Indexing of Handwritten Historical Documents - Recent Progress

R. Manmatha **Toni M. Rath**
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts Amherst

Abstract

Indexing and searching collections of handwritten archival documents and manuscripts has always been a challenge because handwriting recognizers do not perform well on such noisy documents. Given a collection of documents written by a single author (or a few authors), one can apply a technique called word spotting. The approach is to cluster word images based on their visual appearance, after segmenting them from the documents. Annotation can then be performed for clusters rather than documents.

Given segmented pages, matching handwritten word images in historical documents is a great challenge due to the variations in handwriting and the noise in the images. We describe investigations into a number of different matching techniques for word images. These include shape context matching, SSD correlation, Euclidean Distance Mapping and dynamic time warping. Experimental results show that dynamic time warping works best and gives an average precision of around 70% on a test set of 2000 word images (from ten pages) from the George Washington corpus.

Dynamic time warping is relatively expensive and we will describe approaches to speeding up the computation so that the approach scales. Our immediate goal is to process a set of 100 page images with a longer term goal of processing all 6000 available pages.

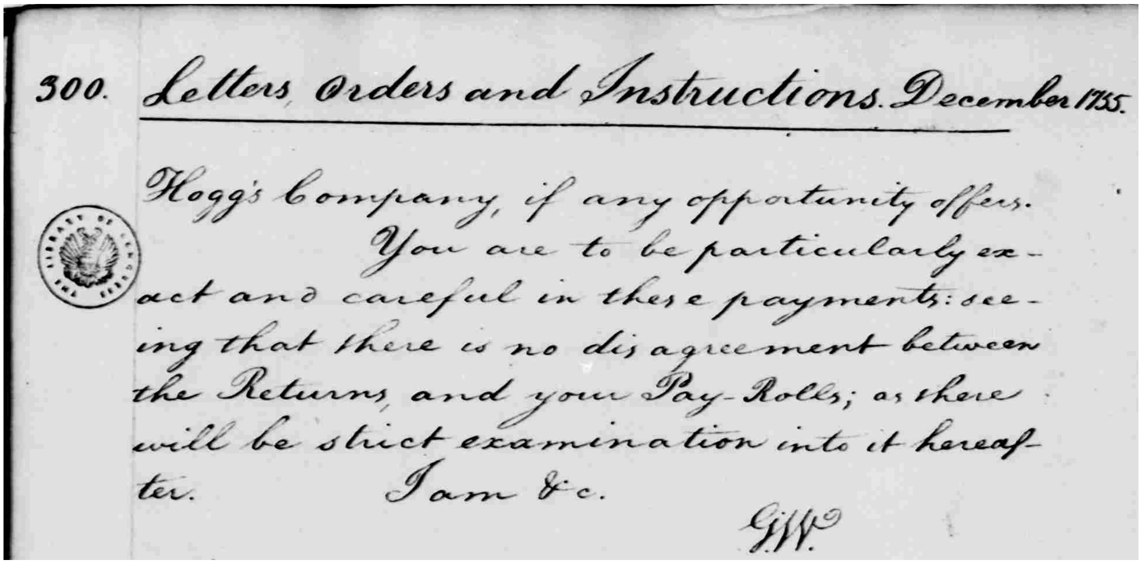
1 Introduction

Libraries contain an enormous amount of handwritten historical documents. Such collections are interesting to a great range of people, be it for historians, students or just curious readers. Efficient access to such collections (e.g. on digital media or on the Internet) requires an index, for example like in the back of a book. Such indexes are usually created by manual transcription and automatic index generation from a digitized version. While this approach may be feasible for small numbers of documents, the cost of this approach is prohibitive for large collections, such as the manuscripts of George Washington with well over 6000 pages.

Using Optical Character Recognition (OCR) as an automatic approach may seem like an obvious choice, since this technology has advanced enough to make commercial applications (e.g. tablet PCs) possible. However, OCR techniques have only been successful in the *online* domain, where the pen position and possibly other features are recorded during writing, and in *offline* applications (recognition from images) with very limited lexicons, such as automatic check processing (26 words allowed for legal amount field). However, for general historical documents with large lexicons, and the usually greatly degraded image quality (faded ink, ink bleed-through, smudges, etc.), traditional OCR techniques are not adequate. Figure 1 shows part of a page from the George Washington collection (this page is of relatively good quality).

Previous work [23] has shown the difficulties that even high-quality handwriting recognizers have with historical documents: the authors aligned a page from the Thomas Jefferson collection with a perfect transcription of the document. The transcription was used to generate a limited lexicon for each word hypothesis in the document. With a limited lexicon, the recognizer only has to decide which of a few possibilities the word to be recognized is. However, even for very small lexicon sizes of at most 11 (ASCII) words per word hypothesis in the image, only 83% of the words on a page could be correctly aligned with the transcription.

The wordspotting idea [12] has been proposed as an alternative to OCR solutions for building indexes of handwritten historical documents, which were produced by a single author: this ensures that identical words, which were written at different times, will have very similar visual appearances. This fact can be exploited by clustering words into groups with image matching techniques. Ideally, each clus-


300. Letters, Orders and Instructions. December 1755.

Hogg's Company, if any opportunity offers.
You are to be particularly ex-
act and careful in these payments: see-
ing that there is no disagreement between
the Returns, and your Pay-Rolls; as there
will be strict examination into it hereaf-
ter. I am &c.
GW.

Figure 1: Part of a scanned document from the George Washington collection.

ter of word images consists of all occurrences of a word in the analyzed document collection. Clusters that contain the most words can then be annotated in order to allow an index generation for the respective words[1].

Apart from the problem of segmenting images from documents (see [14] for a scale-space approach), the crucial step is to determine the similarity between images of words. In this work, we present our recent investigations into robust matching techniques for words in historical manuscripts, and evaluate their performance on a subset of the George Washington collection.

2 Matching Techniques

The matching techniques we have investigated fall roughly into two categories: image matching approaches that compare images pixel-by-pixel, and feature-oriented techniques, that extract image features and compare them after determining correspondences between them.

2.1 Pixel-by-Pixel Matching

The matching techniques in this section compare two images pixel-by-pixel after aligning them initially. The alignments compensate part of the variations which are inherent to handwriting (e.g. shear and scale changes). Some of the matching techniques that have been investigated include:

1. XOR[8, 13]: The images are aligned and then a difference image is computed. The difference pixel count determines the cost.
2. SSD[8]: translates template and candidate image relative to each other to find the minimum cost (= matching cost) based on the Sum of Squared Differences.
3. EDM[8, 13]: Euclidean Distance Mapping. This technique is similar to XOR, but difference pixels in larger groups are penalized more heavily, because they are likely to result from structural differences between the template and the candidate image, not from noise.

Early versions of the above algorithms were proposed by [12, 13]. Kane et al. [8] improved them by using extensive normalization techniques that align images and also conducted more systematic experiments . The above algorithms (including the normalization techniques) are detailed in [8].

2.2 Feature-Oriented Matching

A number of feature based techniques have also been investigated for matching word:

1. SLH[8, 13]: recovers an affine warping transform (using the Scott and Longuet-Higgins algorithm [22]) between sample points taken from the edge of the template and candidate image. The residual between template points and warped candidate points is used as the matching cost.
2. SC [1, 15]: Shape Context matching. This algorithm is currently the best classifier for handwritten digits. Two shapes are matched by establishing correspondences between their outlines. The outlines are sampled and *shape context histograms* are generated for each sample

[1] Clusters of *stop* words, such as 'the' are not annotated.

point: each histogram describes the distribution of sample points in the shape with respect to the sample point at which it is generated. Points with similar histograms are deemed correspondences and a warping transform between the two shapes is calculated and performed. The matching cost is determined from the cost associated with the chosen correspondences. [15] tested this algorithm for word matching in word spotting.

3. DTW[15]: A fixed number of features is extracted per image column, resulting in sets of "times series" (one per extracted feature) with the horizontal axis representing time. These time series can then be jointly aligned and compared with the Dynamic Time Warping algorithm. Examples of the time series features include projection profiles and upper/lower word profiles of the words.

4. CORR[18]: Using correlation, this technique recovers the correspondences between points of interest in two images. These correspondences are then used to construct a similarity measure.

Unlike the other techniques mentioned above, both the dynamic time warping and the point correspondence techniques share the property that they do not assume a global transformation between words[2]. These two techniques turn out to be the best performing techniques with DTW being somewhat better than the other technique. In the following sections we will discuss the DTW and CORR techniques in some more detail. For more details of the use of the SLH and SC algorithm in word spotting, see [8] and [15].

2.3 Matching Words with DTW

A person writing a word usually moves the pen from left to right in English (from right to left in some languages like Arabic and Hebrew). While a word is inherently two dimensional, there is some association between image columns and the time they were written. By carefully pre-processing word images, one can minimize variations in the vertical dimension and then recast word matching as a 1-dimensional problem along the horizontal axis.

The slant and skew angles at which a person writes, are usually constant for single words, and can be normalized using a global transform. On the other hand, the inter-character and intra-character spacing is subject to larger variations. DTW [20] offers a way to compensate for these variations, which is more flexible than linear scaling: in the matching algorithm that we describe here, image columns are aligned and compared using DTW. In our framework, each image column is represented by a time series sample point. Figure 2 shows an example alignment of two time series using dynamic time warping.

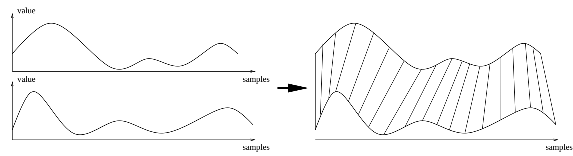


Figure 2: Alignment of two similar time series using dynamic time warping.

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[3]:

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

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[4] 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

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

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

To prevent pathological warpings, global path constraints are used to force the paths to stay close to the diagonal of the DTW matrix [19]. More details of the dynamic time warping algorithm are presented in [15].

[2] The warping transform used in the shape context algorithm is rather rigid.

[3] The notation implies that every feature value is calculated strictly from the pixels in the corresponding image column. This constraint can be relaxed.

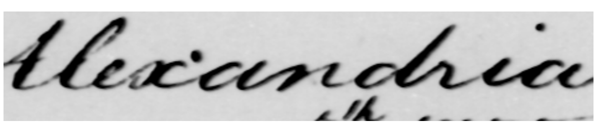
[4] Matching *scores* can be obtained from *errors* by negation.

Another approach, which uses dynamic time warping to compare features from a template word image to feature representations of whole lines of handwritten historical text, was described by Kołcz et al. in [10]. The main differences to our work are the application (retrieval by example in Kołcz's case), the matching framework (independent alignment of time series vs. constrained alignment in our approach) and the limited evaluation (4 query examples vs. thousands in our case).

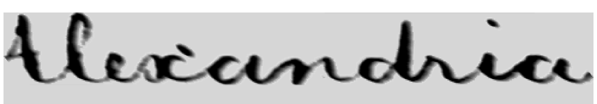
In the following section we present a number of features that we used for matching words as images, using the above dynamic time warping algorithm.

2.3.1 Features for Dynamic Time Warping

The images we operate on are all grayscale with 256 levels of intensity [0..255]. Before column features can be extracted from an image, a number of processing steps have to be performed: first, parts from other words that reach into this word's bounding box have to be removed and the background is cleaned; then inter-word variations such as skew and slant angle have to be detected and normalized; next, the bounding box is cropped so that it tightly encloses the word; finally, the image is padded with extra rows either on top or on the bottom, to move the baseline[5] to a predefined location. Figure 3 shows an original image and the result of the above processing steps.



(a) original image,



(b) cleaned and normalized version.

Figure 3: Original image and result after cleaning and normalization.

All of the column features we describe in the following are normalized to a maximum range of [0..1], so they are comparable across words. Our goal was to choose a variety of features presented in the handwriting recognition literature (e.g. [3] or [24]), such that an approximate reconstruction of a word from its features would be possible.

We use the following four features to represent word images (all of the example Figures were obtained from the image in Figure 3(b)):

[5]The baseline is the imaginary line people write on.

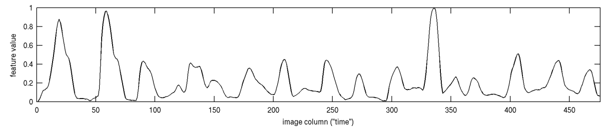


Figure 4: Projection profile feature (range-normalized and inverted).

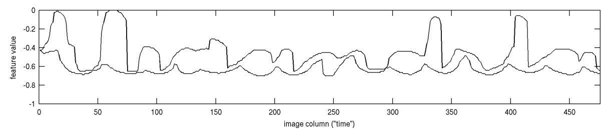


Figure 5: Upper and lower word profile features (normalized and inverted).

1. Projection Profile: these time series result from recording the sum of the pixel intensity values in every image column. The result is a profile that captures the distribution of ink along the horizontal axis of a word. Figure 4 shows a typical result.

2/3. Word Profiles: for every image, we can extract an upper and lower profile of the contained word, by going along the top (bottom) of the enclosing bounding box, and recording the distance to the nearest "ink" pixel in the current image column. Identifying ink pixels is currently realized by a thresholding technique, which we have found to be sufficient for our purposes. For more sophisticated foreground/background separation, see [11]. Together, these features capture the shape of the word outline (see Figure 5).

4. Background/Ink Transitions: for every image column, we record the number of transitions from a background- to an ink-pixel. This feature captures the inner structure of a word (see Figure 6 for an example).

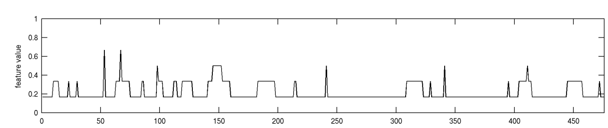


Figure 6: Background/ink transition-count feature (normalized).

We also tried a number of other features, including Gaussian derivatives and projection profiles that are calculated for parts of words. A discussion of the results can be found in [16].

2.3.2 Speeding up Dynamic Time Warping

While the word matching based on dynamic time warping works very well (see results in section 3), the computational load is quite high: for two images of width n, the algorithms complexity is $O(n^2)$. Here we present preliminary investigations into possibilites for speeding up DTW.

Using a global path constraint like the Sakoe-Chiba band [19] speeds up the computation, because the DTW matrix only has to be evaluated in a region around the diagonal. The complexity of the resulting algorithm is still $O(n^2)$, but with a lower constant.

Another approach is to use the *lower-bounding* paradigm (e.g. see [5]): the idea is to use a lower-bounding function $lb(\mathrm{A}, \mathrm{B})$, which always underestimates the real matching distance $merr(\mathrm{A}, \mathrm{B})$:

$$\forall \mathrm{A} \forall \mathrm{B}: \quad lb(\mathrm{A}, \mathrm{B}) \leq merr(\mathrm{A}, \mathrm{B}). \tag{6}$$

Such a lower-bounding function can be used for finding a time series in a collection, that has the lowest distance $merr$ to a given query series.

The approach is still to sequentially scan the data base for the best matching series, but lb is used to compare the query Q to a candidate C: if $lb(\mathrm{Q}, \mathrm{C})$ is greater than the distance $merr(\mathrm{Q}, \mathrm{M})$ to the currently best matching series M, it is not necessary to evaluate $merr(\mathrm{Q}, \mathrm{C})$, since

$$merr(\mathrm{Q}, \mathrm{M}) < lb(\mathrm{Q}, \mathrm{C}) \leq merr(\mathrm{Q}, \mathrm{C}). \tag{7}$$

We only need to evaluate $merr(\mathrm{Q}, \mathrm{C})$, if $lb(\mathrm{Q}, \mathrm{C})$ is less than $merr(\mathrm{Q}, \mathrm{M})$.

Of course, this strategy is only useful if lb has a lower complexity than $merr$. Several researchers have proposed lower-bounding functions for time series comparisons with DTW, with [9] being the tightest. The tightness is an important aspect of lower bounds, since it determines how often $merr$ has to be evaluated for time series that do not yield a distance which is lower than the current minimum. The lower bound in [9] was proposed for univariate time series. We have extended the approach to multivariate time series [17].

2.4 Word Matching using Point Correspondences

The image matching approach based on point correspondences identifies image corners in the input images using the Harris detector. Then, the similarity between these points is determined by correlating their intensity neigborhoods using the sum of squared differences measure. Then the recovered correspondences are used to calculate a measure of similarity between the input word images.

Recovering correspondences for all pixels in one word image would be an expensive operation, considering the search space, which is of quadratic size in the number of sample points in the two input images. Using the Harris detector allows us to select a limited number of points of interest, which are repeatable under a range of transformations and invariant to illumination variations [6].

The Harris detector operates on the matrix

$$\mathbf{M} = \begin{pmatrix} (\frac{\partial I}{\partial x})^2 & (\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ (\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & (\frac{\partial I}{\partial y})^2 \end{pmatrix},$$

where I is the gray level intensity image. A corner is determined when the 2 eigenvalues of M are large since this indicates grayscale variations in both the x and y direction.

2.4.1 Recovering Corner Correspondences

Here we determine pairs of corresponding corner points in the two input images. Most correspondence methods compare the characteristics of the local regions around feature points, and then select the most similar pairs as correspondences. The characteristics of local regions can be represented by either a feature vector (e.g. see [21]), or by windows of gray-level intensities.

We use the sum of squared differences (SSD) error measure to compare gray-level intensity windows, which are centered around detected corner locations. The reasons for selecting the SSD measure are its simplicity and the small number of operations required to calculate it - an important consideration when comparing a large number of image pairs.

In this simple approach, false point matches can be caused by a number of factors. We try to alleviate them with constraints:

- the size of the query word may be different from that of the candidate word image, that is, they have different resolutions. Assuming tight bounding boxes for all words, we resize all candidate images to the size of the query image.

- For a given feature point, there might be several feature points in a candidate image, which result in small SSD errors. to reduce this possibility, we constrain corresponding feature points in the candidate image to lie in the neighborhood of the corner point in the template image. This constraint also has the desirable effect of speeding up the algorithm.

In order to further decrease the computational load, we reduce each image to half-size. In essence, this can be regarded as doubling the size of the SSD correlation windows without slowing down the implementation. Using larger SSD windows can help prevent false matches, because of the added context that is taken into account when comparing image regions.

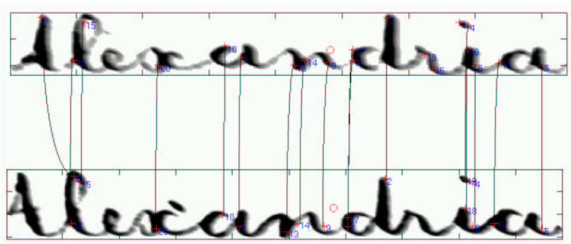


Figure 7: Recovered correspondences in two word images.

Experiments showed that adding the above constraints greatly improved the matching accuracy (see Figure 7 for an example of recovered correspondences).

2.4.2 Distance Measure Calculation

The correspondence between pairs of feature points captures the similarity between *local* regions of two images. In order to judge the similarity of two word images, this local information is now combined into a *global* measure of similarity.

After investigating various approaches for distance measurements, we used the following distance measurement:

$$D(A,B) = \frac{\sum_i \sqrt{(x_{bi}-x_{ai})^2 + (y_{bi}-y_{ai})^2}}{\#\text{correspondences}} \cdot \frac{\#\text{feature points in A}}{\#\text{correspondences}}, \quad (8)$$

where A is the query image, and B a candidate image; (x_{ai}, y_{ai}) and (x_{bi}, y_{bi}) are the coordinates of a pair of corresponding feature points, in A and B respectively. Essentially, we are calculating the mean Euclidean distance of corresponding feature points. Additionally, we penalize for every point in image A, that does not have a correspondence in image B[6] by multipliying the average distance with a weight. Thus, the fewer corresponding feature points are found in the candidate image B, the larger the distance between B and A.

More details on the point correspondence technique can be found in [18].

3 Results

We conducted experiments on two labeled data sets, both 10 pages in size. Data set 1 is of acceptable quality (see Figure 8(a)). The second set is very degraded (see Figure 8(b)) - even humans have difficulties reading these pages. We prepared four test sets for our evaluation:

A: 15 images in test set 1.

B: 2372 images of good quality (test set 1).

C: 32 images in test set 2.

D: 3262 images of bad quality (test set 2).

Test sets A and C are mainly for the purpose of comparison to previously reported techniques (in [8]) and for quick performance tests of new techniques.

3.1 Experimental Methodology

For a given test set/matching algorithm pair, the following evaluation was performed: each image in the test set was regarded as a query, which was used to rank the rest of the images in the collection according to their similarity to the query. Some query/candidate pairs are not compared by the matching function ("pruned"), because they are dissimilar according to a set of simple heuristics (image length, aspect ratio of bounding box, etc.).

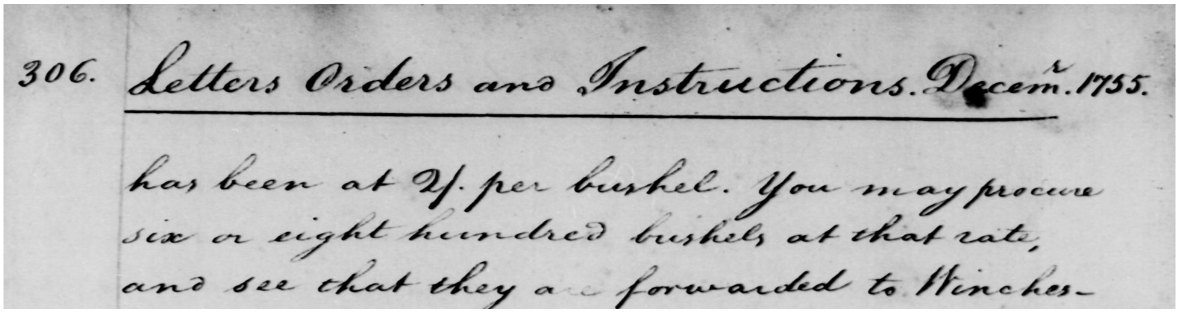
The ranked lists of retrieved word images were evaluated using the mean average precision measure [25], which is commonly used in the information retrieval field. For the purpose of evaluation, a candidate image was considered "relevant" to the query image, if the image labels (ASCII annotation) matched.

Previous work by [8] used a somewhat misleading evaluation, that considered each query image as a candidate. With this evaluation, the results are biased, since most of the techniques always retrieve the query image at rank 1, if it is part of the candidate set. In table 1, we have provided both the old evaluation and a new version, which removes query images from the candidate set. The new evaluation values for some of the discussed matching techniques appear in the 4 right-most columns.

3.2 Result Discussion

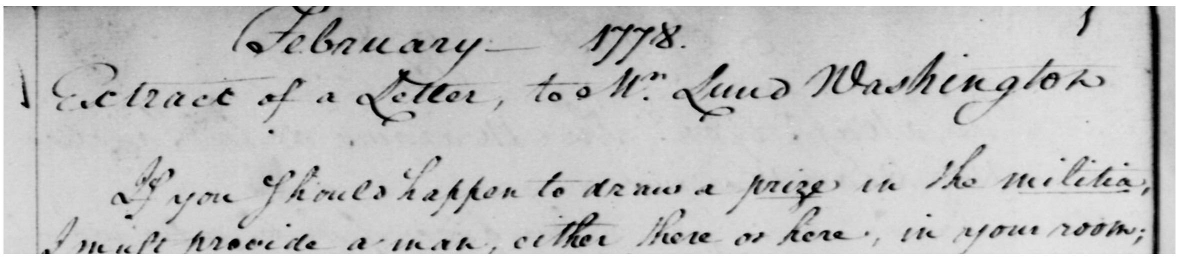
As can be seen from table 1, DTW and CORR work best, with similar performance on all data sets. While CORR performs better on the smaller sets A and C, DTW seems to have a slightly better overall performance (data sets B and D). The EDM matching technique seems to perform well on data set A, but its performance is significantly lower on set C. Additionally, it is unclear what its overall performance is, since we had no raw results available for sets B and D. The rest of the matching approaches (XOR, SSD, SLH and SC) does not perform nearly

[6] This can happen if the search area in B for a corner point in A does not contain any points of interest.



306. Letters Orders and Instructions. Decem.r 1755.
has been at 2/. per bushel. You may procure
six or eight hundred bushels at that rate,
and see that they a[re] forwarded to Winches-

(a) example from test set 1 (good quality),



February 1778.
Extract of a Letter, to Mr. Lund Washington
If you should happen to draw a prize in the militia,
I must provide a man, either there or here, in your room;

(b) example from test set 2 (bad quality).

Figure 8: Examples from the two test sets used in the evaluation.

Run	XOR	SSD	SLH	SC	EDM	DTW	CORR	SC	EDM	DTW	CORR
A	54.14%	52.66%	42.43%	48.67%	72.61%	73.71%	73.95%	40.58%	67.67%	67.92%	69.69%
B	n/a	n/a	n/a	n/a	n/a	65.34%	62.57%	n/a	n/a	40.98%	36.23%
C	n/a	n/a	n/a	n/a	15.05%	58.81 %	59.96%	n/a	n/a	13.04%	14.84%
D	n/a	n/a	n/a	n/a	n/a	51.81%	51.08%	n/a	n/a	16.50%	15.49%

Table 1: Average precision scores for all test runs (XOR: matching using difference images, SSD: sum of squared differences technique, SLH: technique by Scott & Longuet-Higgins [22], SC: shape context matching [1], EDM: euclidean distance mapping, DTW: dynamic time warping matching, CORR: recovered correspondences). Four right-most columns show corrected evaluation results.

as well as DTW and CORR, with mean average precision values in the 40-50% range.

The general performance difference of DTW and CORR on data sets A and B can be explained by the pruning heuristics, which work much better on set A than on set B: in set A, only 10% of the valid matches are discarded in the pruning, while in set B, almost 30% are discarded. A similar observation can be made for the test sets C and D: on both sets, the pruning discards around 45% of the valid matches. This can be seen in the smaller differences in mean average precision for both DTW and CORR on these data sets. The reason for the high rejection rate of valid matches by the pruning lies in the word segmentation, which is heavily affected by the bad quality of the document images.

4 Conclusions and Outlook

Given the challenges in recognizing words from the large vocabularies of handwritten manuscript collections, word spotting involving word matching is a reasonable approach for solving the problem of indexing such manuscript collections. We have discussed a number of different approaches to matching with the best performing ones being dynamic time warping and a point correspondence based technique. Challenges remain, including the creation of word clusters and the necessity of speeding up these algorithms sufficiently, so that large collections can be handled in a reasonable amount of time.

Building a system involves creating a user interface. While it is straightforward to imagine a visual index with pictures and links to pages, it is not clear whether users would be able to use such an index effectively. An ASCII user interface can be created by annotating the matched word clusters manually - permitting a more traditional index. Recent advances in automatic picture annotation [2, 4, 7], using machine learning and information retrieval techniques, may permit a completely different approach to this problem through automatic annotation of word image clusters.

Acknowledgments

We would like to thank Jamie Rothfeder and Shaolei Feng for contributing to the work on point correspondences for word spotting. We also thank the Library of Congress for providing the images of the George Washington collection.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the National Science Foundation under grant number IIS-9909073. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] S. Belongie, J. Malik and J. Puzicha: *Shape Matching and Object Recognition Using Shape Contexts*. IEEE Trans. on Pattern Analysis and Machine Intelligence **24**:24 (2002) 509-522.

[2] D. M. Blei and M. I. Jordan: *Modeling Annotated Data*. Technical Report UCB//CSD-02-1202, 2002.

[3] C.-H. Chen: *Lexicon-Driven Word Recognition*. In: Proc. of the Third Int'l Conf. on Document Analysis and Recognition 1995, Montréal, Canada, August 14-16, 1995, pp. 919-922.

[4] P. Duygulu, K. Barnard, N. de Freitas and D. Forsyth: *Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary*. In: Proc. 7th European Conference on Computer Vision, Copenhagen, Denmark, May 27-June 2, 2002, vol. 4, pp. 97-112.

[5] C. Faloutsos: *Multimedia IR: Indexing and Searching*. In: Modern Information Retrieval, R. Baeza-Yates and B. Ribeiro-Neto; Addison-Wesley, Reading, MA, 1999.

[6] C. Harris and M. Stephens: *A Combined Corner and Edge Detector*. In: Proc. of the 4th Alvey Vision Conf., 1988, pp. 147-151.

[7] J. Jeon, V. Lavrenko and R. Manmatha: *Automatic Image Annotation and Retrieval Using Cross-Media Relevance Models*. CIIR Technical Report MM-41, 2003.

[8] S. Kane, A. Lehman and E. Partridge: *Indexing George Washington's Handwritten Manuscripts*. Technical Report MM-34, Center for Intelligent Information Retrieval, University of Massachusetts Amherst, 2001.

[9] E. Keogh: *Exact Indexing of Dynamic Time Warping*. In: Proc. of the 28th Very Large Databases Conf. (VLDB), Hong Kong, China, August 20-23, 2002, pp. 406-417.

[10] 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 **3** (2000) 153-168.

[11] G. Leedham, S. Varma, A. Patankar and V. Govindaraju: *Separating Text and Background in Degraded Documents Images - A Comparison of Global Thresholding Techniques for Multi-Stage Thresholding*. In: Proc. of the 8th Int'l Workshop on Frontiers in Handwriting Recognition 2002, Niagara-on-the-Lake, ON, August 6-8, 2002, pp. 244-249.

[12] 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 Digital Libraries, Bethesda, MD, March 20-23, 1996, pp. 151-159.

[13] R. Manmatha and W. B. Croft: *Word Spotting: Indexing Handwritten Archives*. In: Intelligent Multi-media Information Retrieval Collection, M. Maybury (ed.), AAAI/MIT Press 1997.

[14] 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, Corfu, Greece, September 26-27, 1999, pp. 22-33.

[15] T. M. Rath and R. Manmatha: *Word Image Matching Using Dynamic Time Warping*. to appear in Proc. of the Computer Vision and Pattern Recognition Conf. 2003.

[16] T. M. Rath and R. Manmatha: *Features for Word Spotting in Historical Manuscripts*. to appear in Proc. of the 7th Int'l Conf. on Document Analysis and Recognition 2003.

[17] T. M. Rath and R. Manmatha: *Lower-Bounding of Dynamic Time Warping Distances for Multivariate Time Series*. CIIR Technical Report MM-40, 2003.

[18] J. L. Rothfeder, S. Feng and T. M. Rath: *Using Corner Feature Correspondences to Rank Word Images by Similarity*. CIIR Technical Report MM-44, 2003.

[19] H. Sakoe and S. Chiba: *Dynamic Programming Optimization for Spoken Word Recognition*. IEEE Trans. on Acoustics, Speech and Signal Processing **26** (1980) 623-625.

[20] D. Sankoff and J. B. Kruskal: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA, 1983.

[21] C. Schmid and R. Mohr: *Local Grayvalue Invariants for Image Retrieval*. IEEE Trans. on Pattern Analysis and Machine Intelligence **19**:5 (1997) 530-535.

[22] G. L. Scott and H. C. Longuet-Higgins: *An Algorithm for Associating the Features of Two Patterns*. Proc. of the Royal Society of London **B224** (1991) 21-26.

[23] C. I. Tomai, B. Zhang and V. Govindaraju: *Transcript Mapping for Historic Handwritten Document Images*. In: Proc. of the 8th Int'l Workshop on Frontiers in Handwriting Recognition 2002, Niagara-on-the-Lake, ON, August 6-8, 2002, pp. 413-418.

[24] Ø. D. Trier, A. K. Jain and T. Taxt: *Feature Extraction Methods for Character Recognition - A Survey*. Pattern Recognition **29**:4 (1996) 641-662.

[25] C. J. van Rijsbergen: *Information Retrieval*. Butterworth, London, England, 1979.