

a) Kodierung & Komprimierung

Mittwoch, 8. Februar 2023 00:49

Information(en)	= Abstrakte Bedeutung von Ausdrücken, Grafiken, ... unabhängig von ihrer Repräsentation/Darstellung = Wissen über die Realwelt (?)
Daten	= Kodierung von Information, die Speicherung und Verarbeitung durch Computer ermöglicht. = eine Form der Repräsentation von Informationen. Wird interpretiert, um Bedeutung zu ermitteln.

- Signale können sein: -> analog (= Kodierung für kontinuierliche Daten)
-> digital (= Kodierung für diskrete Daten)
- Computer können mit digitalen Signalen umgehen
- **Bit** (*binary digit*) (Kleinste „Dateneinheit“): -> Kann zu jedem Zeitpunkt genau einen von zwei Werten annehmen: > AN oder AUS
> 1 oder 0
- Repräsentation umfangreicherer Daten durch Kombination von Bits zu Bitstrings.

Wie viele Bits braucht man, um wie viele verschiedene Zustände darstellen zu können?

- N bits können 2^N Zustände darstellen

- 1 Byte = Bitstring der Länge 8 (kann also jeweils einen von 256 verschiedenen Zuständen speichern)

Kilobyte: 10^3 Byte

Megabyte: 10^6 Byte

Gigabyte: 10^9 Byte

Terabyte: 10^{12} Byte

Petabyte: 10^{15} Byte

KODIERUNG VON ZAHLEN

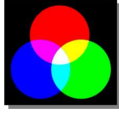


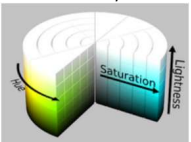
Stellenwertsystem	<ul style="list-style-type: none">• Basis: -Gibt an, wie viele unterschiedliche Zeichen es gibt.• Dezimalsystem: -Basis 10 -$1234 = 1 \cdot 10^3 + 2 \cdot 10_2 + 3 \cdot 10_1 + 4 \cdot 10_0$• Binärsystem: -Basis 2 -$1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9_{10}$
Binärsystem für ganze Zahlen	<ul style="list-style-type: none">• Im Prinzip wie eben beschrieben• Unterschiedliche Varianten zur Darstellung von Vorzeichen: -Betrags-Vorzeichendarstellung (Wert und Vorzeichen werden getrennt abgelegt) -Komplementdarstellung• Maximale Bitstringlänge beschränkt Zahlenbereich: -Häufige mehrere Integerdatentypen in Programmiersprachen -Überlauf kann zu Fehlern führen• Berechnungen liefern exakte Ergebnisse ohne Rundungsfehler
Binärsystem für Reelle Zahlen	<ul style="list-style-type: none">• Wie kann man 1,625 binär codieren? --> $1,101_2 = 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$• Was ist mit 1/3? --> Begrenzte Länge des Bitstrings Genau Darstellung nicht möglich Approximation durch Gleitkommazahlen: Berechnungen liefern ungenaue Ergebnisse!

KODIERUNG VON TEXTEN

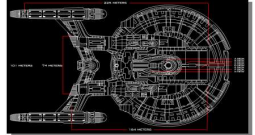
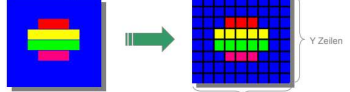
Text	<ul style="list-style-type: none">-Textdateien werden üblicherweise im Computer im ASCII-Code oder Unicode gespeichert.-Im ASCII-Code wird jedem Zeichen einer Nachricht eine 7 Bit-Folge zugewiesen.-Anmerkung: Damit können wir den Text kodieren, aber nicht z.B. sein Layout, die Schriftart,																								
ASCII-Code	<ul style="list-style-type: none">• Problem: -7 Bit ausreichend für 128 Zeichen<ul style="list-style-type: none">-International existieren aber viele Umlaute und Sonderzeichen• Lösung: -ISO 8859-x Standard,<ul style="list-style-type: none">-8-Bit ASCII-Kodierung mit nationalen Erweiterungen (Umlaute)-0-127 identisch mit Standard-ASCII-128-159 seltene Steuerzeichen-160-255 nationale Erweiterungen• Problem: -8 Bit sind ausreichend für 256 Zeichen<ul style="list-style-type: none">-Chinesische, japanische, koreanische oder indische Schriftzeichen lassen sich damit nur schwer repräsentieren																								
Unicode	<ul style="list-style-type: none">-ursprünglich 16-Bit, dann 21 (32)-Bit Kodierung-ermöglicht multilinguale Textverarbeitung-potenziell sind $2.147.483.648 = 2^{31}$ Zeichen möglich-genutzt werden nur 17 Ebenen (planes) mit je 65.536 Zeichen• Grundidee: Jedem potenziellen Zeichen wird ein so genannter Codepoint zugeordnet anstelle einer Glyphie<ul style="list-style-type: none">-> Zeichen (character) = abstrakte Idee eines Buchstabens-> Glyphie = konkrete grafische Darstellung eines Zeichens• Codepoints: -> Identische Zeichen kommen in unterschiedlichen Alphabeten vor<ul style="list-style-type: none">-> Daher können in Unicode einem Zeichen verschiedene Codepoints zugeordnet werden• Unicode Transformation Formate (UTF):<ul style="list-style-type: none">-> Allgemein werden Unicode Codepoints in der folgenden Form dargestellt: U+xxxxxxxx16-> Da aber meist nur Codepoints aus dem BMP benutzt werden, wurden effizientere Kodierungen entwickelt, z.B UTF-8-> UTF-8 kodiert Codepoints mit 1 - 4 Bytes Länge<table border="1"><tr><td>1 Byte</td><td>0xxxxxxx</td><td></td><td></td><td></td><td>(7 Bit)</td></tr><tr><td>2 Byte</td><td>110xxxxx</td><td>10xxxxxx</td><td></td><td></td><td>(11 Bit)</td></tr><tr><td>3 Byte</td><td>1110xxxx</td><td>10xxxxxx</td><td>10xxxxxx</td><td></td><td>(16 Bit)</td></tr><tr><td>4 Byte</td><td>1111xxxx</td><td>10xxxxxx</td><td>10xxxxxx</td><td>10xxxxxx</td><td>(21 Bit)</td></tr></table><ul style="list-style-type: none">-> Wähle für Codepoint stets die kürzeste Kodierungsvariante-> 1 Byte UTF-8 ist kompatibel mit 7-Bit ASCII	1 Byte	0xxxxxxx				(7 Bit)	2 Byte	110xxxxx	10xxxxxx			(11 Bit)	3 Byte	1110xxxx	10xxxxxx	10xxxxxx		(16 Bit)	4 Byte	1111xxxx	10xxxxxx	10xxxxxx	10xxxxxx	(21 Bit)
1 Byte	0xxxxxxx				(7 Bit)																				
2 Byte	110xxxxx	10xxxxxx			(11 Bit)																				
3 Byte	1110xxxx	10xxxxxx	10xxxxxx		(16 Bit)																				
4 Byte	1111xxxx	10xxxxxx	10xxxxxx	10xxxxxx	(21 Bit)																				

KODIERUNG VON FARBEN

Farbe	<ul style="list-style-type: none">-Farben sind die Grundbestandteile des weißen Lichts-Prisma zerlegt weißes Licht in seine spektralen Bestandteile-Lichtstrahlen besitzen keine Farbe sondern eine spektrale Energieverteilung-Menschen können drei Grundfarben wahrnehmen, Rest entsteht durch Mischung-Farben werden aus Farbanteilen der Grundfarben (Rot, Grün, Blau) gemischt und in ein 2-dimensionales Koordinatensystem projiziert
-------	---

RGB-Farbmodell	<p>-additive Farbmischung</p> <p>-Mischung selbstleuchtender Grundfarben (rot, grün, blau)</p> <p>-Farbe wird als Tripel (r,g,b) aus den jeweiligen Farbanteilen angegeben</p> <p>-z.B. bei 8 Bit pro Farbkanal: gelb = (255,255,0)</p> <p>additive Farbmischung</p>  <p>-Zahl darstellbarer Farben hängt von zur Verfügung stehender Bitanzahl ab</p>
CMY(K)-Farbmodell	<p>-cyan, magenta, yellow</p> <p>-subtraktive Farbmischung</p> <p>-Farbe entsteht durch Reflektion/Absorption an unterschiedlichen Oberflächen</p> <p>subtraktive Farbmischung</p>  <p>bestimmte Farbanteile werden reflektiert, andere absorbiert</p>
YUV-Modell	<ul style="list-style-type: none"> • Zerlegung der Farben in: -Helligkeitsanteil (Luminanz) – Y-Komponente -Farbanteil (Chrominanz) – U und V Komponente • Historisch in Verbindung mit dem Farbfernsehen entstanden <ul style="list-style-type: none"> -> Rückwärtskompatibilität mit Schwarzweiß-Empfängern daher separater Helligkeitskanal -> Ausnutzung der unterschiedlichen Empfindlichkeit des menschlichen Auges für Helligkeits- und Farbunterschiede 
HSI- / HSL-Farbmodell	<ul style="list-style-type: none"> • Zerlegung der Farben in: -Farbton (Hue) -Sättigung (Saturation) -Intensität (Intensity) • Modell hinter den meisten Color-Pickern • In der Bildanalyse sehr nützlich: -> Getrennte Farbinformation für bspw. Segmentierung 

KODIERUNG VON BILDERN

Vektorgrafik	<p>-Codierung von Linien, Polygonen und Kurven</p> <p>-Zusätzliche Information wie Farbe, Linienstärke etc.</p> <p>-Ohne Qualitätsverlust beliebig skalierbar</p> <p>-Farbverläufe schwierig</p> <p>-z.B. pdf, svg</p> 
Rastergrafik	<p>-Grafik wird in Matrix aus einzelnen Bildpunkten (Pixel) aufgerastert (Rastergrafik).</p> <p>-Als Pixel bezeichnet man das kleinste, auf einem Computerbildschirm darstellbare Element.</p> <p>-kontinuierliches Bild wird räumlich diskretisiert -> Rasterung</p> <p>-jeder Pixel erhält Farb-/Helligkeitswert -> Quantisierung</p> 

KODIERUNG VON TÖNEN

Analog-Digital-Wandlung	<ol style="list-style-type: none"> 1. Abtastung des Signals (Sampling) <ul style="list-style-type: none"> -> das Signal wird periodisch in bestimmten Zeitabständen t_s abgetastet -> zeitdiskrete, aber wertkontinuierliche Abtastwerte 2. Diskretisierung der Abtastwerte (Quantisierung) <ul style="list-style-type: none"> -> Rundung der kontinuierlichen Abtastwerte auf diskrete Quantisierungspunkte -> zeitdiskrete und wertdiskrete Abtastwerte 3. Kodierung der quantisierten Abtastwerte <ul style="list-style-type: none"> • Problem: -Wie viele Abtastpunkte? (Samplingrate) -Wie viele Quantisierungsintervalle? (Samplingtiefe) • Ziel: -Möglichst exakte Reproduktion des Ursprungssignals bei möglichst geringem Speicheraufwand • Abtasttheorem nach Shannon/Raabe/Nyquist/Kotelnikow <ul style="list-style-type: none"> -> Für jede Größe eines Samplingintervalls Δt gibt es eine bestimmte kritische Frequenz f_s (nyquist critical frequency), die die obere Grenze angibt, bis zu der Frequenzen abgetastet werden können. -> Um eine Schwingung rekonstruieren zu können, werden mehr als zwei Abtastpunkte innerhalb einer Periode benötigt. • Ist vorab die höchste in einem Signal vorkommende Frequenz (f_s) bekannt, kann ein optimales Samplingintervall (Δt) bestimmt werden: $f_s < \frac{1}{2\Delta t}$ <p>Daher folgt für die Samplingrate f_s: $f_s > 2 \cdot f_s$</p>
-------------------------	---

	<ul style="list-style-type: none"> •Bsp.: zu niedrige Samplingrate: Rekonstruktion nicht korrekt möglich, es entstehen hörbare Artefakte •Bsp.: ausreichende Samplingrate: Die Schwingung kann korrekt rekonstruiert werden
Tiefpassfilterung	<p>-in der Praxis müssen daher Frequenzanteile jenseits der kritischen Nyquist-Frequenz (f_a) durch einen Tiefpassfilter entfernt werden, da sonst störende Artefakte auftreten</p> <p>-in der Praxis gibt es aber keinen „idealen“ Tiefpassfilter</p> <ul style="list-style-type: none"> • Ablauf der Digitalisierung <p>kontinuierliches analoges Audiosignal → Sampling → diskontinuierliches analoges Audiosignal → (Analog-Digital-Wandlung) Quantisierung → diskontinuierliches diskretisiertes Audiosignal</p> <p>Samplingrate = #Abtastpunkte pro Zeitintervall Samplingtiefe = #Bits pro abgetastetes Signal</p>
Pulse-Code Modulation (PCM)	<p>= Digitalisierung eines analogen Audiosignals</p> <ul style="list-style-type: none"> • Wie groß sollen die Quantisierungsintervalle gewählt werden? <p>Quantisierungsstufen</p> <ul style="list-style-type: none"> • n Stufen erfordern $k \geq \log_2 n$ Bits zur Kodierung

KOMPRIMIERUNG

Merke	<ul style="list-style-type: none"> -Verarbeitung ist effizienter, wenn sie im Hauptspeicher stattfinden kann •Gründe, Daten zu komprimieren: -Speicherplatz sparen -Speicher- oder effektiven Netzwerkdurchsatz erhöhen -Netzwerkvolumen verringern
-------	---

VERLUSTFREIE KOMPRIMIERUNG

Was ist Information?	<ul style="list-style-type: none"> •Maßgröße für die Ungewissheit des Eintretens von Ereignissen im Sinne der Wahrscheinlichkeitsrechnung = beseitigte Ungewissheit (z.B. durch Auskunft, Aufklärung, Mitteilung, Benachrichtigung über Gegenstände) •Ereignisse = Zeichen (Nachrichtenelemente) •Werden durch Auswahlvorgang aus einem Zeichenvorrat von einer Nachrichtenquelle erzeugt •Durch diese Festlegung wird Information zu einem berechenbaren Maß für die Wahrscheinlichkeit zukünftiger Ereignisse in einem technischen System •Zeichenkette = Folge von Elementen eines Alphabets -> Wirsing - Alphabet = {a,b,c,d,...,A,B,C,D,...} -> 1001001 - Alphabet = {0,1} •Nachricht = übermittelte Zeichenkette, die meist nach bestimmten, vorgegebenen Regeln (Syntax) aufgebaut ist. •durch Verarbeitung erhält die Nachricht Bedeutung (Semantik) •durch die Verarbeitung der Nachricht ändert sich der Zustand des Empfängers der Nachricht (Pragmatik) 								
Wie messe ich Information?	<ul style="list-style-type: none"> •z.B. kürzeste Beschreibung, die eine Nachricht benötigt, welche dieselbe Bedeutung für den Empfänger besitzt, wie die ursprüngliche vorgegebene Information (Beschreibungskomplexität) •Wie viele Bits benötige ich mindestens, um eine Nachricht mit einem bestimmten Informationsgehalt zu kodieren? <p>Alphabet = {a,n,s, <leerzeichen> }</p> <p>Kodierung: Blockcode mit 2 Bit</p> <p>Nachricht: <i>anna an ananas</i></p> <table border="1"> <tr><td>a</td><td>00</td></tr> <tr><td>n</td><td>01</td></tr> <tr><td>s</td><td>10</td></tr> <tr><td><leerzeichen></td><td>11</td></tr> </table> <p>-> Kodierte Nachricht:</p> <p>00 01 01 00 11 00 01 11 00 01 00 01 00 10</p> <p>a n n a a n a n a n a s</p> <p>->Gesamtinformation: 14 x 2 Bit = 28 Bit</p> <p>->Mittlerer Informationsgehalt eines Zeichens: 2 Bit</p> <p>->Tatsächlicher Informationsgehalt einer kompletten Nachricht?</p> <p>-> Kodierte Nachricht:</p> <p>0 1 1 0 10 0 1 10 0 1 0 1 0 0 1</p> <p>a n n a a n a n a n a s</p> <p>->Gesamtinformation: 17 Bit</p> <p>->Aber: Dekodierung ist NICHT eindeutig möglich!</p> <p>->Jede Folge von Bits muss eindeutig dekodierbar sein</p> <p>-> Kodierte Nachricht:</p> <p>1 01 01 1 000 1 01 000 1 01 1 01 1 001</p> <p>a n n a a n a n a n a s</p> <p>->Gesamtinformation: 25 Bit</p> <p>->Code kann auch als Binärbaum dargestellt werden</p> <p>Binärbaumkodierung</p> <p>101011</p> <p>Eindeutige Dekodierung:</p> <ul style="list-style-type: none"> • Starte mit 1. Bit der Folge an der Wurzel des Baums • 0 -> links, 1 -> rechts • Gelangt man an ein Blatt, hat man das Zeichen dekodiert und startet mit dem nächsten Bit wieder an der Wurzel • Gelangt man an einen inneren Knoten, fährt man mit dem nächsten Bit an diesem Knoten fort 	a	00	n	01	s	10	<leerzeichen>	11
a	00								
n	01								
s	10								
<leerzeichen>	11								

Entropie	<p>-ist das Maß für den Informationsgehalt einer Nachricht</p> <p>-Informationsgehalt ist abhängig von Kodierung einer Nachricht</p> <p>-Nach Claude E. Shannon: Entropie H</p> <div>$H(I) = \sum_i^n p_i \log_2 \left(\frac{1}{p_i} \right)$</div> <p>-Nachricht I, besteht aus unterschiedlichen Symbolen {c₁, c₂, ..., c_n}</p> <p>-jedes Symbol c_i (1 ≤ i ≤ n) kommt in Nachricht I mit einer bestimmter Häufigkeit (Wahrscheinlichkeit) p_i vor</p> <p>-Die Entropie H(I) ist der gewichtete Mittelwert der Informationsgehalte aller Zeichen c_i</p> <div><div></div><div>Nachricht: <i>anna an ananas</i> (14 Zeichen)</div></div> <table><tr><th>Zeichen c_i</th><th>a</th><th>n</th><th>s</th><th><leerzeichen></th></tr><tr><td>Häufigkeit</td><td>6</td><td>5</td><td>1</td><td>2</td></tr><tr><td>Relative Häufigkeit p(c_i)</td><td>6/14</td><td>5/14</td><td>1/14</td><td>2/14</td></tr><tr><td>Informationsgehalt log₂ 1/p_i</td><td>1,222</td><td>1,485</td><td>3,807</td><td>2,807</td></tr></table> <div>$\sum_{i=1}^4 p_i \log_2 \left(\frac{1}{p_i} \right) = \frac{6}{14} \cdot 1,222 + \frac{5}{14} \cdot 1,485 + \frac{1}{14} \cdot 3,807 + \frac{2}{14} \cdot 2,807 = 1,727 \text{ bit}$</div> <div><div></div><div>Entropie</div></div> <p>-Informationsgehalt der gesamten Nachricht: Länge x Entropie = 14 Zeichen x 1,727 bit/Zeichen = 24,183 bit</p> <p>-Unsere ursprüngliche Kodierung benötigte 25 Bit</p> <div><div></div><div>Da 24,183 bit = 25 bit (⌈⌋: Zeichen für Aufrundungsfunktion)</div></div> <p>→ unsere Kodierung ist eine optimale Kodierung</p>	Zeichen c _i	a	n	s	<leerzeichen>	Häufigkeit	6	5	1	2	Relative Häufigkeit p(c _i)	6/14	5/14	1/14	2/14	Informationsgehalt log ₂ 1/p _i	1,222	1,485	3,807	2,807
Zeichen c _i	a	n	s	<leerzeichen>																	
Häufigkeit	6	5	1	2																	
Relative Häufigkeit p(c _i)	6/14	5/14	1/14	2/14																	
Informationsgehalt log ₂ 1/p _i	1,222	1,485	3,807	2,807																	
Redundanz	<p>-Anteile einer Nachricht, die keine zur Nachricht beitragende Information enthalten, also aus dieser entfernt werden können, ohne den eigentlichen Informationsgehalt zu verringern</p> <p>-Bsp.: Whnachtsman = Weihnachtsmann (unsere Sprache enthält bereits Redundanz)</p> <p>Funktion:</p> <p>-Information kann selbst bei unvollständiger Übermittlung oder Übertragungsfehlern rekonstruiert werden</p> <p>-Information ist leichter zu lesen/interpretieren</p> <div><div></div><div>Fehlertoleranz und Vereinfachung</div></div> <div><div></div><div>größere Informationsmenge</div></div> <p>-Claude E. Shannon definiert den Informationsgehalt einer Nachricht, die Entropie H</p> <div><div></div><div>→ abhängig von statistischer Natur der Nachrichtenquelle</div></div> <div><div></div><div>→ keine weitere verlustfreie Komprimierung (kleiner als H) möglich!</div></div>																				
Komprimierungsvarianten	<p>-Unter Komprimierung versteht man die Beseitigung oder Verringerung der Redundanz einer Nachricht.</p> <p>-Ziel der Komprimierung ist es, einen möglichst redundanzfreien Code zu erzeugen, aus dem die ursprüngliche Information eindeutig und möglichst ohne Informationsverlust wieder rekonstruiert werden kann</p> <p>-Man kann verschiedene Varianten der Komprimierung unterscheiden:</p> <div><div></div><div>Logische Komprimierung:</div></div> <div><div></div><div>→ fortlaufende Substitution von Symbolen durch andere Symbole</div></div> <div><div></div><div>→ Nutzung der inhärenten Information der Daten</div></div> <div><div></div><div>→ z.B.: „USA“ statt „United States of America“</div></div> <div><div></div><div>Physikalische Komprimierung:</div></div> <div><div></div><div>→ ohne Nutzung inhärenter Information</div></div> <div><div></div><div>→ Austausch einer Kodierung durch eine kompaktere</div></div> <div><div></div><div>→ kann leicht automatisiert werden</div></div> <div><div></div><div>Symmetrische Komprimierung:</div></div> <div><div></div><div>→ Verfahren zur Kodierung und Dekodierung besitzen dieselbe Berechnungskomplexität (d.h. sind gleich schwierig)</div></div> <div><div></div><div>Asymmetrische Komprimierung:</div></div> <div><div></div><div>→ Kodierungs- und Dekodierungsverfahren besitzen unterschiedliche Berechnungskomplexität</div></div> <div><div></div><div>→ In der Regel ist Kodierung komplexer ->ist dann sinnvoll, wenn nur selten auszuführen</div></div> <div><div></div><div>Nicht-adaptive Komprimierung:</div></div> <div><div></div><div>→ Verwendet statisches Wörterbuch mit vorgegebenen Datenmustern (schnell, aufwändiges Wörterbuch)</div></div> <div><div></div><div>Adaptive Komprimierung:</div></div> <div><div></div><div>→ Für den zu komprimierenden Text wird ein eigenes Wörterbuch erstellt (enthält nur Worte aus dem zu komprimierenden Text)</div></div> <div><div></div><div>Semi-adaptive Komprimierung:</div></div> <div><div></div><div>→Mischform aus adaptiver und nicht-adaptiver Komprimierung</div></div> <div><div></div><div>Verlustfreie Komprimierung:</div></div> <div><div></div><div>→ Nach Kodierung und Dekodierung können die ursprünglichen Daten unverändert und ohne Verlust rekonstruiert werden</div></div> <div><div></div><div>Verlustbehaftete Komprimierung:</div></div> <div><div></div><div>→ Beim Komprimieren gehen (unwichtige) Teile der ursprünglichen Information verloren, so dass diese nach dem Dekodieren nicht exakt mit den ursprünglichen Daten übereinstimmt</div></div>																				