

웨이블릿 리프팅과 비트평면 부호화에 기반한 새로운 문서 코덱 시스템

이 호 석[†]

요 약

본 논문은 문서 분할, 웨이블릿 리프팅, 그리고 비트 평면 부호화에 기반한 문서 압축과 복원 시스템을 제시한다. 압축 과정에서 문자들의 모양을 유지하기 위하여 문서 분할을 사용한다. 웨이블릿 변환 방법으로는 웨이블릿 리프팅을 사용하며 웨이블릿 계수 값에 대하여 비트 평면을 구성하여 중요 서브블록 단위로 압축을 수행한다. 중요 서브블록 부호화를 통하여 압축율을 높임과 동시에 압축속도 향상도 이룩하였다. 양자화로는 서브밴드 단위의 비트 쉬프트 방법을 사용하였다. 웨이블릿 리프팅의 전단계로 색차변환, 부표본화를 수행하였으며, 서브블록 부호화의 전단계로 그레이코드 변환을 수행하였다. 여러 영상과 문서를 대상으로 실험하여 시스템의 성능을 확인하였다.

A New Document Codec System based on Wavelet Lifting and Bitplane Coding

Lee Ho Suk[†]

ABSTRACT

In this paper, we present the development of document compression codec using segmentation, wavelet lifting and bitplane coding. We use the segmentation to preserve the text appearance. We performed integer-to-integer wavelet lifting and also performed bitplane subblock coding for document compression. We acquired a high compression ratio and an efficient compression by encoding only the significant subblocks in the bitplane subblock coding. We also implemented scalar quantization by subband-oriented bit shifting. The system performs color conversion and downsampling before wavelet lifting and also performs graycode conversion and quantization before subblock coding. In the experiment, we show the performances of the system by presenting the high compression ratios and high PSNR values.

Key words: Compression, Segmentation, Wavelet Lifting, Bitplane Subblock Coding

1. 서 론

영상이나 문서의 압축은 중요한 연구 주제로서 세계적으로 많이 연구되고 있다. 본 논문은 문서 분할, 웨이블릿 리프팅(wavelet lifting) 그리고 비트 평면(bitplane) 부호화에 기반한 문서 압축 코덱(codec)을 제시한다. 웨이블릿 리프팅은 여러 가지 장점을 가지고 있기 때문에 웨이블릿 변환의 구현에 많이 사용된

다. 비트 평면 부호화에서는 웨이블릿 계수(wavelet coefficients)에 대하여 서브블록 단위로 비트 평면 부호화[1]를 수행한다. 웨이블릿 계수에 대한 부호화 방법으로는 SPIHT 알고리즘[2]이 있다. 그러나 이 알고리즘을 사용하면 수행 속도의 효율화가 필요하다. JPEG-2000[3]에서는 Daubechies (9/7) 실수형 웨이블릿 변환을 사용하였다. 그러나 실수형 웨이블릿 변환을 사용하면 많은 수의 비트 평면이 생성되어 효율이 떨어질 수 있다. 그 밖의 문서 압축 방법으로는 참고문헌[4]가 있다. 이 논문은 주로 이진 문서의

접수일 : 2002년 10월 22일, 완료일 : 2003년 3월 10일

[†] 정회원, 호서대학교 컴퓨터공학부 근무

압축에 대하여 논의한다. 그러나 본 논문은 칼라 영상과 문서의 압축을 소개한다. 그리고 본 시스템은 입력 문서를 배경(background)과 문자에 해당하는 영역인 전경(foreground)으로 분리하는 문서 분할[5]을 수행한다. 문서를 분할하는 이유는, 압축하는 과정에서 문자의 모양을 유지하기 위해서이다.

연구에 대한 간략한 설명은 다음과 같다. 입력 영상은 RGB 형태로 표현된다. 입력 영상의 RGB 칼라는 색차 변환을 통하여 휘도(luminance)값과 색차(chrominance)값을 나타내는 YUV 형태로 변환된다. 변환된 YUV의 각 컴포넌트는 4:1:1 비율로 부표본화된다. 부표본화된 YUV 각 컴포넌트에 대해서 웨이블릿 리프팅을 수행하고, 생성된 웨이블릿 계수는 그레이코드(graycode)[1] 변환과 양자화(quantization) 과정을 거친다. 양자화 과정을 거친 웨이블릿 계수 값들은 비트 평면으로 구성되며, 비트 평면은 다시 서브블록(subblock)[3]으로 나누어진다. 서브블록의 최대 크기는 64×64이다. 서브블록에 대하여는 중요도(significance) 여부를 검사한다. 중요한 서브블록에 대해서는 부호화를 수행하고, 중요하지 않은 서브블록에 대하여서는 그 위치정보만 기록한다. 결과적으로 시스템의 수행 속도는 빨라지고 비트스트림 양은 감소되어 압축률은 상승된다. 마지막으로 산술 부호화[10]를 수행하여 비트스트림을 더욱 압축한다.

2장에서는 웨이블릿 리프팅에 대한 소개와 (5/3) 웨이블릿 리프팅 수식 그리고 구현 방법에 대하여 소개한다. 본 논문에서 제안한 문서 압축 코덱은 참고문헌[7]에서 제시한 정수 웨이블릿 리프팅을 모두 구현하였다. 3장에서는 비트 평면 기반 서브블록 부호화 과정에 대해서 자세히 설명한다. 4장에서는 실험결과를 제시하고 5장에서는 성능 평가를 제시한다. 마지막으로 6장에서는 결론을 제시한다.

2. 웨이블릿 리프팅(Wavelet lifting)

웨이블릿 리프팅은 장점[8,9]을 가지고 있다. 첫째로, 회전(convolution) 방식의 웨이블릿 변환보다 더 빠른 웨이블릿 변환을 구현할 수 있다. 리프팅은 고역통과(highpass)와 저역통과(lowpass) 필터 사이의 유사성을 최적으로 사용함으로써 계산의 속도를 증진시킬 수 있다. 둘째로, 웨이블릿 변환의 계산을

제자리에서 가능하도록 한다. 즉, 원래 영상 신호가 웨이블릿 변환 계수로 바뀌기 때문에 추가적인 메모리가 필요하지 않다는 것이다. 셋째로, 웨이블릿 리프팅에서는 순 방향 변환(forward transform)으로부터 역 방향 변환(inverse transform)을 쉽게 얻을 수 있으며, 웨이블릿 변환이 완전히 가역적이다.

2.1 웨이블릿 리프팅 수식

본 시스템에서는 12 가지 웨이블릿 리프팅[7]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) 리프팅 등이다. (5/3) 리프팅에 대하여서는 순 방향 변환식과 역 방향 변환식을 제시하기로 한다.

• (5/3) 리프팅

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

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

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

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

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. 다음 그림 1은 웨이블릿 리프팅을 수행하기 위한 위한 3-레벨의 필터 뱅크 구조이다.

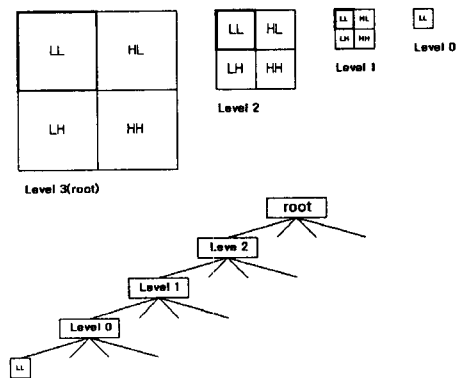


그림 1. 3-레벨 필터 뱅크 구조

3. 비트평면 기반 서브블럭 부호화

이 절에서는 서브블럭 부호화를 통한 웨이블릿 계수의 압축 과정을 설명한다. 웨이블릿 리프팅은 압축을 위한 전 단계로 웨이블릿 계수를 생성한다. 정수 웨이블릿 리프팅을 통한 각 웨이블릿 계수 값은 2바이트의 메모리를 필요로 한다. 일반적인 칼라 영상은 픽셀 하나를 표현하기 위해서 3바이트의 메모리를 필요로 한다. 한 픽셀에 해당하는 웨이블릿 계수는 6바이트 (2바이트×3)를 요구한다. 원 영상에 비해 웨이블릿 계수는 2배의 크기가 된다. 본 장에서는 웨이블릿 리프팅을 통하여 얻은 계수값의 비트평면 부호화 알고리즘을 소개한다. 다음은 시스템 수행 과정이다.

- (1) 입력 영상을 배경과 전경으로 분할한다.
- (2) 배경과 전경에 대해서 RGB에서 YUV로 색 변환을 수행한다.
- (3) Y,U,V 신호에서 색차 값에 해당하는 U, V 값을 부표분화한다.
(휘도 값 Y와 색차 값 U, V의 부표분화 비율은 4:1:1 이다.)
- (4) Y,U,V 컴포넌트에 대해서 웨이블릿 리프팅을 수행한다.
- (5) 웨이블릿 계수를 그레이코드로 변환한다.
- (6) 각 서브밴드별로 양자화를 수행한다.
(LL, LH, HL, HH에 대해서 각기 다른 양자화를 적용한다.)
- (7) 웨이블릿 계수를 서브블럭으로 나눈다.
- (8) 각 서브블럭에 대해서 부호화를 수행하여 비트스트림을 생성한다.
- (9) 비트스트림에 산술 부호화를 적용하여 압축한다.

3.1 문서 분할(Document segmentation)

문서 분할은 입력 문서를 문자에 해당하는 전경과 그림에 해당하는 배경으로 분할한다. 문서 분할은 기본적으로 K-평균(means) 알고리즘[5]을 사용하여 컬러 집합(클러스터링)을 구함으로서 이루어진다. K-평균 알고리즘은 영상의 모든 픽셀 값과 각 클러스터에 속해 있는 픽셀들의 평균값과의 거리를 비교하여 집합하는 알고리즘이다. 다음의 그림 2는 문서 분할의 결과이다.

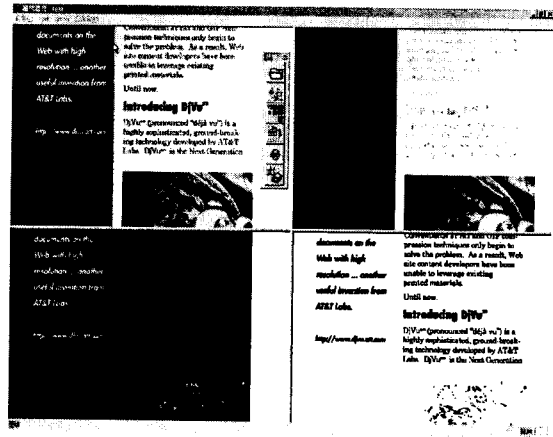


그림 2. 문서 분할의 결과

본 논문에서 문서 분할의 목적은 2 가지이다. 하나는 전경에 포함된 문자들의 모양을 압축과 복원하는 과정에서 유지하기 위해서이고, 다른 하나는 문서에 대한 압축을-왜곡을 해결하기 위해서이다. 압축을-왜곡은 영상이나 문서를 손실 압축을 통하여 압축하는 과정에서, 비트양이 줄면서 복원 영상에 나타나는 왜곡을 설명하는 개념이다[6]. 전경은 문자들을 포함하고 있기 때문에, 문자의 모양과 가독성을 유지하기 위해서는 PSNR(Peak Signal-to-Noise Ratio) 값이 30이상인 한계까지만 압축할 수 있다. 그러나 배경은 영상만을 포함하고 있기 때문에 PSNR 값이 30이하가 되는 범위까지도 압축율을 높일 수 있다. 다시 말하면, 문서에 대하여 압축을-왜곡을 조절하기 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. 그림 13과 그림 14는 실험 결과이다.

3.2 비트평면 부호화(Bitplane coding)

비트 평면 부호화는 웨이블릿 계수에 대하여 비트 평면을 구성하여 부호화를 수행하는 것이다. 본 논문의 시스템은 웨이블릿 리프팅을 수행한 계수값에 대하여 비트 평면에 기반한 서브블럭 부호화를 수행한다. 정수 웨이블릿 리프팅에 대한 계수 값들은 2 바이트의 정수형 자료형에 저장된다. 즉, 하나의 웨이블릿 계수값은 16 개의 비트 평면으로 구성된다. 그러나 비트 평면 부호화의 단점은 픽셀 값의 조그마한

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 제거하기 위해서 픽셀 값을 그레이코드로 변환한다. m 개의 웨이블릿 계수는 m 개의 그레이코드 $g_{m-1} \dots g_2 g_1 g_0$ 로 변환된다. 다음은 그레이코드 변환식이다[1].

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

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

⊕ 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 압축을 더욱 효과적으로 할 수 있도록 한다.

그레이코드로 변환된 웨이블릿 계수는 서브블록으로 나누어진다. 각 서브블록의 크기는 영상의 특성에 따라 가변적으로 변할 수 있다. 본 논문의 시스템은 서브블록을 최소 (8×8)에서 최대 (64×64)까지 사용자가 지정할 수 있도록 하였다.

그림 3의 (a)는 웨이블릿 계수에 대한 비트 평면을 나타낸다. (b)는 각 비트 평면에 대한 서브블록 처리 과정을 나타낸다. 각 비트 평면은 서브블록으로 나누어지고 각 서브블록에 대해서 중요도(significance)

여부를 검사한다. 중요도는 서브블록 속에서 1의 존재 여부에 따라 결정된다. 1 값이 단 한개라도 존재하면 그 서브블록은 “중요한(significant)” 서브블록으로 간주된다. 반대로, 서브블록 내의 모든 값이 0이라면 그 서브블록은 “중요하지 않은(insignificant)” 서브블록으로 간주된다. 0으로만 구성된 중요하지 않은 서브블록은 부호화에서 제외되며, 전체 비트 평면에서 그 서브블록의 위치에 대한 정보만이 서브블록 맵에 기록된다. 중요한 서브블록은 전체 서브블록이 부호화되며 서브블록 맵에는 1값으로 저장이 된다.

비트 평면에 기반한 서브블록 부호화는 전체 비트 평면에 대해서 모두 수행된다. 웨이블릿 계수는 2바이트의 정수형이다. YUV 칼라 모델에서 1개의 픽셀 값을 나타내는 웨이블릿 계수값은 6바이트의 메모리들(Y:2바이트, U:2바이트, V:2바이트) 필요로 한다. 즉, 48(16×3)개의 비트 평면이 생성된다. 즉, 1개의 컴포넌트에 대하여서는, 16개의 비트 평면에 대한 서브블록 부호화 작업을 수행하여야 한다.

그러나 웨이블릿 계수의 자료형이 2바이트(16비트)의 정수형이지만, 실제 비트 평면을 조사해보면 각 계수값의 범위는 최대 10비트를 넘지 않는다. 웨이블릿 계수값은 정수로서 양수와 음수로 표현이 된다. 사인(sign) 부호로서 10번째 비트에 대한 처리를 해줌으로서 비트량을 줄일 수 있다. 본 시스템에서는 웨이블릿 계수에 대한 정수 자료형 2바이트(16비트)에서 10비트만을 처리한다.

“중요한(significant)” 서브블록에 대해서는 비트 스트림을 출력한다. 시스템은 서브블록을 좌측상단에서 우측하단 방향으로 스캔하면서 비트스트림을 출력한다. “중요하지 않은(insignificant)” 서브블록은 비트스트림을 출력하지 않는다. 단지, 서브블록 맵에 그 위치 정보만을 기록한다.

3.3 양자화(Quantization)

본 시스템의 양자화는 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 최소화하도록 하였다. LH,HL,HH 밴드는 쉬프트를 크게 함으로써

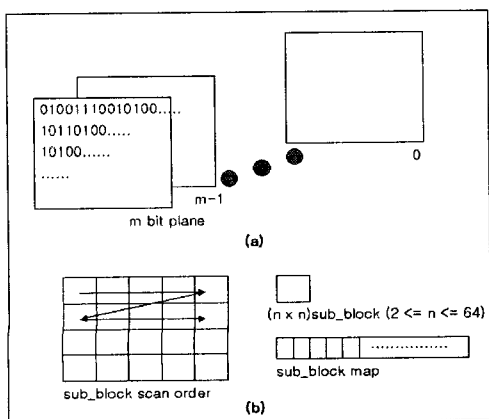


그림 3. 비트 평면 부호화

압축률을 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 압축률을 더 높이도록 하였다.

그림 4는 웨이블릿 계수의 양자화 과정을 나타낸다. 그림에서 오른쪽 쉬프트 연산 후 다시 역 방향으로 쉬프트 연산을 한다. 결과적으로 LL 서브밴드는 n 비트만큼 0값을 오른쪽에 가지게 된다. 비트 평면에 기반한 부호화에서 각 계수 값의 비트에서 0의 증가는 서브블록 부호화에서 “중요하지 않은 서브블록”의 증가를 가져온다. 그리고 이 것은 서브블록 부호화를 수행할 때 비트스트림 양의 감소를 가지고 오게 되며, 따라서 압축율은 증가하게 된다.

다음 그림 5와 그림 6은 압축 시스템과 복원 시스템

의 전체 구조를 나타낸다.

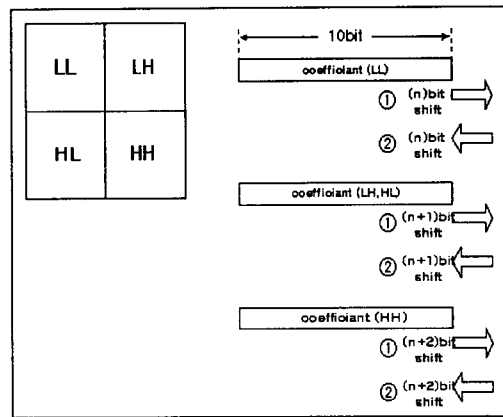


그림 4. 웨이블릿 계수의 양자화

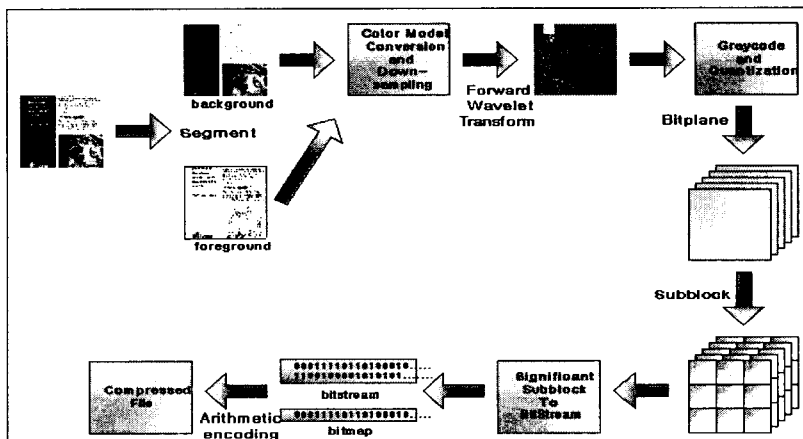


그림 5. 압축 시스템

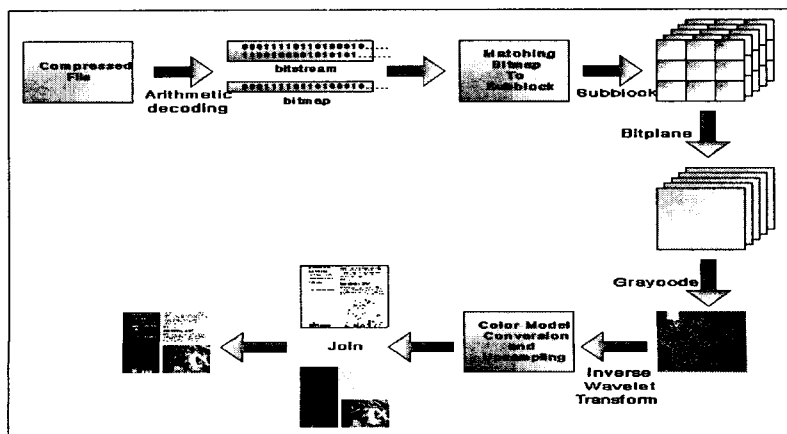


그림 6. 복원 시스템

4. 실험 결과

실험 환경으로 CPU는 셀러론 433MHz, 주메모리는 192 MByte이다. 다음은 본 논문의 문서 압축 코덱의 성능을 실험하기 위해서 사용된 영상이다. 본 시스템은 2가지 모드로 작동된다. 2가지 모드는 영상 압축 모드와 문서 압축 모드이다. (a) Lenna 영상은 영상 압축 모드에서 압축하고 (b)와 (c)는 문서 압축 모드에서 압축하여 실험하였다.

(a) Lenna 영상 (400×255) (b) Hobby 문서 (791×565) (c) AT&T 문서 (512×512)

Lenna 영상에 대하여서는 웨이블릿 리프팅과 비트평면 부호화를 수행하였고, Hobby 문서와 AT&T 문서에 대해서는 문서 분할, 웨이블릿 리프팅, 그리고 비트 평면 부호화를 수행하였다. 실험 결과로는 압축율과 PSNR 값을 제시하였다. 웨이블릿 리프팅 실험에서는 LL, LH, HL, HH 서브밴드에 대하여 9/7/7/6 비트 쉬프트를 디폴트(default) 양자화 값으로 사용하였다. Root-mean-square 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 출력이다.

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

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

255 값은 픽셀의 최대값이다. 그림 7은 실험에 사용된 영상을 나타낸다.

문서 압축 모드에서는 우선 문서 분할을 사용하여 입력 문서를 배경과 전경으로 분할한다. 배경과 전경 각각에 대하여 비트 평면 부호화를 이용하여 압축을 수행하였으며, 압축된 파일과 원래 파일의 크기를 비

교하여 압축율을 계산하고 평가하였다. 또한, PSNR 값을 계산하여 복원 영상에 대한 품질(quality)을 평가하였다. 그리고 실제 복원된 영상을 제시함으로써 입력 영상과의 차이를 비교하여 볼 수 있도록 하였다. 그림 8과 그림 9 그리고 그림 10과 그림 11은 웨이블릿 리프팅에 의한 배경과 전경의 압축율과 PSNR 실험 결과이다. 그림 13과 그림 14는 양자화에 의한 배경과 전경의 압축율과 PSNR 실험 결과이다. 위의 다양한 실험 결과들을 관찰하면 압축율-왜곡에 대처할 수 있는 배경과 전경의 양자화 값을 얻을 수가 있을 것이다.

그림 8과 그림 9는 배경과 전경에 대하여, 8 가지의 웨이블릿 리프팅에 대한 압축율을 그래프와 수치로 보여주고 있다.

실험 결과에 앞에서 소개한 (5/11-A), (SPC), (13/7-C), 그리고 (6/14) 리프팅에 관한 결과는 제시되지 않았다. 그 이유는 (5/11-A) 리프팅은 (5/11-C) 리프팅과 거의 동일한 결과를 내었으며, (SPC)는 (SPB)와 거의 동일한 결과를 그리고 (13/7-C)는 (13/7-T)와 거의 동일한 결과를 내었기 때문이다. (6/14) 리프팅은 배경의 경우에는 (2/6) 리프팅과 그리고 전경의 경우에는 (9/7-F) 리프팅과 거의 동일

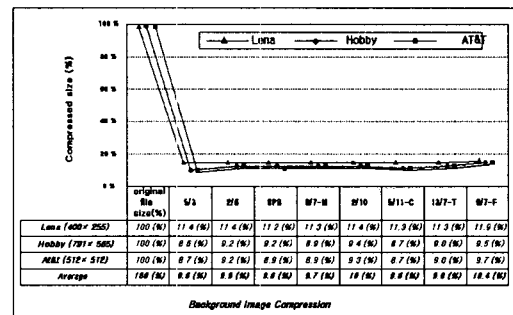


그림 8. 배경 영상의 압축 비율



그림 7. 실험영상 (a), (b), (c)

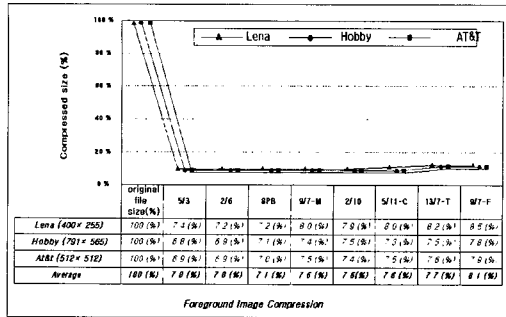


그림 9. 전경 영상의 압축 비율

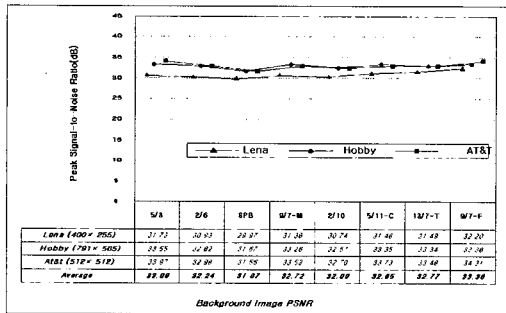


그림 10. 배경 영상에 대한 PSNR

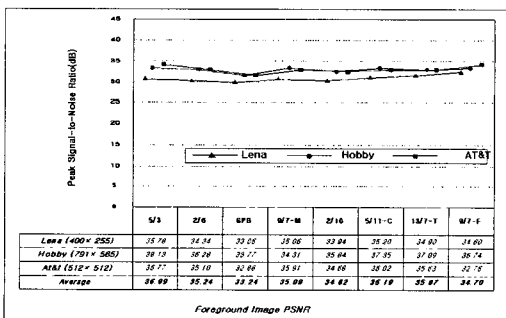


그림 11. 전경 영상에 대한 PSNR

한 결과를 제시하였다.

원래 영상과 복원 영상을 비교한 PSNR 그래프와 수치는 그림 10과 그림 11이 나타낸다.

다음 그림 12는 복원한 영상을 나타낸다.

그림 13과 그림 14는 양자화에 의한 배경과 전경의 압축율과 PSNR 실험 결과이다. 실험은 (5/3) 웨이블릿 리프팅을 가지고 수행하였다. 그림에서, 예를 들어 10/9/8은 양자화로 LL 서브밴드는 10개의 비트, LH와 HL 서브밴드는 9개의 비트 그리고 HH 서브밴드는 8개의 비트를 사용한다는 의미이다. 참고로, 웨이블릿 계수는 총 10 비트를 차지한다. 압축율은 원래의 영상과 그리고 양자화를 수행한 영상과의 비율이다.

	quantization (LL/LH,HL/HH)	10/10/10 (bit)	10/9/8 (bit)	9/8/7 (bit)	9/7/6 (bit)	8/7/6 (bit)	8/6/5 (bit)	7/6/5 (bit)	7/5/4 (bit)
		Hobby (791x 565)	compression rate (%)	24.0	16.1	11.5	8.5	8.4	6.8
	PSNR	39.59	37.90	35.84	33.56	33.27	30.38	29.76	27.29
AT&T (512x 512)	compression rate	22.6	15.0	11.0	8.5	8.4	6.8	6.7	5.6
	PSNR	41.05	39.12	36.64	33.97	33.56	30.48	29.71	27.25

Background (5/3)

그림 13. 양자화에 의한 배경의 압축율과 PSNR

	quantization (LL/LH,HL/HH)	10/10/10 (bit)	10/9/8 (bit)	9/8/7 (bit)	9/7/6 (bit)	8/7/6 (bit)	8/6/5 (bit)	7/6/5 (bit)	7/5/4 (bit)
		Hobby (791x 565)	compression rate (%)	13.6	10.7	8.7	6.9	6.8	5.2
	PSNR	46.47	45.35	42.75	38.06	37.85	32.94	32.71	27.66
AT&T (512x 512)	compression rate	12.8	10.1	8.2	6.6	6.6	5.1	5.1	3.9
	PSNR	43.02	42.30	40.23	36.77	36.57	32.56	32.27	27.96

Foreground (5/3)

그림 14. 양자화에 의한 전경의 압축율과 PSNR



그림 12. 복원 영상

다음 그림 15와 그림 16은 Hobby 영상 그리고 그림 17과 그림 18은 AT&T 영상에 대하여 LL 서버밴드는 7비트, LH와 HL 서버밴드는 6비트, 그리고 HH 서버밴드는 5비트로 양자화하고 복원한 영상이다. 이 실험 결과를 제시한 이유는 7/6/6/5로 양자화를 수행한 경우에, 전경의 PSNR은 32.71과 32.27이고 배경의 PSNR은 29.76과 29.71이었으며, 압축율은 전경이 100:5.1이고 배경은 100:6.7이었으며, 복원 영상의 품질은, 전경에 포함되어 있는 문자의 모양과 가독성을 유지하는 수준에서, 우수한 상태를 보여 주었기 때문이다. 7/5/5/4로 양자화한 경우에는 전경의 PSNR이 27.66과 27.96으로 되어서 문자의 모양과 가독성을 유지하기가 어려운 것으로 판단하였다. 결론적으로, 현재의 실험에서 양자화에 의한 전경의 최대 압축은 7/6/6/5 비트 스위트까지이다. 그리고 전경이 배경에 비하여 압축이 더 많이 된 이유는 전경이 동일한 칼라를 많이 포함하고 있어서 전경에 대하여

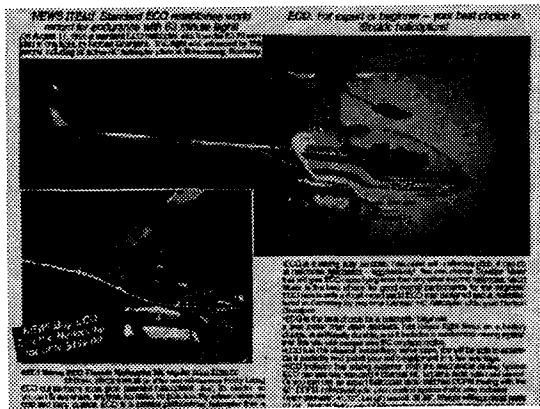


그림 15. 입력 Hobby 영상

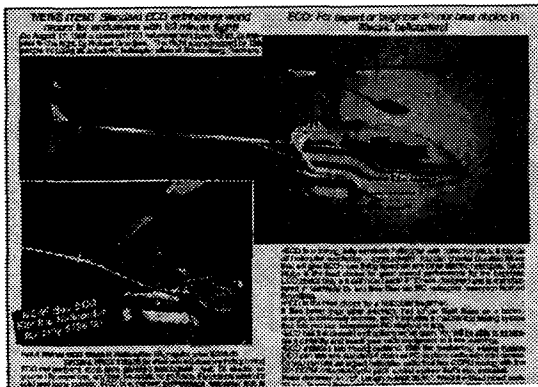


그림 16. 복원 Hobby 영상

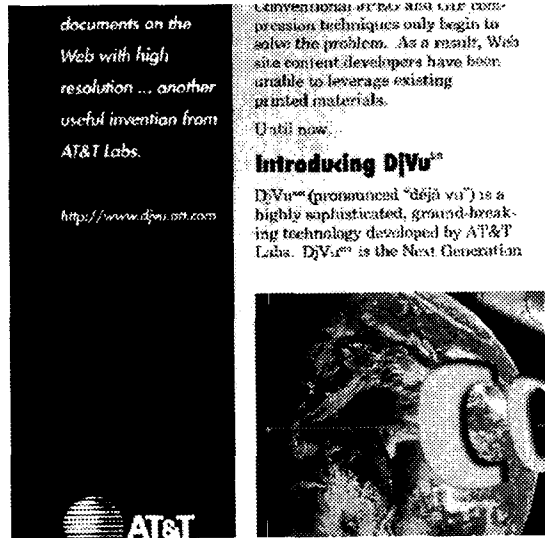


그림 17. 입력 AT&T 영상

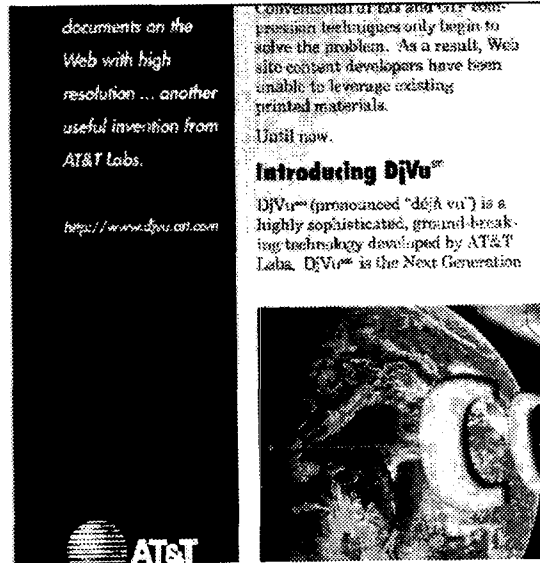


그림 18. 복원 AT&T 영상

웨이블릿 리프팅을 수행하고 양자화를 수행하고 서버블록을 구성하였을 경우에 중요하지 않은 서버블록의 개수가 많아져서 생성되는 비트양이 줄기 때문이다. 배경은 여러 가지의 칼라를 포함하고 있기 때문에 압축율이 전경보다 낮은 것으로 판단하였다.

다음 그림 15와 그림 16은 Hobby 영상에 대한 실험 결과이다.

다음 그림 17과 그림 18은 AT&T 영상에 대한 실험 결과이다.

IJG(Independent JPEG Group)의 JPEG 소프트웨어를 사용하여 Lenna, Hobby, AT&T 영상에 대하여 압축실험을 실시하여 다음과 같이 압축율과 PSNR 결과를 얻었다.

표 1. JPEG 압축율과 PSNR

	파일 크기	압축율	PSNR	비고
Lenna	400 × 255 × 24, 270,054(bytes)	13.6:1	29.92	
Hobby	791 × 565 × 24, 1,342,494(bytes)	12.8:1	26.67	
AT&T	512 × 512 × 24, 786,486(bytes)	13.5:1	27.60	

본 연구의 방법에 의하여 실험하여 다음과 같은 결과를 얻었다. 이 방법에서는 Hobby와 AT&T의 경우에 전경과 배경으로 분할하여 압축을 하고, 압축된 전경과 배경으로부터 원래의 문서를 복원하기 때문에 결과를 각각 구하였다. Hobby의 경우에 전경의 압축율은 27:1, 배경의 압축율은 17.9:1였다. AT&T의 경우에 전경의 압축율은 25.6:1, 배경의 압축율은 17.9:1였다.

표 2. 제안방법의 압축율과 PSNR

	원래 파일크기	압축율	PSNR	비고
Lenna	400 × 255 × 24, 270,054(bytes)	13.5:1	35.78	
Hobby	791 × 565 × 24, 1,342,494(bytes)	전경 27.0:1	27.66	통합 압축율 (10.75:1)
		배경 17.9:1		
AT&T	512 × 512 × 24, 786,486(bytes)	전경 25.6:1	27.96	통합 압축율 (10.53:1)
		배경 17.9:1		

두 실험 결과를 비교해 보면, Lenna의 경우에는 본 연구의 방법이 압축율은 조금 낮으나 PSNR값은 좋은 것을 알 수 있다. 그러나 이 경우는 LL, LH, HL, HH 서브밴드에 9/7/7/6 양자화를 수행한 결과이고, 만약 양자화를 7/5/5/4로 높이면 압축율은 더욱 높일 수 있을 것으로 생각된다. Hobby와 AT&T의 경우에는 전경과 배경의 압축율은 기존 JPEG의 방법보다 훨씬 높다. 그러나 이 경우에는 전경과 배

경으로부터 영상을 복원하기 때문에 통합 압축율을 구하면 Hobby의 경우는 10.75:1이고 AT&T의 경우는 10.53:1이다. 그러나 본 코덱의 목적은 전경부분에 포함된 문자의 모양을 압축하고 복원하는 과정에서 되도록 손상을 덜 입도록 하는 것에 있다. 다음은 Hobby와 AT&T 영상의 경우에 JPEG으로 압축하고 복원한 경우의 그림이다. 앞의 그림 16 복원 Hobby 영상 그리고 그림 18 복원 AT&T 영상과 비교하면, 본 논문에서 제시한 방법으로 압축한 결과에서 문자가 덜 손상된 것을 확인할 수 있다. 즉, JPEG으로 압축하고 복원한 경우에는 전체적으로 문자가 포함된 부분이 더 뿌옇게 손상된 것을 알 수 있다.

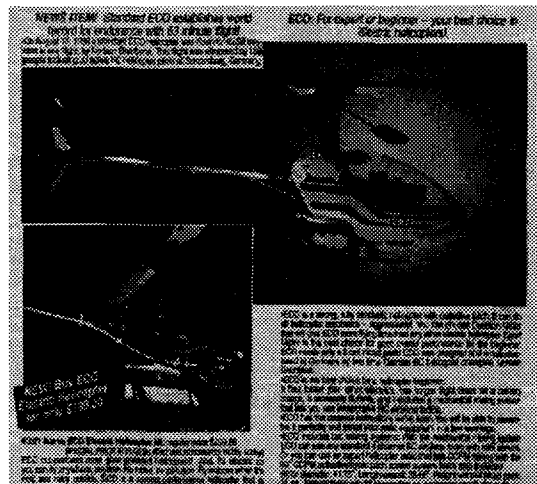


그림 19. JPEG으로 압축하고 복원한 Hobby 영상

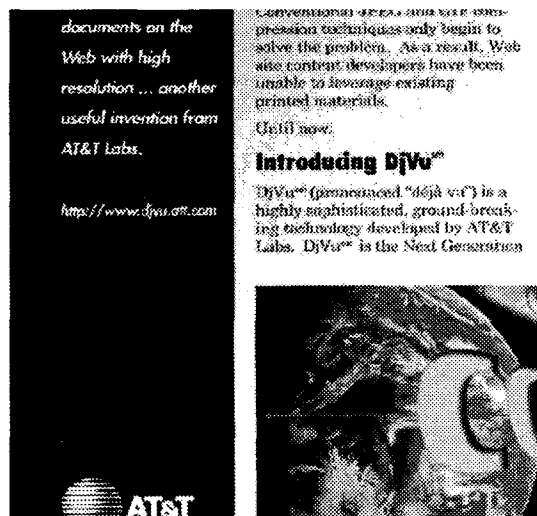


그림 20. JPEG으로 압축하고 복원한 AT&T 영상

5. 성능 평가

본 논문에서 소개한 문서 분할, 웨이블릿 리프팅, 그리고 비트 평면 부호화를 이용한 문서 압축 코덱은 7/6/6/5 양자화를 적용하였을 경우에 배경에 대해서는 최대 100:6.7의 압축율을 보였으며 전경에 대하여서는 최대 100:5.1의 압축율을 보였다.

웨이블릿 리프팅 실험에서는 양자화로 LL, LH, HL, HH 서브밴드에 대하여 9/7/7/6 쉬프트를 디폴트(default) 값으로 사용하였다. 배경 영상에서는 (5/3) 과 (5/11-C) 웨이블릿 리프팅이 가장 높은 압축률을 보였다. (9/7-F) 변환은 압축률이 비교적 좋지 못하지만 PSNR은 가장 높은 값을 나타내었다. (5/3) 과 (5/11-C) 웨이블릿 리프팅은 역시 높은 압축률을 나타내면서도 비교적 높은 PSNR을 보이고 있다. (5/3)의 평균 압축률은 100:9.6 이고 평균 PSNR 값은 33.08 이었다. (5/11-C)의 평균 압축률도 100:9.6 이고 평균 PSNR 값은 32.85 이었다. 전경 영상에서는 (5/3)과 (2/6) 웨이블릿 리프팅이 가장 높은 압축률을 나타내었다. (5/3) 웨이블릿 리프팅은 압축률과 동시에 높은 PSNR을 나타내고 있다. 이 경우, (5/3) 리프팅의 평균 압축률은 100:7.0 이고 평균 PSNR 값은 36.09 이었다. (2/6) 리프팅의 평균 압축률도 100:7.0 이고 평균 PSNR 값은 35.24 이었다. 반면에 (9/7-F)는 압축율과 PSNR 모두에서 가장 좋지 않음을 볼 수 있었다.

여러 웨이블릿 리프팅을 사용하여 시스템을 실험한 결과, (5/3) 웨이블릿 리프팅이 압축율과 PSNR에서 평균적으로 우수하였다. 배경 경우의 평균 압축율은 100:9.6 이고 평균 PSNR 값은 33.08 이었으며, 전경 경우의 평균 압축율은 100:7.0 이고 평균 PSNR 값은 36.09 이었다.

양자화에 대한 실험 (5/3) 웨이블릿 리프팅을 활용하여 Hobby 문서와 AT&T 문서에 대하여 수행하였으며, 7/6/6/5로 양자화한 경우에 전경의 압축율은 두 문서 모두 100:5.1이고 PSNR은 각각 32.71과 32.27이었으며, 배경의 압축율은 모두 100:6.7이고 PSNR은 각각 29.76과 29.71이었다. 여기서 전경의 PSNR 값이 30 이상인 것을 유의하기 바란다. 그리고 이 양자화 단계에서는 복원 영상의 품질은 우수한 상태로 유지되었다. 그러나 양자화를 한단계 더 수행하여 7/5/5/4로 양자화한 경우에는 전경의 PSNR 값은 30 이하가 되었다.

결론적으로, 본 논문에서 제시한 시스템의 압축율은 웨이블릿 리프팅의 선택과 양자화 단계의 선택에 의하여 조절할 수가 있으며 현재 배경에 대해서는 100:6.7까지 그리고 전경에 대하여서는 100:5.1까지 압축율을 높일 수가 있다. 본 논문에서 제시한 방법이 문자가 포함된 문서의 압축에는 기존의 JPEG보다 유리하다는 것을 실험으로 제시하였다. 본 연구에서 압축율 향상을 위하여 사용한 주요 방법은 그레이코드 변환, 양자화, 비트평면 부호화 그리고 중요 서브블록 부호화이다. 압축을 위한 그 외의 방법으로는 칼라 변환, 부표본화, 산술 부호화등을 사용하였다. 시스템 수행 속도의 향상을 위하여 사용한 방법에는 비트평면 부호화와 중요 서브블록 부호화등이 있다.

6. 결 론

본 논문에서는 문서 분할, 웨이블릿 리프팅, 그리고 비트 평면 부호화를 이용한 문서 압축 시스템에 대하여 제시하였다. 시스템을 실험한 결과, (5/3) 웨이블릿 리프팅이 압축율과 PSNR에서 평균적으로 우수하였다. 9/7/7/6 디폴트 양자화 단계에서 배경의 경우의 평균 압축율은 100:9.6 이고 PSNR 값은 33.08이었으며, 전경 경우의 평균 압축율은 100:7.0 이고 PSNR 값은 36.09이었다. 본 논문에서 제시한 시스템의 압축율은 웨이블릿 리프팅의 선택과 양자화 단계의 선택에 의하여 조절할 수가 있으며, 7/6/6/5 양자화에서 배경은 100:6.7 그리고 전경은 100:5.1까지의 압축율을 보여주었다.

본 논문에서 제시한 방법은 문자가 포함된 문서의 압축에는 기존의 JPEG보다 유리하다. 그리고 본 연구에서 압축율 향상을 위하여 사용한 주요 방법은 그레이코드 변환, 양자화, 비트평면 부호화 그리고 중요 서브블록 부호화이다. 시스템 수행 속도의 향상을 위하여 사용한 방법에는 비트평면 부호화와 중요 서브블록 부호화등이 있다.

향후 개선 방향으로, 서브블록 부호화에서 서브블록의 중요도에 따라 부호화하는 방식을 더욱 발전시키고 아울러 좀더 세분화된 비트 평면 부호화를 개발하는 방안을 고려해 보아야 할 것이다.

참 고 문 헌

- [1] Rafael C. Gonzalez, Richard E. Woods, Digital

Image Processing, Addison-Wesley, 1992.

[2] Amir Said, William A. Pearlman, "A New Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees," IEEE Trans. on CSVT Vol. 6, pp. 243-250, June 1996.

[3] David Taubman, "High Performance Scalable Image Compression with EBCOT," IEEE Trans. on IP, Vol. 9, No. 7, pp. 1158-1170, 2000.

[4] Jeong-Kwon Kim, Kyeong Ho Yang, Choong Woong Lee, "Document Image Compression by Nonlinear Binary Subband Decomposition and Concatenated Arithmetic Coding," IEEE Trans. on CSVT, Vol. 10, No. 7, pp. 1059-1067, October 2000.

[5] Leon Bottou, Steven Pigeon, "Lossy Compression of Partially Masked Still Images," Data Compression Conference(J. A. Storer, Martin Cohn, eds), 1988.

[6] Cormac Herley, Jelena Kovacevic, Kannan Ramchandran, and Martin Vetterli, "Tilings of the Time-Frequency Plane : Construction of Arbitrary Orthogonal Bases and Fast Tiling Algorithms," IEEE Trans. on SP, Vol., 41, No., 12, pp.3341-3359, December 1993.

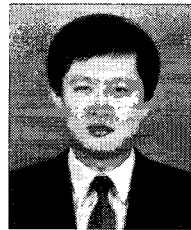
[7] Michael D. Adams, Faouzi Kossentini, "Re-

versible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis," IEEE Trans. on IP, Vol. 9, No. 6, June 2000.

[8] Wim Sweldens, Peter Schroeder, "Building your own wavelets at home," Wavelets in Computer Graphics, ACM SIGGRAPH Course Notes, 1996.

[9] Wim Sweldens, "The lifting scheme : A custom-design construction of biorthogonal wavelets," Applied and Computational Harmonic Analysis, Vo. 3, No. 2, pp. 186-200, 1996.

[10] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," CACM., Vol. 30, pp. 520-540, June 1987.



이 호 석

1979년 3월~1983년 2월 서울대학교 전자계산기공학과 공학사 졸업
 1983년 3월~1985년 2월 서울대학교 컴퓨터공학과 대학원 석사 졸업
 1989년 3월~1993년 8월 서울대학교 컴퓨터공학과 대학원 박사 졸업
 1994년 3월~현재 호서대학교 컴퓨터공학부
 관심분야 : 영상처리, 영상압축, 영상분할, 컴퓨터비전, 웨이블릿, JPEG-2000, MPEG-4, MPEG-7, MPEG-21
 E-mail : hslee@office.hoseo.ac.kr

교신저자

이 호 석 336-795 충남 아산시 배방면 세출리 산29-1 호서대학교 컴퓨터공학부