

순간포착에 의한 칼라 영상 압축과 블록 절단 코딩

박호식* · 차영석* · 최현수* · 권오홍* · 배철수* · 최철재** · 나상동***

*관동대학교 전자통신공학과, **동우대학 전자계산학과, ***조선대학교 컴퓨터공학과

Color Image Compression by Moment-Preserving and Block Truncation Coding Techniques

Ho-sik Park* · Young-seok Cha* · Hyun-soo Choi* · Cheol-soo Bae*

Oh-heung Kwon* · Chul-jae Choi** · Sang-dong Ra***

*Kwandong University, **Dong-U College, ***Chosun University

개 요

새로운 칼라 영상 압축은 순간 포착과 블록 절단 코딩에 기초로 두 개의 대표적인 칼라 중 하나로 할당된 각각의 블록 화소와 겹치지 않은 블록으로 분할된 입력은 블록에서 포착된 어떤 순간으로부터 유래된 해석방식으로 계산된다. 비트맵은 화소의 칼라를 묘사하기 위해 각각의 블록을 일반화하며 별개의 균일한 조건에 의해 설정된 대표적인 칼라는 동일하다고 간주되고 부호 비트를 저장하기 위해 이용된다. 본 연구에서 실험결과에 의해 13이상의 좋은 평균 압축률이 달성될 수 있었다.

I. 서 론

Delp와 Mitchell [1]에 의해 개발된 블록 절단 코딩(BTC) 알고리즘은 출력레벨이 입력 샘플의 첫 번째와 두 번째 순간 포착에 의해 얻어지는 two-level의 겹치지 않는 양자화기이다.

Lema와 Mitchell [2]은 칼라 영상의 3개의 칼라면에 절대 순간 블록절단코딩 방법을 적용했다. 압축률은 영상 독립이고 항상 4이다. 압축 능력을 증가시키기 위해, Wu와 Coll

[3]은 3개의 칼라 면을 모두 양자화하기 위해 단일 비트맵을 사용하였다. 3개의 비트맵 중 하나가 포착되는 것이 필요하다. Kurita와 Otsu [4]는 두 종류로 분류한 픽셀과 각각의 픽셀에 대한 최초의 스코어를 계산하기 위해 칼라 벡터의 평균 벡터와 공분산 매트릭스를 사용했다. 두 종류의 평균벡터와 비트맵은 포착된다. [3]와 [4]에 의해, 블록의 크기가 4×4 일 때의 압축률은 항상 6이 된다.

본 논문에 의해, 순간 포착 원리와 블록 절단 코딩 진행에 기초한 새로운 방법이 제안되었다. 입력 영상은 4×4 크기의 겹치지 않은 블록에서 나누어진다. 각각 블록의 몇몇 칼라 모멘트는 포착되고 해석 공식은 두 개의 대표되는 칼라 C_1 과 C_2 를 계산하기 위해 [5]-[7]에 의해 파생된다.

블록의 3개의 칼라 구성요소(R, G, B)인 $3-k$ 가 거의 동일하고, 또한 $3-k$ 칼라 구성요소의 변화가 매우 작다면 블록은 k -스펙트럼($0 \leq k \leq 3$)이라고 불린다. 블록에서 칼라 조건의 형태는 3-spectral, 2-spectral, 1-spectral, 0-spectral로 정의된다. 3-b 스펙트럼 확인(SI) 코드는 다른 동일한 조건의 칼라를 나타내곤 한다. 이 코딩 방법은 압축률을 개선하는데 효과적임이 밝혀졌다. 블록 픽셀이

C_1 이나 C_2 이든 간에 비트맵에 표시된 각각의 비트를 가지고 일반화되어진다. 압축률을 개선하기 위해 제안된 압축 방법의 확장된 버전으로 제안되었다.

II. 순간포착과 BTC 방법을 사용하여 제안된 칼라 영상 압축

순간포착과 BTC 방법을 사용한 칼라 영상 압축의 제안된 알고리즘은 다음과 같이 설명된다.

step 1 : 겹치지 않은 4×4 블록 f 의 분할과 칼라 영상 f 를 읽어 들인다.

step 2 : f 의 블록 f 를 가지고 다음과 같이 실행한다.

- 1) 블록 f 안의 칼라 구성요소인 픽셀의 값들의 모멘트를 계산한다.
- 2) 블록 f 에서 픽셀들의 칼라 구성요소의 값(R, G, B)들의 변화에 기초한 일정한 스펙트럼 밴드의 수치를 계산한다.
- 3) 블록 f 에 대한 두 개의 대표되는 칼라인 C_1 과 C_2 를 계산하기 위한 방법을 선택한다.
 - a) 0-스펙트럼인 경우 — 블록 f 의 세 개의 칼라 면 모두 거의 동일한 경우; 단 단일 칼라 C 를 $C_1=C_2=C$ 라고 놓고 SI 코드를 000이라 놓고 세 개의 평균을 계산한다.
 - b) 1-스펙트럼인 경우 — 단지 하나의 칼라 면이 일정하지 않은 경우; Tsai [5]에 의해 제안된 순간포착 thresholding 방법을 사용한

다. 두 개의 칼라 면이 일정한 것에 달려있는 SI 코드를 001, 010, 100이라 놓고, 블록 f 에 대한 C_1 과 C_2 를 형성하기 위해 일정하지 않은 칼라 면에 대한 두 개의 대표되는 칼라의 구성요소 값들을 계산하고 두 개의 일정한 칼라 면의 평균을 계산한다.

c) 2-스펙트럼인 경우 — 두 개의 칼라면이 일정하지 않은 경우; 두 개의 일정하지 않은 칼라면의 두 개의 대표되는 칼라 구성요소의 값을 계산하고 일정한 칼라의 평균을 계산하기 위해 Lin과 Tsai [7]에 의해 제안된 무리들의 특징포착 방법을 사용한다. 칼라면이 동일하면 SI 코드를 011, 110 혹은 101로 놓고 f 에 대한 칼라 C_1 과 C_2 를 구성한다.

d) 3-스펙트럼인 경우 — 다음과 같이 사용한다.

- i) 두 개의 대표되는 칼라 $C_1=(R_1, G_1, B_1)$ 과 $C_2=(R_2, G_2, B_2)$ 를 계산하기 위해 확실한 모멘트를 포착하고 결과식을 푼다.
- ii) p_i 의 원래 칼라 값(r_i, g_i, b_i)가 (R_1, G_1, B_1) 혹은 (R_2, G_2, B_2) 이기 위한 closer인지 아닌지에 따라 블록 f 에 대한 각각의 픽셀 p_i 에 새로운 칼라를 지정한다.
- iii) SI 코드를 111로 놓는다.

4) p_i 의 새로운 칼라 값이 C_1 인지 C_2 인지에 따라서 f 에 대한 각각의 픽셀 p_i 에 할당된 0 혹은 1에 의해 블록 f 의 4×4 비트맵이 산출된다.

5) 두 개의 칼라 C_1 과 C_2 의 6번째 8-b 구성요소 값 중에 반복되는 하나를 버리고 출력에 따라 SI 코드와 비트맵과 함께 하나를 남겨둔다. SI 코드에 따

라, 미래의 출력을 사용하여 블록을 재구성할 수 있다. 이 코딩 방법은 압축률을 재구성할 수 있다. SI 코드를 사용하여 복구된 규칙이 표 I 에 나타나 있다.

step 3 : f 에 대한 모든 블록이 처리되었으면 멈춘다.

표 1) 코딩과 재복구 방법을 사용한 SI 코드

SI Code	코딩 규칙		재복구 규칙	
	블록 스펙트럼 상태	6개의 8-b 칼라 구성요소에 대응하는 부분	"0" in bit map	"1" in bit map
000	0-spectral	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)	(none)
001	1-spectral	(R ₁ ,G ₁ ,B ₁)and B ₂	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
010	1-spectral	(R ₁ ,G ₁ ,B ₁)and G ₂	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
100	1-spectral	(R ₁ ,G ₁ ,B ₁)and R ₂	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
011	2-spectral	(R ₁ ,G ₁ ,B ₁)and(G ₂ ,B ₂)	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
110	2-spectral	(R ₁ ,G ₁ ,B ₁)and(R ₂ ,G ₂)	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
101	2-spectral	(R ₁ ,G ₁ ,B ₁)and(R ₂ ,B ₂)	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)
111	3-spectral	(R ₁ ,G ₁ ,B ₁)and(R ₂ ,G ₂ ,B ₂)	(R ₁ ,G ₁ ,B ₁)	(R ₁ ,G ₁ ,B ₁)

언급된 step 2-3중에 블록 f_j 에 대한 포착된 칼라 모멘트는 다음과 같다.

$$m_r = \frac{1}{n} \sum_i r_i, \quad m_g = \frac{1}{n} \sum_i g_i, \quad m_b = \frac{1}{n} \sum_i b_i$$

$$m_{r^2} = \frac{1}{n} \sum_i r_i^2, \quad m_{g^2} = \frac{1}{n} \sum_i g_i^2$$

$$m_{b^2} = \frac{1}{n} \sum_i b_i^2, \quad m_{rgb} = \frac{1}{n} \sum_i r_i g_i b_i$$

n 이 블록 f_j 의 픽셀에 대한 전체 수치이고; m_r, m_g, m_b 은 각각의 칼라 구성요소 값(r, g, b)의 평균이다; $m_{r^2}, m_{g^2}, m_{b^2}$ 은 각각의 (r, g, b)값의 변화와 관계가 있다; m_{rgb} 은 (r, g, b)값 범위내에서 상호관계가 있다.

제안된 알고리즘 CICMPBTC는 다양한 스펙트럼을 나타내고 Step 2-5)에 설명된 것처럼 영상 블록이 일정한 칼라 면을 가질 때 반복적인 칼라 구성요소 값을 나타내는데 사용되는 비트를 구하기 위해서 3-b SI 코드를 사용한다. 자세한 규칙은 표 I 에 나타나 있다.

리시버는 표에 있는 3-b SI 코드에 의해 6개의 8-b 칼라 구성요소 값과 16-b 비트맵에 대응하는 부분에 대한 영상 블록을 쉽게

III. 제안된 알고리즘

인간의 시야는 그렇게 많은 칼라를 구별할 수 없다. 각각의 칼라 구성요소에 대해 8bit로 이루어져 있다. 그러므로 8에서 5b까지 각각의 칼라 구성요소의 대표적인 bit를 줄이기 위해 사용된 간단한 기술 [8]에 의해 적당한 SNR 값을 가지고 더 좋은 압축률을 얻을 수 있는 제안된 방법에 이를 수 있다. 그 밖에, 16-b 비트맵의 결과가 아직까지 출력 코드의 35%이상을 차지하고 있다. 비트맵에서 쓸데없는 부분을 줄이기 위해서 [9]의 방법을 사용할 수 있다. 이 방법은 64로 미리 정의된 표준 비트맵으로 나타낸 6-b index를 사용한다.

작은 영상 블록에서의 edge와 line 패턴에 대한 인간의 시각 시스템의 민감한 관찰력에 따라서 설계된다. 그리고 64 표준 비트맵과 비트맵을 매칭한 후 64 index중 하나에 의해 각각의 영상 블록의 비트맵을 표현한다.

더욱 명확하게, Step 2-4)와 Step 2-5)의 알고리즘 CICMPBTC에 의해 다음과 같이

표 2) 기존의 BTC, 단일 비트 맵 BTC, 제안된 알고리즘 CICMPBTC, CICMPBTC*의 SNR 값과 압축률 비교

	기존의 BTC	단일 비트 맵 BTC	4×4 블록 CICMPBTC	5×5 블록 CICMPBTC	4×4 블록 CICMPBTC*
lena	32.85/4.0	29.03/6.0	31.55/7.03	30.95/9.27	29.14/12.29
pepper	32.46/4.0	27.43/6.0	30.93/6.60	30.12/8.80	28.54/11.47
house	39.93/4.0	31.16/6.0	36.77/7.75	33.32/9.81	33.47/13.67
jet	32.10/4.0	28.45/6.0	31.03/7.71	30.53/10.04	28.63/13.59
candy	39.60/4.0	31.04/6.0	37.52/8.20	33.87/10.51	33.20/14.57
balloon	34.93/4.0	28.32/6.0	33.53/7.36	32.26/9.55	31.16/12.92
average	35.31/4.0	29.23/6.0	33.56/7.44	31.84/9.66	30.69/13.09

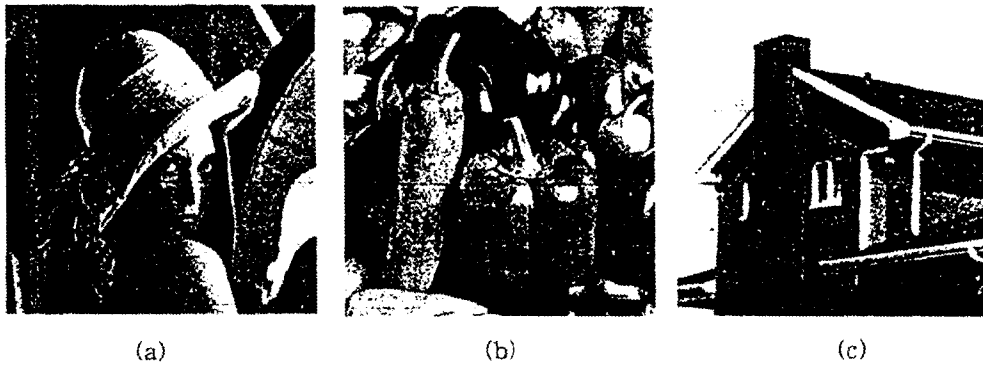


그림 1. "Lena"(a), "Pepper"(b) 와 "house"(c)의 표준 영상 (512×512)

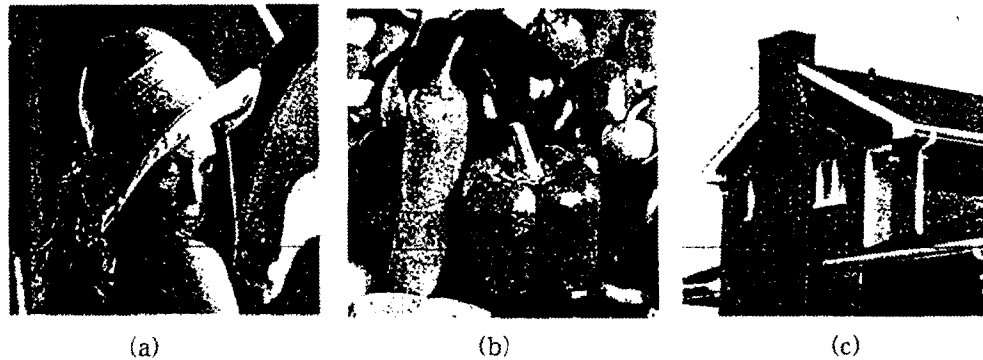


그림 2. 블록 크기 4×4인 CICMPBTC 알고리즘을 사용한 압축결과로부터 얻어진 (a) Lena, (b) Pepper 와 (c) house의 영상, (a) 압축률=7.03 (b)압축률=6.60 (c) 압축률=7.75

다시 쓸 수 있다. (알고리즘의 결과는 Step 2-4):

CICMPBTC*라 한다)

a) (r_i, g_i, b_i) 이 (R_1, G_1, B_1) 혹은 (R_2, G_2, B_2)

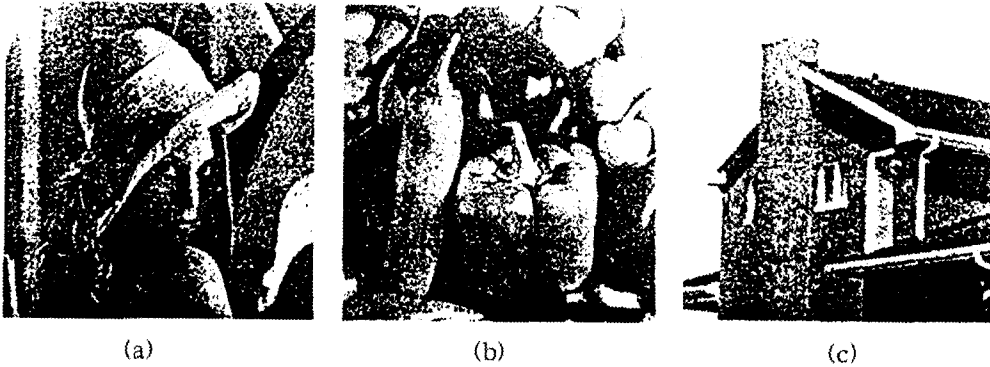


그림 3. 블록 크기 5×5인 CICMPBTC 알고리즘을 사용한 압축결과로부터 얻어진 (a) Lena, (b) Pepper 와 (c) house의 영상, (a)압축률=9.27 (b)압축률=8.80 (c) 압축률=9.87

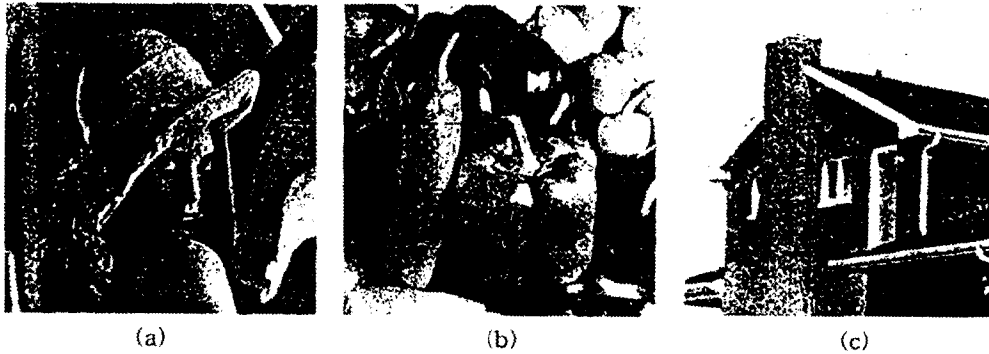


그림 4. 블록 크기 4×4인 CICMPBTC* 알고리즘을 사용한 압축결과로부터 얻어진 (a) Lena, (b) Pepper 와 (c) house의 영상, (a) 압축률=12.20 (b)압축률=11.47 (c) 압축률=13.67

의 폐쇄기인지 아닌지에 따라서 블록 f 의 각각의 픽셀 p_i 를 0과 1로 표시한 것에 의해 블록 f 에 대한 4×4 비트맵 M_f 를 생성한다.

b) 다른 비트 값에 대응하는 픽셀의 가장 작은 수가 되는 M_f 와 가장 유사한 것을 M_k 라 할 때 M_k 와 각각의 64 표준 비트맵에 의해 미리 정해진 M_f 를 비교한다.

c) 각각의 칼라 구성요소를 8에서 5b까지 양자화한다.

Step 2-5):

M_k 의 6-b index와 출력에 의한 SI 코드,

두 개의 대표되는 칼라 C_1 과 C_2 에 의해 겹치지 않는 구성요소 값이 주어진다.

IV. 실험결과 및 확장 버전

알고리즘 CICMPBTC와 CICMPBTC*는 두 개의 방법 모두 몇몇의 칼라 영상을 IRIS

Indigo 워크스테이션에 의해 실험되었으며 각각의 칼라 영상은 한 픽셀당 24bit를 가지고 있고 크기는 512×512이다. 영상 복구의 실험 평가를 위한 기준으로 SNR을 사용한다.

표2는 4×4와 5×5크기의 블록을 가지는 제

안된 알고리즘과 단일 비트맵 BTC, 3개로 나누어진 전통적인 비트맵 방법에 의해 생성된 압축률과 SNR 값을 비교하여 보여주고 있다.

그림1은 "Lena", "peper", 그리고 "house"의 표준 영상을 보여주고 있다. 그림 2와 그림 3은 블록 크기 4×4 와 5×5 에 CICMPBTC 알고리즘을 사용하여 그림 1에 대한 압축결과를 다시 영상으로 만들어 보여주고 있다. 그림 4는 블록 크기 4×4 에 알고리즘 CICMPBTC*를 사용하여 그림1에 대한 압축결과를 다시 영상으로 만들어 보여주고 있다.

표2의 세 번째와 마지막 단을 주의깊게 보면, CICMPBTC*에 의한 실험결과가 압축률과 SNR 값 모두 보다 좋아 졌음을 알 수 있다.

V. 결론

순간포착 방식과 BTC에 기초한 영상 압축으로의 새로운 접근이 제안되었다. 제안된 방법의 실행은 기존의 BTC (3개의 비트맵을 가지는)와 단일 비트맵 BTC 방법과 비교하여 볼 때 CICMPBTC와 CICMPBTC* 기술이 보다 나은 방법임을 알 수 있다. 평균 압축률 13.09와 평균 SNR 30.69는 제안된 알고리즘의 확장된 버전이 주어진다면 실험된 영상으로부터 얻어질 수 있다.

[References]

- [1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," IEEE Trans. commun, vol. COM-27, pp. 1335-1342.
- [2] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," IEEE Trans. Commun. vol. COM-32, pp. 1148-1157, 1984.
- [3] Y. Wu and D. C. coll, "Single bit map block truncation coding for color image," IEEE Trans. Commun. vol. COM-35, pp. 352-356, Mar. 1987.
- [4] T. Kurita and N. Otsu, "A method of block truncation coding for color image compression," IEEE Trans. Commun, vol. COM-35. pp. 352-356, Mar. 1987.
- [5] W. H. Tsai, "Moment-preserving thresholding: A new approach," CVGIP, vol, 29, pp.377-393, 1985.
- [6] C. K. Yang, C. T. Wu, J. C. Lin, and W. H. Tsai, "color image sharpening by moment-preserving technique," signal Processing, vol. 45, no. 3, pp. 397-403, 1995.
- [7] J. C. Lin and W. H. Tsai, "Feature-preserving clustering of 2D data for two-class problems using analytical formulas: An automatic and fast approach," IEEE Trans. Pattern Anal. Machine Intell., vol. 16, pp. 554-560, 1994.
- [8] B. J. Kurz, "Optimal color quantization for color displays," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1983, pp. 217-224. I
- [9] K. Yang and W. H. Tsai, "Improving block truncation coding using ling and edge information for gray-scale image compression," Pattern Recognit. Lett., vol, 16, pp.67-75, 1995.