MULTISCALE SEGMENTATION FOR MRC DOCUMENT COMPRESSION USING A MARKOV RANDOM FIELD MODEL

Eri Haneda and Charles A. Bouman

Purdue University
School of Electrical and Computer Engineering
West Lafayette, IN 47907

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 background layers, represented as a binary mask.

In this paper, we propose a novel multiscale segmentation scheme based on the sequential application of two algorithms. The first algorithm, Cost Optimized Segmentation (COS), is a blockwise segmentation algorithm formulated in a global cost optimization framework. 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 combined COS/CCC segmentation algorithms are then incorporated into a multiscale framework in order to improve the segmentation accuracy of text with varying size.

Index Terms— Image segmentation, Markov random field, MRC compression, multiscale image analysis.

1. 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 encoded raster documents. Perhaps the most critical step in MRC encoding 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 recent 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].

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.

2. 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 \( i = 1, \ldots, M \), and \( j = 1, \ldots, N \), 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,

\[
\begin{align*}
    s_{i,j} = 0 & \implies \tilde{C}_{i,j} = C_{i,j} \quad \text{(Original)} \\
    s_{i,j} = 1 & \implies \tilde{C}_{i,j} = \neg C_{i,j} \quad \text{(Reversed)} \\
    s_{i,j} = 2 & \implies \tilde{C}_{i,j} = \{0\}^{m \times m} \quad \text{(All background)} \\
    s_{i,j} = 3 & \implies \tilde{C}_{i,j} = \{1\}^{m \times m} \quad \text{(All foreground)}
\end{align*}
\]

(1)

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 \( i,j \),

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

(2)
3. 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.

3.1. Statistical model

The Bayesian segmentation model used for the CCC algorithm is shown in Fig. 1. 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. 1. Illustration of a Bayesian segmentation model.](image)

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 \( \partial_i \). To ensure all neighbors are mutual (which is required for an MRF), if \( s \in \partial_i \), we add \( r \) as a neighbor of \( s \) if this is not already the case.

In order to specify the distribution of the MRF, we will first define 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 of the connected component’s center. We measure dissimilarity between the two connected components by using the Mahalanobis distance, \( d_{i,j} \), is normalized using the equation

\[
D_{i,j} = \frac{d_{i,j}}{\frac{1}{2}(d_{i,\partial_i} + d_{j,\partial_j})},
\]

where \( d_{i,\partial_i} \) is the averaged distance between \( CC_i \) and all of its neighbors, and \( d_{j,\partial_j} \) is the averaged distance between \( CC_j \) and all of its neighbors. This normalized distance satisfies the symmetric property, that is \( D_{i,j} = D_{j,i} \).

Using the defined neighborhood system, we adopted a MRF model with pair-wise cliques. Let \( P \) be the set of all \( \{i, j\} \) where \( i \) and \( j \) denote neighboring connected components. The labels, \( X \), are assumed to be distributed as

\[
p(x) = \frac{1}{Z} \exp \left\{ - \sum_{\{i,j\} \in P} w_{i,j} \delta(x_i \neq x_j) \right\}
\]

where

\[
w_{i,j} = \frac{b}{D_{i,j}^p + a}.
\]

As we can see, the classification probability is penalized by the number of neighboring pairs which have different classes. The number is also weighted by the term \( w_{i,j} \). If there exists a similar neighbor close to a given component, the term \( w_{i,j} \) becomes large since \( D_{i,j} \) is small. This favors increasing the probability that the two neighbors have the same class. Note that the parameter \( p \) controls the roll-off of the function, and \( a \) and \( b \) control the minimum and transition point for the function.

With the MRF model defined above, we can compute a maximum a posteriori (MAP) estimate to find the optimal set of classification labels \( x = [x_1 x_2 \ldots x_N]^T \). The MAP estimate is then given by,

\[
x_{\text{MAP}} = \arg \min_{x \in \{0,1\}^N} \left\{ - \sum_{i \in S} \log p(y_i | x_i) + \sum_{\{i,j\} \in P} w_{i,j} \delta(x_i \neq x_j) \right\}
\]

To find an approximate solution of (6), we use iterative conditional modes (ICM) [11].

3.2. Parameter estimation

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 number 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 pseudolikelihood maximization [13, 14] to estimate the prior model parameters, \( \phi = [p, a, b]^T \). In our case, these are given by

\[
\hat{\phi} = \arg \max_{\phi} \prod_{i \in S} p(x_i | x_{\partial_i})
\]

\[
= \arg \max_{\phi} \sum_{i \in S} \left\{ \log Z_i + \sum_{j \in \partial_i} w_{i,j} \delta(x_i \neq x_j) \right\},
\]

where

\[
Z_i = \sum_{x_i \in \{0,1\}} \exp \left\{ - \sum_{j \in \partial_i} w_{i,j} \delta(x_i \neq x_j) \right\}.
\]
4. MULTISCALE-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, 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 \( f_{2}^{(n)} \) is defined for each scale according to Eq. (2).

\[
f_{2}(S^{(n)}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \left\{ f_{1}^{(n)}(x_{i,j}^{(n)}, x_{i,j}^{(n+1)}) \right\} \tag{9}
\]

The term \( V_{4} \) is defined as the number of mismatched pixels within the same block between the current layer segmentation \( g^{(n)} \) and the previous coarser layer segmentation \( g^{(n+1)} \). The exception is that only the pixels that switch from “1” (foreground) to “0” (background) are counted when \( s_{i,j}^{(n)} = 0 \) or \( s_{i,j}^{(n)} = 1 \). 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 \( \Theta^{(n)} = \{\lambda_{1}^{(n)}, \ldots, \lambda_{4}^{(n)}\} \) for each scale. Then, we found the optimal set of \( \Theta^{*} = \{\lambda_{1}^{(0)}, \ldots, \lambda_{4}^{(L-2)}\} \). The error to be minimized was the number of mismatched pixels compared to ground truth segmentations.

5. RESULTS

In this section, we compare multiscale-COS/CCC segmentation results with the results of two existing commercial software packages: Document Express version 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 Photosmart 3300 All-in-One, and Samsung 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 \( p_{MC} \) and \( p_{FC} \). If the total number of correctly detected components is \( N_{gt} \), then we define \( p_{MC} = (N_{gt} - N_{d})/N_{gt} \), where \( N_{d} \) is the number of text components in the ground truth segmentation. The fraction of false components is defined as \( p_{FC} = N_{fa}/N_{gt} \), where \( N_{fa} \) 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 \( p_{MP} \) and \( p_{FP} \). 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 \( p_{MC}, p_{FC}, p_{MP} \), and \( p_{FP} \). 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).

---

\(^1\)https://www.celartem.com
\(^2\)https://www.luratech.com

---

Fig. 2. An example of letter size document segmentations in 300 dpi. The multiscale-COS/CCC exhibits the least missed detections and false detections.

Fig. 3. An example of segmentations in text regions. The region is 165 × 370 pixels in 400 dpi.

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 SnowBatch,
developed by Snowbound Software3. 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.

### 6. CONCLUSIONS

We presented a novel 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 context 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.

### 7. REFERENCES


---

3http://www.snowbound.com/