Chapter 3
Data Representation

Data and computers – Ch. 3.1 .../continued

Analog and Digital Information

- real world is continuous and infinite
- (data on) computers are finite
- need to obtain a finite representation (approximation) of the world, sufficient for our computational needs and our five senses
- analog data: information represented in a continuous form
- digital data: information represented in a discrete form
- examples:
  - sound: vinyl record vs. Compact Disc (CD)
  - clocks, thermometers, etc.
- information is digitized by breaking it into pieces
- the pieces are numbered
- the binary number system is preferred for representing these number
- why? a binary digit is 0 or 1, which can be represented by high and low state in an electronic signal
- electronic signals fluctuate – digital signal is far more resistant to information loss because of distance between the two states

Binary Representation

- n bits produce 2^n different bit patterns
- pair each different bit pattern with the thing being represented
- question: how many bits do you need to represent m unique things
- actual number of bits is often influenced by other factors, i.e. computer handles 8, 16, or 32 bits at a time

Representing Text – Ch. 3.3

- English language character set: 26 letters (upper and lower case), punctuation, numeric digits, etc.
- determine the minimum number of bits needed and pair them off
- what about other languages?

The ASCII Character Set

- each character is coded as byte (8 bits)
- 7-bit code
- 8th bit is unused (or used for parity check or to indicate "extended" character set)
- 128 codes (= 2^7)
- two general types of codes:
  - 95 are "graphic" codes (visible)
  - 33 are "control" codes (invisible)

Convert Hexadecimal to Binary in the ASCII table

Example: `4A` `J` (row 10, column 3)

`4A` ⇒ `01001010`

- convert to decimal: `01001010` = 64 + 8+ 2 = 74

The Unicode Character Set

- 16-bit standard
- 65,536 possible codes (= 2^{16})
- from the standard: "... contains 38,887 distinct coded characters covering the principal written languages of the World."
- see: www.unicode.org for more information
- google: "unicode" and look at Wikipedia entry

Octal - Character

000 NUL	001 SOH	002 STX	003 ETX	004 EOT	005 ENQ	006 ACK	007 BEL
010 BS	011 HT	012 NL	013 VT	014 NP	015 CR	016 SO	117 SI
020 DLE	021 DC1	022 DC2	023 DC3	024 DC4	025 NAK	026 SYN	027 ETB
030 CAN	031 EM	032 SUB	033 ESC	034 FS	035 GS	036 RS	037 US
040 SP	041 !	042 "	043 #	044 $	045 %	046 &	047 '
050 (	051)	052 *	053 +	054 ,	055 -	056 .	057 /
060 0	061 1	062 2	063 3	064 4	065 5	066 6	067 7
070 8	071 9	072 :	073 ;	074 <	075 =	076 >	077 ?
100 @	101 A	102 B	103 C	104 D	105 E	106 F	107 G
110 H	111 I	112 J	113 K	114 L	115 M	116 N	117 O
120 P	121 Q	122 R	123 S	124 T	125 U	126 V	127 W
130 X	131 Y	132 Z	133 [	134 \	135]	136 ^	137 _
140 `	141 a	142 b	143 c	144 d	145 e	146 f	147 g
150 h	151 i	152 j	153 k	154 l	155 m	156 n	157 o
160 p	161 q	162 r	163 s	164 t	165 u	166 v	167 w
170 x	171 y	172 z	173 {	174 |	175 }	176 ~	177 DEL

Hexadecimal - Character

00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL
08 BS	09 HT	0A NL	0B VT	0C NP	0D CR	0E SO	0F SI
10 DLE	11 DC1	12 DC2	13 DC3	14 DC4	15 NAK	16 SYN	17 ETB
18 CAN	19 EM	1A SUB	1B ESC	1C FS	1D GS	1E RS	1F US
20 SP	21 !	22 "	23 #	24 $	25 %	26 &	27 '
28 (	29)	2A *	2B +	2C ,	2D -	2E .	2F /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7A z	7B {	7C |	7D }	7E ~	7F DEL

Decimal - Character

0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK	7 BEL
8 BS	9 HT	10 NL	11 VT	12 NP	13 CR	14 SO	15 SI
16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB
24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
32 SP	33 !	34 "	35 #	36 $	37 %	38 &	39 '
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL

Text Compression

1. Keyword Encoding

- idea: substitute a frequently used word with a single character
- example: as (^), the (~), and (+), that ($), etc.
- problems:
  - these characters can't be part of the text
  - frequently used words tend to be short, so not much compression
  - word variations not handled: The vs. the

2. Run-Length Encoding

- idea: replace long series of a repeated character with a count of the repetition
- example: replace `AAAAAAA` with `*A7`
- actually: 01000001 01000001 01000001 01000001 01000001 01000001 01000001
  with 00101010 01000001 00000111

2. Huffman Encoding

- idea: generalization of Morse Code (my view)
- Morse Code (dots & dashes) is based on distribution of letters in general English usage
- Huffman Encoding is based on distribution in a given message
- the algorithm:
  - encoding:
    - build frequency table of letter usage in message
    - build the code
    - encode the message
  - decoding:
    - Huffman code has the prefix property
    - prefix property: no code is the front part of another code
    - decoding processes the bit stream until a match is found

Example of Huffman Encoding/Decoding

Table

Huffman Code	Character
00	A
01	E
100	L
110	O
111	R
1010	B
1011	D

Message: DOORBELL

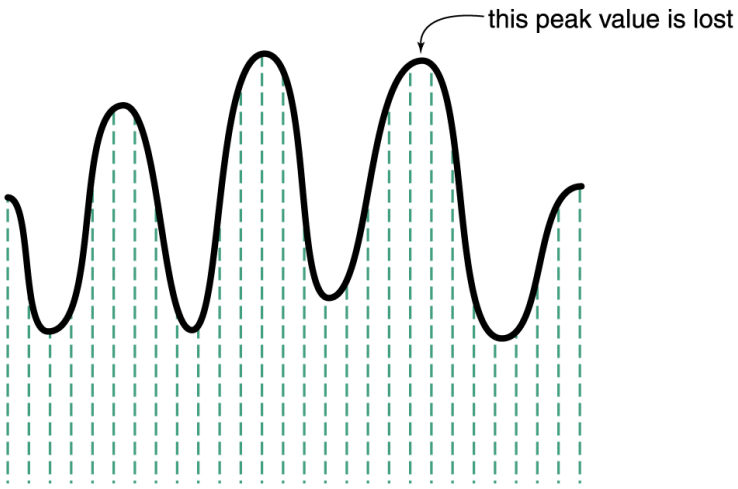
Encoding: 101111011011101001100100

Compression Ratio (vs. ASCII): 25/64 = 0.39

Decode:

Representing Audio Information – Ch. 3.4

- physics of sound
  - speaker to ear of listener through air
  - sound information carried by an electronic wave (e.g. radio)
- electronic wave is analog
- digitize signal by periodically measuring voltage and recording these readings as a series of (binary) numbers
- Nyquist's Theorem: sampling frequency must be 44MHz (44,000 times/sec) for accurate sound reproduction (otherwise: aliasing)

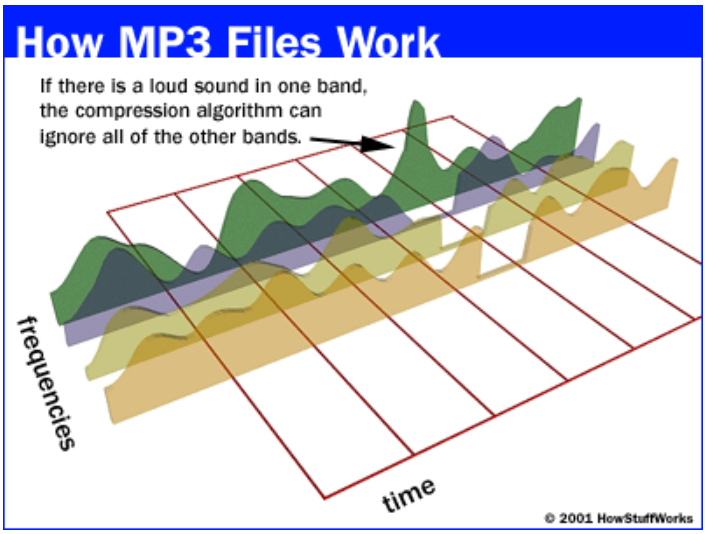


Audio Formats

- Examples: wav, au, aiff, vqf, mp3
- All use some form of compression

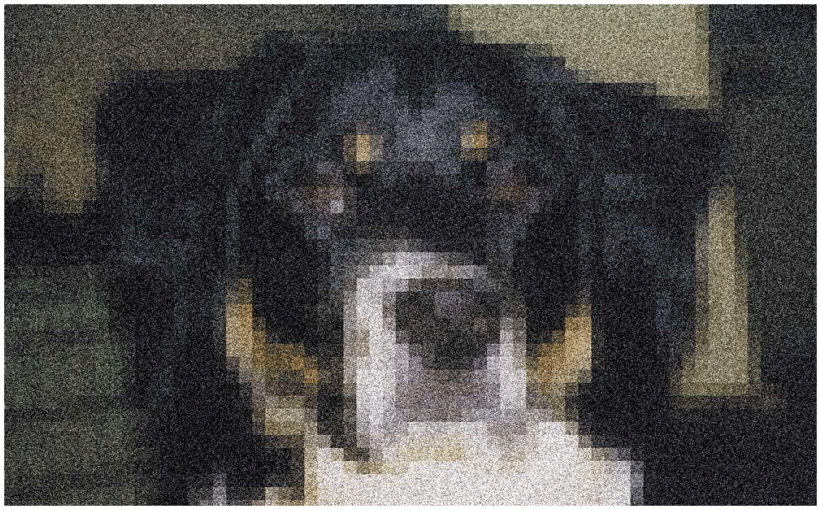
MP3 Audio Format

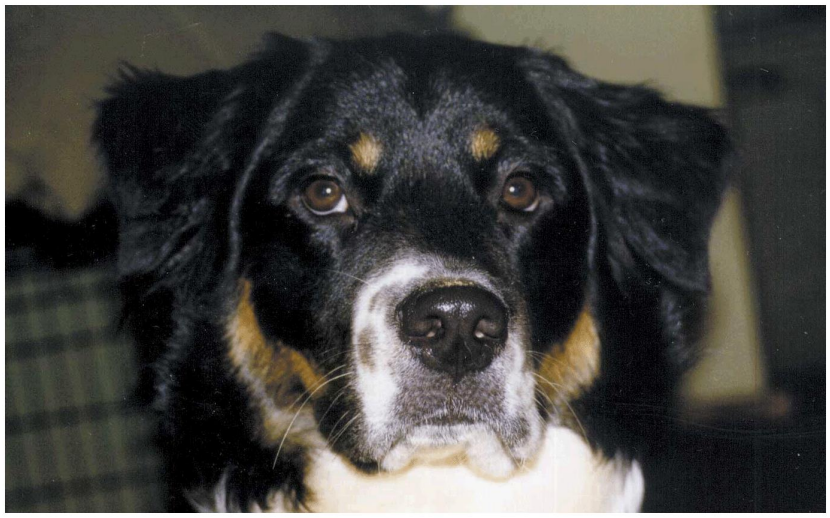
- MPEG-2 audio layer 3
- uses lossy and lossless compression
- lossy: uses mathematical models of human psychoacoustics to discard information the human ear can't hear
- lossless: bit stream compressed by a form of Huffman encoding



Representing Images and Graphics – Ch. 3.5

- raster: image divided into a two-dimensional grid of rectangles
- pixel: each rectangle holds picture information (colour, etc.)
- compression: takes up lots of space (jpeg)
- vector graphics: use mathematical equations to represent shapes
- good for line art and cartoon drawings





Representing Video – Ch. 3.6

- video (film) is a stream of images/frames (at 24 or 30 fps)
- temporal compression: keyframe + series of delta frames that record only changes from keyframe (good if image changes little)
- spatial compression: removes redundant info within a frame (essentially jpeg like compression on each frame)