

비트매트릭스 산술 부호 방식의 정지영상 압축 시스템

이 제 명[†] · 이 호 석^{††}

요 약

본 논문은 David Taubman의 JPEG2000 시스템보다 기능면에서 우수한 비트매트릭스 부호 방식의 정지영상 압축 시스템을 제안한다. 시스템은 서브블록 단위로 비트매트릭스를 구성하여 산술부호화를 수행하여 40:1 이상의 높은 정지영상 압축율을 보여주고 있다. 비트매트릭스 서브블록은 비트평면에 나타나는 비트들을 2x2 비트매트릭스로 파악하여 서브블록을 구성한 것이다. 산술부호기는 구성된 서브블록을 비트매트릭스 단위로 부호화하여 압축한다. 시스템의 입력 모드는 분할(Segmentation) 모드와 ROI(Region Of Interest) 모드로 구성된다. 분할 모드는 입력 영상을 문자 부분과 배경 영상 부분으로 분할하여 입력할 수 있게 한다. ROI 모드는 입력 영상을 관심 영역과 그 밖의 영역으로 구분하여 입력할 수 있게 한다. 현재 시스템이 나타내는 높은 압축율은 다른 JPEG2000 시스템들과 충분히 비교할 수 있는 수준이다. 시스템은 그 밖에 그레이 코딩을 수행하여 압축율을 향상시킨다.

A Still Image Compression System using Bitmatrix Arithmetic Coding

Je Myung Lee[†] · Ho Suk Lee

ABSTRACT

We propose a novel still image compression system, which is superior in its function than the JPEG2000 system developed by David Taubman. The system shows 40:1 high compression ratio using 2x2 bitmatrix subblock coding. The 2x2 bitmatrix subblock is constructed in the bitplanes by organizing the bits into subblocks composing of 2x2 matrices. The arithmetic coding performs the high compression by the bitmatrices in the subblock. The input of the system consists of a segmentation mode and a ROI(Region Of Interest) mode. In segmentation mode, the input image is segmented into a foreground consisting of letters and a background consisting of the remaining region. In ROI mode, the input image is represented by the region of interest window. The high compression ratio shows that the proposed system is competent among the JPEG2000 products currently in the market. This system also uses gray coding to improve the compression ratio.

키워드: 비트매트릭스 산술 부호(Bitmatrix Arithmetic Coding), 분할 모드(Segmentation Mode), 관심영역 모드(ROI Mode)

1. 서 론

영상이나 문서의 압축은 중요한 연구 주제로서 세계적으로 많이 연구되고 있다[1-3]. 본 논문은 영상 분할[4], 웨이블릿 리프팅(wavelet lifting)[5, 6], 그리고 비트매트릭스(bitmatrix) 서브블록 산술 부호화에 기반한 정지영상 압축 시스템을 제안한다. 웨이블릿 리프팅은 웨이블릿 변환[7, 8]의 한 종류로서 많은 장점을 가지고 있으며 영상 압축에 사용된다. 비트매트릭스 서브블록이란 비트 평면에 존재하는 비트들에 대하여 16x16 서브블록을 구성하고 다시 서브블록의 내부를 2x2 비트매트릭스들로 구분한 것을 의미한다. 2x2 비트매트릭스들에는 총 16가지의 종류가 있다. 이렇게 하면 하나의 16x16 서브블록을 구성하는 기호들은 모두 64개가 된다. 이때 16x16 서브블록에서 모든 비트의 값이 0이거나

1인 경우는 중요한 서브블록으로 간주하여 2x2 비트매트릭스로 재구성하지 않는다. David Taubman의 JPEG2000에서는 비트평면을 서브블록 단위로 구분하고 이에 대하여 콘텍스트(context) 기반 적응(adaptive) 산술부호화[9]를 수행하였다. 그 밖의 문서 압축 방법으로는 참고문헌[10]가 있다. 또한 본 시스템은 입력 문서를 문자가 포함되는 영역인 전경(foreground)과 그 나머지 부분인 배경(background)으로 분리하는 영상 분할을 수행하여 압축을 수행할 수 있다. 영상 분할은 K-평균(means) 클러스터링(clustering) 알고리즘[4]을 개량한 multi-scale bicolor 클러스터링 알고리즘으로 구현하였다. 영상 분할의 이유는 압축과 복원을 수행하는 과정에서 크기가 작고 형태가 가는(thin) 문자의 모양을 최대한 그대로 유지하기 위해서이다.

새로운 정지영상 압축 시스템에 대한 간략한 설명은 다음과 같다. 입력 영상은 RGB 형태로 표현된다. RGB 칼라는 색 변환을 통하여 YUV 형태로 변환된다. 변환된 YUV의 각 컴포넌트는 4:1:1 비율로 다운샘플링된다. 다운샘플링된

[†] 준 회 원 : 호서대학교 대학원 컴퓨터공학부

^{††} 정 회 원 : 호서대학교 컴퓨터공학부 교수
논문접수 : 2004년 2월 26일, 심사완료 : 2004년 5월 26일

YUV 각 컴포넌트에 대해서 웨이블릿 리프팅을 수행하고 생성된 웨이블릿 계수는 그레이 코딩(graycoding)[1] 변환과 양자화(quantization) 과정을 거친다. 양자화 과정을 거친 웨이블릿 계수 값들은 비트 평면을 구성한다. 비트 평면에 대하여 비트매트릭스 서브블록을 구성한다. 비트매트릭스 서브블록을 기반으로 산술부호화를 수행한다.

현재 전 세계적으로 많은 JPEG2000 시스템이 개발되어 판매되고 있으며 대표적인 것으로는 참고문헌에서 제시한 [11-15]이 있다. 이들 시스템들은 대부분 40:1 이상의 압축율을 제시하고 있다. 본 연구에서는 비교 대상으로 David Taubman의 JPEG2000[2]을 이용하였다.

본 논문의 구성한 다음과 같다. 2장은 본론으로 웨이블릿 리프팅, 영상 분할, 그레이코딩, 양자화, 비트매트릭스 서브블록 부호, 그리고 전체적인 시스템의 구성에 대하여 설명한다. 3장에서는 제안된 정지영상 압축 시스템에 대한 실험 결과를 제시한다. 마지막으로 4장에서는 결론과 향후 연구 방향을 제시한다.

2. 본 론

2.1 웨이블릿 리프팅

웨이블릿 리프팅[5,6]은 장점을 가지고 있다. 첫째로, 컨볼루션(convolution) 방식을 사용하는 웨이블릿 변환보다 더 빠르게 웨이블릿 변환을 구현할 수 있다. 예를 들면, 입력 데이터 사이의 유사성을 직접 사용하여 웨이블릿 변환을 구현할 수가 있다. 입력 데이터에서 홀수번째 데이터만을 선택하여 원래 데이터의 낮은(coarse) 데이터로 - 저주파통과 데이터 - 인식하고, 짝수번째 데이터는 홀수번째 데이터의 평균값을 계산하여 추정값으로 사용한다. 여기서 실제 홀수번째 데이터와 추정하여 계산된 데이터의 차이값을 - 고주파통과 데이터-보정하여 데이터의 손실을 없도록 한다. 즉, 입력 데이터의 유사성을 최적으로 사용함으로써 빠르게 웨이블릿 변환을 구할 수가 있다. 이것을 웨이블릿 리프팅이라고 한다. 둘째로, 별도의 기억장소 필요없이 입력 데이터가 저장된 메모리에서 직접 웨이블릿 변환이 가능하다. 즉, 원래 영상 신호가 웨이블릿 변환 계수로 바뀌기 때문에 웨이블릿 리프팅을 수행하는데 추가적인 기억장소가 필요하지 않는 것이다. 셋째로, 웨이블릿 리프팅에서는 순 방향 변환(forward transform)으로부터 역 방향 변환(inverse transform)을 쉽게 얻을 수 있으며 웨이블릿 변환이 완전히 가역적이다.

본 시스템에서는 12가지 웨이블릿 리프팅[6]을 구현하였다. 구현된 웨이블릿 리프팅은 (5/3), (2/6), (SPB), (9/7-M), (2/10), (5/11-C), (5/11-A), (6/14), (SPC), (13/7-T), (13/7-C), (9/7-F) 리프팅등이다. 가장 압축 성능이 좋은 (5/3) 리프팅에 대하여서는 순 방향 변환식과 역 방향 변환식을 제시한다.

• (5/3) 웨이블릿 리프팅

$$d[n] = d_0[n] - \left\lfloor \frac{1}{2}(s_0[n+1] + s_0[n]) \right\rfloor$$

$$s[n] = s_0[n] + \left\lfloor \frac{1}{4}(d[n] + d[n-1]) + \frac{1}{2} \right\rfloor$$

$$s[n] = s_0[n] - \left\lfloor \frac{1}{4}(d[n] + d[n-1]) + \frac{1}{2} \right\rfloor$$

$$d[n] = d_0[n] + \left\lfloor \frac{1}{2}(s_0[n+1] + s_0[n]) \right\rfloor$$

위의 웨이블릿 리프팅 수식에서 $s[n]$, $d[n]$ 은 각각 저역통과(lowpass) 신호, 고역통과(highpass) 신호를 나타낸다. 그리고 $s_0[n]$ 은 $x[2n]$ 을 나타내고, $d_0[n]$ 은 $x[2n+1]$ 을 나타낸다. 웨이블릿 리프팅은 필터 뱅크(filter bank) 구조를 통하여 수행된다.

2.2 영상 분할

영상 분할은 입력 영상을 문자에 해당하는 전경과 나머지 그림에 해당하는 배경으로 분할하는 것을 의미한다. 영상 분할은 기본적으로 K-평균 클러스터링 알고리즘[4]을 multi-scale bicolor 클러스터링 알고리즘으로 확장하고 여기에 전경과 배경 반전 알고리즘을 적용하여 개발하였다. 다음은 multi-scale bicolor 클러스터링 알고리즘과 전경과 배경 반전 알고리즘이다.

• Multi-scale bicolor 클러스터링

- ① 전체 영상을 일정한 크기의 블록으로 분할하여 각 블록에 대하여 블록 이전 칼라 클러스터링(block bicolor clustering) 알고리즘을 수행한다.
- ② 전체 영상을 다시 이전 블록 크기보다 작은 크기의 블록으로 분할한다.
- ③ 분할된 블록들의 전경 칼라와 배경 칼라를 현재 블록들을 포함하고 있는 이전 블록의 전경 칼라와 배경 칼라의 값으로 초기화한다.
- ④ 각 블록들에 대하여 블록 안의 모든 픽셀 값을 전경 칼라 그리고 배경칼라와 비교하여 더 가까운 쪽에 포함시킨다.
- ⑤ 전경에 포함되는 픽셀들과 배경에 포함되는 픽셀들을 이전 블록의 전경 픽셀 값과 배경 픽셀 값에 더하여 전경 칼라 값과 배경 칼라 값을 갱신한다.
- ⑥ 각각의 칼라가 모두 모아질 때까지 ④, ⑤를 반복하고 또한 정해진 크기의 블록을 구성할 때까지 ②부터 반복한다.

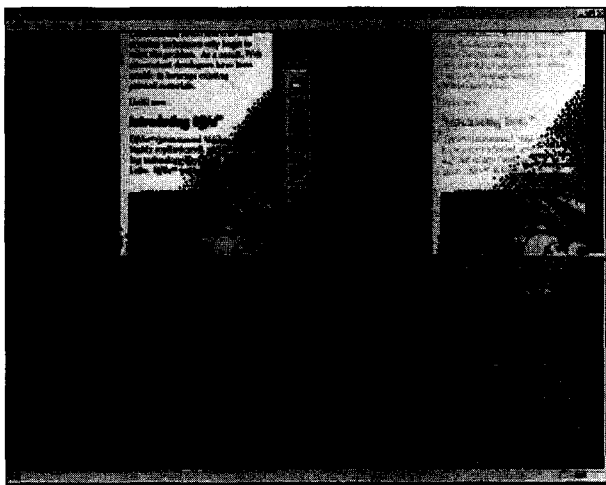
• 전경과 배경 반전

- ① 분할된 전체 영상에서 전경 픽셀 객체와 배경 픽셀 객체를 추출한다.
- ② 각각의 객체가 전체 영상에서 차지하는 면적을 계산

한다.

③ 면적이 큰 전경 객체는 배경으로 반전한다.

문서 분할의 목적은 영상의 전경에 포함된 문자들의 모양을 압축과 복원하는 과정에서 그대로 유지하면서 영상을 최대한으로 압축하기 위해서이다. 전경은 문자들을 포함하고 있기 때문에, 문자의 모양과 가독성을 유지하기 위해서는 일반적으로 PSNR(Peak Signal-to-Noise Ratio) 값이 30dB 이상인 한계까지만 압축하는 것이 좋다. 그러나 배경은 영상만을 포함하고 있기 때문에 PSNR 값이 30dB 이하가 되는 범위까지도 압축할 수 있다. 다시 말하면, 문자를 포함하고 있는 영상을 압축하기 위해서는 문서 분할이 필수적이다. 본 연구에서는 (5/3) 웨이블릿 리프팅을 Hobby 문서와 AT&T 문서에 대하여 수행하고 양자화를 통하여 압축을 단계적으로 높여가면서 압축율과 PSNR을 계산하였다. 다음은 (그림 1)은 문서 분할의 결과이다. 그림에서 윗 부분의 왼쪽이 원래 입력 영상이고 오른쪽이 분할된 배경이다. 아래 부분의 왼쪽이 분할된 전경 부분이고 오른쪽은 비트맵(bitmap)이다. 비트맵은 전경 부분과 배경 부분을 구분하여 주는 역할을 한다.



(그림 1) 문서 분할

2.3 그레이 코딩

정수 웨이블릿 리프팅에 대한 계수 값들은 2바이트의 정수형 자료형에 저장된다. 즉, 하나의 웨이블릿 계수값은 16개의 비트 평면으로 구성된다. 그러나 비트 평면 부호화의 단점은 픽셀 값의 작은 변화가 전체 비트 평면에 큰 변화를 줄 수가 있다는 것이다. 예를 들어, 어떤 픽셀 값이 127(이진수: 01111111)이고 인접한 픽셀의 값이 128(이진수: 10000000)이라면 픽셀 값은 단지 1의 차이만 있으나 전체 비트 평면에서 인접한 모든 비트 값이 다르게 나타난다. 이는 비트의 중복성을 이용한 압축에 효과적이지 못하다. 이러한 현상을 해결하기 위해서 픽셀 값을 그레이코드로 변환한다. m개의

웨이블릿 계수는 m개의 그레이코드 $g_{m-1} \dots g_2 g_1 g_0$ 로 변환된다. 다음은 그레이코드 변환식이다[1].

$$g_i = a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2$$

$$g_{m-1} = a_{m-1}$$

\oplus 부호는 Exclusive OR 연산을 나타낸다. 예를 들어, 127(이진수: 01111111)은 그레이코드로 변환하면 (01000000)이 된다. 128(이진수: 10000000)은 그레이코드로 변환하면 (11000000)이 된다. 즉, 비트 평면에서 이웃하는 비트 사이의 값의 변화는 현격히 줄어든다. 이는 비트사이의 중복성을 증가시켜 압축율을 더욱 높이게 된다.

웨이블릿 계수는 2바이트의 정수형이다. YUV 칼라 모델에서 1개의 픽셀 값을 나타내는 웨이블릿 계수값은 6바이트의 메모리를(Y: 2바이트, U: 2바이트, V: 2바이트) 필요로 한다. 즉, 48(16×3)개의 비트 평면이 생성된다. 즉, 1개의 컴포넌트에 대하여서는, 16개의 비트 평면에 대한 부호화 작업을 수행하여야 한다. 그러나 웨이블릿 계수의 자료형이 2바이트(16비트)의 정수형이지만, 실제 비트 평면을 조사해보면 각 계수값의 범위는 최대 10비트를 넘지 않는다. 사인(sign) 부호로서 10번째 비트에 대한 처리를 해줌으로서 비트량을 줄일 수 있다.

2.4 양자화

스칼라 양자화로서 웨이블릿 계수 값의 크기를 감소시키고 비트 평면에 0값을 생성시키는 역할을 한다. 양자화는 웨이블릿 계수 값을 비트 쉬프트(shift)하는 방식으로 구현하였다. 서브밴드들 중에서 영상의 평균값을 담고 있는 LL밴드의 쉬프트는 LH, HL, HH 밴드보다 상대적으로 적게 함으로써 복원된 영상의 왜곡 현상을 최소화할 수 있도록 하였다. LH, HL, HH 밴드는 쉬프트를 크게 함으로써 압축율을 높이도록 하였다. 또한, 색차변환을 통하여 구한 Y, U, V 신호 값에서 색차 값을 담고 있는 U, V 신호의 쉬프트를 Y 신호에 비해 상대적으로 크게 함으로써 압축율을 더 높이도록 하였다. 양자화는 웨이블릿 계수를 오른쪽 방향으로 쉬프트 연산 후 다시 역 방향인 왼쪽으로 쉬프트 연산을 함으로써 수행한다. 예를 들어 서브밴드를 n비트만큼 양자화시키면 결과적으로 n개의 0값을 오른쪽에 가지게 된다. 각 계수값의 비트에서 0의 증가는 0값을 가진 서브블록의 증가를 가져오며 이 것은 압축율은 증가시키게 된다.

2.5 비트매트릭스 서브블록 산술부호

다음에 형성된 비트평면을 16×16 크기의 서브블록으로 구분한다. 구분된 서브블록 중에서 모든 비트 값이 0 혹은 1인 서브블록은 그 비트 값을 서브블록 전체의 비트 값으로 한다. 다음에 0과 1이 함께 포함되어 있는 서브블록에 대하여서는 그 서브블록을 다시 2×2 비트매트릭스를 사용하여 다

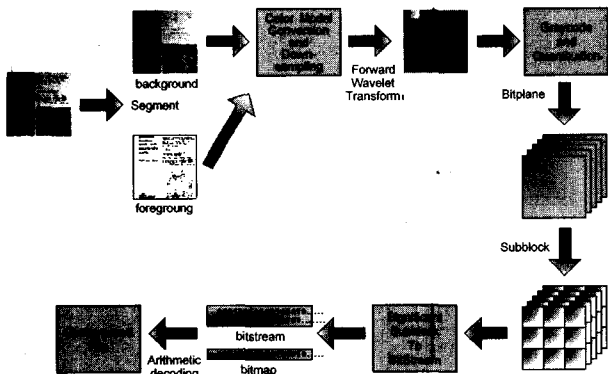
시 나눈다. 2x2 비트매트릭스에는 모두 16가지가 있으며, 16x16 서브블록이 2x2 비트매트릭스로 다시 나누어지면 총 64개의 2x2 비트매트릭스를 포함한다. 산술부호기는 16x16 서브블록을 2x2 비트매트릭스 단위로 부호화한다.

2.6 시스템 수행과정

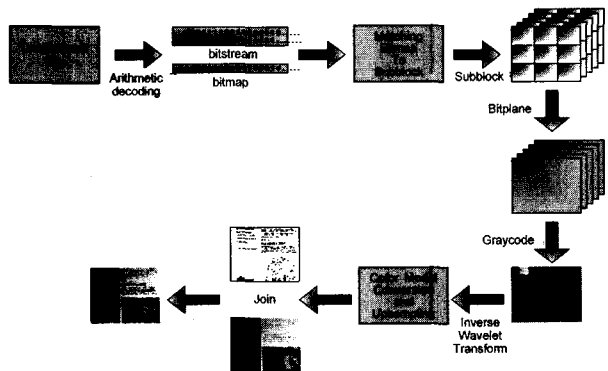
다음은 영상 분할 모드에서의 시스템 부호화 수행 과정을 나타낸 것이다.

- ① 입력 영상을 배경과 전경으로 분할한다.
- ② 배경과 전경에 대해서 RGB에서 YUV로 색 변환을 수행한다.
- ③ Y, U, V 신호에서 색차 값 U, V를 4:1로 다운샘플링한다.
- ④ Y, U, V 컴포넌트에 대해서 웨이블릿 리프팅을 수행한다.
- ⑤ 웨이블릿 계수를 그레이코드로 변환한다.
- ⑥ 각 서브밴드별로 양자화를 수행한다.
- ⑦ 생성된 비트평면에 대하여 비트매트릭스 서브블록을 구성한다.
- ⑧ 비트매트릭스 서브블록에 산술 부호화를 적용하여 압축한다.

다음 (그림 2)와 (그림 3)은 압축 시스템과 복원 시스템의 전체 구조를 나타낸다.



(그림 2) 압축 시스템



(그림 3) 복원 시스템

3. 실험 결과

압축을 실험은 3가지로 구분하여 실시하였다. ① 첫 번째는 분할 모드에서의 전경과 배경 압축을 실험이다. ② 두 번째는 ROI 모드에서의 압축을 실험이다. 그리고 ③ 세 번째는 JPEG, JPEG2000, 제안된 시스템에 대한 압축을 비교 실험이다. 실험 환경은 인텔 P4 1GHz, 512MByte이다. 다음은 압축을 실험에 사용된 실험 영상이다. ①과 ④는 사람 영상, ②와 ③은 자연 영상, 그리고 ⑤와 ⑥은 글자가 포함된 영상이다. 분할 모드 압축을 실험에서는 ⑤와 ⑥ 영상을 이용하였고, ROI 모드에서의 압축을 실험과 시스템 압축을 비교 실험에서는 ①, ②, ③, ④, ⑤, ⑥ 영상을 모두 사용하였다.

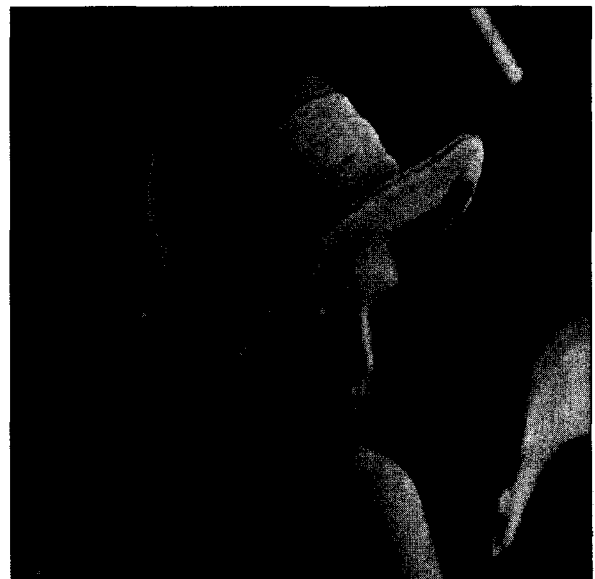
- ① Lena 영상(400x255)
- ② Field 영상(640x480)
- ③ Pier 영상(1024x768)
- ④ Cameraman 영상(256x256)
- ⑤ AT&T 영상(512x512)
- ⑥ Hobby 영상(768x512)

실험 결과로는 압축율과 PSNR 값을 제시하였다. $PSNR_{rms}$ 식은 다음과 같다. 예러는 $e(i, j) = f(i, j) - F(i, j)$ 식으로 정의한다. $f(i, j)$ 는 입력이고 $F(i, j)$ 는 출력을 나타낸다.

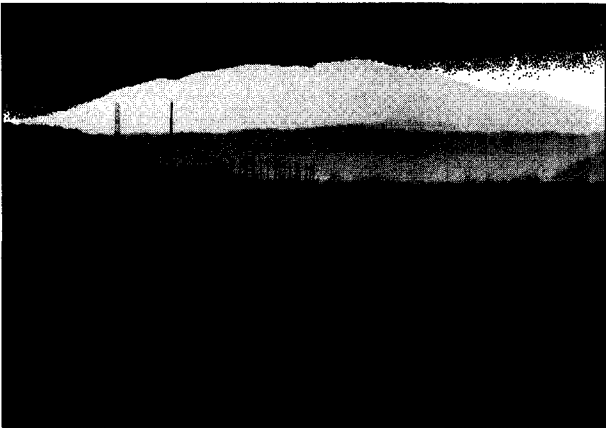
$$PSNR_{rms} = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \text{ dB}$$

$$MSE = \frac{\sum_{i=0}^{W-1} \sum_{j=0}^{H-1} [f(i, j) - F(i, j)]^2}{W \times H}$$

255 값은 픽셀의 최대값이다. 다음은 실험에 사용된 입력 영상들이다.



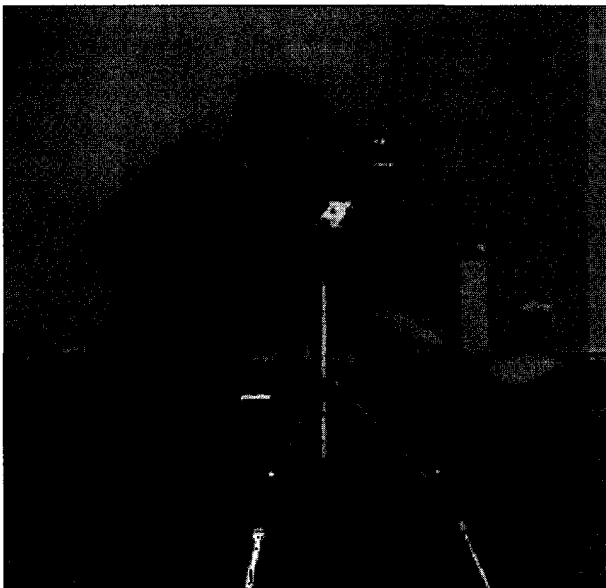
(그림 4) Lena 영상



(그림 5) Field 영상

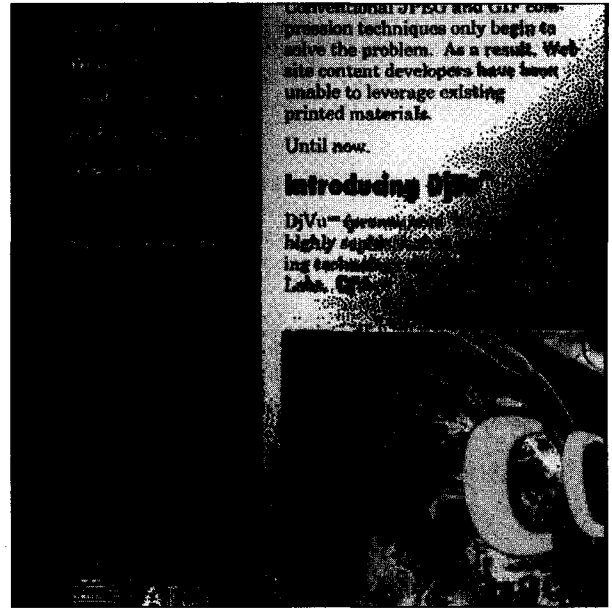


(그림 6) Pier 영상

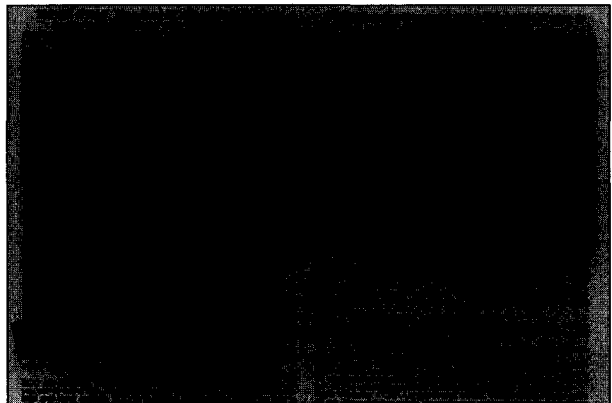


(그림 7) Cameraman 영상

① 첫 번째는 영상 분할 모드에서의 압축율 실험이다. <표 1>은 양자화에 의한 배경과 전경의 압축율과 PSNR 실험



(그림 8) AT&T 영상



(그림 9) Hobby 영상

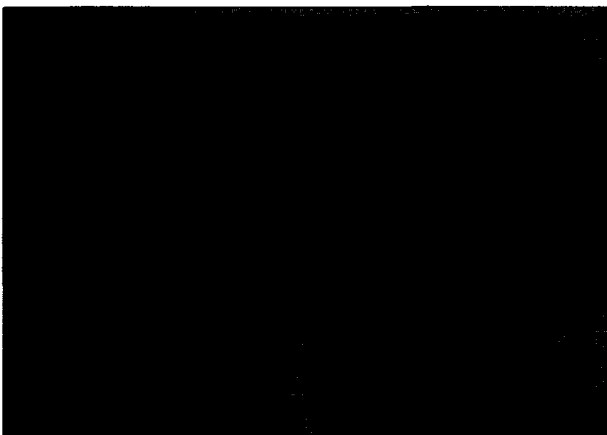
결과이다. 실험은 모두 (5/3) 웨이블릿 리프팅을 사용하였다. 표에서 Level 3은 웨이블릿 리프팅의 분해 레벨(decomposition level)이 3이라는 것을 의미한다. 그리고 (Y)LL /LH/HL/HH는 Y 칼라 성분 에 대한 LL, LH, HL, HH 서브밴드를 나타낸다. 예를 들어, 세 번째 열의 0/2/2/3은 3레벨로 분해한 웨이블릿 계수에 대하여 양자화로 LL 서브밴드는 0개의 비트, LH 서브밴드는 2개의 비트, HL 서브밴드도 2개의 비트, 그리고 HH 서브밴드는 3개의 비트를 사용한다는 의미이다. 참고로, 웨이블릿 계수는 총 10비트를 차지한다. 압축율은 원래의 영상 파일 크기와 압축한 후 파일의 크기에 대한 비율이다. 표에서 5.91%는 원래의 파일 크기가 5.91%(16.9 : 1)로 압축되었다는 의미이다. 실험은 (양자화 1), (양자화 2), (양자화 3)의 방법으로 진행하여 전경의 압축율, 배경의 압축율, 그리고 전체 PSNR을 계산하였다.

<표 1> 양자화에 의한 전경과 배경의 압축율과 PSNR

양자화비트	Level3 : (Y)LL/LH/HL/HH Level2 : (Y)LL/LH/HL/HH Level1 : (Y)LL/LH/HL/HH Level3 : (U)LL/LH/HL/HH Level2 : (U)LL/LH/HL/HH Level1 : (U)LL/LH/HL/HH Level3 : (V)LL/LH/HL/HH Level2 : (V)LL/LH/HL/HH Level1 : (V)LL/LH/HL/HH	(양자화1)	(양자화2)	(양자화3)
			0/2/2/3 0/3/3/4 0/4/4/5 0/3/3/4 0/4/4/5 0/5/5/6 0/3/3/4 0/4/4/5 0/5/5/6 0/3/3/4 0/7/7/8 0/3/3/4 0/4/4/5 0/5/5/6 0/7/7/8	0/2/2/3 0/4/4/5 0/6/6/7 0/3/3/4 0/5/5/6 0/7/7/8 0/3/3/4 0/5/5/6 0/7/7/8 0/3/3/4 0/9/9/10 0/3/3/4 0/6/6/7 0/9/9/10 0/9/9/10
AT&T 512×512×24 786,432 (bytes)	전경압축율(%)	46,536(bytes) 5.91%	30,048(bytes) 3.82%	22,651(bytes) 2.88%
	배경압축율(%)	19,941(bytes) 2.53%	16,295(bytes) 2.07%	13,863(bytes) 1.76%
	PSNR(dB)	31.37	27.11	24.07
Hobby 768×512×24 1,179,648 (bytes)	전경압축율(%)	76,879(bytes) 6.51%	45,094(bytes) 3.82%	35,880(bytes) 3.04%
	배경압축율(%)	26,951(bytes) 2.28%	21,794(bytes) 1.84%	18,830(bytes) 1.59%
	PSNR(dB)	31.79	25.54	24.07

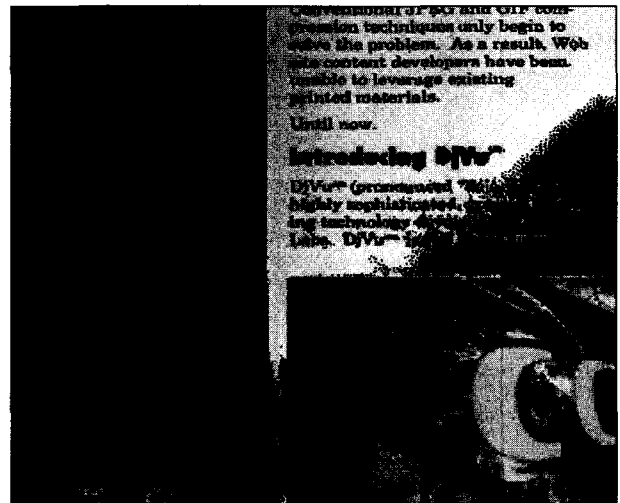
다음 (그림 10) Hobby 영상 그리고 (그림 11) AT&T 영상은 <표 1>의 (양자화 3) 방식을 사용하여 압축한 후 복원한 영상이다. 이 실험 결과를 제시한 이유는 이정도의 압축을 수행한 경우에 두 영상 모두 PSNR은 24.07dB이었지만, 복원 영상의 품질은 전경에 포함되어 있는 문자의 모양과 가독성을 충분히 유지하는 수준에서 매우 우수한 상태를 보여 주었기 때문이다. 그리고 <표 1>이 보여 있듯이, 배경을 전경에 비하여 약 두 배 이상 더 압축할 수 있었다.

다음 (그림 10)은 제안된 정지영상 압축 시스템의 영상 분할 모드에서 Hobby 영상을 <표 1> (양자화 3)의 방법으로 압축한 후 복원한 영상이다.



(그림 10) 복원 Hobby 영상

다음 (그림 11)도 제안된 정지영상 압축 시스템의 영상 분할 모드에서 AT&T 영상을 (양자화 3)의 방법으로 압축한 후 복원한 영상이다.



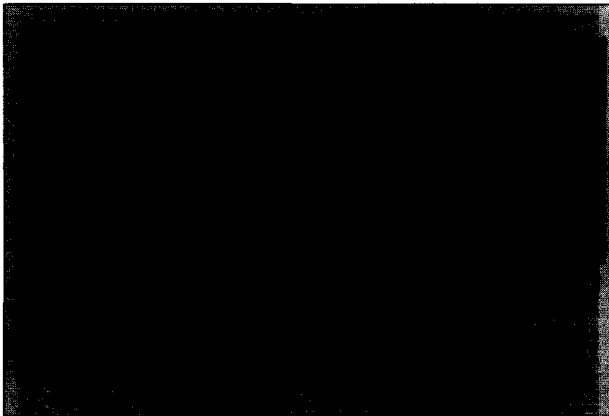
(그림 11) 복원 AT&T 영상

다음에 분할 모드 경우에서의 전체 압축율을 계산하였다. Hobby와 AT&T의 경우에는 전경과 배경으로 분할하여 압축을 하여 전경과 배경의 압축율을 각각 구한 다음에 전체 압축율을 계산하였다. Hobby의 경우에 전경의 압축율은 3.04%, 배경의 압축율은 1.59%, 그리고 전체 압축율은 4.46%(약 21.6 : 1)로 계산되었다. AT&T의 경우에 전경의 압축율은 2.88%, 배경의 압축율은 1.76%, 그리고 전체 압축율은 4.64%(약 22.4 : 1)로 계산되었다.

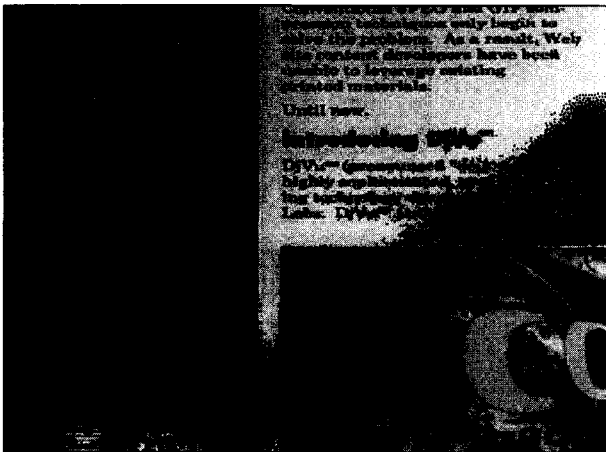
<표 2> 분할모드에서의 전체 압축율과 PSNR

	원영상 파일크기	압축율	PSNR(dB)
AT&T	512×512×24 786,432(bytes)	36,490(bytes), 4.64% 21.6 : 1	24.07
Hobby	512×512×24 1,179,648(bytes)	52,612(bytes), 4.46% 22.4 : 1	24.07

그러나 제안된 정지영상 압축 시스템의 분할 모드 목적은 전경에 포함된 문자의 모양과 가독성을 압축하고 복원하는 과정에서 최대한 유지하는 하는 것에 있다. 다음 (그림 12)와 (그림 13)은 Hobby와 AT&T 영상을 JPEG2000으로 압축하고 복원한 경우의 그림이다. 앞의 (그림 10)의 복원 Hobby 영상 그리고 (그림 11)의 복원 AT&T 영상과 비교하면 제안된 정지영상 압축 시스템으로 압축한 결과에서 문자가 덜 손상된 것을 확인할 수 있다. 즉, JPEG2000으로 압축하고 복원한 경우에는 전체적으로 문자가 포함된 부분이 더 뿌옇고 흐릿하게 손상된 것을 알 수 있다.



(그림 12) JPEG2000으로 압축하고 복원한 Hobby 영상



(그림 13) JPEG2000으로 압축하고 복원한 AT&T 영상

② 두 번째는 ROI 모드에 있어서의 압축을 실험이다. ROI 사각형의 크기는 일반적인 크기이며 최대 압축율을 계산하였다.

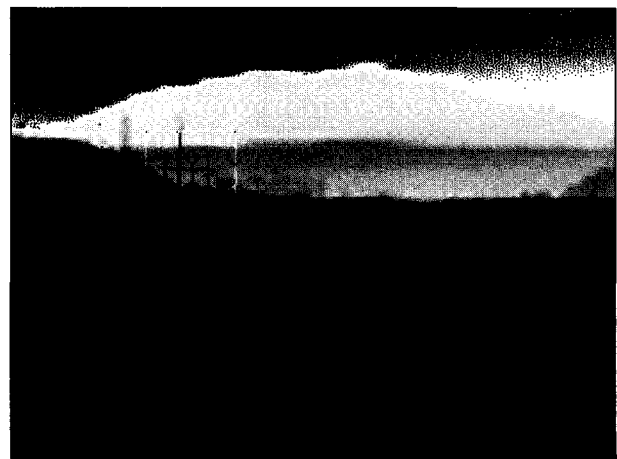
ROI 모드에서의 압축율은 Lena의 경우에 1.41%(70.8 : 1) Cameraman의 경우에 1.92%(51.9 : 1)을 보여주었다. ROI 모드에서의 압축율은 물론 ROI 사각형의 크기에 좌우될 수 있다. 다음 그림들은 ROI 모드에서 압축한 후 복원한 영상이다.

<표 3> 제안된 정지영상 압축 시스템 ROI 모드 압축율 실험

	원영상 파일크기	압축율
Lena	512×512×24 786,432(byte)	11,100(byte), 1.41% 70.8 : 1
Field	640×480×24 921,600(byte)	9,768(byte), 1.05% 94.3 : 1
Pier	1024×768×24 2,359,296(byte)	18,555(byte), 0.78% 127.1 : 1
Cameraman	256×256×24 196,608(byte)	3,785(byte), 1.92% 51.9 : 1
AT&T	512×512×24 786,432(byte)	13,824(byte), 1.75% 56.8 : 1
Hobby	512×512×24 1,179,648(byte)	19,417(byte), 1.64% 60.7 : 1



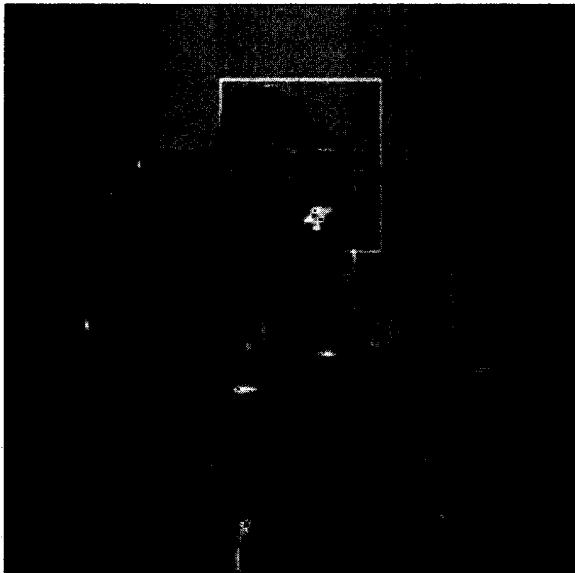
(그림 14) ROI 모드에서 압축한 후 복원한 Lena 영상



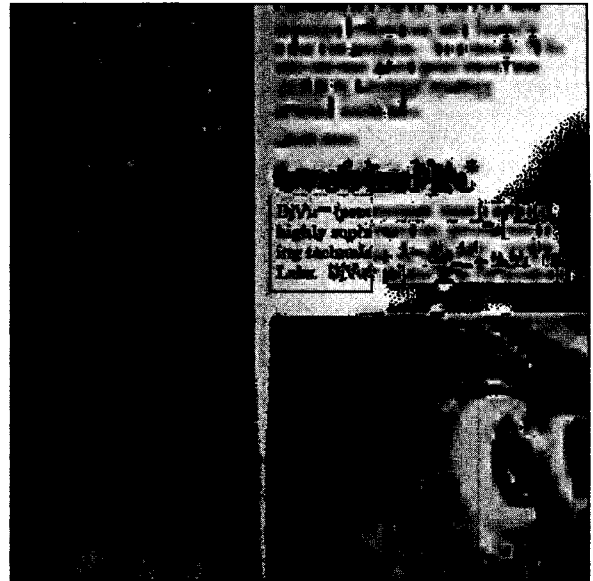
(그림 15) ROI 모드에서 압축 후 복원한 Field



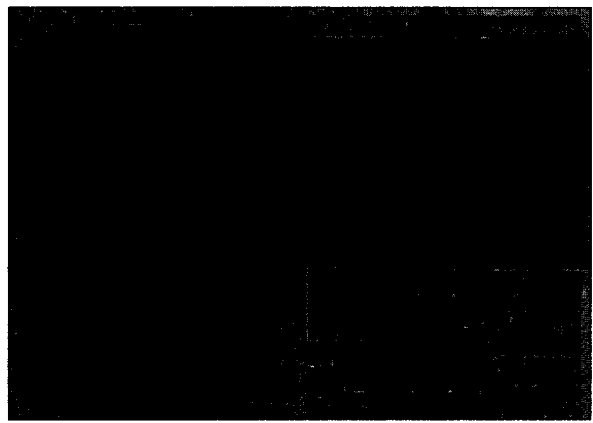
(그림 16) ROI 모드에서 압축 후 복원한 Pier 영상



(그림 17) ROI 모드에서 압축 후 복원한 Cameraman 영상



(그림 18) ROI 모드에서 압축 후 복원한 AT&T 영상



(그림 19) ROI 모드에서 압축 후 복원한 Hobby 영상

ROI 마스크에 의하여 선택된 영상의 관심 영역들은 복원한 후에 매우 선명하며 원래의 영상을 그대로 보여주고 있음을 알 수 있다.

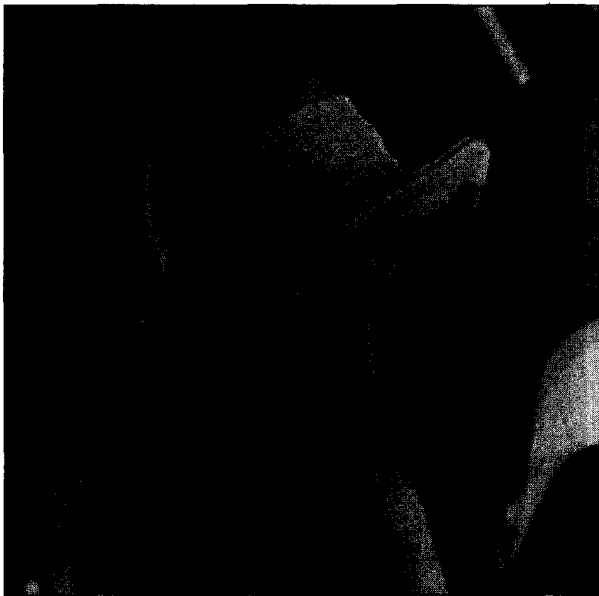
③ 세 번째는 IJG(Independent JPEG Group)-JPEG, JPEG2000, 제안된 정지영상 압축 시스템 사이의 압축율과 PSNR 비교 실험이다.

〈표 4〉 IJG-JPEG, JPEG2000, 제안된 정지영상 압축 시스템 압축율과 PSNR

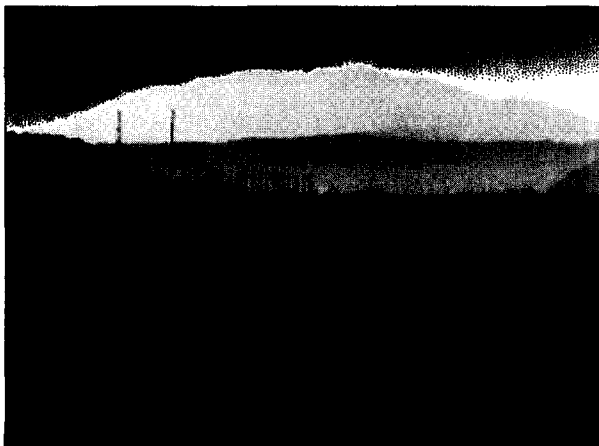
	원영상 파일크기	IJG-JPEG		JPEG2000		제안된 정지영상 압축 시스템	
		압축율	PSNR	압축율	PSNR	압축율	PSNR
Lena	512×512×24 786,432(byte)	4.76%	30.723	2.08%	34.790	2.45%	32.015
Field	640×480×24 921,600(byte)	3.07%	30.375	2.07%	33.499	2.14%	30.114
Pier	1024×768×24 2,359,296(byte)	1.49%	34.449	2.07%	41.274	1.14%	34.529
Cameraman	256×256×24 196,608(byte)	14.48%	28.942	2.05%	29.640	2.81%	26.144
AT&T	512×512×24 786,432(byte)	6.74%	27.183	2.07%	26.856	5.90%	29.180
Hobby	512×512×24 1,179,648(byte)	5.07%	26.919	2.08%	26.309	5.72%	28.497

위 실험은 본 연구에서 개발한 제안된 정지영상 압축 시스템이 이전의 IJG-JPEG 시스템보다는 압축율과 PSNR 측면에서 우수함을 보여주고 있다. 그리고 JPEG2000 시스템과도 충분히 비교할 수 있음을 보여주고 있다. 또한 기능적인 면에서는 제안된 정지영상 압축 시스템은 영상 분할 모드를 지원하여 문자가 포함된 영상의 경우에는 문자의 손상 없이 영상을 압축할 수 있는 장점이 있다.

다음 (그림 20), (그림 21), (그림 22), (그림 23), (그림 24), (그림 25)는 위의 <표 5>에 제시되어 있는 대로, 제안된 정지영상 압축 시스템으로 최대한 압축한 후 복원한 영상들이다. 원래의 입력 영상인 (그림 4), (그림 5), (그림 6), (그림 7), (그림 8), (그림 9)와 비교해 보면 시각적으로 큰 차이를 느낄 수 없음을 알 수 있다. 다만, (그림 23)의 Cameraman 영상은 실험에 사용한 영상의 크기가 256×256으로 상당히 작아서 영상의 품질에 다소의 차이가 발생하였다.



(그림 20) Lena 복원 영상



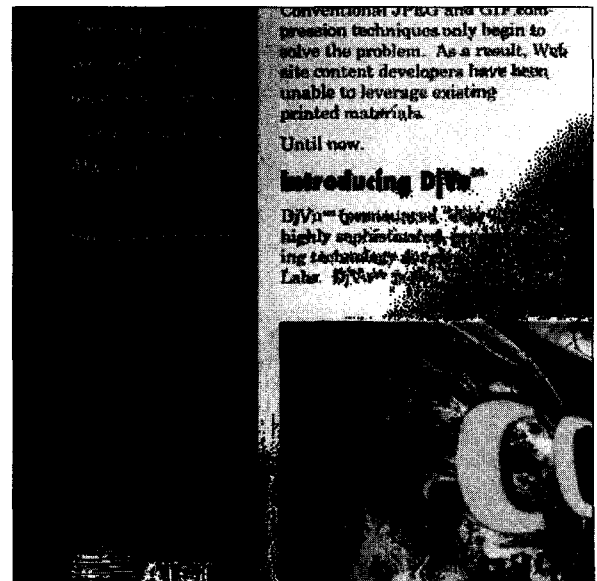
(그림 21) Field 복원 영상



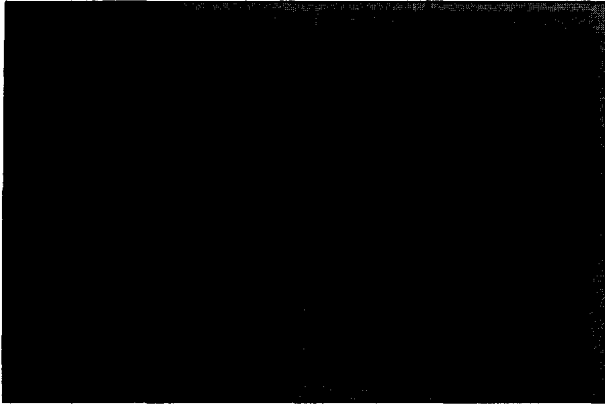
(그림 22) Pier 복원 영상



(그림 23) Cameraman 복원 영상



(그림 24) AT&T 복원 영상



(그림 25) Hobby 복원 영상

4. 결 론

본 논문은 새로운 기능과 높은 압축율의 정지영상 압축 시스템을 제안하였다. 제안된 정지영상 압축 시스템은 Lena 영상의 경우에 PSNR 32.015dB에서 40.7 : 1의 높은 압축율을 보여주었다. Pier 영상의 경우에는 PSNR 34.529dB에서 87.4 : 1의 높은 압축율을 보여주었다. 제안된 정지영상 압축 시스템은 웨이블릿 리프팅, 그레이 코딩, 서브밴드 단위의 양자화, 그리고 비트매트릭스 서브블록 산술부호화 방식을 사용하여 평균적으로 40 : 1 이상의 높은 압축율을 보여주고 있다. 이러한 높은 압축율과 PSNR은 제안된 정지영상 압축 시스템이 실용적 가치가 있음을 충분히 입증하는 것이다. 또한 이 시스템의 입력은 분할 모드를 지원하여 문자가 포함된 영상의 압축에는 기존의 JPEG2000보다 훨씬 유리하다.

향후 연구 방향으로는 압축율과 PSNR을 더욱 높이는 연구와 입력 영상의 명암 대비 비율을 복원영상에서도 그대로 유지할 수 있도록 하는 연구가 필요하다.

참 고 문 헌

[1] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing(2nd edition), Addison-Wesley, 1992.
 [2] David S. Taubman, Michael W. Marcellin, JPEG2000 Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.
 [3] David Taubman, "High Performance Scalable Image Compression with EBCOT," IEEE Trans. on Image Processing, Vol.9, No.7, pp.1158-1170, July, 2000.
 [4] Linda G. Shapiro, George C. Stockman, Computer Vision, Prentice-Hall, 2001.
 [5] Wim Sweldens, "The lifting scheme : A custom-design con-

struction of biorthogonal wavelets," Applied and Computational Harmonic Analysis, Vol.3, No.2, pp.186-200, 1996.
 [6] Michael D. Adams, Faouzi Kossentini, "Reversible Integer-to-Integer Wavelet Transforms for Image Compression : Performance Evaluation and Analysis," IEEE Trans. on IP, Vol.9, No.6, June, 2000.
 [7] David Walnut, An Introduction to Wavelet Analysis, Birkhauser, 2002.
 [8] Stephane Mallat, A Wavelet Tour of Signal Processing(2nd edition), Academic Press, 1999.
 [9] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," CACM., Vol.30, pp.520-540, June, 1987.
 [10] Jeong-Kwon Kim, Kyeong Ho Yang, Choong Woong Lee, "Document Image Compression by Nonlinear Binary Sub-band Decomposition and Concatenated Arithmetic Coding," IEEE Trans. on CSVT, Vol.10, No.7, pp.1059-1067, October, 2000.
 [11] <http://www.leadtools.com>, LEADTOOLS JPEG2000 Plug-in, 2003.
 [12] <http://www.aware.com>, JPEG2000 Codec, 2003.
 [13] <http://www.morgan-multimedia.com>, Morgan JPEG2000 Plug-in, 2003.
 [14] <http://www.pegasusimaging.com>, ImagePress JPEG2000 Plug-in, 2003.
 [15] <http://www.amphion.com>, JPEG2000 Codec, 2003.



이 제 명

e-mail : linuxyi@empal.com
 1997년~2003년 호서대학교 컴퓨터공학부
 공학사
 2003년~현재 호서대학교 컴퓨터공학과
 대학원 석사과정
 관심분야 : 영상압축, 영상처리, JPEG2000



이 호 석

e-mail : hslee@office.hoseo.ac.kr
 1979년~1983년 서울대학교 전자계산기공
 학과 공학사
 1983년~1985년 서울대학교 컴퓨터공학과
 대학원 공학석사
 1989년~1993년 서울대학교 컴퓨터공학부
 대학원 공학박사
 1994년~현재 호서대학교 컴퓨터공학부 교수
 관심분야 : 영상공학, 영상압축, JPEG2000, MPEG-4, MPEG-7