International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 5, May 2020||

A Novel Approach to Text Segmentation for MRC Document Compression

E.Gayathri[1], E.Lahari[2] , J.Madhavi[3], Kanike Vijay Kumar[4]

U.G. Students, Department of Electronics & Communication Engineering, Ravindra College of Engineering for Women, Kurnool, A.P, India[1,2,3]

Assistant Professor, Department of Electronics & Communication Engineering, Ravindra College of Engineering for Women, Kurnool, A.P, India[4]

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 compression is the separation of the document into foreground and background layers, represented as a binary mask. Therefore, the resulting quality and compression ratio of a MRC document encoder is highly dependent upon the segmentation algorithm used to compute the binary mask. In this paper, we pro- pose a novel multiscale segmentation scheme for MRC document encoding based upon the sequential application of two algorithms. The first algorithm, cost optimized segmentation (COS), is a blockwise segmentation algorithm formulated in a global cost opti- mization framework. The second algorithm, connected component classification (CCC), refines the initial segmentation by classifying feature vectors of connected components using an 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. In comparisons to state-of-the-art commercial MRC products and selected segmentation algorithms in the literature, we show that the new algorithm achieves greater accuracy of text detection but with a lower false detection rate of nontext features. We also demonstrate that the proposed segmentation algorithm can improve the quality of decoded documents while simultaneously lowering the bit rate.

KEYWORDS: Document compression, image segmentation, Markov random fields, MRC compression, Multiscale image analysis.

I. INTRODUCTION

With the wide use of networked equipment such as computers, scanners, printers and copiers, it has become more important to efficiently compress, store, and transfer large document files. For example, a typical color document scanned at 300 dpi requires approximately 24 M bytes of storage without compression. While JPEG and JPEG2000 are frequently used tools for natural image compression, they are not very effective for the compression of raster scanned compound documents which typically contain a combination of text, graphics, and natural images. This is because the use of a fixed DCT or wavelet transformation for all content typically results in severe ringing distortion near edges and line-art. 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. The most basic MRC approach, MRC mode 1, divides an image into three layers: a binary mask layer, foreground layer, and background layer. The binary mask indicates the assignment of each pixel to the fore- ground layer or the background layer by a “1” or “0” value, re- spectively. Typically, text regions are classified as foreground while picture regions are classified as background. Each layer is then encoded independently using an appropriate encoder. For example, foreground and background layers may be encoded using traditional photographic compression such as JPEG or JPEG2000 while the binary mask layer may be encoded using symbol-matching based compression such as JBIG or JBIG2. Moreover, it is often the case that different compression ratios and subsampling rates are used for foreground and background layers due to their different characteristics. Typically, the fore- ground layer is more aggressively compressed than the back- ground layer because the foreground layer requires lower color and spatial resolution. Fig. 1 shows an example of layers in an MRC mode 1 document.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 5, May 2020||

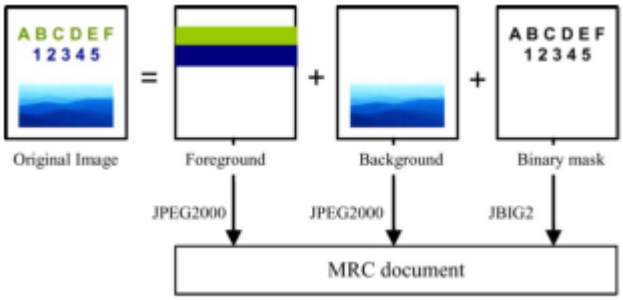


Fig. 1. Illustration of MRC document compression standard mode 1 structure. An image is divided into three layers: a binary mask layer, foreground layer, and background layer. The binary mask indicates the assignment of each pixel to the foreground layer or the background layer by a "1" (black) or "0" (white), respectively. Typically, text regions are classified as foreground while picture regions are classified as background. Each layer is then encoded independently using an appropriate encoder.

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. Segmentation influences both the quality and bitrate of an MRC document. For example, if a text component is not properly detected by the binary mask layer, the text edges will be blurred by the background layer encoder. Alter- natively, if nontext is erroneously detected as text, this error can also cause distortion through the introduction of false edge artifacts and the excessive smoothing of regions assigned to the foreground layer. Furthermore, erroneously detected text can also increase the bit rate required for symbol-based compres- sion methods such as JBIG2. This is because erroneously detected and unstructurenontext symbols are not be efficiently represented by JBIG2 symbol dictionaries.

Many segmentation algorithms have been proposed for ac- curate text extraction, typically with the application of optical character recognition (OCR) in mind. One of the most popular top-down approaches to document segmentation is the – cut algorithm which works by detecting white space using horizontal and vertical projections. The run length smearing algorithm (RLSA) is a bottom-up approach which basically uses region growing of characters to detect text regions, and the Docstrum algorithm proposed in is another bottom-up method which uses -nearest neighbor clustering of connected components. Chen et.al recently developed a multiplane based segmentation method by incorporating a thresholding method. A summary of the algorithms for document segmentation can be found in.

Our segmentation method is composed of two algorithms thatare applied in sequence: the cost optimized segmentation (COS) algorithm and the connected component classification (CCC) algorithm. The COS algorithm is a blockwise segmentation algorithm based upon 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 nontext components (i.e., false text detections) in a Bayesian framework which incorporates an 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 neighbouring 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. The organization of this paper is as follows. In Section II and Section III, we describe COS and CCC algorithms. We also describe the multiscale implementation in Section IV. Section V presents experimental results, in both quantitative and qualitative ways.

II. COS

The COS algorithm is a block-based segmentation algorithm formulated as a global cost optimization problem. The COS al- gorithm is comprised of two components: blockwisesegmenta- tion and global segmentation. The blockwise segmentation di- vides the input image into overlapping blocks and produces an initial segmentation for each block. The global segmentation is then computed from the initial segmented blocks so as to mini- mize a global cost function, which is

carefully designed to favor segmentations that capture text components. The parameters of the cost function are optimized in an offline training procedure. A block diagram for COS is shown in Fig. 2.

A. Block wise Segmentation

Block wise segmentation is performed by first dividing the image into overlapping blocks, where each block contain m x m

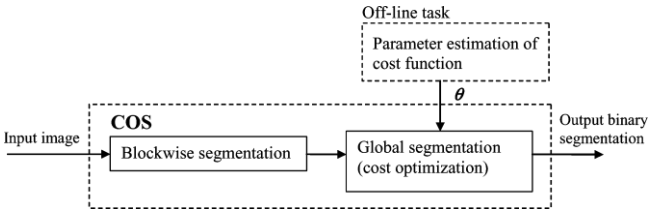


Fig. 2. COS algorithm comprises two steps: blockwise segmentation and global segmentation. The parameters of the cost function used in the global segmentation are optimized in an offline training procedure.

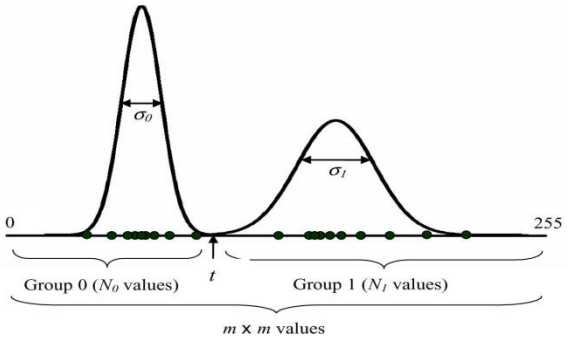


Fig. 3.Illustration of a blockwise segmentation. The pixels in each block are separated into foreground (“1”) or background (“0”) by comparing each pixel with a threshold . The threshold is then selected to minimize the total subclass variance.

The blocks are denoted by O(I,j)for i=1…..M , and j=1……..N, whereMandN are the number of the blocks in the vertical and horizontal directions, respectively. If the height and width of the input image is not divisible by , the image is padded with zeros. For each block, the color axis having the largest variance over the block is selected and stored in a corresponding gray image block, . The pixels in each block are segmented into foreground (“1”) or background (“0”) by the clustering method of Cheng and Bouman [24]. The clustering method classifies each pixel inby comparing it to a threshold . This threshold is selected to minimize the total subclass variance. More specifically, the minimum value of the total subclass variance is given by

$$\gamma_{i,j}^2 = \min_{t\in[0,255]} \frac{N_{0,i,j} * \sigma_{0,i,j}^2 + N_{1,i,j} * \sigma_{1,i,j}^2}{N_{0,i,j} + N_{1,i,j}}$$

where N(0,i,j)andN(1,i,j) are number of pixels classified as 0 and 1 in by the threshold and and are the variances within each subclass (see Fig. 3). Note that the subclass variance can be calculated efficiently. First, we create a histogram by counting the number of pixels which fall into each value between 0 and 255. For each threshold , we can recursively calculate and from the values calculated for the previous threshold of . The threshold that minimizes the subclass variance is then used to produce a binary segmentation of the block.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 5, May 2020||

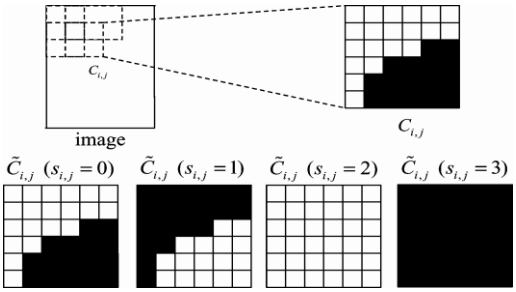


Fig. 4.Illustration of class definition for each block. Black indicates a label of "1" (foreground) and white indicates a label of "0" (background). Four seg- mentation result candidates are defined; original (class 0), reversed (class 1), all background (class 2), and all foreground (class 3). The final segmentation will be one of these candidates. In this example, block size is m=g.

B. Global Segmentation

The global segmentation step integrates the individual segmentations of each block into a single consistent segmentation of the page. To do this, we allow each block to be modified using a class assignment denoted by,

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

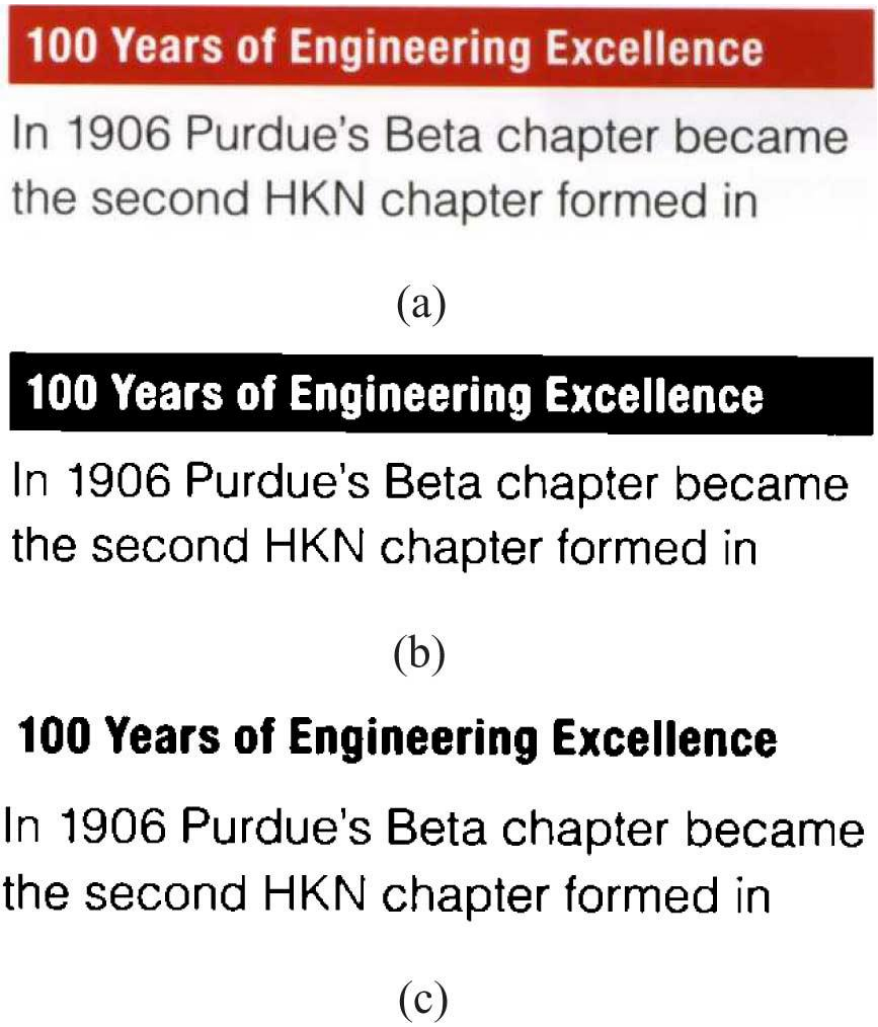
Notice that for each block, the four possible values of correspond to four possible changes in the block's segmentation: original, reversed, all background, or all foreground. If the block class is "original," then the original binary segmentation of the block is retained. If the block class is "reversed," then the assignment of each pixel in the block is reversed (i.e., 1 goes to 0, or 0 goes to 1). If the block class is set to "all background" or "all foreground," then the pixels in the block are set to all 0's or all 1's, respectively. Fig. 4 illustrates an example of the four possible classes where black indicates a label of "1" (foreground) and white indicates a label of "0" (background). Our objective is then to select the class assignments, , so that the resulting binary masks, , are consistent.

We do this by minimizing the following global cost as a function of the class assignments As it is shown, the cost function contains four terms, the first term representing the fit of the segmentation to the image pixels, and the next three terms representing regularizing constraints on the segmentation. The values , , and are then model parameters which can be adjusted to achieve the best segmentation quality. The first term is the square root of the total subclass variation within a block given the assumed segmentation. More specifically where is the standard deviation of all the pixels in the block. Since must always be less than or equal to , the term can always be reduced by choosing a finer segmentation corresponding to or 1 rather than smoother segmentation corresponding to or 3. The terms and regularize the segmentation by penalizing excessive spatial variation in the segmentation. To compute the term , the number of segmentation mismatches between pixels in the overlapping region between block and the horizontally adjacent block is counted. The term is then calculated as the number of the segmentation mismatches divided by the total number of pixels in the overlapping region. Also is similarly defined for vertical mismatches. By minimizing these terms, the segmentation of each block is made consistent with neighboring blocks. The term denotes the number of the pixels classified as foreground (i.e., "1") in divided by the total number of pixels in the block. This cost is used to ensure that most of the area of image is classified as background. For computational tractability, the cost minimization is iteratively performed on individual rows of blocks, using a dynamic programming approach [37]. Note that row-wise approach does not generally minimize the global cost function in one pass through the image. Therefore, multiple iterations are performed from top to bottom in order to adequately incorporate the vertical consistency term. In the first iteration, the optimization of th row incorporates the term containing only the th row. Starting from the second iteration, terms for both the th row and th row are included. The optimization stops when no changes occur to any of the block classes.

Experimentally, the sequence of updates typically converges within 20 iterations. The cost optimization produces a set of classes for overlapping blocks. Since the output segmentation for each pixel is ambiguous due to the block overlap, the

final COS segmentation output is specified by the center region of each overlapping block. The weighting coefficients , , and were found by minimizing the weighted error between segmentation results of training images and corresponding ground truth segmentations. A ground truth segmentation was generated manually by creating a mask that indicates the text in the image. The weighted error criteria which we minimized is given by where , and the terms and are the number of pixels in the missed detection and false detection categories, respectively. For our application, the missed detections are generally more serious than false detections, so we used a value of which more heavily weighted miss detections. In Section III, we The CCC algorithm refines the segmentation produced by COS by removing many of the erroneously detected nontext components. The CCC algorithm proceeds in three steps: connected component extraction, component inversion, and component classification. The connected component extraction step identifies all connected components in the COS binary segmentation using a 4-point neighborhood. In this case, connected components less than six pixels were ignored because they are nearly invisible at 300 dpi resolution.

The component inversion step corrects text segmentation errors that sometimes occur in COS segmentation when text is locally embedded in a highlighted region [see Fig. 5(a)]. Fig. 5(b) illustrates this type of error where text is initially segmented as background. Notice the text “100 Years of Engineering Excellence” is initially segmented as background due to the red surrounding region. In order to correct these errors, we first detect foreground components that contain more than eight interior background components (holes). In each case, if the total number of interior background pixels is less than half of the surrounding foreground pixels, the foreground and background assignments are inverted. Fig. 5(c) shows the result of this inversion process. Note that this type of error is a rare occurrence in the COS segmentation.



| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 5, May 2020||

The final step of component classification is performed by extracting a feature vector for each component, and then computing a MAP estimate of the component label. The feature vector, , is calculated for each connected component, , in the COS segmentation. Each is a 4-D feature vector which describes aspects of the connected component including edge depth and color uniformity. Finally, the feature vector is used to determine the class label, , which takes a value of 0 for nontext and 1 for text. The Bayesian segmentation model used for the CCC algorithm is shown in Fig. 6. The conditional distribution of the feature vector given is modeled by a multivariate Gaussian mixture while the underlying true segmentation labels are modeled byan MRF. Using this model, we classify each component by calculating the MAP estimate of the labels, , given the featurevectors, . In order to do this, we first determine which components are neighbours in the MRF. This is done based upon the geometric distance between components on the page.

A. Statistical Model

Here, we describe more details of the statistical model used for the CCC algorithm. The feature vectors for "text" and "non-text" groups are modeled as -dimensional multivariate Gaussian mixture distributions

$$p(y_i|x_i) = \sum_{m=0}^{M_{x_i}-1} \frac{a_{x_i,m}}{(2\pi)^{D/2}} |R_{x_i,m}|^{-1/2} \times \exp\left\{-\frac{1}{2}(y_i - \mu_{x_i,m})^t R_{x_i,m}^{-1}(y_i - \mu_{x_i,m})\right\}$$

The components of the feature vectors include measurements of edge depth and external color uniformity of the th connected component. The edge depth is defined as the Euclidean distance between RGB values of neighboring pixels across the component boundary (defined in the initial COS segmentation). The color uniformity is associated with the variation of the pixels outside the boundary. In this experiment, we defined a feature vector with four components, , where the first two are mean and variance of the edge depth and the last two are the variance and range of external pixel values.

III. MORKOV RANDOM FIELD MODEL

In the domain of physics and probability, a Markov random field (often abbreviated as MRF), Markov network or undirected graphical model is a set of random variables having a Markov property described by an undirected graph. A Markov random field is similar to a Bayesian network in its representation of dependencies; the differences being that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); on the other hand, it can't represent certain dependencies that a Bayesian network can (such as induced dependencies).

When the probability distribution is positive, it is also referred to as a Gibbs random field, because, according to the Hammersley–Clifford theorem, it can then be represented by a Gibbs measure. The prototypical Markov random field is the Ising model; indeed, the Markov random field was introduced as the general setting for the Ising model. In the domain of artificial intelligence, a Markov random field is used to model various low- to mid-level tasks in image processing and computer vision.[2] For example, MRFs are used for image restoration, image completion, segmentation, texture synthesis, super-resolution and stereo matching.

Given an undirected graph G = (V, E), a set of random variables X = (Xv)v ∈ V indexed by V form a Markov random field with respect to G if they satisfy the following equivalent Markov properties:

Pairwise Markov property: Any two non-adjacent variables are conditionally independent given all other variables:

$$X_u \perp\!\!\!\perp X_v | X_{V\setminus\{u,v\}} \quad \text{if } \{u,v\} \notin E$$

Local Markov property: A variable is conditionally independent of all other variables given its neighbours:

$$X_v \perp\!\!\!\perp X_{V\setminus \mathrm{cl}(v)} | X_{\mathrm{ne}(v)}$$

where ne(v) is the set of neighbours of v, and cl(v) = {v} ∪ ne(v) is the closed neighbourhood of v.

Global Markov property: Any two subsets of variables are conditionally independent given a separating subset:

$$X_A \perp\!\!\!\perp X_B | X_S$$

where every path from a node in A to a node in B passes through S.

As the Markov properties of an arbitrary probability distribution can be difficult to establish, a commonly used class of Markov random fields are those that can be factorized according to the cliques of the graph.

Given a set of random variables X = (Xv)v ∈ V, let P(X=x) be the probability of a particular field configuration x in X. That is, P(X=x) is the probability of finding that the random variables X take on the particular value x. Because X is a set, the probability of x should be understood to be taken with respect to a product measure, and can thus be called a joint density.

If this joint density can be factorized over the cliques of G:

$$P(X = x) = \prod_{C \in \mathrm{cl}(G)} \phi_C(x_C)$$

then X forms a Markov random field with respect to G. Here, cl(G) is the set of cliques of G. The definition is equivalent if only maximal cliques are used. The functions φC are sometimes referred to as factor potentials or clique potentials. Note, however, conflicting terminology is in use: the word potential is often applied to the logarithm of φC. This is because, in statistical mechanics, log(φC) has a direct interpretation as the potential energy of a configuration xC.

Although some MRFs do not factorize (a simple example can be constructed on a cycle of 4 nodes), in certain cases they can be shown to be equivalent conditions:

if the density is positive (by the Hammersley–Clifford theorem),

If the graph is chordal (by equivalence to a Bayesian network).

When such a factorization does exist, it is possible to construct a factor graph for the network.

Any Markov random field (with a strictly positive density) can be written as log-linear model with feature functions f_k such that the full-joint distribution can be written as

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right)$$

where the notation

$$w_k^\top f_k(x_{\{k\}}) = \sum_{i=1}^{N_k} w_{k,i} \cdot f_{k,i}(x_{\{k\}})$$

is simply a dot product over field configurations, and Z is the partition function:

$$Z = \sum_{x \in \mathcal{X}} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right).$$

Here, $\mathcal{X}$ denotes the set of all possible assignments of values to all the network's random variables. Usually, the feature functions $f_{k,i}$ are defined such that they are indicators of the clique's configuration, i.e. $f_{k,i}(x_{\{k\}}) = 1$ if $x_{\{k\}}$

corresponds to the i-th possible configuration of the k-th clique and 0 otherwise. This model is equivalent to the clique factorization model given above, if is the cardinality of the clique, and the weight of a feature corresponds to

$$N_k = \left|dom(C_k)\right|$$

the logarithm of the corresponding clique factor, i.e. $w_{k,i} = \log \phi(c_{k,i})$, where $c_{k,i}$ is the i-th possible configuration of the k-th clique, i.e. the i-th value in the domain of the clique C_k.

The probability P is often called the Gibbs measure. This expression of a Markov field as a logistic model is only possible if all clique factors are non-zero, i.e. if none of the elements of $\mathcal{X}$ are assigned a probability of 0. This allows techniques from matrix algebra to be applied, e.g. that the trace of a matrix is log of the determinant, with the matrix representation of a graph arising from the graph's incidence matrix.

The importance of the partition function Z is that many concepts from statistical mechanics, such as entropy, directly generalize to the case of Markov networks, and an intuitive understanding can thereby be gained. In addition, the partition function allows variational methods to be applied to the solution of the problem: one can attach a driving force to one or more of the random variables, and explore the reaction of the network in response to this perturbation. Thus, for example, one may add a driving term Jv, for each vertex v of the graph, to the partition function to get:

$$Z[J] = \sum_{x \in \mathcal{X}} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}}) + \sum_v J_v x_v\right)$$

Formally differentiating with respect to Jv gives the expectation value of the random variable Xv associated with the vertex v:

$$E[X_v] = \frac{1}{Z} \left.\frac{\partial Z[J]}{\partial J_v}\right|_{J_v=0}.$$

Correlation functions are computed likewise; the two-point correlation is:

$$C[X_u, X_v] = \frac{1}{Z} \left.\frac{\partial^2 Z[J]}{\partial J_u \partial J_v}\right|_{J_u=0, J_v=0}.$$

Log-linear models are especially convenient for their interpretation. A log-linear model can provide a much more compact representation for many distributions, especially when variables have large domains. They are convenient too because their negative log likelihoods are convex. Unfortunately, though the likelihood of a logistic Markov network is convex, evaluating the likelihood or gradient of the likelihood of a model requires inference in the model, which is in general computationally infeasible.

To use an MRF, we must define a neighborhood system. To do this, we first find the pixel location at the center of mass for each connected component. Then for each component we search outward in a spiral pattern until the nearest neighbors are found. The number is determined in an offline training process along with other model parameters. We will use the symbol to denote the set of neighbors of connected component . To ensure all neighbors are mutual (which is required for an MRF), if component is a neighbor of component (i.e.,), we add component to the neighbor list of component (i.e.,) if this is not already the case. In order to specify the distribution of the MRF, we first define augmented feature vectors. The augmented feature vector, , for the th connected component consists of the feature vector concatenated with the horizontal and vertical pixel location of the connected component's center. We found the location of connected components to be extremely valuable contextual information for text detection. For more details of the augmented feature vector, Next, we define a measure of dissimilarity between connected components in terms of the Mahalanobis distance of the augmented feature vectors given by

$$d_{i,j} = \sqrt{(z_i - z_j)^T \Sigma^{-1} (z_i - z_j)}$$

Where sigma is the covariance matrix of the augmented feature vectors on training data. Next, the Mahalanobis distance, , is normalized using the equations

$$D_{i,j} = \frac{d_{i,j}}{\frac{1}{2}(\bar{d}_{i,\partial i} + \bar{d}_{j,\partial j})}$$
$$\bar{d}_{i,\partial i} = \frac{1}{|\partial i|} \sum_{k \in \partial i} d_{i,k}$$
$$\bar{d}_{j,\partial j} = \frac{1}{|\partial j|} \sum_{k \in \partial j} d_{j,k}$$

where is averaged distance between the connected component and all of its neighbors, and is similarly defined. This normalized distance satisfies the symmetry property, that is . Using the defined neighborhood system, we adopted an MRF model with pair-wise cliques. Let be the set of all where and denote neighboring connected components. Then, the are assumed to be distributed as where is an indicator function taking the value 0 or 1, and are scalar parameters of the MRF model. As we can see, the classification probability is penalized by the number of neighboring pairs which have different classes. This number is also weighted by the term . If there exists a similar neighbor close to a given component, the term becomes large since

IV. MULTISCALE-COS/CCC SEGMENTATION SCHEME

In order to improve accuracy in the detection of text with varying size, we incorporated a multiscale framework into the COS/CCC segmentation algorithm. The multiscale framework allows us to detect both large and small components by combining results from different resolutions. Since the COS algorithm uses a single block size (i.e., single scale), we found that large blocks are typically better suited for detecting large text, and small blocks are better suited for small text. In order to improve the detection of both large and small text, we use a multiscale segmentation scheme which uses the results of coarse-scale segmentations to guide segmentationon finer scales. Note that both COS and CCC segmentations are per formed on each scale, however, only COS is adapted to the multiscale scheme.

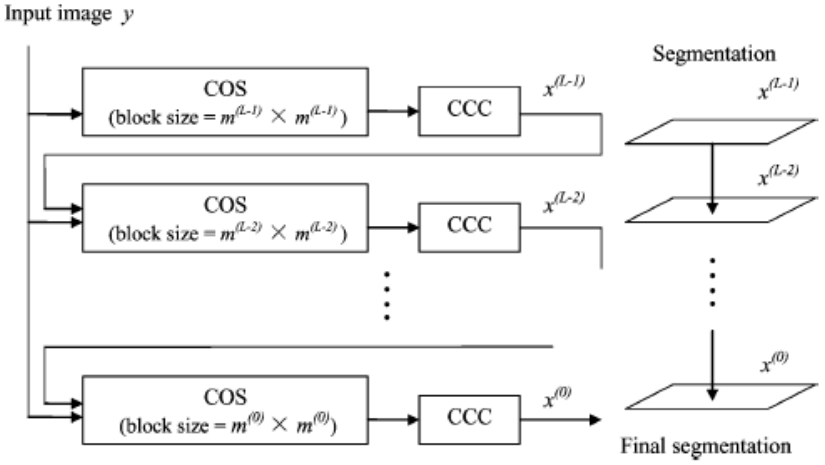


Fig. 8 shows the overview of our multiscale-COS/CCC scheme.

In the multiscale scheme, segmentation progresses from coarse to fine scales, where the coarser scales use larger block sizes, and finer scales use smaller block sizes. Each scale is numbered from L-1 to 0, where L-1 is the coarsest scale and 0 is the finest scale. The COS algorithm is modified to use different block sizes for each scale (denoted as), and incorporates the previous coarser segmentation result by adding a new term to the cost function.

The new cost function for the multiscale scheme is shown in (16). It is a function of the class assignments on both the th And n+1 th scale As it is shown in the equation, this modified cost function incorporates a new term that makes the segmentation consistent with the previous coarser scale. The term , is defined as the number of mismatched pixels within the block between the current scale segmentation and the previous coarser scale segmentation . The exception is that only the pixels that switch from "1" (foreground) to "0" (background) are counted when or . This term encourages a more detailed segmentation as we proceed to finer scales. The term is normalized by dividing by the block size on the current scale. Note that the term is ignored for the coarsest scale. Using the new cost function, we find the class assignments, , for each scale.

V. RESULTS

In this section, we compare the multiscale-COS/CCC, COS/CCC, and COS segmentation results with the results of two popular thresholding methods, an MRF-based segmentation method, and two existing commercial software packages which implement MRC document compression. The thresholding methods used for the comparison in this study are Otsu and Tsai methods. These algorithms showed the best segmentation results among the thresholding methods. In the actual comparison, the RGB color image was first converted to a lumagrayscaleimage, then each thresholding method was applied. For "Otsu/CCC" and "Tsai/CCC," the CCC algorithm was combined with the Otsu and Tsai binarization algorithms to remove false detections. In this way, we can compare the end result of the COS algorithm to alternative thresholding approaches.

The MRF-based binary segmentation used for the comparison is based upon the MRF statistical model developed by Zheng and Doermann . The purpose of their algorithm is to classify each component as either noise, hand-written text, or machine printed text from binary image inputs. Due to the complexity of implementation, we used a modified version of the CCC algorithm incorporating their MRF model by simply replacing our MRF classification model by their MRF noise classification model. The multiscale COS algorithm was applied without any change. The clique frequencies of their model were calculated through offline training using a training data set. Other parameters were set as proposed in the project. We also used two commercial software packages for the comparison. The first package is the DjVu implementation contained in Document Express Enterprise version 5.1. DjVu is commonly used software for MRC compression and produces excellent segmentation results with efficient computation. By our observation, version 5.1 produces the best segmentation quality among the currently available DjVu packages. The second package is Lura Document PDF Compressor, Desktop Version. Both software packages extract text to create a binary mask for layered document compression.

The performance comparisons are based primarily upon two aspects: the segmentation accuracy and the bitrate resulting from JBIG2 compression of the binary segmentation mask. We show samples of segmentation output and MRC decoded images using each method for a complex test image. Finally, we list the computational run times for each method.

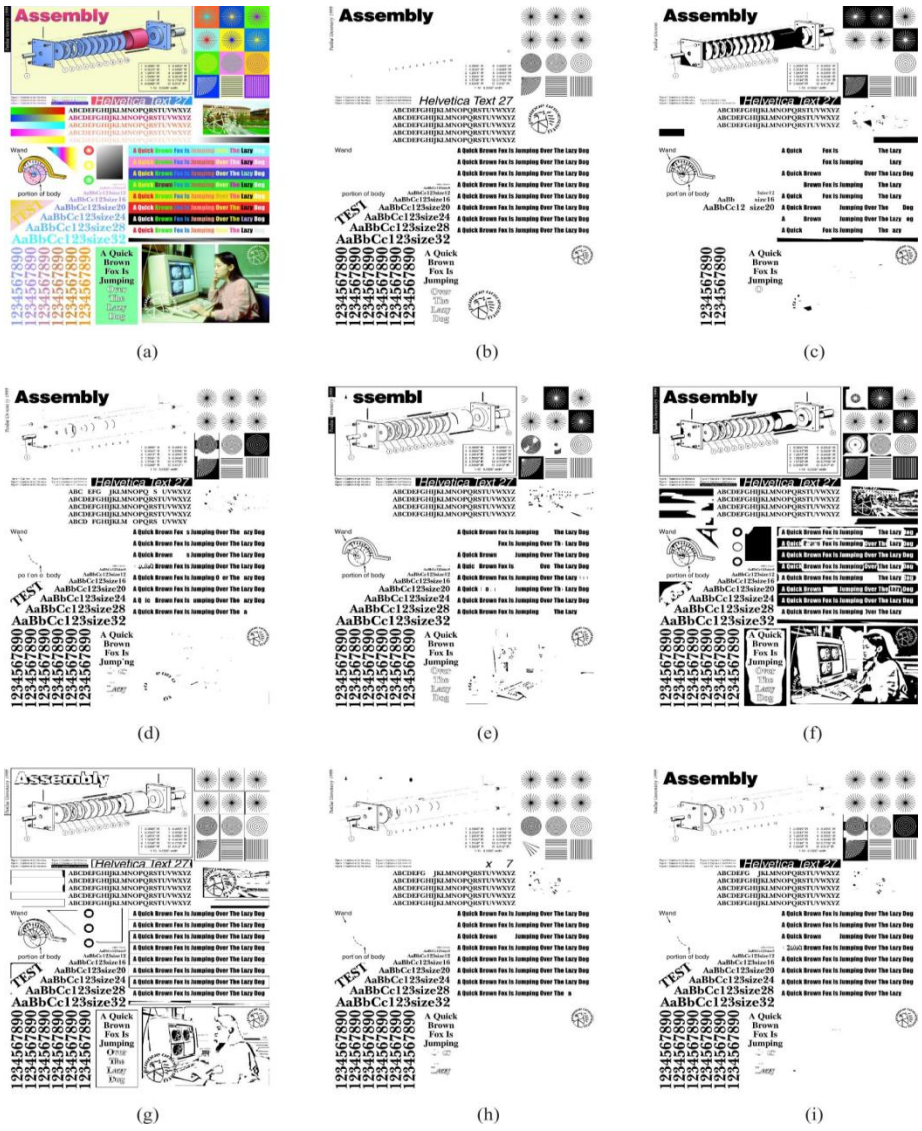


Fig.12.Picture regions inthebinarymask.Picture regionis15161003pixelsat400dpi,whichcorrespondsto9.63cm 6.35cm.(a)Originaltestimage. (b)Groundtruthsegmentation.(c)Otsu/CCC.(d)Multiscale-COS/CCC/Zheng. (e)DjVu.(f)LuraDocument.(g)COS.(h)COS/CCC.(i)Multiscale-COS/CCC.

VI. CONCLUSION

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 multi scale-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

REFERENCES

[1] ITU-T Recommendation T.44 Mixed Raster Content (MRC), T.44, In- ternational Telecommunication Union, 1999.
[2] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document imageanalysis system for technical journals," Computer, vol. 25, no. 7, pp.10–22, 1992.
[3] K. Y. Wong and F. M. Wahl, "Document analysis system," IBM J. Res.Develop., vol. 26, pp. 647–656, 1982.
[4] J. Fisher, "A rule-based system for document image segmentation," inProc. 10th Int. Conf. Pattern Recognit., 1990, pp. 567–572.
[5] L. O'Gorman, "The document spectrum for page layout anal- ysis," IEEE Trans. Pattern Anal Mach. Intell., vol. 15, no. 11, pp.1162–1173, Nov. 1993.
[6] Y. Chen and B. Wu, "A multi-plane approach for text segmentation of complex document images," Pattern Recognit., vol. 42, no. 7, pp.1419–1444, 2009.
[7] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed.Upper Saddle River, NJ: Pearson Education, 2008.
[8] F. Shafait, D. Keysers, and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," IEEE Trans. Pat-tern Anal. Mach. Intell., vol. 30, no. 6, pp. 941–954, Jun. 2008.
[9] K. Jung, K. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," Pattern Recognit.,vol. 37, no. 5, pp. 977–997,2004.
[10] G. Nagy, "Twenty years of document image analysis in PAMI," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 22, no. 1, pp. 38–62, Jan. 2000.

BIBLIOGRAPHY

MsE.GAYATHRI studying herFinal year B.Tech degree inRavindra College of Engineering For Women. Kurnool , Affiliated to J.N.T.U.Ananthapuramu, Andhra Pradesh A.P. Her current interested area is Digital Image Processing andCommunications.

Ms E.LAHARI studying her Final year B.Tech degree inRavindra College of Engineering For Women.Kurnool , Affiliated to J.N.T.U. Ananthapuramu, Andhra Pradesh A.P. Her current interested area is Digital Image Processing andCommunications.

Ms J.MADHAVI studying her Final year B.Tech degree inRavindra College of Engineering For Women. Kurnool , Affiliated to J.N.T.U. Ananthapuramu, Andhra Pradesh A.P. Her current interested area is Digital Image Processing andCommunications.

Mr. KANIKE VIJAYKUMAR received hisB.Tech degree inElectronics andCommunicationEngineering from SJCET,Yemmiganur, Affiliated to J.N.T.U.A Ananthapuram,A.P in 2009 and M.Tech. Post Graduate inCommunication and Signal Processing in GPREC,Kurnool, JNTU Ananthapuram, A.P. in 2012. Currently He is working as Assistant Professor inRavindra College of Engineering For Women,Kurnool, A.P., India. He published Nine technicalpapers in various International journal and Nationalconferences. His current research interests include Microwave Signal, Digital Image Processing andCommunications.