Handwriting Matching and Its Application to Handwriting Synthesis

Yefeng Zheng and David Doermann
Laboratory for Language and Media Processing
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742-3275
E-mail: {zhengyf, doermann}@umiacs.umd.edu

Abstract

Since it is extremely expensive to collect a large volume of handwriting samples, synthesized data are often used to enlarge the training set. We argue that, in order to generate good handwriting samples, a synthesis algorithm should learn the shape deformation characteristics of handwriting from real samples. In this paper, we present a point matching algorithm to learn the deformation, and apply it to handwriting synthesis. Preliminary experiments show the advantages of our approach.

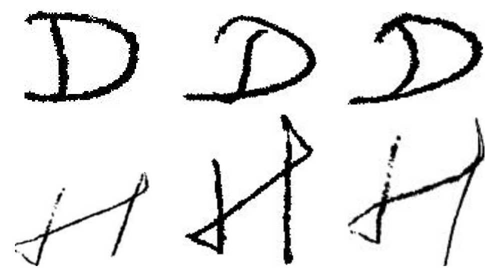

Figure 1. Real and synthesized handwriting samples. Left and middle columns: real samples. Right column: synthesized samples.

1 Introduction

A statistical pattern recognition system heavily depends on the size and quality of the training set. Although it is easy to prepare samples of machine printed text, doing so is expensive for handwriting. Synthesized data can be used as a supplement. The key problem of handwriting synthesis is to generate samples that look *natural*. Otherwise, arbitrarily synthesized samples cannot improve (if not deteriorate) the performance of the system trained on them. Although there are many variations among handwriting samples (such as size, rotation, stroke width), shape is generally used to categorize them into different classes. As shown in Fig. 1, nonrigid deformation of handwriting is obvious. Given two real samples, it is reasonable to assume some person may write with a shape between them (i.e., with similar but less degree of deformation). We argue that a synthesis algorithm should learn the shape deformation characteristics from real handwriting samples.

In this paper, we proposed a handwriting synthesis approach using two training samples. Handwriting is represented as a set of points, which are randomly selected from its skeleton. Point matching is used to learn the shape deformation characteristics. We formulate the matching between two point sets as an optimization problem, which can be interpreted as graph matching. Relaxation labeling is used to solve the optimization problem. After point matching, the thin plate spline (TPS) deformation model, with a parameter tunning the degree of deformation, is used to synthesize new samples. The right column in Fig. 1 shows some synthesized samples using our approach.

1.1 Previous Work

In the previous work, many handwriting synthesis approaches have been proposed to simulate the writing style of a specific person [1], or enlarge the training set for a recognition system [2, 3]. They can be roughly categorized as perturbation-based, model-based, or example-based. Perturbation-based methods only need one handwriting sample. New samples are generated by assigning random parameters to a deformation model, which is then used to deform the sample [2, 4]. A drawback of these approaches is that, without considering the deformation characteristics of handwriting, unrealistic samples may be generated. Instead, model-based approaches learn the deforma-

The support of this research by the Department of Defense of USA under contract MDA-9040-2C-0406 is gratefully acknowledged.

tion of handwriting and explicitly describe it as a distribution (the distribution is often called a model) [1, 5]. After learning, handwriting synthesis is the process of drawing new samples from the distribution. Although theoretically founded, model-based approaches have some drawbacks in real applications: handwriting models are often complex, and many samples are demanded for model training. Our approach belongs to example-based approaches, which use two handwriting samples, and generate new samples with shapes similar to both inputs. Compared with model-based approaches, much fewer samples are needed because it does not need to learn the distribution of deformation. Both model-based and example-based approaches need to perform handwriting matching, which is a challenge problem because handwriting is a nonrigid shape. Among all previous work, the algorithm proposed by Mori et al. [3] is most similar to our approach. They use a handwriting matching method, which is similar to the well-known iterated closed point (ICP) algorithm [6], to get the displacement vector of each pixel. A new sample is generated by move each pixel along its displacement vector. Compared with our approach, it has two drawbacks: (1) the ICP algorithm is not robust under nonrigid deformation [4], and (2) the displacement field is not continuous so the synthesized sample may change the topology.

The remainder of this paper is organized as follows. In the next section, we present our handwriting matching approach. Handwriting synthesis is discussed in Section 3, followed by experiments in Section 4. The paper concludes with a discussion for future work.

2 Handwriting Matching

Given two real handwriting samples, we perform handwriting matching to learn the shape deformation characteristics. In our approach, a handwriting sample is represented as a set of points uniformly sampled from its skeleton. In this section, we present our point pattern based handwriting matching algorithm.

2.1 Problem Formulation

Suppose a template shape T is composed of M points, $S_T = \{T_1, T_2, \cdots, T_M\}$, and a deformed shape D is composed of N points, $S_D = \{D_1, D_2, \cdots, D_N\}$. In order to enforce one-to-one matching, the point sets S_T and S_D are augmented to $S'_T = \{T_1, T_2, \cdots, T_M, nil\}$ and $S'_D = \{D_1, D_2, \cdots, D_N, nil\}$ respectively, by introducing a dummy or nil point. A match between shapes T and D is $f : S'_T \Leftrightarrow S'_D$, where the match of normal points is one-to-one, but multiple points may be matched to a dummy point.

Under a rigid transformation (i.e., translation and rotation), the distance between any point pair is preserved. Therefore, the optimal match $\hat{f}$ is

$$\hat{f} = arg \min_f C(T, D, f), \quad (1)$$

where

$$C(T, D, f) = \sum_{m=1}^{M}\sum_{i=1}^{M}\left(\|T_m - T_i\| - \|D_{f(m)} - D_{f(i)}\|\right)^2 + \sum_{n=1}^{N}\sum_{j=1}^{N}\left(\|D_n - D_j\| - \|T_{f^{-1}(n)} - T_{f^{-1}(j)}\|\right)^2. \quad (2)$$

If nonrigid deformation is present, the distance between a pair of points cannot be preserved. This is especially true for two points that are far apart. In many situations, however, the local neighborhood of a point may not change freely due to physical constraints. Therefore, we define a neighborhood for point m as $\mathcal{N}_m$. Then, (2) can be modified as

$$C(T, D, f) = \sum_{m=1}^{M}\sum_{i\in\mathcal{N}_m}\left(\|T_m - T_i\| - \|D_{f(m)} - D_{f(i)}\|\right)^2 + \sum_{n=1}^{N}\sum_{j\in\mathcal{N}_n}\left(\|D_n - D_j\| - \|T_{f^{-1}(n)} - T_{f^{-1}(j)}\|\right)^2. \quad (3)$$

The absolute distance of a pair of points is not preserved well under scale changes. Therefore, we quantize the distance to two levels as

$$\|T_m - T_i\| = \begin{cases} 0 & i \in \mathcal{N}_m \\ 1 & i \notin \mathcal{N}_m \end{cases} \text{ and } \|D_n - D_j\| = \begin{cases} 0 & j \in \mathcal{N}_n \\ 1 & j \notin \mathcal{N}_n \end{cases}. \quad (4)$$

Then, (3) is simplified as

$$C(T, D, f) = \sum_{m=1}^{M}\sum_{i\in\mathcal{N}_m} d(f(m), f(i)) + \sum_{n=1}^{N}\sum_{j\in\mathcal{N}_n} d(f^{-1}(n), f^{-1}(j)), \quad (5)$$

where

$$d(m, i) = \begin{cases} 0 & i \in \mathcal{N}_m \\ 1 & i \notin \mathcal{N}_m \end{cases}. \quad (6)$$

In order to deal with the case that a point may be matched to a dummy point, we let

$$d(., nil) = d(nil, .) = d(nil, nil) = 1 \quad (7)$$

to discourage matching to dummy points.

Simple mathematical deduction will convert the above minimization problem to a maximization one.

$$\hat{f} = arg \max_f S(T, D, f) \quad (8)$$

where

$$S(T, D, f) = \sum_{m=1}^{M}\sum_{i\in\mathcal{N}_m} \delta(f(m), f(i)) + \sum_{n=1}^{N}\sum_{j\in\mathcal{N}_n} \delta(f^{-1}(n), f^{-1}(j)) \quad (9)$$

$$\delta(i, j) = 1 - d(i, j) \quad (10)$$

This formulation can be interpreted as a graph matching problem. A point set is represented as a graph, where each point is a node in the graph and two nodes are connected by an edge if they are neighbors. The dummy node is not connected to other nodes in the graph. If connected nodes m and i in one graph are matched to connected nodes $f(m)$ and $f(i)$ in the other graph, $\delta(f(m), f(i)) = 1$. Therefore, the optimal solution of (8) is the one that maximizes the number of matched edges of two graphs.

Our definition of neighborhood is as follows. Initially, the graph is fully connected, and we then remove long edges until a pre-defined number of edges are preserved. Suppose, there are M nodes in the graph, the number of preserved edges is $M \times E_{ave}$ ($E_{ave} = 5$ in default). With this neighborhood definition, the graph representation of a point set is translation, rotation, and scale invariant.

2.2 Matching Probability Matrix

We can represent the matching function f in (8) with a set of supplemental variables, which are organized as a matrix P with dimension $(M + 1) \times (N + 1)$.

$$P = \left[\begin{array}{ccc|c} p_{11} & \cdots & p_{1N} & p_{1,nil} \\ \vdots & & \vdots & \vdots \\ p_{M1} & \cdots & p_{MN} & p_{M,nil} \\ \hline p_{nil,1} & \cdots & p_{nil,N} & 0 \end{array}\right] \quad (11)$$

If point T_m in the template shape T is matched to point D_n in the deformed shape D, then $P_{mn} = 1$; otherwise $P_{mn} = 0$. The last row and column of P represent the case that a point may be matched to a dummy point. Matrix P satisfies the following normalization conditions

$$\sum_{n=1}^{N+1} P_{mn} = 1 \quad \text{for } m = 1, 2, \cdots, M \quad (12)$$

$$\sum_{m=1}^{M+1} P_{mn} = 1 \quad \text{for } n = 1, 2, \cdots, N \quad (13)$$

Using matrix P, the objective function of (9) can be written as

$$S(T, D, P) = 2 \sum_{m=1}^{M} \sum_{i \in \mathcal{N}_m} \sum_{n=1}^{N} \sum_{j \in \mathcal{N}_n} P_{mn} P_{ij}. \quad (14)$$

Since $P_{mn} \in \{0, 1\}$, searching for an optimal P that maximizes $S(T, D, P)$ is a hard discrete combinatorial problem. In this paper, we use relaxation labeling to solve the optimization problem, where the condition $P_{mn} \in \{0, 1\}$ is relaxed as $P_{mn} \in [0, 1]$ [7]. After relaxation, P_{mn} is a real number, and the problem is converted to a constrained optimization problem with continuous variables.

2.3 Initialization with Shape Context Distance

Relaxation labeling only converges to a local optimal solution, a good initialization is crucial to get a solution with good quality. In this paper, we use the shape context distance [8] to initialize the matching probability matrix P. Shape context of a point is a measure of the distribution of other points relative to it. Consider two points, m in one shape, and n in the other shape. Their shape contexts are $h_m(k)$ and $h_n(k)$, for $k = 1, 2, \ldots, K$, respectively. Let C_{mn} denote that cost of matching these two points. As shape contexts are distributions represented as histograms, it is natural to use the χ^2 test statistic [8]

$$C_{mn} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_m(k) - h_n(k)]^2}{h_m(k) + h_n(k)}. \quad (15)$$

The Gibbs distribution is widely used in statistical physics and image analysis to relate the energy of a state to its probability. Taking the cost C_{mn} as the energy of the state that points m and n are matched, the probability of the match is

$$P_{mn} \propto e^{-C_{mn}/T_{init}}. \quad (16)$$

Parameter T_{init} ($T_{init} = 0.1$ in default) is used to adjust the reliability of the initial probability measures. We set the probability for a point matching to a dummy point, $P_{m,nil}$ or $P_{nil,n}$, to 0.2. Experiments show that our approach is not sensitive to this parameter.

2.4 Relaxation Labeling

Relaxation labeling was first proposed in a seminal paper by Rosenfeld, Hummel, and Zucker in mid-1970s [7]. The basic idea is to use iterated local context updates to achieve a globally consistent result. Their updating rule is

$$P_{mn} := \frac{P_{mn} S_{mn}}{\sum_{j=1}^{N} P_{mj} S_{mj}}, \quad (17)$$

where S_{mn} is a support function of the match between points T_m and D_n. It represents how much support the current match gets from its neighbors. The denominator is used to enforce the normalization constraint. Theoretic analysis shows that if S_{mn} is defined as the derivative of the objective function, the updating process will converge to a local optimal solution [9]. With the objective function of (14), S_{mn} is defined as

$$S_{mn} = 4 \sum_{i \in \mathcal{N}_m} \sum_{j \in \mathcal{N}_n} P_{ij}. \quad (18)$$

Unlike the original relaxation labeling where many-to-one matching is allowed, we proposed a new method to enforce one-to-one matching. Please refer to our technical report [10] for details.

After relaxation labeling updates, points with maximum matching probability less than P_{min} ($P_{min} = 0.95$) are labeled as outliers by matching them to dummy points.

3 Handwriting Synthesis

After handwriting matching, we use the thin plate spline (TPS) to model the deformation between two handwriting samples. The TPS model is often used for representing flexible coordinate transformations, because it is parameter free with a physical explanation and closed-form representations [11]. Two TPS models are used for the 2-D coordinate transformation. Suppose point (x_i, y_i) is matched to (u_i, v_i) for $i = 1, 2, \cdots, n$, let $z_i = f(x_i, y_i)$ be the target function value at location (x_i, y_i), we set z_i equal to u_i and v_i in turn to obtain one continuous transformation for each coordinate. The TPS interpolant $f(x, y)$ minimizes the following bending energy

$$I_f = \int\int_{\mathcal{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dxdy \quad (19)$$

and has the solution of the form

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i U(\|(x_i, y_i) - (x, y)\|) \quad (20)$$

where $U(r)$ is the kernel function, taking the form of $U(r) = r^2 logr^2$. The parameters of the TPS model w and a are the solution of the following linear equation

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} z \\ 0 \end{bmatrix} \quad (21)$$

where $K_{ij} = U(\|(x_i, y_i) - (x_j, y_j)\|)$, the ith row of P is $(1, x_i, y_i)$, w and z are column vectors formed from w_i and z_i respectively, and a is a column vector with elements a_1, a_x, and a_y.

For handwriting synthesis, we can control the degree of deformation by regularization as follows.

$$H_f = \sum_{i=1}^{n} [z_i - f(x_i, y_i)]^2 + \lambda I_f \quad (22)$$

where λ is a parameter, controlling the amount of smoothing. The regularized TPS can be solved by replacing K in (21) with $K + \lambda I$, where I is the $n \times n$ identity matrix [8]. It has been shown that if $\lambda = \infty$, the deformation is the affine transformation. With a smaller λ, the interpolated shape is closer to the target shape.

4 Experiments

We have tested our nonrigid shape matching algorithm on a public data set and compared it with two state-of-the-art algorithms: the shape context [8] and TPS-RPM [4] algorithms. Experiments show our approach outperforms the other two under nonrigid deformation and noise. Due to the space limit, we cannot show the experimental results here. Interested readers are referred to our technical report for details [10].

In this section, we focus on its application to handwriting synthesis and compare it with two other algorithms [3, 4]. The top row of Fig. 2 shows two handwriting samples and their point matching result. Under this match, we use TPS to deform the first sample to synthesize new samples, as shown in the second row (from left to right, λ takes the value of ∞, 10, 1, and 0.1 respectively). With the decrease of the regularization parameter λ, the synthesized sample is closer to the second real sample. The third row in Fig. 2 shows synthesized samples using a perturbation-based approach [4]. With only one training sample, the deformation of synthesized samples is random, and sometimes unnatural samples may be generated. The third row shows the synthesized samples using the approach of [3]. Suppose point T_i is matched to point $D_{f(i)}$, its displacement is $d_i = D_{f(i)} - T_i$. For an un-matched point, the displacement takes the value of its closest matched point. A new sample is synthesized by moving a point along its displacement.

$$S_i = T_i + \rho d_i \quad (23)$$

If $\rho = 0$, the synthesized sample is just the original temple; and if $\rho = 1$, all matched points will be moved to the target under any matching function. The last row of Fig. 2 shows several synthesized samples with ρ equals 0.2, 0.4, 0.6, and 0.8, respectively. As shown in the figure, both example-based approaches can learn shape deformation characteristics if the point matching results are good. The approach of [3], however, may change the topology (the synthesized handwriting is broken into several parts) due to the discontinuity of the displacement field. This drawback is obvious when point matching errors (which are un-avoidable in real applications) are present. The second and third rows of Fig. 3 show synthesized samples using our approach and [3]. The topology of samples synthesized by our approach is preserved even under substantial errors. Using the approach of [3], unrealistic samples are generated.

More examples on handwriting synthesis are shown in Fig. 4. The ordering of samples is same to Fig. 2. Samples with different slant (within the range of the slant between two training samples) are generated using our approach.

5 Conclusion and Future Work

In this paper, we proposed a new handwriting matching approach and applied it to synthesize new training samples for handwriting recognition. Our approach automatically learns shape deformation characteristics from training samples, so more visually realistic samples are generated. Although it has been shown in several independent experiments that synthesized handwriting samples can improve a recognition system trained on them [2, 3], we will do more experiments to verify it in the future.

Our approach suffers from a number of limitations. In our approach, we assume the same character is written with similar shape though with some degree of distortion. Due to difference in geological location, education and culture, people may write the same character with significant different shapes. Our approach will fail when this happens.

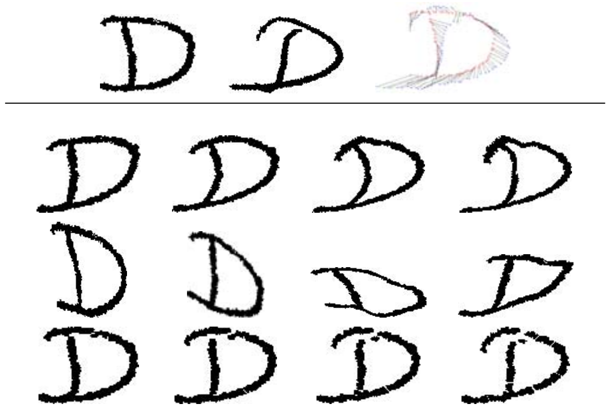

Figure 2. Handwriting synthesis. Top row: training samples. Synthesized samples: second row for our approach, third row for [4], and forth row for [3].

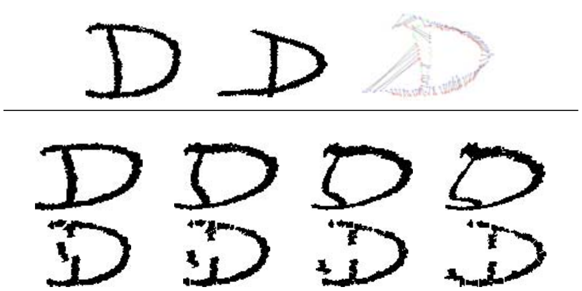

Figure 3. Handwriting synthesis with point matching errors presented. Top row: training samples. Synthesized samples: second row for our approach, third row for [3].

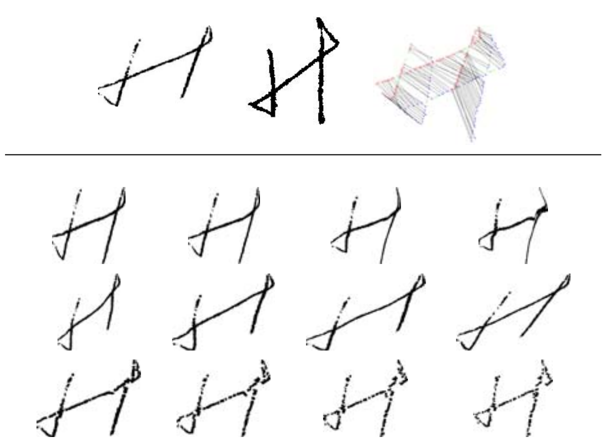

Figure 4. More examples on handwriting synthesis. Top row: training samples. Synthesized samples: second row for our approach, third row for [4], and forth row for [3].

Our approach is more suitable for character level synthesis. There are two challenges applying our approach to synthesize long handwritten words or text lines: (1) the point matching accuracy decreases for a complicated shape represented with a large number of points, and (2) the deformation is more complex, probably beyond the capability of the TPS deformation model.

References

[1] J. Wang, C. Wu, Y.-Q. Xu, and H.-Y. Shum. Combining shape and physical models for online cursive handwriting synthesis. *Int'l J. Document Analysis and Recognition*, Online First, 2004.

[2] T. Varga and H. Bunke. Generation of synthetic training data for an HMM-based handwriting recognition system. In *Proc. Int'l Conf. Document Analysis and Recognition*, pages 618–622, 2003.

[3] M. Mori, A. Suzuki, A. Shio, and S. Ohtsuka. Generating new samples from handwriting numerals based on point correspondence. In *Int'l Workshop on Frontiers in Handwriting Recognition*, pages 281–290, 2000.

[4] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[5] H. Choi, S.-J. Cho, and J.H. Kim. Generation of handwritten characters with Bayesian network based on-line handwriting recognizers. In *Proc. Int'l Conf. Document Analysis and Recognition*, pages 995–999, 2003.

[6] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):239–256, 1992.

[7] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. System, Man and Cybernetics*, 6(6):420–433, 1976.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(4):509–522, 2002.

[9] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323, 1997.

[10] Y. Zheng and D. Doermann. Robust point matching for non-rigid shapes: A relaxation labeling based approach. Technical Report LAMP-TR-117, University of Maryland, College Park, USA, 2004.

[11] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(6):567–585, 1989.