Computers and Mathematics with Applications 64 (2012) 1140–1152

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

The single-pass perceptual embedded zero-tree coding implementation on DSP

Bing-Fei Wu [a], Hao-Yu Huang [a], Yen-Lin Chen [b,*]

[a] Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, 30050, Taiwan

[b] Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

ARTICLE INFO

Keywords:
Block-edge classification
Discrete cosine transform (DCT)
Embedded system
Fast algorithm
Image compression
Zero-tree coding

ABSTRACT

This paper proposes a block-edge-based Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC) method. SPPEZC combines two novel compression concepts, called Block-Edge Detection (BED) and the Low-Complexity and Low-Memory Entropy Coder (LLEC), for coding efficiency and quality. Because the edge information can provide beneficial cues for preserving the perceptual quality of compressed images, this paper presents an effective combinative coding scheme, called Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC), which integrates the improved LLEC and the block-edge information. This approach provides improved perceptual quality in compressed images. Based on the block-edge information, this paper proposes an adaptive architecture for adjusting the quantization table and subsequently coding the quantized coefficients with the LLEC. The proposed SPPEZC approach was implemented and evaluated on both PC-based and DSP-based embedded platforms. Experimental results and comparisons demonstrate that the proposed SPPEZC technique provides computational efficiency as well as satisfactory perceptual quality in compressed images.

1. Introduction

The recent increase in the transmission and storage of visual data has enabled the rapid and continuous progress of image compression technology [1,2]. Transform coding plays an important role in reducing redundancy and maintaining good coding quality in image compression. The common Discrete Cosine Transform (DCT), one of the most popular forms of transform coding, uses coding and static Huffman coding techniques for transformation and entropy coding in prevailing image- and video-coding standards [3,4]. These standards include the Joint Photographic Expert Group (JPEG) [5], JPEG2000 [6], Motion Pictures Expert Group (MPEG) MPEG-1/2 [7,8], MPEG-4 [9], H.261/3 [10], and H.264/AVC [11]. Some of the proposed and applied DCT-based coding approaches include arithmetic coding (AC), embedded zero-tree DCT coding (EZDCT) [12], embedded zero-tree coding in hierarchical DCT (EZHDCT) [13], set partition in hierarchical tree (SPIHT) [14], zero-tree entropy coding (ZTE) [15], embedded block coding with optimized truncation (EBCOT) [16], and warped discrete cosine transform (WDCT) [17]. Although these coders achieve high compression efficiency, they have substantially higher computational costs and memory requirements than other techniques. These disadvantages form a hardware implementation bottleneck in mass-market consumer electronic products.

Zhao et al. [18] proposed a highly efficient method called the Low-Complexity and Low-Memory Entropy Coder (LLEC). This method involves Golomb–Rice (G–R) codes and the Zero-Tree Coding (ZTC) technique. ZTC exploits the zero-tree structure of transformed coefficients for higher compression efficiency, whereas G–R codes perform at a lower complexity

* Corresponding author. Tel.: +886 2 2771 2171.
E-mail addresses: ylchen@csie.ntut.edu.tw, ylchen@cssp.cn.nctu.edu.tw (Y.-L. Chen).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.03.032

level and with lower memory requirements than JPEG Huffman Coding. The LLEC makes block-based transform coding, such as the DCT, suitable for embedded and hardware implementations. Chang et al. presented a block-edge-based method to adjust the quantization table adaptively [19]. Block-Edge Detection (BED) provides useful information on edge features for applications in digital image/video processing, pattern recognition, and computer vision. Determining an efficient and quick mode of identifying edge features is an essential step in the detection and segmentation of objects of interest.

Although integrating the LLEC technique with the DCT is an efficient approach to produce a low-computation and low-memory cost image codec, it does not provide a perceptual strategy to improve the visual quality of compressed images. The quantization tables in the LLEC are fixed for all the DCT blocks, which may vary with different images. Block-edge information [20] can provide improved visual quality during the image compression process, increasing the visual quality of compressed images. Therefore, this study proposes a block-edge-based Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC) method that integrates and improves these two image coding techniques efficiently. This paper also presents an adaptive quantization table adjustment scheme based on the statistical analysis of the DCT blocks in various types of images. Results show that some of the specific DCT coefficients are relatively large. These DCT coefficients, called "dominating coefficients", reflect the characteristics of the directional edges of the DCT blocks. Therefore, the proposed combinative coding scheme adaptively adjusts the quantization tables based on edge feature detection. The LLEC then codes the quantized coefficients using a predetermined quantization table to preserve the highest visual quality in the compressed images.

Numerous studies apply block-edge information to the intra-frame prediction of H.264/AVC video coding [21,22]. This paper applies block-edge information based on DCT coefficients to the LLEC, to increase image compression performance. The information from the block-edge information helps preserve the characteristics of the DCT blocks. The preserved characteristics apply a perceptual strategy to the image coding scheme. This study also introduces a statistical scheme that adaptively and perceptually optimizes the quantization table of each DCT coding block according to the regional edge information obtained by the BED. The block-edge information in each block is analyzed and adopted to fine tune the quantization table. An analysis of the directional edges of DCT blocks further illustrates that the proposed characteristics are applicable to various images. The LLEC codec quantizes the coefficients. The proposed SPPEZC was implemented and evaluated on both PC-based and DSP-based embedded platforms. Experimental results demonstrate that the proposed approach provides satisfactory image compression performance in both peak signal-to-noise ratio (PSNR) and visual quality. Compared to other coding techniques, the SPPEZC demonstrates adequate and effective image compression for various images. The SPPEZC is also a single-pass coding algorithm that requires little computational time. Experimental results on the PC and DSP platforms show that the SPPEZC provides promising performance in both computational efficiency and image coding results.

The features and contributions of this study are as follows:

(1) The proposed SPPEZC improves two novel methods, which are LLEC and BED. With the proposed effective combinative coding scheme, the SPPEZC is a high-efficiency and high-quality image coder.
(2) The SPPEZC adopts the edge information of the DCT blocks to perform adaptive and perceptual image coding.
(3) Additionally, because of the low complexity of the SPPEZC, the proposed architecture is suitable for the embedded application. This study presents the implementation of the SPPEZC on the DSP-based platform.

The rest of this paper is organized as follows: Section 2 provides an overview of the LLEC and the BED based on the DCT; Section 3 presents the method of combining the LLEC and the BED based on the DCT from the analytical results; Section 4 shows the experimental results; and finally, Section 5 offers a conclusion.

2. Overview of the LLEC and the BED

This section briefly introduces the Low-Complexity and Low-Memory Entropy Coder (LLEC) [18] and the Block-Edge Detection (BED) based on the DCT [19]. The LLEC achieves high compression efficiency and low memory cost by combining Zero-Tree Coding (ZTC) and Golomb–Rice (G–R) codes. Furthermore, the LLEC involves only the fundamental sequence (FS) in G–R codes (G–R_FS). To strengthen the edge characteristics of each DCT block, this paper includes a fast BED based on the DCT.

2.1. Low-Complexity and Low-Memory Entropy Coder

The two key elements of this proposed coding technique are the single-pass ZTC and G–R_FS codes. Single-pass ZTC can identify the region in which significant coefficients are distributed for higher compression efficiency. The G–R code [23,24] is a type of variable length code. For applications involving DCT-based transform coding, G–R_FS coding can supplant the look-up-table Huffman coding procedure because it involves simpler computations and lower memory usage.

2.2. Block-Edge Detection based on DCT

By observing the distribution of DCT coefficients, previous researchers discovered that detecting and classifying edge components in images can perceptually improve the performance of image coding in a wide range of applications. This study adopts a fast and systematic scheme [19] to classify the edge patterns of each block for DCT-based image compression.

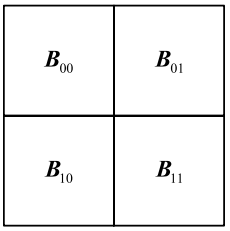


Fig. 1. The coefficients of the 8 × 8 pixel block divided into four subregions.

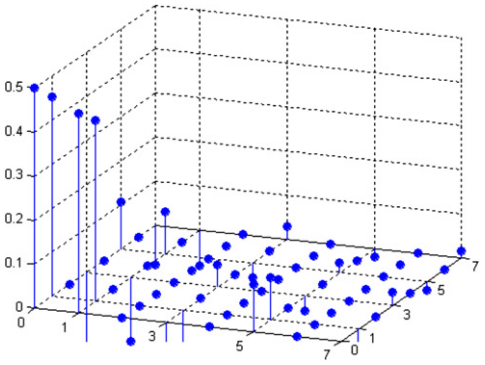


Fig. 2. Plot of $f_{00}(m, n)$.

Fig. 1 shows that each 8 × 8 pixel block includes four subregions for detecting the block edge (BE) orientation in the spatial domain. Eq. (1) indicates the intensity of each pixel in the block.

$$x(i, j), \quad i, j = 0, 1 \ldots, 7 \tag{1}$$

where i and j are the vertical and horizontal indices, respectively.

The following equation defines the average intensity value of each subregion:

$$B_{uv} = \frac{1}{16} \sum_{i=0}^{3} \sum_{j=0}^{3} x(4u + i, 4v + j), \quad u, v = 0, 1 \tag{2}$$

where the definitions of u and v are the same as those of i and j, respectively.

In the DCT domain, 2D IDCT is

$$x(i, j) = \sum_{m=0}^{7} \sum_{n=0}^{7} e(m)e(n)X(m, n)C_{16}^{(2k+1)m} C_{16}^{(2k+1)n}, \quad i, j = 0, 1, \ldots, 7 \tag{3}$$

where

$$C_{2N}^{\tau} = \cos\left(\frac{\pi\tau}{2N}\right), \qquad e(\tau) = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{if } \tau = 0 \\ 1 & \text{elsewhere.} \end{cases} \tag{4}$$

The following represents the form of B_{uv} after combining (2) with (3):

$$B_{uv} = \sum_{m=0}^{7} \sum_{n=0}^{7} f_{uv}(m, n)X(m, n), \quad u, v = 0, 1 \tag{5}$$

where f_{uv} is a filter for edge detection, as shown in (6).

$$f_{uv}(m, n) = e(m)e(n)C_4^m C_8^m C_{16}^m C_4^n C_8^n C_{16}^n (2C_2^m - 1)^u (2C_2^n - 1)^v. \tag{6}$$

Fig. 2 shows the magnitude plot of $f_{00}(m, n)$ in each frequency band. This illustrates that the magnitude decreases along the m and n axes. The experiment and analysis show that edge detection still works well when considering only $X(m, n)$s of $m, n = 0$ or 1. This is a major reason the magnitude of $f_{00}(m, n)$ remains in median. Because high-frequency bands are smaller than low-frequency bands, they have a non-significant influence on the results.

(a) No edge. (b) 0° edge. (c) $\pi/4$ edge. (d) $\pi/2$ edge. (e) $3\pi/4$ edge.

Fig. 3. The five directional edge patterns.

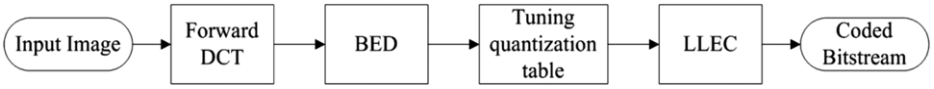


Fig. 4. The flowchart of the proposed encoder.

Therefore, this study rewrites (5) and (6) as

$$\tilde{f}_{uv}(m,n) = \begin{cases} e(m)e(n)C_4^m C_4^n \times (2C_2^m - 1)^u (2C_2^n - 1)^v, & \text{if } m, n = 0, 1 \\ 0, & \text{elsewhere} \end{cases} \tag{7}$$

$$\tilde{B}_{uv} = \sum_{m=0}^{7}\sum_{n=0}^{7} \tilde{f}_{uv}(m,n)X(m,n) \quad u, v = 0, 1. \tag{8}$$

Fig. 3 shows that, by using (7) and (8), the measurements of the five directional edge patterns become

$$\begin{cases} \delta_{\mathrm{NE}} \text{ User Define} \\ \delta_0 = 2\,|V| \\ \delta_{\pi/2} = 3/4 \max |H + V + D|\,, |H + V - D| \\ \delta_{\pi/4} = 2\,|H| \\ \delta_{3\pi/4} = 3/4 \max |H - V + D|\,, |H - V - D| \end{cases} \tag{9}$$

where

$$H = \frac{1}{2}X(0,1) \qquad V = \frac{1}{2}X(1,0) \qquad D = \frac{1}{2}X(1,1). \tag{10}$$

In (9), $X(m,n)$ represents the mth row and the nth column DCT coefficient of the 8×8 pixel DCT block X, where δ_{NE}, δ_0, $\delta_{\frac{\pi}{4}}$, $\delta_{\frac{\pi}{2}}$, and $\delta_{\frac{3\pi}{4}}$ represent the determination factor of the five edge patterns shown in Fig. 3. Eq. (11) shows the determination of the five edge patterns, and T_{QSS} denotes the threshold of a quantization step size. This study defines T_{QSS} as the direct current (DC) value of each DCT block to depict the perceptual information of the DCT block. The image block is the block with the maximum strength among the five directional edge features. The four subregions of a block are multiplied by a proper factor depending on BE. Otherwise, the step size remains unchanged. Hence, four quantization criteria exist for each DCT block, depending on the BE algorithm.

$$\mathrm{BE} = \begin{Bmatrix} \delta_\theta, & \max\left\{\delta_0, \delta_{\frac{\pi}{4}}, \delta_{\frac{\pi}{2}}, \delta_{\frac{3\pi}{4}}\right\} > T_{QSS} \\ \delta_{\mathrm{NE}}, & \max\left\{\delta_0, \delta_{\frac{\pi}{4}}, \delta_{\frac{\pi}{2}}, \delta_{\frac{3\pi}{4}}\right\} < T_{QSS} \end{Bmatrix}. \tag{11}$$

3. The proposed combinative coding scheme

Section 2 shows that the Low-Complexity and Low-Memory Entropy Coder (LLEC) performs efficiently on a DCT-based image coder. Block-Edge Detection (BED) based on the DCT is a high-speed edge detection method based on the DCT coefficient. Because BED can be applied efficiently to DCT coefficients, this paper presents an effective combinative coding scheme called Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC). The proposed coding technique uses the improved LLEC and the block-edge information to provide enhanced perceptual quality on compressed images.

Tables 1 and 2 and Fig. 4 show the algorithmic descriptions and flowchart of SPPEZC. The input image first encounters the macroblocks, and each of which is converted to the DCT coefficients. After completion of the forward DCT, three specific coefficients on each block can determine the directional edge of this block.

Using the obtained block-edge information, this study adaptively adjusts the quantization table of each block to accommodate the optimally fitted block. This paper considers that the Zero-Tree Coding (ZTC) scan lists the approximate order of the DCT frequency domain, where low and high numbers represent the low- and high-frequency elements,

Table 1
Algorithmic description of SPPEZC.

```
Block-based DCT
Directional edge detections of blocks
Modulate the corresponding quantization tables
by the results of edge detections
Use context DPCM scheme to code the DC
coefficient k
Encoding (k) {
  G–R_FS_Encoding (k);
  Zero-Tree_Encoding (k);
}
G–R_FS_Encoding (k){
    C (k); // Category
    P (k); // Position
    Bitstream_Output (k);
}
Bitstream_Output (k){
  Output 1 in C(k) + 1 bits;
  Output P(k) in C(k) bits;
}
Zero-Tree_Encoding (k) {
    if at least one descendant of k is not zero {
      Output 1 in 1 bit;
      for each k's son : s1n
        G–R_FS_Encoding (s1n);
      if at least one descendant of s1 is not zero {
        Output 1 in 1 bit;
      for each k's grandson: s2n
        G–R_FS_Encoding (s2n);
      }
    }
    else
      Output 0 in 1 bit;
}
```

Table 2
Algorithmic description of SPPEZC.

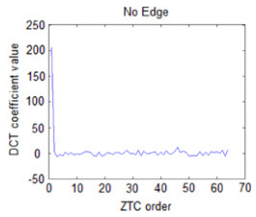
```
Initialize beginning index i of an 8 × 8 DCT
block coefficients as 0
Block-based DCT
Directional edge detections of blocks
Modulate the corresponding quantization tables
by the results of edge detections
Decoding () {
  Blockcoefficients initialize as 0;
  G–R_FS_Decoding ();
  Zero-Tree_Decoding (i);
}
G–R_FS_Decoding (){
    k = Bitstream_Input();
}
Bitstream_Input(){
  Get C bits;
  Get P value according C bits;
}
Zero-Tree_Decoding (i) {
  If get 1 value {
    for each k's son: s1n
      s1n = G–R_FS_Decoding ();
      If get 1 value {
        for each k's grandson: s2n
          s2n = G–R_FS_Decoding ();
        }
      }
    }
```

Table 3
The dominating coefficients of the directional edges.

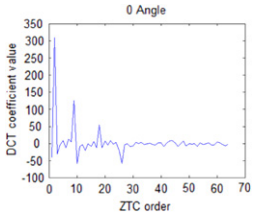
Edge direction	The dominating coefficients
No edge	DC value
0 angle	1st–4th, 6th–12th and 22nd–26th coefficients
90 angle	1st–4th, 6th–12th and 22nd–26th coefficients
45 angle	1st–4th, 14th–20th coefficients
135 angle	1st–4th, 14th–20th coefficients

respectively. Based on the analysis of ZTC characteristics, this paper shows the ZTC graph by observing the characteristics of each directional edge. The x-axis and y-axis of the graph represent the ordered numbers of the ZTC and their corresponding coefficients. Fig. 5 shows the typical graphs of the different directional edges. The different directional edges have the most critical coefficients at the specific order numbers, which are listed in Table 3, in all the 64-point DCT coefficients. These specific coefficients are "dominating coefficients", and represent the local maximum coefficients. Table 3 shows that, in certain rules, except for the case of "no edge", the first to fourth coefficients play important roles in all the coefficients of the DCT blocks for the other directional edges. Therefore, this paper sets the quantization tables of the first to fourth coefficients the same as that of the DC coefficient. This study also applies the more critical points, such as the 6th–12th coefficients in the case of "0 angle", for the quantization table of the DC coefficient to maintain the information of the dominant coefficients. The dominated coefficients represent the characteristics of the directional edges of the DCT blocks. Therefore, the corresponding quantization tables should be small to preserve the dominant coefficients for perceptual strategy.

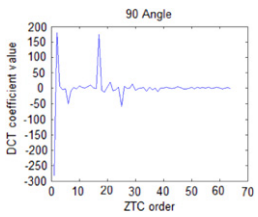
This study verifies and demonstrates the proposed concept by using a statistical analysis experiment. This experiment uses five standard test images: "Barbara", "Lena", "Elain", "Jet", and "Peppers". Table 4 shows the analytical results of the matching percentage of the directional edge blocks. In this statistical experiment, over 80% of the DCT blocks conform to the experimental hypothesis of the characteristics of the DCT coefficients and the directional edges, as Table 3 shows. This indicates that most of the DCT blocks can adopt the proposed modified quantization table to achieve good compression quality.



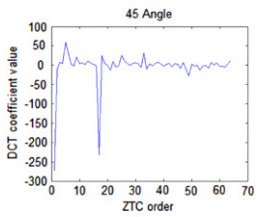
(a) No edge.



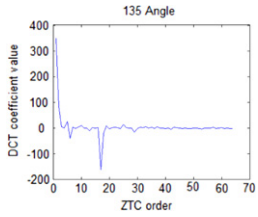
(b) 0 angle.



(c) 90 angle.



(d) 45 angle.



(e) 135 angle.

Fig. 5. Illustrations of the characteristics of the directional edges.

Table 4
Statistical results of the DCT blocks of image samples.

Image samples	Edge direction	The number of blocks in this edge	The number of blocks match the proposed characteristics	Matching ratio (%)
Barbara	No edge	1119	999	89.28
	0 angle	398	374	93.97
	90 angle	962	807	83.89
	45 angle	710	644	90.7
	135 angle	907	767	84.56
Lena	No edge	1500	1248	83.2
	0 angle	149	121	81.21
	90 angle	975	845	86.67
	45 angle	806	757	93.92
	135 angle	666	583	87.54
Elain	No edge	980	915	93.37
	0 angle	389	312	80.21
	90 angle	1090	908	83.30
	45 angle	561	558	99.46
	135 angle	1076	1066	99.07
Jet	No edge	1831	1828	99.84
	0 angle	588	573	97.45
	90 angle	711	641	90.15
	45 angle	309	280	90.61
	135 angle	657	619	94.22
Peppers	No edge	1334	1328	99.55
	0 angle	400	397	99.25
	90 angle	809	707	87.39
	45 angle	736	611	83.02
	135 angle	817	710	86.90

(a) Original. (b) SPPEZC. (c) LLEC. (d) JPEG. (e) JPEG-O.

Fig. 6. BE pattern $\delta = 3\pi/4$ performance demonstration of the "Barbara" image.

This paper multiplies the quantization tables of the other non-dominating coefficients by a weighting value for truncation. Eqs. (12)–(14) show the modulations of the quantization tables, where QT and n represent the quantization tables and their indices, respectively. Table 3 shows that the suffixes NE, 0, 90, 45, and 135 also indicate all the possible directional

Table 5
The compression performance at different bit rates.

Image name	C_R/bit-rate*	PSNR				
		Floating-point SPPEZC (dB)	Fixed-point SPPEZC (dB)	LLEC (dB)	JPEG (dB)	JPEG-O (dB)
Barbara	32/0.25	26.78	26.8	26.34	24.26	25.20
	16/0.50	30.53	29.99	30.08	27.81	28.30
	11/0.75	33.12	32.67	32.77	30.72	31.00
	8/1.00	34.86	34.35	34.59	33.04	33.10
Lena	32/0.25	32.27	32.36	31.80	31.40	31.60
	16/0.50	35.72	35.55	35.39	34.63	34.90
	11/0.75	37.71	37.65	37.40	36.52	36.60
	8/1.00	39.11	38.24	38.76	37.81	37.90
Elain	32/0.25	31.35	31.35	30.80	29.94	31.15
	16/0.50	32.68	32.57	32.40	32.55	32.66
	11/0.75	33.76	33.70	33.47	33.31	33.38
	8/1.00	34.88	34.68	34.54	33.89	34.00
Jet	32/0.25	30.91	30.83	29.66	29.80	30.47
	16/0.50	34.86	35.06	34.00	33.98	34.35
	11/0.75	37.35	36.58	36.77	36.43	36.55
	8/1.00	39.47	38.89	38.79	38.09	38.17
Peppers	32/0.25	31.75	31.73	30.83	30.18	31.20
	16/0.50	34.70	34.70	34.19	33.89	34.08
	11/0.75	36.12	35.89	35.79	35.32	35.37
	8/1.00	37.30	37.30	36.99	36.28	36.37

* C_R denotes the compression ratio.

Table 6
Comparisons of the average gain of PSNRs.

Name	Floating-point SPPEZC vs. LLEC (dB)	Fixed-point SPPEZC vs. LLEC (dB)	Floating-point SPPEZC vs. JPEG (dB)	Fixed-point SPPEZC vs. JPEG (dB)	Floating-point SPPEZC vs. JPEG-O (dB)	Fixed-point SPPEZC vs. JPEG-O (dB)
Barbara	0.38	0.01	2.37	2.01	1.92	1.55
Lena	0.37	0.11	1.11	0.86	0.95	0.70
Elain	0.37	0.45	0.75	0.65	0.37	0.28
Jet	0.84	0.69	1.07	0.77	0.76	1.82
Peppers	0.52	0.46	1.05	0.99	0.71	0.62
Avg.	0.50	0.34	1.27	1.06	0.94	0.99

(a) Original. (b) SPPEZC. (c) LLEC. (d) JPEG. (e) JPEG-O.

Fig. 7. BE pattern $\delta = 3\pi/4$ performance demonstration of the "Elain" image.

(a) Original. (b) SPPEZC. (c) LLEC. (d) JPEG. (e) JPEG-O.

Fig. 8. BE pattern $\delta = 3\pi/4$ performance demonstration of the "Peppers" image.

(a) Original. (b) SPPEZC. (c) LLEC. (d) JPEG. (e) JPEG-O.

Fig. 9. BE pattern $\delta = 3\pi/4$ performance demonstration of the "Jet" image.

Table 7
The comparisons of computational efficiency.

Name		Floating-point SPPEZC (ms)	Fixed-point SPPEZC (ms)	LLEC (ms)	JPEG (ms)	JPEG-O (ms)
Barbara	Encode	124	31	124	141	151
	Decode	110	16	109	125	128
Lena	Encode	110	16	110	125	130
	Decode	109	15	109	109	110
Elain	Encode	125	16	125	140	142
	Decode	93	8	93	109	110
Jet	Encode	109	31	109	125	129
	Decode	94	15	94	109	109
Peppers	Encode	110	16	109	140	146
	Decode	109	15	109	111	111
Avg.	Encode	115.6	22	115.4	134.2	139.6
	Decode	103	13.8	102.8	112.6	113.6

Table 8
The compression efficiency on DSP.

DSP-based platforms	Fixed-point SPPEZC	Coding efficiency (FPS)	Memory consumption (bytes)
TI OMAP3530	Encoder	10.63	25,920
	Decoder	11.56	

(a) Original. (b) SPPEZC. (c) LLEC.

(d) JPEG. (e) JPEG-O.

Fig. 10. Visual comparison results of the sample image "Barbara" at 0.25 bit rate.

edges. The order of the quantization tables is the result of zero-tree scanning. After this operation, the quantized coefficients, which ZTC orders, are coded with G–R_FS. The proposed combinative scheme provides more suitable quantization tables for the specific directional edge of the DCT block. It also helps the LLEC perform perceptual image coding on visual quality.

(a) Original.

(b) SPPEZC.

(c) LLEC.

(d) JPEG.

(e) JPEG-O.

Fig. 11. Visual comparison results of the sample image "Lena" at 0.25 bit rate.

Section 4 presents the comparative experimental results.

$$QT_{NE,n} = QT_n \times 1.15 \quad 64 > n > 0 \tag{12}$$

$$QT_{0,n} = \begin{cases} QT_n \times 1.15 & 64 > n > 31 \\ QT_n \times 0.9 & n = 1 \sim 4, 6 \sim 12, \\ & 22 \sim 26 \\ QT_n & \text{otherwise.} \end{cases} \qquad QT_{90,n} = \begin{cases} QT_n \times 1.15 & 64 > n > 31 \\ QT_n \times 0.9 & n = 1 \sim 4, 6 \sim 12, \\ & 22 \sim 26 \\ QT_n & \text{otherwise} \end{cases} \tag{13}$$

$$QT_{45,n} = \begin{cases} QT_n \times 1.15 & 64 > n > 31 \\ QT_n \times 0.9 & n = 1 \sim 4, 14 \sim 20 \\ QT_n & \text{otherwise} \end{cases} \qquad QT_{135,n} = \begin{cases} QT_n \times 1.15 & 64 > n > 31 \\ QT_n \times 0.9 & n = 1 \sim 4, 14 \sim 20 \\ QT_n & \text{otherwise.} \end{cases} \tag{14}$$

4. Experimental results

This section presents the coding quality and efficiency of the proposed method, called Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC). All experiments introduced in this section were simulated using a PC platform with an Intel Duo Core 1.67 GHz processor and 2 GB RAM. The 512 × 512 pixel grayscale sample images were tested at different bit rates (0.25 bit rate to approximately 1.00 bit rate). In the proposed architecture, the deadzone width of a midtread quantizer was set between approximately 30% and 50% larger than the regular stepsize. The image quality was measured using the peak signal-to-noise ratio (PSNR) method, computed from the original image and decoded images. Table 5 shows the experimental results of the LLEC [18], JPEG (using the default Huffman table) [1], and JPEG-O (using adaptive Huffman coding). This table illustrates the performance of the proposed approach implemented in floating-point and fixed-point versions. The fixed-point SPPEZC adopts the fixed-point DCT module for coding efficiency. The following descriptions in this section evaluate both the floating-point and fixed-point versions of the SPPEZC.

The proposed SPPEZC method displays performance that is substantially superior to that of the LLEC, JPEG, and JPEG-O, at a 0.25 bit rate. Table 6 shows that floating-point SPPEZC consistently outperforms the LLEC, JPEG, and JPEG-O for compression performance. Overall, it is superior to the LLEC by 0.50 dB, the JPEG by 1.27 dB, and the JPEG-O by 0.94 dB. The results of PSNR

(a) Original. (b) SPPEZC. (c) LLEC.

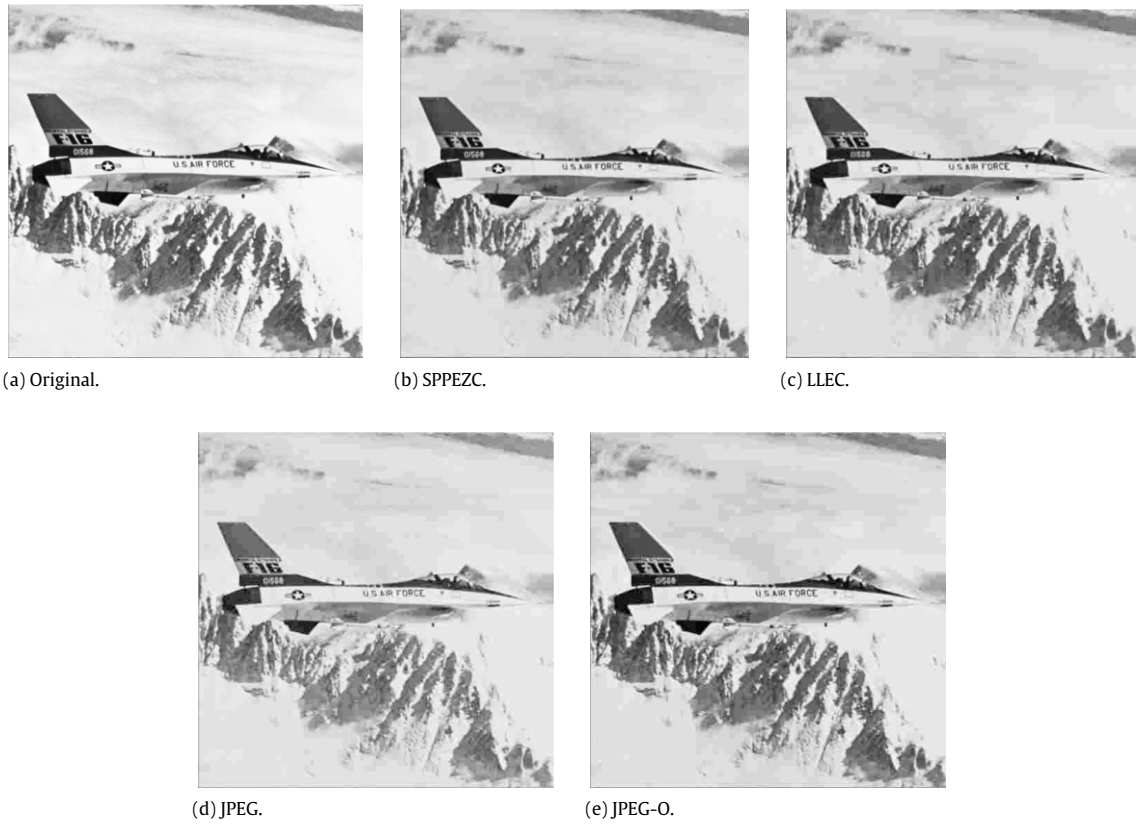
(d) JPEG. (e) JPEG-O.

Fig. 12. Visual comparison results of the sample image "Elain" at 0.25 bit rate.

comparisons confirm the excellent coding quality of floating-point SPPEZC. The following descriptions of Figs. 6–9 compare the visual quality of the compression results obtained by the floating-point SPPEZC, LLEC, JPEG, and JPEG-O. Table 6 also shows the comparative data between fixed-point SPPEZC and other coders. Although fixed-point SPPEZC may not provide the high PSNR measures of the floating-point version, it achieves higher PSNRs than the LLEC by 0.34 dB, the JPEG by 1.06 dB, and the JPEG-O by 0.99 dB, on average. However, for the "Barbara" test image, fixed-point SPPEZC achieves a slightly lower PSNR than that of the LLEC when the compression ratio is set to a 1.0, 0.75, or 0.5 bit rate. Nevertheless, fixed-point SPPEZC achieve better PSNR values than those of the LLEC when the compression ratio is set to a 0.25 bit rate. This is because the fixed-point DCT computation truncates a detailed fractional part of the DCT coefficients. The "Barbara" test image contains many DCT blocks that distribute sufficient DCT coefficients at high frequency. Therefore, the truncation of the fractional part of the DCT coefficients may have a greater effect when handling these types of pictures. However, as Table 6 shows, most of the compression results of fixed-point SPPEZC are still superior to those of the LLEC, JPEG, and JPEG-O.

Table 7 also shows a comparison of the computational costs of the four coders, which are SPPEZC, including the floating-point and fixed-point versions, LLEC, JPEG, and JPEG-O. Table 7 shows that both floating-point and fixed-point SPPEZC provide superior computational efficiency to those of the JPEG and JPEG-O, and especially fixed-point SPPEZC. Floating-point SPPEZC can save approximately 20 ms (ms) on encoding and 10 ms on decoding, compared to the standard JPEG codec. For computation timing, floating-point SPPEZC takes slightly longer for encoding than the LLEC method. This is because floating-point SPPEZC can rapidly and effectively determine the directional edge information of each DCT block using a computational frugal Block Edge Detection (BED) module. Therefore, although the floating-point SPPEZC approach takes slightly more computing time, the proposed approach can provide substantially improved perceptual quality and PSNR performance at the same compression ratio, compared to the LLEC. Fixed-point SPPEZC also improves computational complexity compared to floating-point SPPEZC because the fixed-point version optimizes the fixed-point DCT computation. The results in Tables 6 and 7 indicate that the fixed-point SPPEZC performs extremely well in computational efficiency and achieves satisfactory visual quality and compression performance.

It is crucial for the issues of technical feasibility and consumer electronics to implement fixed-point SPPEZC on embedded system platforms, such as DSP-based platforms. Because DSPs have low power consumption, they can be used in small, handheld, and even battery-operated devices, without additional cooling fans. Conversely, multimedia application developers often experience several unexpected difficulties when using a DSP, and not a desktop personal computer, as the target platform. These difficulties arise from architectural limitations of DSPs, including a lack of operating system



(a) Original. (b) SPPEZC. (c) LLEC.

(d) JPEG. (e) JPEG-O.

Fig. 13. Visual comparison results of the sample image "Jet" at 0.25 bit rate.

services, limited memory resources, lower CPU clock frequencies, and fixed point architectures (without floating point units). Thus, some platform-based optimizations associated with the DSP-based systems are necessary for implementing the proposed SPPEZC efficiently on a DSP system. This study adopts Lee's fast Discrete Cosine Transform (DCT) [25] to optimize the computational efficiency of a 2D DCT. To reduce computational complexity, this study presents a lookup table-based approach to substitute the cosine calculations in the original DCT. To lower memory consumption of the look-up tables, the symmetric and anti-symmetric relationship of the cosine computations were applied [26].

The proposed fixed-point SPPEZC was also implemented on the TI OMAP3530 embedded DSP platform [27], which contains 520 MHz DSP and 720 MHz ARM. Table 8 shows the DSP performance. The coding efficiency of the encoder and decoder reached 10.63 and 11.56 frames per second (FPS) on average for the OMAP3530 platform, respectively. Memory consumption of the SPPEZC on both DSPs was approximately 25 Kb. These results demonstrate that the proposed fixed-point SPPEZC is compatible with embedded applications and achieves promising computational and compression efficiency in both the encoding and decoding of images.

Figs. 6–9 illustrate the visual quality of the perceptual coding scheme versus the LLEC, JPEG, and JPEG-O in four directional edge patterns at a 0.25 bit rate. Fig. 6 shows increased $3\pi/4$ edge detection on the left eye of the Barbara image; because the corner of the eye includes the characteristics of edge information, SPPEZC achieves superior visual quality when coding the corner of the eye. Fig. 7 shows more $\pi/4$ edges detected on the shadow of the hat in the Elain image. For Fig. 7, this study focuses on the visual quality in coding the edge of the hat, demonstrating that SPPEZC provides improved quality, compared to the other methods. Fig. 8 shows the shadow, which includes $\pi/2$ edges of the apple of pepper image for detection. SPPEZC preserves more information of the apple than other methods. In Fig. 9, the shadow of the cloud above the airplane consists of π and $3\pi/4$ edges detected in the jet image. Similarly, SPPEZC determines the edge information of the cloud's shadow for improved visual quality. These results indicate that the perceptual coding scheme enhances the visual quality in image compression applications.

Figs. 10–14 show comparisons of the images at 0.25 bit rate. The analysis is summarized as follows:

- In the proposed approach, SPPEZC provides an improved visual effect in smooth regions. For example, SPPEZC performance is of good quality for coding human skin, as shown in Figs. 10–12.
- Although SPPEZC consistently outperforms JPEG and JPEG-O in PSNR measures, it quantizes more coefficients in the median to high frequency band to reduce visual quality in the sharpness regions (i.e., the lines of pants in the Barbara image).

(a) Original. (b) SPPEZC. (c) LLEC.

(d) JPEG. (e) JPEG-O.

Fig. 14. Visual comparison results of the sample image "Peppers" at 0.25 bit rate.

- Table 5 shows that the increase in the compression ratio is significantly greater than the decrease in coding quality. Therefore, the SPPEZC is effective for further promoting the resulting PSNR performance.
- Because the quantization table of the JPEG and JPEG-O preserves more of the edges of the image than step quantization, it provides enhanced visual quality within the sharpness regions.
- The SPPEZC has also improved and overcome the drawbacks of the original LLEC, because the SPPEZC exploits the block-edge information and varies the quantization tables adaptively to perform perceptual image coding. SPPEZC performance is considerably superior to the original LLEC for the edge regions (i.e., the characters on the body of the jet in Fig. 13).

5. Conclusions

This paper proposes a block-edge based Single-Pass Perceptual Embedded Zero-tree Coding (SPPEZC) method that combines the schemes of the Block-Edge Detection (BED) and Low-Complexity and Low-Memory Entropy Coder (LLEC) methods. The statistical results of the BED in this study show that some dominating coefficients are present in specific directional edges. A statistical experiment of the directional edges of the five test images shows that most of the block edges have the same dominating coefficients. To maintain the information of characteristics, the BED results are used for adaptive adjustment of the quantization tables of the DCT coefficients. Therefore, the LLEC method codes the modulated coefficients for balanced compression quality and complexity. Experimental results confirm that the proposed approach achieves satisfactory image compression performance in both the Peak Signal-to-Noise Ratio (PSNR) and visual quality. Compared to other coding techniques, the SPPEZC exhibits adequate and effective image compression for varying images. Floating-point and fixed-point SPPEZC provide similar performance in PSNR. This study also compares the proposed SPPEZC with other image coders on computational complexity. Floating-point SPPEZC can save approximately 20 and 10 ms on encoding and decoding, respectively, compared with the JPEG standard. Thus, the proposed SPPEZC can provide increased coding quality over the LLEC and only takes slightly more computational times. The fixed-point SPPEZC was also demonstrated on the DSP-based embedded platform TI OMAP3530. The coding computational efficiency of the encoder and decoder reached 10.63 and 11.56 FPS, respectively, on average for the OMAP3530. The proposed SPPEZC on DSP consumes only approximately 25 Kb of memory for the encoding and decoding parts. The SPPEZC outperforms the other three coders because of the proposed perceptual coding strategy on preserving edge information. Therefore, the proposed combinative coding scheme performs well in coding quality and perceptual strategy with lower computational complexity.

Acknowledgments

This paper was supported by the National Science Council of the R.O.C. under contract numbers NSC-100-2221-E-009-041 and NSC-100-2221-E-027-033. The authors would like to thank Mr. Chin-Ming Hsieh for his help on conducting the experiments.

References

[1] Q.-E. Wu, X.-M. Pang, Z.-Y. Han, Fuzzy automata system with application to target recognition based on image processing, Comput. Math. Appl. 61 (5) (2011) 1267–1277.
[2] Y.-L. Lin, M.-S. Wu, An edge property-based neighborhood region search strategy for fractal image compression, Comput. Math. Appl. 62 (1) (2011) 310–318.
[3] T. Shohdohji, Y. Hoshino, N. Kutsuwada, Optimization of quantization table based on visual characteristics in DCT image coding, Comput. Math. Appl. 37 (11–12) (1999) 225–232.
[4] Y. Tian, K. Zhao, Y. Xu, F. Peng, An image compression method based on the multi-resolution characteristics of BEMD, Comput. Math. Appl. 61 (8) (2011) 2141–2147.
[5] W. Pennebaker, J. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1993.
[6] Information technology-JPEG2000 image coding system, Tech. Rep., ISO/IEC 15444-1, 2000.
[7] Coding of moving pictures and associated audio for digital storage media up to Sbout 1.5 Mbit/s, Tech. Rep., ISO/IEC IS 11 172(MPEG-1), 1993.
[8] Generic coding of moving pictures and associated audio, Tech. Rep., ISO/IEC DIS 13 818(MPEG-2), 1994.
[9] R. Koenen, Overview of the MPEG-4 version 1 standard, ISO/IEC JTC1/SC29/WG11 N1909, 1997.
[10] M.C. Chi, M.J. Chen, C.H. Yeh, Region-of-interest video coding based on rate and distortion variations for H.263+, Signal Process. Image Commun. 23 (2) (2008) 127–142.
[11] T. Wiegand, G.J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.
[12] Z. Xiong, K. Ramchandran, M. Orchard, Y. Zhang, A comparative study of DCT- and wavelet-based image coding, IEEE Trans. Circuits Syst. Video Technol. 9 (5) (1999) 692–695.
[13] D. Zhao, Y.K. Chan, Y. Lu, Embedded image coding based on novel organization of DCT coefficients, in: Proc. SPIE—Appl. Digital Image Process. XXIII, vol. 4115, USA, 2000, pp. 153–162.
[14] A. Said, W. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Technol. 6 (3) (1996) 243–250.
[15] A. Martucci, I. Sodagar, T. Chiang, Y. Zhang, A zerotree wavelet video coder, IEEE Trans. Circuits Syst. Video Technol. 7 (1) (1997) 109–118.
[16] D. Taubman, High performance scalable image compression with EBCOT, IEEE Trans. Image Process. 9 (7) (2000) 1158–1170.
[17] N.I. Cho, S.K. Mitra, Warped discrete cosine transform and its application in image compression, IEEE Trans. Circuits Syst. Video Technol. 10 (8) (2000) 1364–1373.
[18] D. Zhao, Y.K. Chan, W. Gao, Low-complexity and low-memory entropy coder for image compression, IEEE Trans. Circuits Syst. Video Technol. 11 (10) (2001) 1140–1145.
[19] H.S. Chang, K. Kang, A compressed domain scheme for classifying block edge patterns, IEEE Trans. Image Process. 14 (2) (2005) 145–151.
[20] Z. Zhang, S. Ma, H. Lui, Y. Gong, An edge detection approach based on directional wavelet transform, Comput. Math. Appl. 57 (8) (2009) 1265–1271.
[21] Y.C. Wei, J.F. Yang, Transformed-domain mode selection for H.264 intra-prediction improvement, IEICE Trans. Inf. Syst. E91-D (3) (2008) 825–835.
[22] H. Li, K.N. Ngan, Z. Wei, Fast and efficient method for block edge classification and its application in H.264/AVC video coding, IEEE Trans. Circuits Syst. Video Technol. 18 (6) (2008) 756–768.
[23] S. Golomb, Run-length encodings, IEEE Trans. Inform. Theory 12 (3) (1966) 399–401.
[24] R. Rice, Some practical universal noiseless coding techniques—part I–III, Jet Propulsion Laboratory, Pasadena, CA, Tech. Rep. JPL-79-22, JPL-83-17 and JPL-91-3, Mar. 1979, Mar. 1983, Nov. 1991.
[25] B. Lee, A new algorithm to compute the discrete cosine transform, IEEE Trans. Acoust. Speech Signal Process. 32 (6) (1984) 1243–1245.
[26] B.F. Wu, H.Y. Huang, Y.L. Chen, H.Y. Peng, J.H. Huang, Low-complexity advanced audio coding optimization and implementation on DSP, IEICE Trans. Inf. Syst. E93-D (5) (2010) 1225–1237.
[27] DaVinci Digital Video Processor—OMAP3530, Available at: http://www.ti.com/product/omap3530.