

4.1 텍스트의 표현

1. (문자, 구두점 등등의)

각 글자에는 고유한 비트 패턴이 할당

A. **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange):
영문 텍스트에서 사용되는 기호를 표현하기 위한 7 비트 패턴

B. **Unicode**: 세계 각국 언어에서 사용되는 주요 기호들을 표현하기
위한 비트 패턴 – 65536 종류
UTF-8, KS-5601

C. **ISO** (**I**nternational **S**tandard **O**rganization) 표준: 세계 각국 언어
에서 사용되는 대부분의 기호들을 표현하기 위한 32 비트 패턴

4.2 ASCII 코드표

1. 128개의 비트 패턴과 문자들을 매핑함

2. 대문자 A
0x41 – 65

3. 소문자 a
0x61 – 97

10진수	16진수	8진수	ASCII	10진수	16진수	8진수	ASCII
68	0x44	104	D	98	0x62	142	b
69	0x45	105	E	99	0x63	143	c
70	0x46	106	F	100	0x64	144	d
71	0x47	107	G	101	0x65	145	e
72	0x48	110	H	102	0x66	146	f
73	0x49	111	I	103	0x67	147	g
74	0x4A	112	J	104	0x68	150	h
75	0x4B	113	K	105	0x69	151	i
76	0x4C	114	L	106	0x6A	152	j
77	0x4D	115	M	107	0x6B	153	k
78	0x4E	116	N	108	0x6C	154	l
79	0x4F	117	O	109	0x6D	155	m
80	0x50	120	P	110	0x6E	156	n
81	0x51	121	Q	111	0x6F	157	o
82	0x52	122	R	112	0x70	160	p
83	0x53	123	S	113	0x71	161	q
84	0x54	124	T	114	0x72	162	r
85	0x55	125	U	115	0x73	163	s
86	0x56	126	V	116	0x74	164	t
87	0x57	127	W	117	0x75	165	u
88	0x58	130	X	118	0x76	166	v
89	0x59	131	Y	119	0x77	167	w
90	0x5A	132	Z	120	0x78	170	x
91	0x5B	133	[121	0x79	171	y
92	0x5C	134	\	122	0x7A	172	z
93	0x5D	135]	123	0x7B	173	{
94	0x5E	136	^	124	0x7C	174	
95	0x5F	137	_	125	0x7D	175	}
96	0x60	140	.	126	0x7E	176	~
97	0x61	141	a	127	0x7F	177	DEL

4.2 ASCII 코드표

1. Hello 문자열의 경우

비트패턴	01001000	01100101	01101100	01101100	01101111
16진법	48	65	6B	8B	6F
	H	e	l	l	o

문자	아스키코드	유니코드
A	1000001	00000000001000001
9	0111001	00000000000111001
한자 생	표현 안됨	0111010100011111
한글 가	표현 안됨	10101100000000000

문자(**Character**)와
문자열(**String**) –
Hello\0

4.3 한글 코드

1. 인코딩 (Encoding)



문자를 컴퓨터에 저장하거나 통신으로 전송할 목적으로 부호화 하는 것 (방법).

2. 한글 - 완성형 코드

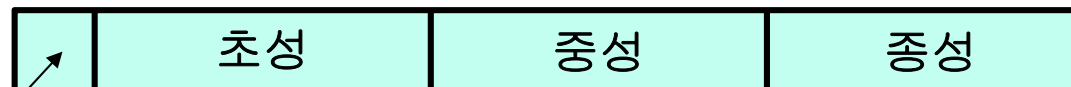
한글의 한글자 한글자에 해당하는 코드표를 만들어 1:1 Mapping하여 글자를 찾는 방식.

3. 한글 - 조합형 코드

논리적 코드생성, 초성, 중성, 종성 코드표 만을 사용하여 코드를 찾는 방식.

예) 장 -  01110 +  00011 +  10111

표현(변환) : 1 + 01110(5bit) + 00011(5bit) + 10111(5bit)



0/1 : 한글 여부

4.3 한글 코드

4. 유니코드

전세계 모든 문자를 하나의 코드표로 통합시킨 개념. 즉, 유니코드 안에는 전세계 모든 언어의 문자가 들어가 있다. 흔히 유니코드라 함은 문자를 2 Byte로 인코딩하는 기법이라 알고 있는데 이는 그렇지 않다. 다만 Euc-KR과 CP949이 2 Byte 처리기법을 사용하고 있다.

5. UTF-8 인코딩 (유니코드)

- 유닉스/윈도우 양쪽 계열 모두에서 사용되는 확장 완성형 코드 페이지
- 전 세계 모든 문자 코드 페이지 포함
- 가변길이 문자 인코딩 방식
- 통상문자 (영어, 특수문자)일 경우 1 Byte 사용
 - ASCII 코드 1 Byte로 표현 (ASCII 코드는 UTF-8의 일부분)
- Euc-KR과 CP949 코드와 호환성 없음

UTF-8 내부 저장방식을 보면 한 문자를 표현/저장하기 위해
최소 1 byte에서 최대 4 byte까지 동적으로 사용.

4.3 한글 코드

6. UTF-16/UTF-32

- UTF-8 A라는 문자 표현을 0x41 1 Byte 로 표현함
- UTF-16 A라는 문자 표현을 0x0041/0x4100 2 Byte 로 표현함
(Little-Endian/Big-Endian)
- UTF-32 A라는 문자 표현을 0x00000041/0x41000000 4byte 로 표현함

7. Euc-KR 인코딩 (Extended Unix Code Page)

- 완성형 유닉스 계열코드 페이지
- 가변길이 방식의 영어, 특수문자, 한글(2,350 자), 일부 한자 만 표현가능
통상문자(영어. 특수문자)일 경우 1 Byte. 한글일 경우 2 Byte
- ISO/IEC 2022 표준

4.3 한글 코드

8. CP949 인코딩 (Extended Windows Code Page)

-확장 완성형 코드 페이지 (Euc-KR과 호환 됨)

완성형 코드에 없는 한글을 조합형으로 처리하는 로직을 가지고 있음

-윈도우 계열의 코드 페이지

-가변길이 방식의 영어, 특수문자, 한글, 일부 한자 만 표현가능

통상문자(영어. 특수문자)일 경우 1 Byte, 한글일 경우 2 Byte

그 외 조합형 병행 사용, 11,172 자의 한글 표현 가능

-Euc-KR보다 많은 한글 수 표현 가능

-Euc-KR을 따라 한 개념으로 ISO/IEC 2022 표준 채택은 안 되어 있음

첨부한

" 컴퓨터 한글 코드.pdf"

를 꼭 읽어 두시기 바랍니다.

4.4 이미지의 표현

1. 비트맵(Bitmap) 기법

- A. 픽셀(**Pixel**): "picture element"의 줄임말 (화소)
- B. Binary (1bits), Gray (8bits), RGB (Red, Green, Blue: 24 bits)

2. 벡터 기법 (Vector Image, 벡터이미지)

- A. 크기 변경이 자유롭다
- B. TrueType와 PostScript

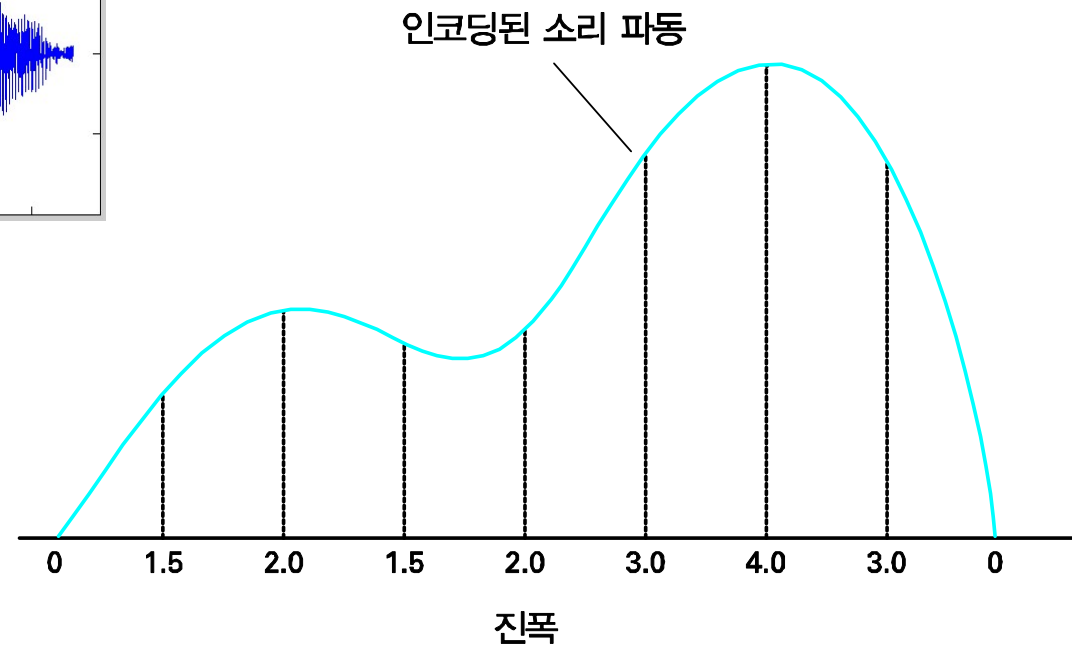
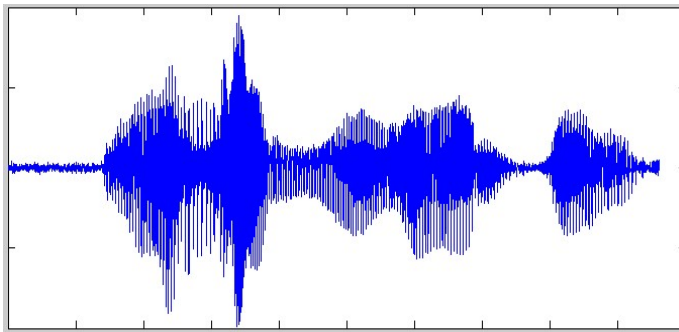
예) 네비게이션 또는 구글 지도에서의 경계선 영상



4.5 소리의 표현

1. 샘플링(Sampling) 기법

- A. 아날로그 파형을 디지털로 리코딩에 사용됨
- B. 가령) X 축을 균일한 간격으로 구분한 후에 측정한 값
(0, 1.5, 2.0, 1.5, 2.0, 3.0, 4.0, 3.0, 0)을 16 bits 로 표현하여 저장함

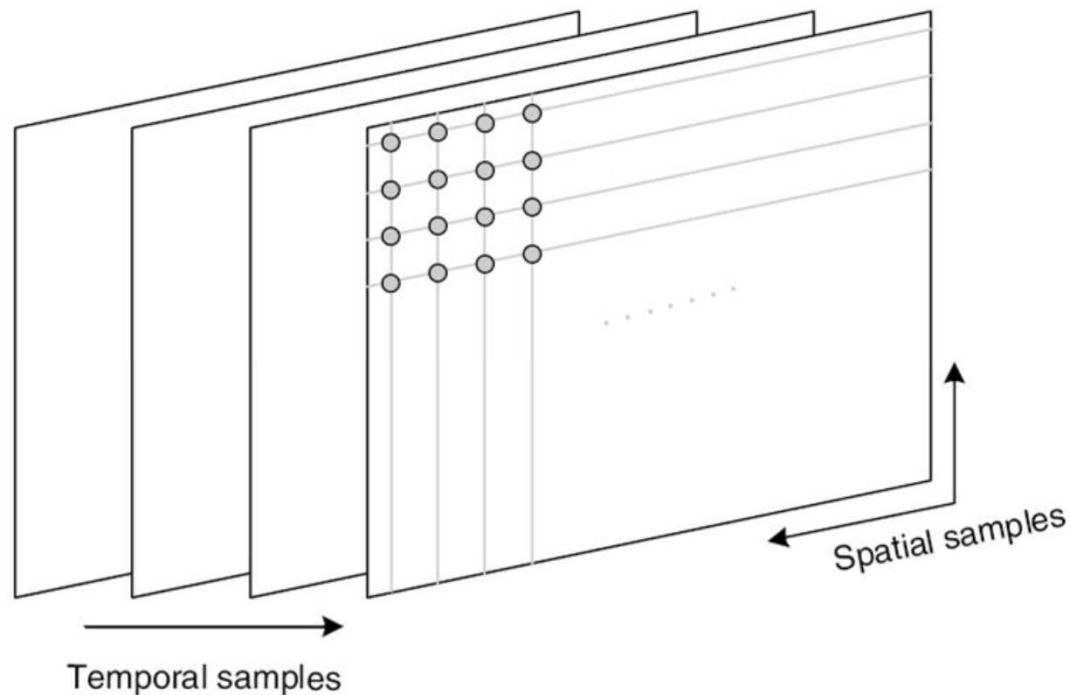


4.6 동영상의 표현

1. 동영상

- A. 다수의 프레임과 음성으로 구성
- B. 1초에 여러 장 프레임을 재생하면 인간은 연속적인 것으로 착시
(25 FPS, Frame Per Second)

2. 시간적 및 공간적인 중복성 존재하며 용량이 방대함



4.6 동영상의 표현

1. Full HD 동영상의 용량

- A. 1시간 크기
- B. 30 FPS
- C. 1920×1080 (픽셀) \times 3Bytes(RGB) \times 60x60(초) \times 30(프레임)
- D. 671 기가 바이트 용량

2. 컴퓨터에서 처리하는 데이터의 경우 압축 등 필요

5.1 데이터의 압축

손실 압축(Lossy compression)과 무손실 압축(Lossless compression)

1. RLE(Run-Length Encoding)

예) 10011100 00110000 → 1(1),0(2),1(3),0(4),1(2),0(4)

2. 빈도 종속 인코딩(Huffman 코드)

예) abacccddeaaccc ...

(부호)	(빈도)	(비트)
a	5	10
b	2	1111
c	9	0
d	3	110
e	1	1110

3 상대적 인코딩(Relative encoding): 차이값을 인코딩

예) 4 5 6 8 9 → 4 1 1 2 1

4. 사전(dictionary) 인코딩 (LZW 인코딩 같은 적응적 사전 인코딩 포함) LZW (Lempel-Ziv-Welsh)

5.2 이미지의 압축

1. GIF: 만화에 유리

GIF - Graphics Interchange Format

2. JPEG: 사진에 유리

JPEG - Joint Photographic Experts Group

3. TIFF: 이미지 보관에 유리

MPEG - Moving Picture Experts Group

5.3 오디오 및 비디오의 압축

1. MPEG (Motion Picture Expert Group)

- A. 고화질 TV 방송
- B. 화상 회의

2. MP3 (MPEG Layer 3)

- A. 시간적 차폐
- B. 주파수 차폐

3. MPEG-4 파트 14(MPEG-4 Part 14)

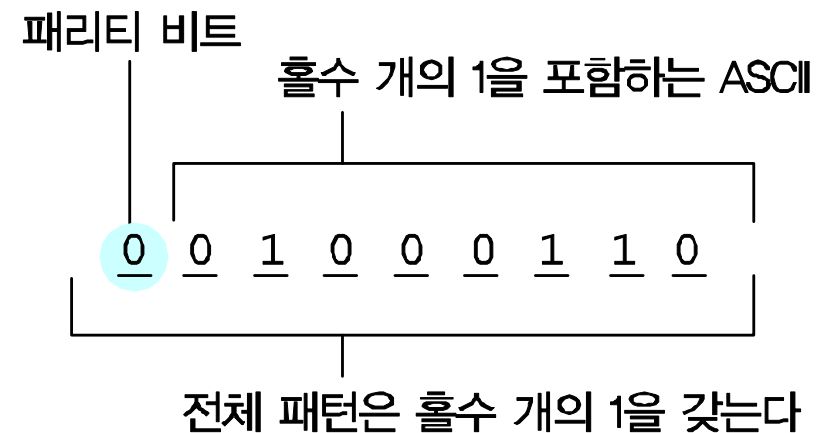
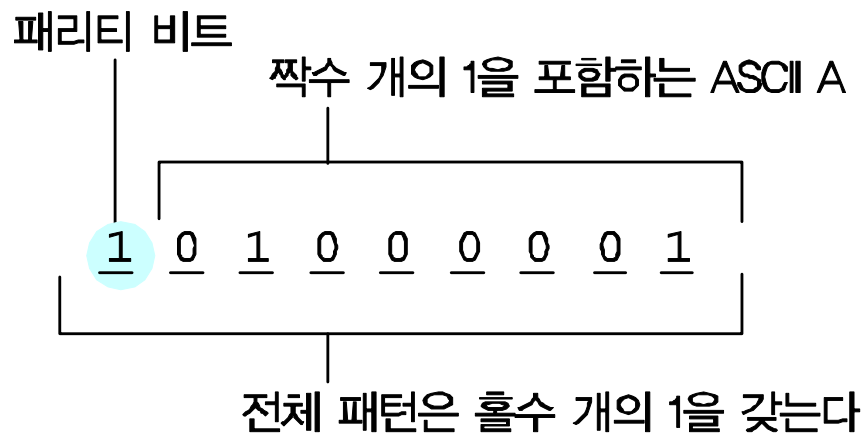
또는 MP4 (공식적으로 ISO/IEC 14496-14:2003)

- A. MPEG-4의 일부로 규정된 멀티미디어 컨테이너 포맷 표준.
- B. 디지털 비디오와 디지털 오디오 스트림(Stream)을 저장하는데 사용.
(자막과 스틸 이미지 따위의 기타 데이터를 저장하는데 사용 가능)
- C. 컨테이너 포맷과 동일하게 MPEG-4 파트 14는 인터넷을 통한 스트리밍 지원.
- D. 파일 내 스트리밍(Streaming) 정보를 포함하기 위해 별도의 힌트 트랙이 사용.
- E. 공식적인 기본 확장자는 .mp4.

5.4 통신 오류

기본적인 아이디어는 비트를 추가로 사용하여
오류를 검출함!

1. 패리티 비트 (짝수 패리티와 홀수 패리티)



2. 검사 바이트 (CRC - Cyclic Redundancy Check)

3. 오류 정정 코드

→ 구체적인 예제는 컴퓨터네트워크 / 데이터통신
교과목에서 학습