



멀티미디어 데이터 압축



차 례

- 압축의 개념
- 압축 알고리즘
- 이미지 압축 표준
- 오디오 압축 기법
- 동영상 압축 표준

1.압축의 개념

- 압축의 개념
- 데이터의 중복성
- 압축 시스템의 평가
- 압축/복원 속도

압축의 개념

- 코딩

- 코딩(coding) = 인코딩(encoding) + 디코딩(decoding)

- 인코딩: 목적에 따라 데이터의 형태를 변환하는 과정

- 디코딩: 인코딩된 데이터를 원래의 데이터로 복원하는 과정

- 목적에 따른 분류

- 저장공간과 전송 대역폭의 효율적 이용 → 압축

- 데이터의 전송 도중에 발생하는 오류의 처리 → 패리티 비트, 해밍코드

- 데이터의 보호 → 암호화 (encryption)

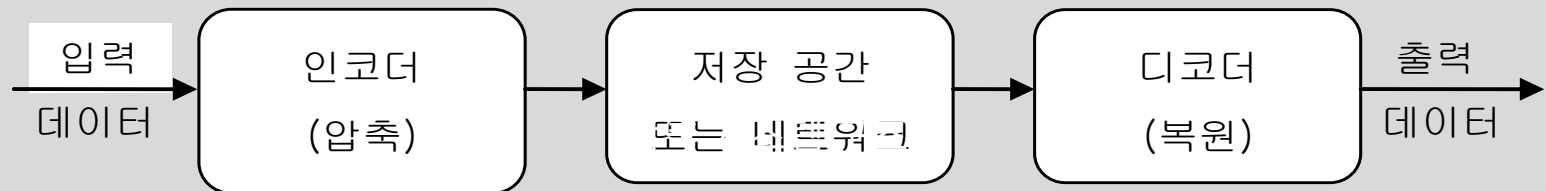
압축의 개념

- 압축

- 저장공간과 전송 대역폭의 효율적 이용을 위해 데이터를 크기 (전체 비트 수)를 줄이는 것

- 데이터 압축 시스템 구조

- 인코더에서 압축을 하고, 디코더에서 복원
- 중간 매체: 데이터 저장 공간 또는 통신/컴퓨터 네트워크
- 코드(**code**): 인코더의 출력



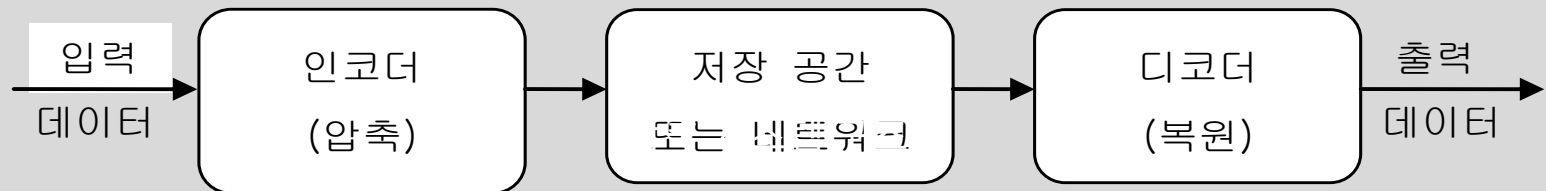
압축의 개념

- 압축

- 저장공간과 전송 대역폭의 효율적 이용을 위해 데이터를 크기 (전체 비트 수)를 줄이는 것

- 데이터 압축 시스템 구조

- 인코더에서 압축을 하고, 디코더에서 복원
- 중간 매체: 데이터 저장 공간 또는 통신/컴퓨터 네트워크
- 코드(**code**): 인코더의 출력

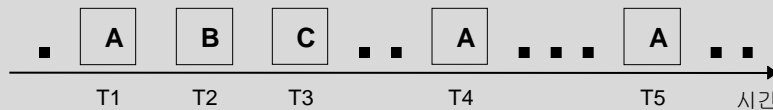


압축의 개념

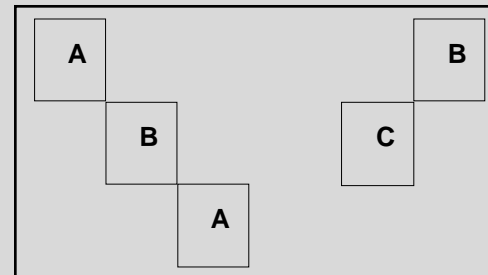
- 3 분 길이의 CD오디오 품질의 음악을 44.1 kHz의 주기로 16 비트 샘플링하여 디지털 데이터로 변환한 데이터의 크기
 - $3 \text{ 분} \times 60 \text{ 초/분} \times 44,100 \text{ 회/초} \times 2 \text{ bytes/회} \approx 15 \text{ Mbytes}$
- 1280 x 1024 해상도의 32 비트 색 이미지를 초당 30 프레임으로 전송하는 비디오 응용에 필요한 전송 대역폭
 - $1280 \times 1024 \times 4 \text{ bytes/프레임} \times 30 \text{ 프레임/초} = 150 \text{ Mbytes/초}$
- 전화선(xDSL)과 모뎀을 이용할 경우: $0.2 \sim 6.3 \text{ Mbytes/초}$ (=1.5 ~ 52 Mbps)
- Ethernet을 이용할 경우: $1.2 \sim 12 \text{ Mbytes/초}$ (=10 ~ 100 Mbps)

데이터의 중복성

- 시간적 중복성(temporal redundancy)
 - 시간의 흐름에 따라 동일한 데이터가 반복하여 나타남
 - 사운드 데이터, 동영상 데이터
- 공간적 중복성(spatial redundancy)
 - 공간상의 여러 위치에 같은 데이터가 중복되어 나타남
 - 정지 영상 데이터나 문서 데이터, 동영상 데이터



(a) 시간적 중복



(b) 공간적 중복

압축 시스템의 평가

- 압축률(compression ratio)

$$\text{압축률} = \frac{B_0}{B_1}$$

(B_0 : 압축 이전의 데이터 비트 수
 B_1 : 압축 이후의 데이터 비트 수)

- 복원된 데이터의 품질(quality)
 - 무손실 압축(lossless compression)
 - 문서 데이터
 - 손실 압축(lossy compression)
 - 소리 데이터, 이미지 데이터

압축/복원 속도

- 대화형(dialogue) 멀티미디어 응용
 - 데이터의 압축과 복원이 거의 동시에 일어남
 - 압축/복원 속도 모두 중요
 - *대칭형(symmetric) 멀티미디어 응용*
예) 비디오 전화, 비디오 회의 등
- 검색형(retrieval) 멀티미디어 응용
 - 복원 속도 중요, 압축 속도는 덜 중요
 - *비대칭형(asymmetric) 멀티미디어 응용*
예) 주문형 비디오(VOD), 비디오 CD 등

2.압축 알고리즘

- 압축 알고리즘
- 무손실 압축과 손실 압축
- 혼성 압축
- 반복길이 코딩
- 허프만 코딩
- 렘펠-지프 코딩
- 변환 코딩
- 빠른 푸리에 변환
- 이산 코사인 변환
- 예측 코딩
- 웨이블릿 코딩
- 보간법

압축 알고리즘

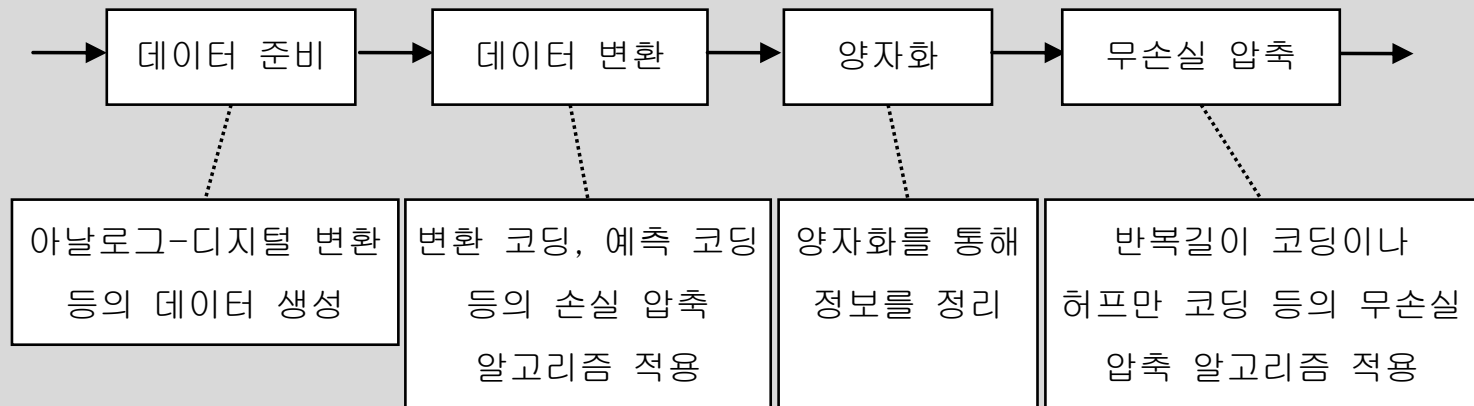
무손실 압축 (lossless compression)	반복길이 코딩 (run-length coding)	
	허프만 코딩 (Huffman coding)	
	렘펠-지프 코딩 (Lempel-Ziv coding)	
손실 압축 (lossy compression)	변환 코딩 (transform coding)	FFT, DCT
	예측 코딩 (predictive coding)	DPCM, ADPCM DM, ADM
	양자화 (quantization)	
	웨이블릿 코딩 (wavelet-based coding)	
	보간법 (interpolation)	
	프랙탈 압축 (fractal compression)	
	혼성 압축 (hybrid compression)	
JPEG, GIF, MPEG, H.261, H.263 ...		

무손실 압축과 손실 압축

- 무손실 압축
 - 이전 데이터에 포함된 모든 정보가 손실 없이 인코딩됨
 - 반복적으로 출현하는 데이터의 중복성만을 제거
 - 문서 데이터 압축에 필수
 - 엔트로피(*entropy*) 코딩
- 손실 압축
 - 포함된 정보 중 내용을 인식하는데 크게 영향을 주지 않는 정보들을 삭제
 - 무손실 압축에 비해 압축률을 높임
 - 원천(*source*) 코딩

혼성 압축

- 손실 압축 기법과 무손실 압축 기법을 혼용
- 20:1 ~ 200:1 정도의 압축률을 얻을 수 있음
- 대부분의 표준 압축 기법에서 이용됨



혼성 압축의 과정

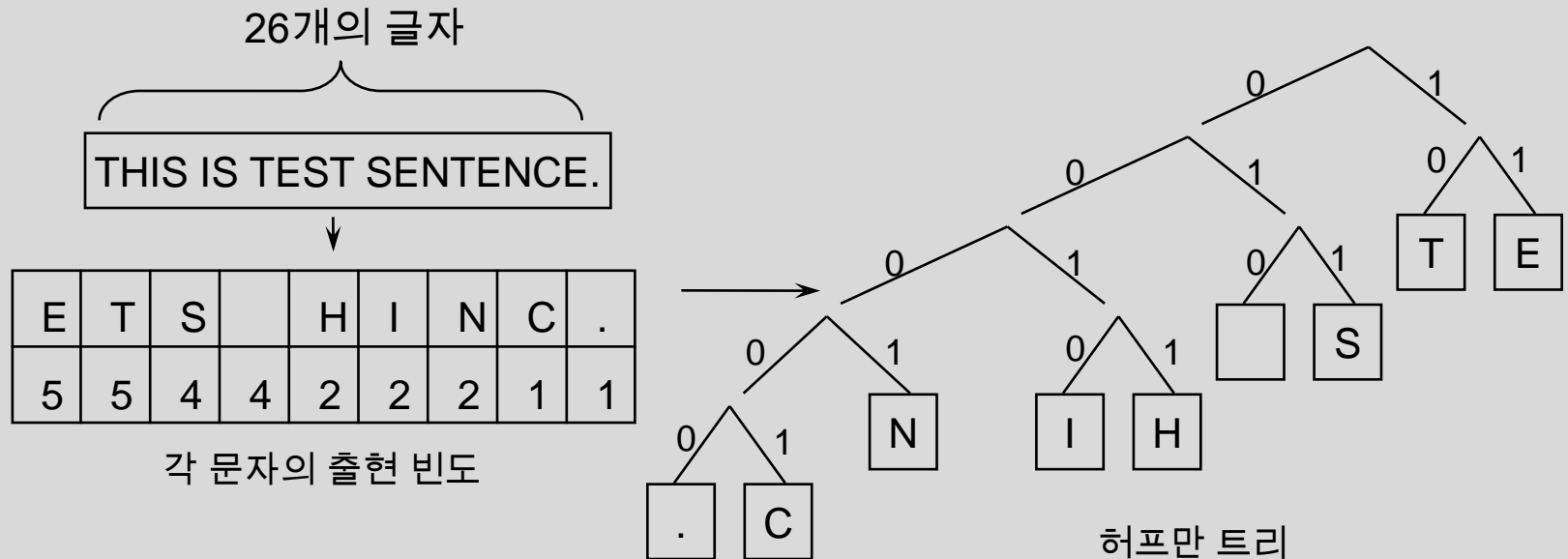
반복길이 코딩

- 반복길이 코딩(Run-Length Coding)
 - 연속적으로 반복되어 나타나는 정보(문자, 픽셀)들을 그 정보와 반복된 횟수(run-length)로 표현하는 코딩 방법

예)  TEAAAAAAAAAAAAALOV → TEA@12LOV

허프만 코딩(Huffman Coding)

- 데이터를 구성하는 단위 정보들의 빈도수를 기반으로, 각 단위 정보를 표현하는 비트 수를 효율적으로 할당
- 가변 길이 코딩 (Variable-Length Coding) 방법
예)



허프만 코딩(Huffman Coding)

E	11	I	0010
T	10	N	0001
S	011	C	00001
	010	.	00000
H	0011		

[각 문자마다 할당된 코드]

- 빈도수가 높은 단위 정보에는 비트 수를 적게 사용하고, 낮은 빈도수의 단위 정보에는 비트 수를 많이 할당
- 8비트 아스키 코드 이용: $8 \times 26 = 208$ 비트
- 허프만 코딩이용: $(2 \times 5) \times 2 + (3 \times 4) \times 2 + (4 \times 2) \times 3 + (5 \times 1) \times 2 = 78$ 비트

허프만 코딩(Huffman Coding)

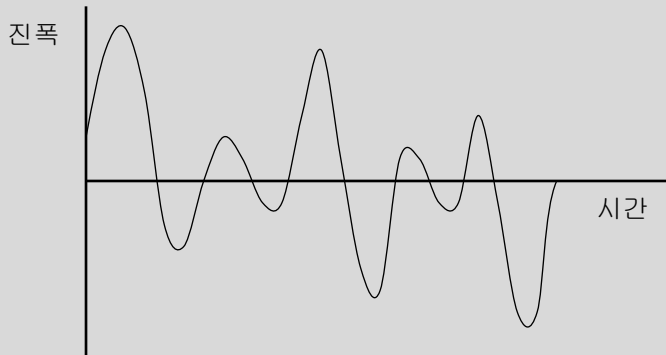
- 속성
 - 유일한 접두사(unique prefix property)
 - 하나의 허프만 코드는 다른 허프만 코드의 접두사가 될 수 없음
 - ← 이유: 모든 입력 기호들을 허프만 트리의 리프 노드에 둬
 - 가장 효율적인 방법(optimality)
 - 인코딩된 데이터의 길이가 최소(minimum-redundancy code)
- 단점
 - 입력 기호의 확률 분포가 변한다면, 효율성을 잃어버림
 - (가변 길이 코딩)
 - 입력 기호의 확률을 인코딩 전에 미리 알아야 함

렘펠-지프 코딩(Lempel-Ziv Coding)

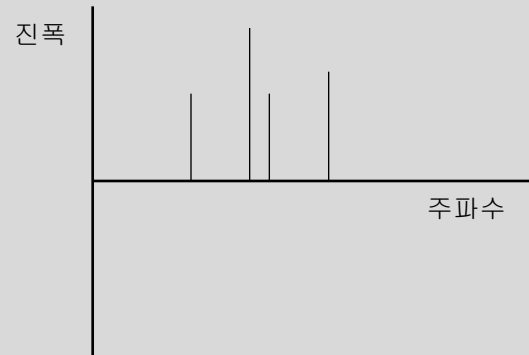
-
- 동일한 어구(phrase)가 반복적으로 나타나는 데이터의 성질 이용
- 처음 나타난 어구는 사전(dictionary)에 어구와 그 어구의 위치 정보를 등록하고 그대로 출력
- 이후 나타난 어구는 사전에 등록된 위치 정보로 치환
- pkzip과 같은 파일 압축 프로그램에 이용
 - 렘펠-지프-웰치(LZW: Lempel-Ziv-Welch) 코딩
 - 어구들 사이의 상대적 위치를 이용
 - UNIX의 compress에 이용
 - GIF(Graphics Interchange Format)의 영상 압축 기법

변환코딩(Transform Coding)

- 변환 함수등을 통해 데이터의 영역(domain)을 바꿈
- FFT(Fast Fourier Transform): 시간 영역 \rightarrow 주파수 영역
- DCT(Discrete Cosine Transform): 공간 영역 \rightarrow 주파수 영역



(a) 시간 영역에서 본 데이터



(b) 주파수 영역에서 본 데이터

사운드 데이터의 영역에 따른 표현

빠른 푸리에 변환(FFT: Fast Fourier Transform)

- CD-Audio의 인코딩에 이용
- 사운드 데이터를 시간 영역에서 주파수 영역으로 변환
 - 시간 영역 내의 n 개의 샘플링된 값들을 주파수 영역 내의 n 개의 값들로 변환하는 경우, 변환된 데이터의 i 번째 값은

$$F(i) = \sum_{k=0}^{n-1} P_k \left(\cos \frac{2ik\pi}{n} + j \sin \frac{2ik\pi}{n} \right)$$

(P_k 는 원래 데이터의 k 번째 값)

이산 코사인 변환(DCT: Discrete Cosine Transform)

- JPEG, MPEG, H.261 등의 표준 압축 기법에서 이용
- 정지 영상 데이터를 공간 영역에서 주파수 영역으로 변환
 - 크기가 $N \times N$ 인 단위 영상 데이터를 변환하는 경우, 변환된 데이터의 (i, j) 번째 값은

$$D(i, j) = \frac{1}{\sqrt{2N}} C_i C_j \sum_x \sum_y p_{xy} \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

(P_{xy} 는 원래 데이터의 (x, y) 번째 값,

$$C_i, C_j \text{는 } C_i = C_j = \begin{cases} 1/\sqrt{2} & \text{if } (i=j=0), \\ 1 & \text{if } (i \neq 0 \text{ 혹은 } j \neq 0) \end{cases}$$

예측 코딩(Predictive Coding)

- 가정 : 정보의 흐름에 있어서, 다음에 나타날 정보는 바로 직전에 나타난 정보와 크게 다르지 않다
- 이전 정보와 차이 정보를 가지고 다음 정보를 생성
- 예측 코딩 기법의 종류
 - 차이 PCM(DPCM: Differential PCM)
 - 적응적 차이 PCM(ADPCM: Adaptive DPCM)
 - 델타 변조 방식(DM: Delta Modulation)
 - 적응적 DM(ADM : Adaptive DM)

양자화(Quantization)

- 양자(quantum)라는 미리 정의된 정보를 이용하여 주어진 데이터를 코딩
- 스칼라 양자화 (scalar quantization)
 - 어떤 스칼라 값을 양자(quantum)로 정하고, 데이터를 구성하는 값들을 양자의 개수로 치환
 - 데이터의 중요도에 따라 크기가 다른 양자를 이용함으로써 덜 중요한 정보를 삭제하는 효과를 얻음
 - 1 차원 변수를 기반으로 양자화하는 것

예)

120	115	55	70	81	83	88	75
30	29	14	17	20	21	22	19
120	116	56	68	80	84	88	76

(a) 양자화 이전의 데이터

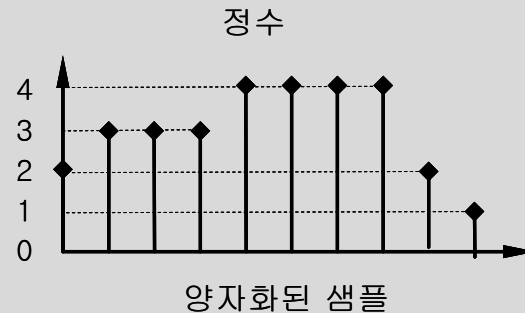
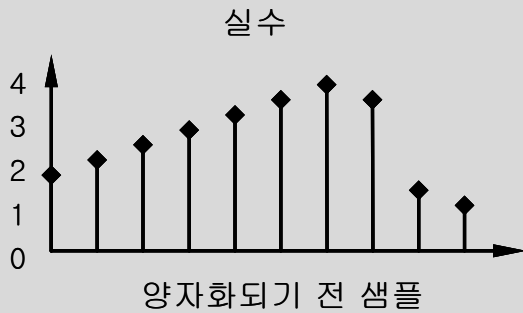
(b) 4의 배수로 양자화된 데이터

(c) 디코딩된 데이터

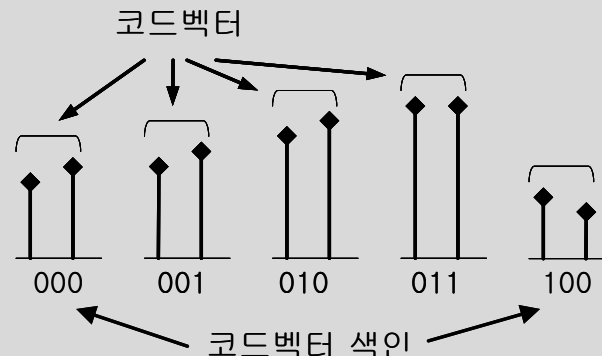
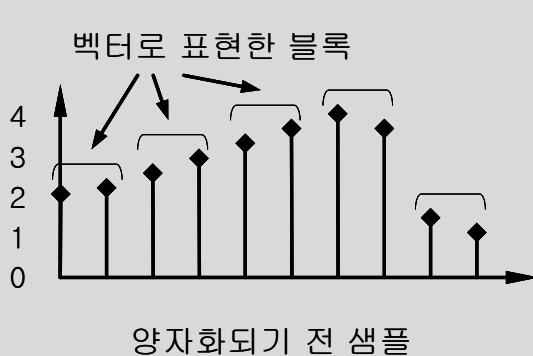
벡터 양자화 (vector quantization)

- 하나의 값에 대해 둘 이상의 변수를 사용하여 양자화
- 코드북(codebook)에 미리 정의된 벡터 정보를 이용하여 데이터 내의 정보들을 코드북에 존재하는 정보의 색인으로 치환
- 인코딩 과정에서는 속도가 느리나, 디코딩 과정에서는 각 단위 정보마다 코드북을 한번만 참조하므로 속도가 빠름
- 효율적인 코드북을 구성하는 것이 중요: 데이터의 특성을 바탕으로 구성됨

스칼라 양자화와 벡터 양자화



(a)



2차원 코드벡터를 가진 코드북

(b)

스칼라 양자화와 벡터 양자화의 비교

웨이블릿 코딩(Wavelet-Based Coding)

- 웨이블릿 변환(wavelet transform) 함수 사용하여 신호 (signal)를 시간과 진동수 측면에서 양측 모두 좋은 신호로 분해하여 압축 성능을 높인 코딩 방법
- MPEG-4와 JPEG2000에 사용됨
- 2 개의 상호 보완적인(complementary) function 이용
 - 웨이블릿(wavelet): 높은 주파수의 세부 이미지를 추출
 - 스케일링 함수(scaling function): 높은 주파수를 제거
- 이미지 압축에 주로 사용
 - 이산 웨이블릿 변환(DWT: Discrete Wavelet Transform)

보간법 (Interpolation)

- 연속된 비슷한 정보들 중 몇 개만을 저장
- 복원 시 보간을 통해 삭제된 정보를 복원

예)



인접한 픽셀들의 값이 유사한 경우에 대한 보간법의 적용

3. 이미지 압축 표준

- JPEG
- GIF
- F.72X

JPEG(Joint Photographic Experts Group)

- 이미지 압축 표준
 - 정지 영상(Color & Grayscale)의 압축 코딩의 표준
 - 1992년 ISO에서 제정
 - 특징
 - 압축률과 영상의 품질을 사용자가 지정 가능
 - 영상의 형태(내용, 크기, 해상도, 화면비, 색상모형)에 관계없이 적용 가능
 - 하드웨어를 이용할 경우, 영상의 품질이 크게 향상됨

JPEG의 약점

- 낮은 픽셀 깊이(0.25 bpp 이하) 압축에서 변형(distortion)이 심함
- 낮은 전송률에서 높은 전송률에 이르는 점진적(progressive) 전송
- 하나의 복원 구조(single decompression architecture)가 없음
- 하나의 비트 스트림(bitstream) 상에서 무손실 및 손실 압축 제공 못함
- 사이즈가 매우 크거나 높은 정밀도를 가진 이미지들을 처리할 수 없음
- 관심 지역 코딩(ROI: Region-of-interest coding)을 지원하지 못함
 - 특정 부분이 이미지의 나머지 부분보다 더 나은 품질로 코딩될 수 있음
- 합성(computer-generated) 이미지(컴퓨터)나 혼합 문서에서 좋은 성능이 나오지 않음

GIF (Graphics Interchange Format)

- CompuServe™의 표준 정지 영상 포맷
- 8bpp(bit-per-pixel) 정지 영상을 대상으로 함
 - 256 색상(8 bit)을 표현할 수 있음
 - 24 bpp영상을 GIF로 인코딩하는 경우
 - 색상테이블에 정의된 256 개의 색상으로 양자화
 - 색상 정보의 손실이 있음
- 렘펠-지프-웰치(LZW) 코딩을 이용
 - 렘펠-지프 코딩의 개선된 기법
 - 색상 정보 이외의 다른 정보는 손실되지 않음

G.72X 표준

- 화상회의나 모뎀과 같은 다양한 원격통신(telecommunication) 응용에 적용
- ADPCM
 - 일련의 데이터의 연속된 샘플의 값들 사이의 차이만을 인코딩하고 저장함으로써 압축을 하는 방식
 - ITU의 음성 압축 표준인 G.721, G.723, G.726, G.727의 핵심 기법
 - 32 kbps에서 음성이 인지되는 품질
 - 표준 64 kbps PCM 전송시보다 약간 떨어짐
 - DPCM보다 좋음
- 제한된 대역의 전화 채널상에서 음성의 효율적인 코딩 지원

G.72X 표준

- G.721: 첫 번째 표준으로, 300에서 3400 Hz의 대역폭을 가진 채널 상에서 32 kbps ADPCM으로 음성을 압축
- G.723: G.721 ADPCM의 24, 40 kbps로의 확장
- G.726: 1990년에 G.721과 G.723을 대체함, 16, 24, 32, 40 kbps에서의 데이터 압축 지원
- G.722: G.726의 개선된 버전으로, 0에서 8 kHz의 광대역 (broadband) 전화 채널 상에서 64 kbps내의 7 kHz 오디오 코딩

4.오디오 압축 표준

- MPEG 오디오 압축
- 서브 밴딩 코딩

MPEG 오디오 압축

- 청각 심리 모델(psychoacoustic model) 이용
 - 인간의 사운드에 대한 감각적인 특성을 모델링
- 3가지 계층
 - 1 계층: 디지털 오디오 테이프
 - 2 계층: 1 계층 보다 높은 복잡도 지님, 디지털 오디오 방송
 - 3 계층: 가장 복잡, ISDN 선상에서의 오디오 전송을 목표로 시작
MP3가 이에 해당

MPEG 오디오 압축

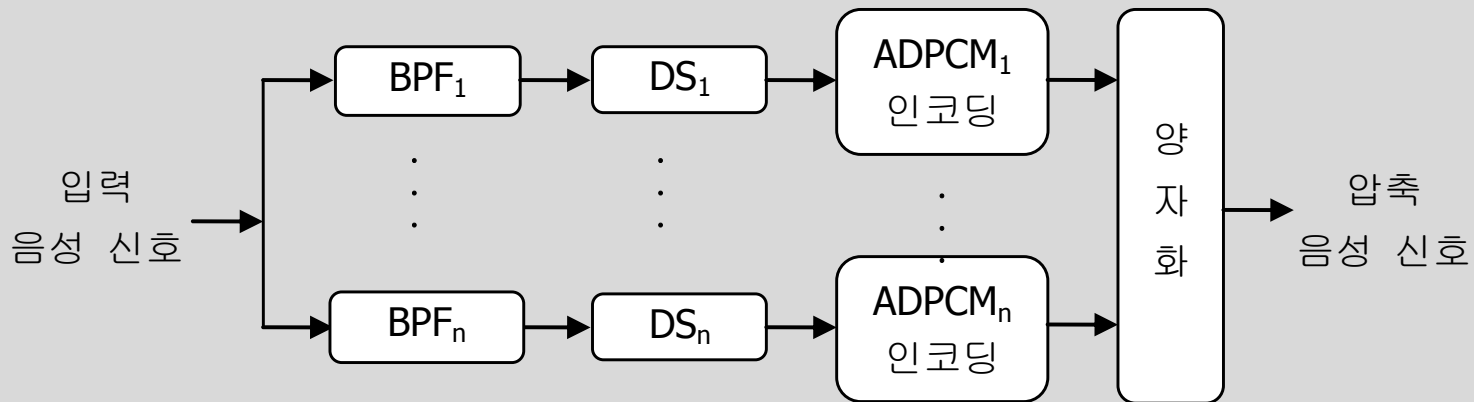
- MPEG-1 layer 2
 - 순방향 적응적(forward adaptive) 비트 할당
 - 부분 비트 양자화 사용
- MPEG-2
 - MPEG-1 오디오에 유연성(flexibility) 추가
 - MPEG-1 샘플링 속도의 $\frac{1}{2}$ 에 해당하는 샘플링 속도 추가
 - 다채널(multichannel)로 확장
 - DVD를 지원하는 표준 차량 등에 널리 쓰임
- MP3 (MPEG-1 layer 3)
- MPEG-4
 - 음성 압축(speech compression), 지각 기반 코더(perceptual coder), TTS, MIDI 등 여러 가지 다른 오디오 요소들을 하나의 표준으로 통합

서브밴드 코딩 (Subband Coding)

- 아날로그 신호를 주파수 대역 상에서 다수 개의 주파수 대역으로 분리하고, 그 후에 각 아날로그 신호에 대해 코딩 방법을 적용
- 선택적인 주파수 변환 기법
 - 주파수 대역 등의 스펙트럼의 미리 정의된 영역에서만 신호를 다룸
- 대역의 개수가 중요한 품질의 척도
- 보통 32개의 서브밴드를 사용
- 장점
 - 음성 압축에 잘 적용됨: 인간의 청각 특성을 이용한 처리가 용이
 - 에너지가 편중된 주파수 대역에 더 많은 비트를 부여하여 양자화함으로써 청각적인 오류를 최소화함
 - 서로 다른 주파수 대역에 대한 코딩 과정에서 일어날 수 있는 양자화 에러는 각 주파수 대역간에 서로 무관

서브밴드 코딩 (Subband Coding)

- 음성 신호는 각 주파수 대역 별로 구분된 여러 개의 대역 통과 필터(BPF: Band Pass Filter)를 통과해 분리된 신호를 얻음



* DS(DownSampling): 신호의 샘플링 개수를 줄이는 것

서브밴드 코딩의 인코딩 과정

5.동영상 압축 표준

- H.261
- H.263
- MPEG
- MPEG-1
- MPEG-2
- MPEG-4

H.261

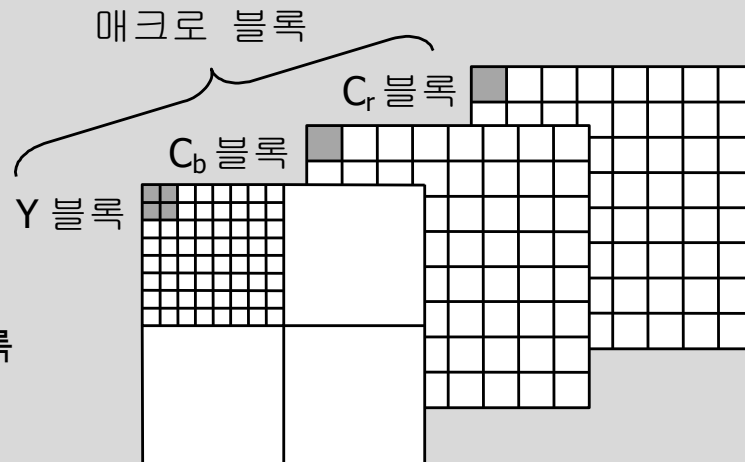
- 1990년에 ITU-T에 의해 제정된 동영상 압축 표준
- ISDN 전화선 상에서 대화형 멀티미디어 응용을 위해 고안됨
- 64kbps의 배수만큼의 전송대역폭을 필요로 함(px64, p:1~30)
- 압축/복원 두 과정에 의한 시간 지연 < 0.15초
- 대상으로 삼고 있는 동영상
 - CIF(Common Intermediate Format)
 - QCIF(Quarter-CIF)

동영상 포맷	밝기 정보 해상도	색채 정보 해상도	압축되지 않은 경우 전송률(Mbps) - 30프레임/초	H.261 지원
QCIF	176 x 144	88 x 72	9.1	Yes
CIF	352 x 288	176 x 144	36.5	Optional

H.261

- 블록과 매크로 블록

- 서브샘플링(2:1:1) 기법에 의해 하나의 밝기 정보(luminance) Y와 두 개의 색채 정보(chrominance) C_b , C_r 로 나뉘어 인코딩
- 블록(block): 8×8 픽셀로 이루어진 단위
- 매크로(macro) 블록: 네 개의 Y 블록과 이에 대응하는 한 개씩의 C_b , C_r 블록



H.261의 블록과 매크로 블록

H.261

- 코딩 알고리즘
 - 인트라코딩(intra-frame coding)
 - 프레임 자체를 인코딩
 - JPEG의 순차적 부호화 기법과 유사
 - DCT, 양자화, 허프만 코딩을 이용
 - 공간적 중복성(spatial redundancy)을 제거
 - 인터코딩(inter-frame coding)
 - 매크로 블록들과 그 이전의 기준 프레임의 매크로 블록들
간에 존재하는 차이 정보를 이용
 - 움직임 보상 기법을 이용
 - 시간적 중복성(temporal redundancy)을 제거

H.261

- 움직임 보상 기법(motion compensation)

- 매크로 블록들 사이의 움직임 벡터와 차이 정보로 화면을 인코딩하는 방법

- 인코딩하려는 프레임과 기준 프레임의 매크로 블록들을 대응

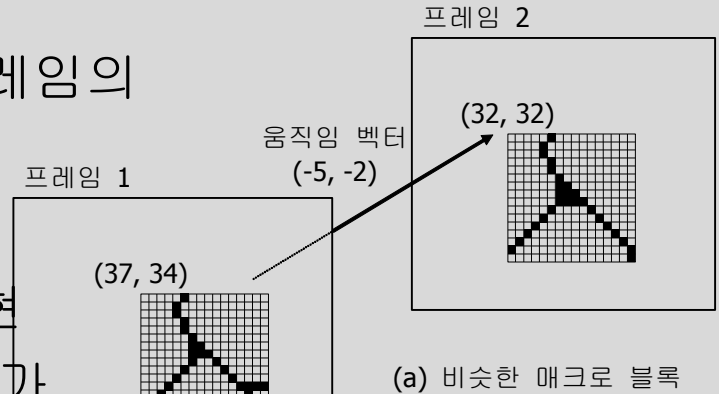
- 위치의 차이는 움직임 벡터

(motion vector)를 이용하여 표현

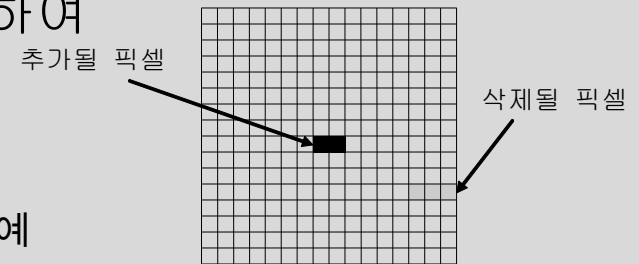
- 매크로 블록들 사이의 차이 정보가 일정 임계값(threshold)을 넘는 경우,

DCT, 양자화, 허프만 코딩을 이용하여

차이 정보를 인코딩



움직임 보상 기법의 예



(b) 차이 정보

H.263

- 1995년에 ITU-T에 의해 제정된 동영상 압축 표준
- 대화형 멀티미디어 응용을 지원하는 향상된 비디오 코드화 표준
- 일반 전화망(PSTN, Public Switched Telephone Networks) 사용
- 64kbps이하의 낮은 전송률을 대상
- 예) PC, 디지털 카메라, 화상 전화, 감시 카메라, 핸드폰, 멀티미디어 플레이어 등

H.263

- 코딩 알고리즘
 - H.261과 유사
 - 인트라코딩: 변환 코딩
 - 인터코딩: 예측 코딩
 - 차이점
 - GOB(Group Of Blocks)
 - » 고정 크기가 아님
 - » 이미지의 왼쪽 경계에서 시작해서 오른쪽 경계에서 끝남
 - 사용할 수 있는(negotiable) 4개의 코딩 option
 - » Unrestricted motion vector mode
 - » Syntax-based arithmetic coding mode
 - » Advanced prediction mode
 - » PB-frames mode

MPEG

- 국제 표준화 기구(ISO: International Organization for Standardization)와 국제 전기 표준 회의(IEC: International Electrotechnical Commission)가 구성한 공동 위원회(JTC1: Joint Technical Committee 1) 산하의 전문 부위원회(SC29: Sub-Committee 29)를 지칭 :1988년 설립
- 동영상, 오디오 압축과 압축된 정보를 전송하는 방법 등에 대한 표준을 제정

MPEG-1

- 1991년에 ISO에 의해 승인된 표준
- 대화형 및 검색형 멀티미디어 응용에 모두 사용
- CD-ROM 등의 일반 디지털 저장매체를 대상으로 함
- 1.5 Mbps의 전송률의 일반 디지털 저장 매체를 대상
- JPEG과 H.261 표준에 사용된 기술 흡수 및 호환성 제공

MPEG-2

- 고품질 디지털 비디오, TV, HDTV의 압축과 복원
- 2 ~ 45 Mbps의 전송률 지원
- MPEG-1과 H.261 등 기존 표준들과의 호환성 제공
 - 여러 단계(level)를 정의: 각 단계마다 거기에 적합한 응용들의 특성에 맞게 여러 프로파일(profile)들이 정의
 - 사운드 데이터에 있어서도 많은 확장이 이루어짐
- 동영상 압축 방식: DCT를 기반
- MPEG-3 흡수: MPEG-2가 HDTV 응용을 지원

MPEG-4

- 사용자와 서비스 제공자와의 쌍방향 통신을 지원하고, 통신, 컴퓨터, 방송기기 등의 여러 매체들을 통합하려는 목적으로 개발됨
- 이동(mobile) 환경에서 멀티미디어 통신을 지원
- 5 kbps ~ 10 Mbps의 낮은 전송률로 동화상을 보내고자 개발된 압축과 복원