Signal Processing 120 (2016) 373–384

Contents lists available at ScienceDirect

Signal Processing

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

Near-Aperiodic Dot-Diffused Block Truncation Coding

Yun-Fu Liu [a,*], Jing-Ming Guo [a], Zong-Jhe Wu [a], Hua Lee [b]

[a] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[b] Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, United States

ARTICLE INFO

Article history:
Received 21 January 2015
Received in revised form
10 September 2015
Accepted 23 September 2015
Available online 8 October 2015

Keywords:
Image compression
Block Truncation Coding
Dot diffusion
Digital halftoning
Optimization

ABSTRACT

Block Truncation Coding (BTC) image compression technique is capable of extraordinary computational efficiency for high-framerate configurations under the trend of high image resolution. However, the recent development such as the Dot-Diffused BTC (DDBTC) method still encounters various challenges of visual artifacts, including impulsive noise, false contours and blocking effects. In this study, a new method, the Near-Aperiodic DDBTC (NADDBTC) technique, is proposed to analyze the tradeoff and balance between computational efficiency and visual artifacts. It is mainly because of the quantization levels and the co-optimized Class Matrix (CM) and Diffused Matrix (DM) during image compression procedure. In addition, the tiling method for bitmap generation is optimized for aperiodic compressed results. Experimental results show that this method is capable of excellent image quality and visual perception, as well as processing efficiency, at a level similar to the DDBTC technique by taking advantage of parallel processing in dot diffusion.

1. Introduction

Block Truncation Coding (BTC), introduced by Delp and Mitchell in 1979, is a technique for lossy image compression [1]. Compared with lossless compressions [2,3], lossy compression is capable of high compression rates. The basic concept of the BTC method is to divide the original image into several non-overlapped blocks, and each is quantized by two different tones. The objective of these two values is to retain the first- and second-moment characteristics of the original block. The traditional BTC method does not necessarily improve the coding gain compared to the existing compression techniques, such as JPEG or JPEG 2000 [4]. The computational complexity of BTC methods is considerably lower. This feature enables us to achieve high framerates for embedded-based surveillance systems with low-power processing constrains while high resolution scenarios. Meanwhile, features of the BTC are also exploited for data hiding or steganography [5].

In the prior studies, techniques have been designed and developed for the improvement of the performance of the BTC methods. The first category involves retaining the moment characteristics of the original image. Halverson et al. [6] generated a family of moment-preserving quantizers by utilizing higher moments. Udpikar and Raina [7] proposed a modified BTC algorithm, preserving only the first-order moment. This algorithm is optimal in the mean-square sense. The second category is to improve image quality by reducing block artifacts. Kanafani et al. [8] decomposed images into homogeneous and non-homogeneous blocks and utilizes BTC or Vector Quantization (VQ) algorithm [9] for compression. Then, block classification can be achieved by image segmentation using the Expectation-Maximization (EM) algorithm. Compared with BTC or VQ methods, this new technique is capable of improving the quality and fidelity of compressed images.

Horbelt and Crowcroft, [10] introduced a video codec algorithm for compression [11], incorporating Discrete Cosine Transform (DCT). The fundamental concept of this

* Corresponding author.
E-mail addresses: yunfuliu@gmail.com (Y.-F. Liu), jmguo@seed.net.tw (J.-M. Guo), m9907501@mail.ntust.edu.tw (Z.-J. Wu), hualee@ece.ucsb.edu (H. Lee).

http://dx.doi.org/10.1016/j.sigpro.2015.09.030
0165-1684/

technique is that traditional BTC is effective in regions with high-contrast details, while DCT performs well for smooth regions. However, under low bit-rate conditions, BTC starts to exhibit poor image quality. Kamel et al. [12] then introduced two approaches for the modifications of BTC. One is to partition an image into variable block sizes instead. The other technique exploits for the optimal threshold for minimum square errors in quantizing the blocks. In addition, Chen and Liu then introduced the Visual Pattern Block Truncation Coding (VPBTC) method, utilizing the bitmap to compute the block gradient orientation and the matched block pattern. The trade-off of this algorithm is the substantial increase of the computation complexity [13].

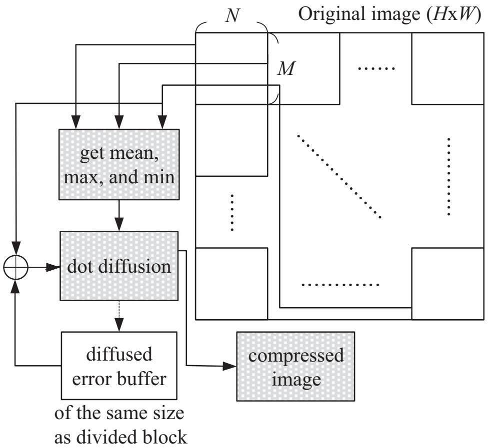

Fig. 1. Flowchart of the DDBTC [24].

Recently, halftoning-based BTC schemes have been developed to improve image quality and minimize computational complexity. Halftoning is a technique for printing documents with two quantized levels in each color channel, and results in continue-tone images when visualized from a distance [14]. In these halftoning-based BTCs, the halftoning is adopted to render the bitmap of BTC images. When the Human Visual System (HVS) is involved, halftoning-based BTC schemes can effectively reduce the blocking artifacts and false contours exhibited in the traditional BTC. Guo exploited error diffusion [15–21] to diffuse the quantization errors to the neighboring pixels for the retention of local average tone, which is known as the Error-Diffused Block Truncation Coding (EDBTC) method [22]. The adopted error diffusion method induces the inherent characteristics to compensate for the quantized error of a position by diffusing the error to the

Table 1
Two optimized pairs of class matrix and the corresponding diffused matrix.

(a) Class matrix of size 8×8

2	32	5	18	35	38	11	12
24	62	61	50	47	45	23	16
15	10	55	56	37	63	13	20
28	41	54	33	57	58	36	29
52	53	46	42	8	51	40	30
60	48	43	9	31	26	49	44
4	14	21	25	27	34	39	59
6	1	19	17	22	3	7	0

(b) Diffused matrix for the 8×8 class matrix

0.2564	1	0.2564
1	x	1
0.2564	1	0.2564

(c) Class matrix of size 16×16

164	144	114	63	39	42	47	44	6	151	95	162	165	76	246	203
244	13	21	50	60	57	82	89	104	145	158	87	177	196	197	206
5	34	27	2	58	70	98	99	15	157	130	186	137	97	208	79
23	25	41	53	67	51	1119	112	142	159	176	201	222	223	225	229
22	16	48	20	94	226	103	133	111	173	166	221	184	170	227	4
31	100	66	35	106	117	155	138	73	182	193	224	232	238	187	32
3	120	102	105	123	135	96	179	198	202	220	230	7	235	17	43
153	122	83	109	160	118	178	183	204	194	150	231	255	11	56	49
40	131	116	147	169	175	113	213	218	242	243	248	247	33	52	68
108	136	140	84	185	128	214	217	233	152	249	28	181	45	74	71
110	141	88	75	192	205	195	234	127	237	253	38	65	77	72	115
146	148	161	174	124	211	168	240	2511	252	254	55	64	12	167	125
149	180	156	191	81	216	236	245	26	0	171	46	92	101	143	132
189	188	107	209	210	228	250	18	37	59	69	91	134	139	85	163
1	199	241	212	93	8	30	36	61	62	90	86	239	129	154	172
200	207	219	9	10	24	14	54	29	78	80	121	126	19	215	190

(d) Diffused matrix for the 16×16 class matrix

0.3165	1	0.3165
1	x	1
0.3165	1	0.3165