Optimizing Block-Thresholding Segmentation For Multi-Layer Compression Of Compound Images

Ricardo L. de Queiroz, *Senior Member, IEEE*, Zhigang Fan, *Member, IEEE*, and Trac D. Tran, *Member, IEEE*

Abstract— **Compound 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. We focus our attention on the mixed raster content (MRC) multi-layer approach for compound image compression. We study block thresholding as a mean to segment an image for MRC. An attempt is made to optimize the block threshold in a rate-distortion sense. Also, a fast algorithm is presented to approximate the optimized method. Extensive results are presented including rate-distortion curves, segmentation masks and reconstructed images, showing the excellent performance of the proposed algorithm.**

I. 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 and are frequently available as bitmaps. Compound documents are assumed here as images which contain a mix of textual, graphical, or pictorial contents. A single compression algorithm that simultaneously meets the requirements for both text and image compression has been elusive. As a rule, compression algorithms are developed with a particular image type, characteristic, and application in mind and 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. The human visual system, however, works differently for typical continuous-tone images, better masking high frequency errors. 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. There is now a focus on new standards to provide color facsimile services over the telephone network and the Internet [1].

When it comes to compound documents different compression algorithms may be applied to each of the regions of the document. This can be accomplished by segmenting the regions or by generating multiple image layers. The multilayer model will be the focus of this paper.

In Section II, we briefly review recent work on compound document compression. The mixed raster content (MRC) imaging model will be explained along with its segmentation framework. In Sec. III, we introduce methods to search for rate-distortion-efficient block segmentation through block thresholding. In Sec. IV we present a method to approximate the performance of the algorithms in Sec. III with lower computation. Experimental results are presented in Sec. V and, finally, the conclusions of this work are presented in Sec. VI.

R. de Queiroz is with Xerox Corp., e-mail queiroz@wrc.xerox.com; Z. Fan is with Xerox Corp., e-mail zfan@crt.xerox.com; T. Tran is with The Johns Hopkins Univ., e-mail ttran@ece.jhu.edu.

II. Mixed Raster Content for Document Compression

A. Document compression overview

Image compression has been very intensively studied and we cannot possibly reference adequately all the most noteworthy 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]. Multi-level 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. Apart from standards, halftones can be compressed with token-based techniques by descreening the halftone and encoding new halftone patterns as objects [9],[10]. Other algorithms do exist which can handle graphic bitmaps well [11] and also algorithms that perform well (not optimally) for both text/graphics and pictures using non-linear filter banks [12]. Once a region is identified it can be encoded with the proper algorithm. For region identification, segmentation algorithms may be employed. Examples are the work in [13] and the algorithms used in Digipaper [14] and DjVu [15],[16], which are already in commercial applications. Multiresolution segmentation was applied successfully in [17] for document compression, while [18] does the same using an approximate block-based object location. Multiscale clustering methods are also effective for segmentation [19]. We will present yet another segmentation algorithm based on block-thresholding in which the thresholds are found to be efficient in a rate-distortion sense.

B. Mixed Raster Content model

The mixed raster content (MRC) imaging model [1],[20],[21], allows for a multi-layer multi-resolution representation of a compound document, as illustrated in Fig. 1. 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 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

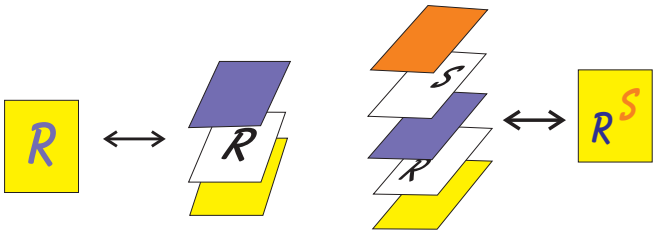

Fig. 1. Illustration of MRC imaging model. The basic 3-layer model can be extended by using a sequence of Mask+Foreground pairs. Less than 3 layers can be used by using default colors for Mask and/or foreground layers.

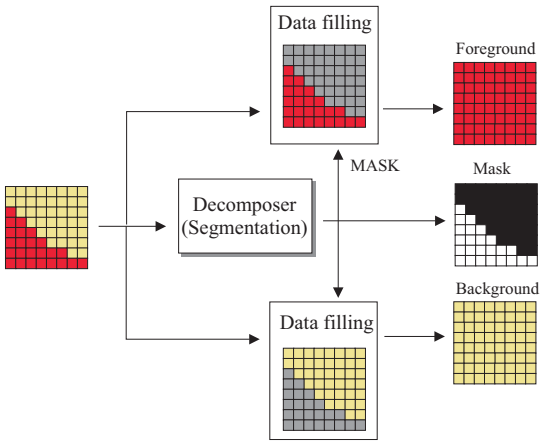

Fig. 2. Diagram of a segmenter.

```
0 0 0 0 0 1 1 1      U U U U U X X X      X X X X X U U U
0 0 0 0 0 1 1 1      U U U U U X X X      X X X X X U U U
0 0 0 0 1 1 1 1      U U U U X X X X      X X X X U U U U
0 0 0 0 1 1 1 1      U U U U X X X X      X X X X U U U U
0 0 0 1 1 1 1 1      U U U X X X X X      X X X U U U U U
0 0 1 1 1 1 1 1      U U X X X X X X      X X U U U U U U
0 1 1 1 1 1 1 1      U X X X X X X X      X U U U U U U U
1 1 1 1 1 1 1 1      X X X X X X X X      U U U U U U U U
     mask                  BG                   FG
```

Fig. 3. Example mask and corresponding block pixel labels.

BG layers when the mask pixel is 1 or 0, respectively, in that position. Thus, the FG layer is essentially poured through the Mask plane onto the BG layer. In effect, the imaging model allows for one, two, three or more layers (see Fig. 1). 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.

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. The compressed layers are then packaged in a format, such as TIFF-FX [22] or as an ITU-T MRC [20] data stream for delivery to the decoder. At the decoder, each plane is retrieved, decompressed, processed (which might include scaling) and the final image is recomposed 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 [22]. 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 such as DigiPaper and DjVu, whose owners built special segmenters for them, and also for check compression [23]. An analysis of the goals of the segmentation algorithm along with a better description of MRC can be found in [21].

C. Redundancy and data filling

In MRC, multiple planes represent a single image. As a result the amount of data to be encoded is multiplied. The reason for that is to allow the use of stock compressors without modifying them. Nevertheless, it comes for a price. In each layer there is redundant data, i.e. each layer (FG or BG) may contain "unused" pixels, as the pixels in that position will be selected from the other layer. Those unused pixels can be replaced by any color in order to enhance compression, since they do not affect reconstruction. This is the function of the data-filling processors illustrated in Fig. 2 for the case of a 3-layer MRC approach. Note that, given the filling algorithm, from Fig. 2 one can see that the segmenter function is to find a binary mask for a given input, from which the data-filling processors can derive the output layers based on the input image.

In this paper, we are interested in block based compression of FG and BG planes. Since blocks are compressed virtually independently, it is only necessary to explain the data filling for a single block. In general, the method can be used to fill larger blocks or perhaps the whole image. If the Mask for a particular block has a mix of 1's and 0's, it means that part of the block will be reproduced from the FG layer and part from the BG layer. The processor inspects the binary mask and labels the input gray-level pixels as either useful (U) or "don't care" (X). An example is shown in Fig. 3. At this point, the Mask block is ignored and the filling processor acts on the input gray level block based on the pixel labels. The X-marked pixels can be replaced by anything else since they are not going to be used for decompression. The filling algorithm 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 input 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 pixels 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 conveyed by the U-marked pixels. Other methods may be used as well with the same goal, however, this simple method has been shown to give reasonable results for JPEG-coded planes.

D. Notation, restrictions and problem statement

The image $x(i,j)$ is segmented generating the Mask $m(i,j)$. The data filling processors generate the image layers $L^{(BG)}(i,j)$ and $L^{(FG)}(i,j)$. The FG/BG planes are processed, compressed and combined into a single stream of bits which is transmitted to a receiver. The length of this stream is the rate R achieved by the coder. At the receiver, the streamed image data is then parsed and the FG/BG layers are decompressed into $\hat{L}^{(BG)}(i,j)$ and $\hat{L}^{(FG)}(i,j)$, while the Mask layer is supposed to be lossless encoded. The image is then recomposed as

$$\hat{x}(i,j) = m(i,j)\hat{L}^{(FG)}(i,j) + (1-m(i,j))\hat{L}^{(BG)}(i,j)\ . \quad (1)$$

The distortion incurred by the compression is

$$D = \sum_{ij} \left(x(i,j) - \hat{x}(i,j)\right)^2 . \quad (2)$$

In the case of block-based segmentation for block-based compression, the image is for simplicity divided into blocks of 8×8 pixels. For the n-th 8×8 input pixel block $x_n(i,j)$, the segmenter generates a binary mask block $m_n(i,j)$ with the same dimensions. The data-filling processor generates the layer blocks $L_n^{(FG)}(i,j)$ and $L_n^{(BG)}(i,j)$. The FG/BG blocks are compressed and decompressed as $\hat{L}_n^{(FG)}(i,j)$ and $\hat{L}_n^{(BG)}(i,j)$, from which the block can be reconstructed as

$$\hat{x}_n(i,j) = m_n(i,j)\hat{L}_n^{(FG)}(i,j) + (1-m_n(i,j))\hat{L}_n^{(BG)}(i,j)\ . \quad (3)$$

The distortion for this block is

$$D_n = \sum_{ij} \left(x_n(i,j) - \hat{x}_n(i,j)\right)^2 . \quad (4)$$

In this paper, the rate achieved for the block is not computed but estimated as

$$R_n = R_n^B + R_n^M + R_n^F, \quad (5)$$

where R_n^B, R_n^M and R_n^F are the estimated rates for compressing $L_n^{(BG)}(i,j)$, $m_n(i,j)$ and $L_n^{(FG)}(i,j)$, respectively. The reason for estimating as opposed to computing the actual rates is twofold. First, for the Mask layer block, it is very difficult (if not impossible) to accurately determine the amount of bits a single block will generate. Binary coders generally rely on run-lengths, or line-by-line differential positions, or even object properties. Hence, the contribution of a single block to the overall rate is not direct. In this paper, to circumvent this apparent deadlock, we estimate the mask rate by counting the number of horizontal transitions, to which we apply a fixed average penalty (e.g. 7 bits per transition). So, for a block with N_t transitions,

$$R_n^M = N_t * \text{penalty}. \quad (6)$$

Second, even though R_n^B and R_n^F can be precisely computed for coders such as JPEG, it is conceivable that all the compressed data may be further subject to entropy coders, which can significantly change the overall rate in high compression cases. In this paper's experiments, the compressed layer files are collected using the Unix programs `tar` and `gzip`. Thus, the bit-stream is subject to Lempel-Ziv compression. Even though $D = \sum_n D_n$, the rate is only approximated, i.e. $R \approx \sum_n R_n$.

The reason to use MRC for compression of compound images is because it is a standard. As a standard it is intended to employ standard compressors as well. JPEG 2000 [7],[8] has not yet been finalized. JPEG [6] is the current standard for lossy compression of continuous-tone material and we apply JPEG to compress non-binary planes. For simplicity, we decided to use MMR [3] to compress the binary plane, although any other binary coder may be applied. The reason for chosing MMR is simply because our local block rate estimator R_n^M better approximates the rate produced by MMR than the rate produced by either JBIG [4] or JBIG2 [5]. Because of the large number of variables involved, we want to keep the MRC model as simple as possible. Thus, we apply a combination of JPEG+MMR+JPEG to compress the FG, Mask, and BG blocks (planes), respectively. The extension of the concept to more than 3 planes is feasible. However, complexity will increase exponentially and we will leave such an extension to another opportunity. A further restriction is to preclude the scaling of the layers so that each 8×8 input block generates equally sized layer blocks.

As we discussed, the Mask and input blocks define the other layers (for a fixed filling algorithms and without spatial scaling). Our goal is to find the best mapping $x(i,j)$ to $m(i,j)$ which will optimize compression in a rate-distortion (RD) sense. We start by breaking the problem (image) into blocks and finding the best mapping $x_n(i,j)$ to $m_n(i,j)$. Then, for each input block, there are 2^{64} possible mask blocks. In order to simplify the search algorithm we also impose restrictions on the quantizer table [6] used to JPEG compress each block. Not only we use the same quantizer table $q(i,j)$ for both FG and BG planes, but we

also use a scaled version of JPEG's *default* table $q_d(i,j)$ as $q(i,j) = Qq_d(i,j)$. This simplification allows us to to control rate and distortion as a function of a single variable Q as opposed to 128 of them. Optimization of tables is possible but would be largely expensive and shadow the segmentation process. In reality, in an efficient implementation, one might scale one or both of the FG/BG planes and use different Q. We avoided this scenario intentionally to simplify our analysis. Despite all the simplifications and limitations of the proposed approach we will show that one can still achieve excellent compression performance.

III. Optimized block thresholding

For a given image $\{x(i,j)\}$, using a particular coding quantizer Q_c, R and D will depend on $\{x(i,j)\}$, Q_c and on the mask $\{m(i,j)\}$. The mask is found separately, so far, through segmentation methods that are independent from the given compression schemes. Here, we want to optimize the segmentation in an RD sense and, therefore, it is assumed a quantizer Q_d for the design phase, and an operating point λ to control the RD trade-off. Hence, $\{m(i,j)\}$ is a function of λ, Q_d and $\{x(i,j)\}$.

A. Thresholding

As the FG/BG planes will be encoded by blocks, we want to find each block Mask $m_n(i,j)$. The simplest model for a compound image is based on the histogram of the block pixels. Pictorial, background and text-edge areas should have histograms which are dense, flat, and bimodal, respectively. One simple approach is to find the bimodal blocks and to cluster the pixels around each of its modes. Whatever method is used to perform clustering or test bimodality, the pixels will be divided by some sort of threshold. In block thresholding the mask is found as

$$m_n(i,j) = u(t_n - x_n(i,j) - 1) \tag{7}$$

where t_n is the block's threshold and $u(k)$ is the discrete step function (equals 1 for $k \geq 0$ and 0 otherwise). In effect, pixels below the threshold are placed in the BG layer. 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 forces the Mask block to be uniform, i.e. all samples imaged from one of the layers[1]. To verify its RD performance, we cannot search all 2^{64} segmentations and compute each (R_n, D_n) pair. Because of that, we experimented on 4×4 blocks. To perform this experiment, the 8×8 input block was reduced to a 4×4 one, but all the layer blocks were interpolated (replicated) into 8×8 blocks before computing an RD point. Thus, there are only 2^{16} possible segmentation masks. Sample results are shown in Fig. 4 where we plot all RD points for each given input block. We, then, marked the RD points in Fig. 4 which correspond 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, tens of blocks were tested showing very similar results. If the results would hold for blocks of 8×8 pixels (without reducing them), then there is a simpler way to find RD-efficient Mask blocks.

[1]A 65th threshold value, t_n greater than the brightest pixel, was ignored as it achieves the same objective as the first.

B. Finding the "best" threshold with all the rest fixed

The quest is to find the best threshold value t_n in an RD sense, from which one finds the mask using (7). In a block there are 64 pixels and therefore only up to 64 threshold values need to be tested. If we sort the block pixels $x_n(i,j)$ into a sequence $p(k)$, for each $t_n = p(k)$, we evaluate

$$J_k = R_n(k) + \lambda D_n(k) \ , \tag{8}$$

where the index k denotes measurements for the k-th threshold tested. We recall that λ was defined as a control parameter and is used here as a Lagrange multiplier, and that the computation of R_n and D_n assumed a particular Q_d. Both λ and Q_d are fixed for all image blocks. This is so, because it is well known that, for optimality, blocks should operate at the same slope on their RD curves [24], and because baseline JPEG does not allow for changing quantizer tables within an image. We test all $p(k)$ in a block and select the index $k = k_o$ for the minimum J_k. Then, $m_n(i,j)$ is found using (7) for $t_n = p(k_o)$.

Two examples are shown in Fig. 5, where it is shown: the input block, RD points, the RD point for minimum J_k, 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.

C. Overall optimization: finding the block slope

In the previous section, we optimized the sequence $\{t_n\}$, block by block, for fixed external variables λ and Q_d. We want now to optimize these variables. As we discussed, in order to compute R and D, we have to decide upon Q_c. It is reasonable to assume $Q_d = Q_c = Q$ for a good RD estimation. In any case, R and D are functions of both λ and Q, i.e. $R(\lambda, Q)$ and $D(\lambda, Q)$. Given a budget R_b (or D_b), the goal is to minimize

$$\min_{\lambda,Q} D(\lambda,Q)\Big|_{R(\lambda,Q)\leq R_b} \quad or \quad \min_{\lambda,Q} R(\lambda,Q)\Big|_{D(\lambda,Q)\leq D_b} , \tag{9}$$

or, equivalently, we are interested in the lower convex hull (LCH) for a bounded RD region. The search of the 2D space can be very expensive. However, there are simplifying circumstances which may reduce the search.

We make the very reasonable assumptions that R and $1/D$ are monotonic increasing functions of both λ and $1/Q$. The higher λ is, the less importance R_n has in (8). Hence, the search will try to increase rate to reduce distortion which has larger weight. The higher Q, the coarser the quantization, thus the image incurs more distortion and is

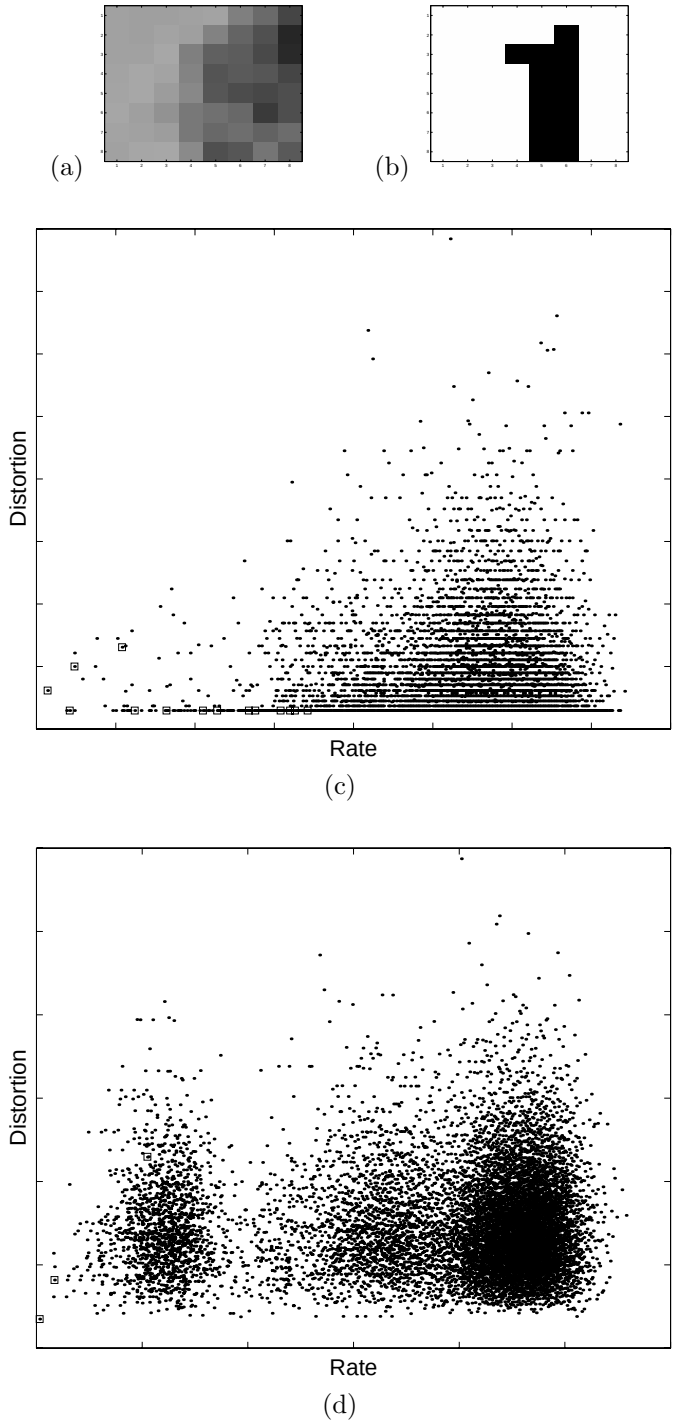

Fig. 4. Sample blocks for the simplified experiment and the corresponding 64K RD points. (a)(b) initial 8×8 pixel blocks before subsampling; (c)(d) RD plots corresponding to (a)(b). RD points obtained by block thresholding are marked.

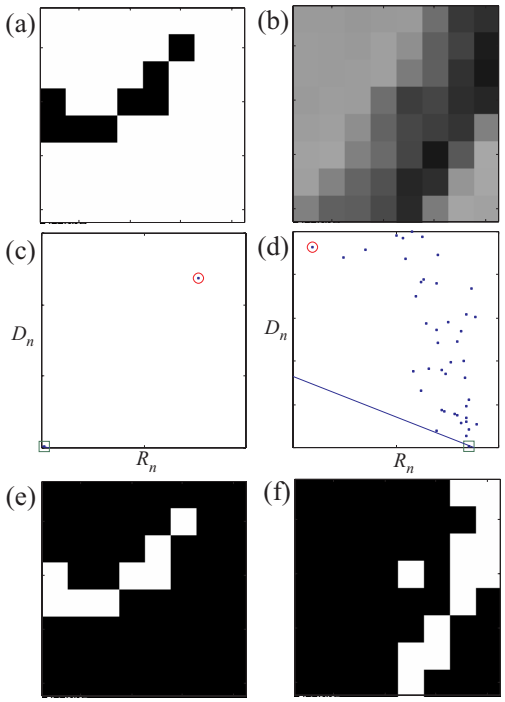

Fig. 5. Sample blocks, RD plots for block thresholding and resulting Mask blocks. The operating slope is indicated along with the best RD point (□) and the RD point for a uniform mask (◯).

more easily compressed (reducing rate). We make yet another assumption. Let us start from an operational point $P_0 = (\lambda_0, Q_0)$. If we vary either Q or λ, which variation would cause a larger RD trade-off? We assume that Q does. The quantizer affects the RD trade-off directly. By changing λ, with Q fixed, the rate is changed only by making the Mask layer more or less active, which has only a modest impact on D. As λ increases, we noted that more portions of pictures yielded non-uniform Mask blocks as shown in the examples in Fig. 6 for varying λ and $Q = 1$. In other words,

$$\frac{D(P_0+\delta_\lambda) - D(P_0)}{R(P_0+\delta_\lambda) - R(P_0)} \leq \frac{D(P_0+\delta_Q) - D(P_0)}{R(P_0+\delta_Q) - R(P_0)} \qquad (10)$$

where $\delta_\lambda = (\Delta\lambda, 0)$ and $\delta_Q = (0, \Delta Q)$ are the minimum changes in λ and Q in order to produce a change in the RD point.

These assumptions imply that the mapping of the plane (λ, Q) into the plane (R, D) only warps the image into the range and the topology is maintained. The mapping is illustrated in Fig. 7, where the the rectangular grid bounded by extreme values of λ and Q in Fig. 7(a) is mapped (distorted and stretched) into the irregular grid in Fig. 7(b) with the same topology (mapped lines do not cross). Thus, at most two of the boundary lines of the image in (λ, Q) are directly mapped to the LCH in the RD plane. However, λ has a limited effect on the RD point, while Q dominates the RD trade-off. As $Q \to \infty$, D diverges and R approaches its minimum (conceptually 0), regardless of λ. As $Q \to 0$, $D \to 0$ and R diverges, again, regardless of λ. Positive val-

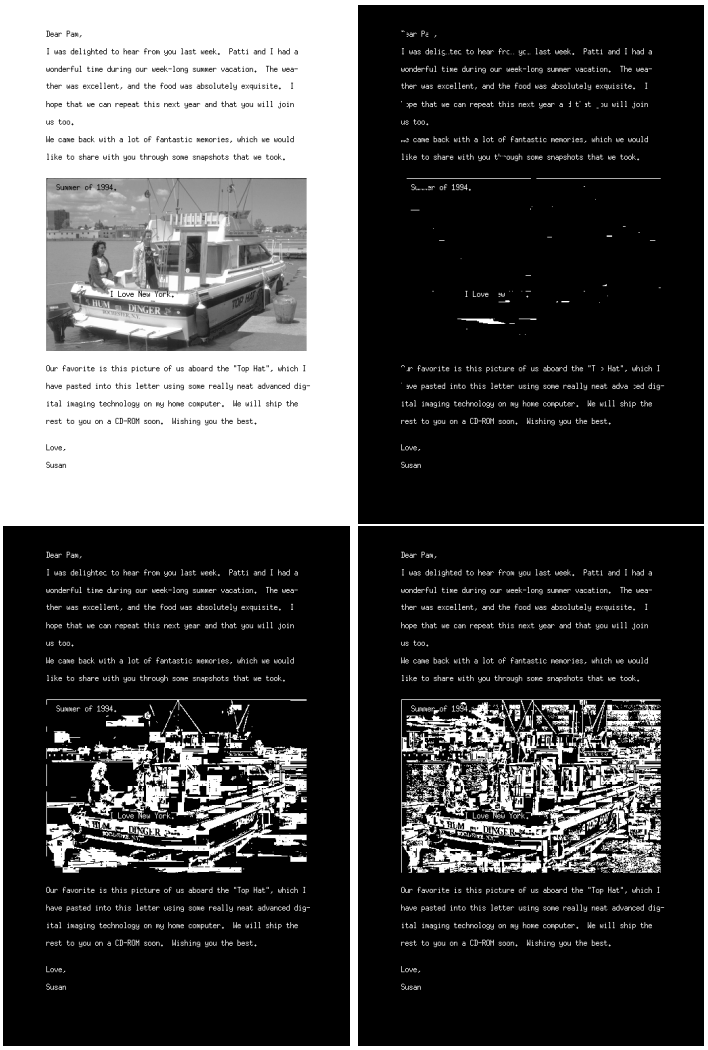

Fig. 6. Mask examples for image "compound1": top left, original image; top right, mask for minimum λ; bottom left, same for intermediary λ; bottom right: same for maximum λ.

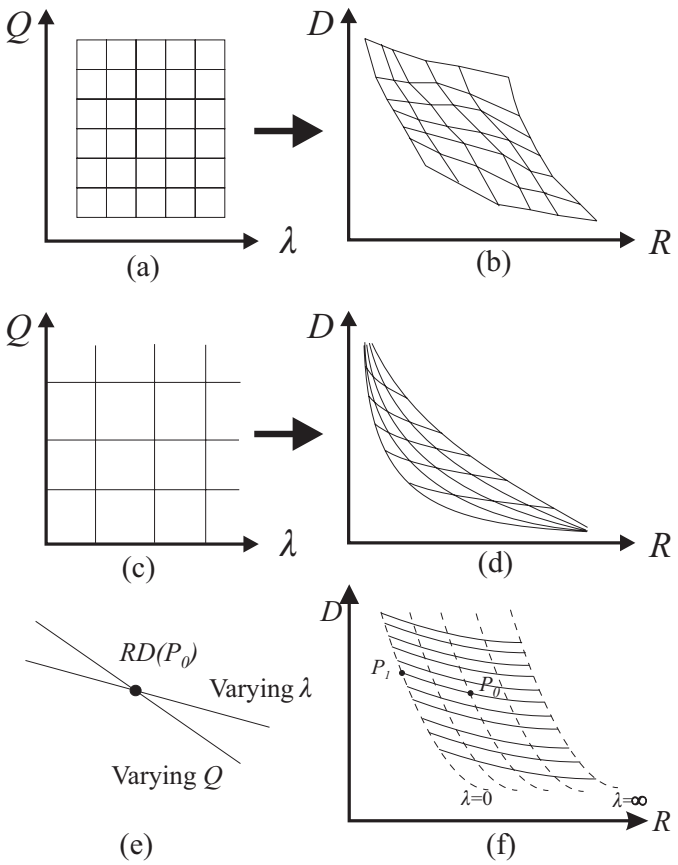

Fig. 7. Mapping $(\lambda, Q) \rightarrow (R, D)$. Image in (a) is warped into (b) because of the monotonicity assumptions. Because of Q domination, the mapping of the allowable image (c) is more like the range in (d). Since variations in Q dominate the RD trade-off (e), the typical topology of the RD range as function of (λ, Q) is as in (f). From the RD plot in (f), the LCH is obtainable by setting $\lambda = 0$.

ues of λ and Q, as illustrated in Fig. 7(c) are then mapped to a region as in Fig. 7(d). Thus, one of the axes is mapped to the LCH. Let $(R(P_0), D(P_0)) \equiv RD(P_0)$. Then, (10) is illustrated in Fig. 7(e). To confirm our speculations, tests were ran for different compound images for different rates and we verified the relations that we hypothesized (monotonicity and (10)). In Fig. 8 it is shown RD curves by varying Q and by varying λ (minimum, intermediary, and maximum values). Circles mark the points where $Q = 1$.

The implications of this result are that if we start from point P_0 in Fig. 7(f), where the solid curves are those obtained by varying λ while the dashed lines are obtained by varying Q, it is easy to see that one will "hit" the LCH by moving leftward from point P_0 until getting to point P_1, i.e. the Q axis is mapped the LCH. In other words, one operates at the LCH if $\lambda = 0$! This unusual result translates in minimizing all R_n without regard to D_n, in order to minimize D for some R or, equivalently, in order to minimize R for some D.

Hence, one can simplify the search and obtain an efficient

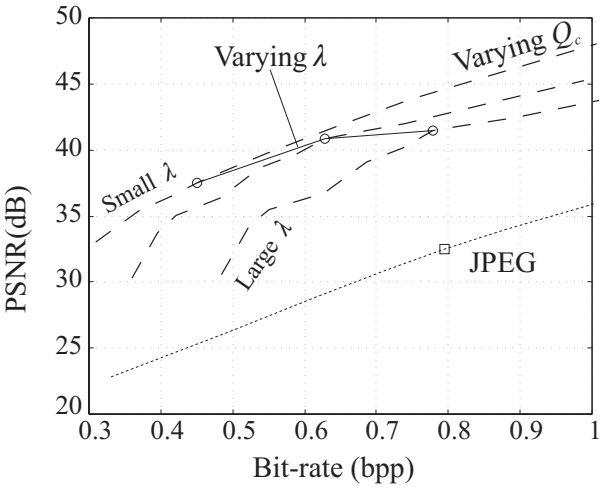

Fig. 8. PSNR plots for MRC and JPEG using image "compound1". Rate can be changed by either varying λ or Q. The points for unscaled quantizer tables are marked for all curves.

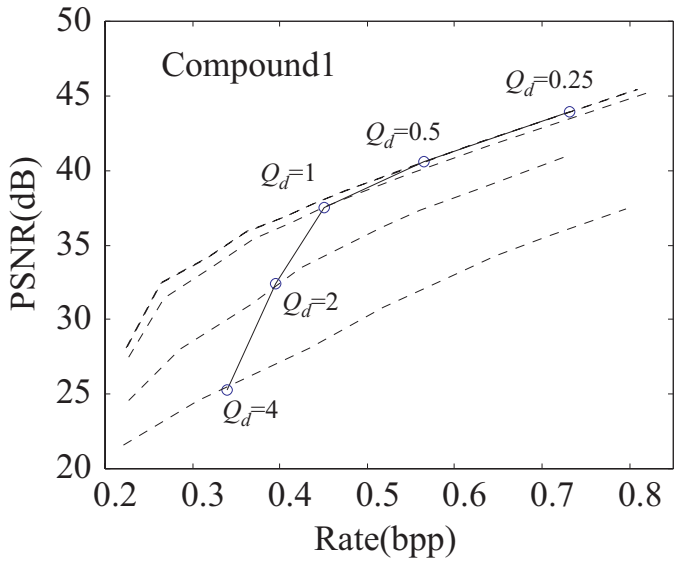

Fig. 9. Comparison between algorithms I (solid line) and II (dashed lines) for image compound1. Curves for $Q_d = 0.5$ and $Q_d = 0.25$ are superimposed.

operating point as:

Algorithm I

1. Select $Q_d = Q_c = Q$.
2. For every block, input $x_n(i,j)$ and find t_n which minimizes $J = R_n$.
3. Obtain mask $m_n(i,j)$ for each block using (7).
4. With resulting Mask layer in hand, compress FG/BG layers using quantizer Q.
5. Verify overall rate R (or D).
6. If R (or D) is not within parameters, adjust Q and go to step 2.

D. Practical optimization: finding the "design" quantizer

Due to practical reasons which we will discuss later, the otherwise logical concept of making $Q = Q_d = Q_c$ does not work very well for low rates. Large Q yields poor segmentation which somehow compromises the performance for moderate to high compression ratios. Note the erratic behaviour of curves in Fig. 8 for lower rates. For that reason we modify the algorithm to use a relatively small fixed value Q_d for segmentation and an adjustable Q_c for changing overall R and D.

Algorithm II

1. Select Q_d.
2. For every block, input $x_n(i,j)$ and find t_n which minimizes $J = R(t_n, Q_d)$.
3. Obtain mask $m_n(i,j)$ for each block using (7).
4. With resulting Mask layer in hand, compress FG/BG layers using scaling factor Q_c.
5. Verify overall rate R (or D).
6. If R (or D) is not within parameters, adjust Q_c and go to step 4.

PSNR plots are shown in Fig. 9 for the image "compound1" from JPEG 2000's test set[2]. We compare the two algorithms in the MRC context. The solid curve represents the points obtained with algorithm I for different Q. The dashed curves correspond to the algorithm II wherein Q_d is the same as Q for the first algorithm's RD curve. Note the poor performance of algorithm I for lower bit rates which is only able to catch up with second algorithm's performance for high bit-rates. Eventually, algorithm I becomes better than II, but only marginally. As we discussed, this effect is caused by the algorithm's inherent imprecision. Also, for high enough bit rates (low enough Q_d), the RD curves are virtually identical regardless of Q_d. This result was consistent with tests performed on other compound images. Thus, we conclude that algorithm II is a more robust substitute for algorithm I and any small enough Q_d should provide a "good-enough" performance.

Summarizing, the preferred algorithm (II) is the one where: $\lambda = 0$, Q_d is small (e.g. ≤ 1), and if necessary, rate is adjusted by changing Q_c. The search in 2D space is avoided, i.e. the Mask layer can be generated with only one point in the (λ, Q) plane. Hence, a single RD-efficient segmentation algorithm can be made to approach or meet the LCH virtually independent of the RD slope, i.e. independent of the actual RD targets.

Let C_J denote the complexity of JPEG compressing a block. It takes about 3 C_J to test one threshold. If, in average, k_t thresholds need to be tested per block, the segmenter complexity per block for Algorithm II is: $C = 3k_tC_J$. For predominantly binary images k_t is very small (minimum 2) while for pictures it can be up to 64, i.e. $6C_J \leq C \leq 192C_J$.

E. On block rate estimation for high compression

The optimization using a common Q is not very accurate for low rates due to several reasons of practical nature. Most of them are tied to the imprecision in estimating the block rates. 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 threshold for a block without looking at implications it may cause to the next block. In this sense, results are not globally optimal. The same reason (interblock dependency) affects largely R_n^M which is the largest inaccuracy of the algorithm, as we discussed. By looking at a single block we cannot compute how many bits are to be spent to encode it. Our simple estimate is better correlated with the one-dimensional MH algorithm, [2] although still imprecise. Also, the entropy coder used for the already compressed stream in largely innefective for high rates (redundancy has already been removed). As compression increases (rate decreases), compressed JPEG blocks become more repetitive and the entropy coder affects more the overall rate, making the block rate estimates less precise. As rate decreases, the Mask layer, whose compressed size does not change, becomes relatively more important within the compressed file. Therefore, innacuracies in estimating R_n^M have larger effect for low R. All those facts, among others, compound to limit the effectiveness of Algorithm I for low bit rates.

[2]The set of images tested in this paper is shown in Fig. 11

IV. Faster approximation

The goal of this section is to present a faster and efficient technique to obtain t_n. The largest complexity in the RD-optimized points comes from pre-processing and simulating compression of the FG/BG layer blocks in order to compute an estimation of the overall RD point. We want to avoid those computations as well as the block-filling operation. As we discussed, segmentation of bimodal blocks is essentially a method of finding a suitable threshold that would divide the block into two nearly-uniform regions. We can modify our cost function to reflect this property. We also want to add a penalty for Mask transitions. For non-bimodal blocks, the cost should be lower for uniform Masks than for non-uniform ones. Once the block is thresholded into two sets, measures of variance or entropy should be good estimators of how similar the set members are. We have chosen to use the variance measure not only because the variance is simpler to compute than the entropy, but it also serves as a good rate estimator. Intuitively, if a block has low variance, it should be compressed well with a small distortion.

For each block, we sort its pixels in $p(k)$ just like in the RD case. However we seek to minimize the following cost function

$$J = \alpha_1 V_{BG} + \alpha_2 V_{FG} + \alpha_3 N_t \tag{11}$$

where α_i are weighting factors, V_{BG} and V_{FG} are the variances of pixels in the BG and FG layer blocks. N_t is the number of horizontal transitions of the Mask block (the first column of the block uses as reference the last column of the previous Mask block, just like in the RD-optimized search case). For a given threshold, a Mask block $m_n(i,j)$ is obtained and we define two sets:

$$\begin{aligned} X_f &\equiv \{x_n(i,j) \mid m_n(i,j) = 1\} \\ X_b &\equiv \{x_n(i,j) \mid m_n(i,j) = 0\}. \end{aligned} \tag{12}$$

We define n_f and n_b as the number of pixels in the set X_f and X_b, respectively, where obviously $n_f + n_b = 64$. Then, variances are computed as:

$$\begin{aligned} V_{BG} &= \frac{\sum_{X_b} x_n(i,j)^2}{n_b} - \left(\frac{\sum_{X_b} x_n(i,j)}{n_b}\right)^2, \\ V_{FG} &= \frac{\sum_{X_f} x_n(i,j)^2}{n_f} - \left(\frac{\sum_{X_f} x_n(i,j)}{n_f}\right)^2 \end{aligned} \tag{13}$$

which can be efficiently implemented. Since thresholds are sorted, as we increment k, we will be effectively moving pixels from the set X_b to the set X_f. Thus, part of the computation does not change in each step. First we set the mask to be all zeros effectively placing the whole block in the background, which is equivalent to set t_0 to be the smallest of $x_n(i,j)$. Then, we set $k = 0$ and initialize the following variables

$$s_b = \sum_{ij} x_n(i,j) \ ; \ v_b = \sum_{ij} x_n^2(i,j) \ ; \ n_b = 64,$$

$$s_f = v_f = n_f = N_t = 0$$

We then compute

$$V_{BG} = \frac{v_b}{n_b} - \left(\frac{s_b}{n_b}\right)^2 \qquad V_{FG} = \frac{v_f}{n_f} - \left(\frac{s_f}{n_f}\right)^2$$

where, if n_f or n_b are 0, the corresponding variance is set to 0. Next, we compute (11). As we increase the threshold to the next pixel in the sorted list, we increment k, and the mask changes so that we need to recompute N_t. Some n_p pixels that form a set X_p are then moved from X_b to X_f. Hence, we have to compute $s_p = \sum_{X_p} x_n(i,j)$ and $v_p = \sum_{X_p} x_n^2(i,j)$. Then, we update our variables as

$$s_f = s_f + s_p \ ; \ v_f = v_f + v_p \ ; \ n_f = n_f + n_p$$

$$s_b = s_b - s_p \ ; \ v_b = v_b - v_p \ ; \ n_b = n_b - n_p$$

and recompute V_{BG}, V_{FG} and J_k. We repeat the process until X_b is empty. We test all 65 possibilities, i.e. from X_f empty to X_b empty in order to make the algorithm symmetric. We select $t = p(k)$ for $\min_k(J_k)$ and we compute the final mask using (7).

The overall computation is much more reasonable than the RD-optimized counterpart. The overall processing has a computational complexity C not superior to simply compressing the block with JPEG once, i.e. $C \approx C_J$.

As for the weights, without loss of generality we can normalize one of them (e.g. $\alpha_1 = 1$). The choice of weights is empirical, however there are some few guidelines that we can use for our advantage. We want to place the complex graphics and pictures in one of the planes and that can be done with $\alpha_2 \neq \alpha_1$. Also, variances are much larger numbers than the number of transitions (given that pixels range from 0 to let us say 255). Thus, α_3 should be a very large number. In our experiments, the values of $\alpha_2 = 5$ and $\alpha_3 = 200$ were efficient for segmentation of images containing sharp contrasting black-on-white text. Some masks are shown in Fig. 10 for the cases where $\alpha_1 = \alpha_2 = 1$, $\alpha_1 = 1$ and $\alpha_2 = 5$, and the case $\alpha_1 = 5$ and $\alpha_2 = 1$. Although all cases segment well the text, the placement of picture blocks differs. It might be advantageous for other reasons to use $\alpha_1 = 1$, $\alpha_2 = 5$ in order to place the picture along with the background, and we recommend those values.

V. Experimental results

We now compare segmentation methods within our MRC framework using the images shown in Fig. 11. The images range from purely graphics (graphics) to purely pictorial (baby), with two other mixed images with graphics (compound1) and pictorial (wine) dominance. Clearly, the more graphics the higher is MRC's advantage over a single coder such as JPEG. For an image such as "baby", our MRC approach has the disadvantage of encoding the overhead of two planes and is expected to be outperformed by JPEG. We compared the following segmenters: (i) $\lambda = 0$,$Q_d = 1$; (ii) its variance-based approximation; (iii) the segmenter from [17]; and (iv) the segmenter from [15]. We also compared the single-layer coders JPEG and JPEG 2000. For

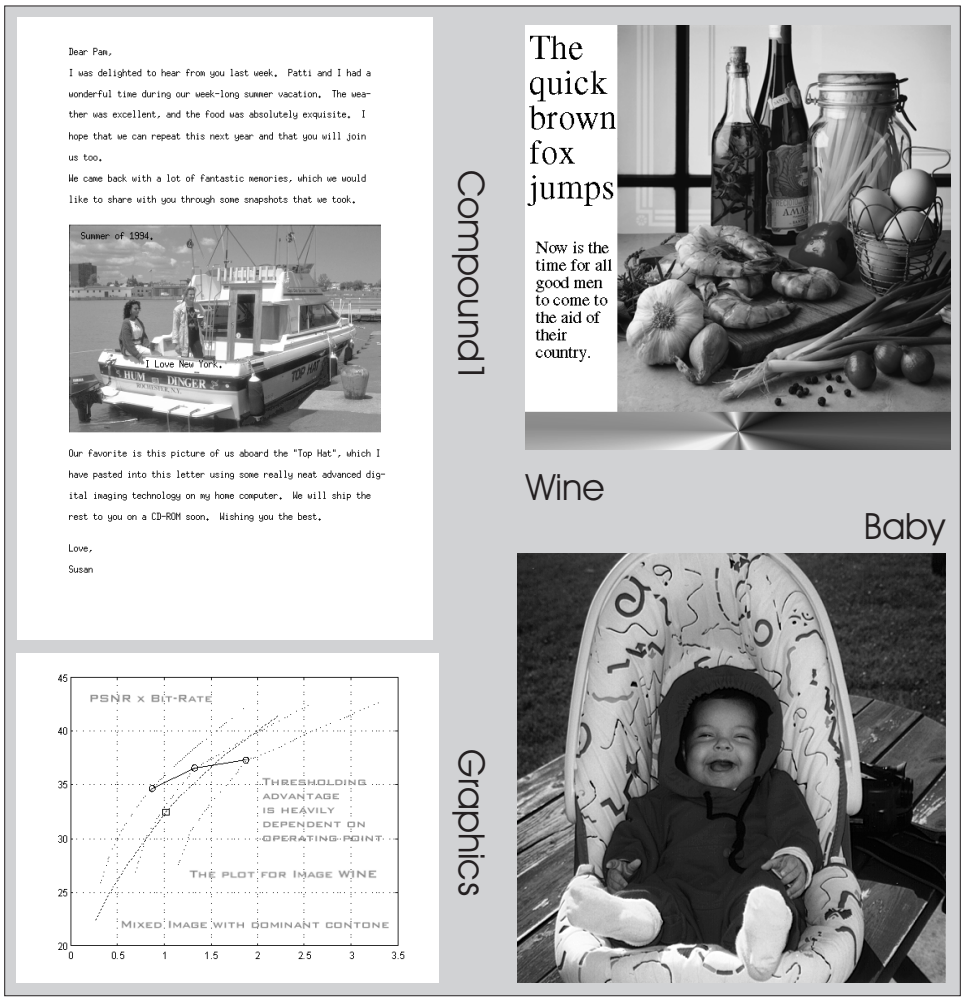

Fig. 11. Test images.

Fig. 10. Mask examples for image "compound1" and $\alpha_3 = 200$. From left to right, $\alpha_1 = \alpha_2 = 1$; $\alpha_1 = 1$ and $\alpha_2 = 5$; $\alpha_1 = 5$ and $\alpha_2 = 1$

the MRC approach, we computed RD curves by varying Q_c, which are shown in Fig. 12. The distortion measure chosen was PSNR and the plots present results in differential PSNR compared to JPEG, i.e. how many dB improvement would there be if we replace JPEG by the respective coder (MRC or JPEG 2000).

The PSNR difference against JPEG is extremely large for the graphics case since MRC quickly approaches the lossless state. The image compound1 is one of the best representatives of the target compound images. In this case, the PSNR difference is a staggering 12 dB over JPEG and several dB over JPEG 2000. The performance of the variance-based method is very close to that of the RD-based one, except for pictorial images. As the image becomes purely pictorial, the losses are about or below 1 dB for the RD-based segmentation compared to JPEG. We consider this small loss a very positive sign: even if by mistake a pictorial image is to be segmented, smart segmentation can minimize losses.

The other segmenters tested do not employ block-thresholding as we do. The one from [17] is RD optimized for a blockwise multilevel mask which was adapted by the authors for our MRC approach, by thresholding blocks in the bimodal class. It slightly outperforms our RD-optimized approach for low bit-rates for image compound1, while being underperformed everywhere else. That result is a combination of their efficient bimodal classification with our defficient segmentation for low bit-rates. The multiresolution clustering segmenter from [15] was designed for high resolution images and is not very precise in identifying small text. Since it is intended for WEB-based, very-

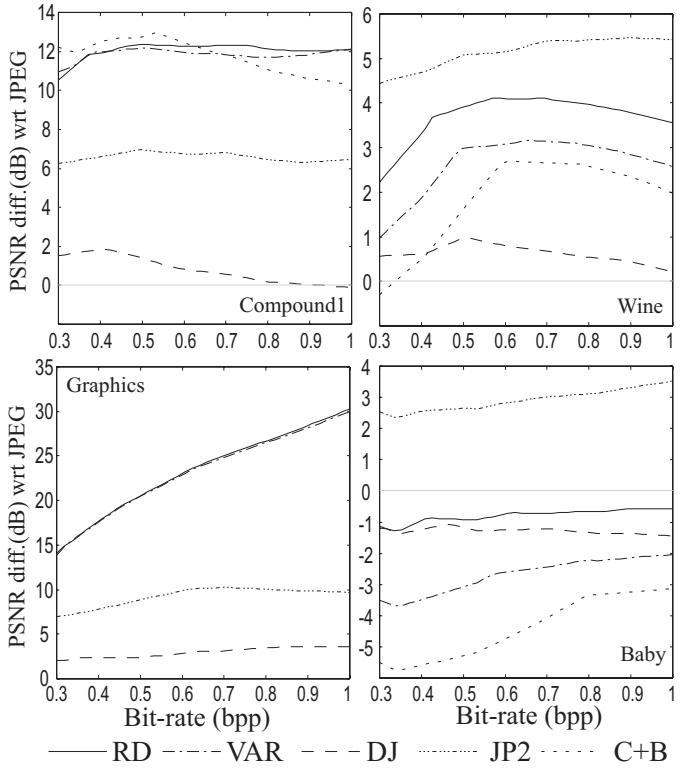

Fig. 12. PSNR difference plots compared to the PSNR achieved by JPEG compression. Solid line at 0 is JPEG reference performance. Line patterns are labelled, where: RD is algorithm II with $Q_d = 1$; VAR is the method in Sec. IV; DJ is MRC with segmenter from [15]; C+B is MRC with segmenter from [17], which is not available for image "graphics"; JP2 is JPEG 2000. Image name is noted for each plot set.

high-compression applications, this is commonly satisfactory, but unfortunately yields modest PSNR performance for our small test images within our MRC approach. Both segmentation approaches serve to illustrate the potential gains one can obtain by optimizing the segmenter, as we propose, for a given MRC setup, instead of using a generic segmentation.

Apart from MRC approaches, JPEG 2000 serves as an upper bound in single-layer performance. It is unfair, though, to compare our approach with JPEG 2000. One can also apply JPEG 2000 and JBIG2 coders to the MRC layers (as DjVu so succesfully does with similar technologies: IW44 and JB2 [15]) instead of simpler and cheaper technologies such as JPEG and MMR, which we used for tests.

A sample of reconstructed images for comparison are shown in Fig. 13. The superior quality of the MRC-coded image over the JPEG-coded one is easily noticeable in both pictorial and text areas.

VI. Conclusions

Optimized block thresholding seems to be an effective way to segment a compound document image for compression. Because of the different curve slopes and of the practical estimation problems at low bit-rates, block rate minimization along with high-quality quantization seem to be the basis for a robust threshold-based segmentation method. The proposed segmentation method is RD-optimized in the sense of leading to points approaching or meeting the LCH, given all constraints, but is robust enough to be virtually independent of the RD slope and of the RD targets, which are basically set after segmentation, during the compression stage. Therefore, the segmenter can be applied to an image independent of the layer compression settings, which simplifies implementation. The variance method is a fast alternative, which can be used as an example where the RD-optimized method can guide the tuning of other segmenter's parameters.

Overall, the trade-off is reasonable: there can be improvement in the order of tens of dB in graphics case, while only a small loss in the pictorial case. This is a result of the optimization algorithm. Furthermore, it was shown a sizable improvement of the proposed methods over other MRC-centric segmenters.

Further efforts will be concentrated on better methods to estimate the rate achieved by compressing the layers and on allowing more flexibility to the framework (e.g.layer scaling, different quantizers, different coders, etc.). Another research direction is to expand optimized thresholding for non-block-based frameworks in order to accommodate coders such as JPEG 2000.

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. Also ITU-T Rec. T.85, Application profile for Recommendation T.82 - Progressive bi-level image compression (JBIG coding scheme) for facsimile apparatus, August 1995.

[5] JBIG2 Working Draft WD14492, ISO/IECJTC1/SC29 JBIG Comm., 21 Aug. 1998, http://www.jpeg.org/public/jbigpt2.htm

[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] M. Valliappan, B. Evans, D. Tompkins, F. Kossentini, "Lossy compression of stochastic halftones with JBIG2," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.2, Kobe, Japan, Oct. 1999.

[10] D. Tompkins and F. Kossentini, "A fast segmentation algorithm for bi-Level image compression using JBIG2," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.4, Kobe, Japan, Oct. 1999.

[11] 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.

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

[13] H. T. Fung and K. J. Parker, "Segmentation of scanned documents for efficient compression," *Proc. SPIE: Visual Comm. and Image Processing,* Orlando, FL, Vol. 2727, pp. 701–712, 1996.

[14] D. Huttenlocher and W. Rucklidge, "DigiPaper: a versatile color document image representation," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.3, Kobe, Japan, Oct. 1999.

[15] 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.

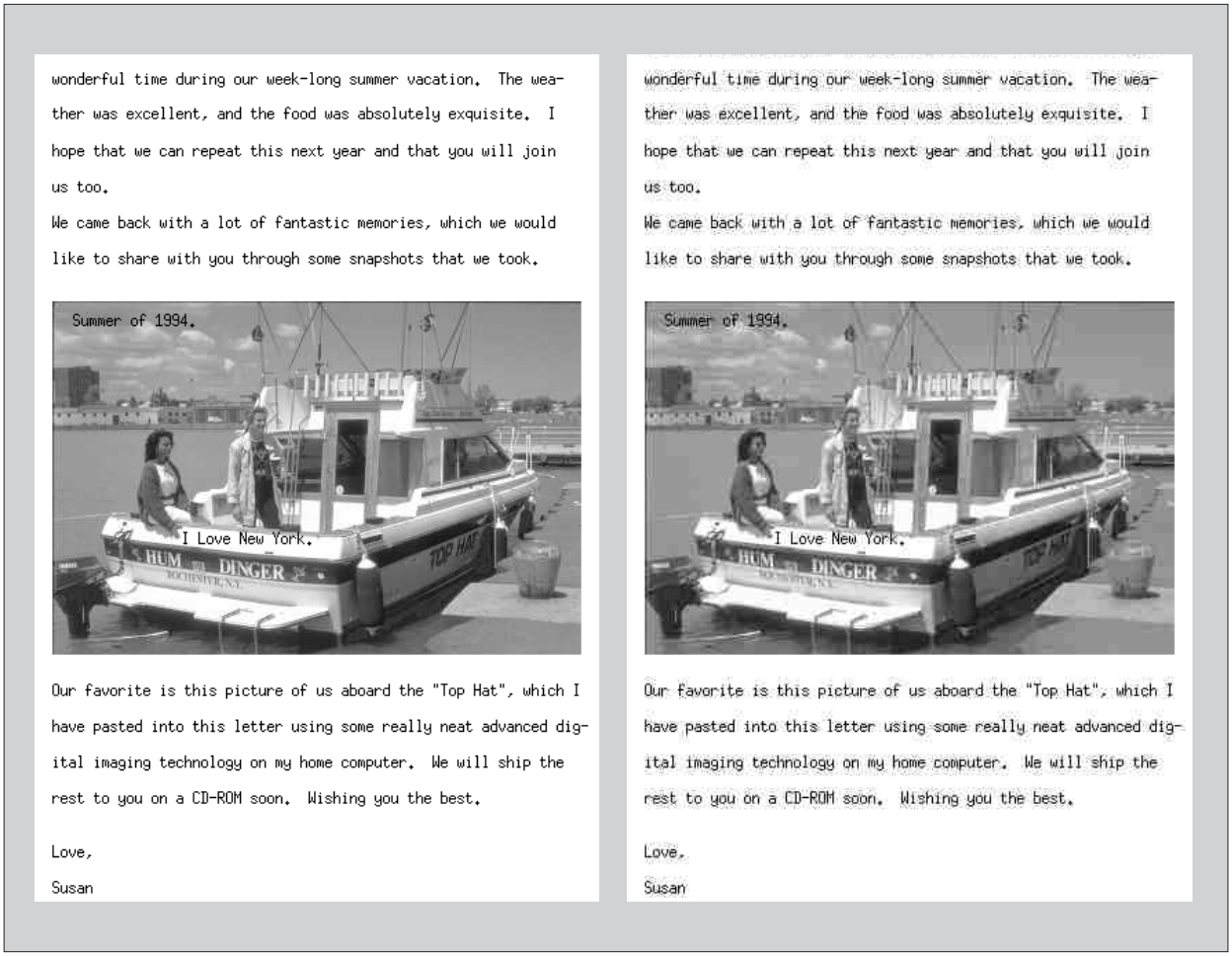

Fig. 13. Reconstructed images after compression at 0.45 bpp. Left: MRC compressed, minimum rate segmentation method (PSNR is 37.49 dB). Right: JPEG (PSNR is 25.33 dB).

[16] L. Bottou, P. Haffner, P. Howard and Y. LeCun, "Color documents on the Web with DjVu," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.7, Kobe, Japan, Oct. 1999.
[17] H. Cheng and C. Bouman, "Document compression based on multiscale segmentation," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.8, Kobe, Japan, Oct. 1999.
[18] A. Said and A. Drukarev, "Simplified segmentation for compound image compression," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.5, Kobe, Japan, Oct. 1999.
[19] D. Mukherjee, S. Acton, "Document page segmentation using multiscale clustering," *Proc. IEEE Intl. Conf. Image Proc.,* 25PS1.6, Kobe, Japan, Oct. 1999.
[20] Draft Recommendation T.44, Mixed Raster Content (MRC), ITU-T Study Group 8, Question 5, May 1997.
[21] 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.
[22] IETF RFC 2301. File Format for Internet Fax. L. McIntyre, S. Zilles, R. Buckley, D. Venable, G. Parsons, J. Rafferty. March 1998. ftp://ftp.isi.edu/in-notes/rfc2301.txt.
[23] 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.
[24] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression,* Hingham, MA: Kluwer Academic, 1992.