

9장: 자료의 표현

-
- 인간은 다양한 **정보**를 컴퓨터로 처리하고 사용한다.
 - 정보를 구체적으로 (숫자, 문자, 소리, 그림, 사진, 동영상, 그래픽) 등의 다양한 **데이터** 형식으로 사용한다.
 - 그러면 인간이 사용하는 정보가 컴퓨터 하드웨어에는 어떻게 **표현되고 저장**되는가?
 - 이 장에서는 정보가 컴퓨터 하드웨어 표현되고 저장되는 개념을 살펴본다.

이장의 목적

- 디지털 표현이란 무엇인가?
- 수를 표현하기 위한 방법을 학습하고 이진수, 팔진수, 16진수의 표현 방법을 익힌다.
- 정수와 실수의 컴퓨터의 내부에서는 어떻게 표현하는지를 학습한다.
- 문자/소리/영상을 표현하기 위한 기본적인 방법을 학습한다.

9.1 디지털 표현

디지털 표현

- 컴퓨터는 0과 1로 된 이진 값을 데이터로 사용한다.
- 연속 신호는 '연속으로 변하는 물리량을 표현하는 용어: 음성, 온도, 풍량, 영상 등. - 아날로그 신호
- 비연속 신호(discrete) 신호는 이산적 물리량: 연도, 날짜, 요일, 개수 - 디지털 신호 - 숫자(digit)로 표현가능
- 컴퓨터는 디지털 신호를 이진수 digit(0이나 1)로 표현
- 0또는 1을 나타낼 수 있는 비트를 사용하여 모든 자료(수, 문자, 사진, 소리, 음성, 동영상,...)를 표현

9.2 수 표현

수 표현

- 이진수는 0과 1로 수를 나타내며, 2를 올림수로 표현한다.
- 4-비트는 모두 $16(2^4)$ 개를 표현
 - 각 자리를 0과 1로 표현하며, 0000(십진수 0)부터 1111(십진수 15)까지 이다.
 - N-비트이면 표현할 수 있는 수는 모두 2^N 개이고, 수의 표현 범위는 $0 \sim 2^N - 1$ 이다.

이진수	십진수
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

P.291

2^N 개에
범위는 $(0 \sim 2^N - 1)$.

수 표현

- 이진수를 십진수로 변환

$$0101_2 = 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 0 + 4 + 0 + 1 = 5$$

$0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 1 = 5$

- 십진수를 이진수로 변환

- 결과는 이진수 100011_2 이다.

$35 \div 2 = 17$	나머지 1
$17 \div 2 = 8$	나머지 1
$8 \div 2 = 4$	나머지 0
$4 \div 2 = 2$	나머지 0
$2 \div 2 = 1$	나머지 0
$1 \div 2 = 0$	나머지 1 ↑

$$\begin{aligned} 35 \div 2 &= 17 \dots 1 \\ 17 \div 2 &= 8 \dots 1 \\ 8 \div 2 &= 4 \dots 0 \\ 4 \div 2 &= 2 \dots 0 \\ 2 \div 2 &= 1 \dots 0 \\ 1 \div 2 &= 0 \dots 1 \end{aligned}$$

$$\begin{aligned} &100011_2 \\ &= 32 + 2 + 1 \\ &= 35 \end{aligned}$$

수 표현

- 소수점이 있는 십진수를 이진수로 변환

- 25.458를 이진수로 변환

- 정수부분은 변환 방법에 의하여 이진수로 변환하면 11001_2

$0.458 \times 2 = 0.916$ 에서 소수점 앞의 정수부분 분리 0

$0.916 \times 2 = 1.832$ 에서 소수점 앞의 정수부분 분리 1, 나머지 부분 0.832

$0.832 \times 2 = 1.664$ 에서 소수점 앞의 정수부분 분리 1, 나머지 부분 0.664

$0.664 \times 2 = 1.328$ 에서 소수점 앞의 정수부분 분리 1, 나머지 부분 0.328

.....

- 소수점 이하가 0이 되거나 정확도에 따라 소수점 이하 자리수를 지정한 만큼의 정수부분만 구하여 순서대로 사용한다

- 25.458의 변환 값은 이진수 $11001.0111..._2$ 이다.

$$0.458 \times 2 = 0.916 \dots 0$$

$$0.916 \times 2 = 1.832 \dots 1$$

$$0.832 \times 2 = 1.664 \dots 1$$

$$0.664 \times 2 = 1.328 \dots 1$$

$$\therefore 11001.0111..._2$$

수 표현

- 8진수란 8을 올림수로 하여 수를 표현한다.
- 표현 방법은 8진수 앞에 수(0이나 소문자 o를) 붙인다(예, 0123 혹은 o123). 또는 8진수 뒤에 아래첨자 8을 붙이기도 한다. 즉, 123₈와 같이 표현한다.
- 이진수 \leftrightarrow 8진수 변환

101 110 010 \rightarrow 142₈

o562(562₈) \rightarrow 001 100 010

8진수	이진수	십진수
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

■ 그림 10.2 ■ 8진수 표현

수 표현

- 41.7_8 을 변환하려면, 먼저 8진수의 각 자리 수에 8의 맥급수를 가중치로 곱하여 더한다.

$$41.7_8 = 4 \times 8^1 + 1 \times 8^0 + 7 \times 8^{-1} = 32 + 1 + 0.875 = 33.875$$

Handwritten note: $4 \times 8^1 + 1 \times 8^0 + 7 \times 8^{-1} = 32 + 1 + 0.875 = 33.875$

- 266.4375 은 아래와 같이 변환한다.

$266 \div 8 = 33$	나머지 2	$0.4375 \times 8 = 3.5$ 에서	3을 취하고	↓
$33 \div 8 = 4$	나머지 1	$0.5 \times 8 = 4.0$ 에서	4를 취한 후	
$4 \div 8 = 0$	나머지 4	$0 \times 8 = 0$	0을 가져온다	

Handwritten notes: Upward arrow under '나머지 4', wavy line under '0' in the last row.

- 십진수 266.4375 는 8진수로 412.340_8 이다.

Handwritten note: (412.340_8)

수 표현

- 16진수는 (16개의 digit로) 값 표현,
- 각 값은 4-비트 패턴에 대응
- 표현방법 xA34 소문자 x를 붙인다(혹은 16진수 뒤에 아래첨자 16을 표기).
- 변환예
 $110011100010 \rightarrow 1100 \ 1110 \ 0010$
 $\rightarrow \text{"xCE2}$
 $\text{x24C} \rightarrow 0010 \ 0100 \ 1100 \rightarrow$
 001001001100

16진수	이진수	십진수
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

그림 10.3 16진수 표현

수 표현

□ 16진수 $2AF_{16}$ 를 십진수로 변환

□ 16진수의 각 자리 수에 16의 맥급수를 가중치로 곱하여 더한다.

$$\begin{aligned} 2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= 687 \end{aligned}$$

□ 십진수를 16진수로 변환하려면 몫이 0이 될 때까지 16으로 반복하여 정수형으로 나누면서 각 단계에서 나머지 값을 취한다.

$$423 \div 16 = 26 \quad \text{나머지 } 7$$

$$26 \div 16 = 1 \quad \text{나머지 } 10$$

$$1 \div 16 = 0 \quad \text{나머지 } 1 \quad \uparrow$$

"1A7"

□ 결과는 16진수 1A7이다.

$$\begin{aligned} 423 \div 16 &= 26 \quad \dots 7 \\ 26 \div 16 &= 1 \quad \dots 10 \\ 1 \div 16 &= 0 \quad \dots 1 \end{aligned}$$

9.3 정수와 실수의 컴퓨터 내부 표현

정수와 실수의 컴퓨터 내부 표현

- 컴퓨터에서 보통 사용하는 데이터 종류와 기술에 의한 유형

• 수 : 정수형, 실수형, 지수형, 복소수형

• 문자 : ASCII코드, EBCDIC 코드, 유니코드

• 이미지 : BMP, JPG

• 오디오 : WAV, MP3

• 비디오 : GIF, H.263, H.264, MPEG

- 어떤 유형의 데이터라도 컴퓨터 내부에 표현될 때는 byte(8 비트 패턴)를 기본 단위로 표현한다.

이미지 : BMP, JPG
오디오 : WAV, MP3

비디오 : GIF

정수와 실수의 컴퓨터 내부 표현

- 정수는 양의 정수와 음의 정수로 나눌 수 있다.
- 부호화/크기 방법 : 특정 비트를 양/음을 표현하는 데 사용한다.
 - 부호가 없는 양의 정수 표현

▶ 255 -> 11111111

B_{N-1}	B_{N-2}		B_2	B_1	B_0
-----------	-----------	--	-----	-----	-------	-------	-------

- 부호가 있는 정수 표현

▶ 127 -> 01111111, -127 -> 11111111

\pm	B_{N-2}	-	-	-	B_2	B_1	B_0
-------	-----------	---	---	---	-------	-------	-------

정수와 실수의 컴퓨터 내부 표현

- 부호화/크기 방법으로 표현하면 -0 (음수 0)과 $+0$ (양수 0) 두 개의 0이 존재하는 문제가 발생한다. 또한 $(-1) + (+1)$ 을 수행하면 0가 되어야 하지만, 아래처럼 이진 더하기 연산을 수행하면 0이 되지 않는다.

$$\begin{array}{r} 1000\ 0001 \\ 0000\ 0001 \\ \hline 1000\ 0010 \end{array}$$

Handwritten annotations: A blue bracket under the first two numbers is labeled "-1", and a blue bracket under the second number is labeled "+1".

정수와 실수의 컴퓨터 내부 표현

- 2의 보수 방법을 사용하여 음수를 표현하고 연산한다.
- 2의 보수 표현에서는 -0 이 없기 때문에 N -비트에서의 범위는 최소값 -2^{N-1} 에서 $+(2^{N-1}-1)$ 까지이다. 즉, 수의 범위는 $-2^{N-1} \sim (2^{N-1}-1)$ 이다.

예
3비트



$$\begin{array}{rcl} & 011 & +3 \\ + & 101 & -3 \\ \hline & 1000 & 0 \end{array}$$

$$\begin{array}{lcl} 011 & \rightarrow & 3 \\ 101 & \rightarrow & -3 \\ 1000 & \rightarrow & 0 \end{array}$$

$$\begin{array}{l} 011 \\ 101 \\ \hline 1000 \end{array}$$

$$(-2^{N-1} \sim 2^{N-1}-1)$$

비트 패턴	표현되는 값
011	3
010	2
001	1
000	0
100	-4
101	-3
110	-2
111	-1

정수와 실수의 컴퓨터 내부 표현

- 실수 값은 정규화를 통해 의미 있는 자리수부분(크기 부분)과 $\times 10^m$ 부분(지수부분)을 컴퓨터 내부에 표현한다.

실수 부호	지수 부호	$\times B^{\pm E}$	$\pm 0.b_{m-1}b_{m-2}...b_0$
±	±	지수 부분(E)	크기 부분(M)

실수부 | 지수부 | 크기부 | 지수부

- 표현을 수학적으로 나타내면 아래와 같다.
 - $\pm M \times B^{\pm E}$
 - M(Mantissa)은 크기,, B(Base)는 밑, E(Exponent)는 지수이다.
 - $0.0000000314 = 0.314 \times 10^{-7}$

$$= 0.314 \times 10^{-7}$$

정수와 실수의 컴퓨터 내부 표현

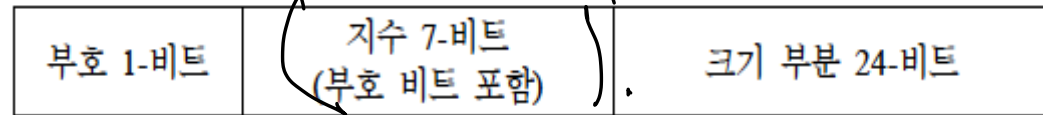
십진수 -25.458는 이진수로 표현하면 -11001.01112×2^0 이다.

정규화 형태는 -0.1100101112×2^5 이다.

$$\begin{array}{r} 25 \\ 212 \dots 1 \\ 216 \dots 0 \\ 213 \dots 1 \end{array} \quad -11001.01112 \times 2^0 = -0.1100101112 \times 2^5$$

실수 형태인 32-비트로 표현하면 아래와 같다.

부호 비트 : 1

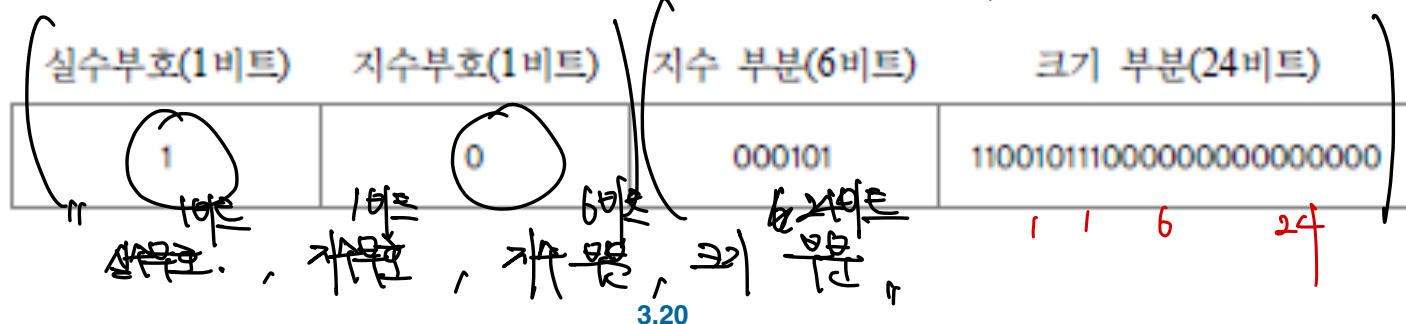


지수 비트 : 0000101

크기 비트 : 110010111000000000000000

최종으로 -25.458는 실수를 위한 지수 표현으로 아래와 같이 32-비트로 변환한다.

1 0 000101 110010111000000000000000



10.4 문자의 표현

문자의 표현

□ 문자를 이진수 비트열로 표현

□ 영어 알파벳이나 기호를 이 때 할당한 수를 코드라 한다.

□ 8-비트: 10000001을 기호 '!'로 해석

□ 부호 없는 정수 129, 부호와 크기 음수 1(-1), 2의 보수 음수(-128),
111110

□ ASCII 코드

□ 대부분 컴퓨터에서 문자를 취급하는 데 사용하는 코드이다.

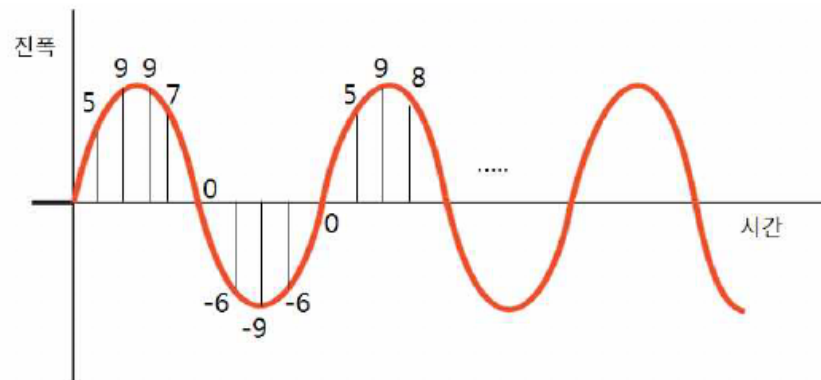
문자의 표현

- 유니코드(UNICODE)코드는 16-비트를 사용하고, 65,536(2^{16})의 문자나 기호를 표현 할 수 있다는 것을 의미한다.(305페이지 2-3줄 삭제)
- 전 세계 모든 문자 표현하는 산업표준 코드표 (2020.3.10 기준, 143,859 문자코드 표준화)
- 컴퓨터 표현방법(USC, UTF 인코딩)
 - UTF 인코딩, 아래를 포함하는 가변길이 인코딩
 - ▶ utf-1, utf-7, utf-8, utf-16, utf-32, utf-EBCDIC

9.4.3 소리의 표현

9.4.3 소리의 표현

- 소리는 인간이 주위에서 청각적으로 접할 수 있는 모든 자연의 아날로그 신호를 말한다.
- 아날로그 소리를 파형으로 보고 단순하게 표현한 것으로, 이 신호를 디지털화한다. 그림 10.5에서 파형의 진폭(높이)은 소리의 크기를 말하고, 주기(T)는 온전한 한 사이클의 파형을 만드는 데 걸리는 시간을 말한다. 주파수(F)는 단위 시간당 사이클의 총 수(사이클/초)로 헤르츠라 하며 $F = 1/T$ 이다.



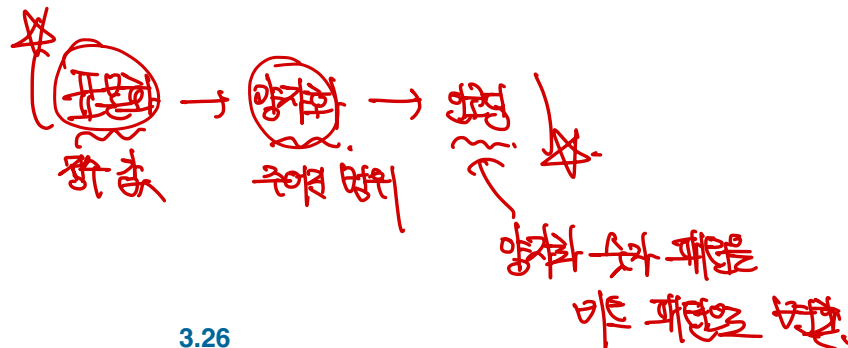
(a) 아날로그 소리 신호의 표본화

|| 그림 10.5 || 소리의 디지털화 예

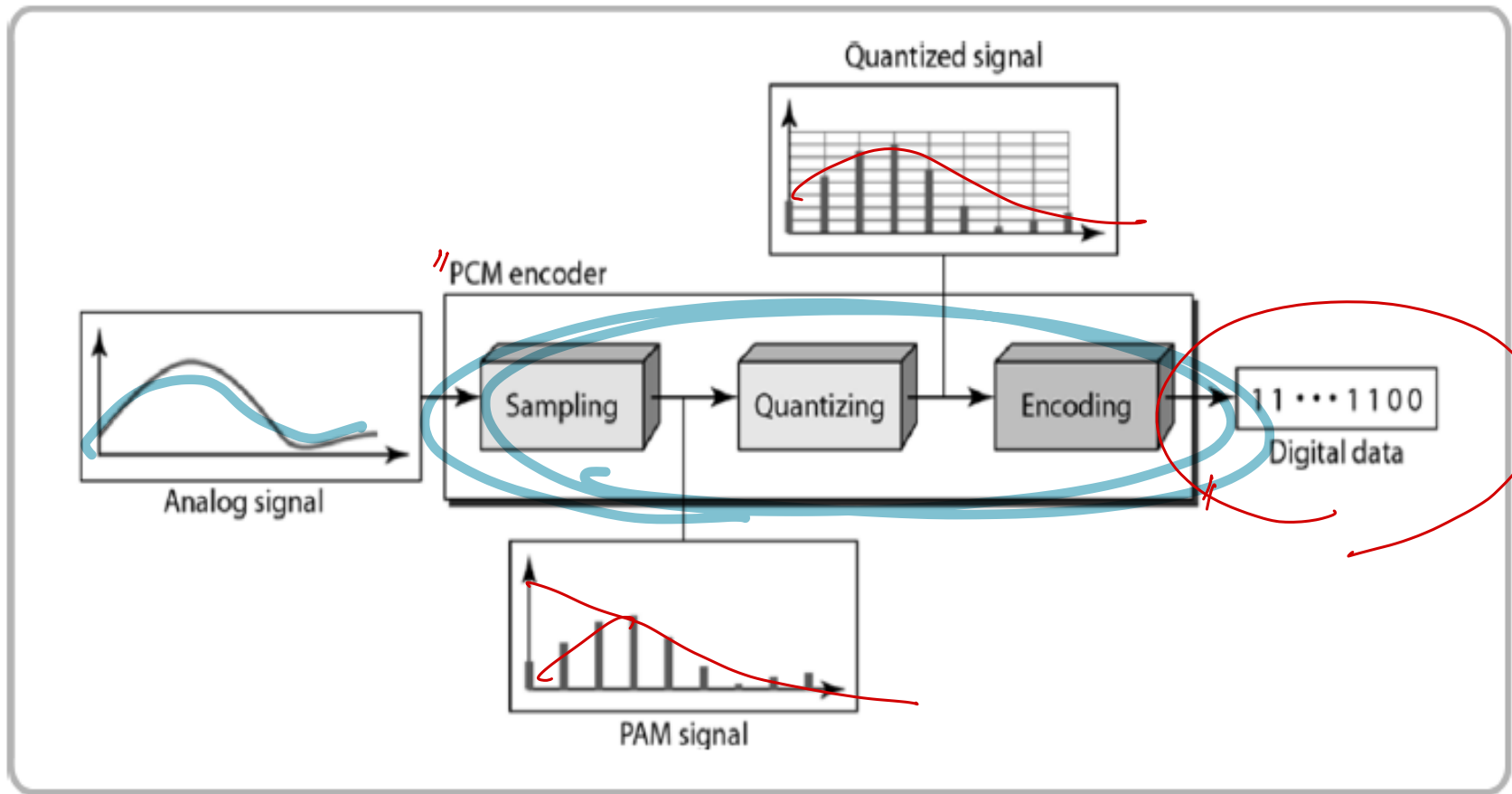
9.4.3 소리의 표현

- 파형의 소리를 컴퓨터에 저장하기 위해 어떻게 디지털화 하는지 살펴보자. ^{저장하기 위해 원근 디지털화 하는 것} 디지털화 과정은 아래 순서의 세 단계로 이루어진다.

- 표본화: 아날로그 소리를 일정 단위 시간 간격으로 쪼개어, 신호의 진폭을 측정하여, 해당하는 정수 값을 저장한다. 그러므로 파형의 진폭이 표본화된 숫자의 연속으로 컴퓨터안에 표현된다.
- 양자화: 샘플링된 연속된 정수 값을 미리 정해진 숫자 단위(범위)로 변환한다.
- 인코딩: 양자화를 거쳐서 생성된 숫자 패턴을 비트 패턴으로 변환한다.



"음원화 → 양자화 → 인코딩"



[그림 9.15] 오디오의 비트패턴 변환 과정

9.4.3 소리의 표현

- 표본화 비율은 파형에서 초당 추출한 표본 수이다.
 - 1초 표본 8,000개를 만들고, 8-비트 패턴으로 변환하면
 - $8,000/\text{초} \times 8\text{-비트} = 64,000 \text{ 비트} = 8,000 \text{ 바이트} = 8\text{KB}$ 의 음원
- CD음원 규격 16bit/44.1Khz *A 44100개의 샘플링 신호*
 - 1초에 44100 개의 샘플링 신호/ 각 신호는 16 bit로 코드화
 - 44100 개 * 16 비트 / 1초 *44100x16비트 / 초*
- 고음질 음원 24bit, 192KHz
 - ^c192000 * 24 bit/second *r*

9.4.3 소리의 표현

- 다양한 압축 포맷 등장 : (WAV, MP3, WMA, AIFF, FLAC)
- WMA AIFF FLAC.

WAV MP3

(WMA
AIFF
FLAC


9.4.2 영상의 표현

- 사진이나 영상도 소리와 마찬가지로 이진 값으로 컴퓨터 안에 저장한다. 영상은 (명암과 색의 연속)이다.
- 영상에 규칙적인 간격으로 놓인 각점의 명암 값을 측정하는 것으로 이루어진다. 이러한 점들을 화소(픽셀, Pixel)라 하는데, 화소를 더 많이 사용하면 영상을 인코딩하는데 더 정확해진다.
but 저장량 ↑
- 영상을 저장하려면 아래와 같이 세 방법을 사용할 수 있다.
 - 흑백 영상으로 저장
 - 회색 영상으로 저장
 - 컬러 영상으로 저장

9.4.2 그림의 표현

□ 흑백 그림으로 저장

□ 흑백으로 그림을 저장하면 각 화소마다 1-비트만 할당하면 된다.
(0은 흰색, 1은 검정색)



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	0	1	1	0	0
0	0	0	0	1	0	1	0	1	0	0	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	1	0	1
0	0	0	0	1	0	1	1	1	1	1	1	1	0	0
0	0	0	1	0	0	1	0	0	1	0	1	0	0	0
0	0	0	0	1	0	1	0	0	1	0	1	0	0	0

흑백 그림

■ 그림 10.6 ■ 흑백으로 저장한 사진

□ 화소가 $100,000 \times 120,000$ 인 영상은 1.5GB의 저장 공간을 차지한다.

9.4.2 그림의 표현

□ 회색 그림으로 저장

- 회색의 표현에 8-비트를 사용하면 ^{다양한 회색} 256개의 회색을 만들 수 있다.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	120	0	0	0	110	0	0	0	0	255	255	255	255	255	255
105	105	105	105	125	0	120	105	0	0						
0		0	0	0	118	0	0	0	0						
0	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0
120	120	120	120	120	118	105	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	105				0	0	0	0	0	0
0	125	125	125	0	0	105	120	120	120	0	105	0	0	0	0
0	0	125	0	0	0	105	0	0	0	0	105	0	0	0	0
0	0	0	0	0	0	105	0	0	0	0	105	0	0	0	0
0	0	115	115	115	115	115	115	115	115	0	105	0	0	0	0
0	0	115	115	115	115	115	115	115	115	120	120	120	120	0	0
0	0	115	115	115	115	115	115	115	115	120	120	120	120	0	0

■ 그림 10.7 ■ 회색으로 저장한 사진

$$1.5 \times 8$$

$$= 12 \text{ GB}$$

$$1.5 \text{ GB} \times 8$$

- 위의 화소가 100,000 × 120,000인 영상을 회색으로 저장하면 12 G바이트(12GB) 저장 공간을 차지한다. 즉, 흑백 영상일 때 보다 8배 저장 공간을 더 차지한다.

9.4.2 그림의 표현

□ 컬러 영상으로 저장

- 영상을 컬러로 디지털로 표현하고 저장하려면 각 화소마다 정보를 더 많이 주어야 하는데, 보통 방법은 빨강, 녹색, 파랑의 정보를 제공하는 기술이다. 이 기술을 RGB(Red, Green, Blue) 인코딩 기법이라 한다. (보통 한 색상에 8비트씩 24비트 사용)
- 순수 빨강과 순수 파랑을 섞으면 ‘보라’가 된다. 즉, RGB = (255, 0, 255)로 인코딩 된다. 마찬가지로 붉은 분홍은 RGB = (255, 105, 180)로 인코딩 된다.
- 화소가 $100,000 \times 120,000$ 인 영상을 컬러영상으로 저장하면 36G바이트(36GB) 저장 공간을 차지한다.

$$1.5 \times 8 \times 3 = 36 \text{ GB}$$

9.4.2 그림의 표현

- 사진 한 장을 전송하기 위하여 많은 데이터를 전송하고 시간도 많이 소요된다. 이런 문제를 해결하기 위해 영상을 압축하여 표현하고 저장한다. 압축 방식에 따라 다양한 파일 확장명이 있다.
→ 이미지, JPG, PMP

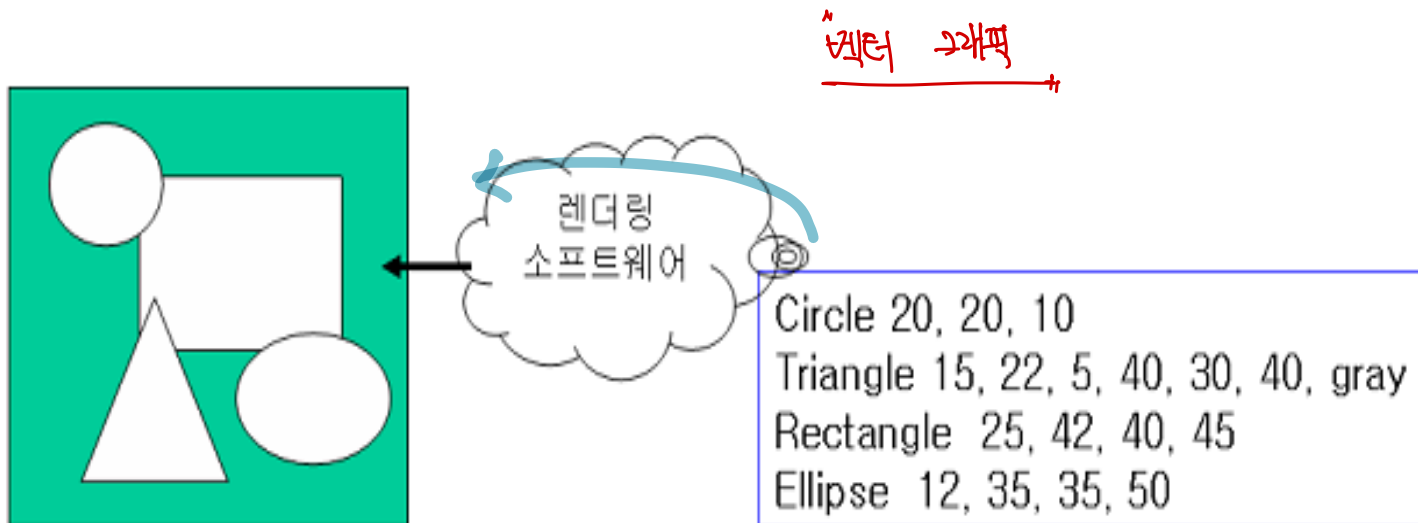
↑ 위에서 설명한 것.

- 영상을 표현하는 이런 비트맵 방식 외에도 객체를 한 단위로 표현하는 벡터 그래픽 방식도 있다. 이 기술은 영상을 여러 기하학적인 다각형 '객체'의 모임으로 보고 각 객체를 선이나 도형을 사용한 수식으로 표현을 한다.
 - 객체 위주의 편집을 지원하기 때문에 그림의 변형이나 파괴가 없다.

벡터 그래픽

■ 벡터 이미지의 디스플레이

- 벡터로 표현한 각 개체에 대한 데이터를 렌더링(rendering) 소프트웨어가 해석하여 컴퓨터 화면의 적당한 위치에 나타나게 한다

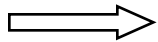


9.4.4 영상의 표현

비트 → BMP, JPG
이진 → WAV, MP3
화상 → MPEG, MOV

동영상

비디오(Video)란 ?



시간의 흐름에 따라 일련의 이미지 혹은 프레임을 출력

⇨ A stream of frames in time (예: 1초에 30장의 프레임 출력)

비디오 압축기술로서 MPEG 등이 있음 (I, P, B 프레임 사용)

(MPEG: Moving Picture Expert Group: ISO/IEC JTC1/SC29)

IPB.

□ UHD 동영상(비압축때)

□ 픽셀 수 : 3840 x 2160 픽셀

□ 정지화상크기 : 3840 x 2160 x 24bit(RGB) = 199,065,600 bit

□ 동영상 비트수 : 3840 x 2160 x 24 x 30(초당프레임수)

5,971,968,000 비트/초

$$3840 \times 2160 \times 24 \times 30 = 5,971,968,000 \text{ bit/sec}$$

3.37 픽셀 8x3 24bit 초당 프레임수

MPEG

■ MPEG: 비디오 압축(표현) 기술

- MPEG-1, 2, 4, 7, 21

■ MPEG 동영상 압축기술

- MPEG-1
 - CD-ROM 저장용 기술(1.5 Mbps 데이터 속도)
- MPEG-2
 - 원격 화상회의나 DVD 용 기술 (3 ~ 6 Mbps 데이터 속도)
- MPEG-3
 - 오디오 압축기술(MPEG-2에 합쳐짐, MP3라함)
- MPEG-4
 - 객체 중심의 씬 (scene) 구성, 씬에 상호작용이 가능한 기술

MPEG

■ MPEG frames

- I-frame (intracoded): 원래 이미지
- P-frame (predicted): I frame과 바뀔 부분
- B-frame (bi-directional): I, P frame과의 차이 부분 (시간차 고려)

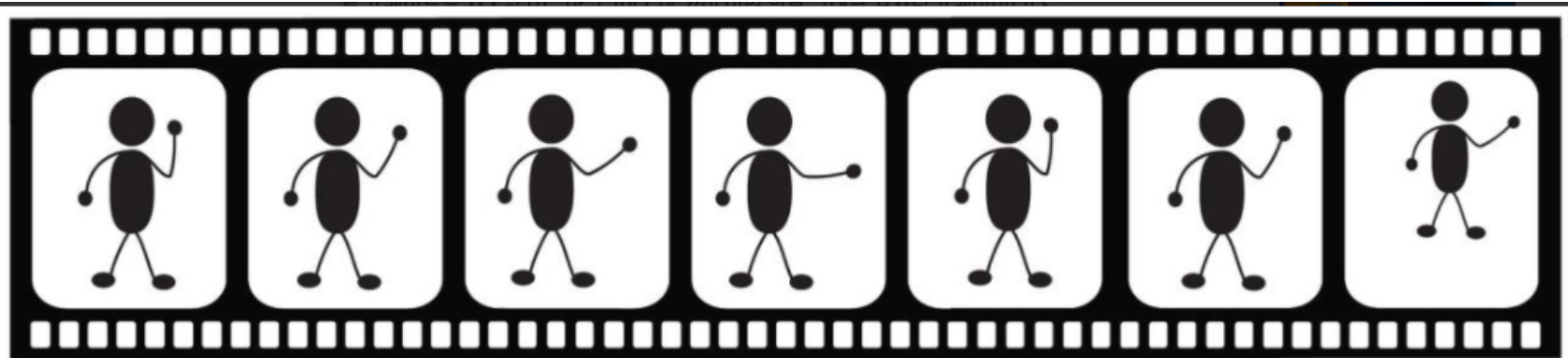
■ GOP (Group of Pictures) = 15 frames

- GOP = IBB PBB PBB PBB PBB
- 예) 2 GOPs per second: 30 frames per second

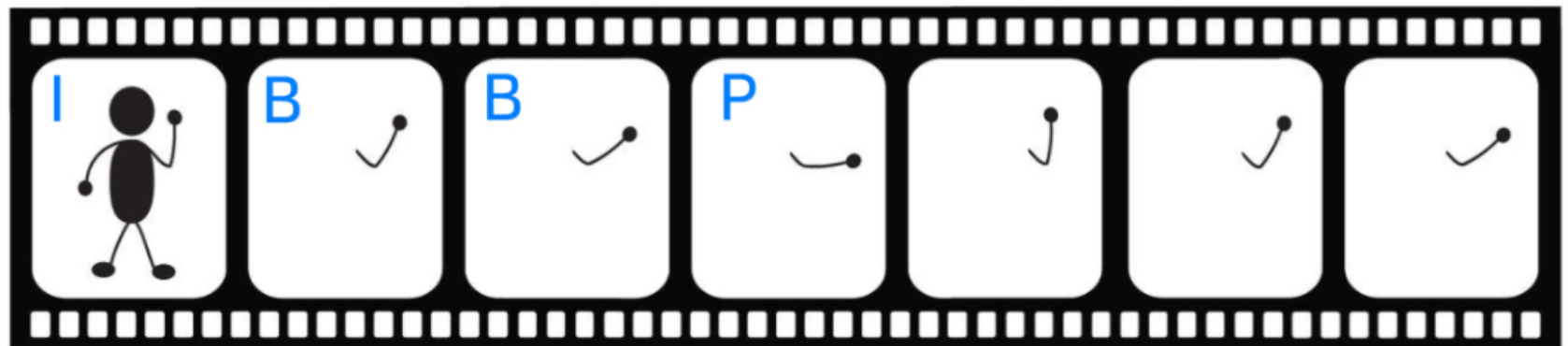
//

I BB PBB PBB PBB , ↓

GOP.



Intraframe Compression
Every frame is encoded Individually



Interframe Compression
Only the differences between frames are encoded for each group of frames



I frame



P frame



B frame

- Red Intra
- Green Inter prediction
- Blue Biprediction
- Yellow Direct mode

- Abundance of Direct macroblock in B frames

Group Of Pictures (GOP)

