2011 International Conference on Document Analysis and Recognition

Document Image Indexing using Edit Distance based Hashing

Ehtesham Hassan, Santanu Chaudhury, M Gopal
Department of Electrical Engineering
Indian Institute of Technology Delhi, New Delhi
hassan.ehtesham@gmail.com, santanuc@ee.iitd.ac.in, mgopal@ee.iitd.ac.in

***Abstract*—We present a novel word image based document indexing scheme by combination of string matching and hashing. The word image representation is defined by string codes obtained by unsupervised learning over graphical primitives. The indexing framework is defined by distance based hashing function which does the object projection to hash space by preserving their distances. We have used edit distance based string matching for defining the hashing function and for approximate nearest neighbor based retrieval. The application of the proposed indexing framework is presented for two document image collections belonging to Devanagari and Bengali script.**

***Keywords*-Document image indexing, Distance based hashing, Edit distance, Shape descriptor**

I. Introduction

Image based document indexing provides an effective solution for storage and retrieval of document collections belonging to scripts having underdeveloped OCR technology. The objective of the present work is to define a novel word based document indexing scheme using edit distance based hashing. A word based indexing scheme fundamentally converts the word images segmented from documents to arithmetic features, and groups the similar words in index space based on defined similarity measure. The framework assigns unique indices to each word image group. The retrieval is done by projecting the query image to a word group and return documents corresponding to words in the group. Two primary challenges involved here are defining unique feature representation for word images and efficient grouping framework for similar word images.

In this direction, the paper presents a novel string based word image representation. The conventional feature based representations capture the global characteristics of the word shape [1][2][3]. In this case, the word spotting based retrieval schemes does not include the substrings based matches existing in the documents. Our work in this direction presents novel clustering based approach to define the word representation. We identify the graphical primitives in the word formulation and apply adaptive clustering for grouping these primitives. A graphical primitive represents the structural part of word image obtained after segmentation. The clustering identifies the equivalence groups of graphical primitives exiting in the document collection. Each group of graphical primitives is assigned a unique code. The word image representation is defined by identifying the graphical primitives and assigning the code based on its nearness to a group of primitives. In this sense, our approach presents a script invariant methodology for word image representation. The nearest neighbor based retrieval provide robust retrieval method. However the linear time search complexity is practically unacceptable for searching large datasets. The hashing based approximate nearest search performs retrieval in sub-linear time complexity with trade-off in accuracy. Our work presents a novel application of edit distance based hashing for document indexing. The effectiveness of the proposed framework has been demonstrated on document collections belonging to Devanagari and Bengali script.

The paper organization is as following: Section II presents related works. Proposed indexing and retrieval framework is presented in section III. Section IV presents discussion on Edit distance based hashing. Experimental evaluation of the proposed framework is presented in section V. The section VI presents conclusion and perspective of our work.

II. Related Works

Large amount of research has been done in the area of word image representation, and document indexing [4][5][6][7]. We present a brief and concise review of the related works in this direction. The string based word representation is generated by extracting character objects from the word image. Objects are assigned codes based on their similarity to example character objects. The concatenation of object codes in respective order defines the word string. In recent works, Shijian et al. [4] proposed a word shape coding without requiring character segmentation. The word image is represented by set of topological character shape features including ascenders/descenders, character holes and reservoirs. Simone et al. [5] have defined collection specific character prototypes from the character objects. The set of character prototypes are further used for word image representation. Nakayama [6] defined word shape tokens for printed words by sequence of character shape codes defined by the set of graphical features. In [7], word shape code is defined using conventional features as ascenders, descenders, character holes, deep eastward and westward concavity and horizontal-line intersections. The existing works on string based word representation is script dependent. Additionally,

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.242

character object segmentation in case of Indian scripts is difficult task. We tackle this problem by defining word representation based on graphical primitives. We apply clustering to identify graphical primitives peculiar to a particular writing style/script for defining the word representation.

The hashing based indexing for various applications is a popular research problem. Hashing based schemes generate the object indices by projecting it to lower dimensional hash space. The process generates hash table containing multiple buckets. Each bucket represent group of objects corresponding to an unique index. The retrieval process includes query index generation by applying the same mapping function to query. The relevant documents are obtained by performing similarity search over the objects in bucket corresponding to query index. The work presented in [8] demonstrated one of the earliest applications of Geometric hashing for handwritten document indexing. Locality Sensitive Hashing (LSH) introduced by Indyk and Motwani is state-of-the-art method for finding similar objects in large data collection [9]. The LSH solves approximate nearest neighbor search in $O(n^{1/1+\epsilon})$ for $\epsilon \geq 0$, by projecting the high dimensional data points to low dimensional hash space with high probability that similar points are mapped to same location. In recent years, many applications have applied LSH for performing similarity search in high dimensional spaces [10], [11], [12], [13]. In this case, the retrieval success probability is increased by generating multiple hash tables and collecting objects corresponding to query buckets of all hash tables for similarity search. However, above discussed methods are not applicable for indexing in arbitrary spaces. In this direction, the paper presents novel application of edit distance based hashing for indexing.

III. Proposed Document Indexing Framework

In this section we present an overview of the proposed document image retrieval system. The word image representation and document indexing scheme are presented as follows.

A. Word Image Representation

We follow the string codes based representation for word images. Extraction of character primitives in Indian script words is a complicated task. The problem of character segmentation from word images has been extensively studied by researchers categorically in the domain of OCR development. However, the process is costly and depends highly on the quality of documents. Therefore to increase the robustness of the string coding, we define word representation based on graphical primitives instead of character primitives. The segmentation of graphical primitives is relatively much simpler and it may have a single letter, or combined letters defined by various grammatical rules. The segmentation of character primitives from word images are performed by identifying the local minima over the vertical profile of the word image. The cut-off points in figure 1 are the local

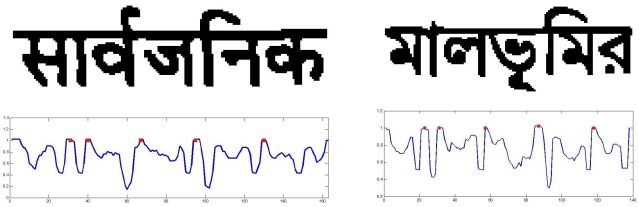

Figure 1. Vertical profile and cut-off points over for graphical primitive segmentation

minima points. Each graphical primitive is defined by the region between cut-off points including the end points. The process generates a large set of graphical primitives. The figure 2 shows set of the graphical primitives obtained after segmentation. The next step for word representation is code

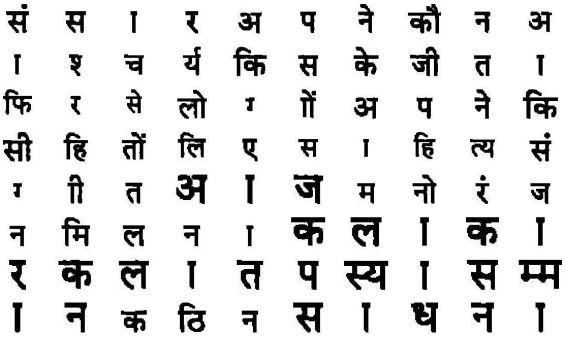

Figure 2. Sample graphical primitives from Devanagari document collection

assignment for each graphical primitive. In the context of Indian scripts, the exact estimation of unique graphical primitives is a difficult task. The problem is further compounded by noisy character primitives segmented wrongly because of the degradation in document images. To increase the robustness of the character code, we follow clustering based approach for learning the codes. A sample set of words is chosen to generate the character primitive dictionary. The words are selected such that segmented primitives cover maximum range in terms of variety. Since the prior knowledge about the dictionary size is not available, we apply DBSCAN for clustering [14]. Each character primitive is assigned to a cluster with respect to its nearness to the cluster. The nearness is computed by Euclidean distance between cluster primitive and cluster centers.

B. Document Indexing

Our document indexing scheme follows conventional hashing based indexing framework (Figure 3). Here we apply distance based hashing functions for generating the indices. The initial steps of off-line processing include document preprocessing (deskewing and binarization). The

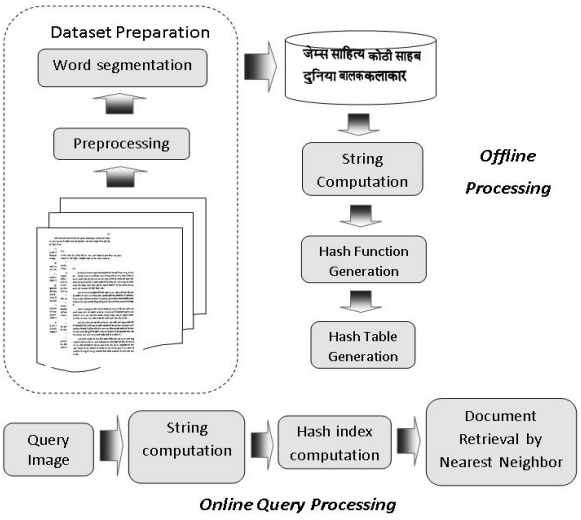

Figure 3. Document Indexing Framework

segmentation routine extracts word images from the document image collection. The meaningful word images for generating the document indices are selected using word length in terms of pixels as thresholding criterion, e.g. most of the Devanagari and Bengali words having less than four characters are not required for indexing. Additionally, the stop words, punctuation and typographical marks are filtered by applying conventional thresholds (aspect-ratio, word length). We generate multiple (L) hash tables by applying the hash function (g) to word images which are defined as numeric string. Each hash table corresponds to a hashing function $g_i = \{f_1, f_2, ..., f_k\}$, here $f \in \mathbb{H}$. $\mathbb{H}$ represent the family of Distance based hash functions. The selection of f_k from $\mathbb{H}$ is based on random selection. The mapping of word images by g_i is based on the assumption that two similar data points which are close in high dimensional space, with high probability will be mapped to same or nearby location in the hash space. Therefore making the indexing able to discover the local geographical structure of the object space. Each hash table contains group of k-bit vectors representing groups of word objects generated by the g_i. In the case of perfect hashing (ideal case), each group will have identical word objects. The complete document image indexing and retrieval framework can be separated as: Off-line Processing and On-line Processing (Figure 3). The Off-line processing includes word image segmentation from the document image collection and index generation for the word images.

The word images are represented as string of codes learned over set of graphical primitives in the word images. The off-line process include generation of distance based hash functions. Each hashing function is defined for a pair of data points defined as pivots. The pivot objects are selected by the random sampling from the word collection. These pivot objects are used to generate a family of hashing functions $\mathbb{H}$. The hashing function generation complexity is $O(\beta^2 N^2)$. Here N is the word image count, and β is the ratio of cluster count to N. For L hash tables we generate g_i for $i = 1, ..., L$ by random selection of k functions from $\mathbb{H}$. Using each function g_i, the indices for word images belonging to complete collection is generated and the procedure is repeated for all the hash tables. Each separable set obtained after the application of g_i over the set forms a separable bucket. In this respect a bucket remains a metric region and organizes all word images from the metric domain falling into it. Each hash table requires $O(kN)$ distance calculations (a distance calculation signifies one projection operation). The storage architecture of the hash table is based on 2-dimensional array of buckets used for storing data points. This requires $O(kLN)$ space to record the k-bit coordinate values corresponding to N word images for L hash tables.

The on-line process includes retrieval by performing the similarity search based on the query. It includes query projection over all the hash tables using string based representation. The data points from the query bucket (bucket to which query is hashed) in all the hash tables are collected. The word images similar to query word are retrieved by performing similarity search over the group of collected data points.

IV. Edit Distance based Hashing

The concept of Distance Based Hashing [15] is fundamentally based on FastMap embedding method [16]. The DBH is an algorithm to map objects to points in k-dimensional space such that the inter object distances are preserved. The primary objective of distance based hashing to perform indexing in arbitrary spaces using arbitrary distance measures. The hashing, therefore considers the distance function as notion of similarity without any theoretical interpretation. The DBH is defined using line projection formula which projects the object x represented as data-point in Euclidean space $\mathcal{X}$ over the line $\mathfrak{L}$ joining objects (x_1, x_2). For objects (x_1, x_2) in space $(\mathcal{X}, \mathcal{D})$, the line projection $F^{x_1,x_2} : \mathcal{X} \rightarrow \mathfrak{L}$ for object x is defined as

$$F^{x_1,x_2}(x) = \frac{\mathcal{D}(x_1,x)^2 - \mathcal{D}(x_2,x)^2 + \mathcal{D}(x_1,x_2)^2}{2\mathcal{D}(x_1,x_2)} \qquad (1)$$

The equation (1) shows that availability of distance $\mathcal{D}$ is the only requirement for projection. Therefore the mapping F is also applicable for arbitrary spaces. The string based representation has been extensively applied for various computer vision applications. In this present paper, we define Edit distance based hashing functions for indexing the word images. The edit distance between two strings is defined by insertion, deletion and substitution cost of characters between strings. Therefore the word image grouping in hash space should group similar words as well as word images having similar sub strings.

For each function F a pair of objects (x_1, x_2) are chosen, a line is drawn between them that serves as coordinate axis, and the coordinate value along this axis for each object is determined by equation (1). If $\mathcal{X}$ is a general non-Euclidean space, then $F^{x_1,x_2}(x)$ does not have a geometric interpretation. However as long as $\mathcal{D}$ is available for $\mathcal{X}$, F^{x_1,x_2} can be defined and provides a simple way to project x on the line defined by (x_1, x_2). The mapping F is independent of the dimensionality of object representation as the inter object distances is the only requirement. The equation (1) defines a rich family of functions. For a collection of N objects in $\mathcal{X}$, $N(N-1)/2$ unique functions can be defined by applying equation (1) to each pair of objects. It is always convenient to have a hashing function that maps objects to $\{0, 1\}$. The functions defined using equation (1) are real valued, whereas we desire discrete-valued hashing functions. The binary hashing functions can be obtained from F^{x_1,x_2} using thresholds $t_l, t_2 \in R$ as:

$$F^{x_1,x_2}_{t_1,t_2}(x) = \begin{cases} 1 & \text{if} \quad F^{x_1,x_2}(x) \in [t_1, t_2] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In practice, (t_1, t_2) should be such that $F^{x_1,x_2}_{t_1,t_2}(x)$ maps approximately half the data points in $\mathcal{X}$ to 0 and half to 1, i.e. F generates balanced hash tables. We formalize this notion by defining for each pair $(x_1, x_2) \in \mathcal{X}$, the set $V(x_1, x_2)$ of intervals $[t_1, t_2]$ such that $F^{x_1,x_2}_{t_1,t_2}(x)$ splits the space in half as

$$V(x_1, x_2) = [t_1, t_2] | \Pr_{x \in \mathcal{X}}(F^{x_1,x_2}_{t_1,t_2}(x) = 0) = 0.5 \tag{3}$$

The hash function family $\mathbb{H}_{\mathbb{DBH}}$ for an arbitrary space $(\mathcal{X}, \mathcal{D})$ is defined as

$$\mathbb{H}_{\mathbb{DBH}} = F^{x_1,x_2}_{t_1,t_2}(x) | x_1, x_2 \in \mathcal{X}, [t_1, t_2] \in V(x_1, x_2) \tag{4}$$

An indexing framework is defined by hash function g as concatenation of k functions selected randomly from $\mathbb{H}_{\mathbb{DBH}}$ i.e. $g = [h_1, ..., h_k]$. The hash values generated by g for an object defines its bucket indices in the hash table. With increase in k, the collision probability $\Pr\{g(x) = g(y)\}$ = $(\Pr\{h(x) = h(y))^k$ decreases exponentially. Therefore L independent hash tables are generated by defining L independent hash functions. The retrieval is performed by computing query hash value $g_i(q)$ for i = 1, ..., L. The value $g_i(q)$ represents the query bucket indices in i^{th} hash table. The similar items are retrieved by performing similarity search over the collection of objects retrieved from buckets $g_i(q)$ in L hash tables. The hashing parameters, function length k and number of tables L are performance parameters. In general, large L will increase the recall with reduction in precision and increase in retrieval cost. Larger function length k increases precision with reduction in recall, therefore requires sufficient number of hash tables for satisfactory recall level.

V. Experimental Results

In this section, experimental results of the proposed document image indexing and retrieval framework is presented. The experiments have been performed on document image collection belonging to Devanagari and Bengali scripts. The Devanagari document collection contains 327 pages scanned from 4 books. The Bengali document collection contains 178 pages scanned from 3 books. The document collection has been prepared as a part of consortium project funded by Government of India [17]. The book pages have been collected from old books and scanning is performed at 300dpi. The scanned images in the collection are of low quality, primarily because of degradation in original document pages. The preprocessing step for Devanagari and Bengali documents included smoothening and deskewing. The conversion of original gray scale images to binary images is performed by Otsu's method. The word segmentation from document image is done by horizontal and vertical profile based technique. After initial filtering, Devanagari word dataset contains 14888 words, Bengali word dataset consists 11231 words. The filtering process removes stop words, punctuation marks, and words having length less than threshold.

The word image representation is generated following the discussion in section III-A. After the initial segmentation, each graphical primitive is represented by shape descriptor presented in [18]. The shape descriptors for graphical primitives have been computed for two set of parameters (distance bins:m, angular bins:n). The document indexing framework for Devanagari collection is tested for 271 queries, for Bengali documents the framework is tested for 211 queries. Conventional performance measures i.e. Precision and Recall are computed over unordered result set. However, the ranking of retrieved results is an important retrieval measure. Therefore we computed Mean Average Precision (MAP) for measuring the performance of proposed indexing scheme. The MAP for a query set X_v is computed as the mean of average precision values [19].

$$\text{MAP}(X_v) = \frac{1}{|X_v|} \sum_{j=1}^{|X_v|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{5}$$

Average precision for query $q \in X_v$ is defined as mean of precision at each relevant recall point in the retrieved results. In equation (5), m_j represents the number of relevant retrieved results for query q_j and R_{jk} represents the ranked retrieval results from the top to k^{th} relevant result. If a retrieved document in R_{jk} is non-relevant, the precision value at this recall point is not considered. In the present indexing scheme, average precision for q is computed over the collection of neighbors obtained from L hash tables having indices $g_i(q)$ for $i = 1, ..., L$. The neighbors are ranked based on the Edit distance from the query string q. The Edit distance for word matching is computed as

Levenshtein distance between two strings. The MAP and average comparisons in the retrieval experiment for different hashing and descriptor parameters $\{(L,\ k)$ and $(m,\ n)\}$ is presented in the table I.

Table I
RETRIEVAL RESULTS

Results with Devanagari documents				
Descriptor paras.	m = 35, n = 36		m = 40, n = 40	
Hashing paras.	MAP	Comps.	MAP	Comps.
L = 18, k = 12	0.854	1413	0.861	1387
L = 18, k = 15	0.813	1028	0.816	984
Results with Bengali documents				
Descriptor paras.	m = 35, n = 36		m = 40, n = 40	
Hashing paras.	MAP	Comps.	MAP	Comps.
L = 18, k = 12	0.861	1127	0.864	1087
L = 18, k = 15	0.837	889	0.842	865

VI. CONCLUSION

We have presented a novel word image based document image indexing scheme. Our approach presents a novel script independent word image representation based on clustering which can used for defining indexing and retrieval framework for various class of document collections. We have showed novel application of edit distance based hashing for document indexing. The efficacy of the proposed framework have been shown experimentally on two printed document collection belonging to Indian scripts. The future direction of the work is to test the framework for other class of document collections.

ACKNOWLEDGMENT

This work is funded by MCIT, Government of India as a part of project *Development of Robust Document Analysis and Recognition System for Printed Indian Scripts*.

REFERENCES

[1] R. Bajaj and S. Chaudhury, "Signature verification using multiple neural classifiers," *Pattern Recognition*, vol. 30, no. 1, pp. 1–7, 1997.

[2] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, vol. 2, pp. 521–527, June 2003.

[3] F. R. Chen, L. D. Wilcox, and D. Bloomberg, "Word spotting in scanned images using hidden markov models," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 1–4, 1993.

[4] S. Lu, L. Li, and C. L. Tan, "Document image retrieval through word shape coding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1913–1918, 2008.

[5] S. Marinai, E. Marino, and G. Soda, "Font adaptive word indexing of modern printed documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1187–1199, August 2006.

[6] T. Nakayama, "Content-oriented categorization of document images," *Proceedings of the* 16^{th} *International Conference on Computational Linguistics*, vol. 2, pp. 818–823, 1996.

[7] S. Bai, L. Li, and C. L. Tan, "Keyword spotting in document images through word shape coding," *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition*, pp. 331–335, 2009.

[8] T. S. Mehmod, "Indexing of handwritten document images," *Proceedings of the 1997 Workshop on Document Image Analysis*, pp. 66–73, 1997.

[9] P. Indyk and R. Motwani, "Approximate nearest neighbor - towards removing the curse of dimensionality," *Proceedings of the* 30^{th} *ACM Symposium on Theory of Computing*, pp. 604–613, 1998.

[10] P. Haghani, S. Michel, and K. Aberer, "Distributed similarity search in high dimensions using locality sensitive hashing," *Proceedings of the* 12^{th} *International Conference on Extending Database Technology*, pp. 744–755, 2009.

[11] H. Shen, T. Li, and T. Schweiger, "An efficient similarity searching scheme in massive databases," *Proceedings of the* 3^{th} *International Conference on Digital Telecommunications*, pp. 47–52, 2008.

[12] W. Weihong and W. Song, "A scalable content-based image retrieval scheme using locality-sensitive hashing," *Proceedings of the International Conference on Computational Intelligence and Natural Computing*, vol. 1, pp. 151–154, 2009.

[13] B. Matei, Y. Shan, H. S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert, "Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1111 – 1126, 2006.

[14] M. Ester, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the* 2^{nd} *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[15] A. Vassilis, P. Michalis, P. Panagiotis, and K. George, "Nearest neighbor retrieval using distance based hashing," *Proceedings of the* 24^{th} *International Conference on Data Engineering*, pp. 327–336, April 2008.

[16] C. Faloutsos and K. I. Lin, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," *Proceedings of the ACM International Conference on Management of Data*, pp. 163 – 174, 1995.

[17] [Online]. Available: http://ocr.cdacnoida.in/

[18] E. Hassan, S. Chaudhury, M. Gopal, and J. Dholakia, "Use of mkl as symbol classifier for gujarati character recognition," *In the Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 255–262, 2010.

[19] C. D. Manning, P. Raghavan, and H. Schtze, *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2009.