Glyph Spotting for Mediaeval Handwritings by Template Matching

Jan-Hendrik Worch
University of Bremen
Am Fallturm 1
D-28359 Bremen
jworch@tzi.de

Mathias Lawo
Berlin-Brandenburgische
Akademie der Wissenschaften
Jägerstrasse 22-23
D-10117 Berlin
lawo@bbaw.de

Björn Gottfried
University of Bremen
Am Fallturm 1
D-28359 Bremen
bg@tzi.de

ABSTRACT

This paper reports on the analysis of different approaches in order to search for glyphs within handwritten mediaeval documents. As layout analysis methods are difficult to apply to the documents at hand, template matching methods are employed. A number of different shape descriptions are used to filter out false positives, since the application of correlation coefficients alone results in too many matches. The overall goal consists in the interactive support of an editor who is transcribing a given handwriting. For this purpose, the automatic spotting of glyphs enables the editor to compare glyphs within different contexts.

Categories and Subject Descriptors

I.4.7 [**Image Processing and Computer Vision**]: [Feature Measurement - Size and shape]

General Terms

Algorithms

Keywords

mediaeval handwriting, transcription assistance, correlation coefficient, glyph spotting, shape descriptions

1. INTRODUCTION

The offline analysis of mediaeval handwritings is a challenging task due to the degradation of old documents which are yellowed, blotted, and distorted, let alone the difficulties arising in offline handwriting analysis, as for instance described in [9]. In this paper, we report on our first results regarding the spotting of single glyphs. The idea is to avoid the application of the standard pipeline in document image processing, since the aforementioned difficulties, in particular regarding the specific character of a single writer, are too complex for the given documents. Instead, we aim at an assistance function which supports the user who is interested in transcribing a given handwriting.

A glyph spotting function searches for specific glyphs in the text. While single glyphs are often difficult to recognise for the human user, the context within a whole word gives him additional information and supports the recognition process. However, some contexts are more helpful than others. Being uncertain in a particular word, it would be helpful to have a look at the same glyph at other places within the text.

The method we shall present below searches automatically for additional appearances of a selected character within the text and emphasises its occurrences visually by colour. This enables the user to get an idea of that character in different contexts. He can jump directly to those appearances, instead of searching for them by hand which would be a cumbersome task. In this sense, the resulting function is to support the user in the analysis of handwritings.

This paper is structured as follows. The methods which have been employed for glyph spotting are explained in the following section. Their application to two very different documents is shown in section 3. The discussion in section 4 identifies the most promising methods for glyph spotting. A summary closes this paper.

This work has been supported by the Deutsche Forschungsgemeinschaft, **DFG-Gz.: GO 2023/4-1, LA 3007/1-1**

DocEng'12, September 4–7, 2012, Paris, France.

2. METHOD

2.1 Template Matching

The editor, whose first aim is the transcription of a handwritten text, can manually extract glyphs from the text which he intends to find within the document. The extracted glyphs are conceived of as templates for a matching method according to [3], as the underlying methodology for all of the following processing steps.

A template T is matched at position (x, y) of the image I using the correlation coefficient τ:

$$\tau_{x,y} = \frac{\sum\limits_{(i,j)\in T} \Big(I(x+i, y+j) \cdot T(i,j)\Big) - K\bar{T} \cdot \bar{I}(x,y)}{\sqrt{\sum\limits_{(i,j)\in T} \Big(I(x+i, y+j)\Big)^2 - K\Big(\bar{I}(x,y)\Big)^2}\,\sigma_T} \quad (1)$$

where

$$\sigma_T = \sqrt{\sum_{(i,j)\in T} \Big(T(i,j) - \bar{T}\Big)^2} \quad (2)$$

is the variance of the template, $\bar{T}$ is the mean value of the template, $\bar{I}(x, y)$ is the mean value of the image region which coincides with the current location of the template and K is the number of pixels the template consists of. The range of the correlation coefficient is $[-1..1]$, but for the sake of implementation and visualisation it is normalised to [0..255]. A non-maximum suppression strategy takes only the highest value within a neighbourhood.

2.2 Postprocessing

The result of template matching depends on the chosen threshold. The threshold splits the results into possible and denied matches and lies, as the range of the correlation coefficient, between 0 and 255. If the threshold is 0, the number of possible matches m between the template T and the image I corresponds to the maximal possible matches for the given image:

$$m = (I_{Width} - T_{Width}) \cdot (I_{Height} - T_{Height}) \tag{3}$$

By contrast a threshold of 255 will normally only find the template itself.

In the context of handwritten documents there will hardly be a single threshold that fits for every glyph. Hence, it is the aim to deploy a robust postprocessing method, which allows a low threshold that finds most instances, but avoids the problem of too many false positives. To analyse a wide variety of possibilities we compare three different kinds of postprocessing techniques that are based on the common categories in shape description: region-based, skeleton-based and contour-based.

2.2.1 *Size Restriction* (λ)

Before the above-mentioned postprocessing techniques are used, many false positives can already be rejected according to their size.

We try to find a glyph at every position in the image where template matching suggests a possible match, that is above a certain threshold. For this purpose, the height of the bounding box of the template is extended, its central point is placed at the point with the highest correlation coefficient within that locality, and the connected components within that region are taken as a probably disconnected glyph. Since the resulting glyph might not completely fill the according rectangular area, the region around the glyph needs to be cropped.

This finally results into a set of possible matches. For each glyph in that set the standard deviation in height is calculated and every glyph which has a size λ that fits the following range will be accepted for further processing:

$$\lambda \in [\text{average}_h + \text{stdv}_h,\ \text{average}_h - \text{stdv}_h] \tag{4}$$

2.2.2 *Moments* (η)

In order to filter out false positives, we use the central, normalised moments as region-based, statistical features [5]. They are invariant w. r. t. scale and translation. The central, normalised moment η of order (p, q) is defined by

$$\eta_{pq} = \mu_{pq} \left(\frac{1}{\mu_{00}} \right)^{(p+q+2)/2} \tag{5}$$

where μ_{pq} is the central moment of order (p, q).

The moments are calculated for order $(0, 0)$ up to $(3, 3)$. For two glyphs, which are to be compared, the sum of the pairwise absolute values of the differences are taken into account. If the resulting distance remains below a certain threshold, the tested glyph is accepted.

2.2.3 *Skeleton Comparison* (ζ)

Another common approach in the field of shape recognition is to compare skeletons [7]. In our approach the skeletons are extracted using the method of [8].

The resulting skeleton image is divided into four subimages. First the skeleton image is divided at half of the height, which leads to the first two subimages. Secondly, it is divided at half of the width. For all subimages the vertical and horizontal projections are computed along their direction of subdivision. In the projections the number of hills and valleys are counted, i. e. the sequences of rows with and without the occurrence of foreground pixels.

A second feature, which can be derived from the skeleton, counts the number of holes within the structure. Both features are concatenated and used to compare the skeletons.

$$\zeta = (num_{hills}, num_{valleys}, num_{holes}) \tag{6}$$

If the values of the feature vector of the template and the corresponding values of the glyph at hand are equal, the skeletons are considered to be similar and the glyph is finally accepted.

2.2.4 *Polyline Comparison* (ρ)

The last approach to compare glyphs with each other is based on polylines [6].

Again, the image is divided into four subimages, the same way as described in section 2.2.3. In a first step, for each of the subimages the upper, the lower, the left, and the right contour profile is calculated. Connecting the single points within one profile leads to a polyline. This is done for every profile so that the result consists of four polylines.

To compare two different glyphs, the distance between corresponding polylines is taken into account. To reduce the effect of noise, the central point of one polyline is used as origin and the points of the other polyline are translated according to the origin of the first polyline. Now, one can calculate the distance between two polylines P_1 and P_2 by using

$$D(P_1, P_2) = \sum_{s_1 \in P_1} \min_{s_2} \{ d(s_1, s_2) | s_2 \in P_2 \} \tag{7}$$

where $d(s_1, s_2)$ is the Euclidean distance between the middle points of the segments s_1 and s_2. If the distance of two polylines stays below a certain threshold, the glyph at hand is said to be similar to the template glyph.

In addition to this distance measure, a qualitative feature, which uses the positional contrast of each segment of a polyline, is taken into account: the so-called extent [4], which is a measure of complexity for polylines. The extent is calculated for each of the polylines and for two glyphs the corresponding values are compared. The absolute value of their distance must be below a certain threshold in order to accept a glyph.

3. EVALUATION

The first step within the transcription process is to mark manually a glyph to which similar glyphs are to be found. To mark a glyph the editor draws a rectangular area with the mouse around that glyph. In this area connected components are searched. Since there might be several connected components, the editor can choose which of them are part of the glyph, and additionally, he can edit the appearance of that glyph to get a proper template (e. g. removing or adding pixels, which were lost due to imperfections during binarisation).

To evaluate the different approaches, two different documents are compared: A mediaeval handwritten document with 1523 glyphs [1] (top of Fig. 1 shows a sample) and a printed one with 2233 glyphs [2] (bottom of Fig. 1). They have been chosen because of their different characteristics, in particular concerning irregular handwritten text as opposed to more regular printed text. Handwritten documents have a rather high variance within single character classes, whereas printed documents look much more regular.

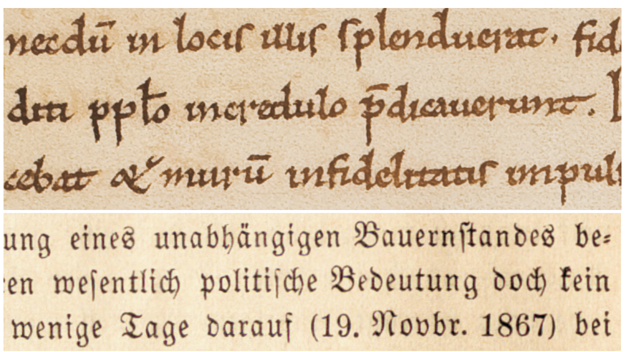

Figure 1: Subimages from [1] (top) and [2] (bottom)

Four experiments have been carried out. First of all, we evaluated template matching alone in order to learn how well the correlation coefficient works for these documents. For the other three experiments we combined template matching and size filtering with each of the three approaches described in sections 2.2.2 to 2.2.4. The glyphs were chosen according to their number of occurences, and additionally, specific glyphs were chosen because of their structure, i. e. some glyphs are substructures of other glyphs (e. g. 'r' is part of 'n' which is part of 'm'). They were taken into account in order to test the associated difficulties. Some examples are shown in Fig. 2.

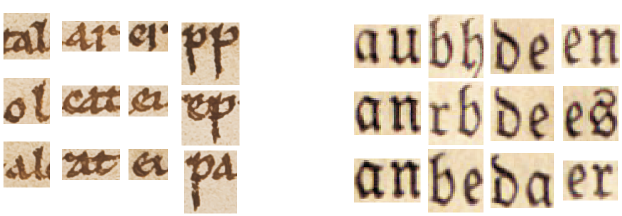

Figure 2: Some glyphs from [1] (left) and [2] (right) showing the intraclass differences.

The evaluation is testing different thresholds for template matching. The other thresholds have been determined experimentally and were not changed during the experiments. In the following sections, the mentioned threshold will always be the chosen one for template matching.

The results of the experiments can be seen in Table 1 and Table 2. The values are rounded average values over all runs of a single experiment. The values state how many matches were accepted at each threshold and how many of them were correct.

As expected, the template matching alone leads to the highest number of possible matches, but at the cost of accuracy. By contrast, the number of possible matches gets lower the higher the threshold of the correlation coefficient. But at the same time, a higher accuracy is achieved. The detailed conclusions from these experiments are drawn in the following discussion.

4. DISCUSSION

We started with the assumption that it would be easier to spot directly for glyphs on the document image than to analyse the layout first. It is in particular difficult to separate characters from each other in a single word, since handwritings do not clearly separate characters. Searching for specific glyphs on a document page calls for template matching methods, as by means of the correlation coefficient. It is our goal to analyse to what extent the correlation coefficient is able to make the physical layout analysis unnecessary, at least in connection with further filter methods.

4.1 The choice of a threshold

Employing the correlation coefficient, a threshold needs to be determined in order to separate accepted and rejected matches. We assume that too many false positives would be accepted with a low threshold. On the other hand, a rather high threshold could result into too many false negatives. However, aiming at the spotting of only a small number of example contexts of a specific glyph, a high threshold should serve for this application purpose: A high threshold might reject a number of true positives, but false positives will be avoided, or at least, small in number. This is what the results show in Table 1.

These results go with the observation that, in the case of a high threshold, there might be still false positives which are quite similar in their appearance. Conversely, true positives could be rejected with a high threshold if the variance of the handwriting is too large. In general, a high variability for the different instances of a class of glyphs results into many false negatives when taking a high threshold and a low threshold becomes necessary.

4.2 Template matching alone

The recall of true positives is quite large, employing the correlation coefficient alone. For instance, in [1] there are 11569 matches when looking for one of the characters as a template and when using a threshold of 25. One might wonder why there are so many matches, inasmuch as there are only 1523 characters on the document page. Omitting the layout analysis including the separation of characters, the correlation coefficient is determined for all conceivable positions on the entire document image, the latter having a resolution of 2469×1988 pixels. There are therefore 4,908,372 potential matches, which will be reduced by taking into account the size of the template, however, in the example this makes approximately 0.24%. Hence, this result is not particularly surprising. It shows the obvious necessity of adding a filter to the template matching method.

Table 1: Results when applying the presented methods to [1]

	possible matches at threshold					true positives at threshold					true positives [%] at threshold				
	25	175	200	225	240	25	175	200	225	240	25	175	200	225	240
τ	11569	351	54	2	1	52	44	30	2	1	0.45	12.47	55.73	100	100
$\tau+\lambda+\eta$	174	46	16	1	1	23	20	13	1	1	13.01	42.50	80.26	100	100
$\tau+\lambda+\zeta$	5	4	2	1	1	4	3	2	1	1	85.71	88.89	90.32	100	100
$\tau+\lambda+\rho$	13	11	8	1	1	9	8	7	1	1	65.43	69.68	87.62	100	100

Table 2: Results when applying the presented methods to [2]

	possible matches at threshold					true positives at threshold					true positives [%] at threshold				
	25	175	200	225	240	25	175	200	225	240	25	175	200	225	240
τ	15018	2103	753	222	86	112	112	112	108	82	0.74	5.31	14.84	48.41	95.55
$\tau+\lambda+\eta$	288	256	179	110	57	72	72	65	65	54	25.12	28.23	36.46	59.04	94.86
$\tau+\lambda+\zeta$	41	39	35	26	18	22	22	22	21	18	52.04	55.48	62.21	81.73	97.96
$\tau+\lambda+\rho$	63	60	52	46	34	39	39	39	39	33	61.71	64.99	74.02	83.72	98.13

4.3 The size constraint

The first filter considers the size of the matches. While the correlation coefficient finds too many matches, at least when taking lower thresholds, there are many false positives which mainly differ in their size. Sorting them out there is in fact a great reduction of the number of possible matches, which speeds up the postprocessing.

4.4 The choice of the threshold revisited

Irrespective of the filter method, a higher threshold means a higher fraction of true positives. For the application scenario, a high threshold seems to be the best choice. In this way, a small number of correct contexts can be found, which can be inspected by the human user who tries to determine the character class for a particular glyph.

4.5 The comparison with printed text

The comparison with a more regular style of glyphs is provided when referring to printed text, instead of finding another more regular handwriting style. As assumed, the recall of true positives is much larger in this case. The results show that this holds even for all methods.

An interesting difference is that, taking a threshold of at least 225, there only remain true positives in the case of the handwriting, while there are still some false positives regarding the printed text, even if for the latter a higher threshold is used. This seems to be related to similar glyphs which are neither sorted out by template matching, nor by the other filters. By contrast, different but similar glyphs are earlier sorted out in the case of the handwritten document.

For both document types, we learn that the filter based on normalised central moments leads to the best results. But moments are not the best filters in any case, if looking at the fraction of the results instead of the absolute values. Some of the other methods, as those based on skeletons, do have a rather small recall when using high thresholds for the correlation coefficient, but a superior fraction.

5. SUMMARY

We have proposed two new approaches to spot similar glyphs based on skeletons and polylines. The experiments point out that those approaches lead to a high accuracy at the cost of a low recall when applyling them to mediaeval handwritings. In contrast, template matching alone and in combination with central normalised moments leads to a high recall at the cost of precision. Additionally, we found out that there is a clear impact of the variance in handwriting, which results in a low recall for all approaches, when applying them to [1].

Hence, the usage of one or the other approach depends on the desired application. In our case, the higher accuracy is the optimal choice: The human user can easily spot a few, but correct glyphs, which are considered to be similar. Based on them he can make a decision on the meaning of a glyph with the aid of different contexts, without searching for them by himself or looking at too many false positives.

6. REFERENCES

[1] Philipps 1870, fol. 11v. Staatsbibliothek zu Berlin Preußischer Kulturbesitz.

[2] Die Grenzboten, 28. Jahrgang, 2. Semester 1. Band, Leipzig 1869. Scan 27 von der Staats- und Universitätsbibliothek Bremen.

[3] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Addison-Wesley, Reading, Mass., [3. ed.] reprint. with corr. edition, 1992.

[4] B. Gottfried. Qualitative similarity measures–the case of two-dimensional outlines. *Computer Vision and Image Understanding*, 110(1):117–133, 2008.

[5] M.-K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179 –187, February 1962.

[6] J.-C. Perez and E. Vidal. Optimum polygonal approximation of digital curves. *Pattern Recognition Letters*, 15:743–750, 1994.

[7] J. Stoppe and B. Gottfried. Skeleton comparisons: the junction neighbourhood histogram. In *Proc. of the 11th ACM symposium on Document engineering*, DocEng '11, pages 93–96, New York, NY, USA, 2011. ACM.

[8] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27:236–239, March 1984.

[9] J. Zhou, Q. Gan, A. Krzyzak, and C. Y. Suen. Recognition of handwritten numerals by quantum neural network with fuzzy features. *Intl. J. on Document Analysis and Recognition*, 2:30–36, 1999.