Word Warping for Offline Handwriting Recognition

Douglas J. Kennard, William A. Barrett, and Thomas W. Sederberg
Department of Computer Science
Brigham Young University
Provo, Utah, USA
{*kennard,barrett,tom*}@*cs.byu.edu*

***Abstract*—We present a novel method of offline whole-word handwriting recognition. We use automatic image morphing to compute 2-D geometric warps that align the strokes of each word image with the strokes of word images of training examples. Once the strokes of a given word are aligned to a training example, we use distance maps to compare how similar the two words are. Like 1-D Dynamic Programming (DP) methods, our warp-based method is robust to limited variation in word length and letter spacing. However, due to its 2-D nature, our method is also more robust than 1-D DP methods in handling variations caused by additional inconsistencies in character shape and stroke placement. Although we use DP for coarse alignment, the novel contribution of this paper is not 2-D DP, but morphing to automatically discover an actual 2-D mesh-based warp, followed by the use of distance maps to compute similarity between words. Early results are encouraging. On two datasets (1,000 training and 1,000 test words each), we get 88.77% and 89.33% recognition accuracy for in-vocabulary words. These are increases of 7.89% and 17.16% above the results of a 1-D DP approach.**

***Keywords*-handwritten word recognition; warping; morphing**

I. Introduction

We present a novel offline whole-word recognition method that uses 2-D warping and distance maps to compare words. Our method, "word warping," successfully handles some of the local variation inherent in handwriting such as inconsistent ink thickness and letters that are unevenly spaced, stretched, compressed, or similarly distorted.

For a given pair of images, we create a regularly-spaced rectangular mesh on the first image and a corresponding warp mesh that defines how to push, pull, bend, and stretch the ink of the first image to align it with the ink in the second image (Figure 1b). Aligning the ink allows us to ignore many of the local differences and variations inherent in handwriting and instead compare words at a more structural level. Once the ink is aligned by warping, we use distance maps to quantify how different the words are.

To define the warp mesh used in alignment, we first coarsely align the warp mesh by using 1-D Dynamic Programming (DP) in both the horizontal and vertical directions (Figure 1a). After coarse alignment, we perform a more detailed alignment by using an image morphing algorithm (Section III-E) to increase the mesh resolution and iteratively adjust the control points (vertices) of the warp mesh (Figure 1b). We only use full-thickness word images for the

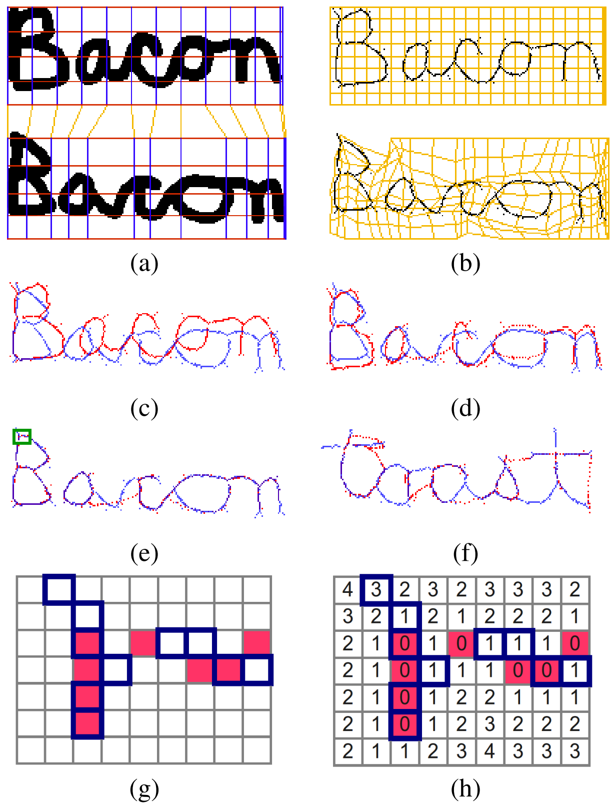

Figure 1. Handwriting recognition by matching two instances of the word "Bacon" using word warping. a) Step 1: Coarse alignment of warp mesh using DP; b) Step 2: Improve warp mesh by morphing; c) Medial axis pixels of each image: red=first instance, blue=second; d) Using only coarse alignment to warp the first instance – not as good as morphing; e) Step 3: Warping with mesh improved by morphing gives good alignment; f) Warping "Bacon" to wrong word "toast" does not align as well. g) Overlay of medial axes from rectangular region in e; h) Step 4: Compute distance map from (red) warped medial axis pixels (0=medial axis). Numbers=distance map with respect to red medial axis. Step 5: Compute word matching cost (normalized sum of blue squares) = "distance" from red medial axis.

coarse alignment. We use medial axis pixels of the words throughout the rest of the process to simplify our morphing algorithm and the distance metric we use to compare words.

II. Related Work

Numerous HR approaches appear in the literature ([4], [8], [13]), including some whole-word recognition methods such as those described by Madhvanath and Govindaraju in [6].

Whole-word approaches exist that use everything from ascenders, descenders, and loops to contour-based features, profile-based features, and graph-based word descriptions.

Rath and Manmatha [10], [9] show that Dynamic Time Warping (DTW) – a 1-D DP method – can be used to match whole words in the context of word-spotting. We see from their work that DTW is robust to some variation in character spacing, width, and shape in the horizontal direction – the direction of the 1-D alignment. Features from two words are aligned using DTW, and the DP cost for the alignment is used as a metric of how different the words are.

We use their DTW method for the coarse alignment of our warp mesh (Section III-C). We also use DTW as the baseline 1-D DP approach for evaluating the performance of our 2-D warping-based recognizer, as described in Section IV. From this evaluation, we see the benefit of moving from 1-D to a recognition method that handles additional 2-D variation.

Pavlidis et al. [7] perform online (not offline) HR by comparing blending costs calculated using the physics based approach to shape blending developed by Sederberg and Greenwood [11] – an algorithm originally used to automatically create smooth graphical blends from one shape to another. Sometimes called shape morphing, the approach models a polygonal shape as a wire that can be bent or stretched into a second shape. The algorithm determines how to manipulate the wire into the second shape using the least amount of work. Singh et al. [12] extend the work of Pavlidis et al. to use shape blending costs to recognize 2-D shapes in general, including a small number of cursive words.

Like shape morphing, image morphing is a graphical technique, but is used to morph one image into another instead of just polygonal shapes. We use principles derived from the work minimization approach to image morphing by Gao and Sederberg [1] in our HR method to improve the warp mesh alignment as described in Section III-E.

Unlike the previous recognition methods that just use DTW cost or shape blending cost as a direct metric of how different words are, we use these methods to align words, but then compute a distinct metric of how different the words are. We describe the metric in Section III-F.

III. Methods

For a given pair of word images, I_0 and I_1, with width/height w_0/h_0 and w_1/h_1, we create corresponding rectangular meshes, M_0 and M_1. Initial control point spacing for M_0 is $max(h_0, 4)$, except for the last row and column of control points, which are placed at $y = h_0 - 1$ and $x = w_0 - 1$. The control points of M_1 (the warp mesh) are not spaced evenly, but instead are coarsely aligned by 1-D DP (Section III-C) and then morphing (Section III-E) is used to adjust them so that the warped medial axis pixels, A'_0, align more closely to the medial axis pixels (A_1) of I_1. We then use D'_0, the distance map from A'_0, to compute $C_{0\to1}$ (Section III-F). $C_{0\to1}$ is the cost to match I_0 to I_1.

I_0, I_1:	The two images being compared
w_0, h_0, w_1, h_1:	Width, height of I_0 and I_1
M_0, M_1:	Meshes defining the 2-D warp from I_0 to I_1
$P^0_{c,r}$:	Control point (vertex) in M_0 at col c, row r
$P^1_{c,r}$:	Control point (vertex) in M_1 at col c, row r
A_0, A_1:	Medial Axis pixels of I_0 and I_1
A'_0:	Pixels of A_0 after being warped (using M_0 to M_1)
D'_0:	Distance map created from A'_0
D_1:	Distance map created from A_1
F_0, F_1:	Feature vectors for DP alignment
$C_{0\to1}$:	Cost of matching I_0 to I_1
$C_{1\to0}$:	Cost of matching I_1 to I_0
$C(I_0, I_1)$:	Total word matching cost between I_0 and I_1

Figure 2. Reference key to symbols and notation.

Since the cost to match I_0 to I_1 is not necessarily the same as the cost to match I_1 to I_0, we repeat the steps with I_0 and I_1 swapped to compute $C_{1\to0}$. The total word matching cost, $C(I_0, I_1)$, is the sum:

$$C(I_0, I_1) = C_{0\to1} + C_{1\to0} \tag{1}$$

Adding the two costs ensures that $C(I_0, I_1)$ is symmetric for a given pair of images regardless of order. $C(I_0, I_1)$ is the distance metric we use for word comparison. We call it "cost" to avoid confusion with distances in distance maps.

Recognition of a word is performed by computing the word matching cost between it and each training example and using the label from the training example resulting in the minimum word matching cost.

A. Preprocessing

We preprocess manually-segmented word images by performing background removal, slant correction, crop/pad, and binarization. Background estimation for background removal is computed with a large median kernel as described in [2]. After background removal, a global threshold value for each page is determined for later binarization using a method described in [3]. Slant correction consists of a horizontal image shear after estimating the slant angle over the central region of each page image. The angle estimation uses ink runlengths accumulated into a histogram based on angle bins. Baseline estimation is used to determine whether to pad the top or bottom of the image, and the image is cropped to the left-most / right-most ink pixel after the slant removal. After all other preprocessing, the word image is binarized using the previously selected threshold.

B. Distance Map and Medial Axis

We compute distance maps for bitonal images using a forward-backward algorithm. Each pixel in the resulting distance map contains the Manhattan distance (in number of pixels) to the nearest edge of an ink component. The greater the distance from ink, the higher the value of the pixel in the distance map. Values within an ink component are zero (on the border with the background) or negative (within the component) – progressively increasing in magnitude as the center of the ink component is approached.

Medial Axis pixels are those in the distance map with values less than or equal to zero not having any 4-connected neighbors more negative than themselves. We remove from the result any pixels for which the North, Northwest, and West neighbors are all also medial axis pixels.

C. Dynamic Programming for Coarse Mesh Alignment

For horizontal alignment of M_1, we use the DTW algorithm described in [10]. Feature vectors F_0 and F_1 are computed from the normalized ink profile, upper word profile, lower word profile, and background to ink transition counts of the respective word images, I_0 and I_1. The DTW function to build the DP alignment table is:

$$D(i,j) = min \left\{ \begin{array}{c} D(i-1,j) \\ D(i,j) \\ D(i,j-1) \end{array} \right\} + d(i,j), \tag{2}$$

where d(i,j) is the cost to align F_0(i) with F_1(j), and is defined as:

$$d(i,j) = \sum_{k=1}^{4} (F_0(i,k) - F_1(j,k))^2, \tag{3}$$

where k is the index to access the four features in the vector at the alignment position (profile, upper/lower indention profile, transition count). We also use the same Sakoe-Chiba band DP constraint with radius 15 as the authors of [10].

The alignment of F_0 and F_1 is found by following the DP path backward through the DP table when the DTW algorithm is complete. The alignment is used to map x-coordinates from M_0 to the corresponding x-coordinate to be assigned to the corresponding control point in M_1. The same is done for y-coordinates using the DTW result for vertical alignment except that we only use a single-dimensional feature vector – just the ink profile of the word images (projected onto the vertical axis). Therefore, in Equation 3, the summation is only for $k = 1$.

D. Warping Coordinates

Since quads in M_0 are rectangular, the s, t coordinate within a quad (s and t having range $[0, 1]$) is easy to calculate for any point x, y. The warped coordinate x', y' is then computed by bilinear interpolation of s, t within the vertices of the corresponding quad of the warp mesh, M_1.

E. Morphing for Warp Mesh Improvement

In image morphing, the start and end mesh define a warp from one image to another. Interpolating positions and pixel colors at evenly-spaced time slices between the start and end results in a series of images forming a graphical morph from one image to the other. The work minimization approach to image morphing [1] automatically generates the end mesh by iteratively *improving* the mesh – adjusting its control points to reduce the overall morph cost according to a work equation – and *refining* the mesh – subdividing it into more detailed quads. The work equation includes costs for work due to angle change, stretching, and pixel color change.

We adapt the morphing algorithm to the application of aligning handwritten words. Our algorithm is as follows:

```
refine_count= 0; $m =$ max$(4, h_0/4)$   // ($h_0$ = height)
while $m > 16$
   $m = m/2$;  refine_count=refine_count+1
for mesh_level=1 to refine_count
   for imp=1 to improve_count          // (we use 3)
     // improve:
       for each $P^1_{c,r}$
          $x, y = P^1_{c,r}$
          $min = placement\_cost_{x,y}$   // (Equation 4)
          $XY_{min} = x, y$
          for each $x, y$ in search area of $P^1_{c,r}$
             if $placement\_cost_{x,y} < min$ then
                $min = placement\_cost_{x,y}$
                $XY_{min} = x, y$
          $P^1_{c,r} = XY_{min}$        // (update control point)
   if mesh_level < refine_count
     // refine:
       increase resolution of $M_0$, $M_1$ by factor of 2
```

The *refine* step doubles mesh resolution by adding control points at the midpoints of each quad/edge in M_0 and M_1.

In the *improve* step, each control point, $P^1_{c,r}$, is in turn moved to the lowest cost position within its current search area. The search area is constrained to a rectangular region surrounding the current position of the control point. The region extends 0.4δ in each direction, where δ is the current control point spacing in M_0 (δ gets halved every time a *refine* occurs). The search area is also constrained by the control points around it. For example, $P^1_{c,r}$ cannot go above any of the 3 control points above it in its 8-neighborhood.

The cost of placing $P^1_{c,r}$ at any given search position x, y within the search area is:

$$placement\_cost_{x,y} = \frac{1}{n+1} \sum_{k=1}^{n} D_1(A'_0[k]), \tag{4}$$

where A'_0 are warped using the search position as the position of $P^1_{c,r}$ in M_1, $n = \|A'_0\|$, and $D_1(A'_o[k])$ is the value in D_1 at the position of the k^{th} warped medial axis point in A'_0. In effect, the cost of placing $P^1_{c,r}$ at this search position is the average distance the medial axis points of I_0 would be from the nearest medial axis pixels of I_1 if we were to place $P^1_{c,r}$ at this search position.

We actually do not use the entire set A'_0 of medial axis pixels during cost calculation for search positions. Since only the pixels within the four mesh quads sharing control point $P^1_{c,r}$ as a vertex move when the control point is adjusted, only the costs associated with those points will affect the cost at any given search position for that control point. To speed up processing, we ignore all A'_0 points outside of the four adjacent quads.

F. Word Matching Cost

After M_1 has been aligned using DP and morphing, we compute the warped medial axis, A'_0, of I_0 (red-shaded pixels in Figure 1g). We then compute the distance map, D'_0, of the warped medial axis A'_0 (Figure 1h). Most of the A'_0 pixels should be closely aligned to pixels of A_1 due to morphing. What tells us if the words are actually similar or not is if the pixels of A_1 (blue-bordered pixels in Figure 1h) also align well to pixels of A'_0, or whether their values in the distance map are high, suggesting that the words are not similar. We compute $C_{0\to1}$, the cost to match I_0 to I_1, as:

$$C_{0\to1} = \frac{1}{\|A_0\| + 1} \sum_{i=1}^{\|A_1\|} D'_0(A_1[i]), \tag{5}$$

where $D'_0(A_1[i])$ is the value in D'_0 of the location of the i^{th} medial axis pixel in A_1.

IV. Experiments

We perform experiments on two datasets of labeled word images. The first dataset consists of words from a set of 20 pages of George Washington's manuscripts [5]. The second consists of words from pages of Jennie Leavitt Smith's diary[1], downloaded from the "Mormon Missionary Diaries" online collection of the Brigham Young University Harold B. Lee Library, available at http://www.lib.byu.edu/dlib/mmd/. We manually segment and label each word to provide ground truth for our experiments.

For each dataset, we select the first 1,000 word images as training examples for which the recognition system is allowed to look at the labels. We use the next 1,000 words (which are not used as training examples) as test data. We compare each test word with the training words and assign it the label from the training word that it most closely matches. This is done both using our 2-D word warping method and also using just the 1-D Dynamic Time Warping alignment cost [10]. We also record the word warping results when using only coarsely-aligned meshes without morphing.

We assess the recognition accuracy of each method by comparing the ground truth labels with the labels assigned by the recognizer. Recognition accuracy is calculated as the number of test words labeled correctly by the recognizer (the number given the same label as its ground truth), divided by the total number of test words. The string comparison between the label and ground truth is case-sensitive.

Since many of the test words are *Out of Vocabulary* (OoV) words, meaning no training examples have the same label as their ground truth, we also report the recognition accuracy with respect to the number of in-vocabulary words (total test words minus the number of OoV test words).

[1]Our preprocessed Smith dataset word images are available upon request.

Table I
Experimental Results – Word Recognition Accuracy

Washington Dataset - 1,000 test words (748 in-vocabulary)		
Method	Total Accuracy (# correct / # possible)	In-Vocab Accuracy (# correct / # possible)
DTW (1-D DP)	60.50% (605 / 1000)	80.88% (605 / 748)
Word Warping (only coarse aligned)	65.30% (653 / 1000)	87.30% (653 / 748)
Word Warping (morphing aligned)	**66.40%** (664 / 1000)	**88.77%** (664 / 748)

Smith Dataset - 1,000 test words (787 in-vocabulary)		
Method	Total Accuracy	In-Vocab Accuracy
DTW (1-D DP)	56.80% (568 / 1000)	72.17% (568 / 787)
Word Warping (only coarse aligned)	64.40% (644 / 1000)	81.83% (644 / 787)
Word Warping (morphing aligned)	**70.30%** (703 / 1000)	**89.33%** (703 / 787)

V. Results and Discussion

Our word warping method is noticeably more accurate than DTW (the baseline 1-D DP method) on both datasets (Table I). Even without using morphing to improve the warp mesh, word warping with coarsely-aligned meshes results in an increase in recognition accuracy of 6.42% for in-vocabulary words with the Washington manuscripts dataset and 9.66% with the Smith diary dataset. Recognition is even better when we include the morphing step. For the Washington dataset, in-vocabulary accuracy is 88.77%, an increase of 7.89% from the baseline (DTW). For the Smith dataset, we see a larger improvement of 17.16% to 89.33%.

Morphing only contributes 1.47% to the accuracy of the Washington dataset, however, it contributes 7.5% to the accuracy of the Smith dataset. We observe that the the Washington penmanship is exceptionally consistent but there is more variation in the Smith dataset, requiring better alignment in order to recognize words. We are encouraged by this result because it suggests that word warping with morphing is adept at handling local variation and should generalize to datasets with multiple authors. This ability to handle variation may even allow us to use synthetically-created training data to improve the OoV recognition accuracy. Figure 3 shows our medial axis alignment using morphing.

Many of the recognition errors that we see are minor, such as differences in case, single letters, or word endings (Figure 4a–4c). Some errors are more blatant (Figure 4d). For many errors, the correct match is ranked very near the

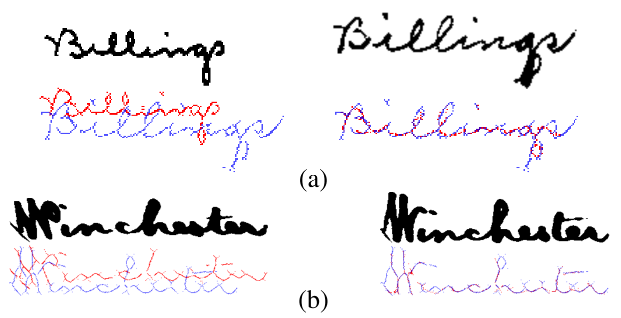

Figure 3. Alignment of medial axes. a) Top: two occurrences of the word "Billings" (Smith dataset); Bottom: corresponding medial axes before alignment (left) and after (right). b) "Winchester" (Washington dataset).

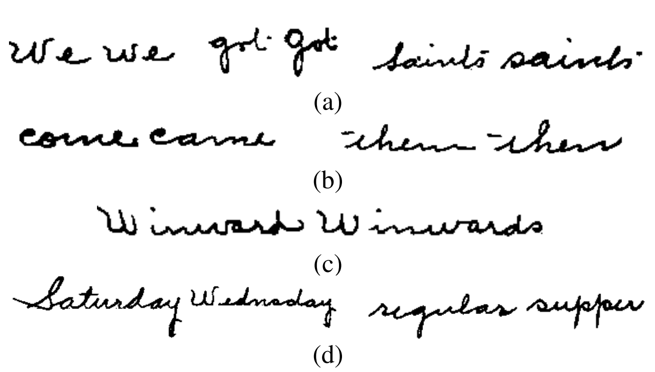

Figure 4. Examples of recognition errors (Smith dataset). Test words followed by the erroneous best match. a) Errors only because of capitalization differences. b) Very similar words: "come" vs. "came" and "them" vs. "then." c) "Winward" vs "Winwards." d) Some more obvious errors.

top (Table II). In fact, the correct result is ranked in the top 3 matches more than 94% of the time for both datasets.

VI. Conclusion

We have presented a 2-D warping method for comparing words to each other for offline handwriting recognition. Our method takes advantage of 2-D warping to get better word matching results. Our early tests on this method are encouraging, showing noticeable improvement over 1-D DP methods.

Table II
Correct Answer In Top-N Results

Washington Dataset - 1,000 test words (748 in-vocabulary)				
Method	Top-1	Top-3	Top-5	Top-10
DTW (1-D DP)	80.88% (605 / 748)	89.17% (667 / 748)	91.58% (685 / 748)	93.98% (703 / 748)
Word Warping	**88.77%** (664 / 748)	**94.52%** (707 / 748)	**96.26%** (720 / 748)	**96.93%** (725 / 748)

Smith Dataset - 1,000 test words (787 in-vocabulary)				
Method	Top-1	Top-3	Top-5	Top-10
DTW (1-D DP)	72.17% (568 / 787)	82.21% (647 / 787)	86.02% (677 / 787)	90.98% (716 / 787)
Word Warping	**89.33%** (703 / 787)	**94.28%** (742 / 787)	**94.79%** (746 / 787)	**96.82%** (762 / 787)

References

[1] Peisheng Gao and Thomas W. Sederberg. A work minimization approach to image morphing. *The Visual Computer*, 14:390–400, 1998.

[2] Luke A. D. Hutchison and William A. Barrett. Fourier-Mellin registration of line-delineated tabular document images. *International Journal on Document Analysis and Recognition (IJDAR)*, 8:87–110, Jun. 2006.

[3] Douglas J. Kennard and William A. Barrett. Separating lines of text in free-form handwritten historical documents. In *International Workshop on Document Image Analysis for Libraries (DIAL)*, pages 12–23, Lyon, France, Apr. 2006.

[4] Alessandro L. Koerich, Robert Sabourin, and Ching Y. Suen. Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications*, 6:97–121, 2003.

[5] Victor Lavrenko, Toni M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Proc. of the Int'l Workshop on Document Image Analysis for Libraries (DIAL)*, pages 278–287, Palo Alto, CA, Jan. 2004.

[6] Sriganesh Madhvanath and Venu Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(2):149–164, Feb. 2001.

[7] Ioannis Pavlidis, Rahul Singh, and Nikolaos P. Papanikolopoulos. On-line handwriting recognition using physics-based shape metamorphosis. *Pattern Recognition*, 31(11):1589–1600, 1998.

[8] Réjean Plamondon and Sargur N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(1):63–84, Jan. 2000.

[9] Toni M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *7th International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 218–222, Edinburgh, Scotland, Aug. 2003.

[10] Toni M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 521–527, Madison, Wisconsin, Jun. 2003.

[11] Thomas W. Sederberg and Eugene Greenwood. A physically based approach to 2-D shape blending. *Computer Graphics*, 26(2):25–34, Jul. 1992.

[12] Rahul Singh and Nikolaos P. Papanikolopoulos. Planar shape recognition by shape morphing. *Pattern Recognition*, 33:1683–1699, 2000.

[13] Tal Steinherz, Ehud Rivlin, and Nathan Intrator. Offline cursive script word recognition — a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 2:90–110, 1999.