# Block-based Fast Compression for Compound Images

*Wenpeng Ding[1*], Dong Liu[2*], Yuwen He[3], Feng Wu[3]*

[1]Dept. of Computer Science, [2]Dept. of Electronic Science and Technology
University of Science and Technology of China
Email: {wpding3, liud} @mail.ustc.edu.cn
[3]Microsoft Research Asia, Beijing 100080 China
Email: heyuwen@tsinghua.org.cn, fengwu@microsoft.com

***Abstract*****—**This paper presents a novel block-based fast compression (BFC) algorithm for compound images that contain graphics, text and natural images. The images are divided to blocks, which are classified into four different types – smooth blocks, text blocks, hybrid blocks and picture blocks with a fast and effective block-based classification algorithm. Four different coding algorithms are carefully designed for each block type according to their different statistical properties to maximize the compression performance. Simulations show that the BFC algorithm we propose has much lower complexity than DjVu with significant better visual quality at high bit rate, and it also outperforms the popular lossy image coding method JPEG.

## I. INTRODUCTION

With the widespread of digital devices such as digital cameras, personal computers, more and more compound images, containing text, graphics and natural images, are available in digital forms such as screen images, web pages. The sensitivity of human eyes for natural image and text is different. The quality requirement of compound image coding is different from general image coding because users cannot accept the quality if text is not clear enough to recognize. How to compress compound image is a hard problem and it is addressed in part6 JPEG2000 [1]. A lot of algorithms have been designed to compress images with different types. The Lempel-Ziv algorithm [2] is designed to compress pure text images, which only have text on the pure color background in the whole images. The JPEG [3] algorithm is suitable for pure picture images which do not have any text in the whole images, but has bad performance on pure text images. Several algorithms are also proposed to compress compound images. One kind of approaches for compound images is layered coding [4~6]. Most layered coding algorithms use the standard three layer mixed raster content representation [4]. One popular method is DjVu [5], which uses a wavelet-based codec (IW44) for background and foreground, and JB2 for mask layer. However, the complexity of mask generating and IW44 wavelet transform is high, which makes DjVu not suitable for real time application. What's more, DjVu shows bad performance on pure text images.

Block-based approaches for compound images are also studied for their low complexity. Said et al. [7] proposed a simple blocked-based scheme, which compresses text blocks using JPEG-LS, picture blocks using JPEG. However, it fails to handle the hybrid blocks, which contains mixed text and pictures.

In text area, there are strong edges which cannot be handled effectively by DCT based coding such as JPEG. In this paper, we present a novel algorithm, which can adaptively compress images with different content types, such as pure text images, pure picture images, and compound images. Four coding algorithms are designed for those blocks with different types based on a fast block classification method. Our BFC algorithm has several advantages over other compound image compression methods.

(1) The block classification algorithm and compression algorithm have low calculation complexity, which makes BFC very suitable for real-time application.

(2) The BFC algorithm can effectively compress the hybrid blocks, which are not well handled by some block-based algorithms.

(3) The BFC algorithm achieves good coding performance on text images, picture images and compound images. It also outperforms DjVu on compound images at high bitrate.

The remainder of this paper is organized as follows. In section 2, the block-based fast compression scheme is introduced. Section 3 presents the block classification technique. Then in sections 4, the algorithms for each block types are discussed in detail. The experimental results are presented in section 5, followed by conclusion in section 6.

## II. BLOCK-BASED FAST COMPRESSION

The framework of the block-based fast compression (BFC) scheme is shown in Fig.1. The compound image is first divided into 16x16 blocks. Then blocks are classified into four types: smooth, text, hybrid and picture according to their different statistical characteristics. Blocks of different type will be compressed with different algorithms, which will be discussed in section III. The block type map is compressed using an arithmetic coder.

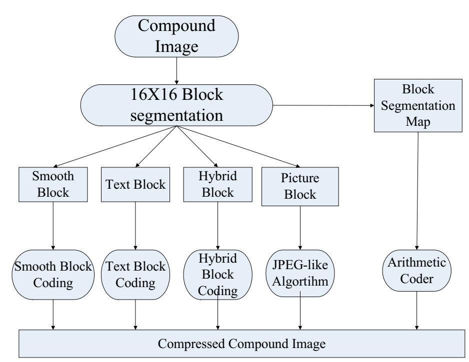



**Figure 1: The block-based fast compression scheme**

* This work was done while the authors visiting Microsoft Research Asia.

## III. BLOCK CLAISIFFICATION

The block classification algorithm is often a computation demanding task. We propose a fast and effective classification algorithm based on two features: histogram and gradient of the block. The pixels of each block are first grouped into three classes: low-gradient pixels, mid-gradient pixels and high-gradient pixels according to pixel's gradient value. Then the histogram distribution for each pixel group is computed. The typical gradient-histogram distribution is shown in Fig.2.

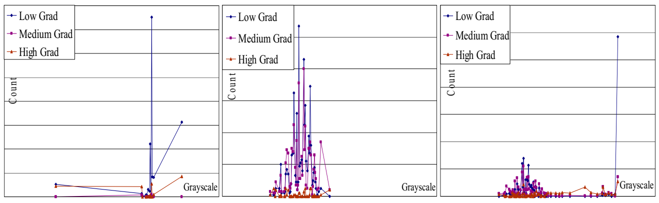



**Figure 2: The three histograms of text block (left), picture block(mid) and hybrid block(right)**

As we observe in Fig.2, blocks of different type shows up different gradient-histogram distributions. We classify the blocks into four types: smooth, text, hybrid and picture blocks based on gradient-histogram distribution. The smooth blocks typically contain only low gradient pixels and show one peak at the low-gradient histogram. The text blocks always show several peaks at the low-gradient and high-gradient histograms. Only a few mid-gradient pixels can be observed in text blocks. If the block contains large numbers of high-gradient pixels and mid-gradient pixels, it will be identified as hybrid block. The blocks mainly consist of mid-gradient pixels are declared as picture blocks. The classification flow is shown in Fig.3. Here T1~T7 are thresholds for judging the block types.

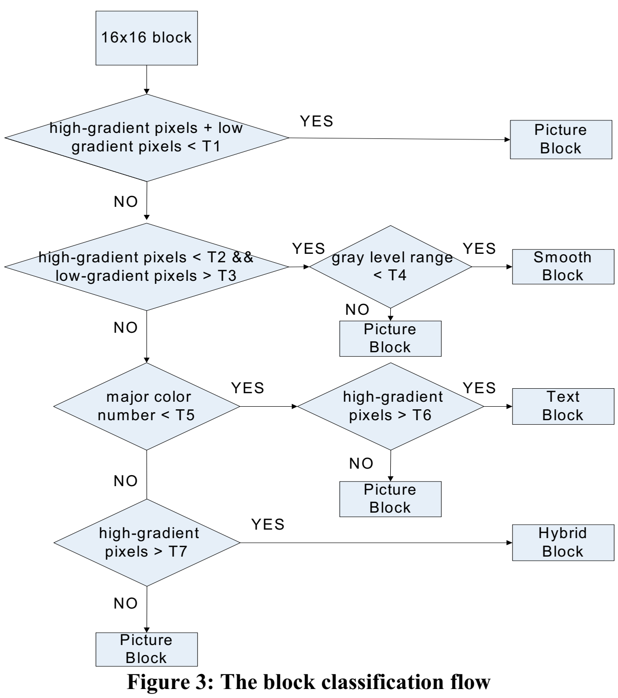



**Figure 3: The block classification flow**

## IV. BLOCK CODING

Blocks of different type are distinct in nature and have different statistics distributions. Smooth blocks are very flat and dominated by one kind of color. Text blocks are more compact in spatial domain than that in DCT domain. The energy of picture blocks is mainly concentrated on low frequency coefficients when they are DCT transformed. Hybrid blocks, containing mixed text and picture images, cannot be compactly represented both in spatial and frequency domain. Four coding algorithms are carefully designed to compress blocks of different types effectively.

### *A. Smooth Block Coding*

The coding of smooth blocks is straightforward. Smooth blocks are dominated by one color and their gray level range is limited to the given threshold. All the colors in smooth blocks are quantized to the most frequent color, which is coded using an arithmetic coder.

### *B. Text Block Coding*

The text blocks are typically dominated by several major colors. The colors with frequency above the given threshold are chosen as major colors. If there are more than four colors satisfying above requirement, only the first four colors with largest number in luminance histogram will be chosen as major colors. The colors close to major colors within the given distance threshold are quantized to their corresponding major colors. The color quantization algorithm can be described as following.

**Algorithm1: Color Quantization**

```
Find first four major colors {M_0, M_1, M_2, M_3};
FOR each pixel P_i
   FOR each major color j
      IF |P_i - M_j| < Th  for some j, THEN
            P_i := M_j;
      ENDIF;
   ENDFOR;
ENDFOR;
```

Every pixel's color in the text block is first converted to color index. The major colors are indexed by 0, 1, 2 and 3 respectively. All the other colors are converted to index 4. The major colors in each block are recorded.

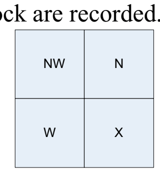



**Figure 4: Text block coding contexts**

The text block index is scanned and compressed in a raster scanning order and the current pixel index is coded based on its causal neighbors as shown in Fig.4 to exploit the spatial relevance in order to improve coding efficiency. "X" is current pixel to be coded. Each neighbor may be five different index values; there are total $5^3 = 125$ contexts for coding the current pixel X. The current pixel index is coded by an arithmetic coder using the context specified by its three casual neighbors [NW, N, W]. If the current pixel index is 4, the pixel value is also coded using an arithmetic coder.

**Algorithm2: Text Block Coding**

```
Quantize the block using Color Quantization;
Convert the block pixels to index;
FOR row = 1 to last row
    FOR column = 1 to last column
        Get index for current pixel;
        /*Generate Context for current pixel*/
        context := NW*25+N*5+W;
        Code current pixel index using the context;
        IF   pixel index is 4 ( other colors) THEN
             code the current pixel value;
        ENDIF
    ENDFOR
ENDFOR
```

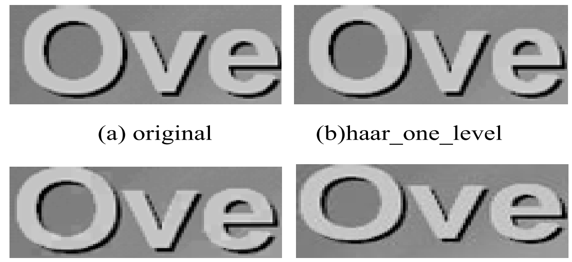
### *C. Hybrid Block Coding*

Hybrid blocks contain mixed text and pictures. There are strong high frequency signals due to text edges and DCT transform is only effective to compact the energy of low frequency signals, so the energy in DCT domain of hybrid block is very diverse and hard to code. When Hybrid blocks are compressed with DCT block transform based coding such as JPEG, it will suffer from ringing effects around the text due to large quantization step for those high frequency components. Wavelet-based schemes such as JPEG-2K fail to compress hybrid blocks effectively. While hybrid blocks are compressed with document image algorithms, the coding performance is too low to be acceptable.

One solution to this problem is layered coding such as DjVu. The text and pictures are separated into different layers and independently coded.

We propose a haar wavelet based coding algorithm for hybrid blocks. As we know, short wavelet bases are helpful to reduce the ringing effect around text (edge), and longer bases are good to improve the coding performance of the picture images. As a tradeoff between two requirements, we choose haar wavelet to code hybrid blocks. Although haar wavelet's poor performance on pure picture images, it can effectively remove the ringing effect on text images. Its coding performance outperforms other coding algorithms both in PSNR and visual quality.

The hybrid block is first transformed with haar wavelet. Here that we only use one level haar wavelet transform, since multilevel haar wavelet transform will produce long wavelet bases not suitable for hybrid blocks. The wavelet coefficients are then coded by a simple arithmetic coder. The coefficients of different subbands are coded using different contexts. The simple haar wavelet algorithm can significantly improve the visual quality and PSNR of images with hybrid blocks. The visual quality of reconstructed images of three methods is compared in Fig.5: coded by one level haar wavelet, three levels haar wavelet and DCT. The PSNR-bitrate curves of the three methods are shown in Fig.6. It is obvious that one level haar wavelet achieves the best coding performance for hybrids blocks.




(a) original (b)haar_one_level

(c) haar_three_level (d) DCT

**Figure 5: Visual quality comparisons of reconstructed images**

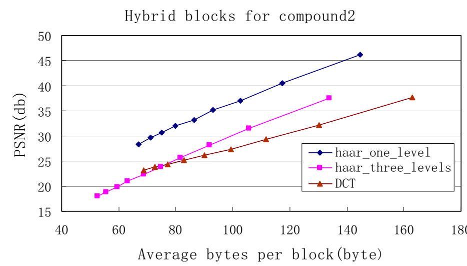



**Figure 6: PSNR-bitrate curves for hybrid blocks of three methods: haar_one_leval haar_three_level and DCT.**

### *D. Picutre Block Coding*

JPEG has been proved to be an effective low complexity algorithm to compress picture images. We propose a JPEG-like algorithm to compress the picture blocks. The difference is that we need to skip the blocks of other types in our block-based scheme. As we expect, the BFC algorithm achieves comparable coding performance to jpeg on pure photographic images with slight overhead of blocks type map.

## V. Experiment Results

Five images are tested in our experiments: compound1, compound2, compoud3, picture1 and text1, which are shown in Fig.7. Compound1~compound3 are examples of typical compound images with mixed text and pictures; Text1 is a document image; Picture1 is the example of pure picture images. The coding efficiency between BFC and JPEG are compared in Fig.8 for three of the compound images. The BFC clearly outperforms jpeg for all the images. In the best case, the BFC outperforms JPEG up to 7db for compound2, which contains a lot of hybrid blocks.

Fig.9 compares the visual quality of reconstructed images coded by JPEG, DjVu, and BFC at the same bitrate. It is obvious that BFC achieves much better visual quality than JPEG and also outperforms DjVu. The ringing affects round text in text blocks and hybrid blocks are successfully removed by BFC.

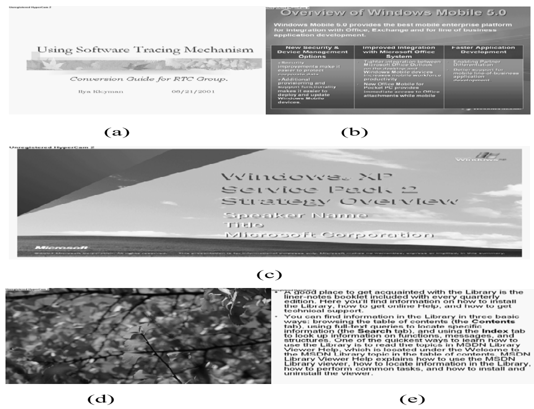


**Figure 7: Test images: (a) compound1, (b) compound2, (c) compound3, (d) picuture1 and (e) text1**

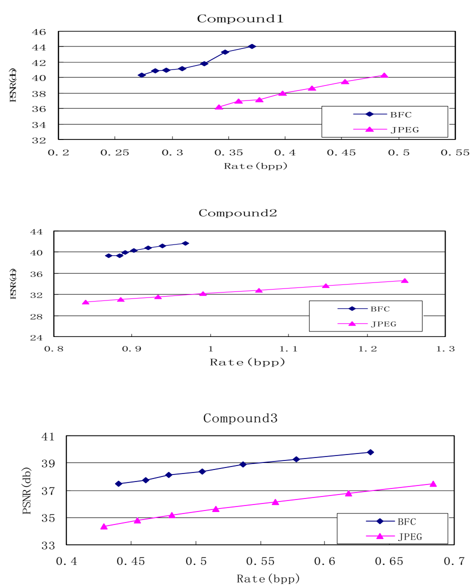



**Figure 8: PSNR-bitrate curves for compound1, compound2 and compound3 of BFC vs. JPEG**


**Figure 9: Visual quality comparisons of portion of reconstructed images: Top (Compound2) and Bottom (Compound3).**

**Table II: Comparison of compression ratio for test images among JPEG, LZW, DjVu and BFC**

| image | JPEG | LZW | DjVu | BFC |
|---|---|---|---|---|
| compound1 | 16.4 | 9.7 | 17.9 | 21.6 |
| compound2 | 6.4 | 4.9 | 6.2 | 8.3 |
| compound3 | 11.7 | 3.0 | 13.0 | 12.6 |
| picture1 | 10.1 | 1.5 | 23.7 | 10.1 |
| text1 | 4.1 | 14.5 | 3.9 | 19.0 |

Table II includes all the comparison ratios for the five test images using four different compression methods: JPEG, LZW, DjVu and BFC. Note that we compare lossless and lossy compression methods, but it is still a valid comparison because we set the quality factor of lossy methods to guarantee visually lossless quality. Here the quality factor of JPEG is set to 75. The DjVu is adjusted to the highest quality factor available. We adjust BFC's parameters to achieve better PSNR and visual quality than JPEG. In table II, the BFC algorithm gives significantly better performance on compound images than other algorithms. It also offers comparable performance on pure picture images with JPEG. The text images, difficult to compress by JPEG and DjVu, can also be efficiently compressed by BFC.

The BFC scheme is of low complexity since the algorithms used have low time complexity. The classification algorithm just uses the histogram of a block; the hybrid coder uses haar wavelet; the text coder uses context-based arithmetic coding; the most computation expensive part of scheme is picture coding, which use a JPEG-like algorithm. In the worst case, the BFC has comparable complexity with JPEG.

## VI. CONCLUSION

We presented a block-based fast compression algorithm for compound image coding. A fast block classification algorithm is proposed and four compression algorithms are carefully designed for blocks of different types. The BFC algorithm achieves comparable or better coding performance, on pure text images and pure picture images, to LZW and JPEG respectively. The BFC algorithm outperforms DjVu on compound images at high bitrate.

## REFERENCES

[1] "Coding of still images", ITU-T, ISO/IEC SG 8 FCD 15444-6, 2001

[2] J.Ziv and A. Lempel, "A universal algorithm for data compression", IEEE Trans. On Information Theory, IT-23(3), pp.337-343, May 1977.

[3] W. p. Pennebaker, J. L. Mitchell, "JPEG: Still image compression standard", Van Nostrand Reinhold, 1993,

[4] "Mixed Raster Content (MRC)", ITU-T Recommendation T.44, Study Group-8 Contribution, 1998,

[5] P. Haffner, L. Bottou, P.G. Howard, P. Simard, Y. Bengio, Y. LeCun, "High Quality document image compression with DjVu", Journal of Electronic Imaging, pp. 410-425 July 1998

[6] B.-f Wu, C.-C Chiu and Y. –L Chen "Algorithms for compressing compound document images with large text/background overlap", IEE Proc.Vis. Image signal Process., Vol. 151 No. 6, pp.453-459 December 2004,

[7] A. Said and A. Drukarev, "Simplified segmentation for compound image compression", Proceeding of ICIP' 1999, pp.229-233