CHARACTER PROTOTYPE SELECTION FOR HANDWRITING RECOGNITION IN HISTORICAL DOCUMENTS

Andreas Fischer and Horst Bunke

Institute of Computer Science and Applied Mathematics
University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
{afischer,bunke}@iam.unibe.ch

ABSTRACT

Handwriting recognition in historical documents is vital for making scanned manuscript images amenable to searching and browsing in digital libraries. A valuable source of information is given by the basic character shapes that vary greatly for different manuscripts. Typically, character prototype images are extracted manually for bootstrapping a recognition system. This process, however, is time-consuming and the resulting prototypes may not cover all writing styles. In this paper, we propose an automatic character prototype selection method based on a forced alignment using Hidden Markov Models (HMM) and graph matching. Besides the predominant character shape given by the median or center graph, structurally different additional prototypes are retrieved with spanning and k-centers prototype selection. On the historical Parzival data set, it is demonstrated that the proposed automatic selection outperforms a manual selection for handwriting recognition with graph similarity features.

1. INTRODUCTION

In the context of cultural heritage preservation, the interest in handwriting recognition for historical documents has grown strongly in recent years [1]. Worldwide, there is a huge repository of scanned or photographed valuable old documents including, e.g., Old Greek manuscripts from Early Christianity, Old German manuscripts from the Middle Ages, and important handwritings from the Modern Ages, such as George Washington's papers at the Library of Congress. In order to make the manuscript images amenable to searching and browsing in digital libraries, i.e., to make them available to a broad readership, automatic handwriting recognition is needed in order to have access to the content of the images [2].

Handwriting recognition in historical documents is an off-line task that is based on the manuscript images only. This task is considered to be harder than on-line recognition, where temporal information is available about the writing process by using special input devices [3]. For large vocabularies underlying natural language, the accuracy of an automatic transcription is far from being perfect [4]. Additional difficulties arise for historical documents by the fact that the image quality, and thus the appearance of the handwriting, are heavily affected by the decay of paper or parchment over time. Furthermore, unlike modern English scripts, the language and basic character shapes vary greatly for different historical manuscripts. Hence, it is often necessary to train a recognition system specifically for a single manuscript. [1]

[1] Manuscripts may consist of one to several hundred pages, often with multiple columns.

In order to bootstrap a new system for the recognition of a given historical handwritten manuscript, the basic character shapes provide information of great value. In the recent literature they were used to make Latin manuscripts searchable by means of a small number of character prototypes and generalized Hidden Markov Models (gHMMs) [5]. A similar approach has been adopted for Arabic scripts in [6]. In [7], character prototypes have been used for template-free word spotting in low-quality medical forms. Recently, the use of character prototypes for HMM-based single word recognition with graph similarity features in historical manuscripts has been proposed in [8].

A drawback of the aforementioned works based on character prototypes is that the character images are selected manually in a first step. This manual selection is a time-consuming process and it is not guaranteed that all variants of the character shapes can be captured.

In this paper, we present an automatic solution to character prototype selection for handwriting recognition in historical documents. Based on forced alignment using analytical features and HMMs, word images are segmented into character prototype candidates. By means of a graph-based representation and graph edit distance, four prototype selection methods are presented, namely median, center, spanning, and k-centers selection. In an experimental evaluation, the selected character prototypes are used for a single word recognition task on the historical Parzival data set in conjunction with graph similarity features and HMM-based recognition. It is demonstrated that the proposed automatic selection is able to outperform traditional manual selection significantly.

The remainder of this paper is organized as follows. First, character image extraction by means of HMM-based forced alignment is discussed in Section 2. Next, Section 3 introduces the graph-based prototype selection methods. The graph similarity features are then presented in Section 4 and the experimental evaluation is discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. HMM-BASED FORCED ALIGNMENT

The goal of HMM-based forced alignment is to segment the training set, consisting of word images and their correct transcription, into individual characters. In case of cursively written text with touching characters, no perfect segmentation can be achieved, in general, because even for humans, the character boundaries are ambiguous. However, the resulting character images are expected to contain large parts of the character shape that can be used as a valuable source of information for handwriting recognition.

In the following, the different processing stages of

HMM-based forced alignment are described. First, the raw word images are normalized in a preprocessing step. Next, a feature vector sequence is extracted using a sliding window taking into account analytical features based on the text foreground. Finally, character HMMs are trained and used in the so-called forced alignment mode, taking into account the correct transcription of the training samples, in order to extract character images from the words.

2.1 Preprocessing

Word image preprocessing consists of binarization of the word images and normalization with respect to the handwriting orientation and size that is applied in order to cope with different writing styles. For binarization, a Difference of Gaussian (DoG) edge detection is used to locally enhance the text foreground, followed by global luminosity thresholding.

For normalization, the skew, i.e., the inclination of the text, is corrected, vertical scaling is applied with respect to the upper and lower baseline, and a horizontal scaling operation is performed using the mean distance of black-white transitions.

For more details on image preprocessing, we refer to [9]. Note that in this work, no errors are taken into account that stem from extracting the word images from the document page, i.e., we consider a perfect, manually corrected word segmentation.

2.2 Feature Extraction

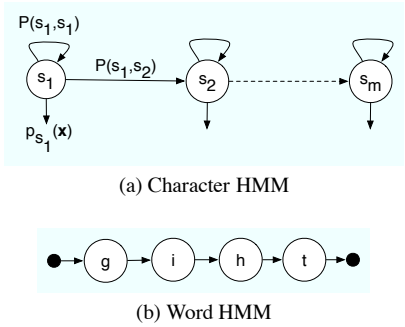
For HMM-based recognition, the two-dimensional information of the normalized binary images needs to be transformed into a one-dimensional signal. Due to the difficulties in reconstructing the original handwriting process from text images, a commonly used workaround is employed by means of a sliding window.

A sequence $\mathbf{x} = x_1, \ldots, x_T$ of feature vectors with $x_i \in \mathbb{R}^n$ is extracted by moving an analysis window with a width of one pixel from left to right over the word image. At each of the T positions of the sliding window, $n = 9$ analytical features are extracted from the foreground pixels. Three global features capture the fraction of black pixels, the center of gravity, and the second order moment. The remaining six local features consist of the position of the upper and lower contour, the gradient of the upper and lower contour, the number of black-white transitions, and the fraction of black pixels between the contours. For a more detailed description of the features, we refer to [10].

2.3 Hidden Markov Models

The basic modeling unit of the handwritten text is given by character HMMs shown in Figure 1a. Each character model has a certain number m of hidden states $s_1, \ldots, s_m$ arranged in a linear topology. The states s_i emit observable feature vectors $x \in \mathbb{R}^n$ with output probability distributions $p_{s_i}(x)$ given by a Gaussian Mixture Model (GMM). Starting from the first state s_1, the model either rests in a state or changes to the next state with transition probabilities $P(s_i, s_i)$ and $P(s_i, s_{i+1})$, respectively, thus taking into account variable character lengths.

The character models are trained using labeled word images. First, a word model is created as a sequence of character models according to the transcription as shown in Figure 1b for the transcription "giht". Then, the probability of this word model to emit the observed feature vector sequence $\mathbf{x} = x_1, \ldots, x_T$ is maximized by iteratively adapting the initial output probability distributions $p_{s_i}(x)$ and the transition probabilities $P(s_i, s_i)$ and $P(s_i, s_{i+1})$ with the Baum-Welch algorithm [11].



(a) Character HMM

(b) Word HMM

Figure 1: Hidden Markov Models

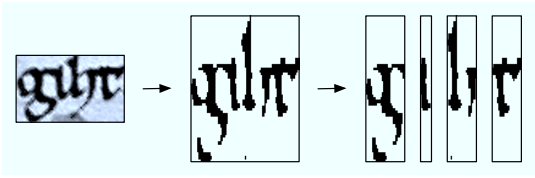


Figure 2: Character Extraction

Important parameters of the character HMMs that need to be optimized on a validation set are the number of states m for the individual characters and the number of Gaussian mixtures G used for the emission GMM.

2.4 Forced Alignment

Using the same word model as for training (see Figure 1b), the optimal likelihood $P(\mathbf{x}|\mathbf{c})$ of the feature vector sequence $\mathbf{x}$ for the transcription character sequence $\mathbf{c} = c_1, \ldots, c_N$ is calculated using the Viterbi algorithm [11]. As a byproduct, the optimal character boundaries are returned and used to extract character images. An example of the complete process is shown in Figure 2 for the word "giht".

3. CHARACTER PROTOTYPE SELECTION

The aim of character prototype selection is to find representative character shapes in the training set that can be used as a valuable source of information for handwriting recognition. In this paper, we propose to represent character images obtained by HMM-based forced alignment (see Section 2) by means of graphs and obtain prototypes using several selection strategies known from general graph-based pattern recognition [12].

In the following, the handwriting graph representation is detailed, followed by a description of the graph edit distance used for graph matching and an introduction to the graph prototype selection methods considered in this paper.

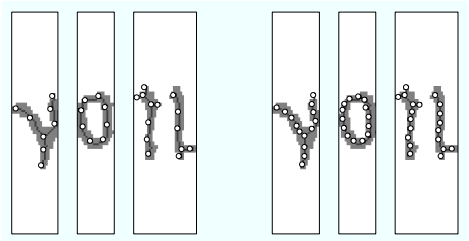


Figure 3: Character Graphs

3.1 Graph Representation

For graph representation of character images, a node-based representation of character skeletons is used that was proposed in [8]. An important property of these handwriting graphs is that no edges are used, while the essential structural information is still preserved with a high density of nodes. The advantage of not using edges is that an optimal graph edit distance, which is used for graph matching (see Section 3.2), can be calculated in polynomial instead of exponential time.

For thinning the binary character images, the 3×3 thinning operator proposed in [13] is applied. Based on two subiterations on a checkerboard pattern, one pixel wide medial curves are extracted while preserving connectivity. An implementation is given by Matlab's `bwmorph` function.

To derive a character graph from the skeleton, a node is added to the graph for each skeleton keypoint and is labelled with its position $(x,y) \in \mathbb{R}^2$. Keypoints include endpoints, intersections and the upper left pixel of circular structures. After all keypoints have been included in the character graph, connection points are added along the skeleton at regular distance D. An example is shown in Figure 3 for the characters "v", "o", and "n" with $D = 9$ (left) and $D = 5$ (right).

3.2 Graph Edit Distance

To calculate the dissimilarity $d(g_1, g_2)$ between two character graphs g_1 and g_2 we use the graph edit distance [14] for error-tolerant graph matching. The edit distance is given by the minimum cost of edit operations needed to transform g_1 into g_2. Possible edit operations include the insertion, deletion and substitution of nodes and edges.

Because no edges are used for the handwriting graphs, only edit operations on the nodes have to be considered. We use a constant cost C for node insertion as well as deletion, and the Euclidean distance between two nodes $||(x_1,y_1) - (x_2,y_2)||$ for node substitution. In the absence of edges, the problem of graph edit distance is reduced to an assignment problem that can be optimally solved by the Hungarian algorithm [15] in polynomial time.

3.3 Prototype Selection

After HMM-based forced alignment and character graph extraction, a set G of graphs is available for each character class present in the training set. By means of prototype selection, a subset $P \subseteq G$ is extracted that aims at representing the different writing styles of a character. Hereby, redundancy should be avoided while maintaining the capability to represent dominant character shapes. Although a random selection might work well in some cases, we use four other approaches that construct the set of prototypes in a more controllable manner. They are described in the following. For more details, we refer to [12].

3.3.1 Median Selection

Using median graph selection, a single prototype $P = \{p_1\}$ is chosen per character. The prototype p_1 is given by the median graph

$$median(G) = \arg\min_{g_1 \in G} \sum_{g_2 \in G} d(g_1, g_2)$$

with respect to the graph edit distance $d(g_1, g_2)$, i.e., the median graph is characterized by minimizing the sum of edit distances to all other graphs in G.

3.3.2 Center Selection

A slightly different concept than the median graph is given by the center graph

$$center(G) = \arg\min_{g_1 \in G} \max_{g_2 \in G} d(g_1, g_2)$$

where the maximum graph edit distance to all other graphs in G is minimized. Again, the center graph is selected as a single prototype $P = \{p_1\}$ to represent G.

3.3.3 Spanning Selection

For spanning prototype selection, the set $P = \{p_1\}$ of prototypes is initialized with the median graph $p_1 = median(G)$. Then, additional prototypes p_i are added iteratively based on the rule

$$p_i = \arg\max_{g \in G \setminus P} \min_{p \in P} d(g, p)$$

until a given number k of prototypes $P = \{p_1, \ldots, p_k\}$ is selected. At each step, the added prototype p_i is the graph that differs most from the previously selected prototypes.

3.3.4 k-Centers Selection

The k-centers selection is based on a k-medians clustering of G [16]. Starting from clusters $C_1 = \{c_1\}, \ldots, C_k = \{c_k\}$ with initial centers c_i obtained from a spanning selection of k prototypes, each of the remaining graphs is added to the cluster with the nearest cluster center. Then, cluster centers are recalculated using

$$c_i = center(C_i)$$

This process is repeated until no more cluster centers are changed. It results in k prototypes $P = \{p_1, \ldots, p_k\}$ given by the final cluster centers.

4. GRAPH SIMILARITY FEATURES

The proposed character prototype selection is evaluated with respect to a handwriting recognition task using graph similarity features that were recently proposed in [8]. The features are based on handwriting graphs and represent structural similarity with respect to a set of character prototypes by means of a sliding window with dynamic context width.

In [8], a manual selection of character prototypes was performed that will be compared to the automatic selection strategies proposed in this paper.

In the following, the graph similarity features are briefly described as well as the HMM-based single word recognition task considered for experimental evaluation. For a more detailed description, we refer to [8].

4.1 Feature Extraction

The extraction of the graph similarity features is based on the graph representation of handwritten text images described in Section 3.1 using the same preprocessing of the word images, i.e., binarization, normalization, and skeletonization.

Feature extraction is performed for each character prototype individually. A sliding window with the width of the character image is moved column-wise from left to right over the word graph. At each window center position i, the graph edit distance $d(g_i, p)$ between the window subgraph g_i and the prototype graph p is calculated. Taking all prototypes into account, this results in a sequence $\mathbf{x} = x_1, \ldots, x_T$ of feature vectors with $x_i \in \mathbb{R}^n$. Each of the n feature dimensions corresponds with the local dissimilarity of the word graph to one of the prototypes $p_1, \ldots, p_n$. The features are finally normalized with respect to the maximum graph edit distance to all prototypes in order to obtain similarity features with $0 \leq x_i \leq 1$.

4.2 HMM-Based Recognition

For HMM-based single word recognition, the character HMMs described in Section 2.3 are used. First, they are trained with respect to the graph similarity features using the Baum-Welch algorithm. For recognition, the character HMMs are then concatenated to word HMMs as shown in Figure 1b and the optimal word

$$w = \arg\max_{w} P(\mathbf{x}|w)$$

is found by means of Viterbi decoding given the feature vector sequence $\mathbf{x}$ of a word image from the test set. Hereby, the possible words are taken from a closed vocabulary $w \in V$ that includes all word classes from the test set. Note that, implicitly, a trivial language model is assumed, i.e., an equal a priori probability $P(w)$ is considered for each word in order to focus on the feature quality in the experimental evaluation.

5. EXPERIMENTAL EVALUATION

The proposed character prototype selection is evaluated for a single word recognition task using graph similarity features. The automatic selection procedures presented in Section 3.3, namely median, center, spanning and k-centers, are compared with the manual selection performed in [8].

The HMM-based single word recognition is performed on word images of the Parzival data set [9]. This data set includes digital images of a medieval manuscript originating in the 13th century. It contains the epic poem *Parzival* by Wolfram von Eschenbach, one of the most significant epics of the European Middle Ages. The manuscript is written in the Middle High German language with ink on parchment. 11,743 word images are considered that contain 3,177 word classes and 87 characters including special characters that occur only once or twice.

Table 1: Word accuracy on the test set with optimal number of prototypes per character k and feature dimension n. The improvement achieved for k-centers selection is statistically significant (t-test, $\alpha = 0.05$).

Selection	Accuracy	Parameters
Manual	94.00	n=79
Median	94.07	n=76
Center	94.31	n=76
Spanning	94.14	k=3, n=195
k-Centers	94.51	k=4, n=244

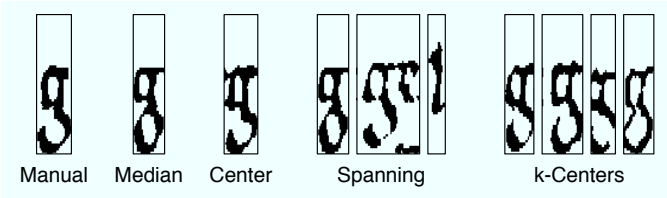


Figure 4: Selected Prototypes

5.1 Setup

First, the word images are divided into three distinct sets for training, validation, and testing. Half of the words, i.e., each other word, is used for training and a quarter of the words for validation and testing, respectively.

The reference system is based on the 79 manually extracted character prototypes reported in [8]. They contain one or two prototypes per character. For the median and center selection, one prototype per character class is chosen, resulting in 76 prototypes altogether. For the spanning and k-centers selection, up to five prototypes are chosen per character and the optimal number is determined with respect to the word accuracy obtained on the validation set. For each number of prototypes k, only those characters were taken into account that occur at least k times in the training set. This results in up to 280 prototypes for $k = 5$, which corresponds to the dimension $n = 280$ of the graph similarity features.

For HMM-based forced alignment, only one Gaussian mixture component is used that has turned out to be optimal for previous forced alignment experiments [17]. The optimal parameters for the graph similarity features, namely the node distance $D = 3.0$ and the node cost function $C = 3.0$ have also been adopted from previous work, as well as the number of states m of the character HMMs [8]. Finally, the number of Gaussian mixtures for single word recognition is optimized over a range of $G \in \{1, 5, 10, 15, 20, 25, 30\}$ on the validation set.

5.2 Results

The word accuracy on the test set is given in Table 1 for the manual prototype selection as well as for the proposed automatic selection methods. The manual selection is outperformed by all automatic selection strategies. In case of the k-centers selection, the improvement is statistically significant (t-test, $\alpha = 0.05$). In accordance to the findings re-

ported in [8], the maximum number of validated Gaussians $G = 30$ was optimal for all selection strategies, i.e., the graph similarity features allow a close adaption to the training set without suffering from overfitting. Since the increase in validation accuracy is asymptotic, only a minor gain is expected for higher values of G.

In Figure 4, the selected prototypes are shown for the different selection strategies, exemplarily for the character "g". Similar results are obtained for the other characters. The spanning selection results are ordered by iteration and the k-centers results by cluster size in descending order. For the median and center selection, the result is astonishingly close to the human choice and it makes sense that nearly the same recognition accuracy is achieved. Including more character prototypes by spanning selection results in the selection of outliers after the first iteration, stemming from wrong character segmentations. The visual inspection confirms the k-centers selection as the most promising prototype selection strategy, since all selected cluster centers are relatively clean character images with different appearances.

6. CONCLUSIONS

In this paper, automatic character prototype selection is proposed for handwriting recognition in historical documents. The proposed procedure can replace the time-consuming manual selection that is frequently performed to obtain character shape information needed for bootstrapping a recognition system.

Based on a forced alignment approach using analytical features and HMMs, word images are segmented into character prototype candidates. By means of a graph-based representation and graph edit distance, four prototype selection methods are presented, namely median, center, spanning, and k-centers selection.

In an experimental evaluation, the selected character prototypes are used for a single word recognition task on the historical Parzival data set in conjunction with graph similarity features and HMM-based recognition. Using median, center, and spanning selection, the same word accuracy was achieved as for manual prototype selection. By means of k-centers selection, the accuracy could even be outperformed significantly.

In future research, an outlier detection of wrongly segmented characters could improve the quality of the selected prototypes. The k-centers selection is furthermore a good candidate for a generalization of the graph similarity features approach to the multi-writer case.

Acknowledgments

This work has been supported by the Swiss National Science Foundation (Project CRSI22_125220).

REFERENCES

[1] A. Antonacopoulos and A.C. Downton. Special issue on the analysis of historical documents. *Int. Journal on Document Analysis and Recognition*, 9(2):75–77, 2007.

[2] G. Nagy and D. Lopresti. Interactive document processing and digital libraries. In *Proc. 2nd Int. Workshop on Document Image Analysis for Libraries*, pages 2–11, 2006.

[3] R. Plamondon and S. Srihari. Online and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. PAMI*, 22(1):63–84, 2000.

[4] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[5] J. Edwards, Y. W. Teh, D.A. Forsyth, R. Bock, M. Maire, and G. Vesom. Making Latin manuscripts searchable using gHMM's. In *Advances in Neural Information Processing Systems*, pages 385–392, 2004.

[6] J. Chan, C. Ziftci, and D. Forsyth. Searching off-line Arabic documents. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pages 1455–1462, 2006.

[7] Huaigu Cao and Venu Govindaraju. Template-free word spotting in low-quality manuscripts. In *Proc. 6th Int. Conf. on Advances in Pattern Recognition*, pages 135–139, 2007.

[8] A. Fischer, K. Riesen, and H. Bunke. Graph similarity features for HMM-based handwriting recognition in historical documents. In *Proc. 12th Int. Conf. on Frontiers in Handwriting Recognition*, pages 253–258, 2010.

[9] A. Fischer, M. Wüthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz. Automatic transcription of handwritten medieval documents. In *Proc. 15th Int. Conf. on Virtual Systems and Multimedia*, pages 137–142, 2009.

[10] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[11] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[12] K. Riesen and H. Bunke. *Graph Classification and Clustering Based on Vector Space Embedding*. World Scientific, 2010.

[13] Z. Guo and R. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989.

[14] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[15] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[16] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[17] E. Indermühle, M. Liwicki, and H. Bunke. Combining alignment results for historical handwritten document analysis. In *10th Int. Conf. on Document Analysis and Recognition*, pages 1186–1190, 2009.