COMPRESSION OF COMPOUND DOCUMENTS

Ricardo L. de Queiroz

Xerox Corporation
800 Phillips Rd., 128-27E, Webster, NY, 14580
queiroz@wrc.xerox.com

ABSTRACT

Compound (or mixed) document images contain graphic or textual content along with pictures. They are a very common form of documents, found in magazines, brochures, web-sites etc. Because of the very distinct nature of those two image classes (text/graphics vs. pictures), their compression invariably involves multiple compression systems and a region segmentation (classification) method. We review state-of-the-art technologies on the subject while focusing our attention on the mixed raster content (MRC) multi-layer approach. We also present new results on segmentation for MRC based on optimized rate-distortion-based block thresholding.

1. INTRODUCTION

Documents are now present in a wide spectrum of printing systems. From offset printers to home desktop computers, documents in digital form are common place. Frequently, documents are available as bitmaps and may contain text, graphics and pictures. Compound documents are images which contain a mix of textual, graphical, or pictorial contents. Those images are invariably large but a single compression algorithm that simultaneously meets the requirements for both text and image compression has been elusive. Many standard compression algorithms are available today and in common use commercially. More are continually being developed to improve on existing methods or to meet special requirements. As a rule, compression algorithms are developed with a particular image type, characteristic, and application in mind. For a different image type or application, a given algorithm either does not apply or does not perform as well as some other, better-tailored algorithm. No single algorithm is best across all image types or applications. When compressing text, it is important to preserve the edges and shapes of characters accurately to facilitate reading. Once the text is binarized, its compression is typically lossless since coding errors in text are easily perceived. The human visual system, however, works differently for typical continuous-tone images because of the richness of patterns and frequency contents. High frequency errors are better masked and lossy compression is usually employed, since lossless compression is often ineffective in this case. In terms of image resolution, text requires much higher resolution than pictures. Actually, roughly speaking, text requires few bits per pixel but many pixels per inch, while pictures require many bits per pixels but fewer pixels per inch.

Document compression is frequently linked to facsimile systems, in which large document bitmaps are compressed before transmission over telephone lines. The facsimile systems that most people are familiar with today are black-and-white (binary images) and conform to international standards set by the ITU-T (Telecommunication Standardization sector of the International Telecommunication Union, formerly known as the CCITT). These standards specify the protocols and bi-level coding procedures that sending and receiving stations use. Together with the ubiquity of the public switched telephone network (PSTN), these standards have led to the explosive growth in Group 3 black-and-white facsimile that has occurred since 1980. The same convenience and ease of use for color facsimile requires wider use of color scanners, displays and printers; faster modems and communication channels to handle the increased data volume; and equivalent standards for color facsimile. These enablers are already being put in place. For example, the ITU-T last year approved V.34 for facsimile, which supports data rates up to 33.6 Kbps, and it is now available commercially in fax machines. There is now a focus on new standards to provide color facsimile services over the PSTN and the Internet [1].

When it comes to compound documents, in order to cope with the differences between text and continuous tone images, different compression algorithms may be applied to each of the regions of the document. For that goal, some segmentation strategy has to invariably be used to discern which regions are to be encoded under which strategy. Another important parameter of a document compression system for compound documents is its imaging model. One can separate the image into different regions of interest and compress each region accordingly. In this case, the imaging model follows space segmentation where each decompressed region can be imaged into the document concurrently. Also, one can generate multiple image layers, compress each one separately and then image all the planes into one. The multilayer model will be the focus of this paper.

2. OVERVIEW

Image compression has been very intensively studied and we cannot possibly reference adequately all the most notable algorithms. However, in terms of international standards the notable algorithms for binary image compression are MH1 [2], MMR2 [3], JBIG [4] and the forthcoming JBIG-2 [5]. Multilevel compression algorithm standards are JPEG [6] and the forthcoming JPEG-2000 [7]. We assume that JPEG is the standard image compression tool while current JPEG 2000 verification model (VM) [8] is the state-of-the-art in image compression, when it comes to pictorial contents. For binary documents, MMR2 is adequate for text. JBIG can use arithmetic coding for improved performance and its multiresolution approach allows for compression of halftones. The new drive, however, in the compression of bi-level images is token-based compression. Contiguous objects are parsed and made

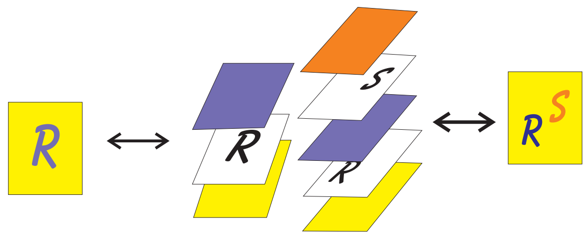

Figure 1. Illustration of MRC imaging model.

into entries in a dictionary. New objects are compared to the dictionary and if a match is found the code for that object is repeated. By making loose matches one allows the introduction of losses in exchange for higher compression. This is one of the key compression methods in document representation formats such as DigiPaper [9] and DjVu [10],[11]. Token-based compression is the heart of the forthcoming standard: JBIG-2 [5]. Even halftones can be compressed with token-based techniques by descreening the halftone and encoding new halftone patterns as objects [12]. For that, a segmentation needs to be performed to identify regions of graphics, text, halftones, etc., in a binary image, in order to improve token-based compression in JBIG-2 [13]. Other algorithms do exist which can handle graphic bitmaps well [14] and also algorithms that perform well (not optimally) for both text/graphics and pictures using non-linear filter banks [15].

Once a region is identified it can be encoded with the proper algorithm. For region identification, segmentation algorithms may be employed. For example the algorithms used in DjVu and Digipaper are already in commercial applications. Multiresolution segmentation was aplied succesfully in [16] for document compression, while [17] does the same using an approximate object location, in order to simplify the implementation. Multiscale clustering methods are also effective for segmentation [18]. We will present yet another segmentation algorithm based on block-thresholding in which the thresholds are optimized in a rate-distortion sense.

3. MIXED RASTER CONTENT

The mixed raster content (MRC) imaging model [1],[19],[20], allows for a multi-layer multi-resolution representation of a compound document. The basic 3-layer MRC model represents a color image as two color-image layers (Foreground or FG and Background or BG) and a binary image layer (Mask). The Mask layer describes how to reconstruct the final image from the FG/BG layers, i.e. to use the corresponding pixel from the FG or BG layers when the mask pixel is 1 or 0, respectively, in that position. An illustration of the imaging model is shown in Fig. 1. The foreground plane is essentially poured through the mask plane onto the background plane. The basic 3-layer model is MRC's most common form. The imaging model, however is composed of basic elementary plane pairs: FG+Mask. The FG layer is imaged onto a BG layer through the mask plane composing a new background image. Another foreground layer can be imaged onto this new background through another mask plane and the process can be repeated several times. The extended MRC model, then, allows for several planes while relying on foreground-mask pairs. A page may be represented as one, two, three or more layers, depending on its content. For example, a page consisting of a picture could use the background layer only. A page containing black-and-white text could use the mask layer, with the foreground and background layers defaulted to black and to white.

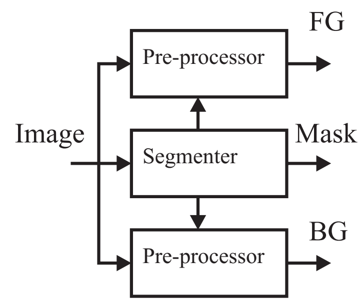

Figure 3. Diagram of a segmenter.

Once the original single-resolution image is decomposed into layers, each layer can be processed and compressed using different algorithms. The image processing operations can include a resolution change or color mapping. Layers may contain different dimensions and have offsets associated with them. If a plane contains only a small object, the effective plane can be made of a bounding box around the object. The reduced image plane is then imaged onto the larger reference plane, starting from the given offset (top, left) with given size (width, height). This avoids representing large blank areas and improves compression. The compression algorithm and resolution used for a given layer would be matched to the layer's content, allowing for improved compression while reducing distortion visibility. The compressed layers are then packaged in a format, such as TIFF-FX [21] or as an ITU-T MRC [19] data stream for delivery to the decoder. At the decoder, each plane is retrieved, decompressed, processed (which might include scaling) and the image is composed using the MRC imaging model.

MRC was originally approved for use in Group 3 color fax and is described in ITU-T Recommendation T.44. For the storage, archiving and general interchange of MRC-encoded image data, the TIFF-FX file format has been proposed [21]. TIFF-FX (TIFF for Fax eXtended) represents the coded data generated by the suite of ITU recommendations for facsimile, including single-compression methods MH, MR, MMR, JBIG and JPEG, as well as MRC. As IETF RFC 2301, TIFF-FX is a Proposed Internet Standard, currently undergoing interoperability testing. MRC has also been proposed as an architectural framework for JPEG 2000.

MRC has been used in products as DigiPaper and DjVu, whose owners built special segmenters for them, and also for check compression [22]. An analysis of the goals of the segmentation algorithm along with a better description of MRC can be found in [20]. Typical segmentation strategies are depicted in Fig. 2, which basically differ in whether one wants to move text and graphics shapes to the FG or the Mask plane. Since each layer (FG or BG) may contain unused pixels (since the pixels in that position will be selected from the other layer), those can be replaced by any color in order to enhance compression. This is the function of the pre-processor. The overall diagram is illustrated in Fig. 3. Given the pre-processors, the segmenter function is that of finding a binary mask for a given input, from which the pre-processor can derive the output layers based on the input image.

In this paper, we are interested in designing the pre-processor and segmenter for optimized compression following a basic 3-layer MRC approach. For simplicity we assume layers have same dimensions, and the encoder for FG and BG layers is JPEG. For each 8×8 input pixel block the pre-processor receives a block of equal dimensions of binary data. By inspecting the binary mask, it labels the input block pix-

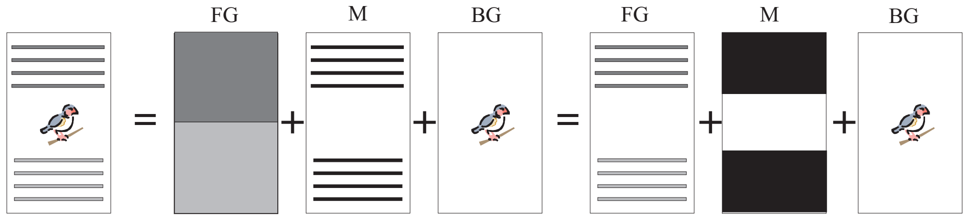

Figure 2. Illustration of typical segmentation strategies for MRC.

els as useful (U) or "don't care" (X). The X-marked pixels can be replaced by anything else since they are not going to be used for decompression. For example:

```
0 0 0 0 0 1 1 1     U U U U U X X X
0 0 0 0 0 1 1 1     U U U U U X X X
0 0 0 0 1 1 1 1     U U U U X X X X
0 0 0 0 1 1 1 1     U U U U X X X X
0 0 0 1 1 1 1 1     U U U X X X X X
0 0 1 1 1 1 1 1     U U X X X X X X
0 1 1 1 1 1 1 1     U X X X X X X X
1 1 1 1 1 1 1 1     X X X X X X X X
```

The block-wise pre-processor we used in this paper works as follows. If there are 64 X-marked pixels, the block is unused and we output a flat block whose pixels have the average of the previous block (because of JPEG's DC DPCM [6]). If there are no X-marked pixels, the input block is output untouched. If there is a mix of U- and X-marked pixels, we follow a multi-pass algorithm: in each pass, pixels marked "X" who have at least one U-marked horizontal or vertical neighbour is replaced by the average of those neighbours and marked "U" for the next pass. The process is continued until there are no X-marked pels left in the block. The aim of the algorithm is to replace the unused parts of a block with data that will produce a smooth block based on the existing data in the U-marked pels.

4. BLOCK THRESHOLDING

Given the preprocessor just described, our goal is to find the best mapping (input block to Mask block) which will optimize compression in a rate-distortion (RD) sense. Rate is given in bits necessary to encode all 3 layers and distortion is given in MSE for the reconstruted block (after decompressing and recombining the layers). For each block, for a fix preprocessor, and without scaling, there are 2^{64} possible Mask blocks. Even if we fix the compression schemes, we cannot possibly investigate all possibilities in search for the segmentation point which yields best RD trade-off. Because of that we devised a simple preliminary experiment: to divide the mask into 16 sets of 2×2 pixels and assign each pixel in the 2×2 set the same value. The image block is also subsampled by 2 and interpolated back using nearest neighbour, so that each 2×2 group in the block has the same intensity level. Now we have only 2^{16} possible arrangements for the mask block. Sample results are shown in Fig. 4 were we plot all RD points for each given input block.

A very curious issue arises when we examine a very simple segmentation strategy: thresholding. In this, for each block a threshold is selected and the mask is found as:

$$mask_n(i,j) = u(x_n(i,j) - t_n)$$

where $x_n(i,j)$ represent the pixels at the n-th block, t_n is the correspondent threshold, and $u(k)$ is the discrete step function. Since there are 64 pixels in a block, there are at most 64 different meaningful threshold values, whereby setting t_n to be less than the darkest pixel the Mask block can be made uniform (all samples imaged from one of the layers). We then mark the RD points with squares in Fig. 4 which correspond

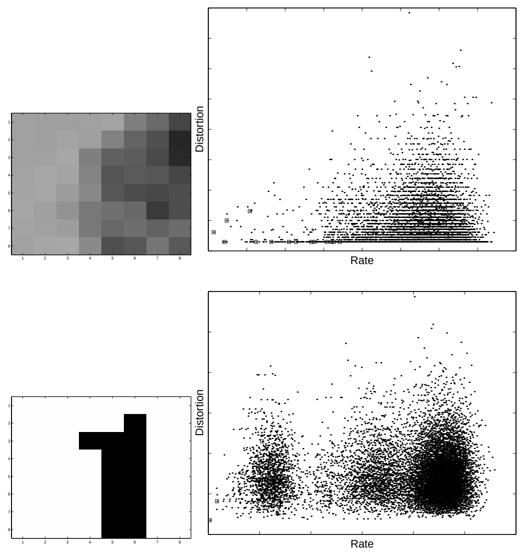

Figure 4. Sample blocks for the simplified experiment and the corresponding 64K RD plots. RD points obtained by block thresholding are marked.

to thresholding the reduced 4×4 block. It is easily seen that the mask obtained using thresholding yields among the best RD points. Although we just have shown two examples, all blocks we tested showed consistent results.

This result is not decisive but is significant. It tells us that if the results would hold for blocks of 8×8 pixels, then there is a simple way to find RD-efficient mask blocks. Note that we said RD-efficient and not optimal, since we cannot claim otherwise. Nevertheless, we pursue thresholding as a means of segmentation. The quest is to find a threshold value t_n for the n-th block. Moreover, we want to find the optimal value of t_n in an RD sense. In a block there are 64 pixels and therefore only up to 64 threshold values need to be tested. Given that the pre-processor algorithm is fixed and so are the compressors (including their parameters such as entropy coders and quantizers) every threshold value t_{kn} (k-th threshold value for the n-th block) yields a set of Mask, BG, and FG blocks, which are compressed at a total rate R_{kn} and are recombined resulting in a distortion D_{kn}. We define the cost function for a block as

$$J_n = R_{kn} + \lambda D_{kn} \ ,$$

where λ is a Lagrange multiplier which is common to all blocks. It is well known that in the optimal point all blocks operate at the same slope on the lower convex hull of the RD points. We test all t_{kn} in a block and select the one that minimizes J_n. Two examples are shown in Fig. 5, where it is shown: the input block, RD points, the RD point for mini-

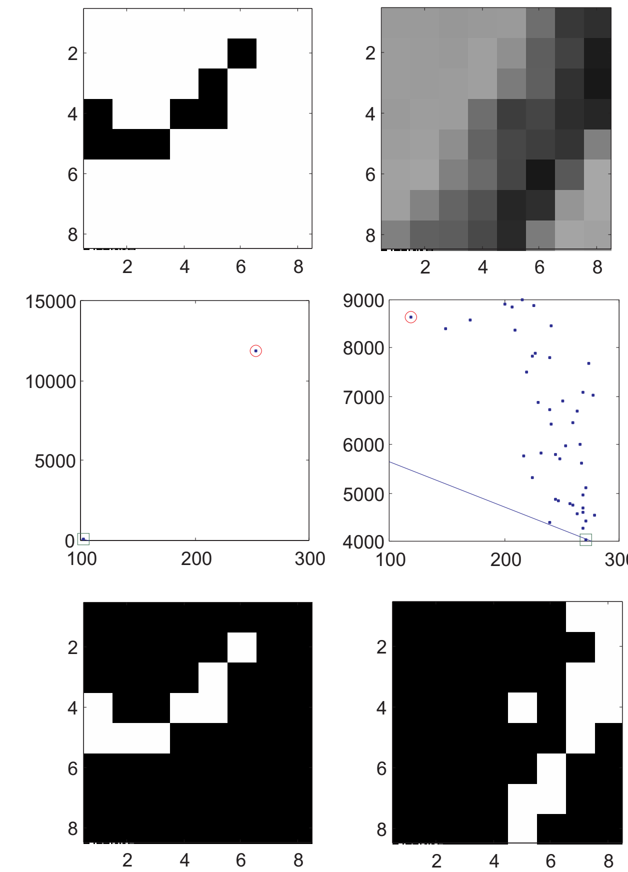

Figure 5. Sample blocks, RD plots for block thresholding and resulting Mask blocks. The operating slope is indicated along with the best RD point (□) and the RD point for a uniform mask (◯).

mum J_n, the RD point for a uniform mask (no segmentation), the line with slope $-1/\lambda$ which defines the best point, and the resulting Mask block. One example is a two-tone block wherein segmentation is clearly advantageous and obvious. The other example is extracted from a picture. Note that a change in the operating point (slope of the line) may result in completely different segmentation.

The main problem in our approach is to accurately compute the rate for a given block mask. The DC term in JPEG is encoded as a function of the DC of the previous block. That forced us to use a slightly greedy approach in which we decide the operating point for a block, calculate the masks, the pre-processed layers and the JPEG compressed data based on the previous layer blocks which were already set. In this sense, results are not globally optimal. The same reason (interblock dependency) affects largely the rate of the mask plane. The rate for the mask plane is by far the largest innacuracy of the algorithm. By looking at a single block we cannot compute how many bits some transition in that mask block would cost to the overall compression. Binary compression often works with transitions and run-lengths (or tokens in the case of JBIG-2). Our simple estimate, is better correlated with the one-dimensional MH algorithm [2] although still imprecise. We simply apply a fixed penalty in bits (e.g. 7 bits) for every horizontal transition of the Mask layer. Globally, this method is a good estimator, but the hope is that it should provide at least an approximation for the sake of the RD optimization.

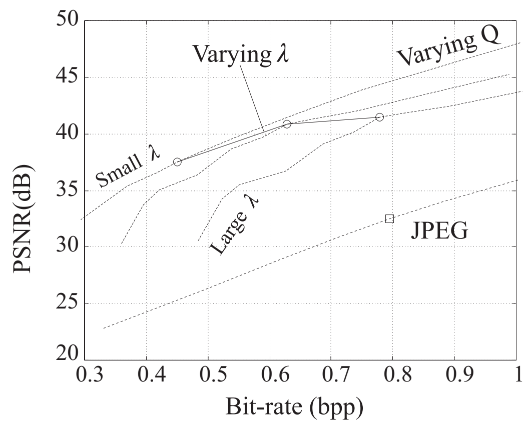

Figure 6. PSNR plots for MRC and JPEG.

PSNR plots are shown in Fig. 6 for the image "compound1" from JPEG 2000's test set. We compare MRC (using the proposed segmentation) and JPEG. The plots were obtained by scaling JPEG's example quantizer table (equal tables in both FG and BG planes) in order to vary the overall bit-rate. For the mask plane we used a simple fax MMR algorithm. The layers are then collected together using `tar` and `gzip`.

In Fig. 6, plots are shown for different values of λ, which is the operating slope of the segmenter. However, it is much more efficient to control the overall rate by modifying the compressors' parameters instead of making the Mask layer more or less complex. As λ decreases, the optimization is more biased towards minimizing rate in exchange of distortion. Nevertheless, as λ decreases the curves improve in Fig. 6. Two factors may contribute to this effect. Firstly, the innacurate calculation of the rate for the Mask layer makes it difficult to control the trade-off. The algorithm might chose to generate very complex masks since the penalty grows linearly with the number of transitions. As λ decreases, we noted that fewer portions of pictures are actually segmented. Secondly, the correlation of thresholding optimality and overall optimality may be weaker for more complex masks. In any case, results for the MRC scheme are far superior to JPEG's in terms of PSNR and can be shown to be superior to JPEG 2000's VM coder as well. A comparison of portions of an image encoded at about 0.4 bits-per-pixel (bpp) is shown in Fig. 7. It shows an MRC compressed image using: segmentation through block thresholding for very small λ; JPEG compression for both FG and BG layers; and CCITT's MMR for the Mask layer. It also shows the result using JPEG and the actual Mask plane used for MRC. Other images and comparisons can be shown but space limitations preclude the presentation of more results.

5. REMARKS

Optimized block thresholding seems to be an effective way to segment a compound document image for compression. If the complexity is not acceptable for a given application, one can use this procedure to guide and train non-RD-based segmentations strategies. Results so far are not decisive. Further efforts will be concentrated on better methods to estimate the rate achieved by compressing the Mask layer and investigating the reasons why minimization of rate is much more important than minimization of distortion, in the segmenta-

Figure 7. Top: portion of a reconstructed image after compression using MRC at 0.37bpp and 35.4dB PSNR. Middle: same for JPEG at 0.39bpp and 23.9dB PSNR. Bottom: mask used for segmentation.

tion algorithm. Further results and details will be presented in a forthcoming full paper [23].

REFERENCES

[1] R. Buckley, D. Venable and L. McIntyre, "New developments in color facsimile and internet fax," *Proc. of IS&T's Fifth Color Imaging Conference,* pp. 296–300, Scottsdale, AZ, Nov. 1997.

[2] ITU-T Rec. T.4, Standardization of group 3 facsimile apparatus for document transmission, July 1996.

[3] ITU-T Rec. T.6, Facsimile coding schemes and coding control functions for group 4 facsimile apparatus, November 1988.

[4] ITU-T Rec. T.82, Information technology - Coded representation of picture and audio information - Progressive bi-level image compression, March 1995.

[5] JBIG2 Working Draft WD14492, ISO/IEC JTC1/SC29 JBIG Committee, 21 August 1998,

[6] W. P. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard,* Van Nostrand-Reinhold, 1993.

[7] ISO/IEC JTC1/SC29 WG1, JPEG 2000 Committee, Working Draft 2.0, June 25, 1999.

[8] ISO/IEC JTC1/SC29 WG1, JPEG 2000 Committee, JPEG 2000 Verification Model (Technical Description), April 22, 1999.

[9] D. Huttenlocher and W. Rucklidge, "DigiPaper: a versatile color document image representation," in this Proceedings.

[10] L. Bottou, P. Haffner, P. Howard, P. Simard, Y. Bengio and Y. LeCun, "High quality document image compression using DjVu," *Journal of Electronic Imaging*, 7(3), pp. 410-425, July 1998.

[11] L. Bottou, P. Haffner, P. Howard and Y. LeCun, "Color documents on the Web with DjVu," in this Proc.

[12] M. Valliappan, B. Evans, D. Tompkins, F. Kossentini, "Lossy compression of stochastic halftones with JBIG2," in this Proceedings.

[13] D. Tompkins and F. Kossentini, "A fast segmentation algorithm for bi-Level image compression using JBIG2," in this Proceedings.

[14] Y. Yoo, Y. Kwon, A. Ortega, "Embedded image domain compression for simple images," *Proc. 32nd Asilomar Conf. on Sig. Sys. Comp.,* Pacific Grove, CA, Nov. 1998.

[15] R. L. de Queiroz, D. A. Florencio and R. W. Schafer , "Non-expansive pyramid for image coding using nonlinear filter banks," *IEEE Trans. on Image Processing,* Vol. 7, pp. 246-252, Feb. 1998.

[16] H. Cheng and C. Bouman, "Document compression based on multiscale segmentation," in this Proc.

[17] A. Said and A. Drukarev, "Simplified segmentation for compound image compression," in this Proceedings.

[18] D. Mukherjee, S. Acton, "Document page segmentation using multiscale clustering," in this Proceedings.

[19] Draft Recommendation T.44, Mixed Raster Content (MRC), ITU-T Study Group 8, Question 5, May 1997.

[20] R. de Queiroz, R. Buckley and M. Xu, "Mixed raster content (MRC) model for compound image compression," *Proc. EI'99, VCIP,* SPIE Vol. 3653, pp. 1106-1117, Feb. 1999.

[21] IETF RFC 2301. File Format for Internet Fax. March 1998. ftp://ftp.isi.edu/in-notes/rfc2301.txt.

[22] J. Huang, Y. Wang and E. Wong, "Check image compression using a layered coding method," *Journal of Electronic Imaging,* 7(3), pp. 426-442, July 1998.

[23] R. de Queiroz, T. Tran and Z. Fan, "Optimizing block-thresholding segmentation for multi-layer compression of compound images," submitted to *IEEE Trans. Image Proc,* 1999.