Orientation-Independent Recognition of Handwritten Characters with Integral Invariants

Oleg Golubitsky, Vadim Mazalov and Stephen M. Watt

The University of Western Ontario
London, Ontario, Canada N6A 5B7
ogolubit@uwo.ca, vmazalov@uwo.ca, watt@uwo.ca

Abstract

We present an approach to recognize handwritten characters independently of their orientation. The method is based on the theory of integral invariants and yields good results in classifying rotated samples. We propose two recognition techniques taking advantage of integral invariants up to second order. Truncated Legendre-Sobolev series are used to represent the invariant functions and recognition is based on proximity to the local convex hulls of known classes. We compare performance of these new methods with another widely used recognition method based on geometric moment invariants. The results obtained indicate that integral invariants give better recognition rates with less computation, confirming they are suitable for classification of rotated handwritten characters in a pen-based environment.

1 Introduction

We are interested in robust methods for the recognition of handwritten mathematical symbols. We view the trace of the symbol, as it is written, as a two-dimensional curve made up of a number of continuous segments and treat recognition as a classification problem. This subject of classifying two-dimensional parametric curves has been gaining importance in recent years. Moreover, mathematical handwriting recognition has received increasing attention with the popularity of hand-held mobile and digital tablet devices [1]. In this setting the accuracy and speed of an online character classification algorithm is important.

There are a number of factors that give the recognition of mathematics additional challenges beyond that of normal text recognition. Among these, we can observe the relatively large "alphabet" of similar looking few-stroke symbols. It is normally the case that symbols tend to be well isolated. There is no fixed dictionary of multi-symbol "words," but it is possible to identify expressions that occur more often in particular fields [2]. In addition, mathematical expressions are two-dimensional objects and the placement of symbols is important in contextual analysis [3]. Character classification algorithms for handwritten mathematics recognition therefore need special consideration.

It has been shown earlier how to classify a curve represented by truncated expansions of its coordinate functions in orthogonal bases [4, 5, 6, 7, 8]. Different bases have been considered, including Chebyshev, Legendre and Legendre-Sobolev bases. Most recently, attention of this research program has focused on Legendre-Sobolev series. These are easy to compute and provide an useful distance measure in the first jet space, taking derivatives into account. Test results confirm that this technique is indeed effective and allows to achieve 97.5% recognition rate.

We now address the problem that the recognition rate may be undermined by the variation in orientation of individual symbols. This may be more of a problem when symbols are written in a well-separated manner than when text is written cursively. Orientation variation is usually addressed by "de-slanting" symbols with a transformation on the coordinate space. One difficulty with this approach in a mathematical handwriting setting is

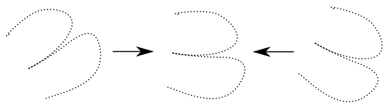

Figure 1: Rotation of a symbol

that different characters may require different correction and the degree of correction is not known in advance. With mathematical handwriting, it can be difficult to detect dominant orientation from symbol features.

Different solutions have been proposed, usually dealing with *ad hoc* rotation of a character after it is completely written (Figure 1). This rotation, as well as symbol resizing, are performed during a normalization stage in most of the online techniques. We propose a different approach: rather than rotating a sample by some estimated amount, we compute from the sample certain functions that are invariant under rotation. We ask to what extent these transformations affect the classification rate and present new algorithms for classifying symbols in the presence of such transformations. We consider methods using classification with integral invariants (CII) and classification with coordinate functions and integral invariants (CCFII). For these we use the theory of integral invariants of parametric curves [9]. To objectively evaluate recognition rate of the proposed techniques, we compare to a similar algorithm that uses geometric moment functions for the rotation-independent classification. We call this last method classification with coordinate functions and moment invariants (CCFMI).

In our methods, curves are represented as points in a vector space formed by the coefficients of their approximating truncated Legendre-Sobolev series. For CII, we take the integral invariants as the curves to be approximated and look for nearest classes in a manner we describe below. For CCFII, the top N classes are selected with integral invariants, then the sample is rotated to determine the angle which gives minimal distance based on coordinate curves. The CCFMI method is similar to CCFII except that it computes geometric moment invariants to obtain top N candidates. The proposed algorithms are online in the sense that most of the computation is performed while the sample is written, with minor overhead after pen-up. The algorithms are as well independent of translation and scaling, which is achieved by dropping the constant terms from the series and by normalizing the coefficient vectors, respectively.

This paper is organized as follows. In Section 2 we summarize the theory of integral invariants as applied in our algorithms. In Section 3 we outline concepts of geometric moments and give examples of moment invariants. Algorithms based on integral invariants are described in detail in Section 4. In Section 5 we present the CCFMI algorithm, relying on explanations in Section 4. Experimental settings and performance comparison are given in Section 6. In the conclusion we discuss causes of misclassification and outline directions for improvement of the proposed methods.

2 Integral Invariants

Integral invariants provide an elegant approach to planar and spatial curve classification under affine transformations. In terms of handwriting recognition, a symbol is given as a parameterized piecewise continuous curve defined by a discrete sequence of points. For a symbol we compute certain integral quantities from the coordinate functions, which are

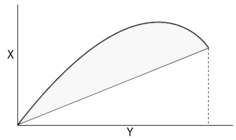

Figure 2: Geometric representation of the integral invariant of the first order

Figure 3: Ambiguity, introduced by shear and rotation

then also functions of the curve parameterization. Exposing the sample to transformations results in the same invariant functions. As opposed to differential invariants, such integral invariants are relatively insensitive to small perturbations, and are therefore applicable to classification of handwritten characters with sampling noise.

As the name suggests, integral invariant functions are constructed from quantities obtained by integration. We consider the following invariants, defined in terms of the coordinate functions $X(\lambda)$ and $Y(\lambda)$:

$$I_0(\lambda) = \sqrt{X^2(\lambda) + Y^2(\lambda)} = R(\lambda),$$

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

$$I_2(\lambda) = X(\lambda)\int_0^\lambda X(\tau)Y(\tau)dY(\tau) - \frac{1}{2}Y(\lambda)\int_0^\lambda X^2(\tau)dY(\tau) - \frac{1}{6}X^2(\lambda)Y^2(\lambda).$$

In our case $X(\lambda)$, $Y(\lambda)$ are parameterized by Euclidean arc length. The invariant $I_1(\lambda)$ can be interpreted geometrically as the area between the curve and its secant (Figure 2). The derivation of these is developed in [9].

Functions $I_1(\lambda)$ and $I_2(\lambda)$ are invariant under $SL(2)$, the group of special linear transformations, while $I_0(\lambda)$ is invariant under the action of the special orthogonal group $SO(2)$. An invariant under the full affine group can be constructed as the quotient $I_2(\lambda)/I_1^2(\lambda)$. This is, however, less stable to compute than I_2. Instead, by translating the origin and normalizing the size of the sample we can restrict our attention to the $SL(2)$ invariants without loss of generality.

Among the actions of special linear group, two are of particular interest in handwriting recognition: rotation and shear transformations. The latter, in its full generality, is a separate nontrivial problem and is not considered in this paper. One of the difficulties is in choosing the appropriate shear-invariant parameterization of the coordinate functions. Another problem, arising with mathematical alphabets, is the requirement of careful analysis of transformation limits to avoid blending classes. For example, a sample L (which initially can be confused with $\lfloor$), when exposed to shear transformation, becomes a subject to misclassification with $\angle$. If, in addition, we allow arbitrary rotation and scaling of the character, the set of matching candidates will include $<$, $>$, 7, V, $\bigwedge$, $\rceil$, $\surd$, Γ and $\wedge$ (Figure 3). Therefore, our attention is currently drawn to the action of the subgroup $SO(2)$. We note that in practice shear is seen most on tall, thin symbols and this can to an extent be corrected by rotation.

The invariant representation of a symbol curve is approximated with Legendre-Sobolev polynomials. Coefficients of the truncated Legendre-Sobolev expansion are used to construct a point for each character in a training set annotated with ground truth. Classification is performed based on the distance from the test point to the convex hull of points in the nearest classes.

Our experiments show that using I_2 gives only a minor increase in the recognition rate (about 1%) in CII, but increases the computational cost significantly. As for CCFII, the nature of the algorithm makes the accuracy of the integral invariant classification less critical. Invariant functions are used only for the purpose of selecting top N candidate classes, which are subsequently analyzed. The function obtained with invariants I_0 and I_1 was therefore chosen as sufficient.

3 Geometric Moments

Similar to integral invariants, moment invariants provide a framework to describe curves independently of orientation. Among moment functions one can select geometric, Zernike, radial and Legendre moments [10]. For the purpose of online curve classification under pressure of computational constraints, geometric moments are of special interest since they are easy to calculate, while invariant under scaling, translation and rotation.

Having been introduced by Hu [11], geometric moments are widely used for shape and pattern classification [10, 12, 13]. A $(p+q)$-th order moment of f can be expressed as

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y)$$

In general, translation invariance is achieved by computing central moments

$$\mu_{pq} = \sum_x \sum_y (x-x_0)^p (y-y_0)^q f(x,y),\ x_0 = \frac{m_{10}}{m_{00}} \text{ and } y_0 = \frac{m_{01}}{m_{00}}$$

and scale normalization is performed as

$$\eta_{pq} = \mu_{pq}/(\mu_{00})^{(p+q+2)/2}$$

The first three moment invariants are derived from algebraic invariants and can be represented as

$$M_1 = \eta_{20} + \eta_{02}, \quad M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \quad M_3 = \eta_{20}\eta_{02} - \eta_{11}^2.$$

Independence of orientation of the above expressions can be verified by substitution with the geometric moments obtained after rotation transformation

$$\begin{aligned}
m'_{20} &= \frac{1+\cos 2\alpha}{2} m_{20} - \sin 2\alpha\ m_{11} + \frac{1-\cos 2\alpha}{2} m_{02},\\
m'_{11} &= \frac{\sin 2\alpha}{2} m_{20} + \cos 2\alpha\ m_{11} - \frac{\sin 2\alpha}{2} m_{02},\\
m'_{02} &= \frac{1-\cos 2\alpha}{2} m_{20} + \sin 2\alpha\ m_{11} + \frac{1+\cos 2\alpha}{2} m_{02}.
\end{aligned}$$

One can omit translation and scale normalization of moments by normalizing a sample's coordinates first. In this case the moment invariants are derived in terms of moments m_{pq}.

4 CII and CCFII

Consider the coordinate functions $X(\lambda)$ and $Y(\lambda)$ of a single- or multi-stroke sample. Multi-stroke symbols are represented by the coordinate functions of consecutively joined strokes. The first step to approximate $X(\lambda)$ and $Y(\lambda)$ as truncated series in basis of Legendre-Sobolev polynomials. These polynomials are orthogonal with respect to Legendre-Sobolev inner product

$$\langle f, g\rangle = \int_a^b f(\lambda)g(\lambda)d\lambda + \mu \int_a^b f'(\lambda)g'(\lambda)d\lambda$$

where the functions $f(\lambda)$ and $g(\lambda)$ are differentiable on the interval $[a, b]$, μ is a numeric parameter and can be chosen experimentally. It has been shown in [14] that $\mu = 1/8$ yields good classification results.

Let x_0, x_1,...,x_d be the coefficients of the approximation for $X(\lambda)$ and similarly for $Y(\lambda)$. Note, that these coefficients are computed while the curve is written with a small constant time overhead after pen-up [7]. We take $d = 12$, because it allows us to achieve accurate enough approximation with error unnoticeable to a human [8].

Since the first polynomial (for any inner product) is 1, point (x_0, y_0) can be thought of as the curve's center. We can therefore normalize the curve with respect to position by simply discarding the first coefficients. Scale normalization is performed by normalizing the vector $(x_1, ..., x_d, y_1, ..., y_d)$, taking advantage of the fact that the norm of the vector is proportional to the size of the curve, to obtain $(\bar{x}_1, ..., \bar{x}_d, \bar{y}_1, ..., \bar{y}_d)$.

With this approximation, the integral invariant functions take the form

$$I_0(\lambda) = \sqrt{\left(\sum_{i=1}^{d} \bar{x}_i P_i(\lambda)\right)^2 + \left(\sum_{i=1}^{d} \bar{y}_i P_i(\lambda)\right)^2},$$

$$I_1(\lambda) = \sum_{i,j=1}^{d} \bar{x}_i \bar{y}_j \left[\int_0^\lambda P_i(\tau)P_j'(\tau)d\tau - \frac{1}{2}P_i(\lambda)P_j(\lambda)\right]$$

Here P_i denotes the i-th Legendre-Sobolev polynomial.

A similar process of approximation is then applied to the invariant functions, yielding a 24-dimensional vector for each sample $(\bar{I}_{0,1}, ..., \bar{I}_{0,d}, \bar{I}_{1,1}, ..., \bar{I}_{1,d})$. Taking the second term in the expression for $I_1(\lambda)$ as precomputed, the Legendre-Sobolev coefficients can be calculated quickly, in time quadratic in d. The coefficients for $I_0(\lambda)$ are computed in the same way.

Classification is based on evaluation of the distance from the sample to the convex hulls of the nearest neighbours and selecting the classes with the smallest distance. Different distance measures were considered in previous work [8] with emphasis on fast computation. Manhattan distance was chosen as the most efficient for pre-classification (selecting the nearest neighbours), while square Euclidean distance gave a lower error rate when used as the distance from a sample to the convex hulls.

Computing the distance from a point to a convex hull can be expensive. We were able to specialize the problem by taking the convex hull to be a simplex, since the number of nearest neighbours in our algorithms is less than dimension of the vector space and the points are in generic position. If the points are not in generic position, a minor perturbation is performed, only slightly affecting the distance. We then apply the algorithm recursively to find the projection from the point on the smallest affine subspace containing the simplex, until the projection happens to be inside the simplex. On each iteration the projection is expressed as a linear combination of the vertices of the simplex, considering the vertices with non-negative corresponding coefficients. The complexity of this algorithm is $O(N^4)$, where N is the dimension. In practice it performs much faster, since at each recursive call the dimension often drops by more than one [14].

The CII algorithm relies on approximation of the invariant functions, as described above. We select the class closest to the sample in the space of coefficients of truncated polynomial series. The algorithm does not depend on the number of classes, since only one class is considered.

As an alternative, in CCFII the coefficients $(\bar{I}_{0,0}, \ldots, \bar{I}_{0,d}, \bar{I}_{1,0}, \ldots, \bar{I}_{1,d})$ are used to select the closest N candidate classes. The value for N may be determined empirically to ensure high probability of the correct class being within the ones chosen. Having a fixed small number of classes with the correct class among them, we evaluate the minimal distance from the sample to each class with respect to various sample rotations. This procedure gives correct class as well as the rotation angle. The angle is determined as the solution to the minimization problem

$$\min_{\alpha} \left(\sum_k (X_k - (x_k \cos\alpha + y_k \sin\alpha))^2 + \sum_k (Y_k - (-x_k \sin\alpha + y_k \cos\alpha))^2 \right),$$

where X_k, Y_k are the coefficients of the Legendre-Sobolev approximation of the coordinate functions of the training symbols, and x_k, y_k are the coefficients of the test sample. The global minimum is selected among the output of the function at the boundary points of the closed interval α and at the stationary point

$$\alpha = \arctan\left(\frac{\sum_k (X_k y_k - Y_k x_k)}{\sum_k (X_k x_k + Y_k y_k)}\right).$$

5 CCFMI

The $(p+q)$-th moment functions of a sample's coordinates can be expressed as

$$m_{pq}(\lambda_\ell) = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} X(\lambda_i)^p Y(\lambda_j)^q f\big(X(\lambda_i), Y(\lambda_j)\big)$$

where $X(\lambda_i)$ and $Y(\lambda_i)$ are the coordinates X and Y at sample point i. We have taken the intensity function to be of the form $f\big(X(\lambda_i), Y(\lambda_j)\big) = \sqrt{X(\lambda_i)^2 + Y(\lambda_j)^2}$ and work directly with moments, since normalization with respect to size and position is already performed in the algorithm. Specifically, we tested the following rotation invariants

$$M_0(\lambda) = m_{00}(\lambda),$$
$$M_1(\lambda) = m_{20}(\lambda) + m_{02}(\lambda),$$
$$M_2(\lambda) = (m_{20}(\lambda) - m_{02}(\lambda))^2 + 4m_{11}(\lambda)^2.$$

As in CCFII, CCFMI selects the top N classes with rotation invariant functions. To make a fair comparison, we considered the classification rate for two combinations of moment invariants: $M_0(\lambda)$, $M_1(\lambda)$ and $M_1(\lambda)$, $M_2(\lambda)$. Classification with $M_1(\lambda)$, $M_2(\lambda)$ in general gave 3% higher error rate. We therefore focused our attention on improving the recognition rate of $M_0(\lambda)$ and $M_1(\lambda)$ by variation of number of classes and number of nearest neighbours. Details of our experiments are described in the following section.

Table 1: Presence (%) of the correct class within the top N classes, CCFII

$N=1$	2	3	4	5	6	7	10	15	**20**	25
87.9	95.1	96.8	97.7	98.3	98.7	98.9	99.4	99.5	**99.5**	99.5

Table 2: Error rate (%), depending on number of nearest neighbours, CCFII

angle (radians)	$K=8$	10	12	14	16	18	19	**20**	21	22
0	4.4	3.9	4.2	4.0	3.9	3.9	3.8	**3.7**	3.8	3.8
0.3	6.2	5.7	5.7	5.4	5.4	5.4	5.3	**5.3**	5.4	5.4
0.5	7.4	6.9	6.8	6.7	6.6	6.5	6.4	**6.4**	6.5	6.5
0.7	8.5	7.9	7.7	7.6	7.4	7.4	7.2	**7.2**	7.3	7.4
0.9	9.3	8.8	8.6	8.3	8.2	8.2	8.2	**8.1**	8.2	8.2
1.1	9.6	9.0	8.7	8.6	8.4	8.4	8.2	**8.2**	8.4	8.4
average	7.5	7.0	7.0	6.8	6.6	6.6	6.5	**6.5**	6.6	6.6

6 Experimental Details and Evaluation of Results

Our dataset comprised 50,703 handwritten mathematical symbols from 242 classes. All samples were represented in a uniform InkML format [15] and stored in a single file. Each symbol definition included the number of strokes and the (X, Y) coordinates of the trace sample points. For some symbols we also had information about timing, pen-up strokes, pen pressure and context (the formula containing the symbol). This additional information was not used in the present experiment.

All symbols had been inspected visually in order to discard samples unrecognizable by a human. Symbols that looked to a human reader as belonging to more than one class were labeled with all those classes. Classes that were indistinguishable without context were merged. For example, we united the classes "Capital O, little o, omicron, zero", capital Greek letters with Latin analogues, etc. As a result, 38,493 samples had one class label, 10,224 samples had 2 class labels, 1,954 samples had 3, 19 samples had 4, and 13 samples had 5. This resulted in 378 composite sets.

To implement 10-fold cross-validation we randomly divided the dataset into 10 parts, preserving the proportions of class sizes. The normalized Legendre-Sobolev coefficient vectors of coordinate functions of randomly rotated symbols, as well as coefficients of integral invariants were pre-computed for all symbols. See [8] for more detailed description of the experimental settings.

According to our tests, CII gives a 88% recognition rate. This recognition rate does not depend on the angle to which test samples are rotated. Neither does the frequency of occurrence of the correct class in the top N classes depend on rotation angle.

To measure the best performance of CCFII, we first determined experimentally the number N of top classes required to contain the correct class most of the time. The results obtained are shown in the Table 1. We find $N = 20$ to be an appropriate balance between accuracy and the complexity introduced by integral invariants. With fixed N, the relationship between the number of nearest neighbours K and the error rate for different

Table 3: Presence (%) of the correct class within the top N classes, CCFMI

$N=1$	2	3	4	5	10	20	30	40	**50**	55
51.5	68.3	77.2	82.2	85.9	95.3	98.8	98.9	99.0	**99.0**	99.0

Table 4: Error rate (%), depending on number of nearest neighbours, CCFMI

angle (radians)	$K=8$	10	12	14	16	18	20	21	**22**	23
0	7.0	6.6	6.4	6.2	6.1	6.1	5.9	5.8	**5.8**	6.0
0.3	8.0	7.8	7.6	7.4	7.2	7.1	7.0	7.2	**7.1**	7.2
0.5	9.3	9.1	8.9	8.5	8.3	8.3	8.2	8.2	**8.1**	8.3
0.7	10.4	10.1	9.9	9.5	9.4	9.2	9.1	9.2	**9.2**	9.3
0.9	11.5	11.1	10.8	10.4	10.2	10.2	10.1	10.0	**10.0**	10.0
1.1	11.4	11.1	10.7	10.4	10.2	10.1	10.1	10.0	**10.0**	10.0
average	9.6	9.3	9.1	8.7	8.6	8.5	8.4	8.4	**8.4**	8.5

Table 5: Error rates of CII, CCFII and CCFMI

α, rad.	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1
CII	12	12	12	12	12	12	12	12	12	12
CCFII	**3.7**	**3.9**	**4.5**	**5.3**	**5.9**	**6.4**	**6.6**	**7.2**	**8.2**	**8.2**
CCFMI	5.8	5.9	6.5	7.1	7.7	8.1	8.7	9.2	10	10

angles is shown in the Table 2. We observed that CCFII gives the best recognition rate for $K = 20$. In this framework, CCFII's rate starts at 96.3% for non-rotated samples and decreases slightly with increase in angle, but never approaches CII (see Table 5).

A similar approach was taken to measure the performance of CCFMI. We first measured the number of classes ($N = 50$) required to contain the correct class most of the time (Table 3) and then found the K that yields the best classification results (Table 4).

A comparison of the performance of CCFII and CCFMI is presented in Figures 4 and 5. Relative classification results are shown in Table 5 and Figure 6. We see that CCFII has a better error rate, while requiring fewer candidate classes and fewer nearest neighbour computations.

7 Conclusion

We have presented methods to classify handwritten characters, independently of orientation, based on integral invariants and have compared them with classification using geometric moment invariants. We have observed that integral invariants perform better while requiring less computation. We therefore conclude that integral invariants are a suitable instrument in the recognition of handwritten characters when orientation is uncertain.

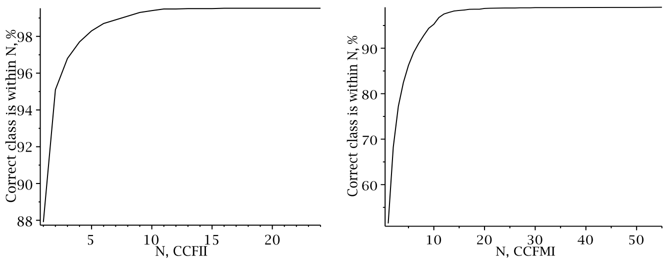

Figure 4: Presence of the correct class within N for CCFII (left) and CCFMI (right)

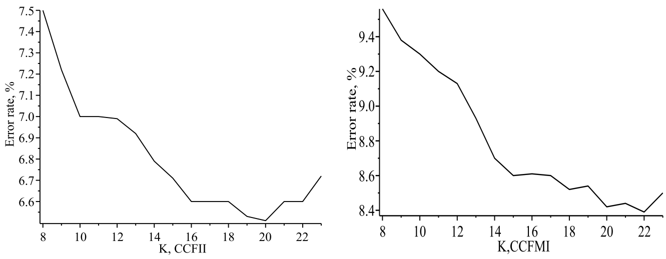

Figure 5: Error rate for different K for CCFII (left) and CCFMI (right)

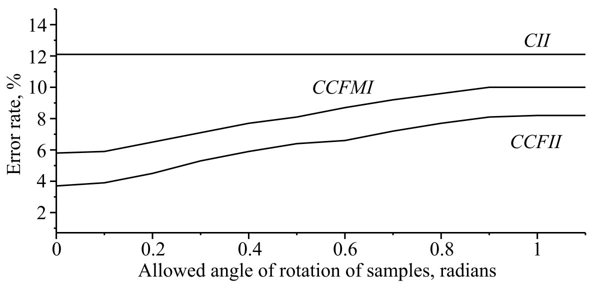

Figure 6: Error rates of CII, CCFII, CCFMI

As expected, we noticed an increase in error rate with the rotation angle for CCFII and CCFMI (Figure 6). The typical misclassifications that arise are when distinct symbols have similar shape and are normally distinguished by their orientation, for example "1" and "/", "+" and "×", "U" and "⊂". As a possible solution to this, a system could consider the tendency to write characters in similar orientations and restrict the range of angles for nearby symbols. A technique similar to CCFII can be applied to classify symbols as part of an expression with small adjustments to the minimization function. This approach has a number of other benefits, such as contextual and notational analysis, and should be considered as a logical continuation of the work presented.

References and Notes

[1] E. Smirnova and S.M. Watt, Communicating Mathematics via Pen-Based Computer Interfaces, In *Proc. 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, (SYNASC 2008)*, IEEE Computer Society, pp. 9-18, 2008.

[2] S.M. Watt, An Empirical Measure on the Set of Symbols Occurring in Engineering Mathematics Texts, In *Proc. 8th IAPR International Workshop on Document Analysis Systems, (DAS 2008)*, IEEE Computer Society, pp. 17-19, 2008.

[3] E. Smirnova and S.M. Watt, Context-Sensitive Mathematical Character Recognition, In *Proc. IAPR International Conference on Frontiers in Handwriting Recognition, (ICFHR 2008)*, pp. 604-610, 2008.

[4] 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)*, IEEE Computer Society, pp. 1198-1202, 2007.

[5] O. Golubitsky and S.M. Watt, Confidence Measures in Recognizing Handwritten Mathematical Symbols, In *Proc. Conferences on Intelligent Computer Mathematics 2009*, Springer Verlag LNCS 5625, pp. 460-466, 2009.

[6] O. Golubitsky and S.M. Watt, Online Computation of Similarity between Handwritten Characters, In *Proc. Document Recognition and Retrieval XVI, (DRR 2009)*, SPIE and IS&T, ISBN 9780819474971, ISSN 0277-786X, pp C1-C10, 2009.

[7] 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)*, IBM Canada, ISSN 1705-7345, pp. 72-80, 2008.

[8] O. Golubitsky and S.M. Watt, Distance-Based Classification of Handwritten Symbols, Technical Report, Ontario Research Center for Computer Algebra, `http://www.orcca.on.ca/TechReports/TechReports/2009/TR-09-03.pdf`.

[9] S. Feng, I. Kogan and H. Krim, Classification of Curves in 2D and 3D via Affine Integral Signatures, to appear in *Acta Appl. Math.*, 2008, `http://arxiv.org/abs/0806.1984`.

[10] R. Mukundan, K. R. Ramakrishnan, Moment Functions in Image Analysis: Theory and Applications, Vol. 37, Number 6, pp.15-19, World Scientific, 1998.

[11] M.K. Hu, Visual Pattern Recognition by Moment Invariants, IRE Transactions on Information Theory, Vol. IT-8, pp. 179-187, 1962.

[12] C.L. Liu, K. Nakashima, H. Sako and H. Fujisawa, Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques, Pattern Recognition, Vol. 37, Issue 2, pp. 265-279, 2004.

[13] M. Mercimek, K. Gulez and T.V. Mumcu, Real Object Recognition Using Moment Invariants, Sadhana, Vol. 37, Number 6, pp.765-775, 2005.

[14] O. Golubitsky and S.M. Watt, Tie Breaking for Curve Multiclassifiers, ORCCA Technical Report TR-09-02, `http://www.orcca.on.ca/TechReports/2009/TR-09-02.pdf` .

[15] Ink Markup Language (InkML) W3C Working Draft, `http://www.w3.org/TR/InkML`, 2006.