



14장 디지털 영상의 압축

- 디지털 영상 압축의 개요
- 디지털 영상의 압축 기법
- 정지영상 표준 압축 부호화 기법
- 동영상 표준 압축 부호화 기법

학습목표

- ✓ 압축의 원리를 이해한다.
- ✓ 무손실 압축 기법을 소개한다.
- ✓ 손실 압축 기법을 소개한다.
- ✓ JPEG의 원리를 공부한다.
- ✓ MPEG의 원리를 소개한다.

Section 01 디지털 영상 압축의 개요

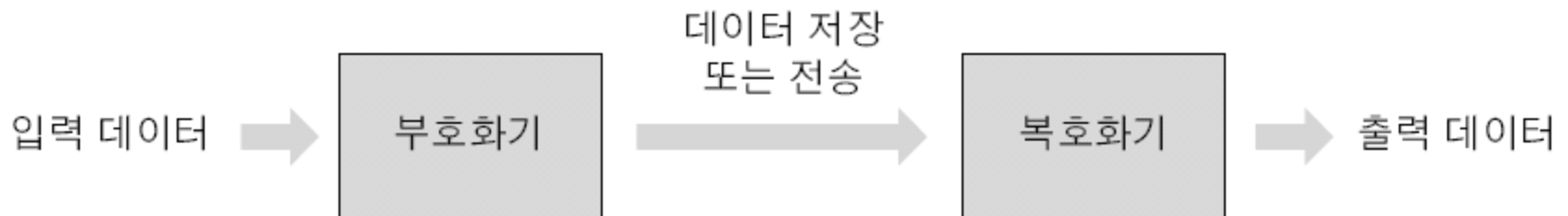
👤 압축의 필요성

- 데이터 양이 많으면 저장장치에 효율적으로 저장하기가 어려움.
- 데이터 양이 많으면 통신 네트워크로 전달하는 데 어려움이 있음.
- 데이터 양이 많은 영상은 그 양을 줄여주는 압축 기술이 필요함.

👤 멀티미디어 정보를 위한 새로운 압축 기법의 요구

- 디지털 영상의 품질 증가와 동영상 활용의 확대, 3차원 입체 동영상의 등장으로 데이터양은 더욱 증가
- 현 상황에 맞는 더 많은 압축 알고리즘이 요구됨.

👤 압축 시스템



[그림 14-1] 압축 시스템

중복성(Redundancy) 제거 기법

- 👤 데이터를 압축하는 원리는 중복된 데이터를 제거하는 것
 - 압축을 중복성 제거 기법이라 함.

통계적 중복성(Statistical Redundancy)

- 정보 이론을 기초로 데이터의 통계에서 나타나는 중복을 제거
- 허프만(Huffman) 부호화 방식이 대표적 기법

주관적 중복성(Subjective Redundancy)

- 인간의 시각 구조를 기초로 눈에 보이는 중복 데이터를 제거

공간적 중복성(Spatial Redundancy)

- 이산 코사인 변환에서 고주파를 제거하는 것처럼 인접한 화소나 블록 단위로 이들 사이의 상관 관계를 제거하여 압축을 수행

시간적 중복성(Temporal Redundancy)

- 주로 동영상에서 수행하는 압축 방법
- 프레임 간 차 영상을 전송하거나 각 오브젝트 단위로 움직임 보상을 수행하여 시간적 중복성을 제거

압축의 분류

👤 압축은 손실과 무손실 압축으로 분류

👤 손실(Loss) 압축

- 중복되고 불필요한 정보의 손실을 허용 데이터양을 줄이는 방법
- 삭제된 데이터가 시각에 미치는 영향은 작지만 영상의 품질은 저하
- 사라진 데이터 때문에 정확하게 영상을 복원하는 것은 어려움.
- 불필요한 데이터를 버렸기 때문에 압축 성능은 우수함
- 절단 부호화, 양자화, 벡터 양자화, 손실 예측 부호화, 변환 부호화

👤 무손실(Lossless) 압축

- 압축 과정에서 데이터의 손실이 전혀 없어 입력 영상과 복원된 영상이 완전히 같음.
- 버린 데이터가 없어 압축 효율은 떨어짐.
- 허프만 부호화, 산술 부호화, 실행 길이 부호화
- 의료 영상, 설계도면 등 자료를 손실하면 안 되는 분야에서 응용

효율적인 압축 방법 선택을 위한 고려 사항

👤 압축률

- 압축률은 원래 자료의 크기를 압축된 자료의 크기로 나눈 것
- 압축률이 높을수록 데이터의 크기는 작지만 저장장치와 전송 상태를 고려하여 정하는 게 바람직

👤 압축(Encoding)/복원(Decoding) 시간

- 압축과 복원을 수행하는 데 걸리는 시간
- 비디오 스트리밍 서비스는 실시간으로 압축하고 복원이 필요하나 전자메일에서 사용하는 영상은 비실시간으로 압축하고 복원해도 됨.

👤 압축 알고리즘의 복잡도

- 압축과 복원을 빠르게 하려면, 복잡도가 낮은 것을 선택해야 함.
- 속도는 구애 받지 않지만 그대신 높은 압축률을 요구한다면 복잡도가 높은 것을 선택하는 게 더 효율적
- 더불어 계산되는 자원의 가용성과 비용도 고려되어야 할 것

👤 표준화 여부

- 표준화된 압축 기법을 사용하면 전송된 압축 데이터의 복원 문제가 발생하지 않아 표준 기법의 고려가 필요함.

다양한 압축 부호화 기법

- 엔트로피 부호화(Entropy Coding) : 무손실 부호화와 같은 뜻
- 정보원 부호화(Source Coding) : 영상의 정보를 표현하는 첫 번째 단계에서 압축을 수행하는 것, 보편적으로 손실 부호화를 수행



[그림 14-2] 다양한 압축 부호화 기법

👤 실행 길이 부호화

- 자료의 반복성을 이용하는 가장 간단한 압축 기법으로, 만화처럼 단순 그림에 적합한 압축 기법

- 20바이트 자료의 실행 길이 부호화

BBBBFFFFFFEEEETPPRR

- 반복되는 자료를 반복 횟수만 표현

4B6F4E1T3P2R

- 압축률의 계산 : 데이터 양이 20바이트에서 12바이트로 감소

압축률 = 20바이트/12바이트 = 1.6

수정 실행 길이 부호화

👤 자료가 반복하지 않는 경우 실행 길이 부호화는 비효율적

👤 19바이트의 자료

WelcomeToImageWorld

👤 실행 길이 부호화를 적용하면 38바이트로 부호화

1W1e1l1c1o1m1e1T1o1I1m1a1g1e1W1o1r1l1d

👤 압축률 : 19바이트/38바이트 = 0.5(성능 저하)

👤 보완 방법

- 반복되지 않는 데이터는 그대로 표현한 채 반복되는 자료에만 실행 길이 부호를 적용
- 반복되는 자료의 실행 길이 부호화 앞에 특수문자를 삽입
- 20바이트의 자료가 수정 실행 길이 부호화를 수행하면 16바이트로 감소

GAFECETAAAAAAEPPPPDE ⇒ GAFECET*6AE*4PDE

영상에서의 실행 길이 부호화

- 👤 실제 영상에서는 연속해서 나오는 화소의 개수를 부호화함.
- 👤 이진 영상에서는 0의 반복 횟수를 먼저 표현한 뒤 1의 반복 횟수를 표현
- 👤 그레이 영상에서는 (화소 값, 반복 횟수) 형식으로 표현

(a) 이진 영상의 예

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

첫 번째 라인 : 8
 두 번째 라인 : 1, 3, 4
 세 번째 라인 : 0, 2, 2, 4

(b) 그레이 영상의 예

10	10	10	10	10	10	10	10
10	10	20	20	20	20	20	20
20	20	20	20	20	10	10	10

첫 번째 라인 : (10, 8)
 두 번째 라인 : (10, 2) (20, 6)
 세 번째 라인 : (20, 5) (10, 3)

[그림 14-3] 영상에서의 실행 길이 부호화

허프만 코딩

- 👤 데이터의 발생 빈도 수에 따라 다른 길이의 부호를 사용하는 가변 길이 부호화(Variable Length Coding) 방법
- 👤 빈번하게 발생하는 자료를 표현할 때는 적은 수의 비트를 할당하고, 드물게 발생하는 자료를 표현할 때는 더 많은 비트를 할당
- 👤 가변 길이의 허프만 부호 예
 - 고정 길이 부호화에서는 문자 하나를 8비트로 표현. 반면, 가변 길이 부호화에서는 문자 하나의 비트 수가 일정치 않음.

toupee t o u p e e (48bit)

 111 0100 10111 10110 100 100 (23bit)

압축률 : 2.08

허프만 부호화의 전치 특성(Prefix Property)

- ❶ 한 코드가 다른 코드 앞에 위치할 수 없다는 것
 - 'e'가 '01'로 표현되면 '010'은 다른 문자로 부호화될 수 없어야 함.
- ❷ 복호화 과정에서 문자의 길이를 알 수 없기 때문
 - '010'이 'f'로 부호화되었다면 복호기는 '01'을 읽고 'e'로 판단할 수도 있고, '010'으로 읽고 'f'로 판단할 수도 있기 때문
- ❸ 복호기에서 '010...'이라는 비트 스트림을 읽을 때 '01'이 들어오면 'e'를 출력하고, 다음의 '0'에서 다시 읽기를 시작할 수 있음.

히프만 코드 생성 알고리즘

📍 일곱 가지 색이 있는 영상

R	K	G	G	B	R	K	R
G	B	R	R	R	G	R	G
G	R	G	G	C	B	R	K
R	K	R	K	Y	K	B	R
G	R	B	R	R	R	R	C
R	K	G	K	R	G	G	K
K	C	G	K	M	C	G	K
M	K	K	K	G	K	K	G

📍 색상의 발생 빈도 수에 따른 배열

색상	빈도 수
빨간색(R)	19
검정색(K)	17
녹색(G)	16
파랑색(B)	5
남색(C)	4
자홍색(M)	2
노랑색(Y)	1

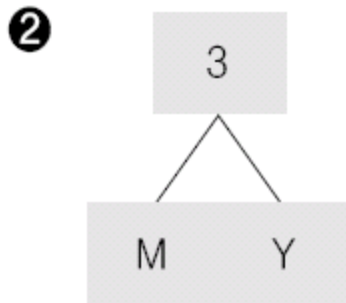
이진 트리를 사용한 허프만 테이블 구성[1]

① 처음에는 모든 색을 각기 독립된 노드인 자유 노드로 간주함.

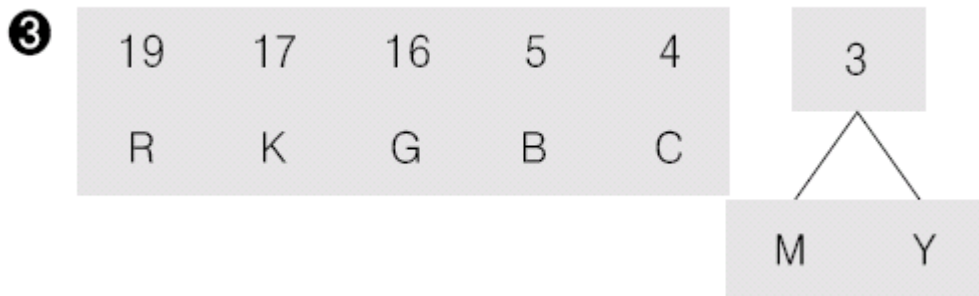
①

19	17	16	5	4	2	1
R	K	G	B	C	M	Y

② 빈도 수가 가장 낮은 자유 노드 두 개를 자식 노드로 하는 부모 노드를 한 개 생성. 부모 노드는 자식 노드의 합과 가중치가 같게 됨.

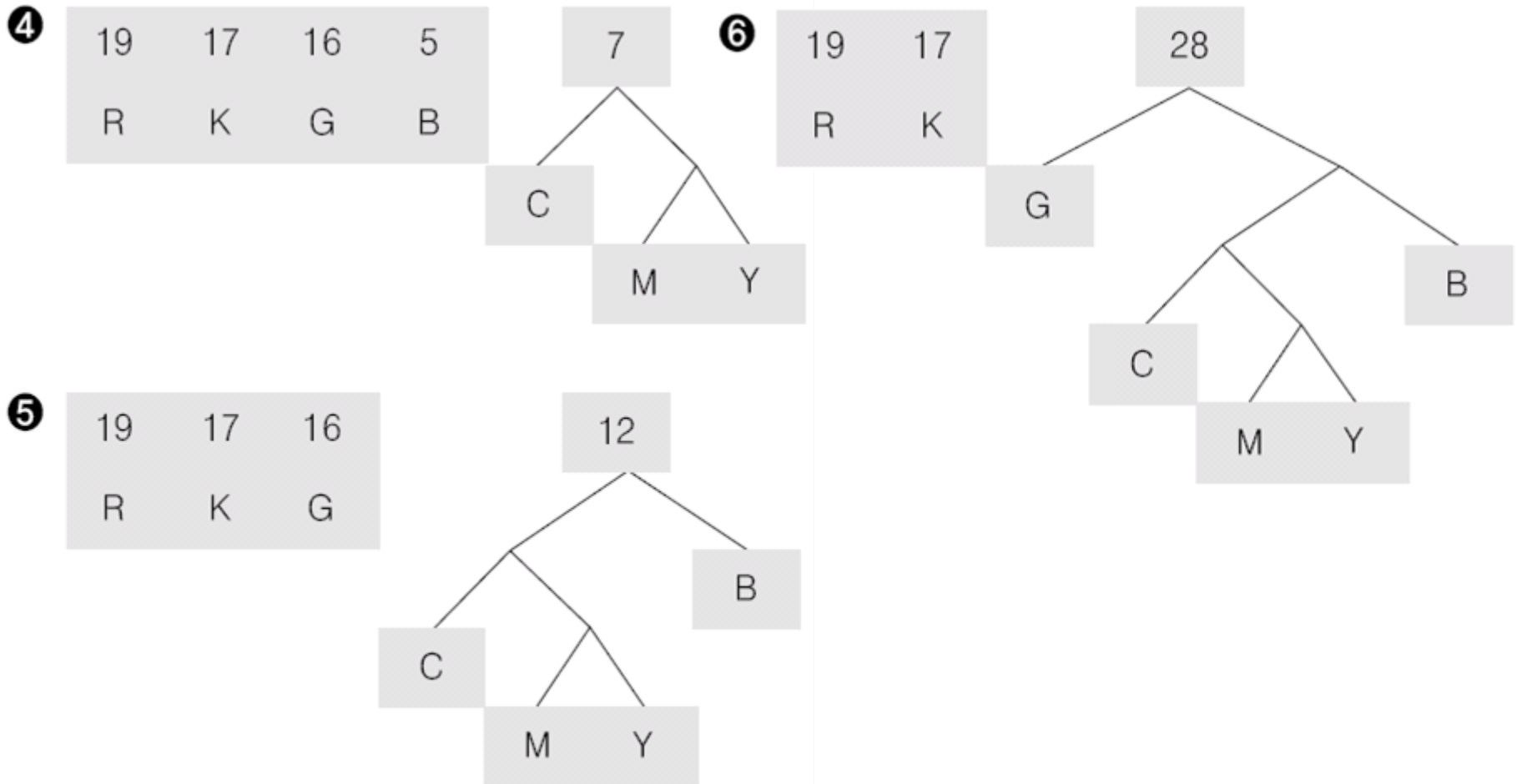


③ 연결한 자식 노드 두 개는 자유 노드 목록에서 제거하고, 새롭게 생성된 부모 노드는 목록에 추가



이진 트리를 사용한 허프만 테이블 구성[2]

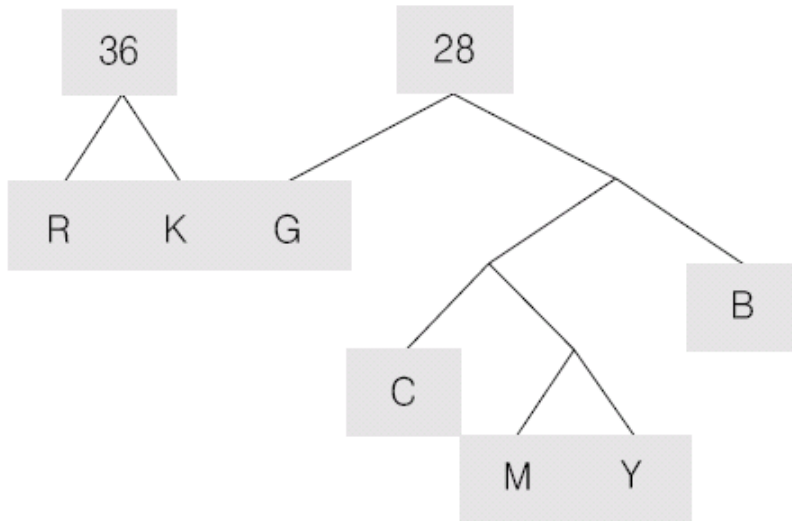
④ 자유 노드가 하나만 남을 때까지 ② 단계와 ③ 단계를 반복 수행. 이 자유 노드는 구성하는 트리의 루트가 됨.



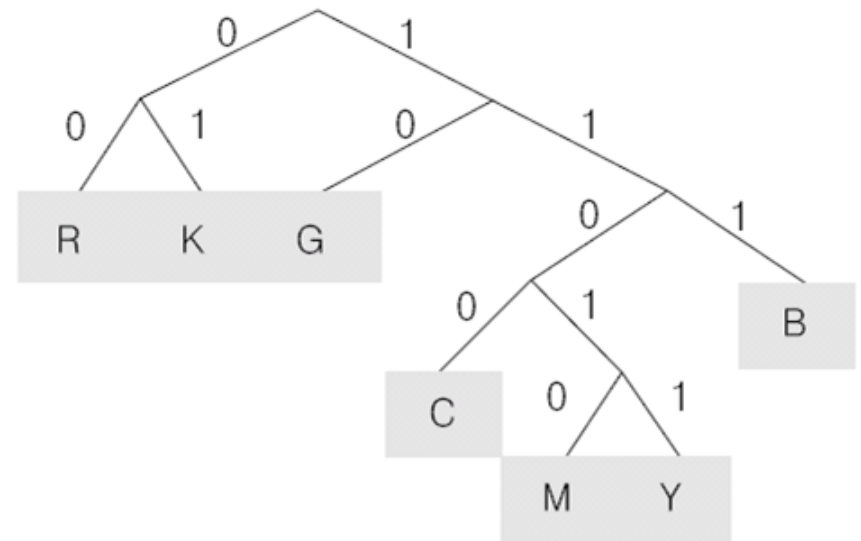
이진 트리를 사용한 허프만 테이블 구성[3]

- ④ 자유 노드가 하나만 남을 때까지 ② 단계와 ③ 단계를 반복 수행. 이 자유 노드는 구성하는 트리의 루트가 됨.

⑦



⑧



각 색상을 허프만 부호화한 결과

- ❶ 최종으로 만든 루트 트리의 각 가지에 임의적으로 0과 1을 할당함.
- ❷ 그런 뒤 아래쪽 가지부터 할당된 비트를 읽어 각 색상의 부호 결정

[표 14-2] 각 색상을 허프만 부호화한 결과

색상	압축 비트
빨간색(R)	00
검정색(K)	01
녹색(G)	10
파랑색(B)	111
남색(C)	1100
자홍색(M)	11010
노랑색(Y)	11011

무손실 차분 부호화

차분 부호화

- 선행 데이터와 현재 데이터 값의 차이만 전송하는 방법
- 값의 차이를 정확하게 표현할 수 있다면 손실이 없는 부호화도 가능
- 값 차이가 너무 크거나 작은 때는 정확한 부호화가 어렵고 손실 부호화가 나타남.

영상에서 무손실 차분 부호화

- 첫 번째 화소는 그냥 부호화
- 나머지 화소는 차이만을 부호화
- 차이 값은 비트 수로도 표현

무손실 차분 부호화

	원본 영상 (8비트)	차분 부호화 (6비트)
화소 #1	30	30
화소 #2	55	25
화소 #3	83	28
화소 #4	64	-19
화소 #5	90	26
총 비트 수	$8 \times 5 = 40$	$6 \times 5 = 30$

[그림 14-5] 영상의 무손실 부호화 예

📍 공간 해상도를 저하시키는 압축

- 영상의 크기를 나타내는 공간 해상도를 줄이면, 데이터가 버려져 영상의 크기는 작아지고, 작아진 영상의 품질은 저하됨.
- 원래 크기로 만들려면 선형 보간(Linear Interpolation)을 이용하여 복원하면 되지만 원본 영상과 똑같지는 않음.



(a) 원본 영상



(b) 다운 샘플링 영상



(c) 보간 영상

[그림 14-6] 공간 해상도를 변경하여 압축한 영상

양자화를 이용한 압축

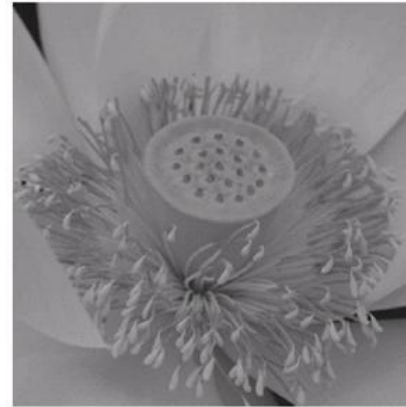
- ❶ 화소 밝기 값의 자릿수 범위를 줄여 데이터를 압축하는 방법
- ❷ 각 화소를 표현하는 비트 수를 줄이면 영상의 데이터양을 줄일 수 있는 압축을 수행
- ❸ 8비트 그레이 영상에서 4비트 그레이 영상으로 변경
 - 밝기 범위가 0~225에서 0~15로 좁아져 정확도가 떨어짐.
- ❹ 극단적으로 1비트로 밝기를 표현
 - 데이터양은 많이 줄어드나 이진 영상이 되어 품질의 저하



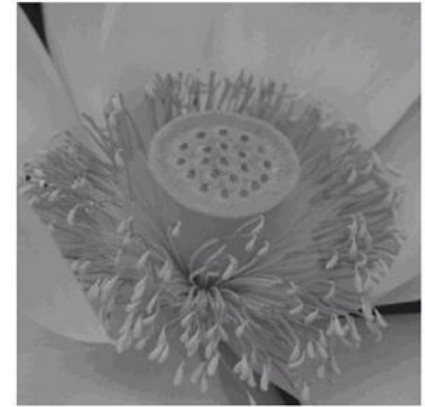
(a) 8비트



(b) 7비트



(c) 6비트

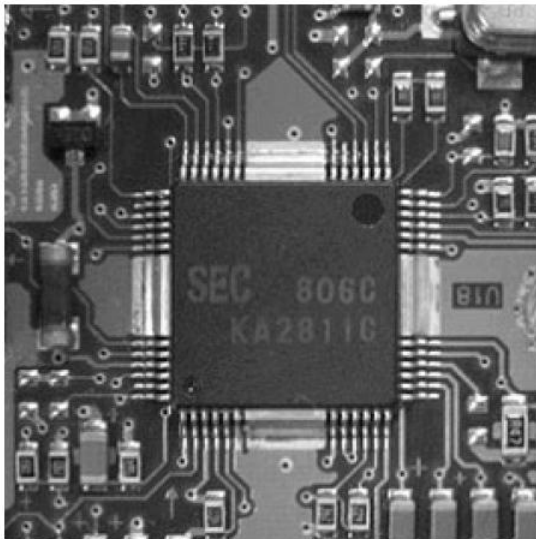


(d) 5비트

[그림 14-7] 화소를 표현하는 비트를 줄여서 압축한 결과 영상

이산 코사인 변환

- 영상의 주파수 성분을 순차적으로 배열
 - 왼쪽 위에는 저주파 성분이 있고, 대각선 방향인 오른쪽 아래로 진행될수록 고주파 성분이 분포
- 고주파 성분은 영상에 중요한 정보가 들어 있지 않으므로 이 성분을 제거해도 눈으로는 품질이 떨어진 것을 알지 못함.
- 고주파 성분을 제거했으므로 데이터의 양은 감소



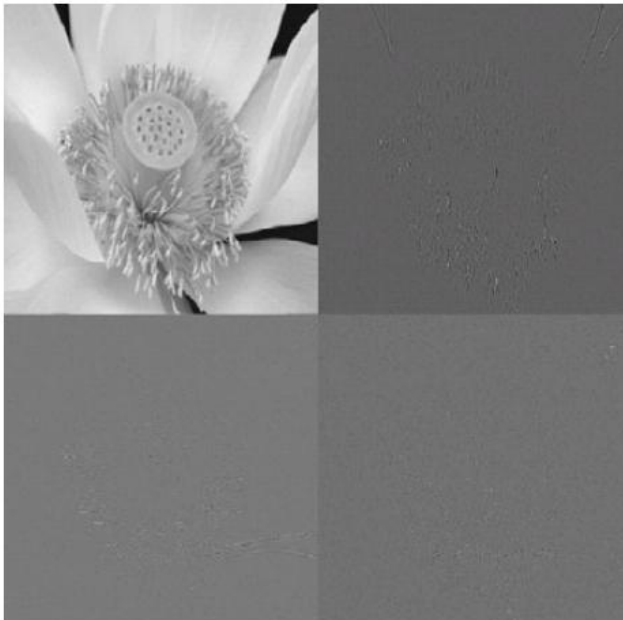
(a) 입력 영상



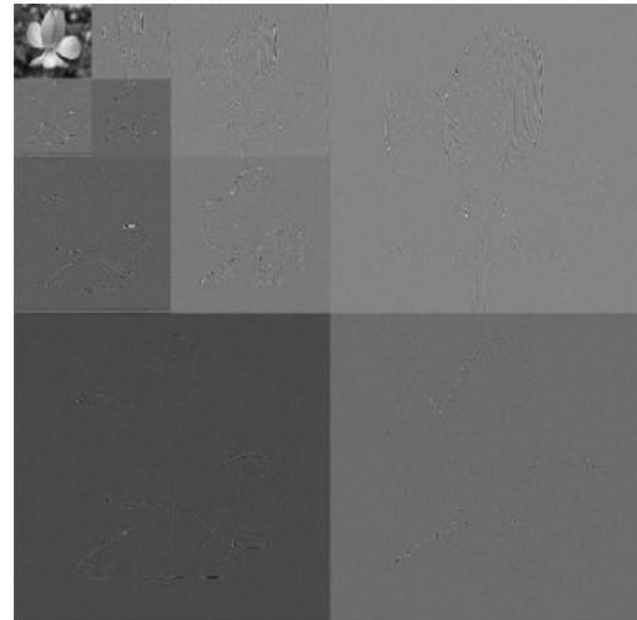
(b) 결과 영상

웨이브렛 변환

- 생성된 부영상을 옥타브 트리 구조로 배치하면 이산 코사인 변환과 같은 주파수 배열을 얻을 수 있음 → 고주파 성분을 쉽게 제거 가능
- 옥타브 트리 구조
 - (a)는 1단계만 수행한 것으로 오른쪽 아래가 고주파 성분
 - (b)는 웨이브렛 변환을 삼 단계로 수행한 것으로 왼쪽 위에서 대각선 방향으로 진행할수록 고주파 성분



(a) 1단계만 수행



(b) 3단계 모두 수행

JPEG(Joint Photographic Expert Group)

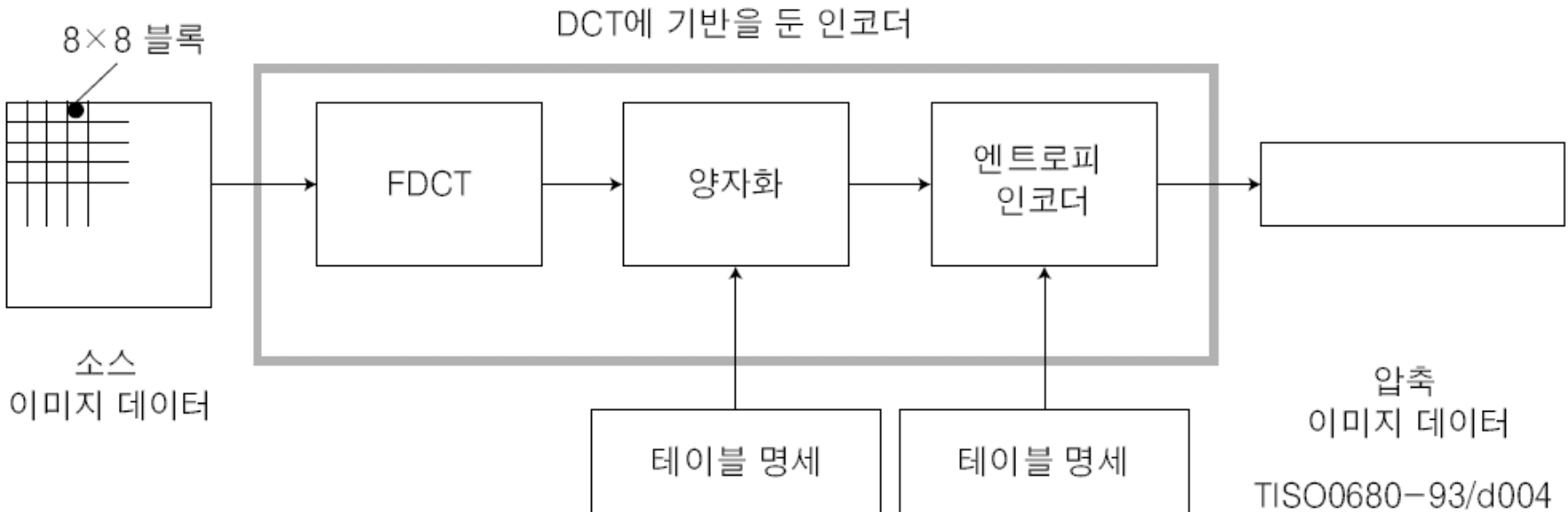
- 국제 표준화기구(ISO)와 국제전기통신연합의 산하 단체인 CCITT에서 1992년 제정한 국제 표준 알고리즘
- 기본적으로 화상의 크기, 컬러 시스템 및 컬러 성분에 제한을 두지 않음.
- 압축 모드로는 순차적 부호화, 점진적 부호화, 계층적 부호화, 무손실 부호화가 있음.

JPEG의 기본 개념

- 여러 가지 응용 목적에 맞게 부호화 방식을 선택하여 필요한 부분을 선택적으로 적용할 수 있음.
- 컬러 정지 화상의 부호화 방식을 폭넓게 수용하여 응용 분야에 따라 부호화 방식이나 사용하는 매개변수 등을 적절히 선택·조합해서 사용할 수 있음.

JPEG의 압축 단계

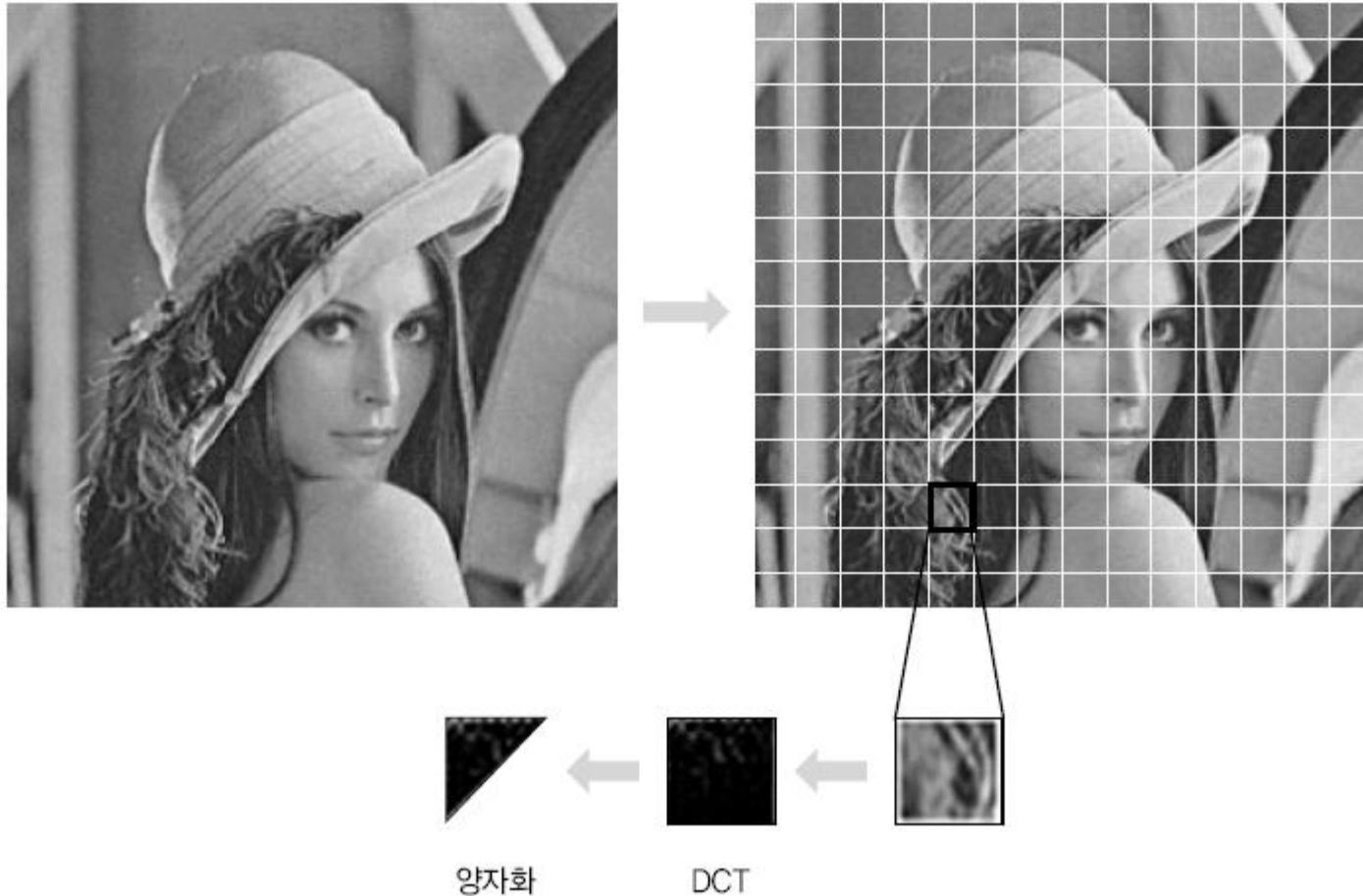
- ① 영상의 컬러 모델을 RGB에서 YCbCr(Y=휘도, CbCr=색도)로 변환
- ② 색상을 나타내는 색도 성분 다운 샘플링(옵션)
- ③ 영상을 8×8 화소의 블록으로 분할하고 각 블록에서 이산 코사인 변환(DCT) 실행
- ④ DCT 계수를 양자화
- ⑤ 양자화된 DCT 계수를 허프만 부호화 방법을 사용하여 부호화



[그림 14-10] JPEG의 압축 단계

실제 영상에서 JPEG 압축

- 양자화까지 수행되면 허프만 부호화로 디지털 스트림의 압축된 영상 데이터가 생성



[그림 14-11] 실제 영상에서 JPEG 압축을 수행하는 과정

RGB 컬러 모델

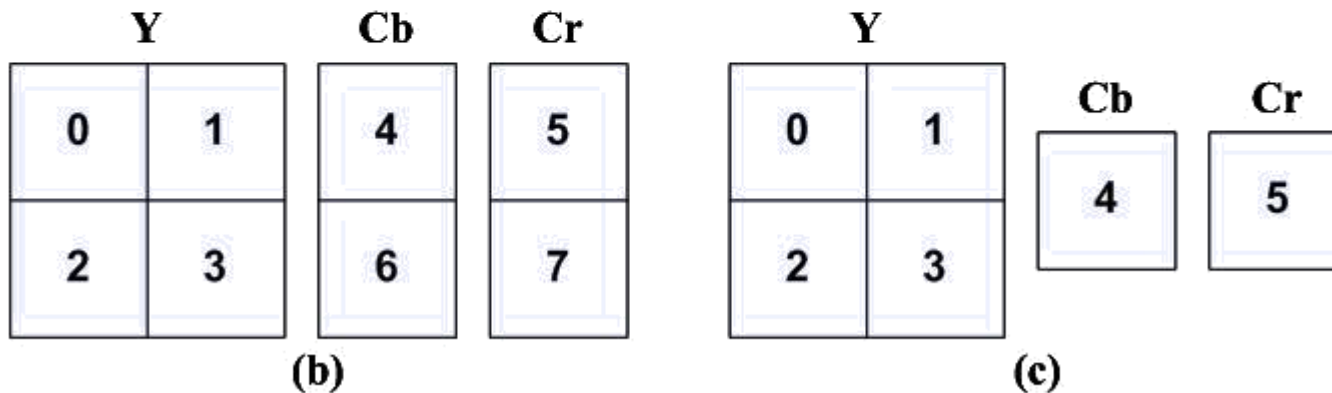
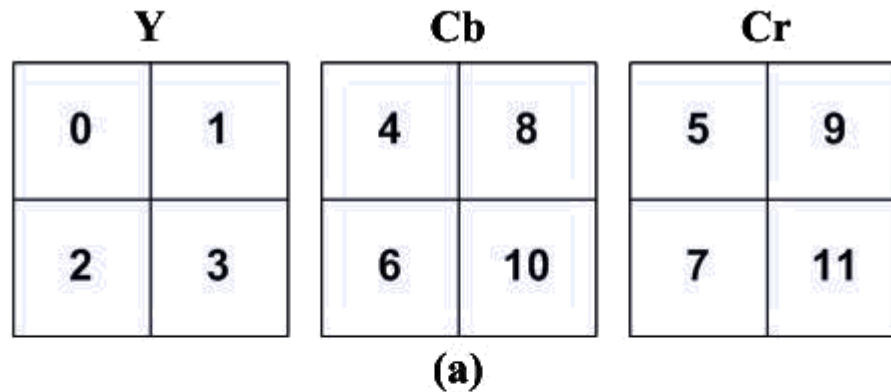
- 채널 간에 상관도가 높아 영상처리 수행 후 원래의 컬러를 보존하기 어려움.
- 색 채널 세 개를 유지해야 하므로 그만큼 데이터양도 많음.

JPEG에서는 YCrCb 모델 사용

- 인간의 시각적인 특성을 이용할 수 있기에 효율적
- 밝기 성분에 해당되는 Y의 휘도 구성요소는 그대로 유지, 색도 구성요소 Cr, Cb는 약간의 정보를 제거해도 눈에 거슬리지 않음.
- 색도 성분은 수평과 수직 방향으로 다운 샘플링됨.

색도 구성요소의 다운 샘플링

- (a): 화소 네 개로 구성된 휘도와 색도 성분
- (b): 색도 성분을 수직 방향으로 다운 샘플링한 결과
- (c): 수직과 수평 방향으로 다운 샘플링된 결과

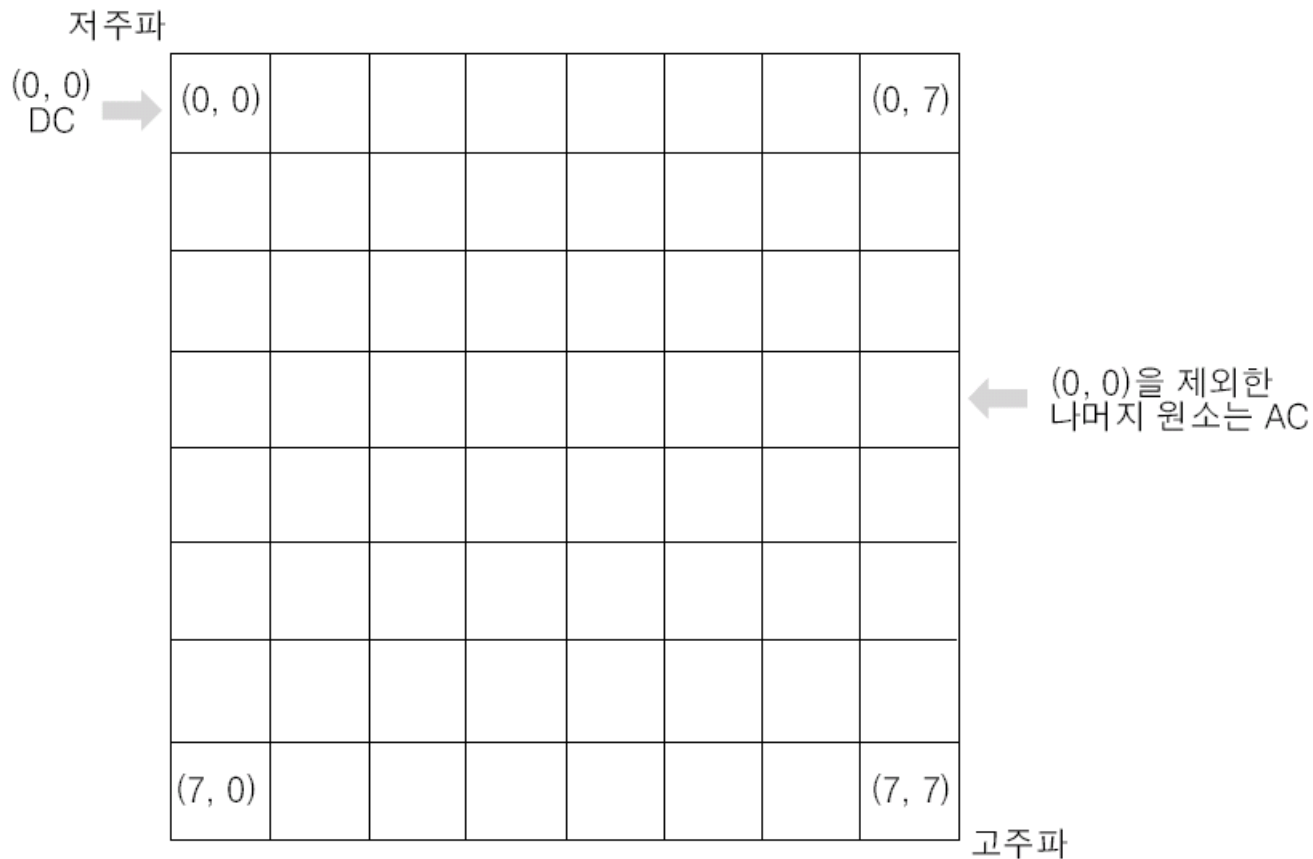


[그림 14-12] 색도 구성요소의 다운 샘플링

이산 코사인 변환 단계



8×8 블록이 DCT되었을 때의 구성

- (0, 0)은 직류 성분이고 나머지 계수는 AC 성분
- 왼쪽 위가 저주파, 대각선 방향으로 오른쪽 아래로 갈수록 고주파



[그림 14-13] 8×8 블록의 DCT 구성

이산 코사인 변환 단계(계속)

-  **DCT 계수에서 DC 계수 값은 아주 큼.**
 - 에너지가 집중되어 있고, 영상의 주요 성분이 포함되어 있음.
-  **고주파쪽으로 갈수록 계수가 작아짐**
 - 에너지와 중요한 정보가 작음.

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	161	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	161	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

(a) 입력 영상

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

(b) DCT 계수

[그림 14-14] 영상의 임의의 한 블록에서 수행한 DCT 결과

DCT 계수의 양자화

- 👤 영상의 중요 정보가 아닌 고주파 성분을 제거하는 과정
- 👤 DCT 계수를 양자화 테이블의 양자화 계수로 나눈 뒤 반올림해서 수행
- 👤 양자화 테이블에서 양자화 계수
 - DC와 저주파 성분이 있는 왼쪽 윗부분에서는 값이 작음.
 - 고주파 성분의 오른쪽 아랫부분으로 이동할수록 크기가 작음.
- 👤 양자화 계수로 DCT 계수를 나누면 뚫은 DC와 저주파 부분은 크고 고주파 부분은 작아서 거의 0에 가까움.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a) 양자화 계수

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	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	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) 양자화 결과

양자화된 DCT 계수의 부호화(계속)

👤 DC 값의 인코딩(1)

- 이전 블록 DC 값과의 차이 값을 부호화하는 차등 부호화 방식을 이용
- 심벌-1
 - 0의 런 길이(RUNLENGTH)는 고려하지 않고, 이전 블록의 DC 값과의 차이를 부호화하는 데 필요한 비트 수를[표 14-3]을 참고하여 표시

[표 14-3] 차분 DC 값의 표현을 위한 비트 수

SIZE	AMPLITUDE	부호어	
		밝기 성분 DC	색 성분 DC
1	-1,1	010	01
2	-3,-2,2,3	011	10
3	-7,-6,-5,-4,4,5,6,7	100	110
4	-15...-8,8...15	101	1110
5	-31...-16,16...31	110	11110
6	-63...-32,32...63	1110	111110
7	-127...-64,64...127	11110	1111110
8	-255...-128,128...255	111110	11111110
9	-511...-256,256...511	1111110	111111110
10	-1023...-512,512...1023	11111110	1111111110

양자화된 DCT 계수의 부호화(계속)

■ 심벌-2

- 이전 블록의 DC 값과의 차이 값을 [표 14-4]를 참고하여 표시

[표 14-4] 계수 값의 부호화 테이블

심벌-2	코드	심벌-2	코드	심벌-2	코드	심벌-2	코드
-1	0	-15	0000	-31	00000	16	10000
1	1	-14	0001	-30	00001	17	10001
-3	00	-13	0010	-29	00010	18	10010
-2	01	-12	0011	-28	00011	19	10011
2	10	-11	0100	-27	00100	20	10100
3	11	-10	1010	-26	00101	21	10101
-7	000	-9	0110	-25	00110	22	10110
-6	001	-8	0111	-24	00111	23	10111
-5	010	8	1000	-23	01000	24	11000
-4	011	9	1001	-22	01001	25	11001
4	100	10	1010	-21	01010	26	11010
5	101	11	1011	-20	01011	27	11011
6	110	12	1100	-19	01100	28	11100
7	111	13	1101	-18	01101	29	11101
		14	1110	-17	01110	30	11110
		15	1111	-16	01111	31	11111

DC 계수를 부호화하는 예

- 이전 블록의 DC 값을 18, 현재 블록의 DC 값을 15라고 가정하면, 차분 DC 값은 -3.
- -3의 부호화를 위한 비트 수(SIZE)
 - 2비트([표 14-3] 참고)
- 숫자 2의 코드
 - 011([표 14-3]의 밝기 성분 DC) → 심벌-1 = 011
- -3의 코드 값
 - 00([표 14-4] 참고) → 심벌-2 = 00
- 결과 코드 값 : 01100

양자화된 DCT 계수의 부호화(계속)

AC 값의 부호화

■ 심벌-1

- (RUNLENGTH, SIZE) 형태로 표기

- RUNLENGTH는 0의 반복되는 횟수

- SIZE는 0이 아닌 수를 부호화하는 데 필요한 비트 수 [표 14-5] 참고

■ 심벌-2에서는 [표 14-4]에 근거하여 0이 아닌 수를 부호화

[표 14-5] AC 계수 값의 허프만 테이블

RUN LENGTH	SIZE	부호어	RUN LENGTH	SIZE	부호어
0	0	1010(EOB)	1	4	1111 1011 0
0	1	00	1	5	1111 1110 110
0	2	01	1	6	1111 1111 1000 1011
0	3	100	1	7	1111 1111 1000 0101
0	4	1011
0	5	1101 0	2	1	1110 0
0	6	1111 000	2	2	1111 1001
...	2	3	1111 1101 11
1	1	1100	2	4	1111 1111 0111
1	2	1101 1	2	5	1111 1111 1000 1001
1	3	1111 001	2	6	1111 1111 1000 1010

양자화된 DCT 계수의 부호화(계속)

👤 AC 계수를 부호화하는 예

- 양자화된 DCT 계수

15, 0, -2, -1, -1, -1, 0, 0, -1, EOB

- DC와 AC 계수를 심벌-1과 심벌-2로 표현

(2) (-3), (1,2) (-2), (0,1)(-1), (0,1)(-1), (0,1)(-1), (2,1) (-1), (0,0)

- (1,2)(-2)

- RUNLENGTH = 1, SIZE = 2 → 심벌-1 = 1101 1([표 14-5] 참고)
- -2 → 심벌-2 = 01([표 14-4] 참고)

- (0,1)(-1)

- RUNLENGTH = 0, SIZE = 1 → 심벌-1 = 00([표 14-5] 참고)
- -1 → 심벌-2 = 0([표 14-4] 참고)

- (2,1)(-1)

- RUNLENGTH = 2, SIZE = 1 → 심벌-1 = 11100([표 14-4] 참고)
- -1 → 심벌-2 = 0([표 14-4] 참고)

부호화 결과

심벌	코드
(2)(-3)	011 00
(1,2)(-2)	11011 01
(0,1)(-1)	00 0
(0,1)(-1)	00 0
(0,1)(-1)	00 0
(2,1)(-1)	11100 0
(0,0)	1010

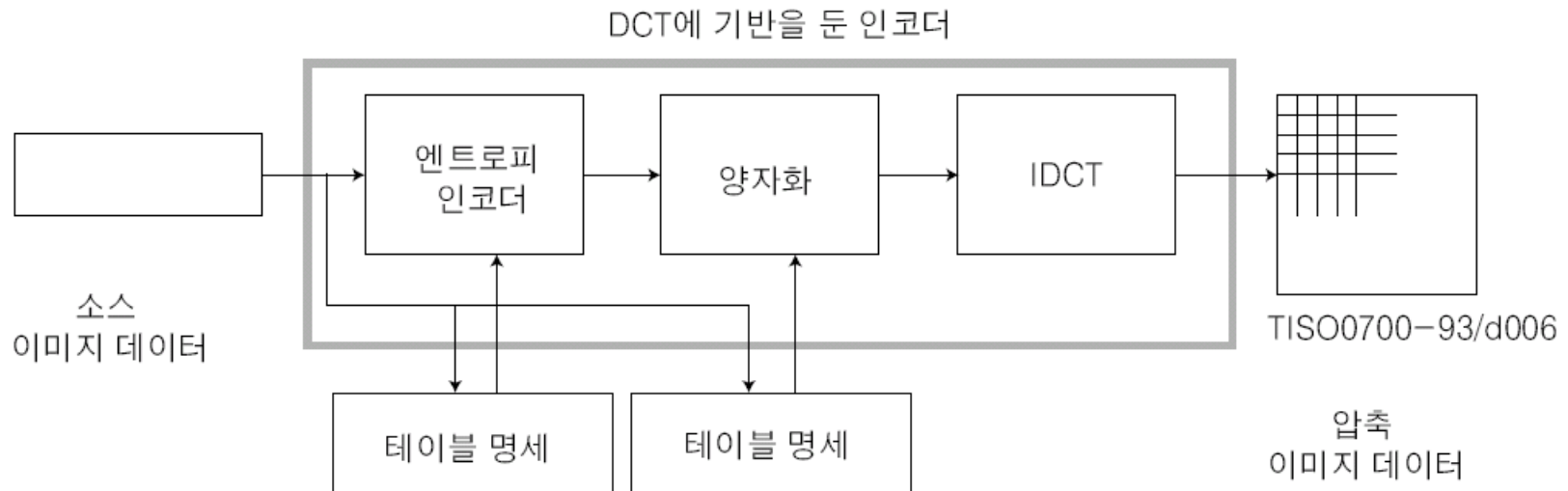
압축의 성능

- 8비트로 화소 값을 표현 : 총 데이터양은 $64 \times 8 = 512$ 비트
- JPEG 압축을 수행 : 31비트로 감소

0110011011010000000001110001010

JPEG 복호화 단계

- 복호화 단계는 압축 과정의 역순으로 진행
 - 압축된 부호화 값을 다시 심벌 형태로 변환
 - 심벌은 양자화 계수 값으로 복원
 - 역양자화를 수행하여 DCT 계수를 얻음.
 - 역방향 DCT(Inverse DCT)를 수행하여 공간 영역의 영상 정보를 생성
 - YCrCb의 컬러 정보를 컬러모델 변환하여 다시 RGB 영상을 생성

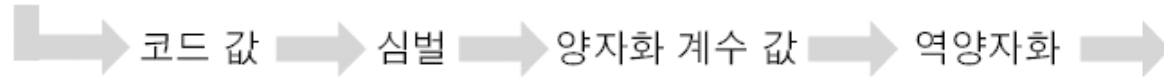


[그림 14-17] JPEG의 복호화 과정

JPEG 복호화 단계

👤 압축 데이터의 복호화 과정에서 생성된 데이터

0110011011010000000001110001010



240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	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	0	0	0	0	0
0	0	0	0	0	0	0	0

IDCT

144	146	149	152	154	156	156	156
155	156	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

[그림 14-18] 압축 데이터의 복호화 과정으로 생성된 데이터

MPEG(Moving Picture Experts Group)

- 동영상을 압축하는 국제 표준
- ISO/IEC(International Electrotechnical Commission)에서 1988년에 설립한 동영상전문가그룹
- 시간에 따라 연속적으로 변화하는 동영상 압축과 부호를 표현하여 정보를 전송할 수 있는 방법을 연구
- 동영상과 오디오를 조합하려는 국제적 표준을 위해 개발
- 최근에는 멀티미디어용 비디오 압축 기술까지도 필요성이 증대되어 이 분야의 개발이 활발히 진행되고 있음.

MPEG-1

- 1991년 ISO에서 규격화한 영상 압축 기술
- CD-ROM 등 디지털 저장매체에 VHS 테이프 수준의 동영상과 음향을 최대 1.5Mbps로 압축·저장
- 해상도는 352×240이고, 초당 보여 주는 화면의 개수를 나타내는 프레임률은 30Frames/Second, 2채널 스테레오 오디오 지원
- 비디오 CD와 CD-I/FMV

MPEG-2

- 1994년 규격화한 영상 압축 기술
- 디지털 TV, 대화형 TV, DVD 등을 압축하려고 MPEG-1을 개선한 것
- 4~15Mb/sec 전송 속도를 목표로 하며, 해상도는 1,920×1,152까지 지원, 프레임률은 30frames/sec을 만족
- 비월주사(Interlaced Video) 방식 지원
- 현재, DVD 등의 컴퓨터 멀티미디어 서비스, 직접 위성 방송, 유선 방송, 고화질 TV 등 방송서비스, 영화나 광고 편집 등에서 널리 사용

MPEG-4

- 멀티미디어 통신을 전제로 만든 영상 압축 기술로, 1998년 완성
- 낮은 전송률로 동화상을 보내려고 개발된 데이터 압축과 복원 기술의 새로운 표준
- 데이터를 객체(Object) 단위로 처리
 - 영상 내의 의미 있는 내용물을 각각 다른 객체로 부호화하고, 이를 혼합하여 하나의 화면을 구성
 - 추후 원하는 객체만을 추출하여 재사용하거나 객체를 이용하여 전체 데이터를 재구성할 수 있음.
- 이런 특징 때문에 64Kbps, 19.2Kbps의 저속 전송으로 동화상 구현
- 내용 기반의 대화형 기능(Content-based Interactivity)과 다양한 전송 환경의 수용성 및 오류에 강한 특성
 - 인터넷 유선망과 이동통신망 등 무선망에서 멀티미디어 통신, 화상회의 시스템, 컴퓨터, 방송, 영화, 교육, 오락, 원격 감시 등의 분야에서 널리 사용

MPEG-7

- 동영상 데이터 검색과 전자상거래 등에 적합하도록 개발된 차세대 동영상 압축 재생 기술로 1996년부터 표준화 작업이 시작
- 색상이나 물체의 모양 정보를 입력하는 것만으로도 웹에서 필요한 멀티미디어 자료를 찾을 수 있는 기술
- 전자도서관, 멀티미디어 디렉터리 서비스(Multimedia Directory Services), TV/Radio 방송국, 멀티미디어 편집(Multimedia Editing) 등의 분야에서 활용이 예상

MPEG-21

- MPEG-1과 MPEG-2, MPEG-4 등 MPEG 관련 기술을 통합하여 디지털 콘텐츠의 제작 및 유통, 보안 등의 모든 과정을 관리할 수 있게 하는 기술
- 콘텐츠 제작자와 유통업자, 최종 사용자가 편리하게 국제적 호환성으로 콘텐츠를 식별하고, 관리하며, 보호할 수 있도록 하는 멀티미디어 프레임워크 핵심 기술의 표준화를 목표

MPEG의 기본 개념

👤 공간 압축 기법

- 프레임 단위로 압축을 수행
- 가장 인접 화소 간의 상관 관계를 제거하려고 DCT를 이용
- DCT 변환 뒤 양자화, 엔트로피 부호화가 수행됨.
- 인트라 프레임 부호화(Intraframe Coding)

👤 시간적 압축 방법

- MPEG에서 추가된 압축 방법
- 인접한 프레임 간의 상관 관계를 이용하는 프레임 간 예측(Frame-to-frame Prediction)을 수행
- 움직임 예측(Motion Prediction)으로 움직임 벡터(Motion Vector)를 구하고, 이것을 이용하여 다시 움직임 보상(Motion Compensation)을 수행
- 움직임 벡터, 원래의 프레임과 움직임 보상된 프레임 간의 차 영상등이 부호화
- 인터 프레임 부호화(Interframe Coding)

MPEG의 기본 개념(계속)

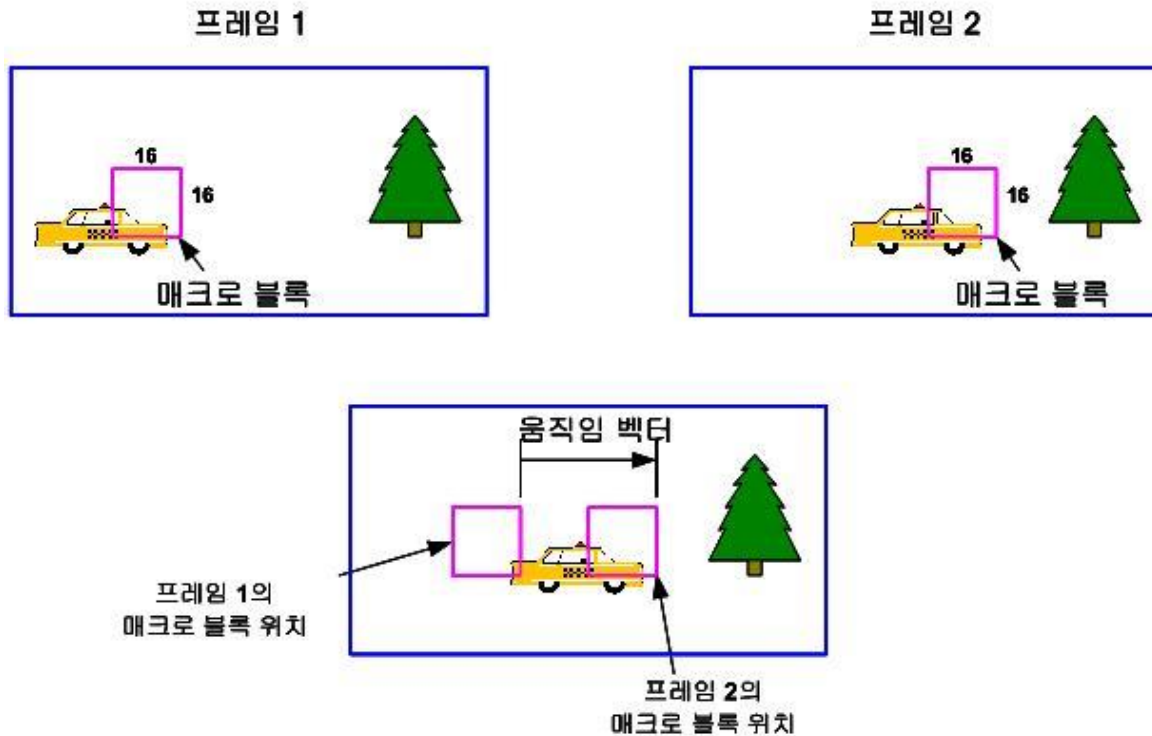
움직임 예측으로 움직임 벡터를 찾는 과정

■ 매크로 블록(Macro Block)

- 움직임 예측을 수행하려고 영상을 크기가 16×16 인 블록으로 분할

■ 매크로 블록이 다음 프레임 어떤 위치로 이동했는지를 검사

- 프레임 1의 자동차 앞부분에 해당되는 매크로 블록이 프레임 2의 어디로 이동했는지를 찾는 것



[그림 14-20] 움직임 예측에서 움직임 벡터를 찾는 과정

움직임 벡터로 움직임 보상을 수행하는 과정

■ 움직임 벡터

- 프레임 1의 매크로 블록이 프레임 2의 어느 곳에 위치하는지를 알려주어 프레임 2는 프레임 1에서 복사하여 사용

■ 움직임 보상

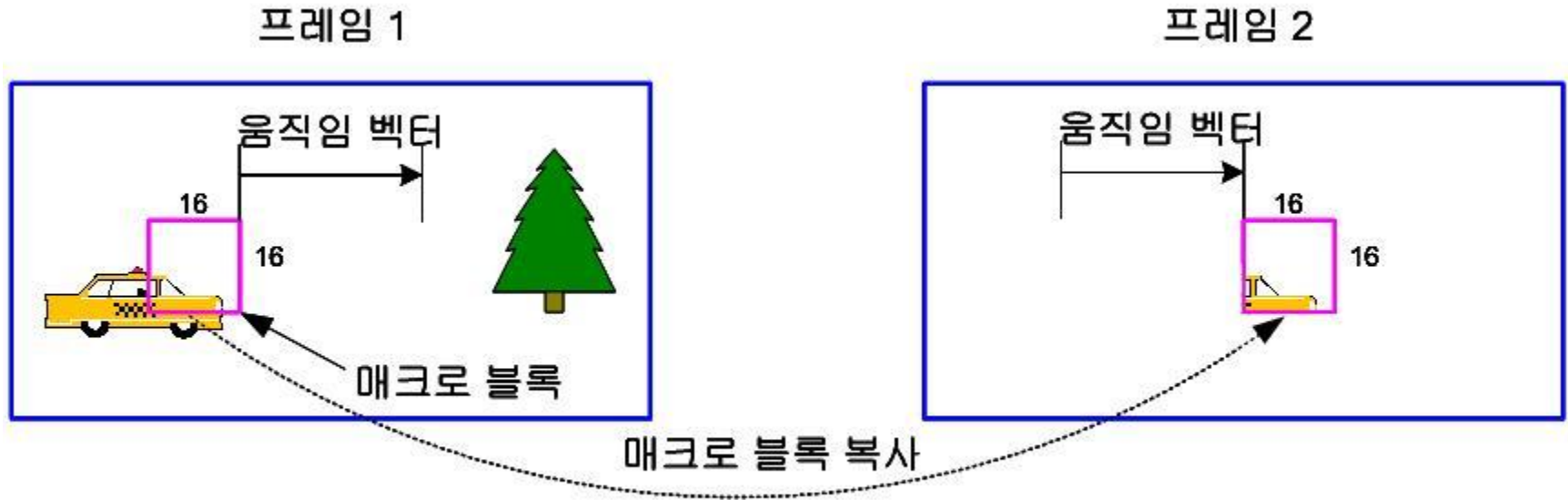
- 프레임 1의 모든 매크로 블록이 움직임 벡터에 근거하여 프레임 2로 복사되어 보상 영상을 구성

■ 차영상과 움직임 벡터 부호화

- 보상 영상과 원래의 프레임 2와의 차 영상을 구하고 이것을 부호화
- 차 영상의 값은 큰 값이 아니므로 적은 데이터양으로 표현이 가능
- 움직임 벡터는 움직임의 시작 좌표와 움직임이 끝나는 좌표를 저장하므로 데이터양이 아주 적음.

- 부호화된 차 영상과 부호화된 움직임 벡터는 하나의 프레임 데이터와 비교하여 크기가 많이 작아짐. → 압축된 데이터

움직임 보상 과정



[그림 14-21] 움직임 보상 과정

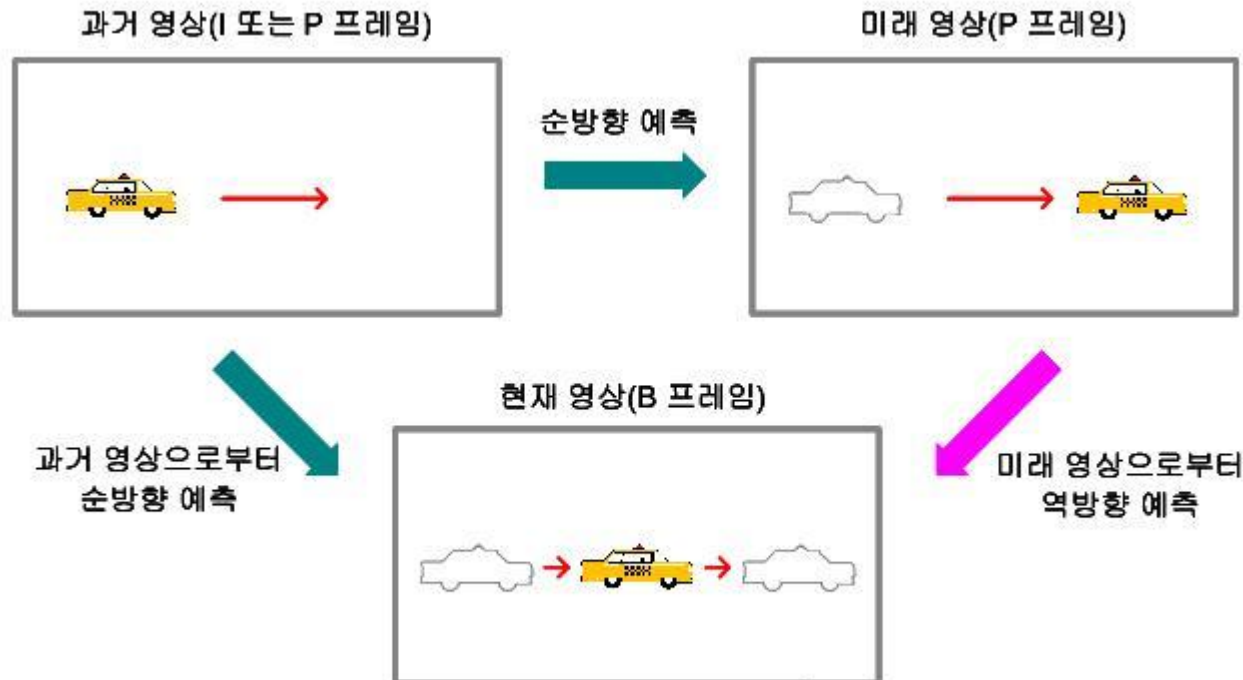
움직임 예측

움직임 예측의 정확도

- 부호화되는 차 영상의 크기를 줄일 수 있으므로 압축률을 결정하는 요소
- 복원할 때 영상의 품질을 결정하므로, 움직임 예측은 정확해야 함.

순방향 예측 : 과거 영상에서 미래 영상을 예측하는 것

양방향 예측 : 현재 영상을 기준으로 과거 영상에서의 순방향 예측과 미래 영상에서의 역방향의 예측을 수행하는 것



[그림 14-22] 순방향 예측과 양방향 예측

예측 영상의 종류와 순서

■ 예측 영상

- I 영상(Intra-Picture)

- 예측 부호화 없이 하나의 프레임에서 압축되어 독립적으로 복원

- P 영상(Predicted Pictures)

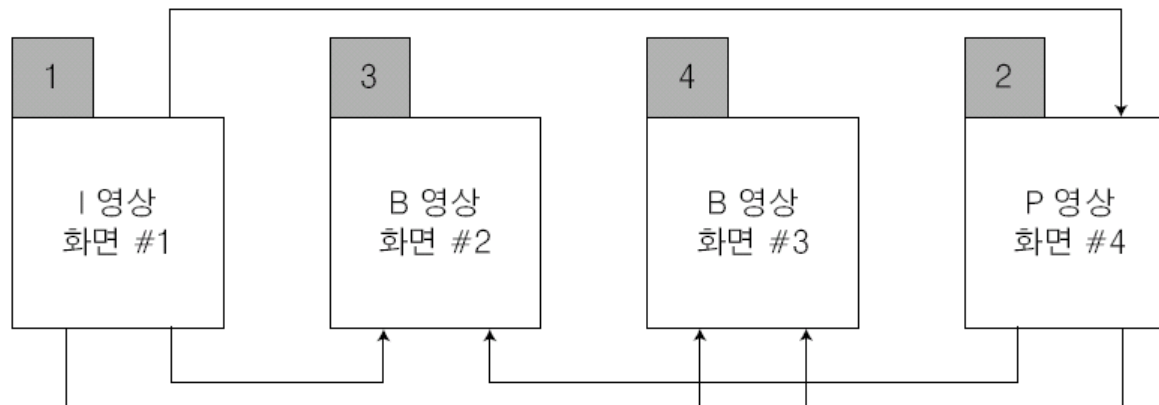
- 순방향 예측 부호화 영상

- B 영상(Bi-directional Predicted Pictures)

- 양방향 예측 부호화 영상

■ 부호화 순서와 화면 순서

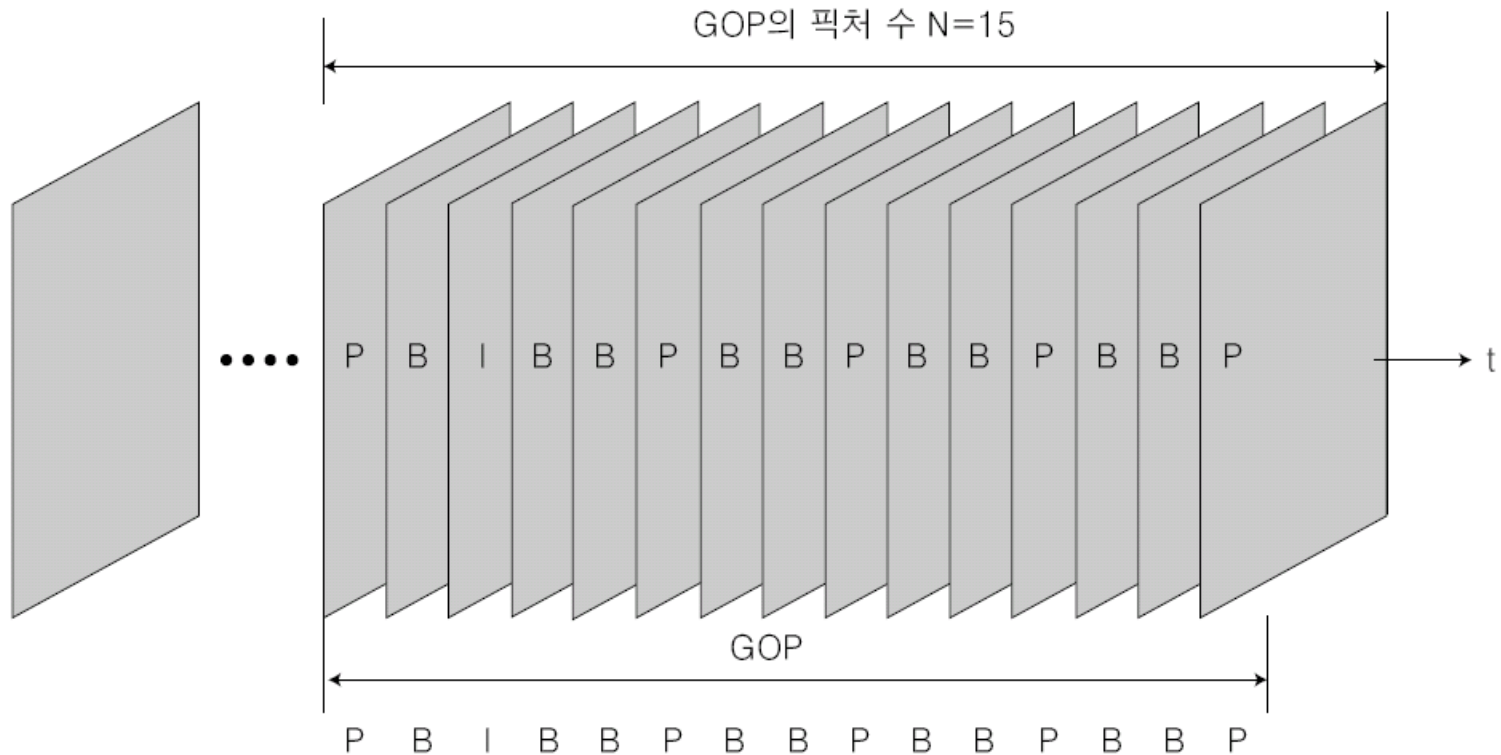
- I와 P 영상을 먼저 처리한 뒤 B 영상을 부호화하므로 원래 화면의 순서



[그림 14-23] 부호화 순서와 화면 순서

👤 GOP(Group Of Pictures) 구조

- 순방향과 양방향 예측을 수행하는 P 영상과 B 영상을 포함하며, 임의 접근(Random Access)이 가능하도록 I 영상이 한 장 이상 주기적으로 들어 있음.
- 동영상 플레이 도중 원하는 시간대의 동영상을 볼 수 있게 해주는 역할을 임의 접근으로 I 영상이 수행함.



[그림 14-24] 크기가 15인 GOP의 구조

👤 데이터 압축 원리

- 중복된 데이터를 제거하는 하는 것
- 중복성 제거 기법이라고 함

👤 손실 압축

- 중복되고 필요치 않은 정보의 손실을 허용하여 데이터양을 줄이는 방법

👤 무손실 압축

- 압축 과정에서 데이터의 손실이 전혀 없어 입력 영상과 복원된 영상과 완전히 같음.

👤 효율적인 압축 방법을 선택할 때 고려해야 할 사항

- 압축률, 압축/복원, 알고리즘의 복잡도, 계산되는 자원의 가용성과 비용, 표준화 등

👤 JPEG(Joint Photographic Expert Group)

- 컬러 정지영상의 부호화 표준을 목표로 ISO와 CCITT에서 ADCT를 기초로 하는 알고리즘을 선택하여 이 부호화 방식을 JPEG라고 명명함.
- 최종적으로 1992년 국제 표준으로 알고리즘이 확정됨.

👤 MPEG

- 동영상을 압축하는 국제 표준
- ISO/IEC에서 1988년에 동영상전문가그룹으로, 시간에 따라 연속적으로 변화하는 동영상 압축과 부호를 표현하여 정보를 전송할 수 있는 방법을 연구함.



Thank you
