

영상 압축 표준

2008. 11. 05

송 성도

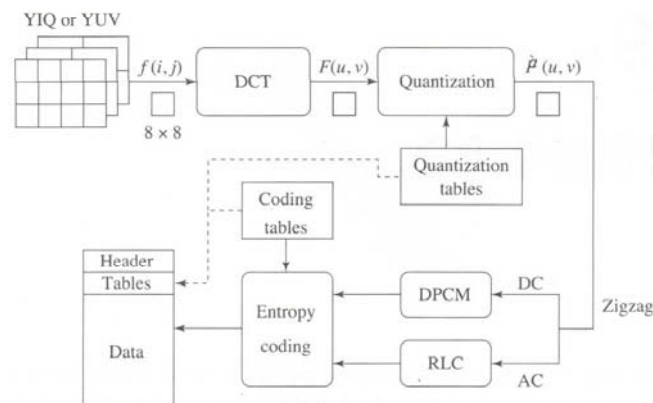
9.1 JPEG 표준

- Joint Photographics Experts Group에 의해 개발된 영상 압축 표준
- 9.1.1 JPEG 영상 압축의 주 단계
 - 디지털 영상 $f(i, j)$ 는 시간 영역(time domain)에서 정의되지 않음
 - 공간 영역(spatial domain)을 통해서 정의
 - 2차원 DCT는 두 정수 u, v 에 의해 인덱스된 공간 주파수 영역(spatial frequency domain)에서 함수 $F(u, v)$ 인 주파수 반응을 가져오기 위해 JPEG에서 한 단계로 사용
 - JPEG은 손실 영상 압축 방식(lossy image compression)

9.1.1 JPEG 영상 압축의 주 단계 (1)

- JPEG에서 DCT 변환 방식의 효율성을 나타내는 관찰 결과
 1. 유용한 영상 내용은 영상 내에서 비교적 천천히 변함
 - 작은 영역에서 명암값이 크게 여러 번 변화하는 것은 일반적이지 않음
 2. 인간은 저주파수 성분보다 매우 높은 공간 주파수 성분의 손실을 훨씬 덜 인지함
공간 중복성: 영상에서 많은 정보들이 반복된다는 것을 의미
 - 주파수가 높아지면, 정확하게 DCT 계수를 표현하는 것이 덜 중요해 짐
 - 약간의 위치 의존 정보를 제거함으로써 블록 안에 있는 화소들을 표현하는 데 더 적은 숫자들의 사용으로 끝낼 수 있음
 3. 시각적인 예민함은 컬러에서보다 명암(흑과 백)에서 훨씬 더 큼

9.1.1 JPEG 영상 압축의 주 단계 (2)

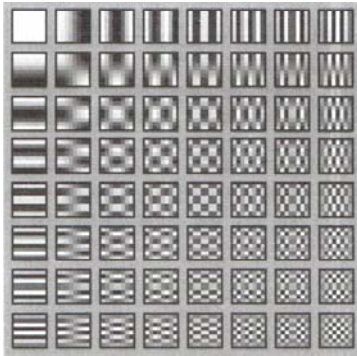


- JPEG 부호화기 블록도
 - RGB를 YIQ나 YUV로 변환하고 색을 부표본화함
 - 영상 블록별로 DCT를 수행함
 - 양자화
 - 지그재그(Zigzag)순서로 정렬, 런-길이(run-length)부호화를 수행
 - 엔트로피(entropy)부호화를 수행

9.1.1 JPEG 영상 압축의 주 단계 (3)

● 영상 블록별 DCT

- 영상은 8×8 블록들로 나뉘짐
- 2차원 DCT는 영상 블록 $f(i, j)$ 에 적용되고, 블록은 DCT 계수 $F(u, v)$ 를 출력으로 가짐
- 블록들을 사용하는 것은 그것에 인접해 있는 배경으로부터 각 블록을 분리시키는 효과
 - 높은 압축률(compression ratio) 지정 시 JPEG영상이 고르지 못함
- 특정한 $F(u, v)$ 를 계산하기 위해 적절한 u 와 v 에 대응하는 기본 영상을 아래 그림에서 선택함
- 주파수 반응 $F(u, v)$ 중의 하나를 얻기 위해 다음 식을 사용



$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i, j)$$

9.1.1 JPEG 영상 압축의 주 단계 (4)

● 양자화

- 압축된 영상을 위해 필요한 전체 비트 수를 줄이는 것을 목표로 함
- 주파수 공간 블록에서 정수에 의해 각 계수를 나누고 라운드하는 것으로 구성

$$\hat{F}(u, v) = \text{round}\left(\frac{F(u, v)}{Q(u, v)}\right) \quad (9.1)$$

- $F(u, v)$ 는 DCT 계수, $Q(u, v)$ 는 양자화 행렬(quantization matrix)
- 결과 값은 엔트로피 부호화에서 사용할 양자화된 DCT 계수를 나타냄
- 휘도(luminance)와 색차(chrominance) 영상 각각을 위한 양자화 행렬 기본값

[표 9.1] 휘도 양자화 표

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

[표 9.2] 색차 양자화 표

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

9.1.1 JPEG 영상 압축의 주 단계 (5)

- $Q(u, v)$ 에서 숫자들이 상대적으로 크기 때문에, $\hat{F}(u, v)$ 의 크기와 분산은 $F(u, v)$ 의 그것보다 상당히 작음
양자화 단계는 JPEG 압축에서 손실의 주 원천
- $Q(u, v)$ 의 값들은 오른쪽 아래로 갈수록 큰 값을 가짐
→ 높은 공간 주파수에서 더 많은 손실을 가지는 것을 목표로 함
- 앞의 행렬에서 간단한 곱셈식으로 스케일링함으로써 압축률을 변화 시킬 수 있음
- JPEG 구현에서 제공되는 화질 인자(quality factor)는 스케일링 인수로 연결
- JPEG는 사용자 양자화 표(custom quantization table)을 허용

9.1.1 JPEG 영상 압축의 주 단계 (b)



An 8 × 8 block from the Y image of 'Lena'

200 202 189 188 189 175 175 175
200 203 198 188 189 182 178 175
203 200 200 195 200 187 185 175
200 200 200 200 197 187 187 187
200 205 200 200 195 188 187 175
200 200 200 200 200 190 187 175
205 200 199 200 191 187 187 175
210 200 200 200 188 185 187 186

$f(i, j)$

515 65 -12 4 1 2 -8 5
-16 3 2 0 0 -11 -2 3
-12 6 11 -1 3 0 1 -2
-8 3 -4 2 -2 -3 -5 -2
0 -2 7 -5 4 0 -1 -4
0 -3 -1 0 4 1 -1 0
3 -2 -3 3 3 -1 -1 -3
-2 5 -2 4 -2 2 -3 0

$F(u, v)$

32 6 -1 0 0 0 0 0
-1 0 0 0 0 0 0 0
-1 0 1 0 0 0 0 0
-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

$\hat{F}(u, v)$

512 66 -10 0 0 0 0 0
-12 0 0 0 0 0 0 0
-14 0 16 0 0 0 0 0
-14 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

$\tilde{F}(u, v)$

199 196 191 186 182 178 177 176
201 199 196 192 188 183 180 178
203 203 202 200 195 189 183 180
202 203 204 203 198 191 183 179
200 201 202 201 196 189 182 177
200 200 199 197 192 186 181 177
204 202 199 195 190 186 183 181
207 204 200 194 190 187 185 184

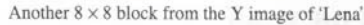
$\tilde{f}(i, j)$

1 6 -2 2 7 -3 -2 -1
-1 4 2 -4 1 -1 -2 -3
0 -3 -2 -5 5 -2 2 -5
-2 -3 -4 -3 -1 -4 4 8
0 4 -2 -1 -1 -1 5 -2
0 0 1 3 8 4 6 -2
1 -2 0 5 1 1 4 -6
3 -4 0 6 -2 -2 2 2

$\epsilon(i, j) = f(i, j) - \tilde{f}(i, j)$

- $f(u, v)$ 는 8×8 블록들 중의 하나, $F(u, v)$ 는 DCT 계수,
 $\tilde{F}(u, v)$ 는 $Q(u, v)$ 에 의해 곱셈으로 결정된 역양자화된 DCT 계수,
 $\hat{F}(u, v)$ 는 양자화된 DCT 계수, $\tilde{f}(u, v)$ 는 재구성된 영상 블록
 $\epsilon(i, j) = f(i, j) - \tilde{f}(i, j)$ 는 손실을 설명하기 위한 오차

9.1.1 JPEG 영상 압축의 주 단계 (7)


$$f(i, j) \qquad F(u, v)$$
$$\hat{F}(u, v)$$
$$\tilde{F}(u, v)$$
$$\tilde{f}(i, j)$$
$$\epsilon(i, j) = f(i, j) - \tilde{f}(i, j)$$

- 급격히 변하는 회도를 가지는 그림
- 오차가 이전 그림에 비해서 큼
- 영상이 빨리 변하는 세부 사항을 가지면 더 많은 손실이 생김

9.1.1 JPEG 영상 압축의 주 단계 (8)

- 엔트로피 부호화를 위한 준비
 - 블록도에서 DCT와 양자화를 제외한 단계들은 양자화된 DCT 계수에 대한 엔트로피 부호화로 이르게 함

32	6	-1	0
-1	0	0	0
-1	0	1	0
-1	0	0	0
 - 이런 데이터 압축 단계들은 무손실

0	0	0	0
0	0	0	0
 - DC들은 DPCM, AC들은 런 길이 부호화가 됨

0	0	0	0
0	0	0	0

$$\hat{F}(u, v)$$

- AC 계층에서 런 길이 부호화(Run-Length Coding, RLC)

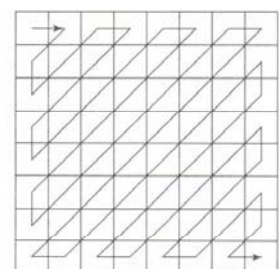
- 런 길이 부호와는 $\hat{F}(u, v)$ 값을 집합으로 변화시킴에 있어 유용
- 지그재그 주사는 8×8 행렬을 64 벡터로 변화시킴
- 이전의 $\hat{F}(u, v)$ 는 다음과 같이 변경됨

$$(32, 6, -1, -1, 0, -1, 0, 0, 0, -1, 0, 0, 1, 0, 0, \dots, 0)$$

- RLC단계는 \hat{F} 의 AC계수에서 각 이들의 런에 대한 한 쌍 (RUNLENGTH, VALUE)에 값으로 대체

- RUNLENGTH는 0의 개수, VALUE는 다른 0이 아닌 개수

- 첫 번째 (DC) 성분을 고려 하지 않으면 값

$$(0, 6)(0, -1)(0, -1)(1, -1)(3, -1)(2, 1)(0, 0)$$


9.1.1 JPEG 영상 압축의 주 단계 (9)

- DC 계수에 대한 DPCM
 - DC 계수는 AC 계수들로부터 분리하여 부호화됨
 - 영상 블록은 오직 하나의 DC 계수를 가짐
 - DC 계수들의 값은 크고 다를 수 있음
 - DC 값은 블록의 평균 명암도를 반영
 - DC 계수가 150, 155, 149, 152, 144이고, i 번째 블록의 예측값이 $d_i = DC_{i+1} - DC_i$ 이고, $d_0 = DC_0$ 라면 DPCM은 150, 5, -6, 3, -8을 생성
- 엔트로피 부호화
 - DC와 AC 계수는 마지막으로 엔트로피 부호화 단계를 거침
 - 허프만 부호화를 사용하고, 기본 엔트로피 부호화 방법만 논의함

9.1.1 JPEG 영상 압축의 주 단계 (10)

- DC 계수들의 허프만 부호화
 - DPCM-부호화된 각 DC 계수는 한 쌍의 심볼(SIZE, AMPLITUDE)로 표현
 - SIZE — 계수 표현을 위한 비트의 수, AMPLITUDE — 실제 비트들을 포함
 - SIZE는 허프만 부호화되고, 가변 길이 부호 -> 추가적인 압축
 - AMPLITUDE는 허프만 부호화되지 않음
 - 오른쪽 표는 다른 가능한 크기들 위한 크기 범주를 설명함
 - 음수를 위해 1의 보수가 사용됨
 - DPCM 150, 5, -6, 3, -8은 (8, 1001110) (3, 101) (3, 001) (2, 11), 4, 111) 로 변경됨

SIZE	AMPLITUDE
1	-1, 1
2	3, -2, 2, 3
3	-7 .. -4, 4 .. 7
4	-15 .. -8, 8 .. 15
⋮	⋮
⋮	⋮
⋮	⋮
10	-1023 .. -512, 512 .. 1023

9.1.1 JPEG 영상 압축의 주 단계 (11)

- AC 계수들의 허프만 부호화
 - 실제 JPEG 구현에서, VALUE는 SIZE와 AMPLITUDE에 의해서 표현됨
 - 비트 절약을 위해 RUNLENGTH와 SIZE는 4비트만 할당되어 한 바이트에 넣어짐

Symbol 1: (RUNLENGT, SIZE)

Symbol 2: (AMPLITUDE)

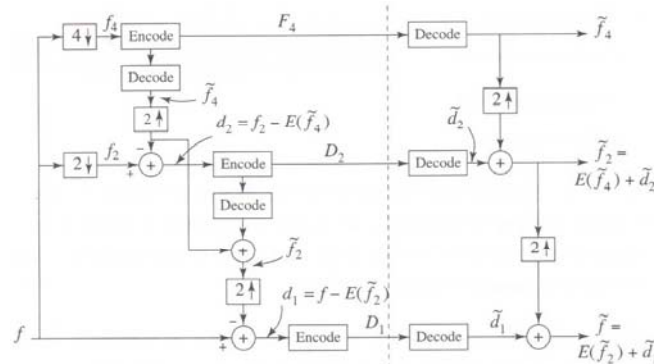
- 4 비트 RUNLENGTH는 0에서 15까지의 0의 런들만 표현 가능
- 0의 런길이가 15보다 커지면 확장 코드 (15, 0)가 Symbol 1을 위해 사용
- DC에서와 같이, Symbol 1은 허프만 부호화되지만, Symbol 2는 그렇지 않음

9.1.2 JPEG 모드 (1)

- 순차 모드(Sequential Mode)
 - JPEG의 기본 모드
 - 회색도 영상과 컬러 영상 성분은 한 번의 왼쪽에서 오른쪽, 위에서 아래로의 주사로 부호화 됨
- 점진적 모드(Progressive Mode)
 - 영상의 낮은 화질의 버전을 빨리 전달하고 뒤이어 높은 화질을 전달함
 - 웹 브라우저에서 폭넓게 지원
 - 통신 선로의 속도가 낮을 때 유용함
 - 구현
 - 스펙트럼 선택
 - DCT 계수의 스펙트럼(공간 주파수 스펙트럼) 특성을 이용
 - 더 높은 AC 성분들은 세부적인 정보만을 제공
 - 연속적인 근사
 - 모든 DCT 계수들이 동시에 부호화
 - 가장 중요한 비트들(MSBs)가 먼저 부호화 됨

9.1.2 JPEG 모드 (2)

- 계층적 모드(Hierarchical Mode)
 - 여러 개의 다른 해상도 계층에서 영상을 부호화
 - 점진적 모드와 비슷하게 점진적으로 화질이 개선되는 다중 전송 형식을 전달 가능
 - 그림은 3-레벨의 계층 JPEG 부호화 기와 복호화 기를 보여줌

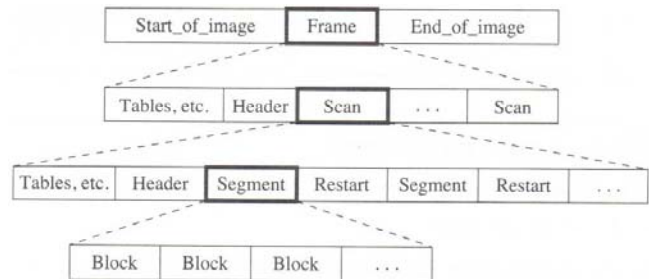


9.1.2 JPEG 모드 (3)

- 무손실 모드(Lossless Mode)
 - 영상 화질에서 전혀 손실이 없는 JPEG의 아주 특별한 경우
 - 변환 부호화를 사용하지 않고 차분 부호화 방식을 사용함
 - 압축률이 낮기 때문에 드물게 사용됨

9.1.3 JPEG 비트스트림에 관한 일견

- 그림은 JPEG 영상들을 위한 비트스트림 구조의 계층적 보기를 제공
- 프레임(frame) - 영상, 주사(scan) - 화소
세그먼트(segment) - 블록의 그룹



- 헤더 정보의 예
 - 프레임 헤더(Frame header)
 - 화소당 비트 수
 - 영상의 (넓이, 높이)
 - 성분의 수
 - (각 성분을 위한) 고유 ID
 - (각 성분을 위한) 수직/수평 샘플링(표본화) 인수
 - (각 성분을 위한) 사용할 양자와 표
 - 주사 헤더(Scan header)
 - 주사하는 성분의 수
 - (각 성분을 위한) 성분 ID
 - (각 성분을 위한) 허프만/산술 부호와 표

9.2 JPEG2000 표준 (1)

- 새로운 JPEG2000 표준은 더 나은 율-왜곡(rate distortion)의 균형과 향상된 주관적 영상 화질 추가적인 기능 제공을 목표로 함
- JPEG2000 표준이 해결하고자 하는 문제
 - 낮은-비트율 압축
 - JPEG은 중간 혹은 높은 비트율에서 높은 율-왜곡 성능을 제공
 - 무손실 및 손실 압축
 - 현재까지 단일 비트스트림에서보다 뛰어난 무손실 압축과 손실 압축을 제공할 수 없음
 - 큰 영상
 - 새로운 표준은 타일링을 하지 않고 64k×64k보다 큰 영상 해상도를 지원할 수 있음
 - 단일 신장 구조
 - 현재 표준은 특정한 응용에서만 사용되고 대부분의 JPEG 복호화기에서 사용되지 않는 4나까지의 모드를 가지고 있음

9.2 JPEG2000 표준 (2)

- 잡음이 있는 환경에서의 전송
 - 무선 네트워크와 인터넷 같은 잡음 있는 환경에서 향상된 에러 복원력을 제공
- 점진적 전송
 - 낮은 비트율에서 높은 비트율까지 끊김이 없는 화질과 해상도의 스케일러빌리티를 제공
- 관심 영역 보호화
 - 관심 영역(Regions of Interest)을 지정해 더 나은 화질로 보호화 가능
- 컴퓨터를 통해 생성된 영상
 - 현재 표준은 자연 영상을 위해 최적화, 컴퓨터를 통해 생성된 영상에 잘 수행되지 않음
- 합성 문서
 - 추가적인 영상이 아닌 데이터를 통합시키기 위해 메타데이터 매커니즘을 제공

9.2 JPEG2000 표준 (3)

- JPEG2000은 256개의 정보 채널을 다룰 수 있음
 - 위성 영상 촬영에서 생성
- JPEG2000은 큰 압축된 영상의 원격 브라우징을 수행할 수 있는 능력 제공
- JPEG2000의 수행 방식
 - DCT 기반 보호화 방식
 - 현재 표준과의 역방향 호환성을 위해 제고
 - 기존선 JPEG(baseline JPEG)를 구현함
 - 웨이블릿 기반 방식
 - 모든 새로운 기능과 향상된 성능이 속함

9.2.1 JPEG2000 영상 압축의 주요 단계 (1)

- 주된 압축 방식은 Taubman에 의해 고안된 알고리즘(EBCOT)
 - 최적화된 절단을 갖는 내장형 블록 부호화
 - Embedded Block Coding with Optimized Truncation
- EBCOT의 기본 개념
 - 웨이블릿에 의해 생성된 서브밴드 LL, LH, HL, HH를 코드 블록이라 불리는 작은 블록들로 분할
 - 코드 블록은 다른 블록의 정보도 사용하지 않고 독립적으로 부호화
 - 독립적이고 스케일러블한 비트스트림이 코드 블록을 위해 생성
 - 블록 기반 부호화 구조를 갖기 때문에 EBCOT 알고리즘은 향상된 오류 복원력을 가짐

9.2.1 JPEG2000 영상 압축의 주요 단계 (2)

- EBCOT 알고리즘 3단계
 - 블록 부호화와 비트스트림 생성
 - 각 서브밴드는 32×32 혹은 64×64 크기의 작은 코드 블록으로 나뉘어짐
 - 코드 블록 B_i 를 위한 스케일러블한 비트스트림 생성
 - 코드 블록 B_i 를 위해 $si(k)=si(k_1, k_2)$ 가 k_1, k_2 를 갖는 서브밴드 샘플의 작은 코드 블록의 2차원 순서가 되게 함

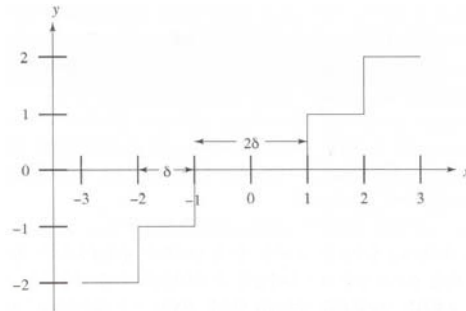


9.2.1 JPEG2000 영상 압축의 주요 단계 (3)

- 알고리즘은 아래 그림의 데드 존(dead zone) 양자화기를 사용함
- 이들 기준으로 양쪽으로 두 배 길이 영역
 $x_i(k) \in \{-1, 1\}$ 이 $s_i(k)$ 의 부호이고 $v_i(k)$ 가 양자화된 크기

$$v_i[k] = \frac{\|s_i[k]\|}{\delta_{\beta_i}} \quad (9.2)$$

- δ_{β_i} 는 코드 블록 B_i 를 포함하는 서브밴드 β_i 의 양자화 단계크기
- $v_i^p[k]$ 를 $v_i(k)$ 의 이진 표현에서 p 번째라고 할 때, $p = 0$ 은 가장 덜 중요한 비트와 부합
- 코드 블록에서 적어도 하나의 샘플을 위해 $v_i^{p_{\max}}[k] \neq 0$ 이도록 p_{\max} p 의 최대값



9.2.1 JPEG2000 영상 압축의 주요 단계 (4)

- 부호화 과정은 가장 중요비트부터 다음 중요비트 순서로 모든 비트평면이 부호화될 때 까지 부호화하는 비트평면 부호화기의 그것과 비슷 ->
 하나이상의 가장 덜 중요한 비트들을 잃음 ->
 질이 낮은 데드 존 양자화기를 사용하는 것과 같음
- 특정 샘플이나 그것의 이웃 샘플들에 관한 이전 $v_i^p[k]=1$ 에 부호화된 정보를 잘 이용하는 것이 중요
- 초기값이 0이지만 첫 번째 0이 아닌 비트평면 이 부호화될 때 1로 변하는 이진값 상태 변수를 정의함으로써 EBCOT에서 행해짐
- 이 이진 상태 변수는 샘플의 중요성이라 불림
- 중요한 샘플들은 클러스터를 이루는 경향이 있음 ->
 단일 이진 심볼을 부호화함으로써 많은 샘플들을 처리하는 것이 가능
- $1b \times 1b$ 크기의 서브블록들까지만 클러스터링 가정을 이용

9.2.1 JPEG2000 영상 압축의 주요 단계 (5)

- $\sigma^p(B_i[j])$ 가 비트평면 p 에서 서브블록 $B_i[j]$ 의 중요성이라면,
중요성 맵은 쿼드 트리를 사용하여 부호화됨
- 코드 블록의 중요성은 $\dagger = \top$ 일 때 루트에서 시작,
 $\dagger = 0$ 일 때 단말을 향해 수행하여 한번에 한 쿼드 레벨을 식별
- 중요성 값은 엔트로피 부호화를 위해 산술 코더로 보내짐
- 중복된 중요성 값은 건너뛰
- 아래의 조건 중 어떤 것이라도 충족되면 중복으로 취급
 - 부모는 중요하지 않음
 - 현재 쿼드는 이전의 비트평면에서 중요했음
 - 이것은 같은 중요도의 부모를 공유하는 것들 가운데 방문된 마지막 쿼드이고,
다른 형제들은 중요하지 않음

9.2.1 JPEG2000 영상 압축의 주요 단계 (b)

● EBCOT의 네 가지 부호화

- 영 부호화
 - 양자화된 샘플이 $v_i[k] < 2^{p+1}$ 을 만족하도록 $v_i^p[k]$ 를 부호화하기 위해 사용
 - 현재 샘플의 중요성은 8개의 이웃의 값에 의존
 - 이웃의 중요성 분류 세 가지
 - 수평 - $h_i[k] = \sum_{z \in \{1, -1\}} \sigma_i[k_1 + z, k_2]$, with $0 \leq h_i[k] \leq 2$
 - 수직 - $v_i[k] = \sum_{z \in \{1, -1\}} \sigma_i[k_1, k_2 + z]$, with $0 \leq h_i[k] \leq 2$
 - 대각선 - $v_i[k] = \sum_{z1, z2 \in \{1, -1\}} \sigma_i[k_1, k_2 + z]$, with $0 \leq h_i[k] \leq 2$
 - 256개의 가능한 이웃 구성은 표에서 나열된 9개의 분명이 다른 내용 할당으로 줄어듦

Label	LL, LH and HL subbands			HH subband	
	$h_i[k]$	$v_i[k]$	$d_i[k]$	$d_i[k]$	$h_i[k] + v_i[k]$
0	0	0	0	0	0
1	0	0	1	0	1
2	0	0	>1	0	>1
3	0	1	x	1	0
4	0	2	x	1	1
5	1	0	0	1	>1
6	1	0	>0	2	0
7	1	>0	x	2	>0
8	2	x	x	>2	x

9.2.1 JPEG2000 영상 압축의 주요 단계 (7)

- 런 길이 부호화
 - 1비트 중요성 값의 런을 만들어내는 것을 목표로 함
 - 런 길이 부호화를 위해 충족되어야 하는 조건
 - 네 개의 연속적인 샘플이 중요하지 않아야 함
 - 샘플들은 중요하지 않은 이웃들을 가져야 함
 - 샘플들은 같은 서브블록 안에 있어야 함
 - 첫 번째 샘플의 수평 인덱스 k_1 은 짝수여야 함
 - 네 개의 심볼들이 조건을 만족할 때, 중요도 식별을 위해 하나의 특별한 비트가 부호화됨
 - 네 개중 어떤 것이 중요해지면, 그런 샘플의 인덱스는 2 비트 양으로 보내짐

9.2.1 JPEG2000 영상 압축의 주요 단계 (8)

- 코드 부호화
 - 기본 코드 부호화는 네 개의 수평, 수직 이웃을 가지고, 중요하지 않거나 양수 혹은 음수이기 때문에 $3^4=81$ 개의 서로 다른 내용 구성이 있음
 - 수평, 수직 대칭 모두를 이용하고 주어진 어떤 이웃 배열에 대해, $\chi_i[k]$ 의 조건분배가 $-\chi_i[k]$ 의 그것과 같다고 가정하면 내용의 수는 5
 - 수평 이웃들 모두가 중요하지 않으면 $\bar{h}_i[k]$ 가 0, 적어도 하나의 수평 이웃이 양수이면 1, 적어도 하나의 수평 이웃이 음수이면 -1
 - $\chi_i[k]$ 가 코드 예측
 - 부호화된 이전 심볼은 $\chi_i[k] \cdot \hat{\chi}_i[k]$

Label	$\hat{\chi}_i[k]$	$\bar{h}_i[k]$	$\bar{r}_i[k]$
4	1	1	1
3	1	0	1
2	1	-1	1
1	-1	1	0
0	1	0	0
1	1	-1	0
2	-1	1	-1
3	-1	0	-1
4	-1	-1	-1

기본 코드 부호화를 위한 내용 할당

9.2.1 JPEG2000 영상 압축의 주요 단계 (9)

- 크기 개선
 - $v_i^p[k]$ 의 값을 부호화하기 위해 사용 ($v_i[k] \geq 2^{p+1}$)
 - 세 개의 내용 모델만이 크기 개선을 위해 사용
 - 처음 $s_i[k]$ 에 크기 개선이 적용된 후에
 - 에서부터 1로 변하는 두 번째 상태 변수 $\tilde{s}_i[k]$ 가 소개
 - $\tilde{s}_i[k] = h_i[k] = v_i[k] = 0$ 이면 $v_i^p[k]$ 는 내용 0으로
 $\tilde{s}_i[k] = 0$ 이고 $h_i[k] + v_i[k] \neq 0$ 이면 내용 1로 $\tilde{s}_i[k] = 1$ 이면 내용 2로 부호화
- 비트스트림 보증을 위한 네 개의 패스
 - 순방향 중요성 전달 패스(P_1^p)
 - 주사선 순서 방문, 중요하지 않은 샘플들은 뛰어 넘음
 - 중요한 샘플은 주사 방향으로 진행되는 중요성 결정 단계에 도움이 되기 때문에 순방향 중요성 전달 패스(forward-significance-propagation pass)라 불림

9.2.1 JPEG2000 영상 압축의 주요 단계 (10)

- 역방향 중요성 전달 패스(P_2^p)
 - 역순서로 진행되는 것을 제외하면 순방향과 동일
- 크기 개선 패스(P_3^p)
 - 이전 두 패스에서 부호화되지 않은 샘플들을 부호화
 - 기본 크기 개선으로 처리
- 표준 패스(P_4^p)
 - 이전의 세 가지 부호화 패스에서 고려되지 않은 샘플을
 기본 코드 부호화와 런 길이 부호화를 사용하여 부호화

S_i^{max}	P_i^{max}	P_i^{max-1}	P_i^{max-1}	P_i^{max-1}	S_i^{max-1}	P_i^{max-1}	...	P_1^0	P_2^0	P_3^0	S^0	P_4^0
-------------	-------------	---------------	---------------	---------------	---------------	---------------	-----	---------	---------	---------	-------	---------

각 블록의 내장된 비트스트림에서 부호화 패스와 쿼트트리 코드의 모양

- S는 초기 부호화 패스가 아니라 마지막 부호화 패스전에 나타남 ->
 처음으로 중요하게 된 서브블록이 마지막 패스까지 무시되어진다는 의미

9.2.1 JPEG2000 영상 압축의 주요 단계 (11)

● 압축 후 율-왜곡(Post Compression Rate Distortion) 최적화

- PCRD의 목표: 비트율 제약에 따라 왜곡을 최소화시키도록 블록의 독립적인 비트스트림의 최적 절단(optimal truncation)을 찾는 것
- 비율 $R_i^{n_i}$ 을 가지는 코드 블록 B_i 의 절단된 내장 비트스트림에서 전체 왜곡

$$D = \sum_i D_i^{n_i} \quad (9.3)$$

- $D_i^{n_i}$ 는 절단 포인트 n_i 을 가지는 코드 블록 B_i 로부터의 왜곡

$$D_i^n = \omega_{bi}^2 \sum_{k \in b_i} (\tilde{s}_i^n[k] - s_i[k])^2 \quad (9.4)$$

- $s_i[k]$ 는 코드 블록 B_i 에서의 서브밴드 샘플의 2차원 순서
 $\tilde{s}_i^n[k]$ 은 절단 포인트 n 과 관련된 샘플들의 양자화 표현
 ω_{bi}^2 값은 코드 블록 B_i 를 포함하는 서브밴드 b_i 를 위한 웨이블릿 기저 함수 L_2 놈(norm)

9.2.1 JPEG2000 영상 압축의 주요 단계 (12)

- 절단 포인트 n_i 의 최적 선택은 다음의 제약을 받는 최소화 문제로 공식화

$$R = \sum_i R_i^{n_i} \leq R^{\max} \quad (9.5)$$

- R^{\max} 는 이용 가능한 비트율

$$(D(\lambda) + \lambda R(\lambda)) = \sum_i (D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda}) \quad (9.6)$$

- 어떤 절단 포인트의 집합 $\{n_i^\lambda\}$ 은 율-왜곡 의미에서 최적
- 전체 율 $R(\lambda) = R^{\max}$ 를 갖는 절단 포인트의 집합을 찾는 것이 최적화 문제를 해결
- 정확한 값을 찾는 것은 불가능 $\rightarrow R(\lambda) \leq R^{\max}$ 를 만족하는 λ 의 가장 작은 값
- N_i 를 가능성 있는 절단 포인트의 집합, $j_1 < j_2 < \dots$ 은 다음 식의 비에 의해 주어진 대응하는 왜곡-율 기울기를 갖는 절단 포인트들의 나열

$$S_i^{jk} = \frac{\Delta D_i^{jk}}{\Delta R_i^{jk}} \quad (9.7)$$

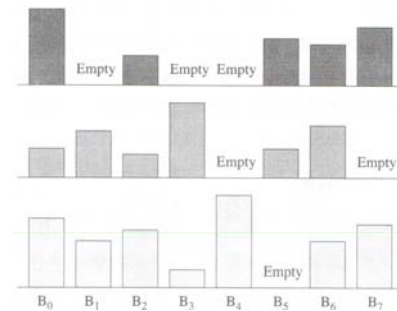
- 고정된 λ 값을 위한 최소한 문제는 사소한 선택

$$n_i^\lambda = \max \{jk \in N_i \mid S_i^{jk} > \lambda\} \quad (9.8)$$

9.2.1 JPEG2000 영상 압축의 주요 단계 (13)

계층 형식과 표현

- ECBCOT 알고리즘은 해상도와 화질 스케일러빌리티 모두 제공
- 이 기능은 계층화된 비트스트림 구조와 2-층을 이루는 부호화 전략으로 이루어짐
- EBCOT의 최종 비트스트림은 화질 계층들의 모임
 - 화질 계층 Q_1 은 코드 블록 B_i 의 처음 $R_i^{n_1}$ 바이트를 포함
 - 다른 계층 Q_q 는 B_i 로부터 증가분 $L_q^i = R_i^{n_q} - R_i^{n_{q-1}} \geq 0$ 을 포함
 - 양 n_q^i 는 q 번째 화질 계층을 위해 선택된
 일-왜곡 임계치 λ_q 에 상응하는 절단 포인트



9.2.1 JPEG2000 영상 압축의 주요 단계 (14)

- 증가분과 함께, 길이 L_q^i , 새로운 부호화 패스의 수 $N_q^i = n_q^i - n_{q-1}^i$
 B_i 가 화질 계층 Q_q 에 대한 p_i^{max} 와 같은 보조 정보가 명확히 저장되어야 함
- 보조 정보는 두 번째-층 부호화 엔진에서 압축
- 2-층 구조에서 첫 번째-층은 내장된 블록 비트스트림 생산,
 두 번째는 화질 계층에 대한 블록 기여를 부호화
- 양 q_i (p_i^{max} 와 B_i 가 처음으로 비어 있지 않은 기여를 만드는 화질 계층의 인덱스)는
 독립적인 내장 쿼드트리 코드를 사용하여 부호화
- $B_i^0 = B_i$ 는 리프, B_i^t 를 전체 서브밴드를 나타내는 트리의 루트
- $q_i^t = \min\{q_j \mid B_j \subset B_i^t\}$ 를 쿼드 B_i^t 에서 B_i 이 비어 있지 않은 기여를 만드는
 첫 번째 계층의 인덱스
- 단일 비트는 각 레벨 t 에서 쿼드에 대해 $q_i^t > q$ 를 식별, 중복된 쿼드는 생략
- 부모 쿼드 B_j^{t+1} 에 대해 $q_i^t > q-1$ 이거나 $q_i^{t+1} > q$ 이면 쿼드는 중복

9.2.1 JPEG2000 영상 압축의 주요 단계 (15)

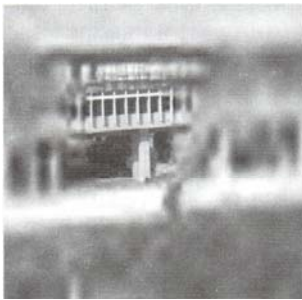
- p_i^{max} 는 화질 계층 Q_q 의 부호화까지는 무관
- p_i^{max} 에 관한 불필요한 정보도 Q_q 를 부호화할 준비가 된 때까지 보낼 필요가 없음
- EBCOT는 루트로부터가 아닌 리프로부터 얻은 내장형 쿼드트리를 사용하여 이것을 수행
- B_i^t 는 서브밴드로부터의 B_i 의 맨 위에 만들어진 쿼드트리 구조의 구성 요소,
 $p_i^{max,t} = \max\{p_j^{max} \mid B_j \subset B_i^t\}$, B_i^t 를 B_i 가 자손인 쿼드의 조상
 p 를 B_i 에 대해 p_i^{max} 보다 더 크다는 것을 보증하는 값
- $p_i^{max} = p_{i_0}^{max,0}$ 의 값 알고리즘
 - $p = p-1, p-2, \dots, 0$ 에 대해
 $p_{i_t}^{max,t} < p$ 인지 아닌지 식별을 위해 이전 숫자를 보냄, 중복된 비트는 뛰어 넘음
 만약 $p_{i_t}^{max} = p$ 이면 멈춤
- 중복된 비트는 $p_{i_{t+1}}^{max,t+1} < p$ 와 같은 조상 이나 다른 B_i 를 위한 p_j^{max} 를 식별하기 위해
 사용되는 부분 쿼드트리 둘 중 하나로부터
 추론될 수 있는 조건 $p_{i_t}^{max} < p$ 에 해당하는 것

9.2.2 JPEG2000으로의 EBCOT 적용

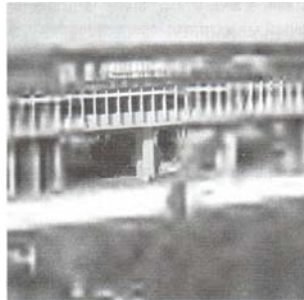
- EBCOT 알고리즘의 수정
 1. 곱셈과 나눗셈이 없는 낮은 복잡도의 산술 부호화기가 대체 사용
 2. HL 서브밴드의 코드 블록들을 전치에서
 영 부호화 내용 할당 맵에서 대응되는 엔트리들이 전치
 3. 순방향, 역방향 중요성 전달 패스에서 단일 순방향 중요성 전달 패스
 4. 블록 크기를 16×16 에서 $4 \times 4 \rightarrow$ 서브블록 중요성을 명확하게 부호화할 필요성 제거
- 기존 알고리즘에 비해서 낮은 손실과 향상된 실행 속도

9.2.3 관심 영역 부호화

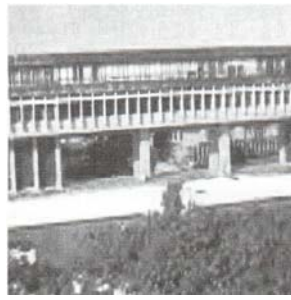
- 특별한 영역을 나머지 부분이나 배경보다 나은 화질로 부호화
- MAXSHIFT라 불리며, 계수의 값을 키우는 스케일링 기반 방법
- 내장형 부호화를 수행하는 동안 결과 비트는 영상의 ROI가 아닌 부분 앞에 놓임
- ROI는 영상의 나머지 부분 전에 복호화되고 개선됨



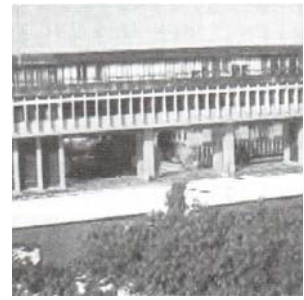
0.4 bpp



0.5 bpp



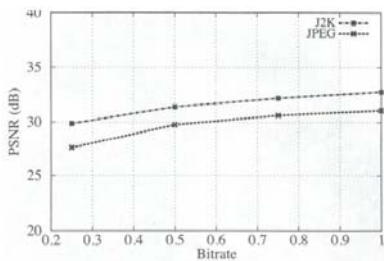
0.6 bpp



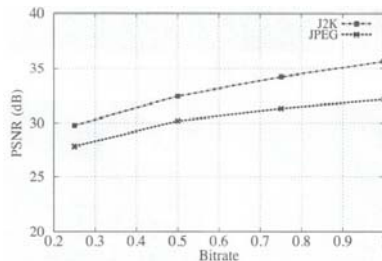
0.7 bpp

9.2.4 JPEG와 JPEG2000 성능 비교 (1)

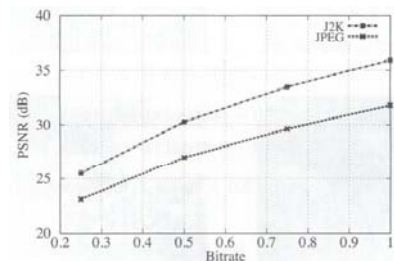
- 압축 효율의 비교



자연 영상



컴퓨터에 의해 생성된 영상



의학용 영상

9.2.4 JPEG와 JPEG2000 성능 비교 (2)

● 복원 결과 비교



원영상



JPEG 0.75bpp



JPEG2000 0.75bpp



JPEG 0.25bpp



JPEG2000 0.25bpp

9.3 JPEG-LS 표준 (1)

- 낮은 복잡도 알고리즘을 기반으로 함
- 연속적인 명암 영상의 무손실 혹은 “무손실에 가까운” 압축을 위한 ISO/ITU 표준
- Hewlett-Packard에 의해 제안된 알고리즘 사용
 - 낮은 복잡도 무손실 압축(LOCO-I)

- LOCO-I는 내용 모델링(context modeling) 개념을 이용

- 입력 소스에서의 구조를 이용

- 영상에서 어떤 화소 값이 각각의 화소를 뒤따를 것인지를 예측 확률을 이용

ex) $P(0) = 0.4, P(1) = 0.6 \rightarrow$ 0차 엔트로피 $H(S) =$

이전의 심볼이 0이면, 현 심볼이 0이 될 확률은 0.8

이전의 심볼이 1이면, 현 심볼이 0이 될 확률이 0.1이라 가정

입력 신호는 0과 1에 해당하며, 0의 확률 0.4와 1의 확률 0.6으로 나타낼 수 있음

$$H(S_2) = -0.1 \log_2(0.1) - 0.9 \log_2(0.9) = 0.47$$

$$\text{평균 비트레이트} = 0.4 * 0.72 + 0.6 * 0.47 = 0.57$$

9.3 JPEG-L5 표준 (2)

- LOCO-I는 오른쪽 그림의 내용 모델을 사용
 - 래스터 주사 순서에서
 - 내용 화소 a, b, c 및 d 는 현 화소 x 전에 나타남
 - 인과적인 내용(causal context)이라 불림
- LOCO-I의 성분
 - 예측
 - 인과적인 템플릿을 사용하여 다음 샘플 x' 의 값을 예측
 - 내용 결정
 - x' 가 일어날 때의 내용 결정
 - 나머지 부호화
 - x' 의 내용에 의해 결정된 예측 나머지에 대한 엔트로피 부호화

c	a	d
b	x	

9.3.1 예측

- JPEG-L5는 낮은 복잡도를 목표로 하기 때문에,
LOCO-I 알고리즘은 수직, 수평 에지 검출을 위해 고정 예측기를 사용

$$\hat{x}' = \begin{cases} \min(a, b) & c \geq \max(a, b) \\ \max(a, b) & c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases} \quad (9.9)$$

- 현 위치의 왼쪽에 수직 에지가 있을 때 a 를 출력
- 현 위치의 위쪽에 수평 에지가 있으면 b 를 출력
- 이웃 샘플들이 상대적으로 평탄할 때 $a+b-c$ 를 출력

9.3.2 내용 결정

- 현재의 예측 오류를 조절하는 내용 모델은 세 가지 성분 내용 벡터 $Q = (q_1, q_2, q_3)$ 을 이용하여 인덱스

$$q_1 = d - b$$

$$q_2 = b - c$$

$$q_3 = c - a \quad (9.10)$$

- 현 샘플을 둘러싸고 있는 극부적인 평탄함 혹은 예지 내용을 획득한 극부적인 변화도를 나타냄
- 잠재적으로 넓은 범위의 값을 가질 수 있기 때문에, 내용 모델 접근 방법은 비실용적
- 문제 해결을 위해 변수 축소 방법의 필요
 - 제한된 수의 값에 의해 표현하기 위해 이 차들을 양자화
 - Q의 성분들은 결정 경계 $-T, \dots, -1, 0, 1, \dots, T$ 를 갖는 양자화기를 사용 양자화됨
 - 내용 크기는 첫 번째 구성 요소가 음수가 되는 내용 벡터 Q를 $-Q$ 로 대신함으로써 더욱 축소
 - 서로 다른 내용 상태의 수는 $\frac{(2T+1)^3 + 1}{2} = 365$
 - 벡터 Q는 $(0, 365)$ 범위의 정수로 2 사상됨

9.3.3 나머지 부호화

- 예측 나머지는 유한한 크기 a 를 가짐
- 주어진 예측 \hat{x} 에 대해 나머지 ε 은 범위 $-\hat{x} \leq \varepsilon < \alpha - \hat{x}$ 에 있음
- \hat{x} 는 복호화기에 의해 생성될 수 있으므로, 나머지 ε 의 동적 범위는 모듈로 (modulo) a 로 축소 시킬 수 있고 $-\lceil \frac{a}{2} \rceil$ 와 $\lceil \frac{a}{2} \rceil - 1$ 사이의 값으로 사상될 수 있음
- 예측 오류 나머지가 두 측면 기하학적 분포(Two-sided Geometric Distribution, TSGD)를 따른다고 볼 수 있음
- 기하학적 분포를 갖는 순서에 대해서는 최적인 Golomb codes에 기반한 적응적으로 선택된 코드를 사용하여 부호화됨

9.3.4 무손실에 가까운 모드

- 무손실 JPEG-LS 모드는 $\delta=0$ 인 무손실에 가까운 모드의 특별한 경우로 고려될 수 있음
- 무손실에 가까운 압축은 양자화를 사용하여 이루어질 수 있음

$$Q(\varepsilon) = \text{sign}(\varepsilon) \left\lfloor \frac{|\varepsilon| + \delta}{2\delta + 1} \right\rfloor \quad (9.11)$$

- δ 은 작은 개수의 정수값만을 가지기 때문에, 나누기 연산은 lookup table을 사용하여 효율적으로 구현될 수 있음
- 예측과 내용 결정 단계는 오직 양자화된 값에 기초함

9.4 이진 영상 압축 표준

- 많은 문서들이 전자적 형태로 처리됨에 따라 이진 영상(흑백 화소)을 압축하기 위한 효율적인 방법은 수요가 많아짐
 - 팩스 영상
- 영상에서 각 주사선은 검은색과 흰색 화소의 런으로서 다루어짐

9.4.1 JBIG 표준

- 무손실 압축으로 프린트되거나 손으로 쓰여진 문서, 컴퓨터에 의해 생성된 문서, 팩시밀리 전송의 조사된 영상을 부호화하기 위해 주로 사용
- 점진적인 부호화와 복호화 기능을 제공
- 회색도 및 컬러 영상을 부호화하는 데 사용가능 하지만, 주된 목적은 아님
- 세 가지 독립적인 동작 모드
 - 점진적 (progressive)
 - 점진적 호환 순차적(progressive-compatible sequential)
 - 점진적 모드와 호환되는 비트스트림 사용
 - 데이터가 이 모드에서 몇 조각으로 나뉘어짐
 - 단일 점진 순차적(single-progression sequential)
 - 오직 하나의 가장 낮은 해상도 계층을 가짐
- JBIG 부호화기의 두 성분
 - 해상도 축소와 차분 계층 부호화기
 - 가장 낮은 해상도 계층 부호화기

9.4.2 JBIG2 표준 (1)

- JBIG표준은 무손실과 점진적인 부호화 능력을 제공하지만, 생성된 손실된 영상은 원본보다 훨씬 더 나쁜 화질을 가짐
 - 원본 영상 화소 수의 1/4만을 가지고 있기 때문
- JBIG2는 좀더 나은 무손실 압축 성능 제공과 시각적 왜손의 감소, 높은 압축률에서 손실 압축과 결합하는 것을 목표로 함
- 화질 점진적이면서 내용 점진적임
 - 화질 점진적: 영상 화질을 낮은 것에서 높은 화질로 개선되어 가는 것
 - 내용 점진적: 서로 다른 타입의 영상들이 점진적으로 합쳐지는 것을 허용함
- 페이지 간의 유사성을 이용하여 하나의 파일에서 한 문서의 여러 페이지를 표현할 수 있음
- 모델 기반 부호화를 통해 우수한 압축 성능을 제공, 추가적인 부호화 이득을 얻음

9.4.2 JBIG2 표준 (2)

- 모델 기반 부호화

- 내용 기반 부호화와 본질적으로 같음
- 영상 내용을 서로 다른 범주로 분리할 수 있고 각각의 범주를 위해 특별한 모델을 유도해 낼 수 있다면, 더 높은 압축률을 얻을 수 있음
- JBIG 스타일의 부호화에서 적응적 템플릿과 모델 템플릿은 영상 안에서 구조를 획득 그러나 텍스트와 중간조 데이터 사이의 구조적 차이를 명백하게 다루지 않음
- JBIG2는 이러한 데이터 타입을 위해 맞춤 모델을 설계하여 이것을 이용함
- 텍스트와 중간조 영역으로 나누어서 독립적으로 부호화함

9.4.2 JBIG2 표준 (3)

- 텍스트 영역 부호화

- 텍스트 영역을 화소 블록으로 좀더 분할함
- 문자의 모든 화소들을 부호화하는 대신에 이 문자에 대한 하나의 대표적인 경우의 비트맵을 부호화하고 사전(dictionary)에 넣음
- 부호화될 문자에 대해, 알고리즘은 먼저 사전에서 문자와 매치되는 것을 찾음
- 발견되면, 사전에서 일치하는 엔트리에 대한 포인터와 페이지상의 문자의 위치가 모두 부호화됨
- 그렇지 않으면, 화소 블록은 직접적으로 부호화되고 사전에 추가됨
- 이 기술은 패턴 매칭과 치환(pattern matching and substitution)으로 불림
- 조사된 문자들에 대해서는 동일한 방법이 가능하지 않음
- 페이지상의 원래 문자를 재생산하기 위해 정련 데이터를 포함하는 선택 사항을 허락함
- 정련 데이터는 사전에서 매치되는 문자의 화소를 이용하여 문자를 부호화함
- 부호화기는 정련이 정확하거나 손실을 허용할 지의 선택권을 가짐
- 이 기술은 소프트 패턴 매칭(soft pattern matching)이라고 불림

9.4.2 JBIG2 표준 (4)

- 중간조-영역 부호화
 - 두 가지 방법을 제안
 - JBIG에서의 내용 기반 산술 부호화와 비슷
 - 새로운 표준은 1b 템플릿 화소만큼 많이 포함하는 내용 템플릿을 허용
 - 디스크리닝(descreeing)
 - 이전 레벨 영역은 $m_b \times n_b$ 크기의 블록으로 나뉘어짐
 - 결과 화색도 영상은 $m_s = \lfloor (m + (m_b - 1)) / m_b \rfloor \times n_s = \lfloor (n + (n_b - 1)) / n_b \rfloor$ 의 크기를 가짐
 - 화색도 값은 상응하는 $m_b \times n_b$ 블록에서 이전 화소 값의 합이 되도록 계산
 - 화색도 영상의 비트평면은 내용 기반 산술 부호화를 사용하여 부호화됨
- 전처리와 후처리
 - 부호화기는 부호화의 효율을 증가시키기 위해 전처리 단계에서 입력 영상을 수정함
 - 후처리는 시각적으로 더 만족스러운 영상을 생성하는데, 중간조를 위해 유용할 수 있음