

교류 계수 분할 압축에 의한 JPEG 정지영상 압축 효율 향상 기법 연구

안영훈[†] 신현준[†] 위영철[†]아주대학교[†]

{amouse,joony,ycwee}@ajou.ac.kr

On Improving Compression Ratio of JPEG Using AC-Coefficient Separation

Younghoon Ahn[†]Hyun Joon Shin[†]Young Cheul Wee[†]Ajou University[†]

요약

본 논문에서는 JPEG 표준의 엔트로피 부호화 과정을 개선하여 압축률을 향상시키기 위한 방법을 제안한다. JPEG은 비교적 높은 화질 대비 압축률을 가지며, 특히 부호화와 복호화가 매우 효율적이어서 많은 응용에 사용된다. 본 논문에서는 JPEG의 이산 코사인 변환과 양자화, 재정렬 등의 과정을 거친 정보에 0, -1, 1 등의 작은 값들이 연속적으로 나타난다는 관찰 결과에 착안하여 이들을 분리하고 JPEG의 방식 보다 효율적인 방법으로 엔트로피 부호화 함으로써 압축률을 향상시키는 방법을 제안한다. 이와 같은 방법을 통하여 다양한 화질의 영상에서 JPEG보다 높은 압축률을 얻을 수 있음을 실험적으로 확인하였다.

Abstract

In this paper, we introduce a novel entropy coding method to improve the JPEG image compression standard. JPEG is one of the most widely used image compression methods due to its high visual quality for the compression ratio, and especially because of its high efficiency. Based on the observation that the blocks of data fed to the entropy coder usually contain consecutive sequences of numbers with small magnitudes including 0, 1, and -1, we separate those sequences from the data and encode them using a method dedicated to those values. We further improve the compression ratio based on the fact that this separation makes the lengths of blocks much shorter. In our experiment, we show that the proposed method can outperform the JPEG standard preserving its visual characteristics.

키워드: 영상 처리, 영상 압축

Keywords: Image Processing, Image Compression

1 서론

JPEG(Joint Picture Expert Group)은 실제 응용에서 가장 널리 사용되는 정지 영상 손실 압축 기법이다. JPEG이 등장한 이후 JPEG2000 등과 같은 보다 높은 성능을 보이는 압축 방법들이 제안되었지만 JPEG은 부호화 및 복호화 과정이 비교적 간단하여 계산량이 비교적 적고 압축률에 비해 높은 화질을 얻을 수 있다는 장점에 기인하여 현재 높은 압축률과 빠른 부호화 요구되는 인터넷을 통한 영상 전송 등의 응용분야나 높은 화질을 유지해야 하는 고화질 영상 촬영 등의 응용 분야에 널리 사용되고 있다[1, 2].

JPEG 압축 기법은 크게 주파수 대역(frequency domain)에서의 양자화(quantization)와 엔트로피 부호화(entropy coding)의 두 단계로 나누어 설명할 수 있다[3, 4]. 주파수 대역에서의 양자화는 공간적으로 분포되어 있는 영상 정보를 이산 코사인 변환(discrete cosine transform)을 통하여 주파수 대역으로 변환하고 주파수 대역별로 적절한 정도의 양자화를 가하는 과정이다. 이는 사람 눈이 고주파 대역의 변화에 비교적 덜 민감하다는 데[5] 착안하여 고주파 영역에 더 강한 양자화를 가하여 정보의 화질을 적절히 유지하면서 엔트로피를 감소시키는 것을 목적

으로 한다[6]. 즉, JPEG에서 영상의 손실을 가져오는 유일한 부분이 양자화 과정이며, 이는 향후 적용될 엔트로피 부호화를 고려하여 엔트로피 부호화에 적절한 형태로 정보를 가공하는 과정으로 볼 수 있다.

JPEG 표준의 엔트로피 부호화는 크게 런 길이 부호화(run-length coding)와 허프만 부호화(Huffman coding)로 이루어져 있다. 런 길이 부호화는 연속된 0의 개수와 0 이후에 처음 나타나는 수를 표현하는데 필요한 비트 수(number of bits)를 동시에 기록함으로써 연속된 0의 길이에 따라서 저장되는 정보의 양을 줄이는 방법이다. 이와 같은 "0 런 길이"와 길이 부호(length code)를 허프만 부호화를 통하여 압축하여 다시 한번 정보의 양을 줄인다. 즉 JPEG의 엔트로피 부호화는 연속적인 0의 길이가 길면 높은 압축률을 보이며, 따라서 양자화 과정 및 재정렬(reordering) 과정이 연속적인 0의 길이를 증가시키는 방향으로 최적화 되어 있다.

본 논문에서는 이와 같은 런 길이 부호화와 허프만 부호화를 개선하여 JPEG의 압축 성능을 보다 향상시키기 위한 방법을 제안한다. JPEG은 양자화 및 재정렬 단계를 통하여 비교적 많은 수의 연속된 0이 나타나도록 설계되어 있지만 실제 데이터에서는 상당 부분 -1, 0, 1 등의 작은 값들이 혼재되어 나타나는

경우가 많다. 이런 경우에 JPEG 표준 엔트로피 부호화 방법으로 최상의 압축률을 기대하기 어렵다. 특히 0과 -1이나 1 등의 작은 값들이 번갈아 가며 나타나는 극단적인 경우에는 런 부호화에 따른 성능 향상을 기대하기 어렵다. 따라서 본 논문에서는 실제 JPEG 영상에서 자주 나타나는 0, -1, 1로 이루어진 연속적인 값(01스트림)을 다른 정보에서 분리하고 이들을 따로 부호화 함으로써 압축률을 향상시키는 방법을 제안한다. 또한 01스트림을 분리하여 부호화 함에 따라서 각각의 부호화 단위(블럭)의 정보 양이 줄어드는 점을 이용하여 압축률을 보다 더 향상시키기 위한 방법을 제시한다.

본 연구 이전에도 JPEG 표준의 압축률을 향상시키기 위한 연구들이 진행되었다. 대표적인 연구로는 JPEG의 양자화 표를 최적화 하여 압축률 대비 영상의 화질을 향상시키기 위한 연구[7, 8]나 엔트로피 부호화를 개선하는 방법[9, 10] 등을 들 수 있다. 본 연구와 관련된 엔트로피 부호화를 개선하기 위한 연구 중 Tu 등이 제안한 방법[9]은 연속적인 0의 길이와 값을 따로 부호화하여 압축 성능을 높인다. 특히 0의 길이와 실값과의 관계를 상관 관계를 고려하여 2진 적응적 산술 부호화(binary adaptive arithmetic coding) 계산량의 증가를 감소하면서 압축 성능을 향상시켰다. Lakhani는 본 논문에서와 유사하게 다양한 방법들로 블럭내의 교류 계수를 몇 개의 영역으로 분리하고 각각을 부호화 하여 JPEG의 압축률을 높이기 위한 방법을 제안하였다[10]. 본 논문의 특징은 블럭을 일반적인 교류 계수, 압축이 용이한 01스트림, 블럭 최후의 연속적인 0 등의 세 부분으로 분리하고 특히 01스트림에 대한 부호화 방법을 최적화 함으로써 보다 높은 압축률을 얻기 위한 방법을 제안한다.

본 논문의 2절에서는 JPEG에서 사용되는 압축 방법, 특히 엔트로피 부호화 방법에 대하여 소개하고, 3절에서는 본 논문에서 제안하는 01스트림을 이용한 부호화 방법과 짝이진 블럭에 대응하여 압축률을 향상시키는 방법을 제시한다. 4절에서는 제안한 방법의 성능을 실험결과를 통하여 알아보고 5절에서는 결론을 맺는다.

2 JPEG 엔트로피 부호화

JPEG에서는 영상의 밝기(luminance)와 색상(chroma)를 분해하여 각각을 독립적으로 부호화 한다. 이때 사용되는 방법이 거의 유사하기 때문에 본 논문에서는 서술의 편의를 위해 밝기 중심으로 부호화 과정을 서술한다. JPEG에서는 먼저 주어진 영상을 8x8 화소의 일정한 크기를 가진 매크로 블럭(macro block)으로 분할한다. 각 블럭은 직류 성분을 제외하고는 독립적으로 처리되고 저장된다. 영상을 주파수 대역으로 구분하기 위해 JPEG에서는 먼저 각 블럭에 대하여 이산 코사인 변환을 적용한다. 이산 코사인 변환의 결과로 블럭 전체의 평균적인 밝기를 나타내는 직류 성분과 주파수 대역별로 교류 성분의 계수(coefficient)를 얻을 수 있다. JPEG에서는 이와 같이 얻은 직류 및 교류 계수를 주파수 대역 별로 각각 양자화 한다. 사람이 인지하기 어렵다고 알려져 있는 고주파 성분을 보다 강하게 양자화 한다. 양자화가 된 값은 일반적으로 점차 작아지는 경향성을 갖게 하기 위해 저주파에서 고주파 순으로 재정렬한다. 이와 같은 과정을 통하여 얻은 값은 경우에 큰 값으로 시작하여 점차 작아지며 블럭의 뒷부분은 대부분 0 혹은 매우 작은 값이 연속적으로 나타나는 경우가 많다[4].

JPEG의 엔트로피 부호화는 이와 같은 특성을 반영하여 정보

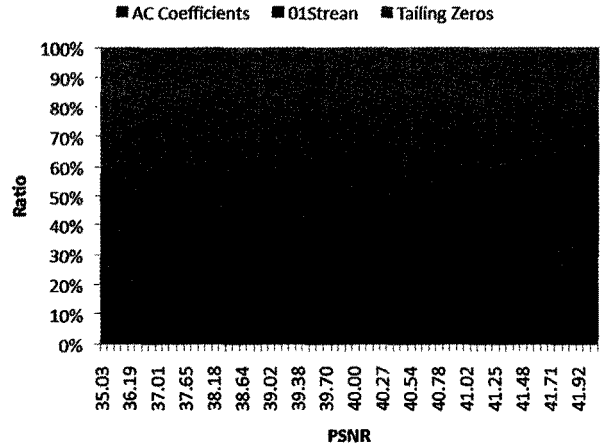


그림 1: 화질에 따른 후반 0스트림과 01스트림의 구성 비

를 압축한다. 전술한대로 JPEG의 엔트로피 부호화는 허프만 부호화와 런 길이 부호화를 적절히 혼합하여 사용한다. 재정렬을 통해 얻은 교류 계수 값은 블럭 뒤 부분에 매우 작은 값들이 밀집되어 있다. 따라서 JPEG에서는 연속된 0을 각각 저장하는 대신 연속 0의 개수를 저장하는 방법을 사용한다. 즉, 매년 0이 아닌 수가 나타날 때 마다 바로 전에 나타난 연속적인 0의 개수를 동시에 기록한다. 이때 0의 개수와 0이 아닌 수의 비트 수를 정보의 단위로 하여 허프만 부호화하여 정보의 양을 줄인다. 또한 블럭의 최후에 존재하는 연속적인 0(tailing zeros; 후반 0스트림)은 기록하지 않고 블럭의 끝을 나타내는 표식을 기록함으로써 저장되는 정보를 줄인다. 따라서 블럭에 연속하는 0이 많을 수록 압축률이 증가하며 후반 0스트림의 길이가 압축률에 상당한 영향을 미친다.

3 교류 계수 부호화 분할 기법

본 연구는 JPEG의 이산 코사인 변환과 양자화 과정을 그대로 유지하며 엔트로피 부호화 과정을 변경하여, JPEG 표준이 가지는 시각적인 특징(visual qualitative characteristic)을 그대로 유지하면서 압축률을 향상시키기 위한 방법을 제안한다. 본 논문에서 제안하는 방법은 먼저 실제 JPEG의 방법대로 양자화되고 재정렬된 정보에는 연속된 0 보다는 0, -1, 1 등의 작은 값들이 혼재되어 있는 경우가 많다는 점에 착안한다. 그림 1은 압축 성능 측정에 널리 사용되는 Lenna 영상을 다양한 화질로 JPEG을 이용하여 압축하고 구성 비율 화질(PSNR)에 따라 표시한 그래프이다. 먼저 녹색 부분은 블럭 전체에 대한 후반 0 스트림의 비율을 보여준다. JPEG에서 의도한 바와 같이 낮은 화질에서는 블럭의 85% 정도가 후반 0 스트림임을 확인할 수 있다.

본 연구에서는 추가적으로 0, 1, -1 등의 작은 값으로 이루어진 스트림이 후반 0스트림 앞에 존재할 것으로 예상하고 이와 같은 정보를 JPEG보다 정교한 방법으로 구분하여 압축함으로써 압축률을 향상시키기 위한 방법을 제안한다. 그림 1의 붉은색으로 표시된 부분은 후반 0스트림 앞에 나타나는 0과 1, -1로만

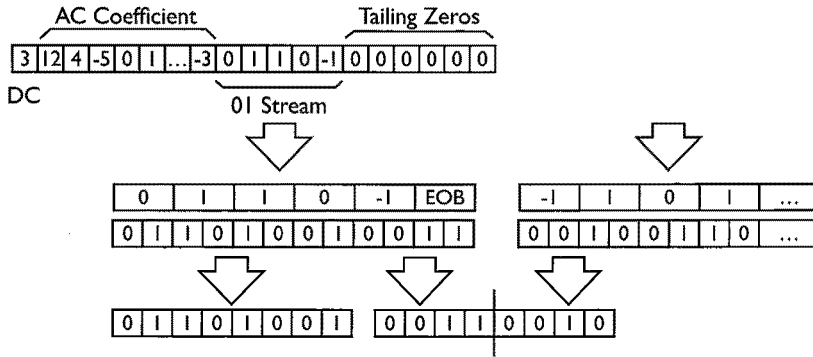


그림 2: OI 스트림의 구성과 바이트 열 생성

이루어진 연속적인 계수의 비율을 나타낸다. 그림에서 전반적으로 후반 0스트림을 제외하면 0, 1, -1로만 이루어진 연속적인 계수들이 전체의 절반 정도가 되는 것을 확인할 수 있다. 따라서 이와 같이 특정한 성질을 가지는 정보를 따로 압축하는 방법을 통하여 압축률 향상에 도움을 줄 것으로 예상할 수 있다.

이와 같은 관찰 결과에 따라 본 논문에서는 블록의 맨 끝에 나타나는 0, 1, -1만으로 구성되는 연속적인 계수들(OI 스트림)을 블록에서 분리하고 독립적으로 부호화하여 압축률을 향상시키기 위한 방법을 제안한다. 또한 OI스트림을 분리함에 따라서 나머지 블록의 길이가 매우 짧아진다는 점에 착안하여 JPEG의 성능을 더욱 향상시키기 위한 방법을 제시한다.

3.1 OI 스트림 부호화

본 논문에서는 독립적인 압축의 대상이 되는 OI스트림은 후반 0스트림을 제외한 블록 최후에 나타나는 0, -1, 1만으로 구성된 연속적인 계수들로 정의한다. OI스트림을 독립적으로 부호화하기 위해 본 논문에서는 먼저 JPEG의 각 블록에서 OI스트림을 추출한다. OI스트림의 추출은 교류 계수를 순차적으로 검사하여 0, 1, -1을 제외한 계수 중 마지막 계수를 찾고 이후의 계수들을 OI 스트림으로 검출할 수 있다. 각 블록당 추출된 OI 스트림은 역시 후반 0스트림을 포함하고 있지 않기 때문에 각각의 블록에 대한 OI스트림의 끝에는 블록의 종결을 나타내기 위한 표식(end of block marker, EOB)를 추가한다(그림 2의 상단 참조).

OI스트림을 구성하는 0, 1, -1의 계수와 블록 종결 표식을 부호화 하기 위해서는 각 계수 당 2비트를 할당하여야 한다. 하지만 2비트 단위로 엔트로피 부호화를 수행하면 높은 압축률을 기대하기 어렵다. 본 연구에서는 OI스트림의 계수들을 순차적으로 연결하는 방법으로 다양한 크기(2비트, 4비트, 8비트, 16비트 등)의 단위 정보를 만들었다. 그리고 다양한 크기를 가지는 정보 단위에 대하여 허프만 부호화를 수행하여 압축률을 평가하였다. 결과적으로 8비트 단위가 실험적으로 가장 높은 성능을 보임을 확인하였고, 따라서 본 논문에서는 계수들을 8비트 단위로 연결하여 허프만 부호화 한다.

OI스트림의 길이가 n 일 경우 독립적으로 부호화된 스트림의 길이는 기본적으로 $2n$ 비트가 된다. 이들을 8비트 단위로 엔트로피 부호화 하기 위해 본 논문에서는 먼저 4개의 계수를

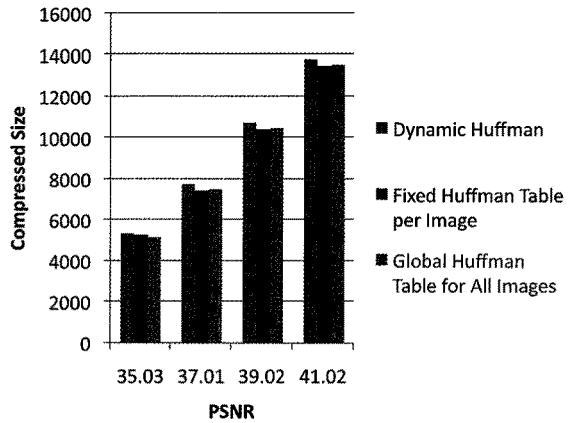


그림 3: 허프만 부호화 방식에 따른 화질별 압축 크기 비교

순차적으로 1 바이트를 구성한다. 이때 블록에 따라 OI스트림의 길이가 다르기 때문에 블록별로 바이트들을 구성하면 블록당 평균적으로 4비트를 낭비하게 된다. 따라서 본 논문에서는 모든 블록의 OI스트림을 순차적으로 연결하여 바이트 열을 구성한다. 예를 들어 첫번째 블록의 OI스트림의 길이가 종결 표식을 포함하여 6이라면 마지막 2개의 계수(EOB 포함)와 다음 블록의 2개의 계수를 순차적으로 조합하여 바이트를 구성하여 비트의 낭비를 최소화 한다(그림 2의 하단 참조).

블록 전체에 대한 OI스트림의 비트열은 전술한 바와 같이 허프만 부호화를 이용하여 부호화 한다. 이는 JPEG의 중요한 특징인 빠른 부호화 및 복호화가 가능하다는 장점을 유지하기 위해서이다. 특히 복호화 성능을 보장하기 위해 정적 부호화표를 이용한 허프만 부호화를 수행한다. 정적 허프만 부호화표는 영상마다 따로 정의된 경우에 최적의 성능을 보이는 것으로 알려져 있다[11]. 하지만 영상마다 허프만 표를 추가하는 가중 부담(overhead)을 방지하기 위해 본 논문에서는 다양한 영상에 대하여 한 개의 최적 허프만 부호화표를 생성하고 부호화와 복호화 과정에 적용하여 추가적인 정보가 발생하는 것을 방지하였다. 본 연구에서는 동적 허프만, 영상 별 정적 허프만 부호화 표, 단일 허프만 부호화 표를 사용하여 각종 영상을 압축하여 압축률을 비교하였다(그림 3 참조). 동적 허프만 부

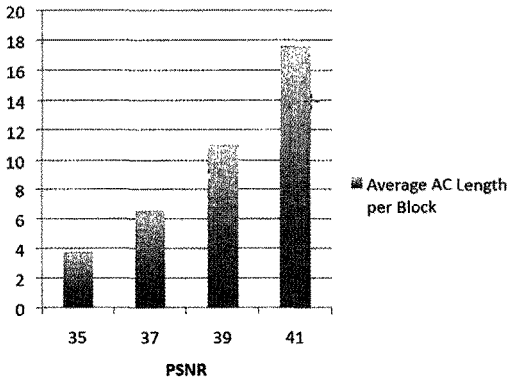


그림 4: 화질에 따른블럭 당 평균 교류 계수 개수

호화를 사용하는 경우는 내부적인 허프만 부호화표가 적절한 수준으로 수립하는데 정보가 필요하기 때문에 정교하게 만들어진 정적인 허프만 부호화표를 사용하는 방법에 비해 압축률이 낮다. 영상 환경에 대한 최적 허프만 부호화표를 사용하는 경우는 많은 영상을 통해 얻은 허프만 부호화표를 사용하는 경우에 비해 압축률의 이익 보다 허프만 부호화표를 저장하기 위한 추가 부담이 많아진다. 따라서 본 논문에서는 화질을 다양하게 변화시킨 많은 수의 영상에 대하여 최적의 허프만 부호화표를 생성하고 이를 이용하여 01스트림을 부호화한다.

본 논문에서 사용한 허프만 부호화 표는 표 1에 나타나있다. 표의 행들은 8비트로 조합된 정보의 상위 4비트의 값을 나타내고 각 열은 하위 4비트를 나타낸다. 각각의 칸의 값은 상위 4비트와 하위 4비트를 조합한 8비트의 값에 해당하는 허프만 부호를 나타낸다. 이때 0에는 1(이진수로 01), 1에는 2(10), -1에는 0(00), 블록 종결 표식에는 3(11)의 값을 지정하여 2비트 숫자를 만들고 이들을 조합하여 8비트 정보를 구성하였다. 이들 중 0으로 나타난 부분은 실제 데이터 상에서 나타나지 않는 부분이다. 예를 들어 7X에 해당되는 부분은 0 다음 블록 종결 표식이 나타나는 경우로 이때 0은 후반 0 스트림에 포함되어야 하기 때문에 이와 같은 경우는 나타나지 않는다.

3.2 교류 길이 표시

JPEG 표준에서는 후반 0스트림을 생략하기 위해 블록의 최후에 나타나는 0이 아닌 계수 다음에 블록 종결 표식을 추가하여 이후의 계수들이 모두 0임을 표시한다. 이때 블록 종결 표식은 일반적으로 4 비트(이진수로 1010)를 사용하여 부호화 된다. JPEG에서는 이와 같은 블록 종결 표식을 이용하면 블록 끝부분에 나타나는 연속적인 0을 4비트로 표현할 수 있기 때문에 압축률 향상에 상당한 도움이 된다. 하지만 본 논문에서는 각 블록에서 01스트림을 제외한 교류 길이와 공간적 상관관계가 있다는 점에 착안하여 압축률을 더욱 향상시킨다.

그림 4에서는 화질에 따라블럭 별로 01스트림과 후반 0스트림을 제외한 나머지 계수들의 개수의 평균 길이를 보여준다. 그림에서 실제로 기존 JPEG 방식을 이용하여 부호화 되어야 할 계수들의 개수가 상당히 적고, 화질과 직접적인 관계가 있다는 것을 확인할 수 있다. 따라서 본 논문에서는 일반적인 JPEG에 사용되는 블록 종결 표식을 추가하는 대신 블록의 교류 계수가 시작하는 부분에 교류 계수의 개수를 증분 부호화 방식을



그림 5: 실험 영상:(왼쪽 위에서 오른쪽으로) Lenna, Airplane, Peppers, Sail Boat, Baboon, Barbara

사용하여 기입함으로써 압축률을 향상시킨다.

증분 부호화는 매 블록의 계수 길이를 저장할 때 바로 앞의 블록의 계수 길이와의 차이를 저장하는 방법이다. 일반적으로 연속적인 블록은 유사한 성질을 가지는 경우가 많기 때문에 교류 계수의 길이 역시 유사한 성질을 가지므로 증분 부호화를 이용하여 교류 길이 표시의 엔트로피를 감소시킬 수 있다. 이와 같이 얻어진 차이값(증분)은 대부분 0과 가까운 값으로 나타나기 때문에 이 정보를 효과적으로 저장하기 위해 허프만 부호화를 적용한다. 증분 부호화 된 교류 길이의 범위는 -63에서 63으로 매우 제한적이기 때문에 필요한 허프만 부호화표가 비교적 작다. 따라서 본 연구에서는 영상 마다 최적의 허프만 부호화표를 생성하여 저장한다.

4 실험 결과

본 논문에서 제안하는 교류계수 분할 기법의 성능을 실험하기 위한 방법은 다음과 같다. 실험에는 다양한 영상 특성을 가지는 그림 5의 6개의 표준 영상을 사용하였다. 특히 유사한 기존 연구와의 공정을 비교를 위해 전술한 바와 같이 영상의 밝기 채널(luminance channel)만을 추출하여 실험하였다. 색상(chroma channel)의 경우에도 부표본추출(subsampling) 과정을 제외하면 동일한 방식이 사용되기 때문에 같은 특징을 보일 것으로 예상할 수 있다. 또한 JPEG과의 정확한 성능 비교를 위해 엔트로피 부호화를 제외한 모든 부분은 JPEG 표준과 동일하게 하였다. 화질을 다양하게 조절하기 위해 널리 쓰이는 방식대로 표준 양자화 표에 화질 설정에 따라서 적절한 상수

표 1: 01 스트림을 위한 허프만 부호화 표(모든 값은 16진수임)

| | | | | | | | | | | | | | | | | |
|----|------|----|-----|------|------|----|------|--------|------|------|------|------|------|------|------|------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0X | 30D | D | 2E7 | BE | 1F5 | 6C | 2 | 0 | 2E4 | 8 | 370 | BC | 50 | 3A4 | 31 | 191 |
| 1X | 1BF | 1D | 1D7 | 67 | 59 | 1A | 58 | 0 | 1DA | 4E | 1DB | 68 | 0 | 0 | 0 | 0 |
| 2X | 2E3 | F | 376 | 506 | 1FB | 63 | 1F4 | 0 | 30C | 6 | 309 | 65 | 63 | 3E3 | 64 | 65C |
| 3X | 39 | 1F | 33 | C3C | 1FE | 9 | 1F9 | 0 | 2E | 1A | 37 | A0A | 1F14 | AC3 | 1F13 | 65F |
| 4X | 1FA | 1E | 1BC | 140 | 1B | 11 | 42 | 0 | 1DE | 1F | 1D5 | 170 | F9 | 5F | EC | 196 |
| 5X | 57 | 4 | 5D | 29 | 1C | 1 | 1E | 0 | 5E | 5 | 51 | 2A | 0 | 0 | 0 | 0 |
| 6X | 1BD | 4D | 1FF | 159 | 43 | 12 | 40 | 0 | 1B9 | 1C | 4 | 6B | FB | 60 | EE | 507 |
| 7X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8X | 2B1 | A | 30A | DB | 1F8 | 6D | 3 | 0 | 3A5 | E | 2E6 | 560 | CA | 3BF | 56 | 369 |
| 9X | 1D3 | 4C | 1D4 | 66 | 5B | 19 | 5A | 0 | 1F0 | 41 | 1D0 | 69 | 0 | 0 | 0 | 0 |
| AX | 30E | B | 2E2 | 2B | 1FD | 62 | 1FC | 0 | 308 | 16 | 37D | BF | 51 | 37C | 33 | 30F6 |
| BX | 63 | 1E | 3A | A09 | 1D6 | 8 | 1 | 0 | 30 | 30 | 38 | B97 | 1F17 | B96 | 1F12 | 172A |
| CX | 65 | 6A | 30 | 368 | 13 | AD | 29 | 0 | 6C | 62 | 54 | 1584 | 1F16 | 1B5 | 1585 | 505D |
| DX | 371 | 7 | 30B | A08 | 1D1 | 4F | 1BA | 0 | 3BE | 0 | 377 | C3E | 0 | 0 | 0 | 0 |
| EX | 14 | 64 | 57 | 65E | B | A1 | 12 | 0 | 55 | 3B | 62 | 1728 | 65D | 17B | 17A | 30F7 |
| FX | 30F5 | C9 | 190 | 30F4 | 172B | 2A | 1F10 | 505CF7 | 3E2A | 1F11 | 3E2B | 2E53 | 2E52 | 1416 | 282F | C3F |

표 2: Lakhani가 제안한 방법과 본 논문에서 제안한 방법의 성능 비교

| Image | Size (bytes) | | | Compression Ratio Gain (%) | | |
|----------|--------------|-------|-------|----------------------------|------|-------|
| | JPEG | OSSC | Ours | OSSC | BASC | Ours |
| Lenna | 19373 | 18295 | 17858 | 5.56 | 6.86 | 7.82 |
| Airplane | 20184 | 19061 | 18193 | 5.56 | 5.98 | 9.86 |
| Baboon | 41958 | 39041 | 38234 | 6.95 | 4.39 | 8.87 |
| Tiffany | 18855 | 17983 | 16289 | 4.62 | 5.72 | 13.60 |
| Sailboat | 26647 | 24679 | 23941 | 7.38 | 6.61 | 10.15 |
| Average | 25403 | 23812 | 22903 | 6.01 | 5.91 | 10.01 |

를 곱하여 양자화표를 만들고 양자화에 적용하여 다양한 화질의 영상을 얻었다. 그림 6은 각각의 영상의 PSNR에 따른 압축 파일 크기를 나타낸다.

그림 6을 통하여 본 논문에서 제안하는 방법의 압축률은 JPEG 표준보다 일반적으로 높은 것을 확인할 수 있다. 특히 PSNR이 높은 경우에는 상대적으로 01스트림의 길이가 길어지기 때문에 더 높은 압축률의 이득을 얻을 수 있다는 것을 실험적으로 확인할 수 있다. PSNR이 낮은 경우에는 후반 0스트림이 매우 길고 01스트림의 길이가 짧기 때문에 압축률의 이득이 거의 없을 수 있다. 또한 01스트림 자체적으로 블록의 끝을 나타내는 표식을 포함하여야 하기 때문에 이와 같은 추가 부담이 증가한다. 하지만 본 논문에서 제안하는 교류 길이 표시를 사용하면 낮은 PSNR의 경우에서처럼 교류의 계수가 전반적으로 작아져 연속적인 블록의 교류 길이가 유사하기 때문에 블록마다 블록 종결 표식을 추가하는 것 보다 보다 효율적으로 블록의 길이를 표시할 수 있고 PSNR이 낮은 경우에도 압축률을 JPEG 보다 낮게 유지할 수 있다.

표 2는 Lakhani가 제안한 교류 계수 분할 방식과 본 논문에서 제안하는 방법의 성능을 비교한 결과이다. 2-4 열은 각 영상에 대하여 IJG-JPEG을 이용하여 압축한 파일 크기, Lakhani가 제안한 방법들 중 평균적으로 가장 높은 압축률을 보이는 OSSC 방식으로 얻은 파일 크기, 본 논문에서 제안한 방법으로 생성된 파일 크기를 각각 나타낸다. 표에서 나타난 바와 같이 모든 영상에서 본 논문이 제안한 방법이 보다 높은 압축률을 보임

을 알 수 있다. 5-7열은 Lakhani가 제안한 OSSC 방식, 특정한 경우에 OSSC보다 높은 성능을 보이는 BASC 방식, 그리고 본 논문에서 제안하는 방식을 비교하였다. 비교의 편이를 위해 JPEG 대비 압축률의 증가 비율을 나타낸다. OSSC나 BASC 공히 평균적으로 6%의 압축률 향상이 있는 반면 본 논문에서 제안하는 방법은 평균 10%의 압축률 향상을 보이는 것을 확인할 수 있다.

본 논문에서 제안하는 엔트로피 부호화 과정은 JPEG에 비해 추가로 2049바이트의 추가적인 메모리 공간을 요구한다. 교류 계수의 길이의 허프만 부호화를 위해 각각의 빈도수를 저장하기 위한 공간(4바이트×128개=512바이트)과 허프만 부호화 표(4바이트×128개=512바이트), 01 스트림의 허프만 부호화 표를 위한 공간(4바이트×256개=1024 바이트)이 요구된다. 이외에 01 스트림은 바이트 단위로 묶어 저장하기 위한 한 바이트의 저장공간이 필요하다. 복호화 과정에서 역시 두 개의 허프만 표를 저장하기 위한 공간과 01스트림의 내용을 풀기 위한 한 바이트의 공간만이 추가적으로 요구된다. 특히 이 한 바이트의 공간을 사용함으로써 JPEG과 동일하게 블록 단위로 각각의 블록을 독립적으로 순차적으로 복호화 할 수 있다.

본 논문에서 제안하는 압축 과정은 기존의 JPEG에 비해 교류 스트림의 길이를 구하는 과정과 길이를 저장하기 위한 허프만 부호화표를 생성하는 과정에 의해 계산량의 증가가 있지만 01 스트림의 값을 4개씩 모아서 허프만 부호화 하기 때문에 이와 같은 계산량 증가를 일부 상쇄할 수 있다. 본 논문에서 제안하는 압축 방법은 대상 화질에 따라서 JPEG 대비 -0.5%에서 2.5% 정도의 미미한 속도 변화를 보였다. 압축을 해제하는 경우에는 허프만 부호의 복호화 과정이 줄어들기 때문에 평균적으로 JPEG의 85%에서 90%의 시간만이 소요됨을 실험적으로 확인하였다.

5 결론

본 논문은 교류 계수를 분할하여 별도로 압축함으로써 JPEG 영상 압축률을 개선하기 위한 방법을 제안하였다. 기존의 JPEG은 교류 계수 영역을 후반 0스트림과 그외의 것으로 구분하는 반면 본 논문에서 제안하는 방법은 01스트림 부분을 추가

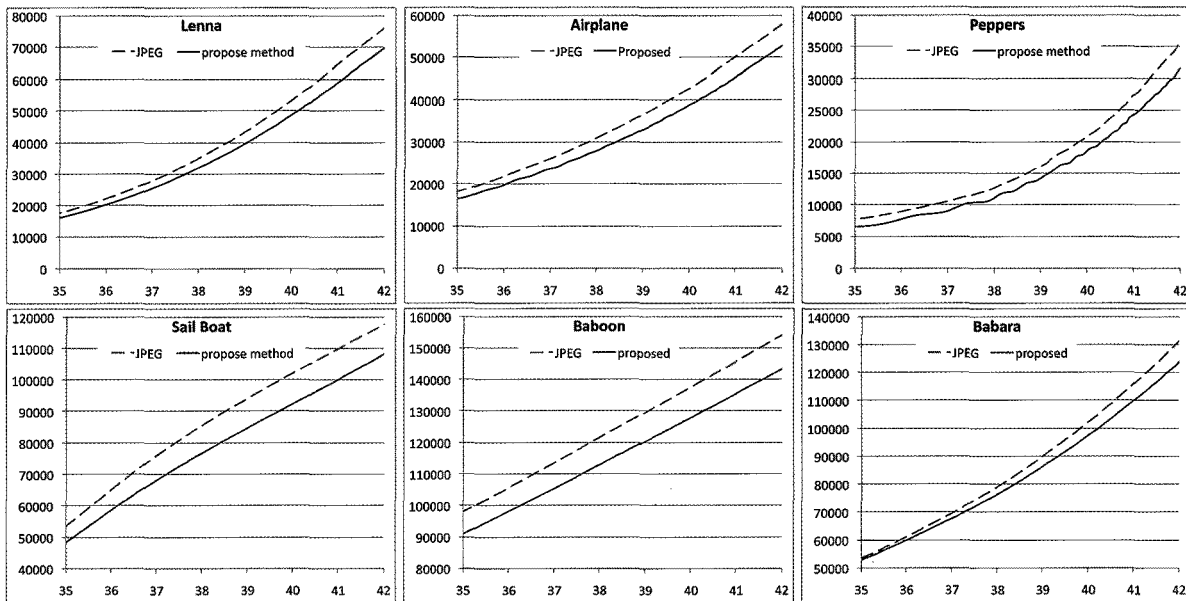


그림 6: PSNR에 따른 압축 파일 크기 비교 (x축: PSNR, y축: 압축된 파일 크기)

하여 세 부분으로 나누고 각각을 압축하는 방법으로 JPEG의 엔트로피 부호화 과정을 개선하였다. 또한 교류 계수의 길이가 연속된 블록에서 유사하다는 점에 착안하여 압축률을 보다 개선하기 위한 방법을 제안하였다. 결과적으로 본 논문에서 제안하는 방법은 JPEG의 화질을 그대로 유지하면서 전반적인 PSNR 영역에 대하여 압축률을 향상시킬 수 있었다.

본 논문의 대표적인 문제점으로는 JPEG 표준과 다른 방식으로 영상을 압축하기 때문에 이와 같은 방법으로 압축된 파일을 표준적인 JPEG 복호화 방식으로는 압축 해제 할 수 없다는 점을 들 수 있다. 이는 본 연구에서 제안하는 방법이 엔트로피 부호화 과정을 일부 개선하여 압축률을 향상시키는 방법이라는 데서 기인한다. 기존 연구에서도 특히 엔트로피 부호화 과정을 최적화하여 압축률을 향상시키기 위한 연구 중 다수가 같은 제한점을 가진다. 따라서 이와 같은 연구 결과는 향후 JPEG의 성능이나 화질에서의 장점을 유지하며 보다 높은 압축률을 가지는 새로운 표준을 제정하는데 기여할 수 있을 것으로 기대한다.

감사의 글

위 연구는 2007학년도 아주대학교 일반연구비 지원에 의하여 연구되었으며 방위사업청과 국방과학연구소(UD060048AD)의 지원으로 수행되었습니다.

참고문헌

- [1] D. Santa-Cruz, R. Grosbois, and T. Ebrahimi, "Jpeg 2000 performance evaluation and assessment," *Signal Processing: Image Communication*, vol. 17, pp. 113–130, 2002.
- [2] J. Miano, *Compressed Image File Formats*. Addison-Wesley, 2000.
- [3] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [4] J. L. Mitchell and W. B. Pennebaker, *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers, 1992.
- [5] F. L. V. Nes and M. A. Bouman, "Spatial modulation transfer in the human eye," *Journal of the Optical Society of America*, vol. 57, no. 3, p. 401, 1967.
- [6] A. B. Watson, "Image compression using discrete cosine transform," *Mathematica Journal*, vol. 4, no. 1, pp. 810–88, 1994.
- [7] A. B. Watson, "Dct quantization matrices visually optimize for individual images," in *Proceedings of the SPIE*, 1993.
- [8] V. Ratnakr and M. Livny, "Rd-opt: An efficient algorithm for optimizing dct quantization tables," in *Proceedings of Data Compression Conference*, 1995, pp. 332–341.
- [9] C. Tu, J. Liang, and T. Tran, "Adaptive runlength coding," *Signal Processing Letters, IEEE*, vol. 10, no. 3, pp. 61–64, Mar 2003.
- [10] G. Lakhani, "Modified jpeg huffman coding," *Image Processing, IEEE Transactions on*, vol. 12, no. 2, pp. 159–169, Feb 2003.
- [11] B. Jeo, J. Parkb, and J. Jeongc, "Huffman coding of dct coefficients using dynamic codeword assignment and adaptive codebook selection," *Signal Processing: Image Communication*, vol. 12, no. 3, pp. 253–262, June 1998.

〈저자소개〉



안영훈

- 2006년 2월 아주대학교 컴퓨터공학부 (학사)
- 2008년 2월 아주대학교 정보통신전문대학원 (석사)
- 2008년 3월 ~ 아주대학교 컴퓨터공학과 (박사과정)
- 관심분야: 이미지 손실/무손실 압축, 이미지 영상처리, 동영상 화질 개선



신현준

- 1995년 2월 한국과학기술원 전산학과 (학사)
- 1997년 2월 한국과학기술원 전산학과 (석사)
- 2002년 2월 한국과학기술원 전산학과 (박사)
- 2002년 3월~2002년 8월 한국과학기술원 박사후연구원
- 2002년 9월~2003년 12월 University of Wisconsin Visiting Scholar
- 2004년 1월~2004년 2월 한국과학기술원 박사후연구원
- 2004년 3월~2005년 2월 서울대학교 박사후연구원
- 2005년 2월~ 현재 아주대학교 미디어학부 교수
- 관심분야: 컴퓨터그래픽스, 영상처리, 그래픽시스템 등



위영철

- 1982년 12월 State Univ. of NY at Albany 전산학과 (학사)
- 1984년 12월 State Univ. of NY at Albany 전산학과 (석사)
- 1989년 12월 State Univ. of NY at Albany 전산학과 (박사)
- 1990년 3월 ~ 1995년 4월 삼성종합기술원 수석연구원
- 1995년 5월 ~ 1998년 2월 현대전자 기획부장
- 1998년 3월 ~ 현재 아주대학교 정보및컴퓨터공학부 교수
- 관심분야: 영상처리, 정보 압축, 컴퓨터그래픽스