Low-level cursive word representation based on geometric decomposition

Jian-xiong Dong[1] and Adam Krzyżak[3] and Ching Y. Suen[2] and Ponson Dominique[1]

[1] IMDS Software
75 rue Queen, Suite 6200, Montréal, Québec, H3C 2N6
{jdong,dponson}@imds-world.com
[2] Center for Pattern Recognition and Machine Intelligence
Concordia University
Montréal, Québec, Canada H3G 1M8
suen@cenparmi.concordia.ca
[3] Department of Computer Science and Software Engineering
Concordia University
1455 de Maisonneuve Blvd. W.
Montréal, Québec, Canada H3G 1M8
krzyzak@cs.concordia.ca

Abstract. An efficient low-level word image representation plays a crucial role in general cursive word recognition. This paper proposes a novel representation scheme, where a word image can be represented as two sequences of feature vectors in two independent channels, which are extracted from vertical peak points on the upper external contour and at vertical minima on the lower external contour, respectively. A data-driven method based on support vector machine is applied to prune and group those extreme points. Our experimental results look promising and have indicated the potential of this low-level representation for complete cursive handwriting recognition.

1 Introduction

Although much progress has been made in off-line cursive word recognition over the past decade [1] [2], this task is still very challenging and the performance of the current recognition systems is far from that of human beings [3]. The solutions to several key problems related to handwritten word recognition remain unknown. One of the most important problems is the efficient low-level representation of a cursive word image for classification. Intuitively, although a handwritten word is concatenated by a small set of handwritten characters (52 characters in English) from left to right, its shape exhibits considerable variations and depends on the uncertainty of human writing. The boundaries between characters in a handwritten word are intrinsically ambiguous due to the overlapping and inter-connections. The changes in the appearance of a character usually depend on the shapes of neighboring characters (*coarticulation* *effects*). As a result, it is very difficult to represent the word image based on characters in early visual processing.

In the current literature several methods have been proposed to alleviate the character segmentation problem [4]. In the first case, the image of the given word is regarded as an entity in the whole. A word is characterized by a sequence of features such as length, loops, ascenders and descenders. No sub-models are used as the part of its classification strategy. Although the method can avoid the difficult problem of segmentation completely, no uniform framework in the literature has been presented to extract those features. It is not clear how to solve the problem of the correspondence of feature points if some features are used as local shape descriptors.

In the second case, the word image is segmented into a sequence of graphemes in left-to-right order [5]. The grapheme may be one character or parts of characters. After the segmentation, possible combinations of adjacent graphemes are fed into a character recognizer. Then a dynamic programming technique is used to choose the best sequence of characters. There are two problems with this method. One is that segmentation and grapheme combination are both based on heuristic rules that are derived by human intuition. They are error-prone. The other is that the computational cost is probibitively high due to the evaluation of a large grapheme combination.

In the third case, features are extracted in a left-to-right scan over the word by a sliding window [6]. In this method, no segmentation is required. But there are several problems related to it. One is that some topological information such as stroke continuity and contour length will be lost. But stroke continuity is a strong constraint for handwritten signals. The other is how to determine the optimal width of a sliding window. Morover, this one-dimensional sampling of two-dimensional word image will resut in information loss.

By reviewing the above methods, we know that none of them imply where the important information is located in a word image and how to organize them efficiently..In this paper, we locate certain extreme points in the vertical direction on a contour, then apply support vector machines to classify those points into two channels: local peaks in the upper external contour and local minima in the lower external contour. For classification task, local feature vectors are extracted at those points. As a result, a cursive word image will be represented by two sequences of feature vectors.

In Section 2, we discuss the relationship of feature points to the process of handwriting production and an algorithm for the extraction of those points will be given. Then we present the method of feature extraction in Section 3. The experimental results are described in Section 4. Finally, we summarize this paper and draw conclusions.

2 Locating extreme points

In the process of handwriting production, strokes are basic units and a handwriting signal can be represented as a sequence of strokes in the temporal dimension.

A stroke is bounded by two points with curvature. In offline handwriting, the image contour can be used to precisely represent a binary (black/white) word image. The high-curvature points can be detected robustly. As a result, the external contour can be broken up into strokes under the assumption of contiguity. In terms of an oscillatory motion model of handwriting [7], we know that the horizontal and vertical directions are more important than the other orientations. Then strokes are split into two groups: strokes in the upper contour and those in the lower contour. These strokes are ordered from left to right. This representation has several characteristics:

1. It is compatible with the study in psychology which shows that strokes are basic units of handwriting signals.
2. The space neighboring relationship of strokes is preserved.
3. It is a local representation of word image. It is easier to extract low-level invariant local features.
4. It is a 2D representation.

Also, unlike wavelet coding, more high-level units such as letters and words can be visually constructed from this representation. As a result, this representation will facilitate us in building a hierarchical system of cursive word recognition.

In order to obtain the above representation, we first need to locate those interesting points with high curvature. In the writing process, the most important part of a curve seems to be where the writing speed has reached a minimum and curvature reaches a maximum [8][9]. The vertical extrema on an external contour are the segmentation points of handwritten strokes. Fig. 1 shows the procedures of the extraction of vertical extrema. The interesting locations are the peaks in the upper external contour and minima in the lower external contour. The neighboring curved segments at those extrema are convex. If the curve are smooth, the curvatures at those points are positive. Theoretically there exists a point with the negative curvature between two neighboring extrema. This indicates that a point with the negative curvature depends on its neighboring points with positive curvature. Therefore, it is reasonable to assume that peaks in the upper external contours are pairwisely independent. So are the minima in the lower external contours. Also, these locations are analogous to the centers of receptive fields [10], which attract visual attention [11]. Separating these extrema into two groups have the following advantages:

- 2D space configuration of these extrema can be approximated by two 1D space configurations. Consequently the problem complexity will be greatly reduced.
- It conforms with the Gestalt principles of perceptual organization: proximity principle (vertical distances) and similarity (local curve shape at these points).
- When we model the signal similarity independently and signals in one group are degraded, the model in the other group is not affected.

In addition, for the inner contour, we represent it as a loop in the stage of feature extraction, rather than as vertical extrema. The loop will be associated with the

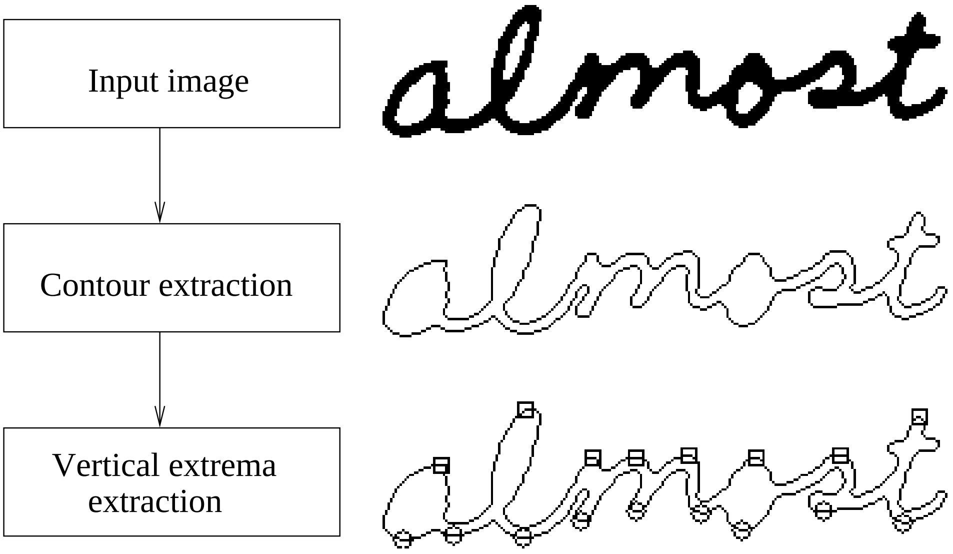


Fig. 1. Vertical extrema on the external contour of a word image. The peaks in the upper contour are marked by rectangles. The minima in the lower contour are marked by circles.

closest extrema in the external contour. In the next stage, local features are extracted at those extrema.

Vertical extrema can be detected on the image external contour. We propose an algorithm to detect those points robustly as below:

```
Algorithm for detection of vertical extrema
Input: contour points $\mathbf{v}[i]$, $i = 1, \ldots, n$ and working space $\mathbf{v}_2$ and $\mathbf{v}_3$.
Output: peaks and minima
1 Identify the index set $B = \{i | \mathbf{v}[i].y \neq \mathbf{v}[i+1].y\}$.
2 Copy elements in set $B$ into the vector $\mathbf{v}_2$ whose elements are sorted in
    an increasing order. The length of vector is $K = |B|$.
3 Calculate the difference of y-coordinate of two neighboring points in $\mathbf{v}_2$.
    $\mathbf{v}_3[k] \leftarrow \mathrm{sign}(\mathbf{v}[\mathbf{v}_2[k]].y - \mathbf{v}[\mathbf{v}_2[k]+1].y)$, $k = 1, \ldots, K$.
4 Median filtering of window size 3 is applied to $\mathbf{v}_3$.
5 Select the candidates of extrema from the two indexed vectors:
    peak$[i] \leftarrow \mathbf{v}_2[k]$ if $\mathbf{v}_3[k] < 0$; minima$[j] \leftarrow \mathbf{v}_2[k]$ if $\mathbf{v}_3[k] > 0$. $P$ and $M$
    denote the number of peaks and number of minima, respectively.
6 Prune invalid minima iteratively.
7 Prune invalid peaks iteratively.
```

In the above algorithm, three primary measures are used to prune invalid peaks: contour length, height difference and bounded variation between two neighboring peaks. For example, if the contour length between two neighboring peaks is small, they will be merged into one peak that will be located at the middle point on the contour. Also, if a local minimum point is located in the upper zone, it will be pruned.

3 Features

Although the algorithm in Section 2 can be applied to group peaks and minima and prune invalid extrema, it is still not good enough due to various variations of word shapes. Therefore, we need to introduce a classifier to refine the grouping and pruning process. Several features have to be extracted at each extreme point. We describe these features as follows:

1. Number of local minima on the current contour (f_1). In Fig. 2, the feature values at points 1 and 2 are 2.

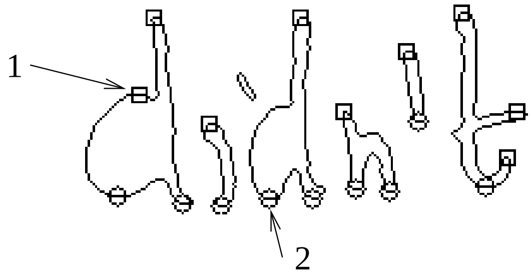


Fig. 2. Illustration for the extraction of number of peaks and minima

2. Number of local peaks on the current contour (f_2). In Fig. 2, the feature values at points 1 and 2 are 2 and 1, respectively.
3. Minimum height difference with neighboring extrema (f_3). When the current point is a local minimum, two neighbors are local minima; When the current point is local peak, two neighbors are local peaks. Neighbors may not be on the same contour as the current one. In Fig. 3, the feature value at point 2 is $\min(|y_1 - y_2|, |y_3 - y_2|)$.
4. Minimum height difference with neighboring extrema (f_4). When the current point is a local minimum, two neighbors are local peaks; when the current point is local peak, two neighbors are local minima; Neighbors may not be on the same contour as the current one. In Fig. 4, two neighbors of point 1 are 2 and 3. The feature value at point 1 is $\min(|y_1 - y_2|, |y_3 - y_1|)$
5. Percent of peaks above the current point (f_5).
6. Percent of minima below the current point (f_6).
7. Relative position in y axis (f_7). The feature is defined by $\frac{y}{h}$, where y is the coordinate of the current point in y-axis and h is the image height.
8. Minimum contour length with neighboring peaks (f_8). Note that neighboring peaks are on the same contour as the current point.

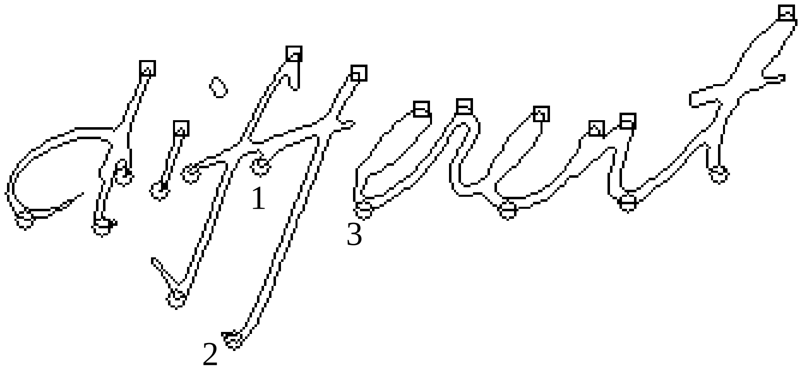


Fig. 3. Illustration for the extraction of minimum height difference. The neighbors have the same convex attributes as the current extreme point.

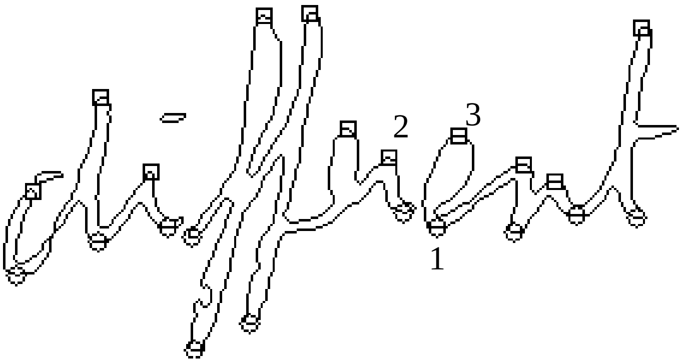


Fig. 4. Illustration for the extraction of minimum height difference. The neighbors have different convex attribute from the current extreme point.

9. Minimum contour length with neighboring minima (f_9). Note that neighboring minima are on the same contour as the current point.
10. Minimum height difference with neighboring peaks (f_{10}). The neighboring peaks must be on the same contour as the current point.
11. Minimum height difference with neighboring minima (f_{11}). The neighboring minima must be on the same contour as the current point.

In the above features, f_1 and f_2 characterizes the information of word length. f_3 represents the information of ascender and descender. For each feature value, a reasonable upper bound will be set. If the feature value is greater than the corresponding bound, it will be rounded to this bound. In biological learning, it is called "peak effect" [12]. Given that the specified bound b_i of the feature f_i, the round operation is given by

$$f_i' = \min(f_i, b_i) \tag{1}$$

For fast learning, the feature values will be first transformed into the interval $[0, 1]$, the variable transformation $x^{0.4}$ is applied to each component of the feature vector such that the distribution of each feature is Gaussian-like [13]. The formula is given by

$$f_i'' = (\frac{f_i'}{b_i})^{0.4} \tag{2}$$

Then a support vector classifier [14] is constructed to classify points to two classes: extrema on the upper contour and extrema on the lower contour. If one local minimum is classified to the upper contour, it will be pruned. If one local peak is classified to the lower contour, it will be pruned. As a result, the valid local peaks will be on the upper contour while the valid local minima will be on the lower contour.

4 Experiments and results

The word representation method described in this paper has been implemented in C++ on Windows XP and compiled by Microsoft visual C++6.0. It is a part of IMDS handwritten word recognition engine. The experiments were conducted on IMDS cursive word database. At IMDS Software we designed a specified electronic form to collect isolated handwritten words such that labelling can be done automatically. Presently the vocabulary are the words from the category of Collins Frequency Band 5, in which these words are most frequently used in daily life. The size of this lexicon is 670. Our samples are written by a variety of 78 persons from different countries such as Arabian, Asian, Canadian, French., from students and professors at universities, employees in companies. No constraints are imposed on the writers in order to get most natural handwritten samples. Each writer writes samples in blank boxes in the form, which contains 670 words. This indicates there are no two samples from the same writer for each word. The form is scanned as a gray-scale image in 300 DPI and is binarized. The samples are randomly split into training and testing sets whose sizes are 38795 and 13733, respectively. Some samples are depicted in Fig. 5.

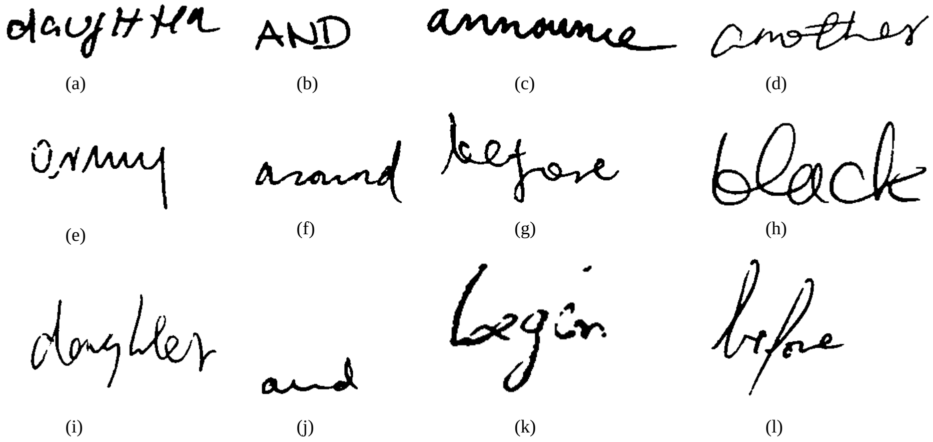
Support vector machine (SVM) is used as a classifier to classify extrema into two categories. The radial basis function (RBF) is chosen as the SVM kernel, given by

$$K(x, x') = \exp(-\frac{\| x - x' \|^2}{0.6^2}) \tag{3}$$

where $\| . \|$ denotes Euclidean norm and the dimension of feature vectors x and x' is 11. The upper bounds b_i, $i = 1, \ldots, 11$ of eleven features in Section 3 are shown in Table 1. The value of C in the dual form of SVM [14] is set to 10.0.

Table 1. Upper bounds b_i, $i = 1, \ldots, 11$.

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}
13.0	13.0	70.0	70.0	1.0	1.0	1.0	120.0	120	70.0	70.0


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 5. Some samples in IMDS cursive word database: (a) “daughter”, (b) “and”, (c) “announce”, (d) “another”, (e) “army”, (f) “around”, (g) “before”, (h) “black”, (i) “daughter”, (j) “and”, (k) “begin”, and (l) “before”.

SVM is trained by HeroSvm2 [1]. Some labelled samples have to be collected before we train support vector machine. Our strategy is first to label a small number of extrema in the word images manually. Then these samples are divided into training and testing sets. A SVM classifier is constructed. The classifier is used to label other extrema. The misclassified errors are corrected manually. Since the recognition rate of SVM classifier is very high, more than 99%, the number of manual corrections is small. Much time-consuming cost has been be saved. Table 2 shows the performance of SVM classifier, where SV and BSV denote the number of support vectors and number of bounded support vectors, respectively. The number of support vectors is small, compared with the size of the whole training set. It may indicate that features are discriminative so that a small portion of SVs can characterize the classification boundary. The above results look very promising. They indicate that the extrema can be grouped into two categories with a high accuracy though cursive word shapes exhibit considerable variations. It also infers that in low-level visual processing the data-driven learning technique with top-down information can eliminate the uncertainty of a decision to a great extent. Traditionally, the baseline information may be used

Table 2. Performance of support vector machine

Training set	Testing set	Training rate	Testing rate	SV	BSV
354,928	119,825	99.78%	99.6%	4377	2936

[1] http://www.cenparmi.concordia.ca/ people/jdong/HeroSvm.html.

to determine the upper and lower zones. But the detection of baseline is not robust due to uneven writing. Moreover, it is difficult to find a baseline for some short words. One of the appealing properties of the proposed method is that the output of SVM classifier can be used as the confidence value. When the absolute value of SVM's output is larger, the decision of the classifier becomes more reliable. Some examples are shown in Fig. 6. It can be observed that the proposed method is insensitive to the word length and image size scale.

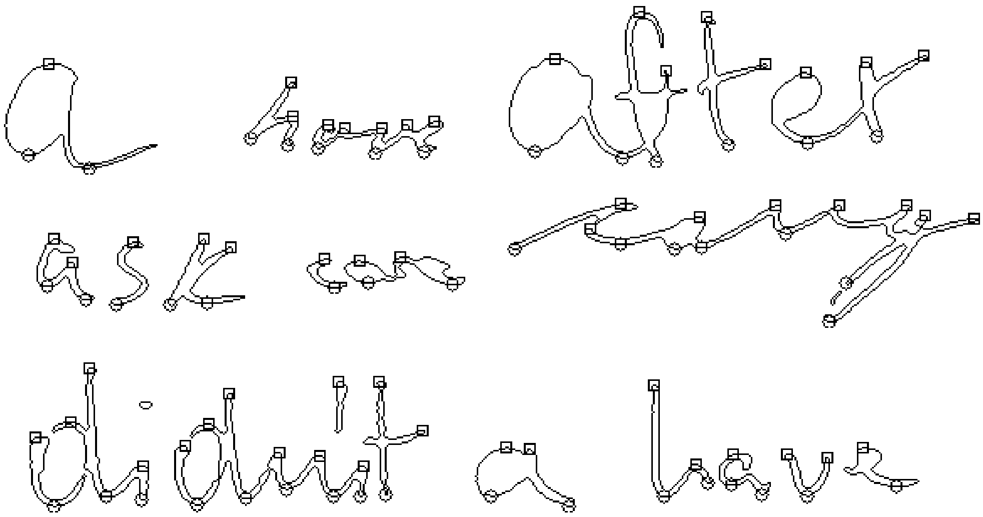


Fig. 6. Some word images where extrema are classified to upper peaks and lower minima. Peaks and minima are marked by squares and circles, respectively.

5 Conclusions

In this paper, we present an efficient low-level representation of cursive word images for classification task, which is an important step to build a general hierarchical word recognition system. A word image can be represented as two sequences of feature vectors which are extracted at vertical peak points on the upper contour and at vertical minima on the lower contour. Some evidences from the process of handwriting production and visual perception are linked to this representation. The experimental results look promising and have shown the potential of this representation for cursive word recognition task, which is our primary goal.

Acknowledgments

This research is support by an Industrial Research Fellowship awarded by Natural Sciences and Engineering Research Council of Canada and IMDS Software.

References

1. Steinherz, T., Rivlin, E., Intrator, N.: Offline cursive script word recognition – a survey. International Journal on Document Analysis and Recognition **2** (1999) 90–110
2. Koerich, A., Sabourin, R., Suen, C.: Large vocabulary off-line handwriting recognition: A survey. Pattern Analyis and Applications **6** (2003) 97–121
3. Bunke, H.: Recognition of cursive roman handwriting – past, present and future. In: Proceedings of IEEE 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland (2003) 448–459
4. Madhvanath, S., Govindaraju, V.: The role of holistic paradigms in handwritten word recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 149–164
5. Knerr, S., Augustin, E., Baret, O., Price, D.: Hidden Markov Model based word recognition and its application to legal amount reading on French checks. Computer Vision and Image Processing **70** (1998) 404–419
6. Mohamed, M., Gader, P.: Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 548–554
7. Hollerbach, J.: An oscillation theory of handwriting. Biological Cybernetics **39** (1981) 139–156
8. Attneave, F.: Some informational aspects of visual perception. Psychological Review **61** (1954) 183–193
9. Schomaker, L., Segers, E.: Finding features used in the human reading of cursive handwriting. International Journal on Document Analysis and Recognition **2** (1999) 13–18
10. Hartline, H.: The response of single optic nerve fibres of the vertebrate eye to illumination of the retina. American Journal of Physiology **121** (1938) 400–415
11. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1254–1259
12. Staddon, J.: Adaptive Behavior and Learning. Cambridge University Press, Cambridge (1983)
13. Fukunaga, K.: Introduction to Statistical Pattern Recognition. second edn. Academic Press (1990)
14. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)