Foundations of Computer Science
Binary World

ONE LOVE. ONE FUTURE.

HUST

hust.edu.vn fb.com/dhbkhn

FaMI 1956

Contents

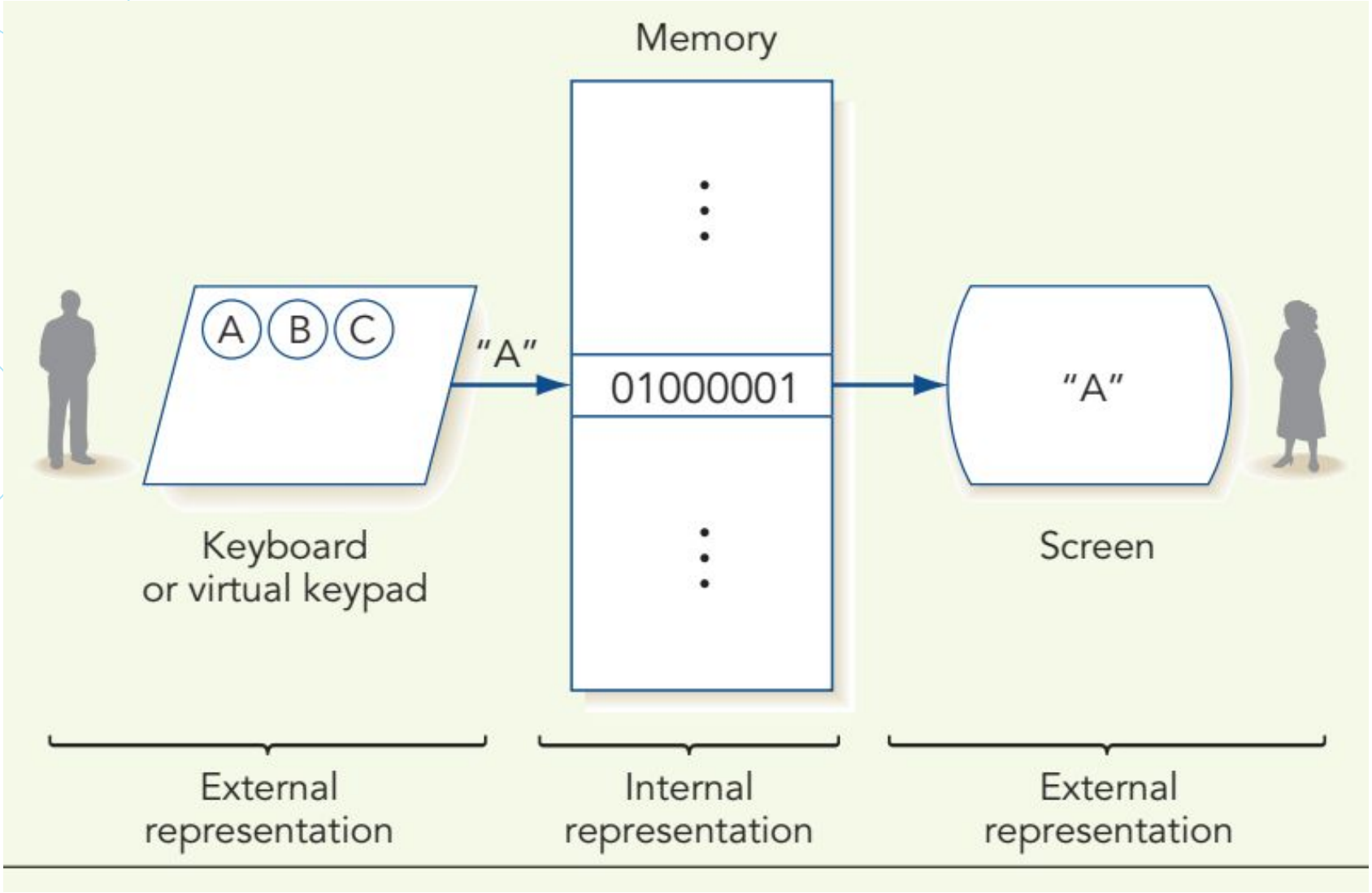
- Binary Number
- Text
- Sound
- Image
- Why Binary?
- Binary Storage
- Boolean Logic
- Logic Gates
- Computer Circuit

Decimal Numbers

- 10 decimal digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, e.g. 459
- *Sign/magnitude notation*: + or – e.g. +31 and – 789 are examples
- *Decimal notation: e.g.* 12.34

$$(2 \times 10^3) + (3 \times 10^2) + (5 \times 10^1) + (9 \times 10^0)$$
$$= 2{,}000 + 300 + 50 + 9$$
$$= 2{,}359$$

Representations



Binary Numbers

$$111001 = (1 \times 25) + (1 \times 24) + (1 \times 23) + (0 \times 22) + (0 \times 21) + (1 \times 20)$$
$$= 32 + 16 + 8 + 0 + 0 + 1$$
$$= 57$$

$$10111 = (1 \times 24) + (0 \times 23) + (1 \times 22) + (1 \times 21) + (1 \times 20)$$
$$= 16 + 0 + 4 + 2 + 1$$
$$= 23$$

Convert the value 19 to binary:

19 ÷ 2	quotient = 9	remainder = 1
9 ÷ 2	quotient = 4	remainder = 1
4 ÷ 2	quotient = 2	remainder = 0
2 ÷ 2	quotient = 1	remainder = 0
1 ÷ 2	quotient = 0	remainder = 1

Stop, because the quotient is now 0.

order for reading the remainder digits

Binary	Decimal	Binary	Decimal
0	0	10000	16
1	1	10001	17
10	2	10010	18
11	3	10011	19
100	4	10100	20
101	5	10101	21
110	6	10110	22
111	7	10111	23
1000	8	11000	24
1001	9	11001	25
1010	10	11010	26
1011	11	11011	27
1100	12	11100	28
1101	13	11101	29
1110	14	11110	30
1111	15	11111	31

Q: Binary-to-decimal Algorithm?

Q: Decimal-to-binary Algorithm?

Binary Numbers

$$111001 = (1 \times 25) + (1 \times 24) + (1 \times 23) + (0 \times 22) + (0 \times 21) + (1 \times 20)$$
$$= 32 + 16 + 8 + 0 + 0 + 1$$
$$= 57$$

$$10111 = (1 \times 24) + (0 \times 23) + (1 \times 22) + (1 \times 21) + (1 \times 20)$$
$$= 16 + 0 + 4 + 2 + 1$$
$$= 23$$

Convert the value 19 to binary:

19 ÷ 2	quotient = 9	remainder = 1
9 ÷ 2	quotient = 4	remainder = 1
4 ÷ 2	quotient = 2	remainder = 0
2 ÷ 2	quotient = 1	remainder = 0
1 ÷ 2	quotient = 0	remainder = 1

Stop, because the quotient is now 0.

order for reading the remainder digits

Binary	Decimal	Binary	Decimal
0	0	10000	16
1	1	10001	17
10	2	10010	18
11	3	10011	19
100	4	10100	20
101	5	10101	21
110	6	10110	22
111	7	10111	23
1000	8	11000	24
1001	9	11001	25
1010	10	11010	26
1011	11	11011	27
1100	12	11100	28
1101	13	11101	29
1110	14	11110	30
1111	15	11111	31

Q: Binary-to-decimal Algorithm?

Q: Decimal-to-binary Algorithm?

Q: What about 65535?

Arithmetic Overflow

Adding

```
  00111
+ 01110
```

```
    0
  00111
+ 01110
      1
```

```
    1
  00111
+ 01110
     01
```

```
    1
  00111
+ 01110
    101
```

```
  00111
+ 01110
  10101
```

Signed Number

$$\underbrace{1}_{-}\ \underbrace{1\,1\,0\,0\,0\,1}_{49} \qquad (2^5 + 2^4 + 2^0 = 32 + 16 + 1 = 49)$$

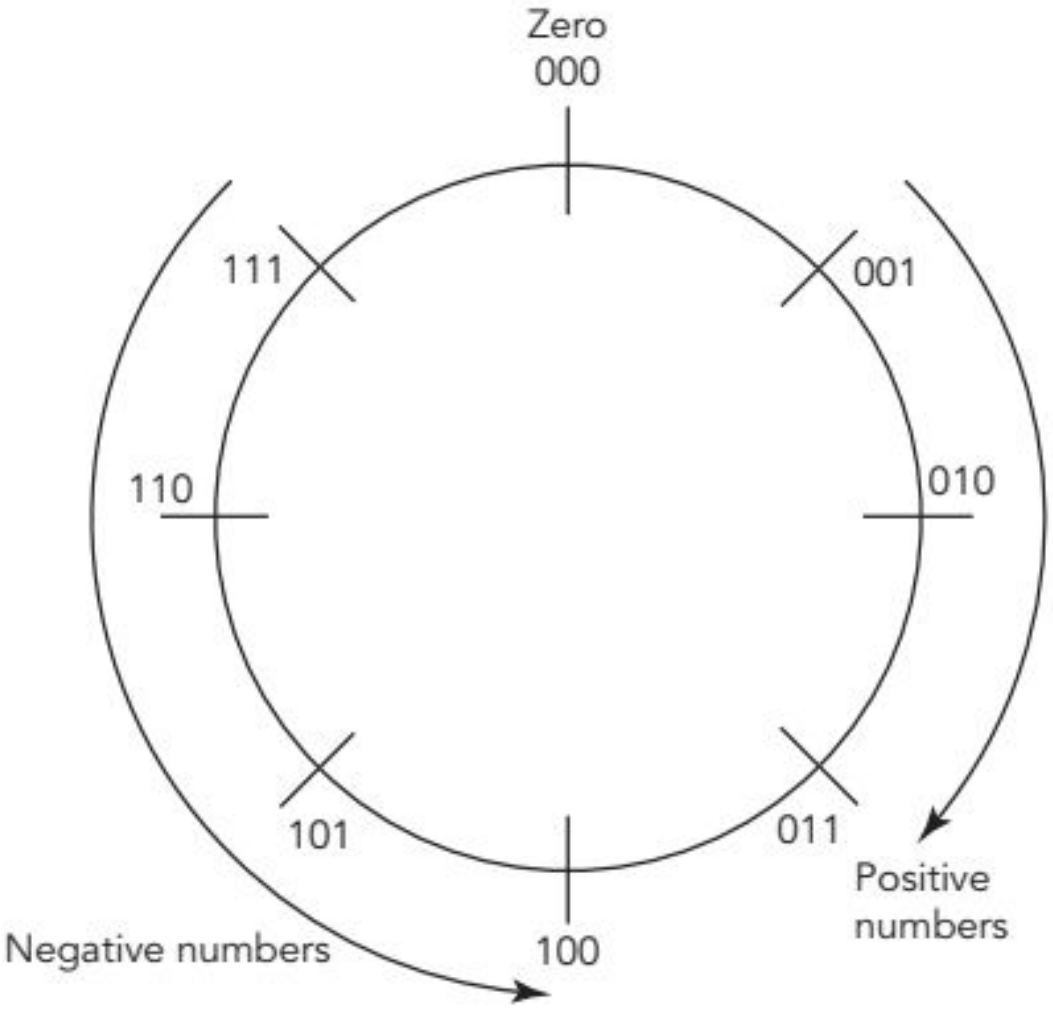
$$\underbrace{0}_{+}\ \underbrace{0\,0\,0\,0\,1\,1}_{3} \qquad (2^1 + 2^0 = 2 + 1 = 3)$$

What about?

$$\begin{aligned} 1110001 &= (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^0) \\ &= 64 + 32 + 16 + 1 \\ &= 113 \end{aligned}$$

What about? 100…00 and 000…000?

Complement Circle



Bit Pattern	Decimal Value
000	0
001	+1
010	+2
011	+3
100	−4
101	−3
110	−2
111	−1

```
  11
 011
 101
-000
```

Fractional number

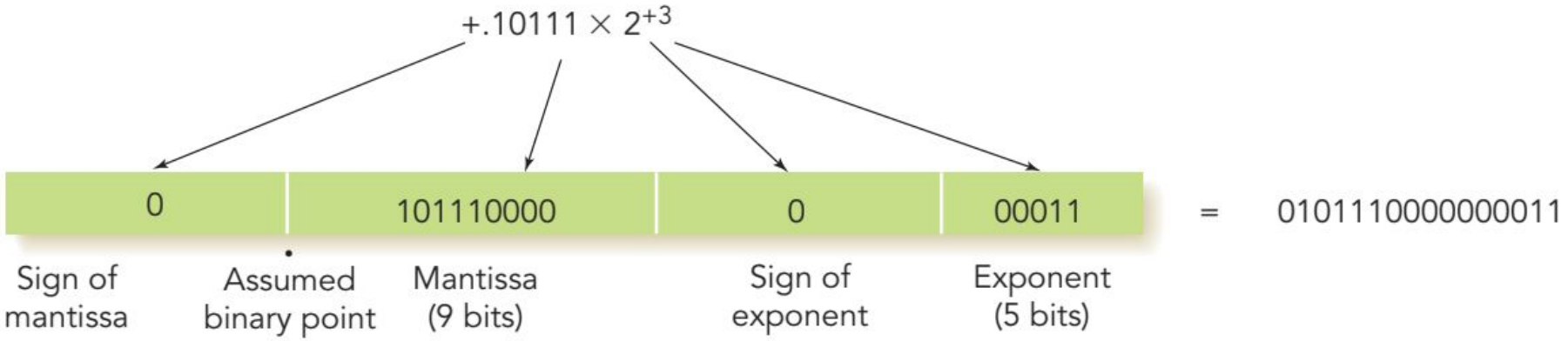
$$\pm M \times B^{\pm E}$$

Mantissa Exponent Base

- Present 5.75

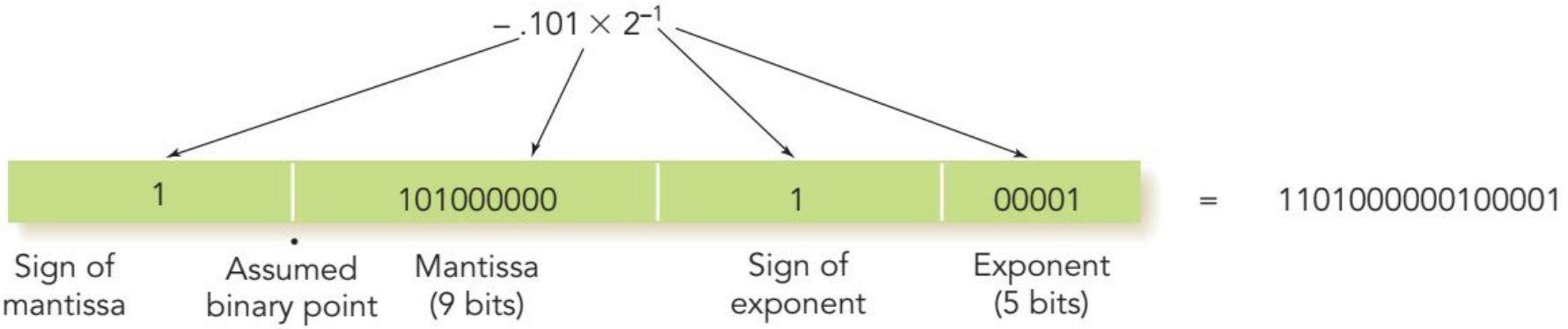
$$0.75 = 1/2 + 1/4 = 2^{-1} + 2^{-2}$$

$$\begin{aligned} 5.75 &= 101.11 \times 2^0 \\ &= 10.111 \times 2^1 \\ &= 1.0111 \times 2^2 \\ &= .10111 \times 2^3 \end{aligned}$$



Example

- Present – 0.3125



Text Symbols

Symbol	*Decimal Value*	*Binary (Using 8 Binary Digits)*
A	1	00000001
B	2	00000010
C	3	00000011
D	4	00000100
⋮	⋮	⋮
Z	26	00011010
⋮	⋮	⋮
@	128	10000000
!	129	10000001
⋮	⋮	⋮

BAD! = 00000010 00000001 00000100 10000001

B A D !

ASCII Conversion Table

- ASCII: American Standard Code for Information Interchange
- Unicode-16 bits
- Unicode-32 bits

Keyboard Character	Binary ASCII Code	Integer Equivalent	Keyboard Character	Binary ASCII Code	Integer Equivalent
(blank)	00100000	32	P	01010000	80
!	00100001	33	Q	01010001	81
"	00100010	34	R	01010010	82
#	00100011	35	S	01010011	83
$	00100100	36	T	01010100	84
%	00100101	37	U	01010101	85
&	00100110	38	V	01010110	86
'	00100111	39	W	01010111	87
(	00101000	40	X	01011000	88
)	00101001	41	Y	01011001	89
*	00101010	42	Z	01011010	90
+	00101011	43	[	01011011	91
,	00101100	44	\	01011100	92
-	00101101	45	]	01011101	93
.	00101110	46	^	01011110	94
/	00101111	47	_	01011111	95
0	00110000	48	`	01100000	96
1	00110001	49	a	01100001	97
2	00110010	50	b	01100010	98
3	00110011	51	c	01100011	99
4	00110100	52	d	01100100	100
5	00110101	53	e	01100101	101
6	00110110	54	f	01100110	102
7	00110111	55	g	01100111	103
8	00111000	56	h	01101000	104
9	00111001	57	i	01101001	105
:	00111010	58	j	01101010	106
;	00111011	59	k	01101011	107
<	00111100	60	l	01101100	108
=	00111101	61	m	01101101	109
>	00111110	62	n	01101110	110
?	00111111	63	o	01101111	111
@	01000000	64	p	01110000	112
A	01000001	65	q	01110001	113
B	01000010	66	r	01110010	114
C	01000011	67	s	01110011	115
D	01000100	68	t	01110100	116
E	01000101	69	u	01110101	117
F	01000110	70	v	01110110	118
G	01000111	71	w	01110111	119
H	01001000	72	x	01111000	120
I	01001001	73	y	01111001	121
J	01001010	74	z	01111010	122
K	01001011	75	{	01111011	123
L	01001100	76	:	01111100	124
M	01001101	77	}	01111101	125
N	01001110	78	~	01111110	126
O	01001111	79			

What is Sound ?

Acoustics is the study of sound.

- Physical - sound as a disturbance in the air
- Psychophysical - sound as perceived by the ear
- Sound as stimulus (physical event) & sound as a sensation.
- Pressures changes (in band from 20 Hz to 20 kHz)

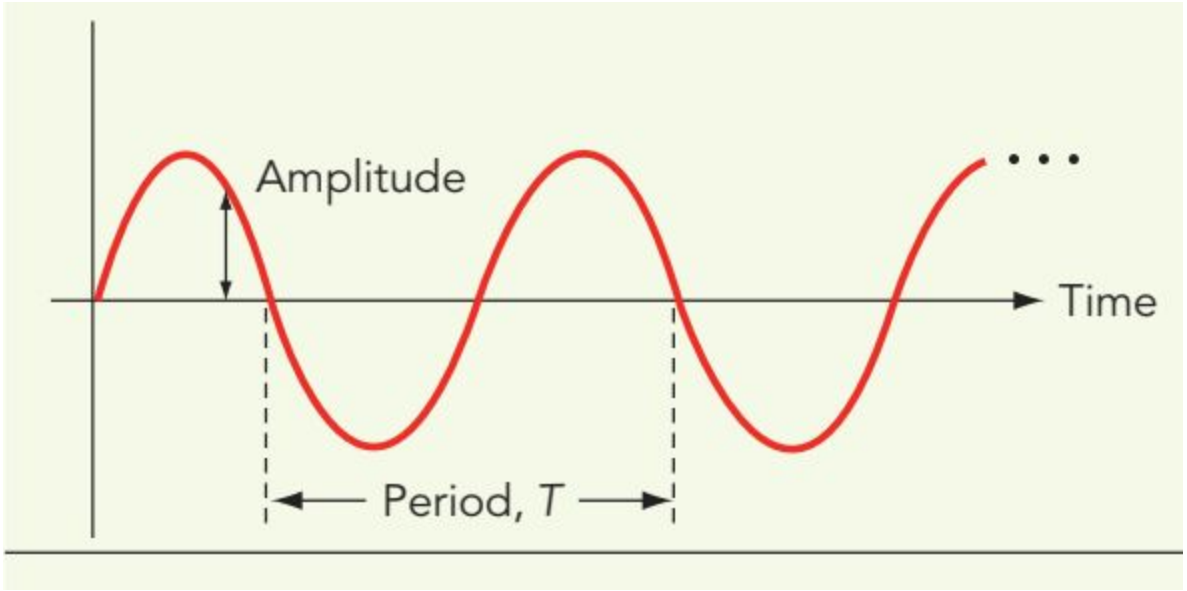
Physical terms

- Amplitude
- Frequency
- Spectrum

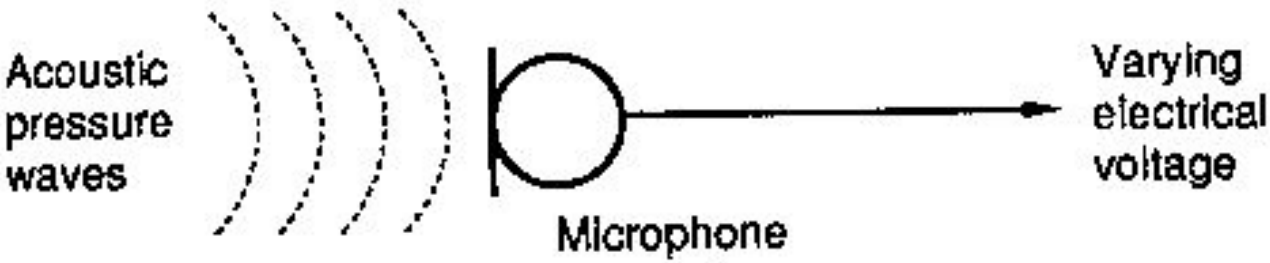
Sound Waves

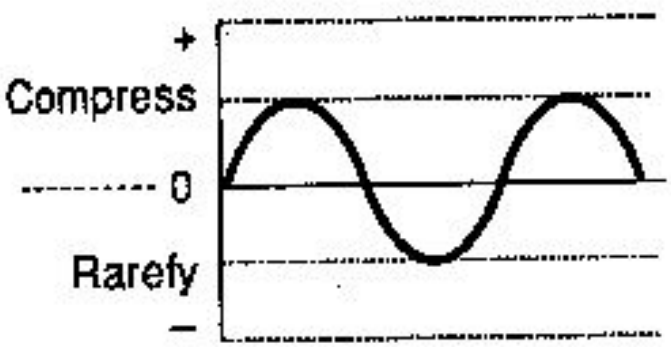
- In a free field, an ideal source of acoustical energy sends out sound of uniform intensity in all directions.
  => Sound is propagating as a spherical wave.
- Intensity of sound is inversely proportional to the square of the distance (Inverse distance law).
- 6 dB decrease of sound pressure level per doubling the distance.

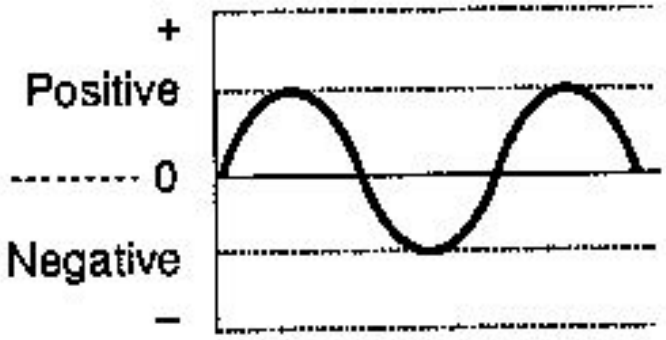
Sound Waves



What is Sound



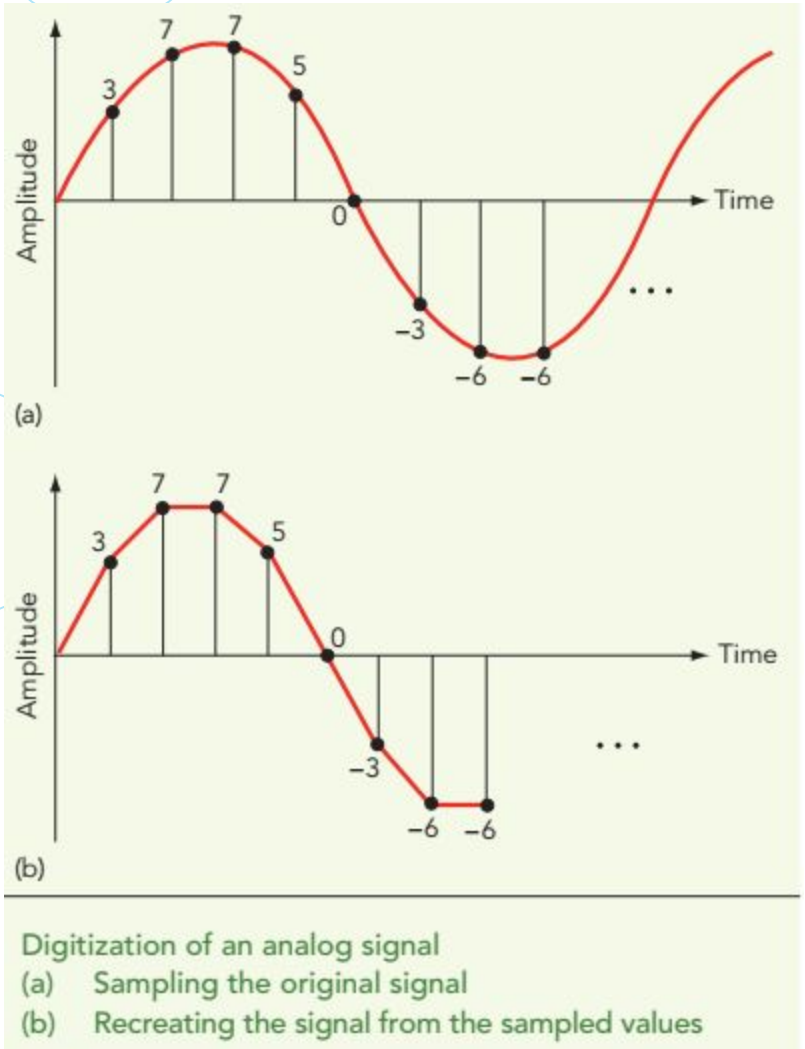




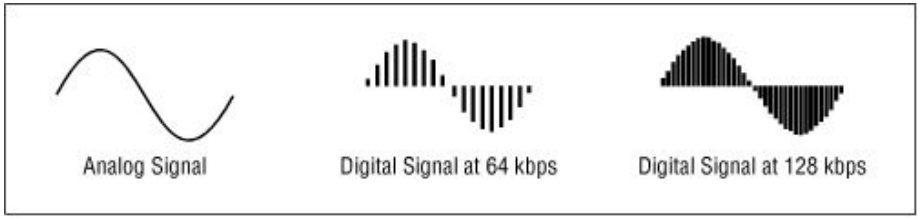
How we hear

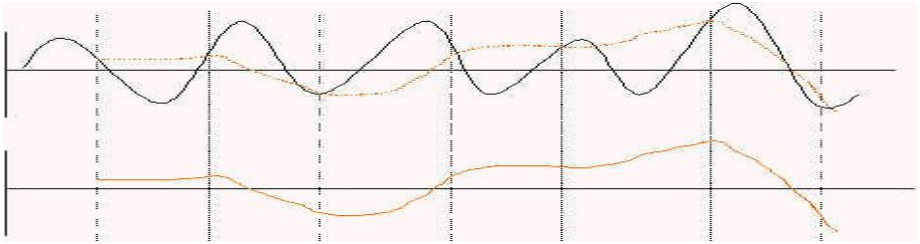
- Ear connected to the brain
  - left brain: speech
  - right brain: music
- Ear's sensitivity to frequency is logarithmic
- Varying frequency response
- Dynamic range is about 120 dB (at 3-4 kHz)
- Frequency discrimination 2 Hz (at 1 kHz)
- Intensity change of 1 dB can be detected.

Digitally Sampling



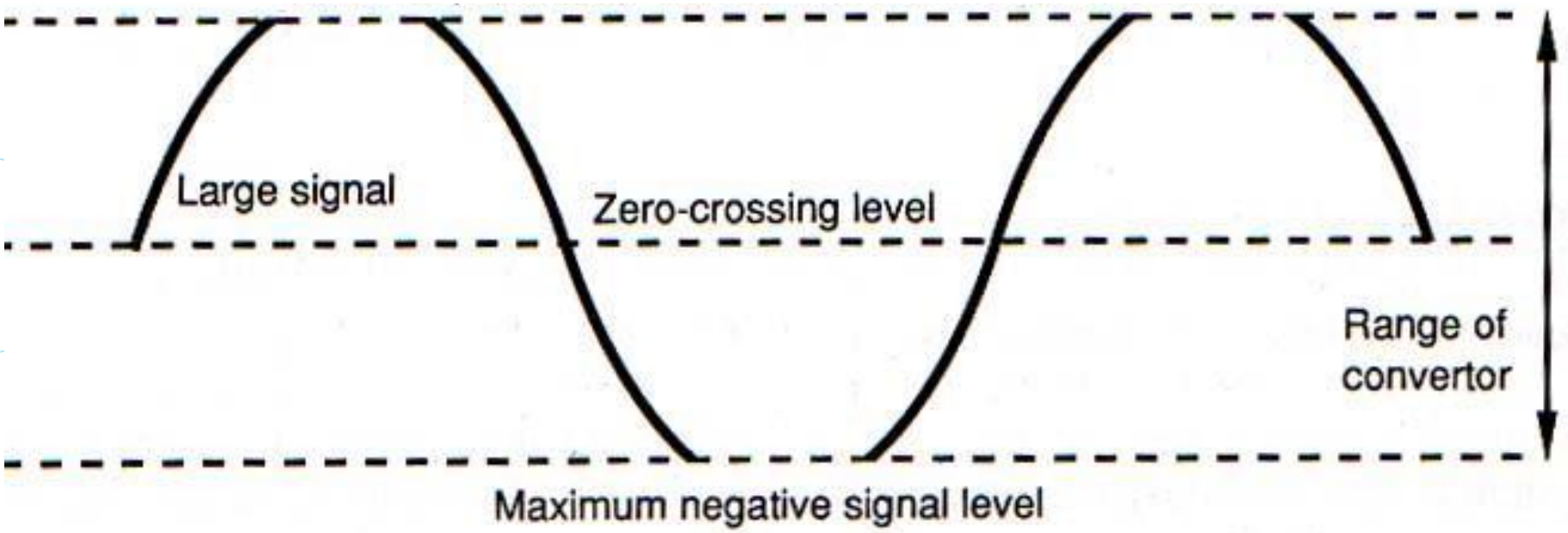
Digitization of an analog signal
(a) Sampling the original signal
(b) Recreating the signal from the sampled values





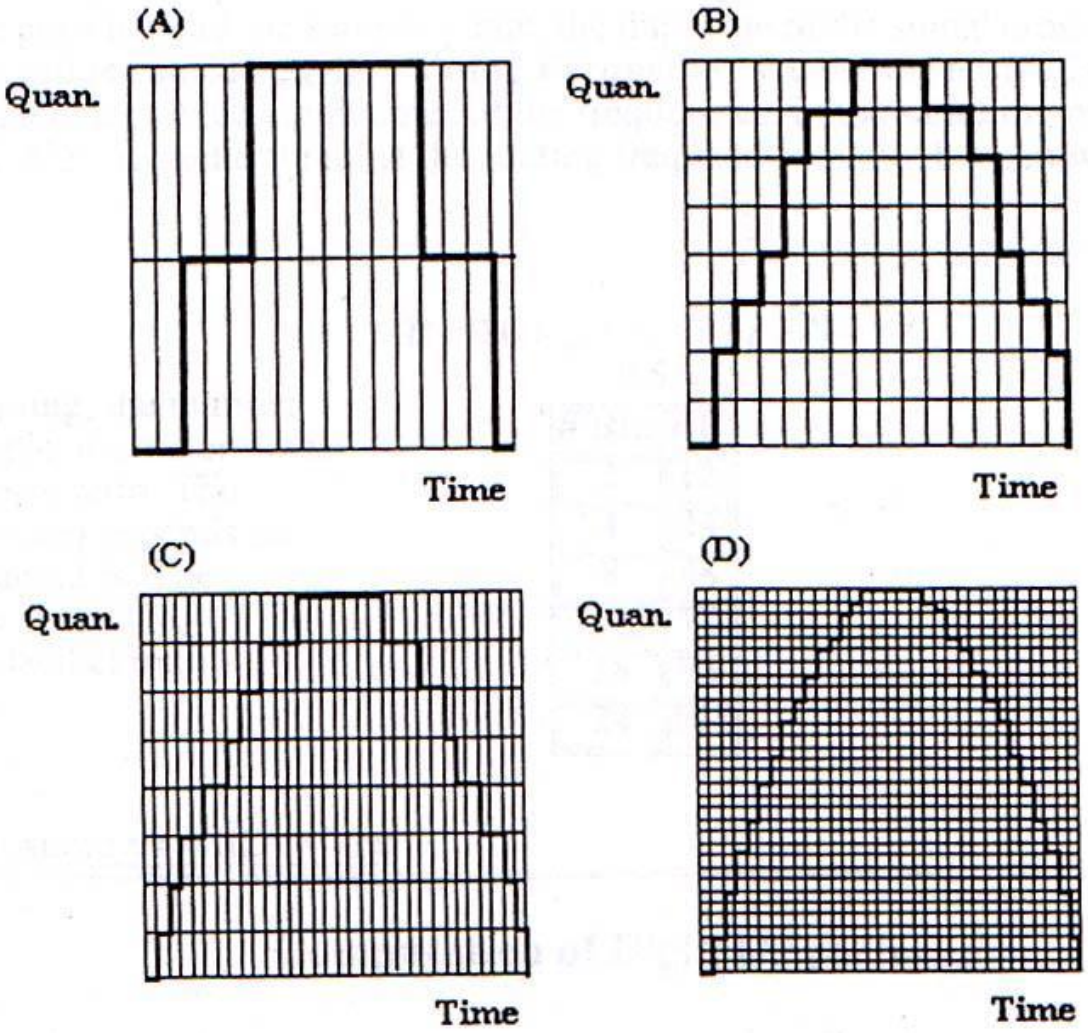
Sampling Rate

Clipping


Large signal
Zero-crossing level
Range of convertor
Maximum negative signal level

Quantization

Bit Depth

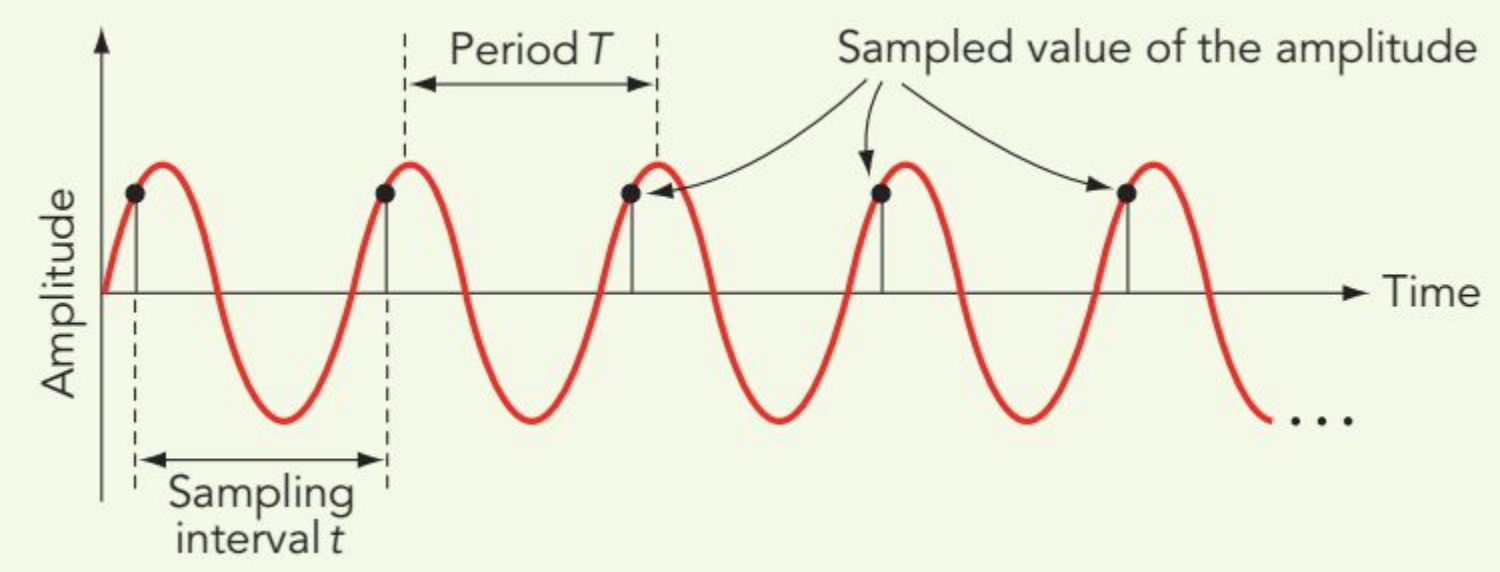


Nyquist's Theorem

- *Sampling* is dictated by the Nyquist sampling theorem which states how quickly samples must be taken to ensure an accurate representation of the analog signal.

$$f_s \geq 2f \quad \text{or} \quad T_s \leq \frac{T}{2}$$

- The Nyquist sampling theorem states that the sampling frequency must be **two** times greater than the highest frequency in the original analog signal.



Audio Encoding Format

- AAC (Advanced Audio Coding): Apple's iPhone and iPad,
- WMA (Windows Media Audio)
- WAV (Waveform Audio File Format)
- MIDI (Musical Instrument Digital Interface)
- MP3: MPEG-1 (updated to MPEG-2), Audio Level 3 Encoding
  - Motion Picture Experts Group (MPEG), a committee of the International Organization for Standardization (ISO)
  - Sampling rate of 44,100 samples/second
  - Bit depth: 16 bits per sample.
  - High-quality sound reproduction,
  - Used format for rock, opera, and classical music

Common Sound Sampling Parameters

- Common Sampling Rates
  - 8KHz (Phone) or 8.012820513kHz (Phone, NeXT)
  - 11.025kHz (1/4 CD std)
  - 16kHz (G.722 std)
  - 22.05kHz (1/2 CD std)
  - 44.1kHz (CD, DAT)
  - 48kHz (DAT)
- Bits per Sample
  - 8 or 16
- Number of Channels
  - mono/stereo/quad/ etc.

Audio Data Rates

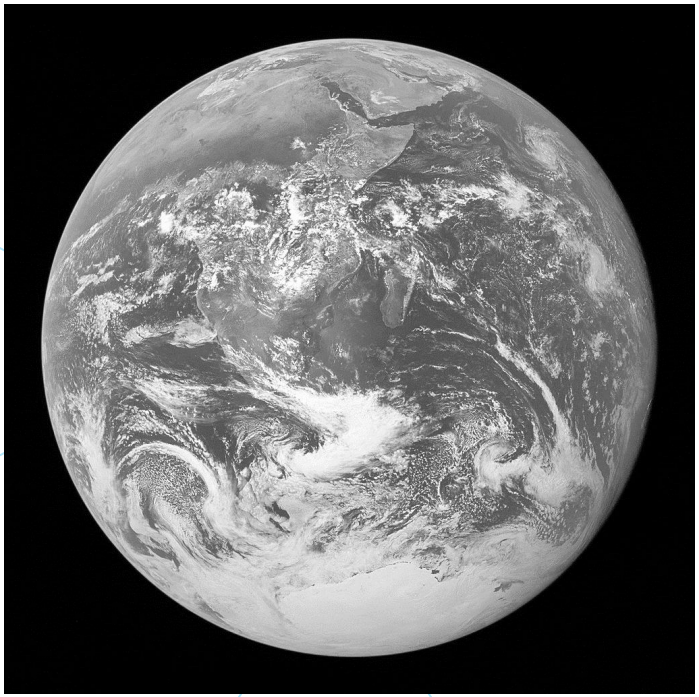
Quality	Format (examples)	Transfer Rate	Disk Space 1 hour	Disk Space 100,000 hours
Netcasting	RealAudio	20 Kbit/s	8.8 MByte	0.9 TByte
Preview	RealAudio	80 Kbit/s	35.2 MByte	3.5 TByte
Preview	MPEG Layer 3 (MP3)	192 Kbit/s	84.4 MByte	8.4 TByte
Broadcasting or Editing	MPEG Layer 2	384 Kbit/s	168.8 MByte	16.9 TByte
Archive (uncompressed)	Waveform PCM	1538 Kbit/s	675.9 MByte	67.6 TByte

Space/Storage Requirements

1 Minute of Sound

Type	Mono	Mono	Stereo	Stereo
Resolution	8 bit	16 bit	8 bit	16 bit
Sampling Rate				
44.1k	2646k	5292k	5292k	10584k
22.05k	1323k	2646k	2646k	5292k
11.025k	661.5k	1323k	1323k	2646k
8k	480k	960k	960k	1920k

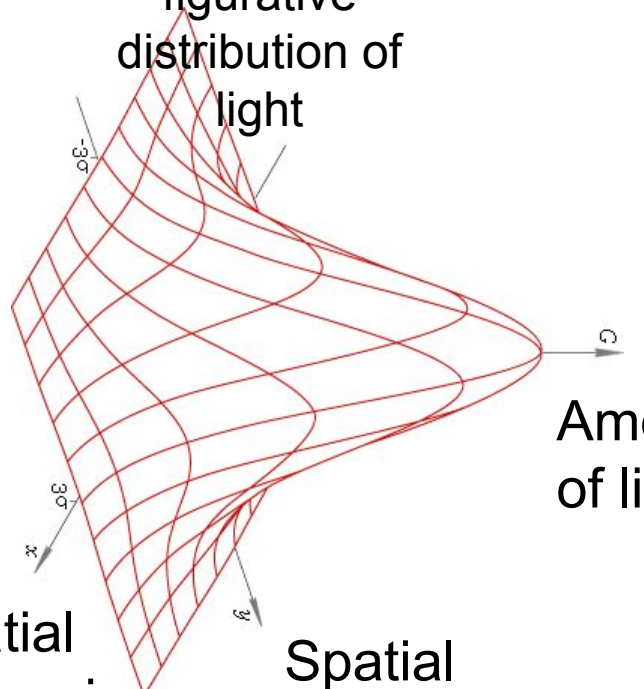
The Blue Marble. NASA | Apollo 17



Light received by camera

[Hasselblad 500 EL - NASA variant – 'Data Camera' (HDC)]

Approximate figurative distribution of light



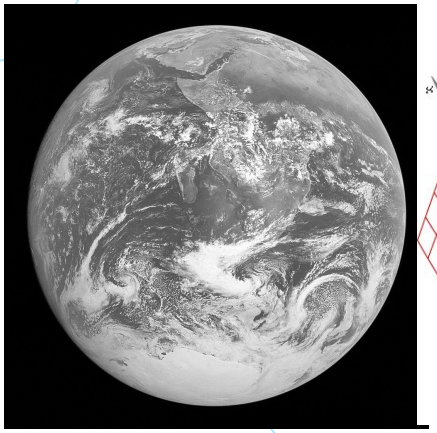
Amount of light

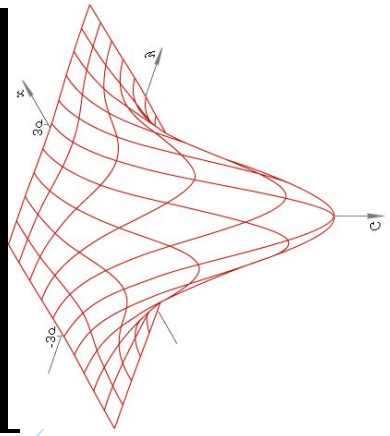
Light received by camera

atial dimension 1

Spatial dimension 2

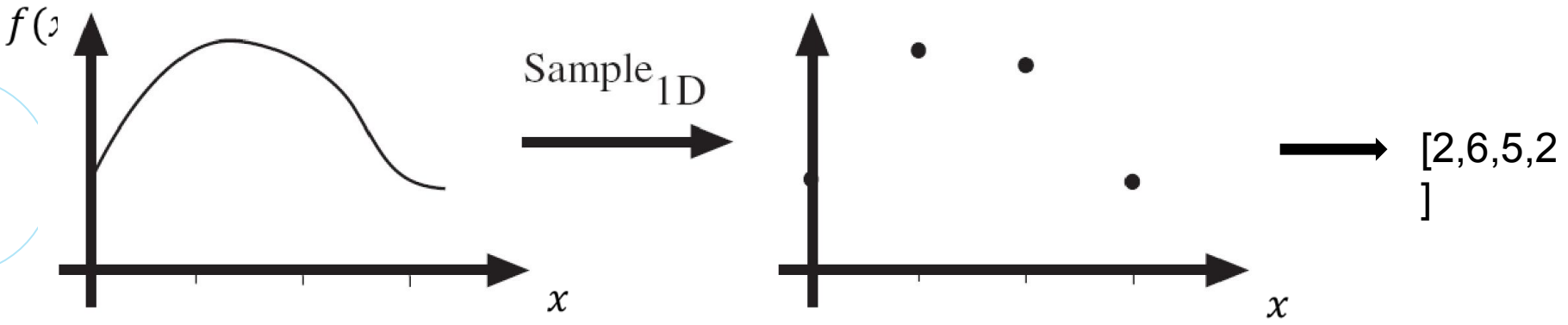
Signal





Sampling in 1D

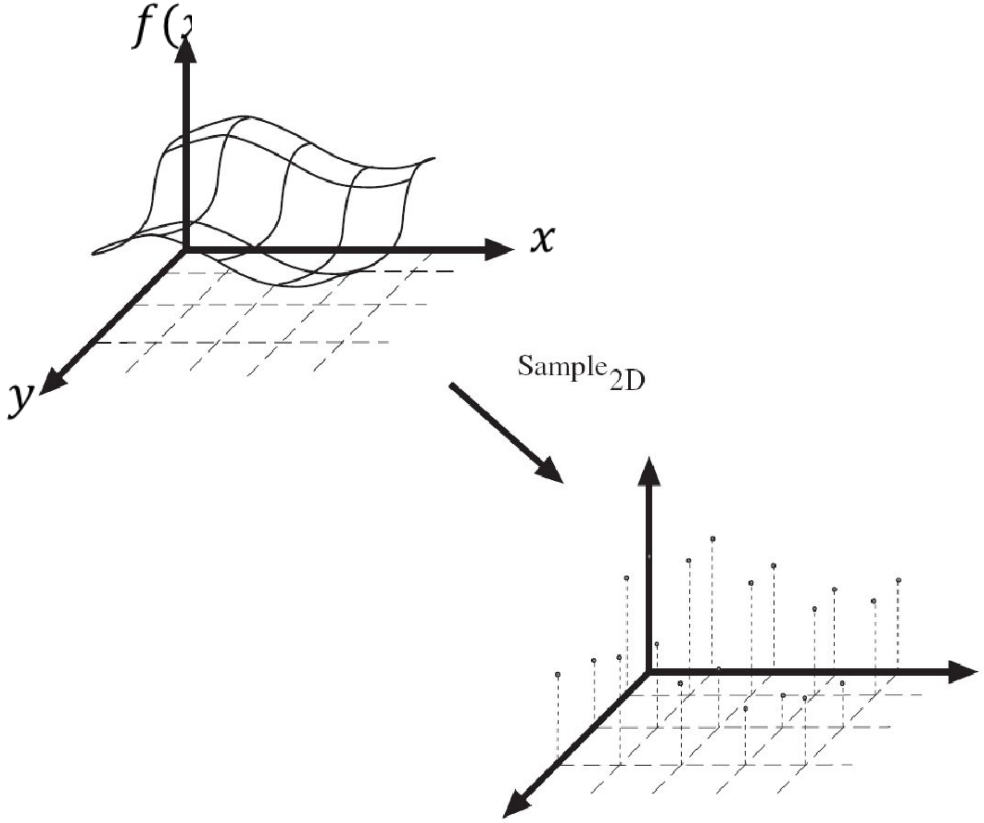
Takes a function f and returns a vector whose elements are values of that function at the sample points.



Danny Alexander

Sampling in 2D

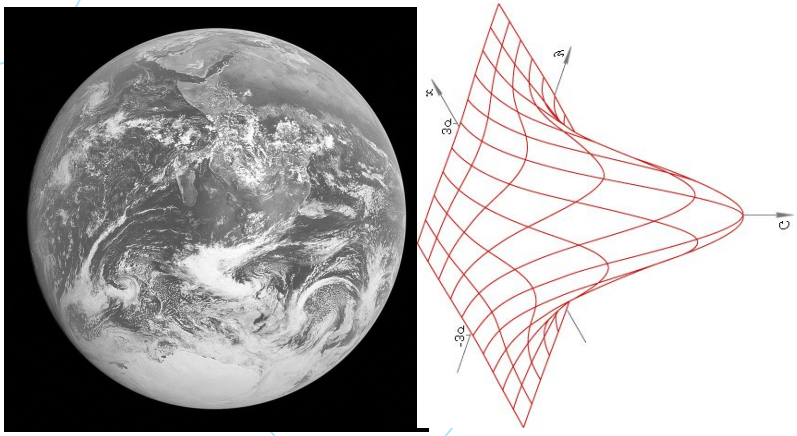
Sampling in 2D takes a function and returns a matrix.



Danny Alexander

2D Image

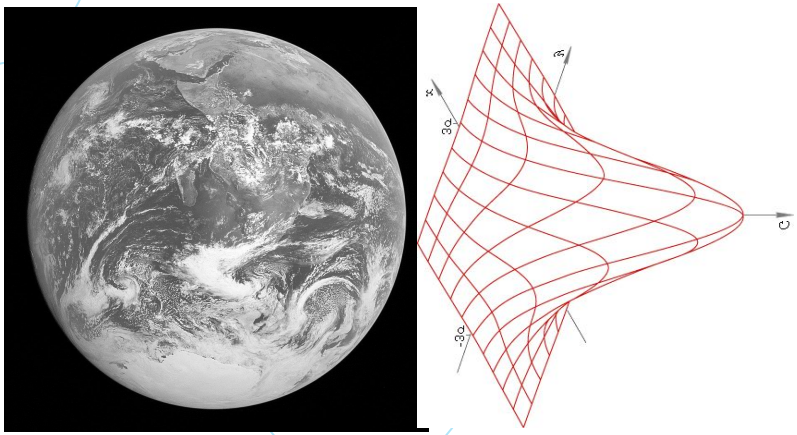
A sampling of a function that contains information about a 2D* signal.



2D Image

Image stores intensity or ‘brightness’

- A sampling of a function that contains information about a 2D signal.

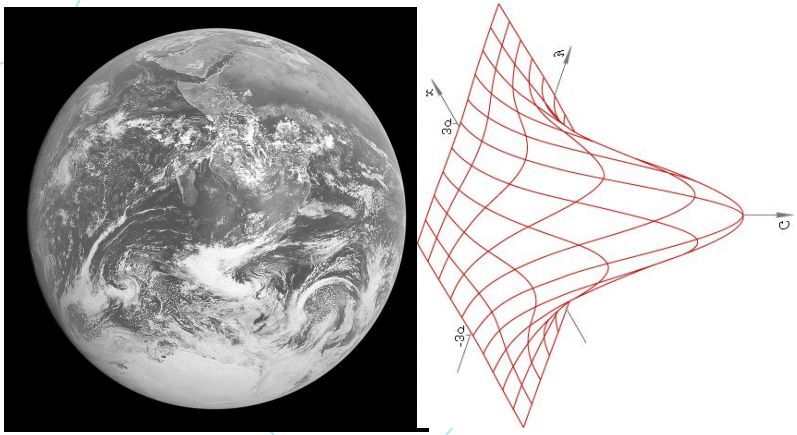


2D Image

Image stores intensity or ‘brightness’

2D visible light signals are special for us

- Brightness along x and y dimensions



2D Image

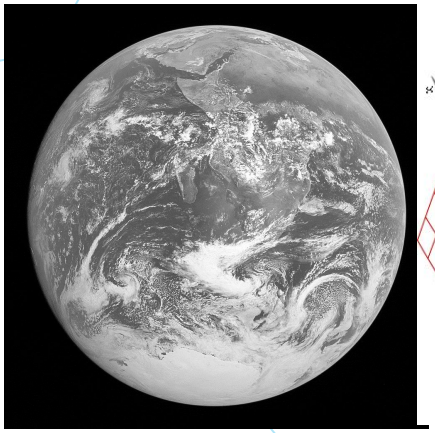
Image stores intensity or ‘brightness’

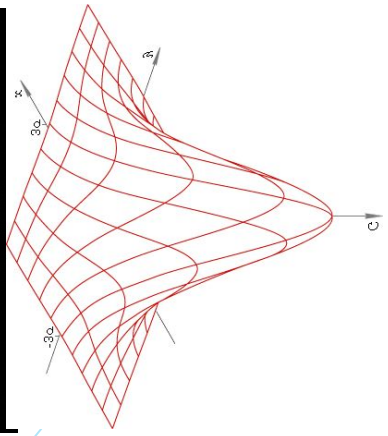
2D visible light signals are special for us

- Brightness along x and y dimensions

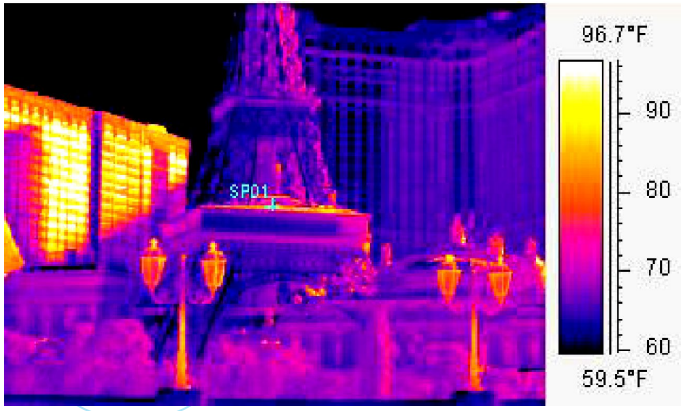
Video: xy-coordinates + time

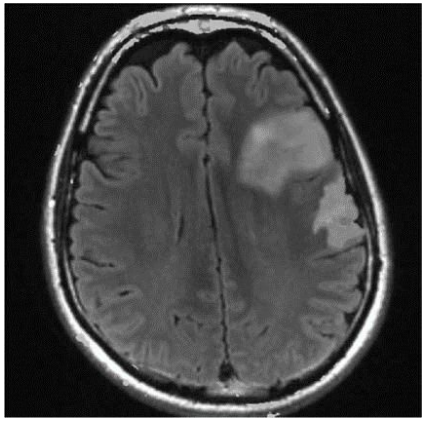
- Time-varying 2D signal





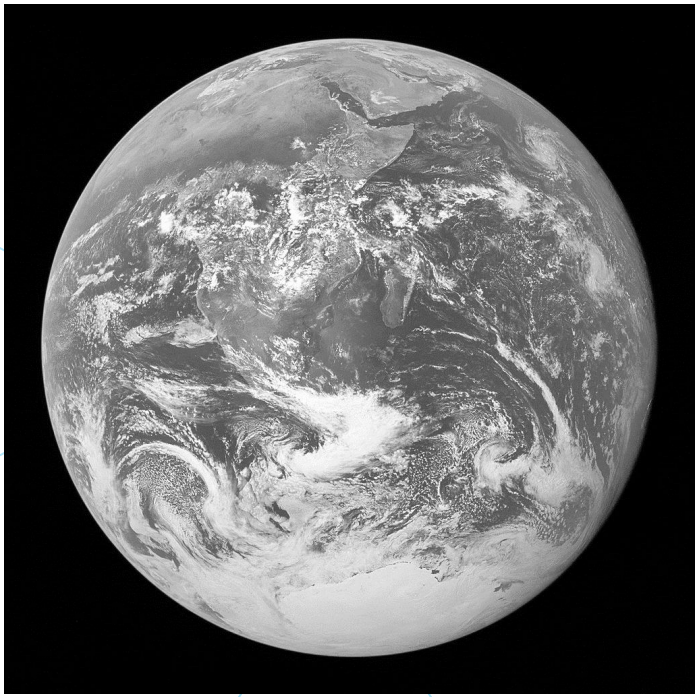
Example 2D Images

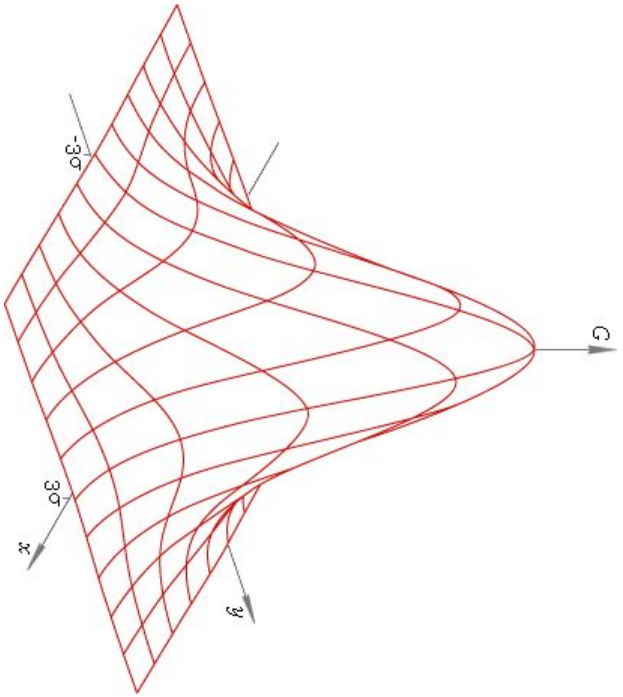


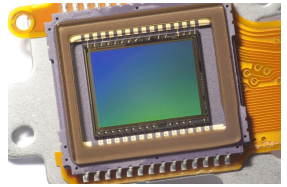


Danny Alexander

Sampling in Practice: Digital Images



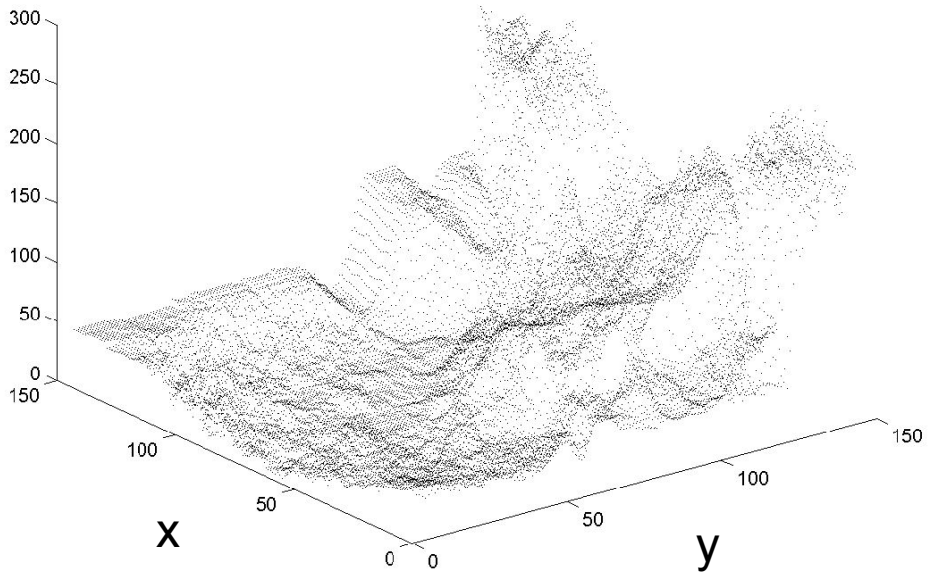




CMOS / CCD

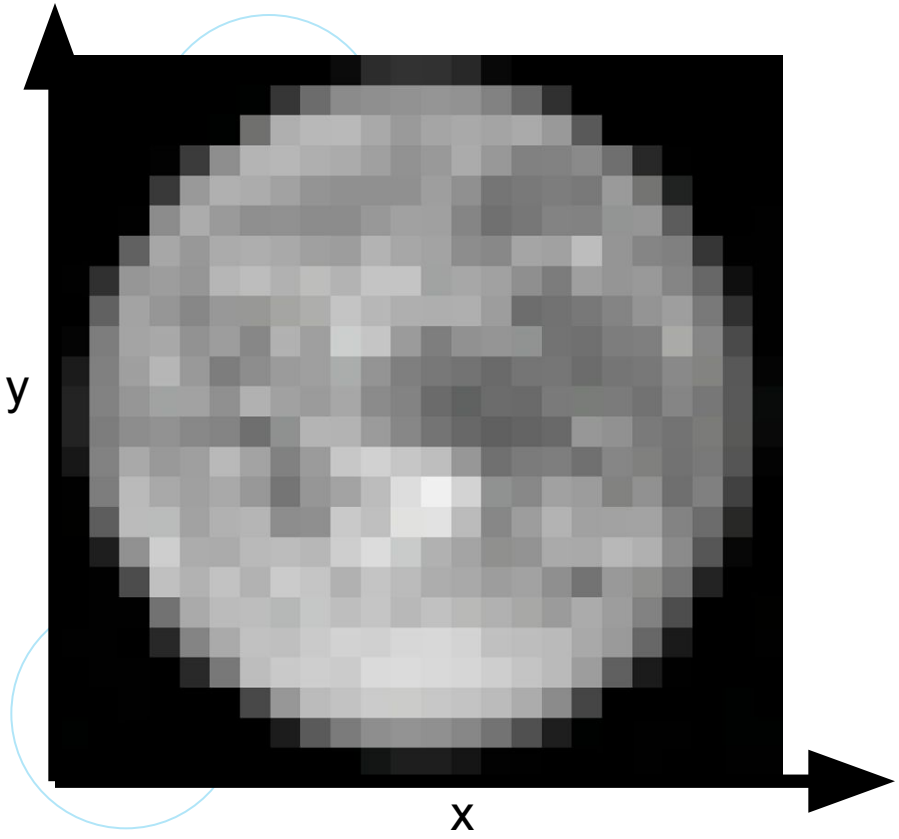
Grayscale Digital Image as 'Height Map'

Brightness or intensity



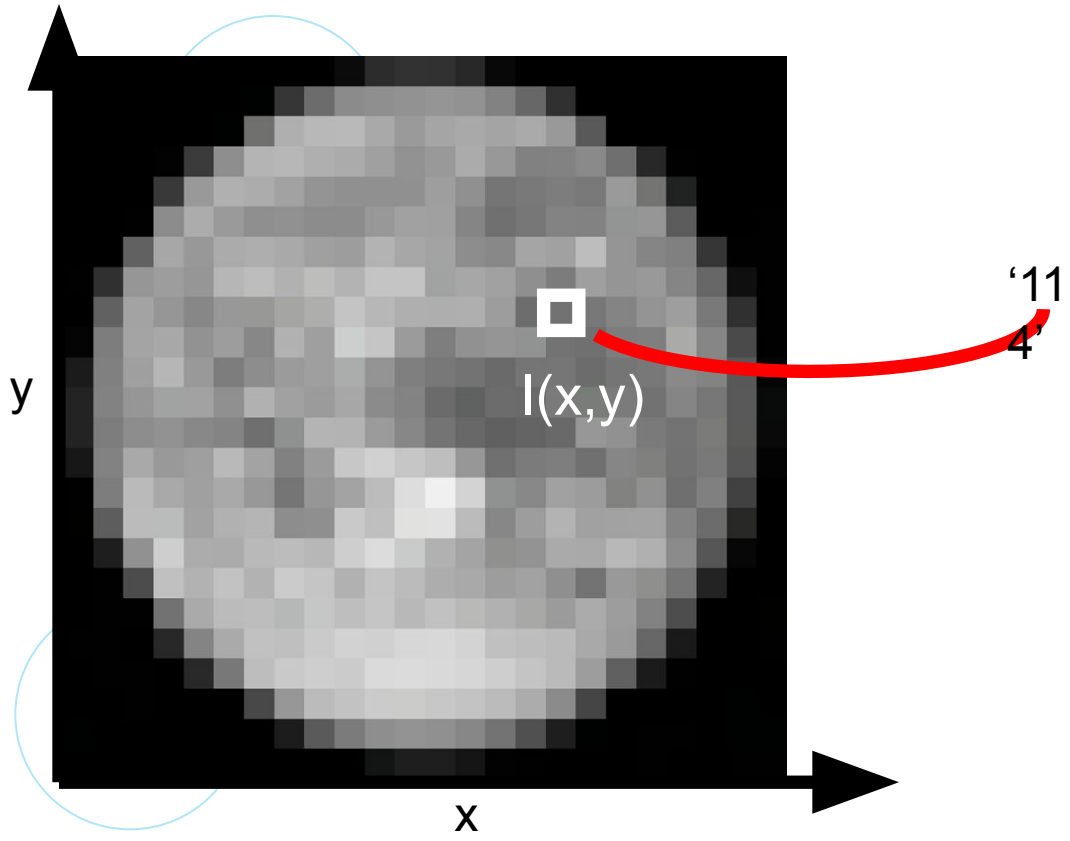
Danny Alexander

Each part of an image is a pixel



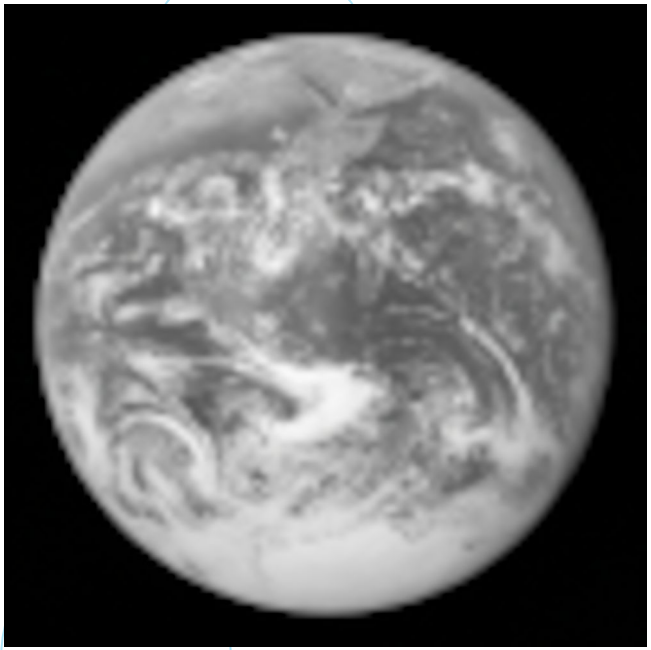
Each part of an image is a pixel

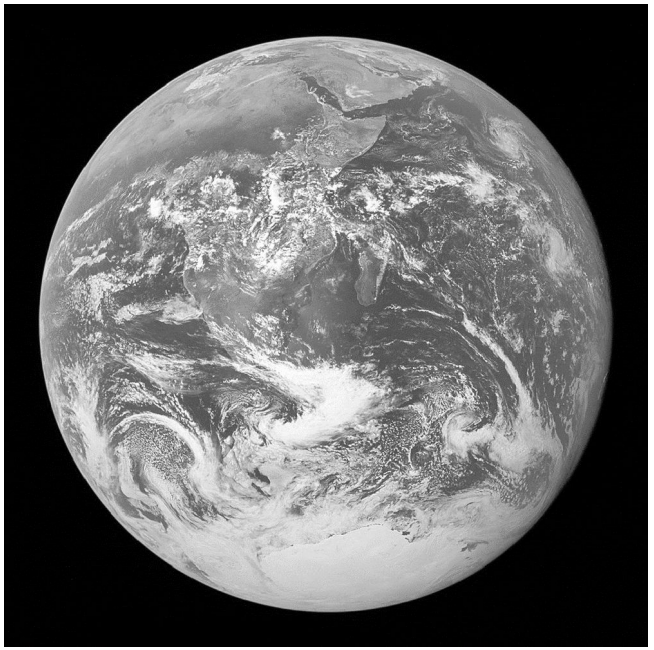
Pixel -> picture element



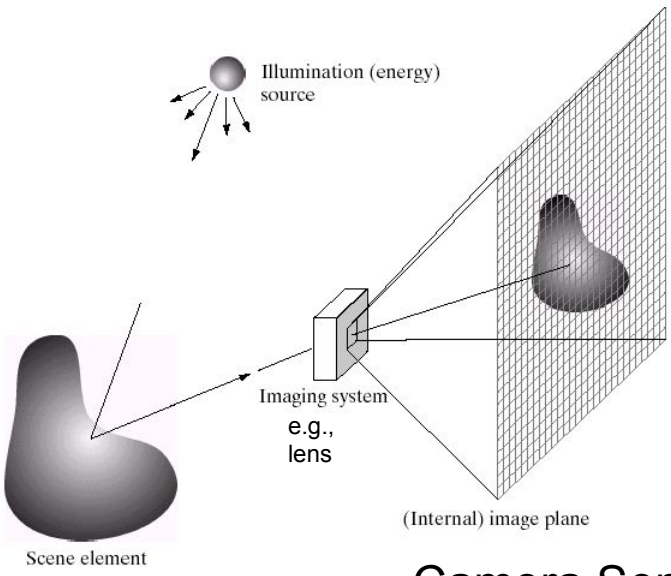
Resolution – geometric vs. spatial resolution

Both images are 1000 x 1000 pixels

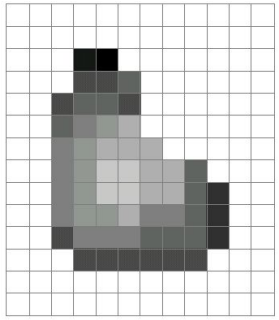




Integrating light



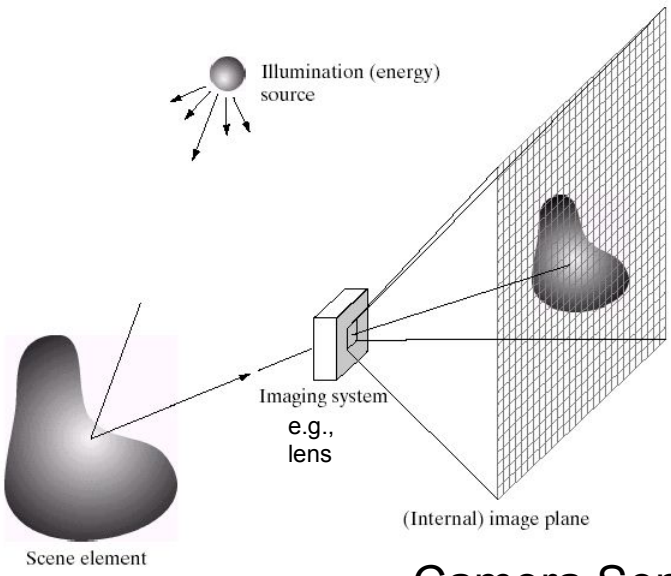
Camera Sensor



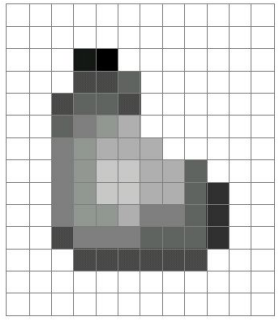
Output Image

James Hays

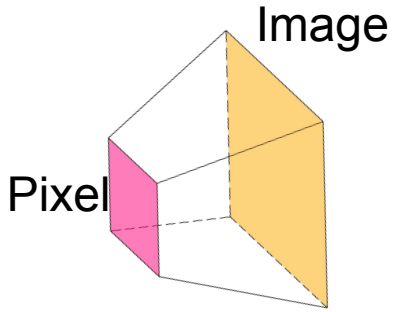
Integrating light over a *frustum*.



Camera Sensor



Output Image

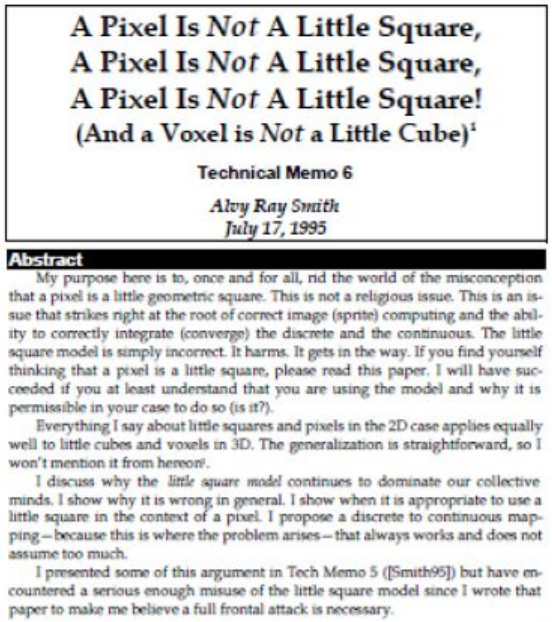


A frustum approximates the ray space that a pixel samples

James Hays

"*A Pixel Is* Not *A Little Square,*
A Pixel Is Not *A Little Square,*
A Pixel Is Not *A Little Square!*
(And a Voxel is Not *a Little Cube)*"

- Alvy Ray Smith,
- MS Tech Memo 6, 1995.



A Pixel Is *Not* A Little Square,
A Pixel Is *Not* A Little Square,
A Pixel Is *Not* A Little Square!
(And a Voxel is *Not* a Little Cube)[1]

Technical Memo 6

Alvy Ray Smith
July 17, 1995

Abstract

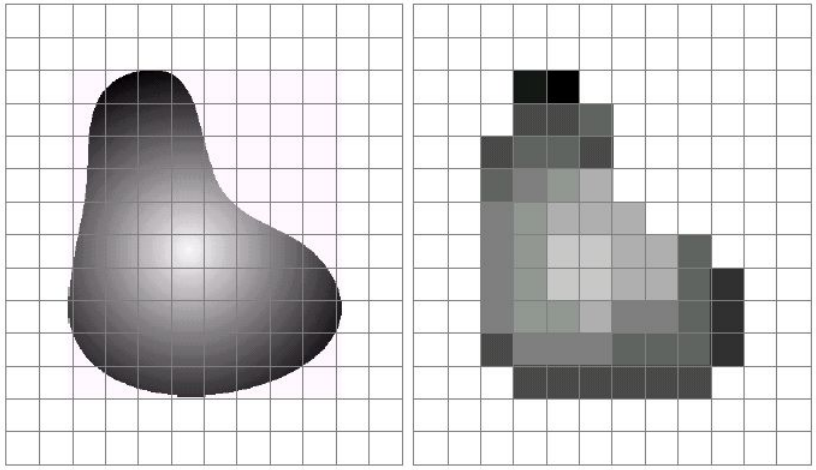
My purpose here is to, once and for all, rid the world of the misconception that a pixel is a little geometric square. This is not a religious issue. This is an issue that strikes right at the root of correct image (sprite) computing and the ability to correctly integrate (converge) the discrete and the continuous. The little square model is simply incorrect. It harms. It gets in the way. If you find yourself thinking that a pixel is a little square, please read this paper. I will have succeeded if you at least understand that you are using the model and why it is permissible in your case to do so (is it?).

Everything I say about little squares and pixels in the 2D case applies equally well to little cubes and voxels in 3D. The generalization is straightforward, so I won't mention it from hereon[2].

I discuss why the *little square model* continues to dominate our collective minds. I show why it is wrong in general. I show when it is appropriate to use a little square in the context of a pixel. I propose a discrete to continuous mapping—because this is where the problem arises—that always works and does not assume too much.

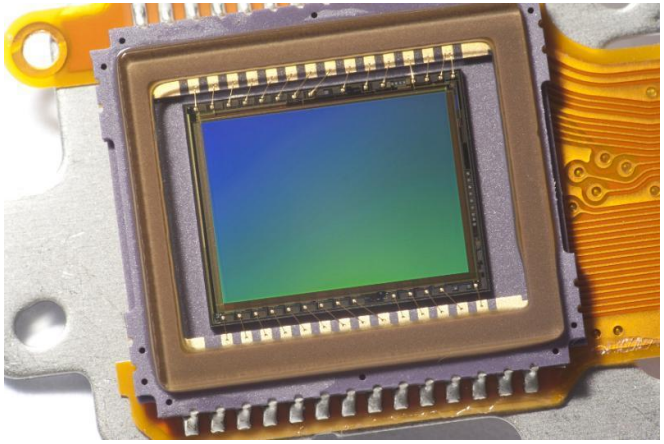
I presented some of this argument in Tech Memo 5 ([Smith95]) but have encountered a serious enough misuse of the little square model since I wrote that paper to make me believe a full frontal attack is necessary.

Sensor Array



a b

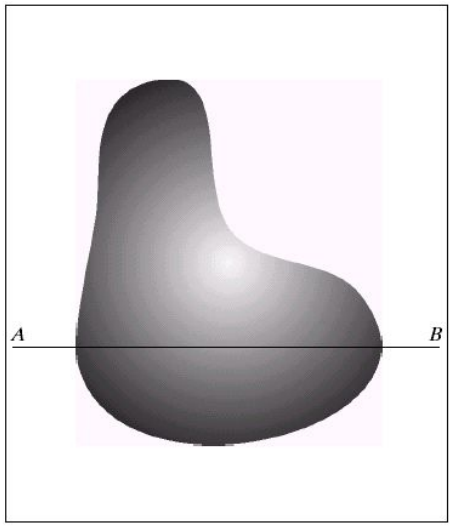
FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.



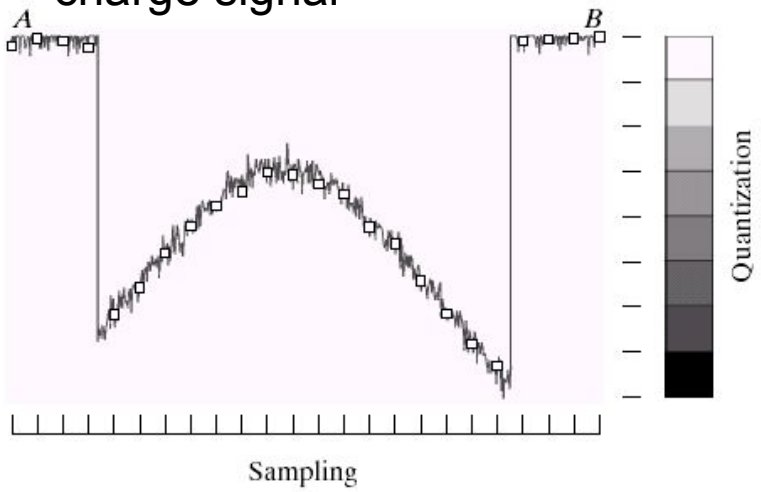
Imaging sensor

James Hay

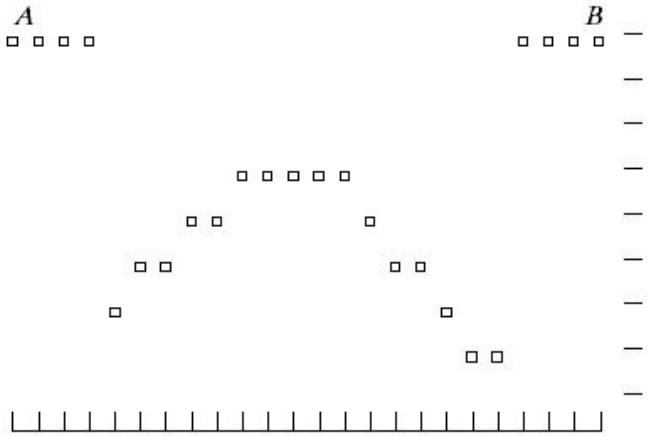
Quantization



Underlying analog charge signal



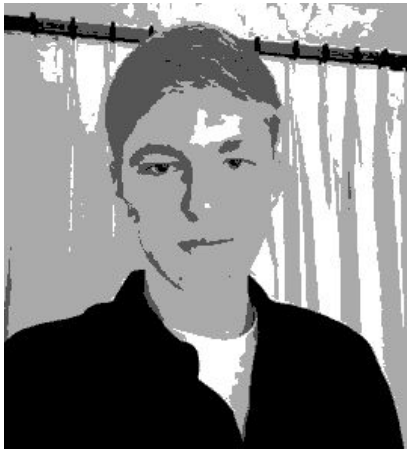
Quantized values



Quantization Effects – Photometric Resolution

8 bit – 256 levels

4 bit – 16 levels



2 bit – 4 levels

1 bit – 2 levels

We often call this *bit depth.*

For photography, this is also related to *dynamic range.*

Radiometric vs. spectroradiometric vs. photometric

Radiometric

The amount of light in physical units, e.g.,

- Radiance -> amount of light (in Watts per unit solid angle (steradian) per square meter)
- Irradiance -> amount of light arriving at a surface over the hemisphere (in Watts per square meter)

Spectoradiometric

Radiometric but for narrow wavelength bands of light

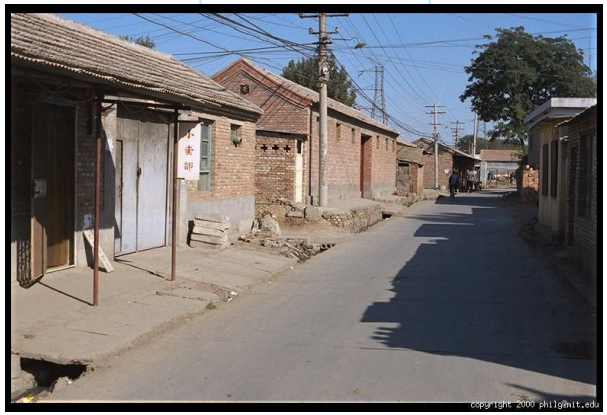
Photometric

The perceived amount of light to an observer, e.g., a human

- Luminous intensity (candelas -> lumens per steradian)
- Illuminance (lux -> lumens per square meter)

Grayscale intensity

Color



James Hays

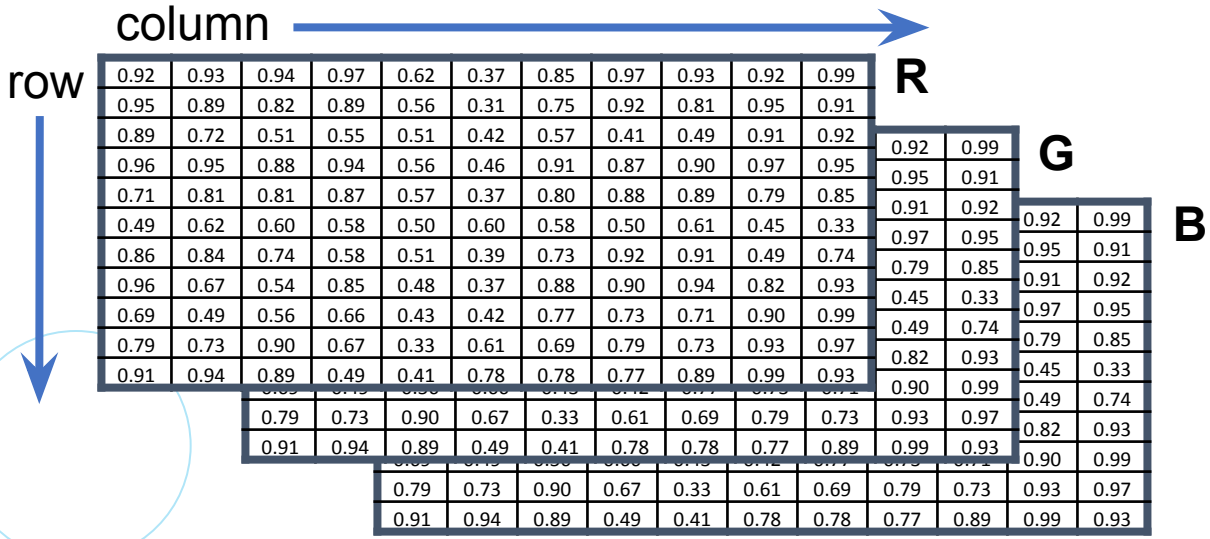
Images in Python Numpy

N x M RGB image “im”

– im[0,0,0] = top-left pixel value in R-channel

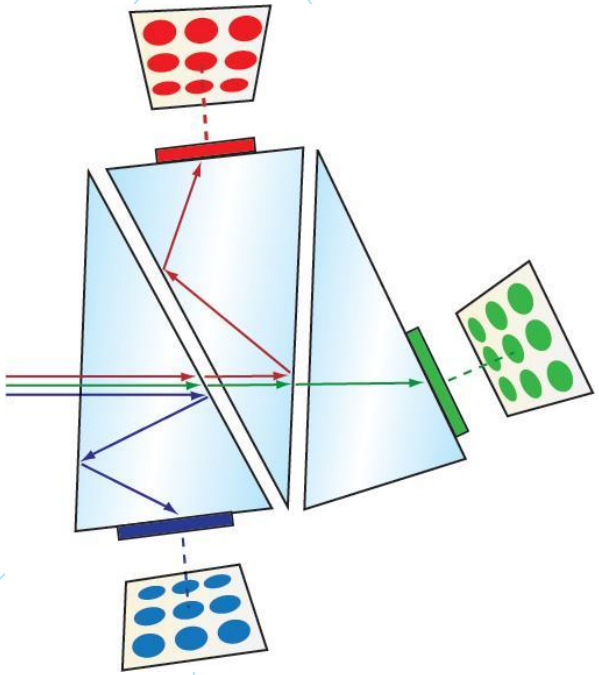
– im[x, y, b] = x pixels to right, y pixels down in the bth channel

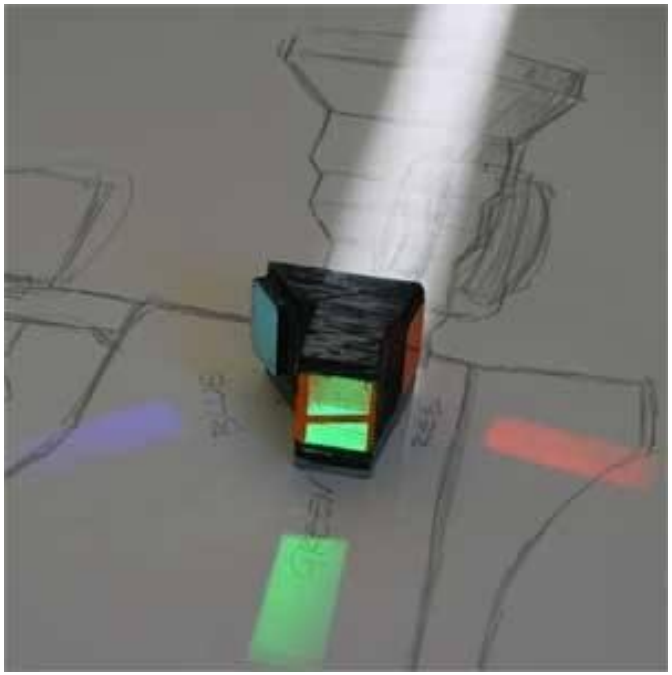
– im[N-1, M-1, 3] = bottom-right pixel in B-channel



James Hays

Cameras with Three Sensors

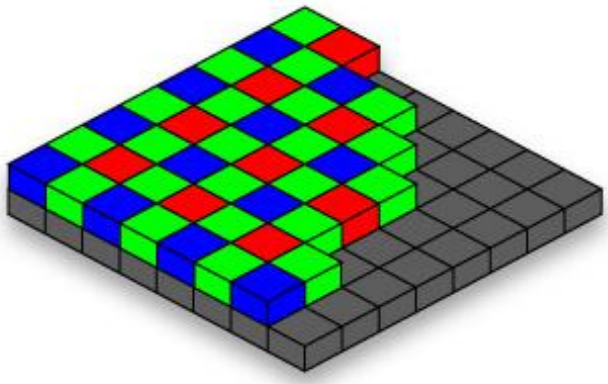


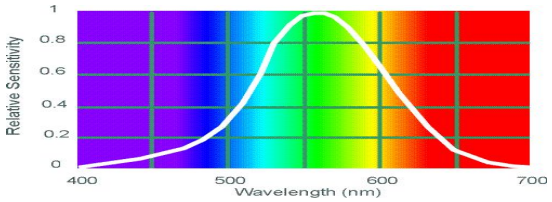


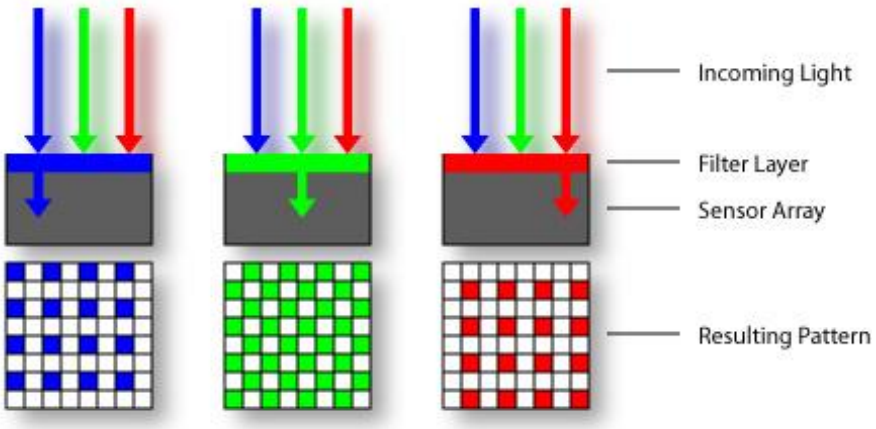
[Edmund Optics; Adam Wilt]

Cheaper/More Compact Color Sensing: Bayer Filter

- Estimate RGB at 'G' cells from neighboring values



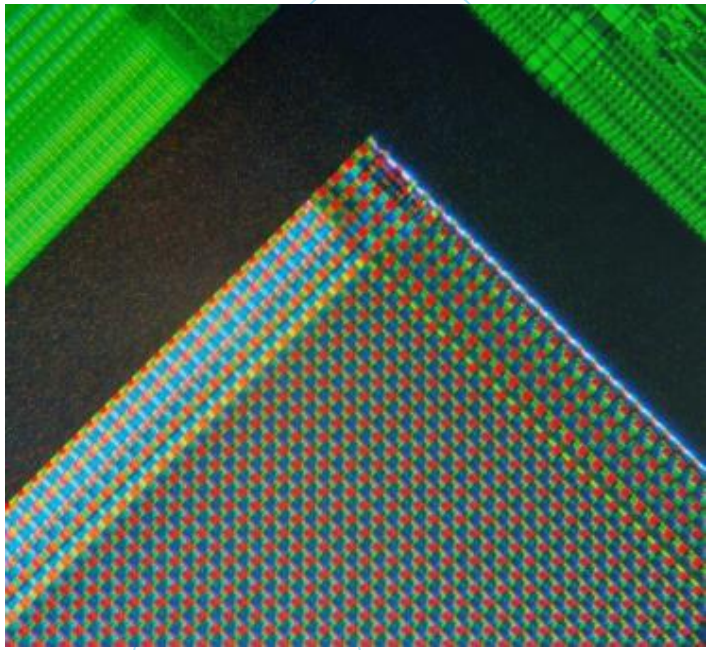




Steve Seitz

A microscope photograph (micrograph) of the corner of the photosensor array of a webcam digital camera.

Credit: Natural Philo, Wikipedia



Reliability

- Binary:
  - Full on/full off
  - Fully charged/fully discharged
  - Charged positively/charged negatively
  - Magnetized/nonmagnetized
  - Magnetized clockwise/magnetized counterclockwise

Voltage Level	*Corresponds to This Decimal Digit*
+0	0
+5	1
+10	2
+15	3
+20	4
+25	5
+30	6
+35	7
+40	8
+45	9

HUST