Robust modelling and tracking of NonRigid objects using Active-GNG

Anastassia Angelopoulou[1]
Alexandra Psarrou[1]
Gaurav Gupta[1]
José García Rodríguez[2]

[1] Harrow School of Computer Science
[2] Dept of Computer Technology and Computation, University of Alicante

Robust Modelling and Tracking of NonRigid Objects Using *Active*-GNG

Anastassia Angelopoulou[1] Alexandra Psarrou[2] Gaurav Gupta[3]
Harrow School of Computer Science, University of Westminster,
Harrow HA1 3TP, United Kingdom
agelopa@wmin.ac.uk, psarroa@wmin.ac.uk, g.gupta1@student.wmin.ac.uk

José García Rodríguez[4]
Department of Computer Technology and Computation, University of Alicante,
Apdo. 99. 03080 Alicante, Spain,
jgarcia@dtic.ua.es

Abstract

This paper presents a robust approach to nonrigid modelling and tracking. The contour of the object is described by an active growing neural gas (A-GNG) network which allows the model to re-deform locally. The approach is novel in that the nodes of the network are described by their geometrical position, the underlying local feature structure of the image, and the distance vector between the modal image and any successive images. A second contribution is the correspondence of the nodes which is measured through the calculation of the topographic product, a topology preserving objective function which quantifies the neighbourhood preservation before and after the mapping. As a result, we can achieve the automatic modelling and tracking of objects without using any annotated training sets. Experimental results have shown the superiority of our proposed method over the original growing neural gas (GNG) network.

1. Introduction

The need for nonrigid shape modelling and tracking occurs in many real world applications. Tasks like registering human brain MRI images in brain mapping, tracking objects in motion, smooth key frame matching in digital animation and $3D$ morphing in virtual reality applications, all require a robust model not prone to noise that can solve for correct correspondences between a set of shapes. However, because of the complexity of non-rigid transformation/mapping, most methods simplify the task and either equally space the point sets along the shape [3] or group the points into higher level structures such as lines, curves or surfaces and parameterise the points along these attributes [15, 14, 4]. An optimal transformation/mapping such as estimating the mean, the covariance or the probability distribution between rigid or non-rigid objects is then achieved. The accuracy of the mapping is assessed by minimising an objective function either over a pair of shapes or along the shape space [11, 7]. The more global the objective function is the better the quality of the built model.

An alternative method to the equally and one-to-one or many-to-one correspondences is to bypass the correspondence problem. With this method the shapes are modeled either linearly by solving a linear optimisation problem such as the least-square problem or nonlinearly by configuring correspondences in non-linear manifolds. The correspondence problem is modeled either as a probability density estimation problem or as an unsupervised classification problem. In the former the probability distributions can be Dirac Delta functions represented as isotropic or oriented Gaussian mixtures [16, 5] and in the latter vectors of a network with or without topological constrains [13]. The objective of unsupervised classification is: given a high dimensional data distribution find a topological structure that best defines the topology of the original distribution.

In this paper the nonrigid tracking and unsupervised model generation, is addressed as a topology learning problem [9, 1]. The contour of the object is described by adding an *active* step to the GNG network which allows the model to re-deform locally, and update its position. The *Active*-GNG takes into consideration not only the geometrical position of the nodes (landmark points) but also the underlying local feature structure of the image, and the distance vector between the modal image and any successive images. To measure the quality of our model we use the topographic product, our objective function, which quantifies the neighbourhood preservation of the map by computing the distance between neighbouring nodes in both the input and the latent space. The advantage of this representation is that the

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

similarity of a pair of nodes before and after the mapping can be calculated. These features (e.g. topographic product, local grey-level and distance vector) of *Active*-GNG allow us to automatically model and track in an unsupervised manner $2D$ hand gestures in a sequence of k frames. The algorithm is computationally inexpensive, can handle multiple open/closed boundaries and can easily be extend to $3D$.

The remaining of the paper is organised as follows. Section 2 describes how we extract landmark points by introducing the main parts of the GNG algorithm. Section 3 introduces the *Active* step of the GNG algorithm and shows how these modifications can be used in local searching and tracking of the landmark points. Section 4 introduces the topographic product, our objective function to quantify the neighbourhood preservation. A set of experimental results are presented in Section 5, before we conclude in Section 6.

2. Landmark Points and Topology Map

One way of selecting landmark points along the contour of shapes is to use a topographic mapping where a low dimensional map is fitted to the high dimensional manifold of the contour, whilst preserving the topographic structure of the data. A common way to achieve this is by using self-organised networks where input patterns are projected onto a network of nodes such that similar patterns are projected onto nodes 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.

For the automatic extraction and correspondence of landmark points we use the GNG network introduced by Fritzke [8]. GNG allows us to extract in an autonomous way the contour of any object as a set of edges that belong to a single polygon and form a topology preserving map (Figure 1). The evaluation of the correspondences is performed by using the topographic product. Since we want the network to converge either globally or locally, we introduce an *active* step to the network which allows the network to re-deform only where differences between successive images exist. Due to the characteristics of the network such as topology preservation, adaptability and growth, and autonomous operation, the *Active* GNG is used to track hands in a sequence of k frames while keeping correct correspondences between successive frames. In addition the *Active* GNG can be used for the unsupervised generation of models from a set of object instances.

2.1. Growing Neural Gas (GNG)

In this subsection, we review GNG and highlight the main parts of the algorithm as used in this work. In GNG, nodes in the network compete for determining the set of nodes with the highest similarity to the input distribution $\mathbb{R}$ [9]. In our case the input distribution is a finite set of all the points along the contour of a shape. The highest similarity reflects which node together with its topological neighbours are nearest to the input sample point w_i which is the signal generated by the network. The n-dimensional input signals w_i are randomly generated from a finite input distribution:

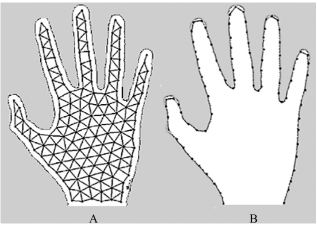

Figure 1. Examples of the two most common topologies. Image A represents the topology preserving map (TPM) of a triangular grid while image B the topology of a line. In both cases the $TPM = \langle N, A \rangle$ is defined by a set of nodes N and a set of edges A that connect them.

$$\vec{W} = \{w_1, ..., w_m\}, \forall w_i \in \mathbb{R} \tag{1}$$

The nodes move towards the input distribution by adapting their position to the input's geometry (Figure 2). During the learning process local error measures are gathered to determine where to insert new nodes. New nodes are inserted near the node with the highest accumulated error. At each adaptation step a connection between the winner and its topological neighbours is created as dictated by the competitive hebbian learning method. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology based on the topographic product [10].

The network is specified as:

- A set N of cluster centres known as nodes. Each node $c \in N$ has its associated reference vector $y_c \in \mathbb{R}$. The reference vectors can be regarded as positions in the input space of their corresponding nodes. Given N reference vectors $\{\vec{y}_c\}_{c=1}^{N} \subseteq \mathbb{R}$ drawn from the random vector $\vec{W}$, we want to find a mapping $G : \mathbb{R} \longrightarrow \mathbb{R}^{\wp}$ and its inverse $F : \mathbb{R}^{\wp} \longrightarrow \mathbb{R}$ such that $\forall c = 1, ..., N$,

$$f(\vec{x}) = E_{\vec{W}|g(\vec{W})}\{\vec{W}|g(\vec{W}) = \vec{x}\}, \forall x \in N \subseteq \mathbb{R}^{\wp} \tag{2}$$

$$g(\vec{W}) = \arg \min_{\mu \in \{\vec{x}_c\}_{c=1}^{N}} \|\vec{W} - f(\mu)\| \tag{3}$$

where $\{\vec{x}_c\}_{c=1}^{N} \subseteq \mathbb{R}^{\wp}$ are the reduced reference vectors drawn from the random vector $\vec{W}$, E is the distance operator and $g(\vec{W})$ is the projection operator. Equations (2) and (3) show that while the forward mapping G is approximated as a projection operator, the reverse mapping F is nonparametric and depends on the unknown latent variable $\vec{x}$. In order to compute $f(x)$ the GNG algorithm evaluates (2) and (3) in an iterative manner. $\wp$ denotes the dimensionality of the latent space. In this work, current experiments include topologies of a line which is the contour of the object ($\wp = 1$) and triangular grid which is the topology preserving graph ($\wp = 2$).

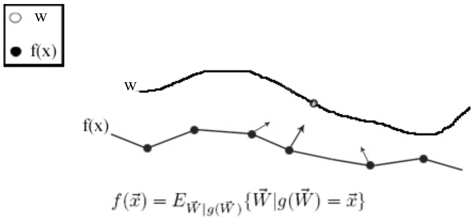
○ w
● f(x)
w
f(x)
$f(\vec{x}) = E_{\vec{W}|g(\vec{W})}\{\vec{W}|g(\vec{W}) = \vec{x}\}$

Figure 2. Every sample point w on the target space is defined as the best matching of all nodes x projecting within a topological neighbourhood of w. For example, the best matching node denoted by the largest arrow, moves towards the sample point while its topological neighbors adjust their position.

- A set A of edges (connections) between pair of nodes. These connections are not weighted and its purpose is to define the topological structure. The edges are determined using the competitive hebbian learning method. The updating rule of the algorithm is expressed as:

$$\Delta\vec{x}_{s_1} = \epsilon_x(w_i - \vec{x}_{s_1}), \Delta\vec{x}_i = \epsilon_n(w_i - \vec{x}_i) \; (\forall i \in N_{s_1}) \tag{4}$$

  where ϵ_x and ϵ_n represent the constant learning rates for the winner node $\vec{x}_{s_1}$ and its topological neighbours $\vec{x}_i$. N_{s_1} is the set of direct topological neighbours of s_1. An *edge aging scheme* is used to remove connections that are invalid due to the activation of the node during the adaptation process. Thus, the network topology is modified by removing edges not being refreshed by a time interval α_{max} and subsequently by removing the nodes connected to these edges.

A complete description of the algorithm can be found in [8].

3. *Active* Growing Neural Gas (*A*-GNG)

When using shape or feature information or combination of the two to track nonrigid objects in video sequences, the most effective models are either 'snakes' introduced by Kass *et al.* [12] or Point Distribution Models (PDMs), Active Shape Models (ASMs) or Active Appearance Models (AAMs) introduced by Cootes and Taylor [6]. In the case of snakes, the deformation of the model to an unseen image is achieved by means of energy minimisation. The snake converges when all the forces achieve an equilibrium state. This dynamic behaviour of the model to minimise its energy function makes the snake *active*. The drawbacks with this method are:

- The snake has no *a priori* knowledge of the domain which means it can deform to match any contour. This attribute is not desirable if we want to keep the specificity of the model which means that the model should deform *only* in ways characteristic of the class of objects it represents.

- The *active* step is performed globally even if parts of the snake have already converged. There is no mechanism in the model to re-deform locally and minimise its energy function only at desirable image properties.

In PDMs, the deformation of the model to an unseen image is specific since *a priori* knowledge such as expected size, shape and appearance is encoded in the model from a training set of correctly annotated images. The ASMs and AAMs have proven to be very powerful tools for interpreting new images. However, as with the snakes the deformation of the model adheres only to global shape transformations.

Since we want the network to converge either globally or locally, we introduce here a nonparametric approach to modelling the objects which makes it ideally suited for learning in dynamic environments. Our model is a modification to the GNG network introduced by Fritzke [8], called *Active* Growing Neural Gas (*A*-GNG) that has the characteristics of a snake, no *a priori* knowledge of the domain and global properties, but is extended in three ways:

1. The correspondence of the nodes is performed locally, so the model re-deforms only where differences in the input space between successive images exist (Figure 3). Therefore, the *active* step is performed locally in contrast to the global properties applied to the image by the snake.

2. The mean vector of the map and of any successive image is calculated and the nodes update their position based on this mean difference. By doing this the map first updates its position into the successive image and then examines a region of the image around each node to determine a better displacement of the node.

3. In order to improve efficiency, we restrict the nodes to their corresponding place by adding a second dimension to the network with information about the local feature structure of the image (Figure 4).

Figure 3 and 4 show the local adaptation of the network and the best matching node denoted by the distance and the underlying feature structure.

The main steps of the *A*-GNG algorithm are as follows:

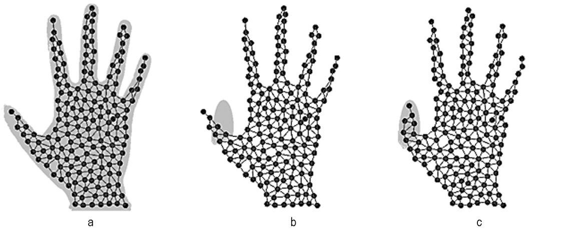

Figure 3. Example of $2D$ local adaptation of the network. Signals are generated only to the new input distribution, Image (b), and the winner node and its direct topological neighbours update their positions, Image (c).

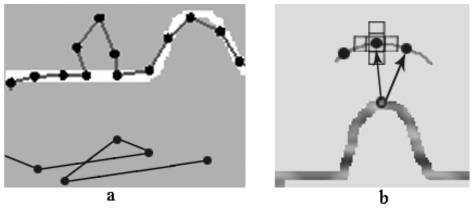

Figure 4. The upper part of Image a shows the convergence of the algorithm to a local minimum. The top node with its direct neighbours can never be winners. The lower part of Image a shows the fold-over that will occur after a number of iterations. Not only point correspondences are lost but also topology relations are violated. To overcome this problem for each node we compute a $2k + 1$ dimensional feature vector which encapsulates grey-level information. Thus, the node with the best feature vector times distance measure will be the winner node. Image b shows the feature vector $2k + 1$ added to each node.

1. Start with a modal image and run the original GNG algorithm.

2. For every node sample k neighbourhood pixels. Thus, we have $2k + 1$ grey-level values which can be put in a vector $\vec{g_i}$.

   $$\vec{g_i} = [g_1, ..., g_{2k+1}]^T \tag{5}$$

   The total shape then is given as:

   $$S_m = \vec{g_i}^T * \vec{x} \tag{6}$$

   where $\vec{x}$ is a $2n$ $\{x_i, y_i\}$ node vector and $\vec{g_i}$ is a $2k + 1$ local feature vector.

3. Given N number of nodes calculate the mean node $\vec{\overline{x_c}}$ of the modal image, where

   $$\vec{\overline{x_c}} = \frac{1}{N} \sum_{i=1}^{N} \vec{x_i} \tag{7}$$

4. Calculate the image difference between the modal image and any other successive image. Let A and B be two sets of elements in $\mathbb{R}$ representing the modal and the successive image respectively. The Minkowski subtraction of B from A is defined as:

   $$A - B = \bigcap_{b \in B} A_b \tag{8}$$

   where the elements b are the pixel coordinates of the successive image.

5. Let $C = A - B$ be the new input distribution of the network.

6. Randomly generate input signals w_i to C and calculate as in step 3 the mean signal $\overline{w_i}$.

7. Calculate the distance vector of the two means and swift the nodes towards C. For each successive image we calculate its deviation from the mean, $d\vec{x_i}$ where

   $$d\vec{x_i} = \vec{x_i} - \vec{\overline{x_c}} \tag{9}$$

8. Randomly generate input signals w_i to C and find the winner node x_{s_1} and its direct topological neighbours x_i.

9. Update the position of the nodes by moving them towards the current signal by the weighted factors ϵ_x, and ϵ_n same as in GNG.

10. Remove the used signal w_i from the input distribution.

11. Repeat iterations $3 - 10$ until the system converges.

The parameters used in all simulations are: $W = 3000$, $N = 150$, $\epsilon_x = 0.1$, $\epsilon_n = 0.005$, $\alpha_{max} = 125$, $k = 5$.

4. Topographic Product

In order to establish correspondence of nodes between successive frames or object instances we use a topology preservation measurement whose attributes derive from nonlinear dynamics. The topographic product P introduced by Bauer and Pawelzik [2] is our objective function which quantifies the neighbourhood preservation of the map by computing the Euclidean distance between neighbouring nodes, in both the input and the latent space. A mapping preserves neighbourhood relations if and only if nearby points in the input space remain close in the latent space. In other words, there is no violation to the topology of the network.

The neighbourhood relationship between each pair of nodes in the latent space $\wp$ and its associative reference vectors in the input space d is given by:

$$P_1(c, k) = [\prod_{l=1}^{k} \frac{d^{\wp}(c, n_l^{\wp}(c))}{d^{\wp}(c, n_l^{d}(c))}]^{1/l} \tag{10}$$

$$P_2(c,k) = [\prod_{l=1}^{k} \frac{d^d(\vec{x}_c, \vec{x}_{n_l^\wp(c)})}{d^d(\vec{x}_c, \vec{x}_{n_l^d(c)})}]^{1/l} \qquad (11)$$

where c is a node, $\vec{x}_c$ is its reference vector, n_l^d is the l-th closest neighbour to c in the input space d according to a distance d^d and $n_l^\wp$ is the l-th nearest node to c in the latent space $\wp$ according to a distance $d^\wp$. Combining (10) and (11) a measure of the topological relationship between the node c and its k closest nodes is obtained:

$$P_3(c,k) = [\prod_{l=1}^{k} \frac{d^d(\vec{x}_c, \vec{x}_{n_l^\wp(c)})}{d^d(\vec{x}_c, \vec{x}_{n_l^d(c)})} \cdot \frac{d^\wp(c, n_l^\wp(c))}{d^\wp(c, n_l^d(c))}]^{1/2k} \qquad (12)$$

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

$$P = \frac{1}{N(N-1)} \sum_{c=1}^{N} \sum_{k=1}^{N-1} \log(P_3(c,k)) \qquad (13)$$

Figure 5 shows an example of a well preserved line topology mapping between two successive frames, where the network has grown sufficiently to reflect the dimensionality of the input distribution. As the input distribution moves the topological relations are updated and correct correspondences are established. A violation of the topology occurs in Figure 6(a) since the distance relations of the data points do not correlate with that of the reference vectors in the network. Figure 6(b) shows the ideal correlation if correct correspondences have been previously established. The problem with the topographic product in cases like in Figure 6(a) is its limitation to take into account the structure of the input distribution since the map of the data points and the reference vectors is one-to-one. In order to overcome this problem where neighbourhood relations are based only on distance measures and not on topological relations, e.g. common borders of Voronoi cells, in every iteration step we update the position of the map towards the image according to the mean vector.

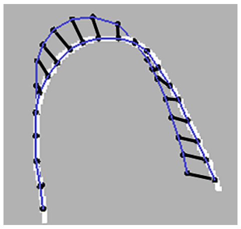

Figure 5. Neighbourhood relations are perfectly preserved since nearby points in the input space remain close to the nearby nodes in the latent space. The mapping is indicated by the lines.

5. Experiments

We demonstrate the performance of our system on a sequence of hand gestures. In order to address the limitations of the existing GNG, and how these have been improved with the A-GNG, we use a combination of artificial and real data sets. In the following experiments we see the superiority of A-GNG against: a) the methodology of active snake model that adheres only to global shape transformations and b) GNG which has only global properties and cannot preserve correspondences when used for tracking. In order to use a clean edge map that serves as the distribution for the algorithm, we have performed the gestures in front of uniform backgrounds avoiding cluttered backgrounds with changes in the illumination. In the following experiments, the parameters in the A-GNG are as follows: $W = 3000$, $N = 150$, $\epsilon_x = 0.1$, $\epsilon_n = 0.005$, $\alpha_{max} = 125$, $k = 5$. Figure 7 and 8 show the tracking of a hand gesture using the A-GNG tracker, and how it outperforms the GNG tracker used in Figure 9.

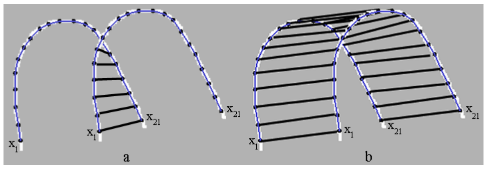

Figure 6. A set of nodes with their reference vectors $\vec{x}_1$, $\vec{x}_2$, up to $\vec{x}_{21}$. As the input distribution moves and the network re-adapts, the distance relations between the data points and the reference vectors are violated (Image a). In the new adaptation the nearest neighbour of $\vec{x}_1$ with its topological neighbours is not $\vec{x}_1$ but $\vec{x}_{21}$. Image b shows correct correspondences if topological information such as closest Voronoi regions and not only metric information has been used.

Figure 7(a) shows the initial A-GNG position. The contour of the first image was extracted using the original GNG and the adaptation of the network at every 10th frame is done with the A-GNG. Images (b) to (i) show the tracking

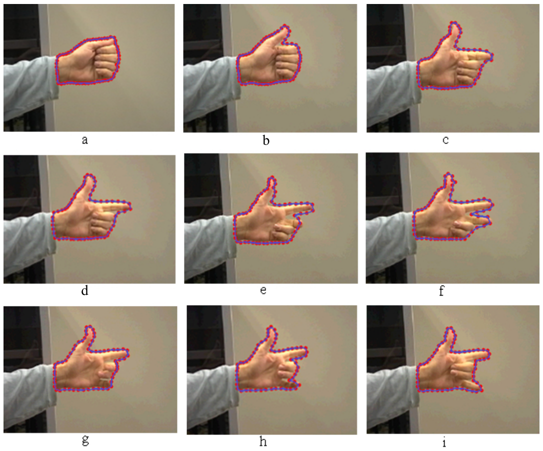

Figure 7. Tracking a gesture. The images correspond from left to right and from top to bottom to every 10th frame of a 90 frame sequence. In each image the red points indicate the nodes and their adaptation after 4 iterations.

of the nodes to a sequence of 90 frames. Our tracker is able to track the fingers and updates the topology of the network every 4 iterations. Figure 8 indicates another tracking example to a sequence of 45 frames. Figure 8(a) shows the initial position of A-GNG. Images (b) and (c) show the adaptation after 1 iteration to a very subtle movement. Images (d), (f) and (h) show the updated position of the network to a more jumping distribution and how the network re-adapts again after 1 iteration.

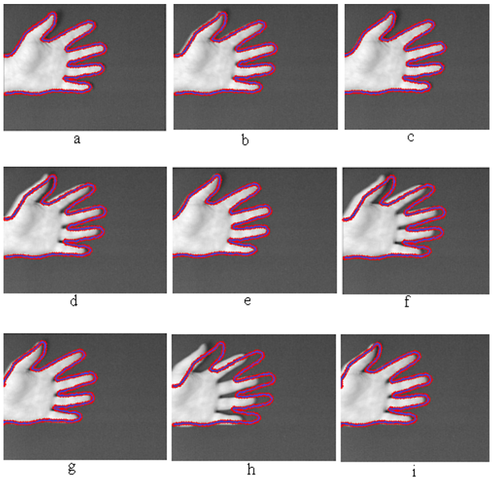

Figure 8. Tracking a hand. The images correspond from left to right and from top to bottom to every 5th frame of a 45 frame sequence.

Figure 9 shows a tracking example using the original GNG. Image (a) shows the initial GNG position and images (b) to (i) show the GNG tracker to a sequence of 120 frames. With the GNG tracker we see that the network is quite far from the real boundaries of the hand and the network is not converging. The top nodes will never be winners and the network collapses to local minima.

Figure 10 is an example of local adaptation of the network between a bump model and a square and how correspondences are improved using the A-GNG compared to original GNG. Image (a) and (b) show the map of the bump model and its superposition to the new image. Image (c) shows the mapping of the GNG based only on distance measures. The network fails to converge since the top nodes can never be winners. The network converges to a local minimum and after a number of iterations a fold-over to the network will occur. Image (d) and (e) show how the convergence is improved by calculating the mean vector of the map and the new image, and then updating the position of the original map according to this difference. The correspondence is improved but still it will take a number of iterations before the top nodes converge unless feature information is incorporated in every node. Figure 11 indicates how feature information can add efficiency to the convergence. Image (a) and (b) show the map and the movement

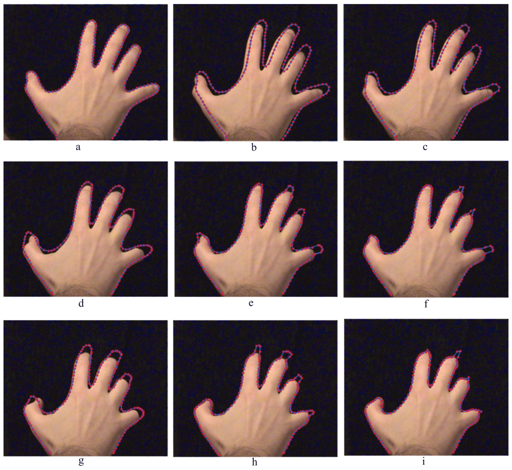

Figure 9. Tracking a gesture. The images correspond from left to right and from top to bottom to every 10th frame of a 120 frame sequence. In each image the red points indicate the nodes and their adaptation after 12 iterations. The GNG tracker is quite far form the hand boundaries and the nodes collapse to local minimum.

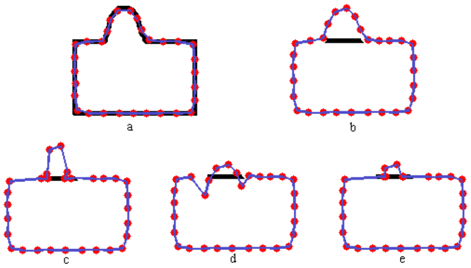

Figure 10. Local convergence using the A-GNG. c shows the map adaptation using the GNG algorithm. d and e show the adaptation using the A-GNG algorithm.

of the finger. Image (c) shows the GNG adaptation and the violation of the map based only on distance measures while Image (d) and (e) show the correct correspondences based on the mean and the feature information added to the network.

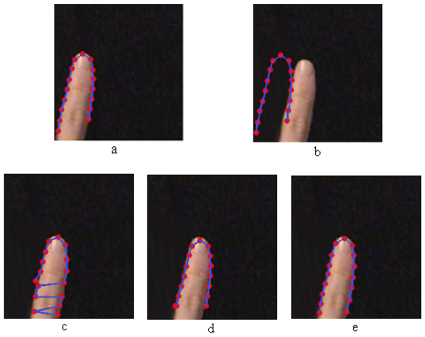

Figure 11. Convergence with and without the *active* steps of the GNG algorithm.

Table 1 shows the topographic product between input

Topographic Product	finger model	bump model
original map	0.049377	0.016662
A-GNG	0.043116	0.036559
GNG	-0.303199	-0.540280

Table 1. Measuring neighbourhood preservation by calculating the map difference between neighbouring nodes, in both the input and the latent space and between successive frames.

and latent space for both the bump and the finger model, and between any successive frames using the GNG and the A-GNG. The topographic product $P \approx 0$ indicates an approximate match while $P < 0$ and $P > 0$ correspond to a too high and a too low match. The first row indicates a match between the input space and the latent space for both the finger and the bump model. The mapping is preserved since nearby points in the input space remain close in the latent space by computing the Euclidean distance between neighbouring nodes. The second and third rows of table 1 show that A-GNG outperforms GNG and correct correspondences are established only when the map is close enough to the new input distribution.

6. Conclusions

We have introduced a new nonrigid tracking and unsupervised modelling approach based on a model similar to snake, but with both global and local properties of the image domain. Due to the number of features A-GNG uses, the topological relations are preserved and nodes correspondences are retained between tracked configurations. The proposed approach is robust to object transformations, and can prevent fold-overs of the network. No background modelling is required. The model is learned automatically by tracking the nodes and evaluating their position over a sequence of k frames. No training set is required and the user interaction is only necessary at initialisation. The algorithm is computationally inexpensive, it can handle multiple open/closed boundaries, and it can easily be extend to $3D$. Our method suffers from some limitations that we are trying to overcome in our future work:

- Our current system uses only absolute grey-level values, because of that if the feature match is below a particular threshold no updates to the position of the nodes is performed, so the network stays inactive.
- A clean edge map is required to serve as the distribution for the algorithm.

Currently we are extending our model by approximating a probability measure on the nodes based on previous and current position, colour information and neighbourhood relations so that A-GNG can be tested on more cluttered backgrounds.

References

[1] A. Angelopoulou, J. García, and A. Psarrou. Learning $2d$ hand shapes using the topology preservation model gng. *In Proc. of the 9^{th} European Conference on Computer Vision, ECCV 2006, LNCS 3951*, pages 313–324, 2006.

[2] H. Bauer and K. Pawelzik. Quantifying the neighbourhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, 1992.

[3] A. Baumberg and D. Hogg. Learning flexible models from image sequences. *In Proc. of the 3^{rd} European Conference on Computer Vision*, 1:299–308, 1994.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[5] K. Chang and J. Ghosh. A unified model for probabilistic principal surfaces. *IEEE Trans. on Pattern Aanalysis and Machine Intelligence*, 23(1):22–41, 2001.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and G. J. Active shape models - their training and application. *Comp. Vision Image Underst.*, 61(1):38–59, 1995.

[7] H. R. Davies, J. C. Twining, F. T. Cootes, C. J. Waterton, and J. C. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Transaction on Medical Imaging*, 21(5):525–537, 2002.

[8] B. Fritzke. A growing neural gas network learns topologies. *In Advances in Neural Information Processing Systems 7 (NIPS'94)*, pages 625–632, 1995.

[9] S. Furao and O. Hasegawa. An incremental network for online unsupervised classification and topology learning. *The Journal of Neural Networks*, 19(1):90–106, 2005.

[10] J. Geoffrey, F. Goodhill, and J. Terrence. A unifying measure for neighbourhood preservation in topographic mappings. *In Proc. of the 2^{nd} Joint Symposium on Neural Computation*, 5:191–202, 1997.

[11] A. Hill, C. Taylor, and A. Brett. A framework for automatic landmark identification using a new method of nonrigid correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):241–251, 2000.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *In Proc. of the 1^{st} Internationl Conference on Computer Vision, IEEE Computer Society Press*, pages 259–268, 1987.

[13] T. Martinez and K. Schulten. Topology representing networks. *The Journal of Neural Networks*, 7(3):507–522, 1994.

[14] A. Rangarajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. *In Proc. of the 15^{th} Conference on Information Processing in Medical Imaging*, pages 29–42, 1997.

[15] S. Scalroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. on Pattern Aanalysis and Machine Intelligence*, 17(6):545–561, 1995.

[16] F. Wang, J. B. Vemuri, and A. Rangarajan. Simultaneous nonrigid registration of multiple point sets and atlas construction. *In Proc. of the 9^{th} European Conference on Computer Vision, ECCV 2006, LNCS 3953*, pages 551–563, 2006.