Fuzzy block truncation coding

Abdel-Ouahab Boudraa
IRENav, Ecole Navale
29200 Brest-Naval
France
E-mail: boudra@ecole-navale.fr

Azeddine Beghdadi
L2TI, Université Paris 13
93430 Villetaneuse
France

Sidi Mohammed Réda Dehak
ENST Département Image
rue Barrault, 75634 Paris
France

Răzvan Iordache
Tampere University of Technology
Institute of Signal Processing
Finland

Abstract. Block truncation coding (BTC) is a well known lossy compression scheme. Due to its low complexity and easy implementation, BTC has gained wide interest in its further development and application for image compression. Based on simple thresholding, BTC retains sharp edges and thus leads to artifacts such as the staircase effect. The second problem encountered in BTC is the splitting of homogeneous regions, which produces false contours. In this work a fuzzy approach of BTC to avoid truncating homogeneous blocks and to preserve smooth edges in two-cluster blocks is proposed. Each image block, viewed as a fuzzy set, is segmented into two clusters using a fuzzy clustering algorithm. The block is then encoded by modified fuzzy weighted means of the two clusters. Initialization strategies of the fuzzy clustering algorithm and a contextual quantization method are proposed. Experimental results show an improvement of visual quality of reconstructed images and peak signal-to-noise ratio when compared to BTC, economical BTC (EBTC), absolute moment BTC (AMBTC), and a minimum mean square error quantizer (MMSEQ). Computation time required by AMBTC, EBTC, and fuzzy BTC methods are reported. [DOI: 10.1117/1.1518031]

Subject terms: image processing; image coding; block truncation coding; fuzzy clustering; fuzzy C-means algorithm; contextual quantization.

Paper 010485 received Dec. 31, 2001; revised manuscript received Apr. 15, 2002; accepted for publication Apr. 17, 2002.

1 Introduction

Block truncation coding (BTC) is a well known lossy image compression scheme.[1] Due to its low complexity and easy implementation, BTC has gained wide interest in its further development and application in image data compression. This block-based method is a two-level quantizer that adapts to local properties of the image. The reconstruction levels of the quantizer are chosen to preserve the first and second moments of the block. The blocks are coded individually and each one is described by the values of mean, standard deviation, and *an* N×N bit plane consisting of ones and zeros, indicating whether pixels are above or below a decision level (e.g., mean value of the block). Blocks are typically 4×4 in size. BTC has found applications in coding color images[2,3] video,[4] and visual pattern coding.[5] BTC is also attractive for hardware implementation.[6] Lema and Mitchell[7] modified the one moment-preserving constraint of BTC and derived a new method, called absolute moment BTC (AMBTC), which outperforms BTC in computational complexity and mean square error (MSE).[7] In addition to the first moment, AMBTC preserves the first absolute central moment rather than the simple variance, which is required in BTC. Instead of using the first absolute moment as a constraint in AMBTC, the minimum mean square error quantizer (MMSEQ) searches for a threshold for each block to meet the criterion of minimizing MSE.[1] It has been shown that performances of AMBTC and MMSEQ coding methods are almost identical.[8] Other techniques are also proposed to find the optimal threshold for BTC using MSE as a fidelity criterion.[9–11] Due to its relatively high bit rate, different variants of BTC aimed at reducing the bit rate using median filtering,[12] discrete cosine transform,[13] or interpolation[14] are proposed. These improvements, however, are achieved at the cost of decreasing image quality. To improve the quality of the reconstructed image, a BTC with more than two quantization levels has been developed.[15] Vector quantization (VQ), which provides a high compression ratio and multilevel quantizer ability, has been combined with BTC to reduce the bit rate and improve edge degradation.[16,17] The main difficulty in applying BTC is finding an adequate decision boundary to separate the data block into two classes. Obviously, the mean of the total pixel values is not an optimal threshold in the sense of minimizing MSE. Furthermore, thresholds generated by BTC[1] and its variants[7,9–11] are difficult ones. Indeed, these thresholds may retain sharp edges, and this leads to artifacts such as the staircase effect, while edge regions are not always sharp but smooth ones. A second problem encountered is the splitting of the homogeneous blocks, which produces false contours. The aim is a two-level coding scheme that avoids truncating homogeneous blocks and preserves smooth edges in two-cluster blocks. For block truncation, a clustering method is more robust against noise than a simple gray-level thresholding-based method. Boundaries of real data structures (clusters) are not well defined (fuzzy) and the transition from one cluster to another is not abrupt but smooth. Thus, formulation of the block truncation should be done fuzzy instead of crisp using a fuzzy set theory. It is convenient and appropriate to regard image regions as fuzzy subsets of the image,[18–19] Fuzzy subsets are characterized by the possibility (degree) of each pixel belonging

PROOF COPY 012212JOE

to them. Note also, since a gray-level picture possesses some ambiguity with pixels due to the possible multivalued levels of brightness in the image, it is justified to apply the concept of fuzzy logic to image compression problems.[20,21] One popular method for assigning multisubset membership values to pixels, for either segmentation or other processing, is the fuzzy C-means (FCM) algorithm. We propose a BTC method based on a FCM algorithm called fuzzy BTC (FBTC).

2 Fuzzy Clustering

A basic problem that arises in a great variety of fields including pattern recognition, machine learning, and statistics, is the so-called *clustering problem*. It plays a key role in searching for structures in data. Each of these structures is called a *cluster*, a region in which the elements (objects) are as similar as possible to each other and at the same time as different from those of the other sets as possible. In this work, data clustering is viewed as a data partitioning problem. For partitioning datasets into groups of similar objects, it has been argued that fuzzy approaches often work better than crisp ones.[22–24] This is the case with many iterative algorithms that converge to a local minimum of the objective function, without any assurance of its proximity to the global minimum. In this situation a fuzzy clustering method evolves more smoothly to the global minimum, whereas a crisp method bears more risk to get stuck in a local minimum.[25] Consider a set of n vectors $\boldsymbol{X}=\{x_1,\ldots,x_n\}\subset\Re^p$ to be clustered into $c\in\{1,\ldots,n\}$ subsets that represent the structure of $\boldsymbol{X}$. Each $x_k\in\Re^p$ is a feature vector consisting of p real-valued measurement (color, length,...) describing the features of the object represented by x_k. Clustering in unlabeled data $\boldsymbol{X}$ is the assignments of labels to objects generating $\boldsymbol{X}$. c-partitions of $\boldsymbol{X}$ are sets of $(c\times n)$ values $\{\mu_{ik}\}$ that can be conveniently arranged as a $(c\times n)$ matrix $\boldsymbol{U}=[\mu_{ik}]$. Each element, μ_{ik}, represents the membership degree of x_k to belong to the i'th cluster. The set of all $c\times n$ nondegenerate constrained fuzzy partitions, A_{cn}, is defined as follows:

$$A_{cn}=\left\{U\in\Re^{c\times n}\left|\sum_{i=1}^{c}\mu_{ik}=1,\sum_{k=1}^{n}\mu_{ik}>1\quad\text{and}\right.\right.$$
$$\left.\mu_{ik}\in[0,1];\ i=1,\ldots,c;\ 1\leqslant k\leqslant n\right\}\subset[0,1]^{cn}. \tag{1}$$

A well known model of fuzzy clustering in $\boldsymbol{X}$ is the following objective function[22]:

$$J_m(\boldsymbol{U},\boldsymbol{V}:\boldsymbol{X})=\sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ik})^m||x_k-v_i||_A^2. \tag{2}$$

$\boldsymbol{U}\in A_{cn}$ is a fuzzy partition matrix, $m\in[1,+\infty[$ is a weighting exponent called the fuzzifier, $\boldsymbol{V}=(v_1,v_2,\ldots,v_c)$ is the vector of unknown cluster centers (prototypes), $v_i\in\Re^p$ for $1\leqslant i\leqslant c$ and $||x||_A=(x^TAx)^{1/2}$ is any inner product norm where A is any positive definite matrix. Good partition $\boldsymbol{U}^*$ of $\boldsymbol{X}$ is taken from $(\boldsymbol{U}^*,\boldsymbol{V}^*)$ that are local minimizers of J_m. The usual method to optimize J_m is to use partial optimization of $\boldsymbol{U}$ and $\boldsymbol{V}$. One can first fix $\boldsymbol{U}$ and find necessary conditions on $\boldsymbol{V}$ to minimize J_m. Then we fix $\boldsymbol{V}$ and minimize J_m with respect to $\boldsymbol{U}$.[22] Approximate optimization of J_m by FCM is based on iteration through the following conditions for its extrema:

$$\mu_{ik}=\left[\sum_{j=1}^{c}\left(\frac{||x_k-v_i||_A}{||x_k-v_j||_A}\right)^{2/(m-1)}\right]^{-1}, \tag{3}$$

$$v_i=\frac{\sum_{k=1}^{n}(\mu_{ik})^m.x_k}{\sum_{k=1}^{n}(\mu_{ik})^m}. \tag{4}$$

The FCM algorithm consists of iterations alternating between Eqs. (3) and (4). This algorithm converges to either a local minimum or a saddle point of J_m.[22]

2.1 *Fuzzy-2-Mean (F2M) Algorithm*

Since the BTC scheme is based on a two-level quantizer, fuzzy clustering is restricted to two clusters. In this case, it is easy to see that membership degrees μ_{ik} can be reduced to:

$$\mu_{ik}=\left[\sum_{j=1}^{2}\left(\frac{||x_k-v_i||_A}{||x_k-v_j||_A}\right)^{2/(m-1)}\right]^{-1} \tag{5}$$

$$\mu_{1k}=\left[1+\left(\frac{||x_k-v_1||_A}{||x_k-v_2||_A}\right)^{2/(m-1)}\right]^{-1} \tag{6}$$

$$\mu_{2k}=1-\mu_{1k}. \tag{7}$$

2.2 *Contextual Quantization Output Levels*

Prototypes v_i [Eq. (4)] can be used as reconstruction levels.[21] In this work, output levels of the quantizer, q_1 and q_2, are estimated as the weighted fuzzy means of the clusters C_1 and C_2 respectively. Indeed, for a better estimation of a level of cluster C_i, only pixels affected to this cluster contribute to the output level. Levels are defined as follows:

$$q_i=\frac{\sum_{k\in C_i}(\tilde{\mu}_{ik})^{\beta(i,k)}.x_k}{\sum_{k\in C_i}(\tilde{\mu}_{ik})^{\beta(i,k)}}, \tag{8}$$

$$\beta(i,k)=\frac{\max_k(\tilde{\mu}_{ik})-\tilde{\mu}_{ik}}{\max_k(\tilde{\mu}_{ik})-\min_k(\tilde{\mu}_{ik})}, \tag{9}$$

$$\tilde{\mu}_{ik}=\max_{1\leqslant l\leqslant 2}(\mu_{lk})\Leftrightarrow x_k\in C_i, \tag{10}$$

where $i\in\{1,2\}$. Relation (10) is the rule of the maximum membership degree (*defuzzification*). $\beta(i,k)$ is a weighting component, which acts like the degree of fuzziness m, and its value depends on the cluster homogeneity. Compared to m, which is fixed for all data points,[22] $\beta(i,k)$ is block and cluster dependent. This component is introduced to give more weight for typical pixels of the cluster C_i to others to compute the output level. It is easy to see that $\beta(i,k)$

$\in[0,1]$. For $\beta(i,k)=0$ (uniform cluster), the weighted fuzzy mean q_i coincides with the arithmetic mean of the cluster. The closer $\beta(i,k)$ gets to zero, the more typical pixel x_k is. The decoder reconstructs the image block using the calculated levels q_1 and q_2. To use less than 8 bits each for q_1 and q_2, only the difference between levels $|q_1 - q_2|$ and one prototype are transmitted.

3 Initialization of F2M Algorithm

As an iterative technique, the FCM algorithm is sensitive to initial starting membership degrees. This problem is not exclusive to the FCM algorithm but shared with many clustering algorithms that work as a hill-climbing strategy. Furthermore, due to the limited number of pixels to be clustered per block (i.e., 16) and to the fact that block pixels are spatially correlated, an adequate block initialization is required to avoid the F2M to be easily trapped inside a local minimum. In this work, four spatial initializations corresponding to four principal edge orientations {0,45,90, and 135 deg} and an initialization based on the mean gray level of the block, like in the BTC scheme, are proposed. The aim of these initializations is to detect possible heterogeneous or homogeneous blocks before fuzzy clustering, thus making FBTC a more robust method. However, one may be restricted to only one initialization. The block is first initialized (truncated) spatially with $\theta=0$ deg [Eq. (11)] and the corresponding mean values of the generated classes are compared. If these two values are different, the block clustering is started. If the spatial initialization for $\theta=0$ deg fails another $\theta\in\{0,45,90, \text{and } 135 \text{ deg}\}$ is tested and so on. If all spatial initializations fail, the block is truncated using its mean gray level value [Eq. (12)]. Let $x(i,j)$ denote the pixel value at location (i,j) in the image, and i and j the row and the column respectively. In this study, a point (i,j) is represented by a scalar k, whose value depends on the scanning order of the block. Spatial initializations are defined as follows:

1. Edge block (θ):

$$\mu_{1k}=\begin{cases}\alpha_k & \text{if} \quad k\in\aleph_\theta\\ 1-\alpha_k & \text{if} \quad k\in\aleph-\aleph_\theta\end{cases} \tag{11}$$

where $\theta\in\{0,45,90, \text{ and } 135 \text{ deg}\}$.

The pixel sets $\aleph$, $\aleph_{0°}$, $\aleph_{45}$, $\aleph_{90}$, and $\aleph_{135\text{ deg}}$ are defined as follows:

$$\aleph=\{k=i+(j-1)\cdot N|1\leqslant i\leqslant N;1\leqslant j\leqslant N\}$$

$$\aleph_{0\text{ deg}}=\{k=i+(j-1)\cdot N|1\leqslant i\leqslant N/2;1\leqslant j\leqslant N\}$$

$$\aleph_{45\text{ deg}}=\{k=i+(j-1)\cdot N|1\leqslant i\leqslant N;1\leqslant j\leqslant N+1-i\}$$

$$\aleph_{90\text{ deg}}=\{k=i+(j-1)\cdot N|1\leqslant i\leqslant N;1\leqslant j\leqslant N/2\}$$

$$\aleph_{135\text{ deg}}=\{k=i+(j-1)\cdot N|1\leqslant i\leqslant N;i\leqslant j\leqslant N\}.$$

2. Mean of the block:

$$\mu_{1k}=\begin{cases}\alpha_k & \text{if} \quad x_k\geqslant\overline{\boldsymbol{X}}\\ 1-\alpha_k & \text{if} \quad x_k<\overline{\boldsymbol{X}},\end{cases} \tag{12}$$

$$\alpha_k=\left[1+\left(\frac{x_k-\overline{\boldsymbol{X}}_1}{x_k-\overline{\boldsymbol{X}}_2}\right)^{2/(m-1)}\right]^{-1} \tag{13}$$

$$\overline{\boldsymbol{X}}=\frac{1}{|\mathcal{N}|}\sum_{k\in\mathcal{N}}x_k \tag{14}$$

$$\overline{X_i}=\frac{1}{|N_i|}\sum_{x_k\in N_i}x_k \quad i\in\{1,2\}, \tag{15}$$

where $\overline{\boldsymbol{X}}$ is the mean of the block. $|\mathcal{N}|$, $|N_1|$ and $|N_2|$ are the cardinals of the block, clusters 1 and 2, respectively. For an edge block, initialization clusters 1 and 2 are defined by the sets $N_1=\aleph_\theta$ and $N_2=\aleph-\aleph_\theta$, respectively.

4 Proposed Image Coding Algorithm

The proposed FBTC works as follows. The image is partitioned into nonoverlapping $N\times N$ blocks (i.e., $N\times N=n=16$). F2M algorithm segments each block pixels into two clusters (i.e., cluster 1, cluster 2). It is achieved as follows:

FBTC algorithm
{
Choose a threshold value ϵ and set iteration counter $t=1$.
Initialize $\boldsymbol{U}^{(t)}$ using equations (7),(11), or (12).
while$(\|\boldsymbol{U}^{(t+1)}-U^{(t)}\|\leqslant\epsilon)$
{
Calculate prototypes using Eq. (4).
Calculate $\boldsymbol{U}^{(t+1)}$ using Eqs. (6) and (7).
Increment t.
}
Defuzzification process
{
if$(\mu_{1k}>\mu_{2k})$
{
$\tilde{\mu}_{1k}\leftarrow\mu_{1k}$;
$\tilde{\mu}_{2k}\leftarrow 0$;
$x_k\in C_1$
}
else
{
$\tilde{\mu}_{2k}\leftarrow\mu_{2k}$;
$\tilde{\mu}_{1k}\leftarrow 0$;
$x_k\in C_2$
}
}
Estimate the two-level block q_1 and q_2 using Eq. (8).
}

5 Results

Thirteen well known gray-scale images, each with 256×256 pixels and 8 bits/pixel resolution are tested on a PC

Table 1 PSNR values obtained from the Lena image from BTC, AMBTC, MMSEQ, EBTC, and FBTC algorithms using different data rates (bpp).

Rate(bpp)	BTC	AMBTC	MMSEQ	EBTC	FBTC
5.0	37.535	37.857	37.886	37.804	38.537
2.0	31.311	31.649	31.885	31.676	32.353
1.25	28.175	28.546	28.656	28.890	29.395
1.0625	25.899	26.312	26.347	26.912	27.227
1.0156	24.092	24.485	24.504	24.861	25.178
1.0039	22.227	22.616	22.620	22.681	22.999

Pentium III (550 MHz). For comparison BTC,[1] EBTC,[11] AMBTC,[7] and MMSEQ[1] algorithms are implemented. Fuzzy clustering is performed with the threshold value ϵ set to 10^{-3}. The fuzzifier m is fixed at 1.5.[22] In this work, the Euclidean inner product norm is used ($\boldsymbol{A}$ is the unit matrix). As a commonly used measure in a variety of coding systems, peak signal-to-noise ratio (PSNR) is adopted for evaluating objective quality. PSNR is defined as

$$\mathrm{PSNR}=10\log_{10}\frac{255^2}{\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\{x(i,j)-\hat{x}(i,j)\}^2}, \tag{16}$$

where 255 is the peak gray level of the image, $x(i,j)$ and $\hat{x}(i,j)$ are the pixel gray levels from the original and the reconstructed images, respectively, and $M\times M$ is the total number of pixels in the image. Data rate R is determined by the block size $|\mathcal{N}|$ and the numbers of bits f_1 and f_2 to quantize q_1 and q_2 respectively. R is given by $(|\mathcal{N}|+f_1+f_2)/|\mathcal{N}|$ bits per pixel (bpp). For instance, for a block size $|\mathcal{N}|=4\times4$ and using $f_1=f_2=8$ bits, the image would be compressed to 2 bpp (see Table 1). Table 2 lists the PSNR values of the images decoded by the AMBTC, EBTC ,and the FBTC, respectively. The average improvements in PSNR are 0.804 and 0.884 dB compared to AMBTC and EBTC, respectively. The improvement of FBTC over AMBTC and EBTC are up to 1.326 and 1.553 dB, respectively, for the same bit rate (2 bpp). To compare AMBTC with FBTC, a region of interest with details of various size and spatial frequencies is zoomed in the original and the reconstructed images (Fig. 1). The obtained results clearly show that FBTC (Fig. 2) outperforms AMBTC (Fig. 3) on the basis of visual criteria observation. Indeed, fine structures and contours are better preserved with FBTC than with AMBTC. See, for example, the leap, the right pupil, and the right cheek where the contouring and staircase effects are clearly reduced in the FBTC-encoded image than in the AMBTC one. This result also illustrates that threshold generated by F2M is optimal compared to that based on the pixels mean block. Results shown in Figs. 2 and 3 are well expected. Indeed, in AMBTC or BTC, an homogeneous block with one pixel (outlier) whose value is far from those of the remaining pixels is split into two classes and this turns in a false contour. While in FBTC, this outlier pixel constitutes a cluster by itself to avoid a false contour.

Fig. 1 Magnified view of Lena image.

Table 2 PSNR performance for AMBTC, EBTC, and FBTC and the corresponding improvement in dB when the block size is 4×4.

Image	AMBTC(1)	EBTC(2)	FBTC(3)	Gain[(3)-(1)]	Gain[(3)-(2)]
Lena	31.649	31.676	32.353	0.704	0.677
couple	32.061	32.010	32.720	0.659	0.710
Zelda	33.089	32.860	33.626	0.537	0.766
goldhill	30.562	30.617	31.242	0.680	0.625
peppers	31.515	31.578	32.275	0.760	0.697
baboon	26.222	26.028	26.697	0.475	0.669
boat	31.588	31.418	32.149	0.561	0.731
Columbia	26.185	26.474	27.288	1.103	0.814
crowd	25.428	25.408	26.630	1.202	1.222
lake	25.531	25.149	26.516	0.985	1.367
plane	27.490	27.556	28.430	0.940	0.874
lax	25.748	26.286	27.074	1.326	0.788
bridge	28.158	27.130	28.683	0.525	1.553

PROOF COPY 012212JOE

Fig. 2 Reconstructed Lena image by FBTC.

Fig. 3 Reconstructed Lena image by AMBTC.

Table 1 shows the PSNR values obtained by reconstructing the 256×256 monochrome Lena image with BTC, AMBTC, MMSEQ, and FBTC coding methods at different R values. Clearly, FBTC exhibits a rate-distortion performance better than other coding systems (Table 1). The results show that AMBTC and MMSEQ have very close performance across the range of compression ratios, and this confirms the findings of Ma and Rajala.[8] It can also be seen from Table 1 that PSNR produced by the EBTC algorithm is slightly better than that of AMBTC and MMSEQ. We list in Table 3 the execution time needed in EBTC, FBTC, and AMBTC methods with block sizes 4×4 and 8×8. As shown in Table 3, the processing time of AMBTC is much faster than that of the FBTC and EBTC methods. On the average, AMBTC, EBTC, and FBTC require 0.05, 0.158, and 0.336 s, respectively. In other words, the computing cost of FBTC is 2.12 times the cost of EBTC and 6.72 the cost of AMBTC. However, the improvement of FBTC over EBTC is up to 3.074 dB for $R=1.25$ bpp (bridge image). We can say that there is a good compromise between the PSNR produced by FBTC and the processing time required. FBTC is computationally more demanding than AMBTC or EBTC due mainly to the five initializations in Eqs. (11) and (12) tested for each block, and to the contextual quantization evaluated by the formula in Eq. (8). If we restrict ourselves to only one initialization, the FBTC scheme becomes faster. Recall that these initializations are introduced to avoid the F2M algorithm being easily trapped inside a local minimum and consequently to make FBTC a more robust method. Different values of the stop criterion, ϵ ranging from 10^{-1} to 10^{-6}, have been tested with block sizes 4×4 and 8×8. The best results, in PSNR, have been obtained for $\epsilon=10^{-3}$. For $\epsilon>10^{-3}$, we have not observed significant improvements of PNSR values. Note that in general ϵ is empirically chosen.[21,22,26]

Although still widely used, the PSNR generally does not correlate well with subjective judgments. Thus, development of the objective quality measure for image compression based on the human visual system is urgently required. Compared to the reconstruction level methods used in Ref. 21, in this work each quantizer (q_1 or q_2) is not estimated using all block pixel values [Eq. (4)], but only the pixels of

Table 3 PSNR values and execution time obtained in different images from AMBTC, EBTC, and FBTC algorithms with block sizes 4×4 and 8×8.

	EBTC		FBTC		AMBTC	
Image	4×4	8×8	4×4	8×8	4×4	8×8
Lena	0.15 s	0.14 s	0.34 s	0.33 s	0.05 s	0.05 s
	31.676 dB	28.890 dB	32.353 dB	29.395 dB	31.649 dB	28.546 dB
Zelda	0.14 s	0.13 s	0.34 s	0.32 s	0.05 s	0.05 s
	32.859 dB	29.820 dB	33.623 dB	30.324 dB	33.089 dB	29.812 dB
goldhill	0.17 s	0.15 s	0.33 s	0.35 s	0.05 s	0.05 s
	30.617 dB	28.060 dB	31.242 dB	28.531 dB	30.562 dB	27.943 dB
peppers	0.15 s	0.15 s	0.34 s	0.36 s	0.05 s	0.05 s
	31.578 dB	28.683 dB	32.275 dB	29.183 dB	31.515 dB	28.537 dB
bridge	0.20 s	0.20 s	0.31 s	0.34 s	0.06 s	0.06 s
	27.130 dB	22.918 dB	28.683 dB	25.992 db	28.158 dB	25.460 dB

the corresponding cluster (C_1 or C_2). This contextual quantization reduces the error reconstruction pixel value, particularly in a heterogeneous block. However, this reconstruction level leads to erroneous output levels when the subblock (q_1 or q_2) contains outliers. This is due to the fact that the FCM algorithm does not distinguish outliers from nonoutliers. To avoid this problem, one may use possibilistic c-means[26] as a clustering algorithm, but it is heavily dependent on initialization and requires the estimation of the bandwidth of each cluster. The FCM clustering method is a gradient-based algorithm that converges to a local minimum. A practical solution to the initialization problem is to perform the clustering several times, initializing randomly each time. Thus, in this work five initialization strategies are proposed. The aim of these initializations is to detect possible heterogeneous or homogeneous blocks before fuzzy clustering. Only four basic orientations, which occur in the coding of nature images, among a large number of edge blocks are considered. No information about the position of the edge block is incorporated.

6 Conclusion

A new technique called FBTC is proposed for encoding still images. The encoding method divides each image block into two clusters using a fuzzy clustering algorithm. Although there is no mathematical proof of the superiority of FBTC over BTC and AMBTC, experimental results performed on 13 well known gray-scale images have shown that FBTC improves the perceptual quality of these encoded images and increases the corresponding PSNR values. There is a tradeoff between the PSNR and the computation time. Like AMBTC, one advantage of FBTC is its simplicity, and it is not time consuming. The proposed coding method can be combined with the VQ method for better image quality and compression, as was done in Refs. 16 and 17.

References

1. E. J. Delp and O. R. Mitchell, ''Image compression using block truncation coding,'' *IEEE Trans. Commun.* **27**, 1335–1342 (1979).
2. L. A. Overturf, M. L. Comer, and E. L. Delp, ''Color image coding using morphological pyramid decomposition,'' *IEEE Trans. Image Process.* **4**(2), 177–185 (1995).
3. C. K. Yang, J. C. Lin, and W. H. Tsai, ''Color image compression by moment-preserving and block truncation technique,'' *IEEE Trans. Commun.* **45**(12), 1523–1516 (1997).
4. D. J. Healy and O. R. Mitchell, ''Digital video bandwidth compression using BTC,'' *IEEE Trans. Commun.* **29**(12), 1809–1817 (1981).
5. B. Barnett and A. C. Bovik, ''Motion compensated visual pattern sequence coding for full motion multisession videoconferencing on multimedia workstations,'' *J. Electron. Imaging* **5**(2), 129–143 (1996).
6. L. G. Chen, Y. C. Liu, T. D. Chiueh, and Y. P. Lee, ''A real time video signal processing chip,'' *IEEE Trans. Consum. Electron.* **39**(2), 82–92 (1993).
7. M. D. Lema and R. O. Mitchell, ''Absolute moment block truncation coding and its application to colour image,'' *IEEE Trans. Commun.* **32**, 1148–1157 (1984).
8. K. K. Ma and S. A. Rajala, ''A comparison of absolute moment block truncation coding and the minimum mean square error quantizer,'' *Proc. IEEE. Int. Symp. Circuits Systems, Singapore*, pp. 296–299 (1991).
9. M. Kamel, C. T. Sun, and L. Guan, ''Image compression by variable block truncation coding with optimal threshold,'' *IEEE Trans. Signal Process.* **39**(1), 208–212 (1991).
10. L. G. Chen and Y. C. Liu, ''A high quality MC-OBTC codec for video signal processing,'' *IEEE Trans. Circuits Syst. Video Technol.* **4**(1), 92–98 (1994).
11. C. Y. Yang and J. C. Chen, ''EBTC: An economical method for searching the threshold of BTC compression,'' *Electron. Lett.* **32**(20), 1870–1871 (1996).
12. G. Arce and N. C. Gallagher, ''BTC image coding using median filter roots,'' *IEEE Trans. Commun.* **31**(6), 784–793 (1983).
13. Y. Wu and D. C. Coll, ''BTC-VQ-DCT hybrid coding of digital images,'' *IEEE Trans. Commun.* **39**(9), 1283–1287 (1991).
14. B. Zeng and Y. Neuvo, ''Interpolative BTC image coding with vector quantization,'' *IEEE Trans. Commun.* **41**(10), 1436–1438 (1993).
15. C. H. Kuo and C. F. Fuu, ''Nearly optimum multilevel block truncation coding based on a mean absolute error criterion,'' *IEEE Signal Process. Lett.* **3**(9), 269–271 (1996).
16. S. A. Mohamed and M. M. Fahmy, ''Image compression using VQ-BTC,'' *IEEE Trans. Commun.* **43**(7), 2177–2182 (1995).
17. C. C. Wang, C. H. Chen, and I. H. Chen, ''Modified VQ-BTC algorithm for image compression,'' *Electron. Lett.* **34**(14), 1390–1392 (1998).
18. J. M. S. Prewitt, ''Object enhancement and extraction,'' in *Picture Processing and Psychopictories*, B. S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York (1970).
19. A. Rosenfeld, ''The fuzzy geometry of image subsets,'' *Pattern Recogn. Lett.* **2**, 311–317 (1984).
20. A. Boudraa, Q. Kanafani, A. Beghdadi, and A. Zergainoh, ''Vector quantization for image compression based on fuzzy clustering,'' *Fifth Int. Symp. Signal Process. Appl.*, **2**, 835–838 (1999).
21. N. B. Karayiannis and P. I. Pai, ''Fuzzy vector quantization and their application in image processing,'' *IEEE Trans. Image Process.* **4**(9), 1193–1201 (1995).
22. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York (1981).
23. M. Roubens, ''Pattern recognition problem with fuzzy sets,'' *Fuzzy Sets Syst.* **1**, 239–253 (1978).
24. E. Trauwaert, L. Kaufman, and P. Rousseeuw, ''Fuzzy clustering algorithms based on the maximum likelihood principle,'' *Fuzzy Sets Syst.* **42**, 213–227 (1991).
25. P. J. Rousseeuw, E. Trauwaert, and L. Kaufman, ''Fuzzy clustering with high contrast,'' *J. Comput. Appl. Math.* **64**, 81–90 (1995).
26. R. Krishnapuram and J. M. Keller, ''The possibilistic approach to clustering,'' *IEEE Trans. Fuzzy Syst.* **1**(2), 98–110 (1993).

Abdel-Ouahab Boudraa received his BS degree in physics (electronics engineering) from the Constantine Institute of Physics, University of Constantine, Algeria. He received his MS in biomedical engineering from INSA, Lyon, the University degree in nuclear magnetic resonance, his PhD degree in image processing, and his University degrees in statistics and modeling and in positron emission tomography all from the University of Claude Bernard, Lyon 1, France. He is currently an associate professor of electrical engineering at Ecole Navale, Brest, France. His current research interests include computer vision, vector quantization, data structures and analysis, data fusion, time frequency analysis, hard and fuzzy pattern recognition, and applications of fuzzy set theories to medical images. He is an Associate Member of IEEE.

Azeddine Beghdadi is currently Full Professor at the University of Paris 13 (Institut Galilée) and a researcher at L2TI, where he does all his research in image processing (image enhancement, segmentation, image compression, image quality, medical imaging, and physics and bio-inspired models for image analysis). He received an education in physics. He obtained his ''Maitrise'' in physics, and Diplôme d'Etudes Approfondies in optics and signal processing from University Orsay-Paris XI in June 1982 and June 1983, respectively. He also obtained his PhD in physics (optics and signal processing) from University Paris 6 in June 1986. He has worked at different places including the Groupe d'Analyze d'Images Biomédicales (CNAM Paris) and Laboratoire d'Optique des Solides (University of Paris 6). He received his Habilitation à Diriger des Recherches from University Paris 13 in May 1994, which was an achievement of ten years of research in the area of image processing fundamentals (image enhancement, nonlinear filtering, segmentation, and image analysis) and their applications to physical and mechanical problems (multifratal and entropic analysis of disordered heterogeneous media - measurement of mechanical deformation from scanning electron microscope images). He was a visiting research professor at the University of Geneva in Switzerland and at Queensland University of Technology in Australia. He has published more than 25 international journal papers and more than 60 conference papers. He is a member of IEEE.

Sidi Mohammed Réda Dehak received his MS degree in signal and image processing in 1998 from the National Institute of Applied Sciences (INSA), Lyon, France and the Engineering degree in computer science in 1997 from University of Sciences and Technology, Oran, Algerie. He is currently working toward his PhD degree in signal and image processing at ENST of Paris, France. His research interests include fuzzy sets theory, spatial relationships, and approximate reasoning.

Răzvan Iordache received his Engineering degree in electronics in 1995 and his MS degree in biomedical engineering in 1996, both from Politehnica University of Bucharest, Romania, and his PhD degree in signal and image processing in 2001 from Tampere University of Technology, Finland. Currently, he is a research engineer at General Electrical Medical Systems, France. His research interests include breast imaging, medical imaging, image coding, and image quality assessment.

PROOF COPY 012212JOE