How to Define Local Shape Descriptors for Writer Identification and Verification

Imran Siddiqi and Nicole Vincent

Laboratoire CRIP5 – SIP, Université Paris Descartes – Paris 5
45, rue des Saints-Pères, 75270 Paris, Cedex 06, France

Abstract. This paper presents an effective method for writer identification and verification in handwritten documents. The idea is that within a handwritten text, there exist certain redundant patterns that a particular writer would use frequently as he writes and these forms could be exploited to recognize the authorship of a document. To extract these patterns, the text is divided into a large number of small sub-images and a set of shape descriptors is extracted from each. Similar patterns are then clustered together for which a number of clustering techniques have been evaluated. The writer of the unknown document is identified by Bayesian classifier. The system trained and tested on 55 documents of the same number of authors, exhibited promising results.

1 Introduction

Writer authentification has received considerable attention over the recent years, not only from the perspective of behavioral biometrics [1,7,9,11] but also in the context of handwriting recognition [5] exploiting the principle of adaptation of the system to the type of writer. A wide variety of techniques have been proposed that are generally classified as either being global [2,8], that are based on the overall look and feel of the writing, or local [1,3,10], which identify the writer based on localized features of writing, which are inherent in the way a writer specifically writes characters. The present communication is based on the later idea. Our method relies on the frequent shapes of the drawing and is linked to the physical way the lines or loops are produced, hence the chosen observation scale is inferior to that of a letter. The method has been detailed in the following section.

2 Proposed Method

We now present our method and its application to writer authentification. We start with the offline training of the system to enroll the authorized authors in a reference base, extracting a set of features for each. The writer of the questioned document is then identified/verified in the classification phase. Feature extraction is based on the argument that within a handwritten text, there exist certain redundant patterns that a

Siddiqi I. and Vincent N. (2008).
How to Define Local Shape Descriptors for Writer Identification and Verification.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 199-204

writer would use frequently as he writes. To extract these patterns, the handwritten document image is first binarized and the connected components in the text are extracted. Each component is then divided into a large number of small windows of size *n*x*n*. The division is carried out following the text as illustrated for the word 'headlines' in figure 1. The window size *n* is chosen empirically [10] and has been set to 13x13 in our case.

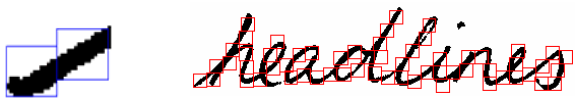

Fig. 1. Division of text into sub-images.

Once the text has been divided into sub-images, we proceed to the extraction of a set of shape descriptors from each window. This set includes: *Horizontal Projection:* The number of text pixels in each row of the sub-image. *Vertical Projection:* The number of text pixels in each column of the sub-image. *Upper Profile:*The distance of first text pixel from the top of each window. *Lower Profile:* The distance of the last text pixel from the top of each window. *Orientation:* The overall direction of the shape (ranging from -90º to 90º). *Eccentricity:* The ratio of the length of the longest chord of the shape to the longest chord perpendicular to it. *Rectangularity:* The ratio area of the object/area of the bounding box. *Elongation:* The ratio of the height and width of a rotated minimal bounding box. *Perimeter:* The number of pixels in the boundary of the shape. *Solidity:* The proportion of the pixels in the convex hull that are also in the region and is computed as Area/Convex Area.

Once all the descriptors have been calculated, the values are normalized (0 – 1) and hence each window is represented by a vector of dimension $d=4n+6$, where n is the window size. In our case, let S be the set of sub-images, thus we have:

$$S = \{Si\},\ with\ each\ S_i = (s_i^1, ..., s_i^d) \tag{1}$$

Once the sub-images have been represented by a set of features, we proceed to their clustering. The objective is that the sub-images that have been produced by the same gesture of hand are grouped in the same classes. We have evaluated a number of clustering algorithms (detailed in the section to follow) and, as a result of clustering, the document is represented by a set of classes C. For each class C_k we have:

$$C_k = \{S_{1,k}, ..., S_{m,k}\},\ m = card(C_k)\ \ And\ each\ S_{i,k} = (s_{i,k}^1, ..., s_{i,k}^d) \tag{2}$$

We calculate its probability of occurrence $P(C_k)$, the mean vector $\overline{S}_k$,and the covariance matrix Cov_k , thus representing the document D as:

$$D^r = \{F_k, k \le card(C)\} \qquad where: F_k = \{P(C_k), Cov_k, \overline{S}_k\} \tag{3}$$

3 Clustering of Sub-images

We now present an overview of the clustering methods (which do not need to know a priori the number of clusters to retain) that we have employed. An important step in any clustering is the choice of similarty measure between two patterns. We calculate the dissimilarity between two patterns using a distance measure (E uclidean distance) defined on the feature space. We have experimented with several algorithms which we will discuss briefly.

3.1 Sequential Clustering (Seq)

We start with a simple sequential clustering algorithm. We choose a similarity threshold and start with the feature vector of the first sub-image as the centroid of the first class. For each of the subsequent patterns, we calculate the Euclidean distance between the current element and the mean of each class. The element is then attributed to the nearest cluster. In case, it is not close enough to any of the clusters (with respect to the threshold), a new cluster is created. The problem however is that the procedure is sensitive to the order in which the patterns are presented.

3.2 Multi-Phase Sequential Clustering (MP-Seq)

To address the problem with sequential clustering, several clustering phases with random selection of the data points are carried out in order to be less sensitive to the initial conditions. Each of the clustering phases provides thus a variable number of clusters. The final clusters are defined as the groups of patterns that are always clustered together during each sequential clustering phase [1]. The leftovers (patterns that are not part of any cluster) are then assigned to the nearest class (using mahanalobis distance).

3.3 Two-Step Sequential Clustering (2Step-Seq)

We now propose a two-step clustering algorithm that could be viewed as a hybrid of the partitional and sequential clustering. We partition the set S into Φ disjoint sub-sets (S_{p1}, ...,$S_{p\Phi}$), setting a criteria on dimension k of the feature vector. Each of the Φ partitions is then separately clustered using the sequential algorithm defined above and the results of these Φ clustering procedures are merged to get the final set of clusters. For our case we chose to partition on the *Orientation* of the trace defining Φ =4, thus creating four equal partitions on the interval –90º to 90º.

3.4 Minimum Spanning Tree Clustering (MST)

We now present the graph based minimum spanning tree clustering. A minimum spanning tree (MST) of a weighted graph connects all the given data points at the

lowest possible cost. From the perspective of clustering: if the weights of the edges represent the distances between the data points, removing edges from the MST leads to a collection of connected components which can be defined to be clusters [6]. For our set of sub-images, we define the weighted (undirected) graph G(S) = (V,E) as follows: the vertex set V = {S_i | S_i Є S} and the edge set E = {(S_i, S_j) | for S_i, S_j Є S and i ≠ j}. Each edge (u, v) Є E has a weight that represents the (Euclidean) distance d(u, v), between u and v. The MST is constructed using Prim's algorithm and the clustering is based on the idea that two data points with a short edge-distance should belong to the same cluster (sub tree) and data points with a long edge-distance should belong to different clusters and hence be cut. The number of clusters obtained, naturally, is sensitive to the threshold value chosen. The algorithm works quite well provided the inter-cluster edge-distances are clearly larger than the intra-cluster edge-distances.

Once the sub-images have been clustered (by one of the methods discussed above), we sort the classes and keep only those having sufficient number of elements. The term sufficient however is relative, so we pick the top most important *M* classes which allow to cover 90% of text pixels in the image (figure 2).

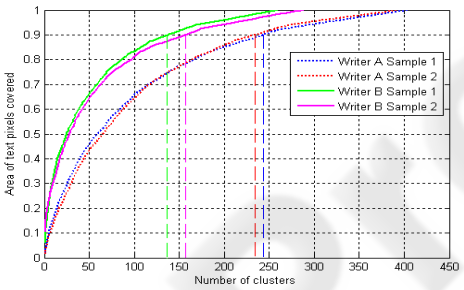

Fig. 2. Number of clusters and the corresponding area of text pixels covered.

Once we find *M*, we have the set of classes C^D for document *D*:

$$C^D = \{C_k \mid k \le M\} \tag{4}$$

4 Writer Recognition

The first step towards writer identification is the extraction of features from the document whose writer is to be identified. We start with a binarization of the test image followed by the division of text into small sub-images and then their clustering (as discussed in the previous sections), thus representing the test document *T* by the set of the mean vector of each class. If the difference between the number of classes of *T* and the reference document *D* is above a certain threshold, we straightaway discard *D* and proceed to the next document of the reference base.

$$\frac{\left|card(C^T) - card(C^D)\right|}{card(C^T)} < \eta \tag{5}$$

If a reference document satisfies condition (5), we proceed to the next step and employ the Bayesian decision theory to calculate the similarity between two documents. We find the similarity index between document *T* and all the documents in the reference base *R* and identify the writer of the questioned document as the author of the document maximizing the index.

Writer verification is performed in the classical Neyman-Pearson framework of statistical decision theory. Varying the decision threshold on the similarity index calculated between two documents, the Receiver Operating Characteristic (ROC) curves are computed and the writer verification performance is quantified by the Equal Error Rate (EER).

5 Experimental Results

To evaluate our system, we have taken a set of 55 writers from the IAM database [4], with one image of each used in training and one in testing. Each image contains on the average, 8-10 lines of text. We present a comparative overview of the recognition rates achieved with different clustering methods as illustrated in figure 3.

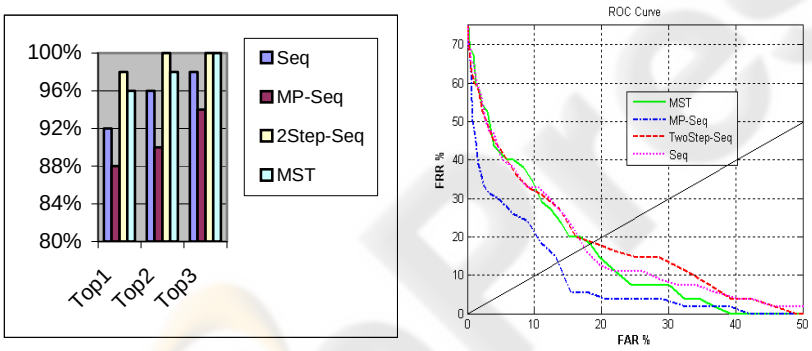

Fig. 3. Comparison of identification and verification results.

As it can be noticed from the result comparison, the two-step sequential clustering out performs the rest achieving an identification rate of 98%. The MST clustering performs approximately equally good (96%), but calculating the MST is quite time consuming. The multiphase sequential clustering, although producing very fine clusters, falls sufficently behind the rest.

For the verification task, we compute the ROC curves for different clustering methods. Contrary to identification, its the MP sequential clustering that performs the best (achieving an equal error rate (EER) of around 13%). Clearly, the performance of the system on writer verification is not as good as that of identification and needs to be improved. One possible way could be to have a writer dependent verification threshold instead of using a global threshold value for all the writers in the reference base. Another option could be to introduce some sort of post processing when the difference between the similarity index of the best match and that of the runner up is less than a certain threshold. These will be addressed in the research to follow.

6 Conclusions

We have presented an effective method for writer recognition in handwritten documents. The method is based on segmenting the writing into small sub-images, extracting a set of shape descriptors from each, hence finding a set of patterns that an individual would use frequently while writing. The realized identification rates are very promising and validate the arguments put forward in this paper. The results on verification, however, are not as good and need to be improved which will be the subject of our future research. Changing the window size *n* during the phase of handwriting division, this method could be applied to non-Latin languages as well. In addition, the system could be made more robust by automatically adjusting the window size depending upon the writing details.

References

1. A. Bensefia, T. Paquet and L. Heutte: A writer identification and verification system, Pattern Recognition Letters, Elsevier Science Inc. New York, Vol 26, issue 13, (2005) 2080-2092
2. V. Bouletreau, N. Vincent, R. Sabourin, H. Emptoz: Handwriting and signature: one or two personality identifiers?, In Proc.of Fourteenth International Conference on Pattern Recognition, Los Alamitos, CA, vol.2, (1998) 1758-1760
3. M. Bulacu, L. Schomaker, and L. Vuurpijl: Writer identification using edge-based directional features, In Proc. of 7th International Conference on Document Analysis and Recognition, volume II, Edinburgh, Scotland, (2003) 937–941
4. U. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition", In Proc. of 5th International Conference on Document Analysis and Recognition, Bangalore, India, (1999) 705-708
5. A. Nosary, L. Heutte, T. Paquet, Y. Lecourtier, "Defining writer's invariants to adapt the recognition task", In Proc. of 5th International Conference on Document Analysis and Recognition, Bangalore, India, (1999) 765-768
6. N. Päivinen: Clustering with a minimum spanning tree of scale-free-like structure, Pattern Recognition Letters, Vol. 26 , Issue 7, Elsevier Science Inc. New York, (2005) 921-930
7. R. Plamondon and G. Lorette, "Automatic signature verification and writer identification – the state of the art", Pattern Recognition, vol. 22, n°2, (1989) 107-131
8. H.E.S. Said, T.N Tan, K.D. Baker, "Personal Identification Based on Handwriting", Pattern Recognition, vol. 33, (2000) 149-160.
9. A. Seropian and N. Vincent, "Writers Authentication and Fractal Compression", Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), (2002)
10. I. Siddiqi and N. Vincent: Writer Identification in Handwritten Documents, In Proc. of the 9th Int'l conference on Document Analysis and Recognition (ICDAR 07), Curitiba, Brazil, (2007) 108-112
11. S. Srihari, S. Cha, H. Arora, and S. Lee, "Individuality of handwriting", J. of Forensic Sciences, 47(4):1.17, (2002)