Indexing Large Class Handwritten Character Database

D.S. Guru and V.N. Manjunath Aradhya

Abstract. This paper proposes a method of indexing handwritten characters of a large number of classes by the use of Kd-tree. The Ridgelets and Gabor features are used for the purpose of representation. A multi dimensional feature vectors are further projected to a lower dimensional feature space using PCA. The reduced dimensional feature vectors are used to index the character database by Kd-tree. In a large class OCR system, the aim is to identify a character from a large class of characters. Interest behind this work is to have a quick reference to only those potential characters which can have a best match for given unknown character to be recognized without requiring scanning of the entire database. The proposed method can be used as a supplementary tool to speed up the task of identification. The proposed method is tested on handwritten Kannada character database consisting of 2000 images of 200 classes. Experimental results show that the approach yields a good Correct Index Power (CIP) and also depicts the effectiveness of the indexing approach.

Keywords: Ridgelet Transform, Gabor Transform, Kd-tree, Handwritten Character Indexing.

1 Introduction

Optical Character Recognition (OCR) is one of the most fascinating and challenging areas of pattern recognition. Research on OCR is popular for its various potential applications in banks, postal departments, defense organizations etc. In such

D.S. Guru
Department of Studies in Computer Science, University of Mysore, Mysore
e-mail: dsg@compsci.uni-mysore.ac.in

V.N. Manjunath Aradhya
Department of Master of Computer Applications,
Sri Jayachamarajendra College of Engineering, Mysore
e-mail: aradhya.mysore@gmail.com

S.M. Thampi et al. (eds.), *Recent Advances in Intelligent Informatics*,
Advances in Intelligent Systems and Computing 235,
DOI: 10.1007/978-3-319-01778-5_23,

applications, response time and search efficiency also have become important in addition to the measure of accuracy because of a large population. In conventional matching process, recognition of a character from a huge collection of characters requires matching of the character against all characters present in the database, which is essentially a more time consuming process. In order to speed up this process, a filtering process is usually brought up to select a minimum number of potential candidates for further matching operation. To select the minimum number of candidate hypotheses, one can think of classification or indexing approaches.

Indexing scheme is to ensure that the system gets a few potential candidates selected in a quick manner for a given query character, so that later a rigorous matching can be carried out in order to identify the correct character. Indexing can be either hash based or tree based. Compared to hash based indexing, a tree based indexing has attained more attention. Hence in this work, we present a mechanism of indexing characters using tree based indexing. Some of the interesting survey made on indexing can be seen in [5, 6, 7, 8].

Feature extraction is one of the most important stage in OCR system. Most of the feature extraction techniques fall into two or combination of three major categories: (i) Statistical (ii) Structural (iii) Global transformation [12]. Due to inherent advantages of using combination of transform techniques and subspace models, in this work we explore the concept of Ridgelet and Gabor transform as a feature extraction method. The proposed method is experimented on offline handwritten characters of Kannada. It is worth to note that the proposed indexing model is first of its kind in the literature on a character database. The main highlights of our work are (i) A speed up approach is used to select a minimum number of potential candidates for matching operation. (ii) An indexing approach based on Kd-tree is used. (iii) Explored the concept of Ridgelets and Gabor for better representation purpose.

The organization of the paper is as follows: Section II presents the proposed model. Experimental results are presented in section III. Section IV concludes the paper.

2 Proposed Model

In this section, we describe the proposed indexing model. The proposed model extracts features by ridgelet and Gabor transforms. The feature vectors obtained by these transforms are projected onto reduced dimensional space by PCA. The reduced feature vectors are then accommodated in Kd-tree for indexing purpose.

2.1 Ridgelet Features

Candes and Donoho [1] introduced a new system of representations named Ridgelets, which they showed to deal effectively with line singularities in 2-D.

In order to apply ridgelets to digital images, Do and Vetterli [2] proposed a Finite Ridgelet Transform (FRIT). FRIT is based on the Finite Radon Transform (FRAT), which is defined as summation of image pixels over a certain set of lines. Those lines are defined in a finite geometry in a similar way as the lines for the continuous Radon transform in the Euclidean geometry. Z_p is denoted as $Z_p = 0, 1, ..., p-1$. Where p is a prime number and Z_p is finite field with modulo p operations.

The FRAT of real discrete function f on the finite grid Z_p^2 is defined as:

$$FRAT_f(k,l) = \frac{1}{\sqrt{p}} \sum_{(i,j)\in L_{k,l}} f(i,j). \tag{1}$$

Here $L_{k,l}$ denotes the set of points that make up a line on the lattice Z_p^2, i.e.

$$L_{k,l} = \begin{cases} (i,j) : j = (ki+l)(mod\, p), i \in Z_p & \text{if } 0 \le k \le p \\ (l,j) : j \in Z_p & \text{if } k = p \end{cases}$$

Most of the energy information can be found in the low-pass of ridgelet image decomposition. Normally, feature vectors are typically several thousands elements wide. In order to reduce the dimension we used PCA [13] method. More details on ridgelets can be seen in [9].

2.2 Gabor Features

Gabor filter is a popular tool for extracting spatially localized features, useful for character recognition.

An even symmetric Gabor filter has the following general form in the spatial domain.

$$G(x,y,f,\theta) = exp\left[\frac{-1}{2}\left[\frac{x'}{\sigma_{x'}}\right]^2 + \left[\frac{y'}{\sigma_{y'}}\right]^2\right] cos(2*\pi*f*x') \tag{2}$$

Where x'=x$*$sinθ + y$*$cosθ and y'=x$*$cosθ - y$*$sinθ. Design of the Gabor filter is accomplished by tunning the filter with a specific band of spatial frequency and orientation by appropriately selecting the filter parameter, such as the spread of the filter $\sigma_{x'}$,$\sigma_{y'}$, radial frequency f and the orientation of the field θ [3] [4]. Hence in this work, filtering is performed with frequency f value 4 and orientation parameters $\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3*\pi}{4}$. The values of $\sigma_{x'}$ and $\sigma_{y'}$ were empirically determined to be 2 and 4 respectively.

The resultant Gabor filter orientation feature vectors is prohibitively high (2500 elements wide for each orientation). In order to compress the dimension of Gabor features, we used PCA [13] method. Detailed description on Gabor features can be seen in [10].

2.3 Indexing Using Kd-Tree

In the proposed indexing model, the obtained feature vectors are indexed using Kd-tree data structure [11]. Given a set of K dimensional data points,the Kd-tree organizes the points in a K-dimensional space, which is useful for searching / retrieving data similar to a query. The construction algorithm of Kd-tree is as follows, at the root, we split the set of points into two subsets of roughly the same size by a hyperplane perpendicular to the first coordinate of the points.

At the children of the root the partition is based on second coordinate and so on, until depth of K-1 at which partition occurs based on last coordinate, where K is the dimension of the feature space. After depth K, again, partitioning is based on first coordinate. The recursion stops only when one point is left, which is then stored at the leaf. Because a K-dimensional Kd-tree for a set of n points is a binary tree with n leaves, it uses O(n) storage with search time being O(nlogn). In addition to this, in Kd-tree there is no overlapping between nodes. When a query feature vector of dimension K is given, search is invoked using Kd-tree and top matches that lie within a specified distance from the query are retrieved.

3 Experimental Results

The proposed character indexing system is tested on offline handwritten Kannada characters. The database has been created with the assistance of 100 writers of different streams such as school children, degree students, university students of different age group. The dataset holds 200 classes (vowels, consonants and modifiers). In this work, we considered 2000 images (i.e., 10 images per class) for experimentation. We have carried out the experiment in four stages. In each stage, we have varied class size c (where c = 50, 100, 150, 200). In every stage of our experiment, the system was trained with 4 and 9 samples and 1 image is used for testing. All our experiments are carried out on a PC machine with P IV, 2.2GHz CPU and 1GB RAM memory under Matlab 7.0 platform.

After PCA process, feature vector of dimension K is indexed through Kd-tree. We varied the number of features K (where K = 5, 10, 15, 20, 25, 30, 40, 45 & 50). Since K has a considerable impact on Correct Index Power (CIP), we choose the value that corresponds to the best CIP on the image set. In this work, we use Correct Index Power (CIP) as a performance evaluation measure for indexing, where CIP is the ratio of the correctly answered queries to the total number of queries.

The graph of Top Matches v/s CIP under different class size are shown in Figure 1-4. The features obtained using Gabor performs quite better compared to Ridgelet features under 50 class size. The same is not observed for 100, 150 and 200 class size. The features of Ridgelet perform better compared to Gabor as class size increases. It is worth to note that, the results of Ridgelet gives better CIP when it is less trained (Refer Figure 1). In case of Gabor, we observed variations under different class size (Refer Figure 2, 3, and 4).

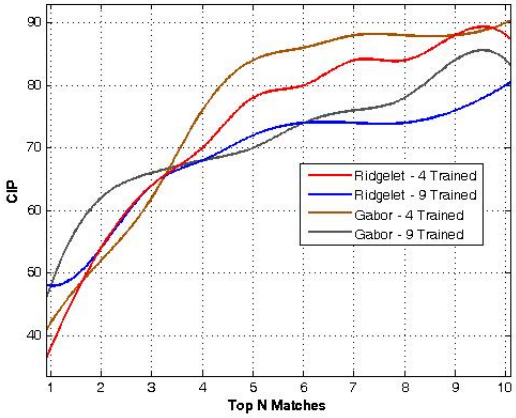

Fig. 1 Top Matches v/s CIP under 50 class size

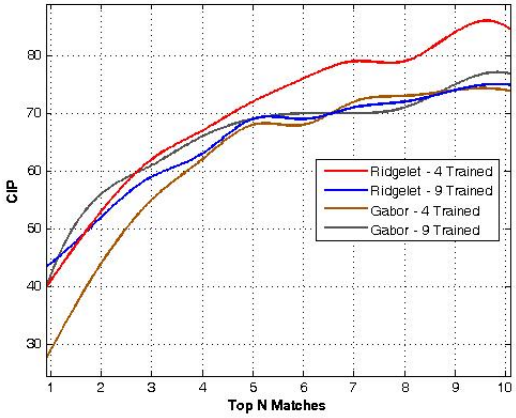

Fig. 2 Top Matches v/s CIP under 100 class size

The efficacy of the proposed indexing scheme lies in its efficiency from the point of search time. Table 1 shows the time analysis for indexing based and conventional identification method for 4 trained images. From this, it is noticed that, the proposed indexing method reduces the search time. It is also worth to note that, the percentage of time reduction is noticeable using Kd-tree based indexing model against the conventional identification method even when class size is increased.

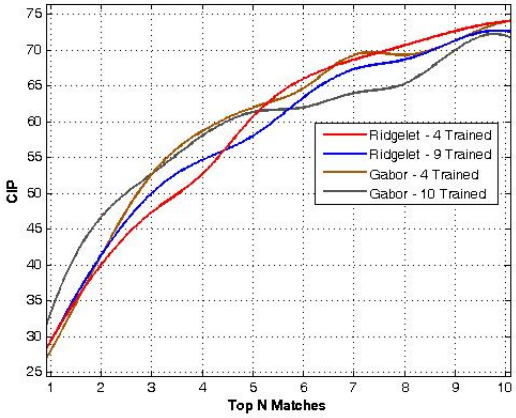

Fig. 3 Top Matches v/s CIP under 150 class size

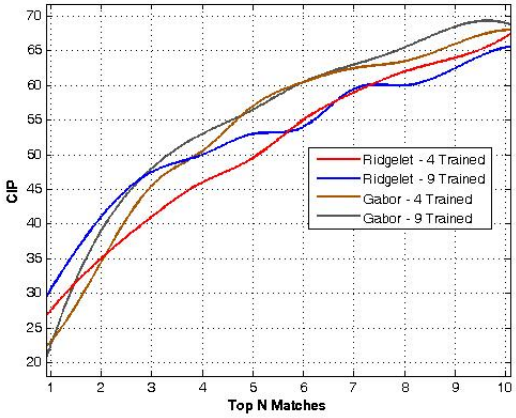

Fig. 4 Top Matches v/s CIP under 200 class size

Table 1 Time analysis for Indexing and Without Indexing

	Time in Secs.	
Class	With Indexing	Without Indexing
50	0.0019	0.016
100	0.0024	0.078
150	0.005	0.18
200	0.0057	0.29

4 Conclusion

We proposed an efficient indexing technique using Kd-tree to reduce the search space for a large class character database. As a part of feature extraction stage, Ridgelet and Gabor features are used to index the database by forming Kd-tree separately. Our first exploration on Kd-tree is proven to be suitable data structure for character identification system particularly in the analysis of execution of range search algorithm. The proposed method is recommended to be used as a supplementary tool to speed up the task of identification. From the experiment, it is clear that indexing prior to character identification is faster than conventional character identification. In near future the authors plan to work on different structural features and also on exploring different indexing schemes.

Acknowledgements. The authors would like to thank Dr. S. Manjunath and K.B. Nagasundara for their support rendered during the work.

References

1. Candes, E.J., Donoho, D.L.: Ridgelets: a key to higher-dimensional intermittency? Phil. Trans. R. Soc. Lond. A, 2495–2509 (1999)
2. Do, M.N., Vetterli, M.: Finite ridgelet transform for image representation. IEEE Transactions on Image Processing (2002)
3. Jain, A.K., Prabhakar, S., Hang, L.: A multichannel approach of fingerprint classification. IEEE Transactions on PAMI 21(4), 349–359 (1999)
4. Jain, A.K., Reisman, J.: A hybrid fingerprint matcher. Pattern Recognition 36(7), 1661–1673 (2003)
5. Jayaraman, U., Prakash, S., Gupta, P.: Indexing multimodal biometric databases using kd-tree with feature level fusion. In: ICISS 2008, pp. 221–234 (2008)
6. Lu, H., Ooi, B.C., Shen, H.T., Xue, X.: Hierarchical indexing structure for efficient similarity search in video retrieval. IEEE Trans. on Knowledge and Data Engineering 18, 1544–1559 (2006)
7. Mukherjee, R.: Indexing techniques for fingerprint and iris databases. Master Thesis, West Virginia University (2007)
8. Nagasundara, K.B., Guru, D.S., Manjunath, S.: Indexing of online signatures. International Journal of Machine Intelligence 3, 289–294 (2011)
9. Naveena, C., Manjunath Aradhya, V.N.: An impact of ridgelet transform in handwritten recognition: A study on very large dataset of kannada script. In: IEEE World Congress on Information and Communication Technologies (WCIT), pp. 622–625 (2011)
10. Naveena, C., Manjunath Aradhya, V.N., Niranjan, S.K.: The study of different similarity measure techniques in recognition of handwritten characters. In: ACM International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 781–787 (2012)
11. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley (1990)
12. Tokas, R., Bhadu, A.: A comparative analysis of feature extraction techniques for handwritten character recognition. International Journal of Advanced Technology Engineering Research (IJATER) 2(4), 215–219 (2012)
13. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)