Invariant Features for HMM Based On-Line Handwriting Recognition

Jianying Hu, Michael K. Brown and William Turin

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA

Abstract. In this paper we address the problem variability in handwriting due to geometric distortion of letters and words by rotation, scale and translation. In general, translation has not been a problem because it is easy to choose features that are invariant with respect to translation. It is more difficult to find features that are invariant with respect to all three types of geometric distortion. We introduce two new features for HMM based handwriting recognition that are invariant with respect to translation, rotation and scale changes. These are termed *ratio of tangents* and *normalized curvature*. Writer-independent recognition error in our system is reduced by a factor of over 50% by employing these features.

1 Introduction

The principal difficulty in the recognition of patterns by computer is dealing with the variability of measurements, or features, extracted from the patterns. There are several sources of variability, depending on the type of pattern data being processed. In on-line handwriting recognition these sources include input device noise, temporal and spatial quantization error, and variability in the rendering of input by the writer.

In this paper we address the spatial component of the last source of variability, that is, the geometric distortion of letters and words by rotation, scale and translation. We introduce two new features for handwriting recognition that are invariant with respect to all three factors of geometric distortion. These are termed *ratio of tangents* and *normalized curvature*. Similar features have appeared previously in the literature for planar shape recognition under partial occlusion [1].

2 Invariant Features

The concept of invariant features arises frequently in various machine vision tasks. Depending on the specific task, the geometric transformation ranges from simple rigid plane motion to general affine transformation, to perspective mapping, etc. [1]. In the case of handwriting recognition, the transformation of interest is *similitude* transformation, which is a combination of translation, rotation

and scaling [1]. In our basic HMM based handwriting recognition system [2], we used the tangent slope feature which is invariant under translation and scaling, but not rotation. In this section we consider features that are invariant under arbitrary similitude transformation.

A similitude transformation of the Euclidean plane $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined by $\mathbf{w} = c\mathbf{U}\mathbf{r} + \mathbf{v}$, where c is a positive scalar, $\mathbf{U} = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix}$, $\mathbf{v} = [v_x\ v_y]^T$, representing a transformation that includes scaling by c, rotation by angle ω and translation by $\mathbf{v}$. We regard two curves as equivalent if they can be obtained from each other through a similitude transformation. Invariant features are features that have the same value at corresponding points on different equivalent curves.

Suppose that a smooth planar curve $\mathbf{P}(t) = (x(t), y(t))$ is mapped into $\tilde{\mathbf{P}}(\tilde{t}) = (\tilde{x}(\tilde{t}), \tilde{y}(\tilde{t}))$ by a reparametrization $t(\tilde{t})$ and a similitude transformation, i.e.

$$\tilde{\mathbf{P}}(\tilde{t}) = c\,\mathbf{U}\,\mathbf{P}(t(\tilde{t})) + \mathbf{v}\ . \tag{1}$$

Without loss of generality, assume that both curves are parametrized by arc length (natural parameter), i.e. $t = s$ and $\tilde{t} = \tilde{s}$. Obviously, $\mathrm{d}\tilde{s} = c\mathrm{d}s$. It can be shown [1] that curvature (the reciprocal of radius) at the corresponding points of the two curves is scaled by $1/c$, i.e. $\tilde{\kappa}(\tilde{s}) = \kappa\left((\tilde{s} - \tilde{s})/c\right)/c$. It follows that:

$$\frac{\tilde{\kappa}'(\tilde{s})}{\left(\tilde{\kappa}(\tilde{s})\right)^2} = \frac{\kappa'\left((\tilde{s} - \tilde{s}_0)/c\right)}{\left(\kappa((\tilde{s} - \tilde{s}_0)/c)\right)^2}\ , \tag{2}$$

where $\tilde{\kappa}' = \mathrm{d}\tilde{\kappa}/\mathrm{d}\tilde{s}$ and $\kappa' = \mathrm{d}\kappa/\mathrm{d}s$, thus eliminating the scale factor from the value of the ratio. Equation (2) defines an invariant feature which we shall refer to as *normalized curvature*.

The computation of the normalized curvature defined above involves derivative estimation of up to the third order. Another set of invariants that require lower orders of derivatives can be obtained by using the invariance of distance ratios between corresponding points. Consider again the two equivalent curves $\mathbf{P}(t)$ and $\tilde{\mathbf{P}}(\tilde{t})$ defined above. Suppose P_1 and P_2 are two points on $\mathbf{P}(t)$ whose tangent slope angles differ by θ; $\tilde{P}_1$ and $\tilde{P}_2$ are two points on $\tilde{\mathbf{P}}(\tilde{t})$ with the same tangent slope angle difference. P and $\tilde{P}$ are the intersections of the two tangents on $\mathbf{P}(t)$ and $\tilde{\mathbf{P}}(\tilde{t})$, respectively (Fig. 1). Since angles and hence turns of the curve are invariant under the similitude transformation, it can be shown that if point $\tilde{P}_1$ corresponds to point P_1, then points $\tilde{P}_2$ and $\tilde{P}$ correspond to points P_2 and P respectively [1]. It follows from (1) that:

$$\frac{\left|\tilde{P}\tilde{P}_2\right|}{\left|\tilde{P}_1\tilde{P}\right|} = \frac{|PP_2|}{|P_1P|}\ . \tag{3}$$

Equation (3) defines another invariant feature which we shall refer to as *ratio of tangents*.

[1] The same transformation was referred to as *similarity* transformation by Bruckstein *et. al.* [1]. We have chosen another term to avoid confusion with the well known similarity transformation of linear algebra.

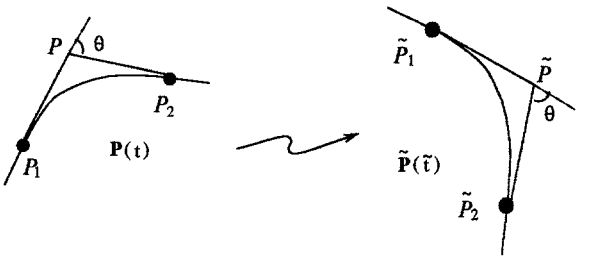

Fig. 1. Ratio of tangents

3 Implementation Issues

3.1 Derivative Estimation

To evaluate accurately the invariant features described above, high quality derivative estimates up to the third order have to be obtained from the sample points. In the following we describe how to use spline smoothing operators for derivative estimation.

Let $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ be the noisy input vector obtained by sampling a "smooth" function $g(t)$ at $\mathbf{t} = (t_1, t_2, \ldots, t_n)$. Given an integer $m \geq 1$, the *natural polynomial smoothing spline approximation of order* $2m$ to $g(t)$ is the unique solution to the problem of finding $f(t)$ with $m-1$ absolutely continuous derivatives, and square integrable mth derivative, which minimizes [3]: $\frac{1}{n}\sum_{i=1}^{n}(f(t_i)-y_i)^2 + \lambda\int_{t_1}^{t_n}\left(f^{(m)}(u)\right)^2 du$. Let $\mathbf{g}$ be the output vector obtained as a result of evaluating the smoothing spline approximation at values of t given by $\mathbf{t}$. Since the smoothing spline approximation is linear we have $\mathbf{g} = \mathbf{A}(\lambda)\mathbf{y}$, where $\mathbf{A}(\lambda)$ is independent of the input vector. In a typical application to data lengths greater than n, the row $k = (n+1)/2$ of $\mathbf{A}(\lambda)$ is convolved with the data to produce the set of filtered data. Operators for estimating the νth derivative of $g(t)$ at the sample points can be constructed by evaluating the derivative of the spline approximation at the sample points. Since this is a linear functional, the result can also be expressed by a matrix-vector multiplication, i.e. $\mathbf{g}^\nu = \mathbf{A}^\nu(\lambda)\mathbf{y}$. Efficient algorithms for the construction of the $\mathbf{A}^\nu(\lambda)$'s exist [3].

To obtain estimates of third order derivatives, m has to be at least 3, yielding a spline of degree 5. The length of the operator is constrained by: $n \geq 4m-1$. Given the degree of the spline, a wider operator provides better support for the spline estimation but also involves more computation. We chose to use a spline filter of degree 5 ($m = 3$) and length $n = 15$. The smoothness parameter λ controls the cut-off frequency f_c if the spline smoothing operator is viewed as a low pass filter. Since the handwriting signal (with the exception of cusps) consists predominantly of low frequencies and the predominant noise sources (mostly quantization error and jitter) are of high frequency content, it is easy to choose λ so that the spline filter cuts off most of the noise without causing significant distortion of the signal. In our system the handwritten scripts are parameterized in terms of arc length by resampling at 0.2 mm intervals before

feature extraction and $\lambda = 20$ ($f_c \simeq 0.425\text{mm}^{-1}$) is used for all our experiments. Cusps tend to be smoothed out when such derivative operators are applied. However this does not pose a severe problem as long as the resulted ratio of tangents and normalized curvature features are used along with the tangent slope feature, since information related to cusps can be captured by the last feature [2].

3.2 Calculation of Ratio of Tangents

First we describe how to compute the ratio of tangents with an arbitrary tangent slope angle difference. Suppose P_1 and P_2 are two points along a script whose tangent slope angles differ by θ, as shown in Fig. (2). P is the intersection of the two tangents. The ratio of tangents at P_1 is defined as: $Rt_\theta(P_1) = |PP_2| \,/\, |P_1P|$. Suppose $\mathbf{u}_1$ and $\mathbf{u}_2$ are unit normal vectors at P_1 and P_2 respectively, using the law of sines we get: $Rt_\theta(P_1) = \sin\alpha / \sin\beta = |P_1P_2 \cdot \mathbf{u}_1| \,/\, |P_1P_2 \cdot \mathbf{u}_2|$. For convenience, we shall call P_2 the θ *boundary* of P_1.

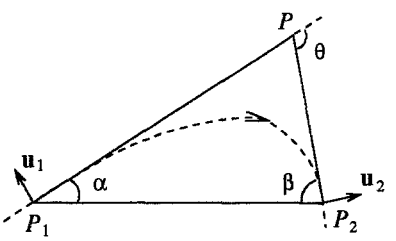

Fig. 2. Calculation of ratio of tangents

In order to use ratio of tangents as an invariant feature in handwriting recognition, a fixed angle difference $\theta = \theta_0$ has to be used for all sample points in all scripts. In real applications we normally have only scattered sample points instead of the continuous script and, in general, we can not find two sample points whose slope angle difference is equal to θ_0. Suppose sample point P_i's θ_0 boundary is between points P_j and P_{j+1}, i.e. P_j is P_i's θ_1 boundary and P_{j+1} is its θ_2 boundary where $\theta_1 < \theta_0 < \theta_2$, $Rt_{\theta_0}(P_i)$ is estimated from $Rt_{\theta_1}(P_i)$ and $Rt_{\theta_2}(P_i)$ using linear interpolation.

Obviously the choice of θ_0 greatly affects the tangent ratio values. If θ_0 is too small, the feature tends to be too sensitive to noise. On the other hand, if θ_0 is too large, the feature becomes too global, missing important local shape characteristics. Currently this value is chosen heuristically and $\theta_0 = 10°$ is used in all our experiments.

In order to enhance the distinctive power of the feature, we augment the ratio of tangents at each point by the sign of the curvature at that point. The resulted feature is referred to as *signed ratio of tangents*, and is used instead of ratio of tangents in the experiments described later.

3.3 Combined Scores

In our discrete HMM system, each feature is quantized into a fixed number of bins. To simplify our models, we chose to treat the features as being independent from each other. When the three features are used together, the joint probability of observing symbol vector $S_{k_1k_2k_3} = [k_1, k_2, k_3]$ in state j is: $b_j(S_{k_1k_2k_3}) = \prod_{i=1}^{3} b_{ji}(k_i)$, where $b_{ji}(k_i)$ is the probability of observing symbol k_i in state j according to the probability distribution of the ith feature. It follows that the corresponding log-likelihood at state j is: $L_j(S_{k_1k_2k_3}) = \sum_{i=1}^{3} \log[b_{ji}(k_i)]$.

In a conventional HMM implementation with Viterbi scoring, the likelihood defined above is used directly in training and recognition. In this case, each of the three features contributes equally to the combined log-likelihood and therefore has equal influence over the accumulated scores and the optimal path. However, our experiments with the three features show that when used alone, the tangent slope feature gives far better recognition performance than each of the two invariant features. This suggests that the tangent slope is a more discriminative feature and therefore should have more influence on decision making than the other two features. In order to adjust the influence of different features according to their discriminative power, we use instead the *weighted log-likelihood*: $\hat{L}_j(S_{k_1k_2k_3}) = \sum_{i=1}^{3} w_i \log(b_{ji}(k_i)) - \log(N_j)$, where N_j is the state normalization factor such that the *weighted probabilities*, defined by $\hat{b}_j(S_{k_1k_2k_3}) = \prod_{i=1}^{3} [b_{ji}(k_i)]^{w_i}/N_j$, sum up to 1 for each state. The normalization factor ensures that the weighted log-likelihood is not biased towards any particular state.

4 Experimental Results

The detailed description of our recognition system can be found in a previous paper [2]. A brief review is provided here.

The handwriting data was collected using a newly developed graphics input tablet [4] at 200 samples per second. Writers were asked to write on a lined sheet of paper, without any constraints on speed or style. The preprocessing steps include cusp detection, smoothing, deskewing and finally resampling at 0.2 mm intervals. Features are then extracted at each resampled point.

Sub-character models called *nebulous stroke models* are used as the basic model units. Currently each stroke is modeled by a single HMM state. A letter model is a left to right HMM with no state skipping, constructed at run time by concatenating the corresponding stroke models. These HMM's are embedded in a stochastic language model which describes the vocabulary. The current vocabulary contains 32 words, targeting an underlying application of a pen driven graphics editor.

8595 samples have been collected from 18 writers, with each word in the vocabulary written 15 times by each writer. 10 writers were chosen (after data collection) to be the "training writers". The training set is composed of about 10 samples of each word from each training writer, a total of 3180 samples. The

test set is composed of all samples not used for training, divided into two groups. Group A contains 1592 samples from the 10 training writers, group B contains 3823 samples from the 8 other writers not used for training.

Table 1 compares the error rates of the system when a single feature – tangent slope was used and when the two invariant features – signed ratio of tangents and normalized curvature were added. As shown in the table, by adding the two invariant features we have achieved an error rate reduction of 46% for training writers, and 54% for non-training writers.

Table 1. Comparison of error rates

	A (Training Writers)	B (Non-training Writers)
single feature	10.3%	23.4%
invariant features added	5.6%	10.7%

5 Concluding Remarks

We have introduced two new features for handwriting recognition that are invariant under translation, rotation and scaling. The use of these features in an HMM based handwriting recognition system has been demonstrated and significant improvement in recognition performance has been obtained. Invariant features have been discussed extensively in computer vision literature. However, they have been rarely used in real applications due to the difficulty involved in estimating high order derivatives. We have demonstrated that high order invariant features can indeed be made useful with careful filtering in derivative estimation.

References

1. Bruckstein, A. M., Holt, R. J., Netravali, A. N., Richardson, T. J.: Invariant signatures for planar shape recognition under partial occlusion. CVGIP: Image Understanding **58** (1993) 49–65
2. Hu, J., Brown, M. K., Turin, W.: Handwriting recognition with Hidden Markov Models and grammatical constraints. Fourth Int. Workshop on Frontiers in Handwriting Recognition, Taipei, Taiwan (1994) 195–205
3. Lyche, T., Schumaker, L. L., Computation of smoothing and interpolating natural splines via local bases. SIAM J. Numer. Anal. **10(6)** (1973) 1027–1038
4. Boie, R., Ruedisueli, L., Wagner E.: Capacitive position sensor. U.S. patent filed (1993)