

산술부호화를 이용한 인덱스 칼라 이미지에서의 효율적인 무손실 압축 방법

정희원 유강수*, 이한정*, 장의선**, 곽훈성*

An Efficient Lossless Compression Algorithm using Arithmetic Coding for Indexed Color Image

Kang-Soo You*, Han-Jeong Lee*, Euee S. Jang**, Hoon-Sung Kwak* *Regular Members*

요 약

본 논문에서는 팔레트 기반 이미지(palette-based image) 또는 인덱스 이미지(indexed image)라고 불리는 256색의 이미지에 대한 압축 성능 향상을 위한 새로운 알고리즘을 소개한다. 제안한 방식은 현재 색상이 갖는 인덱스 값을 중심으로 다음에 나오는 색상의 인덱스가 얼마나 발생하는지를 추정하고, 발생 빈도를 정렬하여 순위를 구한 후에, 색상에 대한 인덱스 값을 순위로 표현하여 원래의 인덱스 이미지를 대체한다. 그렇게 변화된 순위 인덱스 이미지(ranked index image)의 인덱스 분포에서는 순위가 높은 곳에 같은 인덱스들이 더 많이 존재하기 때문에 데이터 중복성(redundancy)을 높일 수 있어, 압축 효율을 기대할 수 있다. 실험 결과에서는, 기존의 산술부호화 방식, 휘도 성분 기반의 JPEG-LS 방식 그리고 인덱스 기반의 GIF 방식들과 비교할 때, 원 이미지에 대한 압축률이 최대 22.5까지 향상되어 제안한 방식의 압축 성능이 훨씬 뛰어나다는 것을 보여준다.

Key Words : lossless compression, palette-based image, markov chains, arithmetic coding

ABSTRACT

This paper introduces a new algorithm to improve compression performance of 256 color images called palette-based or indexed images. The proposed scheme counts each frequency of index values after present index value and determines each rank for every index value by sorting them in descending order. Then, the scheme makes ranked index image instead of original indexed image using the way to replace index values with ranks. In the ranked index image's distribution produced as a result of this algorithm, the higher ranked index value, the more present same values. Therefore, data redundancy will be raised and more efficient performance of compression can be expected. Simulation results verify that because of higher compression ratio by up to 22.5, this newly designed algorithm shows a much better performance of compression in comparison with the arithmetic coding, intensity-based JPEG-LS and palette-based GIF.

I. 서 론

정보통신 기술의 발달과 함께 이미지 데이터는 현대 사회의 다양한 분야에서 폭넓게 활용되고 있다. 여러 매체에서 다양한 형태로 사용되고 있는 이미지

들은 디지털로 표현된 데이터들인데, 이것들의 용량은 다른 데이터들에 비해 비교적 크기 때문에 효율적인 사용을 위해서 대부분 압축하여 저장 혹은 전송을 하게 된다.

이때 사용되는 이미지 압축은 손실 압축(lossy

* 전북대학교 영상공학과 컴퓨터비전 및 영상처리 연구실(mickey@chonbuk.ac.kr, sosim@chonbuk.ac.kr, hskwak@chonbuk.ac.kr)

** 한양대학교 정보통신학부 디지털미디어 연구실(esjang@hanyang.ac.kr)

논문번호 : #KICS2004-10-245, 접수일자 : 2004년 10월 24일

compression)과 무손실 압축(lossless compression)으로 분류된다. 손실 압축은 부호화할 때 매우 높은 압축률(compression ratio)을 가질 수 있으나, 복원 후 이미지 데이터의 손실을 가져온다는 단점을 가지고 있다. 비록 압축 기술의 발달로 인해 정지 영상 압축 표준인 JPEG^[1]의 경우, 20:1의 손실 압축에서도 복원 후 시각적으로 원본과 구별 할 수 없을 정도의 높은 품질(quality)을 가질 수 있고, 더욱이 현재 새로운 칼라 정지 영상 표준으로 채택된 JPEG-2000^[2]에서는 동일한 압축률에서 더욱 좋은 영상 품질을 가질 수 있게 되었지만, 원래의 데이터의 손실을 허용하지 않는 특정 분야들에서는 이러한 이미지 데이터의 손실이 문제가 될 수도 있다.

한편, 무손실 압축의 경우에는 이미지 데이터를 부호화하고 다시 복호화하는 과정을 거치더라도 이미지의 품질을 그대로 유지할 수 있다는 장점을 가지고 있으나, 압축 효율 면에서 그다지 뛰어난 성능을 보이지 못한다는 단점이 있다. 예를 들어, JPEG-LS^[3], CALIC^[4] 그리고 JPEG-2000의 무손실 모드에 이르기까지 여러 무손실 압축 기법들이 소개되고 있으나, 이러한 방식들에서의 압축 효율은 아직까지 1.5:1에서 3:1 정도에 불과하다. 이처럼 무손실 압축에서의 압축 효율의 향상은 아직도 더욱 개선시켜야 할 과제들이다.

이와 같은 문제점을 가지고 있는 무손실 압축 분야에서 특히 좋은 압축 효율을 기대할 수 있는 이미지의 형태는 적은 수의 색상에 대한 인덱스들로 표현되어지는 팔레트 기반 이미지 혹은 인덱스 이미지라 불리는 이미지 데이터들이다. 이러한 이미지들은 다른 파일 형태에서의 색상 값을 표현한 휘도 성분의 크기보다 더 작은 인덱스 값을 저장하기 위한 공간을 사용하는데, 제한된 적은 수의 색상을 가지는 이미지에 대하여 더 좋은 압축 효율을 얻을 수 있기 때문에 무손실 압축에서 자주 사용되고 있는 이미지 형태인 것이다.^{[5][6]}

컴퓨터에서 사용되는 많은 이미지들 중 컴퓨터 그래픽스에 의해 생성된 이미지는 보통 적은 수의 색상으로 구성 되고, 각 색상을 해당 인덱스로 매핑(mapping)하여 인덱스 이미지의 형태로 사용되어지고 있다. 이러한 특징을 갖는 팔레트 기반 이미지들은 인덱스 행렬(matrix of index)과 색상을 참조하기 위한 색상 테이블(color map) 혹은 팔레트(palette)로 표현되어진다. 가장 흔히 볼 수 있는 팔레트 기반 이미지의 대표적인 것으로는 컴퍼서브사(CompServe incorporated)에서 만든 GIF(Graphics Interchange Format) 파일 형식을 들 수 있다.^{[7][8]} 이 형식은 데이

터에서 검출되는 비트 패턴을 가변길이 부호로 대체하기 위해 사용하는 Lempel-Ziv 압축 알고리즘의 개선된 형태인 LZW(Lempel-Ziv-Welch) 압축 알고리즘을 사용한다. LZW 방식은 원래의 데이터에서 나타나는 새로운 패턴들로 테이블을 구성하면서, 다음에 나타나는 동일한 패턴들에 대해서는 해당 인덱스로 대체하여 압축을 해나가는 방식이다.^[9] 이러한 알고리즘을 사용하는 GIF파일 형식의 이미지는 실제 이미지의 표현을 위해 사용되기 보다는, 이미지 데이터의 압축을 통해 인공적으로 생성된 컴퓨터 게임의 유닛을 표현하거나 온라인에서의 컴퓨터 그래픽 이미지들을 전송 및 저장하기 위해 사용되고 있다.

이러한 인덱스 이미지는 색상 팔레트가 적절히 변하는 한, 어떠한 정보의 손실도 없이 다시 인덱스 될 수 있다는 것이 인정되어 왔다. 이것은 압축 성능을 개선하기 위해 적절한 인덱스의 재조정(re-indexing) 방법을 선택할 수 있게 한다. 따라서 본 논문에서는 인덱스 칼라 이미지(indexed color image)에서 인덱스들의 재조정을 통해 압축 효율의 향상을 기대할 수 있는 새로운 방법을 제안하고, 그러한 방법으로 재조정된 인덱스들을 허프만 부호화, LZW 부호화 등의 무손실 압축 기법들 보다 성능이 더 좋은 산술부호화(arithmetic coding)^[10]를 이용하여 압축한다. 다시 말해서, 인덱스 값들에 대한 출현 빈도수의 순위를 구해 원래의 인덱스 이미지를 순위 이미지로 변환한 후, 산술부호화를 통한 손실 없는 이미지 데이터의 압축 및 복원 방법을 제안한다. 그리고 제안한 방식을 GIF 방식, LPEG-LS 방식 그리고 산술부호화만을 사용한 무손실 압축 방식들과 함께 비교하여 분석 및 평가를 행한다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 무손실 압축을 행하는 LZW 및 산술부호화 그리고 인덱싱(indexing)의 기반이 되는 마코프 체인(Markov Chains)^[11]에 대하여 살펴보고, III장에서 인덱싱 기술을 토대로 새롭게 제안한 순위 이미지를 생성하는 방법에 대해서 서술한다. IV장에서는 제안한 방법에 대한 성능 평가를 수행하고, 마지막으로 V장에서 본 논문의 전체적인 결론을 맺는다.

II. 무손실 압축 기법

II.1. LZW(Lempel-Ziv-Welch)

1977년과 1978년에 이스라엘의 Abraham Lempel과 Jacob Ziv가 빈번하게 발생하는 순열들을 활용하

여 데이터를 효율적으로 압축하는 획기적인 방법을 제안하였고, 그 후 1984년에 Terry Welch가 이 알고리즘을 더욱 개선하여 LZW 압축 알고리즘을 제안하였다. CCIT의 V.42vis에 채택되어 있는 이 압축 기법은, 한 번에 한 문자만을 부호화하는 허프만 부호화(Huffman Coding)와는 달리, 입력 데이터 길이를 가변 길이로 하고 출력 부호의 길이를 고정한 기법으로서, 한 번 나타난 순열들을 테이블에 등록시키고 다음에 동일한 순열이 반복되어 발생할 경우 테이블을 참조하여 등록된 순열에 해당하는 인덱스만으로 데이터를 압축하는 방식이다.

이 방법은 많은 실험에서 장문의 영어 텍스트(text)를 압축할 때, 약 50% 정도의 놀라운 압축률을 보여주었다. 또한, 이러한 놀라운 압축률뿐만 아니라, 알고리즘의 단순성으로 인해 내부 연산(operation)의 횟수가 적기 때문에 압축과 복원에 있어서 빠른 실행이 가능하다. 더욱이 이 방법에서는 압축될 자료에 대한 사전 조사가 필요 없고, 압축에 사용된 테이블과 동일한 테이블이 복원 시에도 자동으로 생성되기 때문에 압축된 파일에 부가적인 정보를 첨부하지 않아도 된다는 장점을 가지고 있다. 이처럼 무손실 압축에서 획기적이고 효율적인 LZW 방식은 유닉스(UNIX)에서의 압축, 개인용 컴퓨터 분야에서의 ARC 압축 그리고 GIF 파일 형식의 이미지 압축 등 다양한 응용 분야에서 사용되어왔다.

GIF 파일 형식의 이미지는 현재 웹의 표준으로 JPEG과 함께 널리 사용되고 있으며, 위에서 언급한 LZW 알고리즘을 기본으로 한다. LZW 알고리즘의 특허권을 가지고 있는 유니시스(UNISYS)와 IBM 두 회사가 특허 만료일까지 LZW 알고리즘이 자유 소프트웨어로 사용되는 것을 거절하여 이제까지 GIF 형식

의 파일 사용 및 응용에 많은 제약을 가지고 있는 실정이었으나, 특허의 만료로 인한 활발한 연구 및 폭넓은 응용이 기대된다.

II.2. 산술부호화(Arithmetic Coding)

엔트로피(Entropy) 부호화의 대표적인 것으로는 허프만부호화 이외에도 산술부호화가 있다. 허프만부호화는 한 개의 심볼 또는 심볼열(symbol sequence)에 한 개의 부호를 할당하는 방법이지만, 산술부호화는 차례로 입력되는 심볼열에 대해 한 번에 한 심볼씩 점진적으로 부호를 계산해가는 방식이다. 즉, 전체 자료 스트림을 취하여 하나의 특정 부호어(code word)를 출력하는 방법으로, 사용되는 코드워드는 0과 1사이의 부동소수점 숫자이다. 일단 0에서 1까지의 범위에서 심볼들의 발생 확률(probability)에 대응하여 구간(interval)을 분할하고, 입력 심볼의 발생 확률에기 반한 비율을 사용하여 재귀적인 부분할을 통해 구간을 좁혀나간 후, 최종 분할 된 구간 내의 위치를 표시하는 2진 소수 값을 입력 심볼열에 대한 부호어로 정하는 것이다.

이와 같은 방법을 통해 심볼의 출현 빈도수가 좀 더 높은 것은 적은 수의 비트들로 표현하기 위하여 보다 폭 넓은 범위를 할당하고, 빈번하게 발생하지 않는 심볼에게는 좀 더 많은 비트들로 표현하기 위하여 좁은 범위가 할당한다. 따라서 적게 발생하는 심볼을 부호화할 때, 더욱 더 많은 비트가 필요하게 된다. 이와 같은 산술부호화에서는 입력 자료군이 커지면 커질수록, 숫자에 대한 좀 더 많은 디지트(digit)가 출력되고, 이 유일한 숫자는 복호화 될 때 손실 없이 정확한 입력자료 스트림을 출력하도록 부호화된다. 이 방법은 많은 실험 데이터에서 영어 문장에 적용시켰을 경우, 한 문자 당 2비트 가까이 압축하는 것이 가능하다.

이처럼 현존하는 무손실 부호화 방법 중 가장 이상적인 엔트로피 부호화가 가능한 산술부호화 기법이지만 실행속도가 느리고, 임의의 접근(random access)이 불가능하기 때문에 일부를 복원하기 위해서는 처음부터 복호화를 다시 해야만 하는 단점을 가지고 있다. 하지만 재귀적인 부분할 시에 부호 구간 내에서 실제로 부분할 되는 점은 그때까지 발생된 픽셀 빈도수에 의존하기 때문에 국부적인 이미지 특성에 잘 적응된다.^[6]

II.3. 마코프 체인(Markov Chains)

확률변수 X 가 시간(t)의 변화에 따라 변한다고 가

```

initialize table with single character strings
STRING = first input character
WHILE not end of input stream
    CHAR = next input character
    IF STRING + CHAR is in the string table
        STRING = STRING + CHAR
    ELSE
        output the code for STRING
        add STRING + CHAR to the string table
        STRING = CHAR
END WHILE
output code for string
    
```

LZW 압축 알고리즘

```

LOW = 0.0
HIGH = 1.0
WHILE not end of input stream
  get next symbol
  RANGE = HIGH - LOW
  HIGH = LOW+RANGE*high range of symbol
  LOW = LOW+RANGE*low range of symbol
END WHILE
output LOW
    
```

산술부호화 알고리즘

정한다면, 확률변수 $X(t)$ 는 시계열적으로 나열될 것이고, 이러한 확률변수 $X(t)$ 의 시계열적 집합 $\{X(t)\}$ 를 확률과정이라고 한다. 여기에서 $X(t)$ 를 시점 t 에서의 확률과정의 상태(state)라고 하고, 이러한 상태들의 집합을 상태 공간(state space)이라고 한다.

이러한 확률과정 $\{X(t)\}$ 는 어떤 시점 t 에서의 상태 $X(t)$ 가 t 이전 시점의 상태와 어떻게 관련되어 있는지에 따라 여러 가지 형태로 나눌 수 있는데, 이러한 형태들 중 대표적인 것들로는 가법과정(additive process), 정상과정(stationary process), 마코프과정(markov process) 등이 있다. 이들 중에서, 마코프과정은 시점 t 에서 $X(t)$ 의 상태가 알려져 있을 때 t 보다 이후의 시점 s 에서 $X(s)$ 의 확률법칙이 t 보다 이전의 시점 u 에서의 $X(u)$ 의 값에 관계없이 정해지는 확률과정을 말한다. 이것은 현재의 상태가 알려져 있을 때, 장래의 확률법칙이 과거의 경력에 어떤 영향을 받지 않는다는 것을 의미한다. 즉, 일단 어떤 상태가 주어졌다면, 다른 상태로 전이할 확률은 단지 현재의 상태에 의해서만 영향을 받는다는 것을 의미한다. 따라서 이전의 경력은 현재의 상태에서 다음의 상태로 전이할 확률에 별다른 영향을 미치지 못한다는 것이다.

이것을 조건부 확률에 의해 정의하면, 확률과정 $\{X(t)\}$ 가 있을 때 시점 $t_1 < t_2 < t_3 < \dots < t_n < t$ 에서,

$$P\{X(t) = x \mid X(t_1) = x_1, X(t_2) = x_2, \dots, X(t_n) = x_n\} = P\{X(t) = x \mid X(t_n) = x_n\} \quad (1)$$

이 성립하면 이 과정을 단순 마코프과정(simple Markov process)이라고 한다. 이러한 마코프과정은 상태 공간, 즉 시간이 연속적인(continuous) 형태로 분포하는 확률과정을 일컫는 말이다.

이러한 연속적인 형태와는 다르게 만약 어떤 마코프과정의 상태 공간이 이산적인(discrete) 형태로 분포

하는 경우, 이것을 마코프체인(Markov chain)^[11]이라고 부른다. 마코프과정과 유사하게 어떤 시점의 상태가 단지 직전의 상태에 의해서만 결정되는 마코프체인을 단순 또는 1차 마코프체인(simple or first-order Markov chain)이라고 하며, 이와는 달리 직전의 상태 외에도 그 이전의 사건에 의해서도 현재의 상태가 영향을 받는다면 2차 마코프체인(second-order Markov chain)이라고 한다.

조건확률 $P\{x_j(t) \mid x_i(t-1)\} = p_{ij}(t)$, 또는 이것을 위의 표기대로 나타낸,

$$P\{x_j(t) \mid x_i(t_n)\} = p_{ij}(t) \quad (2)$$

는 상태 i 에서 상태 j 로 천이할 확률을 나타내는데, 일반적으로 이 확률은 시간에 의존하지만 만약 그렇지 않다면, 위에서 논의한 바와 같은 단순 마코프과정에 대한 식 (1)을 만족하게 되고, 이것을 특히 동질적 또는 정상적 마코프과정(homogeneous or stationary Markov process)이라고 한다.

III. 제안한 방법

정지 영상의 무손실 압축을 위한 근거로는 크게 픽셀간 중복성(interpixel redundancy)과 부호화 중복성(coding redundancy)을 들 수 있다.^[10] 이들 중 인접 픽셀간 중복성을 제거하기 위하여, 본 논문에서는 조건부 확률을 이용한 마코프 체인을 응용한다. 마코프 체인은 M 개의 원소를 가지는 상태 집합과 각 상태 사이의 천이(transition)를 나타내는 $M \times M$ 천이행렬 T 로 이루어진다. II장의 II.3.절에서 살펴본 바와 같이 천이행렬 T 의 i 행 j 열 원소는 i 상태에서 j 상태로의 천이 확률을 담고 있다.^[11] 여기에서 각 상태를 색상 인덱스로 생각하면 색상 인덱스간의 천이 확률에 대한 행렬을 얻을 수 있다.

색상 인덱스 값이 256색의 팔레트 기반으로 구성된 이미지의 경우에 인덱스 값의 범위는 0~255의 범위에 놓여진다. 현재 픽셀에 해당하는 인덱스 값을 i 라고 할 때, 다음에 나올 색상 인덱스 값들 또한 0~255의 범위에 있기 때문에, 천이 행렬의 크기는 256×256 이 된다. 이 행렬을 사용하여 각 색상 인덱스를 기준으로 다음 색상 인덱스의 발생 빈도를 구하여 순위로 매길 수 있다. 이때, 높은 순위는 발생 빈도 역시 높으므로 순위 데이터만을 수집하면 결과적으로 부호화 중복성을 극대화할 수 있게 된다. 이러한 변화는 다음 단계에서의 엔트로피 부호화 성능을

향상시킬 수 있게 한다.

본 논문에서 제안한 새로운 기법은 이러한 마코프 체인의 특성을 응용하여 원래 이미지의 색상 인덱스 값 i 에 대하여, 다음에 나올 인덱스 값 j 의 발생 빈도를 기록하여 빈도수가 높은 순서대로 순위를 매기고, 새로운 순위 값으로 인덱스를 대체시켜 순위 이미지를 생성한다. 단, 동일한 발생 빈도수를 가질 경우에는 특정한 동점 처리 규칙을 적용하여 우선순위를 구해 나간다. 다음으로 이렇게 생성된 순위 이미지에 대해 산술 부호화를 행하여 압축 성능을 극대화한다. 이에 대한 알고리즘의 각 단계들은 다음과 같다.

- Step 1.** 2차원 이미지에서의 각 픽셀에 해당하는 인덱스 값들의 형태를 모두 1차원의 형태로 표현한다.
- Step 2.** 현재 픽셀에 해당하는 인덱스 값 i 에 대한 다음 픽셀에 해당하는 인덱스 값 j 의 발생 빈도수를 측정하여 변화계수표(*TCT*: Transition Count Table)를 식 (3)에 의해 생성한다.
- Step 3.** 생성된 *TCT*를 바탕으로 각 인덱스에 대한 모든 인덱스 값의 발생 빈도수를 내림차순으로 정렬하여 순위를 매긴 변화순위표 (*TOT*: Transition Order Table)를 작성한다.
- Step 4.** *TOT*를 참조하여 원래 이미지의 각 픽셀에 해당하는 인덱스 값을 다음 픽셀에 대한 발생 빈도의 순위 값으로 대체하여 순위 이미지(Rank Image)를 만든다.
- Step 5.** 만들어진 순위 이미지 데이터를 산술부호화를 적용하여 무손실 압축한다.

위의 알고리즘에 사용되는 이미지 데이터는 $M=256$ 개의 색상으로 구성된 인덱스 이미지로서, 각 인덱스 값 $0, 1, \dots, M-1$ 들이 이에 대응하는 색상 심볼 $0, S_1, \dots, S_{M-1}$ 을 나타낸다고 가정한다. step 1에서는 이와 같은 인덱스 이미지 데이터를 1차원 배열의 형태로 바꾸어 준다. step 2에서는 1차원의 형태의 이미지 데이터에 대해 변화계수표인 *TCT*를 생성하는데, *TCT*의 x 열(현재 인덱스)과 y 행(다음에 나오는 인덱스)은 256의 길이를 가진다. 그리고 이번 단계에서 사용되는 식 (3)에서의 $C(S_i, S_j)$ 는 1차원 공간상에서 서로 인접한 두 픽셀에 대한 인덱스 값의 변화를 카운트(count)한다는 것이다.

다음으로 step 3에서는 *TCT*에 대한 변화순위표인

*TOT*를 작성하는 데 만약, 같은 빈도수를 가질 때에는 *TCT*의 대각선 방향으로 현재 픽셀이 갖는 인덱스 값에서 가까운 거리에 있는 인덱스 값에 우선순위를 두는 방식으로 순위를 결정한다. 즉 식 (4)의 순위 규칙을 적용하여 식 (5)와 같이 동점 처리된 *TCT*를 구한 후에 *TOT*를 생성한다. 예를 들면, S_{11} 다음에 나오는 심볼이 $\{S_9, S_{10}, S_{11}, S_{12}, S_{13}\}$ 의 5개이고 각각의 빈도수가 $\{30, 200, 200, 35, 35\}$ 라고 하면, 심볼의 순위는 $\{4, 1, 0, 2, 3\}$ 이 된다. 다시말하면, 식 (3)에 의해 생성된 *TCT*의 i 행에서 같은 빈도수가 있을 때에는, 다음과 같은 식 (5)에 의해 $\widehat{TCT}_i(j)$ 를 구하여 정렬 시킴으로서 $TOT_i(j)$ 의 순위를 결정하게 된다.

이제 step 4에서 *TOT*를 참조하여 원래의 인덱스 이미지에 대한 순위 이미지를 작성하고, 최종적으로 step 5에서 산술부호화를 통한 무손실 압축을 수행하게 된다. 순위 이미지의 압축에 대해서 산술부호화를 선택한 이유는 압축할 심볼이, 치우침이 심한 확률분포를 가지고 또한 심볼이 가질 수 있는 경우의 수가 적을 때에는 허프만 부호화나 렘펠-지브 등의 부호화보다 더 좋은 압축률을 보이기 때문이다. 이는 영상의 전역적인 영역 특성에 적용할 뿐만 아니라, 영상의 국부적인 영역 특성에도 빠르게 적용하여 압축 효율이 10~15% 정도까지 향상된다.¹⁶⁾

$$TCT_i(j) = \sum_{j=0}^{M-1} C(S_i, S_j) \quad (3)$$

$$Priority\ rule\ (weight) = \frac{255 - |i - j|}{256} \quad (4)$$

$$\widehat{TCT}_i(j) = TCT_i(j) + \frac{255 - |i - j|}{256} \quad (5)$$

이렇게 효율적으로 압축된 데이터는 *TCT*와 원래 이미지의 색상 테이블과 함께 전송 혹은 저장되어 복원 시에는, 우선 산술복호화를 수행하여 손실 없이 순위 이미지를 먼저 복원한 후, 함께 전송된 *TCT*를 토대로 인덱스 이미지 데이터를 복원하고 색상 테이블의 참조에 의해 실제 이미지 픽셀의 칼라 값들을 출력할 수 있게 된다.

그럼 1은 본 논문에서 제안한 순위 이미지를 이용한 부호기(Encoder)의 압축 시스템을 보여준다. 그림에서처럼 제안된 압축 알고리즘은 입력된 인덱스 이미지에 대한 순위 이미지를 생성하기 위하여 *TCT* 및 *TOT*를 작성하고, *TOT*를 토대로 원래의 인덱스 이미

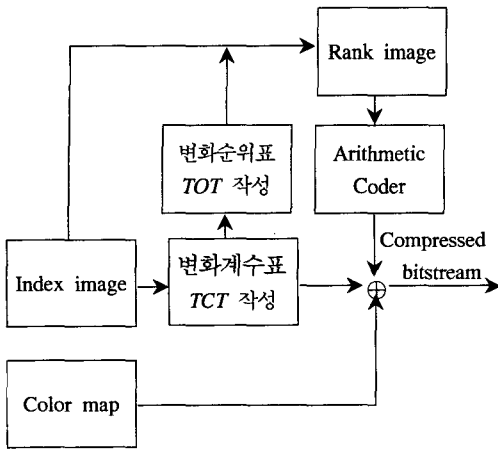


그림 1. 제안한 압축 시스템의 부호기

Figure 1. Encoder of the proposed compression system

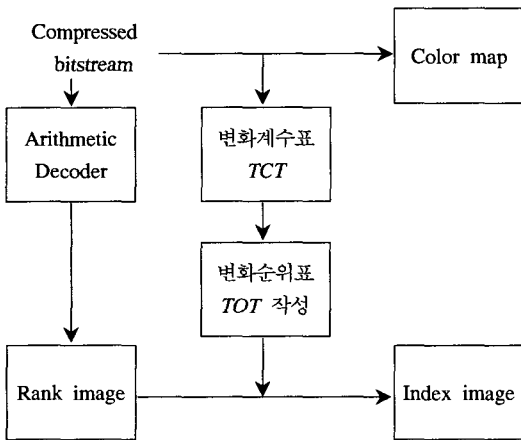


그림 2. 제안한 압축 시스템의 복호기

Figure 2. Decoder of the proposed compression system

지에 대한 순위 이미지를 생성한 후 최종적으로 순위 이미지를 산술 부호화를 하여 TCT와 함께 전송 혹은 저장하게 되는 것이다.

그림 2에서의 복호기(Decoder)는 부호기에서의 압축 과정을 역으로 처리함으로써 구현된다. 부호기를 통하여 압축된 비트열에 대해 산술복호화를 한 후, 함께 전송되었던 TCT를 참조하여 순위 이미지로 변환시키고 원래의 인덱스 이미지로 복원하면 된다. 물론, 무손실 압축을 행하였으므로 복원 이미지는 원래의 이미지와 완전히 동일한 화질 및 크기를 갖는다.

IV. 성능 평가 및 분석

실험에 사용된 “clegg”, “fractal”, “frymire”,

“ghouse”, “party8”, “pc”, “serrano” 등의 이미지들은 모두 8 bits의 팔레트 기반 이미지들이다. 본 실험에서는 압축 효율을 높이기 위하여 원래의 인덱스 이미지를 순위 이미지로 변환한 후 산술부호화를 적용한다. 새롭게 제안한 부호화 방식의 성능을 GIF 방식과 JPEG-LS 방식 그리고 산술부호화만을 사용하여 압축한 방식들과 비교한다. 압축 성능을 평가하기 위해서 압축률(CR: Compression Ratio)을 가지고 계산된 bpp (bits per pixel)를 구하여 판단하는데, 식 (6)과 식 (7)은 각각 압축률 CR과 bpp 를 구하는 방법을 나타낸 것이다.

$$CR = \frac{\text{Original image size}}{\text{Compressed image size}} \quad (6)$$

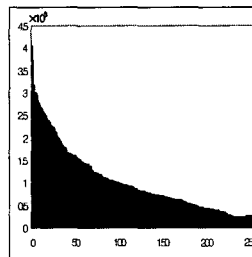
$$CR = 8 \times \frac{\text{Compressed image size}}{\text{Original image size}} = 8 \times CR \quad (7)$$

그림 3은 512×512 크기의 256 색상을 갖는 “lena”에 대하여, (a)는 원래의 이미지를 그림으로 나타낸 것이고 (b)는 원 이미지에 대해 본 논문에서 제안한 방식을 적용한 순위 이미지를 그림으로 표현한 것이다. 또한 (c)는 원래의 인덱스 이미지에 대한 색상 인덱스 분포를 나타내는 히스토그램을 표현하며 (d)는

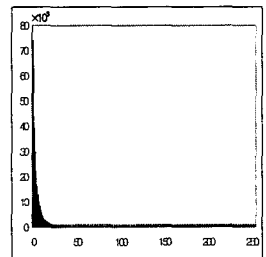


(a) Original image

(b) Rank image



(c) 히스토그램 (Original image)



(d) 히스토그램 (Rank image)

그림 3. “lena” 이미지와 히스토그램
Figure 3. Images and histograms of “lena”

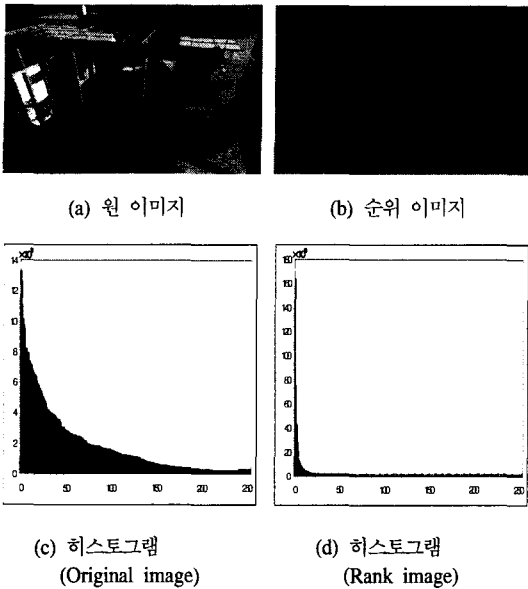


그림 4. "ghouse" 이미지와 히스토그램
Figure 4. Images and histograms of "ghouse"

순위 이미지로 변화시켰을 때의 히스토그램을 보인다. 그림 3의 (d)에서 알 수 있듯이, 히스토그램에서 도수의 분포가 높은 순위 쪽으로 매우 집중되어 있어 높은 압축 효율을 기대할 수 있다.

마찬가지로, 그림 4는 800×600 크기의 256 색상을 갖는 "ghouse"에 대하여, (a)와 (b)는 각각 원 이미지와 순위 이미지를 그림으로 표현한 것이며, (c)와 (d)에서는 각각 원 이미지의 인덱스 히스토그램과 제안한 방식의 순위 이미지로 변환된 인덱스 이미지를 나타낸 것이다.

표 1. 실험에 사용된 이미지들과 bpp
Table 1. The size and *bpp* of sample images

Image	GIF	
	size	<i>bpp</i>
clegg	510,324	5.699
fractal	336,650	6.948
frymire	413,955	2.681
ghouse	300,094	5.002
party8	8,068	0.429
pc	362,704	0.843
serrano	180,845	2.897
bike3	431,560	4.810
lena	214,255	6.535

표 2. 각 방식에서의 *bpp* 비교
Table 2. Comparisons with each method

Image	JPEG-LS	Arithmetic Coding	Proposed Algorithm
clegg	5.087	7.617	4.652
fractal	7.350	7.719	5.903
frymire	3.603	5.895	2.540
ghouse	4.539	7.157	4.759
party8	0.544	1.730	0.355
pc	1.436	1.918	0.789
serrano	3.416	7.208	2.637
bike3	4.981	7.090	4.113
lena	6.984	7.657	4.776

표 1은 실험을 위한 이미지 데이터를 나타내는 것으로서, GIF 형태의 파일들의 크기(단위: bytes)와 각 파일의 *bpp*를 기술한다. 표 1은 무손실 압축의 한 형태이므로 실험 이미지일 뿐만 아니라, 제안한 알고리즘과 비교되어질 자료이기도 하다. 표 2에서는 JPEG-LS 방식과 산술부호화 방식 그리고 제안한 방식으로 실험을 한 후, 압축된 파일의 *bpp*를 각각 기록하였다. 이때의 *bpp*는 압축되지 않은 색상 정보를 포함하여 계산된 것이다. 특히 제안한 방식에서의 *bpp* 계산에는 압축된 이미지의 크기와 색상 정보 그리고 TCT나 TOT 중 하나의 정보를 포함하여야 하는데 본 실험에서는 TCT 크기를 취하여 저장 및 전송하였다. 이 TCT 정보는 이미지의 특성에 따라 원래의 이미지 데이터의 크기에 비해 상대적으로 많은 수의 bits(대략 1KB~56KB)를 차지하게 되는 경우도 발생하는데, 원래의 이미지 크기가 100KB가 넘지 않을 경우에는 비효율적일 수도 있다.

하지만 표 1과 표 2의 실험 결과에서 보면 알 수 있듯이, 일반적인 이미지들에 대해 무손실 압축 실험을 해보았을 때, 제안한 순위 이미지 적용 방식이 GIF 및 JPEG-LS 그리고 산술 부호화 방식보다 평균적으로 비트율(bit rate)을 15%, 19%, 43%까지 각각 감소시킨다는 것을 알 수 있었다. 또한 자연의 칼라 이미지(natural color image)인 512×512 크기의 "lena"와 781×912 크기의 "bike3"를 인덱스 이미지로 변환한 후, 각 방식에서의 *bpp*를 측정하더라도 제안된 방식의 압축 효율이 뛰어나다는 것을 알 수 있다.

그림 5는 이미지 압축 실험 결과인 표 1과 2를 토대로 각 실험 방식에서의 *bpp* 값을 그래프로 표현한 것이다. 여기에서 AC는 Arithmetic Coding 방식을

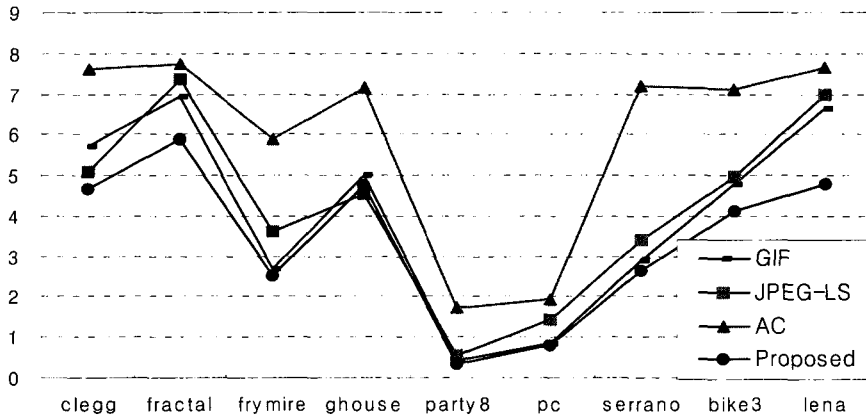


그림 5. 실험 결과에서의 bpp 그래프
 Figure 5. The graph of bpp in simulation results

의미한다. 본 논문에서 제안한 방식을 통해 데이터의 중복성을 극대화 시켰으므로 산술부호화만을 수행한 방식보다 제안한 방식이 훨씬 좋고 나머지 방식들과 비교해서도 압축 효율이 뛰어난을 그림 5를 통해 알 수 있다.

V. 결론

본 논문에서는 칼라 이미지를 그레이스케일이나 원 영상 파일인 raw 형태로 변환할 필요 없이, 팔레트 기반의 256색상을 갖는 이미지를 손실 없이 보다 효율적으로 압축할 수 있는 새로운 알고리즘을 제안하였다. 이를 위해 인덱스 이미지에서의 각 인덱스 정보에 대한 두 인접한 픽셀간의 인덱스 빈도수를 계산하여, 원래의 인덱스 이미지를 순위를 매긴 순위 이미지로 바꾸어 데이터의 중복성을 더 많이 가지게 하였다. 그리고 그 결과를 가지고 효율적인 산술부호화를 적용하게 되었다. 실험 결과, GIF 방식과 JPEG-LS 방식 그리고 Arithmetic Coding 방식보다 평균적으로 각각 0.592, 0.743, 2.607 정도의 bpp를 보이는 개선된 압축 성능이 있음을 확인하였다.

본 실험에서 관찰하였듯이, 인덱스 이미지를 순위 이미지로 변화시켰을 때 TCT에 관련된 부가정보 또한 저장 및 전송을 하여야 한다. 이는 그 크기가 이미지의 크기 및 칼라 정보에 따라서 달라지므로 크기가 충분히 크지 않은 이미지에 대해서는 비효율적일 수도 있다. 따라서 TCT 정보를 전송하지 않아도 될 순위 이미지로의 변환에 있어서, 적응적인 방법이 적용되어야 하는 연구가 있어야 할 것이다.

참고 문헌

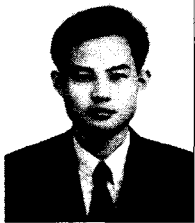
- [1] I. Pitas, *Digital Image Processing Algorithms and Applications*, Jone Wiley & Sons, Inc., 2000.
- [2] "Information Technology - JPEG-2000 image coding system", JTC1/SC29/WG1 FCD15444-1, Mar 2000.
- [3] Marcelo J. Weinberger, Gadiel Seroussi, Fellow and Guillermo Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS" *Image Processing, IEEE Transactions on*, Volume: 9 Issue: 8, pp. 1309-1324, Aug. 2000
- [4] Xiaolin Wu, Nasir Memon "Context-Based, Adaptive, Lossless Image Coding" *IEEE Transactions on Communications*, Volume: 45 Issue: 4, pp.7-444, April 1997,
- [5] W. Zeng, J. Li and S. Lei, "An efficient color re-indexing scheme for palette-based compression," In proc. IEEE Int. Conf image Processing, pp. 476-479, Sep 2000.
- [6] Randy Crane, *A Simplified Approach to Image Processing*, Prentice-Hall PTR, August, 1996.
- [7] James D. Murray and William Vanryper, *Graphics File Formats*, OReilly & Associates, INC., April 1996.
- [8] C. Wu and D. Irwin, *Emerging Multimedia Computer Communication Technologies*,

Prentice-Hall PTR, 1998.

- [9] Mark Nelson, *The Data Compression Book*, M&T Publishing, 1992.
- [10] Khalid Sayood, *Introduction to Data Compression*, San Francisco, Morgan Kaufmann, 2000.
- [11] Leon-Garcia Alberto, *Probability and Random Processes for Electricl Engineering*, 2nd edition, Addison-Wesley Publishing, May 1994.

유 강 수(Kang-Soo You)

정회원

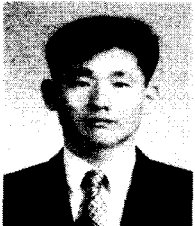


1991년 2월 : 전북대학교 컴퓨터 공학과 공학사
 1994년 2월 : 전북대학교 컴퓨터 공학과 공학석사
 2000년 3월~현재 : 전북대학교 영상공학과 박사과정

<관심분야> 영상신호처리, 영상검색, 멀티미디어

이 한 정(Han-Jeong Lee)

정회원



1992년 2월 : 호원대학교 전자계산학과 학사
 2003년 8월 : 전북대학교 영상공학과 공학석사
 2004년 3월~현재 : 전북대학교 영상공학과 박사과정

<관심분야> 영상신호처리, 영상검색, 멀티미디어

장 의 선(Euee S. Jang)

정회원



1991년 2월 : 전북대학교 컴퓨터 공학과 공학사
 1994년 6월 : 미국 버팔로 뉴욕 주립대학 공학석사
 1996년 2월 : 미국 버팔로 뉴욕 주립대학 공학박사
 1996년~2002년 : MPEG SNHC

Chair, 삼성종합기술원 책임연구원
 1997년~2000년 : MPEG-4 Project Editor
 2000년~현재 : IEEE ICME member, MPEG Forum
 전문위원, 한국방송공학회 편집위원, 한양대학교 정보통신대학 조교수

<관심분야> 컴퓨터그래픽스, 고화질 미디어, 3DAV, 무손실 영상 압축, MPEG/JPEG

곽 훈 성(Hoon-Sung Kwak)

정회원



1970년 : 전북대학교 전기공학과 공학사
 1979년 : 전북대학교 전자공학박사
 1981년~1982년 : 미국 텍사스주립대학 연구교수
 1994년~1995년 : 국가교육연구전

산망 추진위원
 1997년~1998년 : 전주영상축전조직 위원장 및 전북대학교 영상산업특성화사업단장
 1998년 : 과학기술법령정비정책위원
 1999년~현재 : 조달청우수제품(정보통신)심사위원
 1997년~현재 : (사)영상산업연구센터대표
 현재 : 전북대학교 전자정보공학부 컴퓨터공학 교수 및 영상공학과(대학원) 주임교수

<관심분야> 영상신호처리, 인공지능, 컴퓨터비전, 멀티미디어 등