On Foreground-Background Separation in Low Quality Color Document Images

Utpal Garain
Computer Vision & Pattern Recognition Unit,
Indian Statistical Institute
203, B. T. Road, Kolkata 700 108, India
{utpal,bbc}@isical.ac.in

Thierry Paquet and Laurent Heutte
Laboratoire PSI - FRE CNRS 2645
UFR des Sciences, University of Rouen
76821 Mont Saint Aignan cedex, France
{thierry.paquet, laurent.heutte}@univ-rouen.fr

Abstract

This paper proposes an adaptive method for separation of foreground and background in low quality color document images. A connected component labelling is initially implemented to capture the spatially connected similar color pixels. Next, dominant background components are determined to divide the entire image is into number of grids each representing local uniformity in illumination, background, etc. Finally foreground parts are located using local information around them. Several color images of old historical documents including manuscripts of high importance are used in the experiment. Apart from a qualitative evaluation, results are quantitatively compared with one popular foreground/background separation technique.

1. Introduction

With the widespread development of input devices for color images, several types of documents including historical ones are now often scanned in color mode. Therefore, research dealing with color documents has gained considerable attention in the recent past. The work presented here is motivated by this current research trend.

Studies dealing with foreground/background separation in color documents are few in number and quite different from the traditional binarization methods [1] and are broadly based on color clustering/segmentation principle [2]. Some initial studies (e.g. [3]-[6] etc.) in this domain concentrate on extraction of text parts from color images. Experiments have been conducted on images of printed documents [4], of Internet pages [3, 6], scene images from digital camera [5], etc. Images used in these experiments are mostly well contrasted and not very much suffered from several image defects that are generally observed in low quality (e.g. historical) documents.

Studies described in [7, 8] are directly focused on separating foreground and background in color documents. The method in [7] involves a manual registration of template color maps and identifies color map in a target document by a quantization algorithm. The technique presented by Leydier *et al.* [8], achieves foreground-background segmentation by a serialized K-means algorithm. This approach also involves a manual intervention to set the number of logical classes and the color samples for each logical class to initialize the initial cluster centers in the K-means algorithm. Experiment showed that the parameters used in the algorithm can have a heavy impact on the performance as well as on the computational time.

Interestingly, DjVu [9] implements an efficient foreground-background separation in the context of compression. The approach is based on a multi-scale bicolor clustering that considers several grids of increasing resolution. The technique works well for a large class of documents (gray as well as color) but fails for documents with low contrasted foreground and background as often observed in low quality manuscripts. Some failures are demonstrated in section 3 of this paper.

This paper presents an alternative algorithm for foreground/background separation in color documents. An special emphasis is put on processing of historical documents that in general, are suffered from many degradations like uneven illumination, noise, aging effect, etc. The approach during its execution does not require any human intervention and is applicable for printed as well as handwritten manuscripts. The method is quite adaptive in nature to capture nonuniformities in illumination, background, etc. and produces acceptable results on a range of test images starting good to very low quality. The rest of the paper describes the proposed method and experimental results.

2. The Proposed Method

Our proposed method consists several processing steps as explained below:

Preprocessing step: This is an optional one and mainly deals with color smoothing in an input image. In our ap-

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

proach, each pixel color is reassigned to an average color value found within a small region surrounding that pixel. Color of each pixel (X) is redefined as the mean color computed in a 3×3 window surrounding the pixel, X (X being at the center of the window). However, for applications dealing with specific documents, one may incorporate more sophisticated smoothing techniques.

Connected Component Labelling (CCL): CCL is executed on the entire image. Since the traditional CCL algorithm assumes an image in binary mode and only considers spatial connectivity (e.g. 4- or 8-connectivity), we modify this algorithm to capture color information and spatial details at the same time. The Algorithm-I presents the modified algorithm. Please note that for the same image, CCL result returned by the Algorithm-I will differ in different color spaces and even within the same color space results differ with different values of the threshold, τ.

Algorithm-I: Connected Component Labelling (CCL) in Color Space

Input: Color Image (I) and Output: Labelled image consists of k number of connected components, $c_1, c_2, \ldots, c_k$.
Initialization: All pixels are initially unlabelled.
S: is the stack of pixel coordinates;
L: is the label value and initialized to 1;
for i=1 to H (image height in pixels)
  for j = 1 to W (image width in pixels)
    if X = I(i,j) is unlabelled
      push(X);
      while Stack (S) is not empty
        Y = pop();
        Label(Y) = L;
        Compute the current mean color (μ_L) for
        the component c_L
        If $P_1, P_2, \ldots, P_8$ are the 8-connected
        pixels of Y then for each unlabelled P_k
        Compute distance[1] $\mathcal{D}_k = ||\text{color}(P_k) - \mu_L||$;
        If $\mathcal{D}_k < \tau$ (a pre-defined threshold) push(P_k);
      end while;
      L=L+1;
    end if;
  end for;
end for;

Selection of Color Space: Experimentally it is verified that end results presented in section-3 are comparable in all four different color spaces (RGB, HSV (or HSI), YCbCr, and CIE L*u*v*). However, HSV is preferred for the present application because if the color spaces (for the same image) are forced (by controlling the threshold value, (τ)) to generate the same (or nearly the same) number of connected components, a single component in HSV is broken into less number of components as compared to other color spaces. Therefore, CCL under HSV eventually generates less number of connected components than in other color spaces and makes the subsequent processing (as described later) faster. However, instead of using one color space at a time, more than one color (e.g. [8] uses RGB and HSV values together) space can improve the CCL results.

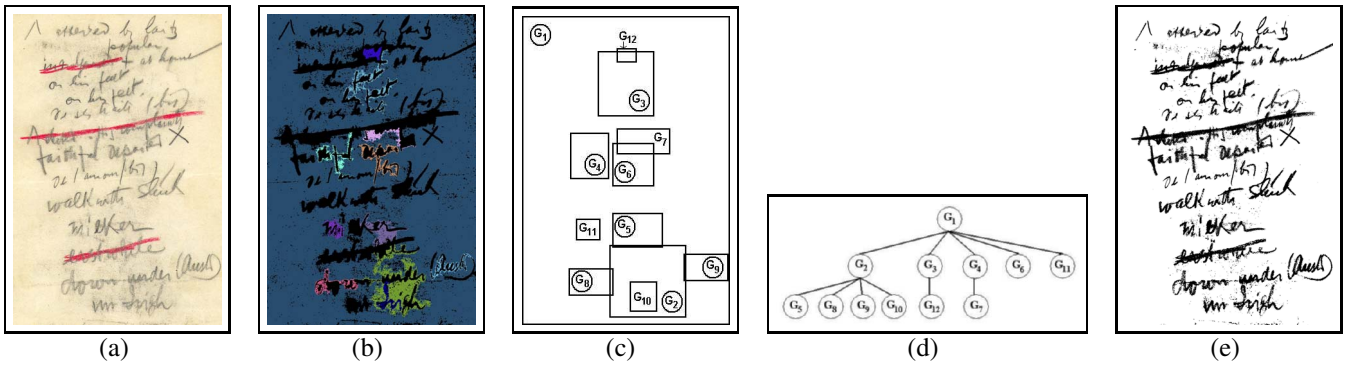
Automatic selection of τ: Considering the varying nature of documents, we prefer to automatically select the threshold value (τ) rather setting it to a predefined value. The τ is calculated from the input image by considering only the adjacent pair of pixels in row and column wise manner. The maximum color distance between such pixel pairs are recorded for each row and column and an average of these values serves as the threshold, τ used in the Algorithm-I.

Identification of dominant background: Let $c_1, c_2, \ldots, c_k$ be the k no. of connected components found by the Algorithm-I. Each component (c_i is the i-th component) is tagged with its label (i), size (S_i) in terms of number of member pixels, center of inertia ($C_i(x, y)$), top-left and bottom-right coordinates of its surrounding bounding box (bbox) and other three values namely μ_{h_i}, μ_{s_i}, and μ_{v_i} representing the mean hue, saturation and value for c_i, respectively.

If the components are sorted in descending order of their S_i values and compared with respect to their mean size (μ_S), it would be clear that very large (w.r.t. to μ_S) components basically represent background parts. If the background of an image is uniform then only a few number of background components are located but this number increases if non-uniformity of the background increases. A component c_i is labelled as a background component if $S_i > \alpha\mu_S$, where α is a scalar and determined dynamically. Fig. 1(b) shows the dominant background components extracted for the image in fig. 1(a). Execution of Algorithm-I for this image results in 16,439 connected components which shows a mean size of 182.69 pixels and only 12 components have been identified as background components.

Identification of these background components shows the non-uniformity in the background of the input image. Components representing foregrounds are much smaller in size; however, there are various background components which are small in size and remain unidentified at this stage. Let l be the number of identified dominant background components and $\mathcal{B}$ represents this set of l components. Let c_b (where $b \in [1 \ldots l]$) be the biggest component and its color is taken as the reference color (B_{ref}) for background parts. A reference foreground color (F_{ref}) is then computed as follows. For each component c_i, ($i \neq b$) its color distance with c_b is measured and the component (say, c_f) with the highest distance is treated as reference foreground part. Use of F_{ref} will be clear in the later sections.

1 Results reported in section 3 are based on an Euclidian distance measure defined in the HSV space (H, S and V are normalized to have values in [0, 1] but other distance measures are now being investigated.



(a) (b) (c) (d) (e)

Figure 1. Foreground extraction: (a) input image, (b) identified dominant background components are painted with randomly chosen colors (black pixels are part of connected components which are either foreground/background unidentified at this stage), (c) grids formed by dominant background components, (d) hierarchical arrangement of grids, (e) extraction results after processing of grids.

Formation of grids and their arrangement: Once the set $\mathcal{B}$ is identified, the entire image is divided into several grids. This is done by using the bounding box (bbox) information available with each component. Hence, l grids are identified for l background components. For example, fig. 1(c) shows the grids formed by 12 dominant background components (shown in fig. 1(b)). Next, these grids are arranged in a hierarchical structure following a graph theoretic approach. Geometric layouts of the grids and the area covered by each grid are considered for such an arrangement which induces a tree structure. The Algorithm-II determines the *parent-child* relationship between two grids.

Algorithm-II: Hierarchical arrangement of grids

Let $g_1, g_2, \ldots, g_l$ be the list of l-grids corresponding to l background components.
Sort the grids in a descending order on the area covered by them.
for i=1 to l-1
  for j = i+1 to l
    if g_i and g_j are overlapping or contained in one another
      then `parent`$[g_j]$ = g_i;
    end if;
  end for;
end for;

The largest grid consequently forms the root of the tree. However, if the largest grid does not overlap or contain other grids, the disjoint sets of overlapping (or contained in) grids will result in. In such cases, each set of overlapping grids generate a tree and final arrangement of grids results in a forest (collection of trees) structure. However, in any case, the Algorithm-II defines a *parent-child* relationship, if exists between two grids. Fig. 1(d) demonstrates the tree that is formed with the grids in fig. 1(c).

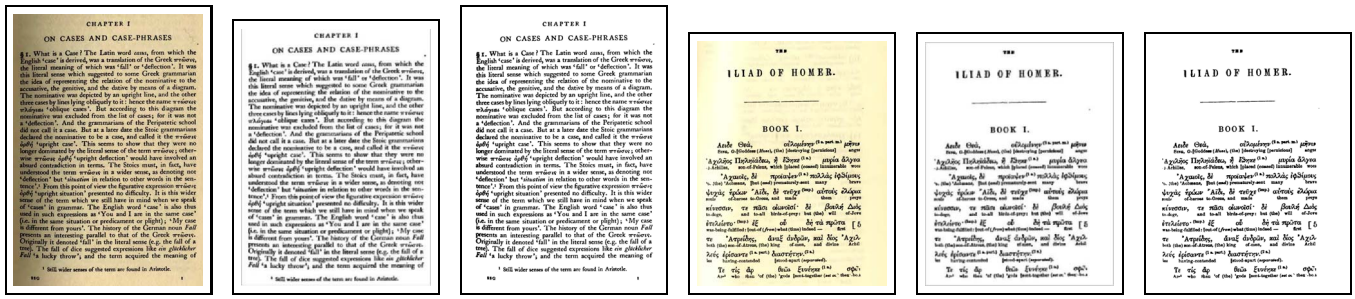
Bicolor clustering: The grids are initially arranged in a hierarchical structure to implement a grid level bicolor clustering of the connected components. The constituent components for a grid is determined by looking at their $C_i(x, y)$ values. The K-means algorithm is used with slight modification for its initialization. The algorithm is initially applied on the root grid for which one cluster is initialized with B_{ref} and another cluster is initialized with F_{ref}.

For the inner grids (leaves and non-leaf nodes) of the tree, one cluster center is initialized with the same color as that of the background component represented by the grid (note that each node representing a grid is originated by a background component identified before) but the second cluster center is initialized by the foreground color found in the parent of the current grid. This modification in initializing the bicolor clustering algorithm introduces a bias in the process to rapidly attract the components towards the true foreground and background clusters. Processing of all grids (arranged in a tree or forest) results in a binarized version of the input image. Fig. 1(e) shows this end result for the image in fig. 1(a).

Note that if the grids representing dominant background components do not cover the entire image surface, some components may remain unprocessed by the bicolor clustering executed on the tree (or forest) of grids. Therefore, a final check is done to detect unclassified connected components and another round of bicolor clustering is executed on them. During this execution, initialization of two cluster centers is done by using (B_{ref}) and (F_{ref}), respectively.

3. Experimental results

As the real challenge lies in dealing with low quality documents showing different degradations due to uneven illu-



(a) (b) (c) (d) (e) (f)

Figure 2. Results on DjVu sample documents:: (a) and (d): Input images; (b) and (e): results obtained (at 300 dpi) by DjVu technique; (c) and (f): results obtained by our approach.

mination, aging, scanning, etc., creation of test data puts emphasis on this issue. At present, documents dominant in text (printed or handwritten) are only taken into consideration. The test dataset contains about 50 documents which are broadly divided into two parts. Most of the images are scanned at 300 dpi (a few are at 150 dpi) and images are quite large in size (average size is about 3M pixels). Under Part-I of the test set, 20 test documents are taken from the '*Ancient books, and Historical Documents*' category of '*DjVu Zone Digital Library*'[2]. Selection of these documents enable us to compare our results with those by DjVu technique [9]. Many documents under this group are century old, some are more than 200 years.

Part-II of the test dataset consists of 30 documents which are color scanned from working notebooks of several famous writers of 19th and 20th century. Samples include images from working notebooks of Gustave Flaubert (1821-1880), James Joyce (1882 - 1941), etc. Many of these manuscripts had been written with quill or lead pencil (not ink) and pencil marks are frequently spread over the background and very low contrast is observed between foreground and background parts.

Fig. 2 shows foreground extraction results for two documents taken from Part-I of the test dataset. Figure (b) & (c) (similarly (e) & (f)) compares the results obtained by DjVu [9] and our proposed technique. These documents are taken from DjVu document database where DjVu achieves good extraction results but our results are also comparable to that of DjVu as checked visually. However, as far as part-II documents are concerned, DjVu very often fails to properly extract the foreground parts because of very low contrast between foreground and background, whereas our method, in major cases successfully locate the foreground elements. Fig. 3 presents a few examples where our technique outperforms DjVu results. An evaluation based on a quantitative method (similar to one in [10]) is presented next.

Evaluation of foreground extraction efficiency: Since the test documents are predominantly textual in nature, quantification of extraction accuracy is done at two different levels: line and word. Evaluation of extraction results is done by manually computing the no. of lines and words in each original image (I_o) and in the corresponding extracted foreground (I_e). Next, extraction efficiency (ϵ_l: at the line level, and η_w: at the word level) is measured as, $\epsilon_l = \frac{\text{No. of lines (correctly extracted) in } I_e}{\text{No. of lines in } I_o}$ and $\epsilon_w = \frac{\text{No. of words (correctly extracted) in } I_e}{\text{No. of words in } I_o}$.

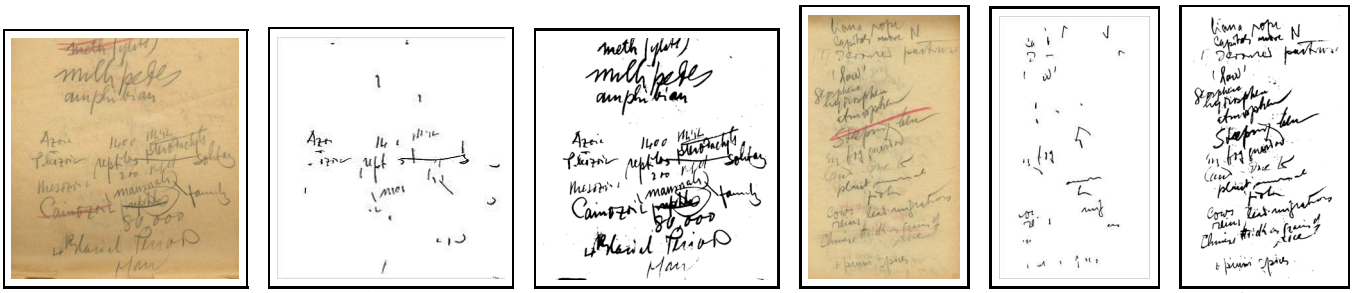
Since word boundaries are sometimes not very clear in some images and therefore, the counts (mainly for words) are based on manual perception only. Errors in extractions are classified into two categories: (i) partially extraction and (ii) completely missed. A line is partially extracted if some its constituent words are missed, whereas extraction of a word is partial if some constituent strokes are missed. Table 1 presents the evaluation results for the combined set of 50 test documents. Evaluation results for DjVu technique is presented for a comparative evaluation.

Eval. level	#No.	Type of extraction			ϵ_l or ϵ_w
		#Correct	#Partial	#Missed	
		DjVu / Our	DjVu / Our	DjVu / Our	DjVu / Our
Line	917	541 / 761	282 / 114	94 / 42	0.59 / 0.83
Word	59,631	42,934 / 54,264	10,007 / 3,636	6,690 / 1,731	0.72 / 0.91

Table 1. Foreground extraction results

Major sources behind extraction errors: Visual examination of extraction results reveals that though the proposed approach successfully works for most of the images, in a few cases, it fails to locate all the visible foreground parts of an input image. It is analyzed that such problems oc-

2 Freely available at http://www.djvuzone.org/djvu/antics/index.html



(a) (b) (c) (d) (e) (f)

Figure 3. Handwritten historical manuscripts:: (a) and (d): Input images; (b) and (e): extraction results obtained (at 300 dpi) by DjVu technique; (c) and (f): results obtained by our approach.

cur more in handwritten manuscripts than in printed documents and it is mainly due to (i) very weak stroke marks, (ii) very low contrast between a foreground stroke and its corresponding background and (iii) spreading of ink or pencil marks over the background.

Bleed-through (show-through) effect impose another problem. It is observed that marks present due to show-through effect are sometimes identified as foreground. However, in such cases bleed-through marks are quite prominent and look similar (as visually perceived) to the true foreground parts and therefore, identifying them as foreground parts should not be judged as any weakness of the proposed algorithm which primarily deals with foreground extraction/binarization. Rather, some specialized technique can be used as a post-processing method to improve binarization results for documents suffered from bleed-through effect.

4. Conclusions

In this paper, an efficient approach is presented for background/foreground separation in color document images including low quality color ones. Test set contain several samples for printed as well as handwritten manuscripts of historical nature. Many of them written by famous writers and written with quill, pencil, etc. generating low contrast between background and foreground. Results show enormous adaptability of the proposed approach with the uneven illumination or local changes in background and foreground color. However, behavior of this algorithm for documents riched with non-textual elements like graphics, half-tones, etc. is considered as future extension of the present work. Moreover, proper assessment of the extraction results needs benchmarking and groundtruthing of foreground and background pixels in sample documents. This needs use of extensive manual intervention and hence finding an efficient way (may be semi-automatic in nature) of achieving it could be treated as another future direction of the current study.

References

[1] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, 33, 225-236, 2000.

[2] H.D. Cheng, X.H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, 34, 2259-2281, 2001.

[3] D. Lopresti and J. Zhou, "Locating and Recognizing Text in WWW Images," *Information Retrieval,* 2, 177-206, 2000.

[4] T. Perroud, K. Sobottka, and H. Bunke, "Text extraction from Color Documents- Clustering Approaches in Three and Four Dimensions," *6th International Conference on Document Analysis and Recognition* (ICDAR), Seattle, USA, 937-941, 2001.

[5] K. Wang and J.A. Kangas, "Character Location in scene images from digital camera," *Pattern Recognition,* 36, 2287-2299, 2003.

[6] P.K. Loo and C.L. Tan, "Adaptive Region Growing Color Segmentation for Text Using Irregular Pyramid," *in 6th International Workshop on Document Analysis Systems* (DAS), Italy, LNCS vol. 3163, 264-275, 2004.

[7] J. He and A.C. Downton, "Colour Map Classification for Archive Documents," *in 6th International Workshop on Document Analysis Systems* (DAS), Italy, LNCS vol. 3163, 241-251, 2004.

[8] Y. Leydier, F.L. Bourgeois, and H. Emptoz, "Serialized k-Means for Adaptive Color Image Segmentation- Application to Document Images and Others," *in 6th International Workshop on Document Analysis Systems* (DAS), Italy, LNCS vol. 3163, 252-263, 2004.

[9] L. Bottou, P. Haffner, and P.G. Howard, "High Quality Document Image Compression with DjVu," *Journal of Electronic Imaging*, 7(3), 410-425, SPIE, 1998.

[10] C. Yan and G. Leedham, "Decompose-Threshold Approach to Handwriting Extraction in Degraded Historical Document Images," *in 9th Int. Workshop on Frontiers in Handwriting Recognition* (IWFHR), Kokubunji, Tokyo, Japan, 2004.