Part-Structured Inkball Models for One-Shot Handwritten Word Spotting

Nicholas R. Howe
Smith College
Northampton, MA 01063
Email: nhowe@smith.edu

***Abstract*—Many document collections of historical interest are handwritten and lack transcripts. Scholars need tools for high-quality information retrieval in such environments, preferably without the burden of extensive system training. This paper presents a novel approach to word spotting designed for manuscripts or degraded print that requires minimal initial training. It can infer a generative word appearance model from a single instance, and then use the model to retrieve similar words from arbitrary documents. An approximation to the retrieval statistic runs efficiently on graphics processing hardware. Tested on two standard data sets, the method compares favorably with prior results.**

I. Motivation

When people write by hand, they produce forms that vary in shape even between two instances of the same word. The differences are most pronounced when comparing the handwriting of different individuals, but even for a single writer some variation will be observed. This makes identifying instances of a target word in handwritten collections a challenging form of pattern recognition. Existing approaches to *word spotting* for handwritten documents often require significant training before retrieval can take place, and the performance still leaves room for improvement. This paper describes a novel approach to word spotting based on recent developments in *part-structured models* that improves retrieval precision with little or no training required.

The method described in this paper has not previously appeared in the scientific literature, but it weaves together elements of several established lines of research. It represents an example of *one-shot learning*, a category of pattern recognition that attempts to learn a concept class from a single positive example [1]. The learning process induces a generative model for word appearance similar to one described by Revow et al. [2]; however the details differ so as to allow for much greater efficiency in spotting applications. The framework in this case takes the form of a part-structured model or *PSM* [3], but somewhat different in form from those recently applied with great success to general-purpose object recognition [4]. One might also loosely view the models as performing a sort of flexible Hausdorff matching [5] using a deformable template.

The method also bears some resemblance to several other techniques in either implementation or spirit. The dynamic programming implementation resembles methods in level sets and snake optimization [6], although work in those areas did not directly influence this effort. The approach also bears some comparison to dynamic time warping [7], although DTW allows for only one-dimensional deformation while the method given here handles two. Finally, comparisons are possible with branch-and-bound methods used in symbol spotting, which also search for likely matches of a model shape to an image [8].

The handwriting recognition and word-spotting fields both sport results notable as points of comparison to the approach described herein. Although word spotting does not require full transcription of a manuscript, a sufficiently accurate recognition engine can produce a transcript with which word spotting becomes trivial. However, training recognition systems can be more involved than spotting mechanisms, and the recognition process is at least as error-prone as word spotting. Recent recognition systems suitable for comparison include Frinken et al.'s method based upon recurrent neural networks [9] and the related work by Fischer et al. [16]. Results for word spotting have also been reported by Lavrenko et al. [10], and Howe et al. [11], among others.

The remainder of this paper includes a section describing the new method, another section describing experiments performed, and concludes with a discussion of strengths and weaknesses.

II. Method

Any generative model of character appearance must account for expected variability by incorporating flexibility to match observed deformations. Doing so in computationally tractable form presents a serious challenge, since allowing flexibility usually implies adding additional degrees of freedom in the model. The solution adopted here is to heavily exploit dynamic programming and to choose a model that allows computation to be reused whenever degrees of freedom are added. The basic model consists of balls of ink connected via springlike potentials, and arranged in a tree structure whose *a priori* minimum energy configuration conforms to the shape of the query word or symbol (see Fig. 1). This model draws from several strands of previous work. In one sense, it corresponds to a simplified version of the model proposed by Revow et al. That work used "ink generators" arranged in a spatial configuration defined by B-splines [2], as opposed to the simpler displacement model used here. Computationally, the current work takes inspiration from the part-structured models introduced by Felzenszwalb et al. for object recognition [4]. In contrast to the bulk of the work in that area, the models here employ a far simpler structure at each node and compensate with a much deeper node tree.

Mathematically, the model consists of a set of nodes $Q = \{q_i | i = 1...m\}$ related in a tree structure with a single root

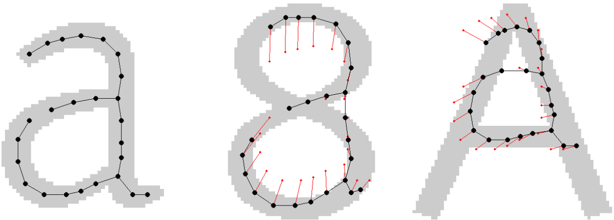

Fig. 1. Part-structured ball-and-spring model of the letter A, left. The shape that generated this model appears in gray. Parent-child relations are indicated by black lines. Note the break in the loop necessary to maintain a tree structure. Remaining images show configurations of the model deformed to match different observations. The result compromises between the rest configuration and the observed ink.

q_r, $1 \leq r \leq m$. Without loss of generality, assume $r = 1$ for convenience. Define the following notation: $q_{i\uparrow}$ denotes the parent of node q_i (undefined for q_r) and $q_{i\downarrow j}$ denotes the jth child of node q_i, if it exists. Furthermore, $Q_i^{\downarrow}$ denotes the set of all children of q_i.

A simple process derives a model of the form just described from a binary word image: First, thin the image to a medial skeleton of single-pixel width using a suitable algorithm[12]. Select all points that are endpoints or junctions of the skeleton to populate the set of nodes. Greedily add to these selections additional skeleton points as far away as possible from existing selections but no more than distance d, such that when finished each skeleton point lies at most $d/2$ away from one or more selected points. The parameter d should be set to some significant fraction of the mean stroke width (80% for example) so that disks of ink placed at their centers will overlap to approximate the original image. Each selected point becomes the default location of a node. The tree structure is induced by selecting the node closest to the center of mass as root, and greedily adding whichever node is nearest to the growing tree, as a child of the closest node already added. Fig. 1 shows at left a model generated using the algorithm just described.

A. *Configurations*

A *configuration* of the model associates each node q_i with a 2d coordinate vector $\vec{v}_i$ representing its position in the image plane; $\vec{V}$ represents the entire $2 \times m$ matrix of positions. Each node q_i also has an associated *default offset* $\vec{t}_i$ which specifies its position relative to its parent in the minimum energy configuration or *rest configuration*, and $\vec{T}$ represents the entire $2 \times m$ matrix of default offsets. The default offset is intrinsic to a model and remains unchanged regardless of the current configuration. Figure 1 shows a model in its rest configuration and two other configurations that incorporate some deformation to better match an observed ink distribution.

The energy of a particular configuration depends upon the intrinsic parameters of the model Q, its configuration C, and observations Ω (image values in the neighborhood of the configuration points). Broadly speaking it consists of two terms, one describing the deformation of the spring connections in the model relative to their rest configuration, and the other describing the match to observed areas of ink. Although not strictly derived from Bayes' Law, the first term may be seen to approximate the negative log prior probability of the model configuration, while the second term approximates the negative log posterior probability of the observations given the model.

$$E(Q, C, \Omega) = E_\xi(Q, C) + \lambda E_\omega(C, \Omega) \tag{1}$$

The deformation energy E_ξ takes the form of a springlike potential at each parent-child link, centered at its default offset. Expressing the spring constant as $\frac{1}{\sigma_i^2}$ leads to an interpretation of the deformation energy as a log-likelihood, where the probability distribution for each position vector is a 2D Gaussian with variance σ_i. The total deformation energy may be defined as the sum of all spring energies.

$$E_\xi(Q, C) = \sum_{j=2}^{m} \frac{\left\|(\vec{v}_j - \vec{v}_{j\uparrow}) - \vec{t}_j\right\|^2}{2\sigma_j^2} \tag{2}$$

This is equivalent to the following recursive formulation:

$$E_\xi(Q, C) = E_\xi^{(1)}(Q, C) \tag{3}$$

$$E_\xi^{(i)}(Q, C) = \sum_{j:q_j \in Q_i^{\downarrow}} \frac{\left\|(\vec{v}_j - \vec{v}_i) - \vec{t}_j\right\|^2}{2\sigma_j^2} + E_\xi^{(j)} \tag{4}$$

Although σ_j could vary for each node, in practice all nodes use the same value, $\sigma_j = 1$.

The observation energy E_ω may take many forms. Previous applications of part-structured models use complex forms such as the output of support-vector machines trained as part recognizers, which have demonstrated high accuracy for detection of diverse objects in photographic images[4]. Although a similar approach could also be applied to handwriting recognition, developing the necessary part recognizers requires extensive training and computational resources. Here the goal is simplicity and computational tractability, leading to a much more basic formulation. On the other hand, the structural complexity already embedded in E_ξ can compensate for the simplification of E_ω.

Bayes' Law suggests that E_ω should correspond to the negative log posterior probability of the observations given the model configuration. Observations containing ink in the vicinity of each node, and background elsewhere, should have high probability and thus low E_ω. The formulation described below captures only the first part of this prescription; it does not contain any terms accounting for image observations away from the nodes. Nevertheless it proves a highly useful approximation because it supports efficient computation algorithms, and its indifference to the background context can actually prove advantageous in applications with noise and/or clutter surrounding the handwritten text. The second half of the prescription can be addressed heuristically by performing two-way matching: plausible candidate matches of template to observation should be confirmed by verifying that the match of observation to template also works. This section considers only the one-way match; details of the two-way confirmation appear with the experimental description in Section III.

Although handwriting stroke width can occasionally convey information, more often it acts merely as yet another source of random variation thwarting accurate model matching. The formulation for E_ω therefore works not with an original document image I, but with a processed version that has undergone binarization [13], then thinning [12] to a set of skeleton pixels S. It assigns low observational energy to configurations where node locations coincide with skeleton pixels, and high energy to configurations where the nearest observed skeleton pixels are far from the node locations.

$$E_\omega(\vec{V}) = \sum_{i=1}^{m} \min_{\vec{s}\in S} \frac{1}{2\sigma_i'^2}\|\vec{s}-\vec{v}_i\|^2 \tag{5}$$

Equation 5 represents another springlike energy based upon the square of the directed chamfer distance [14] from the set of node positions $\{v_i | 1 \le i \le m\}$ to the set of skeleton pixels S. As before, $\sigma_i' = 1$ in all experiments.

B. Energy Minimization

The energy expression developed in the previous section is motivated in part by implementation considerations. In fact, the form chosen permits the efficient evaluation of the minimum-energy configuration at all possible locations in an image via dynamic programming. Furthermore, the algorithm is amenable to parallel implementation and can run on a commodity graphic processing unit (GPU). This section gives the details of the implementation.

Felzenszwalb & Huttenlocher describe the *generalized distance transform* (GDT) as a useful extension of the ordinary distance transform, computable in linear time in the number of pixels[15]. Given a k-dimensional scalar field W, the generalized distance transform Γ_W embodies a related scalar field that optimizes between the value of W and some function of the distance to the location where that value occurs.

$$\Gamma_W(\vec{v}) = \min_{\vec{u}\in\mathbb{R}^k}\left[W(\vec{u}) + D(\vec{u},\vec{v})\right] \tag{6}$$

This paper uses $D(\vec{u},\vec{v}) = \frac{1}{2\sigma^2}\|\vec{v}-\vec{u}\|^2$ throughout, with $\sigma = 1$ in all cases. Eqn. 5 may be written in terms of a GDT $\Gamma_{\tilde{S}}$, with $\tilde{S}$ zero where S is 1 and infinite elsewhere.

$$E_\omega(\vec{V}) = \sum_{i=1}^{m} \Gamma_{\tilde{S}}(\vec{v}_i) \tag{7}$$

Felzenszwalb & Huttenlocher employ the GDT in their paper introducing the concept of part-structured models[3], and this work follows theirs closely. Consider the recursive formula for the deformation energy in Eqn. 4. When combined with Eqn. 5, it yields a recursive formula for the energy as a whole.

$$E^{(i)}(Q,C,\Omega) = \lambda\Gamma_{\tilde{S}}(\vec{v}_i) + \sum_{j:q_j\in Q_i^{\downarrow}} \frac{\left\|(\vec{v}_j - \vec{v}_i) - \vec{t}_j\right\|^2}{2\sigma_j^2} + E^{(j)} \tag{8}$$

Now suppose that the model Q and observations Ω are known and fixed, and one seeks to compute the minimum possible energy along with the configuration that achieves it. In particular, consider constraining the root node position v_1 to an arbitrary vector $\vec{v}$ and minimizing the energy over all possible descendent node configurations $v_2 ... v_m$. If this is performed for every possible root node location, it defines a scalar field of minimum constrained-root energies $\mathcal{E}$.

$$\mathcal{E}(\vec{v}) \equiv \min_{C|v_1=\vec{v}} E(Q,C,\Omega) \tag{9}$$

Subtree energy fields $\mathcal{E}^{(i)}$ are defined by analogy, fixing the subtree root and minimizing over all possible descendant node configurations. A recursive formulation allows for efficient computation. For nodes without children, observations alone determine the energy:

$$\mathcal{E}^{(i)}(\vec{v}) = \lambda\Gamma_{\tilde{S}}(\vec{v}) \text{ if } Q_i^{\downarrow} = \varnothing \tag{10}$$

For nodes with children, one must include the possible child configuration energies in the minimization.

$$\mathcal{E}^{(i)}(\vec{v}) = \lambda\Gamma_{\tilde{S}}(\vec{v}) + \sum_{j:q_j\in Q_i^{\downarrow}} \min_{\vec{v}_j}\left[\frac{\left\|(\vec{v}_j - \vec{v}) - \vec{t}_j\right\|^2}{2\sigma_j^2} + \mathcal{E}^{(j)}(\vec{v}_j)\right] \tag{11}$$

The second term may also be computed using a GDT:

$$\mathcal{E}^{(i)}(\vec{v}) = \lambda\Gamma_{\tilde{S}}(\vec{v}) + \sum_{j:q_j\in Q_i^{\downarrow}} \Gamma_{\mathcal{E}^{(j)}}(\vec{v} + \vec{t}_j) \tag{12}$$

Note that Eqn. 12 can be computed over entire image pixel grids, where the terms in the summation are themselves grid values shifted by t_j with interpolation if necessary, followed by a GDT. Such a grid computation can be used as a detector. Low energy values correspond to root locations of a high probability configuration, so that finding the energy minima equates to detecting plausible configurations of the modeled text. Computing $\mathcal{E}^{(i)}$ using Eqn. 10 requires a single distance transform. Computing $\mathcal{E}^{(i)}$ using Eqn. 12 requires an interpolated translation and a distance transform for each descendant of q_i. Both the translation and the distance transform are linear computations in the number of pixels n. Thus computing $\mathcal{E}$ takes $\mathcal{O}(mn)$ time on a single processor.

If sufficient resources are available the translation and distance transform can be parallelized for each pixel, making $\mathcal{E}$ computable in $\mathcal{O}(m)$ time. Translation and interpolation are easily parallelized on a GPU. The single-processor GDT algorithm described by Felzenszwalb & Huttenlocher[15] is not well suited to current GPU architectures because it generates unpredictable memory access patterns. However, a simple approximation algorithm that simply computes Eqn. 6 for all pixels within a neighborhood of square radius r parallelizes well and works adequately. So long as $r > \max_{j=1}^{m} t_j$ this approximation to the generalized distance transform will accurately compute all low-energy configurations, which are the only ones of interest. On current hardware (NVIDIA GeForce GTX 480), matching a word model to a high-resolution (8 megapixel) page image takes on the order of one second, depending on the number of nodes in the model, compared with a minute or more on a single CPU core.

The value of λ in Equation 1 is a free parameter of the model, governing the tradeoff between configurations that minimize the configuration energy $E_\xi(Q,C)$ and those that minimize the observation energy $E_\omega(C,\Omega)$. It is set to 2.0 in all the experiments for historical reasons. Future work should investigate optimization on this parameter.

C. Recovering Configurations

Eqn. 12 gives a formula for computing the minimal configuration energy across a document image for any part-structured model of the form previously described. It is often useful to know the exact configuration (or one possible configuration if the solution is not unique) corresponding to that energy. This can be recovered using a backtracking technique analogous to those used in other forms of dynamic programming. During computation of the generalized distance transform, the algorithm must keep track of the vector u that gave the minimum value in Eqn. 6.

$$\Lambda_W(\vec{v}) = \arg\min_{\vec{u}\in\mathbb{R}^k} [W(\vec{u}) + D(\vec{u},\vec{v})] \tag{13}$$

Now beginning from a particular root location v_1 the position of its children may computed, and so on until all v_i have been found.

$$\vec{v_{i\downarrow j}} = \vec{v}_i + \vec{t_{i\downarrow j}} + \Lambda_{\mathcal{E}^{i\downarrow j}} \tag{14}$$

Figure 2 visualizes the matching process for a concrete example. At left the first row shows the original character image, with the thinned skeleton superimposed. The next image to the right shows the matching energy for a single node, $\Gamma_{\tilde{S}}$, which can match at any of the skeleton points. The next image shows $\mathcal{E}^{(i)}$ for a subtree of the model containing the six nodes at the top of the letter. This subset of nodes matches the image fairly well at three spots, as indicated by the three dark areas in $\mathcal{E}^{(i)}$; the three corresponding subtree configurations appear to the left in the second row. The rightmost image shows $\mathcal{E}$, the matching energy for the entire model, above the model corresponding configuration. The single deep minimum reflects the fact that there is only one good configuration for a full model 'a' on this image.

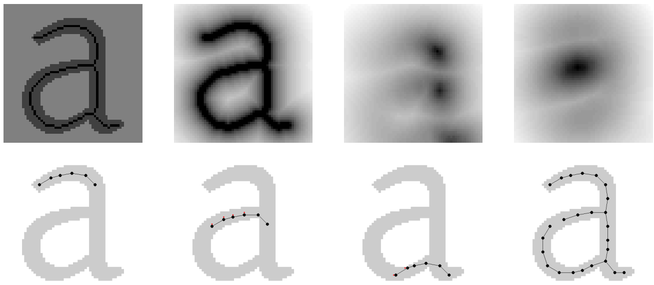

Fig. 2. Partial and complete matches of a PSM to observations.

III. Experiments

The method is validated on several established data sets. The George Washington (GW20) set comprises 20 pages from the official correspondence of George Washington [10]. Some of the images contain vertical rule lines which are removed in a preprocessing stage. The Parzival set comprises 47 pages from a medieval manuscript. Word-spotting results for both have appeared recently [16] and the images are available to researchers for download. Although GW20 includes the writing of several of Washington's secretaries, both show less variation in handwriting style than many multi-writer collections.

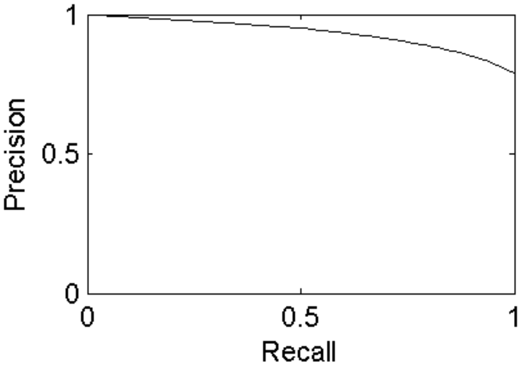

Fig. 3. Mean precision vs. recall for GW20 over all query words.

The public distribution of both data sets include word segmentations. Fitting the PSM does not rely on the segmentation, but it simplifies the experimental procedure to record the lowest configuration energy over all points within the test word bounding box $BB(y)$.

$$M(x,y) = \min_{\vec{v}\in BB(y)} \mathcal{E}_x(\vec{v}) \tag{15}$$

This will identify good matches of the query word x to the test word y, but also allows subset matches such as 'and' to 'Alex**and**ria'. The reverse match $M(y,x)$ must therefore be computed to confirm each candidate match. The final symmetric match score M^* takes the larger of the two one-way matchings.

$$M^*(x,y) = \max(M(x,y), M(y,x)) \tag{16}$$

For a given query word, test words are retrieved in order of ascending M^*. Figures 3 and 4 show the resulting mean precision vs. recall curves for each collection. These are averages over all the words that appear at least once in both the training and test sets, using the same train/test split as Fischer et al. [16].[1] Because word models vary in their response levels, the averaged curves are generated using response rank order without maintaining a global threshold consistency. Also, words that appear more than once in the training set allow for spotting with multiple models, where the match score of a given target word is taken as the best match over all available models. The mean precision values for the two experiments are 93.4% and 88.2% respectively. These exceed the best prior results for GW20 (84%) and approach the best for Parzival (94%), using the same folds and keywords [9].

Both GW20 and Parzival do include many words appearing as singletons in the training set, and the resulting model nevertheless does well at identifying the corresponding word in the test set. This accounts for the high precision at 100% recall, which at 78.9% and 68.4% respectively is much higher than the level seen in prior work [9]. Figure 5 shows some sample matches for a singleton query from the GW20 test set.

[1] Since the part-structured models require no extra data for validation, the training and validation sets are combined.

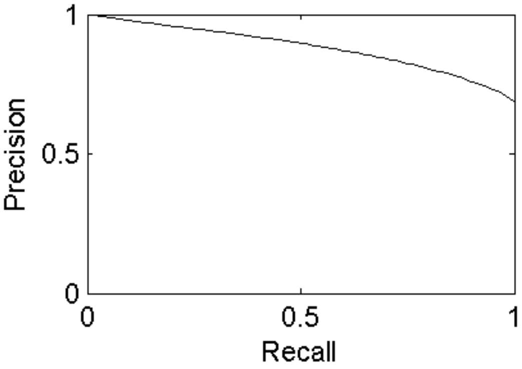

Fig. 4. Mean precision vs. recall for Parzival over all query words.

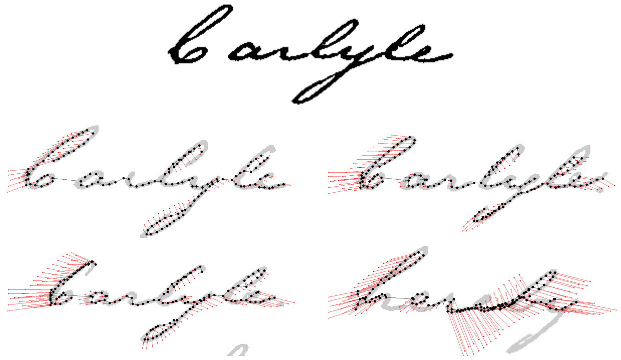

Fig. 5. Sample GW20 retrievals for singleton query word 'Carlyle' (top). The three appearances in the test set have the best three match scores. The next closest word is 'hereby'. Red lines show the deformation of the model.

IV. Discussion

Part-structured models offer several attractive features for word spotting. Because they are generated automatically from one or more query images, they do not require large training sets to perform well. They need not be applied only at the word level; both fragments of words and multi-word phrases can also serve. Although using multiple word instances as a query may add some robustness to variation in form, the method performs well even with a single example. The results presented in Section III exceed the best prior performance on GW20 by a considerable margin.

The method presents a few drawbacks, although none insurmountable. Computation speed is fast enough for small scale searches, taking on the order of one second per full scanned page. Using the technique with larger collections will require either prescreening to reduce the search space, or caching of precomputed results. In addition, part-structured models perform poorly in situations where the query and the target word vary in scale by more than about 20% in either direction. When such size variation appears in the collection to be searched, it will be necessary to match scales by expanding or reducing the model before computing the spotting score. Adjusting the model scale is computationally simple, but the best method for estimating the proper scale factor remains a matter for future work. Finally, a one-shot PSM will struggle in cases where the query and target words differ greatly in underlying letter form, as may be seen with multiple-writer data. More research is necessary to handle such conditions. For example, multiple query examples may be required to cover the target space.

The PSM may also provide other benefits beyond the word spotting application. Since the model configuration is recoverable for any match, the internal structure is accessible. This may open applications that require character segmentation, for example. Models generated from online input may also be interesting, because the temporal information from the online word may be transferrable to offline data. It will be interesting to see what applications can be devised for this new handwriting word model.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[2] M. Revow, C. Williams, and G. Hinton, "Using generative models for handwritten digit recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 592–606, 1996.

[3] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, 2005.

[4] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[5] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[7] L. Rabiner and B. J. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[8] N. Nayef and T. M. Breuel, "Graphical symbol retrieval using a branch and bound algorithm," in *International Conference on Image Processing*, 2010, pp. 2153–2156.

[9] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[10] V. Lavrenko, T. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," in *Proc. of the IEEE Workshop on Document and Image Analysis for Libraries DIAL'04*, 2004, pp. 278–287.

[11] N. Howe, T. Rath, and R. Manmatha, "Boosted decision trees for word recognition in handwritten document retrieval," in *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[12] Z. Guo and R. Hall, "Parallel thinning with two-subiteration algorithms," *Communications of the ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[13] N. Howe, "A Laplacian energy for document binarization," in *International Conference on Document Analysis and Recognition*, 2011, pp. 6–10.

[14] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics, and Image Processing*, vol. 34, no. 3, pp. 344–371, 1986.

[15] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, vol. 8, no. 19, 2012.

[16] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character hmms," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, 2012.