Text Detection and Recognition in Real World Images.

Raid Saabni

Faculty of Engineering,
Tel-Aviv University.
Triangle R&D Center, Kafr
Qarea, Israel
saabni@cs.bgu.ac.il

Moti Zwilling

School of Business
Administration,Academic
Center of Law Business,
Ramat Gan, Israel
Moti.Zviling@gmail.com

Abstract

Detecting and recognizing texts in real world images such as sign boards and advertisements is an important part of computer vision applications. The complexity of the problem comes out of many factors such as non-uniform background, different languages and fonts, and non consistent text alignment and orientation. In this paper, we present a novel approach to detect characters and words in real-world images. The presented approach decompose the gray level image into sequence of images, each one includes pixels with gray level values from different disjoint ranges. This decomposition enables extracting connected components representing characters or other non textual objects separated from their neighborhood background. An interpolation of two classes of features translated to histograms is used by a support vector machine to classify and collect the textual objects generating the textual zones. The Shape Context Descriptor [1], is used by the Earth Movers Distance(EMD) method to recognize the characters within the image. The recognized characters are fed to heuristic rule based system to determine words and give final results. To optimize the speed of the system, we follow the embedding of the EMD metric presented in [22] to a normed space to enable fast approximation of the k-Nearest Neighbors using Local Sensitivity Hashing functions(LSH). Experiments show that our algorithm can detect and recognize text regions from the ICDAR 2005 datasets [17] with high rates.

Keywords: Word Searching; Earth Movers Distance; Embedding; Local Sensitivity Hashing, k-Nearest Neighbor; Text Detection;

1 Introduction

Text detection and recognition in images of real-world scenes has received significant attention recently. Reading words in unconstrained images is a challenging problem of considerable practical interest. In contrast to text recognition in documents (OCR systems), texts acquired in general settings, still considered as an open problem. Recently, there has been a surge of interest in the scene text recognition problem from the computer vision community, therefore, The public ICDAR Robust Reading challenge [17], was collected to highlight the problem of detecting and recognizing scene text. Some authors have focused on subtasks of the scene text recognition problem, such as text localization [5, 4, 19, 20], individual character recognition [25] or reading text from segmented areas of images [24]. In this paper we address the problem of word detection and recognition from out door images mostly taken by mobile phone cameras, giving a set of bounding boxes labeled with the recognized words as an output. Most text recognition systems in natural scenes accepts a gray level image as an input and convert it to a binary image using binarization or edge detection methods[4]. Generally binarization and edge detection for natural scene is different and more complicated than in OCR cases, therefore, binarization results are substantial for sufficient text detection and recognition rates.

The presented system, uses a novel approach of dividing the gray level image to sequence of images including consequent sub ranges of the gray level values. Mostly, texts in a natural scenes have close to uniform texture which is translated to a very close gray level values which are different than pixels of their background neighborhood. Following this observation, we use different ranges of values to collect pixels in each image(range) in the sequence to connected components separated form their immediate background. Connected Components (CC's) from results of each level (image) in the sequence are classified to textual and non-textual zones using a support vector machine. The system, accumulates textual zones from all levels to one image, where character objects in that image are recognized in the final process using the Earth Movers Distance.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.279

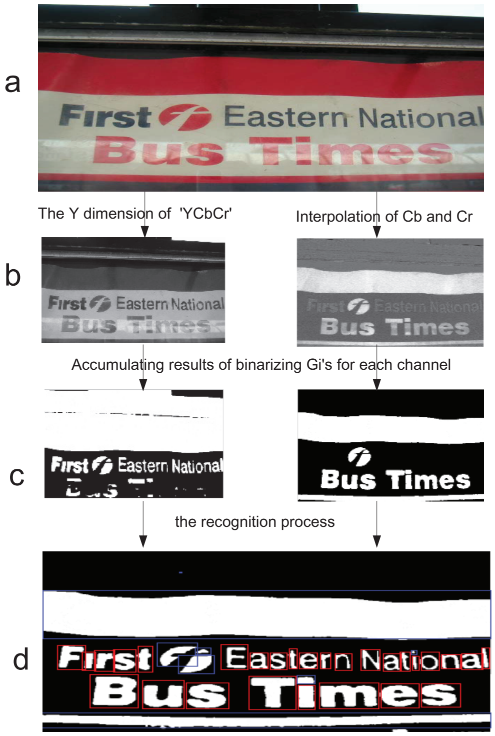
a
The Y dimension of 'YCbCr'
Interpolation of Cb and Cr
b
Accumulating results of binarizing Gi's for each channel
c
the recognition process
d
First Eastern National
Bus Times

Figure 1. (a) The original image. (b) Results of the 'YCbCr' color space divided to two channels. (c) The accumulated results of binarizing each range G_i of gray values.(d) Classification results: text zones are in red bounding boxes and non-text zones in blue.

2 Related Work

Generally speaking, algorithms for text detection and reading from real world images can be broadly categorized into three categories. In the first category, ('texture-based'), algorithms follow the assumption that textual areas have unique textures [3, 9]. These algorithms, scan the image at a number of scales, classifying neighborhoods of pixels based on text properties, such as high density of edges, high variance of intensity, distribution of wavelet or DCT coefficients, etc. The second category is the 'component-based' approach [16, 14], where connected components extracted from the binarized images are recognized after being collected and grouped based on certain properties, such as approximately constant color. This approach is attractive because it can simultaneously detect texts at any scale and is not limited to horizontal texts. The hybrid approach is the third category where the texture based approach is used to extract textual zones and the second approach is used to collect and recognize connected components in these textual zones. A good overview of text detection algorithms can be found in ICDAR 2005 competition report [17]. Approaches also can be divided according to whether or not they use machine learning techniques. A representative example for learning was presented in [3] where Ada-Boost is used for learning of joint-probabilities of features. On the other hand, Epshtein *et al.* [4] did not exploit learning techniques, focusing instead on the fact that text has a constant stroke width.

Many systems use word matching algorithms for spotting and searching words within large sets of shapes [18, 7, 21]. Results were mostly encouraging in the accuracy factor, but less satisfying when considering time efficiency. As a result, a lot of work has been done on embedding finite metric spaces into low-dimensional normed spaces in order to enable efficient and fast nearest neighbor extraction. Such embeddings have been extensively studied in pure mathematics [2, 15, 26], and have found application in a variety of settings [6, 12], usually using one of the l_p norms. In domains with a computationally expensive distance measure, significant speed-ups can be obtained by embedding objects into another space with a more efficient distance measure. Methods among others that can be used for efficient retrieval include Lipschitz embeddings, FastMap [6] and MetricMap [23]. Efficient embedding of the EMD metric to a normed space have been presented by Indyk and Thaper[13] and used later by Grauman and Darrell [10] for contour matching. Indyk and Thaper[13] use a randomized multi-scale embedding of histograms into a space equipped with the l_1 norm. Additional efficient embedding of the EMD metric have been presented by Shridonkar and Jacobs[22], see section 3.4 for more details.

3 Our Approach

In the presented work, we address the problem of detecting and recognizing printed texts in different fonts from out door images. The segmented and recognized characters are used to generate words using a probabilistic rule based system to be matched to a predefined list of words. In the presented work, We use a novel methodology to avoid incomplete data mostly occurs as a result of binarization or edge detection methods frequently used as preprocessing steps. For the recognition process, we use the EMD metric to measure similarity between two shapes using two different features extracted from the boundary. The training sets for classification and recognition were extracted from the ICDAR2003 data set using the associated ground truth. To classify objects to characters and non-characters we have used an interpolation of two feature descriptors with the SVM classifier. For the recognition of each textual object to one of the 62 characters and 10 digits we have used the Shape Context feature set with

Earth Movers Distance. Considering the very large number of positive and negative samples extracted from the data set, we use the mapping presented in [22] for embedding different shapes into a normed space. Embedding is performed using a linear time algorithm for approximating the EMD for low dimensional histograms using the sum of absolute values of the weighted wavelet coefficients of the difference histogram. This embedding enables the use of fast approximation of the k-nearest neighbors methods such as k-d trees and LSH. Generating a short list enables applying the original exact and expensive matching methods, yet keeps sub linear searching time.

Systems for text detection and recognition, usually accept gray level images as an input and use variations of known methods for binarization or edge detection to generate a binary image for the next steps. The efficiency and accuracy of binarization and edge detection methods for images from the real world is crucial, but mostly insufficient as the case of OCR systems for simple text pages. Non uniform background and Illuminance, images of non textual objects and large ranges of colors for different texts is a partial list of factors affecting these results. From the other hand, readable texts in natural scenes, mostly have homogeneous textures and colors which can be distinguished and separated form the neighboring background. Following this observation, in the preprocessing step we divide the gray level Image G, to a sequence of Images G_i, where each G_i contains only the pixels with a specific range of gray values. The fact that a given character have a strict range of gray level values which are different than it's neighboring background, assures that when picking the right range. In many cases pixels of a single character, fall in one of theses G_i's without the neighboring background pixels. To increase homogeneity of the values inside characters we blur the image using average filter before generating the G_i's.

Formally, let G be an input gray level image, and let G_{Mn} and G_{Mx} be the minimum and the maximum value of G respectively. Given Stp, the width of the range of values of each G_i, we divide G to n images with $n = \frac{G_{Mx} - G_{Mn}}{Stp}$, using the following formula:

$$G_i(i,j) = \left\{ \begin{array}{lll} if & G(i,j) < Stp(i+\alpha)_{\&\&} & \\ & G(i,j) > Stp(i-\alpha) & 1 \\ & & \\ else & & 0 \end{array} \right\}$$

In this definition, neighboring pixels that fall in the same range of gray values tend to connect to each other performing one connected component. The value of Stp determines the range width including pixels of characters, but not neighborhood pixels. In order not to lose components where the values fall between two subsequent ranges, ranges are generated with an overlapping area where α defines the size of the overlap (in our case α was 0.75). Following the mentioned observation, characters tend to be extracted in such cases as one connected component. Unfortunately, non character or parts of non character objects, tend to do the same, but unlikely will generate connected components that are similar to characters, therefore a classifier is needed to prune results and distinguish between characters and non ones. In the presented approach, the inverse image is not needed such as in other approaches since the algorithm goes through all gray level values in the range. Some morphological operations such as blurring and closing are used to make colors within characters more homogeneous and close artifact holes. Converting the color image to a gray level image is done using the $YCbCr$ methodology, where the first and interpolation of the last two dimensions are used separately, to overcome problems of non uniform Illuminance and concentrate on the important data.

3.1 Classifying Components

The components extracted in each gray level range many times are not complete. Especially non character objects, which tend to have large ranges of gray level values. We have used the same algorithm which extracts object and partial object(not complete) to generate the two classes of positive and negative samples, using subset of the ICDAR2005 data sets [17], with the appropriate ground truth. We have extracted all objects in these sets and classified them to characters and non characters based on the ground truth of these sets. For the character objects we have refined the shapes in that set to include complete and almost complete characters. For non characters, we have used euclidean distance with the same feature space as in the classifying process, to drop redundant very similar shapes using k-means clustering.

The next step, aims to classify objects to characters and non characters, therefore a simple interpolation of two strong features have been used with the l_2 metric. To classify CC's as characters, two feature classes are extracted, converted to the same size histograms, normalized and merged to one feature vector to be fed to a Support Vector Machine classifier. This feature vector is generated using the Shape Context [1] and the angles of the simplified contour segments. Results are motivated to tolerate non characters similar to characters as false positives which will be filtered out in the recognition phase. The described process can be adapted to other language such as Arabic, by replacing characters with word-parts.

3.2 Feature Extraction

Two feature descriptors were used for the classification and recognition steps. In the first step, we classify objects to textual and non textual using a concatenation of the 'Shape contexts' descriptor and the histogram of the angles of segments of the simplified outer contour. The Angles of the simplified contour segments is a local feature descriptor the authors already used in [21] with good results. We convert each descriptor to a histogram with 18 bins each, normalize and concatenate them to one feature vector for a shape. This feature vector is less detailed than the Shape Context or the angle of segments features, but strong enough to distinguish between textual and non textual objects, Whereas, tolerating overlapping similarity between the two classes.

For the recognition process a stronger and more detailed descriptor is needed to classify shapes of objects to the right character, therefore, we use the original Shape Context descriptor. The Shape Contexts descriptor presented by Belongie and Malik *et al.* [1] is a boundary based descriptor which describes a distribution of all boundary points with respect to each point on the boundary. The Shape Context descriptor of each single boundary point computes the histogram of relative polar coordinates and have been proved to be one of the efficient features for matching binary images.

3.3 Objects Recognition

The presented system uses the Earth Movers Distance (EMD), to measure distance of a given Connected Component (CC), to many shapes of each English character. The Approximate EMD measurement using the Shape Context feature descriptor is used to rank the top k-nearest neighbors of the object (CC) from all shapes of the 62 different classes of English characters and 10 additional digits. All shapes in that feature space, are mapped into a normed space by performing an embedding process for approximating the EMD for low dimensional histograms using the sum of absolute values of the weighted wavelet coefficients of the difference histogram. Using Local Sensitivity Hashing (LSH), we find the approximate k-nearest neighbors to generate a short list on which we apply the original expensive EMD algorithm to fix final results.

The EMD, is a method to evaluate dissimilarity between two multi-dimensional distributions in some feature space where a ground distance between single features is given. Intuitively, given two distributions, one can be seen as a collection of piles of sand and the other as a collection of holes. The EMD measures the least amount of work needed to fill the holes with the sand. Here, a unit of work corresponds to transporting a unit of sand from one pile to a hole depending on their distance. Computing the EMD is based on a solution to the well-known transportation problem [11].

Formally, let us define the first signature to be $\{(q_i, w_i)\}_{i=1}^{m}$ with m entries and the second as $\{(p_i, w_i)\}_{i=1}^{n}$ with n entries. Let the flow between p_i to q_j be f_{ij} and d_{ij} be the ground distance between the entries p_i and q_j. We can solve this problem using the following linear programming problem: Find the flow $F = [f_{ij}]$ that minimizes the following work for the signatures P and Q:

$$Work(P, Q, F) = \sum_{j=1}^{n} \sum_{i=1}^{m} f_{ij} d_{ij}$$

Subject to the following constraints: 1)Flow is only allowed from P to Q and not vice versa, 2) the amount that clusters of P can send are no more than their weights, and the clusters in Q to receive no more than their weights and 3) the last constraint forces to move the maximum amount (total flow) of P that is possible. Once the transportation problem is solved, and we have found the optimal flow, the earth mover's distance is defined as the work normalized by the total flow. The normalization factor is the total weight of the smaller signature, in order to avoid favoring smaller signatures.

3.4 Pre-Processing for recognition Speedup

Shridonkar and Jacobs[22], presented an efficient embedding of the EMD metric for approximating the EMD distance of two histograms using a new metric on the weighted wavelet coefficients of the difference histogram using the l_1 distance. Experimentally they show that this metric follows EMD closely and can be used instead without any significant performance difference. Intuitively speaking, the wavelet transform splits up the difference histogram according to scale and location. Each wavelet coefficient represents an EMD subproblem that is solved separately. For a single wavelet, the mass to be moved is proportional to the volume of $|\psi_j(x)|$, i.e. to $2^{jn/2}$. The distance traveled is proportional to the span of the wavelet 2^{-j} (according to Meyers [14] convention, a wavelet at scale j is the mother wavelet squeezed 2^j times.) The sum of all distances is an approximation to EMD and called the wavelet EMD between two histograms, see equation 1.

$$d(p)_{wemd} = \sum_{\lambda} 2^{-j(1+n)/2} |p\lambda| \tag{1}$$

where, p is the n dimensional difference histogram and p_λ are its wavelet coefficients. The index λ includes shifts and the scale j.

Following the embedding by Shridonkar and Jacobs[22], let DS be the data set of all different shapes of the 62 English characters and the 10 digits, and SS be the set of all available shapes of different appearances of all object in DS. Embedding SS into a normed space, will enable fast approximate search with sub-linear time of the size of SS and improve the efficiency. The process starts by converting all shapes to the feature space using the Shape Context descriptor and normalize them to the same size following the constraints of the embedding process. First, the feature vectors are converted into the wavelet domain using $Coiflets$ of order 3 resulting a 850-coordinate vector of coefficients for each shape. A serious of Locality Sensitive Hashing function are generated on the embedded space, to enable fast search of query images using l_1 to approximately estimate the EMD distance.

Locality Sensitive Hashing (LSH) is a technique for grouping points in space into 'buckets' based on some distance metric operating on the points. Points that are close to each other under the chosen metric are mapped to the same bucket with high probability. This is based on the simple idea that, if two points are close together, then after a projection operation these two points will remain close together. The basic idea is to hash the input items so that similar items are mapped to the same buckets with high probability (the number of buckets being much smaller than the universe of possible input items). LSH [8], uses several hash functions of the same type to create a hash value for each point of the dataset. Each function reduces the dimensionality of the data by projection onto random vectors. The data is then partitioned into bins by a uniform grid. Since the number of bins is still too high, a second hashing step is performed to obtain a smaller hash value. At query time, the query point is mapped using the hash functions and all the data points that are in the same bin as the query point are returned as candidates. The final nearest neighbors are selected by a linear search through candidate data points.

4 Experimental results

To evaluate our algorithm, we employ the publicly accessible benchmark of natural scenes containing text used in the ICDAR2005 competition [17]. The fact that we have extracted the training shapes of characters from real world images data set (a subset of the ICDAR2005 data set), contributes to the efficiency of the system. Since the system addresses printed fonts and deals with individual characters, we have excluded cursive texts out of the testing set. The results in Table 1, present rates in word level with no use of any additional lexicon. The usual rules used in many other systems for collecting characters to words considering lines, heights, locations and other factors were used to generate words.

Precision and Recall of the different configurations of the two systems in terms of counting false positives and false negatives among the automatically extracted components. To be compatible with the results from the ICDAR2005 competition, we have used the same configuration of a 2.4ghz PC running with Windows OS. As we can see in Table 1, results in terms of precision and recall are slightly better than the compared methods. To improve times efficiency, the approximation approach is used and speeds up the system 100 times, while results slightly decrease in only 1%.

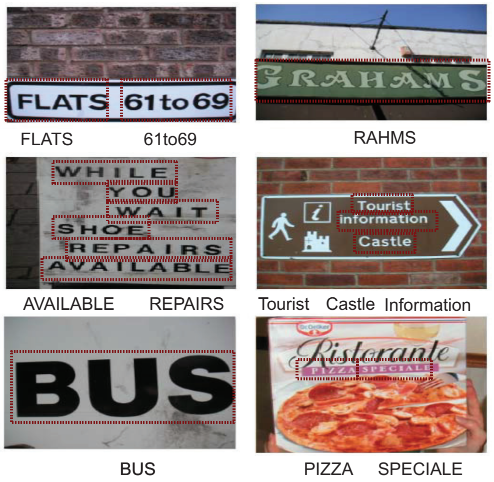

Figure 2. Partial results of texts extracted and labeled. Notice that some characters were missed and cursive words were excluded

Table 1. The entries in the table are the name of the system as appeared in the report of the ICDAR2005 competition, followed by precision and recall. The columns labeled t(s) gives the average time in seconds to process each image. Times of our system are longer due to the time needed for recognizing the texts additionally to text location.

System	Precision	Recall	t(s)
Hinnerk Becker	62%	67%	14.7
Alex Chen	60%	60%	35
Ashida	55%	46%	8.7
HWDavid	44%	46%	0.3
Full_EMD	64%	63%	413.8
Aprox_EMD	63%	61%	31.7

5 Acknowledgments

We would like to thank Prof' D. Ron from Tel-Aviv University for her insightful comments which led to several improvements in the presentation of this paper. This work has been supported in part by the DFG-Trilateral Grant No.8716.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelli-*

gence, 24:509–522, 2002.

[2] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel J. Math*, 25:46–52, 1985.

[3] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004*, pages 366–373, 2004.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR '10: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition.*, 2010.

[5] N. Ezaki. Text detection from natural scene images: towards a system for visually impaired persons. In *In Int. Conf. on Pattern Recognition*, pages 683–686, 2004.

[6] C. Faloutsos and K. Lin. fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings ACM SIGMOD Conference*, pages 163–174, 1995.

[7] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis, and S. Perantonis. A segmentation-free approach for keyword search in historical typewritten documents. In *Proceedings of ICDAR 2005*, volume 1, pages 54–58, 29 Aug.-1 Sept. 2005.

[8] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *In Proceedings of the 25th Intl Conf on Very Large Data Bases*, 1999.

[9] J. Gllavata, R. Ewerth, and B. Freisleben. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *(ICPR'04*, volume 1, pages 425–428, 2004.

[10] K. Grauman and T. Darrell. Fast contour matching using approximate earth movers distance. In *In IEEE Conference on CVPR*, volume 01, page 220227, 2004.

[11] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Math and Phys*, 20:224–230, 1941.

[12] G. Hristescu and M. Farach-Colton. Cluster-preserving embedding of proteins. Technical report, Rutgers Univ., Piscataway, New Jersey, 1999.

[13] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *In 3rd International Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[14] A. Jain and B. Yu. Automatic text location in images and video frames. *Pattern Recognition*, 31(12):2055–2076, 1998.

[15] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Math*, 26:189–206, 1984.

[16] Y. Liu, S. Goto, and T. Ikenaga. A contour-based robust algorithm for text detection in color images. *IEICE TRANS. INF. & SYST*, E89D, NO.3, 2006.

[17] S.M. Lucas. Icdar 2005 text locating competition results. In *Proc. of the 8th Int. Conf. on Document Analysis and Recognition (ICDAR 2005)*, volume 1, pages 80–84, 2005.

[18] R. Manmatha and T. Rath. Indexing handwritten historical documents - recent progress. *the Proc. of the Symposium on Document Image Understanding (SDIUT-03)*, pages 77–85, 2003.

[19] Y. F. Pan, X. Hou, and C.L. Liu. A robust system to detect and localize texts in natural scene images. In *APR International Workshop on Document Analysis Systems.*, pages 35–42, 2008.

[20] Y.F. Pan, Liu Hou, X., and C.L. Text localization in natural scene images based on conditional random field. In *Proc. of the ICDAR2009*, pages 6–10, 2009.

[21] R. Saabni and J. El-Sana. Keyword searching for arabic handwritten documents. In *Proc of the (ICFHR2008), Montreal*, pages 716–722, 2008.

[22] S. Shirdhonkar and D.W. Jacobs. Approximate earth movers distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference.*, pages 1–8, 2008.

[23] X. Wang, J. T. L. Wang, K. I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang. An index structure for data mining and clustering. *Knowledge and Information Systems*, vol. 2:161–184, 2000.

[24] J.J. Weinman, E. Learned-Miller, and A.R. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:1733–1746, 2009.

[25] M. Yokobayashi and T. Wakahara. Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. In *Proc. of the 8th International Conference on Document Analysis and Recognition.*, pages 167–171, 2005.

[26] N.J. Young. An introduction to hilbert space. *Cambridge, UK: Cambridge Univ. Press*, 1988.