

컴개실 PPT HW2

컴퓨터 내부에서 사진, 음성, 영상의 표현과 처리

13조 조강현 백승우 박재완2 배수민

|

컴퓨터에서 사진의 표현

I. 컴퓨터에서 사진의 표현



래스터 그래픽 (Raster Graphics)
JPEG PNG GIF 등의 포맷



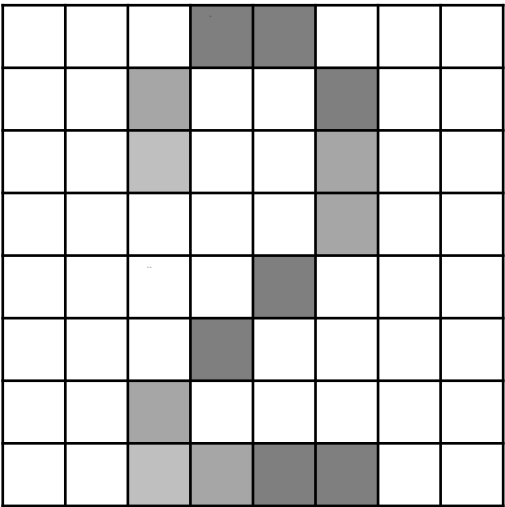
벡터 그래픽 (Vector Graphics)
AI SVG 등의 포맷

01-1 Raster Graphics – 흑백사진 표현



아날로그 이미지
(실제 물체)

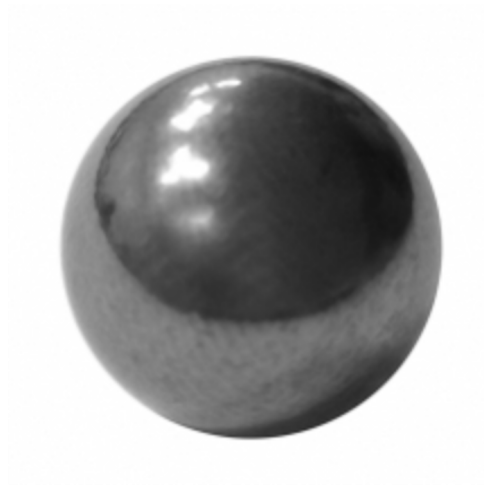
표본화 (Sampling)
양자화 (Quantization)



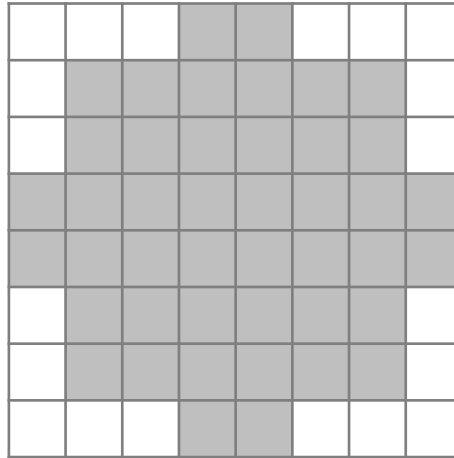
디지털 이미지

01-1 Raster Graphics – 흑백사진 표현

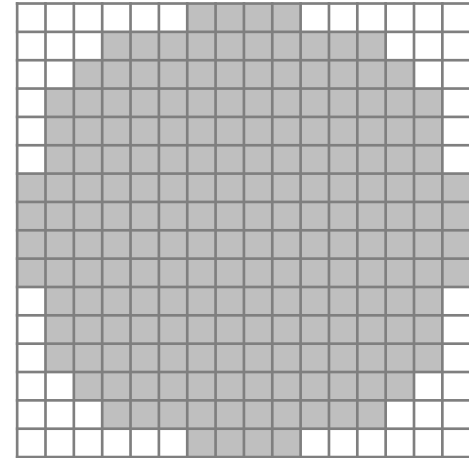
표본화 (Sampling) : 아날로그 이미지의 연속적인 공간을 불연속적인 공간으로 변환하는 과정



아날로그 이미지



8 X 8 pixel



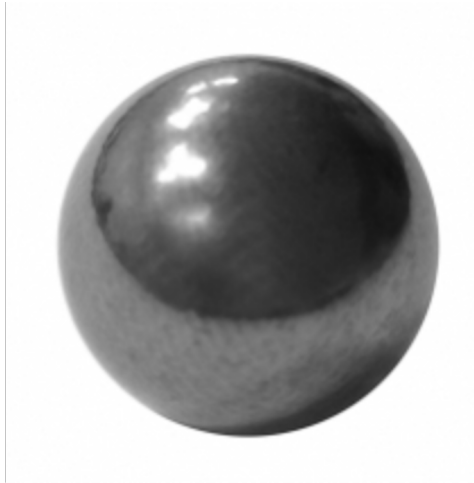
16 X 16 pixel

디지털 이미지의 최소 표현 단위를 화소(pixel)라 하고, 이미지의 화소 수를 해상도(Resolution)라 한다.

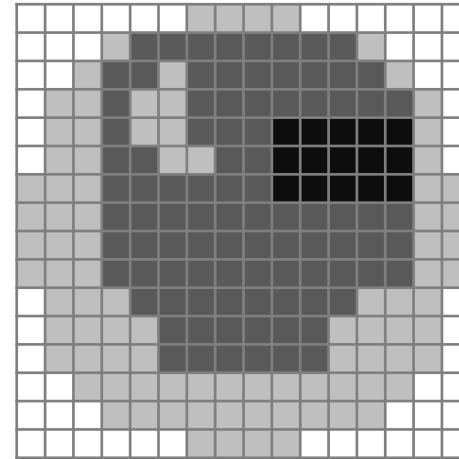
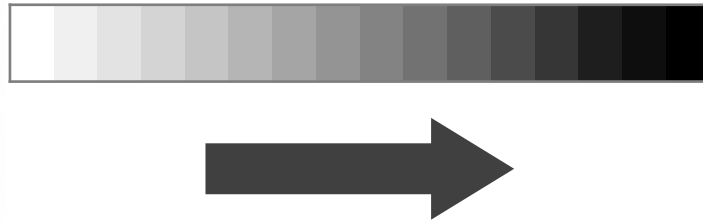
Raster Graphics에서 컴퓨터는 디지털 이미지를 화소의 집합으로 저장한다.

01-1 Raster Graphics – 흑백사진 표현

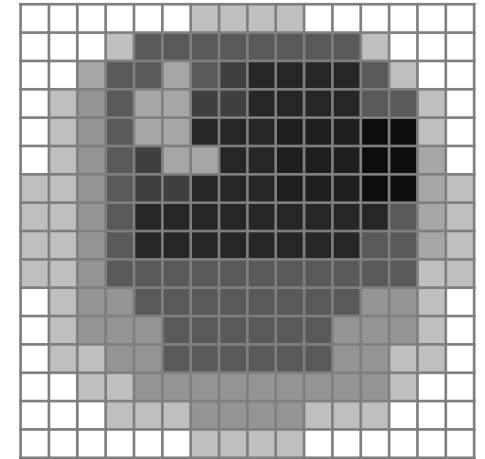
양자화 (Quantization) : 각 화소별 연속적인 밝기 데이터를 불연속적인 데이터로 변환하는 과정



아날로그 이미지



2 비트 (4 레벨) 양자화



4 비트 (16 레벨) 양자화

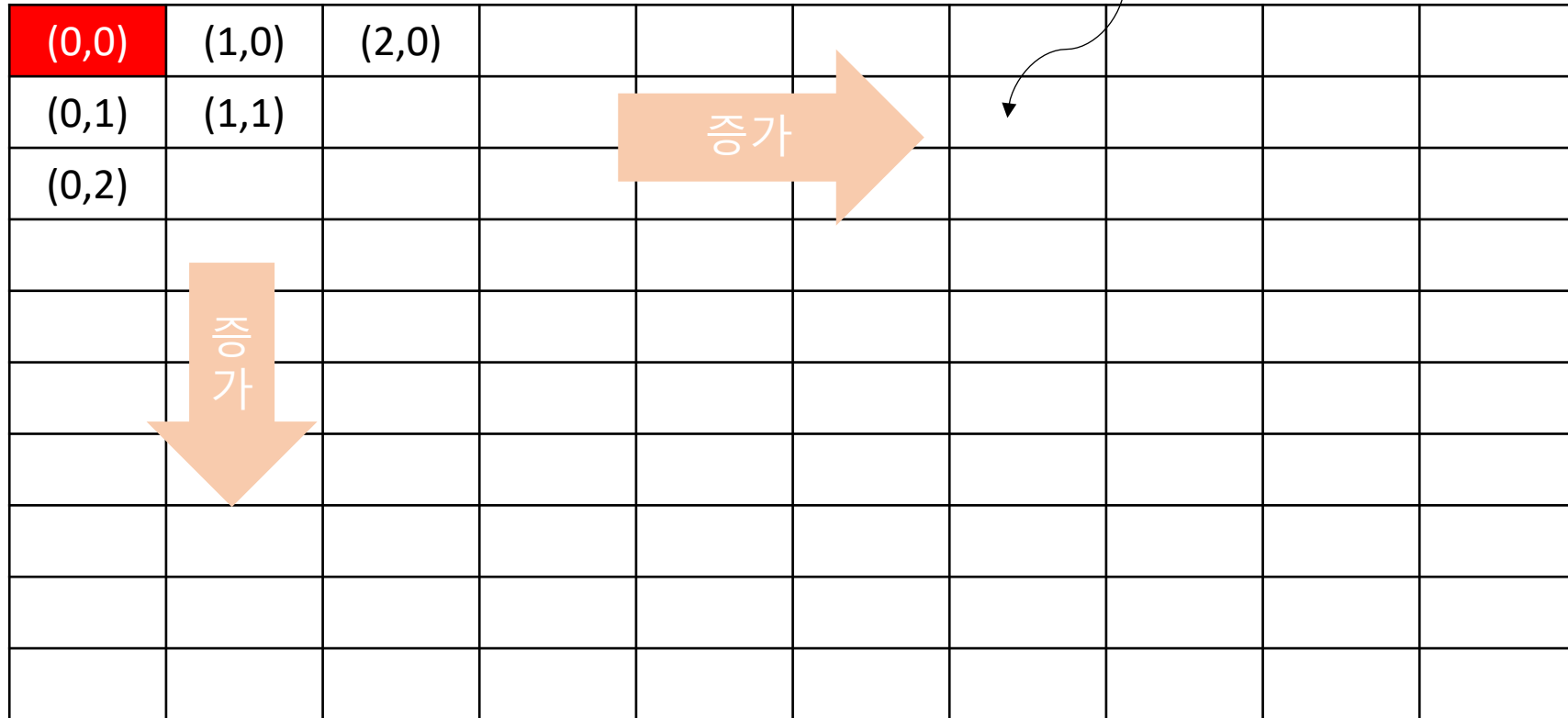
각 화소별 밝기를 불연속적인 단계로 구분하여 숫자로 나타낸다.

G 비트 양자화는 2^G 단계의 색상을 표현할 수 있다. 일반적으로 흑백사진은 8 비트 (256 레벨) 양자화로 표현한다.

01-1 Raster Graphics - 컬러사진 표현

화면 좌표계

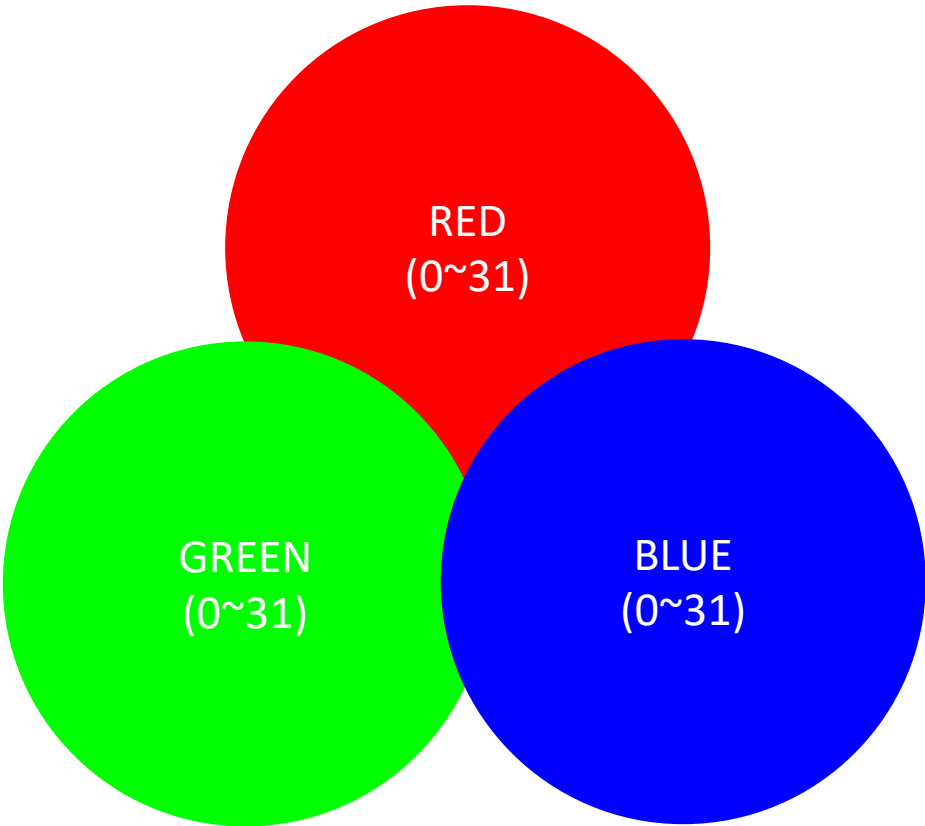
화소(픽셀)



01-1 Raster Graphics – 컬러사진 표현

RGB표현법

하이컬러 (15bit)



총 32,768 가지 색 표현

More Precise!!

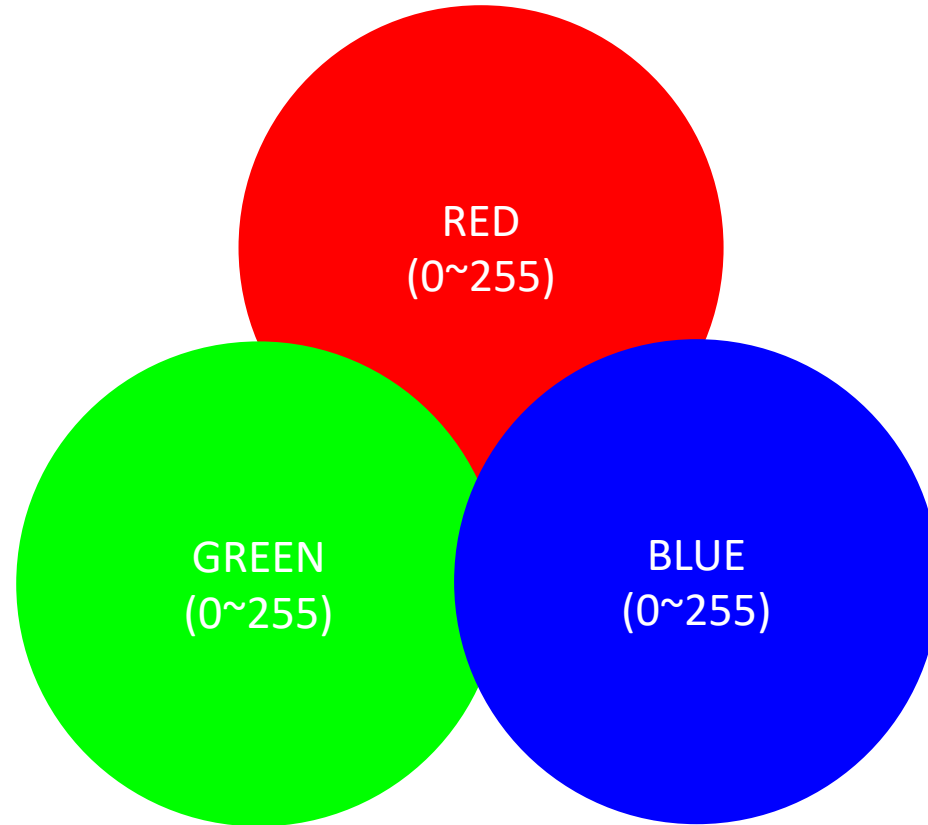
트루컬러 (24bit)

01-1 Raster Graphics – 컬러사진 표현

RGB표현법

트루컬러 (24bit)

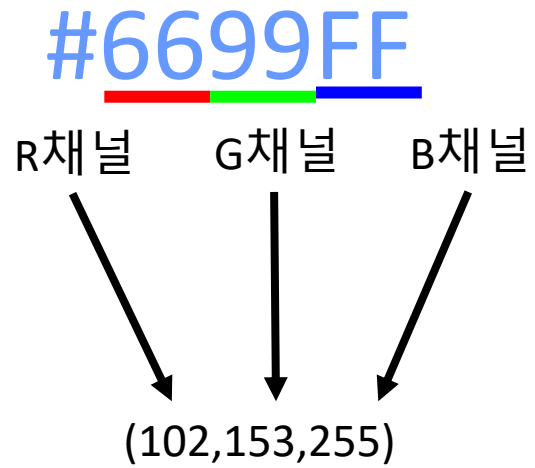
예시) #6699FF



총 16,777,216 가지 색 표현

01-1 Raster Graphics – 컬러사진 표현

표현 방법



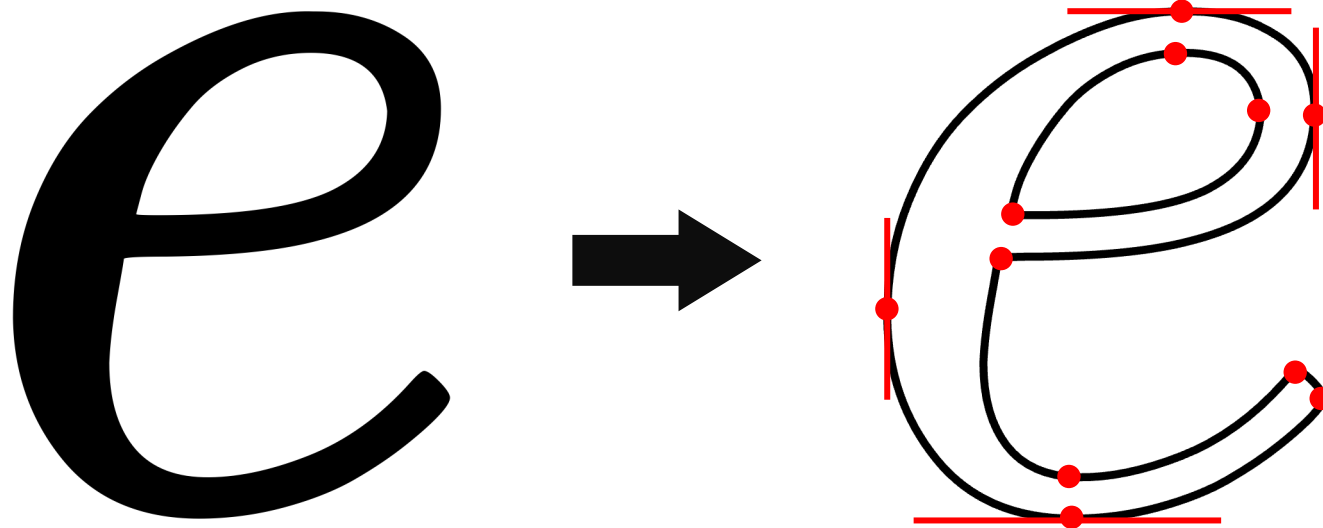
01-1 Raster Graphics – 이미지 파일의 저장

디지털화된 이미지는 다양한 형태로 저장된다.



무압축 포맷	손실 압축 포맷	무손실 압축 포맷
압축을 하지 않고 원본 그대로 저장하는 방식	인간이 지각하기 힘든 범위의 이미지 데이터를 버리고 데이터 크기를 줄여 압축하는 방식	반복 표현되는 데이터를 최대한 줄여 원본과 같은 정보량을 유지하며 압축하는 방식
BMP	GIF, PNG	JPEG

01-2 Vector Graphics



수학적인 방식으로 흑백 혹은 컬러 이미지를 표현하여 저장한다.

정점, 두 개의 점을 이은 선, 세 개 이상의 선이 모인 면 등이 각각 두께, 색상, 곡률 등의 값을 가져 그래픽을 표현한다.

II

컴퓨터에서 오디오의 표현

02 컴퓨터 내부에서의 오디오 데이터 처리

오디오(소리) ∈ 파동

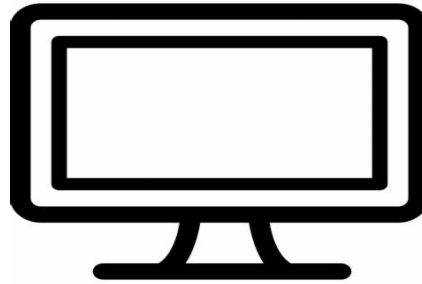
아날로그 신호



How?



입력



How?



출력



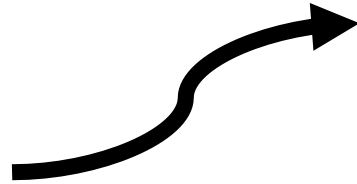
컴퓨터 : 디지털 신호만 이해

Sampling

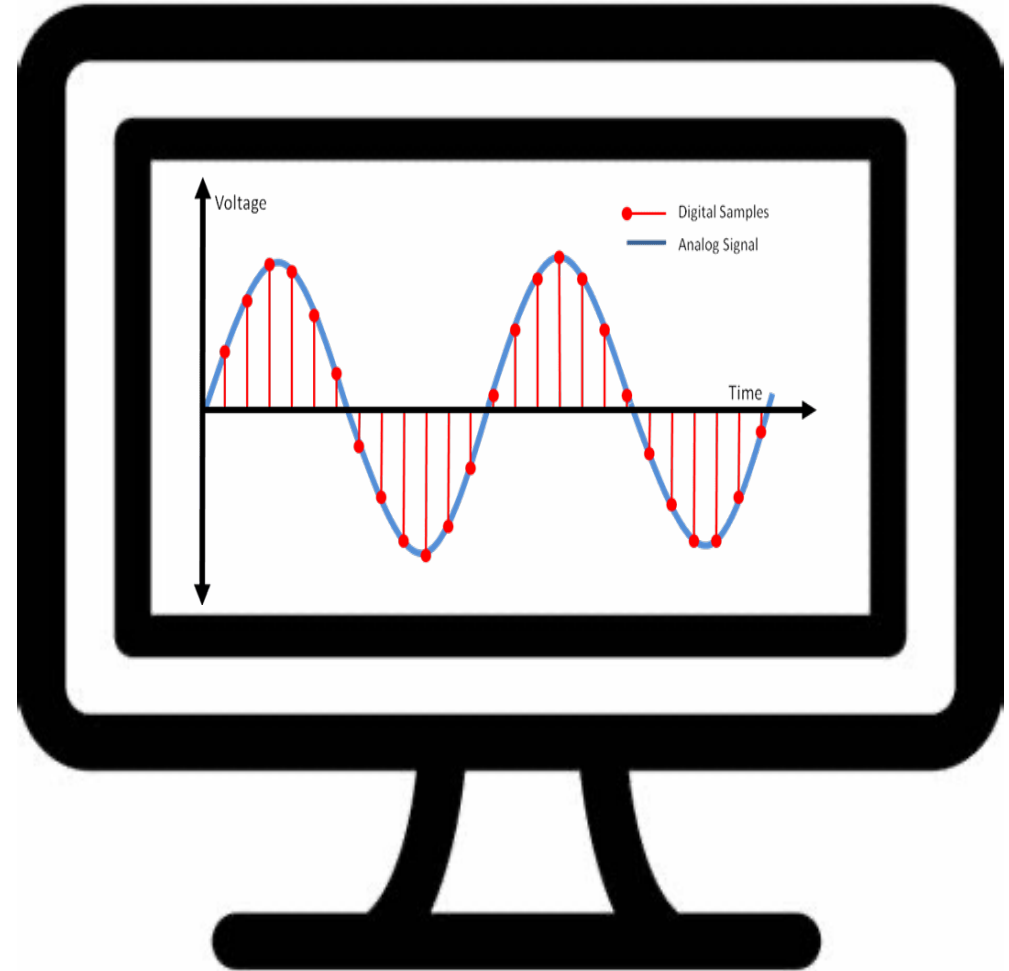
Analog → Digital



Audio
Data

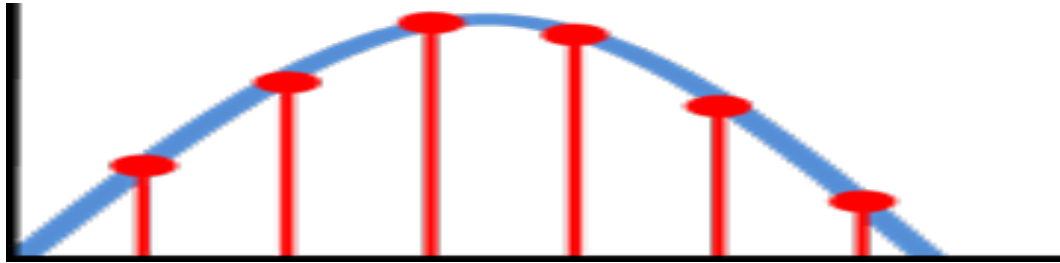


연속적인 파동함수의
그래프를 시간 단위로
쪼개어 그때의 전압을
읽는다. (~구분구적법)



02 컴퓨터 내부에서의 오디오 데이터 처리

Sample Rate \propto 해상도(음질)



**Sample
Rate**

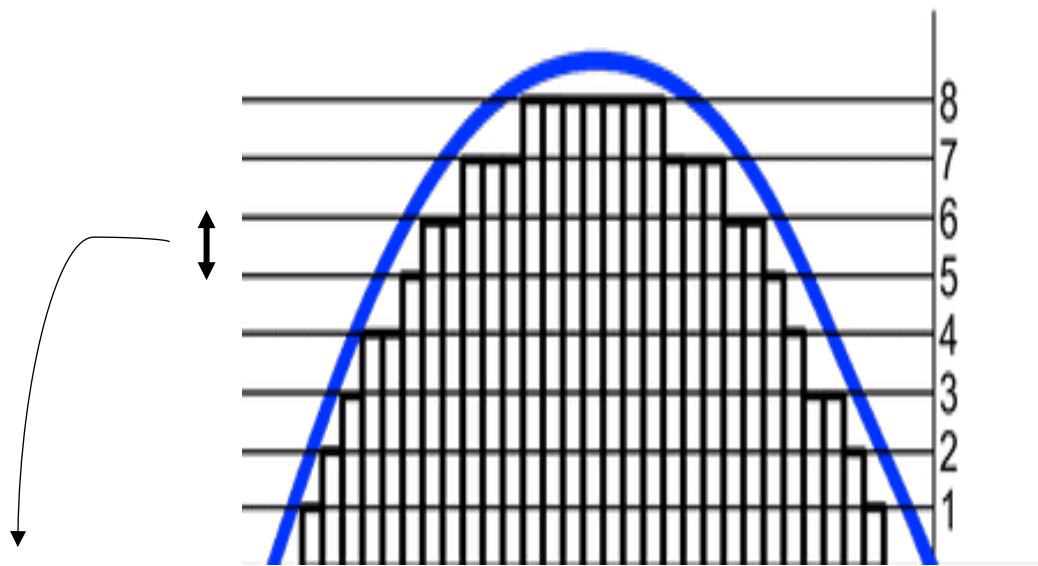
: 신호를 측정하는
빈도



Sample Rate 가 크다.

- == Sampling 시 측정 빈도 ↑
- == Data의 연속성 ↑
- == 실제 Analog Data 와의 유사성 ↑.
- == 실제 소리에 가깝다.
- == **음질이 좋다.**

02 컴퓨터 내부에서의 오디오 데이터 처리

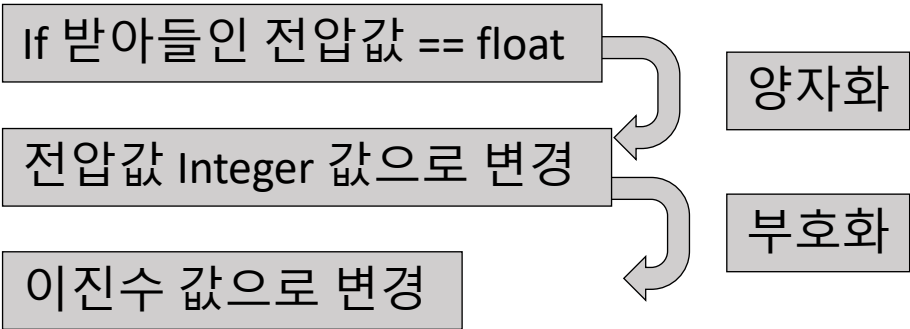


Bit Rate : 신호의 높낮이를 나누는 칸수

24비트면 양의 값으로 12비트만큼, 음의 값으로 12비트만큼 분할 가능

Bit Rate 가 크다.

- == 전압값으로 사용할 비트 수 ↑
- == 전압값의 섬세함 ↑



오디오 파일 포맷 : 컴퓨터에서 디지털 오디오 데이터를 저장하기 위한 파일 포맷



비압축 포맷

WAV

손실 압축 포맷

MP3

비손실 압축 포맷

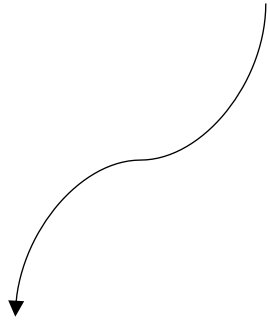
FLAC

02 컴퓨터 내부에서의 오디오 데이터 처리

CD

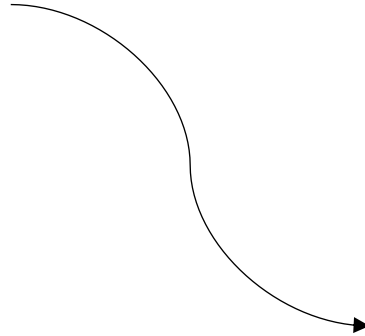
Sample Rate: 44.1kHz

Bit Rate: 16 bit



16자리 이진법 사용가능

== (2^{16}) 개의 값 사용 가능



·인간 가청주파수 : 20~20000Hz

· Sample Rate > 2*(인간 가청주파수)

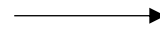
(파동은 압축 -> 팽창 -> 압축 과정을 거치므로 돌아오는 진동까지 컴퓨터가 해석해야 하므로 Sample Rate 가 44.1kHz가 되어야 함.)

02 컴퓨터 내부에서의 오디오 데이터 처리

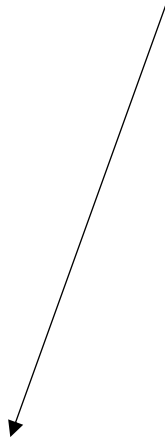
WAV

Sample Rate: 44.1kHz

Bit Rate: 16 bit



비압축 포맷이기 때문 !



CD와 동일한 수준의
음질을 그대로 추출



받아들인 오디오
데이터를 그대로 저장

02 컴퓨터 내부에서의 오디오 데이터 처리

MP3

Sample Rate: 무의미

Bit Rate: 파일마다 다름

압축 효율 결정

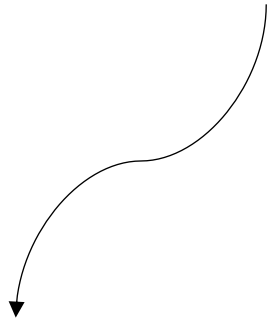
MP3는 손실 오디오 압축 포맷
= 손실 압축 기법 in JPEG 이미지 파일

데이터 크기 ↓

음질 ↓

성능 ↓

FLAC : Free Lossless Audio Codec



MP3와 달리 오디오 스트림에 손실 발생 X

오디오 소스를 온전한 모습으로 보존

||

인코드 후 디코드 -> 똑같은 오디오 데이터 획득 !

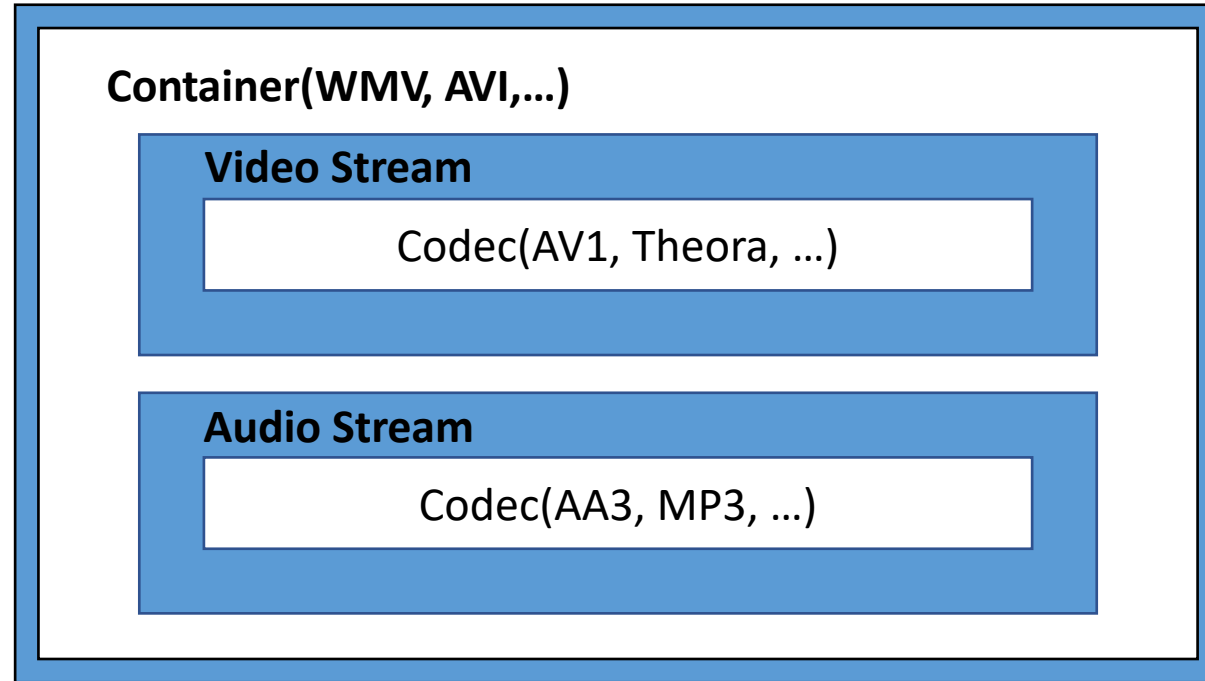
PS. 예전에는 FLAC 파일을 지원하는 기기가 많이 없었음. -> 최근엔 많은 기기에서 지원

III

컴퓨터에서 동영상의 표현

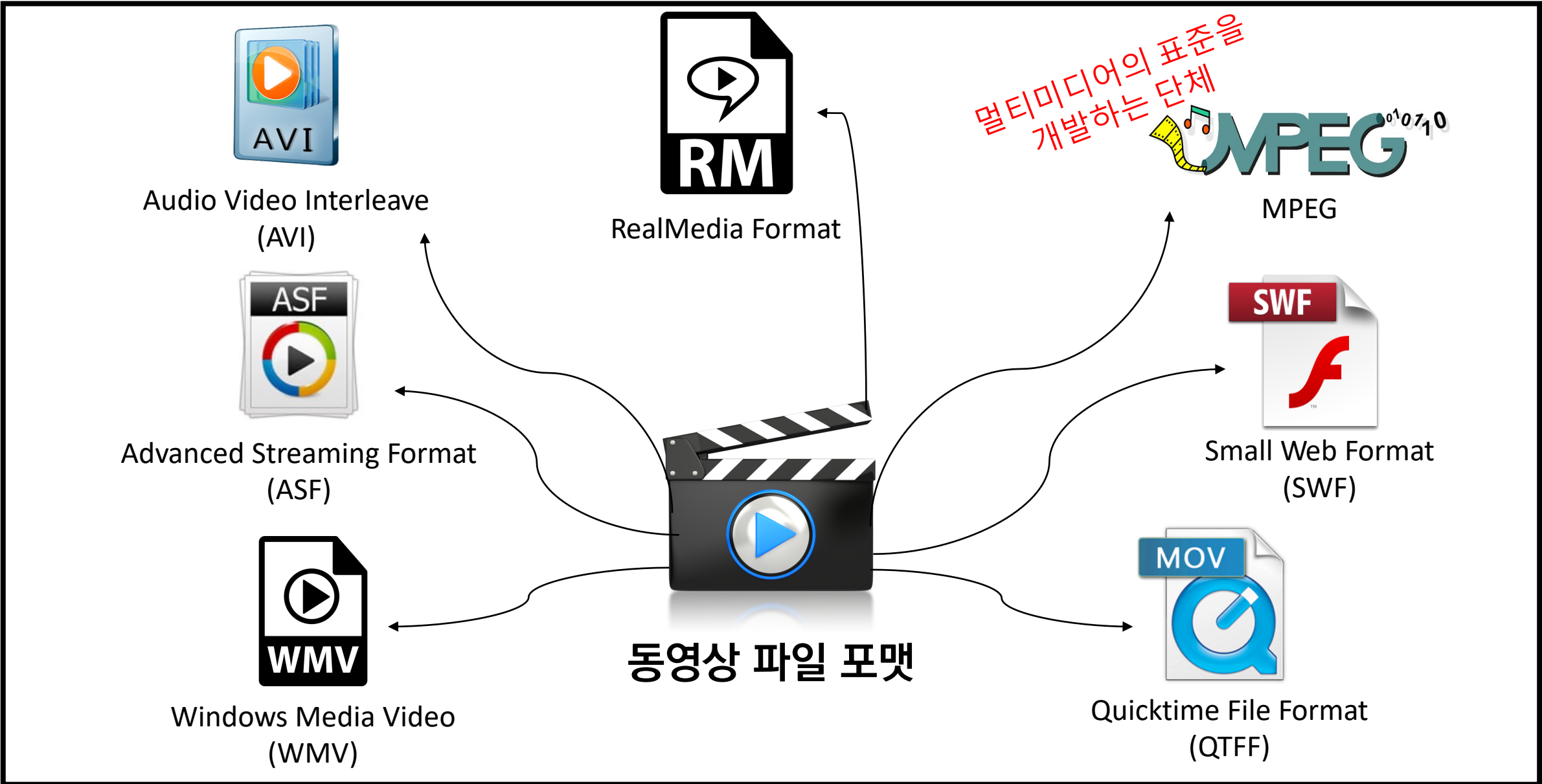
03 컴퓨터에서 동영상의 표현 – 컨테이너 포맷

동영상 파일(Container Structure)



- Stream : 연속성을 가지는 데이터(Video Stream = 비디오 데이터)
- 동영상의 확장자는 동영상의 정보가 담기는 Container의 종류
- Container에는 Codec으로 가공된 Video/Audio Stream이 하나 이상 존재

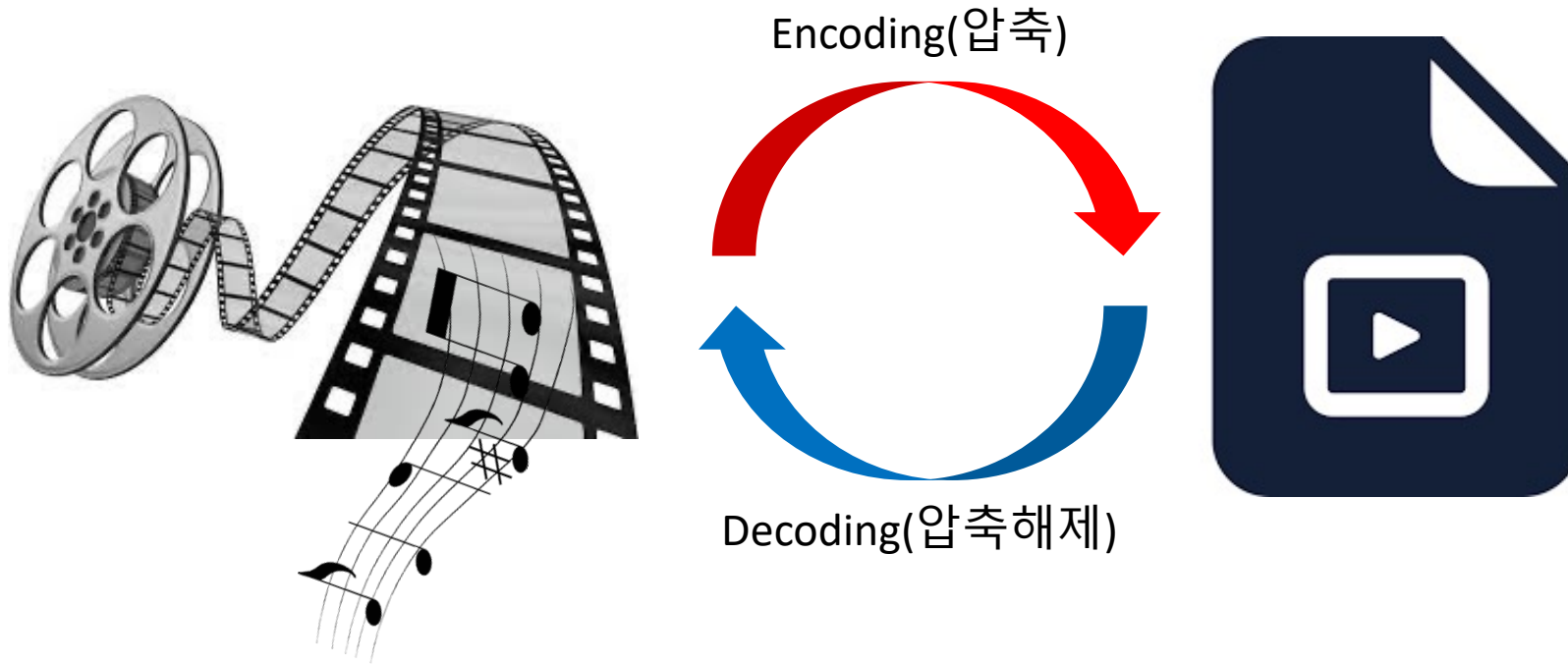
03 컴퓨터에서 동영상의 표현 - 다양한 동영상 파일 포맷



03 컴퓨터에서 동영상의 표현 - 코덱(CODEC)

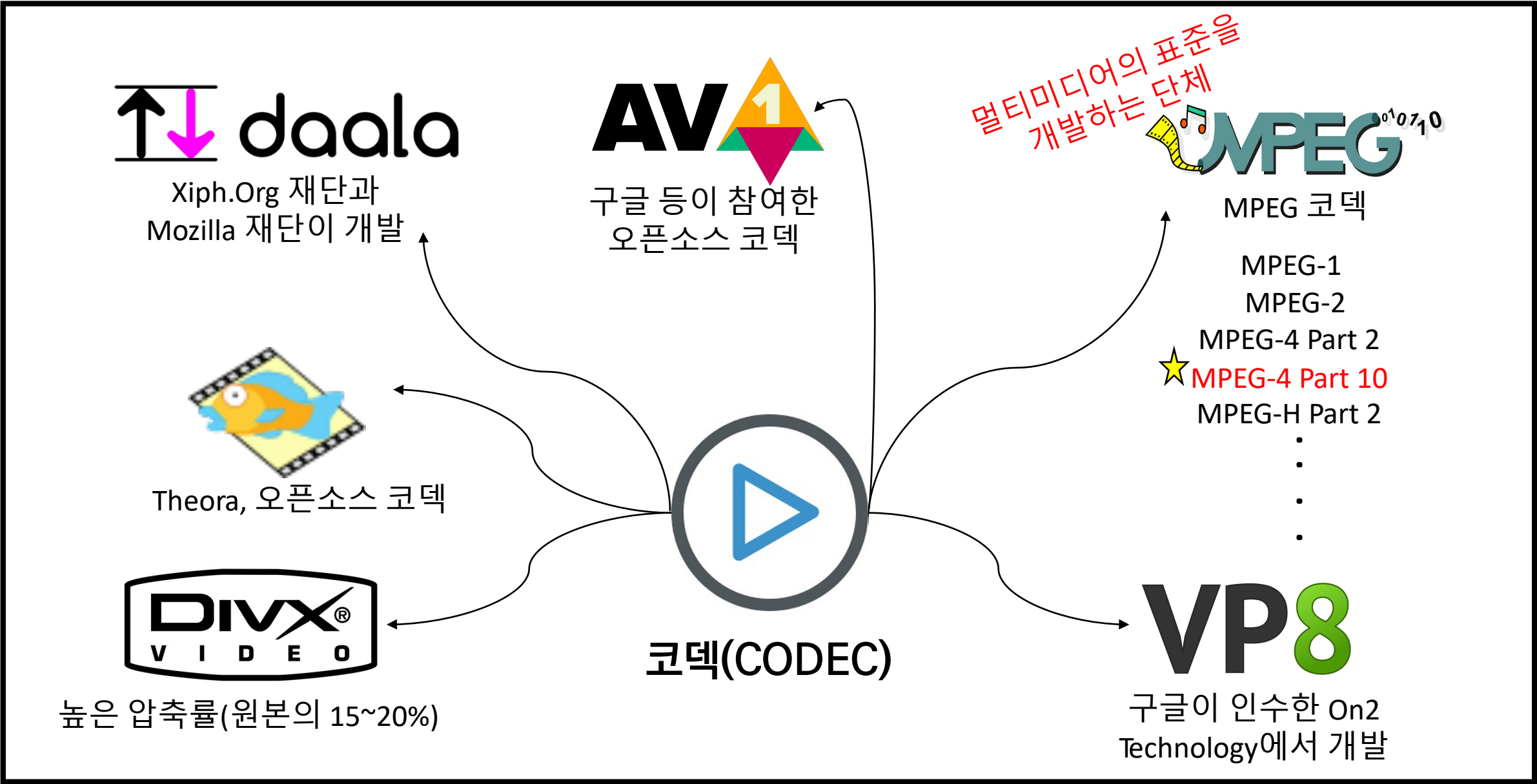
비디오/오디오 데이터를 인코딩/디코딩 할 수 있는 하드웨어/소프트웨어

영상 처리에선 주로 압축/압축해제



***CODEC** = (en)**CO**der-**DE**Coder

03 컴퓨터에서 동영상의 표현 - 코덱의 종류



03 컴퓨터에서 동영상의 표현 - 압축

크게 두 종류로 나뉜다.

- 원본 데이터 손상 없이 압축하는 Lossless 방식
- 일부 데이터를 생략함으로써 파일 크기를 줄이는 Lossy 방식

Run Length Encoding: 무손실 압축

예시) **RRRRBBBRRRRRRRBBRBBBBBBBRRR**  **4R3B6R1B2R7B3R**

중복되는 데이터 부분을 중복되는 횟수와 중복되는 값으로 나타냄

- ✓ 단순한 영상에선 성능 Good
- ✓ 복잡한 영상(실제 영상)에선 성능 Bad

03 컴퓨터에서 동영상의 표현 - 압축

Vector Quantization: 손실 압축

예시)

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

Pattern1: (1, 0, 0, 0)

Pattern2: (1, 1, 1, 1)

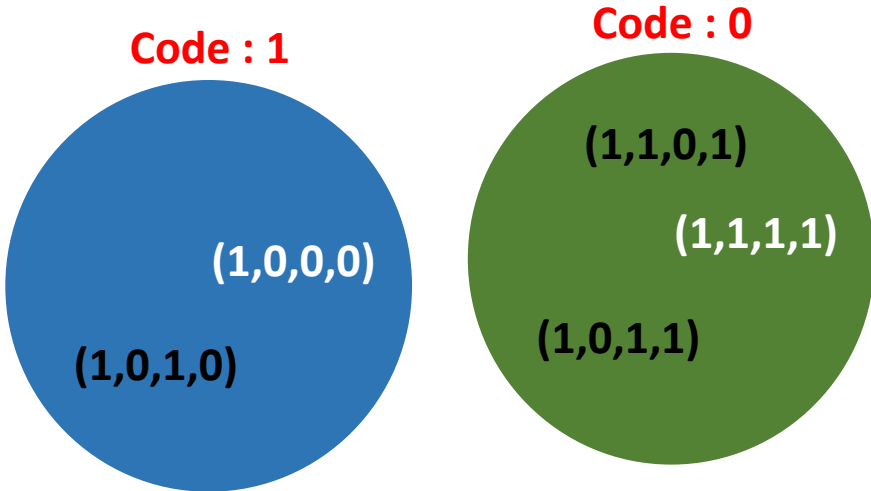
이런 방식으로 2x2 16개에 대해 패턴을 구한다.

(1,0,0,0)	(1,1,1,1)
(1,1,1,1)	(1,0,0,0)
(1,1,1,1)	(1,0,0,0)
(1,0,0,0)	(1,1,1,1)
(1,0,0,0)	(1,0,1,1)
(1,0,0,0)	(1,1,0,1)
(1,1,1,1)	(1,0,0,0)
(1,1,1,1)	(1,0,1,0)

03 컴퓨터에서 동영상의 표현 - 압축

Vector Quantization: 손실 압축

비슷한 패턴끼리 묶어준 다음 묶음마다 코드를 부여한다.



코드를 이용하여 압축해줄 수 있다.

(1,0,0,0)	(1,1,1,1)
(1,1,1,1)	(1,0,0,0)
(1,1,1,1)	(1,0,0,0)
(1,0,0,0)	(1,1,1,1)
(1,0,0,0)	(1,0,1,1)
(1,0,0,0)	(1,1,0,1)
(1,1,1,1)	(1,0,0,0)
(1,1,1,1)	(1,0,1,0)

➡ 1001110001100011

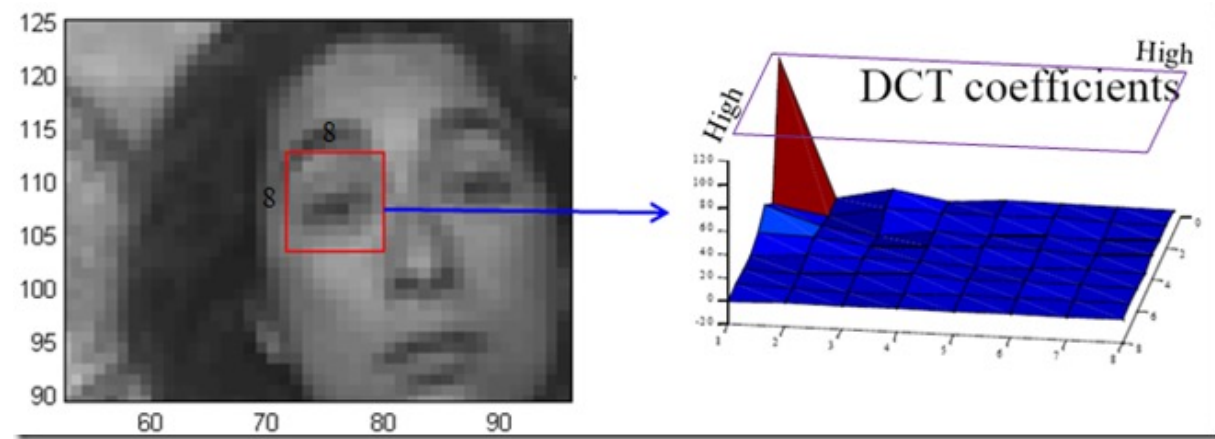
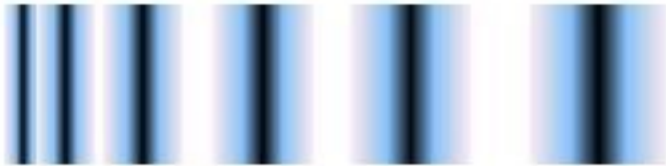
- ✓ 압축해제 할 땐 코드마다 지정해둔 패턴(흰색 글자로 표시)으로 다시 바꾸어주면 된다.
- ✓ Indeo 3.2, Cinepak 코덱 등에서 사용하는 압축 방법

03 컴퓨터에서 동영상의 표현 - 압축

Discrete Cosine Transform(DCT)

한국어로 “이진 코사인 변환”

이미지에서 화소 밝기의 변화율을 파형으로 나타내면 다음과 같다.



사람의 눈은 밝기의 변화율이 큰 부분(높은 주파수)에는 둔감하고, 그 반대에는 민감하다.

따라서 이미지를 파형으로 나타낸 후, 높은 주파수의 파형을 제거해주면 화질에 큰 차이 없이 압축할 수 있다!

✓ JPEG와 MPEG 계열 코덱 등에서 사용하는 압축 방법

출처

파트 1

- https://en.wikipedia.org/wiki/Raster_graphics
- https://en.wikipedia.org/wiki/Vector_graphics

파트 2

- <https://evan-moon.github.io/static/f62ba441843fe9172b18fc5498bd765d/2c288/sampling.png>
- <https://evan-moon.github.io/static/b1bb1c1aa17f6cf158141942536a758f/f2239/bitrate.png>
- [https://en.wikipedia.org/wiki/Sampling_\(signal_processing\)](https://en.wikipedia.org/wiki/Sampling_(signal_processing))
- https://en.wikipedia.org/wiki/Audio_file_format

파트 3

- https://en.wikipedia.org/wiki/Video_file_format
- <https://en.wikipedia.org/wiki/Codec>
- [https://en.wikipedia.org/wiki/Run-length_encoding#:~:text=Run%2Dlength%20encoding%20\(RLE\),than%20as%20the%20original%20run.](https://en.wikipedia.org/wiki/Run-length_encoding#:~:text=Run%2Dlength%20encoding%20(RLE),than%20as%20the%20original%20run.)
- https://en.wikipedia.org/wiki/Vector_quantization
- https://en.wikipedia.org/wiki/Discrete_cosine_transform
- https://en.wikipedia.org/wiki/Discrete_wavelet_transform
- <https://url.kr/5zuej4>
- <https://url.kr/6casw7>