論 文

QUAD TREE를 이용한 BTC에서의 영상 데이타 압축

準會員 白 仁 基* 準會員 金 海 洙**

正會員 趙 成 桓***, 正會員 李 根 泳****

BTC employing a Quad Tree Technique for Image Data Compression

In Ky BAIK*, Hae Soo KIM**, Seong Hwan CHO***,

Keun Young LEE **** *Regular Members*

要 約 계산 과정이 비교적 간단하고 실시간 처리가 가능한 coding방법 중의 하나인 BTC(Block Truncation Coding)를 이용하여 영상 데이타를 압축하기 위하여 2진 영상에서 주로 사용하는 quad tree개념을 도입하여 압축율을 개선시키고, 기존의 BTC와 그 성능을 비교하였다. 이 논문에서 제안하는 방법은 그레이 레벨의 변화가 적은 영역에서는 부화상의 크기를 크게 하고, 그레이 레벨의 변화가 큰 영역에서는 부화상의 크기를 작게 하여 전체 부화상의 갯수를 줄임으로써 영상 데이타의 압축을 행하였다. 또한 비트 평면의 효율적인 전송을 위하여 큰 크기의 부화상에 있어서는 Huffman run-length code를, 작은 크기의 부화상에 있어서는 lookup table 방식을 이용하였다. 컴퓨터 시뮬레이션 결과, 크기가 256×256이고, 그레이 레벨이 256인 영상에서 평균 0.8bit/pel의 압축 효과를 얻었다.

ABSTRACT A conventional BTC has the merit of real time processing and simple computation, but has the problem that its compression rate is low. In this paper, a modified BTC using the Quad Tree which is frequently used in binary image is proposed. The method results in the low compression rate by decreasing the total number of subblocks by means of making the size of a subblock large in the small variation area of gray level and the size of a subblock small in the large variation area of gray level. For the effective transmission of bit plane, the Huffman run - lengh code for the large size of a subblock and the lookup table for the small size of a subblock are used. The proposed BTC method show the result of coding 256 level image at the average data rate of about 0.8 bit/pixel.

I. 서 론

사람이 인식하는 정보의 60~90%는 시각을 통해 받아 들이므로 정보 전송에서도 영상을 이용하여 한꺼번에 많은 정보를 전송 할 수 있다. 디지탈 영상 전송은 1920년 최초로 시작되

*, **, **** 성균관대학교 전자공학과
Dept. of Eleetronics Engineering
Sung Kyun Kwan University
*** 대유공업전문대학
Dae Yeu Technical College
論文番號 : 88-39(接受 1988. 7. 18)

어 현재에 이르기까지 상당한 발전을 이루어 왔으나, 영상 전송에 있어서 일반적인 Monochrome T.V 영상인 경우에도 128 그레이 레벨로 512×512 배열을 이루어야 시각에 무리없이 받아 들여지므로 한 영상을 나타내는 데에는 약1.8 $\times 10^6$ bits 정도의 많은 데이타량이 요구되어 전송이나 저장에 상당한 어려움을 주고 있다.

이러한 많은 데이타량을 좀 더 적은 bit로 영상이나 영상이 포함하고 있는 정보를 나타내는 방법으로 영상 데이타 압축(Image Data Compression)이 연구되고 있다[3].

영상 데이타 압축 방법은 크게 공간 영역 코딩(Spatial Domain Coding)과 변환 코딩(Transform Domain coding)[1],[2]으로 나누어진다.

공간 영역 코딩은 계산 과정이 간단하여 실시간 처리가 가능하며 적은양의 메모리가 필요하지만 압축효과는 작다. 반면에 변환 코딩은 계산 과정이 복잡하고 많은 메모리가 필요하여 실시간 처리가 곤란하지만 압축효과는 뛰어나다.

공간 영역 코딩중에서 계산량이 적어 실시간 처리가 가능하고 하드웨어 구현이 용이한 BTC (Block Truncation Coding)[4]는 1979년 Delp와 Mitchell이 제안하여 현대까지 많은 연구가 이루어져 왔다. 이 방법은 영상에서 부화상(subblock)은 비슷한 그레이 레벨을 갖는 특징을 이용하여 영상을 부화상으로 나누고 각각의 부화상에 대하여 샘플 평균(Sample Mean)과 샘플 표준 편차(Sample Standard Deviation) 및 비트 평면(Bit Plane) 정보를 전송하여 원영상과 재구성된 영상사이에 1차 및 2차 모멘트를 유지하면서 수신측에서 이 변수들로 영상을 재구성하는 방법이다. 그러나 이 방법은 부화상내의 데이타가 2단계의 그레이 레벨값으로 양자화되기 때문에 부화상간의 블럭화 현상이 발생하며, 용도에 따라 화질의 요구조건(저화질 - 고화질)을 가변시킬 수 없고, 압축률이 2 bit/pixel 로 고정되어 있는 단점이 있다.

본 논문에서는 위의 BTC단점을 개선하기 위하여 quad tree 구조[5]를 이용하여 압축율을 향상시키고, 필요에 따라 화질을 가변시킬 수 있는 새로운 BTC방법을 제안하였다.

Ⅱ. 본 론

1. Block Truncation Coding 이론

Delp와 Mitchell이 제안한 BTC 알고리즘을 살펴보면 입력된 영상을 N*N (보통 N=4)크기의 겹치지 않는 부화상으로 나누어 준 다음, 나누어진 각 부화상에 대하여 1차 모멘트의 샘플 평균 $\overline{X}$와 2차 모멘트인 샘플 표준편차 σ를 아래 식에 의해서 구한다.

$$\overline{X}=\frac{1}{m}\sum_{i=1}^{m}X_i$$

$$\overline{X^2}=\frac{1}{m}\sum_{i=1}^{m}X_i^2 \qquad (1)$$

$$\overline{\sigma^2}=\overline{X^2}-\overline{X}^2$$

이 샘플평균 $\overline{X}$를 threshold값 Xth로 선택하여 부화상의 화소

Xi ≥ Xth 인 경우 출력은 "1"로
Xi < Xth 인 경우 출력은 "0"으로

변환하여 "0"과 "1"로 구성된 비트평면을 얻는다. 송신측에서는 각 부화상 단위로 샘플평균(8 bit)과 샘플 표준편차(8 bit) 및 "1", "0"으로 구성된 비트평면 (16 bit)을 전송하므로 전송율

$$R=(8+8+16)/16=2.00\,(\text{bit/pel})$$

이 된다.

수신측에서는 샘플평균 $\overline{X}$와 샘플 표준편차 σ 및 비트평면을 수신하여, 아래 관계식을 이용하여 비트 평면의 "1"은 b로, "0"은 a로 변환한다.

$$m\overline{X}=(m-q)a+qb$$

$$m\overline{X^2}=(m-q)a^2+qb^2$$

$$a=\overline{X}-\bar{\sigma}\sqrt{\left[\frac{q}{m-q}\right]} \qquad (2)$$

$$b=\overline{X}+\bar{\sigma}\sqrt{\left[\frac{m-q}{q}\right]}$$

여기서 m은 부화상의 화소의 갯수를 나타내고, q는 비트평면의 "1"의 갯수를 나타낸다.
이상의 방법으로 BTC가 이루어진다.

2. Quad Tree

영상을 동질성의 부화상으로 나타낼 수 있도록 분할하여 나타내는 구조를 tree라고 한다. 특히 4개의 부화상으로 분할하는 방식을 quad tree라 하며 주로 2진영상을 표현할 때 사용된다.

이 구조는 부화상의 크기와 갯수를 조절하여 화질을 변형할 수 있고, 부화상간의 인접성, 연결성등을 포함한 계층적인 관계를 쉽게 표현할 수 있는 장점을 가진다. 이러한 quad tree표현은 encoding, transform, searching quad tree 등과 같은 분야에서 많은 연구가 이루어지고 있다. Quad tree방식이 많은 region으로 구성된 영상을 비교적 간결하게 표현을 해주는 반면, region의 위치에 따라서 quad tree의 형태가 바뀌는 단점을 안고 있다. $2^m * 2^m$ 영역이 어느 위치에 있느냐에 따라서 하나의 node로, 또는 $f(2^m)$ node로 표현될 수도 있다는 것이다.

Quad tree를 구성하는 방식은 그림 1 에서와 같이 quad tree의 root는 전체 화상을 나타내며 이때 level은 n이다. 임의의 k번째 node에서 "1"과 "0"이 복합되어 있으면 4개의 $2^{k-1} * 2^{k-1}$ 부화상으로 나누어지고 이것을 parent와 4개의 son이라고 한다. 그리고 부화상내에 "0"이나 "1"만이 존재할 경우 더 이상 나누지 않으며 이것을 leaf라고 한다.

이렇게 나누어진 quad tree의 bit string encoding[6]은 다음과 같은 방식으로 이루어진다.

1) Node들은 root에서 시작하여 왼쪽에서 오른쪽 순으로 진행한다.

2) Parent node에는 code "1"을 할당한다.

3) Leaf node에는 code "0"과 leaf의 색깔(흑, 백)을 할당한다.

4) 마지막 단의 leaf에는 code "0"을 할당하지 않고 색깔만 할당한다.

이상의 방법으로 coding이 이루어지며 이 code를 적용하였을 경우의 quad tree와 그에 해당하는 code의 예를 그림 2에 보인다.

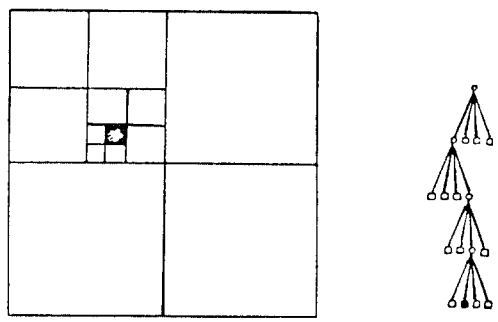


그림 1 영상 분할과 quad tree표현
An image and the corresponding quad tree

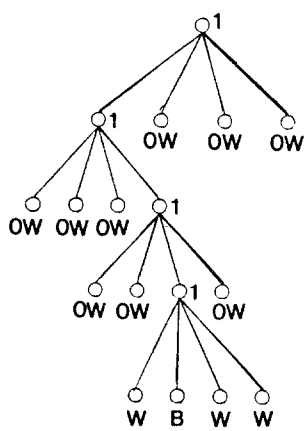


그림 2 Quad tree의 encoding
Encoding of quad tree

3. 제안한 BTC

영상은 그레이 레벨의 변화가 많은 영역(고주파 영역)과 그레이 레벨의 변화가 적은 영역(저주파 영역)으로 구성된다. 영상의 대부분은 그레이 레벨의 변화가 적은 영역으로 구성되어 있기 때문에 기존의 BTC에서와 같이 부화상 크

기를 일률적으로 N*N으로 나누는 것은 데이타 압축면에서 상당히 비효율적이다. 이점을 개선하기 위해서 본 논문에서는 그레이 레벨 변화가 적은 영역에서는 부화상 크기를 크게하고 그레이 레벨변화가 큰 영역에서는 부화상 크기를 작게하여 전체부화상 갯수를 줄임으로써 전송해야 할 샘플 평균, 샘플 표준편차의 갯수를 줄여 데이타 압축 효과를 얻는 방식을 사용하였다.

원래 quad tree는 2 진화상에서 흑색이나 백색만으로 구성된 영역인지, 흑색과 백색이 혼합된 영역인지를 구별하여 더 이상 분할할 것인가를 판별하는 방식이나, 본 논문에서는 표준편차를 이용하여 leaf 인지, 더 나누어 질 지를 판단하여 부화상을 처리하면서 quad tree 형태를 만든다. 이렇게 나누어진 각각의 최종 부화상(leaf)에 대하여 샘플 평균, 샘플 표준편차를 식(1)을 이용하여 구하고 비트평면을 만들어 전송한다. 이상의 방법으로 만들어진 부화상 정보를 quad tree encoding 방식을 이용하여 tree 구조정보, 샘플 평균, 샘플 표준편차, 비트 평면 순으로 전송하게 된다.

본 논문에서 제안한 BTC 방식의 이해를 돕기 위하여 아래에 coding 및 decoding 순서를 기술하였다.

STEP 1. 입력된 영상을 64*64 크기의 부화상으로 나눈다.

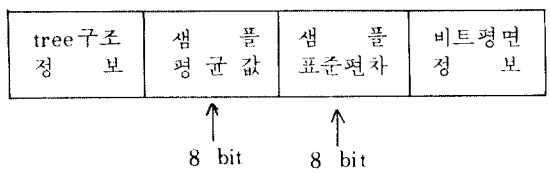
STEP 2. 각 부화상에 대하여 샘플 표준편차를 구하여 이 샘플 표준편차가 $\bar{\sigma}$>Th이면 부화상을 4 개의 더 작은 부화상으로 나누어 quad tree에는 "1"을 할당하고, $\bar{\sigma}\leq$ Th이면 부화상을 leaf로 처리하여 quad tree에 "0"을 할당한다.

STEP 3 : 이 과정을 가장 작은부화상의 크기가 4*.4 일 때까지 반복한다.

STEP 4. 이상의 과정에 의하여 마지막 단계까지 영상이 분할되면 각각의 부화상에 대하여 샘플 평균 $\bar{X}$와 샘플 표준편차 $\bar{\sigma}$를 구한다.

STEP 5. 부화상 내의 화소 Xi가 샘플 평균 값보다 크면 "1"로, 작으면 "0"으로 변환하여 비트평면을 만든다.

STEP 6. 송신측에서는 quad tree와 샘플 평균 $\bar{X}$, 샘플 표준 편차 $\bar{\sigma}$ 및 비트평면 정보를 아래와 같은 정보 형태로 전송한다.



STEP 7. 수신측에서는 위와 같은 정보를 수신하여 식(2)를 이용하여 복원 영상을 재구성한다.

이상의 방법을 이용한 결과는 기존의 BTC방식보다 최고 40% 이상의 압축율 개선을 얻을수 있었다. 영상 데이타를 압축함으로써 발생하는 영상의 왜곡(distortion)은 SNR을 이용하여 측정하였으며 사용된 식은 아래와 같다.

$$SNR\,(dB) = 10\log_{10} \frac{\sum\sum X_{ij}^2}{\sum_i \sum_j (X_{ij} \cdots Y_{ij})^2} \qquad (3)$$

4. 비트평면의 전송

비트평면은 "0"과 "1"로 이루어진 2 진 영상으로 구성되어 있다. 본 논문에서는 이 비트평면을 효과적으로 전송하기 위하여 2 진 영상 전송에 효율적인 Huffman run-length code[8]와 작은 크기의 비트평면에 유용한 lookup table 방식을 이용하였다.

1) Huffman run-length code

이 방식은 run-length code와 Huffman code를 결합시킨 형태로 먼저 run-length code를 적용시켜 run의 갯수를 구한 후 run의 발생 갯수에 대해 Huffman code를 적용시킨다. 표 1은 사용된 code table을 나타내며 이 방식을 이용하여 30-40%의 비트평면압축율개선을 얻을 수 있었다.

표 1 Huffman run-length code table

Number of Run-length	Number of Code bits	Huffman Code
1	2	11
2	3	101
3	3	011
4	4	0101
5	4	0011
6	5	01000
7	5	10010
8	5	01001
9	5	10000
10	5	10011
11	6	001000
12	6	100000
13	6	001010
14	6	001001
15	6	100001
16	6	000011
17	6	001011
18	7	0000000
19	7	0000100
20	7	0000010
21	7	0001110
22	7	0000001
23	7	0000101
24	7	0000011
25	7	0001111
26	8	00011000
27	8	00011010
28	8	00011001
29	8	00011011
over 30	5 + 8	00010+8bits

2) Lookup table 방식

Huffman run-length code는 큰 크기의 부화상에서는 유용하나 작은 크기의 부화상 에서는 overhead로 인하여 비효율적이므로 lookup table방식[9]를 이용하였다.

Lookup table방식은 그림3의 4 * 4 부화상에서 8개의 원으로 표시한 독립적인 화소를 선택한 후 아래 논리식을 이용하여 나머지 8개의 화소를 추정한다.

B=1 iff [(A and D)=1] or [(F and J)=1]
C=1 iff [(A and D)=1] or [(G and K)=1]
E=1 iff [(A and M)=1] or [(F and G)=1]
H=1 iff [(D and P)=1] or [(F and G)=1]
I=1 iff [(A and M)=1] or [(J and K)=1]
L=1 iff [(D and P)=1] or [(J and K)=1]
N=1 iff [(M and P)=1] or [(F and J)=1]
O=1 iff [(M and P)=1] or [(G and K)=1]

이 방식은 수평, 수직 그리고 대각선 경계를 그대로 유지하게 되어 오차가 발생하더라도 큰 무리가 없으며 실제 적용시 95%이상의 복구 성능을 나타내었고 압축율도 0.5 bit/pixel로 만족할 수 있었다.

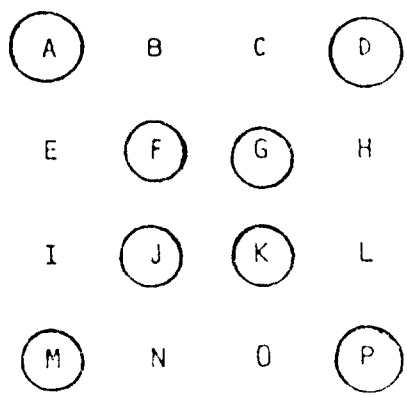
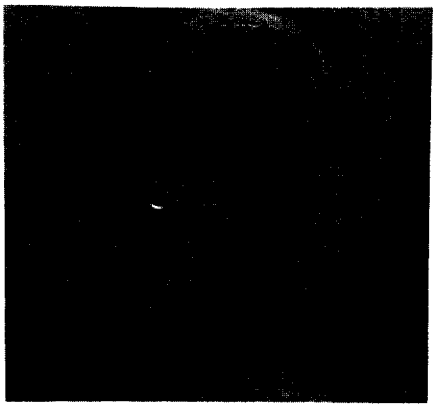


그림3 원으로 표시된 화소는 독립 화소
Circled pels are independent variable

5. 실험 결과

실험에서는 크기가 256*256 pixel이고, 그레이 레벨이 256인 〈girl〉영상과 〈cronkite〉영상을 이용하였으며, 이것을 그림4-(a), (b)에 보였다.

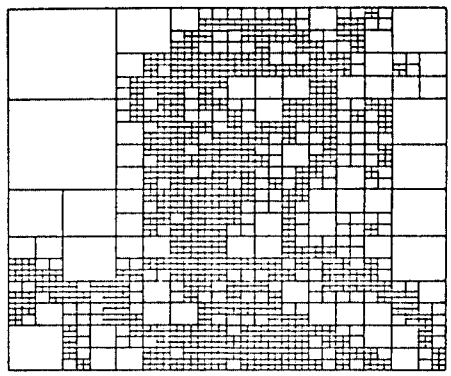
이 영상의 threshold 값을 전체 영상 표준편차의 0.2, 0.3, 0.5 값으로 선택한후 quad tree 구조를 구성하면서 부화상의 표준편차가 threshold 값보다 큰 부화상은 4개의 더 작은 부화상으로 분할하였으며, threshold 값보다 작은 부화상은 분할하지 않았다. 이와 같은 방식을 이용한 경우에 각 threshold 값에 따른 분할 영상을 그림 5-(a), (b)에 보였다.



a

b.

그림 4 원 영상
a. girl b. cronkite
Original Image



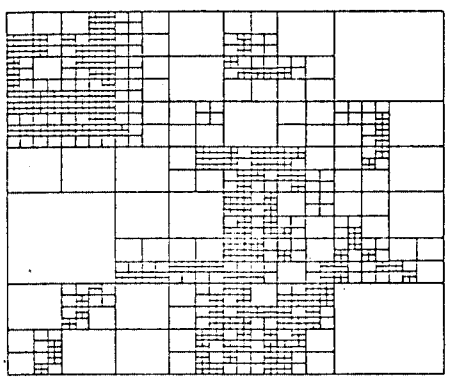


그림 5 분할된 영상(Th=0.3)
a. girl b. cronkite
Dividing blocks

표 2 부화상 크기별 갯수
Numbers of subblocks with the block size

영상 \ 구분		4 * 4	8 * 8	16 * 16	32 * 32	64 * 64	계
GIRL	BTC	4,096	–	–	–	–	4,096
	0.2	2,280	158	22	5	2	2,467
	0.3	1,352	206	48	10	2	1,618
	0.5	676	207	58	10	4	955
CRON	BTC	4,096	–	–	–	–	4,096
	0.2	1,124	203	39	16	2	1,384
	0.3	692	179	56	16	3	946
	0.5	428	145	41	22	4	640

그림 5 의 분할된 영상에서 볼 수 있듯이 배경 부분과 같이 그레이 레벨의 변화가 적은 영역에서는 부화상의 크기가 크게, 경계 부분과 같이 그레이 레벨의 변화가 큰 영역에서는 작게 나누어짐을 알 수 있다. 또한 threshold값을 크게 선택할수록 크기가 큰 부화상의 갯수가 증가한다. 각각의 threshold값에 대한 부화상 크기별 갯수는 표 2 에 나타내었다.

표 2 의 부화상 크기별 갯수에서 기존의 BTC 방식보다 threshold 값 변화에 따라 15~60%의 부화상 갯수가 감소함을 알 수 있다. 이렇게 부화상의 갯수가 줄어듬에 따라 전송하여야 할 평균값, 표준 편차가 부화상 갯수의 감소량 만큼 줄어들어 압축면에서 상당히 효율적임을 알 수 있다.

이렇게 분할된 영상의 각 부화상의 샘플 평균 및 샘플 표준편차를 각각 8 bit로 양자화하고 이것을 이용하여 비트 평면을 식(1)을 이용하여 구하였다.

이 정보를 quad tree encoding 방식을 이용하여 부화상의 크기, 위치, 샘플 평균, 샘플 표준편차 및 비트 평면 순으로 전송하였다. Quad tree를 이용함으로써 발생하는 추가 정보는 약 0.009 bit/pixel로서 비교적 적음을 알 수 있었다. 비트평면의 전송은 압축율과 처리 시간을 줄이기 위하여 크기가 8 * 8 보다 큰 부화상에 있어서는 그레이 레벨 변화가 적은 영역으로 비트 평면내의 변화 역시 적으므로 Huffman run-length code를 이용하였고, 8 * 8 보다 작은 부화상은 그레이 레벨의 변화가 큰 고주파 성분이므로 lookup table 방식을 이용하여 효율적으로 압축율을 개선시킬 수 있었다.

전체 정보 전송에 필요한 데이타량을 표 3 에 보였다.

표 3 의 전송에 필요한 데이타량으로 알 수 있듯이, quad tree 방식을 이용함으로써 아주 적은 잉여 비트가 추가됨에 반해 평균값과 표준편차를 전송하는데 필요한 비트 수는 상당히 감소하였다. 이는 기존의 BTC 방식에서 필요한 비트 수의 35~59% 정도이다.

표 3 전송에 필요한 데이타량
Numbers of datas for transmission

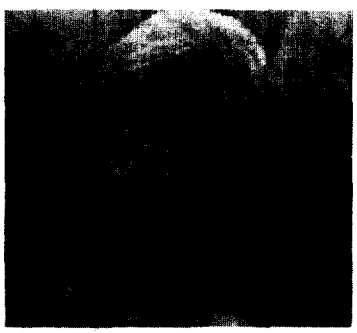
영상 \ 구분 Th		비트평면	평균값 표준편차	Quad tree	총 계
GIRL	BTC	65,536	65,536	0	131,072
	0.2	37,548	39,472	995	78,015
	0.3	38,566	25,888	804	65,176
	0.5	40,299	15,280	597	56,176
CRON	BTC	65,536	65,536	0	131,072
	0.2	39,277	22,144	702	62,123
	0.3	38,775	15,136	550	54,461
	0.5	35,406	10,240	408	46,054

수신측에서는 수신된 정보를 이용하여 영상을 복원하게 되며, 각각의 threshold 값에 대한 복원된 영상을 그림 6 과 그림 7 에 보였고, 데이타를 압축함으로 인하여 발생하는 영상의 왜곡은 식(3)의 SNR을 이용하여 평가하였다. 각 threshold값에 따른 압축율과 SNR을 표 4 에 나타내었다.

a. BTC

b. Th=0.2

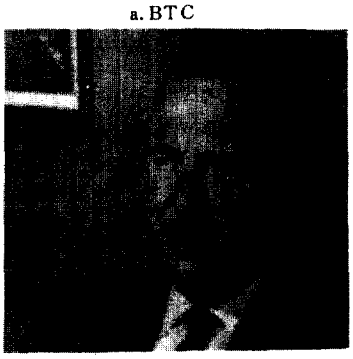
c. Th=0.3



d. Th=0.5

그림 6 복원된 girl 영상
Reconstructed Image "girl"

a. BTC



b. Th=0.2

c. Th=0.3

d. Th=0.5

그림 7 복원된 cronkite 영상
Reconstructed Image "cronkite"

표 4 압축율과 SNR
bit-rate and SNR

영상 / 구분 Th	GIRL		CRONKITE	
	압축율 (bit/pel)	SNR (db)	압축율 (bit/pel)	SNR (db)
BTC	2.0	29.32	2.0	25.40
0.2	1.190	26.00	0.947	21.98
0.3	0.995	25.25	0.831	21.40
0.5	0.857	23.62	0.702	20.02

이상에서 기존의 BTC 방식과 비교할 때 quad tree를 이용함으로써 비교적 좋은 화질과 압축율을 얻을 수 있었으며, 또한 threshold값을 가변시킴으로써 화질의 요구 조건(저화질 - 고화질)을 가변시킬 수 있어 BTC의 단점들을 개선할 수 있었다.

Ⅲ. 결 론

본 논문에서는 영상 데이타 압축 방법의 하나인 BTC를 기본으로 해서 영상의 대부분은 그레이 레벨 변화가 적은 영역으로 구성되어 있다는 점을 이용하여 그레이 레벨의 변화가 적은 영역에서는 부화상의 크기를 크게 하고, 그레이 레벨의 변화가 큰 영역에서는 부화상 크기를 작게 하여 부호화하는 방법을 제안하였다. 기존의 BTC 방식에서는 부화상의 크기를 일률적으로 4 * 4 크기로 영상을 분할하여 처리하기 때문에 데이타 압축면에서 상당히 비효율적이다. 또 용도에 따라 화질의 요구 조건(저화질-고화질)을 가변시킬 수 없고, 압축율이 2 bit/pixel로 고정되어 있는 단점이 있다. 이러한 단점을 개선하고자 주로 2진 영상에서 이용되는 quad tree 개념을 적용하여 전체 영상의 부화상 갯수를 줄임으로써 전송해야 할 평균값, 샘플 표준 편차의 갯수를 줄였다.

비트 평면 전송시에는 8 * 8 이상 크기의 부화상은 변화가 적은 부분이므로 저주파 영역에 적합한 Huffman run-length code를, 8 * 8 이하 크기의 부화상은 경계선과 같이 변화가 큰 부분이므로 고주파 영역에 적합한 lookup table 방식을 이용하여 효율적으로 전송하였다.

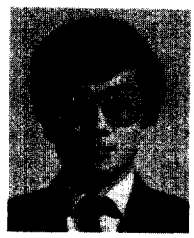
이상의 방법을 이용한 결과 기존의 BTC에 비하여 35-60% 정도의 압축 효과를 얻을 수 있었다. 데이타 압축으로 인한 영상의 왜곡은 SNR을 이용하여 평가하였으며 압축율에 비해 영상의 왜곡은 만족할만한 것이었다. 본 논문의 방법을 이용할 때 개략적인 형태의 영상이 요구될 때는 threshold 값을 0.5 이상으로, 일반적인 형태의 영상이 요구될 때는 threshold 값을 0.3으로, 세밀한 영상이 요구될 때는 0.2 이하의 threshold 값으로 선택하는 것이 적당하다고 판단된다.

본 논문에서는 기존의 BTC와 비교하기 위하여 Delp와 Mitchell의 방식을 기준으로 실험하였으며, 평균값, 샘플 표준 편차를 묶어 10비트로 양자화하는 Healy와 Mitchell의 BTC 방식[10]을 이용하면 더욱 높은 데이타 압축이 이루어 지리라 기대된다.

參 考 文 獻

1. A.N. Netravali and J.O. Limb, "Picture coding: a review", *Proc. IEEE*, Vol.68, No.3, pp.366-406, Mar., 1980.
2. A.K. Jain, "Image data compression: a review", *Proc. IEEE*, Vol.69, No.3, pp.349-389, Mar., 1981.
3. Rafael C. Gonzalez and Paul Wintz, *Digital Image Processing*, Addison-Wesley, 1977.
4. Edward J. Delp and O. Robert Mitchell, "Image compression using block truncation coding", *IEEE Trans. Comm.*, Vol.Com-27, No.9, pp.1335-1342, Sept., 1979.
5. Charles R. Dyer, "The space efficiency of quad trees", *Computer Graphics and Image Processing* 19, pp.335-348, 1982.
6. Markku Tamminen, "Encoding pixel trees", *Computer Graphics and Image Processing* 28, pp.44-57, 1984.
7. D. Jacques Vaisey and Allen Gersho, "Variable block-size image coding", *Proc. ICASSP*, pp.1051-1054, 1987.
8. Thomas J. Lynch, *Data Compression, Techniques and Applications*, Lifetime Learning Press, 1985.
9. O. Robert Mitchell and Edward J. Delp, "Multilevel Graphics Representation Using Block Truncation Coding", *Proc. IEEE*, Vol.68, No.7, pp.868-873, July, 1980.
10. Donald J. Healy and O.R. Mitchell, "Digital video bandwidth compression using block trancation coding", *IEEE Trans. Comm.*, Vol.COM-29, No.12, pp.1809-1817, Dec., 1981.

白 仁 基(In Ky BAIK) 準會員
1959年12月14日生
1986年 2月：成均館大學校 電子工學科 卒業
1986年 1月～1986年11月：現代電子SYSTEM 研究所
1987年 3月～現在：成均館大學校 電子工學科 大學院 碩士課程



金 海 洙(Hae Soo KIM) 準會員
1960年11月20日生
1986年 2月：成均館大學校 電子工學科 卒業
1988年 2月：成均館大學校 大學院 電子工學科 卒業(工學碩士)
1988年 3月～現在：成均館大學校大學院 電子工學科 博士課程

趙 成 桓(Seong Hwan CHO) 正會員
1957年 3月17日生
1980年 2月：成均館大學校 電子工學科 卒業
1982年 2月：成均館大學校 大學院 電子工學科 卒業(工學碩士)
1987年 3月～現在：成均館大學校大學院 電子工學科 博士課程
1982年 9月～1985年 7月：海軍士官學校 教授部 專任講師
1985年 9月～現在：大有工業專門大學 專任講師

李 根 泳(Keun Young LEE) 正會員
1947年12月30日生
1973年 2月：全南大學校 電氣工學科 卒業
1975年 2月：漢陽大學校 大學院 電子工學科 卒業(工學碩士)
1978年 8月：漢陽大學校 大學院 電子工學科 卒業(工學博士)
1979年 3月～1980年 2月：Denmark工科大學(碩究)
1987年 9月～1988年 8月：英國Loughboroough大學(碩究)
1977年 3月～1981年 8月：光云工大 助教授
1981年 9月～現在：成均館大學校 電子工學科 副教授