Vietnam National University HCMC
INTERNATIONAL UNIVERSITY

The Information Layer

Lecturer: Assoc. Phuong Vo, PhD

Contents

Numbers and Computing

- Natural numbers (1, 2, 3,…), negative numbers (-1, -2, …), rational numbers (1.23, 4.67,…), irrational numbers (pi number), and many others that are not to the understanding of computing.

Positional Notation

Positional notation A system of expressing numbers in which the digits are arranged in succession, the position of each digit has a place value, and the number is equal to the sum of the products of each digit by its place value.

943 in decimal number system

$$
\begin{aligned}
9 * 10^2 = 9 * 100 &= 900 \\
+\,4 * 10^1 = 4 * 10 &= 40 \\
+\,3 * 10^0 = 3 * 1 &= 3 \\
&\hline
&943
\end{aligned}
$$

Bases

- Base: The foundational value of a number system, which dictates the number of digits and the value of digit positions.
  - Decimal system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Binary system: 0, 1
  - Octal: 0, 1, 2, 3, 4, 5, 6, 7
  - Hexadecimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Binary, Octal, and Hexadecimal

1010110 in binary

$$
\begin{aligned}
1 * 2^6 &= 1 * 64 = 64 \\
+\,0 * 2^5 &= 0 * 32 = 0 \\
+\,1 * 2^4 &= 1 * 16 = 16 \\
+\,0 * 2^3 &= 0 * 8 = 0 \\
+\,1 * 2^2 &= 1 * 4 = 4 \\
+\,1 * 2^1 &= 1 * 2 = 2 \\
+\,0 * 2^0 &= 0 * 1 = 0 \\
&\qquad\qquad 86
\end{aligned}
$$

754 in octal

$$
\begin{aligned}
7 * 8^2 &= 7 * 64 = 448 \\
+\,5 * 8^1 &= 5 * 8 = 40 \\
+\,4 * 8^0 &= 4 * 1 = 4 \\
&\qquad\qquad 492
\end{aligned}
$$

ABC in hexadecimal

$$
\begin{aligned}
A * 16^2 &= 10 * 256 = 2560 \\
+\,B * 16^1 &= 11 * 16 = 176 \\
+\,C * 16^0 &= 12 * 1 = 12 \\
&\qquad\qquad 2748
\end{aligned}
$$

Quiz

Transform these numbers to decimal numbers

- 11110000 (binary)
- 1101100 (binary)
- ABF (hexadecimal)
- FF (hexadecimal)
- F0 (hexadecimal)
- 154 (octal)
- 630(octal)

Arithmetic in Other Bases

```
  11111        ← carry
   101110
+   11011
  1001001
```

```
     1
    0⁄2     ← borrow
  111001
-   110
  110011
```

```
    02
    ⁄2     ← borrow
  111101
-   110
  110111
```

Converting from binary to octal

Converting 111101100 to octal number:

111	101	100
7	5	4

Converting 1010110 to octal number:

1	010	110
1	2	6

$$
\begin{aligned}
& 1 * 8^2 = 1 * 64 = 64 \\
+\ & 2 * 8^1 = 2 * 8 = 16 \\
+\ & 6 * 8^0 = 6 * 1 = 6 \\
& \qquad\qquad\quad 86
\end{aligned}
$$

Hexadecimal Notation

- Shorthand notation for **bit stream**
- Example: 'H' is represented by 0x48 in ASCII
- Converting from binary to hexadecimal number:

0100 1000 (4 bits ➔ 1 Hex digit)

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Converting from Base 10 to Other Bases

WHILE (the quotient is not zero)
  Divide the decimal number by the new base
  Make the remainder the next digit to the left in the answer
  Replace the decimal number with the quotient

Examples: transform 2748 to hexadecimal

- 2748 = 171x16 + 12 → C
- 171 = 10x16 + 11 → BC
- 10 = 0x16 + 10 → ABC

Why binary?

- Bit - *binary digit* : 0 or 1 (low or high voltage)
- Bit patterns represents
  - numeric values (0110_2 = ?)
  - characters ('H' = 0100 1000)
  - images (greyscale image pixels)
  - sounds

Storage units

- **Bit**: Binary digit
- **Byte**: Eight binary digits
- **Word**: A group of one or more bytes; the number of bits in a word is the word length of the computer
- Kilo byte (KB), Mega byte (MB), Giga byte (GB), Tetra Bytes (TB), Penta Byte (PB)

Quiz

1. 1010101 + 10101 (binary addition)
2. 1010110 – 101 (binary subtraction)
3. 1066 + ABCD (hexadecimal addition)
4. Convert the following binary numbers to hexadecimal 111110110
5. Convert the following numbers from the base shown to base 10.
   a. 111 (base 2)
   b. 777 (base 8)
   c. FEC (base 16)
6. Convert the following hexadecimal numbers to octal. a) A9; b) E7

Contents

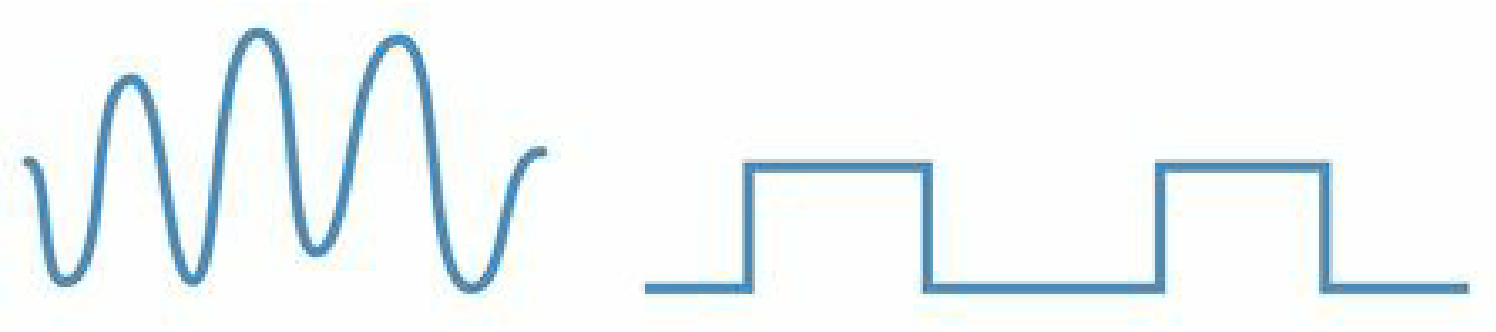
1. Binary Values And Number Systems
2. **Data Representation**

Different types of data

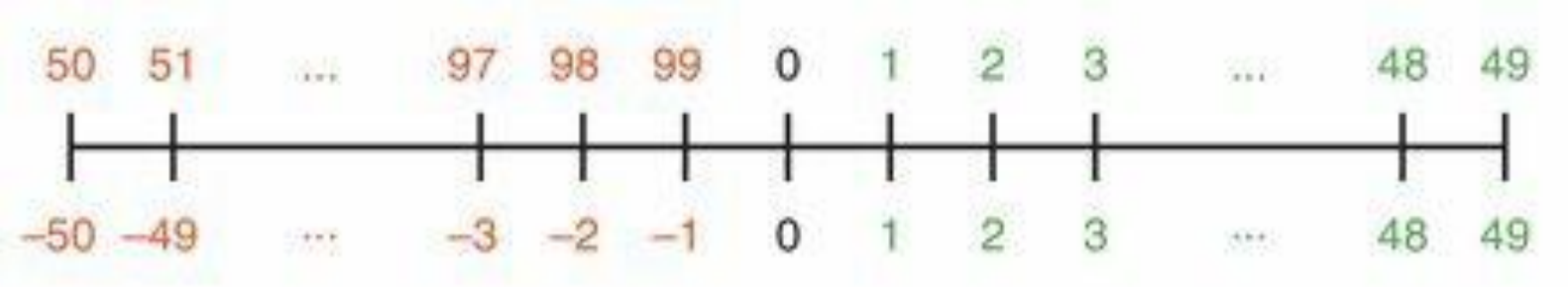
- Numbers
- Text
- Audio
- Images and graphics
- Video

Analog and Digital Data

- **Analog data**: A continuous representation of data
- **Digital data**: A discrete representation of data
- **Digitize**: The act of breaking information down into discrete pieces



Representing Negative Values



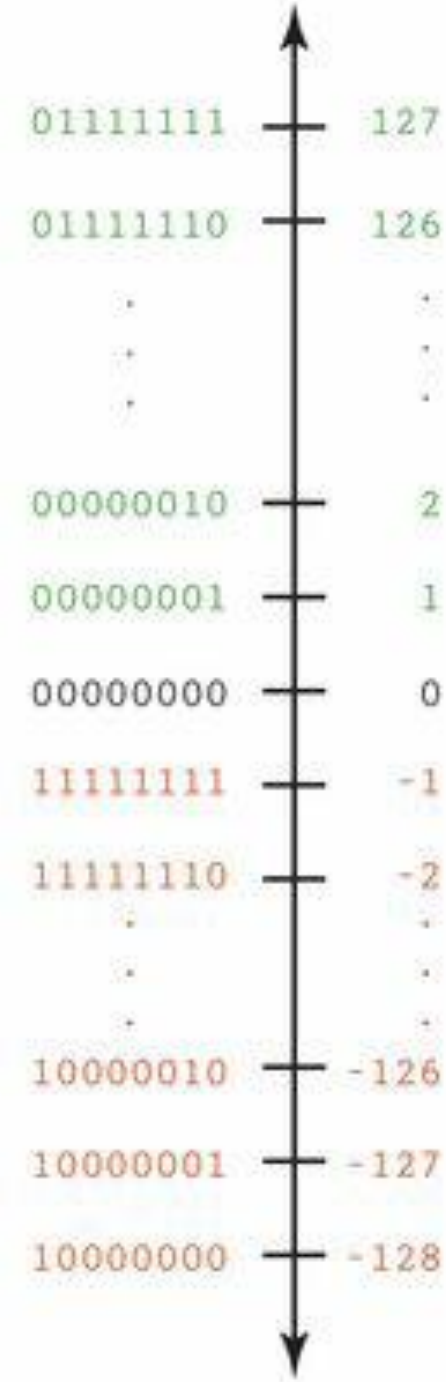
E.g., -3 in two-digit representation is 97.

Ten's complement: Negative(N) = 10^k – N

Signed-Magnitude	New Scheme
5 + – 6 – 1	5 + 94 99
– 4 + 6 2	96 + 6 2
– 2 + – 4 – 6	98 + 96 94

Two's complement

- negative(N) = 2^k - N
  - $-(2) = 2^8 - 2 = 254$
- Number **overflow**: calculated value cannot fit into the number of digits
  - 127+3



Example of two's complement

Evaluate the following expressions, where A is 11111110 and B is 00000010, using the two's complement (k=8).

a. A + B

b. A – B

c. B – A

d. –B

e. – (–A)

Representing Real Numbers

Floating point presentation for *decimal number*: $sign * mantissa * 10^{\exp}$

$$148.69 = 14869 * 10^{-2}$$

TABLE 3.1 Values in decimal notation and floating-point notation (five digits)

Real Value	Floating-Point Value
12001.00	$12001 * 10^{0}$
−120.01	$-12001 * 10^{-2}$
0.12000	$12000 * 10^{-5}$
−123.10	$-12310 * 10^{-2}$
155555000.00	$15555 * 10^{4}$

Representing Real Numbers

In binary: Floating point presentation:

$$sign * mantissa * 2^{\exp}$$

- **E.g. 1**: **0.25** in decimal is 0.11 in binary

since $0.25 * 2 = 0.50; 0.50 * 2 = 1.00$

Hence, $0.25 = 2^{-2} \sim 0.11$(in bin)

- **E.g. 2**: **0.435** in decimal is .111111 since

$$0.435 * 2 = 0.870$$
$$* 2 = 1.740$$
$$* 2 = 3.480$$
$$* 2 = 6.960$$
$$* 2 = 13.920$$
$$* 2 = 27.84$$
$$0.435 = 27 * 2^{-6}$$

Example

Convert the following real numbers to binary (five binary places).

a. 0.50

b. 0.26

c. 0.10

Representing text: ASCII character set (256 characters)

Left Digit(s) \ Right Digit	ASCII 0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
1	LF	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3
2	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
3	RS	US	□	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

Representing text – unicode character set (16 bits)

Code (Hex)	Character	Source
0041	A	English (Latin)
042F	Я	Russian (Cyrillic)
0E09	ฉ	Thai
13EA	Ꮺ	Cherokee
211E	℞	Letterlike symbols
21CC	⇌	Arrows
282F	⠯	Braille
345F	㑟	Chinese/Japanese/ Korean (common)

Data compression techniques

- Why compression?
  We compress the data to improve the efficiency of data transmission.
- Keyword encoding
- Run-length encoding
- Huffman encoding

Keyword encoding

Keyword encoding: Replacing a frequently used word with a single character.

WORD	SYMBOL
as	^
the	~
and	+
that	$
must	&
well	%
these	#

Run-Length Encoding

- Replacing a long series of a repeated character with a count of the repetition.
- *E.g., AAAAAAA => *A7.*
- E.g., decode this string
  **n5*x9ccc*h6 some other text *k8eee*

nnnnnxxxxxxxxxccchhhhhh some other text kkkkkkkkeee

- **Compression ratio**: the ratio between the compressed string and the original string.
- Compression ratio of the above example is 35/51.

Huffman Encoding

- Using a variable-length binary string to represent a character so that frequently used characters have short codes

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

Huffman Encoding

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

DOORBELL => 1011 110 110 111 1010 01 1001 00

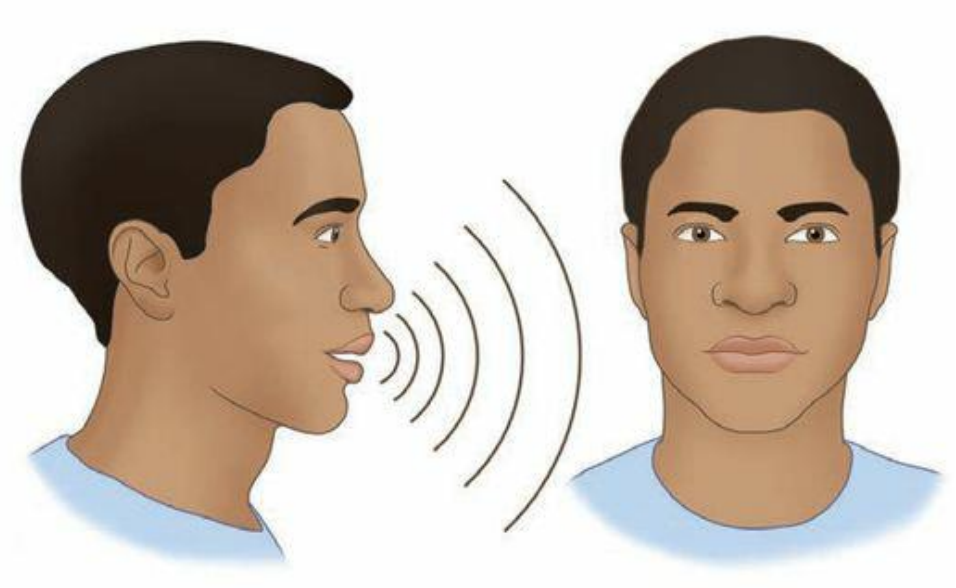
Example

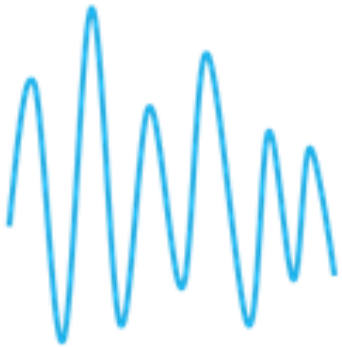
HUFFMAN CODE	CHARACTER
00	A
11	E
010	T
0110	C
0111	L
1000	S
1011	R
10010	O
10011	I
101000	N
101001	F
101010	H
101011	D

10100100101000010001000010100110110

Representing Audio Data

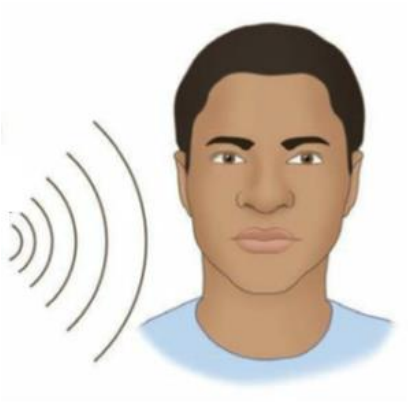
Analog sound:





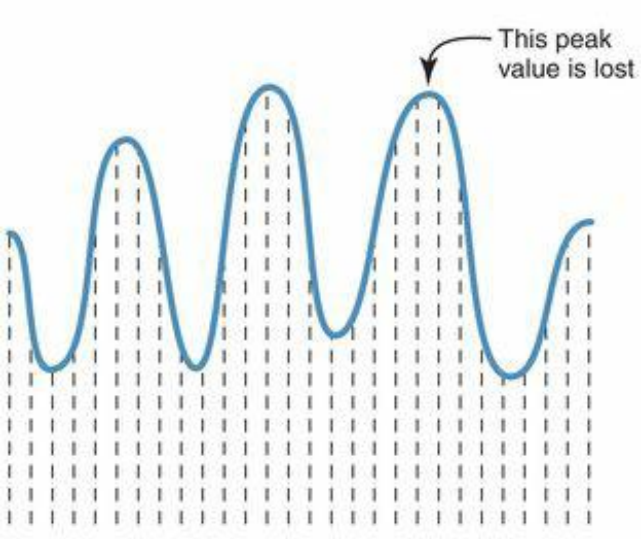
Analog signal

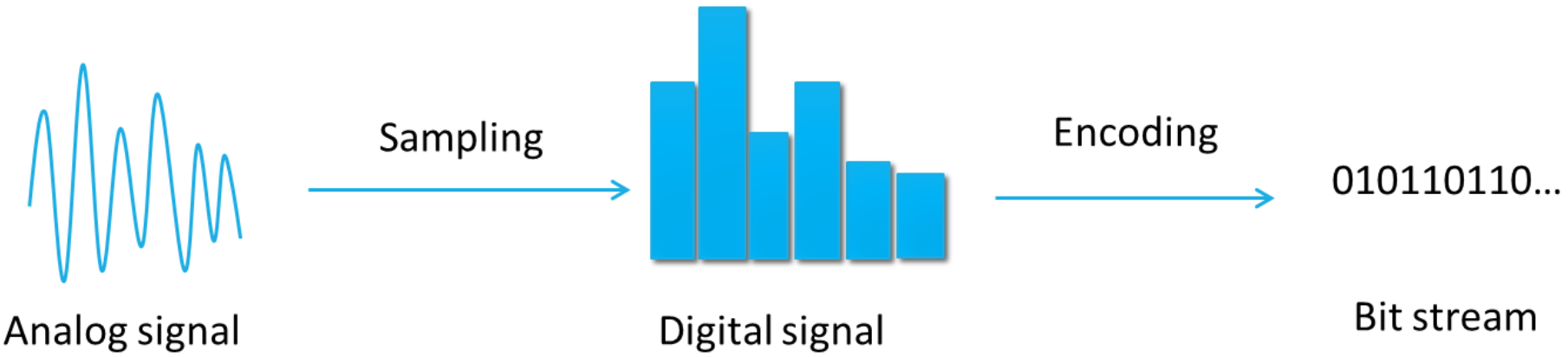
➔



Representing Audio Data

- Sampling an audio signal



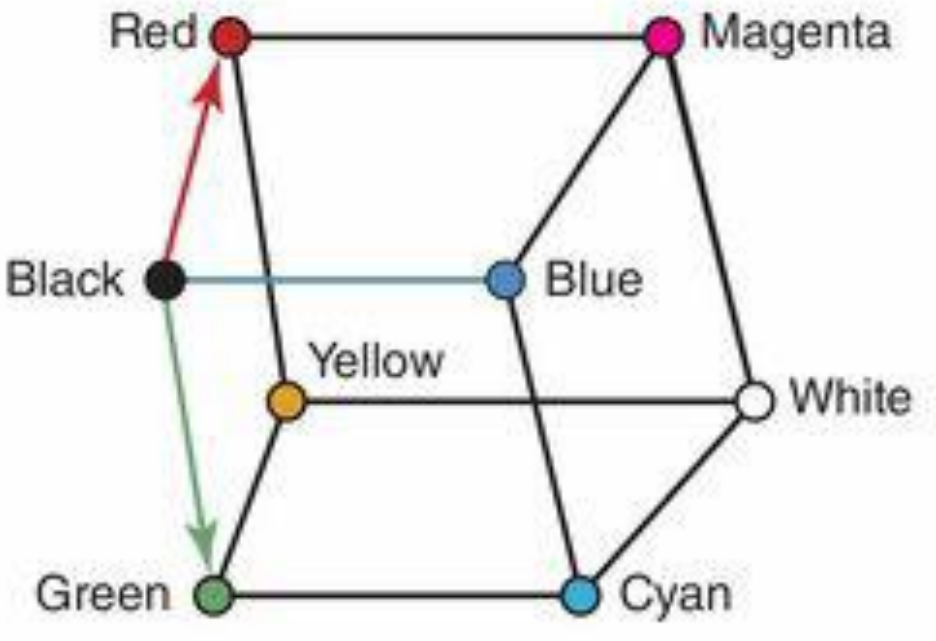


Audio Formats

- WAV, AU, AIFF, VQF, and MP3: recognize the details of the data in different ways and use various compression techniques
- MP3: MPEG-2, audio layer 3 file

Representing Color

- Represent RGB values in 3-D color space

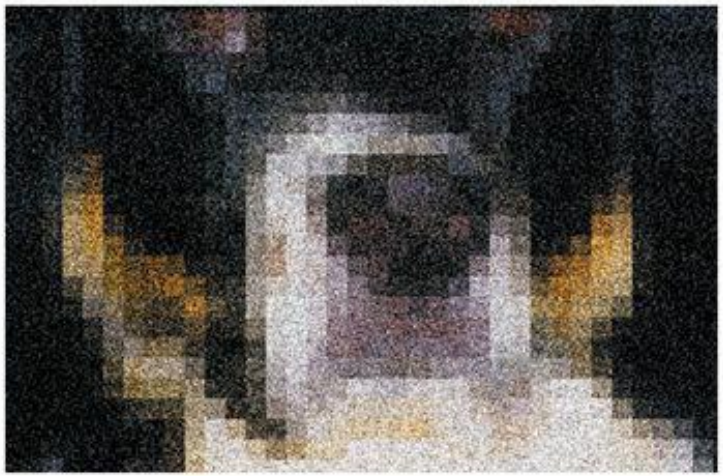


- High color: 16-bit color depth
- True color: 256-bit color depth

RGB VALUE			
Red	Green	Blue	Color
0	0	0	black
255	255	255	white
255	255	0	yellow
255	130	255	pink
146	81	0	brown
157	95	82	purple
140	0	0	maroon

Digitized Images and Graphics

- **Pixels** Individual dots used to represent a picture; stands for picture elements
- **Resolution** The number of pixels used to represent a picture

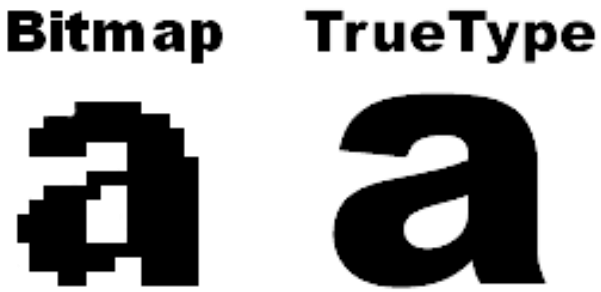


Raster-graphics format

- Storing image information pixel by pixel
- Bitmap file, GIF, JPEG, PNG

Vector graphics format

- Representation of an image in terms of lines and shapes
- A series of commands that describe a line's direction, thickness, and color → small file size
- Flash, SVG



Represent video

- **Video codec**: Methods used to shrink the size of a movie
- **Temporal compression**: Movie compression technique based on differences between consecutive frames
- **Spatial compression**: Movie compression technique based on the same compression techniques used for still images (run-length encoding)

Q&A