Network: Computation in Neural Systems 5 (1994) 241–258. Printed in the UK

A shape-recognition model using dynamical links

Elie Bienenstock†§‖ and René Doursat‡

† Division of Applied Mathematics, Brown University, Box F, Providence RI 02912, USA
‡ Institut für Neuroinformatik–Systembiophysik, Ruhr Universität Bochum, ND03, D-44780 Bochum, Germany

Received 9 March 1993, in final form 22 November 1993

Abstract. A shape-recognition method is proposed, inspired from the dynamic-link theory of von der Malsburg (1981). The quality of a match between two images is assessed through an *elastic* cost functional; the minimal value reached by the cost over a suitably-defined space of maps is viewed as a *distance* between these two images. Experiments on nearest-neighbour classification of handwritten numerals are presented, using a computationally effective procedure for finding a reliable estimate of the matching distance.

1. Introduction

It has been proposed (von der Malsburg 1981, 1987, von der Malsburg and Bienenstock 1986) that the brain may represent dynamical bonds between entities by using suitably defined accurate temporal relationships between neural activity patterns. This idea has recently become a focus of interest, mainly for its potential to solve the so-called *binding*—or segmentation—problem for neural networks. Equally attractive, however, is the suggestion (von der Malsburg 1981) that the brain may use dynamical links—in the form of accurate temporal relationships between the firings of neurons and possibly fast synaptic plasticity—to implement relational descriptions of objects and relation-preserving maps between such descriptions; relational descriptions and relation-preserving maps are likely to be required in many cognitive functions, e.g. perception.

A literal numerical implementation of these ideas in terms of accurate neuronal spiking and fast synaptic plasticity would be impractical. Thus, shape-recognition models inspired from the dynamical-link theory (e.g. Bienenstock and von der Malsburg 1987, Lades *et al* 1993) have generally kept the spirit of the approach, that of a simple, relatively low-level, relational description using dynamical links, and adapted it in various ways to the problem at hand. In this paper, we propose a formulation using a computationally efficient version of *elastic matching* (Burr 1980, Hinton *et al* 1992). An outline of our approach as well as preliminary results have been presented elsewhere (Bienenstock and Doursat 1989, 1991), and a very similar model has been applied to other recognition problems (Buhmann *et al* 1989, Lades *et al* 1993).

In the context of automated pattern recognition, relational-matching and deformable-template methods have been proposed in the past under a variety of forms (e.g. Bajcsy and Kovacic 1989, Grenander *et al* 1990, Amit *et al* 1991, Hinton *et al* 1992, Dickinson *et al*

§ To whom correspondence should be addressed.
‖ On leave from CNRS, Paris, France.

0954-898X/94/020241+18$19.50

1992); see also Hummel and Biederman (1992) for an example of a hierarchical relational-matching model inspired from psychological data and from the dynamical-link theory. Relational-matching methods proceed roughly as follows. A collection of prototype objects is defined, each in terms of *relations* between object subparts, e.g. local features. Upon presentation of an unknown object to recognize, one attempts to build relation-preserving maps between the prototypes and this object, described in the same relational format as the prototypes. The object is recognized—or not recognized—on the basis of the best relation-preserving map(s) found. This strategy sometimes turns out to be impractical, as it may require to search through a very large space of maps.

In the last few years, the task of handwritten-character recognition has become a benchmark for algorithms of pattern recognition, whether neurally inspired (e.g. Le Cun *et al* 1989, Martin and Pittman 1991) or not (e.g. Bozinovic and Srihari 1989, Kundu *et al* 1989, Simard *et al* 1993). Due to the computational difficulty just mentioned, relational-matching methods are not widely used; see, however, Burr (1980), Hinton *et al* (1992), and, in a similar spirit, Simard *et al* (1993). More popular are methods which rely on a blend of feature-extraction techniques followed by conventional statistical classifiers or feedforward neural networks.

The elastic-matching approach presented in this paper can be applied to handwritten-character recognition by defining each character in terms of *geometric* relations between its elementary constituents, i.e. image pixels. Matching is then invariant with respect to shifts, and relatively tolerant of mild rotations and rubber-sheet deformations. As assessed on a medium-size database, this yields good classification performances, taking into account the simplicity of the model relative to alternative methods recently proposed for this task (e.g. Hinton *et al* 1992, Simard *et al* 1993).

The plan of the paper is as follows. Section 2 defines a *matching distance* between two images X and X' as the minimum value of a suitably defined cost functional over a family of maps from X to X'. The cost of a map is an integrated measure of the amount of deformation effected by this map; matching is referred to as *elastic* because of the quadratic form of local contributions to the cost. As a strategy for finding a map realizing the absolute minimum of the cost is not available, we use a *suboptimal* search algorithm, which provides a good approximation to this minimum; the algorithm is outlined in section 3 and described in more detail in the appendix. Statistical experiments are presented in section 4: nearest-neighbour classification is used with our elastic-matching distance instead of, for example, Euclidean distance. Section 5 compares our biologically inspired approach to the methods for handwritten-digit recognition proposed by Hinton *et al* (1992) and Simard *et al* (1993); section 6 is a brief discussion of the model in statistical and biological contexts.

2. The elastic-matching distance

We consider a binary-valued image X on the square lattice S; the value of X at site (pixel) $s \in S$ is denoted $X(s)$. We are interested in images of handwritten numerals, where, by convention, pixel value 1 stands for 'black' or 'numeral', and pixel value 0 stands for 'white', or 'background'. There are ten numeral classes numbered 0 to 9, and numerals within a given class may come in a variety of shapes. As a result, the spread of a given class as assessed by pixelwise distance, i.e. Euclidean distance in X-space, may be considerable.

Our goal is to endow the space of images on S with an alternative metric $\mu(X, X')$ that will reduce this intra-class spread as much as possible. In section 6, we shall characterize this strategy as the *a priori* introduction of a suitable *bias* in the problem. Ideally, one would like any two images belonging to the same class to be closer to each other, in this

new metric, than any two images belonging to a different class. With such a metric, one prototype per class would be enough to achieve error-free classification, using for instance the first-nearest-neighbour method. Unfortunately, one would be hard-pressed to invent a metric $\mu(X, X')$ satisfying this requirement. We shall therefore settle for a more modest goal: the metric μ should be such that, *in general*, $\mu(X, X')$ is small whenever X and X' belong to the same class, and $\mu(X, X')$ is large for X and X' belonging to different classes.

Now given two images X and X' belonging to the same class, i.e. distinct handwritten realizations of the same numeral, one may often view X' as a *deformation* of X (equivalently X as a deformation of X'), where the deformation is a composition of rigid transformations (a shift and a small rotation) with moderate non-rigid distortions. The metric $\mu(X, X')$ we shall define is, roughly, the amount of deformation required to transform X into X'. This definition should be such that: (i) it captures most variations observed in handwritten numerals and only those, and (ii) the computation of $\mu(X, X')$ can be effectively carried out.

For a given image X and a given integer m, let S_X be the union of the black pixels in X (the numeral itself) with a *padding* of white pixels of width m around these black pixels; if $m = 0$, S_X contains no white pixels. For any two images X and X' on S, a map f from S_X to S is called *permissible* if it preserves pixel values, that is, if $X'(f(s)) = X(s)$ for all s in S_X; the family of all permissible maps f from X to X' is denoted $\mathcal{P}_{XX'}$. We wish to measure, for any permissible map f, the amount of deformation effected by f. To this end, we define a *cost*, or *energy*, functional $H(f) = H_1(f) + \kappa H_2(f)$ on $\mathcal{P}_{XX'}$. The first part of the functional, $H_1(f)$, measures the deformation effected by f; κ is a non-negative parameter and the second part, $H_2(f)$, measures the departure of f from injectivity.

Specifically,

$$H_1(f) = \sum_{s,t \in S_X, \|s-t\|=1} \| (s - t) - (f(s) - f(t)) \|^2 .$$

Here, the symbol $-$ is used to denote subtraction between sites considered as points in R^2. Thus, H_1 is the sum, over all pairs of *neighbours* s and t in S_X, of the squared norm of the difference between the vector from point t to point s and the vector from point $f(t)$ to point $f(s)$. Provided S_X is connected, $H_1(f)$ is 0 if and only if $f(s) - f(t) = s - t$ for *any* two sites s and t in S_X, i.e. if and only if f is, *globally*, a shift. Note that H_1 is *locally* composed in the topology of S_X: two sites s and t in the domain of f interact—they contribute to H_1—only if they are nearest neighbours. The penalty contributed by a pair of neighbours is quadratic in the amount of distortion effected there. The main reason for choosing this quadratic form is computational convenience (see next section), but it can also be interpreted as a form of *elastic energy* (think of f as a deformation acting on a rubber sheet).

In short, the first part of the functional H—which we seek to minimize over all permissible maps—embodies a collection of independent soft constraints on f, which collectively tend to make f a shift. In particular, H_1 penalizes rotations; the penalty is small for small-amplitude rotations, and increases rapidly for larger ones.

The second term in $H(f)$, also a collection of quadratic soft constraints, is defined as follows:

$$H_2(f) = \sum_{s' \in S} \left((\mid f^{-1}(s') \mid -1)_+ \right)^2$$

where $\mid A \mid$ is the size of set A, and u_+, the positive part of u, is u if $u > 0$, 0 otherwise. This term is 0 if and only if for each s' in S the set $f^{-1}(s')$ has *at most* one element in it, that is, f is injective. This second term does not play a crucial role in the definition of the

distance; in effect, if the first term vanishes—f is a shift—so does the second. However, including H_2 was found to improve classification performance (see section 4).

Note that the pixel-value constraint, $f \in \mathcal{P}_{XX'}$, could have been implemented as a soft constraint, in the style of H_1 and H_2. However, numerical experiments (not reported in the present paper) showed no clear advantage in doing so, and a hard constraint was found preferable for computational reasons.

Given two images X and X', we may now tentatively define an elastic distance between them as the minimum value reached by H over all permissible maps:

$$\lambda(X, X') = \min_{f \in \mathcal{P}_{XX'}} H(f).$$

This λ, however, is not quite a metric. In particular, it generally is not the case that $\lambda(X, X') = \lambda(X', X)$. Also, λ has the following 'subset problem'. Assume that X and X' are two numerals belonging to *different* classes such that X is (approximately) a subset of X', i.e. such that there exists a map f in $\mathcal{P}_{XX'}$ that is (approximately) a shift. This may for instance occur with the numerals '3' and '8' (see figure 5). Under these conditions, $\lambda(X, X')$ is small, possibly smaller than $\lambda(X, X'')$ for some X'' in the *same* class as X. This is clearly undesirable.

These two problems may be solved by *symmetrizing* λ as follows:

$$\mu(X, X') = \max\{\lambda(X, X'), \lambda(X', X)\}.$$

We shall illustrate in the next section the working of μ on the subset problem†.

3. Computing the elastic-matching distance

Computing the elastic-matching distance $\mu(X, X')$ between two numerals X and X' entails the minimizing of H over two spaces of permissible maps, $\mathcal{P}_{XX'}$ and $\mathcal{P}_{X'X}$. These spaces are clearly too large to allow exhaustive search. We shall therefore content ourselves with an approximation, a suboptimal f. The present section outlines a computationally effective method for finding such a suboptimal solution; a more detailed description of the algorithm is given in the appendix. In the next section, we shall show that the approximated elastic-matching distance yields good classification performances, and we shall argue that these performances are probably nearly as good as would be obtained with the true elastic-matching distance if this were available.

As remarked above, the first term in the cost functional H is made up of a sum of *local* contributions, as each site s interacts only with its nearest neighbours in S_X. This suggests a straightforward iterative-improvement, 'greedy', procedure for minimizing f over the space of permissible maps $\mathcal{P}_{XX'}$. Step k in this procedure consists in visiting a site $s = s_k$ in S_X and updating f at s while keeping it constant at all $t \neq s$. Consider, for a moment, only H_1 and ignore H_2. The only sites $t \neq s$ that matter are then the four neighbours of s: t_i, $i = 1, \ldots, 4$ (here we assume that s is an interior point of S_X). Due to the quadratic form of H_1, the optimal value of f at s given f at the four neighbours of s is the centre of mass of these four values: $\tilde{s} = \frac{1}{4}\sum_{i=1}^{4} f(t_i)$ (see appendix, equation (A1)). However, $\tilde{s}$ is not necessarily a lattice point, nor does it necessarily satisfy the pixel constraint $X'(\tilde{s}) = X(s)$ if it happens to be a lattice point. We also need to take into account the second term H_2 in the cost to find the truly optimal $f(s)$.

† The function μ is still not quite a metric, as it does not necessarily satisfy the triangle inequality. This is of little practical incidence; it can actually be remedied by adding a positive constant C to every $\mu(X, X')$ such that $\mu(X, X') > 0$.

We therefore proceed as follows. After having computed $\tilde{s}$, we visit all sites s' in S that satisfy $X'(s') = X(s)$, in order of increasing distance from $\tilde{s}$. For each site s' visited, we compute $g(s')$, the total change in H resulting from moving $f(s)$ from $\tilde{s}$ to s'. The optimal s' is the site that yields the smallest g; we know when to stop the search because the H_1-component in $g(s')$ is quadratic in $\| s' - \tilde{s} \|$.

This procedure allows us to find, in a computationally effective way, the H-optimal value of f at site s in S_X given f at all sites $t \neq s$ in S_X. Applying this local update scheme iteratively will in general yield convergence to a *local* minimum of H in the space of permissible maps, local in the sense of the topology defined by this greedy single-update scheme: there will be no guarantee that the solution reached is the true optimum. Moreover, as with all such greedy algorithms, one should expect high sensitivity to initial conditions. The local minimum reached will also depend on the visitation sequence for sites s in S_X. However, numerical experiments (see section 4) show that *classification* based on this approximated elastic distance is quite robust. As expected, the single most important factor is the initialization. For instance, if the two images X and X' are '8's, say $X = X'$, it is easy to initialize the algorithm in the 'wrong' way, so that the top circle of the '8' in X will map to the bottom circle of the '8' in X' and *vice versa;* such a map corresponds to a *local minimum* of the energy, with a fairly high cost coming from the mismatch at the centre of X.

In the experiments reported in section 4, we used the following simple initialization procedure, which reliably eliminates the danger of ending in a local energy minimum of the type just described. The map f is first defined on a small number q of randomly chosen black sites $s_1, s_2, \ldots, s_q \in S_X$; typically, q is about one tenth of the number of black sites in S_X. This is done using the following simple *alignment* procedure. Let $c(X)$, resp. $c(X')$, be the centre of mass of the set of black pixels in X, resp. X'; $c(X)$ and $c(X')$ are generally *not* lattice points. We then define $f(s_i)$, $i = 1, \ldots, q$, to be the lattice point s' nearest to $s_i + c(X') - c(X)$ which satisfies $X'(s') = 1$. After f has been defined in this way on q initial sites in S_X, we *extend* it, site-by-site, to the rest of S_X using the greedy update scheme described above (see appendix for details).

Since this initialization procedure does use the update process (except on a small number of sites), we shall refer to it as 'iteration 0' of the optimization. Further iterations consist in re-updating f once on all sites $s \in S_X$, including the first q (in the same order as before). We shall see in section 4 that for purposes of classification iteration 0 is by far the most important.

Before we turn to classification experiments, we illustrate with a few figures the working of the optimization algorithm. Figure 1 shows the successive steps in iteration 0 for the matching of two numerals belonging to two different classes; the match f reached at the end of iteration 0 (panel C) is a severe distortion, heavily penalized by H. Figure 2 shows the result of further optimization (10 iterations) on this matching problem, as well as on the matching of two numerals that belong to the same class and are indeed quite similar. In the latter case, the value of H reached is of course much lower; it is close—possibly equal—to the global minimum for this problem. In both situations, the optimization process has converged; the transformations shown correspond to local minima of H.

Figure 3 illustrates the local minima reached for the same two matching problems as in figure 2, but this time the numerals X and X' have first been *thinned*, using a straightforward thinning algorithm; this reduces substantially the size of S_X, hence the amount of computation required. Still with thinned numerals, figure 4 illustrates the result of the matching algorithm with a larger padding of white pixels ($m = 5$ instead of 1 in the previous figures), resulting in a much larger domain set S_X; as we shall see in the next

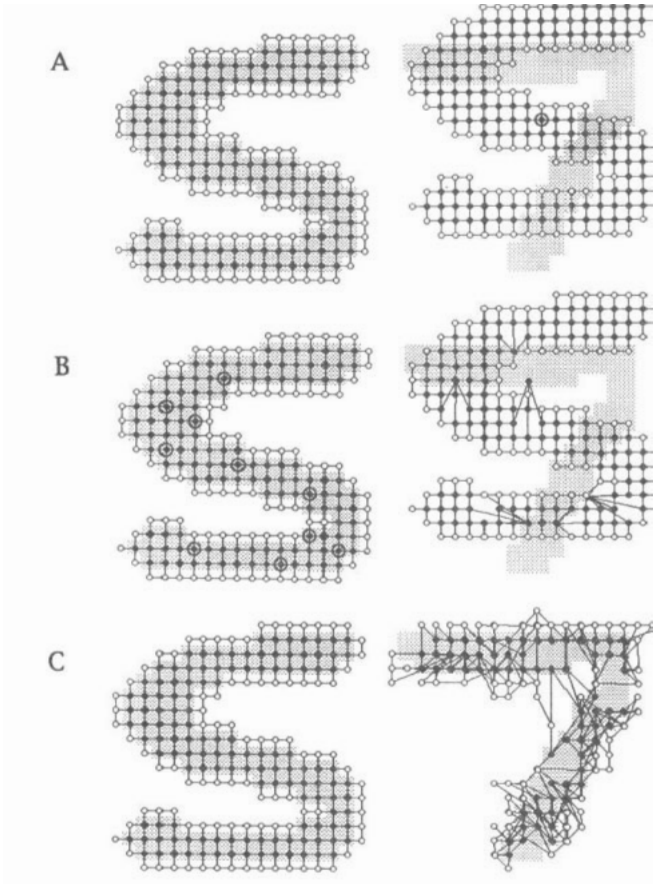

Figure 1. Three steps in iteration 0 ('initialization') of the elastic-matching process. An instance of numeral '5' is to be mapped on an instance of numeral '7'. In the first step (panel A, right) the numerals are registered so that the two centres of mass coincide (circled node). In the second step (panel B) 10 randomly chosen black nodes in numeral '5' (circled nodes, left) are mapped (right) onto the respective closest black nodes in numeral '7'; the images of all other nodes are unchanged. In the third step, each remaining node in numeral '5', black as well as white, is visited once and its image updated according to a greedy update algorithm ('elastic' relaxation into the centre of mass of current images of neighbours). Panel C (right) shows the outcome of this process, i.e. the image, under the resulting map f, of the graph S_X where X is numeral '5'. Note that: (i) all pixel-value constraints are obeyed; (ii) considerable deformation is effected by f; and (iii) images of different nodes often overlap (whenever this is the case, these image nodes are represented slightly offset from each other). The total cost incurred is $H(f) = 878$.

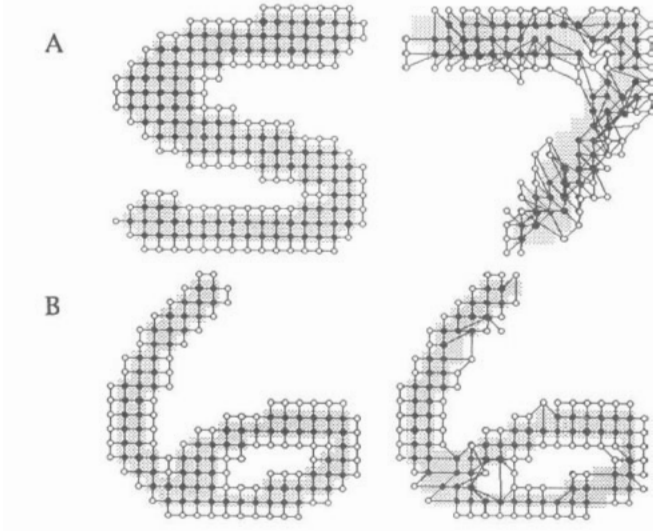

Figure 2. Local minima of the cost functional. Panel A shows the result of 10 further iterations—resulting in convergence to a local minimum—on the matching problem of figure 1; cost is $H(f) = 684$. Panel B shows, under the same conditions, the optimal map of a numeral '6' onto a slightly different realization of the same numeral, with a resulting cost of 88.

section, the width of the padding has little effect on classification performance. Finally, figure 5 shows an instance of the subset problem mentioned in section 2: the optimal map

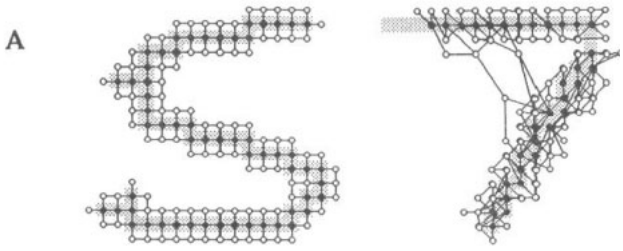

Figure 3. Elastic matching between thinned numerals. Except for thinning, the numerals and parameters are the same as in figure 2. The values of H reached are 478 (panel A) and 73 (panel B). Note that the cost $H(f)$ is, roughly, proportional to $|S_X|$, the area of the domain of f.

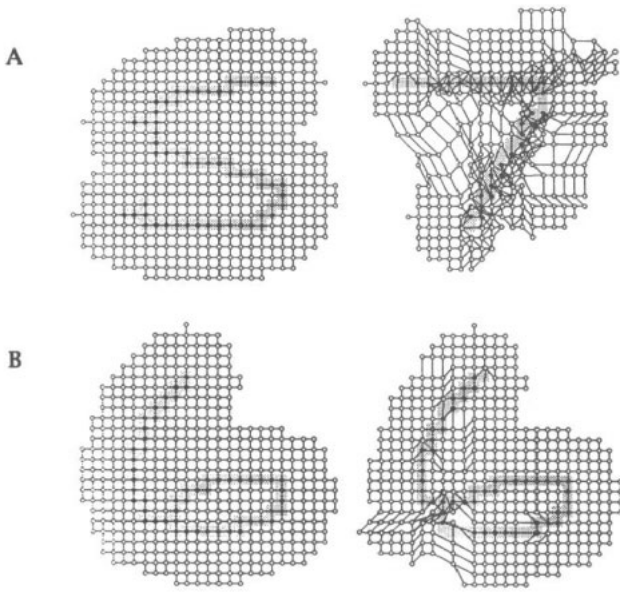

Figure 4. Elastic matching with a large padding of white pixels. Except for the value of m which is now 5 instead of 1, the situation is identical to that of figure 3. Costs are 1444 (panel A) and 198 (panel B). Note that even in the strong-deformation case large portions of the padding are mapped rigidly.

from numeral '3' to numeral '8' effects a relatively moderate distortion, and hence is only mildly penalized by H, whereas the optimal map from numeral '8' to numeral '3' incurs, as expected, a much higher cost. It is the latter that determines the distance μ between these two numerals; this distance is high, as required.

4. Classification experiments

This section reports on classification experiments that were carried out to assess the adequacy of both the distance μ and the optimization procedure described in section 3. We used a database of 1200 handwritten numerals, 120 per class, each a binary-valued image of size 16×16 (courtesy of I Guyon, AT&T Bell Labs). A sample of these images is shown in figure

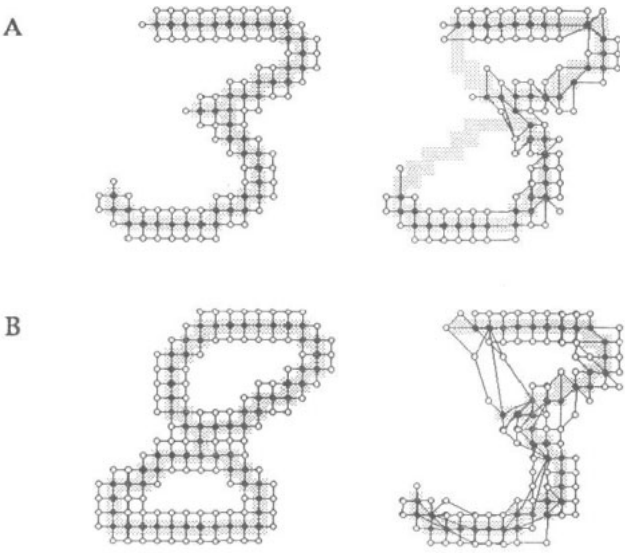

Figure 5. Elastic matching in the subset case. Mapping a numeral '3' on a numeral '8' (panel A) requires little deformation, as the former is a near subset of the latter. Resulting cost is $H(f) = 75$. In contrast, mapping the '8' on the '3' (panel B) entails considerable deformation (note for instance how the bottom circle of the '8' collapses onto the bottom leg of the '3'), resulting in a cost of 302. By definition, the (approximated) elastic-matching distance between these two numerals is the largest of the two values: $\mu = 302$.

6. Note that the numerals are normalized, so that their actual size (the size of the minimum enclosing rectangle) is 16 × 16 (except, for obvious reasons, for numeral '1'). These data were assembled by asking each of twelve individuals to produce 10 numerals of each class, following a given pattern. The shapes of these handwritten digits are therefore relatively uniform within a given class, and the recognition problem for this database is easier than for most currently used zip-code databases (e.g. Simard *et al* 1993). No further preprocessing or feature extraction was applied to the data, except for thinning the characters, as mentioned above.

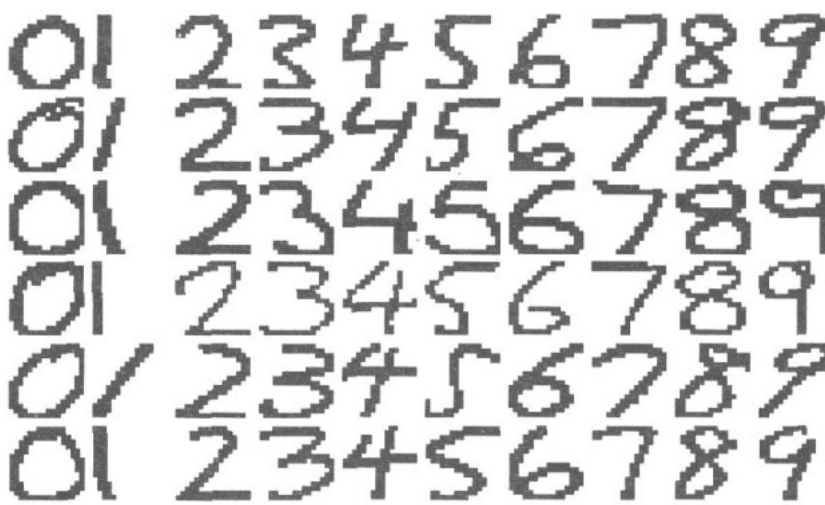

Figure 6. A sample of the 1200 handwritten numerals used in the classification experiments (courtesy of I Guyon).

The experiments reported in this section consist in using, in a non-parametric classification scheme, the elastic-matching metric μ defined in section 2—more accurately the approximated μ given by the update algorithm described in section 3—instead of the

usual pixelwise Hamming distance. We performed experiments using k-nearest-neighbour (k-NN) classification with various values of k, as well as kernel classification (Parzen windows) with various kernel bandwidths σ. Classification performances were found to be very similar for the two methods, and, within certain limits, independent of the 'smoothing parameter' (k or σ as the case may be). Here we shall report only on k-NN classification with $k = 1$. Results of experiments with k varying from 1 to 20 are briefly reported in Geman *et al* (1992) (see figure 17 there).

The default setting, which we shall use unless otherwise stated, is as follows: numerals are thinned; m, the width of the padding of white pixels around each numeral, is set equal to 1; κ, the weight of the injectivity constraint in the cost functional H is set equal to 2; the number of iterations in the optimization process is 0 (which means that we do apply the elastic-update scheme once to most of the sites in the domain of the function).

In all cases, we report on *generalization* error: the database of 1200 numerals is divided into two disjoint sets L and T (the partition is uniform across classes, but random *vis-à-vis* writers). L is used for 'training' (learning), T for 'testing'. There is of course no training in the strict sense here. Rather, numerals in L are used as prototypes; thus, in first-nearest-neighbour classification, the class of a numeral $X \in T$ is simply the class of that numeral $X' \in L$ such that $\forall X'' \in L$, $\mu(X, X') \leqslant \mu(X, X'')$. In order to achieve a robust estimate of error rates, 1000 different random partitions of the data base into two sets L and T were used; the error rate reported is the result of *averaging* over these 1000 partitions.

Figure 7 shows the error rate as a function of the total size of the training set L. As mentioned, the elastic distance is approximated by using only iteration 0 of the optimization process. Three curves are shown, for three different values of the padding width m. The

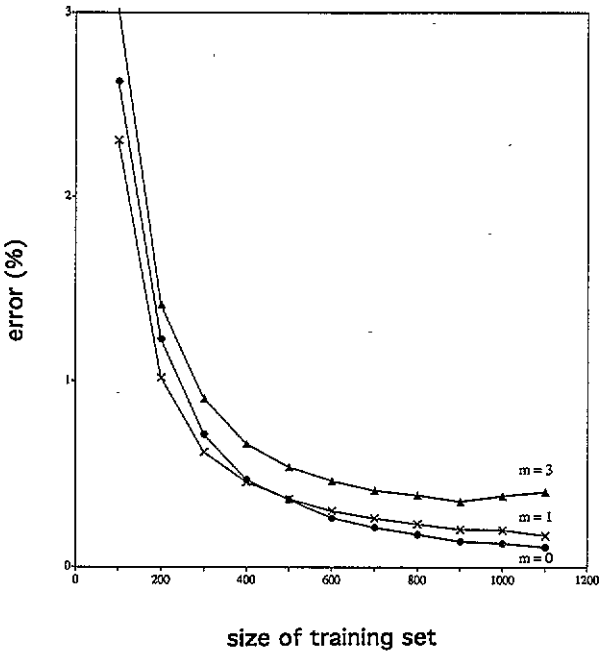

Figure 7. Percent error (generalization) as a function of total training-set size, with various padding widths m. First-nearest-neighbour classification is performed with the elastic-matching metric. Each point is an average error rate over 1000 random partitions of the database into a training set L and a test set T. Results with $m = 0$ and $m = 1$ are hardly distinguishable. Performance degrades slightly with $m = 3$.

curve with $m = 1$ shows for instance that with 500 randomly chosen prototypes (that is, 50 prototypes per class), the error rate is about 0.3%. It falls off to a value of about $0.17\% \approx 2/1200$ when $|L|$ approaches 1200. This is due to the presence of exactly two numerals whose first-nearest neighbours in the *whole* data base are of the 'wrong' class. Figure 7 also shows that performance is fairly insensitive to the presence or width of the padding.

Figure 8 illustrates the influence of κ, the relative weight of the injectivity term in the cost functional H. Including this term significantly improves the performance of the classifier, by a factor of about 3. On the other hand, the magnitude of κ does not appear to be crucial, as long as κ is neither too small (the effect of the second term would be negligible) nor too large (this would result in 'hardening' the injectivity constraint, which clearly is undesirable).

What is the effect of pursuing the optimization, rather than halting it after the initialization pass ('iteration 0')? Figure 9 shows that the improvement of performance with additional iterations is not very significant. Note that increasing the number of iterations beyond 5 does not bring any improvement at all; in effect, the update algorithm generally has *converged* by iteration 5. This is illustrated in figure 10, which shows the evolution of *average* inter- and intra-class approximated distances as a function of iteration number.

Experiments were also performed with different seeds for the random-number generator that determines the site-visitation sequence; the resulting variation of error rate was of the order of 0.1%. These data, along with the results shown in figure 9, may be taken as an indication of the *robustness* of our estimate of μ; they suggest that this approximated elastic distance probably yields essentially as good a classification as one would obtain were the true elastic distance μ available.

Experiments with non-thinned numerals resulted in performances essentially undistinguishable from results obtained with the thinned characters (differences in error rates did not exceed 0.1%). The advantage of thinning is a gain in computation time, as it reduces $|S_X|$ by a factor sometimes as large as 3.

Finally, figure 11 compares the performance of our elastic-matching classifier with a few simple non-parametric techniques. Of particular interest is the comparison with first-nearest-neighbour classification using pixelwise Hamming distance. This comparison shows that substituting the metric μ for Hamming distance results in a very significant drop of error rate, generally by a factor of more than 10. Note also the significant improvement over results obtained with various simple feedforward neural networks (data points from Guyon 1988). Feedforward neural networks introduce no other *bias* than smoothing with respect to the natural distance in input space. In this sense, they function essentially as non-parametric classifiers used with Hamming distance; they indeed yield comparable performances. See Geman *et al* (1992) for a more extensive discussion of this issue, as well as a comparison of the elastic-matching classifier with a backpropagation network including from 1 to 25 hidden units (see figure 17 there).

To summarize, using the elastic-matching metric results in very substantial improvement over methods relying explicitly (nearest-neighbour or Parzen-window classifiers) or implicitly (simple feedforward neural networks) on pixelwise distance.

5. Related work

Various forms of elastic matching for the recognition of handwritten numerals or other line drawings have been proposed in the past, generally independently of any biological consideration; see e.g. Burr (1980) and Tappert (1982). Of particular interest is the approach

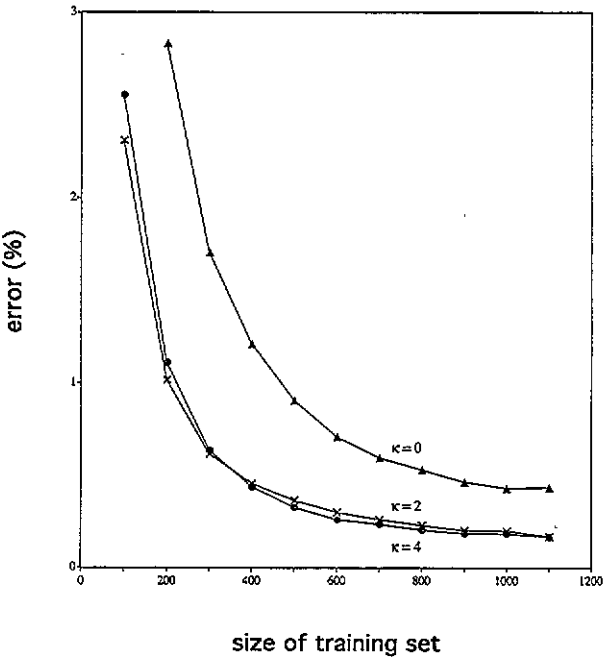

Figure 8. Influence of the injectivity constraint on classification performance. Including the H_2 term ($\kappa = 2$ or $\kappa = 4$) results in substantial improvement over the performance achieved with the sole H_1 'elastic' term.

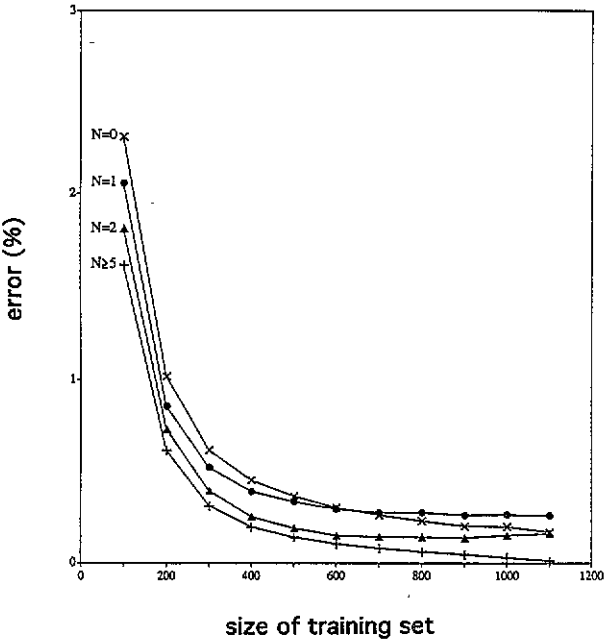

Figure 9. Influence of number of iterations (N) of the cost-minimization algorithm. The upper curve ($N = 0$) shows the error rate when optimization, in the computation of μ, is halted after the 'initialization' pass. The lower curve ($N \geqslant 5$) shows the error rate when the update algorithm is allowed to converge, which requires, in nearly all cases, at most five iterations.

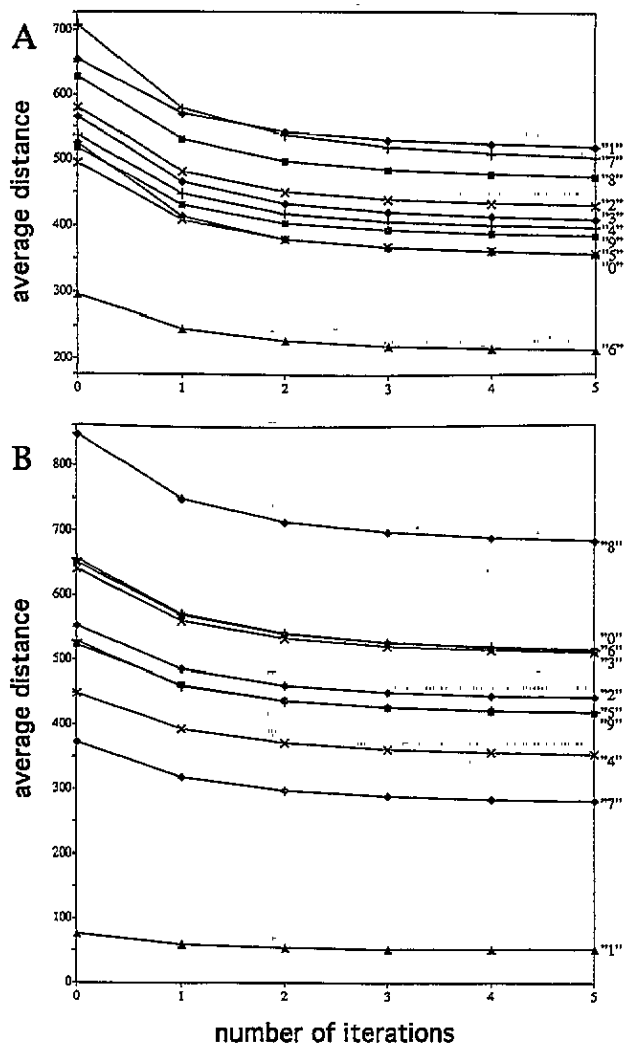

Figure 10. Average distance μ as a function of iteration number N. Panel A (resp. B) shows the distance between class '6' (resp. '1') and all 10 numeral classes. These distances are *averaged* over all numerals in the two classes concerned (for instance class '8' and class '1' for the upper curve in panel B). The x-axis indicates the number of iterations (N) of the optimization process used to compute the estimated value of μ. Although this value decreases in the first iterations, it does so fairly *uniformly* over class pairs, which makes classification relatively insensitive to N (figure 9).

investigated by Hinton *et al* (1992). These authors model a given numeral as a *deformable spline*, whose shape is determined by the positions of eight control points. These control points have *home locations* (adjustable by a learning procedure) that define an 'ideal' shape for the given character. The elastic matching between the image of an unknown numeral and the deformable spline is performed by an iterative procedure which includes, as an important step, the balancing of two types of *forces* acting on the eight control points: data forces that pull the control points towards black pixels in the image, and elastic forces that pull the points back to their home locations in the model; in the probabilistic setting used by Hinton *et al*, this step requires the inversion of a 16 × 16 matrix at every iteration.

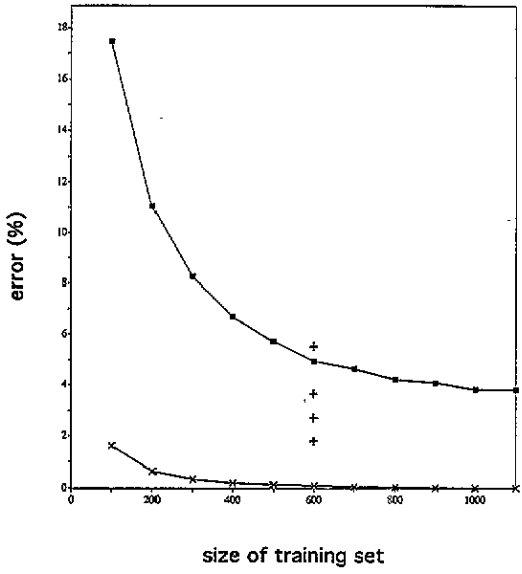

Figure 11. Elastic-matching classification versus Hamming-distance classification. The two curves represent averaged generalization-error rates (over 1000 partitions of the database) obtained by first-nearest-neighbour classification, using one of two alternative metrics: pixel-by-pixel Hamming distance (upper curve), or approximated elastic-matching distance μ (lower curve). The ratio between the two is in all cases larger than 10. Also indicated (from Guyon (1988)) are generalization-error rates obtained with various feedforward neural networks, on the same data base (single partition, $|L| = |T| = 600$). From top to bottom: no hidden layer, pseudo-inverse training rule; one hidden layer, backpropagation training rule; no hidden layer, delta-rule; no hidden layer, delta-rule, pre-processed data (99 extracted features).

The approach of Hinton *et al* bears strong resemblance to ours. It also uses a cost, or 'energy', function, to measure the amount of deformation effected on a character; this function includes an elastic deformation term, as well as a pixel-value term—in our approach, the latter is embodied in the hard constraint $f \in \mathcal{P}_{XX'}$. One important difference is that the deformable-spline approach uses a *higher-level* description: the model of a given numeral is entirely specified by 24 parameters (16 coordinates and 8 variances). This makes it a reasonable strategy to use a *single* model to account for all the variability encountered in each of the ten numeral classes. In contrast, our approach requires *several* exemplars for each numeral class (although not nearly as many as figure 7 would suggest—see below, section 6). The price paid in the deformable-spline approach is of course the rather heavy computation required to fit the data to the model. One advantage of this approach is that it lends itself rather naturally to the incorporation of noise models; it also affords total invariance with respect to substantial affine transformations, which our model does not (our data are normalized in size). However, this also comes at a computational price, since each iteration in the elastic-matching procedure includes the recomputation of the best affine transform between the image and an 'object-based frame'.

Note that full invariance to affine transformations is actually *not* a desirable feature for a character-recognition algorithm. The spread of parameters such as tilt and elongation within a given numeral class is indeed limited. Therefore, a matching algorithm that would attempt to perform the match under an excessively large domain of parameters would likely be less efficient, e.g. be prone to local energy minima; in an extreme situation, it may lead to the recognition of a '6' as a '9'. To address this problem in the deformable-spline approach, it

would probably be necessary to include additional terms in the energy function, in order to penalize large rotations or deformations. In contrast, our much simpler approach penalizes in a natural way *all* affine transformations except shifts, in an amount proportional to the magnitude of the deformation. The results reported by us and by Hinton *et al* (1992) do not make it possible at this point to judge which strategy is better adapted to the specific problem of handwritten-character recognition.

Another approach to handwritten-character recognition that calls for a comparison with ours is the one recently proposed by Simard *et al* (1993). Although these authors do not use an elastic-matching distance in the strict sense, the spirit of their work is, in part, similar to ours. They propose to replace Euclidean distance by a 'tangent distance' better suited to the task at hand, and to use this alternative metric for nearest-neighbour classification. Simard *et al* use grey-level images; their tangent distance D is designed to be *locally* invariant, in the 256-dimensional image space, to a number of standard transformations $\mathcal{T}$: translation, rotation, scaling, shearing, squeezing, and line thickening or squeezing. Given two images X and X', $D(X, X')$ is obtained by considering the manifold $\mathcal{M}_X$ of all $\mathcal{T}$-transforms of X, and the manifold $\mathcal{M}_{X'}$ of all $\mathcal{T}$-transforms of X'; $D(X, X')$ is the Euclidean distance between the hyperplane tangent to $\mathcal{M}_X$ at X and the hyperplane tangent to $\mathcal{M}_{X'}$ at X'. Simard *et al* report very low error rates when using tangent distance for the classification of large databases of handwritten digits.

Our method appears, at first sight, to be somewhat more effective computationally. As a rough indication, it requires about 6000 multiply adds—sometimes significantly less—to perform one match between two normalized digits of size 16×16; compare with the figure $n(m_E + 1)(m_P + 1) + 3(m_E^3 + m_P^3)$, with $n = 256$ and $m_E = m_P = 7$, of Simard *et al* (1993). A direct comparison, however, may be misleading, since we use binary-valued images, which contain significantly less information. An adaptation of our approach to grey-level images would necessitate a third term in the cost functional, to embody a suitable set of pixel-value constraints; this may make the algorithm significantly more computation-intensive. One possible advantage of our approach is that it handles all rubber-sheet deformations, which may be highly nonlinear. In contrast, the metric used by Simard *et al* is designed to be invariant to a standard set of transformations, applied uniformly throughout the image. It is not clear, however, how significant this difference may be for the problem of handwritten-digit recognition.

It would be interesting to assess the performance of our algorithm on larger databases of handwritten characters or numerals, and compare it more accurately with the approaches mentioned above, as well as with feedforward neural networks with shift-invariance constraints on the weights (Le Cun *et al* 1989). Our simple and general approach, designed in the spirit of biological modelling, may well turn out to be less efficient than techniques specifically designed to optimally recognize handwritten characters. The model presented in this paper—using the same elastic cost H_1—has also been applied to the recognition of shapes very different from numerals, e.g. images of human faces (Buhmann *et al* 1989, Würtz *et al* 1991, Lades *et al* 1993); in this application, images are pretreated by a family of 'Gabor-based' wavelet transforms, and a soft-constraint data term is used rather than a hard constraint of the type $f \in \mathcal{P}_{XX'}$.

6. Summary and discussion

This paper proposes a model of shape recognition with a specific biological motivation, namely to illustrate on a concrete problem the capabilities of the dynamic-link approach to brain function (von der Malsburg 1981, 1987, von der Malsburg and Bienenstock 1986).

Our simple elastic-matching formulation retains the spirit of the biological model—a map f from an image X to an image X' is a collection of dynamical links—but adapts it to the computational requirements of the application. Thus, the quality of the map f is assessed through an elastic cost functional $H(f)$, and an elastic-matching distance $\mu(X, X')$ is defined by minimizing H over a suitably defined collection of maps $\mathcal{P}_{XX'}$.

We presented a computationally effective procedure for finding a reliable estimate of $\mu(X, X')$. In experiments performed on a database of 1200 handwritten numerals, substituting the metric μ for Hamming distance in nearest-neighbour classification yielded substantial improvement (figure 11). Also, the performance of elastic-distance classification compared favorably with the performance reached on the same problem by *simple* feedforward neural networks. The implementation of this approach on parallel computing machinery (see e.g. Würtz *et al* 1990) may make it possible in the future to envisage realizations that come closer to the underlying biological model.

No effort was made in our work to optimize the speed of the classification proper. Two straightforward improvements would be: (a) effecting a *prefiltering*, by means a faster but less powerful classification technique, and (b) using a more parsimonious prototype set. The use of a *random* prototype set for first-nearest-neighbour classification is indeed very inefficient, as such a set contains many redundant exemplars. We have briefly experimented with a greedy algorithm designed to reduce the size of the exemplar set without increasing the classification error rate, as assessed on a *given* test set; these experiments (not reported in the present paper) confirm that considerable improvement is possible.

In the context of non-parametric classification, the import of our elastic-matching distance or of other, tailor-made, distances such as proposed by Simard *et al* (1993) can be usefully discussed in the perspective of the *bias/variance dilemma* for non-parametric estimation (see e.g. Geman *et al* 1992). Recall that the bias is the deviation of the average estimator from the theoretically optimal one, while the variance is its intrinsic variability; the stochasticity giving rise to these two terms is that of the *training* data, which obeys a given, unknown, probability distribution. The term 'dilemma' refers to the fact that it is difficult to improve the performance of a classifier by reducing both bias and variance in a fully general way, that is, independently of the problem considered. The way out of this dilemma, which brains must have adopted, is in the devising of appropriate problem-specific biases, which reduce the variance term without appreciably increasing the bias component, in a given problem.

Substituting the matching distance μ for pixelwise Hamming distance may be viewed as a way to introduce a problem-specific bias. In effect, consider generating various images X' from a given image X by flipping the values of n distinct pixels. The Hamming distance between X and X' is always n, whereas $\mu(X, X')$ will depend on the position of the pixels affected by the change. Specifically, $\mu(X, X')$ will be small if there is a low-H map in $\mathcal{P}_{XX'}$ as well as a low-H map in $\mathcal{P}_{X'X}$. This particular bias is well-suited to the problem at hand: we know beforehand, that is, before we are shown any examplars, that numerals related to each other through a moderate distortion are likely to belong to the same class. Therefore, introducing this bias *a priori* in the classifier results in better performance. In this perspective, the fact that the performance of the classifier hardly improves when optimization is pursued beyond iteration 0 may be interpreted by saying that iteration 0 introduces essentially all of the desired bias. Similarly, Simard *et al* (1993) report that the use of an *approximated* tangent distance (see above, section 5) results in no loss of classification performance.

Consider now the issue of neural mechanisms. As in statistical estimation or regression, *unbiased* computation would really mean that the only bias introduced is smoothness with

respect to the natural topology of the input space. An example of unbiased neural machinery might be a multilayer perceptron (MLP), assuming real brains indeed implement MLP-like networks: MLPs interpolate between training data smoothly with respect to the natural topology of the input space.

Biases can be introduced in MLPs by imposing constraints on the architecture and/or synaptic weights (Le Cun *et al* 1989). The dynamical-link approach underlying the present work suggests that a very different kind of bias may be present in *living* brains. Such a bias would rely on an operation of matching characterized by the construction of a relation-preserving dynamical map. Such a map differs from the map implemented by an MLP in two important ways: (i) there are no well-defined 'input' and 'output' spaces; rather, the map establishes a correspondence between two spaces of similar nature, both high-dimensional and containing relationally structured objects; (ii) the map is *dynamical*, that is, the very process of computation consists in the establishment of the map or in the failure to establish it. It has been suggested (von der Malsburg 1981, 1987, von der Malsburg and Bienenstock 1986) that brains may be equipped with a mechanism specialized in the building of dynamical structure-preserving maps; this mechanism could be a fast-enough form of Hebbian plasticity, sensitive to accurate temporal relationships between the firings of different neurons. The brain would then perform interpolation in a space of maps rather than in a space of sensory inputs. This would allow to introduce biases better-suited to handling various types of invariances, as may be pertinent in perception or in other domains of cognition. (For a further discussion of neural implications, see references above.)

In general, matching problems are hard, if not intractable. Thus, subgraph isomorphism is an NP-complete problem (Garey and Johnson 1979). The experiments presented in this paper show that satisfactory matches can be obtained reliably and rapidly (as measured by the number of parallel iterations) provided two general conditions are met: (i) the objects to be matched should be topologically structured, and (ii) initial conditions should provide a rough guess of the map to be constructed. It may be the case that these conditions are reasonably well satisfied in all instances of cognitive tasks—from perception and motor command to linguistic behaviour—that lend themselves to a description in terms of the computation of relation-preserving dynamical maps.

Appendix

Here we discuss in more detail the algorithm for finding a suboptimal match outlined in section 3. For any $f \in \mathcal{P}_{XX'}$, for any $s \in S_X$ and for any $s' \in S$ such that $X'(s') = X(s)$, define the map $f^{ss'}$ in $\mathcal{P}_{XX'}$ as follows:

$$f^{ss'}(t) = \begin{cases} f(t) & \text{if } t \neq s \\ s' & \text{if } t = s. \end{cases}$$

Given $f \in \mathcal{P}_{XX'}$, updating f at a given site $s \in S_X$ means finding a site $u \in S$ that is *optimal* given f on all sites other than s, that is, $X'(u) = X(s)$ and $H(f^{su}) \leqslant H(f^{ss'})$ for all $s' \in S$ such that $X'(s') = X(s)$ (note that u is not always uniquely defined).

Let V_s be the set of sites $t \in S_X$ at distance 1 from s. The size of V_s, $|V_s|$, is 4 if s is an interior point of S_X, less if it is a boundary point. Define

$$\tilde{s} = \frac{1}{|V_s|} \sum_{t \in V_s} f(t) + s - t. \tag{A1}$$

If s is an interior point of S_X, hence $|V_s| = 4$, $\tilde{s}$ simplifies to $\frac{1}{4} \sum_{t \in V_s} f(t)$, the centre of mass of the four points $f(t), t \in V_s$. The site $\tilde{s}$ is readily seen to be optimal with respect

to the 'elastic' component of the cost, H_1. We use $\tilde{s}$ for finding the optimal site u, as follows. We wish to evaluate, for any site $s' \in S$ such that $X'(s') = X(s)$, the total change in $H = H_1 + H_2$ resulting from moving $f(s)$ from a given site s_0 to site s'. This change is easily seen to be given by the following expression:

$$g(s') = H(f^{ss'}) - H(f^{ss_0}) = |V_s| \times \| s' - \tilde{s} \|^2 + \kappa \left(2|f^{-1}(s')| - 1\right)_+ + D \qquad \text{(A2)}$$

where D is a constant depending on s_0 but independent of s'. (A convenient choice of s_0 is $s_0 = \tilde{s}$.) The optimal u is then the site s' in S that minimizes g under the constraint $X'(s') = X(s)$ (or one of the minimizers if there are several).

An efficient search method for u is as follows. Visit all sites s' in S that satisfy $X'(s') = X(s)$ in order of increasing distance from $\tilde{s}$. (This order is not always uniquely defined; which order is used may determine which minimizer of g is found, if there are several.) For each site s' visited, ask whether $g(s')$ is smaller than the smallest value of g encountered so far. If so, provisionally mark s' as a candidate optimal site, and retain $g(s')$ as the current minimal value of g. As soon as a point s' is reached such that the H_1-component of $g(s')$, that is, $|V_s| \times \| s' - \tilde{s} \|^2$, is, *by itself*, larger than the current smallest value of g, discontinue the search and define u to be the site s' with lowest $g(s')$ found.

Note that when we use this update scheme in iteration 0, that is, when we extend the definition of f to all of S_X after having defined it by alignment on the first q black pixels (section 3), it is actually a first assignment that we are making rather than an update; V_s in equations (A1) and (A2) should then be understood as the set of all neighbours of s for which f has *already* been defined, rather than the whole set of neighbours of s in S_X. Thus, in order for the initialization procedure to be applicable, any site $s \in S_X$ to be 'updated' has to have at least one neighbour $t \in S_X$ for which $f(t)$ has already been assigned: either t is one of the initial q black sites, or $f(t)$ has itself already been 'updated'. The site-visitation ordering $s_{q+1}, \ldots, s_{|S_X|}$ of the set $S_X - \{s_1, \ldots, s_q\}$ is therefore random up to the requirement that for all i, $q < i \leqslant |S_X|$, there exist at least one $j < i$ such that $\| s_j - s_i \| = 1$.

Acknowledgments

This work has benefited from many discussions, over several years, with Christoph von der Malsburg and Stuart Geman. We are indebted to Donald Geman for pointing out to us the computational advantages of using a quadratic cost functional. We thank Isabelle Guyon and the AT&T Bell Laboratories for providing the database used in the experiments. Research was supported by the Commission of European Communities (Contract BRAIN ST2J-0416) and Office of Naval Research Grant N00014-91-J-1021.

References

Amit Y, Grenander U and Piccioni M 1991 Structural image restoration through deformable templates *J. Am. Stat. Assoc.* **86** 376–87

Bajcsy R and Kovacic S 1989 Multiresolution elastic matching *Comput. Vision, Graphics, Image Process.* **46** 1–21

Bienenstock E and von der Malsburg C 1987 A neural network for invariant pattern recognition *Europhys. Lett.* **4** 121–6

Bienenstock E and Doursat R 1989 Elastic matching and pattern recognition in neural networks *Neural Networks: From Models to Applications* ed L Personnaz and G Dreyfus (Paris: IDSET) pp 472–82

——1991 Issues of representation in neural networks *Representations of Vision: Trends and Tacit Assumptions in Vision Research* ed A Gorea (Cambridge: Cambridge University Press) pp 47–67

Bozinovic R O and Srihari S N 1989 Off-line cursive script word recognition *IEEE Trans. Pattern. Anal. Machine Intell.* **PAMI-11** 68–83

Buhmann J, Lange J and von der Malsburg C 1989 Distortion invariant object recognition by matching hierarchically labeled graphs *Proc. IJCNN Int. Conf. Neural Networks (Washington)* vol I (Piscataway, NJ: IEEE) pp 155–9

Burr D J 1980 Elastic matching of line drawings *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-3** 708–13

Dickinson S J, Pentland A P and Rosenfeld A 1992 From volumes to views: an approach to 3-D object recognition *CVGIP: Image Understanding* **55** 130–54

Garey MR. and Johnson DS. 1979 *Computers and Intractability* (New York: Freeman)

Geman S, Bienenstock E and Doursat R 1992 Neural networks and the bias/variance dilemma *Neural Comput.* **4** 1–58

Grenander U, Chow Y S and Keenan D 1990 *HANDS: A Pattern-Theoretic Study of Biological Shapes* (Berlin: Springer)

Guyon I 1988 Réseaux de neurones pour la reconnaissance des formes: architecture et apprentissage *Doctoral dissertation* University of Paris VI

Hinton G E, Williams C K I and Revow M D 1992 Adaptive elastic models for hand-printed character recognition *Advances in Neural Information Processing Systems 4* ed J E Moody, S J Hanson and R P Lippmann (San Mateo, CA: Morgan Kaufmann) pp 512–9

Hummel J E. and Biederman I 1992 Dynamic binding in a neural network for shape recognition *Psychol. Rev.* **99** 480–517

Kundu A, He Y and Bahl P 1989 Recognition of handwritten word: First and second-order hidden Markov model based approach *Pattern Recognition* **22** 283–97

Lades M, Vorbrüggen J C, Buhmann J, Lange J, von der Malsburg C, Würtz R P and Konen W 1993 Distortion invariant object recognition in the dynamic link architecture *IEEE Trans. Computers* **42** 300–11

Le Cun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1989 Backpropagation applied to handwritten zip-code recognition *Neural Comput.* **1** 541–51

Martin GL. and Pittman JA. 1991 Recognizing hand-printed letters and digits using backpropagation learning *Neural Comput.* **3** 258–67

von der Malsburg C 1981 *The Correlation Theory of Brain Function* Internal report 81-2, Max-Planck Institute for Biophysical Chemistry, Department of Neurobiology, Göttingen

——1987 Synaptic plasticity as basis of brain organization *The Neural and Molecular Bases of Learning* ed J P Changeux and M Konishi (New York: Wiley) pp 411–32

von der Malsburg C and Bienenstock E 1986 Statistical coding and short-term synaptic plasticity: A scheme for knowledge representation in the brain *Disordered Systems and Biological Organization* ed E Bienenstock, F Fogelman and G Weisbuch (Berlin: Springer) pp 247–72

Simard P, Le Cun Y and Denker J 1993 Efficient pattern recognition using a new transformation distance *Advances in Neural Information Processing Systems 5* ed S J Hanson, J D Cowan and C L Giles (San Mateo, CA: Morgan Kaufmann) pp 50–8

Tappert CC. 1982 Cursive script recognition by elastic matching *IBM J. Res. Develop.* **26** 765–71

Würtz R P, Vorbrüggen J C, von der Malsburg C and Lange J 1991 A transputer-based neural object recognition system *From Pixels to Features II—Parallelism in Image Processing* ed H Burkhardt, Y Neuvo and J Simon (Amsterdam: North-Holland) pp 275–94