Learning 2*D* Hand Shapes Using the Topology Preservation Model GNG

Anastassia Angelopoulou[1], José García Rodríguez[2], and Alexandra Psarrou[1]

[1] Harrow School of Computer Science, University of Westminster,
Harrow HA1 3TP, United Kingdom
{agelopa, psarroa}@wmin.ac.uk
[2] Departamento de Tecnología Informática y Computación, Universidad de Alicante,
Apdo. 99. 03080 Alicante, Spain
jgarcia@dtic.ua.es

Abstract. Recovering the shape of a class of objects requires establishing correct correspondences between manually or automatically annotated landmark points. In this study, we utilise a novel approach to automatically recover the shape of hand outlines from a series of 2D training images. Automated landmark extraction is accomplished through the use of the self-organising model the growing neural gas (GNG) network which is able to learn and preserve the topological relations of a given set of input patterns without requiring a priori knowledge of the structure of the input space. To measure the quality of the mapping throughout the adaptation process we use the topographic product. Results are given for the training set of hand outlines.

1 Introduction

Modelling the shape of a class of non-rigid objects in two-dimensions requires the recovery of their structure from a set of images. A common modelling approach is the observation and analysis of a set of examples of the object or class of objects using standard statistical methods such as principal component analysis (PCA). This approach has turned out to be very effective in image segmentation and interpretation. The basic idea of statistical shape modelling is to establish new unseen legal instances of shapes taken from a given set of training examples, using as few parameters as possible. Shape training sets usually come from manually annotated boundaries. The difficulty arises over the need to automate the process. For example, in a clinical setting the first stage in the post-processing step of a T1-weighted MRI technique is to segment out the ventricles, which can be difficult in many cases if the patient is not properly aligned in the scanner. These post-processing step is laborious and must be very accurate if the purpose of the scan is to help determine the extent of disease progression. In very overburdened medical facilities, performing this task manually may not be feasible. An automated procedure may provide the means of yielding objective and consistent results across various institutions. It is imperative therefore that an accurate, rapid and automated algorithm be developed and deployed.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part I, LNCS 3951, pp. 313–324, 2006.

In literature, various attempts have been made to automate the process of landmark based image registration and correct correspondences among a set of shapes. Baumberg's *et al.* [3] method, which generates flexible shapes models by using equally spaced spline control points around the boundaries of walking pedestrians, is an example of arbitrary parameterisation. The process is automatic, but it is arbitrary since it uses properties of the specific shape being modelled (each shape has a principal axis) thus, not generally applicable.

Davies *et al.* [6] method of automatically building statistical shape models by re-paremeterising each shape from the training set and optimising an information theoretic function to assess the quality of the model has received a lot of attention recently. The quality of the model is assessed by adopting a minimum description length (MDL) criterion to the training set. The MDL is obtained from information theoretic considerations and recently has received a lot of attention due to its ability to locate dense correspondence between the boundaries [18, 6, 7]. This is a very promising method and the models that are produced are comparable to and often better than the manual built models. However, due to very large number of function evaluations and nonlinear optimisation the method is computationally expensive.

Cremer's *et al.* [5] method of automatically constructing statistical shapes from a training set by combining the external energy of the Mumford-Shah functional with the internal energy of the snakes in a single variational framework, has improved segmentation in cases where occlusion or strongly cluttered backgrounds occur. In the case of learning $2D$ shapes the method it's fully automatic as long as no open boundaries or contour splitting are emerged.

Recently, Fatemizadeh *et al.* [8] have used modified growing neural gas to automatically correspond important landmark points from two related shapes by adding a third dimension to the data points and by treating the problem of correspondence as a cluster-seeking method by adjusting the centers of points from the two corresponding shapes. This is a promising method and has been tested to both synthetic and real data, but the method has not been tested on a large scale for stability and accuracy of building statistical shape models.

In this work, we introduce a new and computationally inexpensive method for the automatic selection of landmarks along the contours of $2D$ hand shapes. The novelty in using the Growing Neural Gas method for unsupervised learning is that we can automatically construct statistical shape models independently of closed or open shapes in contrast to Kass *et al.* [11] "Active Contour Models - Snakes" which can be defined only for closed contours. Furthermore, the incremental neural network, the growing neural gas (GNG) is used to automatically annotate the training set without using *a priori* knowledge of the structure of the input patterns. Unlike other methods, the incremental character of the model avoids the necessity to previously specify a reference shape. To evaluate the accuracy of the method we have tested it with other self-organising models such as Kohonen maps and Neural Gas (NG) maps and we applied the topographic product [2] to measure the best topology preservation of the order-preserving map.

The remaining of the paper is organised as follows. Section 2 introduces the statistical shape models. Section 3 provides a detailed description of the topology learning algorithm GNG. Section 4 reviews the topographic product, an existing measure used to quantify the topography of neural maps. A set of experimental results along with qualitative analysis is presented in Section 5, before we conclude in Section 6.

2 Statistical Shape Models

When analysing deformable shapes like hands it is convenient and usually effective to describe them using statistical shape models. The most well known statistical shape models are Cootes *et al.* [4] 'Point Distribution Models' (PDMs) that models the shape of an object and its variation by using a set of n_p landmark points from a training set of S_i shapes. In this work, PDM represents the hands as a set of n_p automatically extracted landmarks (in our case 64, 100, 144 and 169 neurons) in a vector $\mathbf{x} = [x_{i0}, x_{i1}, ..., x_{in_{p-1}}, y_{i0}, y_{i1}, ..., y_{in_{p-1}}]^T$. In order to generate flexible shape models the S_i shapes are aligned (translated, rotated, scaled) and normalised (removing the centre-of-gravity and placing it at the origin) to a common set of axes. The modes of variations of the hands are captured by applying principal component analysis (PCA). The i^{th} shape in the training set can be back-projected to the input space by a linear model of the form:

$$\mathbf{x} = \overline{\mathbf{x}} + \Phi\beta_i \qquad (1)$$

where $\overline{\mathbf{x}}$ is the mean shape, Φ describes a set of orthogonal modes of shape variations, and β_i is a vector of weights for the i^{th} shape. To ensure that the above weight changes describe reasonable variations we restrict the weight β_i to the range $-3\sqrt{\lambda} \leq \beta_i \leq 3\sqrt{\lambda}$ and the shape is back-projected to the input space using Equation (1). PCA works well as long as good correspondences exist. To obtain the correspondences and represent the contour of the hands a self-organising network GNG was used.

3 Topology Learning

One way of selecting points of interest along the contour of 2D shapes is to use a topographic mapping where a low dimensional map is fitted to a higher dimensional manifold, whilst preserving the topographic structure of the data. A common way to achieve this is by using self-organising neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this mapping a representation of the input patterns is achieved that in postprocessing stages allows one to exploit the similarity relations of the input patterns. Such models have been successfully used in applications such as speech processing [12], robotics [17, 14] and image processing [16]. However, most common approaches are not able to provide good neighborhood and topology preservation if the logical structure of the input patten is not known *a priori*. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example, a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case the networks fail to provide good topology preserving as for example in the case of Kohonen's algorithm.

The approach presented in this paper is based on self-organising networks trained using the Growing Neural Gas learning method [9]. This is an incremental training algorithm where the number of units in the network are determined by the unifying measure for neighborhood preservation [10], the topographic product. The links between the units in the network are established through competitive hebbian learning [13]. As a result the algorithm can be used in cases where the topological structure of the input pattern is not known *a priori* and yields topology preserving maps of feature manifold [15].

3.1 Growing Neural Gas

With Growing Neural Gas (GNG) [9] a growth process takes place from minimal network size and new units are inserted successively using a particular type of vector quantisation [12]. To determine where to insert new units, local error measures are gathered during the adaptation process and each new unit is inserted near the unit which has the highest accumulated error. At each adaptation step a connection between the winner and the second-nearest unit is created as dictated by the competitive hebbian learning algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology based on the topographic product [10]. This measure is used to detect deviations between the dimensions of the network and that of the input space, detecting folds in the network and, indicating that is trying to approximate to an input manifold with different dimensions. In addition, in GNG networks learning parameters are constant in time, in contrast to other methods whose learning is based on decaying parameters.

The network is specified as:

- A set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. The edges are determined using the competitive hebbian learning algorithm. An *edge aging scheme* is used to remove connections that are invalid due to the activation of the neuron during the adaptation process.

The GNG learning algorithm to approach the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in R^d.
2. Generate at random an input pattern ξ according to the data distribution $P(\xi)$ of each input pattern. Since the input space is the contour, $1D$ manifold,

the input pattern is the (x, y) coordinate of the edges. Typically, for the training of the network we generated 1000 to 10000 input patterns depending on the complexity of the input space.

3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2 by:

$$s_1 = \arg\min_{c \in A} \parallel \xi - w_c \parallel \tag{2}$$

and

$$s_2 = \arg\min_{c \in A\{s_1\}} \parallel \xi - w_c \parallel \tag{3}$$

4. Increase the age of all the edges emanating from s_1:

$$age_{(s_1,i)} = age_{(s_1,i)} + 1 \ (\forall i \in N_{s1}) \tag{4}$$

5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1 such as:

$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2 \tag{5}$$

6. Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards ξ by a learning step ϵ_w and ϵ_n, respectively, of the total distance:

$$\Delta w_{s_1} = \epsilon_w(\xi - w_{s_1}) \tag{6}$$

$$\Delta w_{s_n} = \epsilon_w(\xi - w_{s_n}) \tag{7}$$

for all direct neighbours n of s_1.

7. If s_1 and s_2 are connected by an edge, set the age of this edge to 0.

$$age_{(s_1,s_2)} = 0 \tag{8}$$

If it does not exist, create it.

8. Remove the edges larger than a_{max}. If this results in isolated neurons (without emanating edges), remove them as well.

9. Every certain number λ of input patterns generated insert a new neuron as follows:

   – Determine the neuron q with the maximum accumulated error:

$$q = \arg\max_{c \in A} E_c \tag{9}$$

– Determine among the neighbours of q the neuron f with the maximum accumulated error:

$$f = \arg\max{}_{c \in N_q} E_c \tag{10}$$

– Insert a new neuron r between q and its further neighbour f:

$$w_r = 0.5(w_q + w_f) \tag{11}$$

– Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.

10. Decrease the error variables of neurons q and f multiplying them by a fraction α:

$$\Delta error_(q) = -\alpha E_q \tag{12}$$

$$\Delta error_(f) = -\alpha E_f \tag{13}$$

11. Initialize the error variable of r with the new value of the error variable of q and f.

$$E_r = \frac{(E_q + E_f)}{2} \tag{14}$$

12. Decrease all error variables by multiplying them with a constant γ:

$$\Delta error_(c) = -\gamma E_c \tag{15}$$

13. If the stopping criterion is not yet achieved (in our case the number of neurons), go to step 2.

The algorithm was tested with three different topology preserving networks so that evaluation of the best topological map can be achieved. The testing involved two cases were the number of neurons were too few or too excessive for the training set of the images. In the former the topological map is lost, not enough neurons to represent the contour of the hands and in the later an overfit is performed. The parameters used in all simulations were: $\lambda = 1000$, $\varepsilon_w = 0.1$, $\varepsilon_n = 0.001$, $\alpha = 0.5$, $\gamma = 0.95$, $\alpha_{max} = 250$.

3.2 Characterising Hand Shape Using GNG

Given an image $I(x, y) \in \Re$ of the object we perform the transformation $\Psi_\nabla(x, y) = \nabla(I(x, y))$ that associates to each one of the pixels its probability of belonging to the contour of the object (Figure 1A, 1B and 1C). If we consider $\xi = (x, y)$ and $P(\xi) = \Psi_\nabla(\xi)$ we can apply the learning algorithm of the GNG to the image I, so that the network adapts its topology to the contours. The result of the learning process is a list of non ordered neurons representing the contour of the

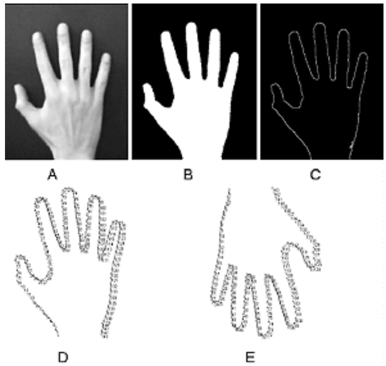

Fig. 1. Image A represents original image in grey level, in B threshold is applied that converts to B/W, in C the contour is obtained, and in D and E the neurons obtained from the adaptation process and the reordering of the neurons

hand. The list of neurons define a graph. To normalise the graph that represents the contour we must define a starting point, for example the neuron on the left-bottom corner. Taking that neuron as the first we must follow the neighbours until all the neurons had been added to the new list. The results of GNG reordering the neurons and the normalised neurons can be seen in Figure 1D and 1E. Since we want to apply the result of the neural network adaptation to the automatically annotation of the $2D$ contour, it is important that the result preserves the topology correctly. For this reason, we have used the topographic product as a measure to quantify this goal.

4 Measuring Topology Preservation

The topographic product [2] was one of the first attempts of quantifying the topology preservation of self-organizing neural networks. This measure is used to detect deviations between the dimensions of the network and that of the input space, detecting folds in the network and, indicating that is trying to approximate to an input manifold with different dimension.

In our case it is used to determine the optimum number of neural units that can be used to describe the $2D$ shape of a hand. This can be thought as an alternative to the MDL objective function introduced by Davies *et al.* [6].

4.1 Topographic Product

This measure compares the neighbourhood relationship between each pair of neurons in the network with respect to both their position on the map ($P_2(j,k)$) and their reference vectors ($P_1(j,k)$):

$$P_1(j,k) = [\prod_{l=1}^{k} \frac{d^V(w_j, w_{n_l^A(j)})}{d^V(w_j, w_{n_l^V(j)})}]^{1/l} \tag{16}$$

$$P_2(j,k) = [\prod_{l=1}^{k} \frac{d^A(j, n_l^A(j))}{d^A(j, n_l^V(j))}]^{1/l} \quad (17)$$

where j is a neuron, w_j is its reference vector, n_l^V is the l-th closest neighbour to j in the input manifold V according to a distance d^V and n_l^A is the l-th nearest neuron to j in the network A according to a distance d^A . Combining (6) and (7) a measure of the topological relationship between the neuron j and its k closer neurons is obtained:

$$P_3(j,k) = [\prod_{l=1}^{k} \frac{d^V(w_j, w_{n_l^A(j)})}{d^V(w_j, w_{n_l^V(j)})} \cdot \frac{d^A(j, n_l^A(j))}{d^A(j, n_l^V(j))}]^{1/2k} \quad (18)$$

To extend this measure to all the neurons of the network and all the possible neighborhood orders, the topographic product P is defined as:

$$P = \frac{1}{N(N-1)} \sum_{j=1}^{N} \sum_{k=1}^{N-1} \log(P_3(j,k)) \quad (19)$$

The sign of P indicates the topological relation of the input and the output space. $P < 0$ corresponds to a too low-dimensional input space, $P \approx 0$ indicates an approximate match, and $P > 0$ corresponds to a too high-dimensional input space [1]. In our case the negative values of the topographic product indicate the low-dimensionality of the input network.

5 Experiments

To illustrate the performance of the convergence algorithm described in Section 3, we present qualitative (Figure 3) and quantitative (Table 1) results for both manually and automatically generated models. The hand database, was composed of images of four individuals who contributed with four images of their right hand and at different poses (two of the fingers, the middle and the ring were captured at various displacements). We used 16 hand shapes which were extracted from the training set by thresholding. All images were of same size 395x500 pixels. The comparison was made by taking two reference models, a manually annotated hand model with 60 landmarks, and an automatic growing neural gas hand model with 144 neurons (Figure 2).

In Figure 3 two shape variations from the automatically generated landmarks were superimposed to the training set and the in between shape instances are drawn which shows the flexing of middle finger and hand rotation. These modes effectively capture the variability of the training set and present only valid shape instances. The quantitatively results (Table 1) show that the automatically generated models are more compact than the manual models since less variance is

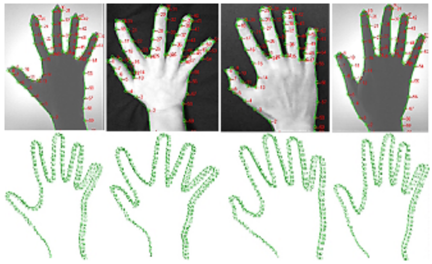

Fig. 2. First row manually annotated landmarks. Second row GNG with 144 neurons.

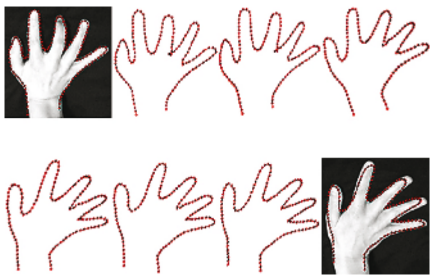

Fig. 3. Superimpose shape instances to the training set and taking the in between steps

Table 1. The results for the hand models

Mode	Manual model	Automatic model (144 neurons)
1	5.6718	1.5253
2	2.3005	1.1518
3	1.6976	0.9808
4	0.9896	0.3968
5	0.6357	0.3716
6	0.4713	0.1980
V_T	13.227	5.1783

captured per mode. It is interesting to note the big difference in the total variance between the two reference models. This may be because of errors in the manual annotation since all points were manually located and because of the difference of the number of points selected in the manual annotation. Table 2 shows the total variance achieved by maps containing varying number of neurons (25, 64, 100, 144, 169) used for the automatic annotation (Figure 4). The map of 144 neurons is the most compact since it achieves the least variance. This is constant with the optimal mapping selected by the topographic product. It is interesting to note that whilst there is significant difference between 25, 64 and 100 neurons (not enough neurons to represent the object) the mapping with

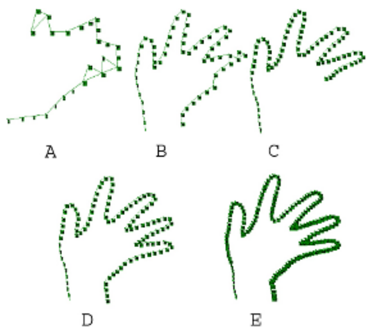

Fig. 4. Network size of 25 (A), 64 (B), 100 (C), 144 (D), and 169 (E), neurons

Table 2. A quantitative comparison of various neurons adapted to the hand model with variances for the first six modes, total variance and the topographic product

Mode	25 (neurons)	64 (neurons)	100 (neurons)	144 (neurons)	169 (neurons)
1	2.1819	4.2541	3.2693	1.5253	2.5625
2	1.2758	2.2512	1.4869	1.1518	0.9266
3	0.6706	0.5681	0.6154	0.9808	0.5734
4	0.4317	0.4645	0.4977	0.3968	0.3101
5	0.3099	0.2844	0.3532	0.3716	0.2491
6	0.2305	0.2489	0.1292	0.1980	0.1927
V_T	5.7486	8.6170	6.4108	5.1783	5.2470
T_P	0.0099	-0.018	-0.023	-0.024	-0.024

Table 3. The topographic product at different input patterns

Patterns	25 (neurons)	64 (neurons)	100 (neurons)	144 (neurons)	169 (neurons)
1000	0.013	-0.017	-0.021	-0.024	-0.025
5000	0.0099	-0.018	-0.023	-0.024	-0.024
10000	0.007	-0.018	-0.022	-0.021	-0.023

169 is good and has no significant difference with the mapping of 144 neurons. The reason is that for the current size of the images the distance between the neurons is short enough so adding extra neurons does not give more accuracy in placement. Thus, the topographic product for 144 and 169 neurons at 5000 input patterns is the same as can be seen from the Table 2. Table 3 shows the topographic product at different neurons and at different patterns. A qualitative representation of the topographic product is given in Figure 5. The introduction of extra neurons slows down the adaptation process. Figure 6 shows a comparative diagram of the learning time of various neurons and at different number of input pattern ξ. The adaptation with the 144 neurons is faster compared to the 169, and it takes 22 seconds at 5000 patterns to adapt to the contour of the hand.

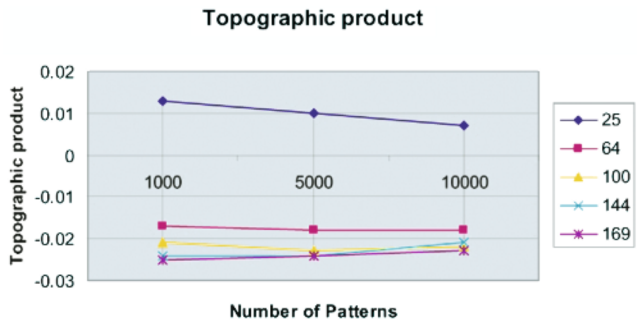

Fig. 5. Topographic product at different input patterns and at different number of neurons as a measure of the topology preservation of the network

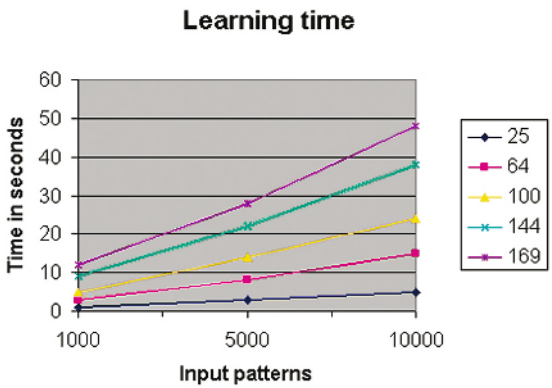

Fig. 6. Learning time for various neurons and at different input patterns

6 Conclusions

In this paper, we have used an incremental self-organising neural network (GNG) to automatically annotate landmark points on a training set of hand outlines. We have shown that the low dimensional incremental neural model (GNG) adapts successfully to the hand manifold, allowing good eigenshape models to be generated completely automatically from the training set. We have shown that these automatic models are more compact than manually landmark models as have been measured in terms of the total variance. Practically we have shown that the optimum number of neurons required to represent the contour depends mainly on the resolution of the input space and if it is not sufficient then the topology preservation is lost. In future work, the method needs to be tested to several sets of outlines since the number of neurons selected depends on the shape of the object.

References

1. H. Bauer, M. Herrman, and T. Villman. Neural maps and topographic vector quantization. *Neural Networks*, 12(4-5):659–676, 1999.
2. H.U. Bauer and K.R. Pawelzik. Quantifying the neighbourhood preservation of self-organizing feature maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.
3. A. Baumberg and D. Hogg. Learning flexible models from image sequences. *3rd European Conference on Computer Vision*, 1:299–308, 1994.
4. T. F. Cootes, C. J. Taylor, D. H. Cooper, and Graham J. Training models of shape from sets of examples. *3rd British Machine Vision Conference*, pages 9–18, 1992.
5. D. Cremers, C. Schnorr, and J. Weickert. Diffusion-snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.
6. H. Rhodies Davies, J. Carole Twining, F. Tim Cootes, C. John Waterton, and J. Chris Taylor. A minimum description length approach to statistical shape modeling. *IEEE Transaction on Medical Imaging*, 21(5):525–537, 2002.
7. Anders Ericsson and Kalle Åstróm. Minimizing the description length using steepest descent. *In 14th British Machine Vision Conference*, 2, 2003.
8. E. Fatemizadeh, C. Lucas, and H. Soltania-Zadeh. Automatic landmark extraction from image data using modified growing neural gas network. *IEEE Transactions on Information Technology in Biomedicine*, 7(2):77–85, 2003.
9. B. Fritzke. A growing neural gas network learns topologies. *In Advances in Neural Information Processing Systems 7*, pages 625–632, 1995.
10. J. Geoffrey, F. Goodhill, and J. Terrence. A unifying measure for neighbourhood preservation in topographic mappings. *Proceedings of the 2nd Joint Symposium on Neural Computation*, 5:191–202, 1997.
11. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 4(1):321–331, 1987.
12. T. Kohonen. *Self-organising maps*. Springer Verlag, 2001.
13. T. Martinez. Competitive hebbian learning rule forms perfectly topology preserving maps. In *ICANN.*, 1993.
14. T. Martinez, H. Ritter, and K. Schulten. Three dimensional neural net for learning visuomotor-condination of a robot arm. *IEEE Transactions on Neural Networks*, 1:131–136, 1990.
15. T. Martinez and K. Schulten. Topology representing networks. *The Journal of Neural Networks*, 7(3):507–522, 1994.
16. M. Nasrabati and Y. Feng. Vector quantisation of images based upon kohonen self-organizing feature maps. In *Proc. IEEE Int. Conf. Neural Networks.*, pages 1101–1108, 1988.
17. H. Ritter and K. Schulten. Topology conserving mappings for learning motor tasks. In *Neural Networks for Computing, AIP Conf. Proc.*, 1986.
18. Hans Henrik Thodberg and Hildur Olafsdottir. Adding curvature to minimum description length shape models. *In 14th British Machine Vision Conference*, 2:251–260, 2003.