

# Stack-Run 알고리즘을 이용한 영상부호기의 설계 및 구현

이 석 희<sup>†</sup> · 최 길 성<sup>†</sup> · 유 재 수<sup>††</sup> · 조 기 형<sup>††</sup>

## 요 약

본 논문에서는 압축률을 높이기 위해 Stack-Run 알고리즘을 이용하여 4개의 심볼만 사용하는 DCT 기반의 새로운 영상부호기를 제안한다. 제안된 영상부호기는 JPEG 부호화 알고리즘을 개선하여 심볼의 개수를 4개만 사용하기 때문에 부호와 복호 알고리즘이 간단하다. 또한 심볼의 발생확률을 기반으로 부호화를 행하는 호프만 부호기 또는 산술부호기와 같은 다른 엔트로피 부호기와의 결합이 용이하다. 제안된 부호기의 압축률을 평가하고 기존의 JPEG 부호기와의 성능비교를 위해, Lena 영상과 Boat 영상을 이용하여 실험을 수행한다. 성능비교 결과, 제안된 영상부호기가 JPEG 베이스라인 부호기에 비해 훨씬 우수한 압축률을 보인다.

## Design and Implementation of a Image Coder Using Stack-Run Algorithm

Seok Hee Lee<sup>†</sup> · Kil Seong Choi<sup>†</sup> · Jae Soo Yoo<sup>††</sup> · Ki Hyung Cho<sup>††</sup>

## ABSTRACT

In this paper, to improve the compression ratio, we propose a new still image coder based on DCT that needs just four symbols by using stack-run algorithm. Since the proposed still image coder uses only four symbols, it is very simple to encode and decode the still images over the JPEG coder algorithm. It is also easy to be combined with other entropy coders such as huffman coder or arithmetic coder based on the probability of symbols. In order to evaluate the compression ratio of the proposed coder and compare the performance of our coder with that of the existing JPEG coders, we perform experiments using lena and boat images. We show through experiments that our proposed coder achieves much better compression ratio than the JPEG baseline mode.

## 1. 서 론

정보화 사회가 진전되면서 수치, 문자, 음성, 화상

※본 논문은 ETRI에서 수행하는 "21세기 DBMS 기술연구"사업과 "정보통신부 우수학교 지원사업"에서 지원받아 수행한 것입니다.

† 정희원: 충북대학교 정보통신공학과

†† 정희원: 충북대학교 전기전자공학과

논문접수: 1997년 7월 15일, 심사완료: 1997년 12월 4일

및 영상 등의 다양한 데이터를 상호 유기적으로 관련시켜 통합하는 멀티미디어 기술이 요구되고 있다. 이 가운데 최근 영상통신 및 멀티미디어 등의 급속한 발달에 힘입어 영상 정보의 효율적인 부호화에 대한 연구의 중요성이 더욱 강조되고 있다. 화상 및 영상 데이터는 인간의 시각 특성을 이용하기 때문에 정보 전달의 효과가 크다. 그러나, 이러한 영상정보는 매우 큰 정보량을 가지고 있어 처리시에 많은 어려움이 있

다. 그래서 효율적인 처리 즉 저장과 전송을 용이하게 하기 위해서는 압축이 필수적이다. 그러나 영상압축은 압축률이라는 수치적인 면과 시각적으로 보기 좋아야 한다는 질적인 면을 동시에 만족해야 한다는 어려운 문제점을 가지고 있다. 영상의 특징은 반복적인 정보가 많다는 것인데 이러한 반복성을 줄이는 것이 영상압축의 기본원리이다. 현재까지 수많은 영상 압축 방법이 제시되었으며 이들은 각각 서로 다른 응용분야에 적용되어 사용되고 있다. [1][2][8][9]

JPEG 부호화 알고리즘은 최근 정지영상의 압축방법으로서 멀티미디어 산업계 전반에 걸쳐 폭 넓게 사용되고 있다. 이 방식은 압축률이 크지 않을 경우 비교적 좋은 화질을 제공하며 효율적인 하드웨어의 구성을 가능하게 하는 장점을 가지고 있다. JPEG 부호화 알고리즘은 처리시간 및 버퍼 요구사항을 줄이기 위하여 DCT 모드의 경우 입력영상을 크기가 8×8개인 블록단위로 분리한 후, 각 블록을 DCT(Discrete Cosine Transform)변환하여 64개의 DCT 계수로 만든다. 이렇게 변환된 계수는 양자화에 의해 양자화가 이루어진 다음 가변길이 부호기에 의해 부호화가 이루어진다. 연속적인 블록들의 DC 계수는 DPCM(Difference Pulse Code Modulation)형태로 부호화가 이루어지고 블록내 AC 계수는 RLC(Run-Length Coder)가 이루어진다. JPEG은 RLC 방식에 있어서 블록당 하나의 연속 비트열을 구성하기 위해 지그재그 형태로 검색을 실시한다. 이때 각각의 계수들은 zero의 길이와 zero가 아닌 계수값의 2차원 테이블을 이용하여 부호화된다. [3-6][11]

본 논문에서는 JPEG에서 2차원 테이블을 이용하여 심볼의 개수를 240개로 만든 다음 그 심볼에 코드를 할당하여 부호화하는 방법을 개선하여 심볼의 개수를 4개만 사용하는 Stack-Run 알고리즘을 적용시킨 새로운 부호기를 설계하고 구현한다. 제안된 부호기에서는 4개의 심볼만 사용하기 때문에 부호화 복호 알고리즘이 간단하며, 심볼의 발생확률을 기반으로 부호화를 행하는 호프만 부호기 또는 산술 부호기와 같은 다른 엔트로피 부호기와의 결합이 용이하다. 본 논문의 구성은 2장에서 JPEG 부호기의 심볼집합과 Stack-Run 알고리즘에 대하여 살펴본 후, 3장에서 Stack-Run 알고리즘을 사용하는 정지영상 부호기를 제안하며, 4장에서는 제안된 부호기의 압축률 측면에

서 성능평가를 수행하고 마지막으로 5장에서 결론을 맺는다.

## 2. JPEG의 심볼집합과 Stack-Run 알고리즘

### 2.1 JPEG 부호기에서 사용되는 심볼집합

JPEG 부호기에 사용되는 심볼들은 (그림 1)에 나타난 것처럼 2차원 테이블에 의한 총 240개의 심볼들로 구성되어 있다. [5] JPEG 부호기의 부호화 과정에서 중요계수들의 그룹번호와 zero 길이의 2차원 테이블로 표현되는 심볼들의 확률 빈도수를 기반으로 호프만 부호화한 다음 부가비트와 합해서 저장한다. 그러므로 JPEG의 부호화 및 복호화 과정에서 계수들을 2차원의 심볼로 변환하여 처리하여야 한다. 이러한 이유 때문에 알고리즘이 복잡하고 코드를 저장하기 위한 별도의 처리와 다수의 기억공간이 필요한 단점이 있다. 알고리즘을 보다 단순화하기 위해서 DC 계수와 AC 계수를 분리하여 부호화하지 않고 함께 부호화하며 양자화된 DCT 계수값을 4개의 심볼로 표현하므로써 보다 간단한 알고리즘이 될 수 있다.

		계수의 분류 그룹														
		0	1	2	...	13	14									
0의 길이	0	EOB														
	...	X	COMPOSITE VALUE													
	...	X														
	...	X														
15	ZRL															

(그림 1) JPEG 부호기의 심볼테이블  
(Fig. 1) Symbol table of JPEG coder

### 2.2 Stack-Run 알고리즘에 사용되는 심볼집합

Stack-Run 알고리즘에서는 다음과 같은 4가지 심볼집합 {0, 1, +, -}을 사용한다. 먼저 심볼 {0}은 중요계수를 이진정보로 표현했을 때 중요계수의 "0"을 표현하는데 사용된다. 심볼 {1}은 중요계수의 "1"을 표현하는 데 사용한다. 그러나 최상위 비트를 표현할 때는 사용하지 않는다. 중요계수의 최상위 비트는 다

음에 설명될 심볼 (+, -)를 사용하여 표현한다. 심볼 (+)는 중요계수가 양수일 경우 중요계수의 최상위 비트를 표현할 때 사용하고, 동시에 zero 길이를 이진 정보로 표현했을 때 zero 길이의 "1"을 나타낸다. 심볼 (-)는 중요계수가 음수일 경우 중요계수의 최상위 비트를 표현하고, zero 길이를 이진정보로 표현했을 때의 zero 길이의 "0"을 나타낸다.

이러한 4개의 심볼집합을 사용하므로써 복호화 할 때 중요계수와 zero 길이 사이를 명확히 구분할 수 있다. 또한, 중요계수를 표현할 때의 심볼 (+)와 (-)는 중요계수의 부호를 나타내는 동시에 최상위 비트의 위치를 나타낸다. 이와 유사한 이점 zero 길이를 표현할 때도 최하위 비트부터 최상위 비트의 순으로 배열하므로써 얻을 수 있다. zero 길이를 이진정보로 표현하면 항상 1로 시작되기 때문에 마지막의 심볼 (+)(MSB) 1개를 정보의 손실없이 모든 zero 길이의 이진정보 표현에서 생략할 수 있다. 그러나 만약 모든 zero 길이의 심볼 (+)(MSB)를 생략한다면 zero 길이가 1과 3일 때 구분할 수 없는 문제점이 발생한다. 이러한 문제점은 zero 길이에 1를 더함으로써 간단히 해결할 수 있다. 이와 같은 문제점은 중요계수를 심볼로 표현할 경우에도 똑같이 발생하므로 중요계수에도 같은 방법을 적용시켜야 한다. zero 길이의 이진정보표현에서 심볼 {0} 또는 {1}이 나타나면 zero 이진정보의 끝을 나타내는 동시에 이것은 다음 중요계수의 LSB가 된다.

### 2.3 Stack-Run 알고리즘

(그림 2)는 DCT계수를 심볼열로 표현하는 과정을 의사코드(pseudo code)로써 나타낸 것이다. 이렇게 데이터를 심볼로 매핑시키는 방법은 산술 부호화를 할 때 다음과 같은 이점을 얻을 수 있다. 첫 번째, 심볼의 개수가 적기 때문에 산술 부호화에서 확률 테이블을 적용적으로 적용시킬 때 용이하다. 두 번째, zero 길이를 나타내는 심볼과 중요계수의 LSB가 중요계수를 나타내는 나머지 심볼들과 구분해 주는 역할을 하므로 결과적으로 zero 길이와 중요계수를 구분해 주는 별도의 심볼을 고려하지 않아도 된다. zero 길이는 심볼 (+)와 (-)로 끝을 나타낼 것이며, 중요계수 값의 끝은 심볼 {0}과 {1}로 나타낼 것이다. 세 번째, 중요계수를 표현할 때 실제적인 값을 표현하는

```

< 중요계수의 심볼화 알고리즘 >
Begin.
  If (입력계수 != 0) ( // 중요계수이면
    If (중요계수 < 0) (
      sign = "-"
      절대값(중요계수)
    )
    else sign = "+"
    중요계수 값을 이진수로 표현
    LSB->MSB순으로 재배열
    for (LSB에서 MSB-1까지) (
      If 1이면 심볼 "1" 출력
      If 0이면 심볼 "0" 출력
    ) // 최상위 비트 생략됨
    부호출력
  )
End.

< zero길이의 심볼화 알고리즘 >
Begin
  zero길이 값을 이진수로 표현
  LSB->MSB순으로 재배열
  for (LSB에서 MSB-1까지) (
    If 1이면 심볼 "+" 출력
    If 0이면 심볼 "-" 출력
  ) // 최상위 비트 생략됨
End

```

(그림 2) 심볼화 과정을 표현한 의사코드  
(Fig. 2) Pseudo code for phased encoding procedure

```

< 중요계수와 zero 길이의 복호화 알고리즘 >
Begin.
  If 읽은 심볼이 "+" 또는 "-" 이면 zero길이 (
    while(0,1 심볼이 Read될 때 까지) (
      심볼을 한 개씩 읽어 이진수로 변환.
    )
    최상위 비트를 추가.
    복원된 zero길이 만큼 0값을 출력.
  )
  else 읽은 심볼이 "0" 또는 "1" 이면 중요계수 (
    while(+,- 심볼이 Read될 때 까지) (
      심볼을 한 개씩 읽어 이진수로 변환.
    )
    If 마지막 심볼이 '-' 이면 (
      중요계수 * -1.
    )
    중요계수 값을 출력.
  )
End.

```

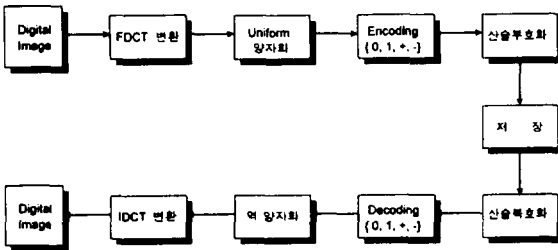
(그림 3) 복호화 과정을 표현한 의사코드  
(Fig. 3) Pseudo code for phased decoding procedure

유효비트만 부호화할 수 있다. 네 번째, Stack-Run 알고리즘은 중요계수의 값이 작고 zero 길이가 긴 값을 가질 때 특히 우수한 성능을 나타낸다. (그림 3)은 복호화하는 과정을 의사코드(pseudo code)로 나타낸 것이다.

### 3. 제안된 영상부호기

#### 3.1 전체구조

제안된 부호기의 주요한 구성도는 (그림 4)와 같다. 대략적으로 부호기의 구성을 살펴보면 먼저 입력영상을 8×8 블록으로 나눈 후 각각을 DCT 변환한다. DCT 변환된 결과를 양자화 테이블을 이용하여 양자화를 수행한 후, DC 계수에 DPCM을 적용시킨 다음, Stack-Run 알고리즘으로 4개의 심볼 {0, 1, +, -}로 표현한다. 이렇게 표현된 4개의 심볼을 적용 산술 부호화 과정을 거쳐서 최종적으로 부호화된 이진 비트열을 얻는다.



(그림 4) 제안된 영상부호기 구성도  
(Fig. 4) Structure of proposed image coder

#### 3.2 FDCT 변환과 IDCT

DCT는 8×8의 영상블록을 64개의 주파수 성분으로 분리하는 역할을 한다. 즉, 입력 영상블록이 가지고 있는 공간영역의 정보를 주파수 크기별로 분리해주는 역할을 한다. 또한 주파수의 분리와 함께 DC 성분 근처에 있는 몇 개의 AC 계수들로 에너지를 집중시키는데 이것이 DCT 변환을 부호화에 이용하는 이유이다.

대부분의 영상 블록에서는 블록의 범위가 작기 때문에 화소의 값이 큰 범위로 변하지 않는다. 이것은 주파수 성분 즉 DCT 계수 값에서 고주파 성분의 값

이 작다는 것을 의미한다. 다시 말하면 블록내의 주파수 성분이 DCT 계수의 DC성분 근처에 몰린다. 더욱이 영상의 인지성 측면에서 고주파 계수는 영상의 세밀한 부분을 의미하므로 이것의 약간의 손상은 전체 화질에 많은 영향을 끼치지 않는다. 그러므로 이 DCT 계수를 주파수 별로 다른 값으로 양자화 하는데 저주파 성분은 작은 값으로 양자화하여 세밀한 값을 부호화하며 상대적으로 고주파 성분으로 갈수록 큰 값으로 양자화하여 약간의 손실로 압축을 극대화한다. 이러한 양자화를 통한 DCT계수는 더욱 더 고주파 성분에서 작은 값을 가지게 되며, 높은 고주파 성분 근처의 계수들은 대부분 zero의 값을 가지게 된다.

#### 3.3 양자화(Quantization)와 역양자화(Dequantization)

양자화 테이블은 64개의 양자화 계수로 구성된 고정된 행렬이다. 이러한 양자화 테이블을 이용하여 블록내 각각의 DCT 계수에 대하여 양자화를 하게 된다. 양자화 과정의 목적은 실제적인 압축 효과를 얻기 위한 것인데, 구현하려고 하는 영상 화질에 맞는 정확도의 DCT 계수를 결과로 얻게 된다. 일반적으로 압축률 조정이 이 과정을 통하여 이루어진다. 구체적으로 양자화 테이블의 값들은 1~255 까지의 범위를 갖는 정수인데 각각의 DCT 계수들에 대한 양자화 폭을 나타낸다. 양자화 테이블에서의 각 양자화 폭은 시각 특성과 직접적으로 관련이 있으며, 영상 특성 및 기타 영상 표시 특성 등과도 관련지어 결정되어야 한다. 양자화 과정은 many-to-one 변환이므로 역양자화 과정을 거치게 될 때에 정보를 유실하는 비가역 변환이다. 따라서 양자화 과정이 DCT 기반의 압축 알고리즘에서의 주요 손실 원인이 된다. 구체적으로 DCT 계수의 양자화는 각 DCT 계수를 이에 대응하는 양자화 폭으로 나눈 후, 가장 가까운 정수로 치환함으로써 이루어진다.

각 계수  $S_{uv}$ 에 대한 양자화 폭( $Q_{uv}$ )은 양자화 테이블에 대응하고 있는 요소의 값이고 양자화 된 2차원 DCT 계수를  $Sq_{uv}$ 라 하면 균일 양자화기는 다음 (식 1)와 같이 정의된다.

$$\begin{aligned} \text{양자화} \quad Sq_{uv} &= \text{round}\left(\frac{S_{uv}}{Q_{uv}}\right) \\ \text{역양자화} \quad R_{uv} &= Sq_{uv} * Q_{uv} \end{aligned} \quad (1)$$

### 3.4 산술부호기

제안된 부호기에서 산술부호기는 선택적으로 적용시킬 수 있다. 더욱 향상된 압축률을 얻고자 할 때 stack-run 알고리즘으로 심분화된 심분들에 대해 산술부호화를 할 수 있다. 본 논문에서 제안된 부호기에서는 참고문헌[7]에서 제공된 적용형 산술부호기를 사용하였다.

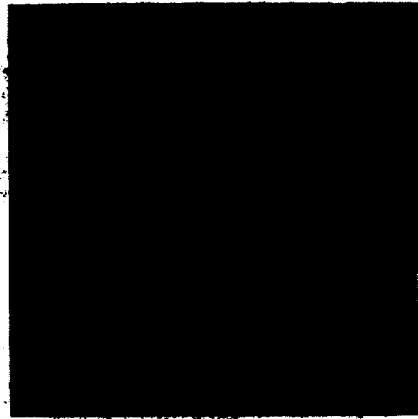
산술부호기에서 부호화된 메시지는 0~1 사이의 수직선상에 한 점을 a라 하면 a와 그곳을 기준으로 c라고 하는 어떠한 폭을 갖는 선분으로 나타내어진다. 부호화가 진행됨에 따라 부호화된 메시지를 나타내는 a와 c의 표현을 위해, 필요한 bit 수는 증가하게 된다. 연속된 data의 입력에 대해 정해진 윈도우의 내용에 따른 확률 값을 갱신하고, 특히 이치 data의 경우, MPS(More Probable System)가 발생할 때 a는 그 값을 유지하며, c는 그 MPS의 확률 값으로 대체된다. 따라서 MPS가 발생할 때 LPS(Less Probable System)보다 a와 c의 값 변화량이 적으며, 부호화된 data에 보다 적은 수의 bit가 추가된다. [7]

## 4. 실험 및 결과분석

### 4.1 실험환경

제안된 부호기의 성능을 평가하기 위하여 크기가 가로, 세로 512×512이며 8비트로 양자화된 (그림 5)의 Lena 영상과 (그림 6)의 Boat 영상을 이용하여 컴퓨터 시뮬레이션을 수행하였다. 영상의 압축 부호화를 평가할 때는 영상 자체가 압축의 대상이 되기 때문에 부호화 방법 자체의 효율뿐만 아니라, 원 영상이 어느 정도로 압축되었고 이를 다시 복원했을 때 얼마만한 정보의 손실이 있었는가를 평가할 수 있는 기준이 있어야 한다. 본 논문에서는 객관적인 평가 결과를 얻기 위해 비교되는 모든 방식에 대하여 똑같이 헤더정보를 고려하지 않았다. 또한 비교되는 모든 방식에 대해 같은 PSNR(Peak Signal to Noise Ratio)에서의 압축률 bpp를 측정하였다. PSNR이 높을수록 좋은 화질을 나타내며 압축률을 나타내는 bpp가 낮을수록 좋은 압축효과를 나타낸다.

실험결과와 비교자료로서 사용된 복원화상의 화질은 4가지 경우로 구분하였다. 비교화질은 제안된 부호기의 압축성능을 알기 쉽게 나타내기 위하여 JPEG



(그림 5) 실험 영상(Lena)  
(Fig. 5) Experiment image(Lena)



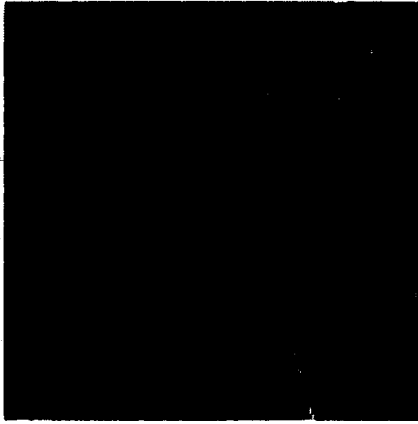
(그림 6) 실험영상(Boat)  
(Fig. 6) Experiment image(Boat)

베이스라인 모드를 사용하여 압축했을 경우 압축률이 각각 0.125bpp, 0.25bpp, 0.5bpp, 1.0bpp로 나타나는 때의 복원화질을 선택하였다.

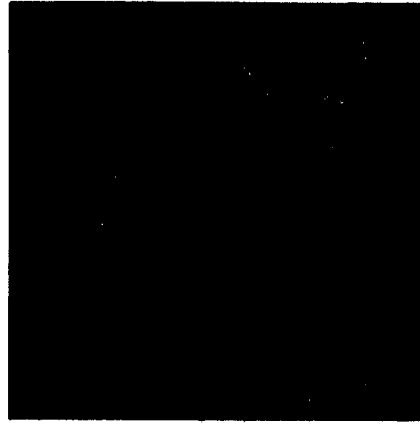
(그림 7)과 (그림 8)에 제안된 알고리즘을 사용한 복원화질을 각각의 PSNR별로 제시하였다.

### 4.2 산술부호기를 사용하지 않은 경우

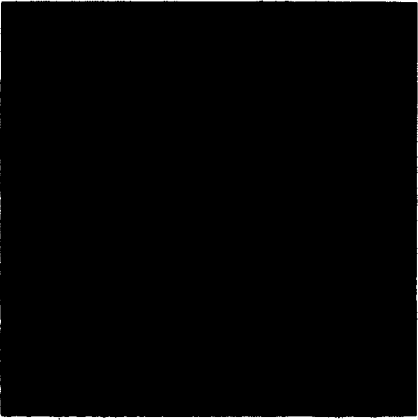
먼저 제안된 부호기에서 산술 부호기나 호프만 부호기와 같은 엔트로피 부호기를 사용하지 않고 단순히 Stack-Run 알고리즘만 적용하여 시뮬레이션을 수



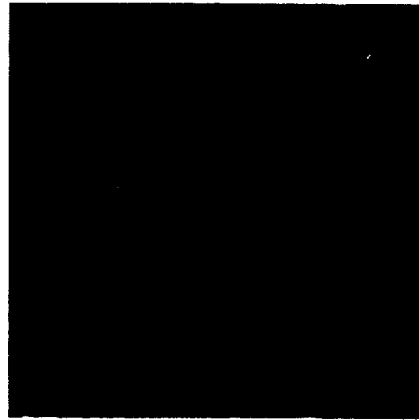
(a) PSNR : 24.23dB



(b) PSNR : 30.84dB

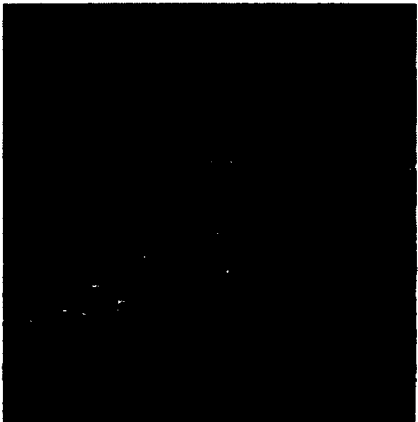


(c) PSNR : 34.76dB

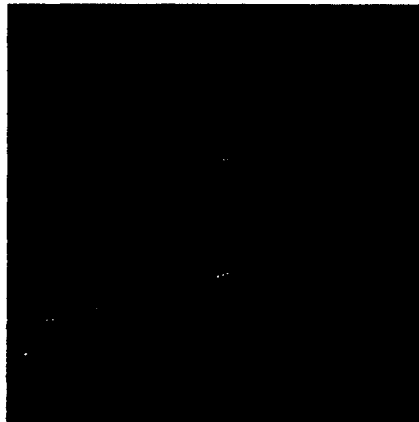


(d) PSNR : 37.92dB

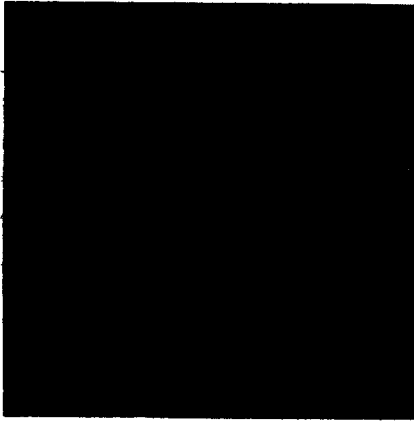
(그림 7) Lena 영상의 복원화질  
(Fig. 7) Restoration of Lena image



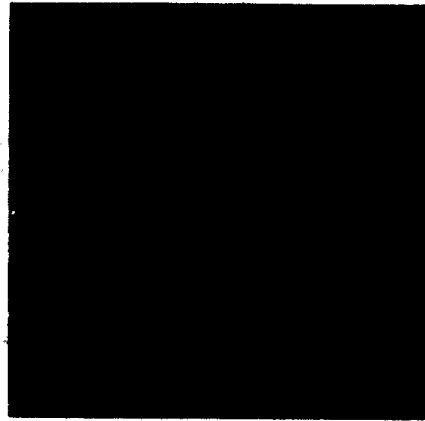
(a) PSNR : 24.26dB



(b) PSNR : 27.54dB



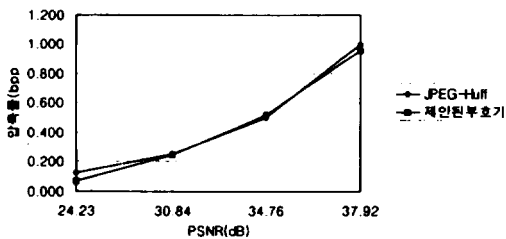
(c) PSNR : 31.02dB



(d) PSNR : 34.56dB

(그림 8) Boat 영상의 복원화질  
(Fig. 8) Restoration of Boat image

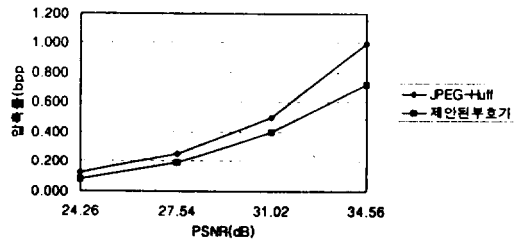
행하였다. 이 실험에서는 Stack-Run 알고리즘의 결과로 만들어진 심볼열에 각각의 심볼당 2비트의 코드를 일률적으로 적용하는 방법을 사용하였다. 각각의 심볼은 다음과 같이 2비트의 이진비트열로 할당되었다. 심볼 "0"에는 00, 심볼 "1"에는 01, 심볼 "+"에는 10, 심볼 "-"에는 11로 각각 할당되었다. 다음의 (그림 9)와 (그림 10)은 JPEG 베이스라인 부호기와 엔트로피 부호기를 사용하지 않을 경우의 압축률을 비교한 결과이다.



(그림 9) Lena 영상의 압축률 비교 그래프  
(Fig. 9) Comparison of compression ratio for Lena image

(그림 9)는 Lena 영상을 대상으로 JPEG 베이스라인 부호기와 압축률을 비교한 결과 그래프이다. (그림 9)를 보면 비교된 모든 화질에서 제안된 부호기가 산술부호기를 사용하지 않아도 JPEG 베이스라인 부

호기보다 향상된 압축성능을 나타냄을 알 수 있다.



(그림 10) Boat 영상의 압축률 비교 그래프  
(Fig. 10) Comparison of compression ratio for Boat image

(그림 10)은 Boat 영상을 대상으로 JPEG 베이스라인 부호기와 압축률을 비교한 결과 그래프이다. (그림 10)은 (그림 9)와 비교해 볼 때 더욱 더 향상된 결과를 나타내었다. Boat 영상이 Lena 영상에 비해 향상된 결과를 나타내는 것은 Boat 영상내에 선 성분과 같은 고주파 성분이 Lena 영상에 비해 많이 포함되어 있기 때문인 것으로 판단된다. 고주파 성분은 제안된 부호기내의 양자화기에서 거의 대부분 제거되기 때문에 제안된 부호기의 성능은 고주파 성분보다는 저주파 성분에 영향이 많은 것을 알 수 있다. 엔트로피 부호기를 사용하지 않았을 때의 성능평가 결과를 <표 1>에서 보였다. 결과로서 Lena 영상, Boat 영상 모두

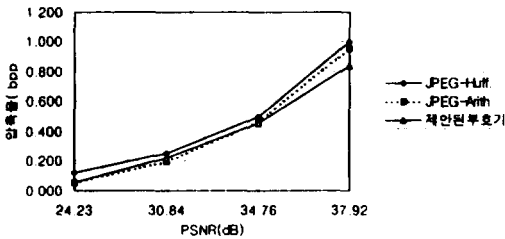
35dB 이상에서 향상된 결과를 나타내고 있음을 알 수 있다.

<표 1> 제안된 부호기의 성능평가 결과(산술부호기 미사용)  
<Table 1> Results of proposed coder(not use arithmetic coder)

Image	PSNR(dB)	JPEG-Huff.	제안된 부호기
Lena	24.23	0.125	0.065
	30.84	0.250	0.248
	34.76	0.500	0.518
	37.92	1.000	0.953
Boat	24.26	0.125	0.080
	27.54	0.250	0.188
	31.02	0.500	0.397
	34.56	1.000	0.720

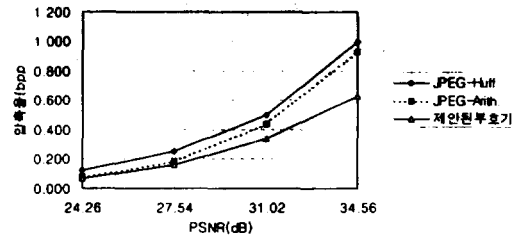
4.3 산술부호기를 사용한 경우

압축률을 더욱 높이기 위하여 Stack-Run 알고리즘을 이용하여 심분화된 결과를 산술 부호기를 사용하여 추가로 압축한 실험결과이다. (그림 11)은 Lena 영상을 대상으로 JPEG 부호기와 본 논문에서 제안된 부호기와의 압축률을 비교한 결과 그래프이다. (그림 11)에 나타난 그래프를 통해 알 수 있듯이 모든 PSNR에서 제안된 부호기가 JPEG 베이스라인 부호기 보다는 훨씬 높은 압축률을 보여 주고 있다. JPEG 확장모드 부호기에 비해서는 PSNR 30dB이하에서 유사한 성능을 나타내었으나 PSNR 30dB 이상에서는 월등히 우수한 성능을 나타내었다.



(그림 11) Lena 영상의 압축률 비교 그래프  
(Fig. 11) Comparison of compression ratio of Lena image

(그림 12)는 Boat 영상을 대상으로 JPEG 표준 부호기와 본 논문에서 제안된 부호기와의 압축률을 비교한 결과 그래프이다. Boat 영상을 대상으로 압축률을 비교한 (그림 12)는 모든 PSNR에서 제안된 부호기가 JPEG 표준 부호기 보다 월등히 높은 압축률로 압축됨을 보였다. 특히 Lena 영상의 경우와 같이 PSNR이 35dB 이상의 좋은 화질에서 더욱 우수한 성능을 보였다.



(그림 12) Boat 영상의 압축률 비교 그래프  
(Fig. 12) Comparison of compression ratio of Boat image

<표 2>에서 Lena 영상과 Boat 영상에 대한 산술 부호기를 사용했을 때의 결과를 보였다. <표 1>보다 더욱 향상된 결과값을 나타내고 있음을 알 수 있다.

<표 2> 제안된 부호기의 성능평가 결과(산술부호기 사용)  
<Table 2> Results of proposed coder(use arithmetic coder)

Image	PSNR(dB)	JPEG-Huff.	JPEG-Arith.	제안된 부호기
Lena	24.23	0.125	0.054	0.056
	30.84	0.250	0.198	0.219
	34.76	0.500	0.463	0.458
	37.92	1.000	0.954	0.835
Boat	24.26	0.125	0.078	0.067
	27.54	0.250	0.184	0.159
	31.02	0.500	0.437	0.340
	34.56	1.000	0.931	0.627

5. 결 론

본 논문에서는 Stack-Run 알고리즘을 적용시켜 4개의 심분만 사용하는 새로운 부호기를 제안하였다.



제안된 새로운 부호기의 성능을 평가한 결과 Lena 영상의 경우 PSNR 37.92dB 일 경우 JPEG 베이스라인 부호기 보다 16.5% 향상된 압축률을 보였고, JPEG 확장모드 부호기에 비해서는 같은 PSNR에서 12.5% 향상된 결과를 보였다. Boat 영상의 경우 PSNR 34.56db 에서 JPEG 베이스라인 부호기에 비해서는 37.3%, JPEG 확장모드 부호기에 비해서는 32.7%의 월등히 향상된 압축률을 보였다. Boat 영상이 Lena 영상에 비해 우수한 성능을 나타내는 것으로 보아 제안된 부호기는 복잡도가 높은 영상에 적합한 부호기임을 알 수 있었다.

제안된 부호기는 기존의 방법(JPEG)에 비해 부호/복호 알고리즘이 간단하며, 심볼의 발생확률을 기반으로 부호화를 행하는 호프만 부호기 또는 산술부호기와 같은 다른 엔트로피 부호기의 사용이 용이한 장점이 있다. 제안된 부호기는 zero 길이의 값이 크고 중요계수의 값이 작을 때 좋은 성능을 보이므로 복잡도가 높아 고주파성분이 다수 포함된 영상의 부호화에 적용시키면 좋은 효과를 얻을 수 있으리라 판단된다.

**참 고 문 헌**

[1] M. J. Tsai, J. D. Villasenor and F. Chen, "Stack-Run Image Coding", IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, pp. 519-521, Oct. 1996.

[2] Jiankun Li, Jin Li and C. -C. Jay Kuo, "An Embedded DCT Approach to Progressive Image Compression", IEEE International Conference on Image Processing, Lausanne, Switzerland, Sept. 1996.

[3] Wallace. Gregory K., "The JPEG Still Picture Compression Standard", Communications of the ACM, vol. 34, No. 4, pp. 30-34, April 1991.

[4] K. R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications", San Diego, CA: Academic Press, 1990.

[5] ISO/IEC DIS 10918-1, "Digital Compression and Coding of Continuous-Tone Still Images, Part 1: Requirements and Guidelines", Nov., 1991.

[6] ISO/IEC CD 10918-2, "Digital Compression and

Coding of Continuous-Tone Still Images, Part 2: Compliance Testing", Jan., 1992.

[7] Mark Nelson, "The Data Compression Book", M&T Pub., pp. 123- 218, 1992.

[8] 윤병주, "벡터양자화에 근거한 프레임간 유클리드 부호화기법", 한국과학기술원 석사논문, pp. 8-17, 1994.

[9] 김성호, "벡터양자화를 이용한 유클리드 영상의 부호화기법", 한국과학기술원 석사논문, pp. 3-9, 1993.

[10] 윤재은, "웨이브릿변환을 이용한 에지분류 적용 영상압축기법", 한국과학기술원 석사논문, pp. 4-16, 1995.

[11] 홍창완, "JPEG을 이용한 이미지 압축의 구현 및 압축률 개선에 관한 연구", 홍익대학교 석사논문, pp. 19-36, 1993.



**이 석 희**

1994년 2월 충북대학교 정보통신공학과 졸업(공학사)  
 1998년 2월 충북대학교 정보통신공학과 졸업(공학석사)  
 1998년 3월~현재 충북대학교 정보통신공학과 박사과정

1994년 3월~1996년 2월 충북대학교 정보통신공학과 조교  
 1996년 8월~1997년 8월 충북대학교 전기전자공학부 조교  
 관심분야: 화상처리, 화상통신, 멀티미디어 데이터베이스, 이미지 정보검색



**최 길 성**

1988년 2월 전산업대학교 전자계산학과 졸업(공학사)  
 1992년 2월 수원대학교 전자계산학과(이학석사)  
 1998년 2월 충북대학교 정보통신공학과 박사수료

1988년 5월~1993년 5월 대전산업대학교 전자계산학과 조교  
 관심분야: 멀티미디어 데이터베이스, OODB, 이미지 정보검색



**유 재 수**

- 1980년 전북대학교 공과대학 컴퓨터공학과(학사)
- 1991년 한국과학기술원 전산학과(공학석사)
- 1995년 한국과학기술원 전산학과(공학박사)
- 1995년~1996년 8월 목포대학교 전산통계학과 전임강사

1996년 8월~현재 충북대학교 공과대학 전기전자공학부 조교수

관심분야: 데이터베이스 시스템, 정보검색, 멀티미디어 데이터베이스, 분산 객체 컴퓨팅 등



**조 기 형**

- 1966년 2월 인하대학교 전기공학과(공학사)
- 1984년 8월 청주대학교 산업공학과(공학석사)
- 1992년 2월 경희대학교 전자공학과(공학박사)
- 1981년~1988년 충주산업대학교 컴퓨터공학과 조교수

1988년~현재 충북대학교 정보통신공학과 교수

관심분야: 데이터베이스, 화상처리 및 통신, GIS, 통신프로토콜