Document Image Segmentation as a Spectral Partitioning Problem

Praveen Dasigi, Raman Jain and C V Jawahar
Center for Visual Information Technology, IIIT Hyderabad,
Gachibowli, Hyderabad-500032, India
{praveend@research.,ramanjain@students.,jawahar@}iiit.ac.in

Abstract

State of art document segmentation algorithms employ adhoc solutions which use some document properties and iteratively segment the document image. These solutions need to be adapted frequently and sometimes fail to perform well for complex scripts. This calls for a generalized solution that achieves a one shot segmentation that is globally optimal. This paper describes one such solution based on the optimization problem of spectral partitioning which makes the decision of proper segmentation based on the spectral properties of the pairwise similarity matrix. The solution described in the paper is shown to be general, global and closed form. The claims have been demonstrated on 142 page images from a Telugu book, in a script set in both poetry and prose layouts. This particular class of scripts has been proved to be challenging for the existing state of the art algorithms, where the proposed solution achieves significant results.

1. Introduction

The objective of a document image segmentation algorithm is to partition a given image into semantically coherent layout units. The output of this process is used as input to many applications including optical character recognition (OCR) systems. Given an input document image and the connected components within, most algorithms employ either top down(divisive) [1, 2, 3] or bottom up(agglomerative) [4, 5, 6] approaches to segment it into text lines and words. The current state of art implementations make use of one or more of the document properties, and strategically set thresholds to make the appropriate decisions in the segmentation process. Some of these algorithms perform reasonably well for a wide class of documents [7]. These solutions are in general greedy and sub-optimal. The local decisions made in the segmentation process need not even relate to the optimization of a meaningful global objective function. From this point of view, these algorithms are highly heuristic in nature. However, in recent years, segmentation of natural images have been successfully formulated and solved as discrete or continuous optimization problem using tools like graph cuts. Normalized cuts [8] and associated literature use a pairwise similarity matrix between pixels or pixel groups to identify the optimal partition. The indicator vector which provides important information as to where the optimal “cut” should lie is a solution to the generalized eigenvector problem over the Laplacian of the pairwise similarity matrix [9]. This class of segmentation algorithms, iterative or otherwise, optimizes a global objective function to obtain a meaningful segmented description. However, the existing document segmentation algorithms lack in such an optimization framework that can achieve meaningful and analyzable segmented descriptions. This aspect can be addressed if the objective function can be formulated in terms of pairwise similarities of connected components in a document image.

The lack of a well structured solution for document segmentation problems becomes even more pertinent for documents in Perso-Arabic or Dravidian scripts. These scripts contain characters which comprise of more than one connected components. They have characters of varying heights with dangling modifiers. Thus most algorithms that use assumptions based on Manhattan layouts or nearest neighbor assignments fail grossly [10] in such cases. Traditional solutions have to be frequently modified for each script, language and class of documents. This paper proposes a generalized solution that optimizes an objective function derived out of multiple clues from document images. The objective function is flexible in the sense that it provides a facility to bring in as many visual clues as needed for a particular class of document. This is done by modelling the visual clues as a single proximity matrix built from a linear combination of multiple proximity matrices. Further the segmentation uses a spectral partitioning approach that tries to maximize the proximities within the partitions while minimizing the proximities across them.

2. Background

A recent paper [11] argues about the required attention towards the seemingly under-addressed problem of Indian language document segmentation. Particular ex-

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.96

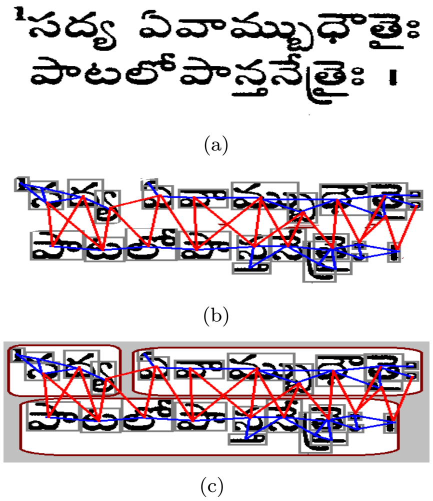
tra emphasis is needed in light of the explosive growth of digital content in Indian scripts. Distribution of connected components in many of the Indian scripts vary widely from that of English. For complex scripts such as those of Dravidian languages, components of character in a line could drift vertically away from the line. This causes the ambiguity for nearest neighbor metrics based on whitespaces or gutters [7, 10]. The top down approaches consider the whole document image as the input and use properties such as gutters, whitespaces or word boundaries to partition hierarchically. One such approach is the recursive XY cut method [1], which is a tree-based top-down algorithm. It splits the document at each level into two or more segments and computes the horizontal and vertical projection profiles at each level. Due to the problem of dangling modifiers for scripts such as Telugu, and Urdu, the projection profile analysis based methods fail since the modifiers obstruct the gutters frequently resulting in merging of text lines. A similar class of algorithms is the constrained textline detection [2] and the whitespace analysis [3] algorithms. They use the maximal whitespace rectangles between components to analyze the textline boundaries. These approaches are also affected by the dangling modifiers which result in poor maximal whitespace covers. Bottom up approaches consider the local neighborhood of components to group them at that granularity. Runlength smearing [5], docstrum [6] and Voronoi diagram based segmentation [4] are some bottom up approaches which use the nearest neighbor based grouping of components. These approaches are also affected by the dangling modifiers. This is because of the fundamental premise of nearest neighbor assignment, which will merge the modifiers with the nearest line instead of the semantically appropriate one. This problem of dangling modifiers have been addressed by using an improved variant of textline detection that employs learning of script priors for proper modifier assignment [12].

In the area of general segmentation, the method of partitioning the image observing the spectral properties has been in popularity in the recent past. This class of algorithms computes a pairwise similarity matrix built over every pair of components (pixels) from the image. The idea is to find an indicator vector from the spectrum of this matrix which can be thresholded to partition the set. A discussion on the applicability of eigenvectors can be found in [9]. The normalized cuts approach [8] is a pioneering work that uses a criterion function which defines the indicator vector used for segmentation as a solution for a generalized eigenvalue problem. Another approach to solve the perceptual grouping problem is to factorize the similarity matrix based on the spectral properties which is proposed in [13]. Though they share common targets, document segmentation departs from this class of natural scene segmentation problems in terms of the performance requirements. What is liberally evaluated in terms of abstract semantic coherence for natural images is a critical necessity when dealing with document images. This is the reason for the huge divide in the formulations for either class.

Popular document segmentation algorithms are shown to be performing reasonably well for documents in Latin scripts. They differ in the visual clue employed for imposing the partitions. Many of them have complementary capabilities. This calls for a framework which can effectively use these complementary document properties to obtain accurate segmentation for documents in complex scripts. The framework will have to be more than just a loosely coupled application of two or more algorithms. The framework should be able to give flexibility for the user to provide a different set of coupling parameters for a class of documents without involving any major design decisions The document signatures to be captured could include the local or global geometry metrics gathered based on the foreground components and whitespaces. Each particular script has a set of properties that provide prior information as to which parameter governs the likelihood of two components falling in the same partition. One such can be the observed class of a modifier that governs the assignment of a dangling modifier to a particular neighbor. In most cases the association of a pair of components due to script specific properties be learned from different labellings. This information will be another factor in the decision making process. The framework should be such that it should allow the possibility of modelling an exhaustive set of such characteristics (optimization parameters) into the decision making process. The next two sections outline such a formulation which satisfies all the requirements identified in this section. The developed algorithm will be shown perform significantly on text-block layouts with complex scripts similar to the results demonstrated in [7]. It can be observed that an extension to text-graphics separation can be easily formulated in the framework proposed

3. Partitioning Problem

The document image which is the input to the segmentation problem is a set of connected components C similar to Fig. **1(a)**. For every such image there will be a set of parameters that can be extracted, which are prominent visual cues that provide an estimate of the document structure. The state of the art algorithms often use a subset of these parameters to judge an optimal partition. For every pair of components in the image $\{c_i, c_j\}$, one can estimate a confidence level or a proximity p_{ij}. This level determines the likelihood of the pair of components belonging to the same text block. p_{ij} is a linear combination of proximities θ_k obtained from each document parameter The full proximity matrix P is a $|C| \times |C|$ matrix, where a row i will contain measures of how likely is component c_i to lie in the same block as every other com-


(a)

(b)

(c)

Figure 1. (a).An example document image fragment to be segmented, with non-manhattan layout (b).A graph constructed over the fragment(c).Ideal partition

ponent. For a typical document image, the parameters that determine the confidence levels are the following

- θ_1 **Euclidean distance**:Distance between centers of bounding boxes of connected components which is inversely proportional to the confidence level
- θ_2 **Co-occurrence probability**:Models the conditional occurrence of a component with respect to another component or another parameter This is useful in the case of Indian language documents since there will be trailing modifiers occurring jointly with main components
- θ_3 **Whitespace area**:The area and shape of the whitespace between two components. The confidence is inversely proportional to the area of the maximal whitespace cover
- θ_4 **Gutter area**: For pairs of components across lines the gutter area between them is inversely proportional to the confidence level of the pair being in one partition
- θ_5 **Global geometry boosts**: An extra factor for enhancing the proximities based on the k-level neighborhood of a component

This approach can be formalized in a graph theoretic framework where the components are a set of points in arbitrary space represented in a weighted undirected graph. In this graph $G : \langle V, E \rangle$, the nodes are the components in the feature space and each edge is labeled with a weight that is a linear combination of proximities obtained from different document parameters. The whole setup is represented in a proximity matrix P which is similar to the adjacency matrix.

$$P = p_1\theta_1 + p_2\theta_2 + p_3\theta_3 + p_4\theta_4 + p_5\theta_5$$

where $p_1 \ldots p_5$ represent the mixing parameters. For segmentation, we seek to partition the components C into two disjoint sets say, P_A and P_B such that the overall confidence level within a partition is maximum and that across partitions is minimum. Let the overall confidence of each partition be $conf(P_A)$ and $conf(P_B)$, and the distance between partitions be $dist(P_A, P_B)$. The best partition is where the distance is minimized with the maximum values of confidence for each of the partition. Thus the following criterion should be minimized to achieve the optimal document segmentation $DocSeg(C)$

$$DocSeg(C) = \arg\min_{P_A,P_B} \left[dist(P_A, P_B) \left(\frac{1}{conf(P_A)} + \frac{1}{conf(P_B)} \right) \right] \tag{1}$$

where $conf(P_A) = \sum_{\{i \in P_A, j \in C\}} P_{ij}$ and $conf(P_B) = \sum_{\{i \in P_B, j \in C\}} P_{ij}$. The distance between partitions is $dist(P_A, P_B) = \sum_{\{i \in P_A, j \in P_B\}} P_{ij}$, which is nothing but the proximity across the partitions.

4. Optimization Framework

The problems of data clustering and image segmentation have been dealt as spectral partitioning problems [13, 9]. In this area there have been approaches using different indicator vectors for spectral partitioning. For example [13] uses the eigenvector corresponding to the largest eigenvalue of the standard eigenvalue problem. The normalized cuts approach [8] uses the Fiedler vector as the indicator vector to partition the images based on algebraic connectivity. This section aims to formulate the document segmentation problem as one class of spectral partitioning problems based on the optimal partitioning criterion introduced in the previous section

4.1. Geometry and Spectral Partitioning

The geometry of the document is modeled in the proximity matrix P. The advantage of the matrix is that the decision of optimal partition can be made from every parameter that contributes to the likelihood of a pair of components. This way both the local geometry and the global properties are aptly represented. From equation (1), the overall confidence of a partition represented by $conf(P_A)$ is the sum of proximities between all elements in partition P_A and all elements in the total component set C. Since the proximity matrix P is the adjacency matrix for the weighted undirected graph representing the document, $conf(P_A)$ can be called the volume of the partition $vol\ P_A$. $dist(P_A, P_B)$ is called the edge boundary with respect to the partitions $E(P_A, P_B)$. Thus the criterion turns out to be,

$$DocSeg(C) = \underset{P_A, P_B}{\arg\min}\ E(P_A, P_B)\left(\frac{1}{vol\ P_A} + \frac{1}{vol\ P_B}\right) \tag{2}$$

In this graph framework, the problem of finding the partitions with the maximum volume is called the isoperimetric problem. An isoperimetric measure is the Cheeger constant of the graph which addresses the question of how to find the optimal partitions P_A and P_B with the maximum edges (volume), such that the edge boundary $E(P_A, P_B)$ contains as few edges as possible (Cost of partition). The Cheeger constant for this graph can be defined as

$$H_C(P_A, P_B) = \underset{P_A, P_B}{\arg\min} \frac{E(P_A, P_B)}{min(vol\ P_A, vol\ P_B)} \tag{3}$$

Let the ideal partition be represented by an indicator vector $\mathbf{x}$ of size $|C|$. It is built such that $x_i = 1$ if $\{c_i \in P_A\}$ and $x_i = -1$ if $\{c_i \in P_B\}$. The indicator vector is the proper identification of the threshold of where the algorithm is supposed to partition the graph into two disjoint sets with the $DocSeg(C)$ criterion being minimum. Let D be the diagonal matrix representing the degree of each node in the graph on its diagonal, i.e., the sum of proximities for each component. The Laplacian matrix of the graph Q is defined as $Q = D - P$. The Laplacian of a graph is positive semidefinite, which means that the eigenvalues are non-negative. Let $k = \frac{\sum_{x_i > 0} d_i}{\sum_{x_i} d_i}$, then $b = \frac{k}{1-k}$ represents the ratio of the volumes, which in the most ideal case should be 1.

¿From the criterion function $DocSeg(C)$, the minimum value is obtained when the partitions are of equal sizes, i.e., $vol\ P_A = vol\ P_B$. So,

$$\begin{aligned} DocSeg(C) &\geq E(P_A, P_B)\frac{2}{vol(P_A)} \\ &\geq 2H_C(P_A, P_B) \end{aligned}$$

In the discussion of Cheeger constants in [14], it has been proved that, for the proper indicator vector x the Cheeger constant of a graph G is bounded by the second lowest Eigenvalue λ_1 of its Laplacian by the inequality, $2H_G \geq \lambda_1$.

$$\underset{x}{\arg\min}\ DocSeg(x) = \lambda_1 \tag{4}$$

The problem of spectral partitioning by normalized cuts has used the same indicator vector with a similar criterion function in [8]. They have proved that with the indicator vector $\mathbf{x}$ the criterion function can be translated into a form of the expression for Rayleigh Quotient. Using the result proposed therein, we have.

$$\min_x DocSeg(x) = \min_y \frac{y^T(D-P)y}{y^T Dy} \tag{5}$$

where $y = (1 + x) - b(1 - x)$. From (4), we have that the ideal indicator vector that achieves the minimum on the document segmentation criterion is the second Eigenvector of the generalized Eigenvalue problem

$$(D - P)y = \lambda_1 Dy$$

Hence the algorithm for document segmentation through spectral partitioning can be outlined as follows

Algorithm 1 Given a document image with $|C|$ connected components, find coherent partitions

```
Build proximity matrix $P$ by calculating $p_1\theta_1 + p_2\theta_2 + p_3\theta_3 + p_4\theta_4 + p_5\theta_5$, with a proper set of mixing parameters $(p_1, \ldots, p_5)$ for every pair of components $(c_i, c_j)$
Find the Laplacian matrix of $P$, $Q(P) = D - P$, where $D$ is the diagonal matrix containing degrees of each node over the diagonal
Solve the eigenvalue problem for $Q(P)$ resulting in eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$
Partition the eigenvector corresponding to the second lowest eigenvalue $\lambda_0$ at a point $x'$ where the following expression is minimum
    $\left[E(P_A, P_B)\left(\frac{1}{vol(P_A)} + \frac{1}{vol(P_B)}\right)\right]$
Assign the label $(1|2)$ for each node $c_i$ using,
    $v \in \{P_1 : \lambda_1(c_i) < \lambda_1(x'),\ P_2 : \lambda_1(c_i) > \lambda_1(x')\}$
Recursively partition the resulting sets until the each set conforms to a stopping criterion
```

4.2. Comments

The potential implications of using normalized cuts to document image segmentation can be viewed from two directions. For the traditional class of nonlinear optimization problems that normalized cuts is generally used, i.e., image segmentation etc., the major problem is that the favorability to define a tightly representative similarity matrix is quite rare. Thus the performance expectation from the perceptual grouping problem and such others is quite low. The problem of document image segmentation however, is a totally different kind of problem where the perceptual requirements are precise and any solution that is not 'right on tick' is highly discredited. Practically speaking, document image segmentation is a totally different problem, almost never dealt with normalized cuts which has the facility of being able to produce a perfectly representative similarity matrix. Thus the application has a entirely new flavour in terms of normalized cuts

For the traditional class of document image segmentation solutions, the convergence process is usually an iterative application of an agglomerative or a divisive scheme until it is coherently segmented. The current algorithm is novel in the sense that it provides a sort of a closed form solution such that given all the cues built into a proper representative structure, the actual process is one shot which makes the best decision based on every aspect needed to be considered. Theoretically speaking, the process of partitioning is completely different from the other methods since it is general and flexible at the same time. Thus the solution has an entirely new flavour in terms of the partitioning method.

Possibly, the closest relative to this approach is found in the work of Kumar et al [11]. There, the problem of proper assignment of trailing components within lines is dealt as an optimization that minimizes the energy based on the smoothness and data constraints. The constraints are calculated from local geometry based cues that determine the confidence of every edge assignment. The proximity matrix calculated in this approach also comprises of confidence measures which determine the conformity of a component with a partition. However a key advantage of this approach is the flexibility to model multiple visual cues from independent sources into a unified framework. That way spectral partitioning based segmentation is easily applicable for a larger class of documents layouts.

4.3. Indeterminacy and soft partitioning

In the graph theoretic formulation discussed above, the pairwise proximities are built by precise decision making clues from various document parameters. Sometimes, it so happens that the parameters do not provide discriminative confidence levels for a component to precisely assign a higher weight to one of its neighbors. In such a case, arbitrary assignment may cause wrong segmentation and the algorithm is discredited. This can be avoided in the case of ambiguities by providing a delay-decision class ρ. Suppose the parameter that governs the conditional probability of a component c_m with its neighboring components be less than a threshold M, which is the minimum confidence level. Then we can include c_m into a “delay-decision” set $\rho : C - C_p$ where C_p denotes the set of confidently assigned components. ρ holds the ambiguous components r_i and the set of labels l'_i with the confidences τ_i for each corresponding label. $\rho : \langle r, l', \tau \rangle, \qquad r \in C - C_p,\ l' \in l_v,\ \Sigma(\tau_i) = 1$

While partitioning, the delay-decision class can be parsed by treating each component as multiple duplicates and assigning each duplicate to one of the neighbors with a high confidence. This will have no effect on the optimization procedure which will segment the component into multiple partitions. Since the delay-decision class is identified prior to segmentation, the multiply assigned labels will have confidences for each labelling. These components will be excluded from the segmentation evaluation. These can be further subjected to a adhoc/semi-automatic assignment. This way the ambiguous assignments will be singled out and can help in directing the focus of the correction routine. Thus the segmentation performance will be reported high with an additional delay-decision set ρ.

5. Implementation

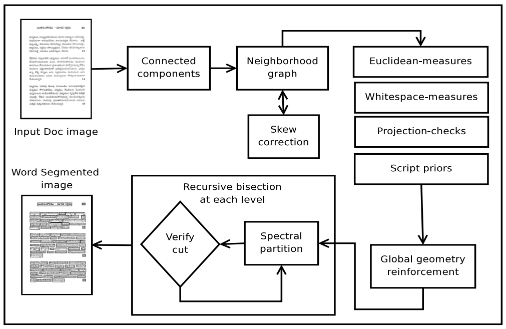


Figure 2. Block diagram showing the segmentation procedure via spectral partitioning

The block diagram in Fig. **2** shows the step-by-step process of extracting the document specific parameters to a single basis-framework.The algorithm takes in as input the document image with the connected components extracted. Each step of the implementation builds one component of the $|C| \times |C|$ proximity matrix.

Neighborhood and Skew detection: The local neighborhood of each component is observed and it is linked to its k-nearest neighbors thus building the initial proximity matrix θ_1. Using these edges, the dominant orientation is detected from the peaks of the angle histogram and any potential skew is corrected (Fig. **3(a)**). θ_1 also serves as the basis for other aspects of proximity calculation

Script priors: In the first step, the dangling modifiers are assigned to the proper component based on the spatial language model as described in [11](Fig. **3(b)**). For each character, the dangling modifiers are all merged by adding high proximities between the corresponding components. This builds the θ_2 part of the proximity matrix. For Dravidian scripts, particularly Kannada and Telugu which contain high amount of dangling modifiers, cooccurence heuristics based on aspect ratio and inter-component distance can be used to as templates to design robust script priors. Due to the complexity of such scripts, we have used a Telugu document collection to emphasize that fact.

Whitespace Analysis: For the whole document the maximal whitespace rectangles are found out that are greater than a predefined area threshold. These whitespaces are used to augment proximity information across edges(Fig. **3(c)**). The θ_3 proximities are inversely proportional to whitespace areas. Thy are calculated by the function $\theta_3(i,j) = e^{\frac{WS_{ij}}{10^4}}$ where WS_{ij} represents the sum of areas of the whitespaces cut by the edge.

Gutter Analysis: The horizontal projection profile of a document gives the whitespace gutters between lines and paragraphs in the document(Fig. **3(d)**). This information is used to build a gutter-based proximity matrix θ_4 by observing the gutters that are cut by each edge. Each gutter is assigned a weight that is a function of the ratio of components on either side of it and the size of the gutter. This ensures similar confidences for edges cutting the same gutter

Neighborhood Analysis: Based on the proximity matrices built so far, the edge weights are reinforced by observing the local neighborhood of each component. For each edge, a chain of edges are built in a direction which is specified by the property of the script. This a linear combination of the edge weights on this chain are used to reinforce the edge in question. This step has the effect of increasing the likelihood of a component in a word having greater proximities to the components in the same word and lesser proximities to the components from lines above and below it.

Proximity integration and relation to other approaches: The final proximity matrix P is a function of individual proximity matrices built. The proper relation to the final proximity is a linear combination of the individual proximities i.e.,$P = p_1\theta_1 + p_2\theta_2 + p_3\theta_3 + p_4\theta_4 + p_5\theta_5$ which is governed by the mixing parameters $\{p_1, \ldots, p_5\}$. The proper set of mixing parameters varies for each class of documents and can be easily decided. P is the final matrix that is used for spectral partitioning to achieve coherent segmentation.

Spectral partitioning of the final proximity matrix will be optimizing the cumulative objective function to find the minimum value of the Cheeger constant. If one observes closely, such objective function uses an optimal mixture of segmentation parameters used by different approaches. Suppose the mixing parameters bias towards one particular feature, eg., whitespace proximities the approach will be finding segments that are covered by maximal whitespace rectangles giving similar results as [3]. Similarly if the gutter analysis based proximity matrix is singly employed, the process would be similar to the projection profile based hierarchical cutting similar to [1]. Thus the framework has the elegance of being able to model any number of parameters into one graphical model such the information combined from different methods will give better performance where singly operating methods are affected by script challenges.

The partitioning process can be dealt as a multiway cut problem or a recursive bisection problem. However for this approach the structure of the affinity matrix is similar to the matrices described in [15] that perform well on recursive bisection. Another important advantage of recursive bisection is that the partition can be checked with the projection profile to determine whether the cut is proper. In some cases owing to ambiguous assignment of dangling modifiers, when sufficient confidence level cannot be established. It will be sent to an ambiguous component list, and assigned multiple labels. This accounts for soft partitioning which can be set aside for delay-decision using semi automatic approaches.

6. Results

Comparison of Per-word errors					
	T_c	T_u	T_o	T_m	T_{dm}
A1	75.6	3.025	0.475	4.25	5.125
A2	75.6	4.2	0	1.2	0.075
A3	75.6	1.3	0.85	0.8	8.4
A4	75.6	2.8	0.45	0.125	6.775
A0	75.6	1.125	0.025	0.025	0
A0+	75.6	0.525	0.025	0.025	0

Table 1. Averages of each class of errors with 75.6 words per document. Observe that $\mathbf{A0+}$ shows improvement in terms of under segmentations due to soft partitioning

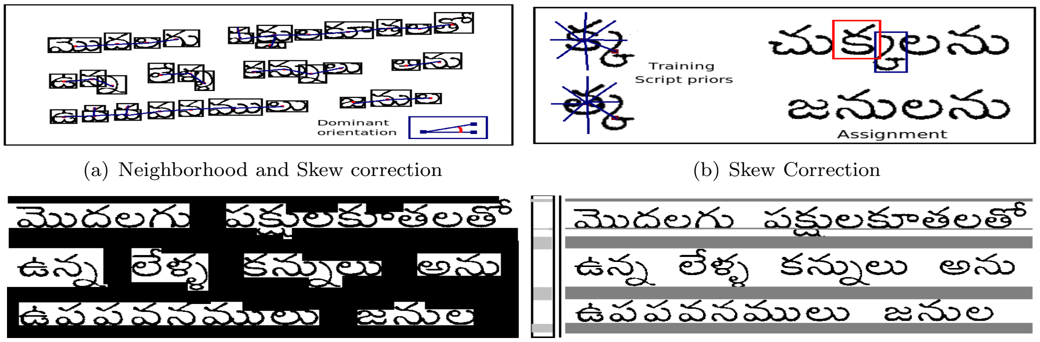
Total performance on 142 pages						
A1	**A2**	**A3**	**A4**	**A5**	**A0**	**A0+**
86.19	87.1	71.65	77.76	76.19	97.42	98.5

Table 2. Performance scores of state of art algorithms compared to$\mathbf{A0}$ and $\mathbf{A0+}$ for 142 pages.

The results shown in this section are divided into three parts to highlight various aspects projected in this work. The primary goal is to show the flexibility offered by the graph based framework to work on complex scripts.

Dataset: To demonstrate the claims we have chosen a dataset of page images gathered from the Telugu book "Rutusamhaaram" written by Kalidasa. This collection contains 142 images of pages scanned with a book scanner. The ground truth has been prepared by initially converting it into text and further employing a routine to map the words. The book used in the experiments contain pages with two kinds of layouts. Around 65 pages contain poetry in Sanskrit resulting in a fairly complex Telugu script. The other set of pages are prose translations in normal layouts, resulting in a total of three font sizes.

Eigenvector based segmentation: Fig. **4** shows six images that depict various stages of the recursive spectral partitioning process. At each stage, it is segmented such that the partitions are balanced with respect to the confidence level. The confidence level is built with respect to multiple visual cues. The cues can be chosen such that the combined contribution describes necessary and sufficient properties of the documents.Thus the outcome will



(a) Neighborhood and Skew correction

(b) Skew Correction

(c) Whitespace Structure

(d) Gutter Analysis

Figure 3. Implementation of various proximity metrics

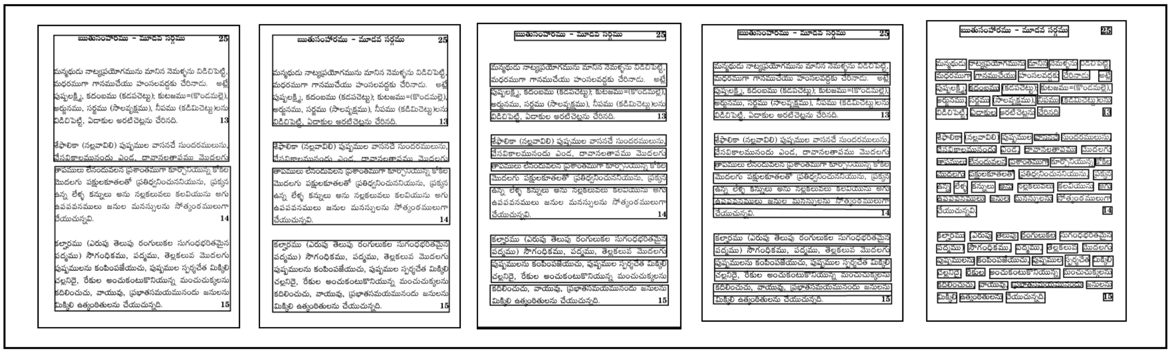


Figure 4. Results after different levels of Eigenvector based recursive segmentation($\mathrm{A0}$). Partitioning stops if a partition conforms with the projection profile, thus determining a text line. The lines are further segmented until they conform to the vertical projection profile of that text line.

reflect the better properties of the state of the art algorithms and thus should compare better than any single algorithm.

Comparision with the state of art: The state of the art algorithms that are compared with the eigenvector based method(**A0**) are the recursive XY cut (**A1**), Runlength Smearing (**A2**), Docstrum (**A3**), Whitespace Analysis (**A4**) and the Area Voronoi diagram based segmentation(**A5**). For the XY Cut the free parameters of the algorithm are the thresholds that govern the split process and stopping criteria [1]. These have been adapted to the documents such that the algorithm favors splits and the words were merged based on post segmentation techniques. The run length smearing approach is applied with the standard thresholds and it fails for any complicated layouts. The DocStrum algorithm [6] has been adapted to Telugu documents. Docstrum usually employs an angle threshold assumption for “within line neighbors” which will result in bad performance in the case of Indian language documents. The adaptation relaxes this constraint such that the complex script can be handled. However its very likely to fail in the case of varying layouts and script properties. The whitespace analysis routine from [3] finds the maximal whitespace rectangles larger than 50 pixels for the whole document. With a large number of dangling modifiers the maximal whitespace covers will be very less and even lesser for skewed documents. This will have a major detrimental effect on its performance. It has been observed experimentally that RLSA, DocStrum and Whitespace analysis are affected by the dangling components such that they are either missed or oversegmented in the final segmentation. RXYC, which is a projection profile based algorithm is not able to segment lines due to heavily trailing components and thus results in under-segmentation. The eigenvector based partitioning assigns the dangling modifiers to the proper components by script priors and thus the segmentation is proper even in the case of trailing components.

The performance analysis is done as follows. For the segmented description of the dataset for each algorithm considered, the number of total missed Dangling modifiers are measured along with the average number of undersegmented and oversegmented components for each image (C_u and C_o). Along with these metrics, the total number of missed components(C_m) are also measured. The performance score is computed using these metrics as follows. Score = $\frac{T-(C_u+C_o+C_m+DM_m)}{T_c}$. This score is an abridged adaptation from [7] and [10]. The results can be seen in Table 1.

Improvement with soft partitioning: While assigning the script priors for a document, suppose a component does not generate enough confidence to be assigned to any of its neighbors concretely. This component will be assigned to all the probable neighbors with a level of confidence for each. This will alleviate the effect of the ambiguous component over the segmentation process. The final segmented partition contains this ambiguous component in all the possible partitions, to be disambiguated manually.See Figure **5**. In our experiments the average size of the delay-decision class is 2% In Figure **5**, observe in the third text line, the ambiguous component '7' is assigned to two partitions due to the ambiguity of assignment which results in soft partitioning. These results are reflected in the Table 1 showing the performance comparision with various algorithms.

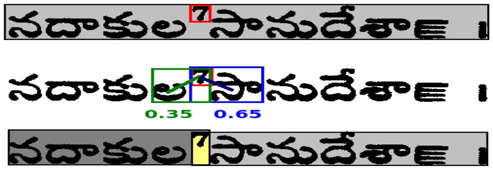


Figure 5. Improper segmentations corrected by soft assignment based on confidences. Delayed decision set contains ambiguous component

The time taken for the implementation of the algorithm can be broken into two parts. The major bottleneck in the whole process is the accumulation of various cues. This ranges anywhere from 15 seconds to one minute for each document. The actual process of segmentation is completed in 2.5 to 4 seconds. The experiments have been implemented in Matlab and run on a 2Ghz Dual-core machine running Linux.

7. Conclusion

In this paper we have developed a method for document image segmentation that departs from the classic adhoc approaches and formulates an optimization framework that achieves a closed form solution to the problem. The proposed method is flexible and general in the sense that it allows the integration of various factors into the proximity matrix that is used for partitioning the document image. The criterion that determines the ideal partition is shown to be achieved by an indicator vector which can be obtained by solving the generalized Eigenvector problem. The results reflect the flexibility of the approach in dealing with complex scripts such as those of Indian language documents

References

[1] G. Nagy, S. C. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," in *IEEE Computer*, vol. 25, no. 7, 1992, pp. 10–22.

[2] T. M. Breuel, "Two geometric algorithms for layout analysis," in *DAS '02: Proceedings of the fifth International Workshop on Document Analysis Systems*, 2002, pp. 188–199.

[3] H.S.Baird, *Document Image Analysis*. World Scientific, 1994, ch. Background Structure in Document Images.

[4] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 370–382, 1998.

[5] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, no. 6, pp. 647–656, 1982.

[6] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1162–1173, 1993.

[7] F. Shafait, D. Keysers, and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 941–954, 2008.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[9] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in *Proceedings of the International Conference on Computer Vision*, vol. II, 1999, pp. 975–982.

[10] K. S. Kumar, S. Kumar, and C. V. Jawahar, "On segmentation of documents in complex scripts," in *International Conf. on Document Analysis and Recognition*, 2007, pp. 1243–1247.

[11] K. S. S. Kumar, A. M. Namboodiri, and C. V. Jawahar, "Learning segmentation of documents with complex scripts," in *Fifth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'06)*, 2006, pp. 749–760.

[12] F. Shafait, A. ul Hasan, D. Keysers, and T. M. Breuel, "Layout analysis of urdu document images," in *Proceedings of IEEE Multitopic Conference (INMIC 06)*, 2006, pp. 293–298.

[13] P. Perona and W. T. Freeman, "A factorization approach to grouping," in *ECCV '98: Proceedings of the fifth European Conference on Computer Vision*, 1998, pp. 655–670.

[14] Fan.R.K.Chung, "Spectral graph theory," *Regional Conference Series in Mathematics,CBMS*, vol. 92, 1997.

[15] H. D. Simon and S.-H. Teng, "How good is recursive bisection?" *SIAM Journal on Scientific Computing*, vol. 18, no. 5, pp. 1436–1445, 1997.