Q-factor변형에 의한 색조영상 압축에 관한 연구

최금수 , 문영득
부산외국어대학교 전자공학과

Image Compressing of Color tone image by transformed Q-factor

Kum-Su Choi , Young-Deck Moon
Dept. of Electronics Engineering. of Pusan University of Foreign studies

Abstract -A storage or transmission of image is difficult without image compression processing because the numbers of generated or reborned image data are very much. In case of the random signal, image compression efficiency is low doing without loss of image information, but compressibility by using JPEG is better. We used Huffman code of JPEG, it assigne the low bit value for data of a lot of generated frequency, assigne the high bit value for data of a small quantity. This paper improved image compression efficiency with transformming Q-factor and certified the results with compressed image. A proposed method is very efficience for continuos a color tone image.

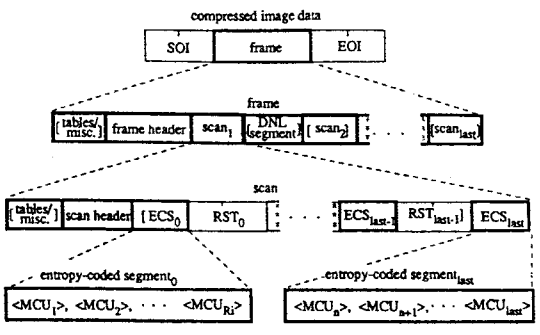
1. 서 론

문자, 오디오 및 비디오 등의 총합체로서 구성되는 멀티미디어 데이터중에서 영상은 그 의미 전달 효과가 매우 크기 때문에 멀티미디어 데이터의 핵심이라 할 수 있다. 그러나, 영상은 단위 시간당 발생 또는 재생되는 데이터량이 방대하여 압축 과정 없이는 저장 또는 전송이 어렵기 때문에 압축은 필수적이라고 할 수 있다. 본 논문에서는 영상압축 기법의 하나인 JPEG의 허프만 부호화의 생성 방법에 대하여 논하고, 생성한 코드에 의한 이미지 압축효율을 실험을 통해 확인하였다.

2. 본 론

2.1 데이타 구조

JPEG의 파일 포맷은 그림 1과 같다. SOI(Start of Image)는 FFD8의 코드값을 가지며 압축 이미지의 처음을 나타내는 마커코드이다. EOI(End of Image)는 FFD9의 코드값을 가지며 이미지의 마지막을 나타낸다.



〔그림 1〕 JPEG의 파일포맷

또한 DQT(Define Quantization Table)는 FFDB의 코드값을 가지며 휘도와 색차의 양자화 테이블이 2개 들어있고 DHF(Define Huffman Table)는 FFC4의 코드값을 가지며 허프만 테이블이 4개 존재한다. FFC0의 코드를 가지는 프레임 헤더(Start of Frame)에는 이미지의 정보와 컴포넌트 수를 가지고 있고 FFDA의 코드를 가지는 스캔헤더 마지막에는 실제적인 데이터 정보를 포함하고 있는 MCU(Minimum Coded Units)가 존재한다.

2.1.1 데이타 압축

JPEG의 인코더는 원영상(Source Image)을 받아서 압축을 행하는 기능을 수행하고, 디코더는 압축된 파일을 읽어서 복원 영상을 만들어 주는 기능을 한다. 이들은 서로 대칭적인 구조를 가지고 있다.

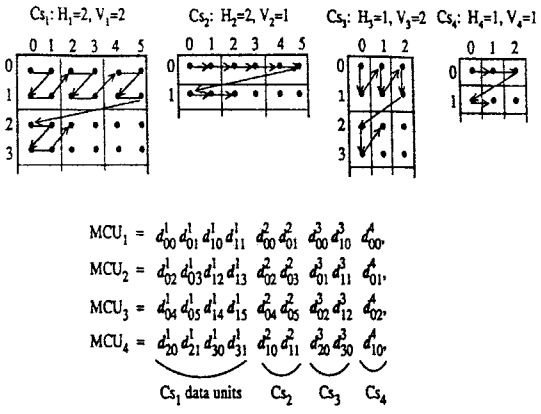
RGB 공간과 새로운 공간 사이의 변환을 보다 빨리 처리하기 위해 CCIR 601 칼라 공간이라 불리는 Y_Cr_Cb 칼라 공간을 정의하였다. 이 칼라 공간은 YUV와 같은 칼라 공간을 사용하나 U와 V의 계수를 조정함으로써 정수 연산과 시프트 연산만으로 RGB와 Y_Cr_Cb와의 변환이 가능하게 하였다. 따라서 현재 JPEG 표준을 기반으로 하는 정지영상 압축에서는 이 칼라 공간을 사용한다.

RGB에서 Y_Cr_Cb로의 칼라공간 변환은 식(1)과 같고

$$Y=(0.299)*R+(0.587)*G+(0.114)*B$$
$$Cr=(0.5)*R-(0.418)*G-(0.081)*B \quad \cdots(1)$$
$$Cb=(0.5)*B-(0.331)*G-(0.168)*R$$

Y_Cr_Cb에서RGB로의 칼라공간 변환은 식(2)와 같다.

$$R=(1)*Y+(1.402)*Cr$$
$$G=(1)*Y-(0.71414)*Cr-(0.34414)*Cb \quad \cdots(2)$$
$$B=(1)*Y+(1.772)*Cb$$



〔그림 2〕 양방향 인터리빙 방식의 MCU

JPEG에서는 CCIR-602에서 정의한 양방향 칼라 샘플링을 이용하고 있다. 이것은 Y 신호를 수평으로 2번, 수직으로 2번 수행하는 동안 Cr, Cb신호는 각각 1번씩 수행하는데, 이런 방법에 의해 구성된 8×8 블록을 인터리빙 방법으로 MCU(Minimum Coded Unit)를 구성하게 된다.

8×8의 영상 데이터 블록을 64개의 공간 주파수 성분(DCT계수)으로 분리하는 FDCT(Forward DCT) 과정을 거친 후 가역 관계에 있는 IDCT(Inverse DCT) 과정을 통해 다시 원영상 데이터 블록으로 환원한다.

8×8 영상 데이터 블록에 있는 각각의 화소들을 FDCT 하는 과정과 역으로 IDCT하는 과정을 식으로 나타내면 식(3), 식(4)와 같다.

FDCT 변환에 대해서는

$$S_{vu}=\frac{1}{4}C_u C_v \sum_{x=0}^{7}\sum_{y=0}^{7} S_{xy}\cos[\frac{(2x+1)u\pi}{16}]\cos[\frac{(2y+1)v\pi}{16}] \quad \cdots(3)$$

IDCT 변환에 대해서는

$$S_{xy}=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_u C_v S_{vu}\cos[\frac{(2x+1)u\pi}{16}]\cos[\frac{(2y+1)v\pi}{16}] \quad \cdots(4)$$

여기서

C_U와 $C_V=\frac{1}{\sqrt{2}}$ 일 때 $U, V=0$이며

C_U와 $C_V=1$일 때 $V, U\neq 0$ 이다.

FDCT의 결과는 64개의 양자화 계수들을 갖는 양자화 테이블에 의해 유니폼 양자화 된다. 이 과정의 목적은 실제적으로 압축효과를 얻기 위한 것인데, 일반적으로 JPEG에서의 압축률 조정이 여기에서 이루어진다.
그 양자화를 위한 관계는 식(5)와 같다.

$$F_q(u,v)=round\frac{F(u,v)}{Q(u,v)} \quad \cdots(5)$$

2.2 허프만 코드 생성 및 이미지 압축

심볼들의 빈도수, 즉 픽셀들의 밝기에 대한 히스토그램을 만들고, 각 심볼 및 그 빈도수를 발생빈도의 내림차순으로 정렬한다. 허프만 트리를 통해, 각 심볼들에 대한 허프만 코드를 쉽게 구할 수 있다.DC계수는 AC계수와는 달리 각 세그먼트마다 하나씩 밖에 없고, 또 인접한 세그먼트간의 DC계수 값의 차이가 많지 않기 때문에 바로 전 세그먼트의 DC계수 값과의 차이 ($DIFF=DC_i-DC_{i-1}$)를 부호화한다.

〔표 1〕 AC, DC의 그룹번호

그룹번호	값의 범위
0	0
1	-1,1
2	-3, -2, 2, 3
3	-7...-4, 4...7
4	-15...-8, 8....15
5	-31...-16, 16...31
6	-63...-32, 32...63
7	-127...-64, 64...127
8	-255...-128, 128...255
9	-511...-256, 256...511
A	-1023...-512, 512...1023
B	-2047...-1024, 1024...2047

부호화 과정은 DIFF가 속한 그룹의 번호를 〔표 1〕에서 찾아 SSSS에 할당한 후, 그룹 번호에 해당하는 수만큼을 DIFF의 비트열에서 취하여 부가한다. 만약 DIFF가 음수일 경우에는 DIFF-1 값에 대한 비트열에 같은 방법으로 부가비트 수를 계산한다.하나의 세그먼트를 구성하는 63개의 AC계수들은 지그재그 순서로 부호화 되는데, 만일 계수값이 0 이면 런랭스 카운터에 의해 0의 런랭스(RRRR)가 계산되고, 0 이 아닌 계수가 나타나면 DC계수과 같은 방법으로 AC계수를 부호화한다.이러한 과정을 한 세그먼트 내에 있는 63개의 AC계수가 모두 처리될 때까지 반복한다. 이 때 주의해야 할 것은 런랭스 RRRR의 값이 16이상인 경우에는 ZRL이라는 코드를 부여하고 RRRR에세 16을 빼주어야한다는 것이다. 이 ZRL부호는 0xF0의 값을 가지고 있는데, 15개의 0이 앞에 있고 이어서 0값을 갖는 심볼이 나타난 경우를 의미한다. 이 부호를 사용하는 이유는 런랭스를 15이하로 제한하기 위해서다. 그리고 끝에 남은 AC계수들이 모두 0인 경우에는 EOB를 사용해서 부호화한다. 즉, 몇 개인가에 관계없이 세그먼트 끝에 남은 0들은 모두 EOB로 처리된다. 실제로 AC계수들은 상당 부분이 0값을 갖는다. 이러한 방법에 의해 압축률을 상당히 높일 수 있다. 모든 AC계수의 값이 0이라면, BOB 한 개로 부호화가 가능하다. 이러한 과정을 통해 영상을 구성하고 있는 모든 세그먼트에 대해 수행하면 허프만 부호화가 이루어진다.

2.3 실험 및 결과

JPEG 영상압축에 있어서 허프만 부호화의 효율성을 입증하기 위해 실제 영상 데이터에 적용시켜 실험하였다. 실험에 사용된 영상은 256×256의 해상도와 8bpp(bit per pixel)을 가진 컬러 영상(Color Image)이다. 부호화 방법은 허프만 부호화와 Arithmetic 부호화 방법을 사용하였으며 압축률 조절을 위해 Q-factor를 1.0에서 3.0까지 조절하였다.

〔표 2〕 영상압축 결과

파일명	코딩타입	Q-factor	압축 사이즈	압축률
girl.raw	huffman	1.0	16548	12:1
girl.raw	huffman	2.0	10009	20:1
girl.raw	huffman	3.0	7425	26:1
girl.raw	Arithmetic	1.0	18916	10:1
girl.raw	Arithmetic	2.0	12438	15:1
girl.raw	Arithmetic	3.0	9481	21:1
mall.raw	huffman	1.0	11883	17:1
mall.raw	huffman	3.0	5922	33:1
mall.raw	Arithmetic	1.0	13183	15:1
mall.raw	Arithmetic	3.0	8197	24:1
f-16.raw	huffman	1.0	8094	24:1
f-16.raw	huffman	3.0	4709	42:1
f-16.raw	Arithmetic	1.0	9450	21:1
f-16.raw	Arithmetic	3.0	5620	35:1

표 2는 각각의 영상에 허프만 부호화와 Arithmetic 부호화 과정을 거친 후 압축된 영상의 크기를 나타낸다. Q-factor에 따라 압축률이 달라지는 것을 볼수 있다.결과를 보면 Arithmetic 부호화보다 허프만 부호화가 영상 압축에 효과적임을 알 수 있다. 또한 영상의 압축률은 원영상에 따라 다르게 나타나는데 영상이 연속적인 색조 영상일 경우 더욱 압축률이 높아진다는 것을 알 수 있었다. girl영상보다 f-16영상이 연속적인 색조가 많은 영상이다. 결과에서 보듯이 f-16영상이 girl영상보다 큰 압축률을 나타내고 있다.

〔a〕원 이미지

〔b〕복원영상 Q-factor1.0 허프만 부호화

〔c〕복원영상Q-factor3.0 허프만 부호화

〔d〕복원영상Q-factor3.0 Arithmetic 부호화

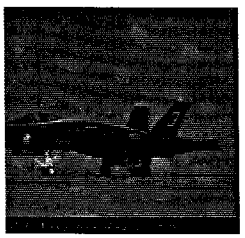
〔그림 3〕girl영상의 원영상과 복원영상

그림3-(a)는 실험에 사용한 256×256 영상이고 그림 3-(b)는 Q-factor를 1.0으로 허프만 부호화한 후 복원한 영상으로 압축률은 12:1이었으며 영상의 화질 저하는 거의 없었다. 그림3-(c)는 Q-factor를 3.0으로 허프만 부호화한 영상으로 압축률은 26:1이다. 그림 3-(d)는 Q-factor를 3.0으로 Arithmetic 부호화한 후 복원한 영상으로 압축률은 21:1이다.

(a)원영상

(b)복원영상Q-factor3.0

(c)원영상



(d)복원영상Q-factor3.0

〔그림 4〕 mall과f-16 복원영상

그림4는 mall영상과 f16영상을 Q-factor 3.0으로 허프만 부호화후 복원한 영상이다. 그림에서 보듯이 사람의 눈으로 화질의 차이를 느낄수 없었을 뿐만 아니라 특히 연속적인 색조 영상에서 높은 압축률을 얻을수 있었다. mall영상의 압축률은 35:1이고 f-16영상의 압축률은 42:1이다.

3. 결 론

Q-factor를 변형하므로 영상압축 효율을 향상시켰으며 그 결과를 압축 이미지를 통해 확인하였다.

허프만 부호화는 발생빈도가 많은 데이터에 대하여는 낮은 비트 값을 할당하고, 발생빈도가 적은 데이터에 대하여는 높은 비트 값을 할당하며, 부여된 코드는 가변길이이고 영상내의 고주파부분과 저주파부분을 DCT와 지그재그 스캔으로 추출하여 이를 허프만 부호화하여 영상을 압축한다. Q-factor의 변형으로 압축효율을 높일 수 있었으며 압축 시 수반되는 화질저하는 미미함을 압축된 이미지를 통해 확인할 수 있었다. 제시한 영상압축 방법은 연속적인 색조영상 압축에 매우 효율적이었다.

〔참 고 문 헌〕

〔1〕 K. T. Mullen. The Contract Sensitivity of Huffman Color Vision to Red-Green and Blue-Yellow Chromatic Gratings. J. Physiol. 359:381-400. 1985

〔2〕 Rafael C. Gonzales. Digital Image Processing. 362-407

〔3〕 E. Feig and E. Linzer. Discrete Cosine Transform Algorithms for Image Data Compression. Proceedings Electronic Imaging '90 East. pp84-7. Boston. MA(Oct. 29-Nov. 1, 1990)

〔4〕 M. Rabbani and P. W. Jones. Digital Image Compression Techniques. Bellingtam, WA: SPIE Optical Engineering Press(1991)

〔5〕 W. B. Pennebaker, J. L. Mitchell, G. L. London, and R. B. Arps. An overview of the basic pronciples of the Q-coder adaptive binary arithmetic coder. IBM J. Res. Develop. 32(6):717-26(Nov.1988).

〔6〕 J. G. Cleary, I. H. Witten, and R. M. Neal. Arithmetic Coding for Data Compression. Commun. of the ACM. 30(6):520-40(Jun. 1987)