Using mixtures of deformable models to capture variations in hand printed digits

Michael Revow, Christopher K. I. Williams and Geoffrey E. Hinton
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4

ABSTRACT

Deformable models are an attractive way for characterizing handwritten digits since they have relatively few parameters, are able to capture many topological variations, and incorporate much prior knowledge. We have described a system [8] that uses learned digit models consisting of splines whose shape is governed by a small number of control points. Images can be classified by separately fitting each digit model to the image, and using a simple neural network to decide which model fits best. We use an elastic matching algorithm to minimize an energy function that includes both the deformation energy of the digit model and the log probability that the model would generate the inked pixels in the image. The use of multiple models for each digit can characterize the population of handwritten digits better. We show how multiple models may be used without increasing the time required for elastic matching.

1 Introduction

The goal of achieving close to human performance in recognizing hand printed digits remains elusive. A major reason for this failure has been the inability to successfully characterize the wide diversity inherent in handwritten digits, resulting from factors such as regional styles, differing writing instruments and psycho-motoric effects [11]. One way to categorize the range of model-based approaches to handling this diversity is to consider the complexity of the procedure used to match model to data (figure 1). Generally as the complexity of the matching increases, the number of models needed decreases. At one end of the spectrum one could imagine a system that stores a large number of different instances of each digit and performs pure template matches to find the digit instance closest to the image. Unless the images are accurately normalized before matching, this involves a huge number of matches. This number can be significantly reduced by considering affine transformations of digit instances. A recognizer [10] trained to be tolerant to small affine tranformations of the input image had better generalization compared to one that had no explicit knowledge about affine transformations. At the other end of the spectrum are elastic deformable matches [3, 12] which attempt to capture all variations with a single model, but using a much more complex matching scheme.

We have described an elastic model [8], based on splines and containing just a few parameters which manages to capture many of the variations of a given digit. Each elastic model contains parameters that define an ideal shape and a deformation energy for departures from this ideal. Affine transformations are not counted as deformations since they

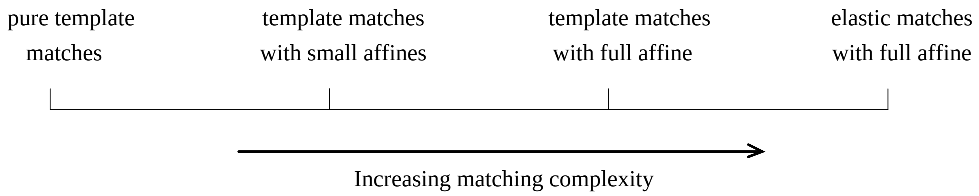

Figure 1: A spectrum of approaches to handling diversity in handwritten character recognition. Moving from left to right across the spectrum, matching complexity increases, while the number of matches decreases.

preserve shape. Our original description used a single model for each digit-class. While being able to capture many variations, it had difficulty characterizing very unusual digit styles. After briefly reviewing this model, we describe a simple extension involving more than one model per digit-class but with very little additional computational overhead. We believe that this allows us to better *characterize* the variations in the instances of each digit while not increasing the matching complexity.

2 Elastic models

Each digit is modelled by a deformable spline whose shape is determined by the positions of at most 8 control points.[1] Every point on the spline is a weighted average of four control points, with the weighting coefficients changing smoothly as we move along the spline. In computing the weighting coefficients we use a cubic B-spline and treat the first and last control points as if they were doubled. To generate an ideal example of a digit we put the control points at their home locations. Deformations are incurred as the control points move away from their home locations. The control points are assumed to have independent, radial gaussian distributions about their home locations. So the negative log probability of a deformation is proportional to the sum of the squares of the departures of the control points from their home locations.

Using a spline it is easy to model topological variants of a digit. The loop of a 2, for example, can smoothly turn into a cusp or an open bend. These variants are produced by small changes in the relative locations of the relevant control points. This advantage of spline models is pointed out by [7] who use a different kind of spline that they fit to on-line character data by directly locating candidate control points in the image.

The deformation energy function only penalizes shape *deformations*. Translation, rotation, dilation, elongation, and shear do not change the shape of an object so we want the deformation energy to be invariant under these affine transformations. We achieve this by giving each model its own "object-based frame" and computing the deformation energy relative to this frame. When we fit the model to data, we repeatedly recompute the best affine transformation between the object-based frame and the image (see section 3). The repeated recomputation of the affine transform during the model fit means that the shape of the digit is influencing normalization. Having an explicit representation of the affine transformation of each digit should prove very helpful for recognizing multiple

[1] Some digits can be adequately modelled with less control points, for example the one model needs only 3 control points.

digits, since it will allow us to impose a penalty on differences in the affine transformations of neighbouring digits. Its use in isolated digit recognition is discussed in section 4.

Although we use our digit models for recognizing images, it helps to start by considering how we would use them for generating images. The generative model is an elaboration of the probabilistic interpretation of the elastic net given in [5]. To generate a noisy image of a particular digit class, run the following procedure:

- Pick a deformation of the model (i.e. move the control points away from their home locations). This defines the spline in object-based coordinates. The *log* probability of picking a deformation is proportional to the sum of squares of the deformations.
- Pick an affine transformation[2] from the model's intrinsic reference frame to the image frame (i.e. pick a size, position, orientation, slant and elongation for the digit).
- Map the spline into image coordinates and space circular gaussian ink generators (beads) uniformly along its length. The number of beads on the spline and their variance can easily be changed without changing the spline itself.
- Repeat many times:
  Either (with probability π_n) add a randomly positioned noise pixel
  Or pick a bead at random and generate an inked pixel from the gaussian distribution defined by the bead.

3 Fitting a model to an image

When classifying an image, we fit each model to the data and choose the model that best "explains" the image. Having selected a digit model, the fitting procedure can be viewed as maximizing the probability of generating the image from a particular model m.

$$P(I|m) = \int P(I|m, \boldsymbol{\alpha}) P(\boldsymbol{\alpha}|m) d\boldsymbol{\alpha} \tag{1}$$

where $\boldsymbol{\alpha}$ is the vector of instantiation parameters for the model. We assume the integrand in (1) has a strong peak around the best fitting model and so $P(I|m)$ can be approximated by the integrand with best fitting instantiation parameters $\boldsymbol{\alpha}^*$.[3] Taking the negative log of the approximate probability gives:

$$E_m = -\log P(\boldsymbol{\alpha}^*|m) - \log P(I|\boldsymbol{\alpha}^*, m) \; + \; Const \tag{2}$$

The first term in (2) is the log-likelihood of a particular deformation under the prior that the control points come from a single multi-dimensional gaussian. We refer to this term as the *deformation energy*, E_{def} and the second term as the *data fit* (E_{fit}). Using independent probabilities for each control point, E_{def} factorizes into the sum of deformation energies for each separate control point. Similarly, by assuming that each inked pixel in the image is generated independently from a distribution defined by the

[2] The one-model uses a similarity transform.

[3] Multiplied by some small volume $\delta\boldsymbol{\alpha}$ which is incorporated into the constant in (2).

gaussian beads and a uniform noise field, the data fit term factorizes into a sum of log probabilities of the inked pixels. The probability (P_i) of inking pixel i is:

$$P_i = \frac{\pi_n}{N} + \frac{1-\pi_n}{B} \sum_{b=1}^{B} P_{ib} \tag{3}$$

where N is the total number of pixels, B is the number of beads, π_n is the mixing proportion of a uniform noise field, and P_{ib} is the probability of inking pixel i under gaussian bead b.

The search for $\boldsymbol{\alpha}^*$ is performed using an iterative, continuation type method [1]. We start with zero deformations and an initial guess for the affine parameters obtained by centering the model over the image. A small number of gaussian beads of large variance are placed along the spline. All beads have the same variance. The large variance beads form a broad, smooth ridge along the spline allowing the model to quickly position itself close to the data. The number of beads are gradually increased while their variance decreases according to predetermined "annealing" schedule[4]. This fitting technique resembles the elastic net algorithm of Durbin and Willshaw [6] except that our elastic energy function is much more complex and we are also fitting an affine transformation.

Each iteration of the elastic matching algorithm involves three steps:

- Given the current locations of the gaussians, compute the responsibility that each gaussian has for each inked pixel. This is just the probability of generating the pixel from that gaussian, normalized by the total probability of generating the pixel ($P_{ib} \ / \ \sum_{b=1}^{B} P_{ib}$).
- Assuming that the responsibilities remain fixed, as in the EM algorithm [4], we invert a 16×16 matrix to find the image locations for the 8 control points at which the forces pulling the control points towards their home locations are balanced by the forces exerted on the control points by the inked pixels. The latter forces arise via the pulls of the inked pixels on the beads.
- Given the new image locations of the control points, recompute the affine transformation from the object-based frame to the image frame. We choose the affine transformation that minimizes E_{def}.

Some stages in the fitting of a model to data are shown in Figure 2. The search technique almost always avoids local minima when fitting models to isolated digits. If the image is not clearly recognized, i.e. the image would be a candidate for rejection (see section 4), we try four other initial positions, translated right, above, left and below the original one and choose the fit with the lowest E_m.

4 Recognizing isolated digits

After fitting all the models to a particular image, we wish to evaluate which of the models best "explains" the data. An obvious measure is the sum of E_{fit} and E_{def} that

[4]Our current schedule starts with 8 beads increasing to 60 beads in 4 steps. The variance decreases to about 1.5% of its initial value.

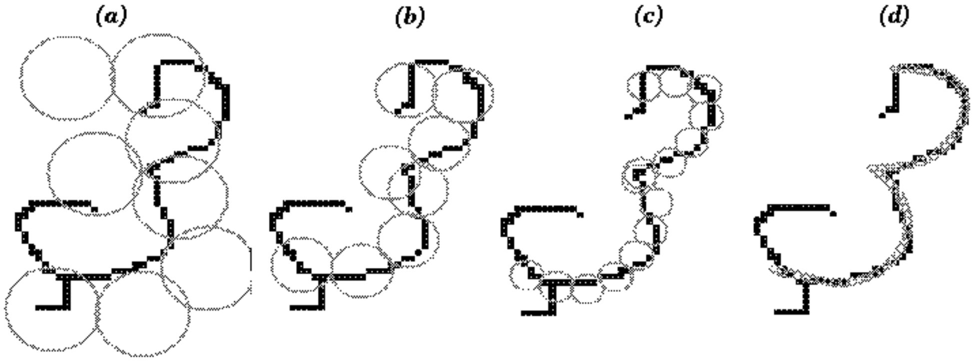

Figure 2: The sequence (a) to (d) shows some stages of fitting a model 3 to some data. The circles represent the gaussian beads, with the radius representing the standard deviation. **(a)** shows the initial configuration, with eight beads equally spaced along the spline. In **(b)** and **(c)** the variance is progressively decreased and the number of beads is increased. The final fit using 60 beads is shown in **(d)**. In this example, we used $\pi_n = 0.3$ which makes it cheaper to explain the extraneous noise pixels and the flourishes on the ends of the 3 as noise rather than deforming the model to bring gaussian beads close to these pixels.

is minimized during the fitting process. However, performance is improved by including four additional measures which are easily obtained from the fitted models.

The description of the generative model in section 3 is deficient as a complete explanation of the image as it does not explicitly penalize fits where there are gaussian beads far from any inked pixels, the "beads in white space" situation[5]. Snakes, [9] are spline contours which are attracted to particular features in an image. Motivated by this concept, we defined another energy term E_w to take into account beads in white space:

$$E_w = -\sum_{b=1}^{B} \log \sum_{i=1}^{N} P_{ib} \tag{4}$$

A bead only makes a large contribution to this cost when all inked pixels are far from the bead. It therefore has the effect of explicitly penalizing models which have beads in white space. This energy term could be easily incorporated into the fitting procedure, but in the present system we simply use it as an evaluated measure of the final fits.

The fitting procedure does not penalize affine transformed images. After fitting a model, there is an explicit representation of the affine for that digit-model. This information should help classification of an image. For example we may want to reject an explanation that requires a model to be highly sheared or elongated. So the elongation, rotation and shear of the affine are used as the remaining three measures for classification.

If M models are fitted to an image then the $6 \times M$ measures are used as input to a simple post processing neural network. The network has 10 input vectors (one for each digit), no hidden layers and uses a "softmax" output layer [2] with 10 output units. There are no cross-connections between the input vectors and the output units. For example, there

[5]There is a small implicit penalty in that beads far from inked pixels are not available for accounting for inked pixels. We have proposed a more elaborate generative model for both inked and non-inked pixels [13].

are weights between the input vector from the four model and the 4 output unit, but not to any other output unit. Including biases, the network has 70 weights and is trained using conjugate gradient minimization to minimize a cross-entropy error function. After training we classify an image according to which of the output units has the largest activation. We reject classifications in which the maximum output activation is below some threshold T.

5 Learning the models

Starting with hand crafted digit models we adjust the home control point locations so that each model maximizes the likelihood of generating instances of that digit in a training set. The maximization is performed iteratively using EM updates. This yields a simple algorithm: the updated home location of each control point (in the object-based frame) is the average location of that control point in the final fits. Learning proceeds rapidly with models learning their final configurations after only a few passes through the training set (figure 3), probably because we start off with good models.

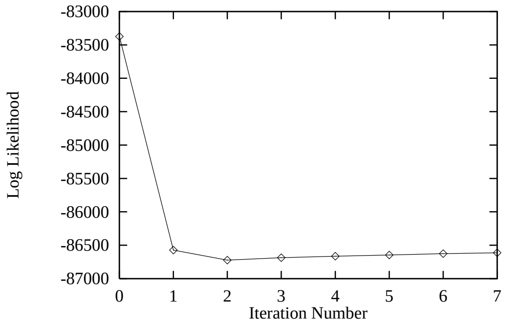

Figure 3: The likelihood of the two-model at each pass through the training set. The model has essentially completed its learning after the second pass through the training set.

An alternative to maximizing the likelihood of the image given the digit is to maximize the mutual information between the correct digit class and the probabilities assigned to the various classes by the digit models. The maximum mutual information criterion emphasizes correct discrimination rather than correct modeling of the image data, and it generally leads to better discriminative performance. Early experiments with both methods gave similar results. However, maximum likelihood learning is much quicker than discriminative learning as the latter requires fitting all models to each image.

6 Using a mixture of local models

Using spline models we can create good generators of digit images using only a small number of parameters. Complete specification of an instantiated model in the object frame needs only the (x, y) locations of n control points. It can therefore be considered as a point in $\mathcal{R}^{2n}$. The entire population of instances of a particular digit form a manifold in this space. Up to this point we have assumed that a generative model based upon a single hyperspherical gaussian can adequately model this distribution. For any deformation,

E_{def} can interpreted as a squared distance from the home locations of the generic model. To improve discrimination, we would like the deformation energy to correspond to the negative log probability density under a distribution that better represents the digit instantiations.

Figure 4 demonstrates the situation in $\mathcal{R}^2$ and how we can get poor approximations. For example, under a single gaussian approximation to the distribution, point **A** would have higher probability than point **B**, which is clearly incorrect. One way of better characterizing the distribution is to use a mixture of K gaussians:[6]

$$P_m(\mathbf{x}) = \sum_{i=1}^{K} \frac{\beta_i}{(2\pi\sigma^2)^n} \exp\{-\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|^2}{2\sigma^2}\} \tag{5}$$

where β_i is the mixing proportion for the i^{th} local model in the mixture.

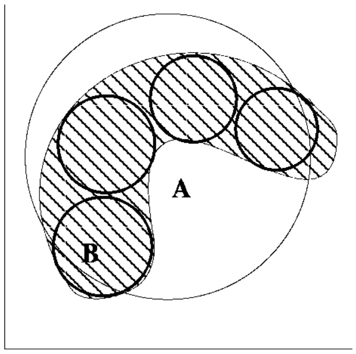

Figure 4: Illustrates how we might approximate some arbitary distribtion in $\mathcal{R}^2$, represented by the shaded region, with a single gaussian of large variance. A better approximation would be to use a mixture of gaussians each with a smaller variance. Under the single gaussian approximation point **A** would be incorrectly considered to be more likely than point **B**

The centres ($\boldsymbol{\mu}$) and common variance (σ^2) are computed using EM to maximize the log likelihood of a training set under distribution (5). Figure 5 shows the 10 local models in the mixture for the two-model. The mixture has been able to capture dominant styles. For example, variations in the presence and size of the loop have been well represented.

One way to use this mixture is to fit each of the local models to an image. This has the disadvantage of increasing the recognition time by a factor of K. Fortunately our generic model nearly always fits correctly. So an alternative strategy is to allow the generic model to fit as before, but in place of evaluating E_{def} as the negative log probability under a single gaussian, we compute it under the mixture distribution (5). This strategy is much more efficient since the most computation intensive portion, the fitting of the model to the data, is done only once. Evaluating the distance of the final fit from each of the local models in the distribution involves only computing $2n$ squared distances and so is negligible compared to the amount of computation incurred in evaluating E_{fit}.

[6] We have experimented with more complex variations such as, allowing each mixture component to have its own adaptive variance, or kernel density estimation, but found no improvement in performance over the more simple characterization.

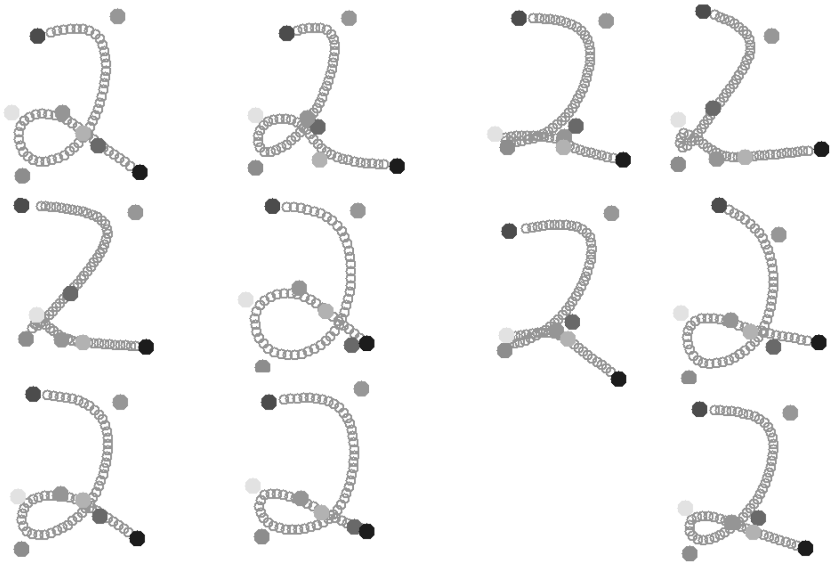

Figure 5: Local models in the mixture for the 2 model. The generic model is shown in the bottom right corner.

Figure 6 illustrates the added classification power obtained using mixture of local models. There is considerable overlap between the distributions of E_{def} for correct and incorrect classification when only a single generic model is used. The overlap has been diminished and increased separation achieved with a mixture distribution. Figure 7 shows specific examples where use of the mixture has been beneficial. In the fit on the left, the model has had to deform considerably from the generic model, yet it still lies within the expected distribution of two's. It would correspond to point **B** in figure 4. The right panel illustrates a converse situation. In fitting to an image of an 8, the six model has not deformed far from the generic model, but the deformation is most unusual. This is the situation of point **A** in figure 4.

There is another way in which the mixture of local models may be used. We would like to retain the advantage of only having to fit a single model, but instead of keeping the original generic model throughout the fit, we would like to modify the home locations of the model *during* the fitting process to a digit instance. At various stages of the fit we can recompute the mixing proportions of the local models and then create a new generic model by choosing the single gaussian that best approximates the current mixture of local models. One problem with this idea is that in some cases a wrong local model dominates early in the fitting process and prevents the correct fit from being found.

7 Experimental results

The performance of the elastic net in recognizing isolated digits has been tested on data from the CEDAR CDROM 1 database of Cities, States, ZIP Codes, Digits, and Alphabetic Characters[7]. The *br* training set of binary segmented digits was subdivided into 3 training sets of size 2000, 7000 and 2000 respectively. A validation set of 2000

[7] Made available by the Unites States Postal Service Office of Advanced Technology.

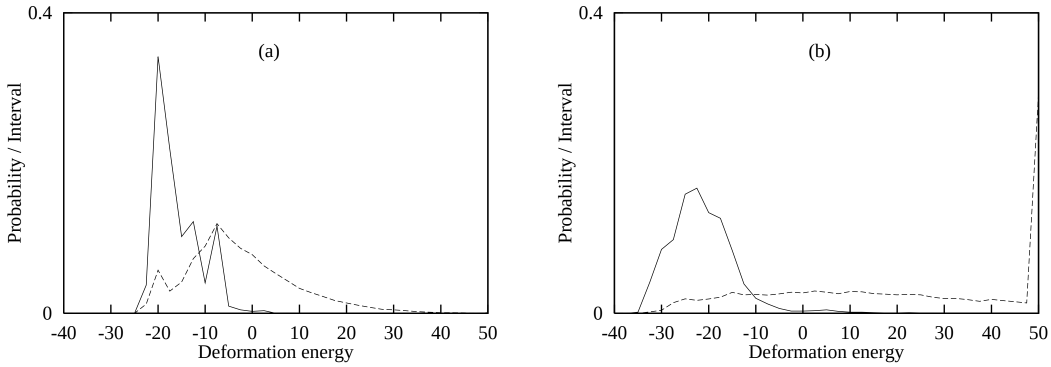

Figure 6: The solid line is the E_{def} distribution of models when fitted to examples images containing the correct digit (eg a three model fitted to an image of a 3). The dotted curve is the E_{def} distribution of the models when fitted to incorrect data. (a) Using a single generic model. (b) Using a mixture of local models. (Many instances of fitting to incorrect data in panel (b) had very large E_{def}. For the sake of display all these were assigned the value 50, accounting for the "spike" at the right edge of panel (b))

examples was also generated to tell us when to stop training the post-processing neural net. The sets were constructed so as to ensure equal representation of all digits in each set. The elastic models were trained (section 5) on the first set, the mixture of local models on the second and the post processing net on the third set. The CEDAR database also includes 2 test sets. The *goodbs* (2213 images) set is a subset of the *bs* (2711 images) set containing only well segmented digits.

	Validation Set	*goodbs* test set	*bs* test set
Generic Model	97.5	97.2	95.0
Mixture of local models	98.2	97.5	95.3

Table 1: Percentage of images correctly classified by the elastic net

Table 1 shows the performance of the elastic net when the rejection threshold (section 4) was set to zero. Even though the differences between the generic and mixture of local models are in the expected direction, they are not statistically significant on testing sets of this size.

Varying the rejection threshold in the post processing neural network allows us to trade off errors against rejects. Figure 8 shows error-rejection curves obtained on the validation and test sets. Again notice that the curve with the mixture of local models consistently lies below that of the single generic model for all data sets.

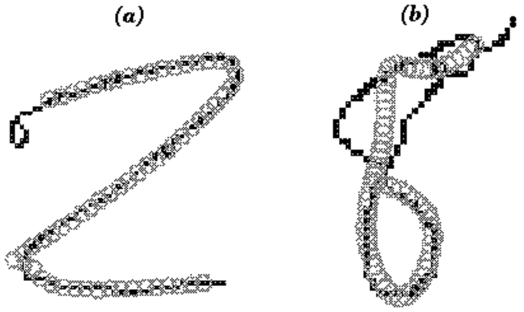

Figure 7: (a) The generic two-model (see bottom right in figure 4) has had to deform significantly in order to fit the data. Under the single gaussian approximation, $E_{def} = -7.7$. When evaluated under the mixture of local models distribution is less, $E_{def} = -16.7$. (b) The six model fitted to an image of an 8. The model has only a small deformation from the generic model ($E_{def} = -16.2$), but the deformation does not lie close to the distribution of sixes in the training set. Under the mixture distribution $E_{def} = -1.45$.

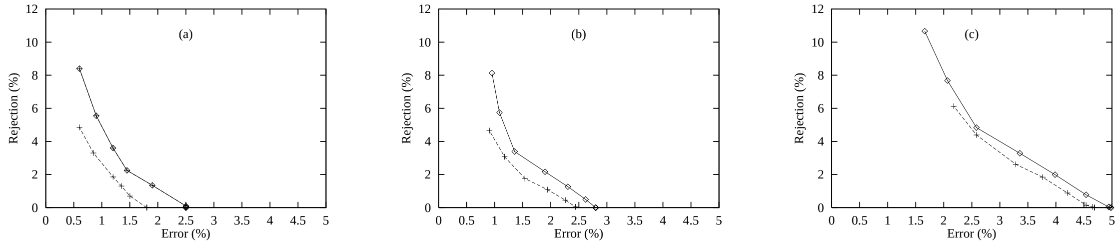

Figure 8: Error-rejection relationships. The lower dashed curve uses the mixture of local models while the upper solid curve uses a single generic model. (a) Validation set (b) *goodbs* test set (c) *bs* test set.

8 Discussion

Our elastic spline models have a lot of prior knowledge about characters built in. This has the advantage of needing only a small number of parameters to characterize the diversity of handwritten digits. We have described a simple extension to better characterize the distribution of handwritten digits without much additional computational effort. Our preliminary experiments using this method indicate that it provides improved recognition rates.

Acknowledgements

This research was funded by Apple and by the Ontario Information Technology Research Centre. We thank Allan Jepson, Richard Durbin, Rich Zemel, Rob Tibshirani and Yann Le Cun for helpful discussions. Geoffrey Hinton is the Noranda Fellow of the Canadian Institute for Advanced Research.

References

[1] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[2] J. S. Bridle. Probabilistic interpretation of feedforward classification network out-

puts, with relationships to statistical pattern recognition. In F. Fougelman-Soulie and J. Hérault, editors, *Neuro-computing: algorithms, architectures and applications.* NATO ASI series on systems and computer science. Springer-Verlag, 1990.

[3] D. J. Burr. Elastic matching of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 3(6):708–713, 1981.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society*, B-39:1–38, 1977.

[5] R. Durbin, R. Szeliski, and A. L. Yuille. An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1:348–358, 1989.

[6] R. Durbin and D. Willshaw. An analogue approach to the travelling salesman problem. *Nature*, 326:689–691, 1987.

[7] S. Edelman, S. Ullman, and T. Flash. Reading cursive handwriting by alignment of letter prototypes. *International Journal of Computer Vision*, 5(3):303–331, 1990.

[8] G. E. Hinton, C. K. I. Williams, and M. D. Revow. Adaptive elastic models for hand-printed character recognition. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kauffmann, 1992.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, Washington, D. C., 1987. IEEE Computer Society Press.

[10] P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. To appear in S. J. Hanson, J. D. Cowan and C. L. Giles, editors *Advances for Neural Information Processing systems 5*, 1993.

[11] J. R. Ward. One view of on-going problems in handwriting character recognition. In Suen C. Y., editor, *Frontiers in handwriting recognition*, pages 101–106, Concordia university, Montreal, Quebec Canada H3G 1M8, April 1990. CENPARMI.

[12] B. Widrow. The 'rubber-mask' technique–I. Pattern Measurement and Analysis. *Pattern Recognition*, 5:175–197, 1973.

[13] C. K. I. Williams, M. D. Revow, and G. E. Hinton. Hand-printed digit recogntion using deformable models. To appear in *Spatial Vision in Humans and Robots*, L. Harris and M. Jenkin eds, Cambridge University Press, 1992.