Fuzzy block truncation coding

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

Azzeddine 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. © 2002 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.1518031]

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

1 Introduction

Block truncation coding (BTC) is a well known lossy image compression scheme. 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, video, and visual pattern coding. BTC is also attractive for hardware implementation. Lema and Mitchell 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). 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. It has been shown that performances of AMBTC and MMSEQ coding methods are almost identical. Other techniques are also proposed to find the optimal threshold for BTC using MSE as a fidelity criterion. Due to its relatively high bit rate, different variants of BTC aimed at reducing the bit rate using median filtering, discrete cosine transform, or interpolation 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. 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. 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 and its variants 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. Fuzzy subsets are characterized by the possibility (degree) of each pixel belonging...
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.\textsuperscript{20,21} One popular method for assigning multiset 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.\textsuperscript{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.\textsuperscript{25} Consider a set of \( n \) vectors \( X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p \) to be clustered into \( c \in \{1, \ldots, n\} \) subsets that represent the structure of \( X \). Each \( x_k \in \mathbb{R}^p \) 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 \( X \) is the assignment of labels to objects generating \( X \). \( c \)-partitions of \( X \) are sets of \( (c \times n) \) values \{\( \mu_{ik} \)\} that can be conveniently arranged as a \( (c \times n) \) matrix \( U = [\mu_{ik}] \). Each element, \( \mu_{ik} \), 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:

\[
A_{cn} = \left\{ U \in \mathcal{R}^{c \times n} \mid \sum_{k=1}^{n} \mu_{ik} = 1, \sum_{k=1}^{n} \mu_{ik} > 1 \text{ and } \mu_{ik} \in [0,1] ; i=1,\ldots,c ; 1 \leq k \leq n \right\} \subset [0,1]^{cn}.
\]

A well known model of fuzzy clustering in \( X \) is the following objective function:\textsuperscript{22}

\[
J_m(U,V,X) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^m ||x_k - u_i||_A^2.
\]

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

\[
\mu_{ik} = \frac{1}{\sum_{k=1}^{n} (\mu_{ik})^m} \left( \sum_{j=1}^{c} \frac{||x_k - u_j||_A}{||x_k - u_i||_A} \right)^{2/(m-1)}, \quad i = 1, \ldots, n \quad j = 1, \ldots, c
\]

\[
u_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^m \cdot x_k}{\sum_{k=1}^{n} (\mu_{ik})^m}.
\]

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 \( J_m \).\textsuperscript{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 \( \mu_{ik} \) can be reduced to:

\[
\begin{align*}
\mu_{ik} &= \frac{2}{\sum_{j=1}^{c} \left( \frac{||x_k - u_{ik}||_A}{||x_k - u_{ji}||_A} \right)^{2/(m-1)}} \quad i = 1, \ldots, n \quad j = 1, \ldots, c \\

\mu_{ik} &= 1 - \mu_{ik}.
\end{align*}
\]

2.2 Contextual Quantization Output Levels

Prototypes \( v_i \) [Eq. (4)] can be used as reconstruction levels.\textsuperscript{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:

\[
\begin{align*}
q_1 &= \sum_{k \in C_1} (\mu_{ik})^{b(i,k)} \cdot x_k \\
q_2 &= \sum_{k \in C_2} (\mu_{ik})^{b(i,k)} \\
\beta(i,k) &= \frac{\max_k (\mu_{ik}) - \mu_{ik}}{\max_k (\mu_{ik}) - \min_k (\mu_{ik})} \\
\tilde{\mu}_{ik} &= \max_{1 \leq i \leq 2} (\mu_{ik}) \phi x_k \in C_i,
\end{align*}
\]

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,\textsuperscript{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 \( \beta(i,k) \)
is represented by a scalar pixel value at location $i$ and $j$ respectively. In this study, a point $(i, j)$ within a block is defined by coordinates $(u_1, u_2)$, where $u_1$ is the scan order of the block. Spatial initializations are 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 $\theta \in \{0, 45, 90\}$ and $135$ 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 ($\theta$):

$$\mu_{1k} = \begin{cases} \alpha_k & \text{if } k \in \mathbb{N}_\theta \\ 1 - \alpha_k & \text{if } k \in \mathbb{N} - \mathbb{N}_\theta \end{cases}$$

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

The pixel sets $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{N}_{45}$, $\mathbb{N}_{90}$, and $\mathbb{N}_{135}$ are defined as follows:

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

$$\mathbb{N}_0 = \{k = i + (j - 1) \cdot N | 1 \leq i \leq N/2; 1 \leq j \leq N\}$$

$$\mathbb{N}_{45} = \{k = i + (j - 1) \cdot N | 1 \leq i \leq N; 1 \leq j \leq N + 1 - i\}$$

$$\mathbb{N}_{90} = \{k = i + (j - 1) \cdot N | 1 \leq i \leq N; 1 \leq j \leq N/2\}$$

$$\mathbb{N}_{135} = \{k = i + (j - 1) \cdot N | 1 \leq i \leq N; i \leq j \leq N\}.$$ 

2. Mean of the block:

$$\bar{x} = \frac{1}{|\mathbb{N}|} \sum_{k \in \mathbb{N}} x_k, \quad i \in \{1, 2\},$$

where $\bar{x}$ is the mean of the block. $|\mathbb{N}|$, $|\mathbb{N}_1|$ and $|\mathbb{N}_2|$ 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 $\mathbb{N}_1 = \mathbb{N}_\theta$ and $\mathbb{N}_2 = \mathbb{N} - \mathbb{N}_\theta$, 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

1. Choose a threshold value $\epsilon$ and set iteration counter $t = 1$.

2. Initialize $U'^{(0)}$ using equations (7), (11), or (12).

3. $\textbf{while} (|U'^{(t+1)} - U'^{(t)}| \leq \epsilon)$


5. $U'^{(t+1)}$ using Eqs. (6) and (7).

6. Increment $t$.

7. Defuzzification process

8. $\textbf{if} (\mu_{1k} > \mu_{2k})$

9. $\mu_{1k} \leftarrow \mu_{1k} - 0.1$; $\mu_{2k} \leftarrow 0.1$; $x_k \in C_1$

10. $\textbf{else}$

11. $\mu_{2k} \leftarrow \mu_{2k} - 0.1$; $\mu_{1k} \leftarrow 0.1$; $x_k \in C_2$

12. Estimate the two-level block $q_1$ and $q_2$ using Eq. (8).

5 Results

Thirteen well known gray-scale images, each with $256 \times 256$ pixels and 8 bits/pixel resolution are tested on a PC.
Pentium III (550 MHz). For comparison BTC, EBTC, AMBTC, and MMSEQ algorithms are adopted. Fuzzy clustering is performed with the threshold value \( e \) set to \( 10^{-3} \). The fuzzifier \( m \) is fixed at 1.5. In this work, the Euclidean inner product norm is used (\( A \) 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

\[
\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} (x(i,j) - \hat{x}(i,j))^2},
\]

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 \( \frac{|\mathcal{N}| + f_1 + f_2}{|\mathcal{N}|} \) bits per pixel (bpp). For instance, for a block size \( |\mathcal{N}| = 4 \times 4 \) 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.

### Table 1

<table>
<thead>
<tr>
<th>Rate(bpp)</th>
<th>BTC</th>
<th>AMBTC</th>
<th>MMSEQ</th>
<th>EBTC</th>
<th>FBTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>37.535</td>
<td>37.857</td>
<td>37.886</td>
<td>37.804</td>
<td>38.537</td>
</tr>
<tr>
<td>2.0</td>
<td>31.311</td>
<td>31.649</td>
<td>31.885</td>
<td>31.676</td>
<td>32.353</td>
</tr>
<tr>
<td>1.0039</td>
<td>22.227</td>
<td>22.616</td>
<td>22.620</td>
<td>22.681</td>
<td>22.999</td>
</tr>
</tbody>
</table>

### Table 2

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

### Fig. 1

Magnified view of Lena image.
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 \( R = 1.25 \text{ bpp} \) (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, \( \epsilon \) 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 PSNR values. Note that in general \( \epsilon \) 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>
<thead>
<tr>
<th>Image</th>
<th>EBTC</th>
<th>FBTC</th>
<th>AMBTC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4×4</td>
<td>8×8</td>
<td>4×4</td>
</tr>
<tr>
<td>Lena</td>
<td>0.15 s</td>
<td>0.14 s</td>
<td>0.34 s</td>
</tr>
<tr>
<td>Zelda</td>
<td>0.14 s</td>
<td>0.13 s</td>
<td>0.34 s</td>
</tr>
<tr>
<td></td>
<td>32.859 dB</td>
<td>29.820 dB</td>
<td>33.623 dB</td>
</tr>
<tr>
<td>goldhill</td>
<td>0.17 s</td>
<td>0.15 s</td>
<td>0.33 s</td>
</tr>
<tr>
<td>peppers</td>
<td>0.15 s</td>
<td>0.15 s</td>
<td>0.34 s</td>
</tr>
<tr>
<td>bridge</td>
<td>0.20 s</td>
<td>0.20 s</td>
<td>0.31 s</td>
</tr>
<tr>
<td></td>
<td>27.130 dB</td>
<td>22.918 dB</td>
<td>28.683 dB</td>
</tr>
</tbody>
</table>
the corresponding cluster \((C_1 \text{ or } C_2)\). 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 \((q_1 \text{ or } q_2)\) 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 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


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 INSAS, 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.

Azedine 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 “Maîtrise” 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’Analyse d’Images Biomédicales (GAIMP Paris) and Laboatoire professor of electrical engineering (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 (multifractal and entropic analysis of disordered heterogeneous media - measurement of mechanical deformation from scanning electron microscope images). He was a visiting re-
search 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 confer-
ence papers. He is a member of IEEE.

Sidi Mohammed Réda Dehak received his MS degree in signal and image pro-
cessing in 1998 from the National Institute of Applied Sciences (INSA), Lyon, France
and the Engineering degree in computer science in 1997 from University of Sci-
ces 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 in-
clude 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 Uni-
versity of Technology, Finland. Currently, he is a research engineer
at General Electrical Medical Systems, France. His research inter-
ests include breast imaging, medical imaging, image coding, and
image quality assessment.