Neural Techniques for Path Linking, with Applications to Image Processing

Robert I. Damper and Stuart J. Gilson

Image, Speech and Intelligent Systems (ISIS) Research Group,
Department of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK.

(emails: `rid|sjg95r@ecs.soton.ac.uk`)

Abstract

Edge linking is a fundamental computer-vision task, yet presents difficulties arising from the lack of information in the image. Viewed as a constrained optimisation problem, it is NP hard – being isomorphic to the classical travelling salesman problem. Self-learning neural techniques boast the ability to solve hard, ill-defined problems and, hence, offer promise for such an application. This paper examines the suitability of four well-known unsupervised techniques for the task of edge linking, by applying them to a test bed of edge point images and then evaluating their performance. Techniques studied are the elastic net, active contours ('snakes'), Kohonen map and Burr's modified elastic net. Of these, only the elastic net and the Kohonen map are realistic contenders for general edge-linking tasks, although special treatment of noise in the image is required.

1 Introduction

Extraction of features from images is an important computer-vision problem. In particular, edge-point images (here, binary points on a two-dimensional plane) are problematic as they contain very little information. A feature-extraction algorithm has to infer from the points a description of the object(s) present. For most non-trivial images, however, there could be many interpretations. Semantic knowledge of the form of the object may help but is not usually available.

One approach is to ignore noise, assume a single object only, and then to link the edge points to form a contour which should represent it. This reduces the problem to that of ordering the image points so as to describe the contour's path, which is isomorphic to the classical travelling salesman problem (TSP) [1, Sect. 4.2]. Hence, established techniques for solving the TSP can be adapted for edge linking.

Neural solutions to the TSP were first proposed by Hopfield and Tank [2], who mapped a cost function onto the Liapunov ('energy') function of a recurrent network. The activation of the ijth neuron is 1 if city j is visited at step i of the tour, and 0 otherwise. As the energy is guaranteed to reduce with each neural update until equilibrium is reached, the cost function for the tour is minimised. However, the approach performs badly on large problems ($>$ 30 cities) [3]. Durbin and Willshaw [4] overcame this limitation by using the same cost function as Hopfield and Tank, but enforcing some of the constraints more strongly based on the notion of a deformable model (elastic net) with an explicit structure. Simic [5] has shown that the techniques share the same statistical-mechanics framework, while Abrantes and Marques [6, 7] show that the elastic net, the Kohonen map [8] and the active contours algorithm of Kass, Witkin and Terzopoulos [9] are actually special cases of a common class of recursive algorithms which lend themselves to the edge-linking task. This paper provides an empirical comparison of these three unsupervised 'neural' techniques for edge linking, together with a modification by Burr [10] to the elastic net algorithm, which is claimed to reduce convergence times significantly without compromising the results.

The remainder of this paper is structured as follows. After formulating the edge-linking problem in the next section, we then describe (Section 3) the four neural techniques to be compared. Each is tested on a common, comprehensive test bed of images, described in Section 4, allowing the characteristic strengths and weaknesses of each technique to be seen. An important issue is how the free parameters of each neural model are set: this is detailed in Section 5. The methods are then evaluated with respect to their performance, using the methods described in Section 6. The results of this evaluation are detailed in

Section 7 before concluding (Section 8).

2 Problem Formulation

Given a vector of units, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]^T$, we wish to map its elements onto the vector of edge points, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N]^T$, such that the elements of $\mathbf{U}$ taken in ascending order represent a minimal-cost path through the points. Both $\mathbf{p}_i$ and $\mathbf{u}_j$ are two-dimensional specifying position in the plane. For a neural solution, the units will be formal 'neurons', inputs to the network are the (x, y) coordinates of the current image point, and weights between inputs and a neuron are interpreted as the 'position' of that neuron. The number of units, M, can be equal to the number of edge points, N, but is usually greater. The cost function (and, hence, the optimal path) depends upon the application, but here the Euclidean metric is used.

Two complications can arise when using TSP solutions for edge linking. First, the shortest path through the edge points is not necessarily the correct path. As edge-point density increases, however, the shortest path will tend towards the object contour. Secondly, the TSP assumes noise-free data, which is not usually realistic in computer-vision applications.

3 Neural Edge Linking

All four techniques examined here are unsupervised: hence, they make no assumptions about what is present in the image. They are all partially model-based – mapping a generic model onto the image by deforming it to achieve a near-optimal match.

3.1 Elastic Net

The vector $\mathbf{U}$ (see Section 2) is represented by the units of the network, with the goal being to move each of the M units onto each of the N cities, while still retaining the essentially circular configuration of the elastic net model. Thus, the algorithm has three constraints:

1. The path must pass through every edge point.
2. Adjacent units of the path should map to neighbouring edge points, ensuring that the distance between units of the network is minimised.
3. The path should be kept as circular and as short as possible, without cross over.

Accordingly, the units are subjected to two forces. One pulls each unit toward the nearest edge point (satisfying 1 and 2), while the other pulls each towards its two immediate neighbouring units (satisfying 3). The forces act in competition, with each controlled by a governing factor allowing the algorithm to favour one over the other. Thus, at each iteration of updating, the change made to the weights of the jth unit, $\mathbf{u}_j$, is given by:

$$\Delta \mathbf{u}_j = \alpha \sum_{i=0}^{N} G_{ij}(\mathbf{p}_i - \mathbf{u}_j) + K\beta(\mathbf{u}_{j+1} - 2\mathbf{u}_j + \mathbf{u}_{j-1}) \quad (1)$$

where α and β are the governing factors mentioned above, and K is a simulated-annealing term which is gradually reduced during convergence to the model's final form. The first term acts to move the jth unit towards whichever edge point has the greatest attraction, given as:

$$G_{ij} = \frac{\phi(|\mathbf{p}_i - \mathbf{u}_j|, K)}{\sum_{k=0}^{M} \phi(|\mathbf{p}_i - \mathbf{u}_k|, K)} \quad (2)$$

where $\phi(d, \sigma) = \exp\left(\frac{-d^2}{2\sigma^2}\right)$ is the Gaussian distribution function, with standard deviation σ. The second (regularisation or 'smoothing') term moves the unit towards the mid-point of its two immediate neighbours, so acting as an elastic or *tension* term.

The net behaves much like an elastic band. Initially, it moves through image features, forming a contour which is coarsely centered and oriented with respect to the distribution of edge points. It then contracts and deforms back towards the edge points as the net cools. During the early stages of convergence, when the 'temperature' is high, the net's structure is enforced quite strongly, allowing it to form a rough or vague solution yet retain its important, essentially circular configuration. Later, the temperature drops and the net becomes less rigid and more sensitive to image features, allowing it to deform more and specialise on a solution.

The simulated annealing is a very significant contributor to the performance of the technique. Abrantes and Marques [6] omitted the annealing schedule in their mathematical framework, and the performance of the simplified algorithm was very poor, often failing to link even simple images. Hence, controlling the annealing schedule is an important consideration. Durbin and Willshaw recommend that the initial temperature, $K(0)$, is set to 0.2 and reduced by 1% every n iterations. Here, $n = 1$ was found to be adequate. Annealing should cease when $K(t)$ reaches around 0.02 or 0.01 at iteration t – the lower value producing more accurate linking. This termination criterion is sufficient to define convergence. Thus, for $n = 1$, the technique will only require just under 300 iterations to find a solution for *any* image. While not as low as some of the other techniques studied here, this number can also be considered as a guaranteed maximum number of iterations. None of the other techniques has such a guaranteed maximum.

However, of the three basic approaches, the elastic net is the most computationally expensive. For every unit, the positions of each of the cities must be considered, each normalised by the positions of all the other units. Thus, each iteration of updating requires M^2N operations, and time complexity is $O(N^3)$.

3.2 Active Contours ('Snakes')

The active contours ('snakes') technique proposed by Kass, Witkin and Terzopoulos [9] is another deformable model, but with much greater regularisation ('smoothing') forces. This means that the behaviour is much more stable and predictable than the elastic net. The approach tries to model the structural aspects of the object, such as its edge-coherence, the smoothness of its outline, whether the object is open or closed, etc. via a deformable model – which has external energy (in much the same way as the elastic net), but which also has internal energy governing the model's behaviour.

The internal energy is composed of two terms. The first governs how it deforms elastically, effectively making it behave like a membrane, while the second component makes the snake behave inelastically (like a thin plate). Part of the difficulty of using the snake is in finding a suitable compromise between these two properties. Too much of the former will make the snake deform too easily, forming false edges, while too much of the latter will make the snake unduly rigid, preventing it from following the desired contour.

To attract the snake towards image features, suitable forces must be applied in the form of external energy. These may take several forms, such as lines, line terminals, and even texture functionals. The one of interest here is an edge-point functional – a Gaussian-scaled Euclidean distance measure (the same as equation (2), but without any normalisation). Further details are given in [11].

3.3 Kohonen Map

The Kohonen map was primarily designed for classification purposes but has been used for many applications in computer speech and vision [8, 12]. Because of its topological ordering property, the Kohonen map will form solutions with some geometric coherence, i.e. similar feature vectors will be nearby in the output map. In the context of edge linking, nearby edge points will be mapped by nearby units of the network. Thus, approach is subject to the same problems as the elastic net since, in the presence of noise, the 'correct' path does not necessarily run through adjacent edge points.

The vector $\mathbf{U}$ is again modelled by the array of network units. Given an input vector of edge points, $\mathbf{P}$, we find:

$$k_w = \arg\min_k |\mathbf{p}_i - \mathbf{u}_k| \quad 0 \le k \le M$$

where k_w denotes the index of the *winner* unit. The network is then updated by modifying the units in the neighbourhood of the winner unit according to:

$$\mathbf{u}_j(t+1) = \mathbf{u}_j(t) - \alpha(t)[\gamma(\mathbf{p}_i - \mathbf{u}_j(t))] \qquad (j = k_w - \beta, \ldots, k_w + \beta)$$

where α is a decay term for γ (the learning step) and β is the radius of the neighbourhood centered on the winner unit, and j is modulo-M. Note the similarity of this to the first term in the update rule, equation (1), of the elastic net. Both the neighbourhood and the learning rate are gradually reduced during convergence onto a solution.

3.4 Burr's Modified Elastic Net

The final technique studied is Burr's modification [10] of the elastic net algorithm, which entails the addition of a new attraction term and annealing mechanism, in an effort to reduce learning times. Whereas the original attraction term dictates where the unit will move by considering all the edge points, the new term only considers the attraction from the nearest edge point. So, although Burr doesn't draw this parallel, the new term effectively implements the Kohonen style of learning [1, p. 246].

The update rule is the same as for the elastic net except that the new term $v_{ij}(\mathbf{p}_i - \mathbf{u}^*(\mathbf{p}_i))$ is added to the summation in equation (1). Here, $\mathbf{u}^*(\mathbf{p}_i)$ is the nearest unit to the point $\mathbf{p}_i$, and:

$$v_{ij} = \frac{\phi(|\mathbf{u}^*(\mathbf{p}_i) - \mathbf{u}_j|, K)}{\sum_{k=1}^{M} \phi(|\mathbf{u}^*(\mathbf{p}_i) - \mathbf{u}_k|, K)}$$

The elastic net algorithm simply allowed the temperature to drop independently of the net's state. Burr, however, uses an interactive (state-dependent) control which reduces K according to how close the net is to a solution:

$$K = \gamma \sqrt{\frac{\sum_{i=1}^{N} |\mathbf{p}_i - \mathbf{u}^*(\mathbf{p}_i)|^2}{2N}}$$

where the factor 2 in the denominator is included to account for some minor differences between Burr's formulation and Durbin and Willshaw's. Hence, K is effectively the root-mean-square of the distance of the network from the edge points. Thus, as a solution is approached, the temperature is reduced allowing the network to become more specific to that solution, and learn more detail. This is a very powerful annealing mechanism, since it is state-dependent. Thus, it avoids some of the 'useless' temperatures (those in which the network doesn't really change much, and only really serve to extend learning times) that other mechanisms suffer from. It is probably the most significant contributor to Burr's claims of rapid ability to find solutions.

4 Image Test Bed

In order to test the techniques thoroughly, a set of artificial images was devised with the aim of demonstrating the sort of features which may occur in real images. There are 20 images, each 100 pixels square and consisting of black pixels on a white background. Space precludes a full specification here: the test bed images are detailed in [11]. They include simple geometric shapes, progressing through to more complex contours with concave corners, to complicated 'signature'-like images. Each was drawn by hand onto the unit square, and was drawn deliberately sloppily to simulate the noisy edge contours expected in real applications.

5 Parameter Settings

For each technique, appropriate parameter values had to be determined empirically. Two points should be made:

1. The values used are not the result of an exhaustive search of the parameter space. Consequently, the parameters used may not be optimal, although we believe they are at least near-optimal.

2. The parameter values were adjusted in an attempt to give best average performance across all the test images; they were not individually set for each image.

There is some compromise when using almost any algorithm. Here speed, accuracy, stability and robustness co-vary, and can be traded. In the context of this paper, *stability* refers to how well a technique handles small changes in its parameters: changing a single parameter should not lead to a wildly different result. On the other hand, *robustness* refers to a technique's ability to handle a wide variety of images without requiring fine tuning of its parameters. Our aim is to find a balance between fast learning and accuracy, while maintaining both the stability and robustness of the technique.

Parameter values actually used in the remainder of this work are specified in Appendix A.

6 Evaluation

The goodness of the edge-linking process can be evaluated computationally. For a given output image, two scores are produced as below. Starting with both set to zero:

1. for each edge point in the image, if there is a unit within a radius δ of the edge point then increment the first score by one and, finally, normalise by the number of edge points;

2. similarly, for each unit of the network, if there is an edge point within a radius δ of the unit then increment the second score by one and, finally, normalise by the number of units.

These two scores give a measure of *support* and *explanation* respectively [13]. The former refers to the way the data support the model (i.e. does the model encompass all the image features?) while the latter considers how well the model explains the data (i.e. does every part of the model refer to an image feature?) The image can only be considered fully linked if these two measures, lying in the range 0 to 100%, have high values.

We used a range of δ values – from 1 to 5. At $\delta = 1$, the scoring is maximally stringent, so support and explanation will both be low. We expect, of course, a general trend for support and explanation to increase with δ.

7 Results

For closed contours, the network was initialised as points on a closed, centered circle of radius 0.4 (in terms of the unit square on which the images were drawn), and which encompassed most, if not all, of the image features. For an open contour, the network was initialised as a straight, horizontal line, centered in the image. In both cases, the M units were evenly distributed over the length of the initial configuration. In all cases, $M = \lfloor \frac{5N}{2} \rfloor$, since $5N/2$ is the value recommended by Durbin and Willshaw [4] for the elastic net.

Figure 1 shows the average values of support and explanation for the four techniques as a function of the radius parameter δ. The predicted trend for increasing support and explanation as δ increases is clearly seen. The elastic net and Kohonen map out-perform the other two techniques. In view of the efforts put into finding suitable parameters, we do not believe these differences can be explained merely on the basis of inappropriate parameter settings for the snake and the modified elastic net.

Since these results are for gross averages taken across the 20 images, we do not infer that the elastic net and Kohonen map are always superior to the two other techniques.

8 Conclusions

This paper compares four unsupervised neural techniques on the task of edge linking of images. The emphasis has been towards real, fully-automatic applications where the techniques would be expected to link a set of images without any user intervention. Each technique has been implemented and then applied to a comprehensive test

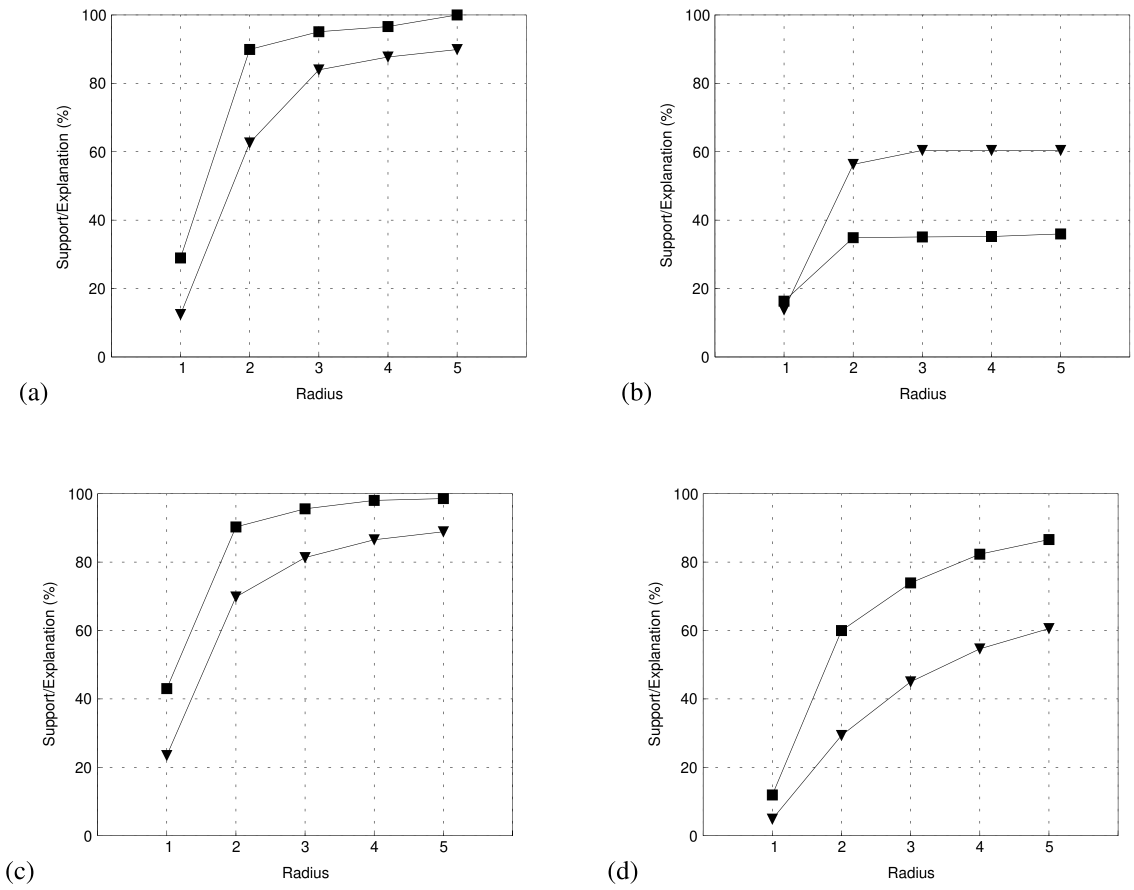

Figure 1: Support (filled box) and explanation (inverted filled triangle) as a function of the radius parameter δ for the four techniques studied: (a) elastic net; (b) active contours ('snake'); (c) Kohonen map and (d) Burr's modified elastic net. Values are averages across all 20 test bed images

bed of 20 images which aim to expose the various characteristics of the techniques. The 'best' set of parameter values has been determined, and the results of these 'best' tests have been evaluated.

To summarise, the elastic net has proven to be a very powerful and stable edge linker. Although relatively slow, it is reliable and seems capable of dealing with large problems. The active contours technique is quite the opposite – unable reliably to link images, and very sensitive to its initial position with respect to image features. It should not be entirely discounted, however, since it does have several characteristics which may make it suitable for particular, well-controlled applications [9]. Although the Kohonen map (unlike the other three techniques) has no regularisation term, it also manages to link the test images smoothly. It is also very fast, which is an important consideration for on-line feature extraction (as in automatic mail sorting based on postcode recognition, for example). Finally, the modified elastic net provides slightly faster linking, but poor results for all problem sizes, large and small.

Several avenues for future work suggest themselves. First of all, the TSP is an inherently noise-free problem. This is not the case in image edge-point linking. Hence, extensions of the methods studied here – to allow them to handle noisy images – are of considerable interest. We have recently extended the elastic net for this purpose [14].

The same value of M in terms of N – that recommended by Durbin and Willshaw [4] – was used for all the networks, as these are the only authors to specify a suitable value. Unfortunately, the recommendation appears incorrect: Simic [5] has shown that the elastic net only approximates the TSP tour length for large M, with the error tending to zero as $M \to \infty$. Hence, future work to explore suitable values for network size is planned.

While Kass *et al* [9] do not suggest an appropriate number of snake units to use for a given problem, it is clear that only enough units are required to give an accurate contour. Thus, there is not necessarily a relationship between M and N, but rather a relationship between M and some measure of the complexity of the target contour.

The Kohonen map, being designed for classification purposes, only requires in principle as many units as there are classes. Using more than the minimum number, however, can allow the map to produce better classifications. In this application, this can result in a more coherent contour.

An interesting area for future research is the investigation of dynamic node creation and/or destruction for these techniques. Amin [15] has done this with an elastic net/Kohonen map derivative with encouraging results.

The purpose of this work was to assess neural techniques for edge linking of images. Initially, we had hoped to identify a 'best' approach, which could then be the subject of further development: such as the incorporation of some notion of a 'correct' contour in the constrained optimisation – rather than just the shortest path. This might involve the development of a new cost function and/or imposing new constraints in addition to the existing ones. For example, Burr's algorithm (for reasons demonstrated above) may benefit from using a different error function for its temperature control. Similarly, the Kohonen map needs some method for dead-unit avoidance.

The work shows, however, that to identify a 'best' algorithm which should be used to the exclusion of all others is unrealistic. The specific characteristics of each and every application mean that no single technique is ideal for all of them. However, both the elastic net and the Kohonen map are quite well suited to most applications, yielding high support and explanation as well as being both stable and robust. As such, they provide a good starting point when searching for a solution to a specific problem. Interestingly, a combination of the two (which is effectively what Burr's technique is) did not necessarily produce a superior technique.

Acknowledgement

Author SJG is supported by a Research Studentship from the Engineering and Physical Sciences Research Council.

References

[1] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991.

[2] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, January 1985.

[3] G. V. Wilson and G. S. Pawley. On the stability of the travelling salesman problem algorithm of hopfield and tank. *Biological Cybernetics*, 58:63–70, January 1988.

[4] R. Durbin and D. Willshaw. An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, 326:689–691, April 1987.

[5] P. D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network*, 1:89–103, January 1990.

[6] A. J. Abrantes and J. S. Marques. Unified approach to snakes, elastic nets, and Kohonen maps. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '95*, 5:3427–3430, May 1995.

[7] A. J. Abrantes and J. S. Marques. A class of constrained clustering algorithms for object recognition. *IEEE Transactions on Image Processing*, 11:1507–1521, 1997.

[8] T. Kohonen. Clustering, taxonomy, and topological maps of patterns. In *Proceedings of 6th International Conference on Pattern Recognition*, pages 114–128, October 1982.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of 1st International Conference on Computer Vision*, pages 259–268, London, June 1987.

[10] D. J. Burr. An improved elastic net method for the tsp. In *IEEE International Conference on Neural Networks*, volume 1, pages 69–76, 1988.

[11] S. J. Gilson and R. I. Damper. An empirical comparison of neural techniques for edge linking of images. *Neural Computing and Applications*, 6:64–78, 1997.

[12] T. Kohonen. The 'neural' phonetic typewriter. *IEEE Computer Magazine*, pages 11–22, March 1988.

[13] C. K. I. Williams. *Combining deformable models and neural networks for hand-written digit recognition*. PhD thesis, University of Toronto, November 1994.

[14] S. J. Gilson, R. I. Damper, and I. Middleton. A semi-localized elastic net for surface reconstruction of objects from multi-slice images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998. Submitted.

[15] S. Amin. A self-organised travelling salesman. *Neural Computing & Applications*, 2:129–133, 1994.

A Network Parameters

This appendix details the parameters which gave the best overall results from the test bed, and which were fixed for the evaluation of each technique. They are not the result of an exhaustive search of the parameter space, but are based on refinements of samples taken from the space.

Note that, for consistency, there are some notational changes here from that used in the original sources. Where a parameter is defined here but does not appear in the text of this paper, it refers to the original source.

A.1 Elastic Net

For the elastic net, the parameters originally suggested in Durbin and Willshaw's paper ($\alpha = 0.2$, $\beta = 2.0$) gave good results. However, it was discovered that reducing β, hence reducing the net's tension, permits higher accuracy linking. Thus, the chosen parameters are: $\alpha = 0.2$, $\beta = 1.0$ and $K(0) = 0.2$, with the latter reduced by 1% every iteration until $K(t) = 0.01$ at iteration t. This annealing schedule allowed the net to link the images in 299 iterations.

A.2 Active Contours

The parameter space for the active contours technique is very large, but the criterion for quick learning helped reduce it to more manageable proportions. Given this reduction, the best parameters found are: $\sigma = 0.01$, $\gamma = 0.0001$, $a = 1000$, $b = 100$ and $h = 1.0$.

The ceiling for iterations was 5000.

A.3 Kohonen Map

The dimensionality of the Kohonen map's parameter space is not as great as the snake's, but was still large because the map can tolerate a wide range of values for each parameter.

Again, our requirement for rapid learning leads to a high value for the learning rate, which then allows some sensible restrictions to be placed on the other parameters: $\gamma = -0.4$, $\beta = \frac{M}{4}$ initially and was reduced linearly such that $\beta = 0$ after 100 iterations. The decay rate, α, was set to -0.004.

A.4 Modified Elastic Net

The best parameters found coincide with those suggested in Burr's paper: $\alpha = 0.5$ and $\beta = 1.0$ with a ceiling on the number of iterations of 100. This allowed most images to be linked correctly, but a few failed to converge fully.