Toward Affine Recognition of Handwritten Mathematical Characters

Oleg Golubitsky
Google, Inc.
340 Hagey Blvd.
Waterloo, Ontario, Canada
oleg.golubitsky@gmail.com

Vadim Mazalov
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada
vmazalov@uwo.ca

Stephen M. Watt
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada
watt@uwo.ca

ABSTRACT

We address the problem of handwritten symbol classification in the presence of distortions modeled by affine transformations. We consider shear, rotation, scaling and translation, since these types of transformations occur most often in practice, and focus most on shear within this framework. We present a distance-based classification method, in which feature vectors are constructed from Legendre-Sobolev expansions of the coordinate functions and of the affine integral invariants of the curves given by the symbol's ink strokes. We analyze different size normalization methods and conclude that integral invariants provide the most robust norm. Finally, we propose a new parameterization, a combination of arc length and time, insensitive to variations in curve tracing speed and affine distortion.

1. INTRODUCTION

Our objective is classification of handwritten mathematical characters, as the basis for mathematical expression recognition. We study techniques primarily for online symbol classification to be applied in digital pen environments, although our algorithms can be adapted to offline classification of general planar curves. Even though considerable work has been done in the area of handwriting recognition, the classification of mathematical symbols requires special consideration. The set of mathematical characters includes numerous similar few-stroke symbols that may appear ambiguous even to a human and errors in classification are unavoidable. Such ambiguities are exacerbated by diverse handwriting styles and are hard to correct in general. However, samples are commonly subjected to a limited number of transformations, and one could try to eliminate certain classification errors by using methods invariant with respect to such transformations. Among these we consider skew, rotation, scaling and translation, which are all affine transformations.

Some advances have been made [1, 2, 3, 4] with classification based on the approximation of coordinate functions by truncated Legendre-Sobolev series and the distance from the sample to be classified to the convex hulls of classes in the space of coefficient vectors. This method was found to yield a 97.5% recognition rate with a dataset of samples, most of which were collected as isolated symbols. Although our samples do exhibit a certain amount of transformation as mentioned above, we expect that symbols collected in a less controlled environment are likely to be much more affected by such transformations. Therefore, we investigate robust algorithms that are not susceptible to such distortions. To evaluate the sensitivity of the original algorithm to angular perturbations, samples were rotated by random angles in a certain interval. We found that the recognition rate for the original algorithm decreases approximately quadratically with respect to the angle and drops to 90% when the angle ranges over an interval of $[-0.3, 0.3]$ radians. Therefore, the original algorithm is quite sensitive to rotations, while being fully translation- and scale-invariant.

A modified algorithm, invariant under rotation, was introduced in [5] and is based on the theory of integral invariants of parametric curves [6]. Out of an infinite family of such invariants, a subfamily was chosen that is fast to compute, numerically stable and sufficiently discriminating. These were used to find the top N classes for character samples, based on the distance to convex hulls of nearest neighbours in the space of Legendre-Sobolev coefficients of the integral invariant functions. The number of top classes was determined empirically to ensure high probability of the correct class being within the selected classes. This helped reduce the number of classes for further processing. The recognition rate of this algorithm was found to be 96.3% for the original unrotated samples and to decrease slightly with greater rotation angle and reached 94.7% for the interval $[-0.3, 0.3]$. We note the small decrease in performance for non-rotated samples. This is due in part to the fact that allowing rotation makes certain symbols more ambiguous and also partly because the coordinate functions serve as a better curve descriptor than invariants. The investigation focused on the interval of $[-0.3, 0.3]$ radians because that is the range of angles that might reasonably occur for handwriting, and not because of any limitations of the rotation-invariant method.

This paper addresses another class of transformations that often occur in practice: shear, or "skew", transformations. This may be seen as a theoretically sound form of "de-slanting". Samples that have been sheared seem to be quite common in handwriting, compared with other transformations. Also, the maximal shear angle, for which a character is still readable by a human can be quite large (Figure 1), compared

DAS '10, June 9-11, 2010, Boston, MA, USA

to the corresponding maximal rotation angle. We therefore expect that, in practice, a large amount of shear can occur, and consider shear invariance as a useful addition to the set of tools for character recognition.

We have also found that shear is harder to deal with than rotation. Since shear does not preserve the length of strokes, parameterization by the Euclidean arc length is no longer robust. Size normalization requires special attention as well. We develop an algorithm, invariant with respect to shear, rotation, scale and translation, and then propose a way to extend the invariance of the method to the full affine group, while keeping the recognition rate higher than that of classification with the affine integral invariants alone.

This paper is organized as follows. In Section 2 we give an overview of some of the existing affine methods and compare them with our approach. In Section 3 we summarize key facts about integral invariants and describe their use in our algorithms. Scale normalization, parameterization of coordinate functions and the algorithm itself are discussed in Section 4. The experimental setting and results using the usual parameterizations are given in Section 5. Section 6 introduces and evaluates a mixed space-time parameterization that is useful for shear. A possible extension of the algorithm to the full affine group is presented in Section 7. Section 8 concludes the paper.

2. OVERVIEW OF AFFINE METHODS

In this section we briefly describe some of the existing methods invariant under the group of affine transformations and the differences with our approach.

Stroke-Based Affine Transformation

This approach was proposed in [7] to minimize distortions in handwriting by applying stroke-based affine transformation. The algorithm denotes stroke-wise uniform affine transformation for a stroke i with A_i and b_i, where A_i is a 2×2 matrix for shear, rotation and scale and b_i is a 2-dimensional translation vector. For a sample, a set of N strokes is selected to construct the objective function in the form of least-squares data fitting to determine the optimal A_i and b_i as

$$F_i = \sum_k \|A_i t_k + b_i - r_k\|^2 \to \min \text{ for } A_i, b_i, (1 \leq i \leq N)$$

where t_k and r_k are k-th feature points of the sample to be classified and a reference sample respectively, and $\|\cdot\|$ is the Euclidean norm. This yields the affine transformation with the least distance between corresponding strokes of the input and a reference sample. This procedure is repeated for each reference sample and then distance-based classification takes place. Recognition of handwritten characters as gray scale images was proposed in [8], using similar ideas.

Minimax Classification with HMM

An online method, robust against affine distortions, was developed in [9], based on continuous-density hidden Markov models (CDHMM). Let N be the number of character classes $C_i,\ i = 1, ..., N$, each containing M_i CDHMMs

$$\left\{ \lambda_i^{(m)}, m = 1, ..., M_i \right\}.$$

In the non-affine approach, an input sample I is classified as member of class C_i in terms of the joint likelihood of the observation I and the associated hidden state sequence S given CDHMM $\lambda_j^{(m)}$, $p(I, S|\lambda_j^{(m)})$, as follows

$$\arg\max_j \left\{ \max_m \left[\max_S \log p(I, S|\lambda_j^{(m)}) \right] \right\}.$$

In order to eliminate affine distortions between the input and training samples, the authors use

$$\arg\max_j \left\{ \max_m \left[\max_S \log p(I, S|\Gamma_{\hat{A}}(\lambda_j^{(m)})) \right] \right\},$$

where Γ_A is a specific transformation of $\lambda_j^{(m)}$ with parameters A, and $\hat{A} = \arg\max_A p(I, \Gamma_A(\lambda_j^{(m)}))$. The authors propose solving this optimization problem with three iterations of the EM algorithm described in [10].

Affine Moment Invariants

Affine moment invariants (AMIs) are independent of actions of the general affine group and can be used in recognition of handwritten characters [11]. A central moment of order $p + q$ for a 2-dimensional object O can be represented as

$$\mu_{pq} = \iint_O (x - x_c)^p (y - y_c)^q \mathrm{d}x\mathrm{d}y$$

where (x_c, y_c) is the center of gravity of the object O. In the paper the first four affine moments were calculated to obtain a description of an isolated character in the form of a 4-dimensional vector. Samples were classified by the minimum Euclidean distance to the training samples. The performance of AMIs is compared with that of the geometric moment invariants, which are invariant under rotation, scale and translation. It was concluded that AMIs gave a better recognition rate than geometric moments.

As opposed to the first two methods described above, we propose a different technique of classifying handwritten characters with integral invariants. Similarly to the classification with AMIs, we compute affine-invariant quantities from the original curve, without using any specific transformations of the input sample. However, AMIs, as originally defined, provide curve-to-value correspondence, unlike the curve-to-curve correspondence with the integral invariants. This allows to obtain a richer description of a curve without excessive computation. In fact, we consider only 2 invariants and find them sufficient for an acceptable classification accuracy of samples under different transformations. In order to further improve the classification rate, we still perform an analysis of a sample to obtain a numerical measure of the distortion (i.e. the angle of rotation or shear). This analysis is performed for a small subset of top classes determined by the affine-invariant classification and is computationally inexpensive.

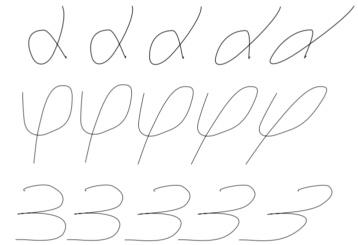

Figure 1: Skews of 0.0, 0.2, 0.4, 0.6 and 0.8 radians.

3. INTEGRAL INVARIANTS

Integral invariants provide an elegant approach to planar and spatial curve classification under affine transformations [6]. Given a symbol, in order to obtain quantities that are independent under special linear transformations, we evaluate invariants constructed from the coordinate functions. Such invariant functions, compared to differential invariants, are less sensitive to small perturbations, and therefore yield more robust methods for classifying handwritten characters with sampling noise.

The following functions invariant under the special linear group $SL(2)$ are considered in the paper

$$I_1(\lambda) = \int_0^\lambda X(\tau)\mathrm{d}Y(\tau) - \frac{1}{2}X(\lambda)Y(\lambda)$$

$$\begin{aligned} I_2(\lambda) &= X(\lambda)\int_0^\lambda X(\tau)Y(\tau)\mathrm{d}Y(\tau) \\ &\quad - \frac{1}{2}Y(\lambda)\int_0^\lambda X^2(\tau)\mathrm{d}Y(\tau) - \frac{1}{6}X^2(\lambda)Y^2(\lambda) \end{aligned}$$

where $X(\lambda)$, $Y(\lambda)$ are coordinate functions.

Function $I_1(\lambda)$ can be geometrically represented as the signed area between the curve and its secant (Figure 2), while $I_2(\lambda)$ can be described in terms of volume [6].

Having computed the coefficients $(x_1, ..., x_d, y_1, ..., y_d)$ of the coordinate functions approximated by truncated Legendre-Sobolev series [5], we can obtain I_1 as

$$I_1(\lambda) \approx \sum_{i,j=1}^{d} x_i y_j \left[\int_0^\lambda P_i(\tau)P_j'(\tau)\mathrm{d}\tau - \frac{1}{2}P_i(\lambda)P_j(\lambda)\right]$$

where P_i is the i-th Legendre-Sobolev basis polynomial. Then expression for I_2 is

$$I_2(\lambda) \approx \sum_{i,j,k,l=1}^{d} x_i x_j y_k y_l \ \mu_{ijkl}$$

where μ_{ijkl} is

$$\begin{aligned} P_i(\lambda)\int_0^\lambda P_j(\tau)P_k(\tau)P_l'(\tau)\mathrm{d}\tau - \frac{1}{2}P_l(\lambda)\int_0^\lambda P_i(\tau)P_j(\tau)P_k'(\tau)\mathrm{d}\tau \\ - \frac{1}{6}P_i(\lambda)P_j(\lambda)P_k(\lambda)P_l(\lambda). \end{aligned}$$

4. A SHEAR-INVARIANT ALGORITHM

In this section we discuss an algorithm, invariant under shear, in addition to rotation, scale and translation [5]. We consider different size normalization methods and parameterizations of coordinate functions to ensure appropriate setting for the method. The algorithm itself is given at the end of the section.

4.1 Size Normalization

Size normalization is traditionally implemented by rescaling a sample to achieve standard values of certain parameters. In our previous algorithm, this parameter was the Euclidean norm of the vector of Legendre-Sobolev coefficients of coordinate functions [4]. While this norm can still be used to rescale rotated samples [5], it is not invariant under shear and affine transformations in general. Instead, we look at the

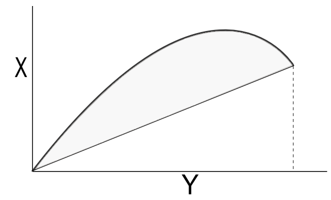

Figure 2: Geometric representation of I_1.

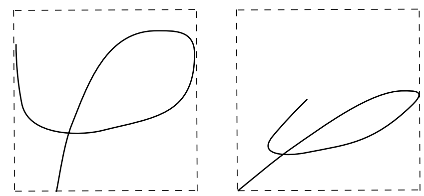

Figure 3: Aspect ratio size normalization.

norm of the Legendre-Sobolev coefficient vector of I_1. We can then normalize the coefficient vectors of the coordinate functions by multiplying them by $1/\sqrt{\|I_1\|}$. Finally, we compute the coefficients of I_2 from the normalized coefficients of the coordinate functions. Computing the norm of I_1 allows us to extend the invariance of I_1 and I_2 from the special linear group, $SL(2, R)$, to the general linear group, $GL(2, R)$. Invariance under the general affine group, $\mathrm{Aff}(2, R)$, is obtained by dropping the first (order-0) coefficients from the coefficient vectors of the coordinate functions [4].

To evaluate the performance of $\|I_1\|$ for normalization, we consider two other normalization approaches typical in handwriting recognition: height and aspect ratio [12]. Both of these are not perfect in the presence of affine transformations. While normalization by height is invariant under horizontal shear, it becomes inaccurate if samples are subjected to rotation. Aspect ratio is suitable for rotation, but becomes inaccurate for larger degrees of shear (Figure 3).

4.2 Parameterization of Coordinate Functions

Parameterization by time and arc length are among the most popular choices in online handwriting recognition. Parameterization by arc length is usually preferable, since it is not affected by variations in writing speed and is invariant under Euclidean transformation. It may be expressed as

$$AL(\lambda) = \int_0^\lambda \sqrt{(X'(\tau))^2 + (Y'(\tau))^2}\mathrm{d}\tau.$$

When one looks at the group of affine transformations, however, parameterization by arc length may no longer be the best choice, since it is no longer invariant. For example, it is changed by shear distortion. Instead, we may consider parameterization by special affine arc length. We use affine arc length in the form

$$AAL(\lambda) = \int_0^\lambda \sqrt[3]{|X'(\tau)Y''(\tau) - X''(\tau)Y'(\tau)|}\mathrm{d}\tau.$$

In our experiments we study the recognition rate using each of the three parameterizations to evaluate performance empirically for different levels of distortion.

4.3 The Algorithm

In online classification algorithms, a symbol is given as a continuous curve defined by a discrete sequence of points. When a symbol is given by multiple strokes, they are joined. The curve is parameterized with an appropriate function (Section 4.2) and the Legendre-Sobolev coefficients of the coordinate functions are computed online, as points are accumulated [2]. Using the representation of $I_1(\lambda)$ in Section 3, coefficients of the invariant are computed as

$$I_{1,i} = \langle I_1, P_i \rangle \ / \ \langle P_i, P_i \rangle, \quad i = 1..d.$$

Here $\langle P_i, P_i \rangle$ is an arbitrary inner product. We use the Legendre-Sobolev inner product, $\langle f, g \rangle = \int fg\mathrm{d}\lambda + \mu \int f'g'\mathrm{d}\lambda$ with $\mu = 1/8$. Similarly, we calculate coefficients for $I_2(\lambda)$ and obtain a $2d$-dimensional vector for each sample

$$(I_{1,1}, ..., I_{1,d}, I_{2,1}, ..., I_{2,d}).$$

Taking the second term in the expression for I_1 as precomputed, each coefficient of the approximation can be computed in time quadratic in d. Each coefficient of I_2 is computed in $O(d^4)$ operations. Since $d = 12$ in our experiments, the coefficients for both invariants are computed quickly. Note that one can also compute invariants of higher degree [6]. We expect, however, higher degree invariants to affect the classification rate only slightly, while introducing a noticeable computational overhead. For example, it would take $O(d^7)$ operations to calculate the coefficients of I_3.

Given the representation of a character in terms of Legendre-Sobolev coefficients of the invariant functions, we classify the sample based on the distance to the convex hulls of nearest neighbours in the same representation. Nearest neighbours are selected with Manhattan distance, which is the fastest distance we are aware of, requiring $2d - 1$ arithmetic operations, where d is the dimension. Distance to the convex hulls is evaluated with the square Euclidean distance, which takes $3d - 1$ operations.

Computation of the distance from a point to a convex hull is generally expensive. To simplify the problem, we consider the convex hulls to be simplices, since the number of nearest neighbours we compute is less than the dimension of the vector space and the points tend to be in generic position. If the points are not in generic position, we perform a slight perturbation with a minor affect on the distance. This procedure is recursively repeated to find the projection of the point on the smallest affine subspace that contains the simplex until the projection happens to be inside the simplex. On each iteration we represent the projection as a linear combination of the vertices with non-negative coefficients. The algorithm has complexity $O(d^4)$, where d is the dimension. It performs much faster in practice, however, because at each recursive call the dimension often drops by more than one [4].

As in [5], we select 20 classes closest to the Legendre-Sobolev coefficient vector of the integral invariants. To find the correct class among these, we solve the following minimization problem for each of these classes C_i:

$$\min_{\phi} \mathrm{CHNN}_k(X(\phi), C_i),$$

where $X(\phi)$ is the sheared image of the test sample curve X and $\mathrm{CHNN}_k(X, C)$ is the distance from a point X (in the Legendre-Sobolev space) to the convex hull of k nearest neighbors in class C.

Table 1: Maximum absolute and average relative errors in coefficients of invariants

	I_1		I_2	
Degree	Abs. Err.	Rel. Err.	Abs. Err.	Rel. Err.
2	9×10^{-12}	3×10^{-19}	3×10^{-11}	9×10^{-20}
3	1×10^{-11}	4×10^{-19}	8×10^{-10}	2×10^{-19}
4	5×10^{-11}	9×10^{-19}	1×10^{-9}	4×10^{-19}
5	6×10^{-11}	3×10^{-18}	3×10^{-9}	1×10^{-18}
6	3×10^{-10}	1×10^{-17}	9×10^{-9}	5×10^{-18}
7	2×10^{-9}	5×10^{-17}	7×10^{-8}	2×10^{-17}
8	3×10^{-8}	2×10^{-16}	1×10^{-7}	1×10^{-16}
9	2×10^{-7}	1×10^{-15}	6×10^{-7}	5×10^{-16}
10	2×10^{-6}	6×10^{-15}	4×10^{-6}	2×10^{-15}
11	5×10^{-6}	3×10^{-14}	2×10^{-5}	1×10^{-14}
12	1×10^{-5}	1×10^{-13}	7×10^{-5}	6×10^{-14}
13	4×10^{-5}	7×10^{-13}	5×10^{-4}	3×10^{-13}
14	3×10^{-4}	4×10^{-12}	3×10^{-3}	1×10^{-12}
15	1×10^{-3}	2×10^{-11}	7×10^{-3}	8×10^{-12}
16	1×10^{-2}	1×10^{-10}	2×10^{-2}	5×10^{-11}
17	5×10^{-2}	5×10^{-10}	3×10^{-2}	6×10^{-10}
18	4×10^{-1}	3×10^{-9}	3×10^{-1}	4×10^{-9}

It is not infeasible to solve the minimization problem by trying all possible angles, given that the precision of 1 degree is certainly sufficient for our purposes. Our error rates were calculated using this method. However, there are also more efficient methods. If we replaced the class C by a single point $(X_0, \ldots, X_d, Y_0, \ldots, Y_d)$ in the Legendre-Sobolev space of the coordinate functions, then we could find the minimum among the values of the distance at the boundary points of the interval of shear (i.e. the smallest and the largest admissible shear angles) and the stationary point

$$\varphi = \arctan \frac{\sum_k (X_k - x_k)}{\sum_k y_k}.$$

Furthermore, as the curve is sheared by different angles, the corresponding point in the Legendre-Sobolev space traces a straight line segment. This follows from the fact that, for each order i, the Legendre-Sobolev coefficient x_i remains unchanged, while y_i is multiplied by $t = \tan(\phi)$ which spans the interval $[\tan(\phi_{\min}), \tan(\phi_{\max})]$. Therefore we are looking at the problem of finding the point in the segment with the smallest CHNN distance. The problem can be reduced to that of the distance between two convex polyhedra. One can make various optimizations, given that one of the polyhedra is just a segment.

5. EXPERIMENTAL RESULTS

In this section we describe the experimental setting and estimate the approximation errors of representing I_1 and I_2 by truncated Legendre-Sobolev series, as described above. We then present classification results for different choices of size normalization and curve parameterization.

Our dataset of handwritten mathematical characters currently comprises 50,703 samples from 242 classes. This set includes 26,139 samples collected at the Ontario Research Centre for Computer Algebra (special mathematical char-

Table 2: Recognition rate (%) for shear from 0.0 to 0.9 radians, parameterized by affine arc length (AAL), arc length (AL) and time.

(a) Size normalization by height

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AAL	82.2	82.2	82.2	82.1	82.1	82.1	82.1	82.1	82.1	82.0
AL	96.4	96.4	96.1	95.6	95.0	94.1	93.0	91.9	90.2	88.0
Time	94.8	94.9	94.9	94.7	94.5	94.4	94.4	94.4	94.4	94.3

(b) Aspect ratio size normalization

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AAL	81.9	81.8	81.6	81.4	81.2	81.0	80.8	80.2	79.4	77.4
AL	96.3	96.4	96.1	95.5	94.7	93.7	92.3	90.1	85.7	77.5
Time	94.7	94.7	94.6	94.3	94.1	93.9	93.7	93.2	91.9	89.0

(c) Size normalization by I_1

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AAL	83.0	83.1	83.0	82.9	82.9	82.8	82.8	82.8	82.8	82.7
AL	**96.3**	**96.3**	**96.1**	**95.7**	**95.1**	**94.4**	**93.3**	**91.9**	**90.2**	**87.9**
Time	94.6	94.7	94.6	94.5	94.5	94.5	94.5	94.5	94.5	94.4

(d) Size normalization by I_1, without linear symbols

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AL	**96.7**	**96.7**	**96.5**	**96.2**	**95.7**	**95.0**	**94.1**	**92.7**	**91.2**	**88.9**

acters, Latin letters and digits), 14,802 samples from the UNIPEN database [13] (mostly digits), and 9,762 samples from the LaViola database [14] (mostly Latin letters and digits). All the samples are stored in a single file in InkML format. The number of strokes is included in the class labels (thus single-stroke and double-stroke "7" are considered as two different classes). Although this raises the total number of classes to 378, we have found that this gives better recognition rates compared to including the number of strokes in the feature vector [15].

Samples that appear unrecognizable even to a human were discarded to measure classification errors more meaningfully. At the same time we united classes that are very hard or impossible to distinguish without contextual information, such as x and $\times$; o, 0 and O. If, however, there was at least one sample in the class that could be recognized by a human with confidence, we retained the label of the class. As a result, we obtained 38,493 samples assigned to single classes, 10,224 to 2 classes, 1,954 to 3 classes, 19 to 4 classes, and 13 samples to 5 classes. Additional details of the experimental setting are given in [4]. To increase the precision of approximation of integral invariants, we precomputed some terms in the formulas for I_1 and I_2 in Maple [16] using rational arithmetic.

To estimate the error of approximation of integral invariants, we have computed the coefficients of the original samples and the same samples sheared by 1 radian. The results are summarized in Table 1 in terms of the maximum error and average relative error, defined as the quotient of the sum of absolute errors and the sum of absolute values, where degree is the degree of approximation.

We observed that there are a few characters that contribute the most to the maximum absolute error. These are geometrically linear samples, such as "–". For these samples, I_1 is close to identically zero, and therefore the normalization behaves unpredictably. Nevertheless, we have found that the 12-th degree of approximation of invariants provides sufficient accuracy for our purposes.

We have examined the error rate of the algorithm for different choices of parameterization of coordinate functions: arc length, time and affine arc length. We have also compared size normalization techniques described above. The results are shown in Figure 4 and Figure 5.

We notice that parameterization of coordinate functions by affine arc length results in relatively low recognition rate. Presence of second order derivatives makes it sensitive to sampling perturbations, even though it is invariant under special affine transformation. Parameterization by arc length, which is not invariant under shear, gives a lower recognition rate than parameterization by time for large distortions. However, for shear up to about 0.45 radians ($\approx$25 degrees) it yields a noticeably better classification rate.

6. SPACE-TIME PARAMETERIZATION AND FURTHER EXPERIMENTS

It is not surprising that parameterization by time gives better performance than parameterization by arc length for large affine distortion. Time is invariant under any transformation of the coordinate plane, while arc length is not. Indeed, arc length can be significantly distorted under horizontal shear, because vertical parts of the symbol get stretched while horizontal parts preserve their lengths. On the other hand, as has been seen previously [4] and confirmed in Tables 2(a)–2(d), arc length performs better when distortion due to affine transformations can be neglected. We therefore ask whether spatial and temporal parameterizations could be combined in a parameterization that would inherit advantages of both.

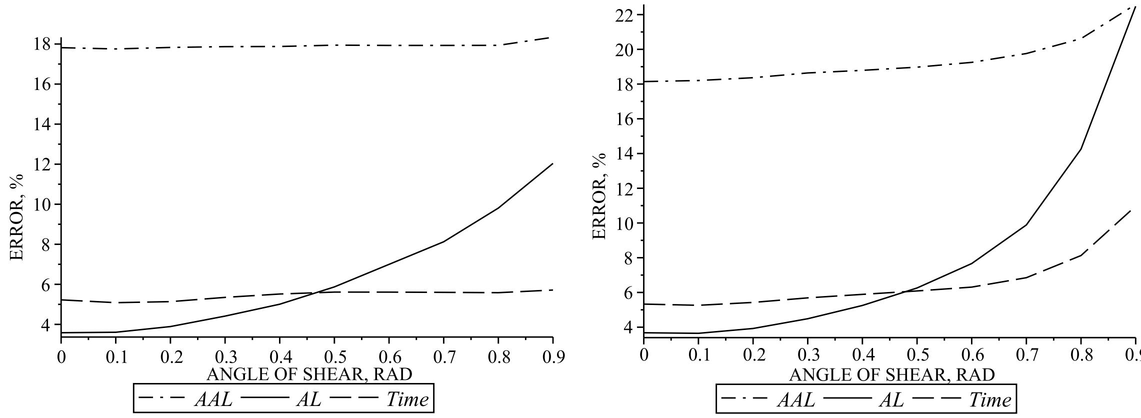

Figure 4: Error rate for size normalization by height (left) and with aspect ratio (right), parameterized by affine arc length (AAL), arc length (AL) and time.

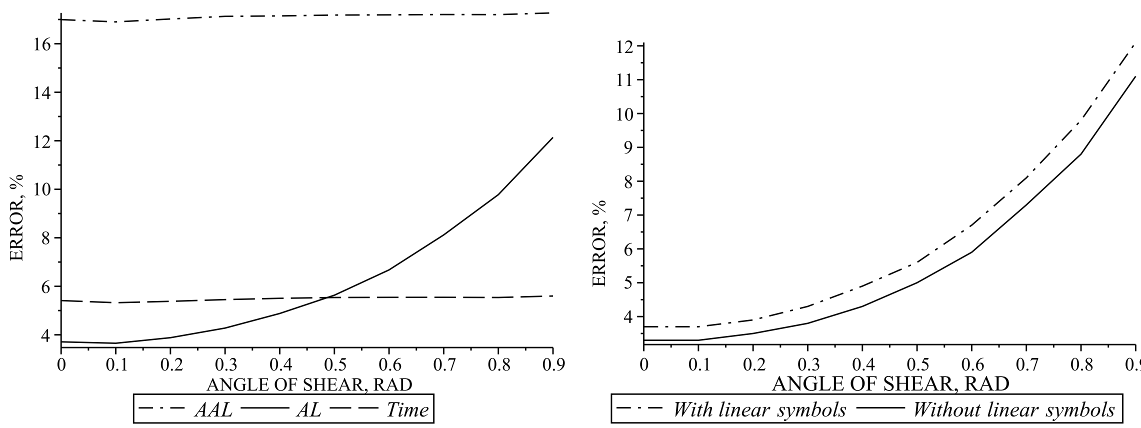

Figure 5: Error rate for size normalization with I_1 (left) and comparison of performance without linear samples, parameterized by arc length and size normalization with I_1 (right).

Because time is distorted mainly by local variations in speed, it is reasonable to expect the cumulative effect of such variations over longer time periods to be close to neutral. For arc length, the situation is reversed: large distortions can accumulate when the direction of the curve does not change. In other words, parameterization by time is more robust globally, and parameterization by arc length is more reliable locally. We therefore combine the two parameterizations as follows: divide the curve in N equal time intervals, and parameterize each interval by arc length. We furthermore notice that, if this algorithm is applied literally, abnormal behavior may occur at the end points of the N time intervals. To avoid this, we can smooth the transition from time to arc length with a mixed metric of the form

$$k\mathrm{d}t^2 + \mathrm{d}x^2 + \mathrm{d}y^2$$

inside the subintervals (here pure arc length would correspond to $k = 0$).

To obtain higher classification rate, we increased the number of classes selected by invariants to 50 (out of 378) and found the optimal values of $N = 2$ and $k = 2$ by cross-validation, see Table 3 for the whole dataset. A similar experiment was performed on the subset of the dataset from LaViola. Table 4 shows the results we obtained for LaViola samples. The results are plotted in Figure 6. We conclude that the mixed parameterization, coarse by time and fine by arc length, works as well as arc length for symbols not affected by affine distortions, while remaining almost as insensitive as time to large affine distortions.

We have discovered that size normalization by height gives the best classification rate under shear transformation (Table 2(a)). Such normalization, however, is not suitable for some other affine transformations, e.g. rotation. Normalization with aspect ratio (Table 2(b)), performs similarly to normalization by height at smaller degrees of shear, but the difference in classification rates becomes noticeable with the increase of deformation. Normalization by I_1 performs just as well as normalization by height (Table 2(c)) and remains invariant under affine transformations. We therefore consider size normalization with I_1 as the most suitable approach, if transformation of characters takes place. However, there is a special situation, when I_1 can not be used as the norm. Being an area between the curve and the secant, I_1 is close to zero if the curve happens to be close to a line;

Table 3: Recognition rate (%) for mixed parameterization for entire dataset.

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$N = 2$, $k = 1$	96.1	96.2	96.0	95.9	95.8	95.7	95.6	95.3	95.2	94.9
$N = 2$, $k = 2$	95.8	95.9	96.0	95.8	95.7	95.7	95.7	95.6	95.5	95.5
$N = 3$, $k = 0.5$	96.1	96.2	96.0	95.9	95.9	95.7	95.5	95.6	95.3	95.0
$N = 3$, $k = 1$	95.9	96.2	96.0	95.8	96.0	95.8	95.7	95.6	95.5	95.2
$N = 4$, $k = 1$	95.9	96.1	95.9	95.7	95.8	95.6	95.6	95.7	95.6	95.4

Table 4: Recognition rate (%) for mixed parameterization for LaViola component.

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$N = 2$, $k = 1$	98.4	98.5	98.5	98.4	98.3	98.4	98.3	98.3	98.2	98.0
$N = 2$, $k = 2$	98.4	98.5	98.4	98.3	98.4	98.3	98.3	98.3	98.2	98.2
$N = 3$, $k = 0.5$	98.3	98.3	98.3	98.2	98.3	98.2	98.3	98.2	98.1	98.0
$N = 3$, $k = 1$	98.4	98.4	98.4	98.4	98.4	98.4	98.3	98.3	98.3	98.2
$N = 4$, $k = 1$	98.4	98.5	98.4	98.4	98.3	98.3	98.3	98.3	98.2	98.2

dividing by the norm then appears to be the cause of partial misclassification, since it yields unpredictable results. This kind of symbol behaves strangely under affine transforms: essentially they can be transformed into a variety of shapes if stretched in the direction orthogonal to the line. This is similar to the way small symbols are affected by size normalization (a bloated period or comma can also look like a different sample). To test this hypothesis, we excluded linear characters such as "-", "\", "/", "l", ".". This excluded 1,442 symbols, or 3% of the total number of samples. We observed an increase in recognition rate of about 0.6% for the corresponding degrees of skew (Figure 5). We therefore consider that some of the methods, designed for 2-dimensional curves, are indeed not suitable for 1-dimensional symbols, and such linear characters require special treatment in classification algorithms.

7. TOWARD UNIFIED AFFINE- INVARIANT CLASSIFICATION

We have seen in [5] how to recognize symbols independently of orientation. We can now apply both rotation and shear to a sample and consider the two parameter minimization problem. We do not consider vertical shear, since this is not typical in handwriting. The solution to the two parameter problem can be found by computing partial derivatives with respect to the parameters and solving a system of two equations in two unknowns. This algorithm is independent of shear, rotation, scale and translation and should yield approximately the same recognition rate, as in the Table 2(c) for parameterization by arc length and time or as in the Table 3 for a mixed parameterization.

There is, however, an alternative to make an algorithm stable under transformations from the general affine group. Since I_1 performs poorly as a norm when it is close to zero, i.e., on linear and small symbols, the appropriate representation should combine normalized and un-normalized features. Perhaps one could put the size of a sample, the Legendre-Sobolev coefficients of approximated coordinate functions, and Legendre-Sobolev coefficients of invariants together in a feature vector. Each of these 3 parts should be scaled by a weight.

The weights can be symbol-dependent, such that for a small symbol, size should be weighted more, while all the rest of the vector less. For a linear symbol, the size and Legendre-Sobolev coefficients of the affine invariants should weight less, while the coefficients of the coordinate functions would be weighted more. For regular non-linear and non-small symbols, size should be weighted less, and the Legendre-Sobolev coefficients of the coordinate functions and invariants would be weighted similarly.

The question of how to calculate the weights prior to classification is important. If one doubles the size of a period or a comma, it may not appear as such anymore. But if size of a sample such as "O" is doubled in size it would still clearly be "O". Similarly, if a non-linear symbol is stretched in any direction by some factor, in most cases it would be classified as the same symbol. Therefore, we can consider a measure of "affinity" of a symbol with respect to a transformation group, as the degree to which the symbol can be transformed while still looking like the same symbol.

8. CONCLUSION

We have presented a classification algorithm that is invariant under shear, rotation, scaling and translation, the subset of affine transformations that are the most important in recognition of handwriting.

To achieve independence of scale we evaluated popular size normalization methods, but concluded that they are not suitable for the affine case. Instead we perform size normalization by normalizing the coefficient vector using the integral invariant of first order.

Another challenge was parameterization of coordinate functions. Parameterization by time gives stable recognition rate for different levels of distortion, while arc length is noticeably better for small transformations. To take advantage of both of these parameterizations we constructed a new mixed parameterization, by dividing the curve into equal time intervals and parameterizing each interval by the combined metric. We experimentally evaluated the parameters, finding those that make the mixed parameterization close to the result of parameterization by arc length, while being almost as invariant under distortions as parameterization by time.

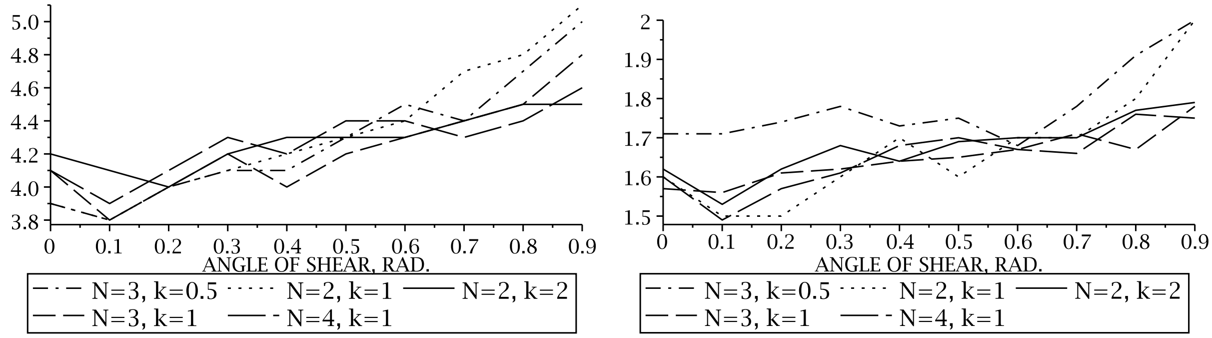

Figure 6: Error (%) for the mixed parameterization for different values of skew for the whole dataset (left) and for the LaViola component (right).

We have shown how to change the minimization problem to include rotation invariance. Finally, we have laid out a more general approach of extending the robustness to the full affine group for unified affine-invariant classification.

9. REFERENCES

[1] B.W. Char and S.M. Watt. Representing and characterizing handwritten mathematical symbols through succinct functional approximation. In *Proc. International Conference on Document Analysis and Recognition, (ICDAR)*, pp. 1198–1202. IEEE Computer Society, September 2007.

[2] O. Golubitsky and S.M. Watt. Online stroke modeling for handwriting recognition. In *Proc. 18th Annual International Conference on Computer Science and Software Engineering, (CASCON 2008)*, pp. 72–80. IBM Canada, October 2008.

[3] O. Golubitsky and S.M. Watt. Online computation of similarity between handwritten characters. In *Proc. Document Recognition and Retrieval XVI, (DRR 2009)*, volume 7247, pp. C1–C10. SPIE and IS&T, January 2009.

[4] O. Golubitsky and S.M. Watt. Distance-based classification of handwritten symbols. *International Journal of Document Analysis and Recognition*, 2010, DOI 10.1007/s10032-009-0107-7.

[5] O. Golubitsky, V. Mazalov, and S.M. Watt. Orientation-independent recognition of handwritten characters with integral invariants. In *Proc. Joint Conference of ASCM 2009 and MACIS 2009*, pp. 252-261, COE Lecture Note Vol. 22, Kyushu University, ISSN 1881-4042.

[6] S. Feng, I. Kogan, and H. Krim. Classification of curves in 2d and 3d via affine integral signatures. *Acta Applicandae Mathematicae*, 2010, DOI 10.1007/s10440-008-9353-9.

[7] T. Wakahara and K. Odaka. On-line cursive Kanji character recognition using stroke-based affine transformation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 19, pp. 1381–1385. IEEE Computer Society, December 1997.

[8] T. Wakahara and S. Uchida. Hierarchical decomposition of handwriting deformation vector field using 2D warping and global/local affine transformation. In *Proc. 10th International Conference on Document Analysis and Recognition*, pp. 1141–1145. IEEE Computer Society, July 2009.

[9] Q. Huo and T. He. A minimax classification approach to HMM-based online handwritten Chinese character recognition robust against affine distortions. In *Proc. 9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, pp. 1226–1230. IEEE Computer Society, July 2007.

[10] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, volume 9, pp. 171–185. 1995.

[11] J. Flusser and T. Suk. Character recognition by affine moment invariants. In *Proc. 5th International Conference on Computer Analysis of Images and Patterns*, pp. 572–577. Springer-Verlag, 2007.

[12] C. Liu, K. Nakashima, H. Sako, H. Fujisawa. Handwritten digit recognition: investigation of normalization and feature extraction techniques. *Pattern Recognition*, volume 37, pp. 265–279. Elsevier, 2004.

[13] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet. UNIPEN project of online data exchange and recognizer benchmarks. In *Proc. 12th International Conference on Pattern Recognition (ICPR 1994)*, pp. 29–33. IAPR-IEEE, 1994.

[14] J.J. LaViola. Symbol recognition dataset. Microsoft Center for Research on Pen-Centric Computing. http://pen.cs.brown.edu/symbolRecognitionDataset.zip

[15] O. Golubitsky, S.M. Watt. Online recognition of multi-stroke symbols with orthogonal series. In *Proc. 10th International Conference on Document Analysis and Recognition, (ICDAR 2009)*, pp. 1265–1269. IEEE Computer Society, 2009.

[16] Maple 13 user manual. Maplesoft, 2009.