High Quality MRC Document Coding*

***Guotong Feng*†, *Hui Cheng*‡, *and Charles Bouman*†**
†*School of Electrical and Computer Engineering*
Purdue University
West Lafayette, IN 47907-1285
‡*Sarnoff Corporation*
Visual Information Systems
Princeton, NJ 08543-5300

Abstract

The mix raster content (MRC) model can be used to implement highly effective document compression algorithms. However, many MRC based methods which achieve high compression ratio can distort some fine details of document quality, such as thin lines, and text edges. In this paper, we present a method called resolution enhanced rendering (RER) to achieve high quality rendering of document containing text, pictures and graphics, while maintaining desired compression ratios. The method applies adaptive dithering to the MRC encoder and then performs a nonlinear prediction in the MRC decoder. Both the dithering and nonlinear prediction algorithms are jointly optimized to produce the best quality rendering.

We present experimental results illustrating the performance of our method and comparing it to some existing MRC compression algorithms.

1. Introduction

Document imaging applications such as scan-to-print, document archiving, and internet fax are driving the need for document compression standards that maintain high quality while achieving high compression ratios. Recently, the mix raster content (MRC) compression model has been adopted as a standard for document encoding [1]. The MRC standard allows raster documents to be coded at very high compression ratios but with much lower distortion than would be possible using conventional image coding methods [2]. While MRC methods are much better than conventional transform coders, they still can substantially distort fine document details, such as thin lines and text edges.

In this paper, we propose a method called resolution enhanced rendering (RER) for jointly optimizing the MRC encoder and decoder to achieve high quality rendering of document text, image and graphics, while maintaining desired compression ratios. The method works by adaptively dithering the mask layer of a three-layer MRC encoding to produce the intermediate tone levels required for high quality rendering. The dithering is performed using a novel adaptive error diffusion algorithm. A tree-based nonlinear predictor is then designed into the MRC decoder to reconstruct the desired intermediate tones. Both the dithering and nonlinear prediction algorithms are jointly optimized to produce the best quality rendering. The optimization is performed by iteratively optimizing the encoder and decoder to achieve the minimum distortion.

This method also has a number of potential advantages. First, it is compatible with the MRC standard. That is to say, the RER enhanced encoder works with a conventional MRC decoder, and the RER enhanced decoder works with a conventional MRC encoder. Second, the method can be implemented using the previously proposed the Rate Distortion Optimized Segmentation (RDOS) method. The RDOS method computes the segmentation that minimizes a combination of bit-rate and distortion. We will present experimental results comparing the performance of RDOS compression with and without the RER method.

2. Conventional MRC Encoder

An MRC encoder is based on the Mixed Raster Content imaging model, which represents a document by layers with different properties. As shown in Figure 1, a three-layer MRC document contains a background layer, a foreground layer, and a binary mask layer. At each pixel, the value of the binary mask is used to select between the foreground and background pixels. In the MRC model, each layer is compressed independently. This adds some inefficiency since the foreground and background layers must be coded even when they are not used [3], but it simplifies the imaging model. Typically, the foreground and background layers are compressed using natural image coders such as JPEG or embedded wavelet coder [4]; whereas the binary mask is typically encoded with a lossless binary encoder such as JBIG or JBIG2 [5].

*This work is supported by the Xerox Foundation.

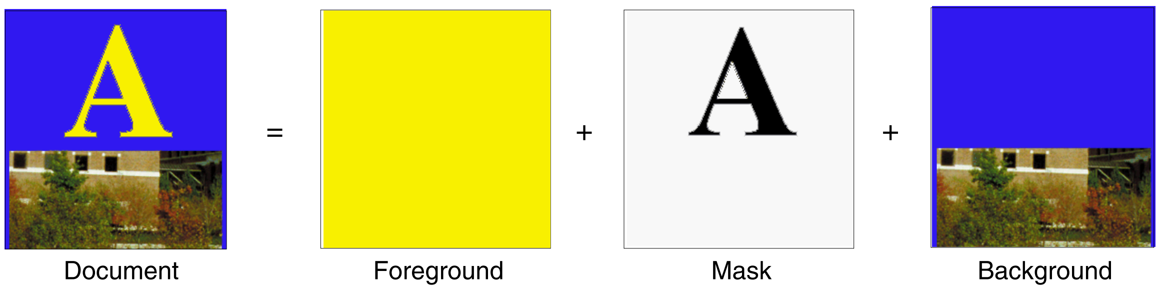


Figure 1: *MRC imaging model forms text and line art by using a binary mask to choose between foreground and background layers.*

In this work, we focus on the rate distortion optimized segmentation (RDOS) algorithm of [6, 7]. Strictly speaking, the RDOS encoder is not a true MRC encoder because it does not encode each layer of the MRC model independently. Nonetheless, the RDOS method can in principal be modified to be a true MRC method, and the methods introduced in this work are equally applicable to any typical MRC encoder.

The RDOS algorithm classifies each 8×8 block of pixels into one of four classes: “picture block”, “two-color block”, “one-color block” or “other block”. Each class corresponds to a different coding method. The picture and other blocks use JPEG block encoders. The one-color blocks are entropy coded using an arithmetic encoder. For each two-color block, both the foreground and background colors are entropy coded using the arithmetic encoders while the 8×8 binary mask is encoded using a JBIG2 encoder. The class of each block is chosen to maximize the rate-distortion performance over the entire document. The optimization is achieved by applying each candidate coding method to each block and then selecting the method which yields the best rate-distortion trade-off.

3. Resolution Enhanced Rendering Method

MRC encoders have an enormous advantage for document encoding because they can efficiently encode text and line art with very high spatial resolution. However, one limitation of conventional MRC encoders is that the mask can only represent binary transitions at text edges. This makes accurate representation of text edges difficult. In principal, it is possible to add edge detail to the foreground or background layers; however, in practice this detail is lost when those layers are encoded using natural image coders at acceptable bit rates.

Figure 2 shows how the resolution enhanced rendering (RER) algorithm adds edge detail while retaining the binary MRC mask layer. First, the RER encoder segments the foreground and background using an adaptive error diffusion method. This error diffusion method effectively dithers the binary mask along the edge of the character to represent the gradual transition of true raster scanned text characters. The error diffusion algorithm uses the local value of the mask to adapt the error diffusion weights so that error is diffused along the 1-D mask boundary.

The RER decoder uses the binary mask, together with the foreground and background colors to estimate the true value of the document pixels. This estimation is done using a nonlinear tree-structured predictor as described in [8, 9]. Importantly, this predictor is trained to identify the characteristic patterns of the RER encoder. Therefore, it can do a much better job of accurately estimating the true pixel values.

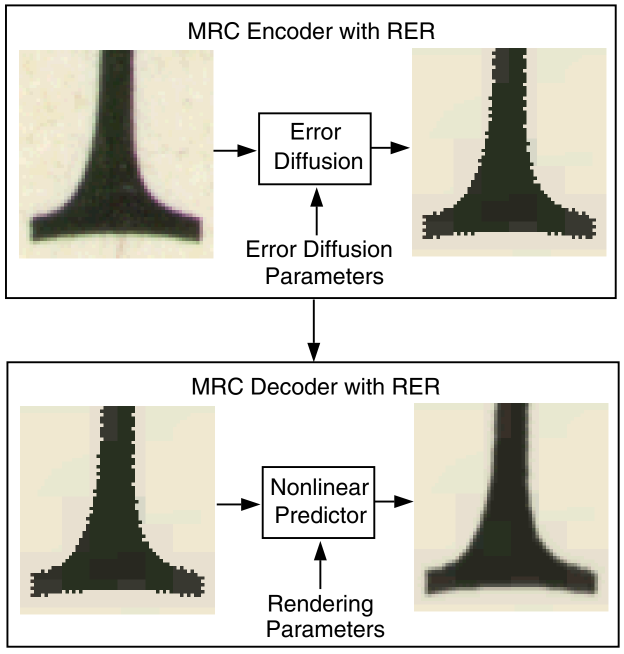


Figure 2: *Illustration of MRC encoder and decoder with RER. Examples were selected from actual RER inputs and outputs.*

Figure 3 illustrates how the RER encoder and decoder are jointly optimized to maximize the quality of the decoded document. As we will see, both the encoder and the decoder have parameters which can be trained to produce

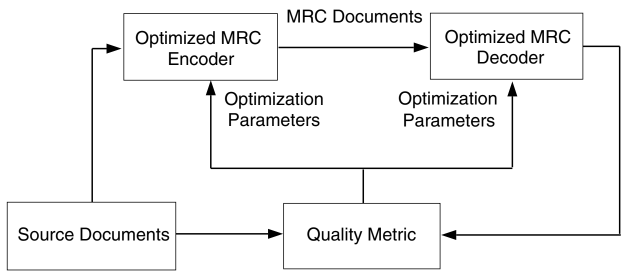


Figure 3: Overview of method used to train the optimized encoder and decoder. Once training is complete, the encoder and decoder function independently.

the best possible result. The error diffusion algorithm has five parameters which control its behavior, and the nonlinear predictor has a large number of parameters which specify the nodes of a nonlinear regression tree.

In each iteration of the optimization, the parameters of the encoder or decoder are alternatively fixed, while the parameters of the other one are optimized. Importantly, two different sets of documents are used for training the encoder and decoder. We have found this improves the robustness of the training procedure. Experimental results are shown for test documents that are not contained in either set of training documents. The experimental results indicate that this training process robustly converges to parameters which reduce the distortion of the decoded document. Moreover, we have found that joint optimization of the encoder and decoder performs substantially better than independent optimization of these two functions.

3.1. The RER Encoder

Let X_s be a pixel in the raster document at location s. In the MRC format, each pixel also has an associated foreground color, F_s, and background color, B_s. The binary MRC mask then determines whether F_s or B_s will be used to represent the true pixel value X_s. In RDOS encoding, the foreground and background colors are constant in 8×8 blocks. But in other MRC encoding methods, the values of the foreground and background colors can change from pixel to pixel.

Next define the scalar value λ_s which determines the relative mixture of foreground and background color in the pixel X_s. More specifically, λ_s is given by the value on the real line which minimizes the squared error

$$||X_s - (B_s + \lambda_s(F_s - B_s))||^2 \tag{1}$$

Figure 4 gives a geometric interpretation of λ_s as the projection of the true pixel color onto the line connecting the foreground and background colors. The solution to this least squares approximation problem is given by

$$\lambda_s = \frac{(X_s - B_s)^t(F_s - B_s)}{(F_s - B_s)^t(F_s - B_s)} \, . \tag{2}$$

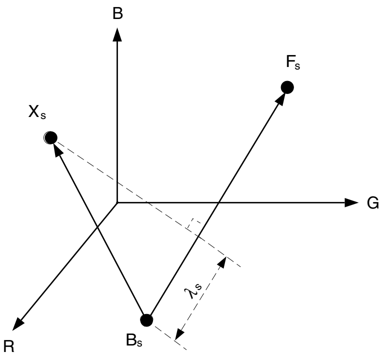


Figure 4: Least squares approximation of pixel color, X_s, by a combination of the background, B_s, and foreground, F_s, colors.

Furthermore, let γ_s denote the value of λ_s constrained to the interval $[0, 1]$.

$$\gamma_s = \min\{1, \max\{0, \lambda_s\}\} \tag{3}$$

So, γ_s forms a gray scale image with minimum value 0 and maximum value 1. Notice that when $\gamma_s = 0$, the pixel s is primarily background, and when $\gamma_s = 1$, the pixel s is primarily foreground.

We also define the minimum approximation error at each pixel, Δ_s, in terms of the value γ_s.

$$\Delta_s = \min\{\gamma_s, 1 - \gamma_s\} \tag{4}$$

Notice that when a pixel is well approximated by either the forground or background color, then Δ_s is small. On the other hand, when a pixel is best approximated by a mixture of foreground and background colors, then Δ_s is large. The value of Δ_s will be used to control the local adaptation of the error diffusion algorithm.

Typically, MRC encoders compute the binary mask by classifying each pixel as foreground or background independently. However, the RER encoder computes the binary mask by applying a form of adaptive error diffusion to the gray scale image γ_s. This effectively dithers the binary mask along character edges so that the mask can more accurately represent fine gradations in the transition between foreground and background colors.

While the RER error diffusion method is similar to serpentine scan Floyd Steinberg error diffusion [10], it is specially designed and optimized to diffuse error along text edge transitions. This is done by adaptively setting the error diffusion weights at each pixel.

Figure 5 illustrates the four future pixels s_0, s_1, s_2, s_3 which neighbor s in serpentine scan order. Then $w_{s_0}, w_{s_1}, w_{s_2}, w_{s_3}$ are the values of the four corresponding error diffusion weights. The values of these four weights are varied at each pixel s using the formula

$$w_{s_j} = \frac{\Delta_{s_j} u(\Delta_{s_j} - \tau)}{\sum_{j=0}^{3} \alpha_j \Delta_{s_j} + 0.001} \tag{5}$$

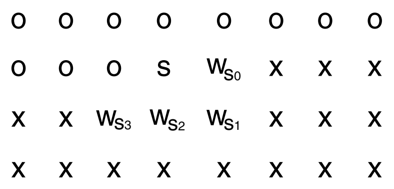


Figure 5: Illustration of error diffusion algorithm. The o's are pixels that have already been processed, the current pixel position is denoted by s, and s_0, s_1, s_2, s_3 denote the four future pixel positions to which error will be diffused. The corresponding error diffusion weights at these four positions are denoted by $w_{s_0}, w_{s_1}, w_{s_2}, w_{s_3}$. These four weights are adapted based on local edge orientation in the binary mask.

where Δ_{s_j} is the approximation error at s_j defined in (4), $u(\cdot)$ is the unit step function, and $(\tau, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$ is a five dimensional vector of scalar parameters, with each parameter falling in the interval $[0, 1]$.

Equation (5) is designed to diffuse the error into the direction with the largest approximation error. In the case of text edges, this means that it tends to diffuse the error along the edge of the character. Moreover, when the parameter $\tau > 0$, the error is not diffused into regions where the approximation is good, since in this case $u(\Delta_{s_j} - \tau) = 0$. This means the error is not generally diffused into smooth interior regions of a character where it could generate isolated holes in the binary mask. The vector $\Theta = (\tau, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$ parameterizes the RER encoder. Section 4 describes how this parameter vector is estimated from training data to maximize decoded document quality. Once the binary mask is generated, it is then losslessly coded using a JBIG2 encoder.

Since the RDOS coder is not a true MRC encoder, it requires that we handle a few special cases that arise. Image blocks that are not assigned the two-color class, do not have foreground and background colors in the RDOS method. For these blocks, we simply assign $\gamma = 0$ which is equivalent to assuming that all pixels are background pixels.

3.2. RER Decoder

Figure 6 illustrates the basic structure of the RER decoder. The decoder works by using a nonlinear predictor to compute, $\hat{\lambda}_s$, the minimum mean squared error estimate of λ_s. Using this estimate, the reconstructed pixel color can be computed as

$$\hat{X}_s = \hat{\lambda}_s F_s + (1 - \hat{\lambda}_s) B_s \ . \quad (6)$$

Here we assume that the foreground and background colors are the same as used in the RER encoder.

The nonlinear predictor works by first extracting the binary mask in a 5×5 window about the pixel in question.

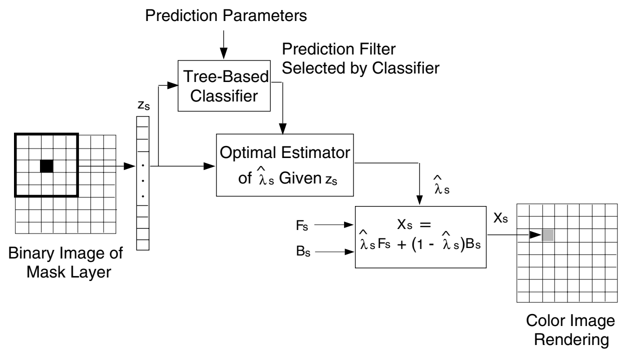

Prediction Parameters
Tree-Based Classifier
Prediction Filter Selected by Classifier
z_s
Optimal Estimator of $\hat{\lambda}_s$ Given z_s
$\hat{\lambda}_s$
F_s
B_s
$X_s = \hat{\lambda}_s F_s + (1 - \hat{\lambda}_s) B_s$
X_s
Binary Image of Mask Layer
Color Image Rendering

Figure 6: Structure of RER decoder using nonlinear predictor.

This data forms a binary vector, z_s, which is then used as input to a binary regression tree predictor known as Tree-Based Resolution Synthesis (TBRS) [8, 9]. The TBRS predictor estimates the value of λ_s in a two-step process. First, it classifies the vector z_s into one of M classes using a binary tree classifier. Each class, then has a corresponding linear prediction filter which is used to estimate the value of λ_s from z_s using the equation

$$\hat{\lambda}_s = A_m z_s + b_m$$

where m is the determined class of the vector z_s, A_m and b_m are the corresponding linear prediction parameters of class m.

The basic idea of TBRS is to use a binary regression tree as a piecewise linear approximation to the conditional mean estimator. The classification step is essential because it can separate out the distinct regions of the document corresponding to mask edges of different orientation and shape.

One additional complication occurs with the RDOS method. Since it is not a true MRC encoder, pixels which fall outside of two-color blocks have no binary mask values. This can cause a problem when the pixel s falls near the boundary of a block, and the 5×5 window about the pixel covers part of the adjacent block that is not a two-color block. In this case, the pixels are classified as either 0, 1, or 2 depending on if they are close to the background color, the foreground color or neither color. Then the values 0, 1, and 2 are encoded as binary values 00, 01, and 10, to insure that the input vector z_s remain binary.

4. Training

The objective of the training process is to optimize the performance of the RER encoder and decoder by selecting the encoder and decoder parameters to maximize the decoded document quality over a training set of documents. The distortion metric used to measure document quality is mean squared error. While mean squared error is not always a good measure of quality, for this application we found that it was always well correlated with our subjective evaluation of quality.

The training process alternated between optimization of the encoder and decoder parameters. So, when optimizing the encoder parameters, the previously obtained decoder parameters were used; and when optimizing the decoder parameters, the previously obtained encoder parameters were used. The training phases for encoder and decoder used different sets of training data. This strategy seemed to produce more robust training results.

The iterative optimization is always started by optimizing the decoder and using the initial encoder parameters

$$\begin{aligned}\Theta &= [\tau, \alpha_0, \alpha_1, \alpha_2, \alpha_3] \\ &= [0, 1, 1, 1, 1] \, .\end{aligned}$$

The iterative procedure is ended after 5 iterations of both the encoder and decoder.

4.1. Decoder Optimization

While the TBRS predictor is very efficient to implement, it can be computationally expensive to train. The training process includes three steps: generating training vector pairs, building the regression tree, and generating the least square prediction filters. The details of the training process are explained in the publications [8, 9].

The regression tree is trained by selecting out relevant pixels in the training image data. At each selected pixel, we extract the values of λ_s and z_s, the binary vector representing the mask values in a 5×5 window about s. These training pairs, (λ_s, z_s), are then used to train the predictor.

The training pairs are only extracted at pixels for which $-0.1 \leq \lambda_s \leq 1.1$. This is very important because if λ_s falls outside this range, then it can not be usefully predicted by the binary vector z_s. We found that selecting training pairs in this manner substantially improved the quality of the decoded documents.

4.2. Encoder Optimization

The encoder is optimized by searching for the value of the vector $\Theta = [\tau, \alpha_0, \alpha_1, \alpha_2, \alpha_3]$ which minimizes the mean squared error averaged over the set of training documents. This search is initialized at the current value of the vector Θ and uses the decoder parameters resulting from the last optimization of the decoder.

The optimization of Θ is done by sequentially perturbing the elements of the vector Θ using an iterative coordinate decent strategy. Each perturbation is made using a step size of $\pm\delta$ which is initialized to the value $\delta = 0.2$. The decision to perturb each component by $\pm\delta$ or to leave it unchanged is made based on the value of the mean squared error computed with the set of training documents.

If the mean squared error does not decrease with the perturbation of every element of the parameter vector, then the size of the perturbation is automatically reduced using the formula

$$\delta \leftarrow \delta/2 \, .$$

The coordinate descent optimization is stopped when the perturbation value is less than 0.01.

5. Experimental Results

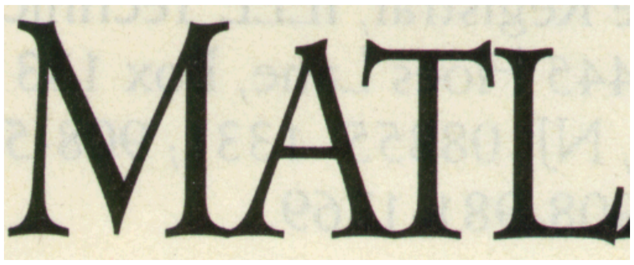
In this section, we present experimental result illustrating the value of the RER methods. For these results, we used an encoder training set consisting of portions of 5 documents. The decoder training set consisted of 16 full documents. All results were obtained from testing data not contained in either of the training sets. All raster document data was scanned at 400 dpi and 24 bits per pixel on the HP 6100C flatbed scanner.

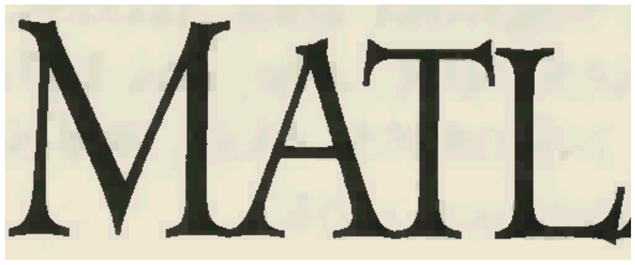
Figure 7 shows the comparison among the original image, the image rendered by standard RDOS, and the image rendered by the RER enhanced RDOS encoder/decoder pair. Both results are taken at a bit rate of approximately 0.18 bits per pixel. Figure 7b exhibits objectionable "jaggie" artifacts around the text edges, which are the common problems existing in most MRC encoders due to the limitation of the binary mask layer approach. Figure 7c shows that the RER method eliminates these artifacts and produces a decoded document which is quite close to the original shown in Figure 7a.

Figure 8 illustrates the comparison of rate-distortion performance between standard RDOS, and RER enhanced RDOS. The two rate-distortion curves show the substantial improvement achieved by the RER method. The rate-distortion curve of the RER method has a sharp knee at about 0.16 bits per pixel (e.g. 150:1 compression ratio), and then decreases with a much slower rate than that of standard RDOS. This implies that over a wide range of bit rates the RER method can reduce distortion with no increase in bit rate.

6. Conclusions

In this paper, we have proposed the resolution enhanced rendering (RER) method for implementation with standard mixed raster content (MRC) encoders and decoders. The RER method works by encoding edge detail into the binary mask layer of the MRC using an adaptive error diffusion method. It then decodes the MRC document by using a nonlinear predictor to determine the relative amount of foreground and background color to apply to each pixel. We propose a method for jointly optimizing the parameters of the RER encoder and decoder to yield maximum document image quality. Our experimental results indicate


(a)


(b)

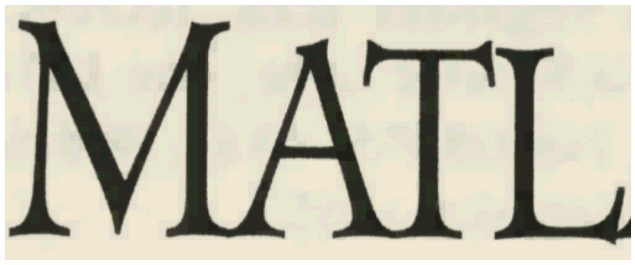

(c)

Figure 7: *Comparison of compression results. (a) A portion of the original test image; (b) Compressed by standard RDOS at 0.184 bpp (130:1 compression ratio); (c) Compressed by RER enhanced RDOS at 0.182 bpp (132:1 compression ratio).*

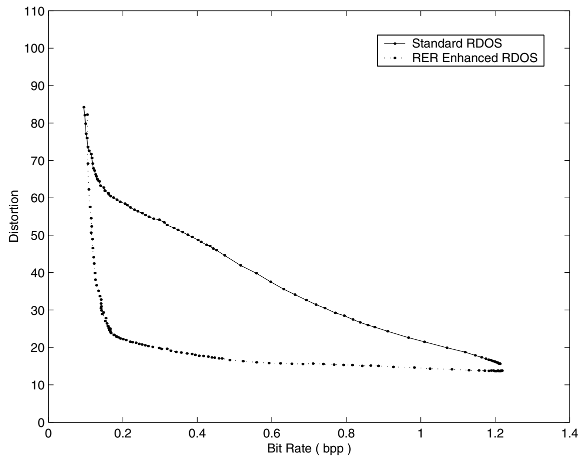


Figure 8: *R-D performance of RER enhanced RDOS and standard RDOS.*

that the RER method can substantially improve document image quality for a fixed bit rate. In addition, the RER method has the advantage that it can be efficiently implemented on general MRC encoders and decoders, and it is fully compatible with the existing MRC standard.

7. References

[1] R. L. de Queiroz, R. Buckley, and M. Xu, "Mixed raster content (MRC) model for compound image compression," in *Proc. IS&T/SPIE Symp. on Electronic Imaging, Visual Communications and Image Processing*, San Jose, CA, Februray 1999, vol. 3653, pp. 1106–1117.

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

[3] M. Ramos and R. L. de Queiroz, "Adaptive rate-distortion-based thresholding: application in JPEG compression of mixed images for printing," in *Proc. of IEEE Int'l Conf. on Image Proc.*, Kobe, Japan, October 25-28 1999.

[4] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[5] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The emerging JBIG2 standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838–848, November 1998.

[6] H. Cheng and C. A. Bouman, "Multiscale document compression algorithm," in *Proc. of IEEE Int'l Conf. on Image Proc.*, Kobe, Japan, October 25-28 1999.

[7] H. Cheng and C. A. Bouman, "Document compression using rate-distortion optimized segmentation," *Journal of Electronic Imaging*, p. to appear, April 2001.

[8] C. B. Atkins, *Classification Based Methods in Optimal Image Interpolation*, Ph.D. dissertation, Purdue University, West Lafayette, IN, December 1998.

[9] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Tree-based resolution synthesis," in *Proc. of the Image Proc., Image Quality, Image Capture Systems Conference (PICS '99)*, Savannah, GA, April 25-28 1999, pp. 405–410.

[10] R. A. Ulichney, "Dithering with blue noise," *Proc. of the IEEE*, vol. 76, pp. 56–79, January 1988.