TexTraCC: Text extraction using color-based connected component labeling

T Kasar*, A G Ramakrishnan*, Amey Dharwadker† and Abhishek Sharma‡
* Department of Electrical Engineering
Indian Institute of Science, Bangalore, INDIA - 560012
{tkasar, ramkiag}@ee.iisc.ernet.in
†Department of Electronics and Communication Engineering
National Institute of Technology, Tiruchirappalli, INDIA-620015
ameydhar@gmail.com
‡Department of Electrical Engineering
Birla Institute of Technology, Mesra, INDIA-835215
sharma.abhishek@ieee.org

***Abstract*—This paper describes a novel method for extraction of colored text from natural scene images. The core segment of the method involves a two-fold smoothing of the color pixels performed along the row and columns sequentially. The 0-1 and 1-0 transitions in the edge image are analyzed to obtain the run-lengths of 0s and 1s in each row and column of the image. The smoothed image is obtained by replacing each edge segment by the median of the color values of pixels in that particular segment. A color-based connected component labeling is then performed on the smoothed image and the stroke width and color information of the resulting components are used to identify the foreground text components. The method is tested on word images from the ICDAR 2003 robust reading competition dataset and is found to enhance the recognition accuracy significantly.**

I. Introduction and motivation

Digital cameras have enabled human interaction with any type of text in any environment. Its ability to capture non-paper document images like scene text has several potential applications like licence plate recognition, road sign recognition, document archiving and retrieval, historical document processing and reading aids for the visually-challenged. Besides hard copy documents, new types of documents such as e-mails, web-pages, still pictures and videos have now emerged. Due to the variations in the imaging condition as well as the target document type, traditional scanner-based OCR systems cannot be directly applied on camera-captured images and a new level of processing is necessary.

While it is relatively easy to segment characters from clean scanned documents, camera-based images are difficult to process due to unconstrained mode of image acquisition. They inherently suffer from various geometric and photometric distortions that are absent in scanned documents. Scene text can appear on any surface, not necessarily on a plane. In addition, text in natural scenes are often designed with artistic fonts, colors and complex backgrounds, posing a big challenge to text extraction. Therefore, scene text extraction systems must be applicable to generic documents and be able to handle as many types of text as possible. Unlike that of processing conventional document images, scene text understanding normally involves a pre-processing step of text detection and extraction before subjecting the acquired image for character recognition task. The subsequent recognition task is performed only on the detected text regions so as to mitigate the effect of background complexity. Robust extraction of text is therefore a critical requirement since it affects the whole recognition process that follows.

II. Review of text segmentation

Most recognition algorithms expect clean documents, with text strokes in black ink against a white background. The simplest method for text extraction is thresholding, that assigns a pixel as the foreground object or the background based on its gray level or other attributes with respect to a reference value. Global thresholding techniques [1], [2], [3] that rely on histogram analysis and statistical methods to derive a single threshold value for the entire image, work well for documents with a well-separated foreground and background intensities. However, they do not have any spatial discrimination and cannot handle non-uniform illumination. Local thresholding methods [4], [5], [6] compute an adaptive threshold for each pixel based on the image statistics in its neighborhood. This provides more robustness to non-uniform illumination. Local methods require that the size of the neighborhood should be so chosen that it encloses at least one character. This limits their applicability to only known document types, that have a uniform font size. In addition, all the above thresholding techniques require a prior knowledge of the foreground-background polarity and hence cannot handle documents that contain inverse text.

Color information is being increasingly used for the analysis of newer document types, scene text in particular. Most approaches [7], [8], [9], [10], [11] involve clustering on the 3D color histogram followed by identification of text regions in each color layer using some properties of text. Zhu *et al.*[12] proposed a CC-based text detection method that uses a non-linear Niblack thresholding scheme for segmentation. Each CC is described by a set of low level features and text components are classified using a cascade of classifiers trained with Adaboost algorithm. However, complex backgrounds and touching characters interfere in the accurate identification of CCs and thus significantly degrade their performance. Badekas *et al.* [13] estimate dominant colors in the image followed by CC labeling in each color plane. Text blocks are identified by CC filtering and grouping based on a set of heuristics. Each text block is applied to a Kohonen SOM neural network to output only two dominant colors. Based on the run-length histograms, the foreground and the background are identified to yield a binary image having black text in white background. The performance of this method rely on the accuracy of color reduction and text grouping, which are not trivial tasks for a camera-captured complex document image. The method does not process isolated characters. An accurate identification of CCs is the key to the success of these algorithms.

An alternative approach is to skip the binarization step altogether. Binarization of complex images is a difficult task and it can lead to noise and significant loss of information. There has been recent interest in OCR algorithms that operate directly on gray scale images without binarization [14]. Unlike the previous approaches to solve the same or similar problems, we employ a new color-based CC labeling on color images directly. Thus, our main focus is to achieve pixel-level segmentation for accurate character extraction. Our method not only avoids the binarization stage but also yields robust segmentation of CCs leading to increased recognition rates.

III. System description

This section describes a novel color segmentation method that uses the edge information to guide the run-length color smoothing process. Our methodology consists of two stages. In the first stage, color segmentation is performed at the pixel-level using a novel color smoothing and labeling technique. In the second stage, we employ stroke and text color information to identify the foreground text components.

A. Run-length color smoothing

The segmentation process starts with color edge detection to obtain the boundaries of homogeneous color regions. Canny edge detection is performed individually on each channel of the color image and the edge map $\mathcal{E}$ is obtained by combining the three edge images. A morphological bridging operation is performed on the resulting edge image to close broken edges. Furthermore, small and spurious edge components are removed before the color smoothing step.

The edge image $\mathcal{E}$ is used to guide the color smoothing process. The 0-1 and 1-0 transitions along the i^{th} row of the edge image are identified from the sequence,

$$\mathcal{Z}_i(j) = |\mathcal{E}(i,j) - \mathcal{E}(i,j+1)|, \quad j = 1, 2, \cdots, (m-1) \quad (1)$$

where m is the number of columns of the edge image and i and j denote the row index and the column index respectively.

The run lengths $\mathcal{R}_k$ of $0^{'s}$ and $1^{'s}$ are then obtained from the sequence $\mathcal{Z}_i(j)$. The smoothing process involves the median filtering of the color values of every segment of the row defined by $\mathcal{R}_k$ and the resulting row-smoothed image is denoted by $\mathcal{I}_r$. The same procedure is applied independently along the columns to obtain a column-smoothed image $\mathcal{I}_c$. These two output images are fused as follows:

$$\mathcal{I}_s(i,j) = \frac{\text{Median}(\mathcal{N}_3(\mathcal{I}_r(i,j))) + \text{Median}(\mathcal{N}_3(\mathcal{I}_c(i,j))}{2} \quad (2)$$

Here, $\mathcal{N}_3(\mathcal{I}(i,j))$ denotes a 3×3 neighborhood around the pixel $\mathcal{I}(i,j)$. This color smoothing process significantly reduce the variations in the RGB values of the original image. This is an important pre-processing step that minimizes the effect of complex backgrounds enabling a more accurate CC extraction.

B. Color connected component labeling

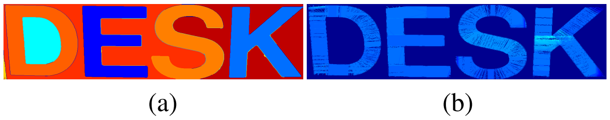
The smoothed image is scanned pixel by pixel from left to right and top to bottom. Let (i,j) denote the pixel at any step in the scanning process. We consider its two upper diagonal neighbors and the left and upper neighbors. The nature of the scanning process is such that these neighbors have already been processed by the time the procedure gets to pixel (i,j). The common RGB color format is not suitable for color grouping tasks because it is not expressed in the way perceived by humans. So, we use a uniform color space, namely CIE $L^*a^*b^*$, in which similar changes in color distance correspond to similar recognizable changes in the perceived color. The color distances of the four neighbors from the pixel at (i,j) are computed and our labeling algorithm proceeds as follows:

1) If none of the neighbors have a color distance smaller than a predefined threshold value T_c, assign a new label to pixel (i,j).
2) If only one of the neighbors has a color distance smaller than T_c, assign its label to pixel (i,j).
3) (a)If two or more neighbors have a color distance smaller than T_c, the pixel at (i,j) is assigned the label of the one that has the least color distance.
   (b) The label of the pixel that has the least color distance is also assigned to the other neighbors pixels.
   (c)All the previously labeled pixels in the image that have the same label as that of the other neighbor pixels are re-assigned with the label of the pixel having the least color distance.

The distance between the colors $\text{C}_1 = (L_1^*, a_1^*, b_1^*)^{\text{T}}$ and $\text{C}_2 = (L_2^*, a_2^*, b_2^*)^{\text{T}}$ is computed as the Euclidean distance:

$$\text{Dist}(\text{C}_1, \text{C}_2) = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} \quad (3)$$

The threshold parameter T_c decides the similarity between two colors and is set to a value 8 after experimentation. It is observed that in some images that have a poor contrast, the color transition at the edges are not sharp and hence the neighbor pixels belonging to different labels can satisfy the color threshold. This condition leads to merging of previously correctly labeled CCs with the background CC. To avoid this problem, we check whether the current pixel is an edge pixel or not. The re-assignment step (step 3(b & c)) of the algorithm is skipped whenever the current pixel belongs to the edge image.



(a) (b)

Fig. 1. (a) Output of color CC labeling (b) Stroke width image from which the dominant stroke width is computed

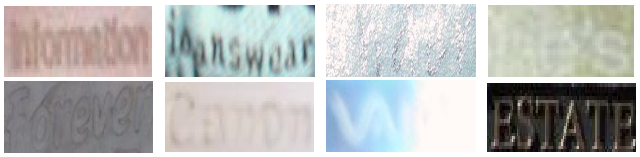


Fig. 3. Some samples of the type of images from the ICDAR 2003 database not considered for evaluation of our work.

The conventional CC labeling methods store equivalent label pairs as equivalent classes and label re-assignment through a second scan. On the other hand, our approach requires only a single scan through the image. Therefore, our method depends only on the image size and not on the number of CCs.

IV. Extraction of foreground text

A. *Heuristic CC filtering*

We make some sensible assumptions about the text regions in the image to filter out the obvious non-text CCs before the character extraction step. Components whose height is less than 6 pixels or area less than 15 pixels are unlikely to be text and are therefore removed. A minimum stroke width of 3 pixels is enforced to remove very thin components. Similarly, components having stroke width greater than one-third of its height are removed. The labels of the pixels situated on the image boundary (pixels from the first row, first column, last row and last column) are used to determine the background CC. If the frequency of the label of a particular component on the boundary pixels exceeds $0.8 \times (m+n)$, such a component is considered to be the background and is therefore removed. Here m and n denote the number of rows and columns, respectively.

B. *Stroke width estimation*

Characters in a word usually have uniform stroke width and color. Based on the observation that the dominant stroke width in the word image normally corresponds to the character stroke width, we seek to estimate the dominant stroke width in the labeled image using a variant of the method proposed in [15]. The stroke width of each CC is estimated using the edge information. At each pixel (i, j) in the edge map, the gradient angle $\theta(i, j)$ is first computed. We start traversal in the direction $\theta(i, j)$ until we reach another edge pixel. A valid stroke-ending edge pixel should have a gradient direction roughly opposite to that of the stroke-starting edge pixel. Thus, we treat the computed stroke segment as valid only if the gradient angle of the stroke-ending pixel is within the range of $\pm\pi/8$ radians from the angle opposite to that of the stroke-starting pixel. All pixels in the stroke segment are assigned this particular stroke width, unless it had been previously assigned a smaller stroke width. This process is repeated in the direction $-\theta(i, j)$ to handle inverse text too.

However, complex areas of the character pixels, especially at the corners, may not contain the true stroke width. Thus, we make a second pass where process of stroke traversal is repeated again. However, this time we compute the median stroke width over all the pixels of each valid stroke segment; and replace each pixel of that segment with this median value unless it already contains a smaller stroke width value. Figure 1 shows the output of color CC labeling and the corresponding stroke width image. A few spurious stroke widths may still be present but since we make use of the dominant stroke width of the components, our results are not affected by such minor inconsistencies.

C. *Foreground text extraction*

For text components, we make use of the fact that the stroke width will be uniform to a large extent. Once the stroke width of the labeled CC is obtained, we compute a dominant stroke width, SW_{dom} over all those components.

$$SW_{dom} = \text{mode}(SW) \tag{4}$$

where SW is the set of valid stroke widths. Potential text components are identified as those CCs whose stroke width lie within $(SW_{dom} \pm 0.5 \times SW_\sigma)$ where SW_σ is the standard deviation of the stroke widths.

We then invoke the color dissimilarity constraint to remove non-text CCs that may have comparable stroke widths as that of the text stroke width. We compute the distance D of the mean color of these potential text CCs from the mean color of the pixels whose stroke width is equal to SW_{dom}. Assuming that the letters are of uniform color, we filter out CCs having a value of D greater than a threshold which is set to 20 after experimentation. The final segmented text CCs are those that remain after the stroke width and color-based filtering. The results of different processing steps such as color smoothing, color-based CC labeling and subsequent text CC extraction from the stroke width and color information are shown in Figure 2 for two sample images.

V. Experiments and results

The proposed method is tested on word images from the ICDAR 2003 robust reading competition data set [16]. The 'TrialTrain' data set contains around 1150 English words. These images include various kinds of degradations such as uneven lighting conditions, complex backgrounds, variable fonts, color, size and may appear on curved or shiny surfaces. Most of these images have a uniform text color and are horizontally aligned. We remove the ones that have low resolution, poor contrast and difficult to read even by humans. Some of these images are shown in Figure 3. Subsequently, we retain 915 images which are very representative of text instances in natural scenes. These selected images contain a total of 5177 characters.

Some example outputs of the proposed color text extraction method are shown in Figure 4. To evaluate the performance

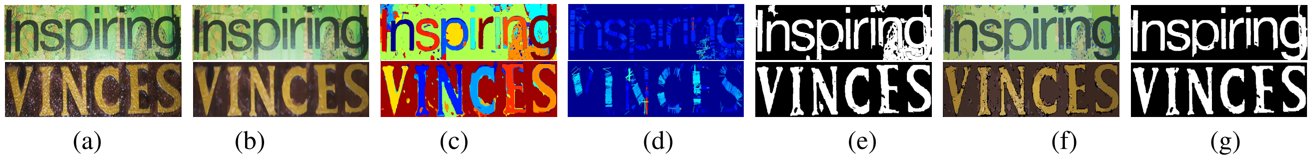


Fig. 2. Illustration of each of the processing steps. (a) Input images (b) Output of color smoothing (b) Output of color-based CC labeling (d) Stroke width of CCs computed from the labeled images (e) Potential text CCs based on stroke width information (f) After replacing the color of each pixel in a CC by its mean color (g) Final extracted text CCs after color filtering

TABLE I
EVALUATION OF CHARACTER SEGMENTATION: PRECISION, RECALL AND OCR EVALUATION USING NUANCE OMNIPAGE PROFESSIONAL 16 SOFTWARE (TRIAL VERSION) WITH AND WITHOUT THE PROPOSED COLOR TEXT SEGMENTATION.

Total number of characters tested	Precision	Recall	Levenshtein distance on raw input image	Levenshtein distance on the output of proposed method
5177	91.7 %	93.4%	2188 (57.7)%	417 (91.9)%

of text extraction quality, *Precision* and *Recall* measures are defined as follows:

$$Precision = \frac{\text{Number of extracted text CCs}}{\text{Number of extracted CCs}} \quad (5)$$

$$Recall = \frac{\text{Number of extracted text CCs}}{\text{Total number of text CCs}} \quad (6)$$

Since there is no pixel-level ground truth available for the data set, we determine the number of CCs correctly identified as text and the number of extracted CCs by visual inspection and quantify the results. We obtain a precision and recall values of 91.7% and 93.4% respectively. We illustrate the effectiveness of our method using Nuance Omnipage Professional 16 (trial version) OCR software. To quantify the OCR results, the Levenshtein distance between the ground truth and the OCRed text is computed and listed in Table I. The results of OCR on the raw input images without using the proposed method is 57.7% which markedly increases to 91.9% when it is fed with the processed outputs.

The proposed method does not assume any script specific-features and is therefore applicable to a number of other scripts. Figure 5 shows some example outputs of the method on 5 Indic scripts (Bangla, Devanagari, Kannada, Malayalam and Tamil).

VI. Conclusions and Future Work

This paper describes an important preprocessing step for scene text analysis and recognition. We have presented a novel run-length color smoothing operation and color-based CC labeling for robust text segmentation from real-world camera-captured images. While the color smoothing significantly reduces the color variability without affecting the edges, the subsequent CC labeling process ensures an accurate identification of CC because of the inherent connectivity property being invoked while grouping the pixels. This mitigates the effect of complex backgrounds and significantly improves the recognition rate. The method is found to be robust from our experiments on a wide range of target document types.

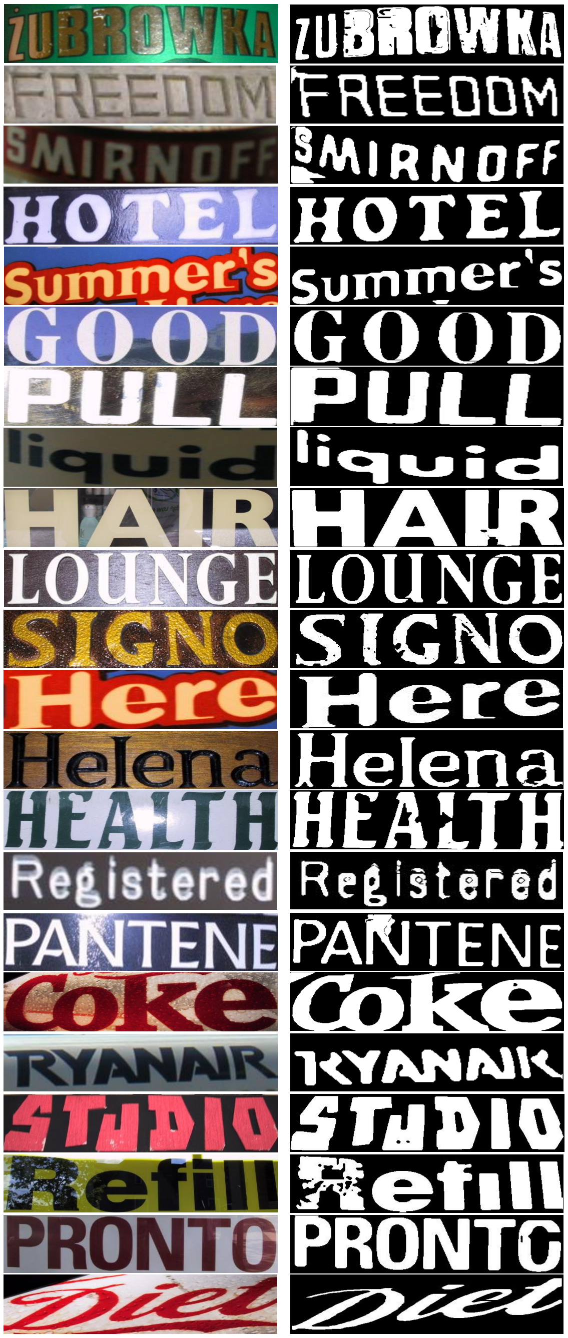


Fig. 4. Representative results of color text segmentation.

CC-based methods are suitable for camera-captured images

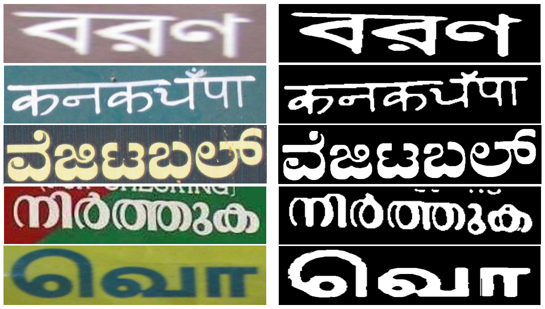


Fig. 5. Results of text segmentation on some Indic script word images.

since they are generally more robust to large variations in font styles, size, color, text orientation and layout. In our future work, we will make use of the proposed color text segmentation method in conjunction with a trained classifier to automatically localize and extract text from natural scene images.

References

[1] J. N. Kapur, P. K. Sahoo and A.K.C. Wong, “A new method for gray-level picture thresholding using the entropy of the histogram”, Computer Vision Graphics Image Processing, 29, 273 - 285, 1985.

[2] J. Kittler and J. Illingworth, “Minimum error thresholding”, Pattem Recognition, 1(19), 41 - 47, 1986.

[3] N. Otsu, “A threshold selection method from gray-level histograms”, IEEE Trans. Systems Man Cybernetics, 1(9), 62 - 66, 1979.

[4] W. Niblack, “An Introduction to Digital image processing”, Englewood Cliffs, N.J., Prentice Hall, 115 - 116, 1986.

[5] J. Sauvola and M. Pietikainen, “Adaptive Document Image Binarization”, Pattern Recognition, 33, 225 - 236, 2000.

[6] C. Wolf, J.M. Jolion and F. Chassaing, “Text Localization, Enhancement and Binarization in Multimedia Documents”, Proc. Intl. Conf. Pattern Recognition”, 4, 1037 - 1040, 2002.

[7] A.K. Jain and B. Yu, “Automatic Text Location in images and video frames”, Pattern Recogniton, 12(3), 2055 - 2076, 1998.

[8] S. S. Raju, P. B. Pati and A.G. Ramakrishnan, “Text Localization and Extraction from Complex Color Images”, Proc. Intl. Symp. Visual Comput., LNCS Springer Verlag, 486 - 493, 2005.

[9] C. M. Thillou and B. Gosselin, “Color text extraction with selective metric-based clustering”, Computer Vision and Image Understanding, 107, 97 - 107, 2007.

[10] B. Wang, X. F. Li, F. Liu and F. Q. Hu, “Color text image binarization based on binary texture analysis”, Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing, 3, 585 - 588, 2004.

[11] Y. Zhong, K. Karu and A. Jain, “Locating text in complex color images”, Pattern Recognition, 28(10), 1523 - 1535, 1995.

[12] K. Zhu, F. Qi, R.Jiang, L. Xu, M. Kimachi, Y. Wu and T. Aizawa, “Using Adaboost to Detect and Segment Characters from Natural Scenes”, Proc. Intl. Workshop Camera Based Document Analysis and Recognition, 52 - 59, 2005.

[13] E. Badekas, N. Nikolaou and N. Papamarkos, “Text binarization in color documents”, Intl. Jl. Imaging, Systems and Technolology, 16, 262 - 274, 2007.

[14] G. Nagy, “Twenty Years of Document Image Analysis in PAMI”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(1), 38 - 62, 2000.

[15] B. Epshtein, E. Ofek and Y. Wexler, “Detecting Text in Natural Scenes with Stroke Width Transform”, IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2963 - 2970, 2010.

[16] ICDAR 2003 Robust reading competition dataset, available at http://algoval.essex.ac.uk/icdar/Datasets.html.