2009 10th International Conference on Document Analysis and Recognition

Shape Descriptor based Document Image Indexing and Symbol Recognition

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

Abstract

In this paper we present a novel shape descriptor based on shape context, which in combination with hierarchical distance based hashing is used for word and graphical pattern based document image indexing and retrieval. The shape descriptor represents the relative arrangement of points sampled on the boundary of the shape of object. We also demonstrate the applicability of the novel shape descriptor for classification of characters and symbols. For indexing, we provide a new formulation for distance based hierarchical locality sensitive hashing. Experiments have yielded promising results.

1 Introduction

Traditional document image indexing applications have applied primarily two approaches, first based on text search which requires efficient and robust optical character recognizers, and second based on word spotting. The word spotting approach for building of document images, ensures that identical words have similar visual appearances. Similarity between word images is ensured by matching the word images.

In recent works, primarily two approaches have been used for word image matching: pixel level matching and feature based matching [2, 3]. Shape context matching is currently most preferred algorithm for feature based matching. In the Shape Context framework, a set of points are sampled on the boundary of the shape. A Shape Context is associated with each point which represents the arrangement of the point with respect to other points. The set of the histograms for the point set is further used for establishing correspondence between points on two shapes. The similarity between two objects is measured by a matching cost, which is obtained by solving a bipartite graph matching problem. However the graph matching problem is computationally demanding and size of the descriptor is also an issue.

We present a novel shape descriptor which is fundamentally based on shape context and tackle the above mentioned problem. The shape descriptor represents structural organization of a shape. It is applied for word and graphic pattern based document image indexing. The indexing framework is provided by hierarchical distance based locality sensitive hashing. The Distance based hashing proposed in [6], is a novel formulation which can be applied for arbitrary distance measures. This advantage is exploited for document image indexing. The proposed shape descriptor with slight advancements is further extended for symbol shape representation and is applied for symbol recognition problem.

The organization of the paper is as follows. In the section 2, we describe the algorithmic framework to generate the shape descriptor representation for word images. The section 3 will have discussion on the hierarchical distance based hashing. The details of the document Image Indexing scheme is presented in section 4. The section 4 will also discuss about symbol retrieval problem. Experimental results are presented in section 5. Finally, we conclude and give perspective of our work.

2 Fourier based Shape Descriptor

We treat the shape of an object in an image as a point set P_i for $i = 1, ..., L$. The shape descriptor represents the relative arrangements of these points in the form of a log polar histogram. We perform bound detection of the shape in the image using horizontal and vertical profiling as initial step. The histogram computation is dependent on relative distance and orientation

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.63

IEEE computer society

of shape descriptor points. Thus for a object, the count of shape descriptor points varies in object's instances in different orientations. We handle this problem by performing Principal Component Analysis and aligning the object to its principal eigen axis. The PCA analysis forms the preprocessing step of shape descriptor computation followed by bound detection.

The sampling of points P_i on the shape is done by placing a logical grid over the shape image. The transition points while traversing over the grid lines are the shape descriptor points. Additionally the points on the grid which lie on the boundary of the shape are also considered as shape descriptor points. This approach of sampling of shape descriptor points gives better representation of complex shapes having multiple contours. In the figure 1 green points are the transition points on horizontal grid lines and red points are transition points on the vertical grid lines.

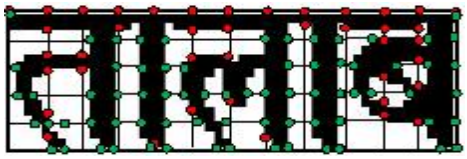

Figure 1. Shape descriptor points

The population of shape descriptor points varies with the complexity of the shape in the image. The distribution of the shape descriptor points based on relative arrangement is represented by logpolar histogram. For a point set $P = P_i$ for $i = 1, ..., L$, logpolar histogram is defined as

$$H_i(k) = \{q \neq p_i | (q - p_i < bin(k))\}$$

The histogram $H_i(k)$ is defined as the shape context [4] of point P_i. The shape context represents the count of points that fall inside each bin k, centering the logpolar origin at the point P_i. The computation of $H_i(k)$ for P_i can be done as:

- $R_{i,j} = log(l_{i,j}) - min(log(l_{i,j}))$ for $i, j = 1, ..., L$ and $i \neq j$ where $l_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
- $\alpha_{i,j} = tan^{-1}\left\{\frac{y_j - y_i}{x_j - x_i}\right\}$ for $i, j = 1, ..., L$ and $i \neq j$
- Input parameters :: Δ_r, Δ_α, M, N. Δ_α and Δ_r are the logpolar histogram resolution parameters
- for point set $P = P_i, i = 1, ..., L$, M and N should be such that $M >= floor(Rm_i/\Delta_r)+1$ and $N >= floor(\alpha m_i/\Delta_\alpha) + 1$, where $Rm_i = max(R_{i,j})$ and $\alpha m_i = max(\alpha_{i,j})$ for $j = 1, ..., L$ and $i \neq j$
- for point set $P = P_i, i = 1, ..., L$, initialize $h_{i,p,q} = 0$ for $p = 1, ..., M$ and $q = 1, ..., N$
- $p = floor(Rm_i/\Delta_r)+1$ and $q = floor(\alpha m_i/\Delta_\alpha)+1$, $h_{i,p,q} \leftarrow h_{i,p,q} + 1$ for $i, j = 1, ..., L$ and $i \neq j$

The integration of all the shape contexts represents the relative arrangement of points sampled on the boundary of shape. Because of noisy nature of document images there can be some isolated noisy bins in the integrated histogram. The noisy bins are filtered out and we select only dominant histogram and then normalize it.

- $h_{integrated,p,q} = \sum_i h_{i,p,q}$ for $i = 1, ..., L$
- $h_{integrated,p,q} = h_{integrated,p,q} / \sum h_{integrated,p,q}$

The rotation invariance can be incorporated in the dominant histogram by transforming it to fourier domain [5]. The absolute value of the fourier coefficients represents the shape descriptor of the object image.

$$F(P) = \|FFT(h_{integrated,p,q})\|$$

The Shape Descriptor is a matrix of MXN. It represents relative distribution of shape descriptor points with respect to distance and orientation.

3 Hierarchical Distance Based Hashing

3.1 Locality Sensitive Hashing

The Locality Sensitive Hashing(LSH) algorithm has been successfully applied for solving different nearest neighbor problems in high dimensional space [1]. LSH takes into account the locality of the points so that nearby points remain nearby, and solves the $(1+\epsilon)$ approximate nearest neighbor problem avoiding the curse of dimensionality. The LSH 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 core of LSH is scalar projection, mapping data points in high dimensional space to hash space, building a hash table. The hash value $h(.)$ obtained for a data point is the index of a bucket in the hash table hash table. A family of hash functions is said to be interesting if it projects nearby points to location close in hash space. This requires that:

- For any points p and q in R_d that are close to each other, the probability P_1 is high that they fall into the same bucket

$$P_H = [h(p) = h(q)] \geq P_1 for \|p - q\| \leq R_1 \quad (1)$$

- For any points p and q in R_d that are far apart, the probability $P_2 < P_1$ that they fall into the same bucket

$$P_H = [h(p) = h(q)] \leq P_2 for \|p - q\| \geq cR_1 = R_2 \quad (2)$$

Where c is a real number greater than 1.

The difference between P_1 and P_2 can further be increased by performing projection by combination of k randomly selected hash function. This increases the ratio of the probabilities of separation because $(P_1/P_2)^k > (P_1/P_2)$. The k bit real number obtained for each data point after projection is the corresponding hash index. The nearest neighbor search for a query point can be performed by mapping the query point to hash space using k hash functions and then performing a linear search through all the points that fall into the same bucket as the query. Within each set of k dot products, we achieve success if the query and the nearest neighbor are in the same bin in all k dot products. To reduce the impact of an *unlucky* quantization in any one projection, we form L independent projections i.e. independent hash tables and and pool the neighbors from all these tables and find the nearest neighbor.

3.2 Distance Based Hashing

The Distance Based hashing(DBH) is a method for applying hash-based indexing in arbitrary spaces and distance measures. The definition of the hash functions depends on distances between objects. The distance measure between the objects is an open choice, we can take euclidean, metric or any user defined distance measure.

Several methods are available for defining functions that map an arbitrary space (X, D) into the real line R [2]. In an arbitrary space (X, D), for two points (X_1, X_2), pseudo line projections $F^{X_1,X_2} : X \rightarrow R$ for a data point x is defined as

$$F^{X_1,X_2}(x) = \frac{D(X_1,x)^2 - D(X_2,x)^2 + D(X_1,X_2)^2}{2D(X_1,X_2)} \quad (3)$$

If (X, D) is a euclidean space, then $F^{X_1,X_2}(x)$ computes the projection of point x on the unique line defined by points X_1, X_2. If X is a general non-euclidean space, then $F^{X_1,X_2}(x)$ does not have a geometric interpretation. However as long as a distance measure D is available, F^{X_1,X_2} can still be defined and provides a simple way to project x onto R. The family of functions(eqn.3) define a rich family of functions, where a pair of data points defines a different function. Given a database of n data points, we can define about $n^2/2$ unique functions by applying eqn.3 to pairs of data points from X.

The functions defined in eqn.3 are real-valued, whereas hash functions need to be discrete-valued. We can easily obtain discrete-valued hash functions from F^{X_1,X_2} using thresholds $t_l, t_2 \in R$:

$$\begin{aligned} F^{X_1,X_2}_{t_1,t_2}(x) &= \{\mathbf{0}, \quad if\ F^{X_1,X_2}(x) \in [t_1,t_2]\} \\ &= \{\mathbf{1}, \quad otherwise \quad\} \end{aligned} \quad (4)$$

In practice t_1 and t_2 should be chosen so that $F^{X_1,X_2}_{t_1,t_2}(x)$ maps approximately half the data points in X to 0 and half to 1, so that we can build balanced hash tables. We can formalize this notion by defining, for each pair $(X_1, X_2) \in X$, 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.

$$V(X_1,X_2) = [t_l,t_2] | Pr_{x \in X}(F^{X_1,X_2}_{t_1,t_2}(x) = 0) = 0.5 \quad (5)$$

Now we can define a family H_{DBH} for an arbitrary space (X, D) as

$$H_{DBH} = F^{X_1,X_2}_{t_1,t_2}(x) | X_1, X_2 \in X, [t_1,t_2] \in V(X_1,X_2) \quad (6)$$

If function family H_{DBH} is locality sensitive, then DBH would be a special case of LSH. Locality Sensitivity property of the H_{DBH} is established by exploiting the geometric information of the auxiliary space which is a complicated task because arbitrary spaces have arbitrary geometries. The performance of H_{DBH} can still be analyzed by exploiting statistical information obtained from sample data, even if geometric constraints (such as euclidean properties and/or the triangle inequality) are not available about the auxiliary space.

3.3 Hierarchical DBH

The points distribution in most of the real datasets is nonuniform. Since the LSH partitions that space uniformly without considering the distribution of points, it may lead to few heavily populated buckets with rest of the buckets having sparse population of points. This considerably reduces the efficiency of LSH in terms of accuracy and reduction in average number of comparisons for nearest neighbor search. The problem can be solved by applying the hierarchical concept in LSH, where the points in certain buckets are successively hashed into new hash tables(figure 2).

The bucket selection for successive hashing can be based on either the population criterion or distribution of points in a particular bucket. The hash functions for successive hash tables are generated by points of respective buckets.

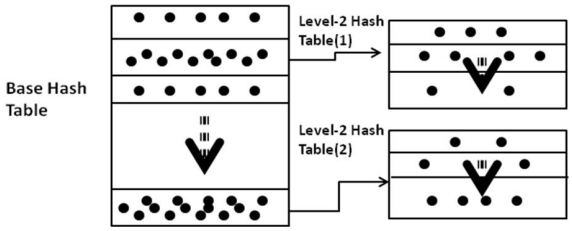

Figure 2. Hierarchical Hash Tables

4 Methodology for document image indexing and symbol recognition

In this section we give brief introduction of our methodology for Document Image Indexing and Symbol Recognition problem using distance based hashing.

4.1 Document Image Retrieval

The steps in our application for Distance based LSH for indexing and retrieval of documents are listed as follows

1. Identification of the text and graphics region in the document image.

2. Extraction of the text lines using horizontal projection profiles of intensity, and segmentation of every text line into constituent word images using vertical projection profiles.

3. Hash index calculation for each graphics and word image using fourier based shape descriptor. The generation of hash function by eqn.3 for base hash table is done off line. The subset of images for function generation is selected randomly from the set of images.

4. Each hash index indicates a bucket or group. Ideally each group of a word image or graphics pattern consists of all respective occurrences in the analyzed document collection. Groups are further annotated in order to allow an index generation for respective image.

5. Query process involves the hash index calculation for using fourier based shape descriptor.

6. The hash index for the query indicates the bucket or group. Using the group annotation the documents are retrieved in the ranked order.

4.2 Symbol Recognition

The variation in symbol images in terms of rotation and scaling in comparison with word images is more. The present form fourier based shape descriptor is insufficient for such cases. The problem can be solved by transforming the image so as to align them to its principal eigen axis using *hotelling transform*. The next step of shape descriptor computation of transformed image is performed as in section 2.

5 Experimental Results

The experimental results of document indexing application in hierarchical DBH framework and symbol recognition are discussed below. We also present the application of shape descriptor for character classification problem.

5.1 Word and Graphic pattern based document image indexing

The word and graphic pattern based indexing scheme is experimented on sample pages from hindi, bengali and malyalam languages. The database consists of 10230 images segmented from 385 documents with 3145 hindi, 3690 bengali, 2771 malyalam and 624 graphics patterns. The preprocessing step include only bound detection because the sample images do not have orientation problem. Since the length of words and graphic patterns vary considerably, we place variable number of grid lines over the shape at resolution of 4 pixels. This scheme preserves the aspect ratio of shape. The number of pixel rows in the image after bound detection is fixed to 48. The shape descriptor parameters are $\Delta_r = 0.05$, $\Delta_\alpha = 6$, $M = 60$ and $N = 60$. The indexing scheme is implemented with hierarchical DBH. The hash functions for the base hash tables are generated by random selection of 10% of the images. The hierarchical hash tables are build for two most populated buckets in the base hash table. We experimented with different combinations of L and k, $L = 4$ and $k = 5$ gave best results. Euclidean distance is selected as the distance measure. The performance evaluation of was done with about 400 queries. The retrieval precision and recall of our algorithm is given in table below.

The performance evaluation is performed using Mean average Precision(MAP). The results are given in the table below.

Language	Queries	Prec.	Rec.	MAP
Hindi	146	87.9	88.4	0.267
Bengali	134	87.4	89.0	0.252
Malyalam	108	84.6	87.6	0280

5.2 Character Classification

The proposed shape descriptor is applied for Bengali Script Characters(figure 3) classification problem. The dataset consists of 17022 characters belonging to 49 categories from different documents. The size of image characters is varying from 22X26 to 52X48. The KNN and SVM in OAA framework is applied for classification. The partitioning of the dataset is done as 70% of character images as testing data and 30% as training data. The images are resized to 32X32 after the bound detection and fixed grid size of 8X8 and 16X16 is applied for the experiment. The Shape Descriptor parameters are $\Delta_r = 0.05$, $\Delta_\alpha = 12$, $M = 35$ and $N = 30$. We have selected cosine distance as measure of distance between symbol images. The results are given in the table below.

-	Acc.(8X8 grid)	Acc.(16X16 grid)
KNN(K=3)	95.20	96.47
SVM(OAA)	99.21	99.81

Figure 3. Bengali Script Characters

5.3 Symbol Recognition

We selected test set of GREC[1] for our symbol recognition experiment. It contains 300 images divided in two subsets. The model set A contains 50 different symbol images(50 classes). The model set B contains 250 instances of 50 symbols obtained by the linear transformations on each symbol image in set A. The frequency of instances of each class in set B is not equal, the maximum frequency is 10 and minimum is 1.

We select set B as the training image set. Before shape descriptor computation, the symbol images are aligned to its principal eigen axis. The preprocessing step include bound detection and then resizing to 100X100 pixels. Two trials of the experiment is performed with grid of size of 25X25, and 50X50. The Shape Descriptor parameters are $\Delta_r = 0.05$, $\Delta_\alpha = 10$, $M = 45$ and $N = 36$. We used Distance based Hashing for retrieval. The hash functions are generated by random selection of 20% of images from the set B. The hash table parameter are finalized as $L = 3$ and $k = 4$. We obtained precision of 78% and recall rate 66% with 25X25 grid and precision as 80% and recall rate as 71% with 50X50 grid.

[1] http://www.cvc.uab.es/grec2003/SymRecContest/

6 Conclusion

We have proposed a novel shape descriptor for shape representation. The effectiveness of the shape descriptor framework is demonstrated by word image and graphic pattern based indexing, symbol recognition and character classification problems. A novel hierarchical distance based hashing framework is introduced for document image indexing.

Acknowledgment: This work was funded by MCIT, Government of India as a part of project ″Development of Robust Document Analysis and Recognition System for Printed Indian Scripts″. The authors are thankful to Prof. B. B. Chaudhuri for providing Bengali Script Character dataset for experiments.

References

[1] G. Aristides, I. Piotr, and M. Rajeev. Similarity search in high dimensions via hashing. *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529, 1999.

[2] C. Faloutsos and K. I. Lin. Fastmap:; a fast algorithm for indexing, data mining and visualisation of traditional and multimedia datasets. *Proceedings of ACM International Conference of Management of Data(SIGMOD)*, pages 163–174, 1995.

[3] G. Harit, R. Jain, and S. Chaudhury. Improved geometric feature graph: A script independent representation of word images for compression, and retrieval. *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 421–425, 2005.

[4] B. Serge, M. Jitendra, and P. Jan. Shape matching and object recognition using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[5] Y. Su and W. Yuanyuan. Rotation invariant shape contexts based on feature-space fourier fourier transformation. *Proceedings of the 4th International Conference on Image and Graphics*, pages 575–579, 2007.

[6] A. Vassilis, P. Michalis, P. Panagiotis, and K. George. Nearest neighbor retrieval using distance based hashing. *International Conference on Data Engineering*, pages 327–336, April 2008.