DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-162

A Strip-based Fast Text Detection for Low memory devices

Jobin J Mathew[a], Yue Wang[a], Eli Saber[a], David Larson[b], Peter Bauer[b], George Kerby[b], Jerry Wagner[b]; Rochester Institute of Technology, Rochester, NY; Hewlett Packard (HP) R&D, Boise, ID

Abstract

This paper proposes a strip-based fast and robust text detection algorithm for low cost embedded devices such as scanners/printers that is designed to operate with minimal memory requirements. Generally speaking, the unavailability of the whole document at once along with other memory and processing speed constraints pose a significant challenge. While conventional approaches process the whole image/page with intensive algorithms to get a desirable result, our algorithm processes strips of the page very efficiently in terms of speed and memory allocation. To this effect, a DCT block based approach along with appropriate pre and post-processing algorithms is used to create a map of text pixels from the original page while suppressing any non-text background, graphics or images. The proposed algorithm is able to detect text pixels from documents of varying backgrounds, colors and non-textual portions. This algorithm is simulated in both MATLAB and C programming languages and tested using a Beagle Board to simulate a low processing CPU on a wide variety of documents. The average execution time for a full 8.5x11 page scanned at 300 dpi is approximately 0.5 sec. in C and about 3 seconds on the Beagle board.

Introduction

Text detection is a widely researched area and is an integral part to many applications such as document rendering and enhancement, character recognition, etc. In low cost embedded devices - such as low to mid-volume scanners/printers - small memory buffers, processing time and strip-based inputs, significantly limit the detection and enhanced rendering of text documents. In these devices, a document page is scanned by first dividing it into multiple strips - due to limited memory resources. So at any given time, only a strip of the page is available to the processor. Hence, a fast and accurate text detection algorithm, which can process the page in individual page-strips, is highly desired to efficiently and effectively render and print documents with enhanced text. This detected text information can also be used for document tagging, search and character recognition, etc. The field of optical character recognition is a well-explored one and has many efficient algorithms available. However, separating the text and non-text regions is not completely solved and algorithms can be found in the current literature that explores this problem.

The algorithms used for text detection can be categorized based on the features they use such as Edge based methods, Texture based methods, Connected-component based, Stroke based and other methods including Maximally Stable Extremal Regions, segmentation and classification methods that use different training techniques [3]. While simple edge-detection based methods are easy to implement, the detection accuracies are usually poor. As the complexity of the algorithm increases, the accuracies do increase but also does the computation time and use of resources including memory.

A Discrete Cosine Transform (DCT) domain text detection by S Lu et al.[6] utilized weighted DCT coefficients as the discrimination parameter for text regions. A Linear Discriminant analysis is also performed on the sum of the coefficients to find a threshold for separating the text. The scenario of text detection is not limited to documents but used in a wide array of applications such as signboards, images, videos, etc. A survey of text information extraction from images and videos by K Jung et al.,[5,7] uses different steps for this extraction such as detection, localization, tracking, enhancement, recognition. The video domain text detection also uses similar methods as the document/image counterpart. Edge based methods make use of the contrast difference between text and background, differential filtering approaches and other edge detectors. Texture based approaches involves the use of Fourier or Wavelet transforms, Gabor filters, etc. and mostly operates based on the spatial variance. Since the videos are in a compressed format, some detection methods involve decoding and making use of the DCT and motion vectors by localizing and tracking blocks of data using block texture intensity projection profiles. The text information extraction methods are widely used in video/image content analysis and indexing, content-based video coding and decoding, etc. To achieve higher accuracies combination of different techniques are used; for instance, classification methods combined with edge analysis and filtering are used in video detection by comparing the high contrast and low contrast video images which tend to increase detection rates compared to using these methods individually [8]. The text to be detected can be present in different languages and many algorithms in the literature are capable of detection in other languages than English. A.R. Chowdhury et al. [4], uses text detection on two major Indian scripts present in natural scene images. The stroke thickness of the scripts is used to distinguish the scripts from the non-text regions.

This research deals with developing a text detection algorithm for scanned documents with emphasis on the following three constraints. The available memory in the device hardware is low. The scanned page is saved in the buffer in strips of around 256 rows by the page width. This implies that only a portion of the whole page is available for processing at any given time. Secondly, the algorithm should limit the use of memory consuming image-processing techniques. Additionally, each stage of processing should be independent of the previous stage such that minimal amount of memory is utilized. Finally, a speed constraint requires all the processing and detection work be done in the least amount of time so as to not inhibit consumer workflows with unwarranted latencies. In that light, the text detection needs to be completed once the scanning process is completed (which is about one to three seconds).

In this paper, we present an approach to segment or extract the text pixels out of a document while considering the above-mentioned constraints of low memory, speed and unavailability of the whole document at once. A Discrete Cosine Transform based approach is proposed where the document is first preprocessed and then a DCT-based frequency filtering is applied which removes most of the non-text components while preserving text regions. A

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-162

series of post-processing approaches such as morphology, connected components, etc. are used next to suppress the non-text areas while enhancing the text pixels. The next sections of this paper discusses about the strip-based approach followed by the DCT-based algorithm and finally the observations/results and conclusions.

Strip-based approach

The extraction of text from a document in majority of the applications has the availability of whole document at the initial processing stage. This advantage can be explored using a variety of intelligent algorithms to effectively segment and separate the text. In contrary to this, one of the constraints in our problem is that the whole document page is not available at once. The term 'strip-based' is used in this regard which means only a strip of the page is available for processing at a given time. The size of a single strip can vary between 64 to 256 rows by *n* columns for an *m* x *n* image. Thus a single strip of document contains all the columns but limited number of rows. This approach is used in low-memory devices where memory cache is not large enough to hold a full image and thus memory has to be freed frequently. Due to this, the text in a strip can be partial depending on the page resolution, strip size and the document type. A block-based approach without any neighboring block dependency is used to deal with this constraint while seldom affecting the detection accuracy.

Properties of text and assumptions

Text has some unique properties compared to other areas in a document such as images, graphics, clip arts etc. and these observations are used to design certain stages of the algorithm. Certain assumptions are also made to narrow down the detection problem. The language of the selected input documents is English; detection in documents with other languages will be addressed in the future phases of this project. Although the font size in documents can vary a lot, in a general day-to-day document class majority of the text falls in the range of 8-24 points. The approach presented in this paper focuses on detecting text which falls in this size range.

The majority of text regions in a document are structured (as sentences or paragraphs) with spaces separating each word and each row of text line. The non-text areas of the document do not have any such pattern throughout. This is also useful in designing the algorithms. The background of the document has a significant effect on the detection rate. If the text is dark on a white or light colored document, the detection accuracies are high while it falls when text is light or colored with a non-white background since in this case it has low contrast compared to the background. The contrast difference between text and background in a document has a major effect on the detection accuracy; a strong contrast difference indicates higher detection accuracies.

Text Detection algorithm

In light of the above, the text detection algorithm was designed based on the constraints associated with this problem. The limitation of processing the whole page at once immediately suppresses the scope of using algorithms which operate on the whole page. The low memory problem contributes to another constraint where the algorithm never uses a large part of the memory. In short, the algorithm needs to utilize the properties of text and process the document effectively such that processing an document strip is fast and independent of the previous or latter strips of the document. A combination of frequency methods and block-based approaches were used and the algorithm is fairly accurate in detecting and separating text regions from non-text portions while meeting all the requirements.

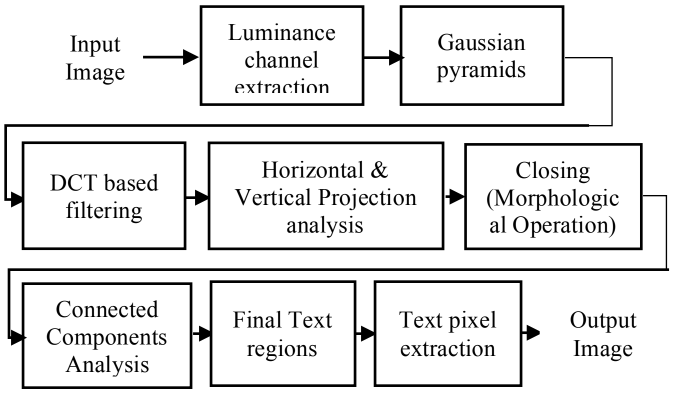
Input Image
Luminance channel extraction
Gaussian pyramids
DCT based filtering
Horizontal & Vertical Projection analysis
Closing (Morphologic al Operation)
Connected Components Analysis
Final Text regions
Text pixel extraction
Output Image

Figure 1. Block diagram of the proposed Text detection algorithm

The block diagram of the algorithm is shown in Figure 1 and can be divided into three major parts. The first is a preprocessing and Discrete Cosine Transform (DCT) based processing where the high frequency components are extracted from the original image after smoothing it. The second stage applies morphological operations and projection analysis on the filtered image to remove false positives and enhance text regions. The final step reduces any noises or falsely detected text regions through a mixture of connected component analysis and fast region classification based on some observed text properties.

Pre-processing – Gaussian pyramids

The pre-processing stage prepares the scanned image such that the text regions are highlighted more compared to other portions of the page. The input document is a three-channel color document, which is scanned at 300 dpi and contains one Luminance channel and two chrominance channels. A strip of the page contains about 256 rows by *n* columns and each strip is processed individually to produce an output strip. The luminance-channel is extracted and used for the detection while the chrominance channels does not contribute to the text detection. This gray-scale/Luminance image is downsized using a Gaussian pyramid where the image is downsized by a factor of 4 in two stages of 2 each. Between each downsizing, a low pass/blurring is applied with a 3x3 Gaussian low-pass filter. The pyramid downsizing effects the processing in two ways: one is that by downsizing huge images the processing speed is greatly improved and since text usually has contrast and sharp edges, the image quality is not compromised. Second, the low-pass blurring employed after each downsizing stage reduces the high frequencies in the images and suppresses insignificant non-text regions. The level of smoothing in this stage affects the frequency processing in the successive stages. A highly smoothed image would result in high frequency components with lower magnitudes and thus some of the text can be miss-detected. While a non-preprocessed image might contain lot of strong high frequency components, which can also contribute from the non-text regions. Thus the filter size and components in the Gaussian pyramid stage are fine tuned such that the smoothing does not highly suppress the text but fairly suppress images and thus text can be easily detected in the frequency

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-162

processing with minimal miss-detections. The pyramid-downsized and smoothed strip is passed to the next step for frequency domain processing.

DCT based filtering

Discrete Cosine Transform is an efficient transformation technique based on the Fourier Transform. For a strip-based algorithm, a block-based DCT approach is a suitable method, which can utilize the limited availability of the page/data. A DCT is used to process the strip in blocks of size 8x8 while converting and separating the block into high and low frequency components. A high-frequency portion of the cosine-transformed block is retained while most of the low and some mid frequency components are neglected. The text has edges and these edges are fairly strong compared to the document background and other graphics. The edges or high frequency parts of the document are the key regions, which can correspond to either text, or some sharp non-textual regions. Thus operating in a frequency regime necessitates the need to separate the high frequencies of text regions from non-text regions. The DCT filtering approach aids to this by filtering out the low and most of the mid-frequency components.

$$G_x(k) = \frac{2}{N}\sum_{m=0}^{N-1} X(n)\cos\frac{(2n+1)k\pi}{2N}, \quad 0 \le k < N \qquad (1)$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} w(k)G_x(k)\cos\frac{(2n+1)k\pi}{2N}, \quad 0 \le n < N \qquad (2)$$

where, $w[k] = \begin{cases} 1/2, & k = 0 \\ 1, & 1 \le k < N \end{cases}$

The DCT of an 8x8 block is computed using equation (1) where gray-level intensity values are used to calculate the magnitude of the different frequency components. The block size is selected as 8x8 since it is large enough to contain relevant textual information and small enough to make the processing faster. The inverse DCT (iDCT) equation in (2) is used to map the intensity values from the corresponding frequency components to obtain a frequency-filtered image.

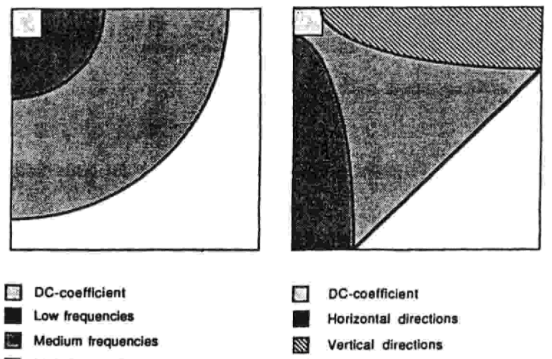

Figure 2: Frequency distribution of the 2D-DCT coefficients and the block features they represent [1]

The frequency separation using the DCT block analysis is based on the frequency distribution diagram given in Figure 2 [1]. This shows the horizontal, vertical and diagonal frequency distribution in any 8x8 DCT of an 8x8 matrix. The selected frequencies are the lower right ones and the upper left AC components are ignored. The manipulated DCT block is remapped into the spatial domain such that the remaining frequency components correspond to the high frequency areas in the image. The threshold of the low frequency elimination is set in such a way that, only strong edges are detected here. Since text have strong edges compared to the edges in non-text regions/images, this thresholding selectively eliminates most of the non-text edges, while retaining majority of the text.

Post processing

The post processing aims to enhance detected text regions and eliminate or reduce any falsely classified regions. The DCT processing eliminates most of the background in the page while retaining text and high-frequency non-text areas. The threshold is set such that majority of text pixels are detected even though false positives can occur. As mentioned earlier, the priority is to reduce the miss-detection rate rather than reducing false positives. The DCT image is a binary image, with *one* representing probable text pixels and *zero* representing background, consists of mostly non-connected text regions and the succeeding stages are designed to connect these regions, enhance them and also to reduce any non-text detections. This is performed using projection analysis, morphological operations, connected components and fast region classification. The nature of printed text shows that the variation in the text happens more frequently in the horizontal direction compared to the vertical direction. Projection analysis is performed in horizontal and vertical directions; each pixel row or a portion of each row is projected and the sums of text and background pixels are identified. The ratio of these quantities represents the contrast and text or content pixel density in comparison to the background. An empirically derived threshold is used to classify this specific group of pixels as text or non-text. The horizontal projection analysis helps in reducing and removing non-text pixels and separated individual pixels mostly belonging to non-text areas while the vertical projection analysis contributes to removing any lines or scattered pixels around the document.

A series of morphological operations (dilation, opening and closing) are used to enhance the detected text regions and eliminate random pixels. In addition, small blobs that are improbable to be text are eliminated by calculating the area of each blob and with a threshold that is set to hundred pixels. The connected component analysis is a time and memory consuming operation, so it is used at a minimal and essential level here. After the projection analysis and morphology, majority of the pixels will be grouped into blocks of text, non-text and mixed text. The mixed text blocks are difficult to classify and the above stages are designed to maximize the separation between text and image borders thereby reducing blocks of mixed text/non-text. Finally a fast region classification method is employed based on the fairly unique property of text seen in general documents. Each text line is preceded and followed by empty spaces; majority of the text regions in a document would be structured and resulting in this space between the lines. This feature is absent in images, graphics and most text-like figures present in documents. Connected component method is used here to identify pixel groups and the groups having area less than 5000 pixels are further classified using this method. Each contender is tested for this feature using a horizontal projection along each of the selected pixel groups superimposed with the original document. This classification step eliminates complex image regions in the document. The resulting image is the text region map consisting of potential text regions and is upscaled by a factor of 4 to account for the downsizing in the pre-processing stage. The final text pixel map is generated from the text region image by superimposing on the original document and using a binary thresholding operation. The final image is thus a binary image with black background and white text pixels.

Observations and Results

The input documents used for testing were obtained from a scanner at a resolution of 300 dots per inch (dpi) and saved in the TIFF format. Due to the scanner pipeline requirement, this RGB image was converted to Luminance-Chrominance image plane, which is similar to the *Lab* color space with one luminance channel and two chrominance channels. The typical size of the test image is about 2550 by 3100 by 3 channels. To improve the processing speed and reduce memory requirements, luminance channel was only used for detection it contains majority of the spatial information compared to the chrominance channels. The algorithm was written in MATLAB and also in C to help with its implementation on an embedded device. The algorithm execution in MATLAB took about 2.5 seconds and in C it took an average of 0.5 seconds per page. To evaluate the performance on a real-time embedded environment, a Beagleboard was used to process these images. The *BeagleBone Black* has a Cortex A8 1 GHZ processor with 512 MB RAM. The Beagleboard processed the images in an average time of 3 seconds, which is fast given the limited-resource embedded environment. The memory management was fine and the processing of all the images were smooth.

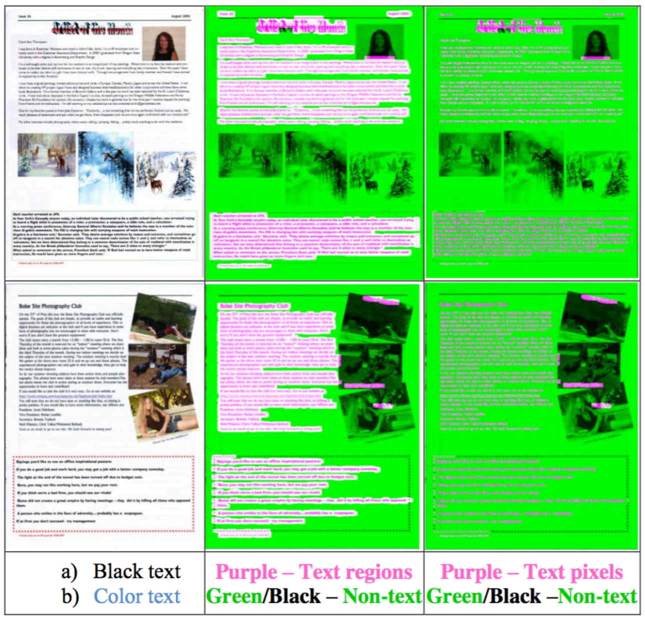

Figure 3. Results of the Text detection on color document with text and images on a white background.

The algorithm was tested on a set of over 100 images and these images were carefully selected with differing complexities, contents, colors, etc. while replicating a general day-to-day document class. The algorithm effectively segments and separates text from non-text regions in majority of the documents and has accuracies above 90s for black and color documents with white/light colored backgrounds. The accuracies were based on detected text pixel counts in comparison with a manually thresholded ground truth. Figure 3 shows two input documents that were scanned at 300 DPI and with a page resolution of 2550 by 3100 by 3 channels. These documents have black and color text on white background with text spread over different parts of the page and with mixed images. The first column shows the original color images and the luminance channel was processed using the proposed algorithm. The second column of images shows final text region image with detected text regions in purple. These text regions are superimposed on the original image and using a binary thresholding, text pixels are found out which are the images in the last column. The first stage of DCT filtering detects fairly all the text regions in the document successfully while including false positive regions. The false positives are sharp or edgy image regions present in the document and get extended through the filtering stage since the high frequency components in the DCT coefficient blocks are retained. The Gaussian pyramid stage in the pre-processing stage was added to smooth the image (while also reducing the size) and the parameters can be adjusted such that most of the non-text regions will be removed after the DCT processing. But an extreme threshold in the Gaussian pyramid stage would also result in loss of text pixels after the DCT filtering. As the emphasis is to avoid miss-detections of text, the pre-processing stage was tuned such that the text pixels were not filtered out in the first stage. The post-processing methods were carefully selected to remove the falsely detected non-text regions from the initial stages.

There were a few failure cases that occurred with a few text colors, specific backgrounds and also when complex images were present in the document. When the text color intensity in comparison to the background is low-contrast (such as light yellow on white background), false negatives arise causing missed text pixels. The background changes from light to dark colors also affect the accuracy as this results in a decreasing contrast difference between text and background. When complex images are present in the document, many false positives occur and it varies depending on the complexity of images; images with sharp or edgy portions are prone to be classified as text. The contrast difference problem can be addressed by employing a contrast-boost algorithm in the pre-processing sections and the complex image problem needs further research and will be addressed in the future work.

Conclusions

A strip-based scheme for detecting text pixels from a document consisting of images, graphics and other non-text areas is presented in this paper. The document text language was English and font sizes varying from 8 to 24 points were tested. Unlike a general text detection problem, there were a few constraints to be accounted for such as speed of the implementation, limited availability of a page for processing and memory constraints. The proposed algorithm proved to be effective and efficient on a wide variety of documents and smoothly running on an embedded environment while meeting the above constraints. A more general algorithm that can be applied to different languages and bigger fonts is currently under research while also accounting for the failure cases.

Acknowledgements

The research project was supported by Hewlett Packard (HP) Company in collaboration with the Chester F. Carlson Center for Imaging Science at Rochester Institute of Technology.

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-162

References

[1] K.R. Rao, P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications", Academic Press Ltd, 1st edition, 1990.

[2] A.K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, 1989.

[3] H Zhang, K Zhao, Y. Song, J. Guo, "Text extraction from natural scene image: A survey" Jour. Neurocomputing 122, pp. 310-323, 2009.

[4] A.R. Chowdhury, U. Bhattacharya, S.K. Parui, "Text detection of two major Indian scripts in natural scene images", CBDAR 2011, LNCS 7139 pp 42-57, 2012.

[5] K. Jung, K.I. Kim, A.K. Jain, "Text information extraction in images and video: a survey", Journal of Pattern Recognition Society, Elsevier, 2004.

[6] S. Lu, K.E. Barner, "Weighted DCT coefficient based Text Detection", ICASSP, IEEE 2008.

[7] X. Qian, G. Liu, H. Wang, R. Su, "Text detection, localization and tracking in compressed video", Signal Processing: Image Communication, pp. 752-768, Elsevier, 2007.

[8] P. Shivakumara, W. Huang, T.Q. Phan, C.L. Tan, "Accurate video text detection through classification of low and high contrast images", Journal of Pattern recognition, Elsevier, 2010.

Author Biography

Jobin J Mathew received his B.Tech in Electronics & Communication from the Kerala University, India (2010) and his MS in Electrical Engineering from Rochester Institute of Technology (RIT), NY (2014). He is currently pursuing PhD in Imaging Science at Chester F Carlson Center for Imaging Science, RIT, NY. The Fast Text Detection was a research project funded by Hewlett-Packard (HP).