

영상 부호화를 위한 DCT 계수 블록 크기 분류

강 경 인[†] · 김 정 일^{††} · 정 근 원[†] · 이 광 배^{††} · 김 현 옥^{††}

요 약

본 논문에서는 양자화 계수가 0이 되는 위치를 예측하여 축소된 영역내에서 DCT(Discrete Cosine Transform)를 수행하는 새로운 알고리즘을 제안한다. 이 제안한 알고리즘은 FDCT(Forward DCT)와 IDCT(Inverse DCT)의 계산량을 줄여 부호화 시간과 복호화 시간을 감소시킬뿐만 아니라, 허프만(huffman) 부호화시에도 각각의 블록에 대하여 분류된 블록 크기에 따라 각기 다른 수평 수직 지그재그 스캔을 수행함으로써 압축률을 증가시킨다. 기존의 영상 부호화 방법은 모든 블록에 대하여 똑같은 DCT 계산과 지그재그 스캔을 행한다. 그렇지만, 제안한 알고리즘은 부호화시에 분류된 블록 크기 밖의 양자화 계수에 대해 FDCT를 계산하는 대신에 0을 대입함으로써 FDCT 계산 시간을 줄인다. 또한, 복호화시에는 분류된 블록 크기내에 존재하는 역양자화 계수만을 가지고 IDCT를 수행함으로써 IDCT 계산 시간을 줄인다. 추가하여, 제안한 알고리즘은 분류된 블록 특성에 적합한 수평 수직 지그재그 스캔을 수행함으로써 Run-Length를 줄여서, 향상된 압축률을 제공한다. 한편, 제안한 알고리즘은 DCT에서 압축률과 화질면에서는 최적이지만 부호화 시간과 복호화 시간이 많이 걸리는 16*16 블록의 처리에도 적용되어질 수 있다. 또한, 실시간을 요구하는 동영상 부호화로 확장되어질 수 있다.

DCT Coefficient Block Size Classification for Image Coding

Kyeong In Kang[†] · Jeong Il Kim^{††} · Keun Won Jeong[†] · Kwang Bae Lee^{††} · Hyen Ug Kim^{††}

ABSTRACT

In this paper, we propose a new algorithm to perform DCT(Discrete Cosine Transform) within the area reduced by predicting position of quantization coefficients to be zero. This proposed algorithm not only decreases the encoding time and the decoding time by reducing computation amount of FDCT(Forward DCT) and IDCT(Inverse DCT) but also increases compression ratio by performing each different horizontal-vertical zig-zag scan according to the classified block size for each block on the huffman coding. Traditional image coding method performs the same DCT computation and zig-zag scan over all blocks, however this proposed algorithm reduces FDCT computation time by setting to zero instead of computing DCT for quantization coefficients outside classified block size on the encoding. Also, the algorithm reduces IDCT computation time by performing IDCT for only dequantization coefficients within classified block size on the decoding. In addition, the algorithm reduces Run-Length by carrying out horizontal-vertical zig-zag scan appropriate to the classified block characteristics, thus providing the improvement of the compression ratio. On the other hand, this proposed algorithm can be applied to 16*16 block processing in which the compression ratio and the image resolution

[†] 준 회 원: 명지대학교 전자공학과

^{††} 정 회 원: 명지대학교 전자공학과

논문접수: 1995년 9월 21일, 심사완료: 1997년 2월 17일



이 상 범

- 1974년 연세대학교 전자공학과 (공학사)
- 1978년 서울대학교 대학원 전자공학과(공학석사)
- 1986년 연세대학교 대학원 전자공학과(공학박사)
- 1984년 미국 IOWA대학교 컴

퓨터공학과 객원교수

1979년~1992년 단국대학교 전자공학과 교수

1993년~현재 단국대학교 컴퓨터공학과 교수

관심분야: 컴퓨터구조, 영상신호처리, GIS

arc optimal but the encoding time and the decoding time take long. Also, the algorithm can be extended to motion image coding requiring real time processing.

1. 서 론

정지영상 압축의 표준인 JPEG(Joint Photographic Experts Group)과 동영상 압축의 표준인 MPEG(Moving Picture Experts Group)에서 사용하는 DCT와 양자화 는 영상을 주파수 영역에서 해석하여 처리한다. JPEG 과 MPEG에서 주로 사용되는 압축 기법은 변환 부호 화 방식과 예측 부호화 방식, 엔트로피 부호화 방식 등이 있다[1, 2, 3, 4, 5]. 영상 압축은 영상들이 갖는 공간적, 시간적 중복성을 제거하거나 변환시키고 거 기에 통계적 특성을 적용함으로써 영상을 표현하는 데이타량을 줄이는 방법이다[6, 7]. 공간적 중복성이란 한 영상에서 인접한 픽셀들이 서로 비슷한 값을 가지고 있다는 것을 말하고, 시간적 중복성이란 대부분의 동영상에서 1초에 수십 장의 영상을 보내어 움직 임을 보여 주고 있으므로 역시 시간적으로 연속적인 두 영상은 비슷하다는 것을 의미한다[8].

영상에 대한 공간상의 중복성을 제거하는 방법으 로는 원래의 신호를 여러 개의 블럭으로 분할한 후 각각을 블럭 변환하여 변환 계수를 구한 뒤, 각 계수 들을 양자화하여 부호화 효율을 증가시키는 변환 부 호화 방법[3], 영상의 통계적 특성 및 인간의 시각적 특성을 고려하여 유한 갯수의 동일 차원 코드 벡터로 이루어진 코드북에서 한 코드 벡터를 찾아서 근사화 하는 벡터 양자화 방법[9], QMF(Quadrature Mirror Filter) बैं크를 사용하여 대상 신호의 주파수 대역을 N개의 부대역(subband) 신호로 분리한 후, 각 부대역 신호를 그 주파수 영역에서 인간의 시각 특성에 적합 한 PCM(Pulse Code Modulation) 또는 DPCM(Dif ferential PCM) 등을 적용하여 대역 압축을 하는 대역 분할 부호화 방법[10], 심볼의 확률적 분포에 따라 발 생 빈도가 높은 심볼은 적은 비트를 할당하고 발생빈 도가 낮은 심볼에 대해서는 많은 비트를 할당함으로 써 전체적으로 비트 발생율을 최소화하는 엔트로피 부호화 방법[3] 등이 있다.

변환 부호화, 벡터 양자화, 대역 분할 부호화, 엔트 로피 부호화가 영상에 대한 공간상의 중복성을 제거

하는데 이용되는 압축 방식인 반면에 예측 부호화 방 식은 영상에 대한 시간적인 중복성을 제거하는 압축 방식이다[1, 2, 9, 10, 11]. 단순히 같은 공간상 위치의 화소를 비교하여 차신호를 구하고, 차신호를 양자화 한 후 전송하는 단순 DPCM 방식이 일찍부터 영상의 시간상의 중복성을 제거하는 부호화 기법으로 사용 되어져 왔다. 그러나 이러한 단순 DPCM은 움직임이 많은 동영상 부호화시에는 예측오차의 증가에 따라 예측 성능이 떨어지기 때문에 움직임이 많은 동영상 을 낮은 비트율로 감축할때의 예측방법으로 이용하 기에는 부적절하다. 이러한 문제점을 보완하면서 예 측 성능을 향상시키기 위하여 이웃한 영상간의 물체 의 움직임을 검출한 후 보상해줌으로써 예측 성능을 크게 향상시킨 움직임 보상 예측 부호화 방식이 널리 사용되고 있다. 움직임 보상 예측 부호화 방식의 대 표적인 방법으로서 블럭당 움직임 벡터를 가지는 블럭 정합 알고리즘이 사용되고 있다[8, 10].

정지영상에 대한 표준으로 채택된 DCT 알고리즘 은 압축률과 화질면에서는 어느 정도 최적화된 것으 로 간주할 수 있으나, 이 DCT 알고리즘을 동영상으 로 확장하기 위해서는 실시간 처리에 대한 문제점을 고려하여야 한다. DCT 알고리즘의 방대한 계산시간 은 실시간 처리에 있어서 상당한 문제점을 제기한다. 이로 인하여 DCT 알고리즘의 최적으로 알려진 16*16 블럭크기로의 처리에 심각한 문제점을 야기시킨다. DCT 알고리즘의 계산량은 처리하는 블럭 크기에 비례하여 증가한다. 따라서, 이 논문에서 제안한 부호화 시간과 복호화 시간을 줄이는 알고리즘은 실시간 처리를 필요로 영상 압축 분야에 좋은 해결책이 될 수 있을 것이라 생각한다.

2. 제안한 알고리즘의 DCT 영역 블럭 크기 분 류 방법

변환 부호화 방식에는 푸리에 변환(Fourier Transform), KLT(Karhunen Loeve Transform), WHT(Walsh-Hadamard Transform), DCT 등이 있으며 KLT가 성

능면에서 최적이라 알려져 있다. 그러나, KLT는 커널(Kernel)이 영상에 따라 바뀌므로 커널 산출을 위하여 많은 계산이 필요하며, 전송시 각 커널을 부가 정보로 보내주어야 하므로 실제적으로 적용하는데 어려움이 있다. KLT의 많은 계산 과정과 부가 정보로 인하여 영상 부호화에는 KLT 다음으로 최적으로 알려져 있는 DCT를 사용하고 있다[12, 13]. 그러나, 이러한 DCT도 여전히 계산 과정이 많다는 문제점을 가지고 있다. 변환 부호화 방식에서는 보통 전체 영상을 8*8 또는 16*16의 블록으로 나누어서 변환을 행하는데 주변 화소들간의 상관 관계로 인하여 16*16 블록이 최적인 것으로 알려져 있다[14]. 그러나, 16*16 블록은 8*8 블록보다 DCT 계산 시간이 많이 걸려 실시간을 고려해야 하는 동영상으로의 확장에는 문제점이 있다.

영상에 적용하는 일반적인 2차원 FDCT와 IDCT의 수학적 모델은 식(1), (2)로 나타낸다[9, 15].

$$F(u, v) = \frac{4C(u)C(v)}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (2)$$

N은 블록 크기

u, v=0, 1, 2, ..., N-1

x, y=0, 1, 2, ..., N-1

C(w)는 w=0인 경우에만 $\frac{1}{\sqrt{2}}$ 이고 나머지는 1이다.

식(1)과 식(2)에서 보면 블록 크기인 N에 따라 DCT 계산 시간이 좌우된다. 8*8인 경우에는 하나의 주파수 영역 F(u, v)는 64개의 f(x, y)의 조합에 의해 이루어지고, 16*16인 경우는 256개의 f(x, y)의 조합에 의해 이루어진다. 이런 이유로 인해서 영상을 처리함에 있어서 8*8의 블록 크기로 처리하는 것보다 16*16 블록 크기로 처리하는 것이 많은 부호화시간과 복호화시간이 소요된다[16][17]. 따라서, DCT에 있어서 최적으로 알려진 16*16 블록 크기로 수행하는 것은 부호화 시간과 복호화시간에서 제한을 받게된다. 한편,

동영상 처리에 있어서는 8*8 블록 크기도 실시간에 관계해서 부호화 시간과 복호화 시간에서 제한을 받게 된다. 따라서, 부호화 시간과 복호화 시간을 줄이는 문제는 해결해야 할 중요한 과제가 된다.

기존의 JPEG에서 사용하는 8*8 블록 크기인 경우에 제안한 알고리즘은 (그림 1)과 같이 빗금친 부분의 양자화 계수만을 구하여 블록 크기를 정한다.

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)
(1, 0)							
(2, 0)							
(3, 0)							
(4, 0)							
(5, 0)							
(6, 0)							
(7, 0)							

(그림 1) 8*8 블록
(Fig. 1) 8*8 block

제안한 알고리즘이 부호화 시간과 복호화 시간을 줄이기 위해 각각의 블록에 대하여 블록 크기를 정하는 방식은 아래와 같다.

단계 1: 블록 크기가 8*8인 경우 수평축의 (0, 7)에 대하여 DCT를 계산하고 양자화 테이블로 나누어 양자화 계수 F(0,7)을 계산한다.

단계 2: 구한 F(0, 7)이 0이 아닌 경우에는 수평축의 블록 크기를 8로 정한다. F(0, 7)이 0인 경우에는 단계3을 수행한다.

단계 3: 수평축의 (0, 6)에 대해서 DCT를 계산하고 양자화 테이블로 나누어 양자화 계수 F(0, 6)을 계산한다.

단계 4: 구한 F(0, 6)이 0이 아닌 경우에는 수평축의 블록 크기를 7로 정한다. F(0, 6)이 0인 경우에는 단계5를 수행한다.

단계 5: 위의 단계들을 반복하면서 수평축의 (0, 0)까지 수행한다.

단계 6: 블록 크기가 8*8인 경우 수직축의 (7, 0)에 대해서 DCT를 계산하고 양자화 테이블로 나누어 양자화계수 F(7, 0)를 계산한다.

단계 7: 구한 F(7, 0)이 0이 아닌 경우에는 수직축의 블록 크기를 8로 정한다. F(7, 0)이 0인 경우에는 단계8을 수행한다.

단계 8: 수직축의 (6, 0)에 대해서 DCT를 계산하고 양자화 테이블로 나누어 양자화 계수 F(6, 0)를 계산한다.

단계 9: 구한 F(6, 0)이 0이 아닌 경우에는 수직축의 블록 크기를 7로 정한다. F(6, 0)이 0인 경우에는 단계10을 수행한다.

단계10: 위의 단계들을 반복하면서 수직축의 (0, 0)까지 수행한다.

단계11: 알고리즘을 마치면서 최종적으로 수평축과 수직축에 대한 블록 크기를 저장한다.

FDCT와 IDCT를 수행해야할 블록 크기를 줄임으로써 부호화 시간과 복호화 시간을 줄이는 제안한 알고리즘에서 사용하는 FDCT와 IDCT의 예를 아래에 나타내었다.

예1: 블록 크기 3*7인 경우 $\Rightarrow F(u, v) = \frac{4C(u)C(v)}{N^2}$

$$\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

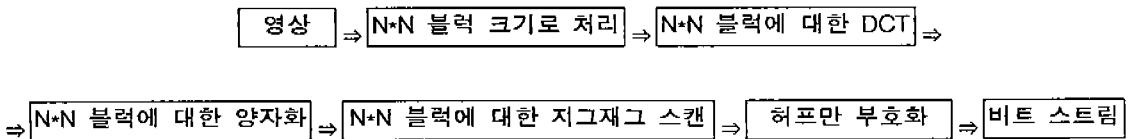
$$f(x, y) = \sum_{u=0}^2 \sum_{v=0}^6 C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

예2: 블록 크기 5*4인 경우 $\Rightarrow F(u, v) = \frac{4C(u)C(v)}{N^2}$

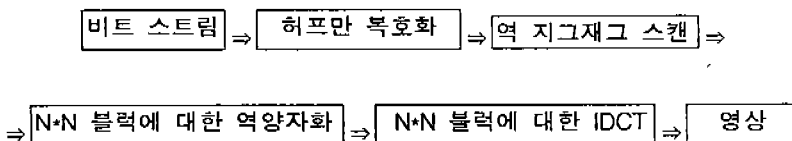
$$\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$f(x, y) = \sum_{u=0}^4 \sum_{v=0}^3 C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

분류된 블록 크기가 3*7인 예1을 보면 F(u, v)가 3*7 블록 크기에만 존재하므로 3*7에 대해서만 FDCT를 계산하고 8*8의 나머지에 대해서는 0을 대입하면 된다. 이 경우 FDCT 계산은 21번만 하게 된다. 그리고, 블록 크기를 정하기 위해 수행한 8번의 FDCT를 포함하여 총 29번의 FDCT 계산을 하면된다. 따라서 64-29=35번의 FDCT계산을 줄일 수 있다. 한편 IDCT의 경우 \sum 의 계산시에 u=2와 v=6으로 제한됨으로써 기존의 u=7과 v=7인 경우에 비해 복호화시에 계산 과정을 줄일 수 있다. 5*4의 블록 크기로 정해진 예2의 경우도 예1의 경우와 마찬가지로 블록 크기에 반비례해 부호화 시간과 복호화 시간을 줄일 수 있다.



(a) 부호화기



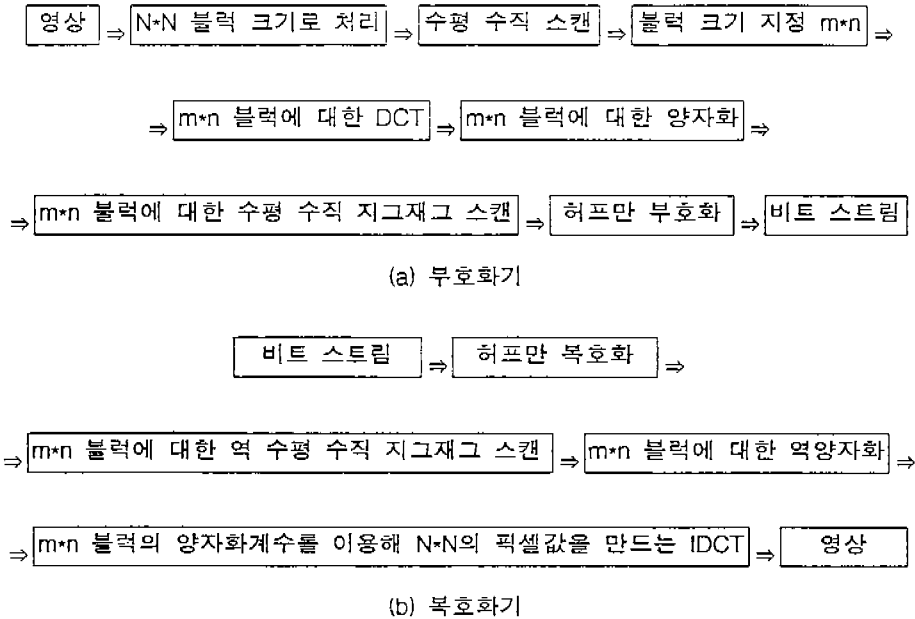
(b) 복호화기

(그림 2) 기존 알고리즘에 대한 블록도

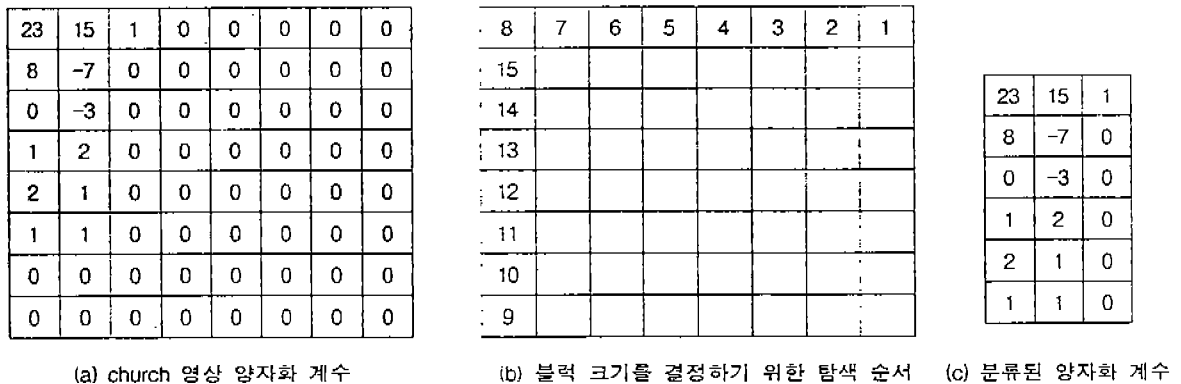
(a) 부호화기 (b) 복호화기

(Fig. 2) Block diagram of traditional algorithm

(a) Encoder (b) Decoder



(그림 3) 제안한 알고리즘에 대한 블록도
 (a) 부호화기 (b) 복호화기
 (Fig. 3) Block diagram of proposed algorithm
 (a) Encoder (b) Decoder



(그림 4) 8*8 블록에 대한 블록 크기 결정을 위한 탐색 순서
 (a) church 영상 양자화 계수
 (b) 블록 크기를 결정하기 위한 탐색 순서
 (c) 분류된 양자화 계수
 (Fig. 4) Scan order for block size determination of 8*8 block
 (a) Quantization coefficient of church image
 (b) Scan order for block size determination
 (c) Classified quantization coefficient

기존의 알고리즘에 대한 블록도를 (그림 2)에 나타내고[18], 제안한 알고리즘이 영상을 부호화하고 복호화하는 과정에 대한 개괄적인 블록도를 (그림 3)에 나타내었다.

제안한 알고리즘에 대한 블록도를 살펴보면, 기존의 방식과 다르게 먼저 블록 크기를 지정하는 알고리즘을 수행하여 블록 크기를 정한다. 그림 3의 블록도에서 8*8인 경우에 N은 8이되고 분류된 블록 크기가 수평축으로 4이고 수직축으로 6이면 m은 6이고 n은 4가 된다.

(그림 4)에는 8*8 블록에 대해서 블록 크기를 정하는 방법을 나타내었다. 8*8 블록의 양자화 계수에 대해서 DCT 계산을 해야하는 블록 크기를 각각의 블록 특성에 맞게 (그림 4)의 (b)와 같이 1에서부터 15까지 탐색하여 분류한다. 탐색하여 분류된 6*3의 블록에 대한 양자화 계수를 (그림 4)의 (c)에 나타내었다. (그림 4)의 (c)의 6*3 블록의 양자화 계수를 제외한 나머지의 양자화계수는 0이므로 DCT 계산을 할 필요가 없다. 따라서 부호화 시간은 감소하게 된다.

(그림 5)는 Lenna 영상의 8*8블록에 대한 양자화 계수를 나타내었다. 이 블록의 경우 양자화 계수가 0이 나오는 것을 제외하고 실제로 DCT를 수행해야하는 블록의 크기가 5*6인 것을 알 수 있다.

제안한 알고리즘에서는 영상을 8*8의 블록 크기로 처리하는 경우에 FDCT와 IDCT를 계산하는 종류가 분류된 블록 크기 1*1에서부터 8*8까지에 따라 다양한 종류가 존재하게 된다.

-25	-4	9	-4	4	-2
20	-3	-9	3	-1	0
1	-6	-2	2	0	0
-1	1	2	0	0	0
-1	1	0	-1	0	0

(b) 분류된 양자화계수

(그림 5) 8*8 블록에 대한 블록크기 결정
 (a) lenna 영상 양자화계수 (b)분류된 양자화계수
 (Fig. 5) Block size determination of 8*8 block
 (a) Quantization coefficient of lenna image
 (b) Classified quantization coefficient

3. 제안한 알고리즘에 대한 수평 수직 지그재그 스캔

기존의 방법은 주파수 영역에서의 FDCT를 계산한 결과인 FDCT 계수를 양자화 테이블에 의해 나눔으로써 나오는 양자화 계수는 왼쪽과 오른쪽으로 갈수록 0이 아닐 확률이 적고, 오른쪽과 아래쪽으로 갈수록 0일 확률이 커진다는 에너지 밀집성의 원리와 Run-Length에 기초하여 최적화된 것으로 알려져 있는 지그재그 스캔에 의하여 부호화된다. (그림 6)은 이러한 지그재그 스캔 순서를 나타내고 있다. 0에서부터 63까지 순서대로 탐색하여 허프만 부호화를 수행하게 된다.

-25	-4	9	-4	4	-2	0	0
20	-3	-9	3	-1	0	0	0
1	-6	-2	2	0	0	0	0
-1	1	2	0	0	0	0	0
-1	1	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

(a) lenna 영상 양자화계수

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

(그림 6) 기존의 지그재그 스캔 탐색 순서
 (Fig. 6) Traditional zig-zag scan search order

(그림 7)의 (a), (b), (c)는 제안한 알고리즘의 여러가지 블록 크기에 대한 수평 지그재그 스캔 탐색 순서를 나타낸다.

기존의 지그재그 스캔 방식은 (그림 8)의 (a), (b), (c), (d), (e), (f)에 나타난 블록들에 대해서 많은 문제점을 발생시킨다. 제안한 알고리즘은 이러한 블록들의 특성을 고려하여 블록 크기를 정하고 수평 수직 지그재그 스캔을 수행함으로써 Run-Length를 줄여 기존 방법의 문제점을 해결한다.

(그림 8)의 (a)의 경우를 살펴보면 기존의 지그재그 스캔을 하게되면 아래와 같다.

-23인 경우는 DC 계수이므로 제외하고 AC 계수에 대해서만 살펴보면, 스캔 결과는 -32 0 0 0 3 -2 0 -1 EOB(End of Block)이 되고 Run-Length가 23개가 된다.

(그림 8)의 (a)의 경우를 제안한 수직 수직 지그재그 스캔으로 하게되면, 먼저 (그림 9)과 같이 블록 크기를 결정하게 된다.

제안한 방식에 의해 DC 계수인 -23을 제외한 스캔 결과는 -32 3 -2 -3 0 0 -1 EOB이 되고 Run-Length는 2개가 된다. 위의 경우에 대한 고찰 결과 기존의 방식에 비해 제안한 방식의 성능이 뛰어난을 알 수 있다.

0	1	2	3	4	5	6	7
8	15	16	20	21	29	30	42
9	17	19	22	28	31	41	43
10	18	23	27	32	40	44	53
11	24	26	33	39	45	52	54
12	25	34	38	46	51	55	60
13	35	37	47	50	56	59	61
14	36	48	49	57	58	62	63

(a) 분류된 8*8 블록 크기

0	1	2	3	4
5	8	9	13	14
6	10	12	15	18
7	11	16	17	19

(b) 분류된 4*5 블록 크기

0	1	2
3	9	10
4	11	13
5	12	14
6	15	17
7	16	18
8	19	20

(c) 분류된 7*3블록 크기

(그림 7) 제안한 수평 수직 지그재그 스캔 탐색 순서

- (a) 분류된 8*8 블록 크기
- (b) 분류된 4*5 블록 크기
- (c) 분류된 7*3블록 크기

(Fig. 7) Proposed horizontal vertical zig-zag scan search order

- (a) Classified 8*8 block size
- (b) Classified 4*5 block size
- (c) Classified 7*3 block size

-23	-32	3	-2	-3	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	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) 수평 양자 화계수(couple 영상)

20	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) 수직 양자화 계수(cman 영상)

-53	-7	5	-2	1	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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c)수평 수직 양자화 계수(cronk 영상)

-14	-5	-4	-3	-2	-1	0	0
-1	1	0	0	0	0	0	0
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	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

-25	-4	9	-4	4	-2	1	0
20	-3	-9	3	-1	0	0	0
-1	1	2	0	0	0	0	0
-1	1	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

11	1	0	0	0	0	0	0
-17	-1	0	0	0	0	0	0
-28	0	0	0	0	0	0	0
-14	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

㉠ 주로 수평 수직 양자화 계수(pepper 영상) (e) 다양한 양자화 계수(lenna 영상) (f) 다양한 양자화 계수(woman 영상)

(그림 8) 기존의 지그재그 스캔 수행시 문제점이 발생하는 블록

- (a) 수평 양자 화계수(couple 영상)
- (b) 수직 양자화 계수(cman 영상)
- (c) 수평 수직 양자화 계수(cronk 영상)
- (d) 주로 수평 수직 양자화 계수(pepper 영상)
- (e) 다양한 양자화 계수(lenna 영상)
- (f) 다양한 양자화 계수(woman 영상)

(Fig. 8) Block generating problem by performing traditional zig-zag scan

- (a) Horizontal quantization coefficient (couple image)
- (b) Vertical quantization coefficient (coman image)
- (c) Horizontal vertical quantization coefficient (cronk image)
- (d) Manily horizontal vertical quantization coefficient(pepper image)
- (e) Various quantization coefficient (lenna image)
- (f) Various quantization coefficient (woman image)

-23	-32	3	-2	-3	0	0	-1
-----	-----	---	----	----	---	---	----

(그림 9) 제안한 방식에 의해 분류된 1*8 블록크기
(Fig. 9) 1*8 block size classified by proposed method

그림7의 각각의 경우에 대해 기존 방식과 제안한 방식의 성능비교를 <표 1>에 나타내었다.

Pepper	기존 방식	0
	제안한 방식	0
Lenna	기존 방식	10
	제안한 방식	2
Church	기존 방식	26
	제안한 방식	0

<표 1> 기존 알고리즘과 제안한 알고리즘의 Run-Length 비교

<Table 1> Run-Length comparison of traditional algorithm and proposed algorithm

영상	방식	Run-Length
Couple	기존 방식	23
	제안한 방식	2
Cman	기존 방식	15
	제안한 방식	0
Cronk	기존 방식	9
	제안한 방식	0

5. 실험 및 검토

본 논문에서는 256*256 크기의 256 그레이레벨 영상을 가지고 펜티엄60에서 시뮬레이션하였고, 사용한 양자화 테이블과 허프만 테이블은 JPEG 표준안에서 정해진 것을 사용하였다.[18]

실험은 블록 크기 분류와 각각의 블록 크기에 맞는 수평 수직 지그재그 스캔을 이용하여 압축률과 PSNR, 부호화시간, 복호화 시간을 계산하여 기존의 방식을 사용하는 JPEG 알고리즘과 비교하였다.

(그림 10), (그림 11), (그림 12), (그림 13)에서 (a)에는 256*256 원 영상을 (b)에는 기존의 방식에 의해 부호화한 후 복호화했을 때 생성된 영상을 (c)에는 제안

한 방식에 의해 부호화한 후 복호화했을 때 생성된 영상을 나타내었다.

<표 2>, <표 3>, <표 4>, <표 5>에는 baboon 영상,

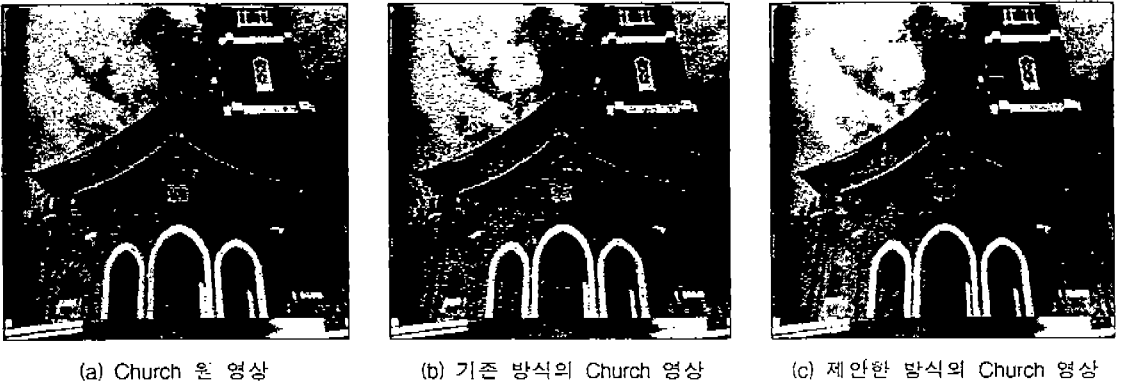


(그림 10) Baboon 영상에 대한 기존 방식과 제안한 방식의 비교

- (a) Baboon 원 영상 (b) 기존 방식의 Baboon 영상
- (c) 제안한 방식의 Baboon 영상

(Fig. 10) Comparison of traditional method and proposed method on baboon image

- (a) Baboon original image
- (b) Baboon image of traditional method
- (c) Baboon image of proposed method



(그림 11) Church 영상에 대한 기존 방식과 제안한 방식의 비교

- (a) Church 원 영상
- (b) 기존 방식의 Church 영상
- (c) 제안한 방식의 Church 영상

(Fig. 11) Comparison of traditional method and proposed method on church image

- (a) Church original image
- (b) Church image of traditional method
- (c) Church image of proposed method



(a) Lenna 원 영상

(b) 기존 방식의 Lenna 영상

(c) 제안한 방식의 Lenna 영상

(그림 12) Lenna 영상에 대한 기존 방식과 제안한 방식의 비교

(a) Lenna 원 영상

(b) 기존 방식의 Lenna 영상

(c) 제안한 방식의 Lenna 영상

(Fig. 12) Comparison of traditional method and proposed method on lenna image

(a) Lenna original image

(b) Lenna image of traditional method

(c) Lenna image of proposed method



(a) Pepper 원 영상

(b) 기존 방식의 Pepper 영상

(c) 제안한 방식의 Pepper 영상

(그림 13) Pepper 영상에 대한 기존 방식과 제안한 방식의 비교

(a) Pepper 원 영상

(b) 기존 방식의 Pepper 영상

(c) 제안한 방식의 Pepper 영상

(Fig. 13) Comparison of traditional method and proposed method on pepper image

(a) Pepper original image

(b) Pepper image of traditional method

(c) Pepper image of proposed method

〈표 2〉 Baboon 영상에 대한 분류된 블럭 크기
 〈Table 2〉 Classified block size of baboon image

y축좌표 (수직크기)	x축좌표 (수평크기)	블럭 갯수	y축좌표 (수직크기)	x축 좌표 (수평크기)	블럭 갯수	y축 좌표 (수직크기)	x축 좌표 (수평크기)	블럭 갯수
0	0	1	2	6	2	5	4	104
0	1	6	2	7	2	5	5	40
0	2	5	3	0	0	5	6	35
0	3	6	3	1	2	5	7	22
0	4	4	3	2	14	6	0	0
0	5	0	3	3	23	6	1	6
0	6	0	3	4	35	6	2	23
0	7	1	3	5	9	6	3	50
1	0	0	3	6	5	6	4	82
1	1	5	3	7	6	6	5	41
1	2	8	4	0	1	6	6	15
1	3	16	4	1	4	6	7	17
1	4	8	4	2	27	7	0	0
1	5	2	4	3	35	7	1	0
1	6	0	4	4	54	7	2	8
1	7	1	4	5	31	7	3	20
2	0	0	4	6	21	7	4	34
2	1	3	4	7	18	7	5	11
2	2	9	5	0	3	7	6	8
2	3	9	5	1	8	7	7	5
2	4	14	5	2	38			
2	5	6	5	3	61			

〈표 3〉 Church 영상에 대한 분류된 블럭 크기
 〈Table 3〉 Classified size of church image

y축좌표 (수직크기)	x축좌표 (수평크기)	블럭 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블럭 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블럭 갯수
0	0	53	2	6	5	5	4	44
0	1	14	2	7	6	5	5	12
0	2	16	3	0	6	5	6	4
0	3	13	3	1	26	5	7	6
0	4	12	3	2	46	6	0	1
0	5	3	3	3	55	6	1	5
0	6	0	3	4	50	6	2	8
0	7	6	3	5	16	6	3	6
1	0	21	3	6	6	6	4	7
1	1	31	3	7	2	6	5	2
1	2	28	4	0	2	6	6	2
1	3	35	4	1	14	6	7	1
1	4	30	4	2	48	7	0	0
1	5	13	4	3	39	7	1	1
1	6	3	4	4	36	7	2	3
1	7	7	4	5	11	7	3	3
2	0	5	4	6	3	7	4	4
2	1	19	4	7	4	7	5	3
2	2	41	5	0	5	7	6	0
2	3	49	5	1	8	7	7	0
2	4	37	5	2	32			
2	5	18	5	3	38			

〈표 4〉 Lenna 영상에 대한 분류된 블록 크기

〈Table 4〉 Classified block size of lenna image

y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수
0	0	17	2	6	6	5	4	19
0	1	50	2	7	6	5	5	11
0	2	34	3	0	3	5	6	5
0	3	38	3	1	13	5	7	1
0	4	10	3	2	31	6	0	0
0	5	1	3	3	51	6	1	0
0	6	8	3	4	49	6	2	1
0	7	3	3	5	27	6	3	0
1	0	34	3	6	12	6	4	2
1	1	74	3	7	6	6	5	2
1	2	70	4	0	0	6	6	1
1	3	53	4	1	5	6	7	0
1	4	32	4	2	19	7	0	0
1	5	19	4	3	38	7	1	0
1	6	10	4	4	40	7	2	0
1	7	7	4	5	16	7	3	0
2	0	14	4	6	12	7	4	0
2	1	28	4	7	1	7	5	0
2	2	47	5	0	0	7	6	0
2	3	33	5	1	2	7	7	0
2	4	33	5	2	6			
2	5	11	5	3	13			

〈표 5〉 Pepper 영상에 대한 분류된 블록 크기

〈Table 5〉 Classified size of pepper image

y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수	y축좌표 (수직크기)	x축좌표 (수평크기)	블록 갯수
0	0	17	2	6	9	5	4	21
0	1	44	2	7	1	5	5	9
0	2	41	3	0	8	5	6	4
0	3	13	3	1	26	5	7	0
0	4	4	3	2	59	6	0	0
0	5	0	3	3	68	6	1	2
0	6	0	3	4	47	6	2	12
0	7	1	3	5	14	6	3	4
1	0	24	3	6	12	6	4	9
1	1	55	3	7	0	6	5	3
1	2	75	4	0	3	6	6	1
1	3	48	4	1	10	6	7	0
1	4	19	4	2	30	7	0	0
1	5	6	4	3	46	7	1	2
1	6	7	4	4	40	7	2	0
1	7	4	4	5	7	7	3	2
2	0	11	4	6	2	7	4	2
2	1	29	4	7	1	7	5	0
2	2	62	5	0	2	7	6	0
2	3	33	5	1	7	7	7	0
2	4	20	5	2	12			
2	5	11	5	3	25			

church 영상, lenna 영상, pepper 영상에 대한 분류된 64가지 종류의 블럭 크기에 대한 블럭 갯수를 나타내었다. <표 2>, <표 3>, <표 4>, <표 5>에서 y축 좌표가 0이고 x축 좌표가 0인 것은 1*1 블럭 크기로 분류되었다는 것을 의미하고, y축 좌표가 7이고 x축 좌표가 7인 것은 8*8 블럭 크기로 분류되었다는 것을 의미한다. 표에서 살펴보면, baboon 영상에 대한 분류된 블럭 크기는 lenna 영상에 대해 분류된 블럭 크기보다 주로 크기가 큰쪽에 많이 분포한다. 이것은 baboon 영상이 lenna 영상에 비해 복잡한 영상임을 의미한다. 제안한 방식에서의 baboon 영상의 부호화 시간과 복호화 시간은 lenna 영상에 비해 거의 2배가 소요된다.

<표 6>에는 기존 알고리즘과 제안한 알고리즘의 성능 평가를 위하여 psnr, bpp, 압축률, 부호화 시간, 복호화 시간을 13개의 표준 영상에 대해 비교하였다.

6. 결 론

본 논문에서는 각각의 블럭에 대해 양자화 계수가 0이 되는 위치를 예측하여 DCT를 계산해야하는 블럭 크기를 줄임으로써 부호화 시간과 복호화 시간을 감소시키고, 분류된 블럭의 크기에 따라 수평 수직 지그재그 스캔을 적용함으로써 거의 화질을 저하시키지 않고 압축률을 향상시킬 수 있는 알고리즘을 제

<표 6> 기존 알고리즘과 제안한 알고리즘의 화질, 압축률, 부호화 시간과 복호화 시간 비교
 <Table 6> Resolution, compression ratio encoding time and decoding time of traditional algorithm and proposed algorithm

영상	방식	psnr [dB]	bpp(bit per pixel)	압축률	부호화 시간 [sec]	복호화 시간 [sec]
Baboon	기존 방식	26.97	1.53	5.21	31.14	33.07
	제안한 방식	26.16	1.27	6.29	16.42	15.21
Church	기존 방식	33.10	0.92	8.74	31.14	33.01
	제안한 방식	32.65	0.78	10.28	11.14	8.01
Cman	기존 방식	31.66	0.90	8.90	30.82	33.73
	제안한 방식	30.48	0.77	10.33	11.86	8.35
Couple	기존 방식	35.42	0.75	10.67	31.14	33.07
	제안한 방식	34.85	0.63	12.73	10.16	6.26
Cronk	기존 방식	39.20	0.58	13.70	30.71	32.52
	제안한 방식	38.65	0.51	15.60	9.33	5.54
Girl	기존 방식	34.72	0.77	10.41	30.88	33.73
	제안한 방식	34.21	0.64	12.52	8.93	5.09
House	기존 방식	37.72	0.59	13.66	30.71	32.57
	제안한 방식	37.02	0.51	15.83	9.40	5.22
Jaguar	기존 방식	28.12	1.51	5.30	31.14	33.07
	제안한 방식	27.38	1.27	6.28	15.82	13.62
Lenna	기존 방식	32.84	0.97	8.36	30.71	32.57
	제안한 방식	31.58	0.77	10.35	10.27	7.03
Miss	기존 방식	39.15	0.38	21.19	30.88	33.73
	제안한 방식	38.71	0.34	23.40	8.29	2.99
Pepper	기존 방식	33.97	0.90	8.92	31.14	33.07
	제안한 방식	33.27	0.76	10.53	11.53	8.23
Sale	기존 방식	34.20	0.89	9.03	30.88	33.73
	제안한 방식	33.38	0.75	10.65	11.75	8.68
Woman	기존 방식	33.42	0.86	9.30	30.71	32.57
	제안한 방식	32.72	0.72	11.12	10.99	7.75

안하였다. 제안한 알고리즘은 부호화 시간과 복호화 시간의 관점에서 살펴보면 대략 2배에서 5배의 성능 향상을 얻을 수 있었고, 압축률면에서도 상당한 성능 향상을 얻을 수 있었다. 화질면에서는 거의 원 영상과 구분할 수 없는 영상을 얻을 수 있었다. 제안한 알고리즘은 부호화 시간과 복호화 시간을 줄임으로써 DCT의 최적으로 알려진 16×16 에도 적용될 수 있을 것이라 평가된다. 또한 제안한 알고리즘은 동영상으로의 확장도 용이하리라 생각된다. 동영상에서는 실시간으로 영상을 압축하고 복원해야하므로 제안한 알고리즘의 시간 감소 효과와 압축률 향상은 이에 쉽게 적용되어질 수 있다. 앞으로의 연구 과제로는 DCT의 최적인 16×16 에 대한 적용을 위하여 최적화된 16×16 양자화 테이블과 허프만 테이블에 대한 연구가 필요하리라 생각된다. 또한 허프만 테이블은 16×16 블록 전체의 양자화계수에 대한 확률적 분포를 계산해서 만드는 것보다 제안한 알고리즘에서 분류하는 블록 크기내의 양자화 계수에 대해서만 확률적 분포를 계산해서 허프만 테이블을 만드는 연구가 수행되어야 하겠다.

참 고 문 헌

- [1] T.J. Lynch, Data Compression, Life Learning Publication, 1985.
- [2] A.K. Jain, "Image Data Compression: A Review," IEEE Proc., Vol.69, pp.349-389, 1981.
- [3] JPEG digital compression and coding of continuous-tone still image, Draft ISO 10918, 1991.
- [4] MPEG1 "Simulation Model Three," ISO/IEC, July 1990.
- [5] Coding of moving picture and associated audio, ISO/IEC JTC1/SC29, November 1993.
- [6] A.K. Jain, Fundamentals of digital image processing, Prentice Hall, 1989.
- [7] Didier Le Gall, MPEG: A video compression standard of multimedia applications, Commun ACM, vol. 34, no. 4, pp. 47-58, April 1991.
- [8] M. Ibrahim Sezan and Reginald L. Lagendijk, Motion Analysis and Image Sequence Processing, pp. 153-187, Kluwer Academic Publishers, 1993.
- [9] K.R.Rao, P.Yip, Discrete Cosine Transform, Academic Press, 1990.
- [10] J. W. Woods, Subband image coding, Kluwer Academic Publisher, 1990.
- [11] Allen Gersho and Robert M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.
- [12] P.A. Wintz, "Transform Picture Coding," IEEE Proc., Vol.60, pp.809-820, 1972.
- [13] R.J. Clark, "Relation between the Karhuned-Loeve and Cosine Transforms," IEEE Proc. 128, pp.359-360, 1981.
- [14] R.J. Clark, Transform Coding of Images, Academic Press, 1985.
- [15] N.Ahmed, T.Nartarajan, and K.R.Rao, "Discrete Cosine Transform," IEEE Trans. Comput., Vol. C-23, January, pp.90-93, 1974.
- [16] Rafael C. Gonzalez and Paul Wintz, Digital Image Processing, Akkison-Wesley, 1987.
- [17] Azrial Rosenfeld and Avinash C. kak, Digital Picture Processing, Academic Press, 1980.
- [18] William B. Pennebaker and Joan L. Mitchell, STILL IMAGE DATA COMPRESSION STANDARD, Van Nostrand Reinhold, pp.509-513, 598, 1993.



강 경 인

1994년 2월 명지대학교 전자공학과 졸업(공학사)
 1996년 2월 명지대학교 전자공학과 졸업(공학석사)
 1996년 3월~현재 명지대학교 전자공학과 박사학위과정 재학중

관심 분야: 영상 압축 및 통신, 음성 압축 및 통신, 영상 인식



김 정 일

1989년 2월 명지대학교 전자공학과 졸업(공학사)
1994년 8월 명지대학교 전자공학과 졸업(공학석사)
1995년 3월~현재 명지대학교 전자공학과 박사학위 과정 재학중

관심 분야: 영상 압축 및 통신, 음성 압축 및 통신, 영상 인식



김 현 옥

1978년 2월 고려대학교 전자공학과 졸업(공학사)
1980년 2월 고려대학교 전자공학과 졸업(공학석사)
1987년 2월 고려대학교 전자공학과 졸업(공학박사)
1980년 3월~1981년 2월 동양공

업전문대학 전자과 전임강사
1981년 3월~1988년 1월 명지대학교 전자공학과 교수
1988년 2월~1990년 1월 Dept. of Computer Science of Arizona State University Adjunct Faculty

1990년 3월~현재 명지대학교 전자공학과 교수
관심분야: 병렬처리 컴퓨터 시스템, 고장 감내 시스템, 영상 통신, 음성 통신



정 근 원

1991년 2월 명지대학교 전자공학과 졸업(공학사)
1993년 8월 명지대학교 전자공학과 졸업(공학석사)
1995년 8월~현재 명지대학교 전자공학과 박사학위 과정 재학중

관심 분야: 영상 통신, 음성 통신, 병렬처리 컴퓨터 시스템



이 광 배

1979년 2월 고려대학교 전자공학과 졸업(공학사)
1981년 2월 고려대학교 전자공학과 졸업(공학석사)
1981년 3월~1982년 3월 삼성 반도체연구소
1982년 3월~1983년 4월 금성 연구소

1984년~1986년 Univ. of Southern California, Computer Engineering 전공(공학석사)

1986년~1991년 Arizona state Univ., Electrical Engineering 전공(공학박사)

1992년~현재 명지대학교 전자공학과 조교수

관심 분야: 멀티미디어, 병렬처리 컴퓨터 시스템, 인공지능, 고장 감내 시스템