

논문 2013-50-12-11

공용 컬러맵(코드북)을 이용한 고품질 인덱스 영상의 구현 (Implementation of High Quality Indexed Image utilizing Common Color Map(Codebook))

최 용 수*, 이 달 호**

(YongSoo Choi[Ⓢ] and DalHo Lee)

요 약

현재의 IT 환경에서 영상 또는 영상처리 기술은 많은 부분에서 응용이 되고 있고 또한 매우 중요하다. 본 논문에서는 벡터 양자화 방법을 통해 원본 BMP영상을 인덱스 영상과 제안하는 codebook으로 재구성하고 동일한 픽셀 깊이 내에서 고품질의 영상을 디스플레이 할 수 있는 방법을 제안하였다. 즉, 2^n 개의 컬러 코드북(맵)를 가지는 영상에 공용 맵 기법을 적용함으로써 2^{n+1} 개의 컬러 코드북 정도의 화질로 표현이 되는 방법을 제안한다. 제안된 방법을 통해 얻어진 출력영상을 원본 BMP영상과 비교했을 때 JPEG과 같은 8 bit 인덱스 영상의 출력 시 보다 약 2 dB 정도 높은 PSNR을 제공해주며 이는 9 bit 인덱스를 사용했을 때의 화질과 거의 비슷함을 알 수 있었다.

Abstract

Image and it's processing techniques are widely applied and very important in the recent IT environment. In this paper, we try to reconstruct original BMP(Bitmap) image into indexed image and codebook utilizing vector quantization and represent high quality image only with same pixel depth of previous indexed image like JPEG etc. That is, By adopting common map method onto index image with 2^n color codebook, image can be represented as high quality as 2^{n+1} color codebook. When proposed output image is compared with original BMP image, it provides as much around 2dB as higher PSNR than conventional 8 bit index image(normal JPEG). Furthermore, this improvement(2 dB higher PSNR) could be provided when using the 9 bit indexed image.

Keywords : 벡터 양자화, 공용 맵, 영상 압축, 컬러 표현, 영상 처리

I. 서 론

최근 컴퓨터상에서 보여 지는 영상은 파일의 크기가 매우 크고 화질이 뛰어나다. 이는 하드웨어의 고용량화,

영상표현 기술의 발전으로 가능해졌다고 할 수 있다. 하지만 하드웨어 메모리는 주기적으로 그 한계를 드러내고 사용자들은 메모리의 업그레이드를 실시해야 한다. 압축은 동일한 크기의 데이터를 더 작은 공간의 메모리에 저장할 수 있도록 해 주는 기술이다. 특히, 정지 영상압축에 있어서는 양자화와 Entropy 기술을 이용한 JPEG Compression, 인덱스를 이용한 TIFF, GIF 등과 같은 포맷들이 개발되어왔다. 그 이외에도 흑백(픽셀이 0과 1로 표현되는)영상을 위한 JBIG과 같은 특화된 압축 포맷들도 존재한다^[1~2, 6~7, 11, 15, 18].

본 논문에서는 위에서 보인 압축포맷의 변형으로 원본 영상인 BMP(Bit MaP File Format, 일반적으로 원

* 정회원, 성결대학교 교양교직부(멀티미디어)
(Division of Liberal Arts & Teaching, Sungkyul University)

** 정회원, 가천대학교 IT대학 전자공학과
(Dept. of Electronic Engineering, Gachon University)

Ⓢ Corresponding Author(E-mail: ciechoi@sungkyul.ac.kr)

※ “이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입(No. 2013R1A1A1013410)”

접수일자: 2013년11월1일, 수정완료일: 2013년11월21일

도우즈에서 디지털 이미지를 저장할 때 사용영상을 JPEG과 같이 인덱스를 가진 압축영상으로 저장할 때 일반적인 Index 영상과 거의 동일한 BPP(Bits per Pixel)를 가지지만 더욱 풍부한 화질을 보일 수 있는 방법을 소개한다. 즉, 고화질의 영상을 논문에서 개발한 공용 컬러 맵을 이용하여 압축 및 저장하는 방법을 제안한다.

즉, 제안하는 알고리즘은 인덱스를 가진 영상을 표현함에 있어 더 적은 메모리의 사용으로 더 좋은 화질의 영상을 저장하고 표현하고자 한다. 제안된 방법은 2^n 크기의 코드북을 가지는 색인 영상의 파일 사이즈를 유지한 채 2^{n+1} 크기의 코드북을 가지는 색인 영상에 상응하는 화질을 제공한다. 여기서 n은 픽셀의 깊이(Depth)를 말한다.

II. Vector Quantization의 이해

풍부한 컬러를 담고 있는 Graphic Image(컴퓨터에 의한 디지털 영상)은 하드웨어적인 한계 또는 사용자의 요구에 따라 더 적은 색상으로 표현되기도 한다. 예를 들어, RGB 각각이 256가지 색상을 가지는 BMP영상이 때로는 8가지(Gray) 또는 2가지(Black & White) 색상으로 표현되거나 출력되기도 한다. 이때 원본이 가진 색상을 8가지 또는 2가지의 색상으로 매핑하는 과정을 컬러 양자화 또는 Image Vector Quantization이라 한다 [3-5,8,16].

1. Image Vector Quantization^[9,10]

본 절에서 소개하는 방법은 실제 Gray 영상을 색인 영상으로 압축할 때 사용하는 Image Vector Quantization 응용이며 그 순서는 다음과 같다.

- ① 원 영상에 대해 전체 영역을 양자화
- ② 양자화된 영상을 n x n 크기의 블록들로 분할
- ③ 각 블록들을 대표 벡터라 하고 대표벡터의 집합을 코드북이라 정함
- ④ 원영상의 블록과 코드북 벡터 각각들에 대한 거리를 계산하여 가장 적은 왜곡(거리)을 가지는 벡터의 색인을 해당 원영상의 Index 값으로 저장

인코딩 시 선택된 벡터(블럭)는 코드북 2와 가장 근

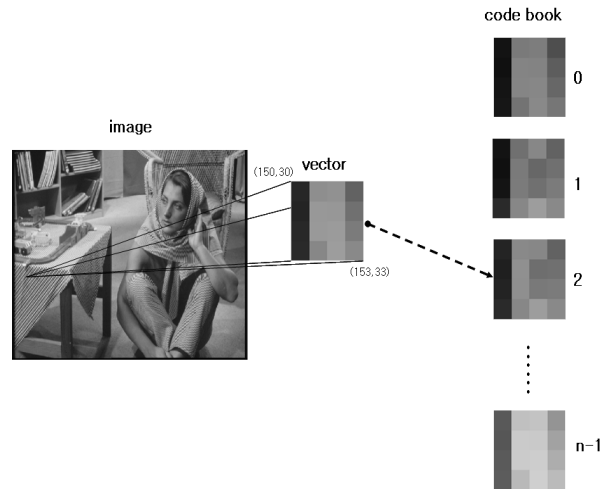


그림 1. 영상 벡터 양자화의 인코딩
Fig. 1. Image Vector Quantization의 Encoding

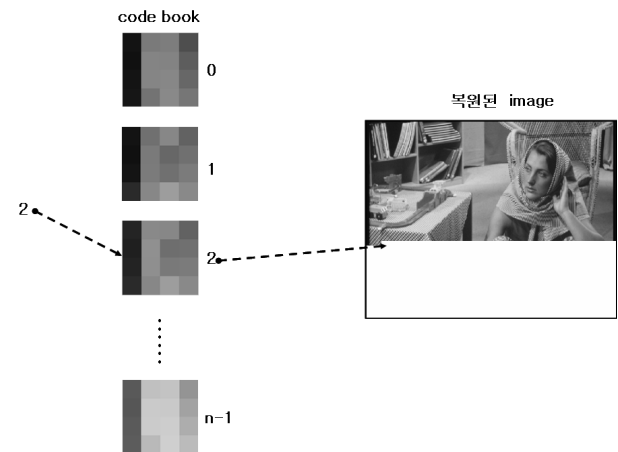


그림 2. 영상 벡터 양자화의 디코딩
Fig. 2. Image Vector Quantization의 Decoding

접하므로(Euclidean Distance가 짧음) 해당 벡터는 색인 2를 가지게 된다. 이러한 Index 기반의 영상표현은 경계색을 표현하기에는 부족하고 각 블록들 간에 Blockiness가 발생하는 단점을 가지고 있다.

디코딩 시 해당 블록을 복원하기 위해 색인 2에 해당하는 코드북의 값을 해당 위치에 대입함으로써 일정오류를 지닌 원 영상이 복원된다. 이와 같은 벡터 양자화는 코드북과의 유사성만을 색인 선택의 기준으로 삼으므로 일반적인 양자화 모델에서는 블록기반보다는 전체 색상에 대한 양자화를 실시한다.

2. Color Vector(codebook) Quantization^[6~9]

컬러의 양자화는 아래와 같은 4가지 방법으로 나눌 수 있다.

① 단일 양자화

- 원래의 각 색상은 균일하게 분포된 색상들 중에서 가장 근접한 기준 색상으로 매핑 된다. 애니메이션과 같은 출력에서 유용하게 쓰이는 방법이다.

② 중간 값 자르기에 의한 양자화

- 컬러 색상의 경우 이 알고리즘은 두 단계로 동작한다. 첫째는 컬러공간을 그룹으로 나누는 것이며 두 번째는 그룹이 새로운 컬러공간으로 할당되어지는 것이다. 최초의 컬러공간 R, G 그리고 B 색상 축을 하나의 사각형으로 보고 가장 긴 축을 반으로 자르면 또 하나의 사각형이 생긴다. 각각의 사각형은 가장 긴 축을 반으로 나눔으로서 두 개로 나뉘어진다. 이러한 동작은 사각형이 원하는 컬러 수 만큼 될 때 까지 반복된다. 마지막으로 각 사각형의 원소들의 중심점 또는 중간 값이 대표 색상으로 할당되어 진다.

③ Histogram에 의한 양자화

- 영상의 색상 중 가장 많이 사용된 색상에 대해 변화를 주지 않고 보존하고 나머지 색상들은 사용 빈도에 따라 근접한 컬러들로 할당되어 지는 방법이다. 이 방법은 원본의 색상 특성을 대체적으로 많이 보존하는 방법이다.

④ 편차최소화 양자화

통계적인 접근 방법으로 원래의 색상과 매핑 되는 대표 색상으로의 거리인 편차를 계산하여 그 변위를 최소화 하는 대표 색상을 계산하는 방법으로 앞에서 보인 단일 양자화 또는 중간 값 자르기에 의한 방법보다 원본 색상에 근접한 대표 색상들을 계산하는 것이 가능하다. 이 방법은 K-Means 또는 LBG(Linde-Buzo-Gray)와 같은 군집알고리즘을 이용하여 대표 색상을 생성해 낸다^[12~14].

III. 제안하는 공용 맵

제안하는 방법의 개념은 그림 3과 같이 개략적으로 도시할 수 있다. 원본(비압축) 영상을 9 bit 인덱스 영상으로 만든 후 양자화를 통해 8 bit 인덱스 영상과 위치 맵으로 변환 한 후 저장되며 영상 표현 시에는 저장된 8 bit 인덱스와 위치 맵을 이용하여 가상적으로 9 bit 깊이의 풍부한 컬러정보를 가지는 영상을 출력하는 것이다.

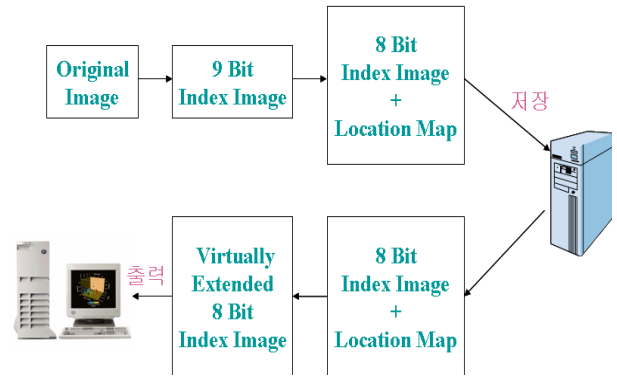


그림 3. 제안하는 색인 영상 압축 과정
Fig. 3. Proposed Indexed Image Compression

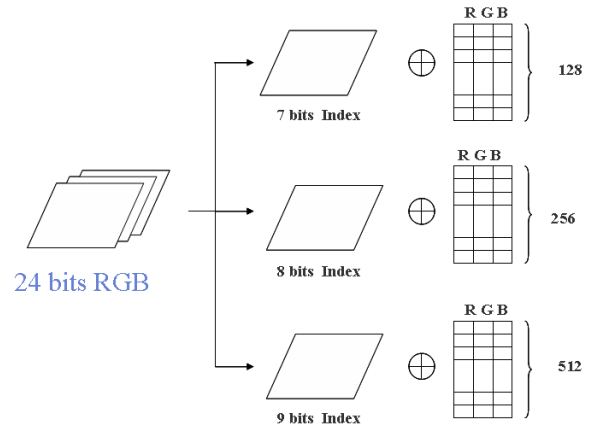


그림 4. 일반적인 색인 영상의 구성
Fig. 4. General Indexed Image Structure.

일반적인 디지털 영상은 그림 4와 같이 24 Bits RGB(Red, Green, Blue) 값으로 표현되어 있으며 256크기의 RGB 코드북을 가진 8 bit Index로 압축 표현된다. 더욱 높은 화질(512 코드북)을 위해 하나의 픽셀 당 9 bit를 쓸 경우 화질은 뛰어나지만 물리적으로 9 bit를 쓰는 것은 메모리의 소비가 많다. 이러한 문제를 해결하기 위해 본 논문에서는 아래 그림 5와 같은 공용 맵 구조를 제안한다.

24 Bits RGB영상의 256 코드북은 2장 2절 ④방법을 통해 480 코드북으로 양자화 되고 코드북의 색인을 표현하기 위해 9 bit가 필요하다. 하지만 앞서서도 언급한 바와 같이 8 bit(256가지) 색인을 사용하기 위해 공용 맵을 쓰게 된다.

공용 맵을 만드는 과정은 다음과 같다. 먼저 생성된 코드북은 영상의 히스토그램 분석에 의해 내림차순으로 재배열된다. 히스토그램 재배열은 자주 이용되는 컬러 code는 코드북의 상위에 사용 빈도가 낮은 컬러 code는

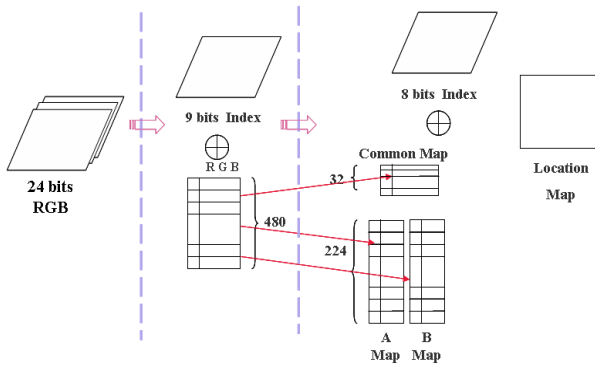


그림 5. 제안된 색인 영상의 구성
Fig. 5. Proposed Indexed Image's Structure.

코드북의 하위에 배치되도록 만든다. 재배열된 코드북의 상위 32개는 공용 맵으로 지정되며, 차 순위 224개는 A Map, 나머지 224개는 B Map으로 지정된다.

위와 같이 공용 맵과 Map A, B를 각각 만든 후 다음과 같은 3장 1절의 과정을 거쳐 9 Bits index를 8 Bits index로 변경(Encoding)한다.

색인영상 생성 및 복원에서 사용되는 약어들의 정의는 다음과 같다.

- $Im(BMP Image)$: 원본 영상
- $Idx(Indexed Image)$: 색인 영상
- $LM(Location Map)$: 위치 지도
- i, j 는 각각 영상에서의 픽셀 위치(행과 열)

1. 제안된 인코딩을 이용한 색인 영상의 생성

- $Im(i, j)$ 의 색인이 0~31이면 공용 맵을 참조하므로 $LM(i, j)$ 에는 0을 할당하고 색인을 $Idx(i, j)$ 에 저장.
- $Im(i, j)$ 의 색인이 32~255이면 맵 A를 참조하므로 $LM(i, j)$ 에는 0을 할당하고 색인을 $Idx(i, j)$ 에 저장.
- $Im(i, j)$ 의 색인이 256~479이면 맵 B를 참조하므로 $LM(i, j)$ 에는 1을 할당하며 해당 픽셀의 저장 색인을 $Idx(i, j)$ 에 저장하며 그 색인은 다음과 같이 구한다(저장되는 색인 값= $Im(i, j)$ 의 색인 - 224).

위의 동작을 원 영상의 모든 픽셀을 읽을 때까지 반복한다. 이와 같은 과정을 거침으로써 480 코드북의 화질을 보존한 채 저장되는 픽셀의 색인은 8 bit (0~255)를 유지하게 된다.

2. 제안된 Decoding을 이용한 영상의 복원

저장 공간으로부터 $Idx, LM, 480$ 코드북을 읽어온다.

- $Idx(i, j)$ 의 색인을 읽고 값이 0~31사이라면 색인을 그대로 보존.
- $Idx(i, j)$ 의 색인이 32~255인 경우 $LM(i, j)$ 의 Flag를 읽어 0인 경우 색인을 그대로 보존하고 1인 경우 현재 색인에 대응하는 Map B의 색인을 복원(즉, 복원 색인 값= $Idx(i, j)$ 의 색인 + 224)

위의 과정을 Idx 영상의 모든 픽셀을 읽을 때까지 반복하고 해당 색인에 대한 480 코드북의 컬러 벡터를 디스플레이 한다.

위의 Decoding 과정을 거친 복원된 영상은 8 bit(256 코드북) 영상 깊이를 가지지만 512 코드북만큼의 PSNR를 보여준다. 즉, 원래의 BMP이미지에 더욱 근사한 출력 영상을 만들어냄을 실험을 통해 알 수 있다.

IV. 구현 결과 및 평가

영상을 이용한 실제 실험은 Windows 운영체제에서 MATLAB 프로그램을 이용하여 이루어졌으며 실험영상은 영상처리에서 대표적으로 사용하는 LENA 512X512 영상을 사용하였다^[17].

그림 6의 원본 영상을 각각 128, 256, 512, 480 크기의 코드북으로 구성된 인덱스 영상으로 표현한 경우 그림 7과 같은 출력영상을 각각 얻을 수 있다. 또한 각 영상에 사용된 색상의 차이를 보이기 위해 각 그림의 오른쪽에 사용된 컬러바를 표시하였다. 실제 모니터에서 정상적인 크기로 볼 경우 그 차이가 확연하지만 논문에

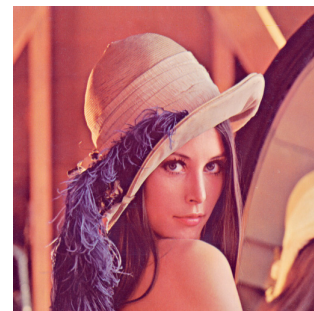


그림 6. 원본 Lenna (512X512)
Fig. 6. Original Image(Lenna)



그림 7. 128, 256, 512, 480 코드북을 이용한 색인영상
Fig. 7. Indexed Lena Image with 128, 256, 512, 480 Color Map respectively(From Top to Bottom).

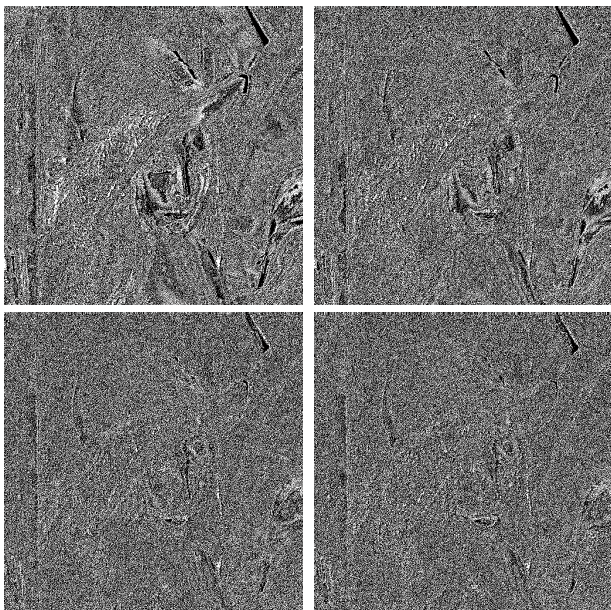


그림 8. 128, 256, 512, 480 코드북을 사용했을 때의 출력영상과 원본과의 왜곡
Fig. 8. Distortion on between original image and 128, 256, 512, 480 Indexed Image respectively.(From Top to Bottom).

서는 그 차이를 보기 어려워 그림 8과 같이 각각의 영상과 원본 영상과의 왜곡(차이 값)을 측정하여 도시하였다.

그림 8에서 보는 바와 같이 128, 256 컬러 맵을 사용



그림 9. 각 256, 512, 480 색인 영상에서의 왜곡지점
Fig. 9. Area of Distortion on each 256, 512, 480 Color Map respectively(From Top to Bottom)

한 경우 원본과의 비교 시 왜곡 부분이 확연하게 드러난다. 하지만 512, 480 컬러 맵을 사용한 경우는 미세한 차이는 있지만 두 경우 모두 원본에의 왜곡이 매우 적음을 원 영상의 윤곽(패턴)을 인지하기 어려움에서 알 수 있다.

그림 9는 가 출력 영상에서 왜곡이 비교적 잘 드러나는 부분에 직사각형을 표시한 것이다. 왜곡은 평면적인(저주파) 영역 또는 서로 다른 컬러들이 만나는 경계선 부근에서 많이 발생함을 알 수 있다.

그림 10은 9 bit 색인을 압축하기 위해 도입한 위치지도(LM)이다. 원래의 LM은 원영상 만큼의 크기를 가지지만 JBIG 계열의 이진 압축방법에 의해 매우 작아



그림 10. Lena 영상에서의 위치도
Fig. 10. Location map of Lena Image.

표 1. Lena 영상에서 컬러 맵 크기에 따른 PSNR
Table 1. PSNR Comparison according to each codebook size.

코드북 크기	128	256	480	512
PSNR	32.335	34.678	36.710	36.942

저 실제 Payload의 증가에 큰 영향을 미치지 않는다.

제안한 방법의 성능을 비교하기 위하여 영상의 화질을 측정하는 척도인 PSNR(Peak Signal to Noise Ration)을 측정하였다[식1~3].

$$MSE = \frac{\sum (Im(i,j) - Im^*(i,j))^2}{M*N} \quad (1)$$

$$RMSE = \sqrt{MSE} \quad (2)$$

$$PSNR = 20 \log_{10} \frac{255}{RMSE} \quad (3)$$

여기서 Im 은 원래의 BMP 영상, Im^* 는 복원된 BMP 영상으로 볼 수 있다. 표 1에서 256코드북의 경우 현재 대표적인 인덱스 영상 압축방법인 JPEG의 경우이며 본 논문에서 제안한 480 코드북 방법과 비교할 기준으로 삼을 경우 대체적으로 만족할만한 수준의 화질 개선을 보인 것을 알 수 있다.

IV. 결론 및 향후 연구

논문에서는 영상의 압축 중에서도 색인 영상에 대한 시각적 개선을 시도하였다. 일반적인 색인 영상은 픽셀의 값(색인)과 그에 대응하는 컬러 코드북을 이용하여 압축된 영상을 디스플레이 한다. 그리고 보편적으로 픽셀을 표현하는 인덱스는 28개의 컬러 코드북으로 구성되어 있으며 본 논문에서는 공통 코드북을 이용하여 기존의 코드북을 재구성함으로써 보다 풍부한 색상으로 영상을 표시할 수 있도록 하였다. 실험을 통해 제안된 방법(480 코드북)은 9 bit 색인을 사용했을 경우와 거의 비슷한 화질을 제공함을 알 수 있었다. 또한, 현재 많이 사용되는 256 코드북 방법과 비교하여 화질개선 측면에서는 많은 장점을 보이며 Payload의 증가는 크기 양을 알 수 있다.

하드웨어적인 한계로 한 픽셀을 9 bit로 표현하지는 못 하지만 제안된 코드북 재구성에 의해 가상적으로 그에 해당하는 시각적효과를 이끌어냈다. 이러한 형태의 코드북을 이용하여 정보은닉과 같은 응용을 개발하는 것 또한 가능하다.

REFERENCES

- [1] A. Somboonkaew, Cheevasuvit, K.Dehan, S.Mitatha and Wongkharn, "Colour Palette Compression via Vector Quantization for Colour Image Transmission", Asian Conference on Remote Sensing, 1997.
- [2] Balasubramanian, R.; Bouman, C.A.; Allebach, J.P. "Sequential scalar quantization of vectors: an analysis", *IEEE Trans. on Image Processing*, VOL 4, Issue 9, pp. 1282 - 1295, 1995.
- [3] Bernd W. Koptzick and Charles A. Bouman, "Optimized Universal Color Palette Design for Error Diffusion," *Journal of Electronic Imaging*, vol. 4, no. 2, pp. 131-143, April 1995.
- [4] Bian Yang, Zhe-Ming Lu, and Sheng-He Sun, "Reversible Watermarking In the VQ-Compressed Domain", *Visualization, Imaging, and Image Processing*, 1993.
- [5] L. Velho, J. Gomes, and M. V. R. Sobreiro, "Color image quantization by pairwise clustering", *Proc. Tenth Brazilian Symp*, pp. 203-210, 1997.
- [6] Moonki Cho, Yungsup Yoon, "BTC Algorithm Utilizing Multi-Level Quantization Method for Image Compression", *Journal of IEEK*, Vol. 50, No. 6, pp. 114-124, 2013.
- [7] N.M. Nasrabadi and R.A.King, "Image coding using vector quantization; A Review", *IEEE Trans. Comm.*, VOL-COM. 36, pp. 957-971, Aug. 1980.
- [8] R. Balasubramanian, C.A. Bouman and J.P. Allebach, "Sequential scalar quantization of vectors: an analysis", *IEEE Trans. on Image Processing*, VOL 4, Issue 9, pp. 1282 - 1295, 1995.
- [9] R. S. Gentile, E. Walowit, and J. P. Allebach, "Quantization and multilevel Halftoning of Color Images for Near Original Image Quality," *J. Opt. Soc. Am. A*, vol.7, no.6, pp.1019-1026, 1990.
- [10] R. W. Floyd and L. Steinberg, "An Adaptive

- algorithm for spatial grayscale,” Proc. SID, vol. 17, no. 2, pp. 75-77, 1976.
- [11] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison Wesley Longman, 1998.
- [12] Robert Y. Li, Jung Kim, and N. Al-Shamakh, “Image Compression using vector quantization”, *Image and Vision Computing* 20, pp. 37-45, 2002.
- [13] S.J. Wan, P. Prusinkiewicz, and S.K.M. Wong, “Variance-based color image quantization for frame buffer display,” *Color Res. Appl.* 15:52 - 5, 1990.
- [14] Xialolin Wu, “YIQ Vector Quantization in a New Color Palette Architecture,” *IEEE Trans. on Image Processing*, VOL. 5, No. 2, pp. 321-329, 1996.
- [15] An Introduction to Image Compression, “<http://www.debugmode.com/imagecmp/>”
- [16] F.S. HILL, Jr, *Computer Graphics using OpenGL*, Prentice Hall, 2001
- [17] USC-SIPI Image Database, “<http://sipi.usc.edu/database/>”
- [18] JBIG Algorithm “<http://www.imagepower.com/>”

— 저 자 소 개 —



최 용 수(정회원)-교신저자
1998년 강원대학교
제어계측공학과 공학사
2000년 강원대학교
제어계측공학과 공학석사
2006년 강원대학교
제어계측공학과 공학박사
2006년~2007년 연세대학교 첨단융합건설연구단
연구교수
2007년~2013년 고려대학교 정보보호대학원
연구교수
2013년~현재 성결대학교 교양교직부(멀티미디어) 조교수
<주관심분야 : Multimedia Hashing, Information Hiding, Watermarking, Steganography, Image Forensics, Fogery Detection>



이 달 호(정회원)
1982년 서울대학교
제어계측공학과 공학사
1985년 서울대학교
제어계측공학과 공학석사
1992년 서울대학교
제어계측공학과 공학박사
1992년~현재 가천대학교 전자공학과 교수
<주관심분야 : 시스템 식별, 필터링 기법, INS 응용, Data Hiding>