A New Gradient based Character Segmentation Method for Video Text Recognition

[a]Palaiahnakote Shivakumara, [a]Souvik Bhowmick, [a]Bolan Su, [a]Chew Lim Tan and [b]Umapada Pal
[a]School of Computing, National University of Singapore, Singapore
{shiva, subolan, tancl}@comp.nus.edu.sg
[b]Computer Vision and Pattern Recognition Unit, Indian Statistical Unit, Kolkata, India
umapada@isical.ac.in

Abstract— **The current OCR cannot segment words and characters from video images due to complex background as well as low resolution of video images. To have better accuracy, this paper presents a new gradient based method for words and character segmentation from text line of any orientation in video frames for recognition. We propose a Max-Min clustering concept to obtain text cluster from the normalized absolute gradient feature matrix of the video text line image. Union of the text cluster with the output of Canny operation of the input video text line is proposed to restore missing text candidates. Then a run length algorithm is applied on the text candidate image for identifying word gaps. We propose a new idea for segmenting characters from the restored word image based on the fact that the text height difference at the character boundary column is smaller than that of the other columns of the word image. We have conducted experiments on a large dataset at two levels (word and character level) in terms of recall, precision and f-measure. Our experimental setup involves 3527 characters of English and Chinese, and this dataset is selected from TRECVID database of 2005 and 2006.**

Keywords- ***Video document analysis, Word segmentation, Video character extraction, Gradient features, Video character recognition***

I. INTRODUCTION

With the rapid growth in size of multimedia databases, thanks to the internet and cable TV, understanding of the contents of such databases is becoming more important for effective content-based indexing and retrieval. However, a video search function, which is categorized by manual indexing, requires excessive time and money. Text such as news caption in a video provides important information of the contents as well as description of the video scenes. Such text can be used as indices of the multimedia database. Therefore automatic extraction of video text, which aims at integrating advanced optical character recognition (OCR), is vitally useful for video annotation and retrieval systems [1]. Hence video text extraction and recognition is crucial to the research in all video indexing and summarization [1-6].

Video text recognition is generally divided into four steps: detection, localization, segmentation, and recognition. The detection step roughly identifies text regions and non-text regions. The localization step determines the accurate boundaries of the text portion. The segmentation finds the character and the word boundary from the text line [7]. There are several algorithms that are reported in the literature for accurate text detection and they have achieved good accuracy even for scene text detection and multi-oriented text detection [8-13]. Therefore, in this work we use the method that works for multi-oriented scene text reported in [13] to locate boundary of text portion in video image. It is also true that the recognition of characters has become one of most successful application of technology in the field of pattern recognition and artificial intelligence. However, OCR systems are developed for recognizing characters written on clear paper. Applying the current OCR system directly on video text leads to poor recognition rates, typically not more than 45% [2]. This is because text characters contained in video have low resolution and embedded with complex backgrounds. To recognize these video characters, it is necessary to segment character and words from the text portion properly, even when the whole text string is already well located. Therefore, segmentation is important and research is badly needed to meet requirements of real time applications such as video events analysis and sports events analysis etc. Hence, we focus on character and word segmentation from detected text lines in video frames.

There exists many text region segmentation methods and they are divided into three classes. Methods in the first class use either global or local or multilevel thresholds to retrieve text region. The second class uses stroke based methods to retrieve text region. These features are used to enhance the contrast at edges that are most likely to represent text. The third class is the color-based methods. However, performance of these methods is poor because of complex background and unfavorable characteristics of video. Recently, a language independent text extraction method [7] is proposed which works based on adaptive, dam point labeling and inward filling. However, this method is sensitive to complex background images.

Chen et. al. [3] proposed a two-step method for text recognition. The first step uses edge information to locate the text and the second step uses features and machine learning to recognize the segmented text. Since the method uses edge information to locate the text, it may not be robust enough for complex backgrounds. Chen and Odobez [2] have proposed a method based on Monte Carlo sampling for segmented text recognition. This method appears to be expensive as it uses probabilistic Bayesian classifier for selecting thresholds. The method requires a sequence of frames to achieve accuracy. However, it may be hard to decide the number of frames for enhancement [14]. Another method for low resolution video character recognition is proposed based on a holistic approach and a component analysis. However, it requires a large number of training samples. There are methods which propose robust binarization algorithm to improve the recognition rate of video character recognition [15-16]. Recently, Zhou et al. [17]

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.34

developed binarization method for video text recognition. This method uses Canny information to binarize and it achieves a reasonably good accuracy compared to baseline thresholding methods. However, these methods focus on graphics text recognition but not on scene text recognition and hence their error rates are higher if scene text is present in the frame.

Most of the previous methods use segmented region by the text detection method as input for binarization and recognition, focusing on graphics text and horizontal text. None of these text region segmentation methods have a perfect solution to the problems of complex background, multi-oriented text and scene text in the video. However, we have found a work that performs character segmentation from the detected text region [7] based on an assumption that characters have uniform color and the text is in horizontal direction. These assumptions may not be valid in the case of scene text. Nevertheless, we are inspired by this work to propose a new method for word and character segmentation from detected text before binarization recognition to achieve better accuracy even for scene text and multi-oriented text in video.

II. PROPOSED APPROACH

Since our focus is on word and character segmentation from video text lines that are detected by the text detection method, we use a method proposed by us in [13] for video text detection. The reason for choosing this method is that the method is able to locate both graphics and scene text and multi-oriented text in complex video background despite its low resolution. In order to ease the problem due to multi-oriented scene text, we take advantage of the angle of text line determined by the text detection method during bounding box fixing. We then propose Bresenham's line drawing algorithm [19] to identify the text pixel direction. As a result, we convert text lines of any direction into horizontal text lines. Hence the problem of multi-orientation of text line has been simplified to the problem of horizontal text line.

The proposed method is structured into three subsections. Bresenham's line algorithm for handling multi-oriented text is described in Subsection A. The gradient features are explained in Subsection B with illustration for obtaining text cluster using Max-Min clustering concept. In Subsection C, we propose a method to combine text cluster obtained in Subsection B with a Canny operation on the input image to restore the missing text candidates for word segmentation using the run length concept. Subsection D presents the proposed character segmentation method from the segmented word image based on text height difference (THd), a top distance and a bottom distance vectors.

A. *Bresenham's Line Algorithm*

The Bresenham's Line Drawing algorithm is used to determine the points in an n-dimensional raster that should be plotted to form a close approximation to a straight line between two given points. Bresenham's algorithm chooses the integer *y* corresponding to the pixel center that is closest to the ideal (fractional) *y* for the same *x*; successive columns y will either remain the same or increase by 1. The general equation of the line through the two endpoints (x_0, y_0) and (x_1, y_1) is given by: $\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$. Since x (column), the pixel's row, y is given by rounding this quantity to the nearest integer:

$$y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0$$

The slope $(y_1 - y_0)/(x_1 - x_0)$ depends on the endpoint coordinates only and can be pre-computed, and the ideal y for successive integer values of x can be computed starting from y_0 and repeatedly adding the slope. In this way, Bresenham's line algorithm helps in finding the direction of pixel to convert any given direction of text line into horizontal text line as shown in Figure. 1. However, the quality of image degrades somewhat. It can be seen in Figure. 1.

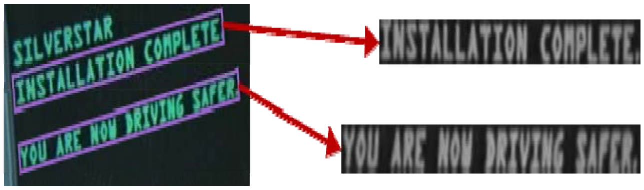


Figure 1. Rotation of non horizontal text line into horizontal using Bresenham's line drawing method

B. *Gradient Features*

Since video images have low resolution and complex background, we propose gradient features for separating text and non-text pixels as gradient gives high value for text and low value for non-text pixels. For a given horizontal text line detected by the text detection method [13], we use [-1 1] sliding window to obtain the gradient features corresponding to each pixel in the text line gray image. It is observed in [18] that performing this mask over a text frame produces high negative and positive gradient values for text pixels. Hence the absolute value of gradient helps in enhancing text pixels in video frame. The gradient image is determined from the gray image as follows.

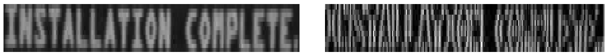
$grad(x,y) = img(x+1,y) - img(x,y)$. For the obtained gradient image, we take the absolute value to consider high negative values. We normalize the gradient matrix as

$$norm\_grad(x,y) = \frac{abs\_grad(x,y) - \min\_ag}{\max\_ag - \min\_ag}$$

where $\mathrm{min\_ag = minimum(abs\_grad(x,y))} \,\forall (x,y)$

and $\mathrm{max\_ag = maximum(abs\_grad(x,y))} \,\forall (x,y)$.

Figure 2 shows the effect of gradient operation on gray text line image and it can be seen in Figure 2(b) that text pixels are brightened compared to non-text pixels. Further to confirm the usefulness of gradient operation, we plot a graph normalized gradient values vs columns for the middle row of the image shown in Figure 2(b) in Figure 3 where one can notice clearly a big gap having zero gradient values for the word gap marked by ellipse in Figure 3.



(a). Gray text line (b) Normalized gradient image

Figure 2. Sample result of gradient operation

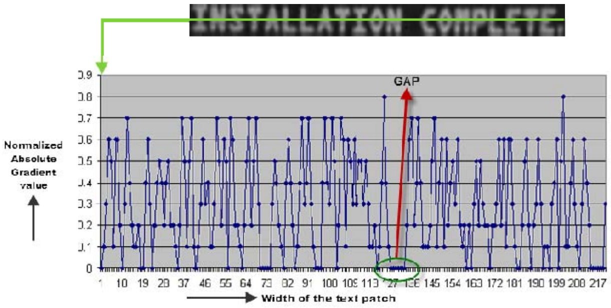
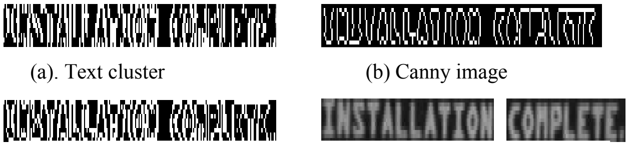


Figure 3. Normalized absolute gradient information for word gaps at the middle row of text patch

For the normalized gradient matrix, we propose a Max-Min clustering algorithm to separate text cluster from the non-text cluster rather than using a threshold as deciding a threshold is hard in case of video images. The Max-Min clustering method selects Max and Min values in the normalized gradient feature matrix and then it compares each value in the gradient feature matrix with Max and Min chosen values to find its nearest neighbor. i.e the value which is close to Max is classified as text and the value which is close to Min is classified as non-text. This results a text cluster. One sample output of clustering is shown in Figure 4(a) for the image in Figure 2(b) where it is observed that most of the pixels are classified as text pixel correctly.



(a). Text cluster (b) Canny image

(c). Union of (a) and(b) (d) Word are segmented

Figure 4. Word gap identification

C. Word Segmentation

Max-Min clustering gives a text cluster but that is still not sufficient to identify the gap between words due to loss of some textual information of the image. To restore the lost text information, we propose a union operation of the Canny edge image obtained by applying Canny edge detection algorithm on the input image as shown in Figure 4(b) for the image in Figure 2(a) with the text cluster obtained by Max-Min clustering as shown in Figure 4(c). As it is known that Canny definitely gives edges for text and that can be used for word segmentation but not for character segmentation due to erratic edges at the character background. Hence the union operation helps in filling the text region and leaving a gap between words.

The gap between the words is identified by introducing the run length information. This idea works when the image has a high number of same consecutive black pixels (background). It is true that where there is a gap between words there we get consecutive black pixels in high number but not in between characters. Hence, this idea gives good results for word segmentation as shown sample results in Figure 4(d) where it is observed that clear space between the words and correct segmentation of words.

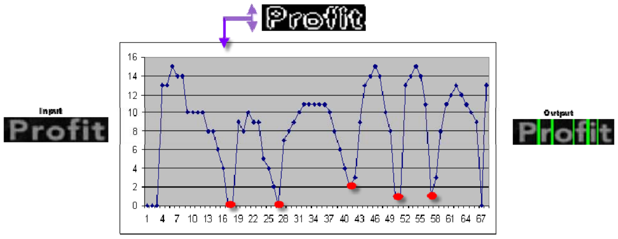


Figure 5. Gap detected depending on the Tv vector

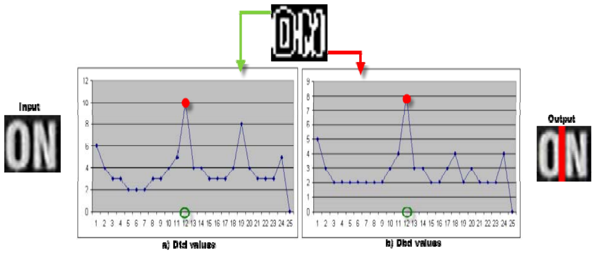


Figure 6. Finding a gap candidate column depending on Dtd and Dbd vectors

D. Character Segmentation

The run length concept used for word segmentation does not work for character segmentation as it is noticed that there is no such a high number of consecutive black pixels between characters. Therefore, we propose a new method based on text height difference (THd), top distance and bottom distance vector of the above union operation. The text height difference vector is defined as the distance between the first pixel and the last pixel of each column in the restored word image. If THd is less than two pixels, we consider that gap as a true character gap. It is illustrated in Figure 5 where the gaps identified by THd are marked by red dots. (In Figure 5, x axis shows column number and y axis shows THd values). The final character segmentation is shown in a word "Profit" on the right side of Figure 5. If THd is greater than 2 then we check the top and bottom distance vectors. The top distance vector (Td) is defined as the distance between the upper boundary of the text and the topmost black pixel of each column and the bottom distance vector (Bd) is defined as the distance between the bottom boundary and the bottommost black pixel of each column. Then the method finds the difference between consecutive distance values in Td and Bd in each column to identify the depth (high difference when character boundary exists between the characters), which is denoted as Dtd and Dbd, respectively. When there is a gap between characters, both Dtd and Dbd give high values and hence it is considered as candidate gaps.

We have observed while doing experimentation that for most of the cases, touching between characters exists at the middle of character boundary but not at top and at bottom. For Dtd and Dbd values, we use the same Max-Min clustering method used in Section B for obtaining text cluster for choosing gap candidate clusters (Cluster with Max value). For each gap candidate in cluster belonging to the top distance vector, the proposed method checks whether the corresponding candidate in the gap candidate cluster obtained from the bottom distance vector is a gap candidate or not. We consider a gap candidate as a true gap candidate for segmenting the character if the above criterion is satisfied. Figure 6 illustrates character gap identification, where we can see the values of Dtd and Dbd for each column of a text patch. Now at 12th column both the Dtd and Dbd values are going towards higher, so here we can conclude that there should be a gap. (here the higher and lower are again decided by the Max-Min clustering among the values of Dtd and Dbd values separately). In the left of Figure 6, we have shown the graph obtained based on Dtd while in the right side the graphs is based on Dbd values.

Table 1.Grayscale Image with Word Gap detected

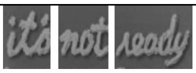
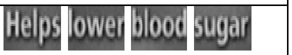
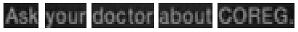
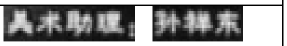
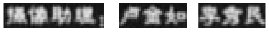
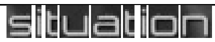
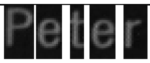
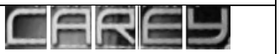
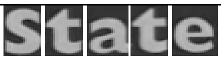
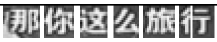
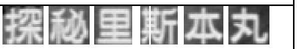
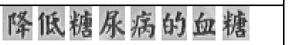
Original Grayscale	Word Gap detected
	
	
	
	
	

Table 2. Gray scale word image with character gap detected

Original Grayscale	Character Gap detected
	
	
	
	
	
	
	

III. EXPERIMENTLA RESULTS

For the experimentation purpose we have created a dataset consisting of 1200 text lines (including both horizontal and non horizontal English and Chinese languages) for word segmentation and 700 words (which results in 3527 characters) are used for character segmentation. Our database includes different kinds of multi-oriented scene text lines in order to show that the proposed method is effective for multi-oriented video text recognition.

To evaluate performance of the proposed method, we consider recall (R), precision (P) and f-measure (F) as metrics in this work. We conduct experiments on the text cluster obtained by Max-Min clustering algorithm (Gradient), Canny operation of the original text line image (Canny) and the restored image (Union) of the proposed method for word and character segmentation. Similarly, we conduct the same experiments on the Chinese data.

The sample results for word and character segmentation are shown in Table 1 and Table 2, respectively, where gaps are marked in white for different varieties of word and character images. Different performances measures of our method are given in Tables 3-5. Tables 3-5 show that F-measure of the proposed method is higher than results given by the gradient feature alone and Canny because text clustering loses significant text information when Max-Min clustering is used while Canny operation gives erratic edges due to the background complexity. On the other hand, the proposed method (Union) gives better results because of the advantage of character segmentation.

Table 5 shows character segmentation results of the proposed method for different classes of data and it does not include results on text clustering (Gradient) and Canny operation as they have already been shown to give poorer result at the word level and hence poorer result at the character level. It is noticed from Table 5 that the proposed method gives promising results for both English and Chinese horizontal and non-horizontal data.

Table 3. Results for word segmentation on English dataset

Method	English Horizontal			English Non Horizontal		
	R	P	F	R	P	F
Gradient	0.94	0.46	0.62	0.83	0.55	0.66
Canny	0.58	0.59	0.58	0.65	0.61	0.63
Union	0.93	0.73	**0.82**	0.82	0.77	**0.79**

Table 4. Results for word segmentation on Chinese dataset

Method	Chinese Horizontal			Chinese Non Horizontal		
	R	P	F	R	P	F
Gradient	0.93	0.58	0.71	0.83	0.53	0.65
Canny	0.79	0.73	0.76	0.73	0.64	0.68
Union	0.93	0.69	**0.79**	0.83	0.78	**0.80**

Table 5. Results for character segmentation on both English and Chinese dataset

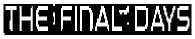
Union	R	P	F
English Horizontal	0.74	0.86	0.80
English Non-Horizontal	0.83	0.89	0.86
Chinese Horizontal	**0.87**	**0.91**	**0.89**
Chinese Non-Horizontal	**0.87**	0.87	0.87

E. Recognition Results

Our primary goal of this work is to show that segmentation of character before binarization and recognition helps in achieving good accuracy in recognition compared to recognition accuracy of binarization and recognition of whole text lines and words without segmentation. To justify that we have reported recognition results on English text here but not recognition results on Chinese due non availability of Chinese OCR. This work uses the latest binarization method reported in [20] for binarizing the given text line, word and characters before passing to OCR for recognition. The reason behind in making the choice of this binarization method is that this work has been compared with state of the art methods to prove its superiority to existing methods on binarization. The OCR engine downloaded from [21] is used for the purpose of English text recognition.

We present the recognition results at three levels namely text line, word and character levels. (Here by text line level recognition we mean to feed a text line into an OCR for its recognition; by word level recognition we mean to feed a text word into an OCR for its recognition; and by character level recognition we mean to feed segmented individual characters of a text word into an OCR for its recognition). An example of text line level based recognition results is shown in Tables 6. Here the input “THE FINAL DAYS” has been recognized as “THE F||'\AL DAVE“. The current OCR performs poorly at the text line level as low recognition accuracy of 36.5% is reported when computed from all text lines of our database. The word level recognition results obtained from the input "THE FINAL DAYS" is “THE F||'\AL DAYS”. There is improvement in recognition results at the word level over text line level as the word “DAYS” recognized correctly at the words level. As a result, higher character recognition accuracy of 62.4% is reported compared to the recognition accuracy at the text line level due to elimination of complex background by the segmentation method. Thus, it can be concluded that word extraction helps in improving recognition accuracy. In the same way, when we test OCR on segmented characters from the words, the recognition result is “THE F||NAL DAYS”. One can notice from the results at the word and character level that recognition accuracy improves at the character level over the word level. Hence, the recognition accuracy at the character level is 65.6% which is higher than that at the text line and word levels. We can observe that word and character levels recognition accuracy doesn't differ much (3.2%) and this is due to use of language model at word level. The language model helps resolving ambiguity that may occur in a word and enhances recognition rate. Otherwise, we should get much higher characters level accuracy than word level accuracy.

Table 6. Recognition results at the text line level

Input	Binarization	Recognition
		THE F||'\AL DAVE

IV. CONCLUSION AND FUTURE WORK

In this work, we have proposed a new segmentation method based on gradient features for word and character extraction from text line image and word image. We have shown that the combination of text clustering and Canny operation is better than gradient and Canny operation alone for classification of text pixels. The run length concept is applied for the first time on word gap identification in video from the restored image. Novel distance vectors are proposed for character segmentation from words. The experimental results of the recognition reveal that segmentation of words and characters is useful to improve the accuracy of OCR recognition. In future, to achieve better accuracy as in document analysis, we are planning to develop a reconstruction algorithm to restore character shapes for segmented characters from text line images.

ACKNOWLEDGMENT

This joint work is done by National University of Singapore, Singapore and Indian Statistical Institute, Kolkata, India and it is supported in part by A*STAR grant R252-000-402-305.

REFERENCES

[1] S. H. Lee and J. H. Kim, “Complementary combination of holistic and component analysis for recognition of low resolution video character images”, Patten Recognition Letters, 2008, pp 383-391.

[2] D. Chen and J. M. Odobez, “Video text recognition using sequential Monte Carlo and error voting methods”, Pattern Recognition Letters, 2005, pp 1386-1403.

[3] D. Chen, J. M. Odobez and H. Bourland, “Text detection and Recognition in images and video frames”, Pattern Recognition, 2004, pp 595-608.

[4] X. Tang, X. Gao, J. Liu and H. Zhang, “A Spatial-Temporal Approach for Video Caption Detection and Recognition”, IEEE Transactions on Neural Networks, 2002, pp 961-971.

[5] D. Doermann, J. Liang and H. Li, “Progress in Camera-Based Document Image Analysis”, In Proc. ICDAR 2003, pp 606-616.

[6] C. Wolf and J. M Jolion, “Extraction and Recognition of artificial text in multimedia documents”, Pattern Analysis and Applications, 2003, pp 309-326

[7] X. Hunag, H. Ma and H. Zhang, “A New Video Text Extraction Approach”, In Proc. ICME 2009, pp 650-653.

[8] J. Zang and R. Kasturi, “Extraction of Text Objects in Video Documents: Recent Progress”, In Proc. DAS 2008, pp 5-17.

[9] K. Jung, K.I. Kim and A.K. Jain, “Text information extraction in images and video: a survey”, Pattern Recognition, 2004, pp. 977-997.

[10] A.K. Jain and B. Yu, “Automatic Text Location in Images and Video Frames”, Pattern Recognition, 1998, pp. 2055-2076.

[11] H. Li, D. Doermann and O. Kia, “Automatic Text Detection and Tracking in Digital Video”, IEEE Transactions on Image Processing, 2000, pp 147-156.

[12] K. L. Kim, K. Jung and J. H. Kim, “Texture-Based Approach for Text Detection in Images Using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm”, IEEE Transactions on PAMI, 2003, pp 1631-1639.

[13] P. Shivakumara, T, Q. Phan and C. L. Tan, “A Laplacian Approach to Multi-Oriented Text Detection in Video”, IEEE Transactions on PAMI, 2011, pp 412-419.

[14] H. Li, D. Doermann and O. Kia, “Text Extraction, Enhancement and OCR in Digital Video”, LNCS, 1999, pp 363-377.

[15] Z. Saidane and C. Garcia, “Robust Binarization for Video Text Recognition”, In Proc. ICDAR 2007, pp 874-879.

[16] G. Guo, J. Jin, X. Ping and T. Zhang, “Automatic Video Text Localization and Recognition”, In Proc. ICIG 2007, pp 484-489.

[17] Z. Zhou, L. Li and C. L. Tan, “Edge based Binarization for Video Text Images”, In Proc. ICPR 2010, pp 133-136.

[18] E. K. Wong and M. Chen, “A new robust algorithm for video text extraction”, Pattern Recognition, 2003, pp. 1397-1406.

[19] Donald Hearn and M. Pauline Baker, “Computer Graphics C Version” 2nd Edition, Prentice-Hall 1994, Bresenham Line Drawing Algorithm.

[20] S. Bolan, L. Shijian and Chew Lim Tan. “Binarization of Historical Document Images Using the Local Maximum and Minimum”. In Proc. DAS 2010, pp 159-165.

[21] OCR Engine used: http://code.google.com/p/tesseract-ocr/