Invariant Extraction Method applied in an Omni-Language Old Document Navigation System

Quang Anh BUI, Muriel VISANI and Rémy MULLOT
Laboratory L3i, University of La Rochelle
La Rochelle, France

quang_anh.bui@univ-lr.fr, muriel.visani@univ-lr.fr, remy.mullot@univ-lr.fr

Abstract—We are currently working on the concept of an omni script and interactive word retrieval system for ancient document collection navigation, based on query composition for non-expert users. To make the query, the user selects and composes writing pieces, which are invariants automatically extracted from the old document collection. In order to extract invariants from documents, strokes must be first extracted and clustered. Stroke extraction raises two main difficulties: detecting the ambiguous zones so as to extract primary strokes (writing pieces which do not contain any ambiguous zone) and grouping the primary strokes so as to form invariants. In this paper, we present existing methods for ambiguity zones detection and compare these methods on documents of different languages and periods to find out which one is more adapted in our context. Once ambiguous zones have been extracted, some neighboring primary strokes are grouped so as to obtain strokes and our clustering algorithm is applied over these strokes to find their representatives, i.e. the invariants. These invariants can further be used by the user to compose his/her query and to retrieve words from the document collection.

Keywords—Word Retrieval, Invariant Extraction, Stroke Extraction, Clustering, Ambiguous Zones Detection

I. INTRODUCTION

A huge amount of human’s knowledge is stored in ancient documents, spread all over the world. Digitization is used to protect this heritage and to make it accessible to everyone. In order to provide fast access to that knowledge even though the masses of ancient documents available and their variability in terms of scripts and languages, different automatic transcription methods based on word recognition systems have been proposed. This process is an alternative to human transcription which is too slow and too expensive. However, the bad quality of old documents makes it difficult to achieve good results. Traditionally, most word recognition systems are specific to a given language. But, many ancient or rare languages and/or scripts do not have any dedicated word recognizer. “Word spotting” or “word retrieval” systems are an alternative to access historical document collections without any recognition system. “Word spotting” [2, 3] consists in locating all the occurrences of a given word image (query) that the user has previously selected in the document. The main drawback of this method is that user has to spot at least one occurrence of the word to search. To circumvent this problem, an alternative solution was proposed: “word retrieval” [1]. In “word retrieval”, the user generates his query by using a predefined coding, where the code represents pieces of characters, pictograms, ideograms, etc. One drawback of most word retrieval systems is that it relies on a coding which is generally specific to a given script.

We are currently working on a generic omni-script and interactive word retrieval system adapted to old documents. This system is based on 3 stages. The first stage (off-line) is the automatic extraction of invariants from a document collection. These invariants will constitute the codebook for the user to compose interactively queries in a second stage. We extract invariants rather than characters because some languages are not based on an alphabet. The third stage consists in retrieving the word images in the document collection which are similar to the query. In this paper, we focus on the first step: invariant extraction. The main objective of the first stage is to extract automatically consistent invariants from a given document collection.

In order to extract invariants, we first extract strokes. Strokes were originally defined as the path between pen-up and pen-down in the case of handwritten material. They are challenging to extract in the off-line case where no temporal information is available. As we consider old documents, we are working on an off-line system, with both handwritten and printed material. In our case, we therefore define a stroke as a pattern that frequently occurs within the document collection, this pattern consisting in a set of connected points of the writing between 2 ends or junctions. Even if many methods have been proposed in the literature [4, 5, 6, 7, 8, 9], stroke extraction from offline image remains an open issue. There are 2 main processes in a stroke extraction system: ambiguous zone detection (which consists in extracting primary strokes) and primary stroke grouping (which consists in solving the segmentation ambiguities by merging primary strokes to form strokes).

Our objective is then to extract, for each document collection, a limited set of invariants that may further be used for interactive query composition and word retrieval. Those invariants must be in a limited number and meaningful to the user (because the user will use them to easily compose his/her query). We select them by clustering strokes.

This paper is organized as follows. In section II, we present some existing methods for ambiguous zone detection from offline documents and provide an experimental comparison of different primary stroke extraction methods using documents of different languages and periods. In section III, we review existing methods for primary stroke grouping and propose a
new method. In section IV, we present our invariant extraction method based on the interactive clustering of strokes and we introduce our user interface for cluster visualization and interaction.

II. AMBIGUOUS ZONE DETECTION – PRIMARY STROKE EXTRATION

A. State-of-the-art approaches

There are parts of the writing where establishing which stroke it belongs to is not straightforward, e.g. crossings or touching components. These zones are called ambiguous zones. Primary strokes are defined as the set of connected points of the writing between 2 ends or ambiguous zones. Methods found in the literature for ambiguous zone detection are mostly based on the skeleton [7, 8, 9] or on the contour of the handwriting [4, 5, 6].

In [7], using skeleton of the image, the author find candidate fork points (CFP) which indicate ambiguous zones in the image, then, for each CFP, an ambiguous zone is located by a polygon whose vertices are contour points with the local minimum distance to this CFP. More generally, the advantage of skeleton based approaches is computational efficiency while maintaining acceptable geometric and topological attributes of the writing. But, the results of these approaches are dependent on the stability of the stroke width inside one document image.

In the case of contour based approaches, each point on the contour matches a position of the writing and it is a clue for ambiguous zone detection. In [4], authors demonstrated that ambiguous zones can be located using dominant points of the contour. These points correspond to stroke’s end points, or to overlaps of consecutive strokes. In [6], authors classified ambiguous zones into 2 types (basic and complex) using dominant points. The main drawback of these approaches is that they rely on curvature estimation algorithms, which are mostly unreliable in the presence of degraded (old) documents. In [5], the authors detect ambiguous zones using a probabilistic model which is based on a parametric representation of strokes. The main drawback of this method is that its parameters must be tuned for any script or language.

Once ambiguous zones are detected and localized, primary strokes can be extracted as the writing segments that do not contain any ambiguous zones.

B. Experimental comparison

Ambiguous zone detection takes an important role in stroke extraction. In order to compare the methods in [7], [4] and [5], regarding multiple scripts and periods, we perform experiments on the database shown in Fig.1. It consists of 2 handwritten Latin pages (containing 2051 connected components) from the “Saint Gall” database (from the 9th century) [18], 2 pages (containing 934 connected components) from the old handwritten Chinese book “Bai shi wen ji” (written in 1618), and 2 pages (containing 1933 connected components) in printed Telugu (Indian) from the modern book “Bhavishya Puranam” (published in 1954).

We created the ground-truth manually, which explains the relatively low number of pages in this database. However, we do not work at the page level, but at the connected component level, and we have from 934 to 2051 connected components per language. We choose to evaluate the different algorithms with not only old documents (the Saint Gall database and the Chinese book) but also modern document (the one written in Telugu) so as to measure their robustness towards the level of degradation of the document. In each document, we first apply a pre-processing method to binarize the image and remove noise. Then, connected components are extracted. Further, ambiguous zones are detected inside these connected components using methods [7], [4] and [5]. We compare the outputs of each algorithm with the ground-truth, varying the parameter values of each algorithm. After selecting, for each database and for each method, we count the number of missing ambiguous zones and the number of wrongly detected ambiguous zones, and calculate precision and recall. (see Table 1)

As the method in [4] relies on an algorithm of curvature estimation, which is unreliable in the presence of degraded documents, it is the least efficient on our databases. This method performs the worst on the Chinese document, which is the most degraded, while it gives its best results on the Indian document, which is the newest and least degraded.

The method of L’Homer [5] provides the best precision results on our 3 databases (the method in [7] giving similar results though). This method uses a probabilistic model for
representing strokes. Therefore, for each database, we have to adapt the parameters of the probabilistic model to the type of script and the characteristics of the document. This requires a huge work and expertise while our goal tends to work with documents of different languages and non-expert users. Therefore, this method is not suitable for our system.

As we can see in Table 1, the method in [7] is very effective on the 3 databases of different languages, giving similar results to the method in [5] in terms of precision. Those good results may be explained by the fact that the contents of our documents are homogenous enough. Indeed, even in handwritten documents, the writer tends not to change stroke width in such old documents. The recall is generally better than with the methods in [4] and [5], except for the Saint Gall database, which might be explained by the fact that the handwriting is highly cursive in that case. Additionally, this method does not require any training stage so, different from the method in [5], does not have to be adapted manually to a given script/language.

Based on these experiments, we can conclude that the method in [7] is the most suitable for our word retrieval system.

III. PRIMARY STROKE GROUPING – STROKE EXTRACTION

A. State-of-the-art approaches

In this stage, primary strokes and ambiguous zones are grouped together in order to find out strokes.

In [7], the authors use a graph where each ambiguous zone or primary stroke corresponds to a vertex. If an ambiguous zone and a primary stroke are connected, then there is an edge between their corresponding vertices. The authors use continuity analysis to determine whether 2 primary strokes joined with the same ambiguous zone can be merged or not. Continuity analysis is based on a Bayesian classifier, using features (such as angle deviation, width difference, curvature variation, etc.) extracted from 2 primary strokes joined with the same ambiguous zone, and based on an algorithm for searching all the simple paths in the graph satisfying a certain number of conditions: end constraint, non-end constraint, smoothest criterion, and Y-junction criterion [7]. Because of the use of a supervised classifier, this method requires a preliminary training step on manually annotated samples of the document collection, which is not possible in our case where we have no a priori information about the language/script.

In [4], the authors use a general decision function to determine whether 2 neighboring primary strokes have to be grouped or not, instead of using a supervised classifier. This method can be used in the general case. But, designing and parameterizing such a general decision function is hardly tractable.

In [5], the authors have a statistical analysis of the NIST database [15] and note that the percentage of ambiguous zones connected to more than 4 primary strokes is very low. Based on this analysis, the authors limit their algorithm to deal with ambiguous zones connected to less than 5 branches. Then, they can list all possible configurations (topologies) for an ambiguous zone. This method uses a priori knowledge for a specific language, so it cannot be used in our case.

B. Our proposed stroke extraction technique

Our objective is to merge the primary strokes extracted using the method in [7] so as to obtain our strokes. Similar to [7], we represent primary strokes and ambiguous zones using a graph. Each ambiguous zone and primary stroke corresponds to a vertex. If an ambiguous zone and a primary stroke are connected, then there is an edge between their corresponding vertices. But, we cannot rely on a supervised classifier to group primary strokes like in [7] because our goal tends to work with documents of different languages and with non-expert users, we prefer to use a decision function. This function is based on Gestalt parameters [17] that are involved in the human visual processing of writings. These parameters are: 1) the section of the stroke (strokes generally have sections of approximately the same width, especially in old documents) and 2) the good continuation rule (the curvature of the stroke cannot abruptly change). To determine if 2 primary strokes $S_i$ and $S_j$ joined by an ambiguous zone $S_a$ should be grouped or not, we use a process in 2 steps:

Firstly, we estimate the average widths $w_i$ and $w_j$ for each primary stroke. The measure $w_{ij} = |w_i - w_j|$ describes the width difference between $S_i$ and $S_j$. Based on the first Gestalt parameter, if $w_{ij}$ exceed a given threshold, the pair of primary strokes $(S_i, S_j)$ cannot be grouped and therefore belong to 2 different strokes.

Secondly, Sample Points of Writing Trajectories (SPWT) are extracted from each writing segment using the algorithm described in [4] (see Fig. 2.). Let us denote $S_{q_i} = (p_{i0}, p_{i1},...,p_{im})$ the sequences of a SPWT corresponding to a given primary stroke $S_i$, joined with an ambiguous zone $S_a$. We define the support chain $(p_{i0}, p_{i1},...,p_{ig_i})$, $g_i \geq 3$ as the subsequence of $S_{q_i}$ starting from $p_{i0}$ (the point being the nearest to $S_a$) for which the direction of the path from $p_{i0}$ to $p_{ig_i}$ remains stable on the whole path. More formally, let us denote $V_{g_i}$ the first principal component (i.e. the main direction) obtained by applying Principal Component Analysis on the set of points $(p_{i0}, p_{i1},...,p_{ig_i})$. We define $g_i$ as the index of the last point in the sequence so that the angle between $V_{g_{i-1}}$ and $V_{g_i}$ remains inferior to a predefined threshold $\phi_{thr}$.
For each ambiguous zone $S_a$, given 2 primary strokes $S_i$ and $S_j$ joined with $S_a$, we then define 2 variables: the angle deviation $\theta_{ij} = \text{angle}(V_{g_i}, V_{g_j})$ and the curvature deviation $\gamma_{ij}$ (calculated as in [7]). Based on the second Gestalt parameter, the primary strokes $(S_i, S_j)$ for which $\gamma_{ij}$ is minimum are grouped if and only if $\theta_{ij} < \theta$ (where $\theta$ is a fixed threshold). We performed experiments on the database presented in Fig. 1 and we found out that the values of the thresholds $\phi_{thr} = 0.3$ and $\theta = \pi/2$ gave satisfying results on all the languages. Of course, these threshold values could be adapted more precisely to the considered language, but then it would require a specific training stage for each language.

### IV. INVARIANT EXTRACTION

**A. Stroke clustering**

Once all strokes in a document collection have been extracted, we apply a clustering algorithm to group strokes into clusters of similar shape and select the cluster prototypes as invariants. Clustering is performed based on a description of the strokes relying on a set of well-chosen features.

**Stroke description:** The features used for stroke description should not be rotation invariant (otherwise, in Latin, the letter “n” could be clustered with the letter “u” for instance). Similarly and because of the traditional homogeneity of the ancient documents contents, the features should not be scale invariant (otherwise, the letters “i” and “l” could be mixed). We use the following set of features:

- **1) Elongation** [14]: Ratio of the height to the width of the shape’s minimal bounding box.
- **2) Solidity** [14]: Ratio of the area of the shape to the convex hull area of the shape. Solidity describes the extent to which the shape is convex or concave.
- **3) Rectangularity** [14]: Ratio of the area of the shape to the area of the minimum bounding rectangle. Rectangularity represents how rectangular a shape is.
- **4) Circularity ratio** [14]: Ratio of the area of the shape to the area of a circle having the same perimeter. Circularity ratio represents how a shape is similar to a circle.
- **5) Bounding box** [16]: Bounding box method approximates the shape of the stroke using a fixed number of rectangles of varying size and computes a vector of features representing the shape of the obtained lattice of rectangles. Normally, bounding box method uses a normalization algorithm to make the features invariant to rotation. In our case, where the features should not be rotation invariant, we do not apply this normalization.

**Invariant extraction:** First, a pre-clustering is performed by applying the BIRCH clustering algorithm [13] using the set of simple features: 1,2,3,4. We choose the BIRCH algorithm because it is fast, hierarchical and is proven to be efficient for image clustering [11]. We choose features 1, 2, 3, 4 for pre-clustering as they are simple features that we can use as filters. Second, for each pre-cluster, re-clustering is applied by using several sequential clustering phases [10] and the feature 5. Feature 5 is used in the second step only to refine clustering results obtained from the first step. We do not include it in the first step, as it is scale invariant and we do not wish to get scale-invariant clusters. The sequential clustering phase provides a variable number of clusters. Final clusters are defined as the groups of strokes that are always clustered together, enabling us to filter outliers.

For each stroke, we calculate its Silhouette Width (SW) score [12] which is an unsupervised measure for evaluating clustering results. SW measures the clusters compactness and separation. Then, for each cluster, we define the corresponding invariant as the stroke inside the cluster with the highest SW score.

**B. Cluster visualization**

After clustering strokes and finding invariants, we use a graphical user interface to visualize strokes and invariants, as shown in Figures 3 to 5.

![Fig. 3. Invariants of the Telugu database](image3)

![Fig. 4. Details of a cluster, showing a stroke in its context (in the original connected component)](image4)

![Fig. 5. Invariants of the Saint Gall database](image5)
All invariants (i.e.
cluster prototypes) are displayed in window 1, in a plane composed of the two first
principal components extracted from their feature vectors using Principal
Component Analysis (see Figures 3 and 5). When the user
moves his/her mouse over an invariant, a red circle is displayed
around the invariant, which indicates the radius of the
Corresponding cluster in the feature space. This prototype is
displayed at the center of window 2, and the closest strokes to
the invariant are displayed at the inner ring while the strokes
farthest to the invariant are displayed in the outer ring. If the
user double-clicks at an invariant, the window 1 displays all the
strokes of this cluster (see Figure 4).

This interface allows the user to visualize the invariants and
analyze the clusters. We are currently working on integrating
an interaction/personalization module that lets the user interact
with the system so as to create (based on strokes) the invariants
he’s more comfortable with for query composition.

V. CONCLUSION

In this paper, we introduce an invariant extraction method
adapted for omni-language word retrieval. There are 3 main
processes in this invariant extraction system: ambiguous zone
detection, primary stroke grouping (stroke extraction) and
invariant extraction (stroke prototype extraction).

We compare existing ambiguous zone detection methods on 3
data sets of different languages and find out that method [7] is
the most suitable in our context. This method gives good result
even on degraded documents and does not require any training
stage, and therefore can be used for any script.

We introduce a new primary stroke grouping method, using a
general decision function to determine whether 2 primary
strokes joined with the same ambiguous zone can be merged or
not, based on parameters that are involved in the human visual
processing of writings.

Then, an adapted clustering method is applied to find
invariants (i.e. stroke cluster prototypes) that can be further
used for the second step: query composition and the third step:
word retrieval. An interface is built, allowing the user to
visualize the invariants and the clusters.

Because it must be generic, our system cannot be
specifically adapted to a given language. If sometimes returns
invariants that are not meaningful enough to the user to
compose his/her query in the second step of query
composition, especially in the case where the script is highly
cursive. We are currently working on interactive primary
stroke grouping/splitting and interactive stroke clustering
interface, where the user can refine the results of the algorithm
according to his/her wishes.

ACKNOWLEDGMENT

We would like to thank Professor Alain Boucher for the
help he provided us, especially for the experimental
comparison of primary stroke extraction methods.

REFERENCES

193-221, April 2007
without layout analysis”, ICPR, volume 3, pp. 57-60, 2002
handwriting”, Proceedings of Computer Vision and Pattern Recognition
(CVPR), pp. 631-637, 1996
handwriting: an approach based on off-line recovery of the motor-
temporal information”, IEEE Transactions on Image Processing, pp. 80-
91, 1999
Recognition, pp. 1147-1160, 2000
writing sequence from offline handwritten chinese character image,
Proceedings of the International Congress on Image and Signal
Processing (CISP), pp. 298-302, 2008
on ambiguous zone detection: a preprocessing step to recover dynamic
information from handwritten Chinese characters”, IJDAR, pp. 109-121,
2009
restoration of writing order from single-stroked handwriting image”,
Pattern Analysis and Machine Intelligence (PAMI), pp. 1724-1737,
2006
[9] K. Liu, Y.S. Huang and J.H. Kim, “Model-based stroke extraction and
matching for handwritten chinese character recognition”, Pattern
Recognition, pp. 2339-2352, 2001
[10] A.K. Menahem Friedman, Introduction to pattern recognition: statistical,
structural, neural, and fuzzy logic approaches, World Scientific, 1999
comparison of clustering methods for content-based indexing of large
345-366, 2012
validation of cluster analysis”, J Comput Appl Math, vol 20, pp. 53-65,
1987
method for very large databases”, Proceedings of the ACM SIGMOD
International Conference on Management of Data, pp. 103-114, 1996
extraction techniques”, Pattern Recognition, pp. 43-90, 2008
Database, 1992
shape classification”, Proceedings of International Conference on Image
Processing, pp. 478-481, 2005
visual processing”, Computer Vision, Graphics and Image Processing,
vol 37, pp. 261-298, 1987
Manuscripts using Hidden Markov Models,” in Proc 1st Int. Workshop
on Historical Document Imaging and Processing (HIP), 2011