Word Image Matching Using Dynamic Time Warping

Toni M. Rath and R. Manmatha*
Multi-Media Indexing and Retrieval Group
Center for Intelligent Information Retrieval
University of Massachusetts
Amherst, MA 01003

Abstract

Libraries and other institutions are interested in providing access to scanned versions of their large collections of handwritten historical manuscripts on electronic media. Convenient access to a collection requires an index, which is manually created at great labour and expense. Since current handwriting recognizers do not perform well on historical documents, a technique called word spotting has been developed: clusters with occurrences of the same word in a collection are established using image matching. By annotating "interesting" clusters, an index can be built automatically.

We present an algorithm for matching handwritten words in noisy historical documents. The segmented word images are preprocessed to create sets of 1-dimensional features, which are then compared using dynamic time warping. We present experimental results on two different data sets from the George Washington collection. Our experiments show that this algorithm performs better and is faster than competing matching techniques.

1. Introduction

Traditional libraries contain an enormous amount of handwritten historical documents that they would like to make available electronically on the Internet or on digital media. However, such large collections can only be accessed efficiently if a searchable or browsable index exists, just like in the back of a book. The current state-of-the-art approach to this task is to manually create an index for the collection. Since manual indexing is expensive, automation is desirable in order to reduce costs.

Success in *offline* handwriting recognition, where only an image of the produced writing is available, has been limited to domains with small vocabularies, such as automatic mail sorting and check processing. In addition, these domains usually provide good quality images, while the quality of historical documents is often significantly degraded due to faded ink, stained paper, and other adverse factors (see Figure 1). Consequently, traditional Optical Character Recognition (OCR) techniques that usually recognize words character-by-character, fail when applied to historical manuscripts.

For collections of handwritten manuscripts written by a single author (or a few authors) – for example the George Washington collection used in this paper – the images of multiple instances of the same word are likely to look similar. For such collections, the *Word spotting* idea [5] provides an alternative approach to index generation: first, each page in the document collection is segmented into words, and the different instances of a word are clustered together using image matching. Then, a human can tag the n most interesting clusters for indexing with the appropriate ASCII-equivalent, which could be used to build a partial index for the analyzed collection. Historical handwritten documents are often of poor quality and unlike printed documents, there is variation in the way the words are written. Thus, both segmentation of a page into words and the matching of word images are challenging problems for such documents.

Previous work by [6] has dealt with the problem of segmenting such images of historical documents. In this work, we present a word matching algorithm that compares word images using Dynamic Time Warping (DTW). DTW has been widely used in the speech processing, bio-informatics and also the online handwriting communities to match 1-D signals. Although the matching of word images is in general a 2-dimensional problem, we recast it as a 1-dimensional problem since there is a loose association of image columns with the time that they were written over. By carefully preprocessing the image we try to minimize the variations in the other dimension. We then extract a number of features from each image column and match the resulting feature sequences with the DTW algorithm. DTW can handle local distortions in word images and is not restricted to a single global transform. We compare this approach to a number of

*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.

other techniques, including affine-corrected Euclidean Distance Mapping, the shape context algorithm, and correlation using sum of squared differences. Our results show that the algorithm proposed here outperforms the other techniques both in terms of accuracy as well as speed.

In the following section, we put our work in context with previous efforts in this direction. Section 2 reviews the dynamic time warping algorithm and introduces our matching technique. After presenting our results and comparing them to other word image matching methods in section 3, we conclude with an outlook on further research.

1.1. Previous Work

In [10] the problem of spotting word images in historical documents using a perfect transcript (obtained manually) is addressed. An OCR is used to recognize the word images and the recognized images are aligned with the transcript. Good results were only obtained when the recognizer's lexicon was restricted to the ASCII versions of the line to be recognized (obtained from the perfect transcript). The word alignment accuracy of just about 83% (on a single page) shows how challenging the task of word spotting for historical documents is, even in the presence of a perfect (manually generated) transcript.

The word spotting idea was proposed by [5]. The authors presented some preliminary work on matching techniques and methods for discarding unlikely matches ("pruning") based on simple image features. In [3], the previously described techniques were extended and refined. Partial results on three annotated data sets, each 10 pages, were reported.

[4] examine the problem of spotting occurrences of a known template word in each line of several pages. Their approach is line based unlike the word based approach used here. Thus, while our algorithm solves a sequence matching problem, their algorithm solves a very expensive subsequence matching problem. Since [4] do not perform segmentation, the word templates are hand generated. In addition, the technique requires multiple ($>$10) handpicked training samples for each word. We believe this makes their technique not practical for automation. In contrast, the templates proposed here are automatically generated and multiple training samples are not needed. The matching algorithm proposed in [4] is also problematic, since it 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. 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. [4] provide results for 4 hand-picked individual words on the Archives of the Indies - this data set seems to have been scanned from the originals and is probably of good quality. It appears from these results that the best result for any individual word template has a precision of 0.4 or less. No statistical results for a set of word templates are provided (presumably because this line-based approach is too expensive to run).

The *shape context* approach [1] for shape matching 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 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. We compare the performance of the shape context algorithm against our technique in section 3.

2. Matching

Previous research [3] indicates that good matching performance can be achieved by a technique that skews, resizes and aligns two candidate word images with respect to each other and then compares them pixel-by-pixel. We use DTW to match word images, because it offers additional flexibility to compensate for handwriting variations.

Running a matching algorithm is expensive with growing collection sizes, so *pruning* techniques which can quickly discard unlikely matches are used. We briefly summarize the applied pruning techniques in the next section. Then, we shortly review the Dynamic Time Warping algorithm before going on to explaining its application in our matching technique.

2.1. Pruning

Pruning is a way to quickly determine whether a pair of images is either dissimilar or likely to match each other. In [5], pruning of word pairs based on the area and aspect ratio of their bounding boxes was performed. The idea is to require word images, which will later be compared, to have similar *pruning statistics* (e.g. area of bounding box).

The authors of [3] extended the pruning based on area and aspect ratio of word bounding boxes. Their technique additionally requires two words to have the same number of descenders (strokes below the *baseline*[1], e.g. bottom part of the letter 'q').

2.2. DTW

Dynamic Time Warping [8] is used to compute a distance[2] between two time series. A time series is a list of samples

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

[2] The terms *distance* and *matching cost* are used synonymously in this work; we do not require the presented distances to obey all metric axioms.

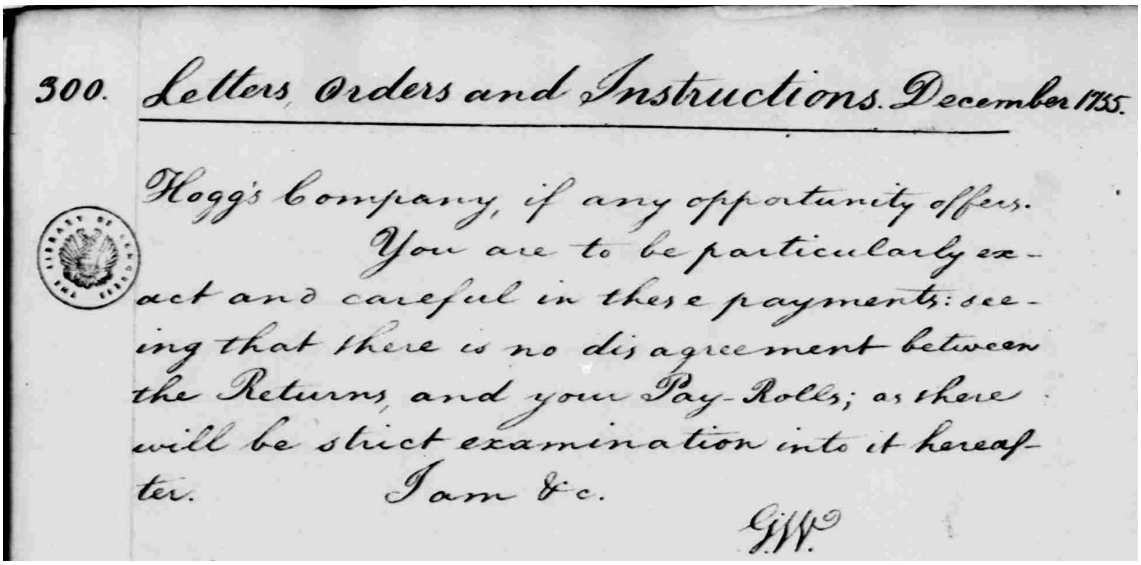
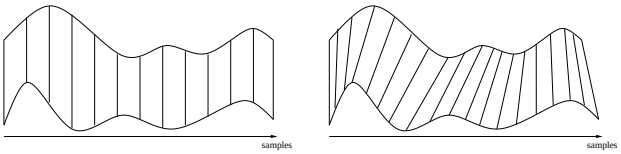


Figure 1: Part of a scanned document from the George Washington collection.

taken from a signal, ordered by the time that the respective samples were obtained.

A naive approach to calculating a matching distance between two time series could be to resample one of them and then compare the series sample-by-sample. The drawback of this method is that it does not produce intuitive results, as it compares samples that might not correspond well (see Figure 2(a)).



(a) naive alignment after resampling, (b) alignment with DTW.

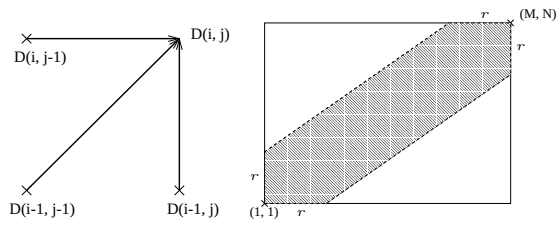
Figure 2: Different alignments of two similar time series.

Dynamic Time Warping solves this discrepancy between intuition and calculated matching distance by recovering optimal alignments between sample points in the two time series. The alignment is optimal in the sense that it minimizes a cumulative distance measure consisting of "local" distances between aligned samples. Figure 2(b) shows such an alignment. The procedure is called *Time Warping* because it warps the time axes of the two time series in such a way that corresponding samples appear at the same location on a common time axis.

The DTW-distance between two time series $x_1 \dots x_M$ and $y_1 \dots y_N$ is $D(M, N)$, which we calculate in a dynamic programming approach using

$$D(i,j) = min \left\{ \begin{array}{l} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{array} \right\} + d(x_i, y_i). \quad (1)$$

The particular choice of recurrence equation and "local" distance function $d(\cdot, \cdot)$ varies with the application. Using the given three values $D(i, j-1)$, $D(i-1, j)$ and $D(i-1, j-1)$ in the calculation of $D(i, j)$ realizes a *local continuity constraint* (cf. Figure 3(a)), which ensures smooth time warping (e.g. no samples left out in warping).



(a) local continuity constraint. (b) global path constraint ($r = 15$ in our implementation).

Figure 3: Constraints used in the current dynamic time warping implementation.

Backtracking along the minimum cost index pairs $(i, j)_k$ starting from (M, N) yields the DTW *warping path*. We use the *Sakoe-Chiba* band constraint [7] to ensure this path stays close to the diagonal of the matrix which contains the $D(i, j)$ (see Figure 3(b)). This way, pathological warpings that align a small portion in one sequence to a large portion in the other are avoided. A more detailed discussion of continuity constraints can be found in [8].

2.3. Matching Words with DTW

While the slant and skew angle at which a person writes is usually constant for single words, the inter-character and intra-character spacing is subject to larger variations. DTW offers a more flexible way to compensate for these variations than linear scaling: in the matching algorithm that we propose, image columns are aligned and compared using DTW.

To do this, we first have to normalize the slant and skew angle of candidate images to compensate for inter-word variations. Then, from each word, four features per image column are extracted and combined into a single time series of multi-variate samples. That is, for each image I with height h and width w, we extract a time series $X(I) = \mathbf{x}_1 \dots \mathbf{x}_w$, where each

$$\mathbf{x}_i = (f_1(I,i), f_2(I,i), f_3(I,i), f_4(I,i))^T.$$
$$0 \leq f_k(\cdot,\cdot) \leq 1, \quad k = 1,2,3,4.$$

This makes $X(I)$ a 4-variate vector of length w, where the f_k are the four extracted features per image column.

In order to run the DTW algorithm on two time series $X(I)$ and $Y(J)$ extracted from images I and J, we have to define a local distance function that compares the feature sets at aligned columns. We have chosen to use the square of the Euclidean distance:

$$d(\mathbf{x}_i, \mathbf{y}_j) = \sum_{k=1}^{4} (f_k(I,i) - f_k(J,j))^2. \tag{2}$$

This penalizes large differences between the extracted features more heavily than the Euclidean distance would.

Now the DTW algorithm can be run to determine a warping path between X and Y. The length K of the warping path $((i_1, j_1), \dots, (i_K, j_K))$ biases the determined distance

$$D(X,Y) = \sum_{k=1}^{K} d(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}). \tag{3}$$

When comparing a template series X to others, shorter series would be favored (i.e. produce smaller costs). For this reason, our final matching cost is normalized by the length K of the warping path:

$$\text{matching\_cost}(X,Y) = D(X,Y)/K. \tag{4}$$

In the following section, the column features used for matching will be described.

2.4. Features

The images we operate on are all grayscale with 256 levels of intensity $[0..255]$. Before column features can be extracted from an image, inter-word variations, such as the baseline offset and the skew/slant angles have to be detected and normalized. All of the column features we describe in the following are normalized to the range $[0..1]$. Specific pixel intensity values in an image I (dimensions $h \times w$) are referred to as $I(r,c)$, where r and c indicate the row and column index of the pixel. Our goal was to choose a variety of features presented in handwriting recognition literature (e.g. [2]), such that an approximate reconstruction of a word from its features would be possible.

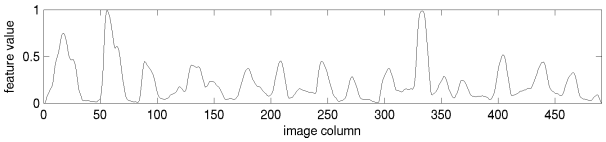
2.5. Projection Profile

Projection profiles capture the distribution of ink along one of the two dimensions in a word image. A vertical projection profile is computed by summing the intensity values[3] in each image column separately:

$$pp(I,c) = \sum_{r=1}^{h} (255 - I(r,c)). \tag{5}$$

Due to the variations in quality (e.g. contrast, faded ink)

(a) original image: slant/skew/baseline-normalized, cleaned.



(b) normalized projection profile.

Figure 4: Original image and projection profile feature.

of the scanned images, different projection profiles do not generally vary in the same range. To make them comparable, the range of the projection profiles is normalized to the range $[0..1]$ which yields $f_1(I,c)$. Figure 4 shows an example projection profile and the original image it was extracted from.

2.6. Word Profiles

Word profiles capture part of the outlining shape of a word. The current word matching algorithm uses upper and lower word profiles: these two features are calculated by going along the upper (lower) boundary of a word's bounding box and recording for each image column the distance to the nearest "ink" pixel in that column. The identification of ink pixels is currently realized using a thresholding technique which we have found to be sufficient for our purposes.

[3] We invert the pixel intensities, because the result is visually more intuitive (peaks for pronounced vertical components in the input word image).

Due to a number of factors, such as pressure on the writing instrument and fading ink, some image columns may not contain ink pixels. The occurrence of such gaps is not consistent for multiple instances of the same word. Therefore, we close these gaps by linearly interpolating between the two closest points where the word profile feature values could be reliably determined.

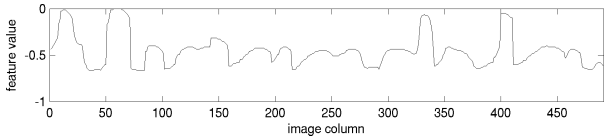


Figure 5: Normalized upper word profile (negative feature value displayed).

The features f_2 and f_3 can be obtained from the upper and lower word profiles by normalizing their maximum range to $[0..1]$. Figure 5 shows an upper word profile feature, generated from the original in Figure 4(a).

2.7. Background/Ink Transitions

So far, the above features represent the distribution of ink in the columns of a word image and the outlining shape of the word. To capture part of the "inner" structure of a word, we chose to record the number of background to ink transitions $nbit(I, c)$ in an image column as the last feature. The range of this feature is normalized with a (conservatively estimated) constant that ensures a range of $[0..1]$:

$$f_4(I, c) = nbit(I, c)/6. \quad (6)$$

With this feature set at hand, we will now demonstrate its effectiveness when used within the proposed DTW matching algorithm (section 2.3).

We tried other features, including Gaussian derivatives, but the above set seemed to work the best.

3. Experimental Results

3.1. Data Sets and Processing

Word matching experiments were conducted on two test sets of different quality, both 10 pages in size. The first set is of acceptable quality, see Figure 6(a)). The second set is very degraded (see Figure 6(b)) - it is difficult even for people to read these documents - and it was used to test how badly the algorithms would perform. A number of algorithms were tested and results are presented on four sets which were constructed as follows:

A: 15 images in test set 1, analyzed in [3].

B: entire test set 1 (2381 images total, 9 do not contain words[4]).

C: 32 images in test set 2, analyzed in [3].

D: entire test set 2 (3370 images total, 108 do not contain words[4]).

The subsets A and C allow us to test algorithms which would otherwise take too long to run on the entire dataset.

Each page in the two test sets was segmented into words using the algorithm described in [6]. The algorithm uses scale-space techniques to determine word boundaries which are then used to extract single word images. For reasons of comparability we used the exact same segmentation results as in [3].

For the matching based on DTW and the shape context run (see below), we normalized the slant and skew of the word images and cleaned the images to remove noise in the background and parts of other words that reach into the bounding box.

Test set	total #queries	$\frac{\#\text{pruned pairs}}{\#\text{total pairs}}$	Recall
A	15	12.71%	90.72%
B	2372	13.57%	71.11%
C	32	13.01%	56.49%
D	3262	14.26%	55.05%

Table 1: Effects of pruning for all analyzed data sets.

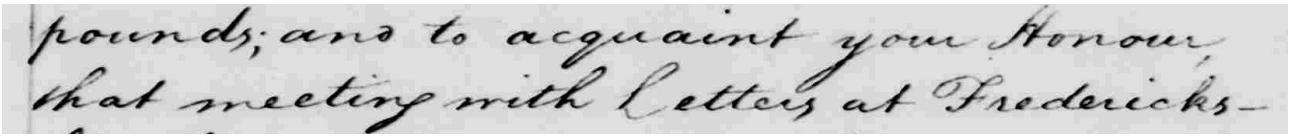
The total number of word pairs, which would otherwise have to be processed by the matching algorithm, was reduced by applying the pruning techniques described in section 2.1. Table 1 shows the effects of pruning on the 4 subsets A, B, C and D. *Pruned pairs* denotes the images left for comparison after pruning, *#total pairs* is the number of query words in the (partial) test set multiplied by the number of words in the enclosing collection (either 2381 or 3370). *Recall* is the proportion of valid matches that remains in the pruned set (100%=no valid matches discarded).

3.2. Evaluation Method

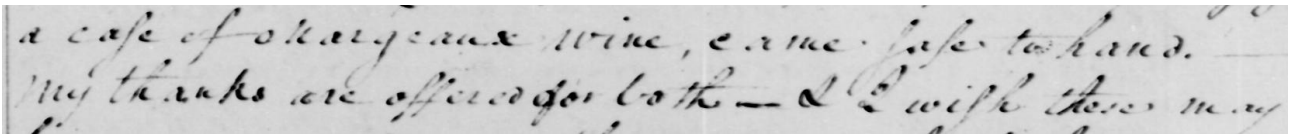
Each word in the data sets was tagged with its ASCII equivalent. In case of segmentation errors, a tag corresponding to all visible characters in the segmented word image was assigned. Based on this annotation, relevance judgments were produced for the data sets. Two word images were considered relevant, if they have the same tags.

To evaluate the word image matching algorithms, we used an information retrieval approach: each image in a data set is viewed as a query which is used to retrieve similar images from the entire collection enclosing the data set (e.g. data set A is enclosed in set 1). Matching the query against other images produces a ranked list of retrieved images, sorted by the matching cost. Using the `trec_eval` program, we calculated average precision scores [11] for all queries in the sets A through D.

[4] These images result from segmentation errors.



(a) acceptable quality (set 1).



(b) significantly degraded (set 2).

Figure 6: Document samples of different quality.

3.3. Results

Table 2 shows average precision results for all data sets obtained with a range of different matching techniques:

1. XOR[3]: The images are aligned to compensate for shear and scale changes, and then a difference image is computed. Difference pixel count determines cost.

2. SSD[3]: translates template and candidate image relative to each other to find the minimum cost (= matching cost) based on the sum of squared differences.

3. SLH[3]: recovers an affine warping transform (using the Scott & Longuet-Higgins algorithm [9]) between sample points taken from the outline of the template and candidate image. Residual between template points and warped candidate points is matching cost.

4. SC [1]: Shape context matching (see section 1.1)

5. EDM[3]: Euclidean distance mapping. In the XOR image, difference pixels in larger regions are weighted more heavily, because they are likely to result from structural differences between the template and the candidate image, not from noise.

6. DTW: the matching technique proposed in this work.

Kane et al. [3] used an evaluation technique that considers each template image as a candidate. Most of the above matching techniques retrieve the template image at rank 1, which biases the evaluation scores. In order to provide a more accurate picture of the actual matching performance, we have re-calculated the average precision scores for test-runs that were available to us in ranked-list (i.e. raw) format. The three right-most columns in table 2 show the average precision scores of three runs, where each query image was not considered a matching candidate. The remaining results in the table were calculated using Kane et al.'s evaluation method.

For data set A, results for all matching algorithms are available. EDM and DTW clearly outperform any of the other techniques. SC was run with a number of sample points proportional to the length of the words being matched, with about 100 sample points for a word like *Alexandria* (see Figure 4(a)). More sample points would probably improve the effectiveness of the technique, but at the cost of further increasing the matching time (for 100 sample points already about 50 seconds, see Table 3).

The DTW algorithm was also run on dataset B (all images used as templates). The other algorithms were too slow to realistically run on this dataset. On set B, the average precision score for DTW is lower than on the smaller enclosed set A. We attribute this effect to the pruning method, which works much better on the smaller set A: while the pruning preserves about 91% of the relevant documents for data set A, it only produces 71% recall on data set B. The lower recall on set B (due to the pruning) then results in a lower average precision score after matching.

For the SC, EDM and DTW techniques we compared the results on the bad data set C. While the performance of SC, EDM and DTW is generally low on this data set, DTW clearly performs better than SC and EDM. DTW also performs similarly on the rest of the data set (51.81% average precision on data set D). This shows that the DTW matching technique is more robust to document degradation than EDM. We would expect the results to be better, if a more careful pruning was applied: after pruning, the recall percentages have already dropped to about 56% for sets C and D (see Table 1). This limits the maximum average precision that is achievable with the matching algorithms.

Algor.	XOR	SSD	SLH	SC	EDM	DTW
time [s]	13	72	121	∼50	14	∼2

Table 3: Run times for the compared algorithms in *Matlab* on a 400MHz machine, including normalization, feature extraction, and similar processing steps.

Test set/Algorithm	XOR	SSD	SLH	SC	EDM	DTW	SC	EDM	DTW
A	54.14%	52.66%	42.43%	48.67%	72.61%	73.71%	40.58%	67.67%	67.92%
B	n/a	n/a	n/a	n/a	n/a	65.34%	n/a	n/a	40.98%
C	n/a	n/a	n/a	48.11%	49.56%	58.81%	9.46%	n/a	13.04%
D	n/a	n/a	n/a	n/a	n/a	51.81%	n/a	n/a	16.50%

Table 2: Average precision scores on all data sets (results for test set A and B have been corrected, XOR: matching using difference images, SSD: sum of squared differences technique, SLH: technique by Scott & Longuet-Higgins [9], SC: shape context matching [1], EDM: Euclidean distance mapping, DTW: dynamic time warping matching). The three rightmost columns show results with the alternate evaluation technique (query images not considered relevant to themselves).

The results show that DTW performs best among the set of algorithms we tried. Comparing the running times of the investigated algorithms (see Table 3) also clearly shows the value of using DTW - it outperforms any of the other techniques. The DTW implementation used here has not been optimized. Rewriting the code could probably improve the performance by a factor of 5 or 10.

4. Summary and Conclusions

We described an approach to matching words using dynamic time warping and showed that this approach produces much better results than a number of other techniques, including shape context matching. Specifically, we showed that on a set of ten pages of good quality, the average precision was around 65% for the technique based on dynamic time warping. The technique is also much faster than the other methods that were examined.

Our future work will focus on improving the accuracy as well as the speed of the techniques used here. Accuracy can be improved by using better pruning techniques as well as using a larger feature set which discriminates words better from each other. Speed can be improved by optimizing our implementation of the dynamic time warping algorithm, as well as looking at related computational techniques to minimize the number of possible matches.

Acknowledgments

We would like to thank the Library of Congress for providing the scanned images of George Washington's manuscripts. We also thank the authors of the shape context algorithm [1] for making their code available online.

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] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

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

[9] 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.

[10] 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.

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