Panel 1

Storing and Retrieving Data

- Integers
- Letters
- Decimals
- Pictures
- Sounds
- Movies

save/retrieve as sequences of 0's and 1's

How?

1

Panel 2

Integers:

Base-2 format

Base 10. $1234 = 4\cdot10^0 + 3\cdot10^1 + 2\cdot10^2 + 1\cdot10^3$

Base-2: $\# \cdot 2^0 + \# 2^1 + \# 2^2 + \# 2^3 + \# 2^4 + \dots$ (# = 0 or 1)

Ex: $110101_{(2)} = 1\cdot1 + 0\cdot2 + 1\cdot4 + 0\cdot8 + 1\cdot16 + 1\cdot32 = 53_{(10)}$

$1+2+4+8+$

$1+2+2^2+2^3+\dots+2^9 = S$ $S = 2^{10}-1 = 1023$

$2 \quad 2 \qquad\qquad 2^{10} \quad 2S$

2

Panel 3

To convert between base-2 and base-10:

Ex: Convert 1010100 to base-10

64	32	16	8	4	2	1	
1	0	1	0	1	0	0	= 84

~~Convert~~ 301 to base-2:

256	128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0	1

45 13 5

3

Panel 4

Letters

American Standard Code for Information Interchange
ASCII codes

A - 65	a = 97	!	1 = 49
B - 66	b = 98	?	2 = 50
C - 67	c = 99	,	3
⋮		.	4
		#	5

ISO different codes.

4

Panel 5

Ex: 01100110 | 00110011 | 00010110 | 11100011

bit = 0 or 1

8 bits = 1 byte

1 letter = 1 byte

How many different letters are possible?

$11111111 = 2^8 - 1 = 255$

ISO code uses 2 bytes per letter ⇒ $2^{16} - 1$

≈ 64K symbols

Integer uses 4 bytes ⇒ 4294967296−1 in length

5

Panel 6

Assume:

integers use 4 bytes

chars use 1 byte

decimals: 8 bytes

clever way to save exp. and decimals

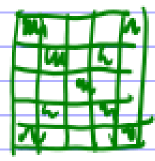
$123.56 \quad 1.2356 \cdot 10^2$

6

Panel 7

Pictures

Step 1: BW only   n×m grid   0=white, 1=black



4 bytes 4 bytes
0...0101 0...0101 1000101010001000101010001
↑ rows   cols

BMP Format

7

Panel 8

Step 2: Grey Scale

4 bytes = rows n
4 bytes = cols m
n×m bytes, i.e. n×m pixels,
each can have 255 shades of grey

0...0101 0...0101 11111111 00000000 00001111

Step 3: Full Color

4 bytes = rows n
4 bytes = cols m
24 bit color
n×m of 3 bytes
RGB

8

Panel 9

Ex: 8 MP camera. File size for

BW: 8000000/8 bytes ≈ 1 MB

Gray: 8000000 bytes ~ 8 MB

Full Color: 8000000 · 3 = 24 MB

8 MP: 3456 × 2306 = 7900000 ≈ 8 MP

9

Panel 10

Compression: use redundancy of data to reduce file size.

Ex: 3000 × 2000 pixels, all black

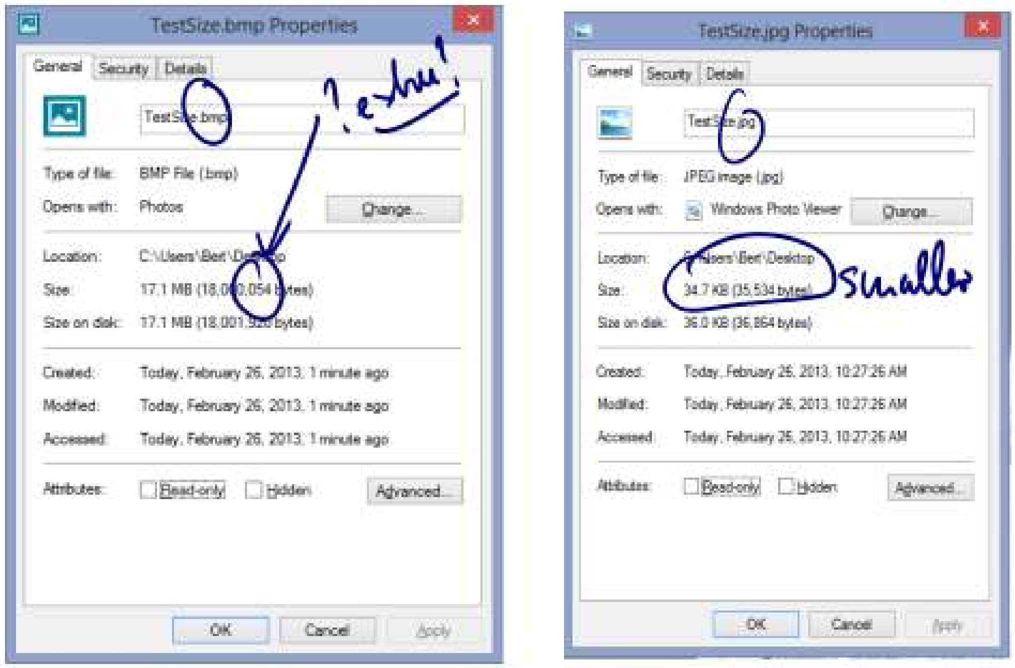
⇒ BMP = 18 MB

JPG: 3000, 2000, 3000×2000, 3

Down side: takes time and CPU power!

10

Panel 11



11

Panel 12

Movie:

color ↓

HDTV: 1920 × 1080 × 3 × 24 × 60 × 60 bytes for 1 hour movie

24 frames per second

537,477,120,000 bytes = 537 GB but only 4.6 GB per DVD

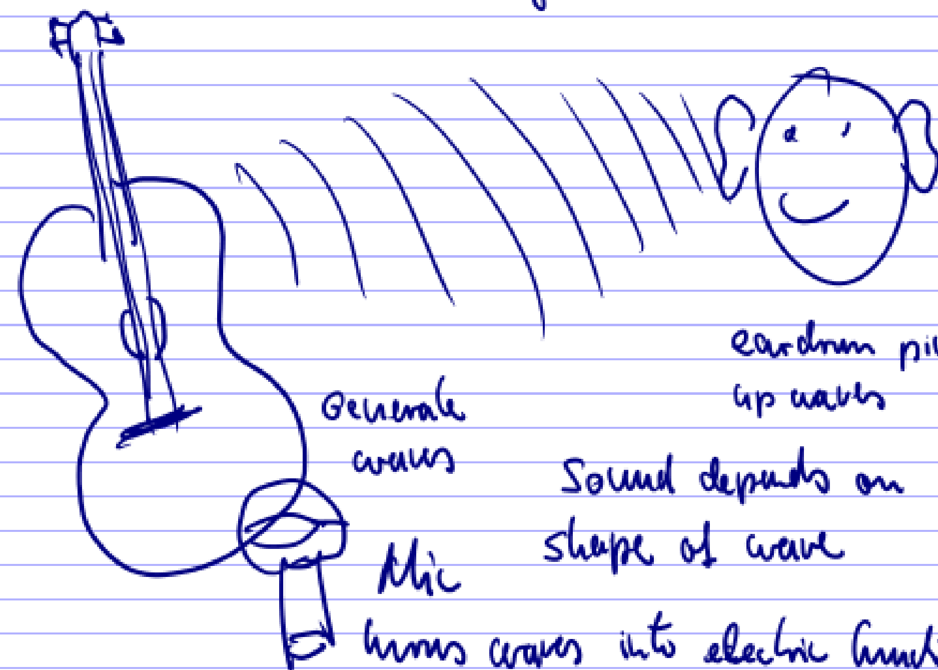
Need: compression optimized for time to decompress pictures (frames) fast enough

→ Audio?

12

Panel 13

Audio: How does hearing work, simplified



eardrum picks up waves

Generate waves

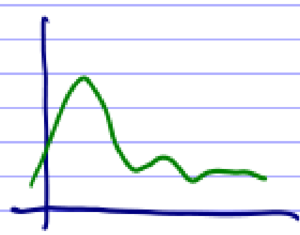
Sound depends on shape of wave

Mic turns waves into electric function!

13

Panel 14

Task: Convert function into sequence of numbers!



Idea I: Taylor series:

$$f(x) = \sum_{n=0}^{\infty} a_n x^n$$

save the a_n:

$1, 3, 5, -1, 2$: $1 + 3x + 5x^2 - x^3 + 2x^4$

Taylor series require very nice (real analytic) functions

Sound waves are not so nice, and they are "wavy"

14