

畫像壓縮을 위한 適應 BTC 方法의 提案

(A New Proposal of Adaptive BTC for Image Data Compression)

張 基 崇*, 吳 性 穆*, 李 榮 哲**

(Ki Soong Jang, Seong Mock Oh and Young Choul Lee)

要 約

본 논문에서는 기존의 BTC 방법을 개선한 새로운 ABTC(adaptive block truncation coding) 방법을 제안하였다. ABTC 알고리즘은 화상데이터의 부분적인 특성에 따라 압축방식을 선택적으로 적용함으로써 높은 압축률로 우수한 화질의 재생화면을 얻을 수 있다. ABTC 화상압축방식은 압축 알고리즘이 간단하면서도 높은 압축률을 얻을 수 있는 특징이 있다. 이 알고리즘을 사용하면 칼라화상을 1.0 (bit/pel)에서 2.56(bit/pel)의 데이터 부호율 범위내에서 임의로 코딩할 수 있으며 원래의 화상에 비해 화질의 큰 손상없이 높은 압축률(1.3~1.5bit/pel)을 얻을 수 있다.

Abstract

This paper proposes a new ABTC (Adaptive Block Truncation Coding) algorithm which improves the BTC algorithm for image data compression.

A new adaptive block truncation coding which adopts a selective coding scheme depending on the local characteristics of an image has been described.

The characteristics of the ABTC algorithm can be summarized as high compression ratio and the algorithm simplicity.

Using this algorithm, color images can be coded at a variable bit rate from 1.0 (bit/pel) to 2.56 (bit/pel) and high compression rate (1.3-1.5 bit/pel) can be achieved without conspicuous image degradation compared with original images.

I. 서 론

화상데이터 압축은 화상신호의 통계적 성질과 인간의 시각특성을 이용하여 화상신호의 冗長性(redundancy)을 제거하고 전송 비트레이트를 저하 시키는

것을 목표로 하고 있다. 화상신호가 갖고 있는 冗長性을 제거하여 화상데이터를 압축하는 방법으로는 여러가지가 있으나 예측부호화(predictive coding)와 변환부호화(transform coding) 방식이 대표적으로 널리 사용되어 왔다. 예측부호화는 주변화소의 값으로 다음에 부호화하는 화소의 값을 예측하여 실제의 표본치와의 차이를 부호화하는 방법이고, 변환부호화는 화면을 적당한 수의 화소로 구성된 블록으로 분할한 다음 각 블록에 대해 직교변환을 하고 저주파 영역에 많은 비트를 할당하여 화상을 압축하는 방법이다.

*正會員, 韓國電氣通信公社 事業支援團
(Korea Telecommunication Authority Research Center)

接受日字: 1988年 9月 20日

예측부호화 방법은 압축률은 떨어지나 시스템 구현이 용이하고 변환부호화는 높은 압축률을 얻을 수 있지만 계산량이 방대하며 시스템이 복잡해지는 단점이 있다.^[1]

BTC(block truncation coding)는 계산량이 적고 상당히 우수한 화질을 갖는 화상압축방법으로서 전송 채널에러에 강한 장점을 가지고 있다.^[2] 그러나 기존의 BTC 방법은 칼라화상에 대해 약 2.13~3(bit/pel) 정도의 압축률을 가지므로 실제 응용분야에 적용시키기 위해서는 이를 더 압축하여 보다 낮은 비트레이트를 실현할 필요성이 있다.^[3]

본 논문에서는 기존의 BTC 방법을 개선한 ABTC(adaptive BTC) 방법을 새로이 제안하였다. ABTC 방법은 화상데이터의 부분적 특성에 관계없이 일률적인 압축알고리즘을 적용하는 기존의 방법을 탈피하여 인간의 시각특성과 화상데이터의 부분적인 특성에 따라 압축방식을 선택적으로 적용함으로써 기존의 방식보다 높은 압축률로 더 우수한 화질을 얻을 수 있다. 또한 ABTC 알고리즘은 여러가지 칼라 화상에 대해 비트레이트를 1.0(bit/pel)에서 2.56(bit/pel)의 범위내에서 임의로 조정할 수 있는 특징이 있다. 본 논문에서는 여러가지 압축률에 대해 ABTC와 기존의 BTC의 성능을 비교 검토하였다.

II. 기존의 BTC 알고리즘

1. 기본 AMBTC 이론

BTC(block truncation coding) 알고리즘은 1979년 Delp와 Mitchell에 의해 처음 발표된 이래^[1] 1984년 Lema와 Mitchell에 의해 BTC 보다 계산이 간단하고 성능이 다소 우수한 AMBTC(absolute moment block truncation coding) 방법이 발표되었다.^[3] BTC 방법은 우선 디지털화된 화상을 (n×n) 크기의 겹치지 않는 여러개의 블록으로 나눈다음 각 블록의 비트평면과 평균 및 표준모멘트(standard moment)를 전송하는 방식이고, AMBTC는 각 블록의 비트평면과 평균 및 절대모멘트(absolute moment)를 전송하는 방식이다. 일반적으로 AMBTC는 BTC 방식보다 계산이 간단하고 성능이 다소 우수한 것으로 알려져 있다.

AMBTC 방법에 대해 자세히 살펴보면^[3] 우선 디지털화된 화상을 (n×n) 크기의 겹치지 않는 블록으로 나눈다음 각 블록에 대해 평균($\bar{\eta}$)과 절대모멘트(α)를 구한다. 평균($\bar{\eta}$)과 절대모멘트(α)는 다음의 (1)식과 (2)식으로 표시할 수 있다.

$$\bar{\eta} = \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\alpha = \frac{1}{m} \sum_{i=1}^m |x_i - \bar{\eta}| \quad (2)$$

여기서 m은 한 블록내의 화소의 갯수(n×n)이고 x_i는 i번째 화소의 gray level이다. 비트평면(bit plane)은 평균($\bar{\eta}$)보다 더 크거나 같은 gray level을 갖는 화소를 1, 평균($\bar{\eta}$)보다 더 작은 gray level을 갖는 화소를 0으로 양자화한 것이다. 이렇게 부호화한 후 송신측에서 각 블록의 $\bar{\eta}$, α 및 비트평면을 수신측으로 전송하면 수신측에서는 다음의 (3)식 및 (4)식과 같이 각 블록에 대해 비트평면이 0인 화소는 a로, 1인 화소는 b로 복원하게 된다.

$$m\bar{\eta} = qb + (m-q)a \quad (3)$$

$$m\alpha = q(b - \bar{\eta}) + (m-q)(\bar{\eta} - a) \quad (4)$$

여기서 q는 한 블록에 대해 비트평면이 1인 화소의 갯수이다.

(3), (4)식으로 a와 b를 구하면 (5)식과 (6)식으로 나타낼 수 있다.

$$a = \bar{\eta} - \frac{\gamma}{m-q} \quad (5)$$

$$b = \bar{\eta} + \frac{\gamma}{q} \quad (6)$$

여기서 $\gamma = \frac{m\alpha}{2}$ 이다.

(5), (6)식의 의미는 평균($\bar{\eta}$)을 기준으로 하여 a는 평균($\bar{\eta}$)보다 작은 화소들의 평균값이고 b는 평균($\bar{\eta}$)보다 크거나 같은 화소들의 평균값을 나타낸다.

2. 칼라 화상에 대한 AMBTC 방법의 적용

칼라화상 데이터는 일반적으로 R, G, B 신호로 구성 되어 있으나 R, G, B 신호 사이의 상관관계(correlation)가 크므로 칼라화상에 대한 AMBTC 방법은 Y, I, Q 신호를 사용하였다.^[3] Y, I, Q 신호와 R, G, B 신호 사이의 관계는 다음식(7)과 같다.^[3]

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (7)$$

Y, I, Q 신호에 있어서 인간의 시각특성은 Y 신호에 가장 민감하고 I, Q의 순서로 민감도가 떨어지게 된다. 칼라화상에 대한 AMBTC 방법은 높은 압축률을 얻기 위하여 인간의 시각특성에 덜 민감한 I 신호 및 Q 신호는 subsampling하여 전송한다. 즉 I 신호에 대해서는 2×2 화소로 구성된 window의 평균을 구하고 Q 신호에 대해서는 4×4 화소로 구성된 window의 평균을 구하여 이러한 window를 마치 한개의 화소인것 처럼 취급하여 subsampling한 효과를 얻는다. 따라서 Y 신호는 16화소(1block)에 대해, I 신호는 64

화소(4blocks)에 대해, 그리고 Q 신호는 256 화소(16 blocks)에 대해 각각 평균과 절대모멘트에 10비트, 그리고 비트평면에 16비트를 할당하여 전송한다. 따라서 Y 신호는 26/16(bit/pel), I 신호는 26/64(bit/pel), 그리고 Q 신호는 26/256(bit/pel)의 비트부호율을 가지므로 결과적으로 칼라화상에 대한 AMBTC 방법은 2.13(bit/pel) (26/16+26/64+26/256)의 비트레이트를 가진다.

III. ABTC 알고리즘

인간의 시각특성은 그림의 경계부분에 대해서 민감하지만 인접한 화소의 상관관계가 큰 배경부분에 대해서는, 덜 민감하다.^[4] ABTC(adaptive block truncation coding)방법은 인간의 이러한 시각특성을 이용하여 그림의 경계부분에 대해서는 AMBTC 방법과 같이 각 블록에 대한 평균, 절대모멘트, 그리고 비트평면에 관한 정보를 전송하지만, 배경부분에 대해서는 평균에 관한 정보만 전송하고 절대모멘트와 비트평면에 관한 정보는 전송하지 않으므로써 높은 압축률을 얻을 수 있다. 또한 AMBTC 방법은 Y, I, Q 신호를 사용하는데 비해 ABTC 방법은 비트평면을 구성하기 위해 Y, I, Q 신호중 인간의 시각특성에 가장 민감한 Y 신호만을 사용하고 칼라신호는 RGB 신호를 그대로 사용한다. 따라서 기존의 AMBTC 방법은 송신측에서 RGB/YIQ 신호변환과 수신측에서 YIQ/RGB 신호변환이 필요하므로 계산량이 많아지지만, ABTC 방법은 전송측에서 RGB/Y 신호 변환만 필요하므로 AMBTC 방법에 비해 계산량이 대폭 줄어들게 된다.

1. 비트평면의 구성

우선 전체화면(256×256)을 여러개의 겹치지 않는 블록(4×4)으로 분할한 다음 각 블록내의 16화소에 대해 (7)식을 이용하여 R, G, B 신호를 Y 신호로 변환한다. 다음에는 아래의 (8)식과 같이 한 블록내의 16화소에 대한 휘도신호(luminance signal)의 평균 \bar{Y} 를 구하고 평균(\bar{Y})보다 크거나 같은 휘도를 갖는 화소는 1, 작은 휘도를 갖는 화소는 0으로 양자화하여 비트평면을 구성한다.

$$\bar{Y} = \frac{\sum_{i=1}^{16} Y_i}{16} \quad (8)$$

여기서 Y_i 는 i 번째 화소의 휘도이다.

2. 배경블럭과 경계블럭의 구분

각 블록의 비트평면을 구성한 후 한 블록내에서 비트평면이 1인 화소와 0인 화소에 대해 아래식(9), (10)과 같이 각각의 휘도신호 평균 $\bar{Y1}$ 와 $\bar{Y0}$ 를 구한다.

$$\bar{Y1} = \sum_{\text{for } Y_i = \bar{Y}} Y_i/q \quad (9)$$

$$\bar{Y0} = \sum_{\text{for } Y_i < \bar{Y}} Y_i/p \quad (10)$$

단, q : 비트평면이 1인 화소의 갯수

p : 비트평면이 0인 화소의 갯수

$$(p+q=16)$$

여기서 각 블록에 대해 $\bar{Y1}$ 에서 $\bar{Y0}$ 를 뺀 dY 를 구하고, 각 블록이 배경부분인가 경계부분인가를 결정 결정하기 위한 휘도신호 문턱치(luminance threshold) Y_{th} 를 정하여 $dY < Y_{th}$ 인 블록은 배경블럭으로 그리고 $dY \geq Y_{th}$ 인 블록은 경계블럭으로 구분한다. 여기서 Y_{th} 의 값은 dY 값의 범위내에서 임의로 정할 수 있으며 이에 대한 구체적인 실험결과와는 IV. 2에 나타내었다.

3. 전송데이터 구조

배경부분에 해당하는 블록은 다음의 (11), (12) 및 (13)식과 같이 블록내의 (4×4) 화소에 대한 R, G, B 신호의 평균 $\bar{R}, \bar{G}, \bar{B}$ 를 전송한다.

$$\bar{R} = \frac{\sum_{i=1}^{16} R_i}{16} \quad (11)$$

$$\bar{G} = \frac{\sum_{i=1}^{16} G_i}{16} \quad (12)$$

$$\bar{B} = \frac{\sum_{i=1}^{16} B_i}{16} \quad (13)$$

경계부분에 해당하는 블록은 그 블록의 비트평면과 비트평면이 1인 화소들의 R, G, B 평균, 즉 (14),(15) (16)식의 $\bar{R1}, \bar{G1}, \bar{B1}$ 값, 그리고 비트평면이 0인 화소들의 RGB 평균, 즉 (17), (18), (19)식의 $\bar{R0}, \bar{G0}, \bar{B0}$ 를 전송하여야 한다.

$$\bar{R1} = \sum_{\text{for } Y_i = \bar{Y}} R_i/q \quad (14)$$

$$\bar{G1} = \sum_{\text{for } Y_i = \bar{Y}} G_i/q \quad (15)$$

$$\bar{B1} = \sum_{\text{for } Y_i = \bar{Y}} B_i/q \quad (16)$$

$$\bar{R0} = \sum_{\text{for } Y_i < \bar{Y}} R_i/p \quad (17)$$

$$\bar{G0} = \sum_{\text{for } Y_i < \bar{Y}} G_i/p \quad (18)$$

$$\bar{B0} = \sum_{\text{for } Y_i < \bar{Y}} B_i/p \quad (19)$$

그러나 일반적으로 한 블록내에서 $\bar{R1}, \bar{G1}, \bar{B1}$ 값과 $\bar{R0}, \bar{G0}, \bar{B0}$ 값 사이에는 상관관계가 크므로 경계블

력에 있어서 비트평면이 0에 해당하는 화소에 대해서는 $\overline{R0}, \overline{G0}, \overline{B0}$ 값을 직접 전송하는 것보다 (20), (21), (22)식과 같이 $\overline{R1}, \overline{G1}, \overline{B1}$ 값과 $\overline{R0}, \overline{G0}, \overline{B0}$ 의 차이값인 dR, dG, dB 를 양자화하여 전송하면 압축률을 더 높일 수 있다.

$$dR = \overline{R1} - \overline{R0} \quad (20)$$

$$dG = \overline{G1} - \overline{G0} \quad (21)$$

$$dB = \overline{B1} - \overline{B0} \quad (22)$$

여기서 dR, dG, dB 를 효율적으로 양자화하기 위해 dR, dG, dB 의 분포도(histogram)^[6]를 구해보면 그림에 따라 다소 차이가 있으나 대체로 다음그림 1과 같은 분포를 갖는다.

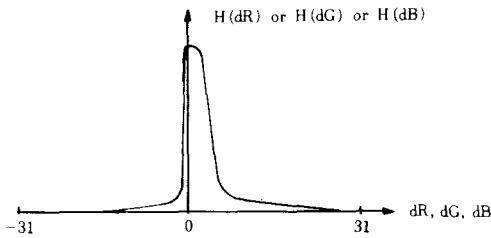


그림 1. dR, dG, dB 의 히스토그램
Fig. 1. Histogram of dR, dG, dB .

따라서 dR, dG, dB 는 각각 표 1과 같이 8가지 레벨로 비균일 양자화 할 수 있다.^[6]

ABTC 알고리즘은 경계블럭과 배경블럭에 대해 각각 전송데이터 구조가 다르므로 송신측에서 각 블럭이 경계블럭인가 배경블럭인가를 나타내주는 over-

표 1. dR, dG, dB 의 양자화
Table 1. Quantization of dR, dG, dB .

dR (or dG or dB)	대표값	코딩 (coding)
-32 ~ 0	0	0 0 0
1	1	0 0 1
2	2	0 1 0
3	3	0 1 1
4	4	1 0 0
5 ~ 7	5	1 0 1
8 ~ 12	8	1 1 0
13 ~ 31	13	1 1 1

head bit가 각 블럭마다 1비트씩 필요하게 된다. 배경블럭과 경계블럭에 대한 전송데이터 구조는 각각 그림 2, 그림 3과 같다.

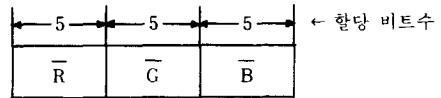


그림 2. 배경블럭의 전송데이터 구조
Fig. 2. Transmission data structure of background blocks.

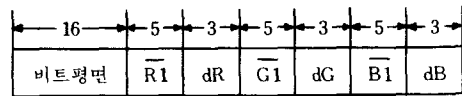


그림 3. 경계블럭의 전송데이터 구조
Fig. 3. Transmission data structure of contour blocks.

4. 화상데이터 압축률

ABTC 방법의 화상데이터 압축률은 III-2의 휘도신호의 문턱치 Y_{th} 에 달려있다. 즉 같은 그림에 대해서 Y_{th} 를 크게하면 비트레이트를 낮출 수 있지만 화질은 떨어지고, 반대로 Y_{th} 를 작게하면 비트레이트는 높아지지만 화질은 향상된다. 그리고 같은 Y_{th} 값에 대해 단순한 그림은 많은 수의 블럭이 배경블럭이 되므로 비트레이트가 저하되고 복잡한 그림은 상대적으로 비트레이트가 높아지게 된다.

만일 한 화면내에서 전체 블럭수를 M , 배경블럭수를 S 라하면 그림 2와 그림 3에서와 같이 배경블럭은 15비트, 경계블럭은 40비트 그리고 각 블럭에 대한 overhead비트가 1비트씩 필요하므로 비트레이트 C 는 다음의 (23)식과 같이 구할 수 있다.

$$C = \frac{15 \times S + 40 \times (M - S) + M}{16 \times M} = 2.5625 - 1.5625r \text{ (bit/pel)} \quad (23)$$

여기서 $r = S/M$

(23)식에서 r 은 0에서 1사이의 값을 가지므로 ABTC 알고리즘의 비트레이트는 1.0에서 2.5625 (bit/pel) 사이의 값을 가진다. 그림 4에서는 이상 설명한 ABTC 알고리즘의 흐름도를 나타내었다.

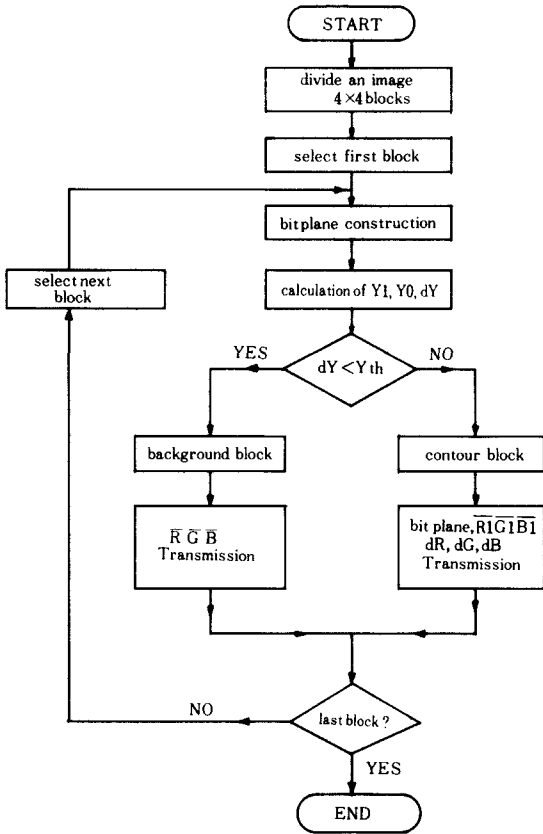


그림 4. ABTC 알고리즘 흐름도
Fig. 4. Flowchart of ABTC algorithm.

IV. 실험결과 및 검토

컴퓨터 simulation을 통하여 기존의 AMBTC 알고리즘과 본 논문에서 제안한 ABTC 알고리즘의 성능을 비교분석하였다. 실험화상은 (256×256) 표준화상을 사용하였으며 원화상 R, G, B 당 비트 수는 각각 5bit이고 ABTC의 경우 블록의 크기는 (4×4)로 하였다.

1. AMBTC 실험결과

AMBTC에 대한 실험은 앞의 II.2, 칼라화상에 대한 AMBTC 방법에 따라 실험하였다. 원화상에 대한 재생화상의 화질을 평가하기 위하여 R, G, B 각각에 대해 (24)식과 같이 RMSE(relative mean square error)를 사용하였다.^[7]

$$\epsilon_R^2 = \frac{(1/N^2) \sum_x \sum_y [R(x, y) - \hat{R}(x, y)]^2}{(1/N^2) \sum_x \sum_y R^2(x, y) - [(1/N^2) \sum_x \sum_y R(x, y)]^2} \quad (24)$$

여기서

ϵ_R^2 : R 신호에 대한 RMSE

$N \times N$: 원화상의 크기 (resolution)

$R(x, y)$: 원화상

$\hat{R}(x, y)$: 재생화상

G 신호와 B 신호에 대해서도 (24)식과 같은 방법으로 RMSE를 계산할 수 있으며 총 RMSE(total RMSE) ϵ_T^2 는 다음의 (25)식과 같이 정의된다.

$$\epsilon_T^2 = \frac{\epsilon_R^2 + \epsilon_G^2 + \epsilon_B^2}{3} \quad (25)$$

원화상(그림5(a))에 대한 AMBTC 재생화상(그림5(b))의 RMSE는 표 2와 같다.

표 2. AMBTC의 RMSE
Table 2. RMSE of AMBTC.

비트레이트 (bit/pel)	ϵ_R^2	ϵ_G^2	ϵ_B^2	ϵ_T^2
2.13	8.86%	4.83%	7.70%	7.13%

2. ABTC 실험결과

ABTC의 경우 여러가지 휘도신호 문턱치(Y_{th})에 하여 실험을 수행하였다. ABTC 재생화상의 화질을 평가하기 위하여 R, G, B 각각에 대해 (25)식의 RMSE를 사용하였으며 그 결과는 표 3과 같다.

표 3. ABTC의 RMSE
Table 3. RMSE of ABTC.

Y_{th}	비트레이트 (bit/pel)	ϵ_R^2	ϵ_G^2	ϵ_B^2	ϵ_T^2
0.34	2.13	3.87%	2.78%	4.09%	3.58%
0.8	1.71	4.02%	2.88%	4.15%	3.68%
1.2	1.57	4.45%	2.93%	4.21%	3.86%
2.0	1.33	5.07%	3.36%	4.58%	4.34%

그림 5(c)와 (d)에서는 비트레이트가 각각 1.71(bit/pel), 1.33(bit/pel)에 해당하는 ABTC 재생화면을 나타내었다.

3. 실험결과 검토

본 실험결과와 같이 ABTC는 기존의 AMBTC 방법보다 더 낮은 비트레이트로 더 좋은 화질을 얻을 수 있었다. AMBTC 방법은 우선 송신측의 RGB/Y IQ 변환과 수신측에서의 역변환과정에서 시간이 많이 소요될 뿐 아니라 I, Q 신호를 subsampling하여 전

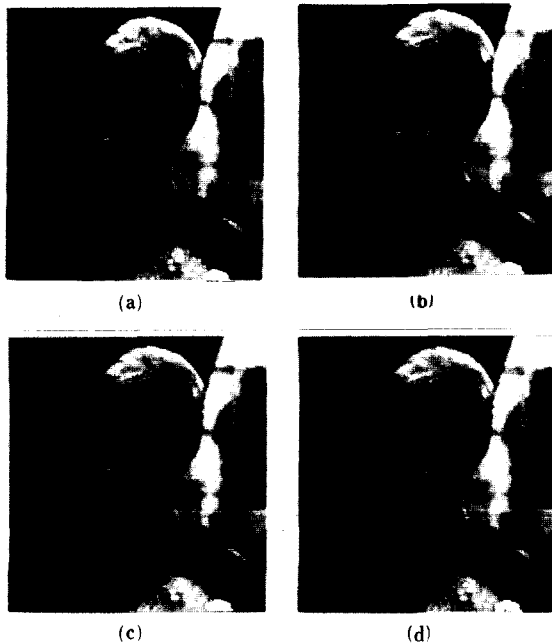


그림 5. (a) 원 화상 (15bit/pel)
 (b) AMBTC 재생화상 (2.13bit/pel)
 (c) ABTC 재생화상 (1.71bit/pel)
 (d) ABTC 재생화상 (1.33bit/pel)

Fig. 5. (a) original image (15bit/pel).
 (b) AMBTC coded image (2.13bit/pel).
 (c) ABTC coded image (1.71bit/pel).
 (d) ABTC coded image (1.33bit/pel).

송하기 때문에 실험결과와 같이 RMSE (relative mean square error)가 커진 것으로 보인다. ABTC 방법에 있어서 화질과 비트레이트는 휘도신호의 문턱치(Y_{th})에 따라 달라지게 된다. 즉 표 3과 같이 휘도신호 문턱치(Y_{th})를 작게 하면 비트레이트는 높아 지지만 화질은 좋고, Y_{th} 를 크게 하면 낮은 비트레이트를 얻을 수 있지만 재생화면에 blocky 현상이 생기면서 화질이 떨어지게 된다. 그리고 ABTC와 AMBTC의 성능을 정확히 분석하기 위해 그림 5의 화상 외에 다른 여러가지 화상에 대해 같은 실험을 해 본 결과, 화상에 따라 다르지만 일반적으로 ABTC 방법은 1.4(bit/pel) 이상의 비트레이트에서 2.13(bit/pel)의 AMBTC 보다 재생화질면에서 성능이 더 우수한 것으로 나타났다.

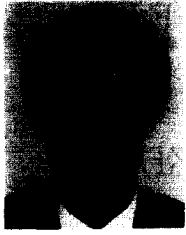
V. 결 론

본 논문에서는 기존의 AMBTC 방법을 개선한 adaptive BTC 방법을 새로이 제안하였다. ABTC 방법은 화상데이터의 부분적인 특성에 관계없이 일률적인 압축 알고리즘을 적용하는 기존의 방법을 탈피하여 인간의 시각특성과 화상데이터의 부분적인 특성에 따라 압축방식을 선택적으로 적용함으로써 기존방법보다 낮은 비트레이트로 더 우수한 화질을 얻을 수 있다. ABTC 방법은 1.0(bit/pel)에서 2.56(bit/pel)의 범위내에서 비트레이트를 임의로 조절할 수 있을뿐 아니라 압축 알고리즘이 기존의 AMBTC 방법보다 더 간단하므로 화상데이터 처리시간을 상당히 단축시킬 수 있는 장점이 있다. 실험결과 ABTC 방식은 1.4(bit/pel) 정도의 비트레이트로 원래의 화상에 비해 화질이 크게 떨어지지 않는 재생 화면을 얻을 수 있었다.

參 考 文 獻

- [1] A.K. Jain, "Image data compression: a review," *Proc. IEEE*, vol. 69, no. 3, pp. 349-389, Mar. 1981.
- [2] E.J. Delp and O.R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.* vol. COM-27, pp. 1335-1342, Sept. 1979.
- [3] M.D. Lema and O.R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Trans. Commun.* vol. COM-32, pp. 1148-1157, Oct. 1984.
- [4] D.J. Healy and O.R. Mitchell, "Digital video bandwidth compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1809-1817, Dec. 1981.
- [5] K.R. Castleman, *Digital Image Processing*, Prentice Hall, pp. 68-83, 1979.
- [6] J. Max "Quantization for Minimum Distortion," *IRE Trans. Inf. Theory*, IT-6, 1, pp. 7-12 Mar. 1960.
- [7] W.K. Pratt "Spatial Transform Coding of Color Images," *IEEE Trans. Comm.* vol. COM-19, pp. 980-992. Dec. 1971. *

著 者 紹 介

**張 基 崇 (正會員)**

1958年 10月 17日生. 1981年 2月 경북대학교 전자공학과 졸업. 1983年 2月 서울대학교 대학원 전자공학과 졸업 공학석사학위 취득. 1983年 2月~1985年 2月 현대중공업 종합연구소 연구원. 1985年 3月~현재 한국전기통신공사 사업지원단 전임연구원. 주관심분야는 위성통신, 디지털 영상신호처리 등임.

**李 榮 哲 (正會員)**

1941年 5月 11日生. 1969年 2月 한양대학교 전기공학과 졸업. 1988年 8月 한양대학교 산업대학원 전자공학과 졸업 공학석사학위 취득. 1970年 1月 체신부 전기통신연구소. 1979年 3月~1983年 12月 한국전자통신연구소 선임연구원. 1984年 1月~현재 한국전기통신공사 사업지원단 전송방식연구실장. 주관심분야는 디지털 영상신호처리, 디지털전송 등임.

**吳 性 穆 (正會員)**

1960年 8月 20日生. 1983年 2月 연세대학교 전자공학과 졸업. 1985年 2月 연세대학교 대학원 전자공학과 졸업 공학석사학위 취득. 1986年 5月~현재 한국전기통신공사 사업지원단 전임연구원. 주관심분야는 화상통신 등임.