Vergleich verlustfreier Datenkompressionsverfahren auf Bilddaten

Nick Schreiber
Technische Hochschule Rosenheim
Master Informatik, Seminar theoretische Informatik
Email: nick.schreiber@stud.th-rosenheim.de

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

I. Einleitung

Datenkompression beschreibt ein Verfahren, das zum Ziel hat, eine Nachricht ohne relevanten Informationsverlust zu verkleinern. Als Nachricht ist jede Art von digitalen Daten gemeint, z.B. Text, Bild, Audio, etc.. Daten können komprimiert werden, indem Redundanz entfernt oder eine Kodierung angewendet wird. Daher wird Datenkompression oft als Kodierung bezeichnet. Kodierung ist ein allgemeiner Begriff, der jede spezielle Darstellung von Daten nach einem bestimmten Schema umfasst. [1]

Es gibt zwei Arten der Datenkompression: die verlustbehaftete und die verlustfreie Kompression. Bei der verlustbehafteten Datenkompression kann eine bestimmte Menge an Information durch die Kompression verloren gehen, was in Kauf genommen wird, da dadurch die Datenmenge erheblich reduziert werden kann oder weil die verlorene Informationen für die Anwendung kaum relevant sind. Das wird auch als Irrelevanzreduktion bezeichnet [2, S. 5]. Ein Beispiel für Irrelevanzreduktion kann bei Audiosignalen beobachtet werden. Der menschliche Hörfrequenzbereich liegt zwischen 20 Hz und 20 kHz [3]. Daher ist es nicht sinnvoll, Frequenzen, die weit außerhalb des hörbaren Bereichs liegen, in Audiodateien zu speichern. Bei der verlustfreien Datenkompression wird die Integrität der Daten bewahrt. Das bedeutet, dass sämtliche Informationen in den komprimierten Daten enthalten sind und die Originaldaten vollständig rekonstruierbar sind. In dieser Arbeit wird nur die verlustfreie Datenkompression untersucht, da Irrelevanzreduktion nicht direkt zum Themengebiet der Datenkompression gehört.

Die Datenkompression von Bildern wird aus verschiedenen Gründen eingesetzt. Speichernutzung: Unkomprimierte Bilddaten können beträchtlich mehr Speicherplatz beanspruchen. Übertragungseffizienz: Bei der Übertragung von Bildern über Netzwerke oder das Internet spielt die Übertragungseffizienz eine entscheidende Rolle. Wenn ein Bild über einen Kanal mit begrenzter Bandbreite gesendet wird, kann es effizienter sein, das Bild zu komprimieren, es zu übertragen und dann beim Empfänger zu dekomprimieren. Dadurch wird die Übertragungszeit verkürzt und das Bild kann schneller bereitgestellt werden. Dies führt zu einer höheren Übertragungsrate und einer reduzierten Bandbreitennutzung.

II. Zielsetzung der Arbeit

Ziel der Arbeit ist es zu untersuchen, ob und warum bestimmte verlustfreie Datenkompressionsverfahren für Bilddaten besser geeignet sind als andere. Dazu werden die theoretischen Aspekte der Kompressionsalgorithmen untersucht. Außerdem wird untersucht, wie Bilddaten aufgebaut sind und welche Besonderheiten in den Bilddaten für die Datenkompression genutzt werden können. Die Arbeit hat einen praktischen Anteil. Verschiedene Algorithmen zur verlustfreien Datenkompression wurden manuell implementiert und an unterschiedlichen Bilddaten getestet. So konnten konkrete Ergebnisse über die Leistungsfähigkeit der Algorithmen gewonnen werden. Die verglichenen Algorithmen sind Run Length Encoding (RLE), Huffman Encoding, Lempel-Ziv 1977 (LZ77), PNG Algorithmus und verschiedene Kombinationen der Algorithmen. Die Ergebnisse werden interpretiert und mit den theoretischen Erwartungswerten verglichen.

III. Grundlagen zur Datenkompression

In der Informatik gehört die Datenkompression zum Teilgebiet der Informationstheorie. Um zu verstehen, wie verlustfreie Datenkompression funktioniert, muss man einige theoretische Grundlagen kennen.

A. Information

Claude Shannon, der Erfinder der Informationstheorie, definiert den Begriff Information als Maß für den Informationsgehlt. Information ist ein Maß der Unsicherheit, das durch das Eintreten eines bestimmten Ereignisses oder das Empfangen einer Nachricht verringert wird. [4] Information ist die Mindestanzahl von Bits, die zur Codierung einer Nachricht verwendet werden müssen. [5] Die grundlegende Idee zu Information besteht darin, dass Informationen umso wertvoller sind, je unerwarteter oder unwahrscheinlicher sie sind.

B. Entropie

Die Quantifizierung des Informationsgehalts erfolgt durch die Entropie (H). Formal drückt die Entropie die durchschnittliche Menge an Bits aus, die benötigt werden, um eine Information zu kodieren. [4] Die Entropie berücksichtigt

die Wahrscheinlichkeiten verschiedener möglicher Ereignisse und erreicht ein Maximum, wenn alle Ereignisse gleich wahrscheinlich sind, was auf maximale Unsicherheit hinweist.

$$H(X) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2(P(x_i)) \tag{1}$$

Formel 1 definiert die Entropie mathematisch. Hierbei steht $H(X)$ für die Entropie der Menge X. $P(x_i)$ steht für die Wahrscheinlichkeit des Auftretens des Ereignisses x_i. Die Summe wird über alle möglichen Ereignisse x_i in X gebildet.

Diese Formel beschreibt die durchschnittliche Menge an Bits, die benötigt werden, um eine Nachricht aus X zu kodieren. Wenn die Entropie hoch ist, ist die Unsicherheit groß, und es werden mehr Bits benötigt, um die Informationen zu repräsentieren. Wenn die Entropie niedrig ist, gibt es weniger Unsicherheit, und somit werden weniger Bits benötigt.

Man kann nun eine direkte Verbindung zwischen Entropie und Kompression herzustellen. Niedrige Entropie bedeutet, dass eine Datenmenge strukturiert ist, bzw. Muster aufweist. Das heißt, dass in den Daten wenig unsicherheit ist und die Daten Redundanz enthalten. Das bedeutet, niedrige Entropie sagt, dass die Daten komprimiert werden können.

C. Redundanz und Mutual Information

Redundanz beschreibt Informationen die in Daten mehrfach vorhanden sind. [6] Einfach gesagt kann man Redundanz als überflüssige Information betrachten. Eine hohe Redundanz sagt aus, dass sich wiederholende oder vorhersehbare Muster innerhalb der Daten befinden.

Um eine Formel für die Redundanz aufzustellen benötigt man die mittlere Codewortlänge. Die mittlere Codewortlänge gibt den durchschnittlichen Bedarf an Bits pro Symbol in einer Nachricht an. Sei X ein Alphabet und $x \in X$. $C(x)$ bezeichnet das zu x gehörende Codewort. $l(x)$ bezeichnet die Länge von $C(x)$. Die mittlere Codewortlänge L(C) einer Nachricht C(x) mit der Wahrscheinlichkeitsverteilung p(x) ist in Formel 2 definiert.

$$L(C) = \sum_{i}^{|X|} p(x_i) \cdot l(x_i) \tag{2}$$

Mit der mittleren Codewortlänge lässt sich nun die Redundanz des Codes, bzw. der Nachricht berechnen. Die Formel 3 definiert die Redundanz einer Nachricht.

$$R_{\text{Code}} = L(C) - H(X) \tag{3}$$

Die Redundanz wird berechnet, indem von der tatsächlichen durchschnittlichen Anzahl an Bits pro Symbol die theoretisch minimale Anzahl an Bits pro Symbol abgezogen werden. Die theoretisch minimale Anzahl an Bits pro Symbol entspricht der enthaltenen Information und ist gleich der Entropie der Nachricht. Daraus ergibt sich, dass die Redundanz ≥ 0 sein muss.

Mutual Information ist ein quantitatives Maß für die gegenseitige Abhängigkeit von zwei Variablen. [4] Es misst, wie sehr die Kenntnis einer Variablen die Unsicherheit über die andere Variable reduziert. Dieses Konzept ist entscheidend, um die Struktur von Daten zu verstehen und voneinander abhängige Informationen zu erkennen.

Wenn die Mutual Information zwischen zwei Variablen hoch ist, bedeutet dies, dass das Wissen über eine Variable bedeutende Informationen über die andere Variable liefert. Hohe Mutual Information sagt dementsprechend aus, dass zwei Variablen stark voneinander Abhängig sind. Das Wissen über den Wert einer Variable trägt bereits wesentlich zur Vorhersage oder zum Verständnis der anderen Variable bei.

Geringe Mutual Informaion sagt aus, dass die beiden Variablen weniger gemeinsame Information teilen. Das Wissen über den Wert einer Variable trägt nicht so stark zur Vorhersage oder zum Verständnis der anderen Variable bei. Das bedeutet eine schwächere Statistische Abhängig der Variablen.

Durch das erkennen von Mutual Information kann gezeigt werden, dass Muster und/ oder Wiederholungen und dementsprechend Redundanz in den Daten enthalten ist. Redundanz spielt im Bezug auf Kompression eine wichtige Rolle. Kompression funktioniert durch das identifizieren und eliminieren redundanter Elemente, um den Informationsgehalt zu maximieren und die Effizienz von Datenrepräsentationen zu steigern.

IV. Informationstheorie

Eine wichtige Punkt in der Informationstheorie ist die Unterscheidung zwischen Daten und Information. Daten und Information werden im normalen Sprachgebrauch häufig als Synonym verwendet, was eigentlich nicht korrekt ist.

Daten sind rohe Fakten oder Symbole, die an sich keine spezifische Bedeutung haben. Informationen entstehen durch die Interpretation, Organisation und Strukturierung von Daten, wodurch ein sinnvoller Kontext geschaffen wird. [7] Daten werden zu Informationen, wenn sie für einen bestimmten Zweck verwendet werden können.

Im Kontext der Datenkompression ist es wichtig zu verstehen, dass nicht alle Daten gleichermaßen informativ sind. Ein effektiver Kompressionsalgorithmen entfernt redundante und nicht informative Teile der Daten. Jedoch bleibt die gesamte Information enthalten. Die Daten sind somit invormativer und komprimierter als zuvor.

A. Quellencodierungstheorem/ Source Coding Theorem

Das Quellencodierungstheorem beschäftigt sich mit der Effizienz der Datenkompression und sagt aus, dass es eine Grenze für die minimale mittlere Codierungslänge gibt, die für die Darstellung von Information aus einer bestimmten Quelle erforderlich ist. [8] Das Quellencodierungstheorem besagt, dass die mittlere Codierungslänge L pro Symbol für eine gegebene Quelle nicht kleiner sein kann als die Entropie H der Quelle. Mathematisch in Formel 4 ausgedrückt.

$$L \geq H \tag{4}$$

Die Entropie stellt dabei die untere Schranke für die mittlere Codierungslänge dar. Das zeigt, dass Datenkompression nicht bis ins unendliche möglich ist ohne Information zu verlieren. Die Maximale Kompression ist genau dann erreicht, wenn $L = H$ entspricht. Es würde also bei solchen Daten keinen sinn machen zu versuchen die Daten zu komprimieren, da das ohne Informationsverlust nicht möglich ist.

Vlt. Todo: Schubfachprinzip, Bedeutung von Entropie in diesem Kontext.

B. Kolmogorov Komplexität

Die Kolmogorov Komplexität ist ein Maß für die Strukturiertheit einer Zeichenkette. Sie entspricht der Länge des kürzesten Programms, das die Zeichenkette erzeugen kann. [9] Die Kolmogorov Komplexität ist dementsprechend ein Maß für die algorithmische Komplexität von Informationen.

Die Kolmogorov Komplexität eines Objekts, z. B. eines Textes, ist die Länge des kürzesten Programms, das das Objekt als Ausgabe erzeugt. Hier ein Beispiel für so ein Programm. Betrachten wir die folgende Zeichenkette: "AAAAAAAAABBBBBCCCCCC". Die Zeichenkette besteht aus 20 Zeichen (9 x A, 5 x B, 6 x C). Mit einem einfachen Programm lässt sich die Zeichenkette deutlich kürzer beschreiben: "9A5B6C". Das Programm gibt die Länge der identischen aufeinanderfolgenden Zeichen an, gefolgt von dem Zeichen. So lässt sich die ursprüngliche Zeichenkette der Länge 20 in nur 6 Zeichen darstellen. Wir haben ein Programm gefunden, das die Länge der Beschreibung der Zeichenkette erheblich reduziert. Die Kolmogorov Komplexität dieser Zeichenkette ist deutlich geringer als die ursprüngliche Länge der Zeichenkette. Das zeigt, dass in der Zeichenkette Strukturen vorhanden sind.

Es ist wichtig zu beachten, dass die tatsächliche Kolmogorov Komplexität für allgemeine Zeichenketten wegen des Halteproblemes nicht praktisch berechenbar ist. [10] Allerdings können Abschätzungen gemacht werden. Wenn ein Algorithmus gefunden wird, der eine Zeichenkette in einem kürzeren Programm darstellt entspricht die Kolmogorov Komplexität der Zeichenkette maximal der Länge des Programms. Es ermöglicht den Informationsgehalt von Daten in Bezug auf die kürzeste mögliche algorithmische Beschreibung zu verstehen.

C. Datenanforderungen, Komprimierbarkeit

Daten können in verschiedene Gruppen eingeteilt werden. Auf der einen Seite stehen strukturierte Daten und auf der anderen Seite unstrukturierte Daten.

Strukturierte Daten sind Daten in denen wiederkehrende oder vorhersagbare Muster enthalten sind. Die Entropie der Daten ist niedrig. Ein Beispiel für strukturierte Daten sind Tabellen.

Unstrukturierte Daten sind Daten, in denen keine wiederkehrende oder vorhersagbare Muster enthalten sind. Die Entropie der Daten ist hoch. Ein Beispiel für unstrukturierte Daten sind zufällig erzeugte Daten.

Strukturierte Daten enthalten meist eine höhere Redundanz als unstrukturierte Daten. Das ist dem Fakt geschuldet, dass die maximal mögliche Kompression, der Information in den Daten, der Entropie, entspricht. Daten können theoretisch maximal auf das Niveau der Entropie der Daten komprimiert werden. Strukturierte Daten mit geringer Entropie haben ein tieferes Limit als unstrukturierte Daten. Deshalb können strukturierte Daten meist mehr komprimiert werden als unstrukturierte.

1) Vorverarbeitung: Eine Möglichkeit aus etwas unstrukturierten Daten strukturiertere zu machen ist über eine Vorverarbeitung der Daten. So eine Vorverarbeitung ist meist eine Normalisierung oder eine Datenfilterung. Wichtig ist, dass der Vorverarbeitungsschritt umkehrbar ist. Eine Art wie Daten für die Kompression Vorverarbeitet werden, wird im Lauf der Arbeit anhand des PNG Algorithmus gezeigt. PNG verwendet eine Datenfilterung.

2) Limitationen: Ein Beispiel für maximal unstrukturierte Daten die scheinbar nicht komprimierbar sind, sind normalverteilte Zufallszahlen. "The Random Compression Challenge" von Mark Nelson [11] betrachtet genau dieses Problem. Das Ziel der Challenge ist es eine Datei, die etwa ein halbes Megabyte groß ist zu komprimieren. Die Datei besteht aus einer Millionen Zufallszahlen, die gleichverteilt sind und aus dem Buch "A Million Random Digits with 100,000 Normal Deviates" [12] kommen.

Es gibt zwei Arten wie die Challenge gewonnen werden kann. Die eine Möglichkeit ist es, die Kolmogorov Komplexität zu verwenden und ein Programm zu schreiben, dass die ursprüngliche Datei erzeugt. Die größe des Programms muss kleiner sein, als die zu komprimierende Datei. Es geht dabei darum zu zeigen, dass die Kolmogorov Komplexität kleiner ist als die Größe der Datei selbst.

Die andere Möglichkeit ist es, ein System zu entwickeln, dass Dateien mit normalverteilten Zufallszahlen komprimieren und vollständig wieder aus der komprimierten Datei dekomprimieren kann. Die größe des Systems spielt dabei keine Rolle, weil das System mehr als nur eine Datei erfolgreich komprimieren und dekomprimieren muss. Es ist bewiesen, dass dieser Ansatz unmöglich ist [11]. Der Grund dafür ist, dass die Entropie der größe der Datei selbst entspricht und ohne Informationsverlust nicht verkleinert werden kann.

Bis heute hat noch niemand die Challenge gewonnen. Das ist nicht verwunderlich, da die zweite Möglichkeit zu gewinnen bewiesenermaßen unmöglich ist. Hingegen ist die erste Möglichkeit mit dem Kolmogorov Komplexität nur vermutlich unmöglich. Das bedeutet es könnte eine Lösung geben, ist aber nach jetzigem Stand unwahrscheinlich.

Die Challenge ist ein perfektes Beispiel dafür, wieso es Sinn macht die Daten zu kennen die man komprimieren will. Sie zeigt die Limitationen der Datenkompression auf.

V. Aufbau und Struktur von Bilddaten

Dieser Abschnitt beschäftigt sich mit der dem Aufbau und der Struktur von Bilddaten. Es gibt verschiedene Möglichkeiten Bilddaten zu speichern und darzustellen. In dieser Arbeit werden die Bilder in Form einer Rastergrafik/ Bitmap gespeichert, dem RGB Format. Eine Rastergrafik ist

eine pixelbasierte Darstellung eines Bildes. Die Auflösung eines Bildes gibt an, wie viele Pixel in der Breite und Höhe vorhanden sind. Ein Pixel ist die kleinste diskrete Einheit in einem digitalen Bild.

Das RGB Format ist eine Möglichkeit, Farbinformationen in digitalen Bildern zu repräsentieren und zu speichern. RGB steht für die Farben Rot (R), Grün (G) und Blau (B). Es ist ein additives Farbmodell und jede Farbe wird durch eine Kombination der drei Grundfarben erzeugt, auch weiß und schwarz. [13]

Ein Beispielbild mit einer Auflösung von 800 Pixel Breite und 800 Pixel Höhe wird in dem RGB Format durch eine Matrix gespeichert. Die zugehörige Matrix hat eine Größe von (800, 800, 3), (Breite, Höhe, Farbkanäle). Die dritte Dimension ist die Anzahl der Farbkanäle. Für jeden Pixel werden drei Werte, die die Farbe des Pixels beschreiben, gespeichert. Der erste Wert ist der Rotanteil, der zweite der Grünanteil und der dritte den Blauanteil. Das RGB Format verwendet normalerweise 8 Bit pro Farbkanal, was eine Farbtiefe von 24 Bit pro Pixel ergibt. Jeder Farbwert eines Pixels hat 8 Bit, bzw. 1 Byte Speicher zur Verfügung und kann Werte zwischen 0 und 255 annehmen.

Bilder im RGB Format sind nach einem bestimmten Schema aufgebaut, weshalb Strukturen in den Bilddaten entstehen. Diese Strukturen können bei der Datenkompression helfen die Bilder zu komprimieren. Eine Struktur im RGB Format ist die Darstellung von Nichtfarben wie schwarz und weiß. Um schwarz, bzw. weiß darzustellen muss ein Pixel für alle drei Farbkanälwerte entweder den Wert 0 oder 255 annehmen. Um reine Farben darzustellen wie rot, wird nur ein Pixelwert für den Rotanteil benötigt, während der Grün- und Blauanteil auf Null gesetzt ist. Ebenso gibt es viele Farbmischungen bei denen eine der Grundfarben nicht benötigt wird und somit deren Anteil im Pixel auf 0 gesetzt wird. Diese Strukturen entstehen durch das Speichern des Bildes im RGB Format. Eine weitere Struktur, die in den meisten Bildern vorhanden ist, ist, dass benachbarte Pixel meist ähnliche oder die gleiche Farbe besitzen. Das hat damit zu tun, dass es in Bildern meistens Regionen gibt, die zu einem Objekt oder Bildteil gehören, die eine homogene Farbe besitzen.

Diese Strukturen und Wiederholungen in Bilddaten können ausgenutzt werden um die Daten zu komprimieren. Bei der Datenkompression ist es entscheidend, welche Algorithmen diese in Bildern spezifisch enthaltenen Redundanzen erkennen und ausnutzen kann.

VI. Messbarkeit der Kompressionsalgorithmen

- Messbarkeit definieren um Ergebnisse der Algorithmen zu vergleichen und zu entscheiden was ist ein guter Kompressionsalgorithmus

Wie ist Kompressionsalgorithmus aufgebaut: Daten die zu komp sind -¿ Kompression (Vorverarbeitung möglicherweise) -¿ Dekompression (Vorverarbeitung der Daten rückgängig machen) -¿ Weil verlustfrei müssen nach komp und dekomp Originaldaten rauskommen

- Kompressionsrate, wie wird berechnet, Formel - Kompressionsgeschwindigkeit, wie wird gemessen, möglicherweise Vorverarbeitungsschritte, die zur Zeitmessung dazugehören - Dekompressionsgeschwindigkeit, wie wird gemessen, aus komp Daten Unkomprimierte Daten

Abwiegen und Gewichten je nach Anwendung. Bsp. echtzeit, nutzlos lange dekomp oder komp aber beste Kompressionsrate Vlt. nur Dekomp entscheidend, Bsp. Inet. Möglicherweise nur 1 interessant Komp oder Dekomp, Bsp.

VII. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page [14] limits.

VIII. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

References

[1] F. M. Ingels, *Information and coding theory*. Scranton, Intext Educational Publishers, 1971.

[2] J. S. Peter Maluck, "Quellencodierung, gelenktes entdeckendes lernen," *EducETH*, 2009.

[3] E. Burke, "Schallquellensystem zur untersuchung der wahrnehmung von infraschall in kombination mit hörschall," *Physikalisch-Technische Bindesanstalt*, 2019.

[4] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948. [Online]. Available: http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

[5] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

[6] B. Friedrichs and B. Friedrichs, "Grundlagen der shannon'schen informationstheorie," *Kanalcodierung: Grundlagen und Anwendungen in modernen Kommunikationssystemen*, pp. 33–68, 1996.

[7] S. Pieper, "Wo liegt der unterschied zwischen daten, informationen und wissen?" https://www.artegic.com/de/blog/wo-liegt-der-unterschied-zwischen-daten-informationen-und-wissen/, 2017, [Online; accessed 01.12.2023].

[8] D. Sharma and S. Saxena, "Generalized coding theorem with different source coding schemes," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 253–257, 2017.

[9] M. Li, P. Vitányi *et al.*, *An introduction to Kolmogorov complexity and its applications*. Springer, 2008, vol. 3.

[10] P. M. Vitányi, "How incomputable is kolmogorov complexity?" *Entropy*, vol. 22, no. 4, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/4/408

[11] M. Nelson, "The random compression challenge," https://marknelson.us/posts/2012/10/09/the-random-compression-challenge-turns-ten.html, 2012, [Online; accessed 03.12.2023].

[12] R. Corporation, *A Million Random Digits with 100,000 Normal Deviates*. Santa Monica, CA: RAND Corporation, 2001.

[13] X.-R. Color, "Additive farbmischung," https://www.xrite.com/de/blog/what-is-additive-rgb-color-mixing, 2022, [Online; accessed 06.12.2023].

[14] D. Autor, "Titel des artikels," *Name des Journals*, pp. 123–145, 2023.