Efficient Coding of Computer Generated Compound Images

Tony Lin and Pengwei Hao
National Laboratory on Machine Perception
Peking University
Beijing, China
{lintong, phao}@cis.pku.edu.cn

Sang Uk Lee
School of Electrical Engineering
Seoul National University
Seoul, Korea
sanguk@ipl.snu.ac.kr

***Abstract*—A new compound image compression algorithm is proposed, based on Shape Primitive Extraction and Coding (SPEC). The SPEC first segments a compound image into text/graphics pixels and pictorial pixels, by extracting the shape primitives of text/graphics. Then all the shape primitives are losslessly compressed with a combined shape-based and palette-based coding algorithm. The remaining pictorial pixels are coded with JPEG2000. Experimental results show that the SPEC has very low complexity and provides visually lossless quality, while yielding competitive compression ratios.**

Keywords-compound image compression; compound image segmentation; shape primitive extraction; shape-based coding; palette-based coding

I. INTRODUCTION

Computer generated compound images have been popular in recent years as they deliver abundant information by exhibiting a combination of text, graphics, and natural pictures. However, traditional image compression algorithms such as JPEG produce annoying artifacts around the text and graphics in the compound images, while these artifacts can be easily noticeable. Fig. 1 shows such examples. On the other hand, lossless coding methods such as GIF often achieve poor compression ratios for these compound images.

Compound image compression has been extensively studied for scanned document images [1-6]. Most algorithms use the standard 3-layer Mixed Raster Content (MRC) representation. For example, DjVu [1-2] uses a wavelet-based coder (IW44) for the background and the foreground layers, and JB2 for the mask layer. The segmentation is based on hierarchical color clustering and uses a variety of filters. Commonly, these scanned image compression algorithms can not achieve satisfactory performance for computer generated compound images. Only recently, a lossless coding [7] and a lossy coding [8] are proposed for computer generated images, but obviously they can not offer a preferable tradeoff between compression ratio and image quality.

This work was supported by NSFC (60302005), FANEDD (200038), NKBRPC (2004CB318005) and KFAS ISEF.

In this paper, we propose a low complexity and high quality compression algorithm – Shape Primitive Extraction and Coding (SPEC). The SPEC accurately segments text/graphics from pictures, and provides a new lossless coding method for text/graphics. SPEC has two unique features: 1) Shape and color serve as two basic features to the segmentation and the lossless coding; 2) Segmentation and coding are tightly integrated in the SPEC algorithm.

Fig. 1. Examples of JPEG compressed compound images.

II. SPEC – THE PROPOSED ALGORITHM

A. The System

As shown in Fig. 2, the proposed SPEC algorithm consists of two stages: segmentation and coding. The algorithm first segments 16x16 non-overlapping blocks of pixels into text/graphics class and picture class, and then compresses text/graphics with a new lossless coding algorithm and all the remaining pictorial pixels with JPEG2000, respectively.

Shape primitives refer to those elementary building units that compose text/graphics in a compound image. For simplicity, four types of shape primitives are used in the SPEC: isolated pixels, horizontal lines (one pixel high), vertical lines (one pixel wide), and rectangles (with horizontal and vertical sides). A shape primitive is defined to have the same interior color. Straightforward, a shape primitive can be represented by a color tag and its position information, i.e., (x, y) is for an isolated pixel, (x, y, w) for a horizontal line, (x, y, h) for a vertical line, and (x, y, w, h) for a rectangle, respectively.

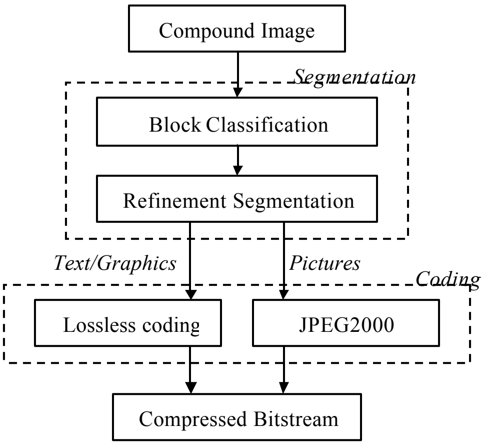

Fig. 2. Flow chart of the SPEC system.

B. Segmentation

The segmentation is a two-step procedure, including block classification and refinement segmentation. The first step is to classify 16x16 non-overlapping blocks into text/graphics blocks and picture blocks by thresholding the number of colors in each block. Each block is scanned to count the number of different colors. If the color number is larger than a certain threshold T_1 (T_1=32 is used for SPEC), the block is classified as the picture block. Otherwise, it is classified as text/graphics block. The underlying assumption is that a continuous-tone picture generally exhibits a large number of different colors. The block classification based on counting different colors can be extremely fast.

Since the block classification is a coarse segmentation, a refinement segmentation is followed to extract text and graphics pixels from picture blocks to enhance the performance. For efficiency, pictorial pixels in text/graphics blocks are not segmented. The segmentation is implemented by scanning each picture block to extract four types of shape primitives. A size-first strategy is adopted to represent the text/graphics in one block with a short list of shape primitives.

Apparently, we should refine the extracted shape primitives to separate text/graphics pixels from pictorial pixels. Because the shape primitives include isolated pixels, every pixel in picture blocks may be misclassified into shape primitive pixels. Moreover, for monotonous regions in a picture, several adjoining pixels may have the same color.

Whether a shape primitive is classified into a text/graphics class depends on its size and color. If the size is larger than the specified threshold T_2 (T_2=3 is used for SPEC), the shape primitive is extracted as text/graphics pixels. Otherwise, the color of the shape primitive is compared to a dynamic palette of recent text/graphics colors. If an exact color match is found, then the shape primitive is considered as text/graphics. If the size of the shape primitive is larger than a threshold T_3 (T_3=5 for SPEC), its color is put into the dynamic color palette. The dynamic color palette is implemented with a first-in first-out buffer and eight entries.

There are several reasons for designing such a procedure to detect the shape primitives of text and graphics. First, most pictorial pixels are found to be isolated pixels, because there is little possibility that several neighboring pictorial pixels yield exactly the same color. Even if this happens, the size of these pictorial regions is usually small. Moreover, if the neighboring pictorial pixels are of a large number, they are classified into text/graphics and can be still efficiently coded by lossless coding. Second, for shape primitives of small sizes, we can make decision based on most recent colors of text/graphics pixels. In computer generated images, the color of textual and graphical content generally has some coherence, and it is unusual that text/graphics colors change frequently. Finally, this procedure is computationally efficient.

The two step segmentation successfully segments the image into two parts – text/graphics pixels and pictorial pixels, where text/graphics pixels include all pixels of text/graphics blocks and shape primitive pixels, and pictorial pixels are the remaining pixels in the picture blocks. Fig. 3 shows an example of the segmentation.

(a) (b) (c)

Fig. 3. An example of the segmentation by SPEC. (a) Source image; (b) Text/graphics pixels; (c) Pictorial pixels.

C. Lossless Coding of Text/Graphics Pixels

The proposed lossless coding of text/graphics pixels is mainly based on the shape primitives. To compress the text/graphics blocks, the shape primitives are extracted first. The extraction procedure is similar to that of the picture blocks. A simple shape-based coding is used to represent the shape primitives. In a 16x16 block, 8-bit (x, y), 12-bit (x, y, w), 12-bit (x, y, h), and 16-bit (x, y, w, h) are used to represent isolated pixels, horizontal lines, vertical lines, and rectangles, respectively. For each color, we use a run-length encoding scheme to represent the counts of four types of the shape primitives.

Sometimes there are too many small shape primitives in a complicated block, making shape-based coding inefficient. Therefore, palette-based coding is a good alternative. For multiple colors, palette-based coding uses a multiple-bit mask. We choose one from the following three cases to achieve the minimal code length: 1) all colors are shape-based coded, 2) all colors are palette-based coded, 3) only the most shape-complicated color is palette-based coded, while other colors are shape-based coded. The most shape-complicated color is the color that generates the maximum coding length when all colors are shape-based coded.

We apply a color table reuse technique to represent the colors of shape primitives. Generally, most colors are the same in the color tables of two consecutive blocks. If a color of the current block is found to match the color table of the previous block, it is represented by a 1-byte index. Otherwise, it is represented by a 3-byte (R, G, B) format.

For each block, the lossless coding stream is organized as follows. First, the color number and the color table are recorded. Second, the encoding method is specified by the number of colors being shape-based coded, the number of colors being palette-based coded. If there is a background color, we record its index in the block color table. If there is a color being coded by 1-bit palette, we record its index, too. Then, the shape primitives in each color are represented by a combined shape-based and palette-based coding algorithm. Finally, the coded stream is fed into a LZW coder for further compression.

D. Pictorial Pixel Coding with JPEG2000

We employ JPEG2000 to compress all the remaining pictorial pixels. In order to alleviate the ringing artifacts and to achieve higher compression ratio, text/graphics pixels in the picture blocks are removed before the JPEG2000 coding. Though the values of these removed pixels can be arbitrarily chosen, it would be better if these values are similar to the neighbor pictorial pixels. This produces a smooth picture block. We therefore fill in these holes with the average color of pictorial pixels in the block.

III. EXPERIMENTAL RESULTS

Fig. 4 shows the test image used in our experiments. First, we compare the performance of SPEC and DjVu on compound image segmentation. The segmentation results shown in Fig. 5 demonstrate that SPEC achieves more accurate segmentation than DjVu. Fig. 6 compares the visual quality of the reconstructed images compressed by JPEG, JPEG2000, DjVu, and SPEC at compression ratio 16:1. From Fig. 6, we can see that SPEC achieves the best visual quality than other algorithms. In addition, SPEC spends only 600ms to encode the image while DjVu need around 4 seconds on a 3.2GHz P4 machine.

IV. CONCLUSTION

We have presented a compound image compression algorithm called SPEC. Two main contributions of this work are: 1) an accurate segmentation algorithm is developed to separate text/graphics from pictures; 2) a lossless coding method is designed for text/graphics compression. Experimental results demonstrated that SPEC is an algorithm of low complexity. It also provides excellent visual quality and competitive compression ratio.

REFERENCES

[1] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun, "High quality document image compression with DjVu," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410-425, July 1998.

[2] P. Haffner, L. Bottou, P. G. Howard, and Y. LeCun, " DjVu: Analyzing and compressing scanned documents for Internet distribution," *Int. Conf. Docu. Anal. and Recog.*, 1999.

[3] D. huttenlocher, P. Felzenszwalb, and W. Rucklidge, " DigiPaper: A versatile color document image representation," *Proc. ICIP*, vol. I, pp. 219-223, Oct. 1999.

[4] H. Cheng and C. A. Bouman, " Document Compression Using Rate-Distortion Optimized Segmentation," *Journal of Electronic Imaging*, vol. 10, no. 2, pp. 460-474, April 2001.

[5] D. Mukherjee, N. Memon, and A. Said, " JPEG2000-matched MRC compression of compound documents, *Proc. ICIP*, vol. 3, pp. 73-76, 2002.

[6] D. Mukherjee, C. Chrysafis, and A. Said, " Low complexity guaranteed fit compound document compression," *Proc. ICIP*, vol. I, pp. 225-228, 2002.

[7] X. Li and S. Lei, " On the study of lossless compression of computer generated compound images," *Proc. ICIP*, vol. 3, pp. 446-449, 2001.

[8] A. Said, "Efficient and reliable dynamic quality control for compression of compound document images," *Proc. ICIP*, 2004.

Fig. 4. The test image (700x500).

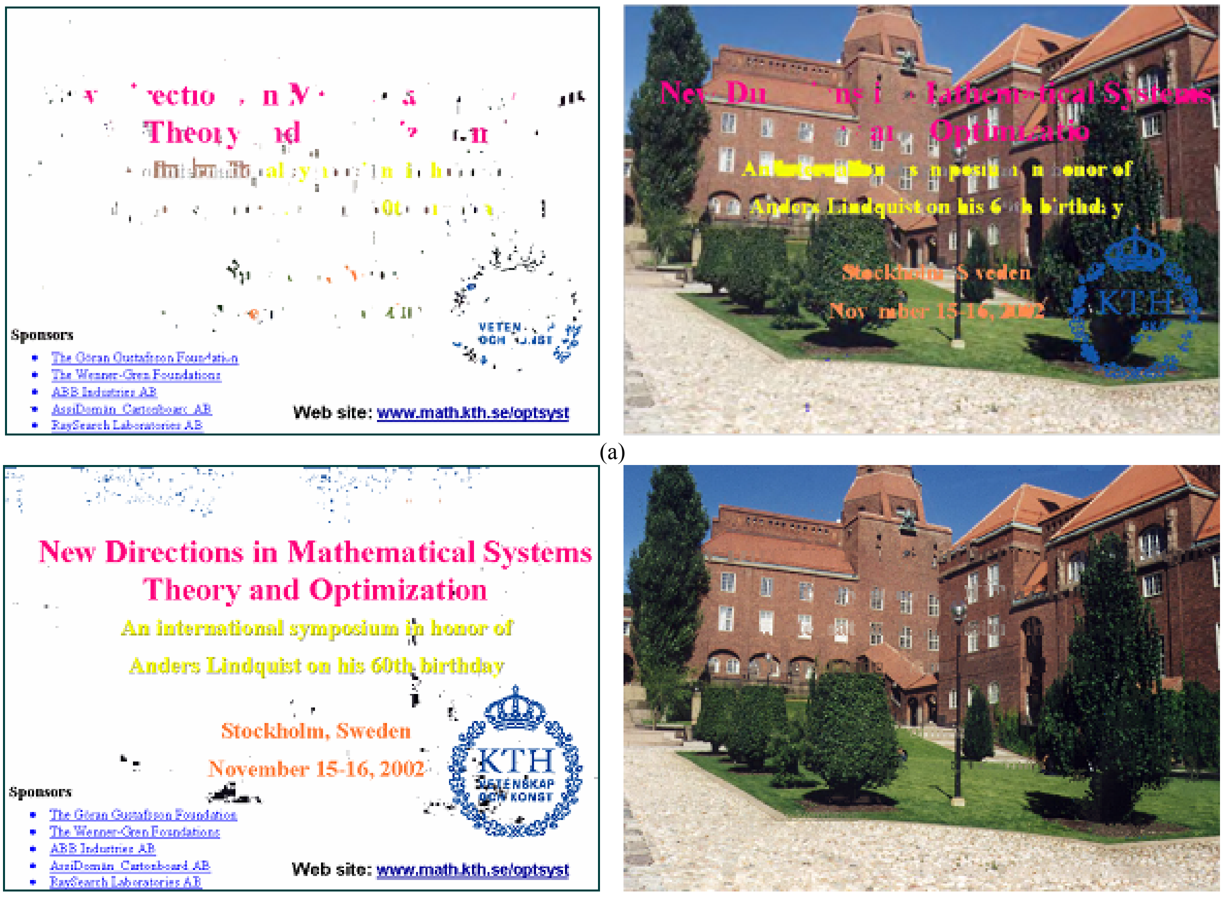

Fig. 5. Segmentation results: (a) DjVu; (b) SPEC.

Fig. 5. Compression results (with compression ratio 16:1): (a) JPEG; (b) JPEG2000; (c) DjVu; (d) SPEC.