Generative Models and Bayesian Model Comparison for Shape Recognition

Balaji Krishnapuram*, Christopher M. Bishop, Martin Szummer
Microsoft Research, 7 J J Thomson Avenue, Cambridge CB3 0FB, UK
balaji@ee.duke.edu, {cmbishop,szummer}@microsoft.com

Abstract

Recognition of hand-drawn shapes is an important and widely studied problem. By adopting a generative probabilistic framework we are able to formulate a robust and flexible approach to shape recognition which allows for a wide range of shapes and which can recognize new shapes from a single exemplar. It also provides meaningful probabilistic measures of model score which can be used as part of a larger probabilistic framework for interpreting a page of ink. We also show how Bayesian model comparison allows the trade-off between data fit and model complexity to be optimized automatically.

1. Introduction

In this paper we study the problem of recognizing hand-drawn sketches captured using an online digitizer, such as a Tablet PC computer, which records temporal information. The problem comprises three related tasks: robust fitting of shapes to ink samples, recognition of a given piece of ink as a template (e.g. a square), and segmentation of a whole page of ink into disjoint subsets each of which is recognized as a template. These tasks are illustrated in Figure 1.

While robust fitting of ink samples with elementary shapes such as ellipses and lines has been studied widely in the past, the algorithms described here are capable of extension to templates of essentially arbitrary shape. Further, the proposed methods can also handle dashed lines or repeated inking of strokes. Finally, our probabilistic framework provides principled methods for comparing multiple shape hypotheses, accept/reject decisions, and classifier combination. We also outline how segmentation can be considered as an extension to the basic fitting and recognition algorithms that form the main focus of this paper.

Temporal information can be helpful in interpreting online drawings. However, we also wish to be invariant to some aspects of the temporal information. For instance, the identity and parameters of a triangle are independent of the number of strokes used to create it or their order or direction. The algorithms presented here use temporal information only for segmentation, but not for fitting or recognition.

In our applications (unlike signature verification for instance) we also want the recognizer to be independent of the pen velocity. As the first step we therefore resample the ink to give pen tip locations distributed uniformly along the strokes, with fixed arc length spacing.

We shall denote the resulting samples of user drawn ink by $\mathbf{x}_n$ where $\mathbf{x}$ represents a Cartesian location in the two-dimensional 'screen space', and $n = 1, \ldots, N$ indexes the individual sample points. The whole user figure of N points is denoted $X \equiv [\mathbf{x}_n]$. Similarly, we denote the uniformly sampled points from each template by $\mathbf{Y}^f = [\mathbf{y}_m^f]_{m=1}^M$ where $\mathbf{y}$ represents a Cartesian location in the two-dimensional 'template space', $m = 1, \ldots, M$ indexes the individual template samples, and $f = 1, \ldots, F$ indexes the figure templates. For simplicity, we have assumed that each figure template is represented by the same number of samples M.

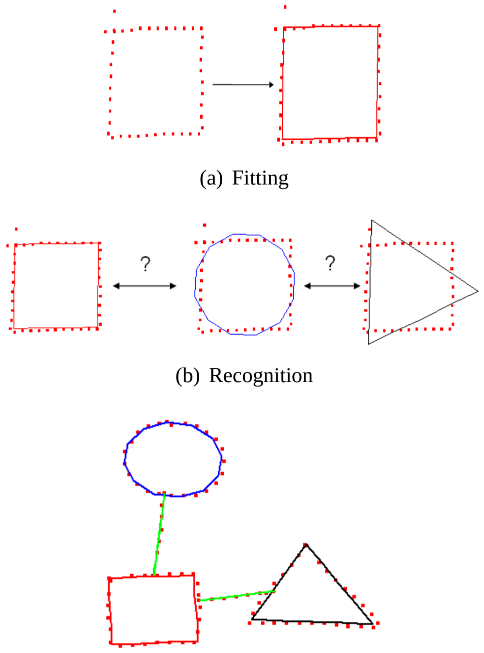
In the illustrations and benchmarks presented here we have used the square, circle, unit line segment, and equilateral triangle as canonical templates (i.e. $F = 4$). However our algorithms allow the use of any other set of shapes. For both fitting and recognizing the templates we restrict our attention to affine transformations from the canonical 'figure space' to 'screen space'. This is a mildly restrictive assumption since certain figures such as arrows cannot be directly fit or recognized by means of only affine transformations. With these kinds of figures, we shall need to take recourse to a higher level inference procedure. For instance such a procedure would recognize arrows from combinations of line segments.

2. Affine Transformations

Consider a general linear mapping corresponding to an *affine* transformation of the form

$$\boldsymbol{\mu} = \boldsymbol{C}\mathbf{y} + \boldsymbol{b}, \tag{1}$$

* Now at Hudson Hall, Duke University, Durham, NC 27708, USA



(a) Fitting

(b) Recognition

(c) Segmentation

Figure 1. Fig. (a) shows the ink, resampled uniformly along arc length, and the result of fitting the parallelogram model using EM. Fig. (b) compares the result of fitting three different shape models to the same ink. The three fitting scores can be used to determine the identity of shape. Fig. (c) shows a diagram consisting of multiple strokes after combined segmentation and recognition is performed.

where $\boldsymbol{C}$ is a 2×2 matrix, $\boldsymbol{b}$ is a two-element vector, $\mathbf{y}$ is a point in 'canonical figure space', and $\boldsymbol{\mu}$ is a point in screen space. We can conveniently combine $\boldsymbol{C}$ and $\boldsymbol{b}$ into a single $2{\times}3$ matrix $\boldsymbol{A} = [\boldsymbol{C}\ \boldsymbol{b}]$. This six degree of freedom transformation incorporates translations, rotations, shear and scaling. We augment the coordinates $\mathbf{y}$ by appending a constant one to form a vector $[\mathbf{y}\ 1]$ of dimension 1×3. In the remainder of the paper we refer to the augmented vector simply as $\mathbf{y}$, so that $\boldsymbol{\mu}_i = \boldsymbol{A}\mathbf{y}_i$.

Note that this parameterization may sometimes involve redundant degrees of freedom (for instance a straight line is governed by just four parameters corresponding to the Cartesian coordinates of its end points) so either care must be taken while fitting and recognizing such figures, or alternatively transformations involving fewer parameters have to be employed. In order to robustly fit and recognize ink, we next describe probabilistic models that measure the likelihood that a sample of ink was generated from a template under a specific affine transformation.

3. Generative Probabilistic Models

Given the electronic ink corresponding to a set of captured strokes, there are many methods available for deciding which of a set of figures they represent (i.e. recognition), and for estimating the corresponding parameters (i.e. fitting). These may be based on template matching, Hough transforms, geometrical heuristics and so on. In this section we focus on probabilistic models which represent a generative process for ink formation. These define a model $p\left(\mathbf{x} \mid \boldsymbol{A}, \mathbf{Y}^f\right)$ for the probability density of ink at any screen point $\mathbf{x}$ given the figure template $\mathbf{Y}^f$ and its affine parameters $\boldsymbol{A}$. Generative probabilistic models offer important advantages in sketch understanding, as discussed in Section 6.

Our generative model produces sample ink as follows:

1. Choose a figure template f from a predefined set of such figures $\mathcal{F} = \{1, \ldots, F\}$ according to some prior distribution $p(f)$. Here we assume that all such figures (square, circle etc.) are equally probable.
2. Choose a value for the affine parameters $\boldsymbol{A}$ from a prior probability distribution $p\left(\boldsymbol{A} \mid f\right)$ for the chosen figure, and use these parameters to map a canonical version of the figure into the screen space.
3. Decide whether a sample is generated from the figure $(\rho_n = 1)$ or not $(\rho_n = 0)$. This decision can be based on a prior probability of δ for generating stray ink and $(1 - \delta)$ for generating ink from the figure.
4. If the sample is not from the figure then choose it according to a uniform distribution over the screen space. In other words $p(\mathbf{x}_n \mid \rho_n = 0) = 1/S$, where S represents the area of the screen space.
5. If the sample is from the figure:
   (a) Select a point $\boldsymbol{\mu}_n$ in screen space lying on the figure. This is chosen from a uniform distribution $p(\boldsymbol{\mu})$ along the arc length of the transformed stroke.
   (b) Generate a sample ink point $\mathbf{x}_n$ from a Gaussian distribution centered on $\boldsymbol{\mu}_n$ having diagonal covariance with precision (inverse covariance) matrix $\tau\boldsymbol{I}$ where $\boldsymbol{I}$ is the unit 2×2 matrix. Hence, $p(\mathbf{x}_n \mid \boldsymbol{\mu}_n, \rho_n = 1) = \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_n, \tau)$.
6. Repeat steps 3 to 5b a total of N times to generate the data set $X = [\mathbf{x}_n]$.

The introduction of the background probability of generating arbitrary ink (via the binary latent variable ρ_n) provides robustness to small levels of stray ink. The mixing proportion between the figure and the background, δ, can be estimated in the fitting algorithm by maximizing likelihood. Similarly, for the fitting and recognition stages we shall omit the priors on $\mathcal{F}$ and $p(\boldsymbol{A} \mid f)$ and simply treat the figure identity and its affine parameters as unknown.

The probabilistic model contains another latent variable $\boldsymbol{\mu}_n$ for each observed ink sample $\mathbf{x}_n$, corresponding to the position along the figure arc responsible for generating that sample. The overall density model for $p(\mathbf{x}_n \mid \boldsymbol{A}, f, \tau, \rho_n = 1)$ requires an integration over $\boldsymbol{\mu}$:

$$p(\mathbf{x}_n \mid \boldsymbol{A}, f, \tau, \rho_n = 1) = \int \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}, \tau) p(\boldsymbol{\mu} \mid \boldsymbol{A}, f)\, \mathrm{d}\boldsymbol{\mu}. \tag{2}$$

Since this integration will in general be intractable (see Section 6 for some tractable special cases) we borrow a technique developed originally by [5] for digit recognition, and used as the basis of the *Generative Topographic Mapping* (GTM) [1], in which we replace the integral along arc length by a summation over a discrete uniform array of points. Thus our model becomes

$$p(\mathbf{x}_n \mid \boldsymbol{A}, f, \tau, \rho_n = 1) = \frac{1}{M}\sum_{i=1}^{M} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_i(\boldsymbol{A}, f), \tau\right) \tag{3}$$

which corresponds to a constrained Gaussian mixture model, with equal mixing coefficients $1/M$. The component centers $\boldsymbol{\mu}_i$ are related through a common set of affine parameters $\boldsymbol{A}$. In practice, we obtain $\boldsymbol{\mu}_i(\boldsymbol{A}, f)$ by a uniform arc length resampling of $\boldsymbol{A}\mathbf{Y}^f$.

The generative model including the background ink probability thus becomes $p(\mathbf{x}_n \mid \boldsymbol{A}, f, \tau, \delta) =$

$$\delta\left(\frac{1}{S}\right) + (1-\delta)\frac{1}{M}\sum_{i=1}^{M} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_i(\boldsymbol{A}, f), \tau\right). \tag{4}$$

In the next section we discuss how to efficiently use this model for robust fitting and recognition of a single figure.

4. Fitting and Recognition

The likelihood function is the probability of the observed data set given the parameters, viewed as a function of the parameters. Since the data samples are assumed to be independent and identically distributed, given the parameters and the noise precision, the log likelihood function is given by $\ln p(X \mid \boldsymbol{A}, f, \tau, \delta) =$

$$\sum_{n=1}^{N} \ln\left\{\frac{\delta}{S} + \frac{(1-\delta)}{M}\sum_{i=1}^{M} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_i(\boldsymbol{A}, f), \tau\right)\right\}. \tag{5}$$

We fit a figure f to user ink by finding the affine parameters that maximize the likelihood function:

$$\boldsymbol{A}_f^*, \delta_f^*, \tau_f^* = \underset{\boldsymbol{A}, \delta, \tau}{\operatorname{argmax}} \ln p(X \mid \boldsymbol{A}, f, \tau, \delta). \tag{6}$$

As a byproduct of the same optimization, the maximum likelihood values for each figure $f \in \{1, \ldots, F\}$ can be compared to recognize the most likely figure. Further, the posterior probability distribution over the figures can also be obtained via Bayes theorem and any uncertainty can be propagated to subsequent layers of higher level inference.

Note that the summations inside the logarithm prevent further simplification. We could use nonlinear optimization strategies, such as conjugate gradients, to maximize this likelihood function, which would require analytical evaluation of the derivatives of the log likelihood. A more elegant approach, however, is to use the EM algorithm [4] to fit the parameters of each figure.

4.1. EM Optimization

The EM algorithm optimizes the likelihood function by alternating between E- and M-steps. In the E-step we use the current settings for the parameters $\boldsymbol{A}$ to evaluate the posterior probability of correspondence between $\mathbf{x}_n$ and each latent point $\boldsymbol{\mu}_i$, given by

$$\xi_{ni} \equiv p(i, \rho_n = 1 \mid \mathbf{x}_n) = \frac{1}{Z}(1-\delta)\tfrac{1}{M}\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \tau) \tag{7}$$

$$\xi_{n(M+1)} \equiv p(\rho_n = 0 \mid \mathbf{x}_n) = \frac{1}{Z}\delta/S \tag{8}$$

$$\text{where } Z = \delta/S + (1-\delta)\frac{1}{M}\sum_{j=1}^{M}\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \tau). \tag{9}$$

In our implementation, we have assumed $\boldsymbol{\mu}_i = \boldsymbol{A}\mathbf{y}_i$. We can view ξ_{ni} as the *responsibility* which latent point $\boldsymbol{\mu}_i$ takes for explaining data sample $\mathbf{x}_n$. Additionally, $\xi_{n(M+1)}$ denotes the responsibility of the uniform background probability for explaining stray ink so that $\sum_i \xi_{ni} = 1$ for every n. From this, δ may also be re-estimated:

$$\delta = \frac{\sum_{n=1}^{N} \xi_{n(M+1)}}{\sum_{n=1}^{N}\sum_{m=1}^{M+1} \xi_{nm}} = \frac{\sum_{n=1}^{N} \xi_{n(M+1)}}{N}. \tag{10}$$

In the M-step the responsibilities $\{\xi_{ni}\}$ are held fixed, and the parameters $\boldsymbol{A}$ are optimized by maximizing the expected complete-data log likelihood given by $F(\boldsymbol{A}, \tau) =$

$$\sum_{n=1}^{N}\left[\sum_{i=1}^{M} \xi_{ni} \ln \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_i = \boldsymbol{A}\mathbf{y}_i, \tau) + \xi_{n(M+1)} \ln \tfrac{1}{S}\right]. \tag{11}$$

Note that the logarithm and the summation have been interchanged compared with (5). By substituting the standard expression for the Gaussian and removing additive and multiplicative constants, we see that maximization of F with respect to $\boldsymbol{A}$ is equivalent to maximizing

$$\widetilde{F}(\boldsymbol{A}, \tau) = -\frac{\tau}{2}\sum_{n=1}^{N}\sum_{i=1}^{M} \xi_{ni}\|\mathbf{x}_n - \boldsymbol{A}\mathbf{y}_i\|^2 + \ln\tau \sum_{n=1}^{N}\sum_{i=1}^{M} \xi_{ni}. \tag{12}$$

Maximizing with respect to $\boldsymbol{A}$ we obtain the equation

$$\boldsymbol{A}\left(\sum_{i=1}^{M}\mathbf{y}_i{\mathbf{y}_i}^T\left[\sum_{n=1}^{N}\xi_{ni}\right]\right)=\sum_{n=1}^{N}\sum_{i=1}^{M}\xi_{ni}\mathbf{x}_n\mathbf{y}_i^T, \qquad (13)$$

which is easily solved to determine $\boldsymbol{A}$. Similarly, maximizing (12) w.r.t. τ, we obtain:

$$\tau^{\star}=\left(\frac{\sum_{n=1}^{N}\sum_{i=1}^{M}\xi_{ni}\|\mathbf{x}_n-\boldsymbol{A}\mathbf{y}_i\|^2}{2\sum_{n=1}^{N}\sum_{i=1}^{M}\xi_{ni}}\right)^{-1}. \qquad (14)$$

4.2. Symmetric Explanation of Model and Ink

The fitting objective of (6) has a smaller penalty for explaining user drawn ink with wasted model ink as compared to the penalty for not explaining some section of the user ink [3]. To further improve accuracy, we investigated an alternative method for fitting and recognizing figures that explicitly enforces symmetry in the explanations. This procedure measures how accurately the model explains user ink, and conversely, also how accurately the user ink explains the model. This may be summarized as $\boldsymbol{A}_f^*, \delta_f^*, \tau_f^* =$

$$\underset{\boldsymbol{A},\delta,\tau}{\operatorname{argmax}}\left[\ln p(X \mid \boldsymbol{A}\mathbf{Y}^f,\tau,\delta)+\ln p(\boldsymbol{A}\mathbf{Y}^f \mid X,\tau,\delta)\right], \qquad (15)$$

where we assume the same form of generative probabilistic model as above for both of the probability distributions, sharing the same τ and δ parameters. The EM algorithm for the above maximization needs two sets of binary indicator variables, $\rho_n^{(1)}$ and $\rho_n^{(2)}$, as well as two sets of responsibilities. The first set of $\xi_{ni}^{(1)}$ measures the responsibility which each model sample takes for explaining a given user ink sample, calculated as in eqs. (7, 8) substituting ξ_{ni} and ρ_n by $\xi_{ni}^{(1)}$ and $\rho_n^{(1)}$ respectively. The second set $\xi_{im}^{(2)}$ measures the responsibilities which each given user ink sample assumes in explaining a model sample:

$$\xi_{im}^{(2)} \equiv p(i,\rho_m^{(2)}=1 \mid \boldsymbol{A}\mathbf{y}_m)=\frac{1}{Z}(1-\delta)\tfrac{1}{N}\mathcal{N}(\boldsymbol{A}\mathbf{y}_m \mid \mathbf{x}_i,\tau) \qquad (16)$$

$$\xi_{(N+1)m}^{(2)} \equiv p(\rho_m^{(2)}=0 \mid \boldsymbol{A}\mathbf{y}_m)=\frac{1}{Z}\delta/S \qquad (17)$$

$$\text{where } Z=\delta/S+(1-\delta)\frac{1}{N}\sum_{j=1}^{N}\mathcal{N}(\boldsymbol{A}\mathbf{y}_m \mid \mathbf{x}_j,\tau). \qquad (18)$$

Hence, δ is easily re-estimated by maximum likelihood,

$$\delta=\frac{\sum_{n=1}^{N}\xi_{n(M+1)}+\sum_{m=1}^{M}\xi_{(N+1)m}}{N+M}. \qquad (19)$$

Denoting the mutual responsibility as $\xi_{ij}=\frac{1}{2}(\xi_{ij}^{(1)}+\xi_{ij}^{(2)})$, it turns out that the M-step update for $\boldsymbol{A}$ and τ remain identical to (13) and (14) respectively.

	P-gram	Ellipse	Triangle	Line
P-gram	**126, 125**	2, 2	1, 2	1, 0
Ellipse	0, 5	**121, 124**	0, 2	0, 0
Triangle	2, 0	2, 6	**126, 121**	4, 0
Line	3, 1	10, 3	1, 3	**125, 130**
# Figures	131	135	128	130

Table 1. Confusion matrix for recognizing templates. Rows indicate the recognized figure while columns indicate the true figure. In each column the accuracy of the generative model based method is indicated on the left and symmetric scoring scheme is indicated on the right. Correctly recognized figures appear on the diagonal.

4.3. Initialization

In our implementations we have found it useful to start the algorithms with a small τ and δ and to re-estimate them as part of the EM algorithm for fitting figure templates. However, care must be taken in the recognition stage while comparing the posterior probabilities to utilize the same τ for all the figures. This is necessary to avoid problems due to different noise scales for different models.

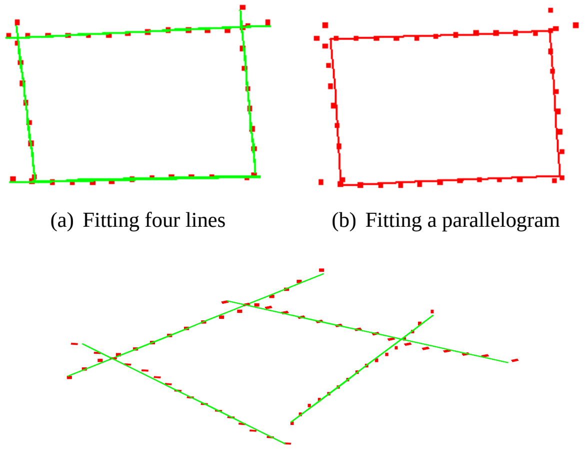
Note that $\boldsymbol{A}\mathbf{Y}^f$ may not be uniformly sampled along the arc, even though $\mathbf{Y}^f$ was. Thus we may wish to resample it in every step of the algorithm. However, in practice we have found this unnecessary when using a reasonable initialization procedure.

We use the mean of the ink samples to initialize the translation parameters in $\boldsymbol{A}$. We initialize scale and rotation using a simple eigenvector decomposition of the covariance matrix (this is equivalent to linear principal components analysis). For the purposes of initialization we ignore the contribution from shear.

4.4. Assessment of Recognition Accuracy

In order to assess the performance of these fitting and recognition methods we asked a set of thirteen subjects to draw parallelograms, ellipses, line segments and triangles using a Tablet PC pen computer. A total of 524 figures were collected. The confusion matrix for both methods described above are provided in Table 1. The recognition accuracy of both methods is equally good, and the difference between their performances is not statistically significant. Further evaluation with a larger database of user drawn figures would probably be useful in future.

The recognition accuracy is affected to an extent by the sampling rate of both the figure and the user ink. In particular, finer sampling proves advantageous in disambiguating



(a) Fitting four lines

(b) Fitting a parallelogram

(c) Data better explained with four lines

Figure 2. Example of Bayesian model selection. The same ink is explained in Fig. (a) using four straight lines whereas in Fig. (b) it is explained using a single parallelogram. The straight lines are less constrained and so will always give a better fit to the data, whereas Bayesian model selection correctly prefers the parallelogram. In the Fig. (c) the more probable explanation is given by the model consisting of four line segments since this gives a much better data fit.

very small figures. We assessed the failure modes of these algorithms by visually inspecting the misclassified figures. Most of these figures were very small and from the sampled versions we found it difficult to accurately identify the figure even by eye. This leads us to believe that further gains in performance are not immediately achievable, at least not until we move to a continuous contour representation instead of a sampled point based representation. It remains to be seen if a continuous representation improves the quality of these methods.

Recognition of the entire set of 523 figures took 924.8 s for the generative model based method, and 863 s for the symmetric scheme, using a Matlab implementation.

5. Segmentation and Bayesian Model Selection

A practical shape recognition system must deal not just with isolated shapes but with whole pages of ink. This can lead to multiple alternative explanations of the same ink, and it is important to have a principled framework for selecting the best interpretation. Consider the example shown in Figure 2. The model comprising a probabilistic mixture of four straight lines will always give a better fit to the data than the single parallelogram since it has fewer constraints (the lines do not have to be perpendicular to each other, nor do their ends have to touch). Thus maximum likelihood will always prefer the more complex model.

From a Bayesian perspective, given alternative figure explanations $\mathcal{H}_k$, where $k = 1, \ldots, K$ for the observed data $\mathcal{D}$, the posterior probabilities are given by

$$p(\mathcal{H}_k \mid \mathcal{D}) \propto p(\mathcal{D} \mid \mathcal{H}_k) p(\mathcal{H}_k). \qquad (20)$$

If we assume that the explanations have equal prior probability (we can trivially use unequal priors if we wish) then the posterior probabilities are determined by the *evidence* $p(\mathcal{D} \mid \mathcal{H}_k)$ of each explanation, which itself is found by marginalizing over the model parameters $\boldsymbol{\theta}_k$:

$$p(\mathcal{D} \mid \mathcal{H}_k) = \int p(\mathcal{D} \mid \boldsymbol{\theta}_k, \mathcal{H}_k) p(\boldsymbol{\theta}_k \mid \mathcal{H}_k) \, \mathrm{d}\boldsymbol{\theta}_k. \qquad (21)$$

We can obtain a rough approximation to this integral if we assume that each parameter has a prior which is uniform over some region of width Δ_0 and that the corresponding posterior is sharply peaked around the model $\boldsymbol{\theta}_k^*$ and has width Δ, so that

$$p(\mathcal{D} \mid \mathcal{H}_k) \simeq p(\mathcal{D} \mid \boldsymbol{\theta}_k^*)(\Delta/\Delta_0)^{M_k}, \qquad (22)$$

where M_k is the number of parameters in $\boldsymbol{\theta}_k$. We can refine this estimate somewhat by noting that in practice the posterior distribution will be multi-modal due to the presence of multiple equivalent parameter values. For example, if the hypothesis comprises L line segments there will be $L!$ permutations of the parameter values all of which lead to the same probability density over the ink. Similarly, in the case of a square, for each parameter value there are three other different values which give rise to the same density, corresponding to rotational symmetry. If we denote the overall redundancy factor for a given shape hypothesis by $\mathcal{F}_k$ then the log evidence is given by

$$\ln p(\mathcal{D} \mid \mathcal{H}_k) \simeq \ln p(\mathcal{D} \mid \boldsymbol{\theta}_k^*) + M_k \ln C + \ln \mathcal{F}_k, \qquad (23)$$

where $\ln p(\mathcal{D} \mid \boldsymbol{\theta}_k^*)$ is the log likelihood for the fitted model, as calculated previously, and $C = \Delta/\Delta_0$ denotes the posterior to prior width ratio (where $C \ll 1$). The redundancy factor $\mathcal{F}_k$ is easily calculated for any given model, and we choose a small value for C which can be tuned by cross-validation if desired.

In order to parse an entire sketch containing multiple templates, we first group the ink sample points into larger units. In particular, we apply an efficient divide and conquer algorithm to divide a stroke into roughly straight line fragments. This method is quite robust, and box-shaped strokes typically split into just a few fragments.

Next, we employ a 'wrapper' approach around the previous algorithms. We consider partitions of the set of all ink fragments and use the previous fitting and recognition algorithms on each subset. The search space over all partitions is too large, so for efficiency we consider only individual subsets of at most 7 fragments. We consider only subsets of temporally consecutive fragments so that we can apply a dynamic programming algorithm to score the whole sketch. The similarity between the user drawn ink and the postulated explanation is scored using either the generative or symmetric models.

6. Discussion

In this paper we have proposed a probabilistic approach to shape modelling based on generative models and EM to find maximum likelihood solutions. We have also shown that Bayesian model selection techniques can be used to find the most probable explanation for a set of strokes amongst competing hypotheses.

We could also apply the EM algorithm to jointly fit multiple figures. We would modify the E-step trivially to assign responsibilities for each ink sample to multiple figures. The M-step can be performed independently for each figure as earlier, and the computational cost of this joint optimization is the same as fitting the figures separately. For efficiency, the line fragments produced by preprocessing can reasonably be assumed to belong to a single figure.

In building a generative model of ink based on a local Gaussian noise model we have to find a tractable approach to the arc length marginalization. Here we have proposed the use of a fixed uniform discrete sampling, eq. (3), following the framework of GTM. This allows very general templates to be considered. More flexible models based on optimizing the control points in a spline representation, have been used by [5] for modelling handwritten digits. These allow non-affine deformations of the template, at the expense of a more complex fitting procedure in the M-step.

For the specific case of a straight line segment the marginalization with respect to a uniform distribution of ink can be expressed in closed form as the product of a Gaussian with the difference of two ‘erf’ functions. Figures composed of straight lines can be expressed as constrained mixtures of these. If the same analysis is applied to a circle it again leads to a closed form solution in terms of Bessel functions. However, the M-step will now require nonlinear optimization, offsetting some of the gain in avoiding the GTM discretization.

An alternative framework, also limited to figures composed of straight line segments, was discussed by [7]. The center location, orientation and length of each line segment are given Gaussian distributions, the parameters of which define the structure of the figure. Each line segment then defines a pseudo-Gaussian distribution of ink given by $\exp(-d^2)$ where d is the distance from a point to the line, defined to be the perpendicular distance if the orthogonal projection exists and the distance to the nearest endpoint otherwise. Note that in this model there are no constraints such that, for instance, the three edges of a triangle should coincide at their end points.

Our proposed model uses independent Gaussian noise on each of the samples to represent the discrepancy between the captured ink and the canonical stroke. This treats the ink samples as independent and identically distributed, given the model parameters, and ignores the strong sequential correlations between successive samples. It is not clear how much practical advantage could be gained by modelling these correlations. However, there exist techniques which could be employed such as hidden Markov models and Kalman filters. Another possible framework is that of Gaussian processes, which has already been used to provide an alternative formulation of the GTM [2]. Such approaches, however, will be complicated by the need to consider the temporal ordering of strokes and of pen direction within a stroke.

Acknowledgements

We are grateful to Michel Gangnet for many valuable discussions and software contributions, and to Hannah Pepper for collecting the data set.

References

[1] C. M. Bishop, M. Svensén, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1–3):203–224, 1998.

[2] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[3] W. Cheung, K, D.-Y. Yeung, and R. T. Chin. Bidirectional deformable matching with application to handwritten character extraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1133–1139, 2002.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Jrnl. of the Royal Statistical Society B*, 39(1):1–22, 1977.

[5] G. E. Hinton, C. K. I. Williams, and M. Revow. Adaptive elastic models for character recognition. In *Advances in Neural Information Processing Systems 4*. Morgan Kauffman, 1992.

[6] E. Saund. Finding perceptually closed paths in sketches and drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):475–491, 2003.

[7] G. E. Valveny and E. Martí. Learning of structural descriptions of graphic symbols using deformable template matching. In *Proc. IEEE 6th Intl. Conf. on Document Analysis and Recognition*, pages 455–459, 2001.