Core Points - a framework for structural parameterization

Abstract

For structural handwriting recognition methods the most common method of discrete parameterization for samples of on-line characters have been to place a fixed number of points on the curves, spaced according to the total arclength of the sample. A problem for this approach is that the points on the reparameterized curves of different samples from a character class may then actually correspond to parts of different structural significance. Many methods such as DTW and HMM have been successful partly because they are less sensitive to parameterizational differences. However, obtaining a structural correspondance between points in samples is greatly beneficial for other recognition methods. It has also been shown that the structural parameterization presented here manages to remove redundance harmful to the recognition task. A framework for obtaining structural correspondances between samples have been implemented in a recognition toolbox-like environment such that various recognition methods and datasets could be evaluated under identical conditions. Results are evaluated both in qualitative form as figures and quantatively as hitrates when incorporated into various recognition systems.

1. Introduction

Most of the state-of-the-art recognition methods today have abandoned a structural approach to the single character recognition problem. A main reason for this are the difficulties in automatically generating reliable structural models from data displaying the vast structural variations as that of handwritten data [1]. Of the remaining template matching systems most have adopted strategies, such as DTW or HMM, that are less sensitive to differences in structural function of corresponding sample points [2, 3].

In this paper we present a method to parameterize samples of handwritten characters according to their structural content. We accomplish this by extracting a rigid frame of points that we call *core points*. Previous research show how the dimensionality of a curve can be reduced by encoding structural content (i.e curvature) [4]. Utilizing this information to segment cursive script was a hot topic in the early nineties and has recently gotten new attention [5, 6].

We will start by recapitulating some facts about handwriting that although obvious are often neglected when discussing the performance of handwriting recognition systems. We will then show how we can anchor our ideas of structural parameterization on a new formalism in these facts. We will conclude the paper by presenting qualitative and quantitative results of recognizers implementing the new parameterization technique.

2. Core Points

The main obstacle preventing formal definitions concerning shapes in the field of HWR is the interaction of a human being with a free mind. In other words the difficulty lies in that each symbol actually bears the meaning that the writer *intended*. A fact well-known in the HWR community is that the writer's intention is not always interpretable since he migh actually have drawn a shape that bears more resemblance to another symbol. Although intricately intertwined with the dataset used in the experiment, there are results indicating that humans have a 96% hitrate when reading context-free handwritten characters in a cursive word [7]. Misinterpretation of an isolated handwritten symbol by a human reader can occur for two reasons. Either the symbol portrayed is itself ambiguous or the deformed shape of a handwritten sample of the symbol actually resembles some reference shape of another symbol. A natural question is then - What are these shapes that we use for reference? They are the set of shapes that each human being with knowledge of the particular script has been taught to interpret. We will here refer to these shapes as *ideal* and we will define them through their characteristic properties. In the field of HWR it is common to refer to different shapes of the same symbol as allographs so we will use this terminology from now on.

Definition 1 *Let a* ***script*** *be defined as a set of ideal allographs representing symbols. Any two* ***ideal*** *allographs representing different symbols of a script has a distinct difference in shape and/or size. We define* ***knowledge of a script*** *as the ability to discriminate between these ideal allographs* *without failure. We call ideal allographs that have a unique symbol reference* ***distinct*** *and allographs with more than one symbolic meaning for* ***contextual*** *allographs.*

With the terminology of Parizeau et al. ([7]) the isolated single character recognition problem is a strict morphological discrimination problem and hence it is only meaningful to present recognition results for handwritten samples of a script that someone with knowledge of the script would be able to pair with a morphologically distinct allograph. In fact it is natural to suspect that any system that manages to correctly classify samples of a symbol that are morphologically more similar to an allograph of a different symbol, is overtrained on that particular dataset.

2.1. Definition

One may view the handwriting process of a symbol of a script is an attempt at portraying the shape of one of the distinct ideal allographs of the script. We will refer to the ideal allograph aimed at by the writer as the **aim** of a sample.

Definition 2 *We can analytically define a set of ideal allographs as defined in Definition 1 as a set of polynomial paths of at most second degree along with an optional requirements on each segment. We call these segments the* ***ideal segments*** *of an ideal allograph.*

A requirement on a segment could for instance be a non-zero curvature requirement. With this terminology fixed we can move on to define what we mean by our ideal set of core points.

Definition 3 *With* ***core points*** *of an ideal allograph we mean the points required to connect all the ideal segments as well as an extra point in the middle of each ideal segment that has a non-zero curvature requirement.*

Albeit their hypothetical nature these definitions provide an intuitive approach to structural parameterization of handwritten character samples. The first step is to realize that for recognition purposes we are only interested in discrete (i.e sampled) handwritten characters that contain the core points of their respective aims. Any other samples are simply impossible to recognize out of context, even for humans.

2.2. Extraction and Structural Reparameterization

To practically extract the points corresponding to the ideal core points of the aim from a sample without knowing the aim is impossible. There are, however, several approaches to extract a reduced set of points that are probable to contain the set of points corresponding to the ideal core points. Some methods for extracting key points are provided in papers [5, 6]. We have implemented a most simple method to extract a structurally reparmeterized set that we call **potential** core points.

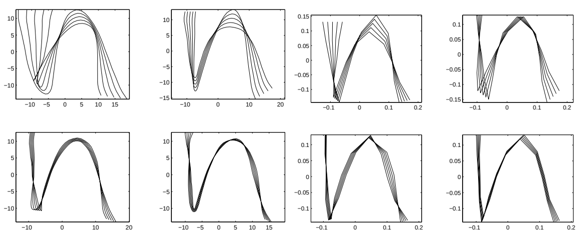

Figure 1. The four first components of principal component analysis of one allograph of *n*, automatically extracted with the core point scheme of Section 3, with and without core point reparameterization. The four figures on the right are the core point reparameterized versions.

We assume that samples have been centered and normalized so that the standard deviation is one and that they have been rotationally aligned so that the x-axis is aligned with the writing direction. In this coordinate system we extract a coarse frame as the extreme points in y that have a local depth exceeding a threshold value, we call these points N, S analogously to their respective bearing on a map. We then use this frame, that we call the **core point frame** as our basis for the structural reparameterization. However, since each segment in this frame may actually correspond to a segment containing a core point, we try to parameterize each segment without missing these. To accomplish this we add a fixed number of points to each segment, but instead of just spacing the points evenly on the segment as we would if we were to use conventional arclength parameterization, we first try to find all points that have a significant curvature. This search is done recursively by picking any point with a diversion from the line between the start and end point that exceeds a threshold. If the number of curvature points chosen in this manner is less than our fixed value of points per segment we add points and try to space them as evenly as possible. The first modes of some samples with a common aim that have been structurally reparameterized in this way are shown in Figure 1. It is clear how our reparameterization has managed to model the movement of the core points independently from the statistical variations of the curvatures on each segment. Figure 1 shows the first modes of a shape model as obtained by a singular value decomposition of all the samples and one can see that the model is significantly more accurate. Verification of this has also been performed quantatively by implementing a recognizer based on Gaussian Active Shape models as can be seen in Table 1.

Method	Original data	CPP-MC	CPP-BLS
1-nn-Euclidean	86.3%	89.2%	83.9%
3-nn-Euclidean	86.5%	**89.4**%	83.9%
AS	77.2 %	82.4 %	-

Table 1. Recognition results on the MIT database.

2.3. Normalization and matching of core point models

Since points in a core point parameterization are unevenly spaced in space, we also investigated an alternative normalization method for the core points. Instead of moving samples to the center of mass we tried to use a Bookstein approach by placing all samples on the consistently longest N-S segment observed in a training set [8]. We call this normalization procedure Bookstein Longest Segment (BLS). Recognition results of kNN classifiers with a Euclidean metric was not improved by this method as seen in Table 1. For very simple models with only one or two segments, however, the Bookstein normalization proved to be better for alignment. To utilize this for more complex models we need some way to make Bookstein coordinates on the individual segments. One way to do this is by calculating the thin plate splines between the core point frames of different samples while leaving the middle parameterization fixed. As can be seen in Figure 2 the results were promising. It seems as if it would be possible to divide the recognition process into two tasks, calculation of the bending energy of the frame, followed by normal pattern matching techniques for the points in the bent frame. It is however likely that there are more suitable ways of modelling the bending energy of such a frame than to model it as a thin plate, although there have been successful applications of this energy model to character recognition in the past [9].

3. Allograph Separation

Another important topic for building handwriting recognition methods is the separation and classification of different allographs. One reason is that many statistically based methods have very poor performance if they are required to build one model of input data of distinctly different shapes. For optimal performance of a handwriting recognition system one may also need to consider that many samples of handwritten data may actually be contextual allographs. A common example of this is that many writers make no clear distinction between the letters *n*, *a*, and *u* when writing. Consequently one needs some way of asserting that models are clearly separated. Support Vector Machines with some slack may be used to obtain a good border estimate between two non-separable classes [10]. There are also several prototype editing strategies for kNN classifiers. One example is the CNN rule which is the kNN operating on a consistent subset of the training samples obtained by some clustering technique [3]. However such methods can only rely on the statistical distribution of the samples in the training set and actually has no knowledge of the true ideal allographs that were intended. If one wants a classifier that performs well in a context-free environment it is important that the samples with an aim belonging to a contextual allograph are disregarded. This is a fact that is often neglected when evaluating handwriting recognition methods. To obtain good recognition results without adaptation there will be requirements on the writer. In a sense this is the reverse strategy of adaptation in that such methods needs to get the user to adapt to the system. We will here show how our structural parameterization technique can be used for semi-automatic allograph separation and clustering.

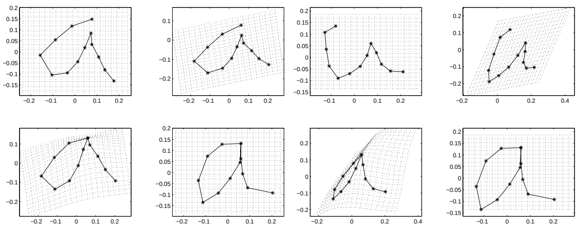

Figure 2. Bending the core point frame while leaving the parameterization fixed. To the four figures to the left is an in-class example of bending one example of an *a* to another. The right four figures shows the action of bending a sample of a *u* to an *a*. The four figures in both cases display the original sample, the affine approximation to the target sample, the complete thin plate spline of the core point frame and finally the target sample.

We obtain a label sequence of our potential core points by further differentiating the label of the extremum by adding information of the sign of the curvature. We obtain a first allograph separation by separating according to these labels (which also differ in length). The extraction of stable subsets of the allographs to include in a database can be conducted by a voting scheme such as the one described in paper [3]. It is however likely that manual intervention could provide even better results. Especially this is true for training sets containing contextual allographs.

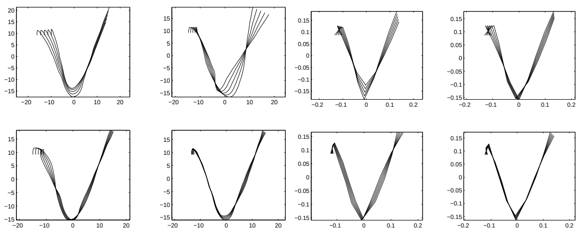

Figure 3. The four first components of principal component analysis of one allograph of *v*, automatically extracted with the core point scheme in Section 3, with and without core point reparameterization. The four figures on the right are the core point reparameterized versions.

4. Experiments

In addition to the qualitative validation of the structural reparameterization technique as shown in Figure 1, 2, 3, we have also conducted recognition experiments on the MIT single character database [11]. We chose the set of 37161 samples from the w section (single characters from words) as the test set and the 2825 samples from the l section as the training set. The results are shown in Table 1. We varied the settings for kNN with simple Euclidean versus DTW metric, mass center (MC) versus BLS (Bookstein of Longest Segment) normalization. We also tried the effect of the reparameterizations and normalizations on a gaussian Active Shape model (AS).

The most interesting part of the results is that we achieve a higher recognition rate even with a kNN recognizer when we structurally reparameterize the characters. In other words we have shown that our reduced set of structurally parameterized points actually increases the discrimination capability. Our core point reparameterization in other words manages to remove harmful redundance in an effective way.

5. Discussion and Conclusions

In this paper we have presented a method to achieve structural parameterizations for characters. It has been seen that a structural parameterization increases the discriminatory power of a character recognition system. What is especially interesting is that our method actually proves to increase recognition rate even though the dimension of the point configurations in the samples is reduced. We have thus proved that our structural reparameterization is superior to arclength parameterization - it requires the storage of less data at the same time that the discriminatory power of the reduced dataset is increased. Some methods such as Active Shape, which is commonly used in many pattern recognition applications, assumes that data points correspond. With the method presented in this paper we have shown that structural correspondence as the one achieved with the method in this paper enables usage of Active Shape for character recognition. We have also indicated that our definition of core point frame could enable a new way of dealing with the recognition problem. Any deformation of a character can be viewed upon as a combination of a deformation of its core point frame along with curvature differences on each segment. This way of viewing the recognition problem will be further pursued in future research.

References

[1] R. Plamondon and S. Srihari, “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey”, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 22, n. 1, pp. 63–84, January 2000.

[2] J. Hu, S.G. Lim and Michael K. Brown, “Writer independent on-line handwriting recognition using an HMM approach”, *Pattern Recognition*, , n. 33, pp. 133–147, 2000.

[3] V. Vuori, J. Laaksonen, E. Oja and J. Kangas, “On-line Adaptation in Recognition of Handwritten Alphanumeric Characters”, in *ICDAR ’99, Proc. of the 5th International Conference on Document Analysis and Recognition*, pp. 792–795, September 1999.

[4] S. Marshall, “Review of shape coding techniques”, *Image and Vision Comp.*, vol. 7, n. 4, pp. 281–294, 1989.

[5] Marc Parizeau and Re’jean Plamondon, “A Handwriting Model for Syntactic Recognition of Cursive Script”, in *Proc. of the 11th International Conference on Pattern Recognition (ICPR)*, volume II, pp. 308–312, August 31 to September 3 1992.

[6] Claudio De Stefano, Marco Garutto and Angelo Marcelli, “A Saliency-Based Multiscale Method for On-Line Cursive Handwriting Shape Description”, in *Ninth International Workshop on Frontiers in Handwriting Recognition Proceedings*, pp. 124–129, 2004.

[7] M. Parizeau and R. Plamondon, “Machine vs Humans in a Cursive Script Reading Experiment without Linguistic Knowledge”, in *Proc. International Conference on Pattern Recognition, Jerusalem, Israel*, pp. 93–98, 1994.

[8] F. L. Bookstein, “Size and Shape Spaces for Landmark Data in Two Dimensions”, *Statistical Science*, vol. 1, n. 2, pp. 181–242, 1986.

[9] S. Belongie, J. Malik and J. Puzicha, “Shape Matching and Object Recognition Using Shape Contexts”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, n. 24, pp. 509–522, 2002.

[10] B. Haasdonk C. Bahlmann and H. Burkhardt, “On-line Handwriting Recognition with Support Vector Machines - A Kernel Approach”, in *IWFHR VIII, 8th International Workshop on Frontiers in Handwriting Recognition*, pp. 49–54, August 2002.

[11] Rob Kassel, “The MIT on-line character database”, ftp://lightning.lcs.mit.edu/pub/handwriting/mit.tar.Z.