Features for Word Spotting in Historical Manuscripts

Toni M. Rath and R. Manmatha*
Multi-Media Indexing and Retrieval Group
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA 01002

Abstract

For the transition from traditional to digital libraries, the large number of handwritten manuscripts that exist pose a great challenge. Easy access to such collections requires an index, which is currently created manually at great cost. Because automatic handwriting recognizers fail on historical manuscripts, the word spotting technique has been developed: the words in a collection are matched as images and grouped into clusters which contain all instances of the same word. By annotating "interesting" clusters, an index that links words to the locations where they occur can be built automatically.

Due to the noise in historical documents, selecting the right features for matching words is crucial. We analyzed a range of features suitable for matching words using dynamic time warping (DTW), which aligns and compares sets of features extracted from two images. Each feature's individual performance was measured on a test set. With an average precision of 72%, a combination of features outperforms competing techniques in speed and precision.

1. Introduction

With the widespread use of computers and the Internet, libraries and institutions would like to make their collections of handwritten historical manuscripts available online or on digital media, such as DVDs. Due to the vast amounts of information contained in such collections, convenient access can only be achieved by generating some sort of index, very much like in the back of a book.

Since the current approach, manual transcription and index generation from the transcript, is extremely expensive and time-consuming, automatic approaches would be favorable. However, due to the typically significant degradation present in historic documents (faded ink, smudges, etc.; see [7] for examples) traditional handwriting recognizers based on Optical Character Recognition (OCR) do not perform well on this task.

The word spotting idea has been previously proposed as an alternative solution to this problem for single-author document collections [4, 5]. The approach is to segment pages into words, match the words as images, and use the match scores to cluster word images. Each word image cluster contains instances of the same word throughout the analyzed collection. By tagging a number of the resulting clusters, a partial index can be constructed for the collection.

We believe that the problem of deciding whether two given words are the same is easier than the recognition of a degraded handwritten word. In this work we present the results of our investigation into features which can be used for successful word image matching. In the following section we put our work in context with previous research efforts. Section 3 briefly explains the word spotting framework and the basic matching algorithm we use before we present the features we have tried in section 4. After presenting experimental results in section 5 we conclude with a summary and an outlook on further research.

2. Previous Work

The work by Tomai et al. [8] has shown the difficulty of historical handwritten manuscript recognition. Their goal was to produce a word-by-word mapping between a scanned document image and a manual transcript of that document. This would allow transcription words to be exactly located on a page. For each line of the document, multiple segmentation hypotheses are generated and the segments recognized by an OCR. The recognizer is using a limited lexicon, which is obtained from the perfect transcript. Even at a lexicon size of at most 11 words, the recognition performance was poor. This clearly shows that OCR is not

a viable option for historical manuscript recognition.

The word spotting idea for handwritten manuscripts was initially proposed by Manmatha et al. [4, 5]. They presented preliminary work on matching techniques and "pruning" methods, which can quickly discard unlikely matches for a given word by using simple word features such as the aspect ratio of a word's bounding box. Extensions to the matching algorithm and partial results on three data sets of 10 pages each from the George Washington collection can be found in [2].

In [3], matches for a user-provided template word image are searched for in each line of several pages using dynamic time warping on a number of features. This line based approach is expensive since the line is not segmented into words and the word has to be searched for at every possible position in the line. A number of heuristics are used to limit the search along the lines and also to re-orient portions of lines for matching. In addition, the matching algorithm aligns each feature using a separate dynamic time warp and combines the results heuristically. This means that for the same word-line pair, each feature may produce a different alignment. They provide results for 4 hand-picked queries with multiple templates (examples) on the Archives of the Indies - it appears that the best result for any individual word template has a precision of 0.4 or less. In this paper on the other hand, we correctly align the entire feature vector simultaneously so as to produce a common alignment over all feature vectors and also show much better results.

3. Matching Algorithm

The word spotting idea is to use word image matching results for building clusters, which contain words with the same ASCII-equivalent. At the current stage of our project, our research is solely focused on word matching techniques and no clustering is performed. Instead, every image in a given collection of documents (George Washington's handwritten manuscripts in our case) is treated as a query, which is used to retrieve a list of image matches from the collection, ranked by the match score. The following is an outline of the current system (see [7] for a more detailed explanation):

1. Segment each page in the collection into words (see [6] for a detailed explanation) and preprocess each segmented word image.

2. For each image t in the collection:

   (a) Determine the set of images P which have an appearance similar to t, based on some features that can be quickly calculated (e.g. aspect ratio of bounding box). This is the *pruning step*.

   (b) Compare image t against all images in the set P.

   (c) Sort images in P by their match score and build a ranked list of results.

Analyzing ranked result lists is a common task in the information retrieval community, and tools that calculate widely used statistics, such as *average precision* and *R-precision*, are readily available. We used the *trec_eval* program to judge the effectiveness of the features we present here.

Our matching algorithm uses dynamic time warping to align and compare sets of features which have been extracted from the matched images. A single feature vector consists of one feature value per column of the image it is calculated for. For example, if image $A = (a(i, j))$ is w_A pixels wide, a feature $f(A)$ would be a vector of length w_A:[1]

$$f(A) = (f(a(1, \cdot)), f(a(2, \cdot)), \ldots, f(a(w_A, \cdot))). \quad (1)$$

The dynamic time warping matching algorithm simultaneously aligns two *sets* of feature vectors F_A and F_B, which are extracted from the images A and B (F_B similarly):

$$F_A = (F_A(1, \cdot), F_A(2, \cdot), \ldots, F_A(w_A, \cdot)), \quad (2)$$

where every entry $F_A(x, \cdot)$ is a d-dimensional vector consisting of all extracted feature values for image column x. That is, F_A and F_B consist of d individually calculated features that will be aligned together by the dynamic time warping algorithm. The matching error[2] for matching images A and B is defined as

$$merr(A, B) = merr(F_A, F_B) = \frac{1}{l} DTW(w_A, w_B), \quad (3)$$

where l is the length of the warping path recovered by the dynamic time warping algorithm $DTW(\cdot, \cdot)$, which uses the recurrence equation

$$DTW(i, j) = min \begin{Bmatrix} DTW(i-1, j) \\ DTW(i, j) \\ DTW(i, j-1) \end{Bmatrix} + d(i, j), \quad (4)$$

$$d(i, j) = \sum_{k=1}^{d} (F_A(i, k) - F_B(j, k))^2. \quad (5)$$

In the following section we present a number of features that we used for matching words as images, using the above dynamic time warping algorithm.

4. Features

We have analyzed the performance of a number of features for use in conjunction with the DTW matching algorithm. Here we describe a selection of the more useful features, some of which have been previously reported in the

---

[1]The constraint - which is implied by the notation here - that every feature value is calculated strictly from the pixels in the corresponding image column can be relaxed.

[2]Matching *scores* can be obtained from *errors* by negation.

literature (e.g. see [1, 3]). Their performance, when used for matching word images, is analyzed in section 5.

Features are extracted from preprocessed rectangular word images, which are slant/skew-normalized and do not contain ascenders and descenders from other words. Some features are sensitive to translations along the vertical axis. For this reason, each image is padded on the bottom or top in order to shift the lower baseline to a position, which breaks the image into two areas at a ratio of 2/1. Figure 1 shows a typical result of this processing. All feature plots presented in this work are extracted directly from this image.
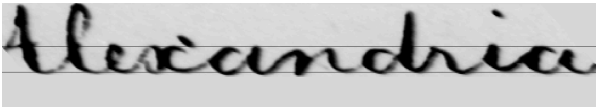

Figure 1. Preprocessed example image with upper and lower baseline displayed.

Comparability of the feature values across multiple words is ensured by normalizing the feature range to $[0, 1]$. The individual matching performance of the features when used with the dynamic time warping algorithm is analyzed in section 5.

4.1. Single-Valued Features

All of the features which are described in this section are single-valued, i.e. one scalar value is calculated per column in the original image.
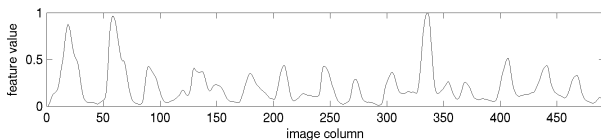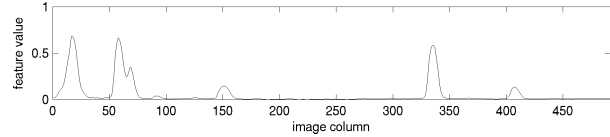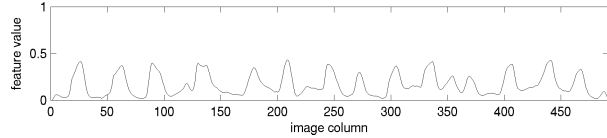

Figure 2. Normalized projection profile feature. All feature values are inverted for visualization purposes.

Projection Profile Each feature vector value is calculated by summing over the pixel values in the corresponding image column. Figure 2 shows the plot of a typical feature vector.

Partial Projection Profile The three partial profiles in Figure 3 result from calculating projection profiles for three horizontal strips in the original image: above, between and below both baselines. A single normalization factor is used for all of the three profiles, because the top and bottom strip can exhibit low variation (words with no ascenders/descenders). If all profiles would be separately normalized, slight errors in the baseline position estimation could seriously affect the result.
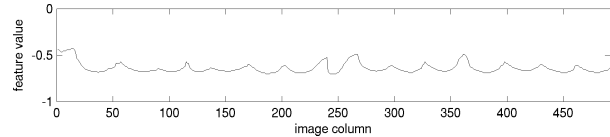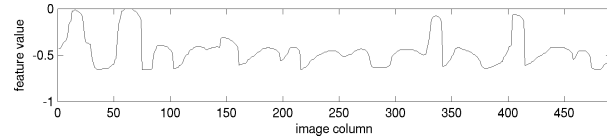

(a) above baselines.

(b) between baselines.

Figure 3. Partial projection profile features (lower profile omitted).

Upper/Lower Word Profile Upper/lower word profile features are computed by recording, for each image column, the distance from the upper/lower boundary of the word image to the closest "ink" pixel. If an image column does not contain ink, the feature value is computed by linear interpolation between the two closest defined values. Figure 4 shows two typical profiles (feature values are inverted).

(a) lower word profile.

(b) upper word profile.

Figure 4. Word profile features.

Background to Ink Transitions This feature (see Figure 5) records, for every image column, the number of transitions from the background to an "ink" pixel (determined by thresholding).
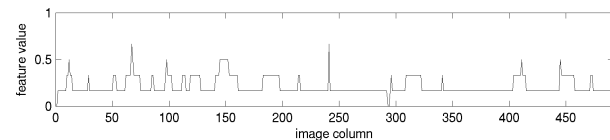

Figure 5. Normalized number of background-to-ink transitions feature.

Grayscale Variance The normalized variance of the grayvalue intensities in every image column is recorded for this feature (see Figure 6).
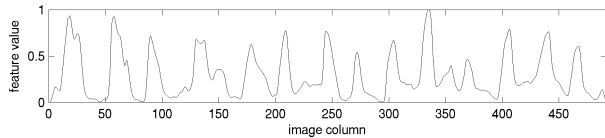
Figure 6. Normalized variance of column pixel intensities feature.
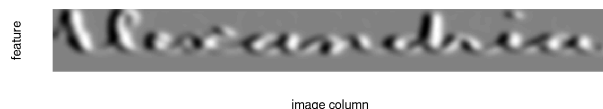
4.2. Feature Sets

All of the following features are multi-variate, i.e. a fixed number of values is calculated per image column.

Gaussian Smoothing   The original image is smoothed with an isotropic Gaussian kernel and resized to a generic height. Each line of the resulting image is now viewed as a separate feature. Figure 7 shows the feature set extracted from the original in Figure 1 (feature values are displayed as grayscale intensities).
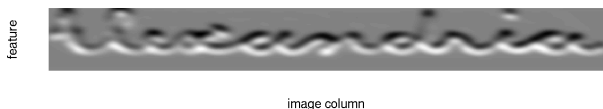

Figure 7. Gaussian-smoothed image ($\sigma = 4$ pixels), resized to generic height (15 lines).

Gaussian Derivatives   Similar to the feature set obtained from Gaussian-smoothing, these two sets are obtained from convolving the input image with a horizontal/vertical partial derivative of a Gaussian kernel. These filters respond to edges in the original image, which are widely used as features, because they can usually be reliably located. Figure 8 shows the resulting feature sets after convolution with horizontal/vertical derivative kernels and resizing to a generic height.

(a) with horizontal partial derivative Gaussian kernel.

(b) with vertical partial derivative Gaussian kernel.

Figure 8. Images convolved with Gaussian partial derivative kernels ($\sigma = 4$ pixels), then resized to generic height (15 lines).

5. Experimental Results

The features presented above have been used with the dynamic time warping algorithm for word image retrieval on a subset of the George Washington manuscript collection. 15 images from that collection of 2381 words were used to retrieve ranked lists of similar word images[3]. In order to cut down on the number of comparisons, a number of simple features (such as the aspect ratio of the image bounding box) where used to rule out unlikely matches. This process reduces the set of image pair comparisons to 12.57% of its original size, while retaining 90.33% of the true positive matches in the reduced set.

After the test runs, the *trec_eval* program was used to compute average precision and R-precision scores for the results (we refer the reader to standard information retrieval literature for an explanation of these statistics). Table 1 shows (among other things) each individual feature's performance on the set of 15 queries. Since the data set was preprocessed differently, the combination results differ from those reported in [7]. Also, the results for EDM have been adjusted as described in that work.

Among the single-valued features, the *upper word profile* feature clearly performs best with about 64% average precision. The two next best features (*projection profile* and *upper projection profile*) follow at a significant distance (both about 50%), but still perform well. Not surprisingly, the *lower projection profile* feature comes in last (25%), a confirmation for the perceived low level of information contained in the region of the word below the lower baseline: only words with different descender characteristics can be distinguished by this feature. The different levels of information content in the upper and lower part of words can also be noticed in the difference in average precision achieved by the *upper and lower word profile* features. The *lower word profile* performs better than the *lower projection profile*, since it also collects information from above the lower baseline.

Among the *Feature Set* runs, which use responses to Gaussian- and Gaussian derivative-kernels as features, the *Gaussian Smoothing* feature performed best (62.78%). The good performance of all three of these features can be attributed to the great amount of information that they contain: an approximate reconstruction of the original word can be easily achieved and is likely to be identifiable by a human reader. The difference in performance between the *vertical derivative* response feature and the two others can be explained by the lower stability of image locations in horizontal direction in the vertical derivative response.

To test the matching performance based on a combination of features, two runs with the best 4 and 3 features of the *Single-Feature* and *Feature Set* runs were performed (*Upper Proj. Profile* was not used because of its similarity to *Projection Profile*). Each feature vector in the combined run was constructed by merging corresponding entries of all feature vectors into multi-variate entries. The

---

[3]The same set was analyzed in [2].

Test Run	Description	Avg. Prec.	R-Prec.	Prec@5	execution time
Single-Feature	Projection Profile ($*$)	50.29	45.02	58.67	1.48
	Upper Projection Profile	49.91	49.26	52.00	1.15
	Middle Projection Profile	30.83	26.47	42.67	0.95
	Lower Projection Profile	24.85	20.59	17.33	0.96
	Upper Word Profile ($*$)	**64.29**	58.07	69.33	1.48
	Lower Word Profile ($*$)	42.99	41.69	53.33	1.48
	Bg./Ink Transitions ($*$)	42.46	44.49	38.67	1.37
	Graylevel Variance	37.88	36.27	46.67	1.62
Feature Set	Gaussian Smoothing ($**$)	**62.78**	57.26	73.33	1.42
	Gauss. Horizontal Der. ($**$)	59.63	53.22	68.00	1.74
	Gauss. Vertical Der. ($**$)	52.49	51.04	62.67	1.42
Feature Combinations	(features marked with $*$)	**72.56**	65.17	77.33	1.68
	(features marked with $**$)	67.31	64.02	72.00	2.42
Previously Reported [2]	XOR[a]	54.14	50.11	n/a	13
	SSD[a]	52.66	47.06	n/a	72
	SLH[a]	42.43	38.46	n/a	121.4
	EDM	**67.67**	62.85	74.67	14.3

Table 1. Performance statistics (all in percent) and execution time (in seconds, for one image pair) of various word matching algorithms.

first combined run ($*$) performs best (72.56%), which can be attributed to the higher redundancy present in the filter response features.

The best combination run ($*$) not only outperforms previously reported results [2] (72.56% vs. 67.67%), but it is also faster in execution time. Therefore, the feature-based matching approach based on dynamic time warping is clearly the best choice.

6. Conclusion

We have presented a number of features suitable for word image matching using the dynamic time warping algorithm. Each feature's individual matching performance on a set of 10 pages has been analyzed and we showed the increased benefit of combining the features, which yielded performance superior to previously published results in both precision and execution time.

The current feature combination assigns uniform weights to all features. We hope to further increase the matching precision by training weights for an improved feature combination. Also, with a successful matching algorithm at hand, we can now tackle the scalability issues in the word spotting project: we are currently working on improving our pruning mechanisms so we can move to larger data sets.

References

[1] C.-H. Chen. Lexicon-driven word recognition. In *Proc. Third Int'l Conf. on Document Analysis and Recognition*, pages 919–922, 1995.

[2] S. Kane, A. Lehman, and E. Partridge. Indexing George Washington's handwritten manuscripts. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts Amherst, 2001.

[3] A. Kołcz, J. Alspector, M. Augusteijn, R. Carlson, and G. V. Popescu. A line-oriented approach to word spotting in handwritten documents. *Pattern Analysis & Applications*, pages 153–168, 2000.

[4] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten archives. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval Collection*. AAAI/MIT Press, Menlo Park, CA, 1997.

[5] R. Manmatha, C. Han, E. M. Riseman, and W. B. Croft. Indexing handwriting using word matching. In *Digital Libraries '96: 1st ACM Int'l Conf. on Dig. Lib.*, pages 151–159, 1996.

[6] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten manuscripts. In *Proc. 2nd Int'l Conf. on Scale-Space Theories in Computer Vision*, pages 22–33, 1999.

[7] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proc. Computer Vision and Pattern Recognition Conf. (to appear)*, 2003.

[8] C. I. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *Proc. 8th Int'l Workshop on Frontiers in Handwriting Recognition*, pages 413–418, 2002.

---

[a]Prec@5 could not be calculated, since ranked result lists were not available.