

畫像데이터壓縮을 위한 프레임內/프레임間 벡터양자화된 블록切斷부호화에 관한 研究

(A Study on Intra/Interframe Vector Quantized Block Truncation Coding for Image Data Compression)

高 亨 和*, 李 忠 雄**

(Hyung Hwa Ko and Choong Woong Lee)

要 約

본 論文에서는 畫像데이터압축을 위한 벡터양자화된 블록切斷부호기(VQBTC)를 새로이 提案하였다. 既存의 block truncation coding (BTC)에 벡터量子化기법을 導入하여 BTC에 比해서 3~6배의 데이터壓縮을 이루었다. 벡터양자기는 프레임內에서와 프레임間에서 動作하도록 構成하였다. 컴퓨터 시뮬레이션을 통해서 프레임內에서는 0.7~1.0 bit/pel, 프레임間에서 시행할 경우 0.3~0.7 bit/pel의 데이터傳送率로 우수한 성능을 얻었다.

Abstract

This paper propose a novel vector-quantized block truncation coder for image data compression. A data compression ratio of about 3- 6 times larger than that of the BTC can be achieved by utilizing a vector quantizer with the BTC. A vector quantizer was realized by computer simulation. The compressed data rate of 0.7 ~ 1.0 bit/pel with intraframe coder and that of 0.3 ~ 0.5 bit/pel with interframe coder gives a good performance.

I. 序 論

畫像信號를 디지털로 傳送하거나 또는 저장하고자 할 때 대폭 늘어나는 情報量을 壓縮시키는 것이 重要的 課題이다. 예를 들어 PCM으로 화상을 實時間전송할 경우 약 90Mbps의 전송속도가 必要하게 된다.¹⁾ 따라서 매우 높은 채널容量을 갖는 傳送路가 요구된다. 또, 정지화상 데이터를 저장하고자 할때 필요한 메모리량을 생각해 보면, 1024×1024의 解像도를 갖는 畫像이 256레벨로 量子化되면 1Mbyte의 메모리가 必要

하다. 따라서, 경제적인 面에서 볼 때 畫像데이터를 壓縮하는 것은 必要하다. 다행스럽게도 화상은 인접화소간 또는 인접프레임間에 매우 높은 相關性이 있으므로 畫像信號는 많은 重複性(redundancy)을 갖고 있다. 1970년대에는 이러한 重複성을 除去하기 위한 研究가 매우 활발하게 進行되었다.^{1),2),3)}

중복성제거의 代表的인 方法으로 예측부호화와 변환부호화가 있다. 前者는 비교적 시스템이 간단하고 實時間처리가 可能하지만 압축율이 그다지 크지 않은 것이 短點이다. 後者の 경우는 畫像이 非相關性(uncorrelatedness)이 되도록 적절한 變換을 해준다음 zonal coding이나 threshold coding을 利用해 情報량을 압축시킨다. 그 예로써, Karhunen-Loeve, Hadamard, Harr, Cosine, Slant 變換 등이 있다. 壓縮率은 높으

*正會員, 光云大學 電子通信工學科

(Dept. of Elec. Comm., Eng., Kwang Woon Univ.)

**正會員, 서울大學校 電子工學科

(Dept. of Elec., Eng., Seoul National Univ.)

接受日字: 1986年 3月 8日

나 대부분 實時間處理가 어려운 것이 短點이다.

1979년경에 Delp와 Mitchell,⁴⁾ Kishimoto⁵⁾ 등에 의해 提案된 block truncation coding(BTC)도 새로운 화상압축技術이다. 이 方法은 성능이 adaptive cosine transform에 가까우면서도³⁾ 데이터전송율(data rate)이 1.6bits/pel밖에 되지 않는다. 특히 실시간처리가 可能하고 하드웨어실현면에서 有利하다. 또한 16~32 레벨의 中間調畫像을 전송할 경우 BTC가 有利하다.⁶⁾ 黑白화상신호뿐만 아니라 컬러信號의 경우에도 좋은 結果를 보여준다.⁷⁾ BTC는 모멘트를 保存하는 量子器를 利用하여 畫像을 블록별로 나누어 각 블록별로 平均과 分散을 保存하는 符號化方式이다. 한 블록내에서 각 畫素의 레벨이 平均치를 넘으면 (1)로, 平均치보다 작으면 (0)으로 부호화하여 (1, 0)의 비트평면(bit plane)을 만든다. 平均과 分散 및 비트평면을 전송하면 受信측에서는 이 세가지 情報로부터 원래의 화상을 再現한다.

最近에 새로운 데이터압축技術으로 각광을 받기 시작한 것 중에 벡터양자화(vector quantizer)가 있다.^{8,9)} 벡터양자화는 종래의 스칼라양자화가 하나의 샘플에 대해 量子化하는 것에 비해, 여러개의 샘플을 하나의 벡터로 取扱하여 量子化한다. $\{x_1, x_2, \dots, x_k\}$ 의 입력벡터 X 가 들어왔을 때 송신측은 코드북(codebook) 내의 대표벡터 $\{Y_1, Y_2, \dots, Y_n\}$ 중에서 $d\{X, Y_i\}$ 를 최소화하는 i 를 찾아서 전송한다. 여기서 $d\{X, Y_i\}$ 는 X 와 Y_i 간의 차이를 말한다. 수신측에서는 송신측과 똑같은 코드북을 갖고 있어서 傳送되어온 i 로부터 Y_i 를 再生벡터로 再生해 준다. 이때 전송율은 (1)식과 같다.

$$r = (\log_2 N) / k \tag{1}$$

스칼라量子化와는 달리 정수가 아닌 전송율을 許容한다.

本 論文에서는 block truncation coding(BTC)한 후에 비트평면을 벡터양자화함으로써 더욱 많은 데이터 압축을 實現시키고자 한다. 프레임내에서와 프레임간에서 양자화를 하였으며 사용된 코드북은 프레임내에서는 128個의 코드를 가지며 프레임간에서는 2048個의 코드를 갖고 있다.

본 논문의 구성은 서론에 이어 제Ⅱ장에서 프레임내에서의 벡터양자기에 대해 기술하였고, 제Ⅲ장에서 프레임간에서의 벡터양자기를 기술한 다음 제Ⅳ장에서는 컴퓨터 시뮬레이션 結果를 보였다.

Ⅱ. Intraframe 벡터양자기의 構成

디지탈化된 화상신호를 $(N \times N)$ 크기의 블록으로 나누어 각 블록을 BTC하면 (1, 0)의 비트平面이 形成

된다. 이때 $(N \times N)$ 크기의 비트평면의 發生可能한 패턴의 총數는 $2^{N \times N}$ 이다. 그런데, 비트평면은 인접화소간의 強한 相關性으로 인하여 實際로 發生할 수 있는 패턴은 이보다 훨씬 적다. 即, 비트평면의 중복성을 나타내고 있다. 이러한 重複性을 제거하기 위해 여러개의 畫像에 대해 많이 發生하는 패턴을 調査한 다음, 조사한 패턴중에서 임의의 개수의 패턴만을 코드북에 記錄한다. 입력되는 비트평면을 코드북내의 패턴들과 비교하여 가장 차이가 작은 패턴으로 代置하고 그 패턴의 인덱스를 전송함으로써 데이터압축을 이룰 수 있다.

입력데이터 X 를 $(X_{ij} | i, j = 1, 2, \dots, N)$ 의 벡터로 생각한다. 코드북 Y 를

$$Y = \{(Y_{ij} | n = 1, 2, \dots, M)\}$$

이라고 하면 X 와 Y_n 사이의 제곱誤差 $d(X, Y_n)$ 은

$$d(X, Y_n) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (X_{ij} - (Y_{ij})_n)^2 \tag{2}$$

이 된다. 여기서 X 와 Y_n 은 모두 (1, 0) 값만을 가지므로 (2)식은 다음의 (3)식으로 표현할 수 있다.

$$d(X, Y_n) = \frac{1}{N^2} (\text{number of positions where } X \text{ and } Y_n \text{ are different})^2 \tag{3}$$

입력 X 에 대해 코드북내의 모든 Y_n 에 의해 $d(X, Y_n)$ 을 求한 다음 최소가 되는 $d(X, Y_k)$ 를 찾아서 K 를 代表值로 전송한다.

여기서 코드북을 만드는 方法이 가장 큰 課題이다. 여러개의 畫像에 대해 비트평면을 調査하여 그 가운데 발생빈도가 높은 順序대로 취하면 되겠으나 畫像의 複雜性, 대역폭, 상관도 등에 따라 크게 달라진다. 따라서 歪曲을 적게 하기 위해서는 使用하는 畫像의 종류에 따라 코드북이 달라져야 하는 점이 발생한다. 실제로 實驗에 사용된 코드북은 128個의 패턴으로 構成했다. 이 가운데 일부를 그림 1에 보였다. 32개의 패턴만을 보여주고 있다. 이 코드북은 실제화상을 조사하여 발생순서대로 취한 것 중에서 약간의 變更을 가하여 만든 것이다.

데이터전송율을 살펴보면 BTC의 경우 블록의 크기를 (4×4) 로 했을 때 2.0bits/pel이 된다. 그런데 본 章에서 제안한 방법으로는 데이터전송율이 $(7+16)/16 = 1.4\text{bits/pel}$ 로 BTC에 비해 30% 정도의 減縮을 이룰 수 있다. 이때 7비트는 128개의 패턴을 부호화하는데 필요하고 16비트는 平均과 분산을 부호화하는데 必要하다. 블록의 크기를 (8×8) 로 확장하면 데이터전송율은 $(4 \times 7 + 16)/64 = 0.7\text{bit/pel}$ 이 된다. 역시 BTC에 비해 60% 이상의 감축을 이룰 수 있다.

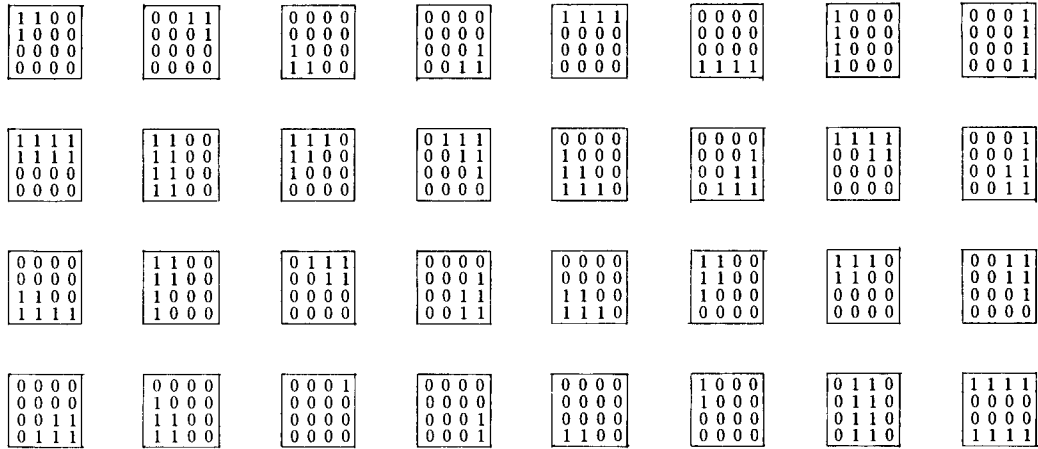


그림 1. 코드북내용의 일부
Fig 1. Partial contents of codebook.

성능평가를 위해 SNR면과 실제화상을 코딩한 후 화질을 비교하였다. SNR을 구하는 식은 (4)식과 같다.

$$SNR = \frac{\sum_{i=1}^N \sum_{j=1}^N x_{ij}^2}{\sum_{i=1}^N \sum_{j=1}^N (x_{ij} - y_{ij})^2} \quad (4)$$

이때 x_{ij} 는 $(N \times N)$ 크기를 갖는 화상의 각 화소를 표시한다. y_{ij} 는 코딩한 후 재현한 화상의 화소를 표시한다.

III. Interframe 벡터양자기의 構成

Intraframe 벡터양자기가 약 1bit/pel 정도에서 우수한 성능을 보여주는데 대해 아직 프레임間에서의 상관성은 利用하지 않고 있다. 여기에 착안하여 프레임間에서 비트平面的 벡터양자화를 하면 더욱 많은 데이터 압축을 이룰 수 있다. 프레임내에서 행할 때와 比較해 데이터전송율을 약 절반정도 낮출 수 있다.

연속하는 3개의 프레임의 畫像을 BTC한 다음 비트 평면에 대해 벡터양자화를 適用한다. 블록의 크기를 $(4 \times 4 \times 3)$ 으로 할 경우 총 發生 가능한 패턴은 2^{**} 이 된다. 이 가운데 2048개의 패턴을 검색에 의해 구한 다음 코드북에 기억시킨다. 즉, 48비트의 비트평면이 11비트로 압축된다. 이것이 가능한 理由는 3次元的인 상관성 때문이다. 코드북을 만드는 方法은 프레임內에서 使用한 128個의 패턴을 이용해 3프레임에 걸쳐 벡터양자화를 적용한 다음 $(128 \times 128 \times 128)$ 개의 패턴 가운데 2048개를 順次的으로 취한다. 이같은 작업을 여러 다양한 화상에 대해 광범위하게 실시하여 최종적으로 2048개의 패턴을 갖는 코드북을 確定시킨다. 입력화상

을 $(4 \times 4 \times 3)$ 의 블록크기로 BTC를 한 다음 비트평면을 코드북의 패턴과 매칭(matching)시킨다. 그다음 최소의 差異를 발생하는 패턴을 그 블록의 대표치로 전송한다. 이때 데이터전송율은 $(16+11)/48=0.56\text{bit/pel}$ 이 된다. 만일 블록의 크기를 $(8 \times 8 \times 3)$ 으로 확장시키면 더욱 큰 압축율을 얻을 수 있다. 即, 데이터전송율은 $(16+4 \times 11)/192=0.31\text{bit/pel}$ 이 된다. 이 경우 블록크기가 커지면 화질의 저하를 막기 위해 더욱 많은 패턴을 갖는 코드북이 필요하다.

IV. 실험 및 結果

실험은 VAX-11/780을 사용하였고, 사용된 디지털화상은 'Walter Kronkite'이다. 연속하는 3개의 프레임을 가지고 시뮬레이션하였다. 실험은 4가지의 다른 方法으로 시행하였다. 첫째로, (4×4) 블록 크기로 나누 뒤 BTC와 벡터量子化를 시행하였다.

둘째로, (8×8) 블록 크기로 하여 BTC와 프레임內 벡터양자화를 하였다. 두 方法公히 (4×4) 의 크기를 갖고 있는 128개의 패턴을 이용하여 벡터양자화를 행한다. SNR 및 데이터전송율 등을 計算하고 양자화한 후 화상을 再現하였다.

셋째로, $(4 \times 4 \times 3)$ 의 블록크기로 나누 뒤 3프레임에 걸쳐 프레임間 BTC 및 벡터양자화를 시행하였다.

넷째로, $(8 \times 8 \times 3)$ 의 블록크기로 나누 뒤 3프레임에 걸쳐 프레임間 BTC와 양자화를 하였다. 이때 사용된 코드북은 2048개의 $(4 \times 4 \times 3)$ 크기의 패턴으로 구성되어 있다.

시뮬레이션 結果는 표 1과 표 2와 같다. 표 1은 프레

표 1. 프레임내에서의 실험결과

Table 1. The results of intraframe simulation.

Method	Block Size	S/N [dB]	데이터전송율 (bit/pel)	패턴의수
BTC	4×4	25.6	2.0	-
BTC	8×8	21.5	1.25	-
V. Q	4×4	23.5	1.06	128
V. Q	8×8	21.2	0.69	128

표 2. 프레임간에서의 실험결과

Table 2. The results of interframe simulation.

Method	Block Size	S/N [dB]	데이터전송율 (bit/pel)	패턴의수
BTC	4×4×3	24.5	1.33	-
BTC	8×8×3	21.3	1.08	-
V. Q	4×4×3	21.7	0.51	2048
V. Q	8×8×3	18.8	0.31	2048

임내에서 시행한 결과이고, 표 2는 프레임간에서 시행한 결과를 보여준다. 프레임내에서 (4×4) BTC가 2.0 bits/pel의 데이터전송율로 25.6(dB)의 S/N를 유지했다. 프레임내에서 (8×8) VQBTC는 21.2(dB)의 S/N과 0.7bit/pel의 데이터전송율을 갖는 것으로 나타났다. 프레임간에서는 (4×4×3) BTC가 1.3bits/pel에서 24.5(dB)을 보여줌으로써 프레임내에서의 경우에 비해 S/N이 1(dB) 밖에 低下되지 않지만 데이터전송율은 60(%)로 압축된다. 즉, 프레임간에서 BTC를 하더라도 화질의 劣化는 거의 發生하지 않는다. 프레임간에서의 시뮬레이션 결과는 (4×4×3)의 벡터量子化했을 때 S/N이 21.7(dB), 데이터전송율은 0.51bit/pel 이 되었다. BTC에 비해 4(dB)의 S/N 저하가 있었지만 데이터전송율은 1/4로 줄어든다.

그림 2~그림 5는 재현된 화상을 나타낸다. 그림 2는 프레임내에서 實驗한 결과이다. 그림 2 (a)는 원래의 화상이고, 그림 2 (b)는 (4×4) BTC후 재현한 화상이다. 그림 2 (c)는 (4×4) 벡터양자화한 다음 재현한 화상이다. 또, 그림 2 (d)는 (8×8) 벡터양자화한 후 재현한 화상이다. 그림 3은 프레임간에서 實驗한 결과이다. 그림 3 (a)는 원래의 화상이고 그림 3 (b)는 (4×4×3) BTC한다음 再現한 화상이다. 그림 3 (c)는 (8×8×3) BTC한다음 재현한 것이고 그림 3 (d)는 (4×4×3) 벡터量子化를 시행한 다음 재현한 화상이다. 그림 4와 그림 5는 각각 프레임내에서와 프레임간에서 원래 화상과의 差 異 部 分만 을 나타낸다.

V. 結 論

본 研究에서는 畫像데이터압축技法의 하나인 block

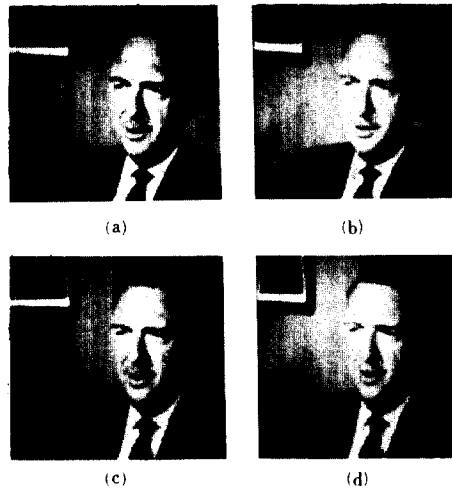


그림 2. 프레임내에서의 실험결과

- (a) 원래의 화상 (8bits/pel)
- (b) (4×4) BTC후 재현한화상 (2bits/pel)
- (c) (4×4) VQBTC후 재현한화상 (1.1bits/pel)
- (d) (8×8) VQBTC후 재현한화상 (0.7bits/pel)

Fig. 2. Simulation Results at intraframe.

- (a) original image (8 bits/pel).
- (b) (4×4) BTC image (2 bits/pel).
- (c) (4×4) VQBTC image (1.1 bits/pel).
- (d) (8×8) VQBTC image (0.7 bit/pel).

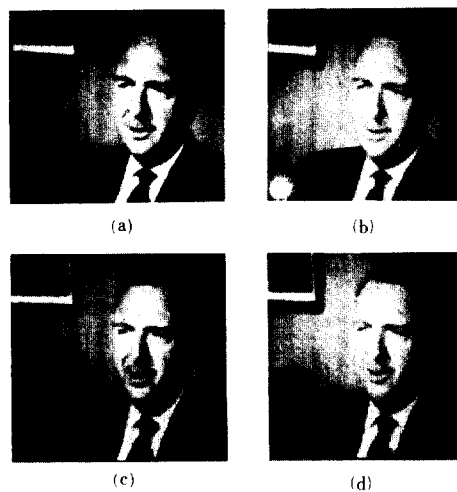


그림 3. 프레임간에서의 실험결과

- (a) 원래의 화상 (8bits/pel)
- (b) (4×4×3) BTC후 재현한 화상 (1.3 bits/pel)
- (c) (8×8×3) BTC후 재현한 화상 (1.1 bits/pel)
- (d) (4×4×3) VQBTC후 재현한 화상 (0.5 bit/pel)

Fig. 3. Simulation Results at interframe.

- (a) original image (8 bits/pel).
- (b) (4×4×3) BTC image (1.3 bits/pel).
- (c) (8×8×3) BTC image (1.1 bits/pel).
- (d) (4×4×3) VQBTC image (0.5 bit/pel).

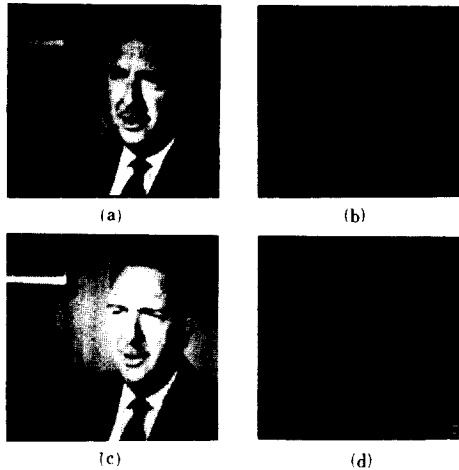


그림 4. Intraframe에서의 error 화상

- (a) (4×4) BTC 후 재현한 화상(2.0bits/pel)
- (b) (4×4) BTC 후 재현한 error 화상
- (c) (4×4) VQBTC 후 재현한 화상(1.1bits/pel)
- (d) (4×4) VQBTC 후 재현한 error 화상

Fig. 4. Error image at intraframe.

- (a) (4×4) BTC image (2 bits/pel).
- (b) (4×4) BTC error image.
- (c) (4×4) VQBTC image (1.1 bits/pel).
- (d) (4×4) VQBTC error image.

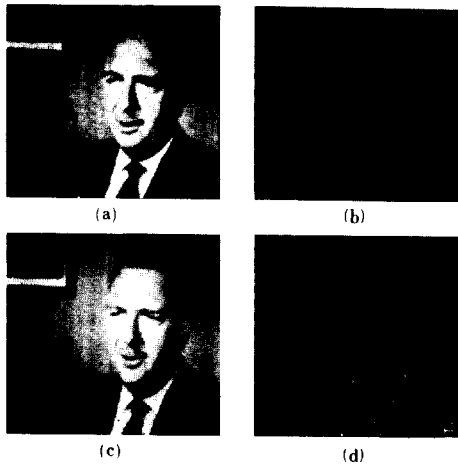


그림 5. Interframe에서의 error 화상

- (a) $(4 \times 4 \times 3)$ BTC 후 재현한 화상(1.3bits/pel)
- (b) $(4 \times 4 \times 3)$ BTC 후 재현한 error 화상
- (c) $(4 \times 4 \times 3)$ VQBTC 후 재현한 화상(0.51bit/pel)
- (d) $(4 \times 4 \times 3)$ VQBTC 후 재현한 error 화상

Fig. 5. Error image at interframe.

- (a) $(4 \times 4 \times 3)$ BTC image (1.3 bits/pel).
- (b) $(4 \times 4 \times 3)$ BTC error image.
- (c) $(4 \times 4 \times 3)$ VQBTC image (0.5 bit/pel).
- (d) $(4 \times 4 \times 3)$ VQBTC error image.

truncaion coding (BTC)를 기본으로 하여 비트평면의 相關性을 利用하는 벡터量子化를 導入하였다. 벡터양자기를 프레임內와 프레임間에서 각기 구성하고 그性能을 컴퓨터 시뮬레이션을 통해 실험하였다. BTC의 데이터전송율이 2.0bits/pel인데 비해 프레임內 vector quantized block truncation coder (VQBTC)는 데이터전송율이 0.7bit/pel, 프레임間 VQBTC는 0.3bit/pel의 값을 갖는다. 화질면에서는 원래의 화상과 비교해서 큰劣化는 없지만 블록별로 처리했기 때문에 blocky 현상이 나타났다. 앞으로의 연구과제는 고속매칭 방법의 개발과 codebook설계에 최적화 algorithm을 적용하면 더 우수한 성능을 얻을 수 있겠다. 아울러 blocky 현상을 줄이기 위한 노력과 윤곽을 살리는 방법들의 연구도 필요하다.

參 考 文 獻

- [1] P.A. Wintz, "Transform picture coding," *Proc. IEEE*, vol. 60, pp. 809-820, 1972.
- [2] A.N. Netravali and J.O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366-406, 1980.
- [3] A. Rosenfeld, *Digital Picture Processing*, Academic Press, pp. 116-208, 1982.
- [4] E.J. Help and O.R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 1335-1342, Sep. 1979.
- [5] T. Kishimoto, et al., "A method of still picture coding by using statistical properties," *National Conf. Inst. Electron. Commun. Eng.*, Japan No. 974, Mar. 1978.
- [6] E.J. Delp and O.R. Mitchell, "Multilevel graphics representation using block truncation coding," *Proc. IEEE*, vol. 68, pp. 868-873, July 1980.
- [7] 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.
- [8] R.M. Gray, "Vector quantization," *IEEE ASSP Magazine* pp 4-29, Apr. 1984.
- [9] Yamada and Tazaki, "Vector quantizer design for video signals," *IECE Trans. J66-B*, pp 965-972, 1983. *