ON DATA FILLING ALGORITHMS FOR MRC LAYERS

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

ABSTRACT

A recently adopted standard, the Mixed Raster Content (MRC) imaging model, represents compound images as a superposition of layers. Since layers are superimposed, large regions may not be imaged onto the final raster. Thus, those regions are redundant. In this paper we focus on techniques to replace the redundant data, i.e. on data filling redundant regions based on non-redundant ones. We present techniques to minimize the rate and distortion achieved by MRC compression through data filling. We start with a general method, then narrowing the presentation to DCT based compression of planes. Iterative block filling algorithms are presented in both spatial and DCT domain.

1. INTRODUCTION

Documents are now present in a wide spectrum of printing systems and they are frequently available as bitmaps containing a mix of text, graphics and pictures. These compound documents often challenge traditional compression mechanisms, which are generally developed with a particular image type and application in mind. Document compression is frequently linked to facsimile systems, in which large document bitmaps are compressed before transmission over telephone lines. Although the facsimile systems that most people are familiar with today are black-and-white (binary images) there is now a focus on new standards to provide color facsimile services over the telephone and the Internet [1].

The mixed raster content (MRC) imaging model [1]–[3], has been proposed as 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, thus allowing for multiple layers. 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.

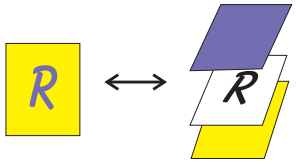

Figure 1. Illustration of MRC imaging model.

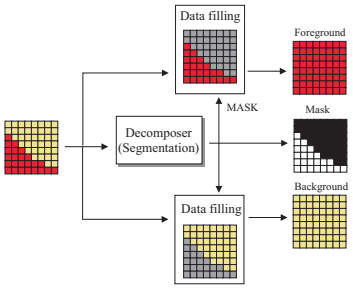

Figure 2. Diagram of a segmenter.

Layers may contain different dimensions and have offsets associated with them. 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 [5] or as an ITU-T MRC [2] data stream for delivery to the decoder. MRC has been proposed and/or accepted for several standards [1],[2],[5],[6], as well as been used in several products [7]–[10].

2. THE PROBLEM: DATA FILLING

Each layer (FG or BG) may contain unused pixels, since final pixels in some positions will be selected from the other layer. Thus, these pixels can be replaced by any color in order to enhance compression. This is the function of the pre-processor as illustrated in Fig. 2. 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. The problem dealt in this paper is the one of data filling over the redundant (unused) data. Redundant regions are marked in each FG/BG layer in Fig. 3. Our goal is to replace the redundant data with other data that will enhance compression.

For the remainder of this paper we will just refer to one layer, which can be either FG or BG, and consider replenishement of redundant regions.

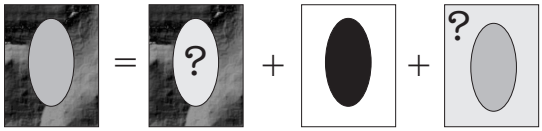

Figure 3. Redundant data.

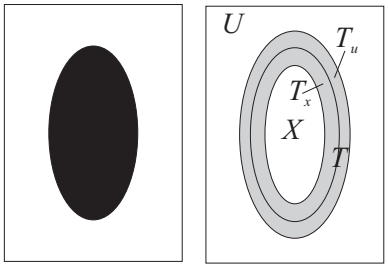

Figure 4. An example mask and labelling of the reference layer regions as useful (U), redundant (X), or transition (T). The transition can be further subdivided (T_U and T_X).

3. GENERAL APPROACH

A mask layer is illustrated in Fig. 4. along with the respective labelling of the reference layer (FG or BG). Without loss of generality, we assume the internal region to be the redundant one and label it with an X for "don't care". The external region is labelled useful (U). We also define a transition region (T) as the one where the transform bases (if any are used for compression) overlap across both the useful and redundant regions.

We want to replace the data in the X region (and T_X in Fig. 4) by any data that would improve compression. The overal goal is to reduce both the amount of bits produced (rate) and total distortion. For simplicity, we use a linear distortion model and aim to minimize a linear weight of rate and distortion, i.e. to minimize

$$J = R + \lambda D. \tag{1}$$

Since the rate and distortion are additive per regions (assuming they are independent), we have

$$J = R_U + R_X + R_T + \lambda(D_U + D_X + D_T) \tag{2}$$

where the subscript indicates the regions. Note that the redundant region is irrelevant for reconstruction, so that $D_X = 0$. Also note that since the replacement of redundant data does not affect the U region, the minimization of the above cost function is equivalent to minimizing

$$J = R_X + R_T + \lambda D_T \ . \tag{3}$$

i.e. to minimize rate in the X region and to make it RD efficient at transitions.

True optimality is a very ambitious goal which is beyond this paper's scope. The alternatives are too many to consider: layers can be resized or further processed and there are too many compression options, ranging from the transform type, through the wavelet type, to the choice of quantizers and entropy coders, etc. It is impractical to optimize the redundant data without fixing all these compression parameters, however we will attempt to offer a good compromise with a practical solution which aims to work well across a wide range of applications.

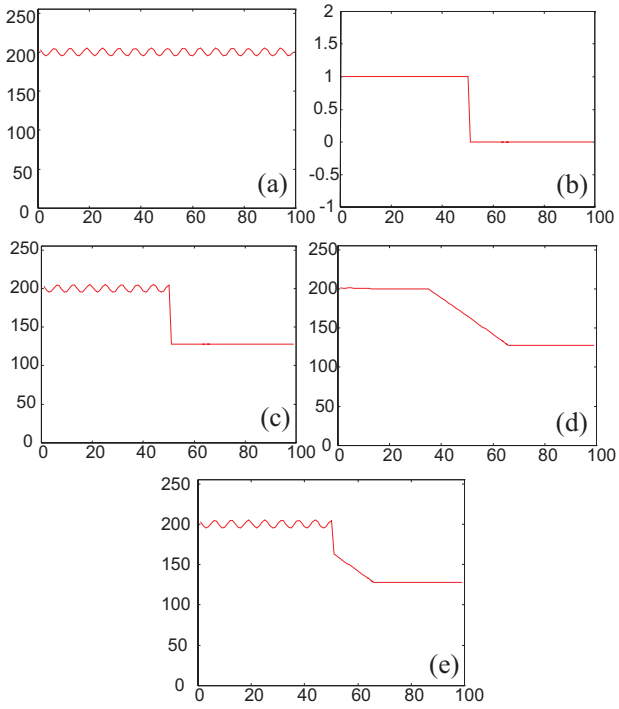

Figure 5. 1D example. (a) signal; (b) mask; (c) masked layer; (d) filtered version of (c); (e) masked filtered layer.

In order to minimize (3), it seems to be reasonable to apply smooth (flat) data to the redundant region. That would definitely reduce R_X to its virtual minimum. The question is what to do with the transitions, i.e. how to minimize $R_T + \lambda D_T$. If we would do that blindly, the best intuitive solution is to make the transition as smooth as possible. Smoother patterns tend to produce less bits and cause less distortion in most popular image coders.

The problem is to generate smooth transitions. Fig. 5 has an illustration of a 1D signal (a) which is masked using the mask in (b) yielding the signal in (c), where the redundant area was replaced by a constant value. If we simply filter the signal in Fig. 5(c) we obtain the signal in (d). After applying the mask, assuming we do not touch the useful region, the reconstructed signal is as in Fig. 5(e). Note that there still exist a discontinuity, which is caused by the symmetry of the filtering process around the edge. We propose to use a segmented filtering, wherein filter weights are exaggerated for pixels in the useful region of the layer. In 1D,

$$y(n) = \frac{\sum_{k=-L}^{L} h(k,n)x(n+k)}{\sum_{k=-L}^{L} h(k,n)} \tag{4}$$

where $h(k,n)$ is a time varying filter of $2L+1$ taps. Its weights are dependent on the mask values $m(n)$ as:

$$h(k,n) = \begin{cases} 1 & \text{if } m(n+k) = 0 \\ Mf(k) + 1 & \text{if } m(n+k) = 1 \end{cases} \tag{5}$$

where $f(k)$ is a filter window such as Hamming or Kaiser to de-emphasize distant pixels within the useful region. The

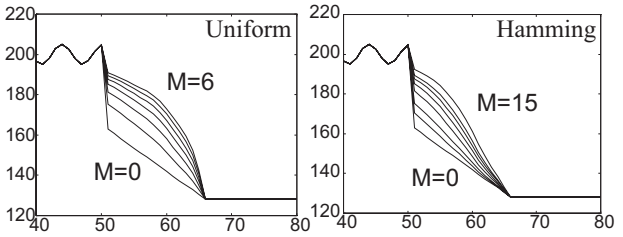

Figure 6. Segmented filtering for the 1D example for different M and for two windows.

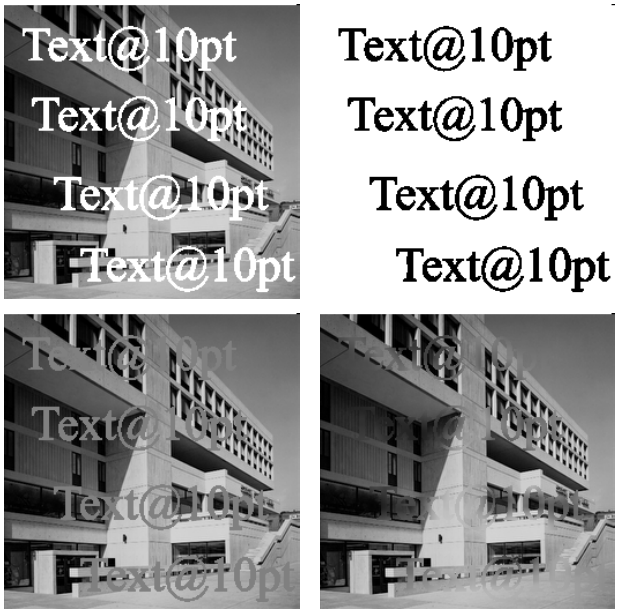

Figure 7. Top: the compound image and its associated mask. Bottom, processed layers with mid-gray (128) filling; left: no filtering; right: segmented filtering.

result of applying the segmented filter is shown in Fig. 6 for $L = 16$, by varying M and for a rectangular (uniform) and for a Hamming window. Note how the discontinuity is largely decreased. The case $M = 0$ is equivalent to a straight averaging filter without any emphasis. This filtering is very easy to implement. For uniform windows, like the averaging filter case, complexity is virtually independent of window size, and very large filters are easily implemented.

One example: processed layers images in Fig. 7 were compressed at 0.5 bpp using JPEG 2000 [6]. We used $L = 10$, i.e. 21×21 filters with uniform windows. For $M = 6$, $M = 0$ (no segmented filter) and without any filter ($L = 0$) the reconstruction PSNR (dB) are

rate	$M = 6$	$M = 0$	No filtering
0.5 bpp	29.3	29.0	27.3

where it can be seen segmented filtering can be advantageous to simple filtering or to no filtering at all. This result is typical. Several other images and bit-rates were tested with similar results.

The present approach was devised having in mind that the coder is generic. If the coder is known, there are ways to further optimize the data filling process. One example is the method used by DjVu [8] in which the wavelet transform is known and a set of interative projections are made in order

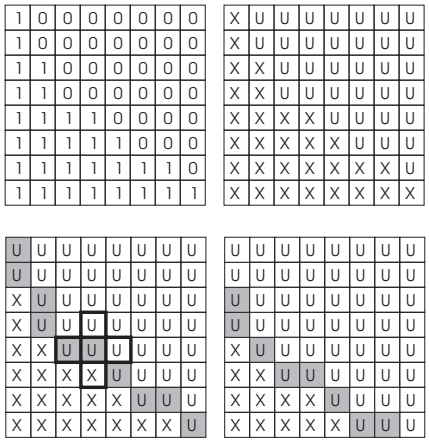

Figure 8. Top: an example block mask and respective labels for layer pixels. Bottom: two further stages of the iterative spatial domain block filling algorithm.

to slowly approach a stable and efficient solution.

4. ITERATIVE BLOCK SMOOTHING

For block transforms, we assume JPEG compression, and without loss of generality we assume 8×8 pixel blocks, although the methods here described can be easily scaled to other sizes. As in the previous session, the pre-processor receives an input block and, by inspecting the binary mask, labels the input block pixels as useful (U) or "don't care" (X). The substitution of redundant pixels is performed through two methods which are discussed next.

4.1. Spatial domain

The spatial domain algorithm is relatively simple and inexpensive. 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 [4]). 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, X-pixels which have at least one U-pixel horizontal or vertical neighbour (N_4 NSEW neighbourhood as indicated in Fig. 8) are replaced by the average of those neighbours as illustrated in Fig. 8. In the next pass, those pixels that were replaced are marked U for the next pass. Fig. 8 illustrates two steps, while the process is continued until there are no X-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 in the U-marked pixels. Its disadvantage is that the X-marked pixels are just influenced by the bordering U pixels, i.e. an internal U pixel does not affect data filling. This is acceptable for most applications. This method is actually the one used for optimized segmentation in [11].

In a real compression example, MRC layers after data-filling for a given target image, are shown in Fig. 9. The method used is the iterative spatial domain algorithm just described.

4.2. DCT domain

There is a more expensive alternative to account for all pixels in the block for data filling. In this method

- Initialize X-pixels in any way, for example as the average of the U-pixels.

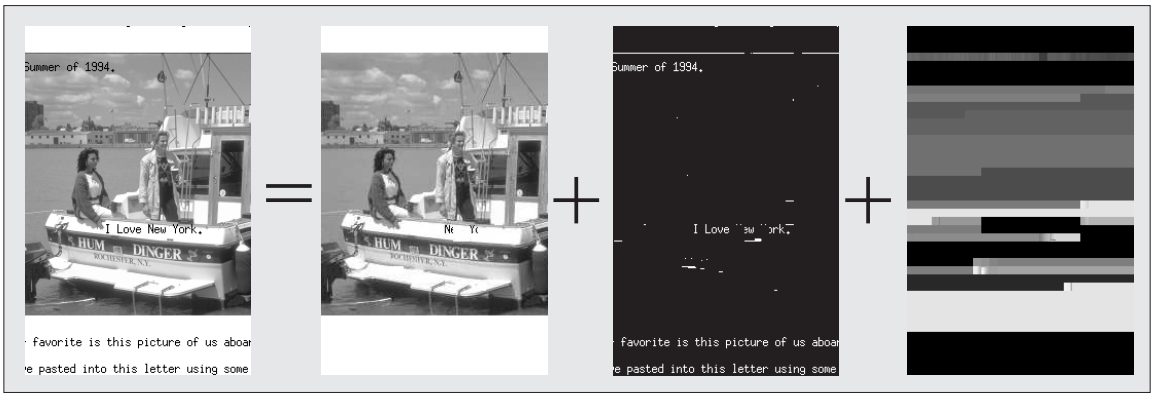

Figure 9. Portions of spatial-domain block filled layers.

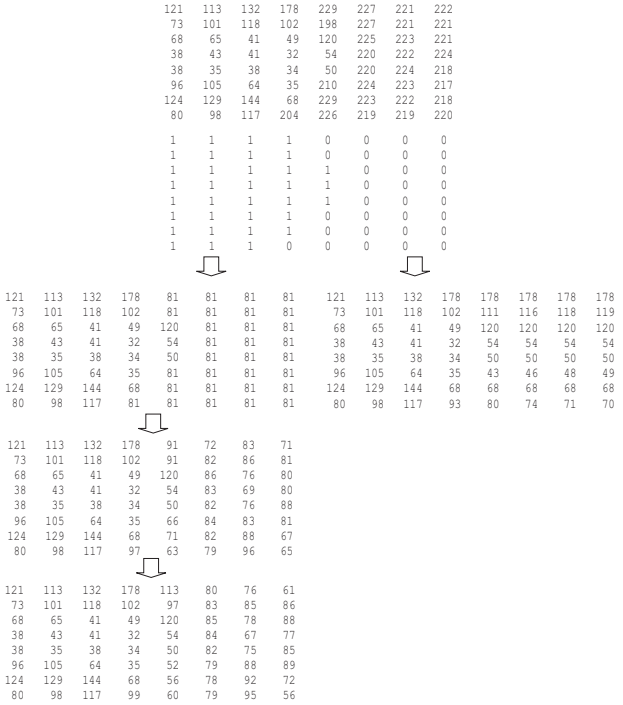

Figure 10. Iterative DCT-domain block filling. Top: block and mask. Left: 3 steps in the DCT domain algorithm. Right: spatial domain method result.

- Transform, quantize and inverse transform the block, obtaining a new set of pixels in the block (call them X' and U' pixels).
- Replace X-pixels by X'-pixels in the original block.
- Repeat the transformation and replacement process until convergence is reached.

Convergence is achieved when X' and X pixels are identical or close within a certain prescribed tolerance. It usually happens after very few iteractions. An an illustration, Fig. 10 shows an example block, its respective mask and the resulting block using DCT domain algorithm. For comparison, a block resulting from the spatial domain algorithm is also presented.

5. CONCLUSIONS

Data filling algorithms were presented not only to cover DCT-based compression, but also as a generic method aiming at a broad spectrum of applications. There is still much more to be done. Ideally, segmentation and data filling should be done in one step, i.e. the decision of what to place in the FG, BG and Mask players is interdependent. For now, smart data filling seems to be the best solution for reducing the effect that MRC redundancy has on compression.

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

[3] 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.

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

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

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

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

[8] 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.

[9] M. Thierschmann, K. Bartel, S. McPartlin, U. Martin, "New technology for raster document image compression," *EI'2000, Document Recognition and Retrieval VII, Proc. SPIE,* Vol. 3967, Jan. 2000.

[10] 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.

[11] R. de Queiroz, Z. Fan. T. Tran, "Optimizing block-thresholding segmentation for MRC compression," in this Proceedings.