Document Compression Using H.264/AVC

Alexandre Zaghetto
Department of Computer Science
Universidade de Brasília - UnB
Brasília, Brasil
alexandre@cic.unb.br

Ricardo L. de Queiroz
Department of Electrical Engineering
Universidade de Brasília - UnB
Brasília, Brasil
queiroz@ieee.org

***Abstract*—It has been verified that H.264/AVC, the newest video compression standard, can also be used to encode still images. In many cases, it outperforms state-of-art coders such as JPEG2000. For compound documents, the gains over JPEG2000 are even more expressive. In this scenario, the contributions of the present paper are distributed over four document encoding methods that use the H.264/AVC as a basic functional element, namely: method 1, *Advanced Video Coding - Compound*, which, based on a macroblock content analysis, adaptively encodes text and image regions; method 2, *MRC Compression of Electronically Generated Documents using H.264/AVC-I and JBIG2*, which combines MRC (Mixed Raster Content) with H.264/AVC and JBIG2, and proposes a new data-filling technique based on the H.264/AVC intra prediction; method 3, *MRC Compression of Scanned Documents using H.264/AVC-I and JBIG2*, which offers pre/post-processing techniques as extensions of the MRC imaging model; and method 4, *Compression of Scanned Books using H.264/AVC*, which explores pattern recurrence to encode pages of scanned books. Many experiments were carried out in order to verify the efficiency of the proposed methods. Results showed objective and/or subjective gains over known approaches.**

***Keywords*-Image processing; compound document compression; Mixed Raster Document; H.264/AVC; scanned book compression.**

I. Introduction

The newest video coding standard, the H.264/AVC [1], has been well explained in the literature [2], [3], [4], [5]. Many papers have illustrated its performance showing many comparative results against coders such as MPEG-2. Apart from the factor-of-two improvement over other standards, there are a few unexpected advantages that come with the AVC package. H.264/AVC is a video compression standard and it was not conceived to be applied as a still image compression tool. Nevertheless, the many coding advances brought into H.264/AVC, not only set a new benchmark for video compression, but they also make it a formidable compressor for still images [6], [7], [8]. One of the components of these advances is the intraframe macroblock prediction method, which, combined with the context-adaptive binary arithmetic coding (CABAC), turns the H.264/AVC into a powerful still image compression engine. If we set our H.264/AVC implementation to work on a sole intraframe it will behave as a still image compressor. We refer to this coder as AVC-I. The big surprise is that it also outperforms previous state-of-art coders such as JPEG2000 [9]. Gains of the AVC-I over JPEG2000 are typically in the order of 0.25dB to 0.5dB in PSNR for pictorial images [6], [7], [8]. However, for compound images (mixed pictures and text) the PSNR gains are more substantial, even surpassing the mark of 3 dB improvement in some cases, as shown in Fig. 1.

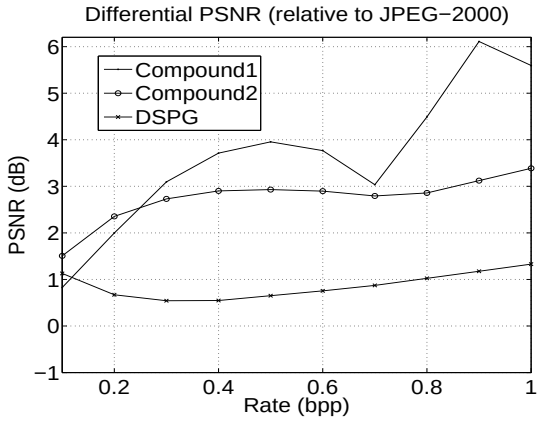


Figure 1. Differential PSNR (relative to JPEG2000) plots comparing AVC-I against JPEG2000 for "compound1", "compound2", and "DSPG" images. The compoundN images belong to the JPEG2000 test set. Because of the very large size of "compound2" we selected only a portion of it for tests.

Given the perspective of storing human intellectual production using digital media and the need for executing this task in an economic way, the present paper assembles the most advanced techniques on image coding, in order to propose new methods that enable efficient digital document compression. Since H.264/AVC has revealed itself to be a very efficient encoder also for still images, the contributions of the present paper are distributed over four methods that use this standard as a basic functional element.

II. Method 1: Advanced Video Coding - Compound

Compression algorithms are developed with a particular image type, characteristic and application in mind and no

Note: This work relates to a Doctoral Thesis.

single algorithm is best across all types of images or applications. When compressing text, it is important to preserve the edges and shapes of characters accurately to facilitate reading. The human visual system (HVS), however, works differently for typical continuous-tone images, better masking high-frequency errors [10].

Compound raster documents (mix of text and pictorial contents) have typically been compressed as a single image. However, different compression algorithms may be applied to each of the regions of the document. That is the way multiple-coder based algorithms work [11]. Instead of a multiple-code approach, the method here presented proposes a single-coder algorithm based on a modified version of the AVC-I that adjusts itself as an effort to encode text and pictorial regions differently.

A. Segmentation-driven rate allocation

A few authors dealt with compressing documents with one single coder. For example, Konstantinides and Tretter [12] used adaptive quantization within the JPEG extensions framework to compress compound (mixed) images. The idea is to use less aggressive quantizer steps for text regions in order to keep edges sharp, while being more forgiving to high frequency losses in pictures. Ramos and De Queiroz [13] used a single JPEG coder for the compression of mixed documents, stealing bits from background and images to give to text and sharp graphics edges.

In general, for RD (rate-distortion) optimized transform coding, the signal is divided into units x_i, each contributing to the overall bit-rate R by R_i bits, i.e. $R = \sum_i R_i$. Distortion is some function of the quantization error $\hat{x}_i - x_i$, where $\hat{x}_i$ is the reconstructed unit. By using a well behaved distortion function such as MSE, then $D = \sum_i D_i$ where D_i is the distortion for the i-th unit as $D_i = ||\hat{x}_i - x_i||$. RD optimization involves the minimization of a cost function $J = R + \lambda D$, where λ is a Lagrangian multiplier. Hence,

$$J = \sum_i R_i + \lambda ||\hat{x}_i - x_i||. \quad (1)$$

We imply a space varying meaning for distortion as opposed to adapting the algorithm, i.e. $D_i = ||\hat{x}_i - x_i||u_i$, where u_i is a distortion weighting factor specific for the i-th unit. In conventional human visual system weighted error measures, we can use a frequency-based weighting system in the transform domain. Since the HVS response is not completely understood and cannot be easily modeled, one can classify the image blocks into a discrete number of representative classes and devise HVS weights for each of the classes. For simplicity we assign weights u_i for the error norm rather than weights in the transform domain. Hence,

$$J = \sum_i R_i + \lambda_i ||\hat{x}_i - x_i||, \quad (2)$$

where $\lambda_i = u_i \lambda$.

(a) (b)

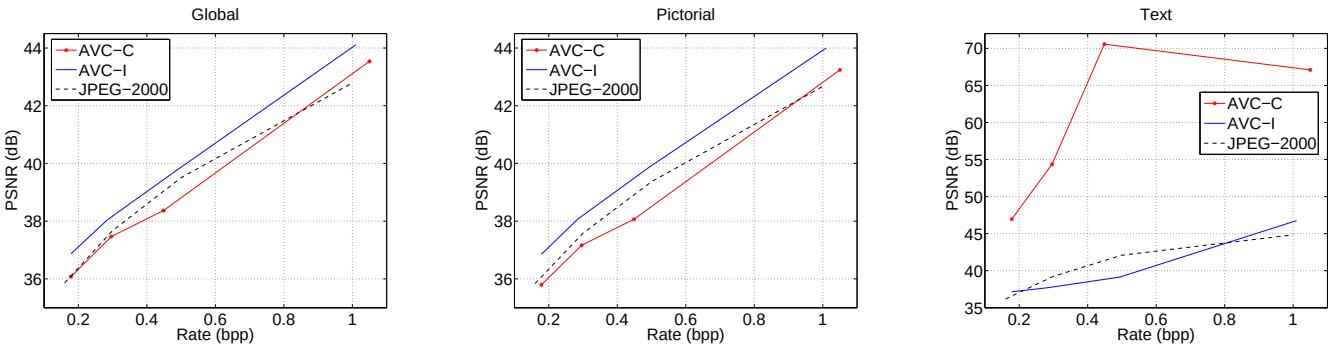
Figure 2. Classification algorithm: (a) original grayscale image (2592x1952 pels); (b) its coding map.

H.264/AVC allows for the change of the quantizer parameter Q_p at each macroblock. The adjustment of λ, or λ_i, in the quantization step, is translated into an adjustment of Q_p by an exponential equation. The quantizer adjustment is the most effective way to control rate and distortion. It controls more intensively the RD balance than for example using RD analysis to select the best macroblock prediction mode, or the size of the DCT. Therefore we can cut corners and adjust RD by modifying directly the quantizer parameter at each macroblock.

We propose to adapt the analysis on a macroblock by macroblock basis to be more economic in some blocks as opposed to others. First, we apply a region classification algorithm that will identify text and pictorial regions. This classification algorithm is derived from an edge detector and needs to identify edges belonging to text as opposed to textures. We assume that in these text regions the viewer would pay greater attention to edges. Since text segmentation is not the focus of this paper, any text segmentation algorithm, such as the one proposed by Fan [14], can be used.

The next step is to classify each macroblock (16x16 pixels block), denoted here as MB. The binary image containing the segmented text is analyzed and each MB is classified as type 0, 1 or 2 and a coding map is constructed. MBs of class 0 (pictorial regions) are composed exclusively by pixels marked as background. Class 1 MBs (text interior regions) are those composed exclusively by pixels marked as text. MBs which present a mixture of background and text interior, in any proportion, are considered as class 2 (text border MBs). Fig. 2(b) shows the coding map for the image shown in Fig. 2(a). To make it easier to visualize, MB classes 0, 1 and 2 were represented as white, black and gray, respectively.

The coding map is passed on to a modified version of AVC-I, which will adapt the value of Q_p for each MB, according to the class it belongs. The idea is to "transfer" quality of a MB class to another. Class 0 and 1 regions are encoded with a quantizer parameter Q_p, while class 2 regions are encoded with a quantizer parameter Q_{pText}, being $Q_{pText} < Q_p$. This means that more compression is applied where there is texture, and less compression is



(a) (b) (c)

Figure 3. Objective performance comparison between AVC-C, AVC-I and JPEG2000 for "DSPG" image: (a) global PSNR; (b) pictorial regions PSNR; (c) text regions PSNR. Notice that text regions quality can be considerably improved with little global quality loss.

applied to the text letter borders. This algorithm is referred to as H.264/AVC-INTRA Compound, or simply AVC-C.

B. The text vs. picture balance

We want to lower the quality of pictorial and text interior regions to improve text border regions until they become sufficiently clear, without compromising the quality of the whole document. Our Q_p and Q_{pText} selection algorithm works as follows:

ALGORITHM 1

1 The document is encoded using all possible (Q_p, Q_{pText}) combinations.
2 A bitrate R is chosen.
3 A bitrate variation δr around R is set.
4 Among all possible (Q_p, Q_{pText}) combinations, those which present bitrates inside the interval $R \pm \delta r$ are selected.
5 Among all selected combinations, the maximum PSNR value, $PSNRmax$, is determined.
6 A PSNR variation δq is set, and a minimum PSNR value, $PSNRmin = PSNRmax - \delta q$, is calculated.
7 Among all selected (Q_p, Q_{pText}) in step 4, those whose PSNR values are greater than $PSNRmin$ are chosen as candidates.
8 Select the candidate with the largest $d = Q_p - Q_{pText}$.

C. Results

The image shown in Fig. 2(a) was compressed by AVC-C, AVC-I and JPEG2000 with different parameters, and results are shown in Fig. 3.

D. Conclusions

AVC-I is very effective for compound documents because of its intraframe prediction mode. With AVC-C, for the same bitrate, it is possible to improve significantly the quality of text regions, with a user controlled quality loss to pictorial regions. Even though there is not an overall objective gain over AVC-I, the proposed AVC-C encodes text regions at higher quality. Furthermore, the proposed AVC-C encoder is compatible with AVC-I decoder.

III. METHOD 2: MRC COMPRESSION OF ELECTRONICALLY GENERATED DOCUMENTS USING H.264/AVC-I AND JBIG2

The Mixed Raster Content (MRC) ITU document compression standard (T.44) [15], [16], [17], [18], [19]. specifies a multi-layer representation of a compound document. In this section we present a basic 3-layer MRC codec that uses the H.264/AVC operating in intra mode (AVC-I) to encode BG/FG layers and JBIG2 [20] to encode the binary mask layer. The main objective is to show that MRC coding based on H.264/AVC and JBIG2, combined with appropriate layer segmentation and data-filling procedures, can yield better compression rates than schemes that use other state-of-the-art still image coders.

The basic 3-layer MRC model represents an image as two image layers (foreground or FG and background or BG) and a binary image layer (mask or M), which determines if a pixel belongs to BG or FG. Figure 4 illustrates the described model.

Once the original single-resolution image is decomposed into layers, each layer can be processed and compressed using different algorithms. BG and FG processing operations can include a resolution change and a data-filling procedure. The compression algorithm used for a given layer would be matched to the layer's content, allowing for improved compression while reducing distortion visibility [18], [21], [22]. The compressed layers are then packed and delivered to the decoder. At the decoder, each plane is retrieved, decompressed, processed and the image is composed using the MRC imaging model.

A. Layer segmentation

The first step of MRC compression is the layer segmentation algorithm [23]. This paper uses a variation of the

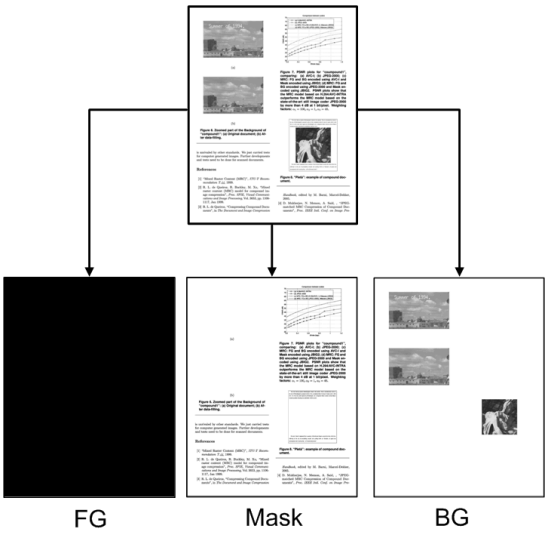


Figure 4. Illustration of MRC imaging model. The original document is represented using 3 layers: Foreground (FG), Background (BG) and mask.

block-thresholding segmentation algorithm proposed by De Queiroz [11], which will be described next.

As the FG/BG planes will be encoded by macroblocks (16×16 pixels block), we want to find each macroblock mask $m_n(i,j)$. In macroblock thresholding the mask is found as:

$$m_n(i,j) = u(t_n - x_n(i,j) - 1), \tag{3}$$

where t_n is the block's threshold, $x_n(i,j)$ represents the original image macroblock and $u(k)$ is the discrete step function (equals 1 for $k \geq 0$ and 0 otherwise).

In a macroblock there are 256 pixels and therefore up to 256 thresholds. For each macroblock, we select a set of $n \leq 256$ sorted thresholds, $t_n(k)$, and seek to minimize the following cost function:

$$J_n = \alpha_1 V_{BG} + \alpha_2 V_{FG} + \alpha_3 N t, \tag{4}$$

where α_i are weighting factors, V_{BG} and V_{FG} are the variances of pixels in the BG and FG layer macroblocks, respectively. N_t is the number of horizontal transitions of the mask block (the first column of the current block uses as reference the last column of the previous block). For a given threshold, a mask macroblock $m_n(i,j)$ is obtained and we define two sets,

$$\begin{aligned} X_{FG} &\equiv \{x_n(i,j) | m_n(i,j) = 0\} \\ X_{BG} &\equiv \{x_n(i,j) | m_n(i,j) = 1\}. \end{aligned} \tag{5}$$

We define n_{FG} and n_{BG} as the number of pixels in the set X_{FG} and X_{BG}, respectively, where obviously $n_{FG}+n_{FG} = 256$ and, then, variances are computed as,

$$\begin{aligned} V_{FG} &= \frac{\sum_{X_{FG}} x_n(i,j)^2}{n_{FG}} - \left(\frac{\sum_{X_{FG}} x_n(i,j)}{n_{FG}} \right)^2 \\ V_{BG} &= \frac{\sum_{X_{BG}} x_n(i,j)^2}{n_{BG}} - \left(\frac{\sum_{X_{BG}} x_n(i,j)}{n_{BG}} \right)^2. \end{aligned} \tag{6}$$

As for the weights, without loss of generality we can normalize one of them (e.g., $\alpha_2 = 1$). The choice of the other two weights is empirical.

B. Data-filling

Once the image is segmented there will be "don't care" regions on BG and FG layers. Pixels assigned to the BG will be marked as "don't care" on the FG, and vice-versa. These pixels can be replaced by anything to enhance compression [23], [24], [25]. There are many methods for the replacement (data-filling), such as the filter-based and iterative block-filling algorithms [23]. If the coder is known, there are ways to optimize the data-filling process. Since AVC-I is used to encode the FG and BG layers, this section proposes a method based on the intra prediction of H.264, which will be described next.

Let F and B represent the pixel positions where M indicates FG or BG, respectively. First, we compute averages as,

$$\begin{aligned} m_{BG} &= mean(x(i,j) | (i,j) \in B) \\ m_{FG} &= mean(x(i,j) | (i,j) \in F), \end{aligned} \tag{7}$$

where $x(i,j)$ represents the original image. Then we compute,

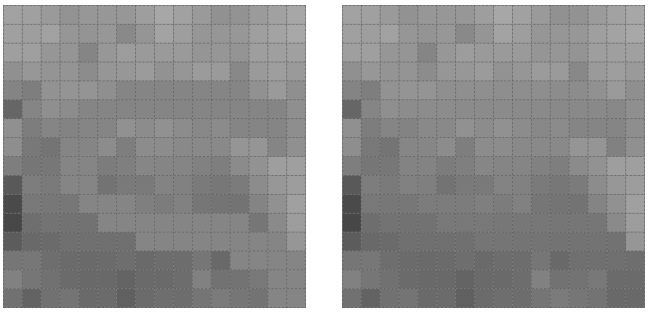
$$\begin{aligned} BG_0 &= M(i,j) \cdot x(i,j) + m_{BG} \cdot (1 - M(i,j)) \\ FG_0 &= (1 - M(i,j)) \cdot x(i,j) + m_{FG} \cdot (M(i,j)). \end{aligned} \tag{8}$$

Let BG_0' and FG_0' represent the AVC-I encoded/reconstructed versions of BG_0 and FG_0, respectively. If BG_{pred} and FG_{pred} are the predicted versions of BG_0' and FG_0' (using the 16×16 pixels intra prediction of H.264/AVC), then, after the the data-filling procedure, the processed BG and FG layers are defined by,

$$\begin{aligned} BG_{df} &= M(i,j) \cdot BG_0(i,j) + BG_{pred} \cdot (1 - M(i,j)) \\ FG_{df} &= (1 - M(i,j)) \cdot FG_0(i,j) + FG_{pred} \cdot (M(i,j)). \end{aligned} \tag{9}$$

Figures 5 (a) and (b) show an exemple of BG_0 and BG_{df}, respectively.

The last step is to encode the processed layers. In our method we use AVC-I to encode BG/FG layers and JBIG2 to encode the binary mask layer.



(a) (b)

Figure 5. AVC-matched data-filling: (a) BG_0; and (b) BG_{df}.

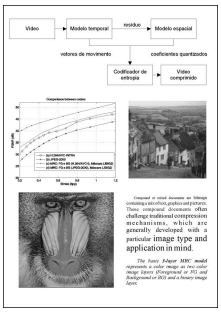


Figure 6. Example of compound document: "mixed1".

C. Results

The documents shown in Fig. 6 was compressed using AVC-I, JPEG2000, MRC-JPEG2000/JBIG2 and MRC-H.264/JBIG2. PSNR plots are shown in Fig. 7.

AVC-I single-coder seems to have an extra capacity of adapting itself to heterodox content [8]. In spite of this extra capacity of AVC-I, the multiple-coder MRC model proposed here offers results that outperform the AVC-I single-coder approach, surpassing the mark of at least 4 dB improvement at 1 bit/pixel. PSNR plots shown in Fig. 7 also demonstrate that the MRC model based on AVC-I outperforms the MRC model based on the state-of-the-art still image coder JPEG2000.

D. Conclusions

Results show that in most cases the MRC model achieves better performance than single coder approaches, such as JPEG2000 and AVC-I. Furthermore, using AVC-I to compress BG and FG yields better results than schemes based on JPEG2000. Without a doubt MRC schemes based on AVC-I set a new level of performance that is unrivaled by other standards. We just carried tests for electronically computer generated documents.

IV. Method 3: MRC Compression of Scanned Documents using H.264/AVC-I and JBIG2

MRC model is very efficient for representing sharp text and graphics onto a background. However, since the mask

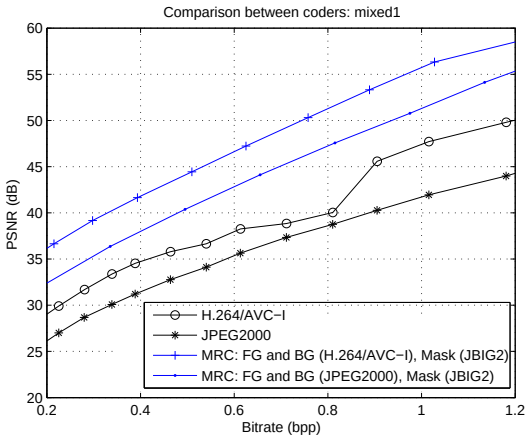


Figure 7. Objective performance comparison between coders: PSNR plots for "mixed1" document.

layer is binary, it is difficult to deal with scanned data and soft edges. The edge transitions do not fully belong to FG neither to BG, and cause some "halo" to the object edges using the MRC model. This section presents an algorithm that builds an edge sharpening map and iteratively parameterizes the original edge "softness" at the encoder. The generated map and the "softness" parameters are, then, used to reconstruct the original soft edges at the decoder. Regarding data-filling and compression, we propose a 3-layer MRC codec proposed in Sec. III. Experimental results are presented, showing that the method can yield 1.5 dB gains in PSNR, in the compression ranges of interest.

A. Edge sharpening and Softening

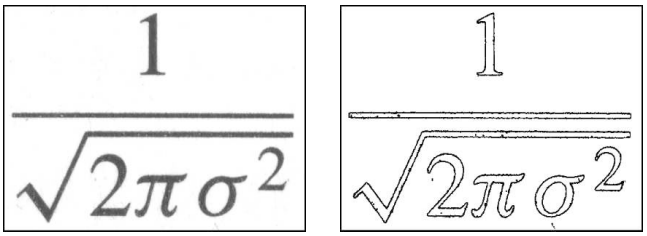
In order to remove the halo effect we are forced to change the data itself. The first step is to estimate where the halo will possibly occur. Our approach is to find transitions by applying the Sobel operator to the binary mask. The resulting transitions are morphologically dilated by a $d \times d - pixels$ structured element in order to mark a neighborhood. The image pixels that coincide with the dilated mask transitions are marked as possible processing targets. Let E be the set of pixel locations comprising this region.

The next step is to find pixels which are supposed to cause the halo effect. First we compute averages, as defined by Eg. 7. Then, we mark any pixel in the candidate region whose gray level is far apart from its layer average, i.e.,

$$C_{FG} = \begin{cases} 0 & : \quad |x(i,j) - m_{FG}| > \epsilon \mid (i,j) \in (F \cap E) \\ 1 & : \quad otherwise \end{cases}$$

$$C_{BG} = \begin{cases} 0 & : \quad |x(i,j) - m_{BG}| > \epsilon \mid (i,j) \in (B \cap E) \\ 1 & : \quad otherwise \end{cases} , \tag{10}$$

where ϵ is a tolerance value.



(a) (b)

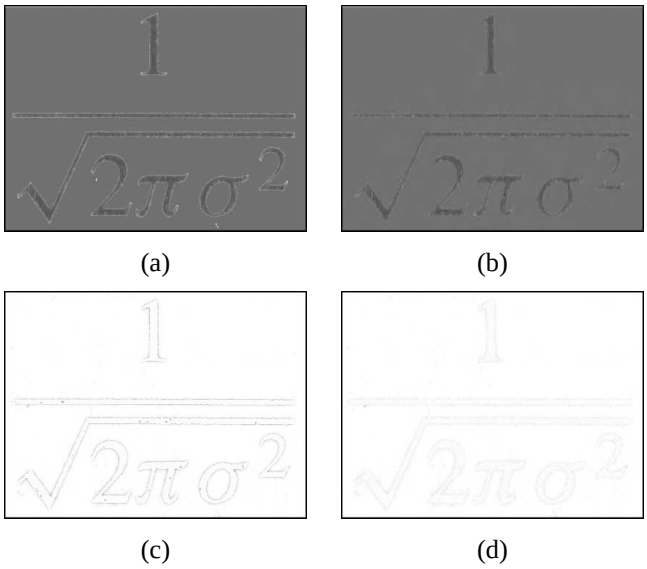
Figure 8. (a) Original scanned material; and (b) pixels to be changed C.

Next, we find the pixels marked by C_{BG} whose values are less than $(m_{FG} + \epsilon)$ and transfer them to the FG layer. This means that pixels in the candidate region $(B \cap E)$ that are distant from m_{BG} but are close enough to m_{FG} will not be pre/post-processed. Similarly, we find pixels marked by C_{FG} whose values are greater than $(m_{BG} - \epsilon)$ and transfer them to the BG layer, i.e., those in candidate region $(F \cap E)$ that are distant from m_{FG} but are close enough to m_{BG} will not be affected by pre/post-processing. The mask M, as well as the maps C_{FG} and C_{BG} are updated to accommodate the inter-layer pixel transfer.

For the image in Fig. 8 (a), and for $\epsilon = 16$ (out of 256 gray levels), the map of the pixels to be changed, i.e., $C = C_{FG} \cup C_{BG}$, is shown in Figs. 8 (b). In order to clean up the edge spots, we replace the values of the pixels in C_{FG} by m_{FG} and the values of the pixels in C_{BG} by m_{BG}. This is equivalent to changing the original image itself in order to make transitions sharper. Figure 9 shows FG/BG planes before and after halo processing. Note how the pre-processing improved the quality of the FG/BG planes. If we send the JBIG2 encoded C map as side information, we can blur only the pixels that belong to this map using an $h \times h$ Gaussian filter with standard deviation σ.

B. Estimation of Pre- and Post-Processing Parameters

Since the edges are sharpened to accommodate the mask, in order to reconstruct soft edges, we have to somehow estimate the transition of the image edges. The quest is to determine the best values of parameters ϵ, h and σ in a rate-distortion sense. For this, we determine the solution by minimizing the following cost function $J(\epsilon, h, \sigma) = D + \lambda R$, where λ is a weighting factor, D is the distortion incurred by the pre-processing, MRC encoding/decoding and post-processing algorithms, and R is the bitrate for compressing the document layers. Algorithm 2 is used to determine the best values for pre-processing parameters, ϵ, h and σ.



(a) (b) (c) (d)

Figure 9. Original (a) FG and (c) BG; processed (b) FG and (d) BG. Note the halo around the unprocessed FG/BG text. Pre-processing improves the quality of FG/BG planes.

ALGORITHM 2

1 $h \leftarrow h_0$;
2 $\sigma \leftarrow \sigma_0$;
3 **for** $\epsilon \leftarrow \epsilon_0$ **to** ϵ_k
4 **do** Generate map C using ϵ;
5 Sharpen the edges using C;
6 Run data-filling algorithm;
7 MRC encode/decode FG, BG and M;
8 Encode/decode C;
9 Filter edges using a Gaussian filter with parameters (h_0, σ_0);
10 Calculate and store cost $J(\epsilon, h_0, \sigma_0)$;
11 Find ϵ that results in the minimum cost J and make it ϵ_{best};
12 Generate map C_{best} using ϵ_{best};
13 Sharpen the edges using C_{best};
14 Run data-filling algorithm;
15 MRC encode/decode FG, BG and M;
16 Encode/decode C_{best};
17 **for** $h \leftarrow h_0$ **to** h_i
18 **do for** $\sigma \leftarrow \sigma_0$ **to** σ_j
19 **do** Filter edges using a Gaussian filter with parameters (h, σ);
20 Calculate distortion D;
21 Find (h, σ) pair that minimizes D and make it (h_{best}, σ_{best});

Since FG and BG are encoded using AVC-I, a design quantizer parameter, QP_D, needs to be set for the MRC encoder in steps 7 and 15. The H.264/AVC quantizer parameter, QP, may vary from 0 to 51. Since we are interested

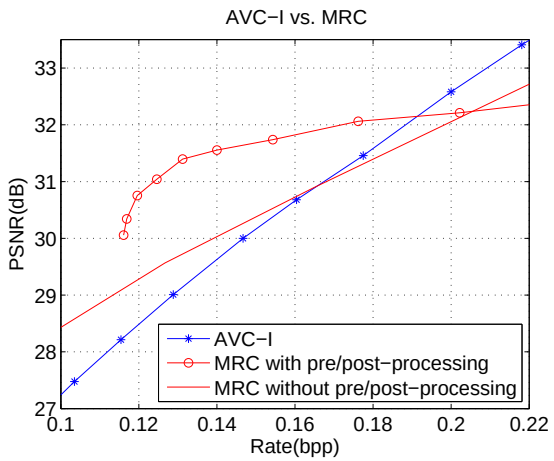


Figure 10. AVC-I and MRC performance for a scanned text/graphics image. The proposed MRC with pre/post-processing (ϵ_{best}=28, h_{best}=25, σ_{best}=1.5) outperforms MRC without pre/post-processing by more than 1.5 dB and AVC-I by more than 2 dB, at 0.13 bpp.

in very low bitrates, a high QP_D (above 30) is suggested.

MRC imaging model also allows resolution change of FG/BG layers. Resize factor, S, of 1, $1/2$ and $1/4$ were used. The performance of the codec was evaluated for those values, as described by Algorithm 3.

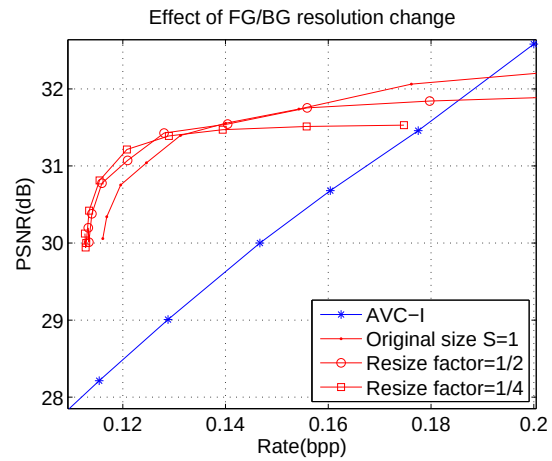
ALGORITHM 3

for $S \leftarrow 1, 1/2$ and $1/4$
  do for $QP \leftarrow QP_0$ To QP_k
    do Generate rate-distortion points (R, D);
Sort (R, D) points along R, in ascending order;
$N \leftarrow$ number of (R,D) points;
for $i \leftarrow 1$ To N
  do if $D_i < D_{i-1}$
    then Select (R_i, D_i) point;

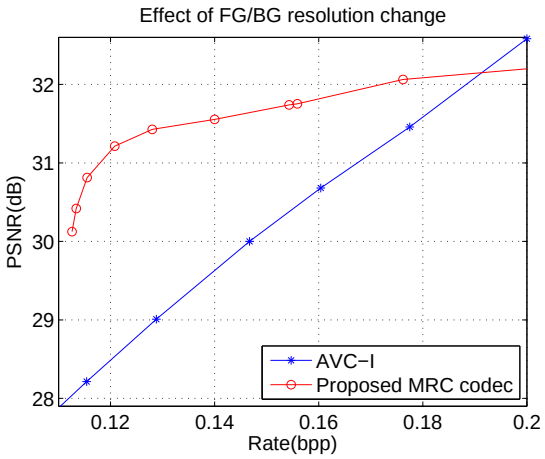
C. Results

Figure 10 shows a PSNR plot comparing AVC-I and MRC performance with and without the pre/post-processing steps for one typical document. No FG/BG resolution change was applied ($S = 1$). MRC with pre/post-processing outperforms MRC without pre/post-processing by more than 1.5 dB and AVC-I by more than 2.0 dB, at 0.13 bpp. However, there is only a short interval wherein there are gains using the proposed scheme (from 0.12 bpp to 0.18 bpp, when compared to AVC-I). This occurs because the method is bounded in PSNR due to the edge sharpening/softening procedure. Also, the achieved bitrate has a lower bound because of the number of bits needed to encode C and M losslessly.

Layer downsampling procedure has been included as an effort to improve the compression range of interest. The rate-distortion effect of this procedure is shown in Fig. 11 (a). Figure 11 (b) shows the resulting rate-distortion performance



(a)



(b)

Figure 11. Effect of FG/BG resolution change: (a) MRC codec performance evaluation for resize factors S = 1, 1/2 and 1/4; (b) PSNR performance after running Algorithm 2 for the curves shown in Fig. 11 (a).

after running Algorithm 3 for the curves shown in Fig. 11 (a). Note how the bitrate lower bounds are shifted left and the PSNR upper bounds are shifted upwards for bitrates closer to 0.12 bpp.

D. Conclusions

We have proposed a method that counter balances the effects of soft edges in MRC compression of scanned documents.

Although the proposed method is meant to deliver a reconstructed image which should be as similar as possible to the original scanned one, in some particular applications the post-processing procedure may be turned off. Subjectively, sharpened (pre-processed only) documents may present better quality than re-softened (post-processed) ones. Hence, the decoder might chose between softening or not the text.

Furthermore, regular MRC decoders would ignore the C map and decode the sharper version.

The proposed approach improves the reconstruction fidelity in the MRC compression of scanned documents. In effect, our results have shown that the method enables competitive MRC compression of soft-edge document images.

V. Method 4: Compression of Scanned Books using H.264/AVC

This section shows how H.264/AVC can also be used as an efficient compressor for scanned books. In such documents, the pages are tipycally individually compressed by some continuous-tone image compression algorithm, such as JPEG [26], JPEG2000 or AVC-I. Considering the recurrence of text patterns across pages, or across different areas of the same page, the main idea here presented is to use the many improvements brought into H.264/AVC to enable a hybrid approximate pattern matching/transform-based scanned book encoder.

It is important to place our coder within the proper scenario. First, the use of one single coder is proposed, thus avoiding the inconvenience of handling multiple coders, as in the MRC imaging model. Second, the encoded document should be decoded by a codec that common users have access to. Third, the codec must output high quality reconstructed versions of scanned documents. This is specially important when rare books of historical value must be digitally stored. In this case, one must guaranty a reconstructed version of the document which is as close as possible to the original one.

A. The Proposed Method

Giving that the book will be compressed using H.264/AVC, the proposed encoding method organizes the scanned pages in such a way the interframe prediction may find on previously encoded macroblocks (16×16 pixels blocks) text patterns that are similar to those on the macroblock currently being encoded. Figure 12 illustrates the proposed page processing algorithm.

First, each scanned $H \times W$ pixels page is segmented into four $H/2 \times W/2$ pixels sub-pages. Then, these sub-pages are used to build a video sequence. The only reason page segmentation should be used is that in some cases similar text patterns are more likely to be found on the same page rather than on different pages. If the text style is constant throughout the whole book, each page may be converted into one single frame and segmentation may be skipped. The final step is to compress the resulting video using H.264/AVC.

The basic idea of the interframe prediction is to exploit similarities between video frames in order to reduce the amount of information to be encoded. Based on previously encoded blocks, it first constructs a prediction of the current frame and then creates a residual frame by subtracting the prediction from the current frame. In H.264/AVC, the luma component of each current macroblock is predicted as one 16×16 partition, two 16×8, two 8×16 or four 8×8 macroblock partitions. In case partitions with 8×8 pixels are chosen, the 8×8 sub-macroblocks may be further partitioned in one 8×8 partition, two 8×4, two 4×8 or four 4×4 sub-macroblock partitions. The prediction of each luma block is constructed by displacing an area of the reference frame, determined by a motion vector and a reference frame index.

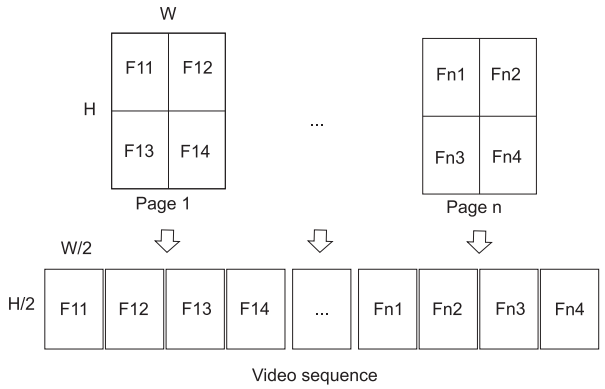


Figure 12. Proposed page processing algorithm.

Figure 13 illustrates the effect of using interframe prediction as an approximate pattern matching algorithm. Figures 13 (a) and (b) show examples of a reference and a current text area, respectively. Figures 13 (c), (e) and (g) represent the predictions of the current text using 16×16, 8×8 and 4×4 block partitions. Figures 13 (d), (f) and (h) are the corresponding residual data. Notice that the 4×4 prediction generates a lower-energy residual, when compared with the 16×16 and 8×8 prediction. However, smaller partitions require a larger number of bits to encode the motion vectors. This implies that partition size selection has a major impact on compression performance.

Examples shown in Fig. 13 suggest that previously encoded text areas (reference frames) can be seen as a dictionary used by the pattern matching (interframe prediction) algorithm. The dictionary is updated in parallel with the encoding process, since new reference frames become constantly available. Furthermore, a rate-distortion optimization algorithm is used to estimate which intra/inter modes combination should be applied.

Once the residual data is available, H.264/AVC utilizes an integer transform with similar properties as the DCT (Discrete Cosine Transform) and the resulting transformed coefficients are quantized and entropically encoded using CABAC.

In the next section we show results that demonstrate the efficiency of the proposed method.

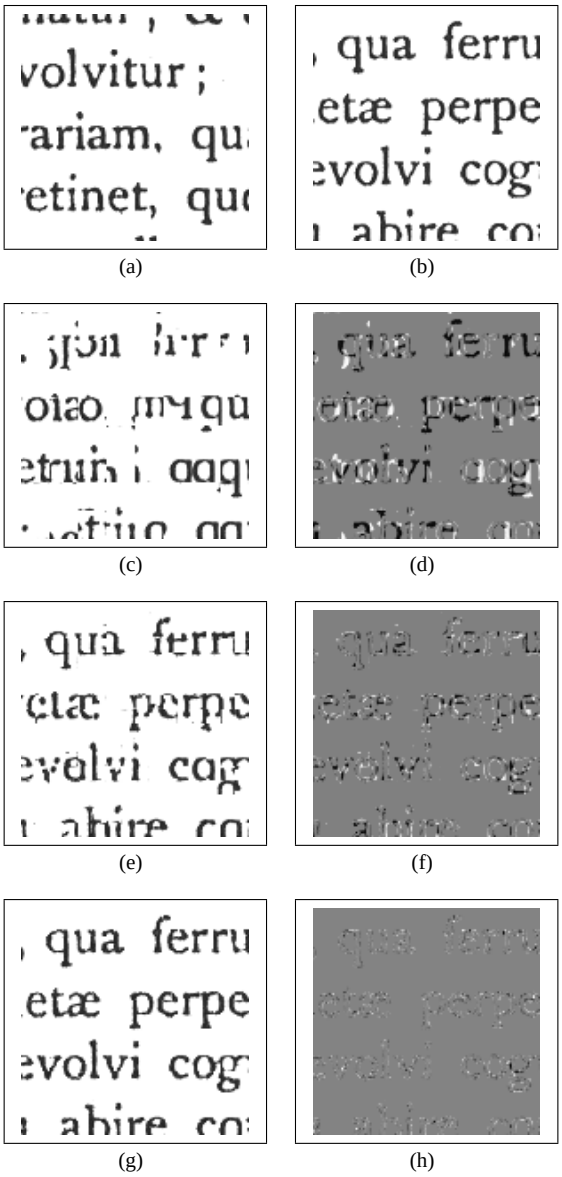


Figure 13. Approximate pattern matching using interframe prediction: (a) reference text; (b) current text; (c) predicted text (block size: 16×16 pixels); (d) prediction residue (block size: 16×16 pixels); (e) predicted text (block size: 8×8 pixels); (f) prediction residue (block size: 8×8 pixels); (g) predicted text (block size: 4×4 pixels); and (h) prediction residue (block size: 4×4 pixels).

B. Results

Two configuration parameters have greater influence on the encoder performance. One is the number of reference frames (R_f), the other is the search range (S_r). In our tests, different page sets were compressed using JPEG2000, AVC-I and H.264/AVC. In JPEG2000 and AVC-I compression, the pages are encoded separately. As for H.264/AVC, the first frame of the sequence is encoded as an I-frame (only intraframe prediction modes are used) and all the remaining

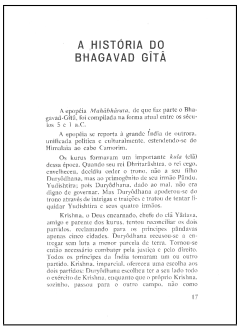


Figure 14. First page of test set "guita" used as example: total number of pages: 2, page size: 1568×1024 pixels.

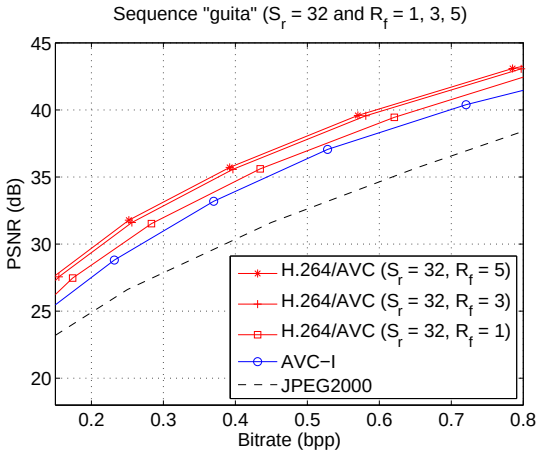
frames are encoded as P-frames (in addition to intraframe prediction, only past frames are used as reference by the interframe prediction). Figure 14 and Figures 15 (a) and (b) show the first page of test sequence "guita" and PSNR plots comparing JPEG2000, AVC-I and H.264/AVC, for different combinations of S_r and R_f, respectively. The PSNR was calculated using the global mean square error (MSE). The higher S_r and R_f values, the better rate-distortion performance. In particular, for S_r = 32 pixels and R_f = 5 frames, H.264/AVC outperforms AVC-I by more than 2 dB and JPEG2000 by more than 5 dB, at 0.5 bit/pixel (bpp).

C. Conclusion

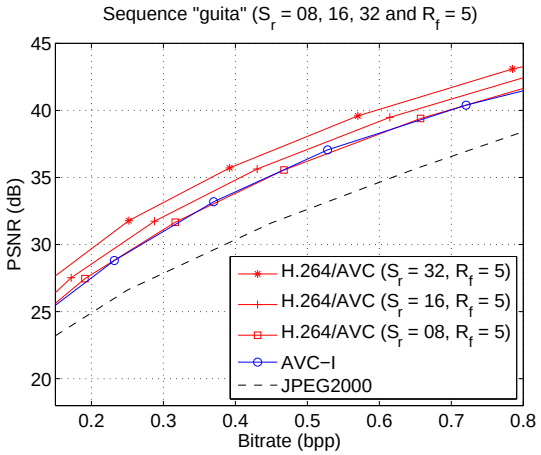
In this section we have shown how H.264/AVC, a video compression standard, may be used as a book compressor. Once the proposed method uses the pages of a book to construct a video sequence, H.264/AVC enables a hybrid pattern matching/transform-based encoder for this class of documents. Results show that the proposed method objectively outperforms AVC-I and JPEG2000 by up to 3 dB and 5 dB, respectively. Furthermore, the encoder outputs documents with superior subjective quality. Future works may include single-page compound document and multi-page compound book compression.

VI. Conclusion

This paper presented four document encoding methods that use H.264/AVC as a basic functional element, namely: method 1, *Advanced Video Coding - Compound*; method 2, *MRC Compression of Electronically Generated Documents using H.264/AVC-I and JBIG2*; method 3, *MRC Compression of Scanned Documents using H.264/AVC-I and JBIG2*; and method 4, *Compression of Scanned Books using H.264/AVC*. Many experiments were carried out in order to verify the efficiency of the proposed methods. Results showed objective and/or subjective gains over known approaches, thus contributing with more efficient document compression alternatives.



(a)



(b)

Figure 15. PSNR plots for test sets shown in Fig. 14. (a) and (b) "guita": comparison between JPEG2000, AVC-I and H.264/AVC, for different combinations of search ranges (S_r) and number of reference frames (R_f); (c) "principia" ($S_r = 32$ and $R_f = 5$).

REFERENCES

[1] JVT, "Advanced Video Coding for Generic Audiovisual Services. ITU-T Recommendation H.264," Novembro 2007.

[2] I. E. G. Richardson, *H.264 and MPEG-4 video Compression*. EUA: Wiley, 2003.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Julho 2003.

[4] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *Proceedings of SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC*, vol. 5558, pp. 53–74, Agosto 2004.

[5] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, (Primeiro quarto) Março 2004.

[6] D. Marpe, V. George, and T. Wiegand, "Performance Comparison of Intra-only H.264/AVC and JPEG2000 for a Set of Monochrome ISO/IEC Test Images," *Contribution JVT ISO/IEC MPEG and ITU-T VCEG, Doc. JVT M-014*, Outubro 2004.

[7] D. Marpe, V. George, H. L. Cycon, and K. U. Barthel, "Performance Evaluation of Motion-JPEG2000 in Comparison with H.264/AVC Operated in Pure Intra Coding Mode," *Wavelet Applications in Industrial Processing*, vol. 5266 of Proceedings of SPIE, pp. 129–137, Outubro 2004.

[8] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe Benefits of the H.264/AVC," *Proceedings of International Telecommunications Symposium*, pp. 208–212, Setembro 2006.

[9] JPEG, "Information Technology - JPEG2000 Image Coding System - Part 1: Core Coding System. ISO/IEC 15444-1," 2000.

[10] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. EUA: W. H. Freeman, 1982.

[11] R. L. de Queiroz, Z. Fan, and T. D. Tran, "Optimizing Block-thresholding Segmentation for Multilayer Compression of Compound Images," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1461–1471, Setembro 2000.

[12] K. Konstantinide and D. Tretter, "A JPEG Variable Quantization Method for Compound Documents," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1282–1287, Julho 2000.

[13] M. Ramos and R. de Queiroz, "Classified JPEG Coding of Mixed Documents," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 716–720, Abril 2000.

[14] J. Fan, "Text Extraction via an Edge-bounded Averaging and a Parametric Character Model," *Proceedings of SPIE Document Recognition and Retrieval X*, vol. 5010, pp. 8–19, Janeiro 2003.

[15] MRC, "Mixed Raster Content (MRC). ITU-T Recommendation T.44." 1999.

[16] R. L. de Queiroz, R. Buckley, and M. Xu, "Mixed Raster Content (MRC) Model for Compound Image Compression," *Proceedings of SPIE Visual Communications and Image Processing*, vol. 3653, pp. 1106–1117, Janeiro 1999.

[17] 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, pp. 410–425, 1998.

[18] G. Feng and C. A. Bouman, "High-quality MRC Document Coding, volume = 15, year = 2006," *IEEE Transactions on Image Processing*, no. 10, pp. 3152–3169, Outubro.

[19] A. Zaghetto, R. L. de Queiroz, and D. Mukherjee, "MRC Compression of Compound Documents using Threshold Segmentation, Iterative Data-filling and H.264/AVC-INTRA," *Proceedings of Indian Conference on Computer Vision, Graphics and Image Processing*, Dezembro 2008.

[20] JBIG2, "Information Technology - Coded Representation of Picture and Audio Information - Lossy/Lossless Coding of Bi-level Images. ITU-T Recommendation T.88," Março 2000.

[21] D. Mukherjee, C. Chrysafis, and A. Said, "JPEG2000-matched MRC Compression of Compound Documents," *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 73–76, Setembro 2002.

[22] D. Mukherjee, N. Memon, and A. Said, "JPEG-matched MRC Compression of Compound Documents," *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 434–437, Outubro 2001.

[23] R. L. de Queiroz, *Compressing Compound Documents, in The Document and Image Compression Handbook*. EUA: by M. Barni, Marcel-Dekker, 2005.

[24] ——, "On Data-flling Algorithms for MRC Layers," *Proceedings of IEEE International Conference on Image Processing*, pp. 586–589, Setembro 2000.

[25] G. Lakhani and R. Subedi, "Optimal Filling of FG/BG Layers of Compound Document Images," *Proceedings of IEEE International Conference on Image Processing*, pp. 2273–2276, Outubro 2006.

[26] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. Chapman and Hall, 1993.