IMAGE SEGMENTATION AND TEXT EXTRACTION: APPLICATION TO THE EXTRACTION OF TEXTUAL INFORMATION IN SCENE IMAGES

Danial Md Nor¹, Rosli Omar², M. Zarar M.Jenu³, and Jean-Marc Ogier⁴

¹,²,³ Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Onn Malaysia 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA.

⁴L3I University of La Rochelle, 17042 La Rochelle cedix 1, FRANCE.

¹danial@uthm.edu.my, ²roslio@uthm.edu.my, ³zarar@uthm.edu.my, ⁴jean-marc.ogier@univ-lr.fr

This paper proposes a solution to the problem of extraction of textual information in presentation scene images. The problem has gained numerous of interest by document analysis and recognition (DAR) community. As an extension in DAR, new research domain, Camera Based Document Analysis and Recognition (CBDAR) has been established which deals with the textual information in scene images taken by low cost hand held devices like digital camera, cell phones, etc. A lot of applications like text translation, reading text for visually impaired and blind person, information retrieval from media document, e-learning, etc., can be built using the techniques developed in CBDAR domain. The proposed approach of extraction of textual information is composed of three steps: image segmentation, text localization and extraction, and Optical Character Recognition. First of all, for pre-processing the resolution of each image is checked for re-sampling to a common resolution format (720 X 540). Then, the final image is converted to grayscale and binarized using Otsu segmentation method for further processing. In addition, looking at the mean horizontal run length of both black and white pixels, the proper segmentation of foreground objects is checked. In the post-processing step, the text localizer validates the candidate text regions proposed by text detector. We have employed a connected component approach for text localization. The extracted text is then has been successfully recognized using ABBYY FineReader for OCR.

Keywords: Image Segmentation, Scene text, Textual Information, Text Extraction, OCR

Scope: Science Engineering.

1. Introduction

The presence of the text data in images and videos containing information that is useful for clearing automatically, indexing, and structuring an image. In the extraction of this information involves the detection, localization, tracking, extraction, enhancement, and recognition of text from the image provided. Nevertheless, because of differences change the text size, style, orientation, and alignment, and low image contrast and complex background to the problem of automatic text extraction extremely challenging [1]. Although a comprehensive survey of related problems such as face detection, analysis, documents, and indexing of images & videos can be found, the problem of extracting the text information is not well explored. In this paper, we present a generic framework and methodology for automatically extract the text content of the image obtained from slide video recordings. In particular, we use the video lectures as our initial target because it can be the basis for other scenarios such as meetings and conferences. In addition to OCR, we also discuss how the unique information available in text layout. This information could be used for indexing and retrieval.
2. Related Work

Development and progress of various approaches to the extraction of text information from the image and video have been proposed for specific application, including page segmentation [2], text color extraction [3], video frame text detection [4] and content-based image or video indexing [5, 6]. However, extensive research, it is not easy to design series general-purpose systems. This is because there many possible sources of variation when extracting text. Shaded from the textured background or, from the low-contrast or complex images, or images with variations in font size, style, color, orientation, and alignment. This variation makes the problem very difficult to draw automatically. Generally text-detection methods can be classified into three categories. The first one consists of connected component-based methods, which assume that the text regions have uniform colors and satisfy certain size, shape, and spatial alignment constraints. However, these methods are not effective when the text have similar colors with background. The second one consists of the texture-based methods, which assume that the text regions have special texture. Though these methods are comparatively less sensitive to background colors, they may not differentiate the texts from the text-like backgrounds. The third one consists of the edge-based methods [7]. The text regions are detected under the assumption that the edge of the background and the object regions are sparser than those of the text regions. However, this kind of approaches is not very effective to detect texts with large font size. Chen et al. [4] compared the Support Vector Machines (SVM) based method with the multilayer perceptrons (MLP) based one for text verification over four independent features, namely, the distance map feature, the grayscale spatial derivative feature, the constant gradient variance feature and the DCT coefficients feature. They found that better detection results are obtained by SVM rather than by MLP. Multi-resolution-based text detection methods are often adopted to detect texts in different scales [8]. Texts with different scales will have different features in the each sub-band. Integrating the detected text in different sub-bands achieves better performance. Furthermore, the available redundant temporal information of videos is often used in candidate text region verification and falsely detected text region elimination. Multi-resolution text regions can be detected from the edge maps down-sampled from that of the original images. Moreover, they also made full use of the temporal redundancy to eliminate the falsely detected text regions [9]. The main contribution of this paper lies in the following three aspects: (1) it is a fast OCR method as in Fig.1, (2) it is proposed for feature extraction of textual information, and (3) it is enhancing textual information for its feature extraction.

3. Text Extraction

The aim of Optical Character Recognition (OCR) is to classify optical patterns (often contained in a digital image) corresponding to alphanumeric or other characters. The process of OCR involves several steps including segmentation, feature extraction, and classification. In principle, any standard OCR software can now be used to recognize the text in the segmented frames. However, a hard look at the properties of the candidate character regions in the segmented frames or image reveals that most OCR software packages will have significant difficulty to
Document images are different from natural images because they contain mainly text with a few graphics and images. Due to the very low-resolution of images of those captured using handheld devices, it is hard to extract the complete layout structure (logical or physical) of the documents and even worse to apply standard OCR systems. For this reason, a shallow representation of the low-resolution captured document images is proposed. In case of original electronic documents in the repository, the extraction of the same signature is straightforward; the PDF or PowerPoint form of the original electronic documents is converted into a relatively high-resolution image (TIFF, JPEG, etc.) on which the signature is computed. Finally, the captured document’s signature is compared to with all the original electronic documents’ signatures in order to find a match.

4. Experimental Results

4.1 Image segmentation

First of all, the resolution of each document i.e. both the image version of the original electronic documents and the pre-processed captured documents is checked for re-sampling to a common resolution format (720 X 540). If the perceived document image is resolution-wise different than the common resolution, then the re-sampling is required to bring the image up to the common resolution. This is necessary as in this scenario, the geometrical properties of the various visual features are considered during the matching for the identification of the captured document images. Due to poor resolution, it is not feasible to go up to the character level as long as the adjacent characters are overlapped in the captured documents. First, the captured document images are pre-processed for the perspective correction and noise removal. Then, the final image is converted to grayscale and binarized using Otsu segmentation method for further processing [Otsu, 1979]. Furthermore, looking at the mean horizontal run length of both black and white pixels the proper segmentation of foreground objects is checked. For example, for the document images having dark background and light foreground, the output of the binarization is reversed i.e. black background (represented as 0’s) and white foreground (represented as 1’s) (Fig. 4.1).

Figure 4.1: Otsu segmentation: (a) original slide document, (b) output of Otsu segmentation

and white pixels in the output of the Otsu segmentation is computed. Normally, the mean horizontal run length of the background pixels is much higher than that of foreground pixels. If the mean horizontal run length of the black pixels is comparatively higher than that of the white pixels in the binary images of the output of Otsu segmentation then the black and white pixels are simply swapped for the required image. Fig.1 illustrates one of such images having dark background and lighter foreground. For this particular image (Fig. 4.1(b)), the mean horizontal length of the black pixels is 32.3 and for white pixels it is 6.1. The image in Fig. 6.1b is
4.2 Text localization and extraction

4.2.1 Run-Length Smoothing Algorithm

The method RLSA is applied row-by-row and column-by-column to the above mentioned binary document images representing white pixels by 1’s and black pixels by 0’s. The RLSA transforms a binary sequence $x$ into an output sequence $y$ according to the rules described by Behera as follows [10]:

i) 1’s in $x$ are changed to 0’s in $y$ if the number of adjacent 1’s is less than a pre-defined limit, $T$.

ii) 0’s in $x$ are unchanged in $y$.

For example, with $T = 5$ the sequence $x$ is mapped into $y$, which is illustrated in Fig. 4.2.

$$x = (1110111110110111111111011111111111111111111111)$$

$$y = (000011111100111111111111111111111111111)$$

**Figure 4.2: RLSA algorithm (adapted from [10])**

The basic idea of the RLSA is to connect the neighboring black areas when they are separated by less than $T$ pixels. The degree of connectivity depends on $T$, the distribution of white and black pixels in the document and the ‘dpi’ (dots per inch) resolution of the document. The two distinct bit-maps are generated using the RLSA in both horizontal and vertical directions. Often, the spacing between the components in the document image tends to differ horizontally and vertically. Therefore, two different thresholds $Th$ and $Tv$ are used for the RLSA in respective horizontal and vertical direction. For the slide documents, these thresholds are tuned as $Th = 80$ and $Tv = 100$. The two bit-maps of respective RLSA output in horizontal and vertical directions are combined using a logical AND operator to detect various components in the document images. Additional horizontal smoothing using the RLSA ($Ts = 15$) produces the final segmentation result. Fig. 4.3 illustrates the RLSA algorithm in horizontal, vertical and combining output of the image in Fig. 4.3 (c). The values of these thresholds have been evaluated and tuned using about a hundred slide images.

**Figure 4.3: Output of the RLSA: (a) horizontal direction, (b) vertical direction and (c)**
combining both the directions, of the binary image in Fig. 4.1 (b)

4.3 Optical Character Recognition

Binarization is achieved with a gray threshold value derived from Otsu’s [38] method. Additional steps are done for a better binary image. Firstly, a horizontal projection profile analysis is done to detect white text on black background. If the text color is deemed to be white, the OT-region is inverted. This is necessary as most OCR software works best with black text on white background. An example of Otsu segmentation is depicted in Fig. 4.4.

Finally, a vertical projection profile analysis is done to discard unwanted pixel regions that are less than a calculated threshold. The final binarized image is shown in Fig. 4.5 (a) and its connected components is illustrated in Fig. 4.5 (b). Connected components will be explained later. We used ABBYY FineReader 8.0 (http://finereader.abbyy.com) for OCR. We found that it was sufficient for our purpose.

Figure 4.4: Otsu segmentation: (a) original slide document, (b) output of Otsu segmentation

Figure 4.5: The extracted text; (a) result for OCR of a binarized frame, (b) connected components.
4.3.1 Analysis of Extracted Characters

ABBYY FineReader is a good OCR software but not for low resolution documents. After extracting the individual characters in a document and determining their properties, we compare them to the original set. Letters like "g", "j" & "y", which extend below the baseline are in one group, tall letters, such as "I" and "T" are in another, short characters, like "a" and "c" another, and floating characters, such as "'" and """" are in the last. Once classified, an extracted character is compared to learned characters which are in the same group. If no good match is found, the extracted character is then compared to the other original characters, regardless of group. Because we found that some characters made it past the original character recognition algorithm, we deemed it necessary to perform additional operations on poorly recognized characters. The mainly observable cause of misrecognition in our original program was linked characters as wide character. An "r" would just barely touch an "i", and the character would be recognized as an "n". To alleviate this problem, we split the character at its most narrow point. This algorithm could possibly cause problems with something like "mi" -- with a poorly scanned "m", the joined character could be broken in the middle of the "m", find an "n", and do something unpredictable with the remnants of the "m" and the "i". Another common cause of misrecognition were split characters. An "i'" might be split down the middle, leaving an "l"-like figure on the left, and something incomprehensible on the right.

5. Enhancing Textual Information

In addition to OCR which was obtained from the binarized frame, we propose several other unique features that can be used to represent low resolution images. The layout shape signature can be used as a feature vector to measure the distance between two images in a query by example searching system. Here we briefly discuss the connection of two features derived from the results of the extraction of textual information Length of Sentences and the Minimum Spanning Tree of Sentences (MST).

5.1 Length of Sentences

Using text as a feature vector will result in dissimilarity between the original text and the generated ones. Hence an algorithm is proposed to classify horizontal words into three categories namely short, medium and long sentences. The sentences are determined from a number of connected components as in Fig 5.1. A connected component is by a set of character.

![Figure 5.1: The histogram of classified connected components.](image-url)
5.2 Minimum spanning trees

Basically a minimum spanning tree is a subset of the edges of the graph, so that there’s a path from any node to any other node and that the sum of the weights of the edges is minimum. We consider each sentence as a node. Here’s the minimum spanning tree of the example (Fig.5.2) of Prim’s Algorithm by Robert C. Prim. The algorithm (greedily) builds the minimal spanning tree by iteratively adding nodes into a working tree:

1. Start with a tree which contains only one node.
2. Identify a node (outside the tree) which is closest to the tree and add the minimum weight edge from that node to some node in the tree and incorporate the additional node as a part of the tree.
3. If there are less then n – 1 edges in the tree, go to 2

For the example graph, here’s how it would run. The resulting graph will have 8 MST as in Fig. 5.1 (c).

![Figure 5.1: Prim’s Algorithm; (a) Nodes, (b) Edges, (c) MST](image)

6.0 Conclusions

Image Segmentation is an important task and requires careful scrutiny. We have presented an innovative and novel framework for the separation of information in an image. This research, which aims to obtain text information including OCR and layout signature, has focused on the captured images from slide show videos. While OCR enables us to search words in the document, layout signature will be used in indexing. The good indexing scheme of an image or frame will be able to produce fast and accurate retrieval. However, the results are very dependent on the quality of OCR images or documents.

In general, the proposed algorithm can give a reliable OCR results. This allows keyword searching for the retrieval to be implemented. However, for low-resolution images, we propose a new signature taken from textual information. A new feature from textual information that we obtain as categorized nodes and the MST is unique for an image. In the future, we intend to develop Content-Based Indexing and Retrieval (CBIR) system using the obtained feature vector. We believe that through proper indexing, the proposed CBIR system is able to address the problem of low resolution document. In this paper, our focus is on scene images from the meetings and conferences. After we have an efficient CBIR, it is not difficult to make a similar system for video and image from other scenarios.
References