MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
4-2

Automatic landmark extraction using Growing Neural Gas (GNG)

Anastassia Angelopoulou
Dept. of Computer Science
University of Westminster, London, UK
agelopa@wmin.ac.uk

José García
Dept. de Tecnología Informática y Computación
Universidad de Alicante, Alicante, Spain
jgarcia@dtic.ua.es

Alexandra Psarrou
Dept. of Computer Science
University of Westminster, London, UK
psarroa@wmin.ac.uk

Abstract

A new method for automatically building statistical shape models from a set of training examples and in particular from a class of hands. In this method, landmark extraction is achieved using a self-organising neural network, the Growing Neural Gas (GNG), which is used to preserve the topology of any input space. Using GNG, the topological relations of a given set of deformable shapes can be learned. We describe how shape models can be built automatically by posing the correspondence problem on the behaviour of self-organising networks that are capable of adapting their topology to an input manifold, and due to their dynamic character to readapt it to the shape of the objects. Results are given for the training set of hand outlines, showing that the proposed method preserves accurate models.

1 Introduction

The construction of a detailed model from complex objects such as the human hands is problematic because of the large number of correspondences that need to be made to capture the shape variations. The correspondences can be achieved by generating a set of landmarks manually or automatically. The manual correspondence is both laborious and subjective especially when applied to three-dimensional images.

In this paper, we define a new method for automatically labeling the training set based on the theoretic framework of self-organising neural networks. The key insight is that if enough random patterns are used when the algorithm is applied then the network will place the neurons always in relatively closer positions to the object contour. If any point has more probability of being selected then the algorithm will place a neuron in that point. When the best neuron is selected then all the topological neighbours move towards that neuron.

This work opens up a new research interest in computer vision applications, because GNG can be used as a study for the representation of the contour of two-dimensional objects.

2 Automatic Model Building Methods

Various attempts have been made to automatically extract landmark points from a set of training examples.

Baumberg and Hogg [8] describe a system, which generates flexible shapes models from walking pedestrians using automatic landmark extraction. Landmarks are generated by computing the principal axis of the boundary and by generating a number of equally spaced points along the boundary. While the process is satisfactory, the parameterisation of the process is arbitrary and is described only for 2D shapes.

Hicks and Bayer [4] describe a system that automatically extracts landmark features from biological specimens, and is used to build an Active Shape Model (ASM). Their approach is based on identifying shape features such as regions of high curvature that can be used to establish point correspondences with boundary length interpolation between these points. While this method works well for diatom species where the heights and the relative position of the contour curvature local maxima and minima change little, it is unlikely that it will be generally successful for shapes such as hands where there are a lot of variations in the shape.

Angelopoulou and Psarrou [7] use 8-connectivity Freeman chain code boundary descriptor to obtain automatically the coordinate of the boundary pixels and the direction of the boundary. It works well with closed boundaries, but it has not been tested on open boundaries. Also, a reference point is required. Because the diameter of the shape boundary varies, it is not clear that corresponding points will always lie to regions with similar curvatures.

Hill and Cris [5] employ a binary tree of corresponded pairs of shapes to generate landmarks automatically. In order to solve the pairwise correspondence problem they use a polygon-based correspondence algorithm. While the algorithm works well with different classes of objects, it assumes that the objects are represented by closed boundaries.

Davies *et. al*. [2] describe a method for automatically building statistical shape models by using the Minimum Description Length (MDL) principle. The MDL is obtained from information theoretic considerations and the model order is defined as the model that minimises the description length. This is a very promising method for measuring the model quality of a statistical shape, but due to the very large number of function evaluations, this optimisation method is computationally expensive.

3 Statistical Shape Models

A statistical shape model can be built from a training set of examples after labeling and aligning them to a common coordinate frame. The number of shapes from

the training set is denoted as $\{S_i\}$ and n are the landmark coordinate points for each of the $\{S_i\}$ shapes. The vector describing the n landmark points of the $\{S_i\}$ shape in the training set is given as:

$$\mathbf{x}_i = [x_{i_0}, x_{i_1}, x_{i_2}, ...x_{i_{n-1}}, y_{i_0}, y_{i_1}, y_{i_2}, ...y_{i_{n-1}}]^T \quad (1)$$

The data were aligned using the Generalised Procrustes Analysis algorithm [9]. The outcome of the alignment is N aligned shapes $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_N$ with mean shape $\bar{\mathbf{x}}$. The modes of variation are defined by applying the principal component analysis (PCA) to the set. Each example is represented as a shape vector and the mean shape is given as:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \quad (2)$$

The deviation from the mean is given by:

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (3)$$

The covariance matrix of the $2n$ x $2n$ landmark points is:

$$C_{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} d\mathbf{x}_i d\mathbf{x}_i^T \quad (4)$$

The modes of variation can be derived by applying an eigen-decomposition of the Covariance matrix $C_{\mathbf{x}}$ such that:

$$C_{\mathbf{x}}\mathbf{p}_i = \lambda_i \mathbf{p}_i \quad (5)$$

where λ_i ($\lambda_i \geq \lambda_i + 1$) is the ith eigenvalue of $C_{\mathbf{x}}$ and $\mathbf{p}_i$ is the associated i^{th} eigenvector. Most of the variation can be described by a small number of t modes. One method for calculating t is to calculate the sum of the λ_i and choose t such that:

$$a(\lambda_T = \sum_{i=1}^{2n}\lambda_i) \leq \sum_{i=1}^{t}\lambda_i \quad (6)$$

where $0 \leq a \leq 1$ will govern how much of the variation seen in the training set can be represented by a small number of t modes. Any shape in the training set can be approximated using the mean shape and a weighted sum of the principal components from the t modes.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_t \beta_t \quad (7)$$

where

$$\mathbf{P}_t = (\mathbf{p}_1 \mathbf{p}_2 \cdots \mathbf{p}_t) \quad (8)$$

is the matrix of the first t eigenvectors and

$$\beta_t = (\beta_1 \beta_2 ... \beta_t)^{\mathrm{T}} \quad (9)$$

is a vector of weights for each eigenvector. The weight vector describing the best approximation, can be found from:

$$\beta_t = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}) \quad (10)$$

To ensure that that above weight changes describe reasonable variations we restrict the weight $\mathbf{b}_t$ to the range:

$$-3\sqrt{\lambda_i} \leq \beta_i \leq 3\sqrt{\lambda_i} \quad (11)$$

where we see that most of the population is in the order of $3\boldsymbol{\sigma}$ from the mean. Finally, we obtain the data by adding the mean to the original data using the equation below:

$$\beta' = \bar{\mathbf{x}} + (\mathbf{P}^T\mathbf{P}(\mathbf{x} - \bar{\mathbf{x}})) \quad (12)$$

PCA has given the original shapes in terms of their differences and similarities. Since the variations can be performed with the most significant eigenvectors the dimensionality of the data is reduced and the variations are described with fewer variables. PCA works well as long as good correspondences exist. To obtain the correspondences and represent the contour of the hands the self-organising network GNG was used.

4 Topology Learning

4.1 Growing neural gas (GNG)

The Growing Neural Gas (GNG) [1] is an incremental neural model able to learn, as the other self-organizing networks do, the topological relations of a given set of input patterns by means of hebbian learning.

Unlike other methods, the incremental character of this model, avoids the necessity to previously specify the network size. On the contrary, from a minimal network size, a growth process takes place, which continues until an ending condition is fulfilled. Also, learning parameters are constant in time, in contrast to other methods whose learning falls basically in decaying parameters.

4.2 Learning algorithm

We consider a neural network as:

- a set $\mathcal{N}$ of nodes (neurons). Each neuron $c \in \mathcal{N}$ has its associated reference vector $w_c \in \mathcal{R}^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons,
- a set $\mathcal{A}$ of edges (connections) between pairs of neurons. Those connections are not weighted and its purpose is to define the topological structure.

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 $\mathcal{R}^d$.
2. Generate an input signal ξ according to a density function $\mathcal{P}(\xi)$.
3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.
4. Increase the age of all the edges emanating from s_1.
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1:

$$\Delta error(s_1) = \left\| w_{s_1} - \xi \right\|^2 \quad (13)$$

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 distane:

$$\Delta w_{s_1} = \varepsilon_w (\xi - w_{s_1}) \qquad (14)$$

$$\Delta w_{s_n} = \varepsilon_n (\xi - w_{s_n}) \qquad (15)$$

7. If s_1 and s_2 are connected by an edge, set the age of this edge to 0. 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 signals generated, insert a new neuron as follows:
   - Determine the neuron q with the maximum accumulated error.
   - Insert a new neuron r between q and its further neighbour f:

     $$w_r = 0.5(w_q + w_f) \qquad (16)$$

   - Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
   - Decrease the error variables of neurons q and f multiplying them with a constant α. Initialize the error variable of r with the new value of the error variable of q and f.
10. Decrease all error variables by multiplying them with a constant β.
11. If the stopping criterion is not yet achieved, go to step 2.

4.3 Characterising hand posture

Given an image $I(x,y) \in \mathcal{R}$ 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 1 shows the transformation.

Figure 1. Image A represents original image in grey level, in B a threshold is applied that converts to B/W, and in C the contour is obtained

If we consider $\xi = (x,y)$ and $\mathcal{P}(\xi) = \psi_\nabla(\xi)$, we can apply the learning algorithm of the GNG to the image I, so that the network adapts its topology to the contours [3]. Figure 2 shows the adaptation process.

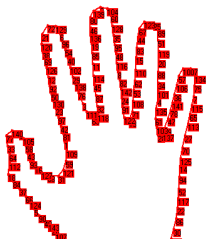

Figure 2. Adaptation process of the GNG

Sometimes in the contour, if the fingers of the hand are too close wrong edges can be created between neurons that are located in different fingers (Figure 3). This problem is both fast and easy to solve by defining a rule to delete the edges drawn onto a part of the input space that does not belong to the contour, or by removing from the list of neurons created in the learning process all the inappropriate cycles produced.

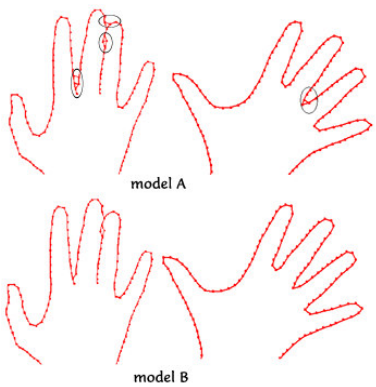

Figure 3. Examples with "wrongly" obtained neurons model A and the applied corrections, model B

4.4 Obtaining the contour and normalise

When the learning process is finished, a set $\mathcal{N}$ of nodes (neurons) and a set $\mathcal{A}$ of edges (connections) between pairs of neurons are obtained representing the contour of the hand preserving its topology [6].

The list of neurons and edges define a graph. To normalise the graph we must define a start point, for example the neuron on the left-bottom corner. Taking that neuron as the first we must follow the neighbours avoiding the cycles until all the neurons had been added to the new list.

If necessary we must apply a scale and a rotation to the list with respect to the centre of gravity of the list of neurons. We achieved the required alignment by applying a transformation T composed by a translation (tx,ty), rotation θ, and a scaling s:

$$T\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{bmatrix} s(\cos\theta)x_i - s(\sin\theta)y_i \\ s(\sin\theta)x_i + s(\cos\theta)y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (17)$$

The results of GNG reordering the neurons and the

normalised neurons can be seen in Figure 4.

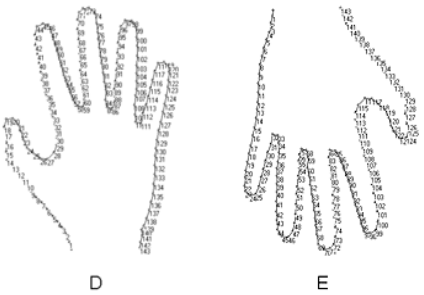

Figure 4. D is the result of GNG reordering the neurons and E shows the list of normalized neurons.

5 Results

In our experiments, 12 hand outlines were segmented from images of different poses. Using GNG we have generated 144 neurons, which represent the landmarks (hence 288 parameters) of each of the image of the training set. Table 1 shows the computational time for a various number of neurons and input patterns.

Table 1. Time spent (in seconds) to adapt the GNG to the contour and reorder the neurons depending on number of neurons and input patterns

Neurons \ Patterns	**1000**	**2000**	**5000**
144	8	9	11
100	5	5	7
64	2	2	3

Experiments were made as well with other self-organizing models as Kohonen maps [10], Neural Gas [11] and the same GNG for obtaining the 2D map but erasing all neurons except those that belong to the contour.

Figure 5 shows the adaptation process using three different topology-preserving networks. The topology preservation of the Kohonen maps in comparison to GNG is very poor. On the contrary, with the NG the topology, preservation is excellent and the contour can be defined very accurately, but the learning time is more than ten times higher than the time for GNG version for contours.

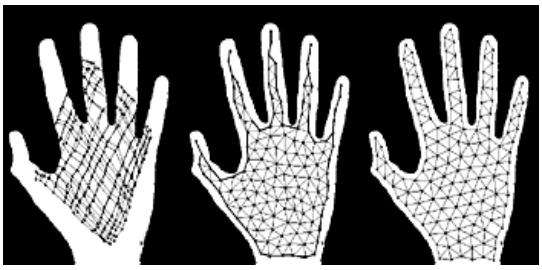

Figure 5. Adaptation process with Kohonen maps, NG and GNG obtaining the contour neurons from the 2D input space of the hand

6 Discussion and Conclusions

This paper presents a new method for automatically extracting landmarks from a training set by using a self-organising neural network (GNG). By establishing a suitable transformation function, the model is able to characterize the shape of the object and adapt its topology to the contour of the object.

In this work the correspondence problem is implicitly solved by the learning process of the network and new areas in shape representation and analysis are opened.

Future work will be aimed at processing contours in unconstrained environments, that is, with coloured or heterogeneous backgrounds and extended it to high level three-dimensional shape variations.

References

[1] Fritzke, B., 1995. A Growing Neural Gas Network Learns Topologies. *In Advances in Neural Information Processing Systems 7*, G. Tesauro, D.S. Touretzky y T.K. Leen (eds.), MIT Press 625-632.

[2] Davies R. *et .al*., 2002. A Minimum Description Length Approach to Statistical Shape Modeling. *IEEE Trans Med. Imaging*, vol. 21, pp. 525-537.

[3] Flórez, F., 2002. Modelo de representación y procesamiento de movimiento para diseño de arquitecturas de tiempo real especializadas. *PhD Thesis*, Universidad de Alicante.

[4] Hicks Y. *et. al*., 2002. Automatic Landmarking for Building Biological Shape Models. *Int. Conf. Image Processing, Rochester*, USA, vol. 2, pp. 801-804.

[5] Hill A. *et. al*., 2000. A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence. *IEEE Trans. Pattern Anal. Machine Intell*., vol. 22, pp. 241-251.

[6] Martinetz, T., Schulten, K., 1994. Topology Representing Networks. *Neural Networks*, 7(3) 507-522.

[7] Angelopoulou A., Psarrou A., 2004. Evaluating statistical shape models for automatic landmark generation a class of human hands. *Int. Arc. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IAPRS, Vol.XXXV, ISSN 1682-1750, Istanbul.

[8] Baumberg A., Hogg D., 1994. Learning Flexible Models from Image Sequences. *Proc. Third European Conf. Computer Vision*, pp. 299-308.

[9] Cootes T., Taylor C., Cooper D., and Graham J, 1995. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1): pp. 38-59.

[10] Kohonen T., 1995. Self-Organizing Maps. Springer-Verlag.

[11] Fritzke B., 1997. Some competitive learning methods. *Tech. rep*., Institute for Neural Computation Ruhr-Universitat Bochum.