MULTISCALE SEGMENTATION FOR MRC DOCUMENT COMPRESSION USING COST FUNCTION

Sreekesh Namboodiri.T , PG Student
Department of CSE, PSN College of Engineering and Technology, Tirunelveli, India

G.Kharmega Sundararaj, Associate Professor,
Department of CSE, PSN College of Engineering and Technology, Tirunelveli, India.

ABSTRACT

The Mixed Raster Content (MRC) standard (ITU-T T.44) specifies a framework for document compression which can dramatically improve the compression/quality tradeoff as compared to traditional lossy image compression algorithms. The key to MRC's performance is the separation of the document into foreground and back- ground layers, represented as a binary mask. In this paper, we propose a integrated segmentation algorithm which is based on the sequential application of two algorithms. Cost Optimized Segmentation (COS), is a blockwise segmentation algorithm . The second algorithm, Connected Component Classification (CCC), refines the initial segmentation by classifying feature vectors of connected components using a Markov random field (MRF) model. The integrated COS/CCC segmentation algorithms are then incorporated to a resolution enhanced rendering (RER) method ie to achieve high quality rendering of document containing text, pictures and graphics, while maintaining desired compression ratios

Index Terms—Document compression, image segmentation, Markov random fields, MRC compression, Multiscale image analysis.

I. INTRODUCTION

The Mixed Raster Content (MRC) standard is a framework for layer- based document compression defined in the ITU-T T.44 [1] that enables the preservation of text detail while reducing the bitrate of en- coded raster documents. Perhaps the most critical step in MRC en- coding is the segmentation step, which creates a binary mask that separates text and line-graphics from natural image and background regions in the document. The most traditional approach to document binarization is Otsu's method [2] which thresholds pixels in an effort to divide the document's histogram into objects and background. However, many re- cent approaches to document Binarization have been based on statistical models. One of the best commercial document binarization algorithms, which is incorporated in the DjVu document encoder, uses a hidden Markov model (HMM) [3, 4]. Zheng *et al.* [5] used a MRF model to exploit the contextual document information for noise removal. Similarly, Kumar [6] *et al.* used a MRF model to refine the initial segmentation generated by wavelet analysis. J. G. Kuk *et al.* and Cao *et al.* also developed a MAP-MRF document binarization framework which incorporates their proposed prior model [7, 8].

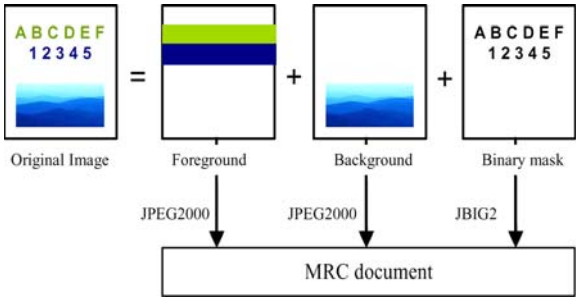

Fig 1 standard MRC document model

In this paper, we introduce a multiscale segmentation algorithm for both detecting and segmenting text from a complex document. This overall segmentation algorithm is performed by applying two algorithms in sequence: the Cost Optimized Segmentation (COS) algorithm and the Connected Component Classification (CCC) algorithm. The COS algorithm is a blockwise segmentation algorithm based on cost optimization. The COS produces a binary image from a gray level or color document; however, the resulting binary image typically contains many false text detections. The CCC algorithm further processes the resulting binary image to improve the accuracy of the segmentation. It does this by detecting non-text components (i.e. false text detections) in a Bayesian framework which incorporates a Markov random field

(MRF) model of the component labels. One important innovation of our method is in the design of the MRF prior model used in the CCC detection of text components. In particular, we design the energy terms in the MRF distribution so that they adapt to attributes of the neighboring components' relative locations and appearance. By doing this, the MRF can enforce stronger dependencies between components which are more likely to have come from related portions of the document.

II. COST OPTIMIZED SEGMENTATION (COS)

The procedure for Cost Optimized Segmentation (COS) is as follows. The image is first divided into overlapping blocks. Each block contains $m \times m$ pixels, and adjacent blocks overlap by $m/2$ pixels in both the horizontal and vertical directions. The blocks are denoted, $O_{i,j}$ for $i = 1, .., M$, and $j = 1, .., N$, where M and N are the number of the blocks in the vertical and horizontal directions. The pixels in each block are segmented into foreground ("1") or background ("0") by the clustering method of Cheng and Bouman [9]. This results in an initial binary mask for each block denoted by $C_{i,j} \in \{0, 1\}^{m \times m}$. However, in order to form a consistent segmentation of the page, these initial block segmentations must be merged into a single binary mask. To do this, we allow each block to be modified using a class assignment, $s_{i,j} \in \{0, 1, 2, 3\}$, as follows,

$$\begin{aligned} s_{i,j} = 0 &\Rightarrow \tilde{C}_{i,j} = C_{i,j} && \text{(Original)} \\ s_{i,j} = 1 &\Rightarrow \tilde{C}_{i,j} = \neg C_{i,j} && \text{(Reversed)} \\ s_{i,j} = 2 &\Rightarrow \tilde{C}_{i,j} = \{0\}^{m \times m} && \text{(All background)} \\ s_{i,j} = 3 &\Rightarrow \tilde{C}_{i,j} = \{1\}^{m \times m} && \text{(All foreground)} \end{aligned}$$

Our objective is then to select the class assignments, $s_{i,j} \in \{0, 1, 2, 3\}$, so that the resulting binary masks, $\tilde{C}_{i,j}$, are consistent. We do this by minimizing the following global cost as a function of the class assignments, $S = [s_{i,j}]$ for all i, j,

$$f_1(S) = \sum_{i=1}^{M} \sum_{j=1}^{N} \{\mathcal{E}(s_{i,j}) + \lambda_1 V_1(s_{i,j}, s_{i,j+1}) + \lambda_2 V_2(s_{i,j}, s_{i+1,j}) + \lambda_3 V_3(s_{i,j})\}$$

III. CONNECTED COMPONENT CLASSIFICATION (CCC)

The Connected Component Classification (CCC) algorithm is a Bayesian text detection procedure operating on the binary image produced by COS. The CCC algorithm refines a segmentation by removing false detections (non-text components). The CCC algorithm works by first extracting foreground connected components using a 4-point neighborhood search. Next, a feature vector, y_i, is calculated for the i^{th} connected component (CC_i). Each y_i is a 4 dimensional feature vector which describes aspects of the i^{th} connected component such as edge depth and color uniformity. Each connected component also has a label, x_i, which is 1 if the component is text, and 0 if it is not. The Bayesian segmentation model used for the CCC algorithm is shown in Fig. 2. The conditional distribution of feature vector y_i given x_i is modeled by a Gaussian mixture while the underlying true segmentation labels are modeled by a Markov random field (MRF). The feature vectors are conditionally independent given the class labels x

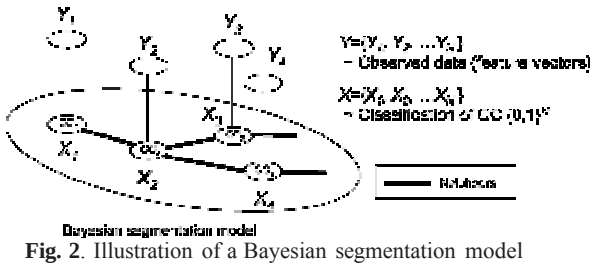

Fig. 2. Illustration of a Bayesian segmentation model

To use a Markov random field model (MRF), we need to define a neighborhood system. To do this, we first find the pixel location at the center of each connected component. Then, for each connected component, we search outward in a spiral pattern until the k nearest neighbors are found. The neighbors of the i^{th} connected component are denoted by ∂I In order to specify the distribution of the MRF, we will first de-fine augmented feature vectors. We define the augmented feature vector for the i^{th} connected component, z_i, consisting of the feature vector y_i concatenated with the horizontal and vertical pixel location.

III NMULTISCALE-COS/CCC SEGMENTATION SCHEME

We incorporate the COS/CCC segmentation algorithm into a multiscale framework [15] in order to improve its accuracy in the detection of text with varying size. The Multiscale-COS/CCC divides the segmentation process into several scales. Each scale is numbered from 0 to $L - 1$, where 0 is the finest scale and $L - 1$ is the coarsest

scale. Segmentation is performed from coarse to fine scales, where the coarser scales use larger block sizes, and the finer scales use smaller block sizes. The segmentation on each scale incorporates results from the previous coarser scale. Both COS and CCC are performed on each scale, however only COS requires adaptation to the multiscale scheme. Equation (9) shows the new cost function used for the n^{th} scale, where $n \in \{0, \ldots, L-1\}$. The term $f^{(n)}$ is defined for each scale according to Eq. (2).

$$f_2(S^{(n)}) = \sum_{i=1}^{M}\sum_{j=1}^{N}\left\{f_1^{(n)} + \lambda_4^{(n)} V_4(s_{i,j}^{(n)}, x^{(n+1)})\right\}$$

The term V_4, is defined as the number of mismatched pixels within the same block between the current layer segmentation $x^{(n)}$ and the previous coarser layer segmentation $x^{(n+1)}$. The exception is that only the pixels that switch from "1" (foreground) to "0" (back- background) are counted when s(n)i,j = 0 or s(n)i,j = 1. This term encourages a more detailed segmentation as we proceed to finer scales. The COS parameter estimation for the multiscale-COS/CCC was also performed in an off-line task. To simplify the optimization process, we first performed optimization to find $\Theta^{(n)} = \{\lambda(n)\ 1 \ldots \lambda(n)3\}$ for each scale. Then, we found the optimimal set of $\Theta = \{\lambda(0)\ 4 \ldots \lambda(L-2)\ 4\}$. The error to be minimized was the number of mismatched pixels compared to ground truth segmentations. All of the parameters in the statistical model were estimated in an off-line training procedure. The parameters of the Gaussian mixture distribution were estimated using the EM algorithm while the num- ber of clusters in each Gaussian mixture for text and non-text were determined using the minimum description length (MDL) estimator by Rissanen [12].We used pseudo likelihood maximization [13, 14] to estimate the prior model parameters, $\varphi = [p, a, b]^T$. In our case, these are given by

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} \prod_{i \in S} p(x_i | x_{\partial i})$$

$$= \underset{\phi}{\operatorname{argmin}} \sum_{i \in S}\left\{\log Z_i + \sum_{j \in \partial i} w_{i,j}\delta(x_i \neq x_j)\right\}$$

where,

$$Z_i = \sum_{x_i \in \{0,1\}} \exp\left\{-\sum_{j \in \partial i} w_{i,j}\delta(x_i \neq x_j)\right\}.$$

IV Proposed technique

In the proposed system the most traditional approach to text segmentation is Otsu's method which thresholds pixels in an effort to divide the document's histogram into objects and background. There are many modified versions of Otsu's method. While Otsu uses a global thresholding approach, Niblack and Sauvola use a local thresholding approach. Kapur's method uses entropy information for the global thresholding, and Tsai uses a moment preserving approach. A comparison of the algorithms for text segmentation can be found. In order to improve text extraction accuracy, some text segmentation approaches also use character properties such as size, stroke width, directions, and run-length histogram. Other binarization approaches for document coding have used rate-distortion minimization as a criteria for document binarization. Many recent approaches to text segmentation have been based upon statistical models. One of the best commercial text segmentation algorithms, which is incorporated in the DjVu document encoder, uses a hidden Markov model (HMM).

Advantages:

- The content used here is a standard framework for layer-based document compression.
- It reduces the bit rate of encoded raster documents.
- The mixed raster content detects and segments the text in complex documents in background gradations.

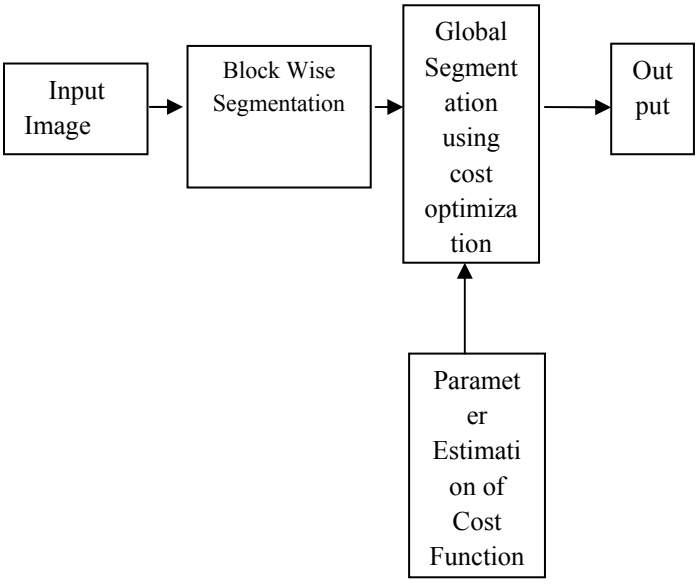

Fig-3 Proposed technique flow diagram

A. The RER Encoder and Decoder

Let X_S be a pixel in the raster document at location s. In the MRC format, each pixel also has an associated fore- ground 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

$$\|X_S - (B_S + \lambda_S (F_S - B_S))\|^2$$

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 $\hat{X}_S = \hat{\lambda}_S F_S + (1 - \hat{\lambda}_S) B_S$. 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. 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 bi- nary 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.

V RESULTS

In this section, we compare multiscale-COS/CCC segmentation results with the results of two existing commercial software packages: Document Expressers ion 5.1 with DjVu[1] and LuraDocument PDF Compressor Desktop [2] . Our comparison is primarily based on two aspects: the segmentation accuracy, and the bitrate resulting from JBIG2 compression of the binary segmentation mask. First, 38 documents were chosen from different document materials, including flyers, newspapers, and magazines. The 17 documents scanned by EPSON STYLUS PHOTO RX700 at 300 dpi were used for the training, and 21 documents scanned by EPSON STYLUS PHOTO RX700, HP Photo smart 3300 All-in-One, and Sam- sung SCX-5530FN at 300 dpi were used to verify the segmentation quality. To evaluate the segmentation accuracy, we measured the percent of missed detections and false detections of segmented *components*, denoted as pMC and pFC . If the total number of correctly detected components is Nd, then we define $\mathrm{pMC} = (\mathrm{Ngt} - \mathrm{Nd})/\mathrm{Ngt}$,where Ngt is the number of text components in the ground truth segmentation. The fraction of false components is defined as $\mathrm{pFC} = \mathrm{Nfa}/\mathrm{Ngt}$, where Nfa is the number of components which are falsely detected. We also measured the percent of missed detections and false detections of individual *pixels*, denoted as pMP and pFP . These numbers were divided by the total number of pixels in the ground truth document. Table 1 shows comparisons of multiscale-COS/CCC, DjVu, and LuraDocument for pMC , pFC , pMP , and pFP . Notice that multiscale-COS/CCC exhibits the lowest error rate in all categories. The qualitative results for a letter size document, text regions, and picture regions are shown in Fig. 2,3, and 4. Black indicates a label of "1" (foreground) and white indicates a label of "0" (background).

Fig. 4. An example of letter size document segmentations in 300 dpi. The enhanced multiscale-COS/CCC exhibits the least missed detections and false detections.

We also compared the bitrate after compression of the binary mask layer generated by multiscale-COS/CCC, DjVu, and LuraDocument in Table 2. For the binary mask compression, we used JBIG2 (defined in ITU-T T.88) encoding as implemented in the Snow Batch, developed by Snowbound Software[3] . JBIG2 is a symbol matching based compression algorithm that works particularly well for documents containing repeated symbols such as text. Notice that the bitrates of multiscale-COS/CCC are similar or lower than DjVu, and substantially lower than LuraDocument. This is likely due to the fact that the multiscale-COS/CCC segmentation has fewer false components than the other algorithms.

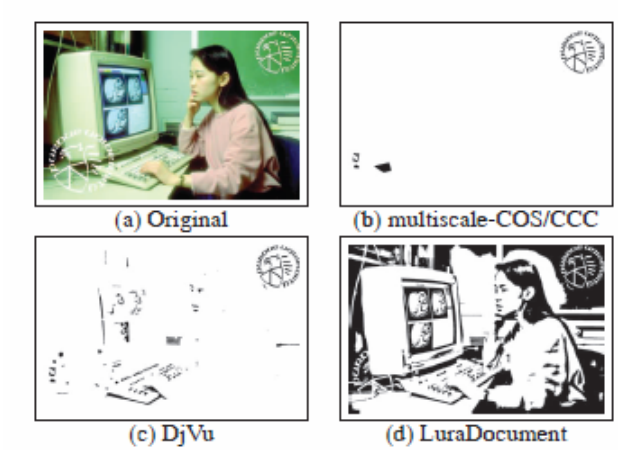

Fig. 5. An example of segmentations in picture regions. The region is 1516 × 1003 pixels in 400 dpi.

EPSON	Multi-COS/CCC	DjVu	LuraDocument
p_{MC}	0.48%	0.61%	4.71%
p_{MP}	0.34%	0.48%	0.76%
p_{FC}	10.3%	13.3%	20.8%
p_{FP}	0.46%	1.06%	6.66%
HP	Multi-COS/CCC	DjVu	LuraDocument
p_{MC}	0.44%	0.62%	5.08%
p_{MP}	0.20%	0.49%	0.68%
p_{FC}	16.1%	18.5%	41.0%
p_{FP}	0.70%	1.20%	6.33%
Samsung	Multi-COS/CCC	DjVu	LuraDocument
p_{MC}	0.49%	0.69%	4.55%
p_{MP}	0.32%	0.44%	0.70%
p_{FC}	8.57%	12.1%	20.3%
p_{FP}	0.50%	0.82%	5.75%

Table 1. Missed component/pixel error and false component/pixel error.

VI. CONCLUSIONS

We presented a segmentation algorithm for the compression of raster documents. While the COS algorithm generates consistent initial segmentations, the CCC algorithm substantially reduces false detections through the use of a component-wise MRF con- text model. The MRF model uses a pair-wise Gibbs distribution which more heavily weights nearby components with similar features. We showed that the multiscale-COS/CCC algorithm achieves greater text detection accuracy with a lower false detection rate, as compared to state-of-the-art commercial MRC products. Such text- only segmentations are also potentially useful for document processing applications such as OCR.

REFERENCE

[1] International Telecommunication Union, *ITU-T recommendation T.44 Mixed raster content (MRC)*, April 1999.

[2] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *IEEE trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[3] 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, 1998.

[4] P. Haffner, L. Bottou, and Y. Lecun, "A general segmentation scheme for DjVu document compression," in *Proc. of ISMM 2002*, Sydney, Australia, April 2002.

[5] Y. Zheng, "Machine printed text and handwriting identification in noisy document images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 337–353, 2004.

[6] S. Kumar, R. Gupta, N. Khanna, S. Chaundhury, and S. D.Joshi, "Text extraction and document image segmentation using matched wavelets and MRF model," *IEEE Trans. on Image Processing*, vol. 16, no. 8, pp. 2117–2128, 2007.

[7] J.G. Kuk, N.I. Cho, and K.M. Lee, "MAP-MRF approach for binarization of degraded document image," in *Proc. of IEEE Int'l Conf. on Image Proc.*, 2008, pp. 2612–2615.

[8] H. Cao and V. Govindaraju, "Processing of low-quality hand- written documents using Markov Random Field," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1184–1194, 2009.

[9] H. Cheng and C. A. Bouman, "Document compression using rate-distortion optimized segmentation," *Journal of Electronic Imaging*, vol. 10, no. 2, pp. 460–474, 2001.

[10] E. Haneda, J. Yi, and C. A. Bouman, "Segmentation for MRC compression," in *Color Imaging XII: Processing, Hardcopy, and Applications*, San Jose, CA, 29th January 2007, vol. 6493.

[11] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy.Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986.

[12] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The annals of Statistics*, vol. 11, no. 2, pp. 417–431, 1983.

[13] J. Besag, "Statistical analysis of non-lattice data," *The Statis- tician*, vol. 24, no. 3, pp. 179–195, 1975.

[14] J. Besag, "Efficiency of pseudo likelihood estimation for simple Gaussian fields," *Biometrika*, vol. 64, no. 3, pp. 616–618,1977.

[15] C. A. Bouman and B. Liu, "Multiple Resolution Segmentation of Textured Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 99–113, 1991