# Modelling and tracking objects with a topology preserving self-organising neural network

**Anastassia Angelopoulou**

School of Electronics and Computer Science

# Modelling and Tracking Objects with a Topology Preserving Self-organising Neural Network

**Anastassia Angelopoulou**

1. Supervisor: Dr. Alexandra Psarrou
2. Supervisor: Prof. Vladimir Getov

A Thesis submitted in partial fulfilment of the
requirements of the University of Westminster
for the degree of Doctor of Philosophy

School of Electronics and Computer Science
June 2011

*"Basic research is like shooting an arrow into the air and, where it lands, painting a target"*

Homer Burton Adkins (American chemist, 1892-1949)

# Affirmation

Herewith I declare that the work submitted is my own. Appropriate credit has been given to thoughts that were taken directly or indirectly from other sources.

Anastassia Angelopoulou

This thesis was created at the School of Electronics and Computer Science (ECS) at University of Westminster between 2002 and 2010. During this time this work was supported internally by the Computer Vision and Imaging Research Group (CVIR).

# Acknowledgements

This thesis would not have been possible to materialise without the support and guidance of various people. I would like to thank all these people who stood by me and believed in me and have influenced with their invaluable source of ideas, support, and guidance this research for so many years. I am deeply grateful to my family and my beloved grandparents for their support, to my partner for his encouragement and passion to carry on, to my supervisors Dr. Alexandra Psarrou and Prof. Vladimir Getov for their support, guidance, academic challenge and constructive criticism, and to my colleagues, friends and researchers Dr. José García Rodríguez, Dr. Gaurav Gupta, Markos Mentzelopoulos, Dr. Peter.M.Roth and Dr. Michael Walter for their crazy ideas, and the time and effort they spent to proof read this thesis. I would also like to thank Prof. Shaogang Gong for enriching my research experience. Finally, to my friends in UK, Greece and beyond for the endless coffees and discussions. I would also like to thank the University of Westminster and especially the School of Electronics and Computer Science for giving me the opportunity to attend various excellent conferences and workshops.

*Dedicated to my family and Mouchette*

# Abstract

Human gestures form an integral part in our everyday communication. We use gestures not only to reinforce meaning, but also to describe the shape of objects, to play games, and to communicate in noisy environments. Vision systems that exploit gestures are often limited by inaccuracies inherent in handcrafted models. These models are generated from a collection of training examples which requires segmentation and alignment. Segmentation in gesture recognition typically involves manual intervention, a time consuming process that is feasible only for a limited set of gestures. Ideally gesture models should be automatically acquired via a learning scheme that enables the acquisition of detailed behavioural knowledge only from topological and temporal observation.

The research described in this thesis is motivated by a desire to provide a framework for the unsupervised acquisition and tracking of gesture models. In any learning framework, the initialisation of the shapes is very crucial. Hence, it would be beneficial to have a robust model not prone to noise that can automatically correspond the set of shapes. In the first part of this thesis, we develop a framework for building statistical $2D$ shape models by extracting, labelling and corresponding landmark points using only topological relations derived from competitive hebbian learning. The method is based on the assumption that correspondences can be addressed as an unsupervised classification problem where landmark points are the cluster centres (nodes) in a high-dimensional vector space. The approach is novel in that the network can be used in cases where the topological structure of the input pattern is not known *a priori* thus no topology of fixed dimensionality is imposed onto the network.

In the second part, we propose an approach to minimise the user intervention in the adaptation process, which requires to specify *a priori* the number of nodes needed to represent an object, by utilising an automatic criterion for maximum node growth. Furthermore, this model is used to represent motion in image sequences by initialising a suitable segmentation that separates the object of interest from the background. The segmentation system takes into consideration some illumination tolerance, images as inputs from ordinary cameras and webcams, some low to medium cluttered background avoiding extremely cluttered backgrounds, and that the objects are at close range from the camera.

In the final part, we extend the framework for the automatic modelling and unsupervised tracking of $2D$ hand gestures in a sequence of $k$ frames. The aim is to use the tracked frames as training examples in order to build the model and maintain correspondences. To do that we add an *active* step to the Growing Neural Gas (GNG) network, which we call *Active* Growing Neural Gas (*A*-GNG) that takes into consideration not only the geometrical position of the nodes, but also the underlined local feature structure of the image, and the distance vector between successive images. The quality of our model is measured through the calculation of the topographic product. The topographic product is our topology preserving measure which quantifies the neighbourhood preservation.

In our system we have applied specific restrictions in the velocity and the appearance of the gestures to simplify the difficulty of the motion analysis in the gesture representation. The proposed framework has been validated on applications related to sign language. The work has great potential in Virtual Reality (VR) applications where the learning and the representation of gestures becomes natural without the need of expensive wear cable sensors.

# Abbreviations

| | |
|---|---|
| GNG | Growing Neural Gas |
| A-GNG | Active Growing Neural Gas |
| VR | Virtual Reality |
| PDMs | Point Distribution Models |
| Snakes | Active Contour Models |
| SOMs | Self-Organising Maps |
| GCS | Growing Cell Structures |
| NG | Neural Gas |
| FAM | fuzzy ARTMAP |
| NN | Neural Networks |
| HMMs | Hidden Markov Models |
| DOF | Degrees of Freedom |
| SVD | Singular Value Decomposition |
| FEM | Finite Element Model |
| CPD | Critical Point Detection |
| GA | Genetic Algorithm |
| MDL | Minimum Description Length |
| TPS-RPM | thin plate spline robust point matching |
| ICP | Iterative Closest Point |
| JS | Jensen-Shannon |
| GNG-U | Growing Neural Gas with Utility |
| RBF | Radial basis function |

| | |
|---|---|
| GWR | Grow When Required |
| ART | Adaptive Resonance Theory |
| IGNG | Incremental GNG |
| RGNG | Robust GNG |
| EM | Expectation Maximisation |
| MRI | Magnetic Resonance Imaging |
| MGNG | modified Growing Neural Gas |
| ANNs | Artificial Neural Networks |
| CHL | Competitive Hebbian Learning |
| ASMs | Active Shape Models |
| AAMs | Active Appearance Models |
| PCA | Principal Component Analysis |
| KLT | Karhunen-Love transformation |
| RMSE | root-mean-square error |
| SG | Single Gaussian |
| MGM | Mixture of Gaussian models |
| TP | Topographic Product |
| MSE | Mean Squared Error |
| IQE | Inverse Quantisation Error |
| GTP | Geodesic Topographic Product |
| FSONN | Fully Self-Organising Artificial Neural Network Models |
| MOE | Mixture-of-Experts |
| VQ | Vector Quantisation |

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

> *"Whereas verbal behavior is assumed to be closely tied to a speaker's thoughts, non verbal behavior, including gesture, has traditionally been assumed to reflect the speaker's feelings or emotions."*
>
> *W. Wundt*

## 1.1 Motivation

Accurate nonrigid shape modelling and tracking is a challenging problem in machine vision with applications in human computer interaction, motion capture, nonlinear registration, image interpretation and scene understanding. In the area of human computer interaction, the last decades with the availability of more powerful computers and a wide range of camcorders, it only became natural to search for more intriguing and natural interfaces to interact with computers.

Human gestures form an integral part in our verbal and non-verbal communication. We use them to reinforce meaning not always conveyed through speech, to describe the shape of objects, to play games, to communicate in noisy environments, and to convey meaning to elderly people and people with special needs. We can use gestures as expressive body motions or to translate non-verbal languages that consist of a set of well defined gestures and hand postures with complete lexical and grammatical specifications as in the case of sign languages. Our visual system is able to interpret a remarkable variety of different gestures with often subtle characteristics of hand configuration, posture and orientation.

Throughout this thesis we use gestures found in sign languages with underlying spatial and temporal structure defined only by the hand motion. Our system integrates the vision module, and not high calibre wear cable sensors attached to the users, and can operate on a low to medium cluttered environment. The system receives images from acquisition devices at video frequency and take decisions under a set of requirements such as time constraints, accurate segmentation, and medium processing speed of hand gestures.

Machine vision systems that exploit gestures are often limited by inaccuracies inherent in handcrafted models. These models are generated from a collection of training examples which requires segmentation and alignment. This task is ill-conditioned due to measurement noise, manual intervention and hand labelling of image sequences, and human variation in the performance of a gesture. Ideally models should be automatically acquired via a learning scheme that enables the acquisition of detailed behavioural knowledge only from topological and temporal observation. As such, a robust model not prone to noise that empirically evolves over time using contextual information directly derived from an observation sequence is required. In addition, this model should vary its shape and can segment, match and track images of anatomic structures. In the case of non-rigid shape matching where natural gestures belong to the modelling can be performed using deformable models which can be classified as:

- Statistical models where *a priori* knowledge is incorporated in the model such as expected size, position, shape and appearance [32, 39, 80, 124, 130].

- Flexible models which can deform to any shape, with no *a priori* knowledge about the object domain and with object-specific varying parameters [92, 106, 158, 188].

- Self-Organising Neural Network Models where no restrictions are applied upon the network model [10, 60, 93, 94, 117, 118].

From the statistical models, the most well known models are the Finite Element Models and the Statistical Shape Models. These models share in common the prior knowledge the user has about the object of interest [48]. The finite element models capture the variability of the different objects by incorporating *a priori* knowledge about the expected physical attributes of the object [32, 39]. Nastar and Ayache [124], for example, build models from a prototype represented by a set of nodes attached to springs. By solving a generalised eigenmode problem they derive different modes of vibrations with the first modes used to model $2D$ and $3D$ medical images. The statistical shape models capture the variability of a class of objects by estimating the population statistics from a set of training examples. The training set consists of a set of points along the contour of the object. The collection of training examples requires the manual segmentation and alignment of an observation sequence, which is an ill-conditioned task due to measurement noise and human variation in the observation. However, if the manual intervention, a time consuming and labor intensive process that is only feasible for a limited set of examples, can be replaced by semi-automatic or fully automatic methods, these models are very compact since they deform in ways only similar to the training set [40, 80]. Cootes *et al.* [32], for example, model and segment $2D$ medical images by constructing Point Distribution Models (PDMs) from training sets of $2D$ boundaries. New shapes are generated by solving an eigenshape problem and reconstructed by the principal vectors that best capture the variation of the training set. With these models we build statistical human brain MRI and hand gesture models since are the most prominent in terms of specificity and generalisation.

From the flexible models, the most well known models are the Hand Crafted Models [79, 83, 106], the Fourier Series Shape Models [19, 150], and the Active Contour Models (Snakes) [92]. These models share in common that no *a priori* knowledge is applied, but object-specific parameters are used as constrains for the model deformation. Hand crafted models can be built up from simple subcomponents, such as circles, lines or arcs, which are allowed object-specific deformations [32]. Yuille *et al.* [188], for example, model parts of a face, such as eyes and mouth, by allowing only specific parameters, such as the radius of the iris, to deform. Modifying the trigonometric basis functions from the Fourier Series, shapes are modelled as a function with specific parameters [32]. Staib and Duncan [158], for example, use Fourier models to interpret medical images. By varying the parameters and the number of terms used, different shapes can be generated. These models have no prior shape information and can deform infinitely since no shape constraints are applied. The Active Contour Models (Snakes) were popularised by Kass *et al.* [92] and describe energy minimising spline curves attracted toward features, such as lines and edges. The snake can deform to any smooth contour with few constrains on the overall shape. The constrains are defined by a combination of internal and external forces and the snake converges when the forces achieve equilibrium. Furthermore, the snake has global constrains and it has no mechanism to minimise its energy function at desirable (local) image properties. A comparison of our proposed method to the snake's methodology is presented in Chapter 5.

From the Neural Network models, the most well known Self-Organising Networks are the Self-Organising Maps (SOMs) or Kohonen Maps [93, 94], the Growing Cell Structures (GCS) [59], the Neural Gas (NG) [117] and the Growing Neural Gas (GNG) [60]. These models share in common the attributes of dynamically generating and removing processing elements (vectors of a network) and dynamically generating and removing synaptic links (neighbourhood connections). Furthermore, these models make no assumption about the global structure of the shape to be modelled or more generally of the problem to be learned [9, 162]. From the above models, which are quite similar in the system architecture, we have used the GNG model since is superior in terms of computational efficiency, is robust against noise, and can handle complex distributions [7, 55, 164, 168]. Furthermore, Heinke and Hamker [78] made a comparative study between fuzzy ARTMAP (FAM), GCS and GNG, and found that a well trained GNG outperforms the other incremental networks with respect to the number of inserted nodes, the number of epoches and convergence speed.

In the statistical and flexible models, the segmentation of the training examples typically involves manual intervention and hand labelling of image sequences, a time consuming and labor intensive process that is only feasible for a limited set of gestures [80]. Moreover, most of the common tracking schemes require a good representation of the posterior distribution so that low-degree parametric models can be applied to the observation [11]. This has motivated many researchers to consider nonparametric representations, including particle filters and nonparametric belief propagation [110, 166, 167]. In the case of the nonparametric belief propagation the geometric hand model's configuration like the structural and the kinematic constrains of the hand are considered. In this thesis, the tracking is based only on the given $2D$ image observation.

Since we want our model to converge both globally and locally, in this thesis we introduce a nonparametric approach to modelling the gestures which makes it ideally suited for learning in dynamic environments. A nonrigid shape modelling framework should be able to ($1$) generalise and capture the natural variability of the objects; ($2$) self-organise to reflect the nonlinear correlations between inputs and outputs, and should have no topological constraints; ($3$) model motion so the object of interest can be tracked in a sequence of frames; ($4$) preserve topological relations based on global and local transformations; ($5$) use as few parameters as possible; ($6$) have a low computational complexity.

In the above framework, it is important that a reliable segmentation system exists that takes into consideration some illumination tolerance, images as inputs from ordinary cameras and webcams, some low to medium cluttered background avoiding extremely cluttered backgrounds, and that the objects are at close range from the camera. In the case of shape variation and modelling where the extracted feature of the object is the contour, a strong segmentation between foreground and background is required since any failure will prevent the successful extraction of the feature. However, when the topology of an object is incorporated into the model the segmentation can be relaxed since the gestures can be recovered from features already saved in the network.

Besides segmentation hand gestures, which are effectively a $2D$ projection of a $3D$ object, can become very complex for any recognition system. Systems that follow a model-based method [1, 126, 168], require an accurate $3D$ model that captures efficiently the hand's high Degrees of Freedom (DOF) articulation and elasticity. The main drawback of this method is that it requires massive calculations which makes it unrealistic for real-time implementation. Since this method is too complicated to implement, the most widespread alternative is the feature-based method [36, 95] where features such as the the geometric properties of the hand can be analysed using either Neural Networks (NN) [171, 184] or stochastic models such as Hidden Markov Models (HMMs) [49, 180]. We decided to use the former for the representation of human gestures since our model should perform at high computational efficiency making it ideal for real time environments, have low quantisation error, and obtain accurately the topology of the hand.

## 1.2 Contributions

The research described in this thesis is motivated by a desire to address the above model building limitations and to provide a framework for the automatic model acquisition and unsupervised tracking of nonrigid objects using topological relations and underline features. In particular the main contributions are:

- We develop a method for the automatic extraction and correspondence of landmark points using only topological relations derived from Competitive Hebbian learning. Correspondences, which are the point-to-point matching between two or more shapes, are the vectors (nodes) of a network without topological constrains and are solved in nonlinear manifolds. The automatic generation of the nodes is performed with the Growing Neural Gas (GNG) network and we show how it can be applied automatically in a set of objects. Furthermore, we have improved its parameters by removing wrong edges and re-ordering the network. The re-ordered list of nodes is then projected into the shape space where synthesised shapes similar to the training set are generated using statistical models.
- Based on the capabilities of GNG to readjust to new input patterns without

restarting the learning process, we propose an approach to minimise the user intervention in specifying the number of nodes needed to represent an object by utilising an automatic criterion for maximum node growth. The termination of the network uses knowledge obtained from information-theoretic considerations. The model is then used to the representation of motion in image sequences by initialising a suitable segmentation that separates the object of interest from the background.

- This approach is extended by adding an *active* step to the GNG network, which we call *Active* Growing Neural Gas (*A*-GNG) that takes into consideration not only the geometrical position of the nodes, but also the underlying local feature structure of the image and the distance vector of the maps between successive images. The network has both global and local properties, and the nodes move at key areas in the image and no training set is required. This extended framework allows us to automatically model and track in an unsupervised manner $2D$ hand gestures in a sequence of $k$ frames.

- Based on the tracked frames provided by *Active*-GNG we measure the validity of the best model, in terms of maintaining correspondences, by computing the distance in successive frames between neighbouring nodes in both the input and the latent space. The advantage of this representation is that the similarity of a pair of nodes before and after the mapping can be calculated, which means that a mapping preserves neighbourhood relations if nearby points in the input space remain close in the latent space. The best neighbourhood preservation is measured by the topology preserving measure, the topographic product. This measure evaluates the similarity of pairs of points before and after the neighbourhood mapping by computing the distance and taking into account the structure of the Delaunay triangulation between neighbours in both the input and the latent space.

## 1.3 Thesis Outline

The remaining part of this thesis is structured as follows. In Chapter 2 we provide a review of related research in the field of solving correspondences across a set of $2D$ shapes. The review focuses on the importance of correspondences in automatic shape modelling and on methods that best approach the correspondence problem. In Chapter 3 we develop a framework for building $2D$ statistical shape models of hand gestures using only topological relations. For validity, we have also experimented with other nonrigid objects such as human brain MRI, which are discussed in Appendix C. We first discuss the Growing Neural Gas (GNG) network and then we show how it can be used for the automatic extraction and correspondence of nodes in a set of objects. The validity of the model, in terms of maintaining correspondences, is measured with a topology preserving function. Based on the capabilities of the GNG network, in Chapter 4 we propose an approach to minimise the user intervention in the adaptation process. We achieve that by defining an optimal number of nodes without overfitting or underfitting the network, based on the knowledge obtained from information theoretic considerations. In Chapter 5 we extend the framework for the automatic modelling and unsupervised tracking of $2D$ hand gestures in a sequence of $k$ frames. The modified network consists of descriptors obtained from a spatial transformation of the network, an automatic criterion for maximum node growth and local features for object tracking. In Chapter 6 we conclude with a formal discussion, an outline of future work, and possible applications like the online tracking and representation of previously unmodelled objects and their incorporation in environments such as augmented reality. This work can also be used in cases where minimum time is required like online learning for detecting obstacles in robotics. In addition, in the Appendices we give an overview of unsupervised learning and the GNG algorithm as used in this thesis, an introduction to shape alignment, experimental results for MRI data sets, an overview of the EM algorithm for skin colour segmentation, and a list of publications that established the basis for this thesis.

# Chapter 2

# Solving the Correspondence Problem Across Sets of $2D$ Shapes

*In this chapter we review related research in shape modelling, a principal approach in many computer vision applications. In particular, we discuss the ideas of constructing optimal models of shape variation, which are fundamentals for the methods proposed in the following chapters. Additionally, we introduce the idea of unsupervised learning and show how the self-organising models can improve shape modelling.*

## 2.1 Shape Modelling

In shape modelling a robust model is a model that can generalise to legal unseen instances of the selected class of shapes. The generalisation is successful if and only if correct correspondences have been established between shapes. One common approach is to hand-annotate the shapes from the training set by placing landmark points around the contours. This process is both laborious, time-consuming and error-prone in $2D$ and $3D$ - especially if the person doing the annotation is non-expert- that an accurate, rapid and automated system should be developed and deployed. In literature, several approaches have been proposed to automate the process of model building [3, 11, 40, 81, 97, 146]. In all cases the aim is to build automatically a model that best captures the shape variation with minimal representation error [30].

In this chapter the methods that best describe the correspondence problem in $2D$ shapes have been grouped accordingly: the equally spaced method [11, 40, 81], the pairwise method [13, 69, 81, 91, 151, 169], the groupwise method [39, 97, 98], the non correspondence [14, 23, 176], and the proposed unsupervised learning method. In the following sections we give a review of related research, and conclude the chapter with our proposed method based on growing neural models.

## 2.2 Equally Spaced Correspondence

Equally spaced is a semi-automatic or automatic method that can be used to construct models by equally spacing control points along the boundaries of the training set. It is a semi-automatic method if it is used in conjunction with a small number of manually placed points at key locations, like points placed manually at the tip of each finger. Researchers have used this method as comparison to their automatically generated models [40, 81]. In this thesis this method was used for comparison and results are presented in Chapter $3$. Since the method uses a number of hand-annotated points in key locations (curvature points) it gives good results but is considered semi-automatic and time consuming. It is an automatic method if it is used by selecting a starting point on each example from the training set and equally space a number of points on each boundary.

Baumberg and Hogg [11] have used this automatic technique to build statistical shape models. At first they obtain a set of un-ordered boundary points from walking pedestrians. The boundary points are re-ordered according to a reference point. The reference point is selected as the lowest point of the principal axis that passes though the centroid of the boundary points, the principal axis is defined by using the least square approximation method. Then, they construct the model by approximating the boundary points to equally spaced B-spline control points around the boundaries of walking pedestrians. The model is then refined by adding direction of motion to the model and is used to extrapolate direction from shape.

This method generally results in poor models because a) the length of each shape differs, b) points do not correspond at physical locations across the set, and c) the points are not allowed to redistribute around the boundaries.

## 2.3 Pairwise Correspondence

Pairwise correspondence is a method that seeks to automate the manual process of identifying similar points on each example from the training set by performing a sequence of point-to-point correspondences between a pair of shapes and by optimising a pairwise metric [97]. The pairwise correspondence can be performed either between a pair of **point sets** [13, 69, 136, 146, 151, 152] not necessarily connected or between two **curvatures** by minimising their difference [29, 47, 81, 91, 169]. The difference between the two is that the former treats all points of equal importance while the later uses as a measure of correspondence points with maximum curvature. However, the problem with pairwise correspondence is generalisability, since the metric is optimised over a pair of shapes rather than over the whole set and therefore global properties of the model are lost. In addition the pairwise metric like curvature-based matching is arbitrary and adapted to the problem at hand.

### 2.3.1 Point Sets

Scott and Longuett-Higgins [151] developed an algorithm of matching $2D$ features such as edges and corners across a pair of shapes using a weighted proximity matrix. The mapping scheme is based on the criteria of favouring matches across shorter distances between a pair of features and favouring only one-to-one matches. This is achieved by performing Singular Value Decomposition (SVD) on the Gaussian-weighted proximity matrix which holds all the possible distances between a pair of features. Only strong correspondences are selected from the proximity matrix and an overall minimum squared distance mapping is ensured. This technique however fails, if the shape undergoes large rotations and distortion such as skewness.

Shapiro and Brady [152] have extended this technique by incorporating in the proximity matrix intra-image feature distances rather than inter-image feature distances; distances between features within the image and not between two images. This matrix and its corresponding eigenvectors form an orthogonal basis that captures the modes of the shape. In other words, shape description is added to the algorithm at a low level. Corresponding points are found by comparing the modes of a pair of shapes which they call it *modal matching*. This algorithm was implemented and the result is that it works well for rigid shapes and for transformations such as rotations, translations and uniform scaling but it becomes unstable when the shape performs nonrigid transformations. This is due to the fact that there is no mechanism behind to allow for redistribution of the points around the boundaries.

Sclaroff and Pentland [146] find correspondences between features of pair of shapes via *modal matching* similar to Shapiro and Brady [152], but the method is extended to nonrigid correspondences and with greater generalisability and accuracy. The algorithm builds a finite element model (FEM) based on the cloud of feature points for each shape. Modal analysis of the FEM produces the eigenmodes or shape modes (rigid-body modes, intermediate modes and high-order modes) for each shape. Correspondences are computed by matching the two sets of modes directly. Hill and Taylor [82] have implemented this algorithm and found that works well on certain shapes (airplanes, cars, etc.) but it fails on deformable shapes such as hands, ventricles and generally medical shapes. The reason is that since no connectivity of the data to form boundaries is enforced the parameterisation of the data points is not restricted to the surface of the object and this does not guarantee legal set of correspondences.

Rangarajan *et al.* [136] use the softassign procrustes matching algorithm to establish correspondences between a pair of point sets. Correspondences are achieved via a binary match matrix that assigns points in one set to points in the other and discarding points that do not match as outliers. An optimisation method similar to solving the assignment problem is used to produce a match matrix of correspondences. The problem with this method is similar to Sclaroff and Pentland [146]. Since there is no notion of boundary connectivity invalid correspondences can be achieved.

Gold *et al.* [69] solve the correspondence problem between a pair of $2D$ and $3D$ point sets by using a combination of optimisation techniques and minimising an objective function which handles both pose and correspondences. The algorithm is a two step iterative algorithm. In the first step the correspondence parameters are estimated using the softassign optimisation technique while in the second step the pose parameters are estimated using coordinate descent. The pose and the correspondence are calculated simultaneously in an iterative manner. The problem with this method is similar to the previous methods. It works well for rigid object with transformations such as scaling, rotation or swearing but not with nonrigid objects since no connectivity is enforced. Furthermore, there is no mechanism behind that allows for redistribution of the points around the boundaries.

### 2.3.2 Curvature Information

Kambhamettu and Goldgof [91] use Gaussian curvature changes of the surface at a given point to estimate point correspondences in nonrigid motion. To determine point correspondences they assume small motion of the surface and hypothesize all the possible correspondences the point of interest can have with its neighbours. Curvature changes are then computed for each hypothesis. Then an error metric is used to evaluate the hypotheses. The hypothesis with the minimum error gives true point correspondences and surface stretching.

Similar to Kambhamettu and Goldgof [91], Cohen *et al.* [29] use curvature information based on the idea that curvature points possess anatomical meaning (for example, on a face curvature points correspond to the nose, chin and eyebrows) to track nonrigid motion of deformable $2D$ shapes. Their method is based on energy minimisation between a pair of curves to be matched and is obtained through the mathematical framework of Finite Element Analysis. The idea is to weight differently the salient regions where points of high curvature exist from other points in the boundaries. Their method is an improvement to Duncan *et al.* [47] method and can generalise to $3D$.

Hill and Taylor [81] use the curvature of the boundary to define correspondence between a pair of closed $2D$ shapes. Their algorithm is a two-stage process. At first a curvature matching dynamic programming algorithm is used to generate a pixel-to-pixel mapping between pair of shapes. Then for each matched pair a mean shape is constructed until there is one generic mean shape. The construction of the mean shape follows the structure of a bottom-up binary tree. A set of landmarks are now generated automatically on the mean shape and a top-down approach is followed. In the second stage an optimisation scheme is performed. The objective function is the trace of the model covariance matrix plus a correction term that penalises points moving outside the boundary. Although the method works well the corresponding metric is arbitrary and the method is not easily extensible to $3D$ [97].

Tagare *et al.* [169] have shown that the above methods based on curvature information are problematic for two reasons:

1. Curvature is a rigid invariant of shape. Rigid invariance means that points separated by a certain distance in one curve are paired with points separated with the same distance in the other curve. If the distance is different then the pairing is nonrigid. Thus, methods that use numerical values as curvature based difference at corresponding points will fail in nonrigid shapes.

2. The curvature based difference objective function is not symmetric since the optimal correspondence from one curve to another is not guaranteed in the reverse direction.

Their method in finding correspondences is based on the notion of taking the product of a pair of curves and creating the relevant product space with its topological structure. These *biomorphism* correspondences follow a pairwise mapping. These correspondences can be one-to-one or many-to-one. With the later allowing curve segments to shrink to points and vice versa. Since the correspondences are based on biomorphism properties can only be defined for closed shapes. An objective function is then used to measure the quality of the correspondences by comparing the local shape and local stretching of the curves. The algorithm gives very good results both to rigid and nonrigid shapes but is not extensible to open shapes since it contradicts the definition of *what is biomorphism* or to $3D$. Furthermore, the authors claim the algorithm may not be relevant for correspondences based on landmarks [169, 170].

In [81] Hill and Taylor used a dynamic programming algorithm based on curvature difference to establish pairwise correspondences between points. They found this method problematic because of the reasons pointed out by Tagare [170]. In [80, 82] they have improved their framework by using a polygonal approximation method between a pair of boundaries without comparing curvature difference. Their algorithm performs in three stages:

- A sparse polygonal approximation to one of the two nonrigid closed boundaries is first performed. The sparse polygon is generated with the critical point detection (CPD) algorithm described by Zhu and Chirlian [191]. CPD is an effective algorithm of detecting points with maximum curvature on boundaries. The algorithm separates the critical points from the rest of the points by applying a critical level to every point and keeping those with the highest level. The critical level is the area of the triangle confined by the given point and its two neighbour points. In a recursive manner the points with the lowest critical level are deleted until no point has lower critical level than the specified level. When this threshold is reached the algorithm converges. This threshold is set up by the user and depends on the level of details of the object. The idea behind the polygonal approximation is to determine the number of ordered pairs between the two shapes.

- Then a back projection based on a path matching correspondence algorithm is performed onto the second boundary resulting on an initial estimate of the corresponding polygon. At this stage a rigid pairing between points is established.
- Finally, these initial set of correspondences are then refined and an optimisation greedy algorithm is used to minimise a cost function.

The algorithm gives good results in nonrigid shapes but it cannot handle multiple open/closed boundaries, the objective function is arbitrary - based on the representation error of two shapes- and computationally expensive to be extended to $3D$ surfaces.

## 2.4 Groupwise Correspondence

Groupwise correspondence is a method similar to pairwise with the only difference that correspondences are calculated by optimising an objective function across the whole set of shapes. Since the correspondence metric is optimised over the volume of the shape space and not between a pair of shapes, as in the case of the pairwise method, the global properties of the model are perceived. This method is preferable compared to pairwise since the similarity of shapes is measured globally thus the quality of the model is measured in a more precise mathematical form which leads to compact models.

Kotcheff and Taylor [97, 98] address the correspondence problem in terms of finding the correct pose and parameterisation of each shape in the training set such as a chosen objective function is minimised. As an objective function they have used the determinant of the covariance matrix which effectively measures the volume of the shape space and concentrates the variance into a few modes with large variances, thus by minimising it should lead to more compact models. Correspondences are calculated by explicitly re-parameterasing, using a piecewise linear function, the original arc-length parametesised shapes. Points are then allowed to move on the boundary of each shape. For legal correspondences to exist the mapping between the arc-length parameterisation and re-parameterisation of the points should be "$1-1$". In their method this problem is addressed by explicitly reordering the points within each shape. This constrain is problematic since it is very computational consuming, the method takes many hours to converge, and cannot be extended in $3D$. A Genetic Algorithm (GA) is used to optimise the objective function by manipulating the re-parameterisation function of each shape. The method of Kotcheff and Taylor [98] is improved by Davies *et al.* [39] who modify the re-parameterisation function so that it can be differentiable. This improvement has direct extension to surface re-parameterisation. Davies *et al.* [39] measure the quality of their model by optimising the Minimum Description Length (MDL) objective function. The method is very promising and it leads to automatically built shape models. However, due to very large number of function evaluations and nonlinear optimisation the method is computationally expensive.

## 2.5 Non Correspondence

The equally-spaced, pairwise and groupwise methods solve the correspondence problem either by equally spacing the shape function, and no redistribution of the points along the shape path is involved, or by grouping the points into higher level structures such as lines, curves or surfaces and parameterising the points along these attributes. An optimal transformation/mapping such as estimating the mean, the covariance or the probability distribution between rigid or nonrigid 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. The more global it is the better the quality of the built model.

The method discussed here bypasses the correspondence problem rather than solve it and is known as the non correspondence method [39]. 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 nonlinear manifolds.

When using nonlinear methods correspondences are calculated either as a probability density estimation function where both **correspondence and shape transformation** are solved in an iterative manner or as we propose vectors derived from **competitive hebbian learning** where the landmark points are the cluster centres in a high-dimensional space. In the former the probability distributions can be Dirac Delta functions represented as isotropic or oriented Gaussian mixtures [23, 176] and in the latter neurons with or without topological constrains. The objective of unsupervised classification is: given a high dimensional data distribution find a topological structure that best defines the topology of the data distribution. In this paper the correspondence problem, with its applications to nonrigid tracking and unsupervised model generation, is addressed as a topology learning problem and the automatic extraction and correspondence of landmark points is achieved through the calculation of the topographic product.

### 2.5.1 Correspondence and Shape Transformation

Belongie *et al.* [13, 14] solve the correspondence problem between a pair of shapes and the transformation for shape-based recognition by introducing a descriptor to each point. Each chosen point is connected by lines to the rest of the points and the length and the orientation of each line is calculated. This distribution of the length and the orientation of the lines is computed through histogram difference and is used as the *shape context* descriptor for the first point. A cost function between all pairs of points from the first and the second shape is then minimised and correspondences are obtained by solving a bipartite matching problem. These correspondences are then used to estimate the transformation that best aligns the two shapes. By using the thin plate spline model shape deformations are allowed and the best transformation map is achieved. Their model is then used for object recognition. The results on various databases including handwriting recognition are good and the algorithm can work for both open and closed boundaries. However, the convergence properties of this algorithm are unclear since there is no global objective function that is being minimised.

Chui *et al.* [28] extend their previous work on pose estimation and correspondence by including spline-based deformations to their transformation scheme, originally restricted to affine and piecewise-affine mapping, and by developing a general purpose nonrigid point matching algorithm. Their thin plate spline robust point matching (TPS-RPM) algorithm is very similar to the Iterative Closest Point (ICP) algorithm but with improvements such as one-to-one correspondences are guaranteed and outliers can be part of the point sets. In an iterative process both correspondence and transformation are solved by minimising a linear assignment least square energy function. Optimisation problems such as the linear assignment discrete problem and the least square continuous problem are handled with the techniques of softassign and deterministic annealing [69, 136]. Although the algorithm gives good results for both rigid and nonrigid objects there is no notion of connectivity which means that in large deformations (e.g. caterpillar images) the algorithm fails and invalid correspondences are ensued.

Wang *et al.* [176] use the probabilistic correspondence method to simultaneously compute the mean shape from unlabeled $2D$ and $3D$ rigid and nonrigid point sets and to register them nonrigidly. The point sets are modelled as kernel probability distributions and the distance between these distributions is quantified by the Jensen-Shannon (JS) divergence measure. This cost function is then optimised, using a gradient based numerical optimisation technique such as the Quasi-Newton method, over a space of coordinate transformations yielding to the desired registration. The JS cost function is used to measure the similarity/closeness between the distributions. If these distributions are statistically similar then the point sets can be properly aligned under nonrigid transformations. The advantage of their work is that it can be used in atlas construction since it emerges during the registration process. The alignment of the unlabeled point sets and the atlas construction work very well but the location of the points need to be known *a priori* since there are manually extracted by experts. This is time consuming and it is not relevant to our approach which is both the automatic extraction and correspondence. Furthermore, this method like the previous will fail in large deformations since points are not connected and therefore not parameterised.

## 2.6 Proposed Method - Unsupervised Learning

In recent years, there have been a number of papers that have used self-organising models in applications related to computer vision, man-machine interaction, and biometric systems including: image compression [12, 18, 65], segmentation and representation of objects [53, 144, 182], tracking objects [6, 22, 55], recognition of gestures [16, 64], biomedicine [4, 37], and 3D reconstruction [2, 35, 73, 84, 138].

These share in common the attributes of dynamically generating and removing processing elements (vectors of a network) and dynamically generating and removing synaptic links (neighbourhood connections). Furthermore, these models make no assumptions about the global structure of the shape to be modelled or more generally of the problem to be learned [162]. In this thesis, and based on Fritzke's neural networks architecture, Growing Neural Gas (GNG) [60], we overcome temporary restrictions on problems such as tracking objects or the recognition of gestures, by processing and matching them over time, using the positions of the nodes in the network.

Many researchers have modified growing models to make them adapt to different applications. Fritzke [61] presented variations of the original GNG algorithm to deal with non-stationary distributions, which he called the Growing Neural Gas with Utility (GNG-U), and a semi-supervised variation SNG [58] combined with Radial basis function (RBF) networks. In recent years, many variations of the GNG algorithm have been proposed. Marsland *et al.* [114] present a variation, named Grow When Required (GWR) algorithm able to add nodes whenever the network does not sufficiently match the input. Furao and Hasegawa [62] introduced an incremental learning GNG model to handle online non-stationary problems. Prudent and Ennaji [133] also proposed an incremental learning model based on the Adaptive Resonance Theory (ART) mechanism, called the Incremental GNG (IGNG), to handle semi-supervised learning. Qin and Suganthan [134] proposed the Robust GNG (RGNG) algorithm for unsupervised clustering. The algorithm added several techniques to the original GNG algorithm to reduce the sensitivity of the algorithm to prototype initialization, input sequence, and outliers. Fatemizadeh *et al.* [51] modified the growing neural gas to automatically correspond important landmark points from two related shapes. The algorithm treats the problem of correspondence as a cluster-seeking method by adjusting the centers of points from the two corresponding shapes. Angelopoulou *et al.* [5] also used the GNG to automatically obtain interest points in medical shapes and built statistical shape models. Frezza-Buet [55] has slightly modified the original GNG algorithm, called GNG-T, by continuously performing vector quantisation over a distribution that changes over time. This method has been applied to people tracking. Wu *et al.* [182] and Stergiopoulou *et al.* [165] suggest the use of self-organising networks for human-machine interaction. Xiang Cao *et al.* [22] and Vasquez *et al.* [172] propose amendments to self-organising models for the characterisation of the movement. From the cited works, only [55] represents both the local as well as the global movement, however there is no consideration of time constraints, no exploitation of knowledge gained from previous frames, and the condition of finalisation for the GNG algorithm is defined by the insertion of a predefined number of nodes.

Considering the work in the area and previous studies about representation capabilities of self-growing neural models, we present an enhanced model which accelerates the learning process and makes it suitable for modelling and tracking an object in a sequence of $k$ frames. Specifically, we present a model where (1) the automatic extraction and correspondence of points are derived from competitive hebbian learning; (2) the network preserves the topology independently of global or local transformations; (3) the input space derived from the Expectation-Maximisation (EM) algorithm is incorporated in the model to track objects in a sequence of $k$ frames; (4) the classification of the gestures takes into account domain knowledge information that respects always some proportions found in hands; (5) the topology is best preserved with an optimal similarity threshold that maximises topology learning versus adaptation time.

Based on the above, we address the correspondence problem as an unsupervised classification problem where landmark points are the cluster centres (nodes) in a high-dimensional vector space [3]. The automatic extraction of landmark points is achieved using the GNG network and the model is built by using nearest neighbour relationships derived from competitive hebbian learning. The approach is novel in that the network can be used in cases where the topological structure of the input pattern is not known *a priori* thus no topology of fixed dimensionality is imposed onto the network. Furthermore, the correspondence of the nodes is achieved by adding an *active* step to the GNG network, which we call *Active* Growing Neural Gas (*A*-GNG) that allows the model to re-deform locally, and update its position [6]. The *Active*-GNG takes into consideration not only the geometrical position of the nodes 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 topology preserving 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 similarity of a pair of nodes before and after the mapping can be calculated. These features (e.g. topographic product, colour 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 experiments include nonrigid shapes like medical MRI brain ventricular images, hands, and rigid artificial data like squares and circles. The algorithm gives good results in rigid and nonrigid shapes, it is computationally inexpensive, it can handle multiple open/closed boundaries and it can easily be extend to $3D$ surfaces.

### 2.6.1 Constraints Applied to the Proposed Method

Specific restrictions have been applied to simplify the difficulty of the motion analysis in the gesture representation. In most systems these are based on the detection and tracking of the object of interest. Below we summarise our assumptions.

- There are limited changes in the velocity and the direction of the gestures. It is assumed that neither the speed nor the direction of the hand changes drastically.
- The camera does not move and any motion from the objects is in a plane parallel to the camera.
- The backgrounds are low to medium cluttered avoiding extremely cluttered scenes.
- There are no occlusions.
- Relative motion: a movement is assumed relative with respect to the morphological changes in the object. Local and global changes are perceived by the observer.

## 2.7 Summary

Research on nonrigid shape modelling provides the base for image understanding and classification. The most important fact however is that the model should be robust not prone to noise and able to handle occlusions. In order for the model to be robust correct correspondences should be established between a set of shapes. However, because of the complexity of nonrigid shape transformation/mapping, most methods simplify the task and either equally space the point sets along the shape or group the points into higher level structures such as lines, curves or surfaces and parameterise the points along these attributes. An optimal transformation/mapping such as estimating the mean, the covariance or the probability distribution between rigid or nonrigid 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. 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.

Since we want to solve for correspondences and in parallel use the model to track objects in a sequence of $k$ frames, we bypass the correspondence problem and we built topology preserving graphs based on nearest neighbour relationships derived from competitive hebbian learning. By following this representation and by incorporating in the network features such as the topographic product, local grey-level and distance vector, we can automatically model and track in an unsupervised manner $2D$ objects.

# Chapter 3

# Object Representation with Self-Organising Neural Networks

*The contribution of this chapter is the automatic extraction, labelling and correspondence of points using only topological relations derived from competitive hebbian learning. In the beginning, we discuss the self-organising neural network Growing Neural Gas (GNG) and show how it can be used for the automatic extraction and correspondence of nodes in a set of objects. Additionally, now that we have obtained the points, we build statistical human brain MRI and hand gesture models using the Point Distribution Model (PDM).*

## 3.1 Introduction

The objective of accurate nonrigid shape modelling is the construction of decision boundaries based on unlabeled training data that can solve for correct correspondences between a set of shapes. Such correspondences can be classified as the problem of finding homogeneous landmark points in a multidimensional data set.

In medical imaging, Magnetic Resonance Imaging (MRI) techniques provide a non-invasive and accurate method for determining the ultra-structural features of human anatomy. The cerebral ventricles are buried within the centre of the brain parenchyma and are the source of cerebral spinal fluid, which provides nutritive and cushioning support to the brain and spinal cord. Neuropathologies involving the ventricles range from severe hypertrophy diagnostic for hydrocephalus, to mild and diffuse enlargements associated with AIDS, Alzheimer's Disease and Schizophrenia [44, 67]. Currently, MRI techniques are employed routinely in the diagnosis of ventricular related diseases. In many cases, the extent of disease progression can be determined by quantifying the extent of the change in ventricular morphology and/or volume [44]. The usual practise in a clinical setting is to perform a high resolution T1-weighted MRI followed by laborious post-processing steps. These post-processing steps are 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. In addition, in a multi-centre study or when a patient visits multiple medical facilities, there is little assurance that the post-processing steps will be performed in an identical fashion. 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.

There are several algorithms that have been employed in the medical imaging domain which can be broadly classified into landmark and non-landmark based approaches. Typical non-landmark based techniques have been published using region-growing algorithms [148], level set [8], and rough sets based [179].

Landmark based techniques can be classified as manual, semi-automatic and automatic. Because the first two are laborious and subjective especially when applied to $3D$ images, various attempts have been made to automate the process of landmark based image registration and correct correspondences among a set of shapes. Souza and Udupa [157] use the landmarks of the mean shape of an MRI foot data set as a reference to automatically generate the landmarks to the training set by locally searching the distance between the given landmark point from the mean shape and the nearest strong edge in the image. This method is arbitrary since the mean shape can be defined only for closed boundaries and for set of images that are mainly aligned and have small variations. Davies *et al.* [40] 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. 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. Fatemizadeh *et al.* [51] have used modified Growing Neural Gas (MGNG) 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 chapter we develop a framework for an automatic, unsupervised statistical hand pose model using only topological relations. One way of extracting 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 neural 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.

Such models have been successfully used in applications such as speech processing [76, 93], robotics [66, 111, 113, 116, 143], biology [125, 177, 186], clustering [26, 187], medicine [5, 24, 37, 51], and image processing [123, 140]. These models share in common the attributes of dynamically generating and removing processing elements (vectors of a network) and dynamically generating and removing synaptic links (neighbourhood connections). Furthermore, these models make no assumptions about the global structure of the shape to be modelled or more generally of the problem to be learned. The method is based on the assumption that correspondences are the nodes (the cluster centres in a high-dimensional vector space) of a network. The automatic extraction, labelling and correspondence of nodes is performed with the Growing Neural Gas (GNG) network introduced by Fritzke [60]. 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 graph (Figure 3.1). Statistical nonrigid shape models are then built by using the point distribution model (PDM) introduced by Cootes *et al.* [31].

The rest of this chapter is organised as follows. Section 3.2 describes the self-organising models and the modifications we have applied to the network to eliminate wrong edges and to reorder the nodes in the map. Section 3.3 shows how

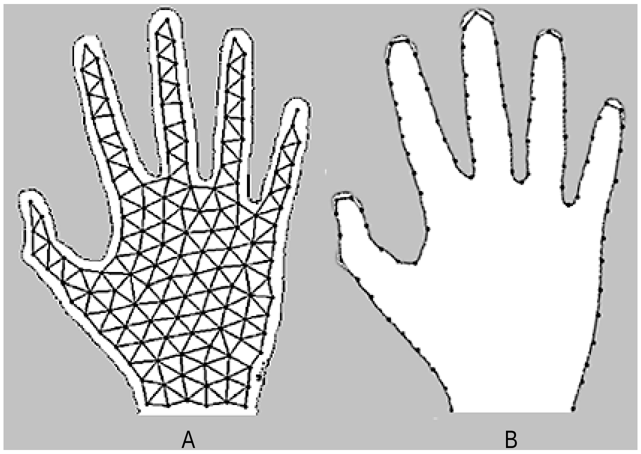


Figure 3.1: Examples of the two most common topologies. Image A represents the topology preserving graph ($TPG$) of a triangular grid while image B the topology of a line. In both cases the $TPG = \langle A, C \rangle$ is defined by a set of nodes $A$ and a set of connections (edges) $C$ that connect them.

we can build statistical shape models. Section 3.4 presents the evaluation criteria we have used in all our experiments. These topology measures are also used in Chapters 4 and 5. Experiments on different data sets and comparisons with other self-organising models are presented in Section 3.5. We summarise our method in Section 3.6.

## 3.2 Self-Organising Neural Networks

The term 'Neural Networks' is a biological term, and it's being used interchangeably with the term Artificial Neural Networks (ANNs). Neural Networks have been inspired from studies of biological nervous systems, and they attempt to create machines that work in a similar way to the human brain. They are called like that because the networks consist of interconnected elements similar to the architecture and operation of biological neurons [132, 178]. Most of them, apart from the Boolean ones, are composed of elements which are direct descendants of the model of a biological neuron created by McCulloch and Pitts [120]. A neural network is composed of a number of nodes or units, connected by links. Each link has a numeric weight associated with it, and learning takes place by updating the weights. The training or the learning phase of these networks can be categorised into two areas - supervised learning and unsupervised learning [132, 147]. In supervised learning the output is provided by the user during training and the learning parameters re-adjust until the data can be correctly analysed by the network. With unsupervised learning the network is allowed to produce its own output which is then further evaluated. The advantage of unsupervised learning is that the network finds its own energy minima and therefore tends to be more efficient in terms of the number of patterns that it can store and recall.

### 3.2.1 Preliminaries

Self-organising neural networks introduce the concepts of self-organisation and unsupervised learning. These networks have been used in recent years in different applications: compression [65], segmentation of objects [182], reconstruction of surfaces [138], economics [107], industrial applications [139], and biology [125]. The aim of self-organisation is to represent high-dimensional data as a low dimensional array of numbers that captures the structure in the original data [178]. Representing an unknown continuous density probability function by a finite set of few vectors reduces the information and allows us to analyse, compress or represent the complexity of the problem.

There are a number of self-organising neural models which share several architectural properties as presented below.

A self-organising network $A$ is formed by a set of $N$ nodes:

$$A = \{c_1, c_2, ....., c_N\} \tag{3.1}$$

where each of the nodes has an associated reference vector (weight) belonging to the space of input signals $\mathbb{R}^q$:

$$\{x_c\}_{c=1}^{N} \in \mathbb{R}^q \tag{3.2}$$

to indicate the area of the input space to which the node is more influenced. These reference vectors or weights in self-organising networks represent the coordinates of the topological map (rectangular or hexagonal) and are adjusted during the adaptation process.

Between the nodes of the network there exists a (possibly empty) set

$$C \subset AxA \tag{3.3}$$

of symmetric neighbourhood connections.

$$(i, j) \in C \Leftrightarrow (j, i) \in C \tag{3.4}$$

These connections have nothing to do with the weighted connections found, for example, in multi-layer perceptrons. They are used so that a node $c$ has a set of topological neighbours $N_c$:

$$N_c = \{i \in A/(c, i) \in C\} \tag{3.5}$$

Learning is based on a set of $q$-dimensional input signals that are generated following a probability density function

$$p(W), W \in \mathbb{R}^q \tag{3.6}$$

For each input signal $\xi_w$, through a competitive process between $q$ nodes, the winner node $x_\nu (x_\nu \in A)$ is the node with the nearest reference vector to $\xi_w$:

$$x_\nu = \arg\min_{c \in A} \|\xi_w - x_c\| \tag{3.7}$$

where $\| \cdot \|$ denotes the Euclidean vector norm.

Subsequently, all or part of the nodes of the network (based on the neighbourhood) adapt their reference vectors to the input signal according to the Hebb's law [115]:

$$\Delta x_c = \alpha \cdot (x_c - \xi_w) \tag{3.8}$$

where $\alpha$ weights the adaptation step.

Once the self-organising process is finished, we obtain a map of the input signals $\mathbb{R}^q$ onto the neural network $A$ such that:

$$f_x : \mathbb{R}^q \to A, W \in \mathbb{R}^q \to f_x(W) \in A \tag{3.9}$$

where $f_x(W)$ is obtained from the condition:

$$\| X_{fx(W)} - \xi_w \| = \min_{c \in A} \|\xi_w - x_c\| \tag{3.10}$$

An introduction to competitive learning and the principles of computational geometry is presented in Appendix A.1.

### 3.2.2 Growing Neural Gas (GNG)

GNG [60] is an unsupervised incremental self-organising network independent of the topology of the input distribution or space. It uses a growth mechanism inherited from the Growth Cell Structure [59] together with the Competitive Hebbian Learning (CHL) rule [118] to construct a network of the input date set. In some cases the probability distribution of the input data set is discrete and is given by the characteristic function $\xi_w : \mathbb{R}^q \to \{0, 1\}$ with $\xi_w$ defined by

$$\xi_w = \begin{cases} 1 & \text{if } \xi \in W \\ 0 & \text{if } \xi \in W^c \end{cases} \tag{3.11}$$

In the network $\xi_w$ represents the random input signal generated from the set $W \subseteq \mathbb{R}^q$ and $W^c$ is the complement of $W \in \mathbb{R}^q$. The growing process starts with two nodes, and new nodes are incrementally inserted until a predefined conditioned is satisfied, such as the maximum number of nodes or available time. 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 and new connections between the winner node and its topological neighbours are created.

The GNG algorithm consists of the following:

- A set $A$ of cluster centres known as nodes. Each node $c \in N$ has its associated reference vector $\{x_c\}_{c=1}^{N} \in \mathbb{R}^q$. The reference vectors indicate the nodes' position or *receptive field centre* in the input distribution. In our examples, the input probability distribution is a discrete distribution (Figures 5.3, 5.7) and a mixture of Gaussians probability density function representing skin colour (Figure 4.4). The nodes move towards the input distribution by adapting their position to the input's geometry using a winner take all mapping. Generating $\xi_w$ input signals from the random vector $W$, we want to find a mapping $G : \mathbb{R}^q \longrightarrow \mathbb{R}^A$ and its inverse $F : \mathbb{R}^A \longrightarrow \mathbb{R}^q$ such that $\forall c = 1, ..., \mid N \mid$,

$$f(x) = E_{W|g(W)}\{W|g(W) = x\}, \forall x \in \{x_c\}_{c=1}^{N} \subseteq \mathbb{R}^q \tag{3.12}$$

$$g(W) = \arg \min_{\nu \in \{x_c\}_{c=1}^{N}} \|W - x_\nu\| \tag{3.13}$$

  where $E$ is the distance operator of the data points from the random vector $W$ projecting onto $f(x)$, $g(W)$ is the projection operator, $\{x_c\}_{c=1}^{N} \subseteq \mathbb{R}^q$ are the reference vectors of the network and $x_\nu$ is the winner node. Equations $(3.12)$ and $(3.13)$ 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 $x$. In order to compute $f(x)$ the GNG algorithm evaluates (3.12) and (3.13) in an iterative manner. $q$ and $A$ denote the dimensionality of the input space and the reduced latent topology. In this work, current experiments include topologies of a line which is the contour of the object ($A = 1$) and triangular grid which is the topology preserving graph ($A = 2$). Figure 3.2 shows an example of a $1D$ GNG network.

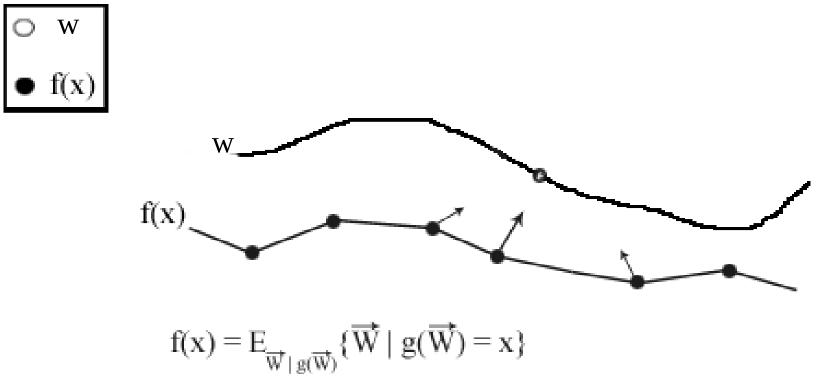


Figure 3.2: Every sample point $w$ on the target space is defined as the best matching of all nodes $x$ projecting within a topological neighbourhood of $w$. For example, the best matching node denoted by the largest arrow, moves towards the sample point while its topological neighbors adjust their position.

Figure 3.3 shows an example of a $2D$ GNG network with its associated Voronoi diagram in $2D$ discrete distribution.

- Local accumulated error measurements and insertion of nodes. Each node $c \in N$ with its associated reference vector $\{x_c\}_{c=1}^{N} \in \mathbb{R}^q$ has an error variable $E_{x_c}$ which is updated at every iteration according to:

$$\Delta E_{x_\nu} = \|\xi_w - x_\nu\|^2 \tag{3.14}$$

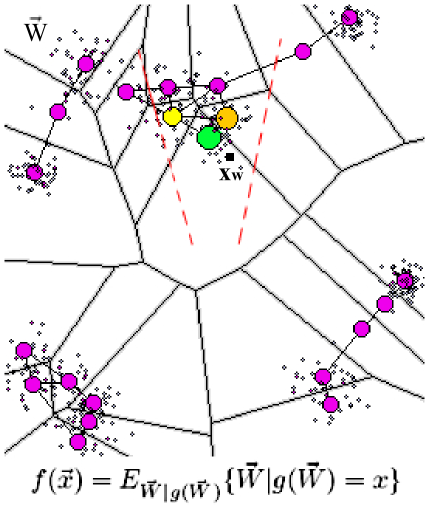

$\vec{W}$
Xw
$f(\vec{x}) = E_{\vec{W}|g(\vec{W})}\{\vec{W}|g(\vec{W}) = x\}$

Figure 3.3: A random signal $\xi_w$ on the discrete input distribution and the best matching within the topological neighbourhood of $\{x_c\}_{c=1}^{N} \subseteq \mathbb{R}^q$. In this example, the green node is the winner node of the network among its direct topological neighbours (orange and yellow nodes). The orange node is the second nearest node to the random signal $\xi_w$.

The local accumulated error is a statistical measure and is used for the insertion and the distribution of new nodes. Nodes with larger errors will cover greater area of the input probability distribution, since their distance from the generated signal is updated by the squared distance. Knowing where the error is large, if the number of the associated reference vectors belonging to the input space is an integer multiple of a parameter $\lambda$, a new node $x_r$ is inserted halfway between the node with the largest local accumulated error $x_q$ and its neighbour $x_f$.

$$x_r = \frac{x_q + x_f}{2} \tag{3.15}$$

All connections are updated and local errors are decreased by:

$$\Delta E_{x_q} = -\alpha E_{x_q} \tag{3.16}$$

$$\Delta E_{x_f} = -\alpha E_{x_f} \tag{3.17}$$

A global decrease according to:

$$\Delta E_{x_c} = -\beta E_{x_c} \tag{3.18}$$

is performed to all local errors by a constant $\beta$. This is important since new errors will gain greater influence in the network resulting in a better representation of the topology.

- A set $C$ 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:

  $$\Delta x_\nu = \epsilon_x(\xi_w - x_\nu) \tag{3.19}$$

  $$\Delta x_c = \epsilon_n(\xi_w - x_c), \forall c \in N \tag{3.20}$$

  where $\epsilon_x$ and $\epsilon_n$ represent the constant learning rates for the winner node $x_\nu$ and its topological neighbours $x_c$. 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 $\alpha_{max}$ and subsequently by removing the nodes connected to these edges.

The learning process of the GNG in the form of a pseudo code is summarised in Algorithm 1. The analytical steps of the algorithm are discussed in Appendix A.2.

**Algorithm 1** The GNG algorithm

```
**Input:** input vectors $x_c$
**Output:** $TPG$
```

```
1. Initialise two vector prototypes $A = \{c_1, c_2\}$ at random positions $\{x_{c_1}, x_{c_2}\}$,
   and the connection set $C$, $C \subset AxA$ to an empty set $C = \emptyset$
2. **while** the current number of prototypes $\leq$ to the maximum number of prototypes **do**
   (a) **for** every input signal $\xi_w$ **do**
       • Determine the winner prototype $x_\nu$ and the second nearest $x_\upsilon (x_\nu, x_\upsilon \in A)$ by Equation 3.7
       • Add the squared distance between the input vector $\xi_w$ and the winner $x_\nu$ to a local accumulated error variable (Equation 3.14)
       • Adjust $x_\nu$ position and its topological neighbours by Equations 3.19 and 3.20
       • Update connections between prototypes
       • Remove any dead nodes
         i. **if** the current number of prototypes is an integer multiple of a parameter $\lambda$ **then**
            – Add a new prototype by Equation 3.15
            – Update the connections between the prototypes
            – Decrease local errors by Equations 3.16 and 3.17
         ii. **end if**
   (b) **end for**
3. Decrease the error for all prototypes by Equation 3.18
4. **end while**
```

The current problems with the GNG are the *dead nodes*, when a non-stationary input distribution occurs, and keeping nodes correspondence between successive frames. The second is very important in cases where subtle changes to the shape of the objects occur. Examples can be found in morphology where shape differences can suggest a connection or not between normal structures [39]. Thus, instead of adopting a global approach where all the nodes need to re-adapt their position, local adaptations of the nodes would be more appropriate since they will maintain the correspondences without violating the network. We tackle the above problems by proposing *Active*-GNG in Chapter $5$.

### 3.2.3 Characterising $2D$ Objects Using GNG

Identifying the points of the image that belong to objects allows the network to obtain an induced Delaunay triangulation of the objects. Let an object $O = [O_G, O_A]$ be defined by its geometry and its appearance. The geometry provides a mathematical description of the object's shape, size, and parameters such as translation, rotation, and scale. The appearance defines a set of object's characteristics such as colour, texture, and other attributes.

Given a domain $\mathbf{S} \subseteq \mathbb{R}^2$, an image intensity function $\mathbf{I}(x, y) \in \mathbb{R}$ such that $\mathbf{I} : \mathbf{S} \rightarrow [0, \mathbf{I}_{max}]$, and an object $O$, its standard potential field $\Psi_T(x, y) = f_T(I(x, y))$ is the transformation $\Psi_T : \mathbf{S} \rightarrow [0, 1]$ which associates to each point $(x, y) \in \mathbf{S}$ the degree of compliance with the visual property $T$ of the object $O$ by its associated intensity $\mathbf{I}(x, y)$.

Considering:

- The input distribution as the set of points in the image:

$$\mathbf{A} = \mathbf{S} \tag{3.21}$$

$$\xi_w = (x, y) \in \mathbf{S} \tag{3.22}$$

- The probability density function according to the standard potential field obtained for each point of the image:

$$p(\xi_w) = p(x, y) = \Psi_T(x, y) \tag{3.23}$$

Learning takes place with the GNG algorithm. So, during this process, the neural network is obtained which preserves the topology of the object O from a certain feature $T$. Therefore, from the visual appearance $O_A$ of the object is obtained an approximation to its geometric appearance $O_G$. Henceforth, the Topology Preserving Graph $TPG = \langle A, C \rangle$ is defined with a set of vertices (nodes) $A$ and a set of connections (edges) $C$. To speed up the learning we used the faster Manhattan distance [121] compared to the Euclidean distance in the original algorithm. Figure 3.4 shows that different $TPGs$ can be obtained from different features $T$ of objects without changing the learning algorithm of neural gas. It is only necessary to define a different potential field. Different potential field $\Psi_T(x, y)$ can cause different structures in the network. Figure 3.4 (d) represents the topology of a $2D$ object while Figure 3.4 (e) represents the contour.

It is useful to obtain a contour composed solely of sequentially linked nodes, so that angles of curvature can be analysed. In many cases, the GNG can be formed with more than two edges emanating from some nodes, such as in Figure 3.5. This can happen when there are either sharp corners or complicated junctions in the silhouette, particularly when the network is not given enough time or sufficient number of nodes to model the contour more accurately. Thus, we need a method to convert a complicated network into one comprised only of sequentially linked nodes. 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 nodes created in the learning process all the inappropriate cycles produced. The rule is with respect to the reordering of the nodes (Figure 3.6). In the process of reordering the nodes when we find a node with more than two neighbours we need to calculate what is the best candidate to include in the list of nodes that define the contour. For this purpose, we compare the slope

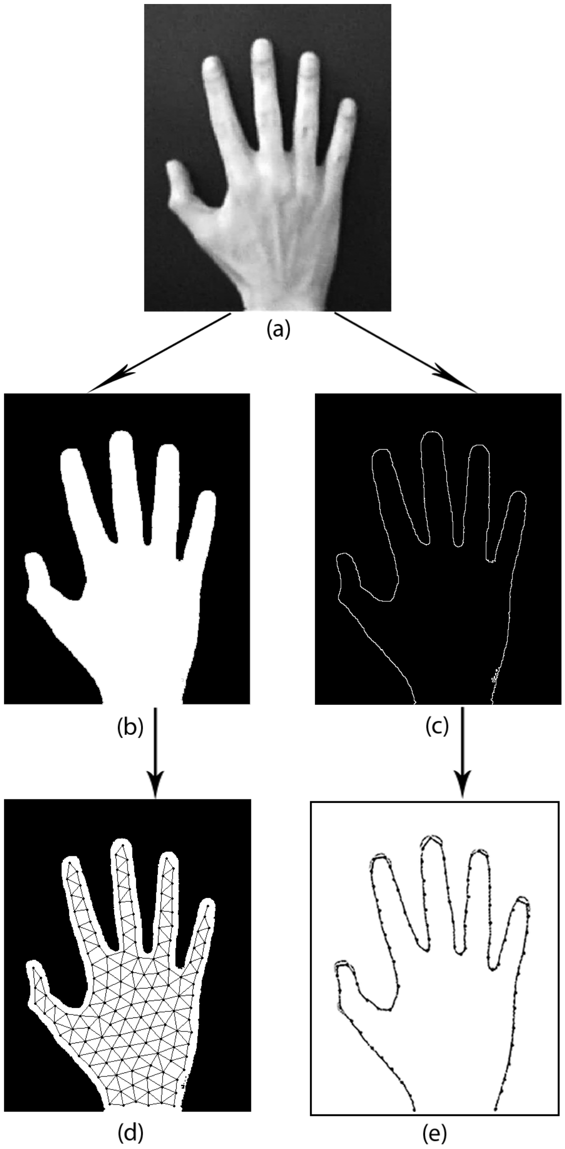


Figure 3.4: (a) Intensity image under normal illumination. (b) Binary image of the hand input space. (c) Binary image of the contour. Different adaptations of the GNG to the same object: (d) $TPG$ of a $2D$ object, (e) $TPG$ of a $1D$ contour representation.

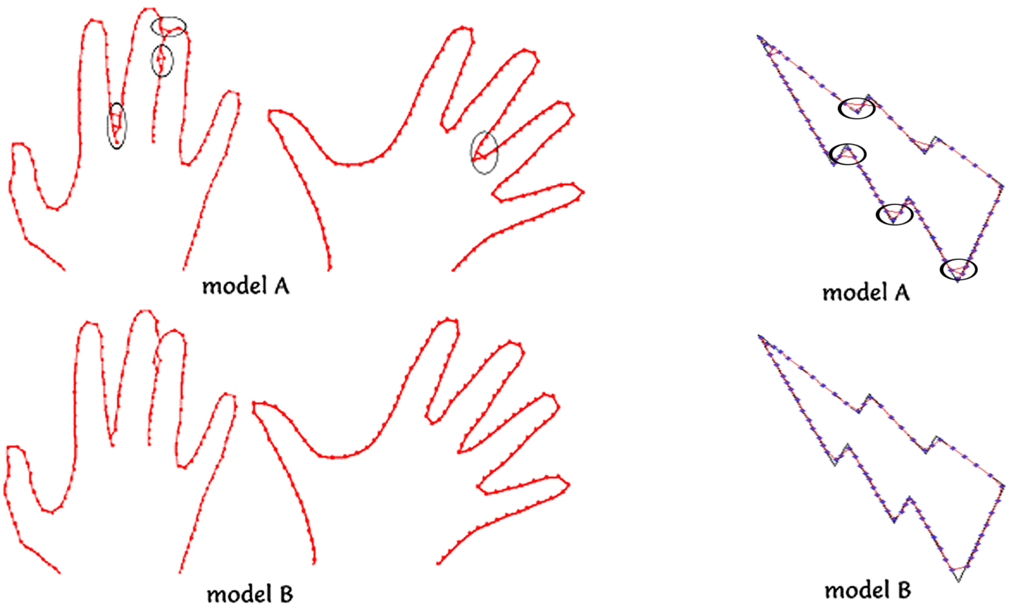


Figure 3.5: Modification of the GNG network to eliminate multiple connections and to attempt to reduce the network to a single series of sequentially linked nodes. model A is the original network with the wrong connections (circled corners), while model B is our modified network.

defined by the edge formed from the last two nodes inserted, with the slopes of the edges defined by the last node inserted and any of the candidate neighbours. We choose to add to the list the one with the least change of the slope.

- For example, n1 is an extreme and it has only one neighbour the node n2.
- Once we have inserted n1 in the ordered list if in the future we find n1 as a neighbour of any other node we must ignore it because it is already in the list.
- node n2 has as neighbours nodes n1 (already in the list) n3 and n4. In this case we have two possibilities, so we need to calculate the slope and decide

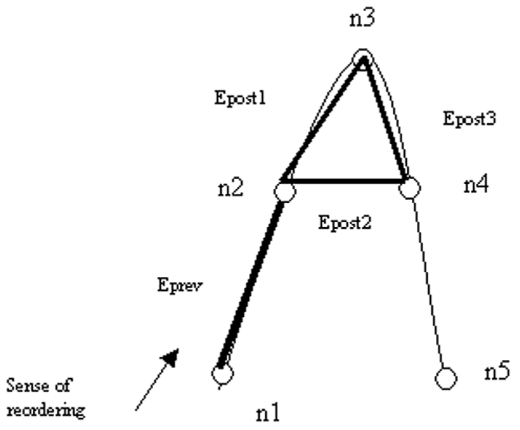


Figure 3.6: Reordering of neighbouring nodes.

if node n3 or n4 is the correct one to add to the list. In this example it is n3 because if we compare the slope of the edge n1-n2 to n2-n3 and to n2-n4, the change of the slope is clearly smaller for n2-n3.

- n3 has now two nodes since n2 is already in the list n4 is added to the list as well.
- n4 has three options, but if all of the previous steps are correct the nodes n2 and n3 have already been inserted in the list and the correct option is n5.
- If n5 is an extreme it has only one neighbour.

In summary, if a node has more than two neighbours connected by edges (structure of Delaunay Triangulation) like the node $n2$ in Figure 3.6, which has $n3$, $n4$, and $n1$ as neighbours, we consider the slope of the edges of the last node inserted in accordance to these neighbours ($Epost1$, $Epost2$, $Epost3$). We calculate the slope of the edge formed by the last two nodes inserted ($Eprev$) and we consider the correct neighbour is the one with the least change of slope. In Figure 3.6 the change of slope of the edge $n1 - n2$ ($Eprev$) respect to $n2 - n3$ ($Epost1$) and $n2 - n4$ ($Epost2$) is clearly smaller for $n2 - n3$ and we would add node $n3$ to the list. The procedure is summarised in Algorithm 2.

**Algorithm 2** Eliminating Wrong Edges

```
Input: $TPG$
Output: reduced $TPG$
1. We start with a set of nodes $A_N$ defined by coordinates $(x_i, y_i)$ and edges $C(A_s, A_t)$ $s \neq t$, and aim to represent these in terms of a sequence of lengths $l_j$ and turning angles $\theta_j$. The distance between two nodes $x_v$ and $x_u$ is their Euclidean distance $d(x_v, x_u)$. If $\exists A_i \in S : S = \{C(A_i, A_y) \cup C(A_y, A_i)\}, |S| = 1$, then the contour network has hanging ends consisting of some nodes with only one emanating edge. We select one of these as the starting node. If not, then every node $x_c$ has two neighbours, so the contour is closed and we arbitrarily select some starting node $x_c$.
2. Initialise new representation with $A_s = x_c$, $j = 0$ and previously considered node $f = -1$.
3. Set $\theta_j = 0$ and $l_j = d(x_v, x_u)$.
4. Set $f = c$ and $c = y$.
5. Find new neighbour $x_z$ of node $x_c$ along edge $C(A_z, A_c)$ such that $z \neq f$.
6. for $j = 1, ....., N$ do
   Set $\theta_j = \Theta(A_{j-1}, A_{j+1}, A_{j+1})$
      if $[\Theta(A_{j-1}, A_j)$ and $\Theta(A_j, A_{j+1})] <$ $[\Theta(A_{j-1}, A_j)$ and $\Theta(A_j, A_{j+2})]$ then
         update GNG, remove $C(A_j, A_{j+2})$
      end if
7. end for
8. Repeat from Step 4.
```

The list of nodes define a graph. To normalise the graph that represents the contour we must define a starting point, for example the node on the left-bottom corner. Taking that node as the first we must follow the neighbours until all the nodes have 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 nodes. We achieved the required alignment by applying a transformation $T$ composed by a translation $(t_x, t_y)$, rotation $\theta$, and a scaling $s$:

$$T\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} s(cos\vartheta)x_i & -s(sin\theta)y_i \\ s(sin\vartheta)x_i & s(cos\theta)y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.24}$$

The ordering of the nodes is summarised in Algorithm 3.

**Algorithm 3** Graph Normalisation

```
Input: reduced $TPG$
Output: nodes re-ordered
1. Start with reduced $TPG = \langle A, C \rangle$.
2. Assign starting node one of the two terminating nodes defined by coordinates $A_N(x_i, y_i)$ and edges $C(A_s, 1)$.
3. Find $nextNode$, the one connecting from the last found node (in this case first node) and update $previousNode$.
   for $i = 1, ....., S$ do
      if $C(A_i, 1) == nextNode$ do
         $previousNode == nextNode$
         $nextNode == C(A_i, 2)$
      end if
      if $C(A_i, 2) == nextNode$ do
         $previousNode == nextNode$
         $nextNode == C(A_i, 1)$
```

```
        end if
    end for
    Remove the connection that has already been followed.
    Reduce the number of connections and update list.
4.  while the stopping criteria such as the network size is not satisfied do
    (a) Find if there are more than one follow ons.
        if none exist then
            $C(A_i, 1) == 0$
        else if decide which one is next based on their difference
            $prevPrevNode = prevNode$
            $prevNode = nextNode$
            $nextNode = C(A_i, 2)$
            $C(A_i, :) = []$
            for $j = 1, ....., N$ do
                $xVec = Nodes(prevPrevNode, :)$
                $yVec = Nodes(prevNode, :)$
                $zVec = Nodes(nextNode, :)$
                $grad1 = diff(xVec - yVec)$
                $grad2 = diff(yVec - zVec)$
                $angleVals(j) = grad1 - grad2$
            Find minimum angle.
            end for
        end if
    (b) Update list, update connections.
5.  end while
```

The non-ordered nodes and the normalised nodes can be seen in Figure 3.7 A, and B. Figure 3.8 shows another example of normalised nodes for a closed shape.

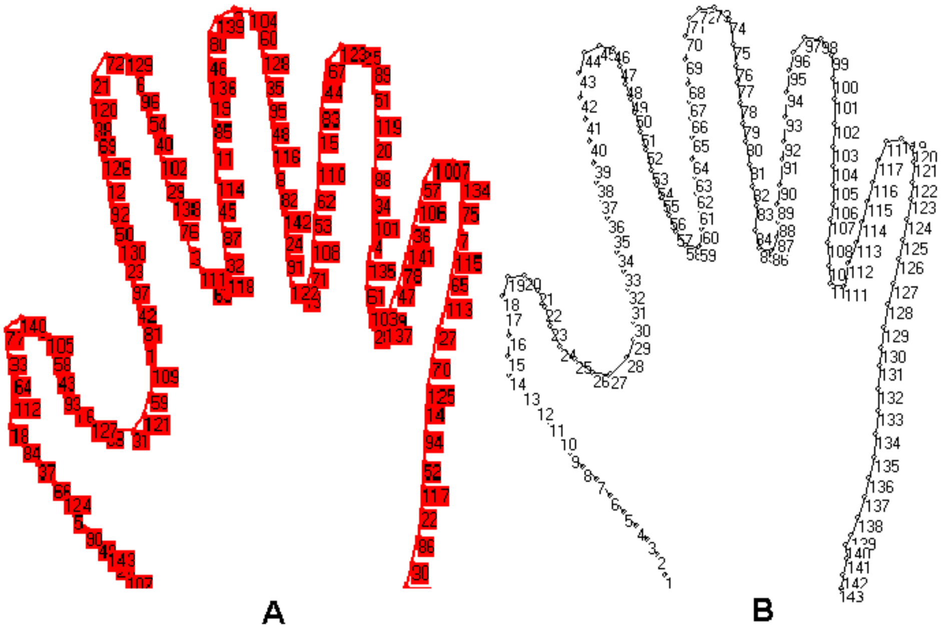


Figure 3.7: Image A shows the automatic node extraction and position before any reordering is applied. B shows the nodes after normalisation is performed. The shape of the hand is represented with a GNG map of 143 nodes.

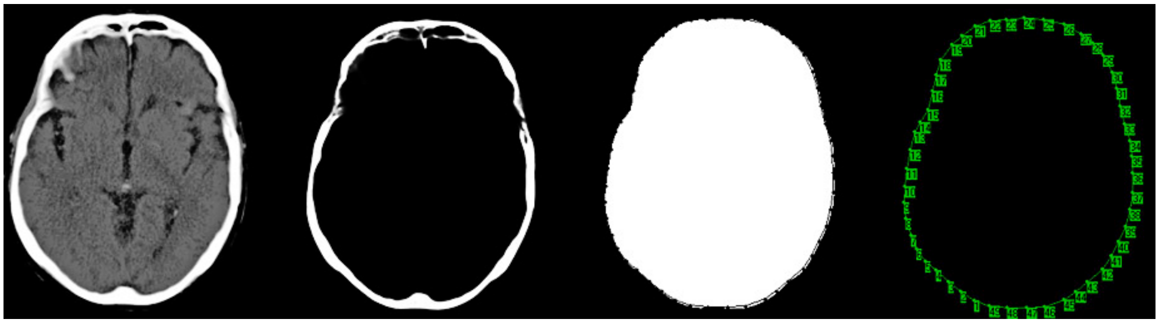


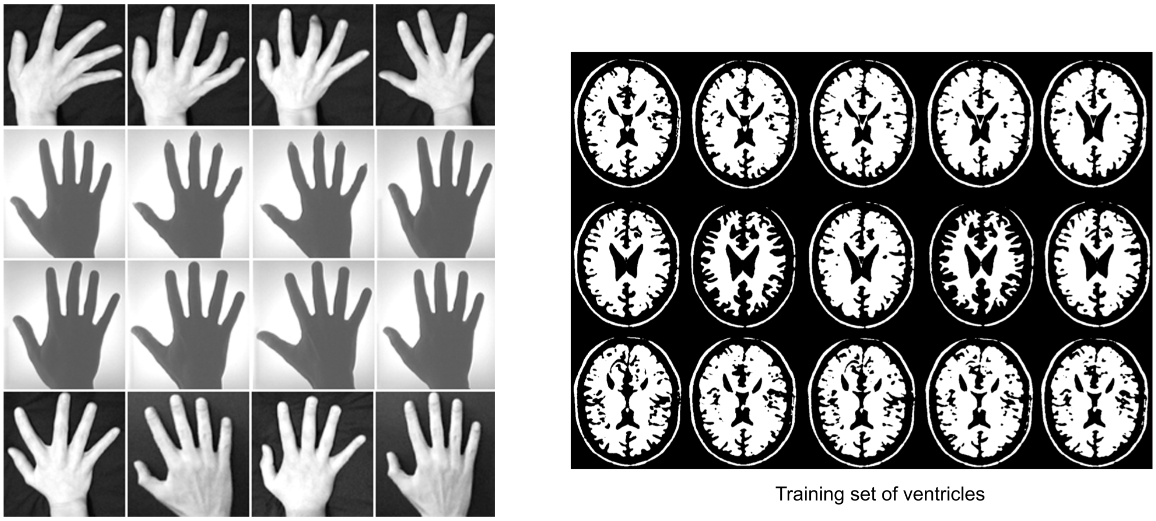
Figure 3.8: Normalisation of nodes in an MRI image after segmentation is performed. The starting node is on the left-bottom corner. The shape of the brain hemisphere is represented with a GNG map of 49 nodes.

## 3.3 Statistical Shape Models

Statistical shape models [30, 96, 159] are flexible models that have been used to capture the variance of the shape of a class of deformable objects by performing statistical analysis on the training set. Shape can be defined as any positive real function of points connected together, which is invariant under some transformation, and remains unchanged when the shape of the object is moved, rotated or scaled [30, 163]. By using the statistical shape models the variations in the shape are analysed over the training set and the derived models synthesise shapes similar to the training set.

To build a statistical shape model the set should include the types of variations one wishes to represent with the model. For example, if there is interest only in hands with changes in the position of the fingers then the training set should include only the required changes. If, however, more complex variations are required, like the bending of different fingers or changes in the pose of the hand, the training set should include these changes. Figure 3.9 shows the training sets of the hands and the ventricles from a series of high-resolution T1-weighted MRI brain images as used in this study.

The most well known statistical shape models are the Point Distribution Models (PDMs) [31], the Active Shape Models (ASMs) [32], and the Active Appearance Models (AAMs) [50]. In particular the PDM, which is the shape descriptor for both the ASM and the AAM models, models the shape of an object and its variation by analysing the statistics of the landmarks placed manually, semi-automatic or automatic in the training set. Each landmark has a certain distribution in the image space, thus the name point distribution model. These landmarks are used to represent the correspondences across the training set. Thus the same number



Training set of hands

Training set of ventricles

Figure 3.9: The training sets of hands and ventricles with various displacements.

of landmarks should be used for each image in the training set and at equivalent positions -in a $2D$ or $3D$ space- on different instances of the shape.

Below we review the four stages needed to obtain a PDM.

### 3.3.1 Labelling the Training Set

Every shape from the training set is represented by locating a number of points along the outline. Dryden *et al.* [46] named these points as landmarks where a landmark is a point of correspondence on each object that matches the same feature in the shape of the class of objects. The labelling is very important since it represents a particular part of the object such as eyes or nose in face recognition or high curvature points along the fingers in hand silhouettes. In order to establish correct correspondences among the shapes the following constraints should appear:

- The number of points used to describe each shape should be the same.
- The $i_{th}$ point on each shape should correspond to the same feature in the shape. For example, in the hand model the first and the last point represents the beginning and the end of the boundary. If the labelling is incorrect, with a particular point placed at different sites then the model building process will incorporate variation in the positioning of points into the model rather than simply the variation of the shape itself. If that happens then the model cannot be used for generalisation and specificity since illegal variations of the shapes will be synthesised.
- Enough points to adequately describe the shapes. Complex shapes require greater number of points.

In this thesis, the $2D$ shape of an object is represented as a set of $n_p$ automatically extracted points in a vector $\mathbf{x} = [x_{i0}, x_{i1}, ...., x_{in_{p-1}}, y_{i0}, y_{i1}, ..., y_{in_{p-1}}]^T$. Given $S_i$ training shapes $S_i$ such vectors $x$ are generated. In order to build statistical shape models the $S_i$ shapes are aligned in a $2D$ Euclidean transformation (translation, rotation and scaling) and normalised (removing the centre-of-gravity and placing it at the origin) to a common set of axes. The data were aligned using the Generalised Procrustes Analysis technique as derived by [17, 31, 72].

### 3.3.2 Aligning the Training Set

The alignment is necessary in the statistical shape modelling since ill-conditioned shapes will be generated based on the fact that the derived statistics would be based on the comparison of non-equivalent points. Figure 3.10 shows an example of non-aligned and aligned hand shapes. We achieved the required alignment by applying the transformation that minimises the sum of the squared distances

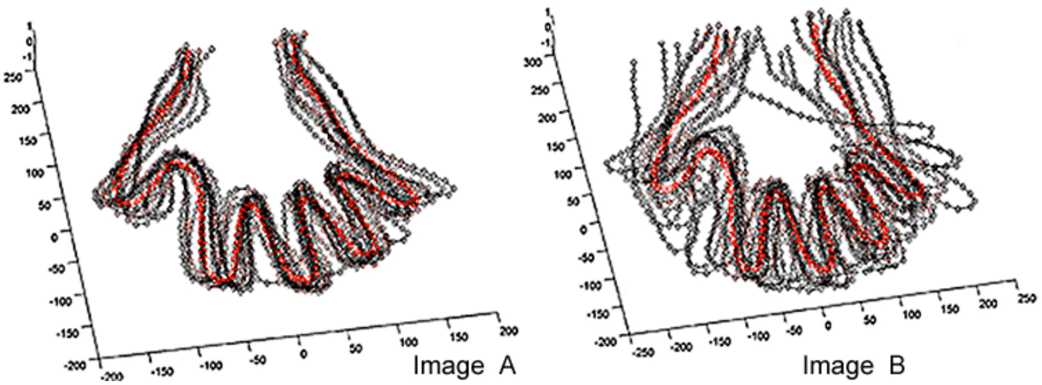


Figure 3.10: Image A represents aligned shapes with the centre-of-gravity removed and the mean shape (red shape) superimposed while Image B represents original non-aligned shapes.

between equivalent points on different shapes. The best fit may be found by minimising in a least-squares approach the expression:

$$E = |x_i - T(x_j)|^2 \tag{3.25}$$

where $x_i$ and $x_j$ represent the $i_{th}$ and the $j_{th}$ pair of shapes to be aligned and $T$ is the transformation matrix composed of a translation $(t_x, t_y)$, a rotation $\theta$, and a scaling $s$ such as:

$$T(x_j) = R\begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} s(cos\vartheta)x_j & -s(sin\theta)y_j \\ s(sin\vartheta)x_j & s(cos\theta)y_j \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.26}$$

By partially differentiating $E$ with respect to the unknowns $(\theta, s, t_x, t_y)$ and equating to zero four linear equations need to be solved. This approach is based on Cootes *et al.* [31] formulation. Details are given in Appendix B.1. The above method is known as the Unweighted Orthogonal Procrustes Analysis [34] and assumes that each point in the training set varies equally. If this is not the case, a weighted matrix should be used to signify those points which vary less than the others. This is important since points which vary a lot with respect to the others would be given a low weight, and vice versa. The weighted matrix is a diagonal matrix of weights which are chosen to give more significance to the points that vary less. This method is know as the Weighted Orthogonal Procrustes Analysis [34]. This weighted matrix as described by [31, 75] computes the distance between points $k$ and $l$ in a shape. Then the variance of the distance between every pair of points over all the shapes is calculated. $V_{kl}$ is the variance in these distances. If the sum of variances is large which means high mobility of particular points then a low weight $w_k$ should be given to those points and vice versa. This can be done by introducing a diagonal weighted matrix $W$ into Equation $(3.25)$:

$$E = W|x_i - T(x_j)|^2 \tag{3.27}$$

where

$$W = \sum_{k=1}^{n} w_k = \sum_{k=1}^{n} (\frac{1}{\sum_{l=1}^{N} V_{kl}}) \tag{3.28}$$

and $n$ is the number of landmark points. A weighted least-squares approach is explained in Appendix B.2. A different formulation is presented by Hamarneh [75].

If more than two shapes need to be aligned then the procrustes analysis can be modified to allow for this and is termed Generalised Procrustes Analysis [34]. To solve numerically the above method different algorithms exist [34]. In our case the alignment of the hands is performed by using the following algorithm derived by [156]:

- Rotate, scale, and translate each shape $x_i$ from the training set to the first shape $x_1$, for $i = 2, 3, ..., M$.

- Calculate the mean $\bar{x}$ of the transformed shapes.

- Rotate, scale, and translate the mean shape $\bar{x}$ to the first shape $x_1$.

- Rotate, scale, and translate each of the shapes $x_i$ to the adjusted mean $\bar{x'}$.

- If the adjusted mean $\bar{x'}$ has not converged return to step 2.

The algorithm for aligning all shapes is given in Appendix B.3. After convergence the training set is analysed for shape variations.

### 3.3.3 Statistical Modes of Variation

Let us say a set of $n_p$ points as vectors in $\delta$ dimensions from a training set of $M$ shapes exists. These vectors form a distribution in $n_p\delta$ dimensional space. In order to model this distribution, the assumption of being Gaussian is made since the training sets contain modest viewpoint variations with no rotations and no occlusions, a parameterised model of the form $x = M(\beta_i)$, where $\beta_i$ is a vector of weights for the $i^{th}$ shape is used. This model can be used to generate new examples similar to those in the training sets and to examine the validity of these examples. One core goal in model building is to select these parameters from the model that best describe the training set and be as few as possible. An effective approach in minimising the parameters while retaining as much information as possible is to apply principal component analysis (PCA) [85, 87, 129] also known as Karhunen-Love transformation (KLT) [108, 154] to the data.

Below we summarise the steps of PCA as used in the PDM model:

Let $X$ be the $M$ x $N$ data matrix whose rows $x_1, ...., x_M$, $x_M \in \Re^N$ are observations of a signal; in the context of hand recognition and modelling, $M$ is the number of available hand images in the training set, and $N = n_p\delta$ is the number of points in $\delta$ dimensions. In our case $\delta = 2$.

- Calculate the mean shape:

$$\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{3.29}$$

- Calculate the normalised covariance matrix:

$$\Sigma = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{3.30}$$

- Solve for the eigenvalues $\{\lambda^k\}$ and corresponding eigenvectors $V = \{v^k\}$ of $\Sigma$ such as:

$$\Sigma v^k = \lambda^k v^k \tag{3.31}$$

Any shape can be back-projected to the input space by a linear model of the form:

$$\mathbf{x} = \bar{\mathbf{x}} + V_s\beta_i \tag{3.32}$$

where $\bar{\mathbf{x}}$ is the mean shape, $V_s = (v_1|v_2|....|v_t)$ describes a set of $t$ orthogonal modes of shape variations (the eigenvectors), and $\beta_i$ is a vector of weights for the $i^{th}$ shape (the principal components). Instead of computing the covariance of the data one can compute the singular value decomposition (SVD). Both are intimately related as can be seen by [54, 75, 153]. To ensure that the above weight changes describe reasonable variations the weight $\beta_i$ is restricted to the range $-3\sqrt{\lambda} \leq \beta_i \leq 3\sqrt{\lambda}$ and the shape is back-projected to the input space.

### 3.3.4 Choice on Number of Modes

The number of the $t$ orthogonal modes of shape variations that best describe the data set was chosen by first checking the percentage of the variance captured. A common proportion is $96\%$ to $98\%$ and corresponds to the first six to eight modes of variations. The calculation is performed by first taking the sum of all the eigenvalues in the training set, $\Lambda_T = \sum \lambda_i$. We then choose the highest $t$ eigenvalues such that $\sum \lambda_i \geq P_T \Lambda_T$, where $P_T$ defines the proportion of the variance captured.

Additional confidence was obtained by calculating the root-mean-square error (RMSE) between the training set and the back-projected. The root-mean-square error (RMSE) is given as:

$$RMSE = \sqrt{\frac{1}{n_p} \sum \sum [M - x_i]^2} \tag{3.33}$$

where $n_p$ are the points as vectors, $M$ are the shapes in the training set and $x_i$ are the generated shapes.

## 3.4 Evaluation Criteria

In this section, we discuss the evaluation criteria for our experiments. We first introduce the concept of topology preservation, and then we discuss the measure we used to quantify the network. For problems such as object categorisation the evaluation criteria are well defined. In fact, a classifier is evaluated on a single image and the decision may either be correct or wrong. In contrast, in topology preserving networks a mapping between input space and network is perfectly topology preserving if and only if connected nodes $i, j$ that are adjacent in the network $A$ have weight vectors $x_i, x_j$ adjacent in the input space $\mathbb{R}^q$. In other words, a network can only perform a perfectly topology preserving mapping if the dimensionality of the map reflects the dimensionality of the input space.

### 3.4.1 Topology Preserving Networks

In any self-organising network the result of the *competitive learning*; the output neurons of the network compete among themselves to be activated or fired with the result that only one output neuron or one neuron per group is on at any one time, is a Delaunay triangulation graph. Traditionally, it has been suggested that this triangulation was sufficient to preserve the topology of the input space. However, Martinetz and Schulten [118] introduce a new condition which restricts this quality.

It is proposed that the mapping $f_x$ of $\mathbb{R}^q$ in $A$ preserves the vicinity when vectors that are close in the input space $\mathbb{R}^q$ are mapped to nearby neurons from network $A$. It is also noted that the inverse mapping preserves the neighborhood if and only if nearby neurons of $A$ have associated feature vectors close in the input space.

$$f_x^{-1} : A \rightarrow \mathbb{R}^q, c \in A \rightarrow x_c \in \mathbb{R}^q \tag{3.34}$$

Combining the two definitions, can be established the Topology Preserving Network (TPN) as the network $A$ whose mappings $f_x$ and $f_x^{-1}$ preserve the neighbourhood.

Thus, self-organizing networks such as Kohonen maps or Growing Cell Structures [57] are not TPN as has traditionally been considered, since this condition only would happen in the event that the topology or dimension of the map and the input space coincide. Since the network topology is established *a priori*, possibly ignoring the topology of the input space, it is not possible to ensure that the mappings $f_x$ and $f_x^{-1}$ preserve the neighborhood.

In the case of the NG and GNG, the mechanism for adjusting the network through a competitive learning generates an Induced Delaunay triangulation (Figure 3.11 (b)), a graph obtained from the Delaunay triangulation, which has only edges of the Delaunay triangulation of points which belong to the input space $\mathbb{R}^q$. Martinetz and Schulten [118] demonstrate that these models are TPN.

This feature is very important in the representation and the tracking of $2D$ hand gestures in a sequence of $k$ frames.

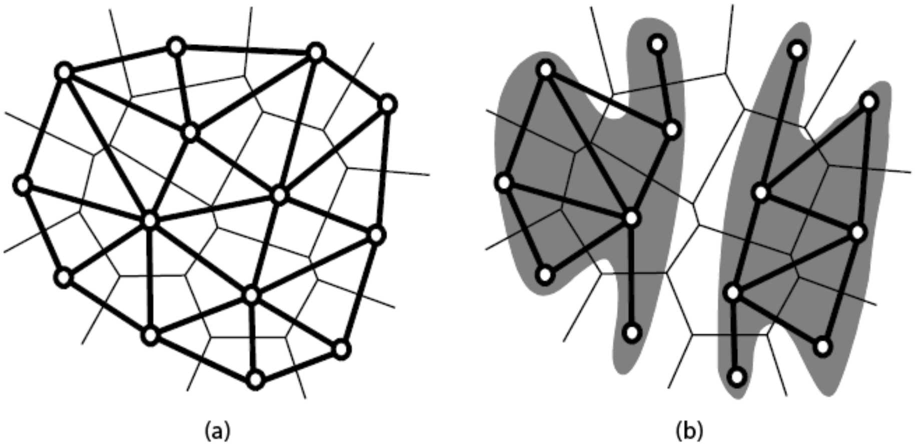


Figure 3.11: (a) Delaunay triangulation. (b) Induced Delaunay triangulation.

### 3.4.2 Measuring Topology Preservation

This section describes the measure we used to quantify the topology preservation of the GNG network. This measure is used to estimate the impact of time and network parameters in the topology preservation of different input spaces.

The adaptation of a self-organising neural network is often obtained by its resolution and its topology preservation of the input space. The measure of resolution can be calculated by minimising the *quantisation* or *distortion error*, which is to find the values of the reference vectors $\{x_c\}_{c=1}^{|N|} \in \mathbb{R}^q$ from the set $W \subseteq \mathbb{R}^A$ such that the error:

$$E = \sum_{\forall \xi_w \in \mathbb{R}^q} \parallel w_{s_{\xi_w}} - \xi_w \parallel^2 P(\xi_w) \tag{3.35}$$

is minimised, where $w_{s_{\xi_w}}$ is the nearest node to the input signal $\xi_w$.

With regards to the preservation of the topology, there are several measures used in the literature [63, 114]. The most relevant are: the Topographic Product (TP) by Bauer and Pawelzik [10], the topographic function by Villman *et al.* [174] and the C measure by Goodhill and Sejnowski [70]. All of these measures permit to quantify the preservation of the topology of the input space, however since our data sets are limited to linear data manifolds the best measure to use, regarding efficiency and computational cost, is the topographic product.

#### 3.4.2.1 Topographic Product

The topographic product $P$ introduced by Bauer and Pawelzik [10] is our topology measure which quantifies the neighbourhood preservation of the map by computing the Euclidean distance between neighbouring nodes, in both the input and the latent space. This measure is used to detect deviations between the dimensionality of the network and the input space. Folds in a network indicate that it is trying to approach a different input space dimension. 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 $\mathbb{R}^A$ and its associative reference vectors in the input space $\mathbb{R}^q$ is given by:

$$P_1(c,k) = [\prod_{l=1}^{k} \frac{d^A(c, n_l^A(c))}{d^A(c, n_l^q(c))}]^{1/l} \tag{3.36}$$

$$P_2(c,k) = [\prod_{l=1}^{k} \frac{d^q(x_c, x_{n_l^A(c)})}{d^q(x_c, x_{n_l^q(c)})}]^{1/l} \tag{3.37}$$

where $c$ is a node, $x_c$ is its reference vector, $n_l^q$ is the $l$-th closest neighbour to $c$ in the input space $\mathbb{R}^q$ according to a distance $d^q$ and $n_l^A$ is the $l$-th nearest node to $c$ in the latent space $\mathbb{R}^A$ according to a distance $d^A$. Combining (3.33) and (3.34) a measure of the topological relationship between the node $c$ and its $k$ closest nodes is obtained:

$$P_3(c,k) = [\prod_{l=1}^{k} \frac{d^q(x_c, x_{n_l^A(c)})}{d^q(x_c, x_{n_l^q(c)})} \cdot \frac{d^A(c, n_l^A(c))}{d^A(c, n_l^q(c))}]^{1/2k} \tag{3.38}$$

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

$$P = \frac{1}{N(N-1)} \sum_{c=1}^{N} \sum_{k=1}^{N-1} \log(P_3(c,k)) \tag{3.39}$$

Figure 3.12 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 vi-

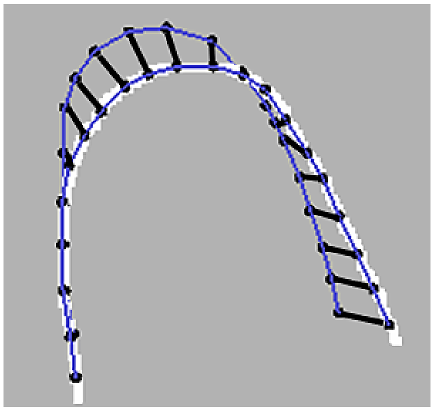


Figure 3.12: Neighbourhood relations are perfectly preserved since nearby points in the input space remain close to the nearby nodes in the latent space. The mapping is indicated by the lines.

olation of the topology occurs in Figure 3.13($a$) since the distance relations of the data points do not correlate with that of the reference vectors in the network. Figure 3.13($b$) shows the ideal correlation if correct correspondences have been previously established. The problem with the topographic product in cases like in Figure 3.13($a$) 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.

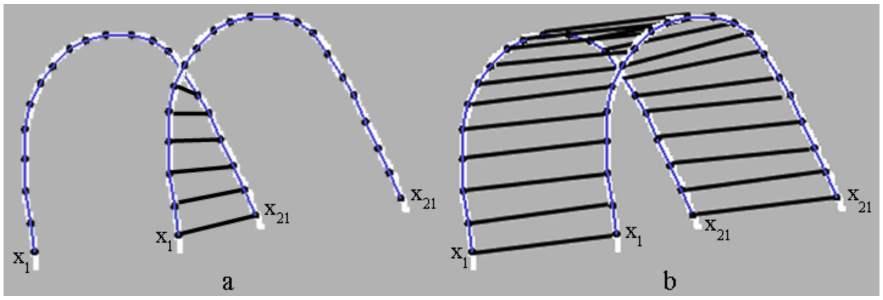


Figure 3.13: A set of nodes with their reference vectors $x_1$, $x_2$, up to $x_{21}$. As the input distribution moves and the network re-adapts, the distance relations between the data points and the reference vectors are violated (Image $a$). In the new adaptation the nearest neighbour of $x_1$ with its topological neighbours is not $x_1$ but $x_{21}$. Image $b$ shows correct correspondences if topological information such as closest Voronoi regions and not only metric information has been used.

## 3.5 Experiments

In this section, we conduct our experiments on a hand data set and compare the GNG algorithm with the Kohonen maps and the Neural Gas (NG) algorithms. Experiments on ventricles from human brain MRI, where accurate topology preservation can discriminate between correct and incorrect shape variations; an approach very important in applications such as morphological analysis [17, 119], are also presented in Appendix C. Both training sets can be seen in Figure 3.9.

### 3.5.1 Hand Model

Our proposed method has been evaluated on a data set of 16 hands, segmented by adaptive thresholding from video images with image resolution $800 \times 600$. The images were obtained from participants from the Computer Vision and Imaging Research Group[1], University of Westminster, UK. The outlines are represented as open curves to better represent the parameterisation of the nodes. 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). For computational efficiency, we have resized the images to $395 \times 500$ pixels.

All constant parameters have been fixed based on Fritzke [60] original paper and on our experience representing different objects in images. The parameters should not be set too high, as this will result in an unstable network with nodes moving too fast, thus violation of the $TPG$, or too low, as this will make the adaptation slow and ineffective. Experimenting with these values has lead to the following boundary values: $\epsilon_x = 0.05$, $\epsilon_n = 0.0006$, $\alpha_{max} = 150$, $\Delta x_{s_1} = 0.5$, $\Delta x_i = 0.0005$,

[1] http://perun.hscs.wmin.ac.uk/cvir/

and $\lambda = 1000$ to $10000$. All experiments have been performed on a 2.26 GHz Pentium IV processor and MATLAB and C++ Builder have been used to code and compile the algorithms.

#### 3.5.1.1 Extracting landmark points

Three different topology preserving networks were used for the evaluation. The testing involved two cases where the number of nodes were too few or too excessive for the training set of the images. In the former the topological map is lost, not enough nodes to represent the contour of the hands and in the later an overfit is performed.

To illustrate the performance of the convergence algorithm we present qualitative (Figure 3.15 and 3.16) and quantitative (Table 3.1) results for both manually and automatically generated models. The comparison was made by taking two reference models, a manually hand built model with 60 landmarks manually located around the boundaries, and an automatically hand built model with 144 nodes automatically generated around the boundaries (Figure 3.14). The optimum number of nodes in the network are determined by the topographic product.

In Figure 3.15 the modes are displayed by varying the first three shape parameters $\beta_i\{\pm 3\sigma\}$ over the training set.The first mode $\beta_1$ varies the shape of the thumb and increases the distance between the middle and the index finger. The second mode $\beta_2$ varies the distance between the thumb and the index finger, and bends the middle finger. The third mode $\beta_3$ varies the shape of the middle finger and the thumb.

In Figure 3.16 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

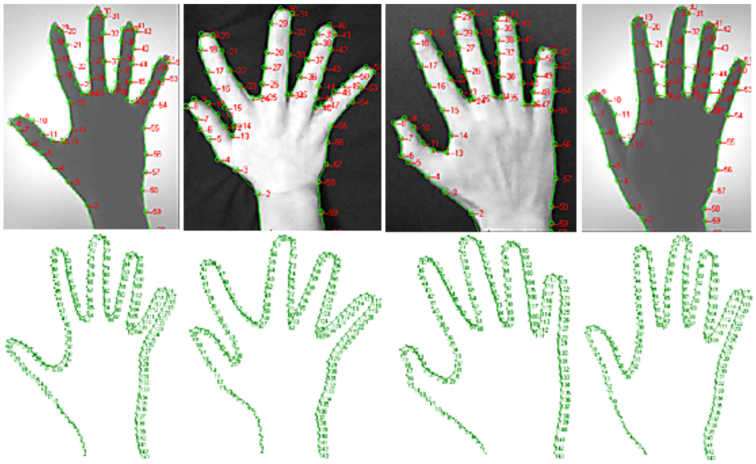


Figure 3.14: First row manually annotated landmarks. All landmarks are major and have been located manually. Second row hand adaptation with 144 nodes.

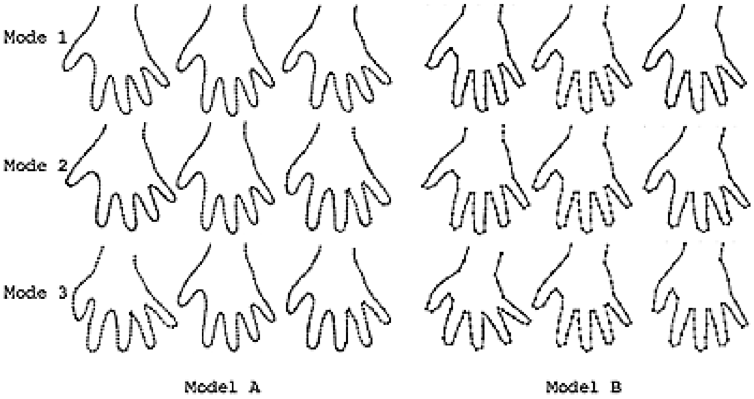


Figure 3.15: Model A shows the first three modes of variation of the automatically hand built model. Model B shows the first three modes of variation of the manually hand built model. Range of variation $-3\sqrt{\lambda} \leq \beta_i \leq 3\sqrt{\lambda}$.

modes effectively capture the variability of the training set and present only valid shape instances.

The quantitative results (Table 3.1) show that the automatically generated models are more compact than the manual models since less variance is captured per

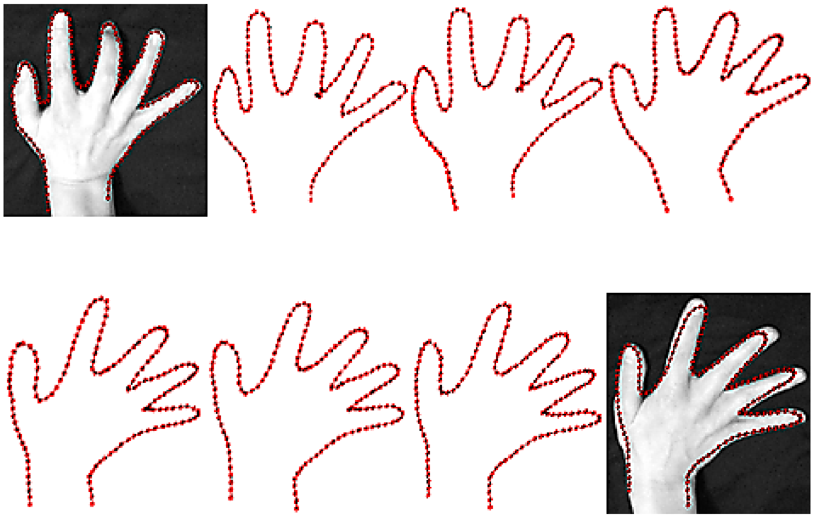


Figure 3.16: Superimpose instances to the training set and taking the in-between steps.

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. In Table 3.1 $V_T$ represents the variance for the first six eigenvectors. $T_P$ measures the topology preservation before and after the neighborhoud mapping between the input and the latent space. In the manual annotated model the match is too high since $P > 0$. In the automatic model $P \approx 0$ which indicates an approximate match.

Table 3.2 shows the total variance achieved by maps containing varying number of nodes $(25, 64, 100, 144, 169)$ used for the automatic annotation (Figure 3.17). The map of $144$ nodes is the most compact since it achieves the least variance. This is consistent with the optimal mapping selected by the topographic product. It is interesting to note that whilst there is significant difference between $25, 64$ and $100$ nodes the mapping with $169$ is good and has no difference, in terms of topographic representation, with the map obtained inserting $144$ nodes. The reason

Table 3.1: Performance evaluation for different variances

| Mode | Manual model (60 landmarks) | Automatic model (144 nodes) |
|---|---|---|
| 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 |
| $T_P$ | 0.039 | 0.012 |

Table 3.2: A quantitative comparison of various nodes adapted to the hand model with variances for the first six modes, total variance and the topographic product

| Mode | 25 (nodes) | 64 (nodes) | 100 (nodes) | 144 (nodes) | 169 (nodes) |
|---|---|---|---|---|---|
| 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 |
| $T_P$ | -0.053 | 0.037 | 0.025 | 0.019 | 0.022 |

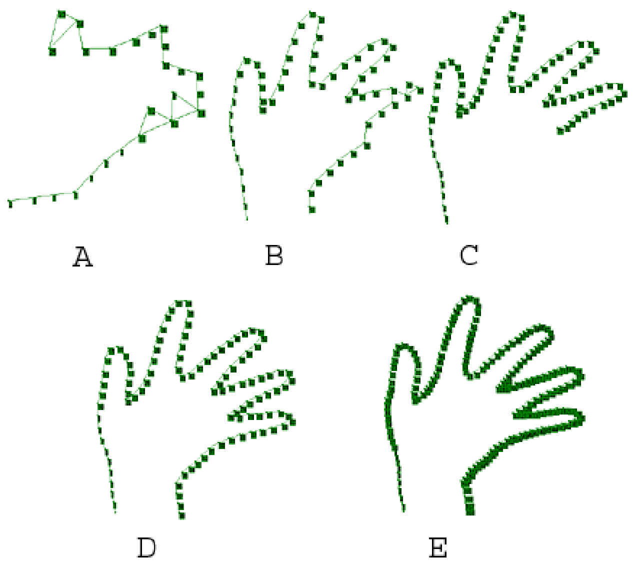


Figure 3.17: Adaptation to an object with network of 25 (Image A), 64 (Image B), 100 (Image C), 144 (Image D), and 169 (Image E), nodes.

is that for the current size of the images ($395 \times 500$ pixels) the distance between the nodes is short enough so adding extra nodes does not give more accuracy in placement. Thus, the topographic product for $144$ and $169$ nodes at $1000$ input patterns is nearly the same as can be seen from the Table 3.2. Table 3.3 shows the topographic product at different nodes and at different patterns. A qualitative representation of the topographic product is given in Figure 3.18. Furthermore, the insertion of more nodes increases the computation time and slows down the adaptation process. Figure 3.19 shows a comparative diagram of the learning time of various nodes and at different number of input patterns $\lambda$. The adaptation with the $144$ nodes is faster compared to the $169$, and it takes $22$ seconds at $5000$ patterns to adapt to the contour of the hand.

Figure 3.20 shows the adaptation process using three different topology preserving networks. The topology preservation of the Kohonen maps in comparison

Table 3.3: The topographic product at different input patterns

| Patterns | 25 (nodes) | 64 (nodes) | 100 (nodes) | 144 (nodes) | 169 (nodes) |
|---|---|---|---|---|---|
| 1000 | -0.053 | 0.037 | 0.025 | 0.019 | 0.022 |
| 5000 | -0.099 | 0.028 | 0.024 | 0.015 | 0.015 |
| 10000 | -0.07 | 0.028 | 0.022 | 0.012 | 0.016 |

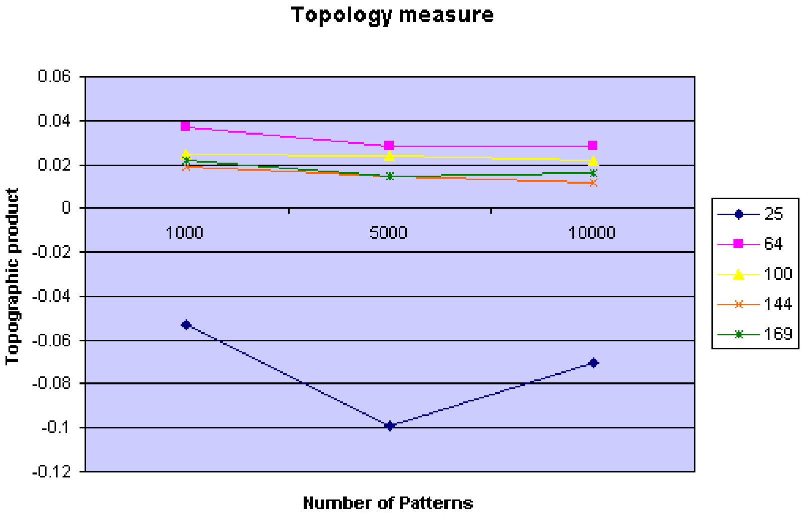


Figure 3.18: Topographic product at different input patterns and at different number of nodes as a measure of the topology preservation of the network.

to GNG is very poor. This is the case because in the original Kohonen map, the topology is constrained to be a two-dimensional grid and does not change during the self-organization. Furthermore, in order to provide good neighbourhood and topology preservation the logical structure of the input pattern (two-dimensional grid) should be known *a priori*. In other words, it specifies in advance the number of nodes in the network and the graph that represents topological relationships

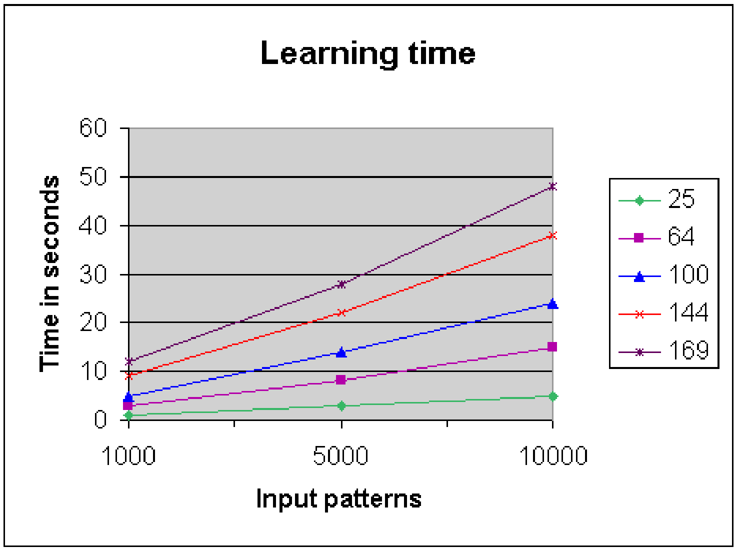


Figure 3.19: Learning time for various nodes and at different input patterns.

between the nodes. On the contrary, with the NG the topology preservation is well defined but the learning time is more than ten times higher than the time for GNG. This happens because in step 3 of the algorithm all the nodes in NG need to be ordered according to their distance and this is computationally very expensive compared to GNG where only the first and the second nearest nodes are ordered.

## 3.6 Summary

We developed an approach to automatically extract and label the contour of an MRI and hand-pose module using only topological relations derived from competitive hebbian learning. We defined the landmark points as the cluster centres in a high-dimensional vector space where correspondences are solved in nonlinear manifolds. The landmark points were extracted in two steps. First, the complete training set was segmented and the contours of the objects were extracted using an

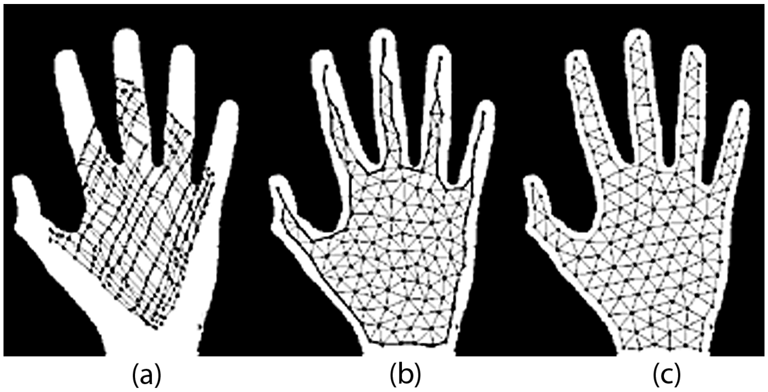


Figure 3.20: Adaptation process using three different topology preserving networks. $2D$ representation of the hand using (a) Kohonen maps, (b) NG, and (c) GNG networks.

edge detection scheme. Second, GNG was used to extract landmark points along the contours and to form topology preserving maps. The result of this adaptation was a list of non-ordered nodes that defined a graph. The graph was then normalised by defining a re-ordering rule of the nodes. The re-ordered list was then projected into the shape space where synthesised shapes similar to the training set were generated using the PDMs. These synthesised shapes are generated by independently varying the shape parameters from the distribution and reconstructed by the principal vectors that best capture the variation of the training set. Furthermore, we have improved the parameters of GNG by removing wrong edges between nodes that can be obtained either due to limited time of the network to adapt or the nodes are too close and the topology preserving graph cannot differentiate between winner and immediate neighbours. We have achieved that by defining a rule that compares the slope defined by the edge formed from the last two nodes inserted, with the slopes of the edges defined by the last node inserted and any of the candidate neighbours. By doing so we remove from the list all nodes created in the learning process with inappropriate cycles.

Experiments were performed on a training set of ventricles and hand poses. In both cases we have shown that GNG adapts successfully to the high dimensional manifold of the ventricles and the hands, allowing good eigenshape models to be generated completely automatically from the training set. The accuracy of our method was compared to other related self-organising networks.

# Chapter 4

# Adaptive Learning

*Based on the capabilities of neural models to readjust to new input patterns without restarting the learning process, we propose an approach to minimise the user intervention in specifying the number of nodes needed to represent an object by utilising an automatic criterion for maximum node growth. Furthermore, this model is used to the representation of motion in image sequences by initialising a suitable segmentation that separates the object of interest from the background.*

## 4.1 Introduction

In the previous chapter, we have demonstrated the capabilities of GNG to represent $2D$ objects by improving its parameters (e.g. removing wrong edges, node reordering) and we have automatically initialised a statistical model given the input distribution from the GNG algorithm based on binary or gray level images. In the learning framework, it is very crucial the initiliasation of the object, i.e. different segmentation methods can be applied depending on the application, to be correct because only then the model can be used for learning. The main idea is to find a suitable segmentation that separates the object of interest from the background.

Segmentation is a pre-processing step in many computer vision applications. The goal is to determine pixels in an image that are significantly different to other previous images. These applications include visual surveillance [27, 38, 77, 100, 104], and object tracking [99, 128, 145, 190]. While a lot of research has been focused on efficient detectors and classifiers, little attention has been paid to efficiently labeling and acquiring suitable training data. The collection of training data requires the segmentation and alignment of an observation sequence, which is an ill-conditioned task due to measurement noise and human variation in the observation. Furthermore, it is a time consuming and tedious task.

Obtaining a set of training examples automatically is a more difficult task. Existing approaches to minimise the labeling effort [103, 105, 122, 155] use a classifier which is trained in a small number of examples. Then the classifier is applied on a training sequence and the detected patches are added to the previous set of examples. Levin *et al.* [105] start with a small set of hand labeled data and generate additional labeled examples by applying co-training of two classifiers. Nair and Clark [122] use motion detection to obtain the initial training set. Lee *et al.* [105] use a variant of eigentracking to obtain the training sequence for face recognition and tracking. Sivic *et al.* [155] use boosting orientation-based features to obtain training samples for their face detector. However, to learn the model for the feature position and appearance a great amount (e.g., $1000$ images) of hand-labeled face images is needed.

A disadvantage of these approaches is that either a manual initialization [103] or a pre-trained classifier is needed to initialise the learning process. Having a sequence of images this can be avoided by using an incremental model. One of the most important characteristics of the GNG is that it does not require the restarting of the initialisation of the network for every image in a sequence of $k$ frames. This is achieved by using the position of the nodes in the network as features to follow where a new TPG representation is obtained for every image sequence. A detailed discussion is given in Chapter 5.

In this chapter, we are interested in the initialisation of the first frame of the GNG network. We are interested in the problem of background modelling where the goal is to get a segmentation of the background, i.e. the irrelevant part of the scene, and the foreground. If the model is accurate, the regions that represent the foreground (objects of interest) can then be extracted. In our experiments, the key to successful hand segmentation relies on reducing meaningless image data. We achieve that by taking into consideration that human skin has a relatively unique colour and we apply appropriate parametric skin distribution modelling.

The rest of this chapter is organised as follows. Section 4.2 summarises the initialisation of the object using probabilistic colour models. Section 4.3 proposes an approach to minimise the user intervention in the termination of the network using knowledge obtained from information-theoretic considerations. In Section 4.4 our method is applied to real and artificial shapes before conclusions are drawn in Section 4.5.

## 4.2 Background Modelling

We subdivide background modelling methods into two categories: (1) background subtraction methods; and (2) statistical methods. In background subtraction methods, the background is modeled as a single image and the segmentation is estimated by thresholding the background image and the current input image. Background subtraction can be done either using a frame differencing approach or using a pixel-wise average or median filter over a number of $n$ frames. A more detailed discussion on background subtraction can be found in [74, 131]. In statistical methods, a statistical model for each pixel describing the background is estimated. The more the variance of the pixel values, the more accurate the multi-modal estimation. In the evaluation stage of the statistical models, the pixels in the input image are tested if there are consistent with the estimated model. The most well known statistical models are the eigenbackgrounds [42, 127], and the Single Gaussian (SG) [21, 181] and Mixture of Gaussians models (MGM) [56, 160].

The methods based on background subtraction are limited in more complicated scenarios. For example, if the foreground objects have similar colour to the background these objects cannot be detected by thresholding. Furthermore, these methods only adapt to slightly changing environmental conditions. Changes, like turning on the light cannot be captured by these models. In addition, these methods are limited to segment the whole object from the background, but for many tasks such as face recognition, gesture tracking, etc., this is not possible since specific parts need to be detected. Since most image sources (i.e. cameras) provide colour images we can use this additionally information in our model for the segmentation of the first image. This information can then be stored in the network structure and used to detect changes between consecutive frames.

### 4.2.1 Probabilistic Colour Models: Single Gaussian and Mixture of Gaussians

Image segmentation based on colour is a field studied by many researchers especially in applications of object tracking [20, 25, 109, 156] and human-machine interaction [16, 64, 182]. Also, a lot of research has been done in the field of skin-colour segmentation [88, 89, 135] since the human skin can create clusters in the colour space and thus be described by a multivariate normal distribution. First, we attempt to model skin-colour using a Single Gaussian distribution. With SG the model can be obtained via the maximum likelihood criterion which looks for the set of parameters (mean and covariance) that maximises the likelihood function. Figure 4.1 illustrates the SG model into different colour spaces (normalised rgb, HSV, CIE X,Y,Z, and CIE L*, a*, b*).

As can be seen in Figure 4.1, SGM model covers the entire area of the distribution for both skin and background. In some colour spaces the differentiation is greater, but the overlapping between skin and non-skin regions is sufficient to produce high FPR (False Positive Rate). It is evident that a SG distribution cannot model all possible variations in the skin-colour data. The existing approaches [21, 181] were extended by using Mixture of Gaussians [137, 185]. Below we summarise the steps involved in a MG skin-colour model.

- Firstly, the variance caused by the intensity is removed. This is achieved by normalizing the data or by transforming the original pixel values into a different colour space (e.g., rg colour-space [149] or HSV colour-space [135]).

- Secondly, a colour histogram is computed, which is used to estimate an initial mixture model.

- Finally, a Gaussian mixture model is estimated, which can efficiently be done by applying the iterative EM-algorithm [43]. A detailed discussion of the EM can be found in Appendix D.

The Gaussian Mixture Models obtained after $5$ EM iterations are shown in Figure 4.2. The results were obtained from a database containing approximately half

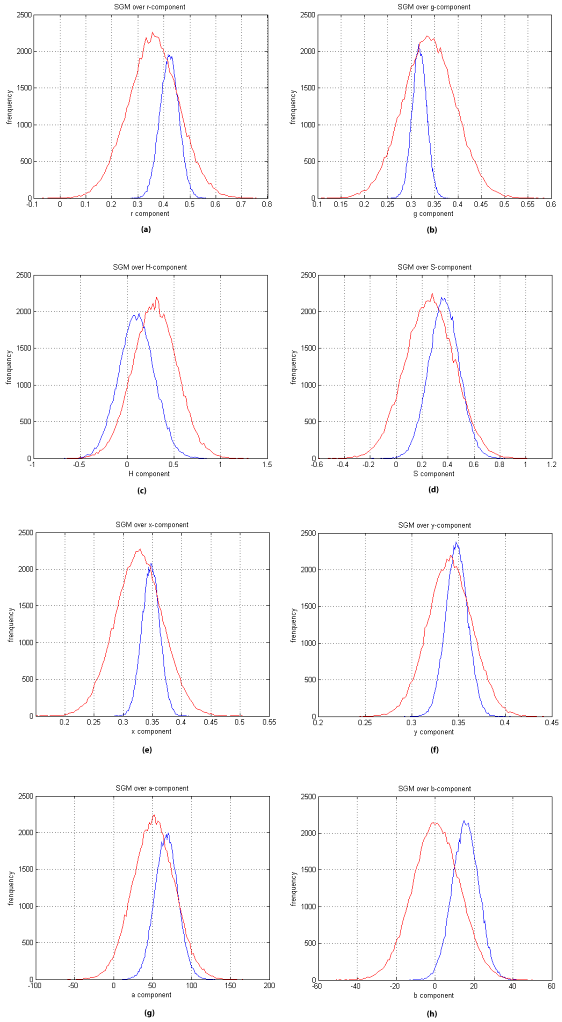


Figure 4.1: Blue line represents skin and red line represents background SGM. (a) Estimated SGM for r-component of normalised-rgb. (b) Estimated SGM for g-component of normalised-rgb. (c) Estimated SGM for H-component of HSV. (d) Estimated SGM for S-component of HSV. (e) Estimated SGM for x-component of CIE X,Y,Z. (f) Estimated SGM for y-component of CIE X,Y,Z. g) Estimated SGM for a-component of CIE L*, a*, b*. (h) Estimated SGM for b-component of CIE L*, a*, b*.

million pixels. Figure 4.3 shows the probability map for the skin colour which can then be used to initialise the network for the proposed learning algorithm.

In both probabilistic colour models the colour space used to represent the input image plays an important part in the segmentation. Levels of illumination are affected in a scene by changes in natural and artificial lighting present at different times. Also, some models are more perceptually uniform than others and some separate out information such as luminance and chrominance. We next discuss how we deal with these problems and a suitable colour space to work in. At each node of the network we experimented with the perceptually uniform colour model CIE L*, a*, b*, and the non perceptually uniform colour models, the normalised RGB, and CIE X,Y,Z (Figure 4.4). We also experimented with the HSV colour model (Figure 4.4), which separates the brightness component from the hue and the saturation, to compensate for changes in illumination. Unlike CIE L*, a*, b*, HSV is not perceptually mapped to the human visual system, meaning changes in colour values are not proportional to changes in the perceived significance of the change. A detailed discussion of the different colour models can be found in [89, 161, 173].

Given consideration of perceptual uniformity our best option is the CIE L*, a*, b* colour space. Thus, we converted the RGB values to the CIE L*, a*, b* values. The colour conversion from RGB to CIE L*, a*, b* is undergoing a linear conversion from RGB to CIE X,Y,Z, and a nonlinear conversion from CIE X,Y,Z to CIE L*, a*, b*.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.433910 & 0.376220 & 0.189860 \\ 0.212649 & 0.715169 & 0.072182 \\ 0.017756 & 0.109478 & 0.872915 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.1}$$

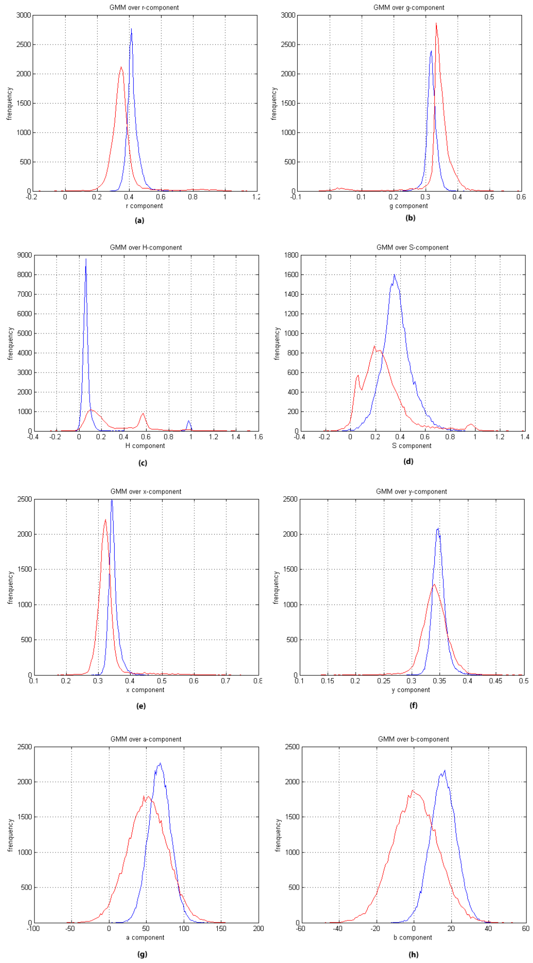


Figure 4.2: Blue line represents skin and red line represents background MGM. (a) Estimated MGM for r-component of normalised-rgb. (b) Estimated MGM for g-component of normalised-rgb. (c) Estimated MGM for H-component of HSV. (d) Estimated MGM for S-component of HSV. (e) Estimated MGM for x-component of CIE X,Y,Z. (f) Estimated MGM for y-component of CIE X,Y,Z. g) Estimated MGM for a-component of CIE L*, a*, b*. (h) Estimated MGM for b-component of CIE L*, a*, b*.

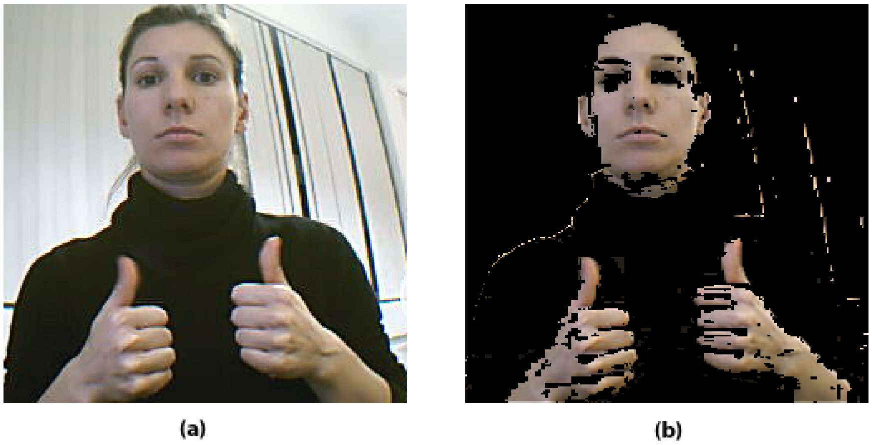


Figure 4.3: Image segmentation based on skin colour information. (a) original input image, and (b) probability map for the skin colour.

$$L^* = 116 * f\left[\frac{Y}{Y_n}\right] - 16$$

$$a^* = 500 * \left[f\left[\frac{X}{X_n}\right] - f\left[\frac{Y}{Y_n}\right]\right] \tag{4.2}$$

$$b^* = 200 * \left[f\left[\frac{Y}{Y_n}\right] - f\left[\frac{Z}{Z_n}\right]\right]$$

where

$$f(r) = \begin{cases} r^{\frac{1}{3}} & \text{if } r > 0.008856 \\ 7.7867 * r + \frac{16}{116} & \text{if } r \leq 0.008856 \end{cases} \tag{4.3}$$

The $X_n$, $Y_n$ and $Z_n$ refer to the CIE X,Y,Z values for a specified white point.

## 4.3 Automatic criterion for GNG

To get the learning process started, we need a simple but robust method to obtain the topology preserving graph with as minimum as possible user intervention. To achieve that we propose an automatic method for maximum node growth based

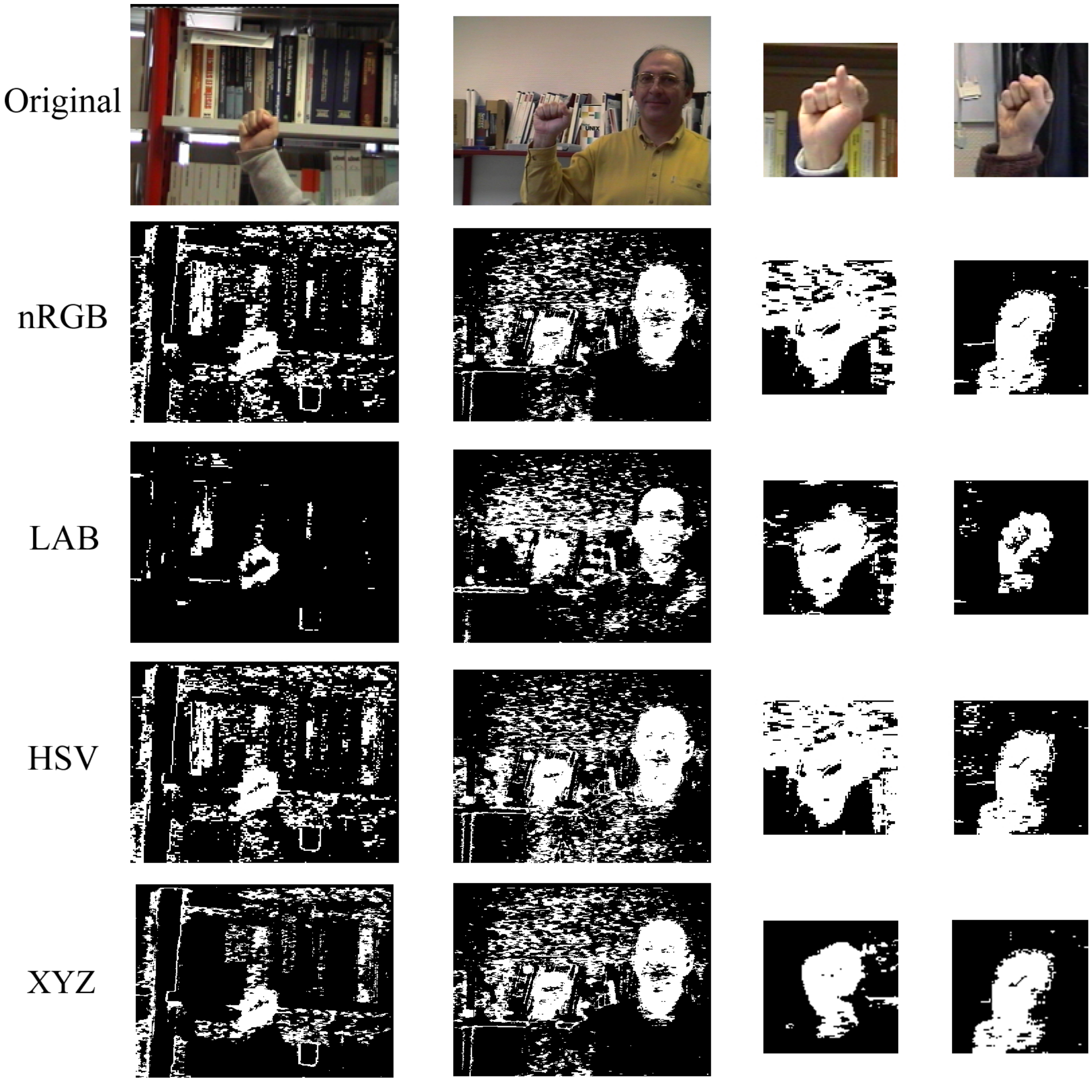


Figure 4.4: Skin colour images in cluttered backgrounds. Top row shows the original images followed by the various colour spaces.

on the likelihood of the generated nodes to sufficiently describe the topology of the objects of interest. In the following, we will discuss the method we have successfully applied to the existing GNG for maximum node growth.

In the GNG algorithm, the stopping criterion or network termination can either be applied by specifying a predefined number of nodes or by applying time constrains to the network. The critical point in both cases is that the quality of the network depends either on the arbitrary selection of the maximum number of nodes or the available time of the network to converge. For example, Figure 4.5 shows topology preservation for variant number of nodes of the same object. It is evident that 20 nodes are not enough to describe the topology of the object. However, the mapping with 101 nodes is good enough and has no difference, in terms of topology representation, with the map obtained inserting 181 nodes. In the latter case, 181 nodes is an example of overfitting which adds no further value to the recognition of the gesture.

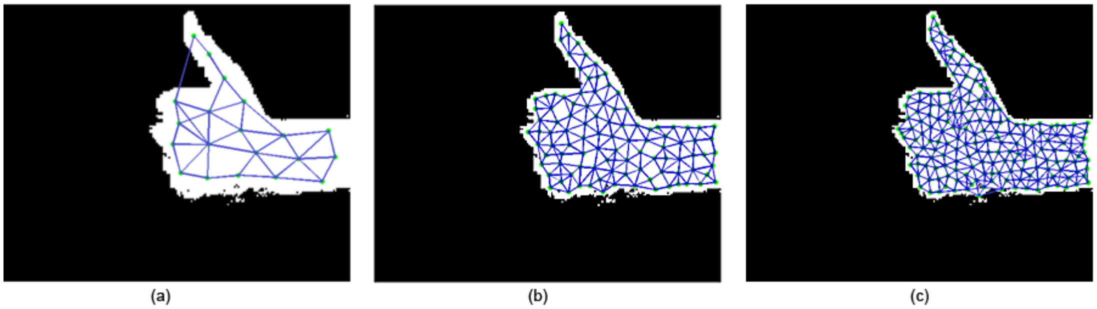


Figure 4.5: Mapping of the same object with image resolution $200 \times 160$ and network map of (a) 20, (b) 101 and (c) 181 nodes.

Figure 4.6 shows the GNG growth on different objects. Based on the type of the object and its parameters (e.g. size, more than one objects of interests or same object re-scaled) the size of the network should differ. In figure 4.6 the hand requires greater number of nodes to achieve topology preservation compared to the two squares and the circles. If the stopping criterion is left to the user, this can be an ill-conditioned task with human variation based on the observation, noise due to measurement, and a tedious task.

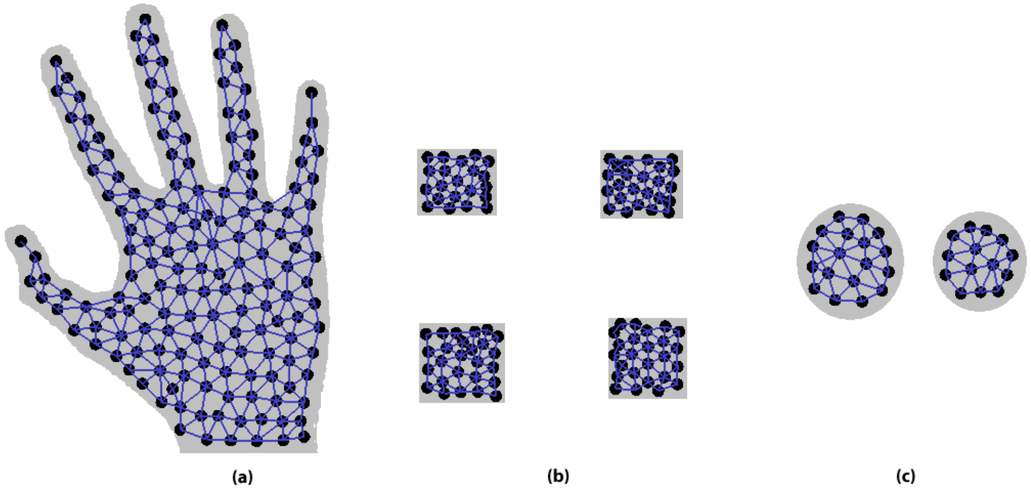


Figure 4.6: Representation of different objects with GNG. (a) hand is represented with $133$ nodes, (b) the four squares are defined with $92$ nodes, and (c) the two circles preserve topology with $32$ nodes.

To overcome this problem, we introduce an automatic method of the stopping criterion that defines the insertion of maximum nodes by calculating the image size and the probability of the objects in the image. In GNG and RGNG [134] the maximum number of nodes ($prenumnode$) to grow is set manually and chosen according to the scale of the clustering tasks. In our case, the maximum number of nodes is defined automatically by the system based on Equation ($4.4$). In our examples we use hand configurations and we model the colour distribution $p_{skin}$ of skin pixels by a Mixture of Gaussians in CIE L*, a*, b* space with mean and covariance estimated from hand-selected training patches. We assume that non-skin pixels have a uniform distribution $p_{bkgd}$.

Let $\Omega(x)$ denote the set of pixels in the objects of interest based on the configuration of $x$ (e.g. colour, texture, etc.) and $\Upsilon$ the set of all image pixels. The likelihood of the required number of nodes to describe the topology of an image $y$ is:

$$\begin{aligned} p(y|x) &= \{ \prod_{u\in\Omega(x)} p_{skin}(u) \prod_{v\in\Upsilon\backslash\Omega(x)} p_{bkgd}(v) \\ &\propto \prod_{u\in\Omega(x)} \frac{p_{skin}(u)}{p_{bkgd}(u)+p_{skin}(u)} \} * e_T \end{aligned} \tag{4.4}$$

and $e_T \leq \prod_{u\in\Omega(x)} p_{skin}(u) + \prod_{v\in\Upsilon\backslash\Omega(x)} p_{bkgd}(v)$. Figure 4.7 plots the likelihood node ratios for different images. $e_T$ is a similarity threshold and defines the accuracy

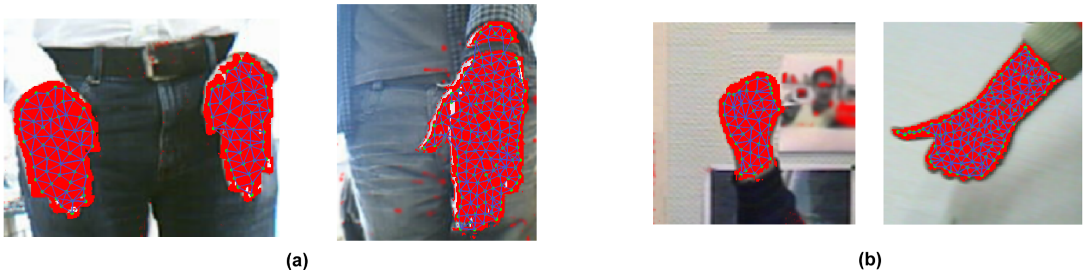


Figure 4.7: Likelihood node ratios for images with same image resolution but different skin to background ratio. (a) Network adaptation to images of $46,332$ pixels with maps of $102$ and $162$ nodes. (b) Network adaptation to images of $21,903$ pixels with maps of $46$ and $132$ nodes.

of the map. If $e_T$ is low the topology preservation is lost and more nodes need to be added. On the contrary, if $e_T$ is too big then nodes have to be removed so that Voronoï cells become wider. For example, let us consider an extreme case where the total size of the image is $I = 100$ pixels and only one pixel represents the object of interest. Let us suppose that we use $e_T = 100$ then the object can be represented by one node. In the case where $e_T \geq I$ then overfit occurs since twice as many nodes are provided. In our experiments the numerical value of $e_T$

ranges from $100 \leq e_T \leq 900$ and the accuracy depends on the size of the objects' distribution. The difference between choosing manually the maximum number of nodes and selecting $e_T$ as the similarity threshold, is the preservation of the object independently of scaling operations. The automatic criterion for the GNG algorithm is summarised in Algorithm 4.

**Algorithm 4** Stopping Criterion for Maximum Node Growth

```
Input: Segmented pixels $\Omega(x)$ from an unknown image I
Output: likelihood $p(y|x), TPG$
1. Obtain skin colour pixels $x$ by a Mixture of Gaussians in CIE L*, a*, b* space.
2. Set the $e_T$ value between $100 \leq e_T \leq 900$.
3. for every pixel $x$ do
     if $100 \leq e_T < 500$ and $e_T \geq 900$ then
       Set $e_T == 100$.
       Find the number of maximum prototypes.
       if Number of prototypes $x_c \leq 50$ then
         Increment $e_T$ until $x_c \geq 120$.
       end if
       if Number of prototypes $x_c > 200$ then
         Decrement $e_T$ until $120 \leq x_c < 200$.
       end if
     else
     $500 \leq e_T \leq 800$
     $x_c =$ maximum prototypes.
     end if
4. Let $p(y|x)$ by Equation 4.4 to find the $x_c$ most informative number of prototypes.
5. end for
```

### 4.3.1 On the number of Similarity Threshold

The determination of accurate topology preservation, requires the determination of best similarity threshold and best network map without overfitting. We can describe the optimum number of similarity thresholds, required for the accuracy of the map for different objects, as the unknown clusters $K$, and the network parameters as the mixture coefficients $W_K$, with $d$-dimensional means and covariances $\Theta_K$. To do that, we use a heuristic criterion from statistics known as the Minimum Description Length (MDL) [142, 175]. Such criteria take the general form of a prediction error, which consists of the difference of two terms:

$$E = modellikelihood - complexityterm \tag{4.5}$$

a likelihood term that measures the model fit and increases with the number of clusters, and a complexity term, used as penalty, that grows with the number of free parameters in the model. Thus, if the number of cluster is small we get a low value for the criterion because the model fit is low, while if the number of cluster is large we get a low value because the complexity term is large.

The information-criterion MDL of Rissanen [142], is defined as:

$$MDL(K) = -\ln[L(X|W_K, \Theta_K)] + \frac{1}{2}M\ln(N) \tag{4.6}$$

where

$$L(X|W_K, \Theta_K) = max\prod_{i=1}^{N} p(x_i|W_K, \Theta_K) \tag{4.7}$$

The first term $-\ln[L(X|W_K, \Theta_K)]$ measures the model probability with respect to the model parameter $W_K, \Theta_K$ defined for a Gaussian mixture by the mixture coefficients $W_K$ and $d$-dimensional means and covariances $\Theta_K$. The second term $\frac{1}{2}M\ln(N)$ measures the number of free parameters needed to encode the model and serves as a penalty for models that are too complex. $M$ describes the number of free parameters and is given for a Gaussian mixture by $M = 2dK + (K - 1)$ for $(K - 1)$ adjustable mixture weights and $2d$ parameters for $d$-dimensional means and diagonal covariance matrices.

The optimal number of similarity thresholds can be determined by applying the following iterative procedure:

- For all $K$, $(K_{min} < K < K_{max})$
  (a) Maximize the likelihood $L(X|W_K, \Theta_K)$ using the EM algorithm to cluster the nodes based on the similarity thresholds applied to the data set.
  (b) Calculate the value of MDL(K) according to Equations $4.6$ and $4.7$
- Select the model parameters $(W_K, \Theta_K)$ that corresponds to minimisation of the MDL(K) value.

Figure 4.8 shows the value of MDL(K) for clusters within the range of $(1 < K < 18)$ which correspond to the similarity thresholds $100 < e_T < 900$. We have doubled the range in the MDL(K) minimum and maximum values so we can represent the extreme cases of $1$ cluster which represents the whole data set, and $18$ clusters which over classify the distribution and corresponds to the overfiting of the network with similarity threshold $e_T = 900$. A global minimum and therefore optimal number of clusters can be determined for $K = 9$ which indicates that the best similarity threshold that defines the accuracy of the map without overfitting or underfitting the data set is $e_T = 500$. To account for susceptibility for the $EM$ cluster centres as part of the MDL(K) initialisation of the mixture coefficients the measure is averaged over $10$ runs and the minimal value for each configuration is selected.

The procedure is summarised in Algorithm 5.

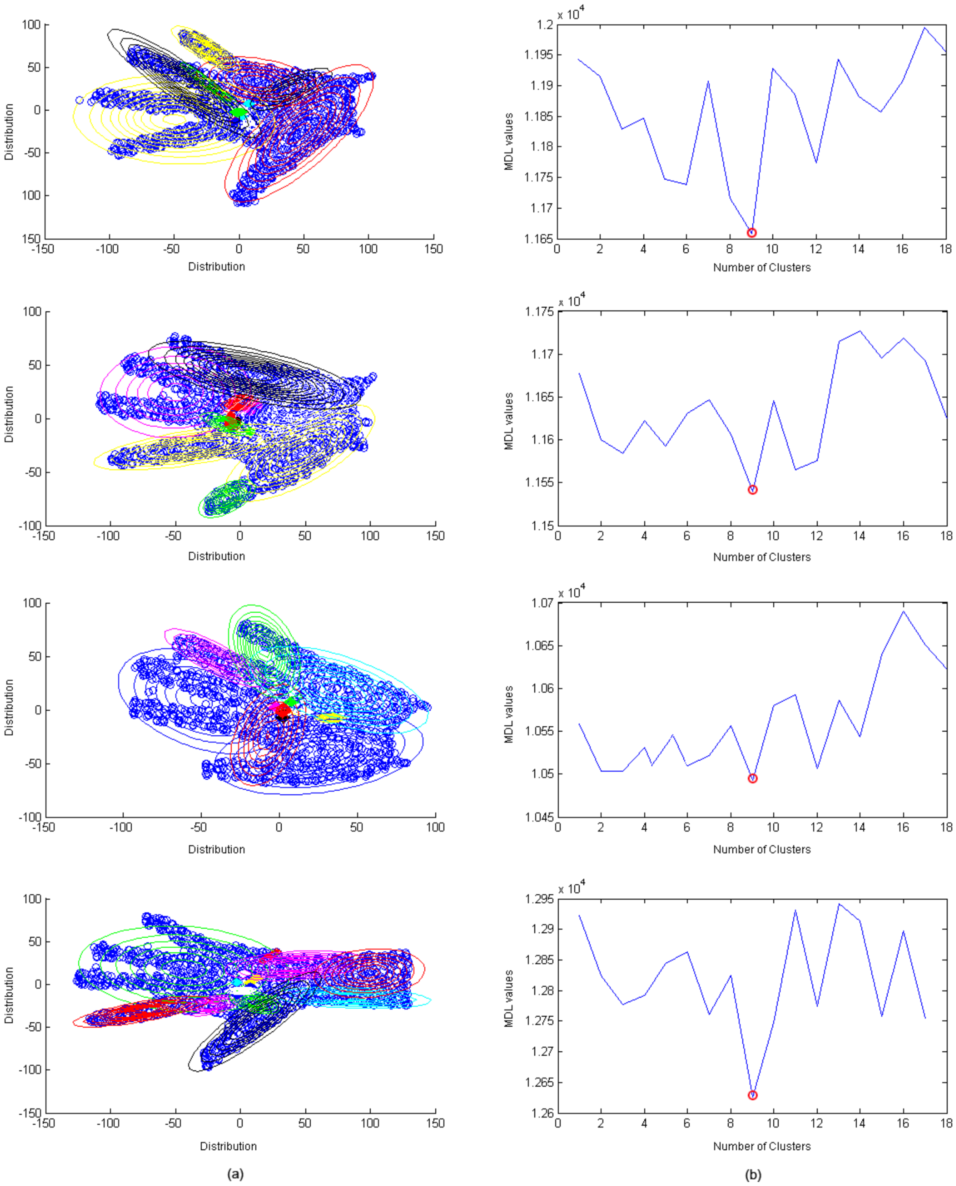


Figure 4.8: (a) Plot of hand distributions. (b) Plot of the MDL values versus the number of cluster centres. The Minimum Description Length MDL(K) is calculated for all cluster configurations with $(1 < K < 18)$ clusters, and a global minimum is determined at $9$ (circled point).

**Algorithm 5** MDL(K) Value

```
Input: $TPG$
Output: MDL(K)
Initialise $TPG$ (Algorithm 4)
while current number of prototypes $= x_c$ do
    Calculate MDL(K) according to Equations 4.6 and 4.7.
    Save position of all prototypes and average MDL(K) over 10 runs.
end while
```

## 4.4 Experiments

In this section we analyse and discuss the behaviour of the GNG network based on the MDL criterion. We show that the topology is best preserved with an optimal similarity threshold that maximises topology learning versus adaptation time as defined in Section 4.3.1. For that purpose, we created a benchmark data set with: (1) different gestures but similar image size; and (2) same object but scaled under linear transformations. The insertion of maximum nodes per object distribution and similarity threshold ($e_T$) determines the extent to which the topology mapping is preserved or not.

### 4.4.1 Benchmark Data

We tested our method on a data set of hand images recorded from 5 participants each performing a set of different gestures. A detailed description of the acquired gestures can be found in Section 5.3.1. To create this data set we have recorded images over several days and a simple webcam was used with image resolution $800 \times 600$. In total, we have recorded over 7500 frames, and for computational efficiency, we have resized the images from each set to $300 \times 225$, $200 \times 160$, $198 \times 234$, and $124 \times 123$ pixels. We obtained the data set from the University of Alicante, Spain and the University of Westminster, UK. Also, we tested our method with 49 images from Mikkel B. Stegmann[1] online data set. In total we have run the experiments on a data set of 174 images. We are interested in the topology preservation of the hands and the time spent when different number of nodes is generated from the similarity threshold (Figure 4.9). Since the background is unambiguous the network adapts without ocllusion reasoning. For our experiments only complete gesture sequences are included. There are no gestures with partial or complete occluded regions, which means that we do not model multiple objects that interact with the background. If in our data set we have gestures concealed by the background, we discard them from the modelling process.

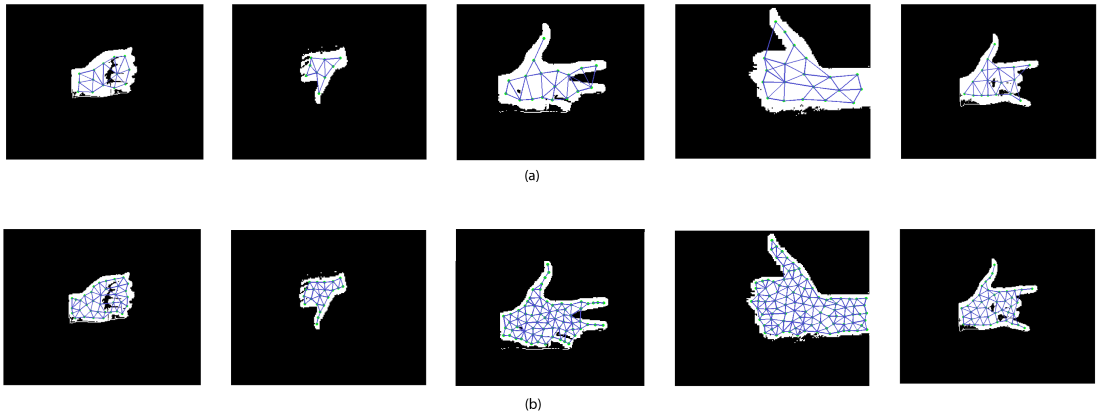


Figure 4.9: (a) Restricted topology of hand gestures with similarity threshold $e_T = 100$. (b) Improved topology of the gestures with optimal similarity threshold $e_T = 500$.

As with the constant parameters in Section 3.5.1, the parameters of the network

[1] http://www2.imm.dtu.dk/˜aam/

Table 4.1: Topology Preservation and Processing Time Using the Quantisation Error and the Topographic Product for Different Variants

| Variant | Number of Nodes | Time (sec) | QE | TP |
|---|---|---|---|---|
| GNG$_{\lambda=100,K=1}$ | 23 | 0.22 | 8.932453 | 0.4349 |
| GNG$_{\lambda=100,K=9}$ | 122 | 0.50 | 5.393949 | -0.3502 |
| GNG$_{\lambda=100,K=18}$ | 168 | 0.84 | 5.916987 | -0.0303 |
| GNG$_{\lambda=300,K=1}$ | 23 | 0.90 | 8.024549 | 0.5402 |
| GNG$_{\lambda=300,K=9}$ | 122 | 2.16 | 5.398938 | 0.1493 |
| GNG$_{\lambda=300,K=18}$ | 168 | 4.25 | 4.610572 | 0.1940 |
| GNG$_{\lambda=600,K=1}$ | 23 | 1.13 | 0.182912 | -0.0022 |
| GNG$_{\lambda=600,K=9}$ | 122 | 2.22 | 0.172442 | 0.3031 |
| GNG$_{\lambda=600,K=18}$ | 168 | 8.30 | 0.169140 | -0.0007 |
| GNG$_{\lambda=1000,K=1}$ | 23 | 1.00 | 0.188439 | 0.0750 |
| GNG$_{\lambda=1000,K=9}$ | 122 | 12.02 | 0.155153 | 0.0319 |
| GNG$_{\lambda=1000,K=18}$ | 168 | 40.98 | 0.161717 | 0.0111 |

are as follows: $\lambda = 100$ to $1000$, $\epsilon_x = 0.1$, $\epsilon_n = 0.005$, $\Delta x_{s_1} = 0.5$, $\Delta x_i = 0.0005$, $\alpha_{max} = 125$. For the MDL(K) value we have experimented with cluster centres within the range of $1 < K < 18$.

Table 4.1 shows topology preservation, execution time, and number of nodes when different variants in the $\lambda$ and the $K$ are applied in a hand as the input space. Faster variants get worse topology preservation but the network converges quickly. However, the representation is sufficient and can be used in situations where minimum time is required like online learning for detecting obstacles in robotics where you can obtain a rough representation of the object of interest in a given time and with minimum quality.

### 4.4.2 Test performance of topology preservation: different object shapes

The test consists of two processes: learning and evaluation. During learning, we choose different similarity thresholds taken from a threshold vector set, then input the results to the GNG, and report number of nodes per similarity threshold, computational time of the network, and Mean Squared Error (MSE) as the results of the adaptation. In the evaluation process, we measure the topology preservation with the topographic product.

The termination of the network depends on the efficient selection of the similarity threshold $e_T$ which ranges from ($100 \leq e_T \leq 900$) and the likelihood that the set of pixels belonging to the object's of interest are above this similarity threshold. It is worth noting that the network stabilises and can represent sufficiently the object when $e_T = 500$. This is an optimum number obtained by MDL(K) that maximises topology learning versus adaptation time and MSE (Table 4.2). Figure 4.10 shows the plots of the MDL(K) values versus the number of clusters for minimum, maximum and approximate match similarity threshold ($e_T = 100$, $e_T = 900$, and $e_T = 500$). As the similarity threshold increases the optimum number for the MDL(K) values increases as well with an optimum growth at $K = 9$. Figure 4.11 shows the topology preserving graphs for different similarity thresholds. As the number of nodes increases the system recognises better the gesture.

Table 4.3 shows the topographic product for a number of nodes. A qualitative representation of the topographic product is given in Figure 4.12. We can see that the insertion of more nodes as $e_T$ increases makes no difference to the object's topology. The graph shows the Topographic Product (TP) for four different hand postures and nodes varying from $26$ to $230$. The topology is best preserved

Table 4.2: Execution time and performance for various number of nodes

| Data sets | | Number of nodes | Time (sec) | MSE |
|---|---|---|---|---|
| Set$_{300x225}$ | H$_{e_{T=100}}$ | 24 | 1.63 | 5.83 |
| | H$_{e_{T=200}}$ | 51 | 4.22 | 2.10 |
| | H$_{e_{T=300}}$ | 77 | 9.80 | 1.33 |
| | H$_{e_{T=400}}$ | 102 | 15.60 | 0.75 |
| | **H$_{e_{T=500}}$** | **124** | **22.32** | **0.55** |
| | H$_{e_{T=600}}$ | 153 | 24.14 | 0.81 |
| | H$_{e_{T=700}}$ | 179 | 33.98 | 0.74 |
| | H$_{e_{T=800}}$ | 205 | 51.15 | 0.49 |
| | H$_{e_{T=900}}$ | 230 | 72.05 | 0.33 |
| Set$_{200x160}$ | H$_{e_{T=100}}$ | 23 | 1.00 | 12.06 |
| | H$_{e_{T=200}}$ | 37 | 2.48 | 4.56 |
| | H$_{e_{T=300}}$ | 56 | 4.77 | 2.42 |
| | H$_{e_{T=400}}$ | 75 | 8.04 | 1.58 |
| | **H$_{e_{T=500}}$** | **112** | **12.02** | **0.98** |
| | H$_{e_{T=600}}$ | 122 | 17.65 | 0.87 |
| | H$_{e_{T=700}}$ | 131 | 24.64 | 0.68 |
| | H$_{e_{T=800}}$ | 150 | 32.19 | 0.56 |
| | H$_{e_{T=900}}$ | 168 | 40.98 | 0.47 |
| Set$_{198x234}$ | H$_{e_{T=100}}$ | 26 | 1.80 | 5.15 |
| | H$_{e_{T=200}}$ | 54 | 6.19 | 1.82 |
| | H$_{e_{T=300}}$ | 81 | 11.16 | 0.99 |
| | H$_{e_{T=400}}$ | 108 | 17.59 | 0.64 |
| | **H$_{e_{T=500}}$** | **131** | **27.83** | **0.46** |
| | H$_{e_{T=600}}$ | 162 | 38.37 | 0.35 |
| | H$_{e_{T=700}}$ | 190 | 57.15 | 0.27 |
| | H$_{e_{T=800}}$ | 217 | 75.62 | 0.23 |
| | H$_{e_{T=900}}$ | 244 | 103.20 | 0.19 |
| Set$_{124x123}$ | H$_{e_{T=100}}$ | 22 | 1.15 | 13.26 |
| | H$_{e_{T=200}}$ | 44 | 3.20 | 4.67 |
| | H$_{e_{T=300}}$ | 66 | 6.27 | 2.55 |
| | H$_{e_{T=400}}$ | 88 | 10.58 | 1.66 |
| | **H$_{e_{T=500}}$** | **125** | **15.92** | **0.90** |
| | H$_{e_{T=600}}$ | 132 | 24.14 | 0.81 |
| | H$_{e_{T=700}}$ | 154 | 32.98 | 0.72 |
| | H$_{e_{T=800}}$ | 176 | 43.95 | 0.59 |
| | H$_{e_{T=900}}$ | 198 | 57.85 | 0.49 |

Table 4.2, shows the number of nodes for different similarity thresholds on different data sets. The optimal similarity threshold that maximises topology learning versus adaptation time and MSE is $e_T = 500$.

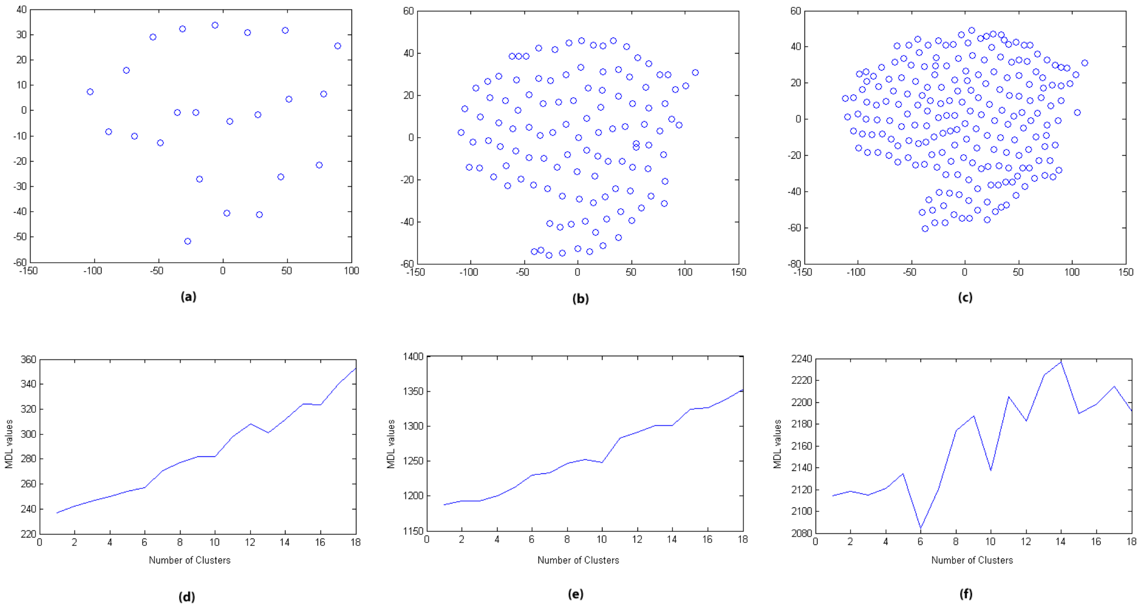


Figure 4.10: (a), (b), and (c) Plot of data set for similarity thresholds $e_T = 100$, $e_T = 500$, and $e_T = 900$. (d), (e), and (f) Plot of the MDL(K) values versus the number of clusters centres generated by the similarity thresholds during the growth process of the GNG.

with maps containing enough nodes ($> 100$) to represent the topology but without overfitting the network ($< 155$) while fewer nodes ($< 60$) are not enough to the recognition of the gesture. Furthermore, the more nodes added during the learning process the more time it takes for the network to grow (Figure 4.13).

### 4.4.3 Test performance of topology preservation: scaled image

Figure 4.14 presents the adaptation with a network map of $72$ nodes to the same object (class probability of pixels belonging to the objects of interest $P(O)$ is set to reflect the size of objects in an image and be $1 - P(B)$ the background probability) but with different image size. The image has been re-scaled by half the size of the

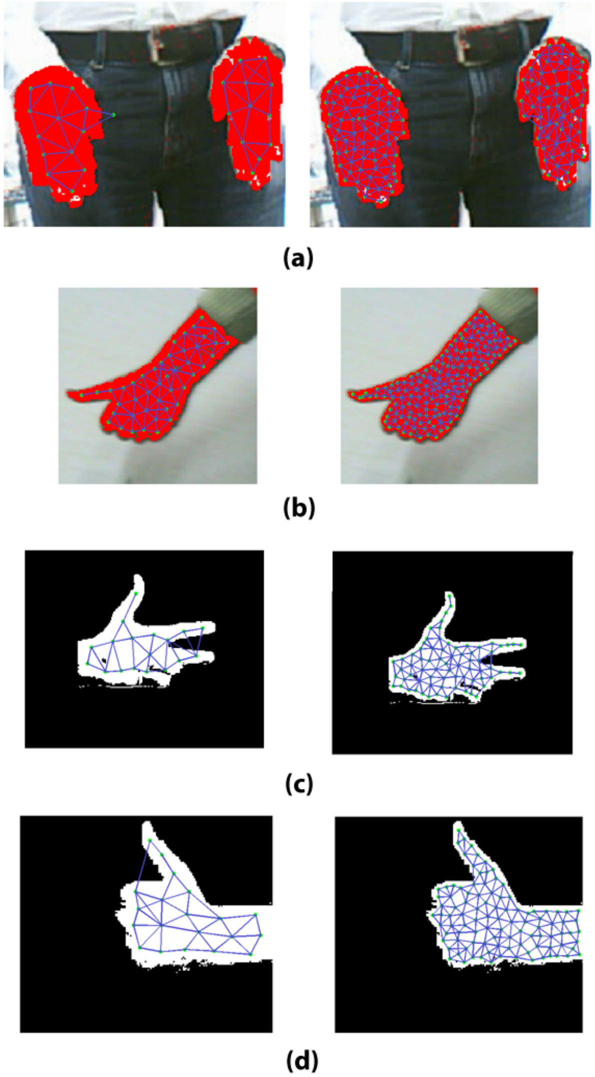


Figure 4.11: Topology preserving graphs with similarity thresholds $e_T = 100$ and $e_T = 500$. (a) network is split into two clusters and the maximum number of nodes is $26$ (left image) and $131$ (right image). (b), (c), and (d) single gestures with maximum number of nodes varying between $23$ to $125$.

original image resolution. In both cases the adaptation is correct and the topology is preserved independently of the scaling of the image. With existing GNG

Table 4.3: The Topographic product for various data sets

| Image (a) | | Image (b) | | Image (c) | | Image (d) | |
|---|---|---|---|---|---|---|---|
| Nodes | TP | Nodes | TP | Nodes | TP | Nodes | TP |
| 26 | -0.0301623 | 26 | -0.021127 | 24 | -0.017626 | 19 | -0.006573 |
| 51 | -0.030553 | 51 | -0.021127 | 47 | -0.047098 | 37 | -0.007731 |
| 77 | 0.04862 | 77 | 0.044698 | 71 | 0.046636 | 56 | 0.027792 |
| 102 | 0.048256 | 102 | 0.021688 | 95 | 0.017768 | 75 | 0.017573 |
| 128 | 0.031592 | 128 | 0.011657 | 119 | 0.014589 | 94 | 0.018789 |
| 153 | 0.038033 | 153 | 0.021783 | 142 | 0.018929 | 112 | 0.016604 |
| 179 | 0.047636 | 179 | 0.017223 | 166 | 0.017465 | 131 | 0.017755 |
| 205 | 0.038104 | 205 | -0.013525 | 190 | 0.017718 | 150 | 0.007332 |
| 230 | 0.037321 | 230 | 0.017496 | 214 | -0.007543 | 168 | 0.007575 |

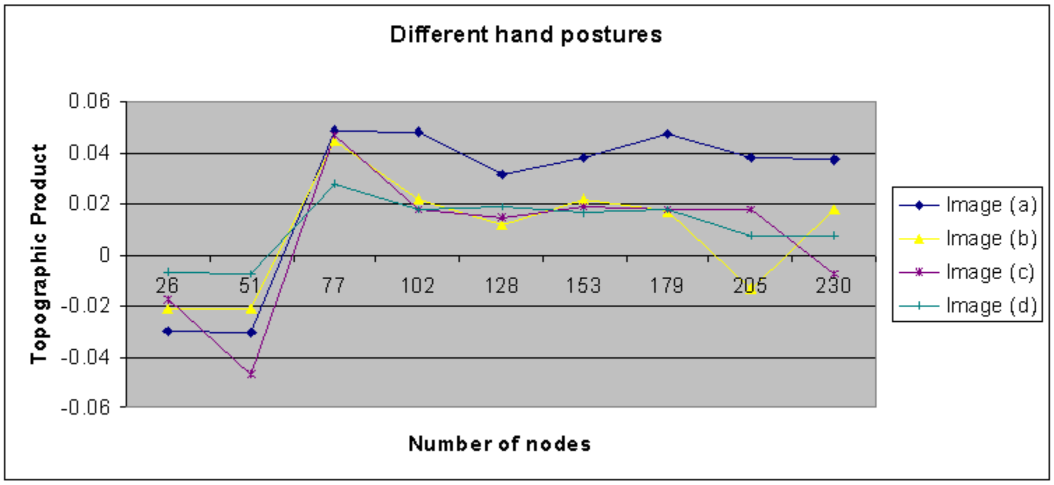


Figure 4.12: Comparative study for various hand postures. In all data sets the approximate match is achieved with $e_T = 500$ where $P \approx 0$. $P < 0$ indicates low match while $P > 0$ indicates a high match between the input and the latent space.

where the network size needs to be defined *a priori* the adaptation for smaller images can become excessive and fewer nodes can be used to define the condition of finalisation.

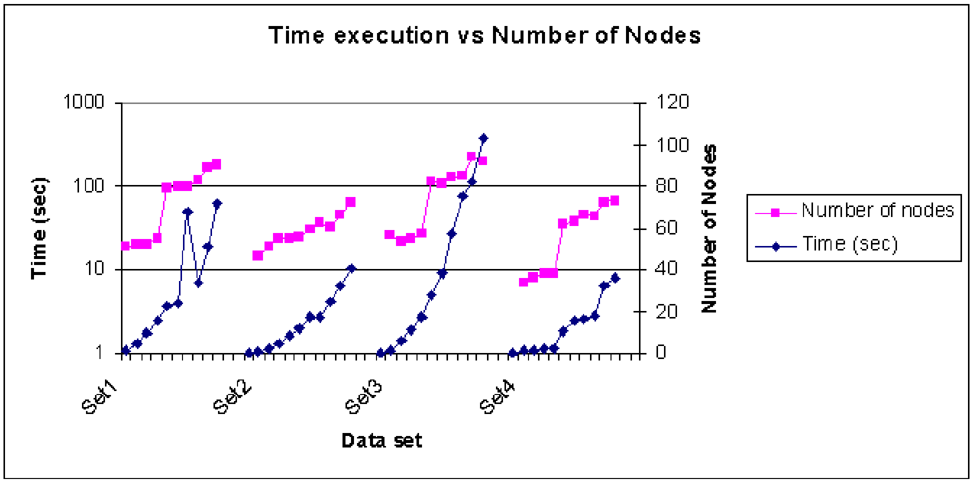


Figure 4.13: Time taken to insert the maximum number of nodes per data set.

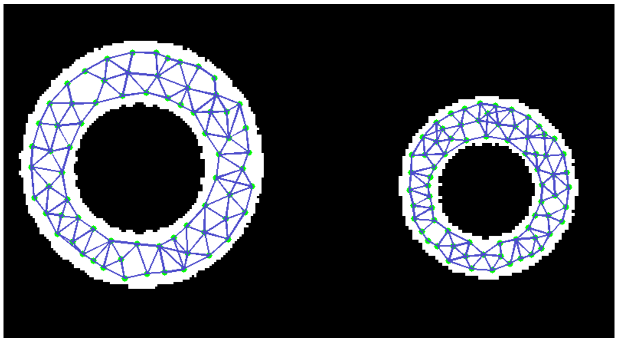


Figure 4.14: Network adaptation to images sizes of $148 \times 186$ (left image) and $74 \times 93$ (right image) pixels with a network map of $72$ nodes.

## 4.5 Summary

Based on the capabilities of GNG to readjust to new input patterns without restarting the learning process, we developed an approach to minimise the user intervention by utilising an automatic criterion for maximum node growth. This automatic criterion for GNG is based on the object's distribution and the similarity threshold

($e_T$) which determines the preservation of the topology. The model is then used to the representation of motion in image sequences by initialising a suitable segmentation. During testing we found that for different shapes there exists an optimum number that maximises topology learning versus adaptation time and MSE. This optimal number uses knowledge obtained from information-theoretic considerations.

# Chapter 5

# Automatic Gesture Model Acquisition and Neural Maps for Motion

*In this chapter we introduce and discuss the Active-GNG. The main idea is to use the capabilities of the GNG and extend it so shapes can re-deform locally if common regions are found. We achieve that by adding properties to the network like restricted movement of the nodes based on local changes in the shape, distance vector between the $1st$ frame and any successive $k$ frames, and the probability of the node to belong to the skin distribution. Applying these updated rules an incrementally better learning algorithm is obtained.*

## 5.1 Introduction

To motivate the representation of motion in image sequences with growing neural models we first discuss the main limitations of existing methods. When using shape or feature information or combination of the two to segment and track non-rigid objects in video sequences, the most effective models are the Active Contour Models (Snakes) introduced by Kass *et al.* [92] and their extensions [102, 183], the Point Distribution Models (PDMs) and the Active Shape Models (ASMs) introduced by Cootes and Taylor [32], and the Active Appearance Models (AAMs) [50]. 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 or preserve the physical attributes such as geometry, topological relations, etc.

- 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 PDM, which is the shape descriptor for both the ASM and the AAM models, 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. However, as with the snakes the deformation of the model adheres to global shape transformations.

In the above models either a training set is required with correct annotation, otherwise the model will not converge, or the models deform globally without taking into consideration common regions in the shape or texture of the image. 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 [60], 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 5.1). 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 5.2).

Figure 5.1 shows the local adaptation of the proposed method. The adaptation of the network and the new topology preserving graph is achieved only to the changes between the two shapes. The rest of the network stays unchanged. With our method we overcome common restrictions in motion analysis like stiffness, where the distance between points of objects does not change along the sequence, and the assumption that movement is assumed common for all the points or regions of the object.

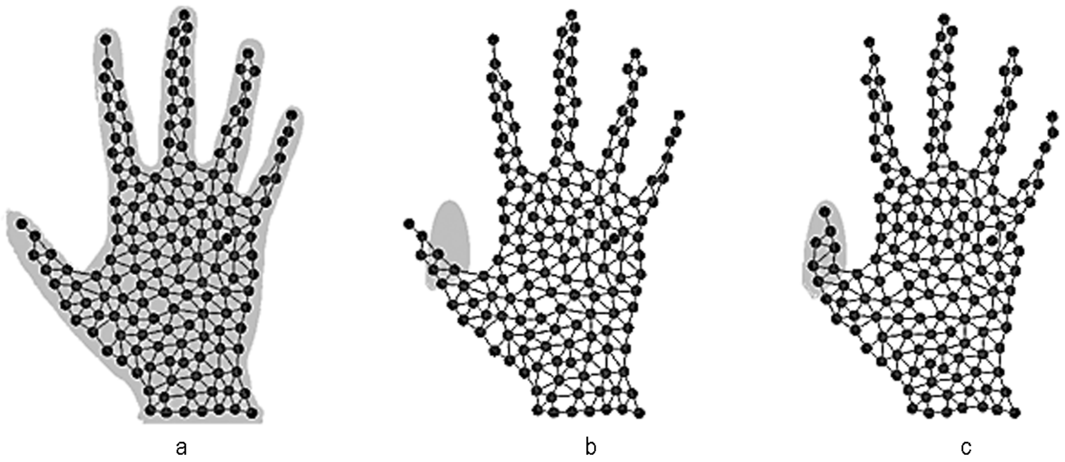


Figure 5.1: Example of $2D$ local adaptation with $A$-GNG. (a) Original shape. (b) Signals are generated only to the new input distribution while the rest of the topology of the network remains unchanged. (c) The winner node and its direct topological neighbours update their positions.

Figure 5.2 shows the best matching node denoted by the distance and the feature vector. Since we are interested in obtaining the geometry of the objects without using *a priori* knowledge such as expected size, shape and appearance encoded in the model from a training set, we compare our model with the neural growing models NG and GNG. The advantage of our model is that it is unsupervised and can be used for automatic model building.

The rest of this chapter is organised as follows. Section 5.2 introduces our approach to follow a non-stationary distribution by calculating the differences between successive frames and re-adjusting the nodes that belong to those differences. In Section 5.3 we apply it to track gestures based on the *Active*-GNG model representation before conclusions are drawn in Section 5.4.

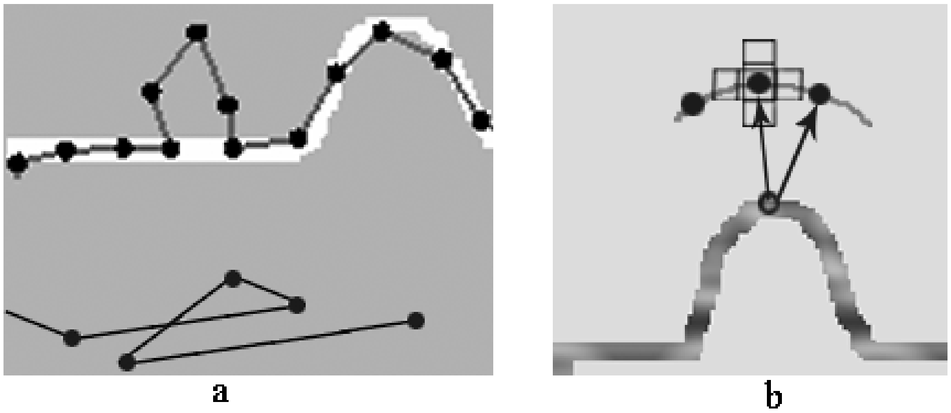


Figure 5.2: The upper part of Image $a$ shows the convergence of the GNG algorithm to a local minimum. The top node with its direct neighbours can never be winners. The lower part of Image $a$ shows the fold-over that will occur after a number of iterations. Not only point correspondences are lost but also topology relations are violated. To overcome this problem for each node we compute a $2k + 1$ dimensional feature vector which encapsulates feature 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.

## 5.2 *Active*-Growing Neural Gas (*A*-GNG)

To tackle the problems of GNG we propose *Active*-GNG, which features the same network characteristics as GNG but with adding properties such as local adaptation of the nodes, a translation vector between the $1st$ frame and any successive $k$ frames, and the probability of the node to belong to the skin distribution.

The main extensions of *A*-GNG are summarised as follows:

1. The correspondence of the nodes is performed locally, so the model re-deforms only where differences in the input distribution between successive images exist. In the GNG example (Figure 5.3$_{GNG}$ and Figure 5.4) the topology is lost

and the network collapses since the winner node $x_\nu$ and its direct topological neighbours $x_c$ move towards the input signal $\xi_w$ without any stopping criterion being applied to the nodes to prevent the network from shrinking. In contrast, in $A$-GNG (Figure 5.3$_{A-GNG}$ and Figure 5.5) the topology is preserved since the updating rule $\Delta x_\nu$ and $\Delta x_c$ is performed only to the nodes where changes in the input distribution have occurred. In the adaptation

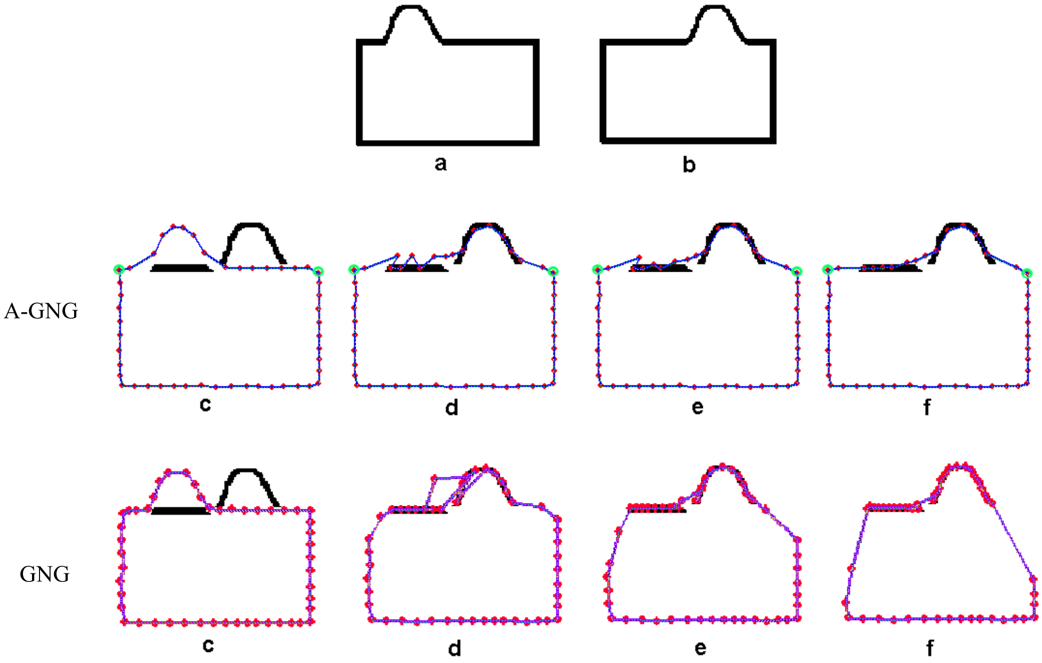


Figure 5.3: Example of tracking a bump model using $A$-GNG and GNG. Sequences $a$ and $b$ represent the original bump model used for tracking. $c$ - $f$ show the local and global convergence of the nodes with $A$-GNG and GNG respectively. With A-GNG correspondences are kept (circled corners) and only nodes closest to the new distribution $Y$ re-adjust their position. In GNG correspondences are lost and the network gradually moves all the nodes to the new distribution $Y$.

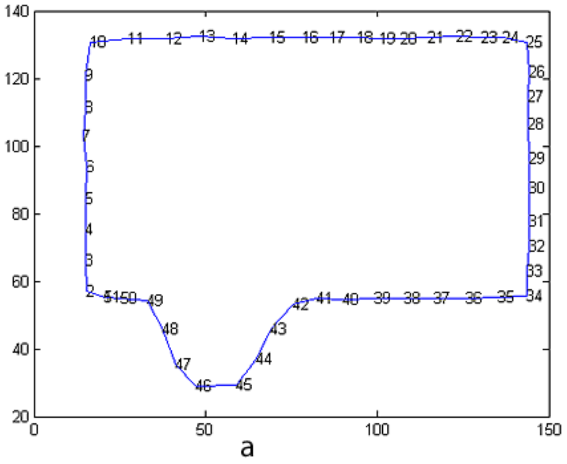


Figure 5.4: Image $a$ shows the original map of the bump model. Image $b$ shows the fold-overs that occur after a number of iterations. Fold-overs of the network occur between points $40$ and $41$ and between $46$ and $47$. Not only point correspondences are lost, but also topology relations are violated.

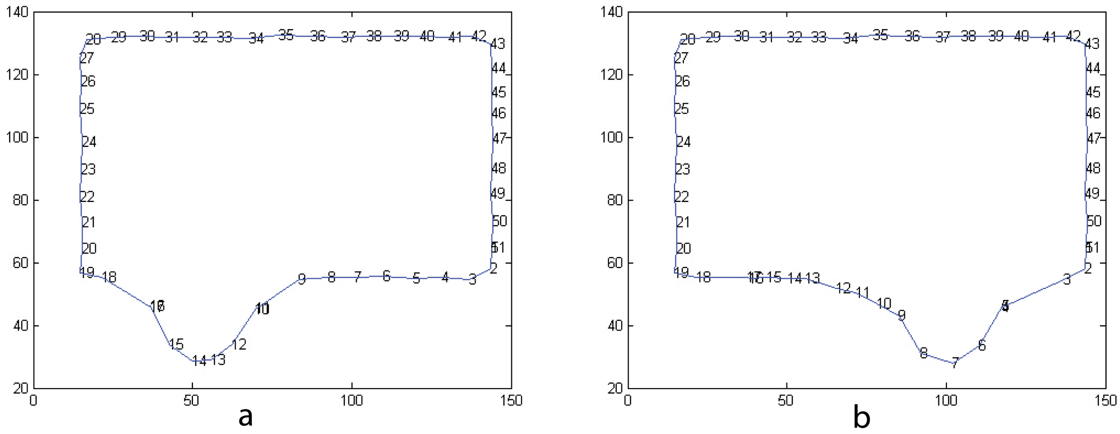


Figure 5.5: Image $a$ shows the original map. In image $b$ only nodes $10$ to $17$ readapt since they are the closest to the new input distribution. The rest of the map remains constant.

process the following action is performed. Let $N$ be the set of nodes from the original input distribution $W$ and $M < N$ the reduced set of nodes from the new distribution $Y$. For every generated input signal $\xi_y$ drawn from the random vector $Y$, if the winner node and its direct topological neighbours $(x_\nu, x_c) \in W - Y$ then move the nodes towards the signal and update their position. If not, then $(x_\nu, x_c) \in W \cap Y$ and no movement is performed (Figure 5.6). By applying this restriction the network keeps its original topology structure.

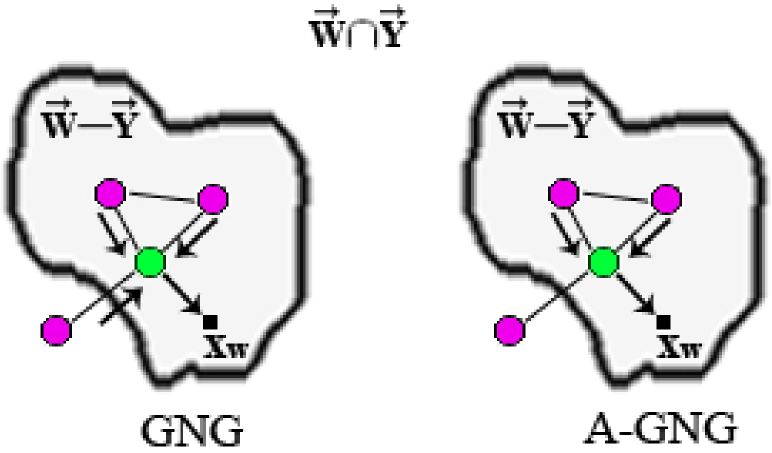


Figure 5.6: In GNG all the nodes are moving (winner and direct topological neighbours) while in $A$-GNG only the nodes that belong to $W - Y$ which is the difference between the original and the new distribution. The rest of the nodes, even if there are direct neighbours to winner remain stationary.

2. In order to track a non-stationary distribution, we first calculate the mean vectors $x_i$ and $x_j$ of the $i_{th}$ and the $j_{th}$ frame, and then we translate the network $\|x_i - x_j\|$ distance. The network readapts its position by examining a

region of the image around each node to determine a better displacement of the nodes. Calculating the translation vector is very important when a rapidly moving distribution occurs. Figure 5.7 illustrates the ability of *A*-GNG to track a jumping distribution due to the calculation of the distance vector between the original and the rapidly moving distribution. In contrast to GNG (Figure 5.8) which fails to readapt to the non-stationary input distribution and the nodes become *dead nodes*.

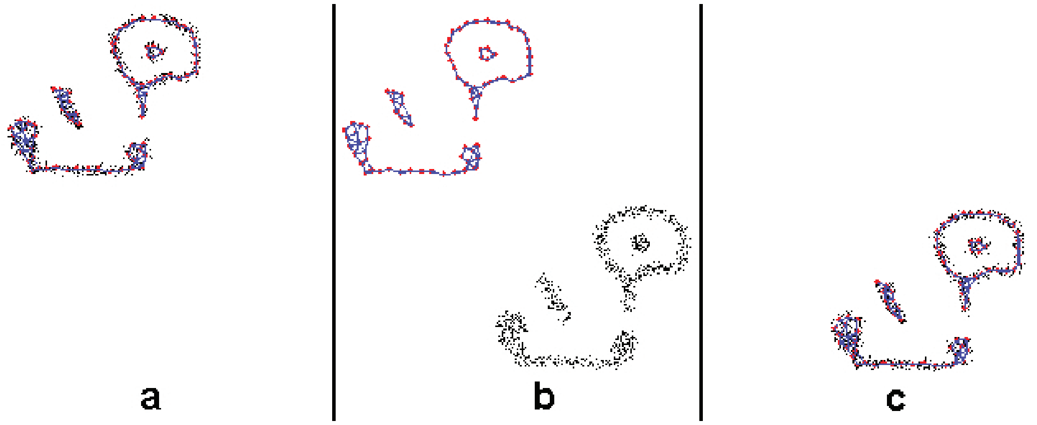


Figure 5.7: *A*-GNG adaptation and tracking to a non-stationary discrete distribution. Sequences $a$ - $c$ represent the original and the final state of the nodes adaptation to the non-stationary distribution.

3. In order to track a hand gesture in a cluttered background we add a new component in the vector feature with skin colour information taken at each node. The skin dataset is a combination of our own dataset from the University of Westminster, and the dataset from the University of Alicante. Skin samples were collected from images taken under varying lighting conditions and from various parts of the hand. The skin domain mainly contains white

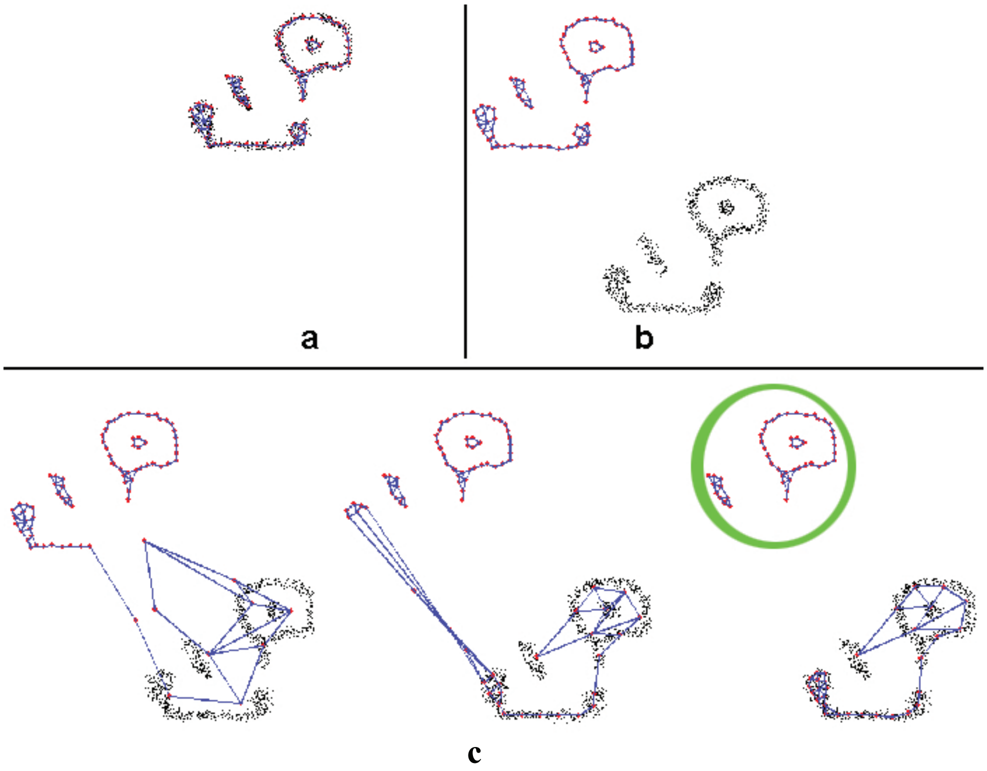


Figure 5.8: Incorrect adaptation and tracking to a moving distribution by GNG. Sequences $a$ and $b$ represent the original states of the distribution as with $A$-GNG (Figure 5.7). Sequence $c$ represents the in-between states of the nodes adaptation before convergence, and shows how the nodes become inactive (circled network), and fail to readapt their position to the moving distribution. The topology is violated and the resources of the network are wasted.

skin samples from European and Asian origins. The skin distribution derives from the standard technique of modelling a distribution using a mixture of $K$ Gaussians [15]. If a node is represented as $(x, y, P(g(x, y)))$, where $x, y$ are positions, $g(x, y)$ colour at $x, y$ and $P(g(x, y))$ posterior probability of that gray level, the strength of the node can then depend on the value of the posterior probability (Figure 5.9).

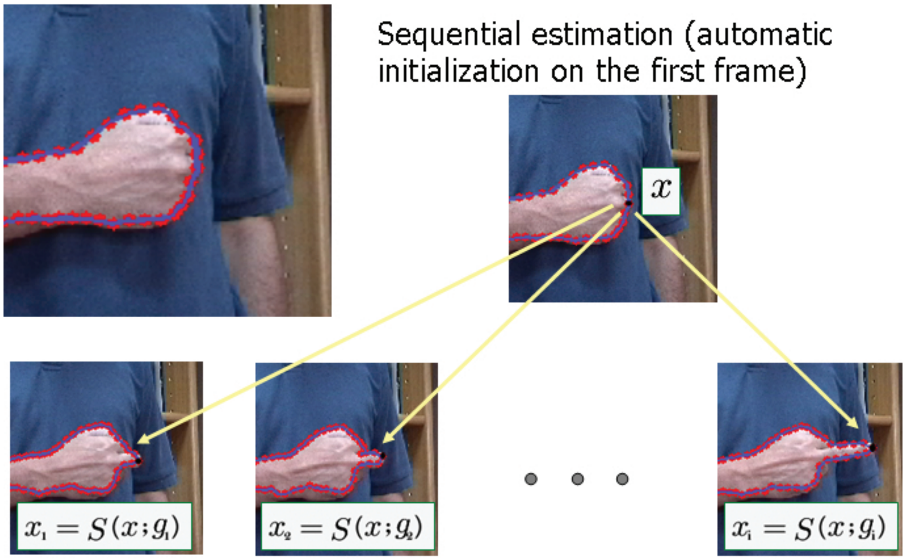

Sequential estimation (automatic initialization on the first frame)
$x$
$x_1 = S(x;g_1)$
$x_2 = S(x;g_2)$
$x_i = S(x;g_i)$

Figure 5.9: Network convergence after a sequence of $k$ frames. The network is defined by the shape S$(x; P(g(x, y)))$ and the movement of the nodes depend on the posterior probability $P(g(x, y))$. The highest the probability of a node to belong to the skin prior probability the faster the node will re-adjust its position to the new input distribution.

The main steps of the algorithm are as follows:

1. For every node calculate the probability of belonging to the skin Gaussian probability density function by using the Bayes's theorem:

$$p(k|x) = \frac{p(k)p(x|k)}{p(x)} = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j)} \tag{5.1}$$

where $\mu$ and $\Sigma$ represent the mean and the covariance of the $k$th Gaussian. The total shape then is given as S$(x; P(g(x, y)))$ where $x$ is a $2n(x, y)$ node vector and $P(g(x, y))$ is the posterior probability of the node at $g(x, y)$. The information stored in the $1st$ and any subsequent frames is added to the $TPG$ map and can be used for the learning in a sequence of $k$ frames. The segmented frame and the stored colour information in each node is given by:

$$S(x; P(g(x,y);t) = p(k|x) \propto P(g(x,y), t-1), TPG_{t-1} \tag{5.2}$$

2. Given $\mid N \mid$ number of nodes calculate the mean vector $\overline{x}$ of the network, where

$$\overline{x} = \frac{1}{\mid N \mid} \sum_{i=1}^{|N|} x_i \tag{5.3}$$

3. Calculate the image difference between the $i_{th}$ and the $j_{th}$ frame. Let $W$ and $Y$ be two sets in $\mathbb{R}^n$ representing the original and the new distribution. The Minkowski subtraction of $Y$ from $W$ is defined as:

$$W - Y = \bigcap_{y \in Y} W_y \tag{5.4}$$

   where $y$ are the pixel coordinates of the successive frame.

4. Let $C = W - Y$ be the new input distribution of the network.

5. Randomly generate input signals $\xi_w$ to $C$ and calculate as in step 2 the mean vector for the $j_{th}$ frame.

6. Calculate the distance vector of the two means and swift the nodes towards $C$. For each successive frame we calculate its deviation $\Delta x$ from the mean where

$$\Delta x = \overline{x_i} - \overline{x_j} \tag{5.5}$$

7. Randomly generate input signals $\xi_w$ to $C$ and find the winner node $x_\nu$ and its direct topological neighbours $x_c$.

8. For every frame, update the feature matrix $P(g(x,y))$ that underline each node.

9. For every frame, update the position of the nodes by moving them towards the current signal by the weighted factors $\epsilon_x$ and $\epsilon_n$ same as in GNG.

10. Remove the used signal $\xi_w$ from the input distribution.

11. Repeat iterations $1 - 10$ until the system converges.

As with the constant parameters in Section 3.5.1, the parameters for the *A*-GNG algorithm should not be set too high or too low, if topology preservation is required from the network. Experimenting with these values has lead to the following boundary values: $\lambda = 100$ to $1000$, $e_T = 500$, $\epsilon_x = 0.1$, $\epsilon_n = 0.005$, $\Delta x_{s_1} = 0.5$, $\Delta x_i = 0.0005$, $\alpha_{max} = 125$, $k = 5$.

## 5.3 Experiments

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. The performance of the network is compared using the benchmark models, hand and bump.

### 5.3.1 Hands

In our experiments, eight gestures (Figure 5.10) that frequently appear in sign language were used as examples to testify our system performance. The gestures were obtained from the University of Alicante, Spain and the University of Westminster, UK using a simple webcam with image resolution $800 \times 600$. In total, we have recorded over 12000 frames. In order to use a clean edge map that serves as the distribution for the algorithm, we have performed all the gestures in front of a low to medium cluttered background avoiding extremely cluttered backgrounds. We have performed the experiments having in mind specific applications, thus limiting its applicability. The quality and stability of the results at close range makes it worthwhile for webcam or green screen sign language applications which share a close range viewing distance and a relatively uncluttered background.

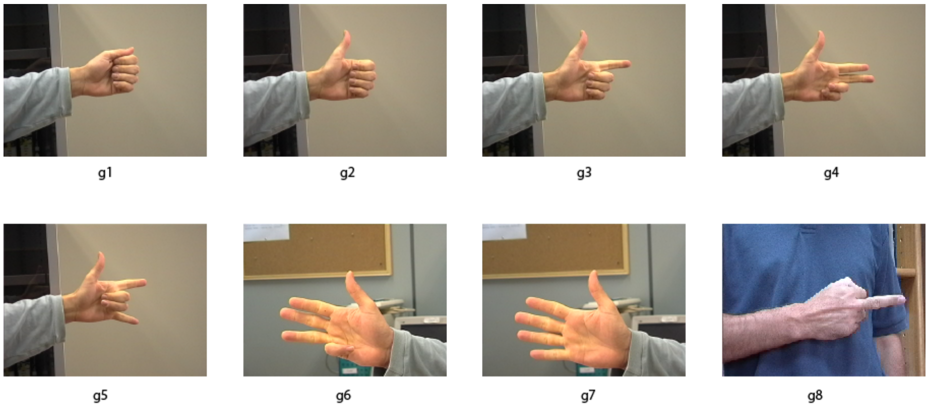


Figure 5.10: g1 to g8 represent the most common gestures used in sign language.

We have also tested the system in a more generic background where shadows, changes in lighting and extremely cluttered backgrounds are common. Figure 5.11 shows that when colour information is incorporated into the network the system is able to represent the gesture and only a few nodes adjust to nearby similar pixels. However, this is not the case when only intensity values are used in the map. The input space is violated and the object representation is lost. However, since the network has the ability to break up its map, objects of heterogeneous spaces will be represented independently by groups of nodes. Gesture representation is possible as long as no homogeneity is applied around the gesture.

For our experiments, tracking is possible without occlusion reasoning, since all fingers are the same colour and the background is unambiguous. Therefore, the network adapts either the fingers are self-occluded or not. However, if finger

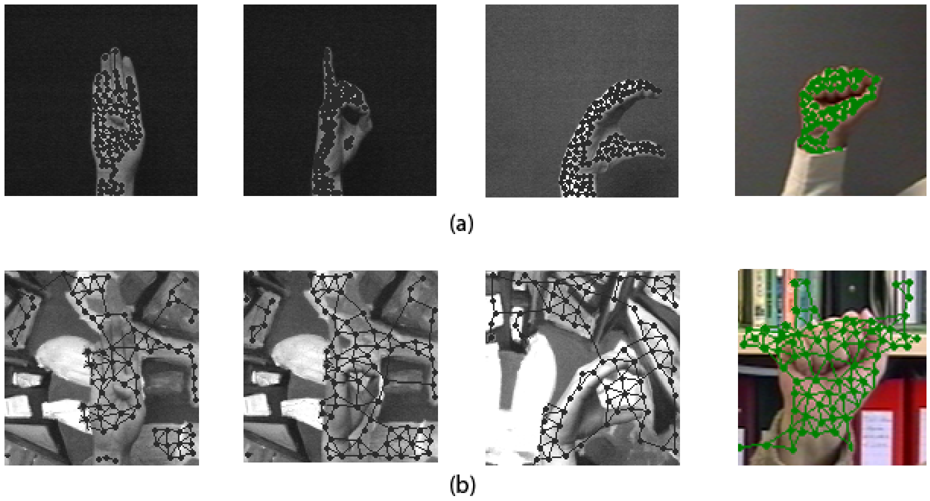


Figure 5.11: (a) Gestures in uniform backgrounds. (b) Examples of gestures in cluttered backgrounds.

recognition is important, for example the system should be able to distinguish between the middle and the ring fingers, then occlusion reasoning that takes into account the structural and temporal kinematics of the hand should be applied.

To classify a region as a hand or face we take into account domain knowledge information that respects always some proportions found in hands and human faces [71]. To do that we find the centroid, height and width of the connected nodes in the networks as well as the percentage of skin in the rectangular area (Figure 5.12). Since the height to width ratio for hands and human faces fall into a small range, we are able to reject or accept if the topology of a network represents or not a hand. Studies [52, 71], have shown that the height to width ratio of human face and hands fall within a range defined based on the well known Golden Ratio (Equation 5.6). Thus, we consider a network as a hand or not if the height to width ratio of the region falls within a range of the Golden Ratio ± $Tolerance$. In the case where the hand is in a folded posture the rule still applies but with different percentage for the $Tolerance$. The values for the $Tolerance$ were found by experimentation, and range from 0.5 to 0.7 based on the hand posture.

$$\varphi \equiv \frac{height}{width} \equiv \frac{(1+\sqrt{5})}{2} \tag{5.6}$$

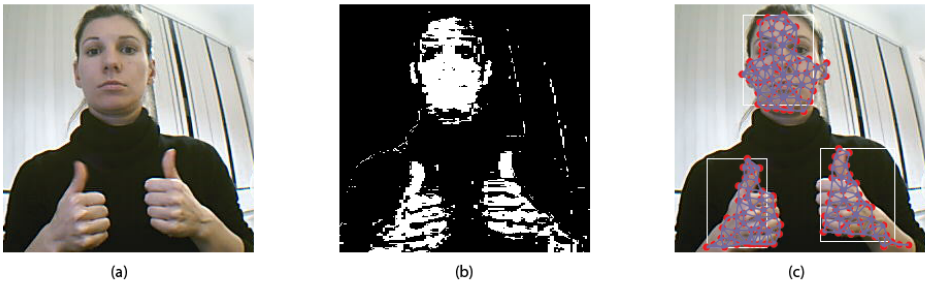


Figure 5.12: Example of correctly detected hands and face based on the golden ratio regardless of the scale and the position of the hands and the face. (a) original image, (b) after applying EM to segment skin region, and (c) hand and face detector taking into account the connected nodes in the networks as well as the percentage of skin in the rectangular area.

### 5.3.2 Bump

The Bump model used in Section 5.2 and in the Comparison study below, is a synthetic object that exhibits a single mode of shape variation where the bump moves along the top of the box. The model has one constant parameter, the rectangle and only the bump represents the new input space to the network. The model captures correctly the correspondence problem when the *A*-GNG is applied compared to the GNG network, as shown in Figures 5.3 and 5.21.

### 5.3.3 Comparison Study

In the following experiments we see the superiority of $A$-GNG against:

1. the methodology of active snake model that adheres only to global shape transformations and

2. GNG which has only global properties and cannot preserve correspondences when used for tracking.

Figure 5.13 show the tracking of a hand gesture using the $A$-GNG tracker, and how it outperforms the GNG tracker. Figure 5.13($a$) 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 $10$th frame is done with the $A$-GNG. Images ($b$) to ($i$) show the tracking of the nodes to a sequence of $190$ frames. Our tracker is able to track the fingers and updates the topology of the network every $5$ iterations at $\lambda = 1000$. Also, the execution time for $A$-GNG is approximately $3$ times less compared to the GNG. The computational and convergence results for these gestures are summarised in Table 5.1.

Figure 5.14 shows the fitting results of a snake applied to the same gesture. Figure 5.14 (a) is the original state of the snake after manually locating an area around the hand. The closer we allocate landmark points around the hand the faster the convergence of the snake. The snake after a number of iterations converges to the palm of the hand but fails to convergence around the thumb. The parameters for the snake are summarised in Table 5.2.

Figure 5.15 indicates another tracking example to a sequence of $45$ frames. Figure 5.15($a$) 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

Table 5.1: Convergence and Execution Time Results of GNG and $A$-GNG in $1D$ topology

| Method | $\lambda$ | Convergence (Iteration times) | Time (sec) |
|---|---|---|---|
| GNG | 100 | 7 | 2.53 |
| | 300 | 8 | 4.21 |
| | 600 | 11 | 7.01 |
| | 1000 | 14 | 15.04 |
| $A$-GNG | 100 | 2 | 0.73 |
| | 300 | 2 | 1.22 |
| | 600 | 3 | 2.17 |
| | 1000 | 5 | 4.88 |

Table 5.2: Parameters and Performance for Snake

| Hand | Constants | Iterations | Time (sec) |
|---|---|---|---|
| Sequence (a) | $\alpha = 0.05$<br>$\beta = 0$<br>$\gamma = 1$<br>$\kappa = 0.6$<br>D$_{min} = 0.5$<br>D$_{max} = 2$ | 40 | 15.29 |
| Sequence (b) | $\alpha = 4$<br>$\beta = 1$<br>$\gamma = 2$<br>$\kappa = 0.6$<br>D$_{min} = 0.5$<br>D$_{max} = 2$ | 50 | 15.20 |
| Sequence (c) | $\alpha = 4$<br>$\beta = 1$<br>$\gamma = 3$<br>$\kappa = 0.6$<br>D$_{min} = 0.5$<br>D$_{max} = 2$ | 40 | 12.01 |
| Sequence (d) | $\alpha = 4$<br>$\beta = 1$<br>$\gamma = 3$<br>$\kappa = 0.6$<br>D$_{min} = 0.5$<br>D$_{max} = 2$ | 20 | 5.60 |

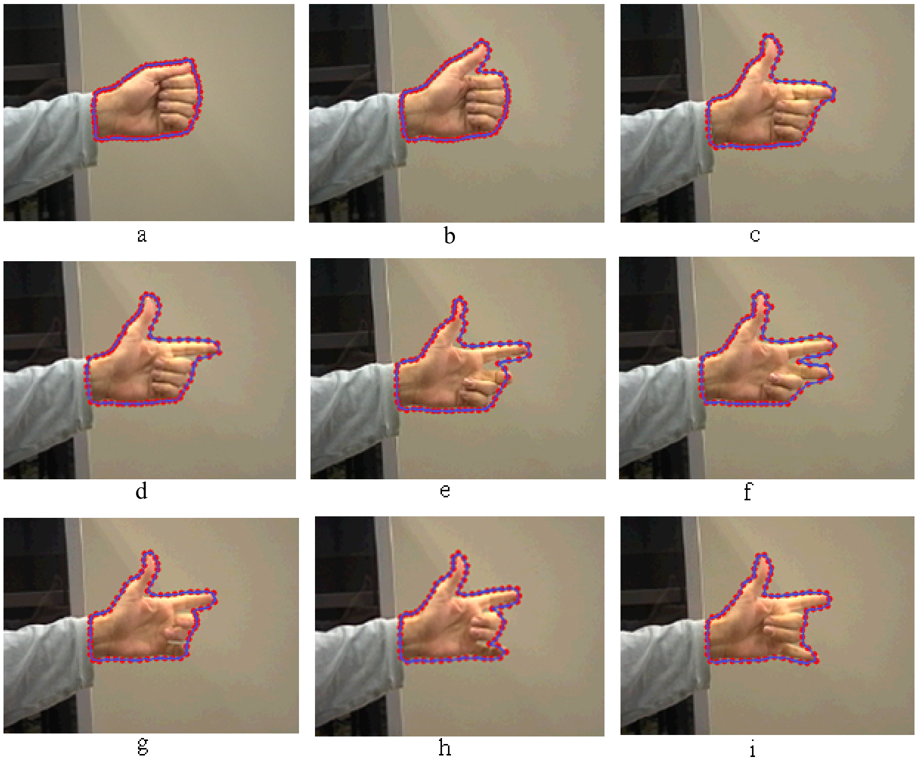


Figure 5.13: Tracking a gesture. The images correspond from left to right and from top to bottom to every $10$th frame of a $190$ frame sequence. In each image the red points indicate the nodes and their adaptation after $4$ iterations.

the updated position of the network to a more jumping distribution and how the network re-adapts again after $5$ iterations.

Figure 5.16 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 5.17 shows the $2D$ topology preserving map of the network on a skin colour distribution. The topology of the $1st$ frame is extracted with the GNG net-

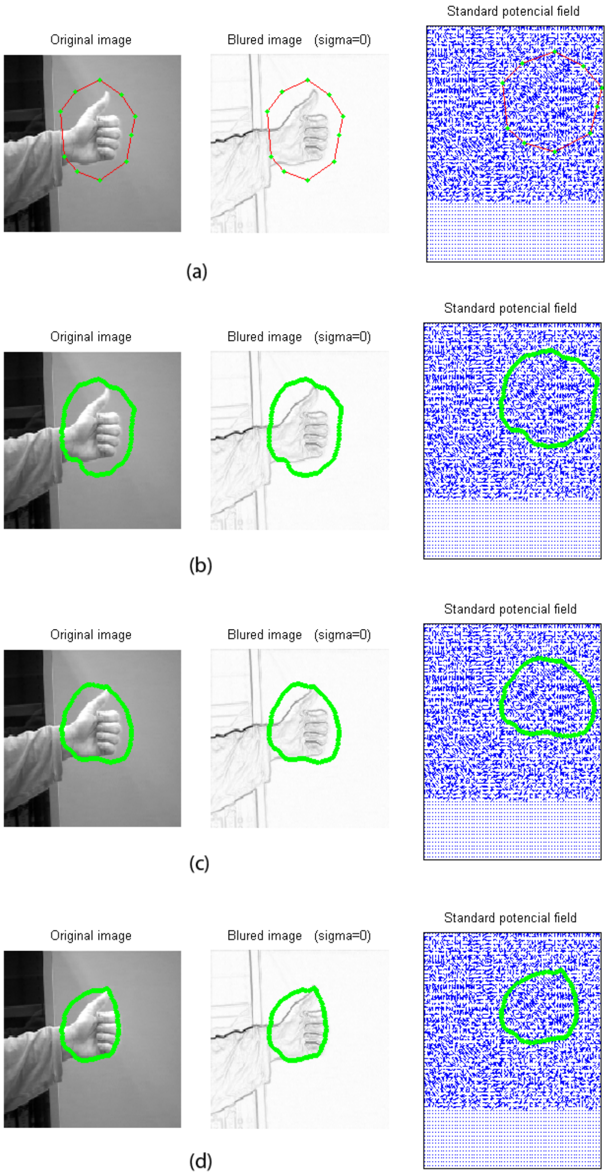


Figure 5.14: (a) Manual initialisation of the snake. (b) to (d) adaptation of the snake after a number of iterations.

work and the adaptation is performed with the *A*-GNG. If skin colour is falsely detected the network will split into smaller networks. To classify each of these networks as hands or not we calculate the Golden Ratio as described in Section 5.3.1.

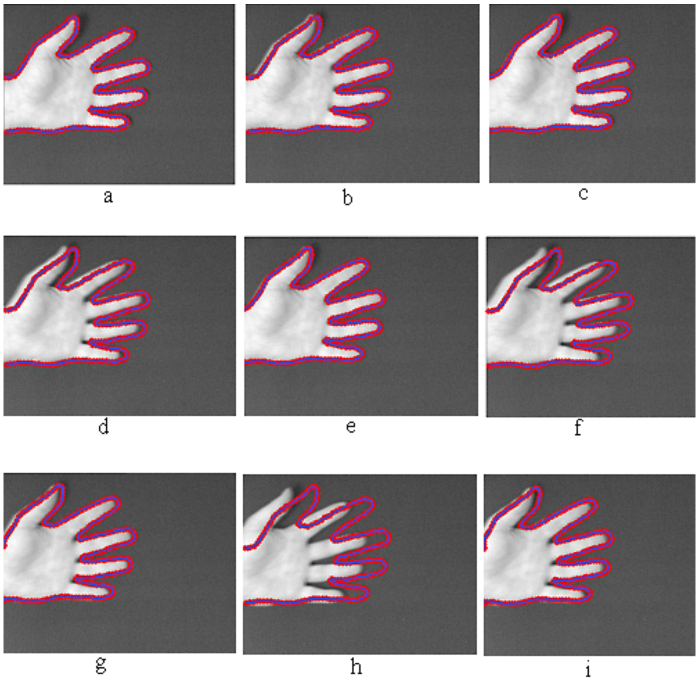


Figure 5.15: Tracking a hand. The images correspond from left to right and from top to bottom to every $5$th frame of a $45$ frame sequence.

The computational and convergence results for these gestures are summarised in Table 5.3.

Figure 5.18 shows a contour tracking example on a uniform background. The contour of the first image was extracted using the original GNG and the adaptation of the network at every $10$th frame is done with the $A$-GNG. Sequences $(b)$ - $(i)$ show the tracking 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 $5$ iterations.

Figure 5.19 indicates another $2D$ topology tracking example to a sequence of $90$ frames. Sequences $(a)$ - $(h)$ show the adaptation of the network after $5$ iterations using the skin colour distribution as the input distribution to the $A$-GNG network.

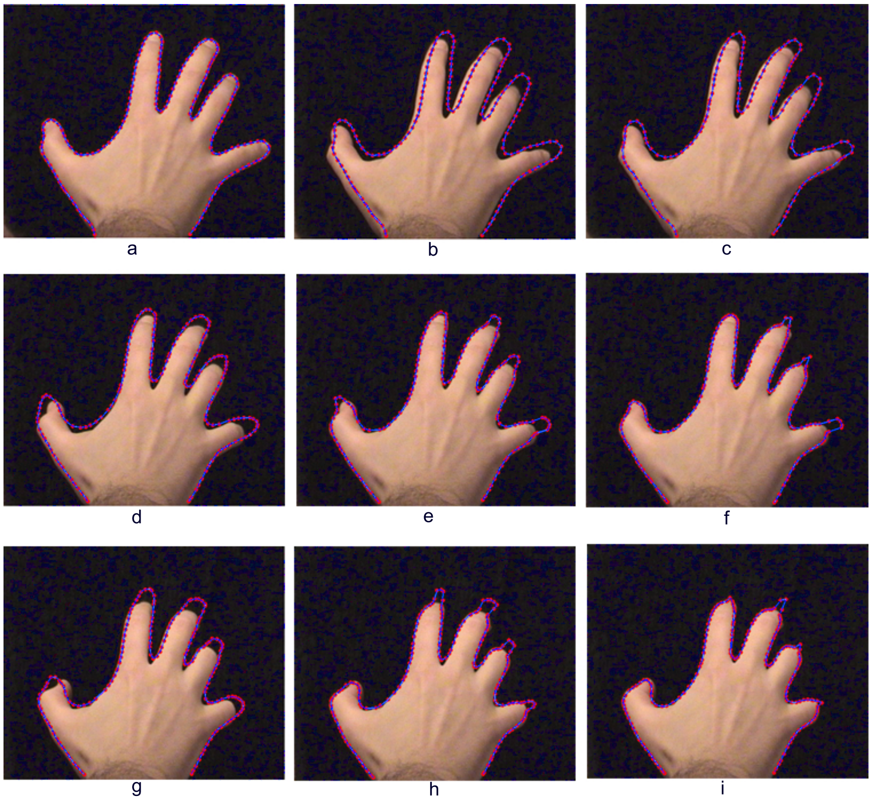


Figure 5.16: Tracking a gesture. The images correspond from left to right and from top to bottom to every $10$th frame of a $120$ frame sequence. In each image the red points indicate the nodes and their adaptation after $5$ iterations. The GNG tracker is quite far form the hand boundaries and the nodes collapse to local minimum.

Figure 5.20 shows the graphs with the topology preservation measures, Inverse Quantisation Error (IQE), Topographic Product (TP) and the Geodesic Topographic Product (GTP) for both images. It is observed that all measures considered are suitable candidates to be used for the quantification of the topology preservation of $A$-GNG. The low inverse quantisation error indicates the accuracy of the map in representing its input, while the two topology measures with $P \approx 0$ indicate good adaptation.

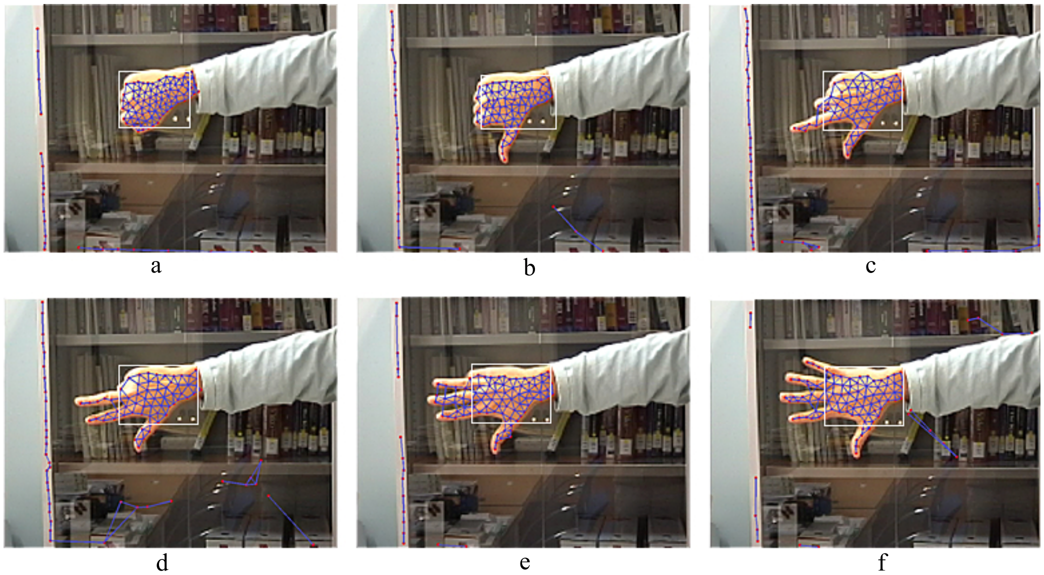


Figure 5.17: Tracking a gesture every $20th$ frame. The white rectangle around the hand identifies the network whose height to width ratio falls under the golden ratio.

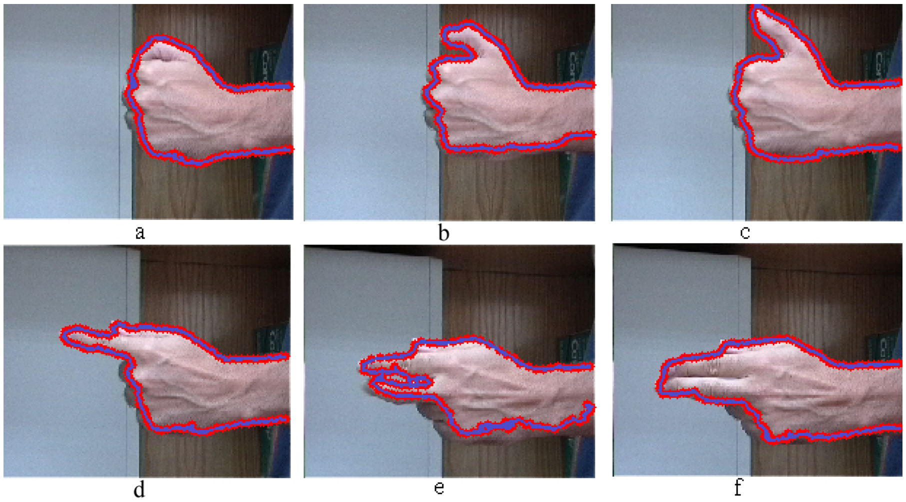


Figure 5.18: Tracking a gesture every $10th$ frame.

Table 5.3: Convergence and Execution Time Results of GNG and $A$-GNG in $2D$ Topology

| Method | $\lambda$ | Convergence (Iteration times) | Time (sec) |
|---|---|---|---|
| GNG | 100 | 5 | 6.29 |
| | 300 | 7 | 16.34 |
| | 600 | 11 | 22.01 |
| | 1000 | 15 | 29.38 |
| $A$-GNG | 100 | 3 | 2.34 |
| | 300 | 5 | 6.36 |
| | 600 | 7 | 8.48 |
| | 1000 | 11 | 12.62 |

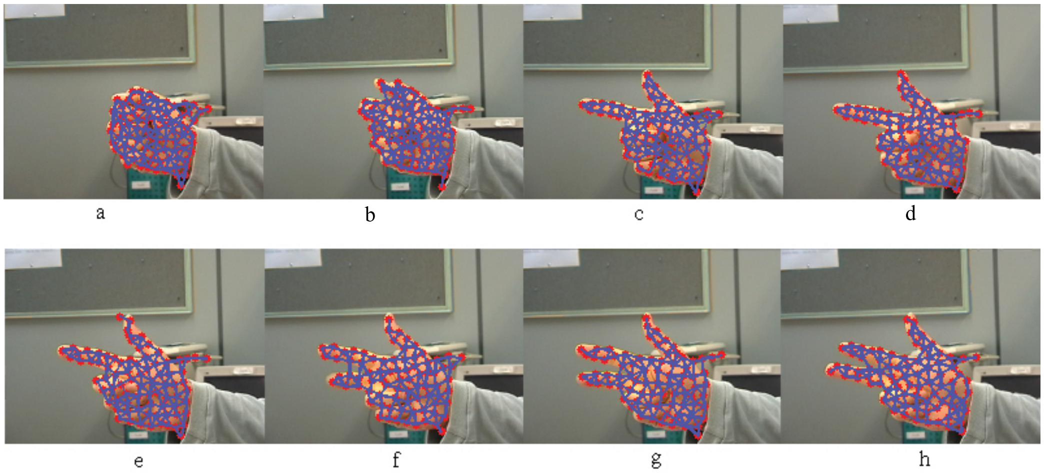


Figure 5.19: Tracking a gesture in a cluttered background. The images correspond from left to right and from top to bottom to every $10$th frame of a $90$ frame sequence.

Figure 5.21 is an example of local adaptation of the network between a bump model and a rectangle and how correspondences are improved using the $A$-GNG compared to the original GNG. Image $(a)$ and $(b)$ show the map of the bump model

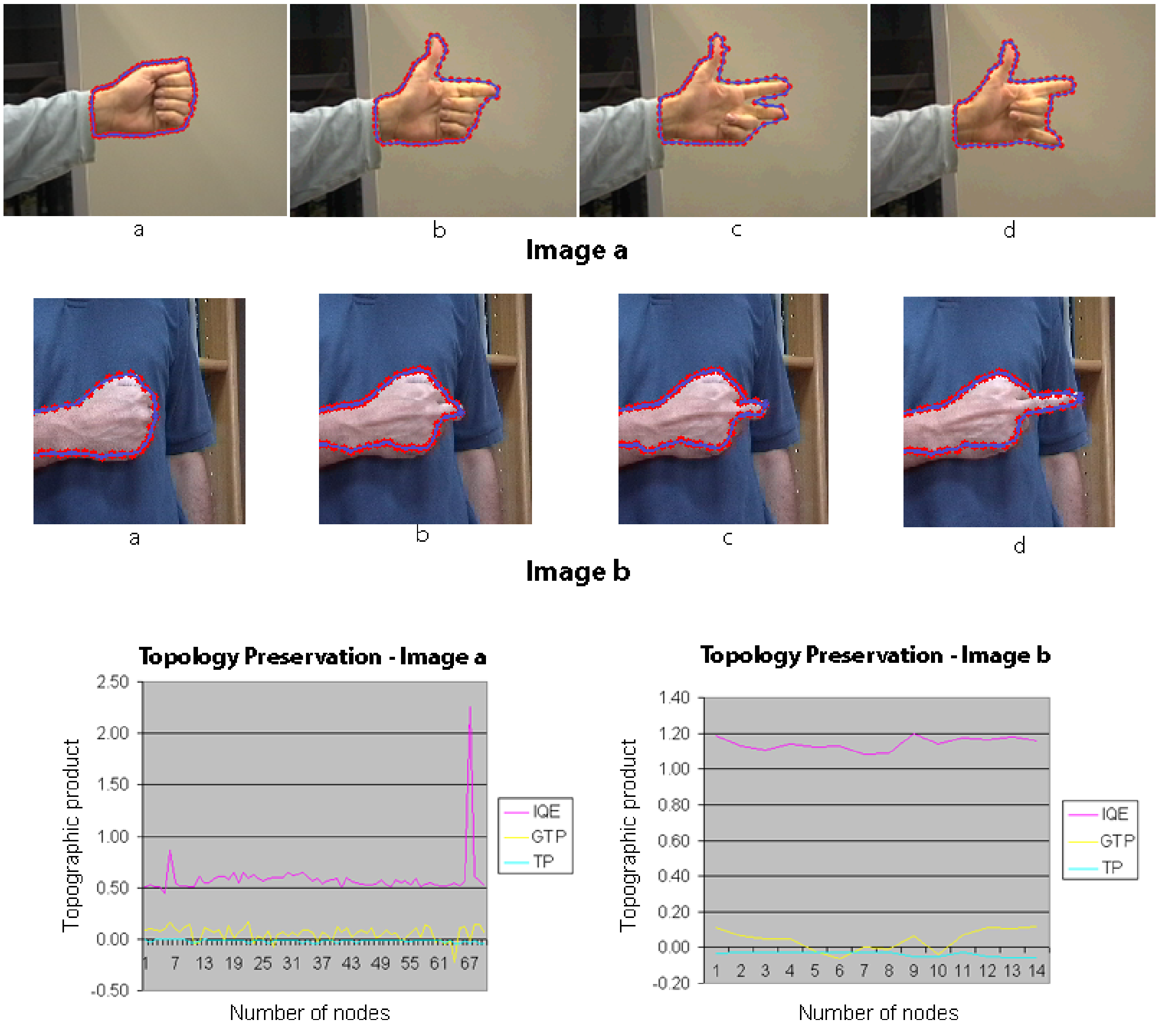


Figure 5.20: Topology preservation during the tracking process of $A$-GNG for images a and b measured with the Inverse Quantisation Error (IQE), the Topographic Product (TP) and the Geodesic Topographic Product (GTP).

and its superposition to the new image. Image (*d*) 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 (*c*) shows 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.

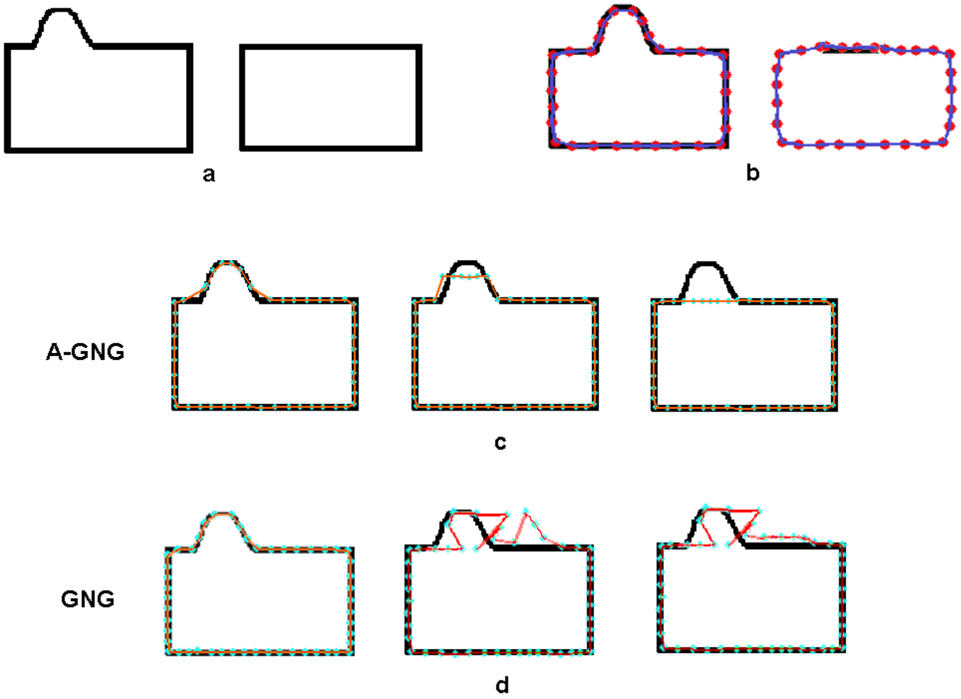


Figure 5.21: Local convergence between a bump model and a rectangle. $a$ shows the models: the bump model represents the original state of the network and the rectangle represents the final state after a number of $5$ iterations. $b$ shows the network superimposed to the bump model and where should locally converge. $c$ shows the steps of the network map using the *A*-GNG algorithm and the correct convergence to the rectangle. $d$ shows the adaptation using the GNG algorithm and how the network fails to converge after $5$ iterations.

Figure 5.22 indicates how feature information can add efficiency to the convergence. Image ($a$) and ($b$) show the map and the movement 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.

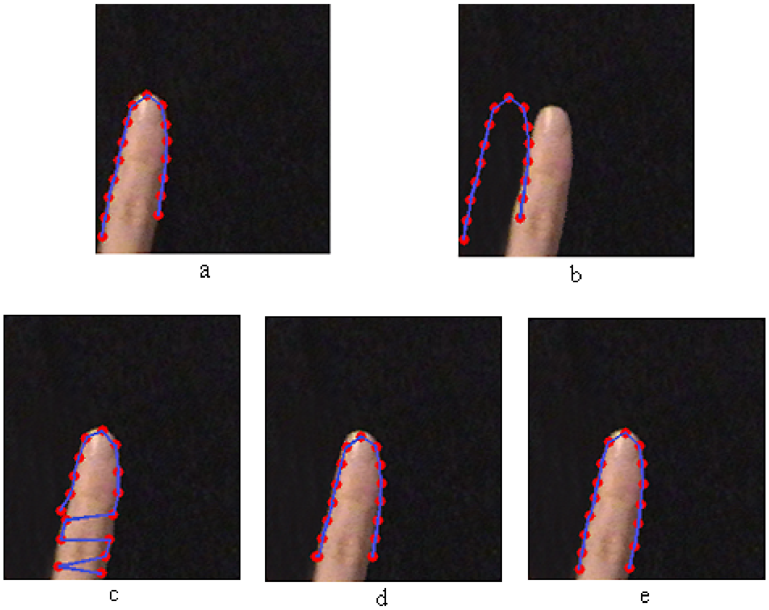


Figure 5.22: Convergence with and without the *active* steps of the GNG algorithm. (a) shows the original image and the network map obtained using GNG. (b) after the object has moved $k$ frames. (c) adaptation of the GNG algorithm. (d) and (e) adaptation with the $A$-GNG after $2$ iterations.

Table 5.4 shows the topographic product between input 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

Table 5.4: Method comparison

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

Table 5.4, measures the neighbourhood preservation by calculating the map difference between neighbouring nodes in successive frames. In both examples the topographic product for the GNG algorithm is $P < 0$ which indicates a low match between the input and the latent space. For $A$-GNG $P \approx 0$ which indicates a match and accurate topology preservation.

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. Table 5.4 shows that $A$-GNG outperforms GNG and correct correspondences are established only when the map is close enough to the new input distribution.

## 5.4 Summary

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. This is done by updating the map every $5th$ frame based on the information obtained from a small region around each node. Then the displacement is achieved on the features obtained from the skin distribution. No training set is required and the user interaction is only necessary at initialisation. The algorithm is computationally inexpensive, and can handle multiple open/closed boundaries. Experiments were performed in hand gestures and the superiority of our algorithm was compared, in real and artificial data set, with the GNG algorithm. The quality of the object's representation for both networks was measured with the topographic product.

# Chapter 6

# Conclusions

*In this thesis we addressed the main limitations of nonrigid shape modelling and tracking, and proposed an approach that overcomes problems like training a set of examples, specifying shape features* a priori *in the model thus, model and track in a supervised manner, and deforming the objects globally by using an unsupervised framework with global and local properties for automatic model building. Below we summarise the entire work covered in the previous chapters and propose areas of interest for future work.*

## 6.1 Discussion

In Chapter 3 we presented an approach to automatically extract, label and correspond points using only topological relations derived from competitive hebbian learning. We addressed the correspondence problem as an unsupervised classification problem where landmark points are the cluster centres (nodes) in a high-dimensional vector space. We assumed that in a pre-processing stage the contours of the objects were extracted using an efficient segmentation scheme. Points were extracted in the following way. First we used GNG to extract landmark points along the contours and to form topology preserving maps. The result of this adaptation was a list of non-ordered nodes that defined a graph. Then we normalised the graph by defining a re-ordering rule of the nodes. We have achieved that by 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 nodes created in the learning process all the inappropriate cycles produced. The re-ordered list of nodes was then projected into the shape space where synthesised shapes similar to the training set were generated using the PDM. Furthermore, we have improved the parameters of GNG by removing wrong edges between nodes that can be obtained either due to limited time of the network to adapt or the nodes are too close and the topology preserving graph cannot differentiate between winner and immediate neighbours. We have achieved that by defining a rule that compares the slope defined by the edge formed from the last two nodes inserted, with the slopes of the edges defined by the last node inserted and any of the candidate neighbours. Experimental evaluation was performed on two different data sets and comparisons with other self-organising models were conducted. The accuracy of the model was evaluated with the topology preserving measure the topographic product.

In Chapter 4 we presented an approach to minimise the user intervention in the learning process of the network by utilising an automatic criterion for maximum node growth based on different parameters. First we discussed various methods on background modelling so to get an accurate initialisation of the first frame of the GNG. We decided to use skin-colour segmentation based on probabilistic colour models since our interest is in applications of gestures and hand shape modelling. Then we introduced the automatic criterion of GNG based on the parameter that the class probability of pixels belonging to the objects of interest should be above a particular similarity threshold. The global minimum and as such the optimal number of this similarity threshold that best describes the topology of the network without overfitting or underfitting the data set is derived from information theory which can describe the complexity or simplicity of the model. Experimental evaluation was performed on a set of images with various gestures and hand postures. During learning the efficiency of the network based on the optimum number that maximises topology learning versus adaptation time and MSE was evaluated with the topographic product.

In Chapter 5 we introduced *Active*-GNG which builds on the capabilities of GNG but is extended in the following ways. First the correspondence of the nodes is performed locally compared to the global approach we followed in Chapter 3. This is achieved by adding a distance vector between the $1st$ frame and any successive $k$ frames. This distance vector calculates the mean of the input distribution between the current and successive frames, and the nodes update their position based on this mean difference. This is done by updating the map every $5th$ frame based on the information obtained from a small region around each node. Then the displacement is achieved on the features obtained from the skin distribution. Gestures are tracked in an unsupervised manner in a sequence of $k$ frames. The highest the probability of a node to belong to the skin prior probability the faster the node will adapt to the new input distribution. Experimental evaluation was performed on hand gestures and the superiority of our algorithm was compared, in real and artificial data set, with the GNG algorithm. The accuracy of the objects' representation was evaluated with the topographic product.

## 6.2 Future Work

Further research can be carried out in the following ways:

- Currently the $A$-GNG like all the Fully Self-Organising Artificial Neural Network Models (FSONN) has no *a priori* knowledge about the object domain. This attribute allows the model to vary its topology and model objects of variable topology. There are many cases in medical imaging where this is important, for example, osteoarthritis (OA) causes articulating cartilage of load-bearing joints to erode thus changing the topology of the examples [39]. Due to the nature of $A$-GNG topological changes can be accommodated to the shape model. However, what our model lacks is specificity, which means to be able to represent only legal instances of the class of objects. For example, in the case where a hand gesture is tracked in a sequence of frames and occlusion occurs since no prior shape information is incorporated in the model our tracked algorithm will fail to track the object and the quality of the segmentation will degrade. This limitation could be overcome by combining shape information with topological constrains in $A$-GNG. By doing so assumptions about the global structure of the objects are incorporated in the model.

- We defined global and local deformations by restricting the movement of the nodes, taking the differences between successive frames and re-adjusting only the nodes that belong to those differences, and by incorporating the probability of the skin distribution to the network. While this is true in applications where skin colour information is obtainable, GNG requires initialisation only for the $1st$ frame since new $TPGs$ are obtained from the features added to the nodes, it is not the case in applications like surveillance systems where colour cannot be used to perform segmentation. In that case the segmented patches derived by frame differencing can be combined with predictive algorithms like Kalman filters [90] to estimate the velocity and acceleration of objects which can then be passed to the network.

- Finally, the framework has been developed for $2D$ (open and closed curves) objects. We would like to extend our framework so $3D$ objects can be automatically build and tracked in a sequence of $k$ frames. This extension can be of great interest in the recognition of gestures in a Virtual Reality (VR) system using only visual information instead of the current complex electromagnetic trackers.

# Appendix A

# Unsupervised Learning

*In this appendix, we discuss the principles of competitive learning that are relevant for this thesis. First, we give an exact definition of concepts derived from computational geometry that are used in all self-organising neural networks. Second, we present the learning algorithm as used in this thesis and its application in computer vision.*

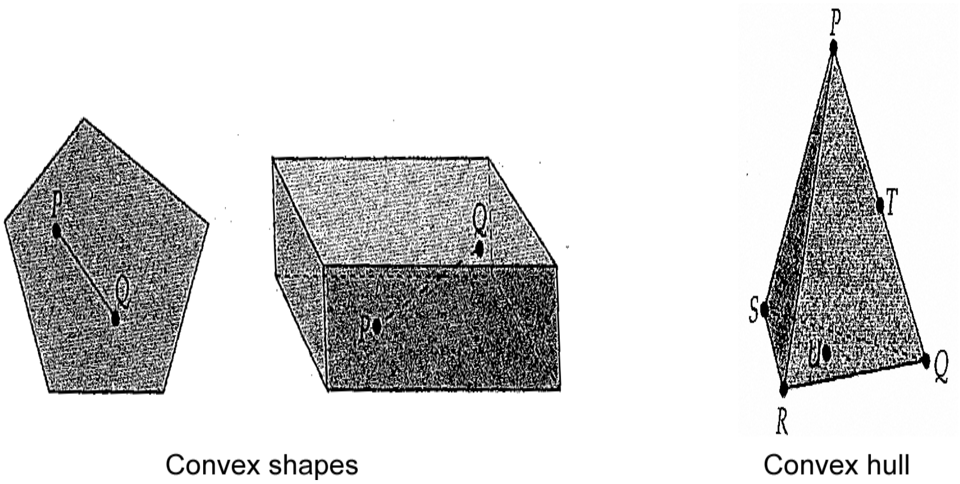
## A.1 Computational Geometry

Computational geometry emerged from the field of algorithmic design and analysis back in the 1970s, and elevates real life problems as geometric problems that require carefully designed geometric algorithms for their solution [41]. In the following, we give definitions of the most important geometric concepts as used in this thesis.

### A.1.1 Convexity

**Definition**

A set $S$ of points in a Euclidean space is **convex** if, for each possible choice of two points $P, Q \in S$ all the points on the line segment joining $P$ and $Q$ also lie in $S$. The **convex hull** of $S$ is the smallest convex set containing $S$ and is denoted by $\langle S \rangle$. To be more precise, it is the intersection of all convex sets that contain $S$. Figure A.1 shows examples of convex shapes and convex hulls.



Convex shapes

Convex hull

Figure A.1: Left image, shows examples of $2D$ and $3D$ convex shapes. Right image shows that the convex hull $\langle P, Q, R, S, T, U \rangle$ is the tetrahedron $PQRS$.

### A.1.2 Triangulations

**Definition**

A **triangulation** is a strict subdivision in which each face is a triangle. A **strict subdivision** of a surface is a subdivision in which:

- Any two faces meet at a single edge, at a single vertex, or not at all;
- Each non-boundary edge belongs to two faces;
- Each edge that is part of the boundary belongs to just one face;
- No face meets itself, either at a vertex or at an edge;
- The union of all the faces meeting a given vertex, together with the vertex and its incident edges, is homeomorphic to an open disc - or if the vertex lies on the boundary, to an open half-disc.

Figure A.2 shows the conversion from strict subdivision into a triangulation.

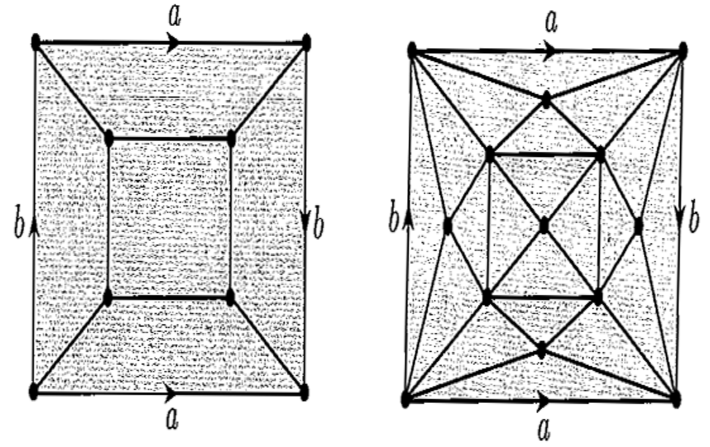


Figure A.2: Conversion from strict subdivision into triangulation.

### A.1.3 Voronoi Diagram and Delaunay Triangulation

**Definition**

Let $P$ be a set of $n$ nodes in the plane, the **Voronoi diagram** of $P$ is the subdivision of the plane into $n$ regions, such that the region of a node $p \in P$ contains all points in the plane for which $p$ is the nearest node. The Voronoi diagram of $P$ is denoted by $Vor(P)$ [41]. Assume there exists $10$ nodes in $R^2$ as depicted in Figure A.3. The Voronoi diagram has the property that for each node every point in the region around that node is closer to that node than to any of the other nodes.

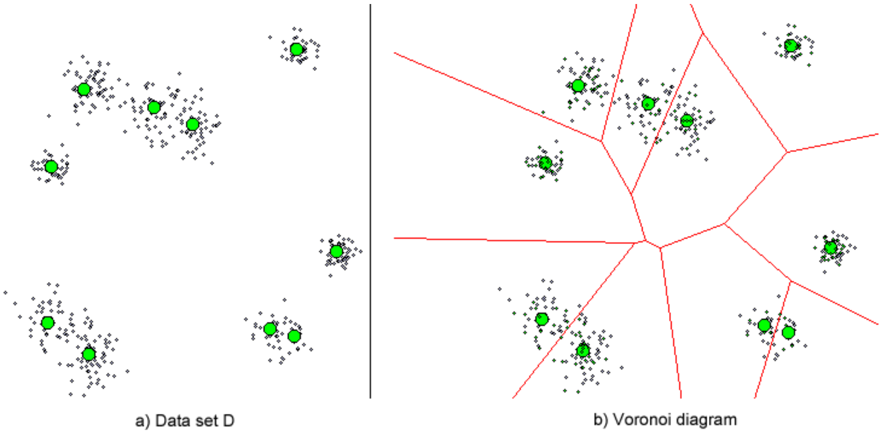


Figure A.3: An input Data set $D$ with $10$ nodes in $R^2$ is shown in $(a)$. The Voronoi diagram for this particular set of nodes is shown in $(b)$.

If one connects all pairs of points for which the respective Voronoi regions share an edge one gets the **Delaunay Triangulation**. This triangulation is special among all possible triangulations since it has the additional property that for each triangle of the triangulation, the circumcircle of that triangle does not contain any other nodes. Figure A.4 shows the previous example together with the corresponding Delaunay Triangulation.

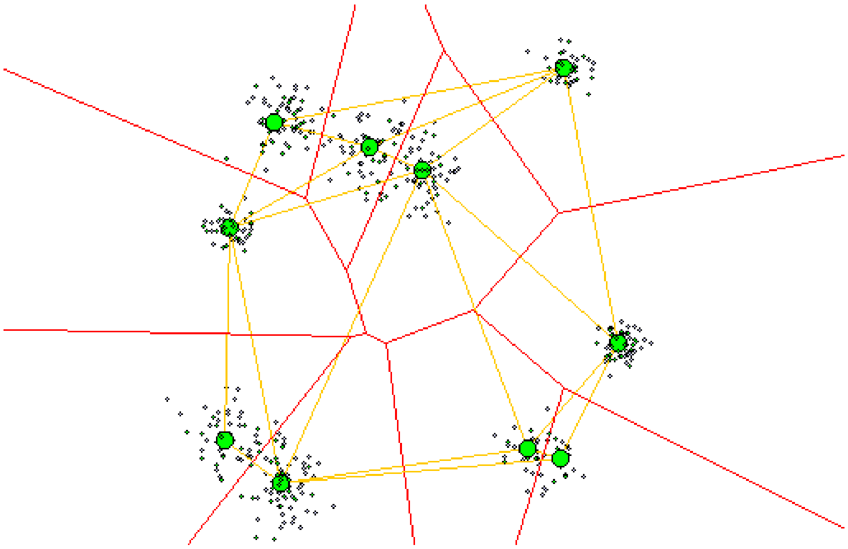


Figure A.4: Both the Delaunay triangulation and the Voronoi diagram.

## A.2 GNG Algorithm

From the Neural Gas model [117, 118] and the Growing Cell Structures [59], Fritzke developed the Growing Neural Gas model [60]. The growth mechanism inherited from the Growth Cell Structure [59], and the Competitive Hebbian Learning (CHL) rule [115] are combined to a new model that starts with two nodes and new ones are inserted successively (Figure A.5).

This model has been used in applications such as robotics [66, 112], face recognition [189], clustering [33, 45, 86], and 3D reconstruction [2, 35] among others. The learning algorithm is as follows:

1. Initialise the set $A$ with only two nodes $c_1$ and $c_2$

$$A = \{c_1, c_2\} \tag{A.1}$$

with their respective reference vectors (weights) $x_{c_1}$ and $x_{c_2}$ randomly generated from the probability density function $p(W)$. Initialise the connection set

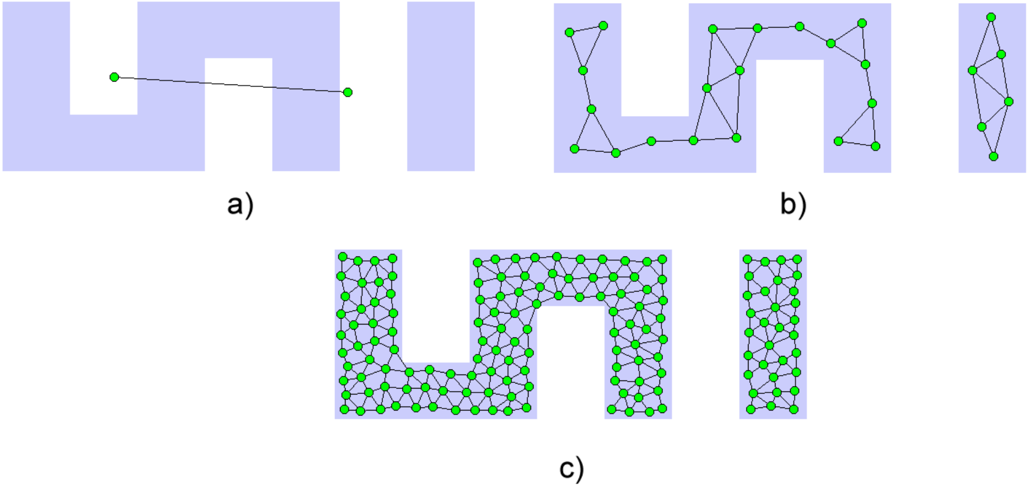


Figure A.5: a) Initial, b) intermediate and c) final state of the GNG algorithm.

$C$, $C \subset AxA$, to the empty set:

$$C = \emptyset \tag{A.2}$$

2. Generate at random an input signal $\xi_w$ according to the data distribution $p(W)$.

3. Find the nearest node (winner node) $x_\nu$ and the second nearest $x_\upsilon(x_\nu, x_\upsilon \in A)$ by:

$$x_\nu = \arg\min{}_{c \in A} \parallel \xi_w - x_c \parallel \tag{A.3}$$

and

$$x_\upsilon = \arg\min{}_{c \in A \backslash \{x_\nu\}} \parallel \xi_w - x_c \parallel \tag{A.4}$$

4. If $x_\nu$ and $x_\upsilon$ are not connected, create it:

$$C = C \cup \{(x_\nu, x_\upsilon)\} \tag{A.5}$$

Set the age of the connection between the two nodes to $0$.

$$age_{(x_\nu, x_\upsilon)} = 0 \tag{A.6}$$

5. Add the squared distance between the input signal and the winner node to a counter error of $x_\nu$ such as:

$$\Delta error(x_\nu) = \parallel \xi_w - x_\nu \parallel^2 \tag{A.7}$$

6. Move the winner node $x_\nu$ and its topological neighbours towards $\xi_w$ by a learning step $\epsilon_x$ and $\epsilon_n$, respectively, of the total distance to the input signal:

$$\Delta x_\nu = \epsilon_x(\xi_w - x_\nu) \tag{A.8}$$

$$\Delta x_c = \epsilon_n(\xi_w - x_c), \forall c \in N \tag{A.9}$$

7. Increase the age of all the edges emanating from $x_\nu$:

$$age_{(x_\nu,i)} = age_{(x_\nu,i)} + 1 \ (\forall i \in N_{x_\nu}) \tag{A.10}$$

8. Remove the edges larger than $a_{max}$. If this results in isolated nodes remove them as well.

9. Every certain number $\lambda$ of input signals generated, insert a new node as follows:

   - Determine the node $q$ with the maximum accumulated error:

   $$q = \arg\max{}_{c \in A} E_c \tag{A.11}$$

   - Determine among the neighbours of $q$ the node $f$ with the maximum accumulated error:

   $$f = \arg\max{}_{c \in N_q} E_c \tag{A.12}$$

   - Insert a new node $r$ between $q$ and its further neighbour $f$:

   $$A = A \cup \{r\} \tag{A.13}$$

   $$x_r = \frac{(x_q + x_f)}{2} \tag{A.14}$$

- Insert new edges connecting the node $r$ with nodes $q$ and $f$, removing the old edge between $q$ and $f$.

$$C = C \cup \{(r,q),(r,f)\} \tag{A.15}$$

$$C = C \cup \backslash \{(q,f)\} \tag{A.16}$$

- Decrease the error variables of nodes $q$ and $f$ multiplying them by a fraction $\alpha$:

$$\Delta error_(q) = -\alpha E_q \tag{A.17}$$

$$\Delta error_(f) = -\alpha E_f \tag{A.18}$$

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

$$E_r = \frac{(E_q + E_f)}{2} \tag{A.19}$$

10. Decrease all error variables by multiplying them with a constant $\beta$:

$$\Delta error_(c) = -\beta E_c \tag{A.20}$$

11. If the stopping criterion is not yet achieved (e.g. maximum size of the network, time, etc.) go to step 2.

The parameters used in Figure A.5 are: $\lambda = 600$, $\varepsilon_x = 0.05$, $\varepsilon_n = 0.0006$, $\alpha = 0.5$, $\beta = 0.0005$, $\alpha_{max} = 88$.

In summary, the adaptation of the network to the input distribution is produced in step $6$. Step $4$ with the insertion of connections, provides the topological relations between the nodes. The elimination of connections (step $8$) removes edges that are no part of the topology. This is done by removing the connections between nodes that are no longer near or that have other nodes that are closer. The accumulation of the error (step 5) can identify those areas where it is necessary to increase the number of nodes to improve the mapping.

Figure A.6 shows the learning algorithm.

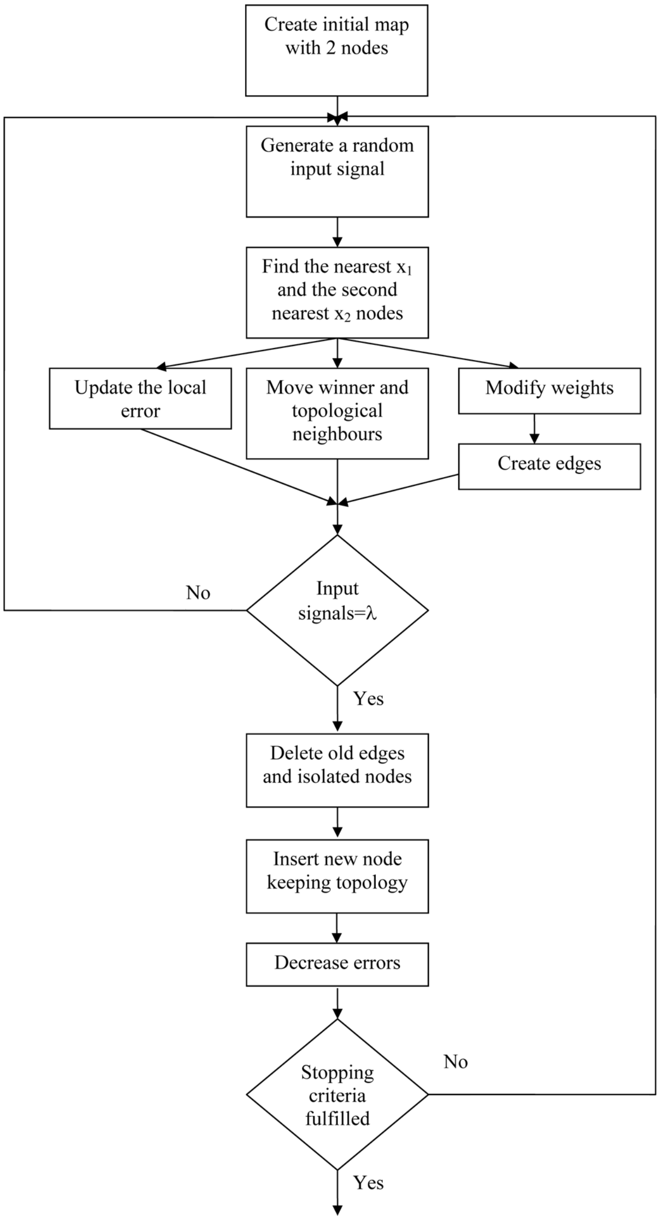

Create initial map with 2 nodes
Generate a random input signal
Find the nearest $x_1$ and the second nearest $x_2$ nodes
Update the local error
Move winner and topological neighbours
Modify weights
Create edges
No
Input signals=λ
Yes
Delete old edges and isolated nodes
Insert new node keeping topology
Decrease errors
Stopping criteria fulfilled
No
Yes

Figure A.6: Flow chart of the growing neural gas network.

# Appendix B

# Aligning the Training Set

*In section 3.3 PDM was discussed as a method for statistical shape modelling. In order to study the variations of the position of each landmark point throughout the training set, all the shapes must be aligned to each other. This appendix discusses the alignment of a pair of shapes with or without using a weighted matrix. It generalises to many shapes.*

## B.1 Aligning A Pair of Shapes

Given two shapes $x_1$ and $x_2$ which are described by a vector of $N$ coordinates:

$$x_1 = (x_1, y_1, x_2, y_2, ......, x_N, y_N)^T \tag{B.1}$$

$$x_2 = (x_1, y_1, x_2, y_2, ......, x_N, y_N)^T \tag{B.2}$$

we wish to find the parameters ($\theta, s, t_x, t_y$) of the transformation matrix $T$ that minimises the sum of the square distances:

$$E = |x_1 - T(x_2)|^2 \tag{B.3}$$

where

$$T\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} s(cos\vartheta)x_i - s(sin\theta)y_i \\ s(sin\vartheta)x_i + s(cos\theta)y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{B.4}$$

for simplicity we write $a = s(cos\vartheta)$ and $b = s(sin\vartheta)$. We can write equation (B.3) as:

$$E(a, b, t_x, t_y) = |x_1 - T(x_2)|^2 = \sum_{i=1}^{n}(ax_{1i} - by_{1i} + t_x - x_{2i})^2 + (bx_{1i} + ay_{1i} + t_y - y_{2i})^2 \tag{B.5}$$

This minimisation is a standard application of a least-square approach where partial derivatives of $E$ are calculated with respect to the unknown parameters $(\theta, s, t_x, t_y)$. In order to find the parameters that best align shape $x_2$ to $x_1$ we partially differentiate equation (B.5) with respect to $(a, b, t_x, t_y)$ and by equating $\frac{\partial E}{\partial a} = 0$, $\frac{\partial E}{\partial b} = 0$, $\frac{\partial E}{\partial t_x} = 0$, $\frac{\partial E}{\partial t_y} = 0$ we solve a system of four linear equations:

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ z & 0 & x_1 & y_1 \\ 0 & z & -y_1 & x_1 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ c_1 \\ c_2 \end{pmatrix} \tag{B.6}$$

where

$$x_1 = \sum_{i=1}^{n} x_{1i} \tag{B.7}$$

$$y_1 = \sum_{i=1}^{n} y_{1i} \tag{B.8}$$

$$x_2 = \sum_{i=1}^{n} x_{2i} \tag{B.9}$$

$$y_2 = \sum_{i=1}^{n} y_{2i} \tag{B.10}$$

$$z = \sum_{i=1}^{n} {x_1}_i^2 + {y_1}_i^2 \tag{B.11}$$

$$c_1 = \sum_{i=1}^{n} x_{1i}x_{2i} + y_{1i}y_{2i} \tag{B.12}$$

$$c_2 = \sum_{i=1}^{n} x_{1i}y_{2i} - y_{1i}x_{2i} \tag{B.13}$$

By solving the equations and by assuming that the centre of gravity of $x_2$ is at the origin we obtain:

$$a = \frac{c_1}{|x_1|^2} \tag{B.14}$$

$$b = \frac{c_2}{|x_1|^2} \tag{B.15}$$

$$t_x = x_2 \tag{B.16}$$

$$t_y = y_2 \tag{B.17}$$

and

$$\theta = \arctan(\frac{b}{a}) = \arctan(\frac{c_2}{c_1}) \tag{B.18}$$

Knowing $a$ and $b$ we can calculate the scaling parameter $s$ from $s^2 = a^2 + b^2$. By knowing $s, \theta, t_x, t_y$ we can align two pair of shapes in Euclidean transformation.

## B.2 Aligning A Pair of Shapes Using a Waited Matrix

The alignment of two shapes using a weighted matrix is exactly the same with the previous method, with the only difference that the weighted matrix gives more significance to the landmark points that tend to be more stable. The stability of a point is measured by having less movement relative to the other points. The weights given to the points are calculated based on: (1) the distances between every pair of points in all the shapes; (2) the variance of the distance between every pair of points over all the shapes; (3) the inverse of the summation of the variances of the distances from this point to all others.

The weighted matrix is given as:

$$w_k = (\sum_{l=1}^{n} V_{kl})^{-1} \tag{B.19}$$

where $0 \leq k \leq n$, $V_{kl}$ is the variance of the distance between landmark points $k$ and $l$, and $n$ is the number of landmark points. The weighted matrix can be written as the diagonal matrix $\mathbf{W} = diag(w_{1x}, w_{1y}, ......., w_{nx}, w_{ny})$.

Given two shapes $x_1$ and $x_2$ we wish to find the parameters $(\theta, s, t_x, t_y)$ of the transformation matrix $T$ that minimises the weighted sum:

$$E = (x_1 - T(x_2))^T \mathbf{W}(x_1 - T(x_2)) \tag{B.20}$$

By equating $\frac{\partial E}{\partial a} = 0$, $\frac{\partial E}{\partial b} = 0$, $\frac{\partial E}{\partial t_x} = 0$, $\frac{\partial E}{\partial t_y} = 0$ we solve a system of four linear equations:

$$\begin{pmatrix} x_1 & -y_1 & W & 0 \\ y_1 & x_1 & 0 & W \\ z & 0 & x_1 & y_1 \\ 0 & z & -y_1 & x_1 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ c_1 \\ c_2 \end{pmatrix} \tag{B.21}$$

where

$$x_1 = \sum_{i=1}^{n} w_k x_{1i} \tag{B.22}$$

$$y_1 = \sum_{i=1}^{n} w_k y_{1i} \tag{B.23}$$

$$x_2 = \sum_{i=1}^{n} w_k x_{2i} \tag{B.24}$$

$$y_2 = \sum_{i=1}^{n} w_k y_{2i} \tag{B.25}$$

$$z = \sum_{i=1}^{n} w_k({x_1}_i^2 + {y_1}_i^2) \tag{B.26}$$

$$c_1 = \sum_{i=1}^{n} w_k(x_{1i}x_{2i} + y_{1i}y_{2i}) \tag{B.27}$$

$$c_2 = \sum_{i=1}^{n} w_k(x_{1i}y_{2i} - y_{1i}x_{2i}) \tag{B.28}$$

## B.3 Aligning All Shapes

Figure B.1 shows the algorithm for aligning a set of $N$ shapes.

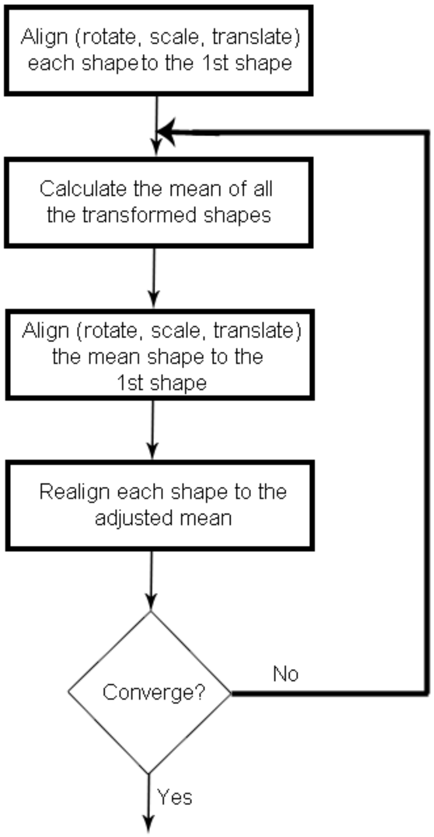

Align (rotate, scale, translate) each shapeto the 1st shape
Calculate the mean of all the transformed shapes
Align (rotate, scale, translate) the mean shape to the 1st shape
Realign each shape to the adjusted mean
Converge?
No
Yes

Figure B.1: Flow chart for aligning a set of $N$ shapes.

# Appendix C

# Human Brain MRI Data Set

*This appendix compares to and presents results for the automatic landmarking and shape modelling of a ventricles data set using three different self-organising models.*

## C.1 Ventricles

The data set was obtained from the MNI BIC Centre for Imaging at McGill University, Canada. These images are $1$ mm thick, $181 \times 217$ pixels per slice ($1.0mm^2$ in-plane resolution), $3\%$ noise and $20\%$ INU. These images are used as ground truth segmentation, as every voxel in the entire volume has been correctly labelled to a tissue class by the McGill Institute. The entire brain volume consisted of $181$ slices, from which we extracted those that contained ventricles (slices $49-91$). The images are $16$ bit grey scale, which were manually segmented to remove all but the outline of the ventricles. Since most typical clinical MRI volumes are on average $5$mm thick, we selected $4$ groups of $5$ contiguous slices to produce our point distribution model.

In Figure C.1 the modes of variation for all four groups are displayed by varying the first shape parameter $\beta_i\{\pm 3\sigma\}$ over the training set. The qualitative results show that GNG leads to correct extraction of corners of anatomical shapes and are compact when the topology preservation of the network is achieved (Figure C.3). In Figure C.2 two shape variations from the automatically generated landmarks were superimposed to groups $4$ and $3$ from the training set. These modes effectively capture the variability of the training set and present only valid shape instances. It is interesting to note that whilst there is significant difference between $64$, and $169$ nodes -not enough nodes to represent the object at specific time constraints (Image A) and too many nodes (Image D)- the mapping with $100$ is good and has no significant difference with the mapping of $144$ nodes. The reason is that for the current size of the images the distance between the nodes is short enough so adding extra nodes does not give more accuracy in placement.

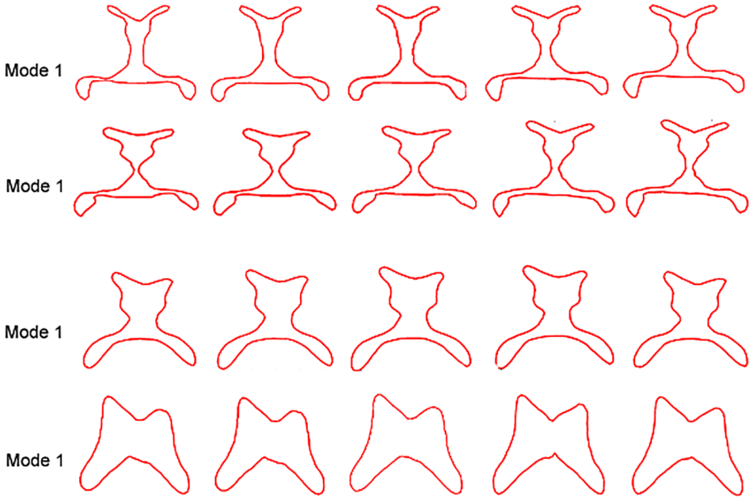


Figure C.1: The first mode ($m = 1$) of variation for the four groups of 5 contiguous slices taken from MR brain data. Range of variation $-3\sqrt{\lambda} \leq \beta_i \leq 3\sqrt{\lambda}$.

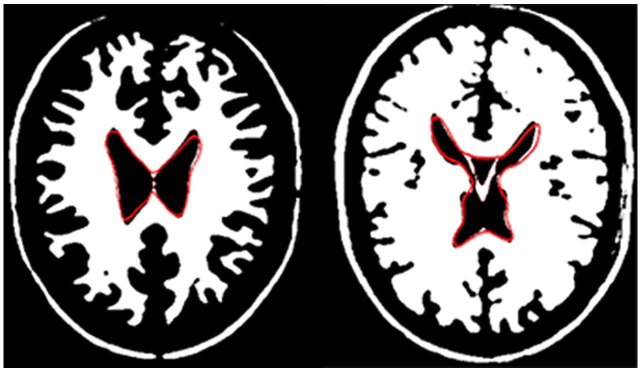


Figure C.2: Superimposed shape instances to groups 4 and 3 from the training set.

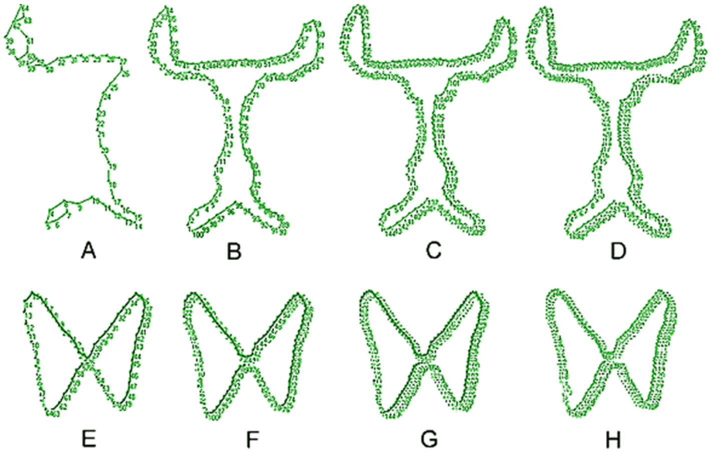


Figure C.3: Automatic annotation with network size of $64$ (Image A, E), $100$ (Image B, F), $144$ (Image C, G) and $164$ (Image D, H) nodes for two groups of the MRI volumes of the ventricles.

Table C.1 shows the total variance achieved by maps containing varying number of nodes $(64, 100, 144, 169)$ used for the automatic annotation (Figure C.3). The map of $100$ nodes is the most compact since it achieves the least variance compared to $64$, $144$ and $169$ nodes among the four groups. Figure C.4 shows superimposed the mean shapes of each group and for all nodes. The red shape referring to the

Table C.1: A quantitative comparison of various nodes adapted to the ventricle model with total variance per group

| Groups | 64 (nodes) | 100 (nodes) | 144 (nodes) | 169 (nodes) |
|---|---|---|---|---|
| $V_{T_1}$ | 9.8340 | 1.9385 | 3.9668 | 3.9235 |
| $V_{T_2}$ | 13.1873 | 1.7284 | 4.3672 | 3.1617 |
| $V_{T_3}$ | 6.7822 | 2.0109 | 3.2260 | 4.0057 |
| $V_{T_4}$ | 2.2567 | 1.6198 | 2.8398 | 3.5861 |

$100$ nodes is the most compact mean shape.

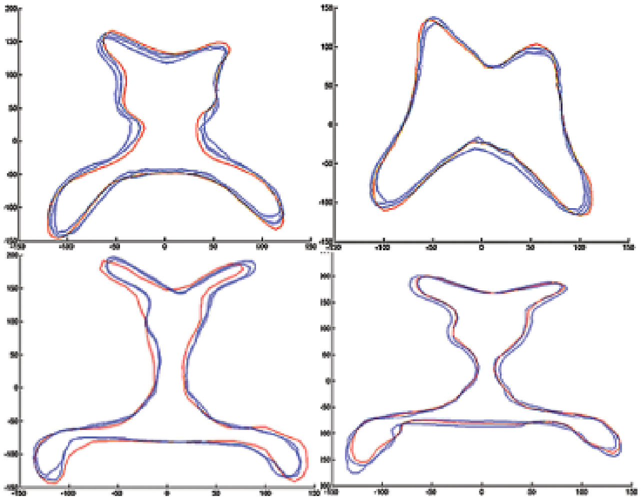


Figure C.4: The means of the four groups and for different nodes. The blue outlines represent the means of the 64, 144 and 169 nodes. The red outline represents the most compact mean achieved with the mapping of 100 nodes.

We have tested and compared our method with two other SOMs, the Kohonen map and the NG map. Figure C.5 shows the quantisation error for the three self-organising maps (SOMs) for different number of nodes. From Figure C.5 one can see that the distortion error for Kohonen is very big compared to NG and GNG but for GNG the results are slightly better to NG, since it has less distortion error thus better topology preservation, and the learning time is 20 times faster compared to NG (Figure C.7). However, as the number of neurons increases the distortion error decreases and stabilises for both networks. The better representation of the GNG over the NG network is also calculated by taking the Mean Squared Error (MSE) between the original shape and the back-projected from the PCA space. Figure C.6 shows the comparative diagram. Kohonen and NG networks assume that the numbers of weights are known *a priori* and do not change during the adaptation process. GNG overcomes this as it is a growth mechanism and new nodes are inserted based on local error measurements. Thus, GNG can give better preservation compared to the other two. The quantitative results show that GNG is significantly faster compared to Kohonen and NG. Figure C.7 shows a comparative diagram of the learning time of various SOMs and at different num-

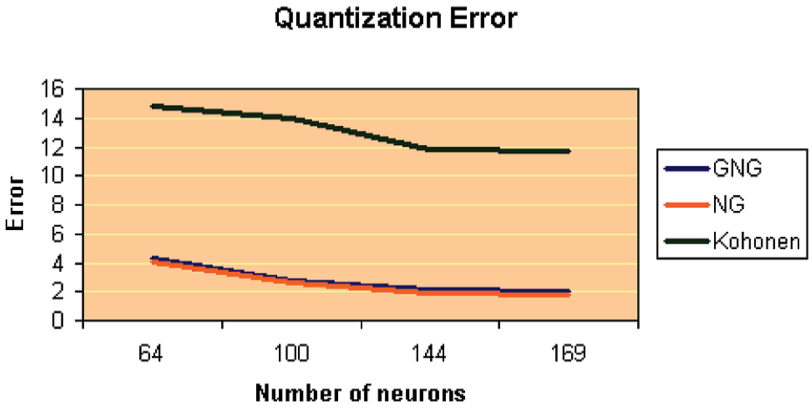


Figure C.5: Quadratic error for different SOMs and neurons.

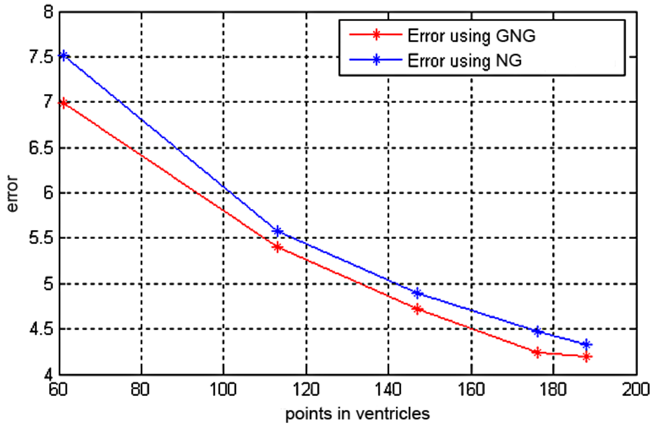


Figure C.6: Mean Squared Error for NG and GNG.

ber of nodes. The adaptation with 64 nodes is only 3 sec with GNG compared to the 57 sec and 52 sec with Kohonen and NG, but with 64 nodes the topology preservation in most of the shapes is lost independent of the selection of the SOM. A good adaptation with 100 and 144 nodes takes 6 and 11 seconds respectively at 1000 patterns to adapt to the contour of the ventricles.

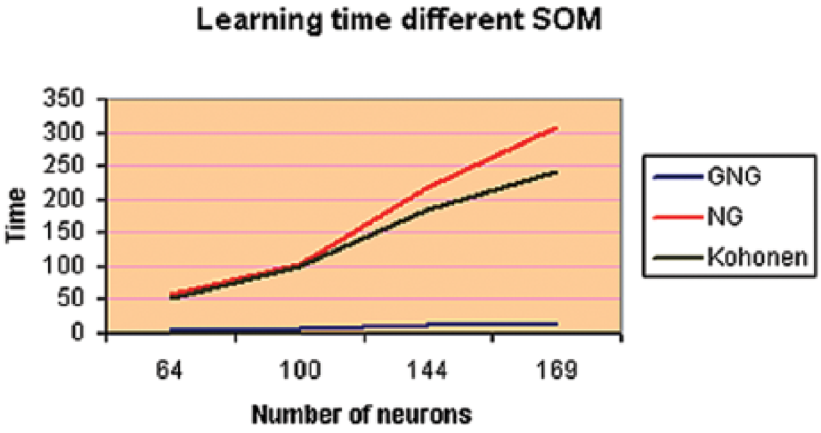


Figure C.7: Learning time for various SOMs and at various nodes.

# Appendix D

# Expectation-Maximization (EM) Algorithm

*In this appendix, we briefly discuss the background of a statistical model which has received wide application in Computer Vision. In particular, we discuss the Expectation-Maximization (EM) algorithm, which is being used to find maximum likelihood estimators in a problem with unobserved data.*

## D.1 Introduction

The classic EM algorithm can be dated back to Dempster, Laird, and Rubin's paper in 1977 [43]. It is a very general parameter estimation method applicable to many statistical models such as Mixture-of-Experts (MOE), Gaussian Mixture Models (GMM), and Vector Quantisation (VQ). These models are inter-related with VQ being a special case of GMM, which in turn is a special case of the more general MOE [101]. In GMM, as used in the work described in Chapter 4, EM is seeking a maximum likelihood solution by iterating two steps, the Estimation (E) and the

Maximization (M). The M-step maximizes a likelihood function that is refined in each iteration by the E-step. The advantages of the algorithm are that it avoids the calculation and storage of derivatives, is usually faster to converge than general purpose algorithms, and can also be extended to deal with missing data [68, 141].

### D.1.1 EM Algorithm for GMMs

The following notations are adopted:

- $X = \{x_t \in \mathbb{R}^D; t = 1, ......, T\}$ is the observation sequence, where $T$ is the number of observations and $D$ is the dimensionality of $x_t$.
- $C = \{C^{(1)}, .....C^{(J)}\}$ is the set of cluster mixture labels, where $J$ is the number of mixture components.
- $Z = \{z_t \in \mathbb{C}; t = 1, ......, T\}$ is the set of unobserved data.
- $\theta = \{\theta^{(j)}; j = 1, ......, J\}$ is the set of unknown parameters that define the density function for approximating the true probability density of $X$.
- $\theta^{(j)} = \{\pi^{(j)}, \varphi^{(j)}\}$ where $\pi^{(j)}$ denotes the prior probability of the $j-th$ component density and $\varphi^{(j)}$ defines the $j-th$ component density.

In a mixture model a density distribution is expressed as a linear combination of basis functions, for example, a linear combination of Gaussians.

#### D.1.1.1 A GMM

Assume a Gaussian mixture model:

$$\theta = \{\pi^{(j)}, \varphi^{(j)} = \{\mu^{(j)}, \Sigma^{(j)}\}; j = 1, ......, J\} \tag{D.1}$$

where $\pi^{(j)}$ denotes the prior probability of expert $j$, and $\varphi^{(j)} = \{\mu^{(j)}, \Sigma^{(j)}\}$ denotes the parameters mean $\mu^{(j)}$, and full-rank covariance matrix $\Sigma^{(j)}$ of the expert. The GMM's output is given by:

$$p(x_t|\theta) = \sum_{j=1}^{J} \pi^{(j)} p(x_t|\delta_t^{(j)} = 1, \varphi^{(j)}) \tag{D.2}$$

where

$$p(x_t|\delta_t^{(j)} = 1, \varphi^{(j)}) = (2\pi)^{-\frac{D}{2}} |\Sigma^{(j)}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(x_t - \mu^{(j)})^T (\Sigma^{(j)})^{-1} (x_t - \mu^{(j)})\} \tag{D.3}$$

is the $j - th$ Gaussian density of the GMM.

The method for determining the parameters of a Gaussian mixture model from a data set is based on maximising the data likelihood.

$$L(X|\theta_n) \equiv \log p(X|\theta_n) = \sum_{Z} P(Z|X, \theta_n) \log p(X|\theta_n) \tag{D.4}$$

Because the likelihood is a differentiable function it is possible to use general purpose optimisation algorithms. One such iterative approach is the EM algorithm which provides fast convergence.

### D.1.2 EM steps

After the initialisation of $\theta_0$, the EM iteration is as follows:

1. **E-step**. As we do not know the class labels, but do know their probability distribution, what we can do is to use the expected values of the class labels given the current parameters.

   For the $n - th$ iteration we form the function $Q(\theta|\theta_n)$ as follows:

$$\begin{aligned} Q(\theta|\theta_n) &= E\{\log p(Z, X|\theta)|X, \theta_n\} \\ &= \sum_{t=1}^{T} \sum_{j=1}^{J} E\{\delta_t^{(j)}|x_t, \theta_n\} \log[p(x_t|\delta_t^{(j)} = 1, \varphi^{(j)})\pi^{(j)}] \end{aligned} \tag{D.5}$$

and define

$$h_n^{(j)}(x_t) \equiv E\{\delta_t^{(j)}|x_t, \theta_n\} = P(\delta_t^{(j)} = 1|x_t, \theta_n) \tag{D.6}$$

Using Bayes' theorem we can calculate $h_n^{(j)}(x_t)$ as:

$$h_n^{(j)}(x_t) = \frac{p(x_t|\delta_t^{(j)} = 1, \varphi_n^{(j)})\pi_n^{(j)}}{\sum_{k=1}^{J} p(x_t|\delta_t^{(k)} = 1, \varphi_n^{(k)})\pi_n^{(k)}} \tag{D.7}$$

which is actually the expected posterior distribution of the class labels given the observed data. In other words, the probability that $x_t$ belongs to group $j$ given the current estimates $\theta_n$ is given by $h_n^{(j)}(x_t)$. The calculation of $Q$ is the E-step of the algorithm and determines the best guess of the membership function $h_n^{(j)}(x_t)$.

2. To compute the new set of parameter values of $\theta$ (denoted as $\theta^*$) we optimise $Q(\theta|\theta_n)$; that is:

$$\theta^* = \arg\max_{\theta} Q(\theta|\theta_n) \tag{D.8}$$

This is the M-step of the algorithm. Specifically, the steps are:

- Maximise $Q(\theta|\theta_n)$ with respect to $\theta$ to find $\theta^*$.
- Replace $\theta_n$ by $\theta^*$.
- Increment $n$ by $1$ and repeat the E-step until convergence.

To determine $\mu^{(k)*}$, differentiate $Q$ with respect to $\mu^{(k)}$ and equate to zero $(\frac{\vartheta Q(\theta|\theta_n)}{\vartheta \mu^{(k)}} = 0)$ which gives:

$$\mu^{(k)*} = \frac{\sum_{t=1}^{T} h_n^k(x_t)x_t}{\sum_{t=1}^{T} h_n^k(x_t)} \tag{D.9}$$

To determine $\Sigma^{(k)*}$, differentiate $Q$ with respect to $\Sigma^{(k)}$ and equate to zero $(\frac{\vartheta Q(\theta|\theta_n)}{\vartheta \Sigma^{(k)}} = 0)$ which gives:

$$\Sigma^{(k)*} = \frac{\sum_{t=1}^{T} h_n^k(x_t)(x_t - \mu^{(k)*})(x_t - \mu^{(k)*})^T}{\sum_{t=1}^{T} h_n^k(x_t)} \tag{D.10}$$

To determine $\pi^{(k)*}$, maximise $Q(\theta|\theta_n)$ with respect to $\pi^{(k)}$ subject to the constraint $\Sigma_{j=1}^{J}\pi^j = 1$ which gives:

$$\pi^{(k)*} = \frac{1}{T}\sum_{t=1}^{T} h_n^k(x_t) \tag{D.11}$$

A detailed derivation of the above equations is given in [101].

# Appendix E

# Publications

This thesis is mainly based on articles that were published during the research work presented below in an inverse chronological order.

**2011**

A. Angelopoulou, A. Psarrou, and J. García. A Growing Neural Gas Algorithm with Applications in Hand Modelling and Tracking. *In Proc. of the* $11^{th}$ *International Work-Conference on Artificial Neural Networks, IWANN 2011, Advances in Computational Intelligence, LNCS 6692*, pages 236–243, Springer, 2011.

**2010**

A. Angelopoulou, A. Psarrou, J. García, and G. Gupta. Tracking Gestures using a Probabilistic Self-Organising Network. *In Proc. of the International Joint Conference on Neural Networks (IJCNN 2010), IEEE WCCI 2010*, pages 1–7, IEEE Catalogue Number: CFP10IJS-DVD, ISBN: 978-1-4244-6917-8, 2010.

**2009**

G. Gupta, A. Psarrou, and A. Angelopoulou. Generic colour image segmentation via multi-stage region merging. *In Proc. of the* $10^{th}$ *International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'09*, pages 185–188, IEEE Xplore, 2009.

**2008**

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Active-GNG: Model Acquisition and Tracking in Cluttered Backgrounds. *In Proc. of the ACM workshop on Vision Networks for Behaviour Analysis, VNBA 2008, in conjunction with the ACM Multimedia*, pages 17–22, 2008.

**2007**

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Nonparametric Modelling and Tracking with *Active*-GNG. *Human Computer Interaction, IEEE International Workshop, ICCV-HCI 2007, in conjunction with the ICCV 2007, LNCS 4796*, pages 98–107, Springer, 2007.

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Robust Modelling and Tracking of NonRigid Objects Using *Active*-GNG. *IEEE Workshop on Non-rigid Registration and Tracking through Learning, NRTL 2007, in conjunction with ICCV 2007*, IEEE Xplore, 2007.

**2006**

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 Com-*

*puter Vision, ECCV 2006, LNCS 3951*, pages 313–324, 2006.

J. García, A. Angelopoulou, and A. Psarrou. Growing Neural Gas (GNG): A Soft Competitive Learning Method for $2D$ Hand Modelling. *Transactions on Information and Systems, E89-D(7):2124-2131*, Oxford University Press, ISSN: 0916-8532, 2006.

**2005**

A. Angelopoulou, A. Psarrou, J. García, and R. Kenneth. Automatic Landmarking of $2D$ Medical Shapes Using The Growing Neural Gas Network. *In Proc. of the IEEE Workshop on Computer Vision for Biomedical Image Applications, CVBIA 2005, LNCS 3765*, pages 210–219, Springer, 2005.

**2004**

A.N. Angelopoulou, and A. Psarrou. Evaluating Statistical Shape Models for Automatic Landmark Generation on A Class of Human Hands. *ISPRS-International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume 35: Part 3, Natural Resources*, pages 749–753, ISSN: 1682-1750, 2004.

# Bibliography

[1] I. Albrecht, J. Haber, and H. Seidel. Construction and animation of anatomically based human hand models. *In Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 98–109, 2003.

[2] C. Alonso-Montes and M. F. González Peredo. 3D Object Surface Reconstruction Using Growing Self-organised Networks. *In Proc. of the* $9^{th}$ *Iberoamerican Congress on Pattern Recognition, CIARP 2004, LNCS 3287*, pages 163–170, 2004.

[3] A. Angelopoulou, J. García, and A. Psarrou. Learning $2$D 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.

[4] A. Angelopoulou, A. Psarrou, and J. García. A Growing Neural Gas Algorithm with Applications in Hand Modelling and Tracking. *Advances in Computational Intelligence -* $11^{th}$ *International Work-Conference on Artificial Neural Networks, IWANN 2011, LNCS 6692*, pages 236–243, 2011.

[5] A. Angelopoulou, A. Psarrou, J. García, and R. Kenneth. Automatic Landmarking of $2$D Medical Shapes Using the Growing Neural Gas

Network. *In Proc. of the IEEE Workshop on Computer Vision for Biomedical Image Applications, CVBIA 2005, LNCS 3765*, pages 210–219, 2005.

[6] A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Robust Modelling and Tracking of Nonrigid Objects Using Active-GNG. *IEEE Workshop on Non-rigid Registration and Tracking through Learning, NRTL 2007, in conjuction with ICCV 2007*, pages 1–7, 2007.

[7] A. Atsalakis and N. Papamarkos. Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas. *Engineering Applications of Artificial Intelligence*, 19(7):769–786, 2006.

[8] C. Baillard, P. Hellier, and P. Barillot. Segmentation of 3D brain structures using level sets and dense registration. *IEEE Workshop on mathematical Methods on Biomedical Image Analysis*, pages 94–101, 2000.

[9] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Trans. on Systems, Man and Cybernetics, Part B*, 29(6):786–801, 1999.

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

[11] 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.

[12] E. Bavafa and M. Yazdanpanah. Image Compression using an Enhanced Self Organizing Map Algorithm with Vigilance Parameter. *In Proc. of the*

*International Joint Conference on Artificial Neural Networks, IJCNN'06*, pages 944–949, 2006.

[13] S. Belongie, J. Malik, and J. Puzicha. Matching Shapes. *In Proc. of the* $8^{th}$ *Internationl Conference on Computer Vision*, 1:454–463, 2001.

[14] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[15] C. M. Bishop. Neural Networks for Pattern Recognition. *Oxford Uni. Press*, 1995.

[16] H. Boehme, A. Brakensiek, U. Braumann, M. Krabbes, and H. Gross. Neural Networks for Gesture-based Remote Control of a Mobile Robot. *In Proceedings of the IEEE World Congress on Computational Intelligence*, 1:372–377, 1998.

[17] F. L. Bookstein. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–244, 1997.

[18] L. Bougrain and F. Alexandre. Unsupervised Connectionist Clustering Algorithms for a better Supervised Prediction: Application to a radio communication problem. *In Proc. of the International Join Conference on Neural Networks, IJCNN1999*, 5:3451–3456, 1999.

[19] H. Bozma and J. Duncan. Model-based recognition of multiple deformable objects using a game-theoretic framework. *Information Processing in Medical Imaging, LNCS 1991*, 511:358–372, 1991.

[20] C. Cdras and M. Shah. Motion-based recognition: a survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[21] S. Caetano, S. Olabarriaga, and B. A.C. Performance Evaluation of Single and Multiple-Gaussian Models for Skin Color Modeling. *In Proc. of the Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*, pages 275–282, 2002.

[22] X. Cao and P. Suganthan. Video shot motion characterization based on hierarchical overlapped growing neural gas networks. *Multimedia Systems*, 9(4):378–385, 2003.

[23] 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.

[24] G. Cheng and A. Zell. Double Growing Neural Gas for Disease Diagnosis. *In Proc. of Artificial Neural Networks in Medicine and Biology Conference (ANNIMAB-1)*, pages 309–314, 2000.

[25] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color Image Segmentation: Advances and Prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

[26] S. Cheng, H. Fu, and H. Wang. Model-Based Clustering by Probabilistic Self-Organizing Maps. *IEEE Trans. on Neural Networks*, 20(5):805–826, 2009.

[27] S. Cheung and C. Kamath. Robust Techniques for Background Subtraction in Urban Traffic Video. *In Proc. SPIE Visual Communications and Image Processing*, pages 881–892, 2004.

[28] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *In Proc. of Computer Vision and Image Understanding (CVIU)*, 89:114–141, 2003.

[29] I. Cohen, N. Ayache, and P. Sulger. Tracking Points on Deformable Objects Using Curvature Information. *In Proc. of the* $2^{nd}$ *European Conference on Computer Vision*, pages 458–466, 1992.

[30] T. F. Cootes and C. J. Taylor. *Statistical* M*odels of* A*ppearance for* C*omputer* V*ision*. Tech. report on Active Shape Models and Active Appearance Models, 2005.

[31] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training Models of Shape from Sets of Examples. *In Proc. of the* $3^{rd}$ *British Machine Vision Conference*, pages 9–18, 1992.

[32] 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.

[33] J. A. F. Costa and R. S. Oliveira. Cluster Analysis using Growing Neural Gas and Graph Partitioning. *In Proc. of the International Joint Conference on Neural Networks, IJCNN2007*, pages 3051–3056, 2007.

[34] T. F. Cox and M. A. A. Cox. *Multidimensional* S*caling*. Chapman and Hall, 2001.

[35] A. Cretu, E. Petriu, and P. Payeur. Evaluation of growing neural gas networks for selective 3D scanning. *In Proc. of IEEE International Workshop on Robotics and Sensors Environments*, pages 108 – 113, 2008.

[36] J. Crowley, F. Berard, and J. Coutaz. Finger Tracking as an Input Device for Augmented Reality. *In International Workshop on Gesture and Face Recognition*, pages 195–200, 1995.

[37] Z. Cselényi. Mapping the dimensionality, density and topology of data: the growing adaptive neural gas. *Computer Methods and Programs in Biomedicine*, 78(2):141–156, 2005.

[38] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, I:886–893, 2005.

[39] H. R. Davies. Learning Shape: Optimal Models for Analysing Natural Variability. *PhD Thesis, University of Manchester*, 2002.

[40] 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 Transactions on Medical Imaging*, 21(5):525–537, 2002.

[41] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer Verlag, 2008.

[42] F. De la Torre and M. Black. Probabilistic Principal Component Analysis. *In Proc. of the IEEE International Conference on Computer Vision*, I:362–369, 2001.

[43] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B 39(1):1–38, 1977.

[44] Y. Ding, J. McAllister II, B. Yao, N. Yan, and A. Canady. Axonal damage associated with enlargement of ventricles during hydrocepahlus: A silver impregnation study. *Neurological Research*, 23(6):581–587, 2001.

[45] K. Doherty, R. Adams, and N. Davey. Hierarchical Growing Neural Gas. *Adaptive and Natural Computing Algorithms*, pages 140–143, 2005.

[46] I. Dryden and K. V. Mardia. *The* S*tatistical* A*nalysis of* S*hape*. Willey, London, 1998.

[47] J. S. Duncan, R. L. Owen, L. H. Staib, and P. Anandan. Measurement of non-rigid motion using contour shape descriptors. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 318–324, 1991.

[48] N. Duta, A. K. Jain, and M.-P. Dubuisson-Jolly. Automatic construction of 2D shape models. *IEEE Transactions on PAMI*, 23(5):433–446, 2001.

[49] S. Eddy. Hidden Markov Models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[50] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. *In Proc. of the International Conference on Face And Gesture Recognition*, pages 300–305, 1998.

[51] 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.

[52] A. Fernandez, M. Ortega, B. Cancela, and M. G. Penedo. Contextual and skin color region information for face and arms location. *In Proc. of the 13th*

*international conference on Computer Aided Systems Theory - EUROCAST 2011*, 6927/2012:616–623, 2012.

[53] F. Flórez-Revuelta, M. García-Chamízo, J. García-Rodríguez, and A. Hermández. Representation of 2D Objects with a Topology Preserving Network. *In Proc. of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS'02)*, pages 267–276, 2002.

[54] D. A. Forsyth and P. Ponce. *Computer Vision A Modern Approach*. Prentice Hall, 2003.

[55] H. Frezza-Buet. Following non-stationary distributions by controlling the vector quantisation acccuracy of a growing neural gas network. *Neurocomputing*, 71(7-9):1191–1202, 2008.

[56] N. Friedman and S. Russel. Image segmentation in video sequences: A probabilistic approach. *In Proc. of the Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[57] B. Fritzke. Growing Cell Structures A Self-organising Network for Unsupervised and Supervised Learning. *Neural Networks*, 7:1441–1460, 1993.

[58] B. Fritzke. Fast learning with incremental RBF networks. *Neural Processing Letters*, 1(1):2–5, 1994.

[59] B. Fritzke. Growing Cell Structures - a Self-organising Network for Unsupervised and Supervised Learning. *The Journal of Neural Networks*, 7(9):1441–1460, 1994.

[60] B. Fritzke. A growing Neural Gas Network Learns Topologies. *In Advances in Neural Information Processing Systems 7 (NIPS'94)*, pages 625–632, 1995.

[61] B. Fritzke. A Self-Organising Network that can follow Non-stationary Distributions. *In Proc. of the International Conference on Artificial Neural Networks, ICANN-97*, pages 613–618, 1997.

[62] S. Furao and O. Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106, 2006.

[63] J. Garcia. *Self-organizing neural netwrok model to represent objects and their movement in realistic scenes*. PhD Thesis, University of Alicante, 2009.

[64] J. García-Rodríguez, A. Angelopoulou, and A. Psarrou. Growing neural gas (GNG): A soft competitive learning method for 2D hand modeling. *IEICE Transactions on Information and Systems*, E89-D(7):2124–2131, 2006.

[65] J. García-Rodríguez, F. Flórez-Revuelta, and M. García-Chamízo. Image Compression Using Growing Neural Gas. *In Proc. of the International Joint Conference on Artificial Neural Networks*, pages 366–370, 2007.

[66] J. García-Rodríguez, F. Flórez-Revuelta, and M. García-Chamízo. Learning topologic maps with growing neural gas. *Lecture Notes in Artificial Intelligence*, 4693(2):468–476, 2007.

[67] B. Gelman, S. Dholakia, S. Casper, T. Kent, M. Cloyd, and D. Freeman. Expansion of the cerebral ventricles and correlation with acquired immunodeficiency syndrome neuropathology in 232 patients. *Arch Pathol Lab Med*, 120(9):866–871, 1996.

[68] Z. Ghahramani and M. Jordan. Supervised learning from incomplete data via an em approach. *Advances in Neural Information Processing Systems*, 6:120–127, 1994b.

[69] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness. New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.

[70] G. Goodhill and T. Sejnowski. A unifying objective function for topographic mappings. *Neural Computation*, 9:12911304, 1997.

[71] V. Govindaraju. Locating human faces in photographs. *International Journal of Computer Vision*, 19(2):129–146, 1996.

[72] J. C. Gower. Generalised Procrustes Analysis. *Psycometrica*, (40):33–51, 1975.

[73] H. Guan, R. Feris, and M. Turk. The Isometric Self-Organizing Map of 3D Hand Pose Estimation. *In Proc. of Automatic Face and Gesture Recognition, FGR2006*, pages 263–268, 2006.

[74] D. Hall, J. Nascimento, P. Andrade, M. Plinio, S. List, R. Emonent, R. Fisher, J. Santos-Victor, and J. Crowley. Comparison of Target Detection Algorithms using Adaptive Background Models. *In Proc. of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 113–120, 2005.

[75] G. Hamarneh. *Active* Shape Models: Modeling Shape Variations *and* Gray Level Information *and an* Application *to* Image Search *and* Classification. Tech. report on Active Shape Models, 1998.

[76] J. Hammerton. Learning to Segment Speech with Self-Organising Maps. *Language and Computers, Computational Linguistics in the Netherlands*, 14:51–64, 2003.

[77] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[78] D. Heinke and F. Hamker. Comparing neural networks: a benchmark on growing neural gas, growing cell structures, and fuzzy artmap. *IEEE Transactions on Neural Networks*, 9(6):1279–1291, 1998.

[79] A. Hill and C. Taylor. Model based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5):295–300, 1992.

[80] 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.

[81] A. Hill and C. J. Taylor. Automatic Landmark Generation for Point Distribution Models. *In Proc. of the $5^{th}$ British Machine Vision Conference*, 2:429–438, 1994.

[82] A. Hill and C. J. Taylor. A Method of Non-rigid Correspondence for Automatic Landmark Identification. *In Proc. of the $7^{th}$ British Machine Vision Conference*, pages 323–332, 1996.

[83] G. Hinton, C. Williams, and M. Revow. Adaptive Elastic Models for Hand-Printed Character Recognition. *Neural Information Processing Systems - NIPS*, pages 512–519, 1991.

[84] Y. Holdstein and A. Fischer. Three-dimensional surface reconstruction using meshing growing neural gas (MGNG). *The Visual Computer: International Journal of Computer Graphics*, 24(4):295–302, 2008.

[85] H. Hotelling. Analysis of a Complex of Statistical Variables with Principal Components. *Journal of Educational Psychology*, 24:417–441, 1933.

[86] C. Hung and S. Wermter. A novel self-organising clustering model for time-event documents. *The Electronic Library*, 26(2):260–272, 2008.

[87] I. T. Jolliffe. *Principal* C*omponent* A*nalysis*. Springer-Verlag, 2002.

[88] M. Jones and J. Rehg. Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[89] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recongition*, 40(3):1106–1122, 2007.

[90] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[91] C. Kambhamettu and D. B. Goldgof. Point Correspondence Recovery in Non-Rigid Motion. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–227, 1992.

[92] 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.

[93] T. Kohonen. *Topology* R*epresenting* N*etworks*. Springer Verlag, 1994.

[94] T. Kohonen. *Self-organising maps*. Springer Verlag, 2001.

[95] H. Koike, Y. Sato, and Y. Kobayashi. Integrating Paper and Digital Information on Enhanced Desk: A Method for Real Time Finger Tracking on an Augmented Desk System. *ACM Transactions on Computer-Human Interaction*, 8(4):307–322, 2001.

[96] A. Koschan, S. Kang, J. Paik, B. Abidi, and M. Abidi. Color active shape models for tracking non-rigid objects. *Pattern Recognition Letters, Special issue: Colour image processing and analysis*, 24(11):1751–1765, 2003.

[97] A. C. W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by genetic algorithm. *In Proc. of the* $15^{th}$ *Conference on Information Processing in Medical Imaging*, pages 1–14, 1997.

[98] A. C. W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2(4):303–314, 1998.

[99] H. Kruppa. *Object* Detection *using* Scale-specific Boosted Parts *and a* Bayesian Combiner. PhD Thesis, ETH Zrich, 2004.

[100] H. Kruppa, C. Santana, and B. Sciele. Fast and Robust Face Finding via Local Context. *In Proc. of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 157–164, 2003.

[101] S. Kung, M. Mak, and S. Lin. *Biometric* Authentication: A Machine Learning Approach. Prentice Hall, 2004.

[102] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.

[103] M. Lee and R. Nevatia. Integrating Component Cues for Human Pose Estimation. *In Proc. of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 41–48, 2005.

[104] B. Leibe, E. Seemann, and B. Sciele. Pedestrian Detection in Crowded Scenes. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, I:878–885, 2005.

[105] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. *In Proc. of the IEEE International Conference on Computer Vision*, I:626–633, 2003.

[106] P. Lipson, A. Yuille, D. O'Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal. Deformable Templates for Feature Extraction from Medical Images. *1st European Conference on Computer Vision*, pages 413–417, 1990.

[107] P. Lisboa, B. Edisbury, and A. Vellido. *Business* A*pplications of* N*eural* N*etworks:* T*he* S*tate-*O*f-*T*he-*A*rt of* R*eal-*W*orld* A*pplications*. World Scientific Publishing Company, 2000.

[108] M. Love. Probability theory. *Graduate Texts in Mathematics*, 2(46), 1978.

[109] L. Lucchese and S. Mitra. Color Image Segmentation: A State-of-the-Art Survey. *In Proc. of the Indian National Science Academy (INSA-A)*, 67(2):207–221, 2001.

[110] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface quality hand tracking. *In Proc. of the* $6^{th}$ *European Conference on Computer Vision, ECCV 2000*, 2:3–19, 2000.

[111] S. Marsland, U. Nehmzow, and J. Shapiro. Novelty Detection for Robot Neotaxis. *In Proc. of the 2nd International Symposium on Neural Compuatation*, pages 554–559, 2000.

[112] S. Marsland, U. Nehmzow, and J. Shapiro. A Real-Time Novelty Detector for a Mobile Robot. *In Proc. of EUREL European Advanced Robotics Systems Masterclass and Conference*, 2000.

[113] S. Marsland, U. Nehmzow, and J. Shapiro. A Real Time Novelty Detector For A Mobile Robot. *In Proc. of the EUREL Conference on Advanced Robotics Systems*, 2000.

[114] S. Marsland, J. Shapiro, and U. Nehmzow. A self-organizing network that grows when required. *Neural Networks*, 15:1041–1058, 2002.

[115] T. Martinez. Competitive Hebbian learning rule forms perfectly topology preserving maps. *ICANN93: International Conference on Artificial Neural Networks*, pages 427–434, 1993.

[116] 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.

[117] T. Martinez and K. Schulten. A neural-gas network learns topologies. *Artificial Neural Networks*, pages 397–402, 1991.

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

[119] R. McCarley, C. Wible, M. Frumin, Y. Hirayasu, J. Levitt, I. Fisher, and S. M. MPI anatomy of schizophrenia. *Biol Psychiatry*, (45):1099–1119, 1999.

[120] W. McCulloch and W. Pitts. *A logical calculus of the ideas imminent in nervous activity*. Number 5. Bulletin of Mathematical Biophysics, 1943.

[121] M. Mignotte. Segmentation by fusion of histogram-based k-means clusters in different color spaces. *IEEE Trans. on Image Processing*, 5(17):780–787, 2008.

[122] V. Nair and J. Clark. An Unsupervised, Online Learning Framework for Moving Object Detection. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recongition*, II:317–324, 2004.

[123] 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.

[124] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. *In Proc. of the* $4^{th}$ *International Conference on Computer Vision, ICCV'93*, pages 275–279, 1993.

[125] T. Ogura, K. Iwasaki, and C. Sato. Topology representing network enables highly accurate classification of protein images taken by cryo electron-microscope without masking. *Journal of Structural Biology*, 143(3):185–200, 2003.

[126] R. O'Hagan, A. Zelinsky, and S. Rougeaux. Visual Gesture Interfaces for Virtual Environments. *Interacting with Computers*, 14(3):231–250, 2002.

[127] N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modelling Human Interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[128] C. Papageorgiou, M. Oren, and T. Poggio. A General Framework for Object Detection. *In Proc. of the IEEE International Conference on Computer Vision*, pages 555–562, 1998.

[129] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Sciences*, 6(2):559–572, 1901.

[130] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modelling and Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):715–729, 1991.

[131] M. Piccardi. Background subtraction techniques: a review. *In Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104, 2004.

[132] P. Picton. *Neural* Ne*tworks (*G*rassroots)*. Palgrave Macmillan, 2000.

[133] Y. Prudent and A. Ennaji. An incremental growing neural gas learns topologies. *In Proc. of the International Joint Conference on Neural Networks,IJCNN'05*, 2:1211–1216, 2005.

[134] A. K. Qin and P. Suganthan. Robust growing neural gas algorithm with application in cluster analysis. *Neural Networks*, 17(8-9):1135–1148, 2004.

[135] Y. Raja, S. McKenna, and S. Gong. Colour Model Selection and Adaptation in Dynamic Scenes. *In Proc. of the European Conference on Computer Vision*, pages 460–474, 1998.

[136] 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.

[137] R. Redner and H. Walker. Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239, 1984.

[138] R. Rêgo, A. Araújo, and F. de Lima Neto. Growing Self-Organizing Maps for Surface Reconstruction from Unstructured Point Clouds. *In Proc. of the International Joint Conference on Artificial Neural Networks, IJCNN'07*, pages 1900 – 1905, 2007.

[139] C. Rehtanz and C. Leder. Stability assessment of electric power systems using growing neural gas and self-organizing maps. *In Proc. of ESSAN 2000*, pages 401–406, 2000.

[140] A. Reyes and C. Constantino. Image Segmentation with Kohonen Neural Network Self Organising Maps. *In International Conference on Telecommunications ICT*, 2000.

[141] B. Ripley. *Pattern* Recognition *and* Neural Networks. Cambridge University Press, 1996.

[142] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[143] H. Ritter and K. Schulten. Topology conserving mappings for learning motor tasks. *Neural Networks for Computing, AIP Conf. Proc.*, pages 376–380, 1986.

[144] J. Rivera-Rovelo, S. Herold, and E. Bayro-Corrochano. Object Segmentation Using Growing Neural Gas and Generalized Gradient Vector Flow in the

Geometric Algebra Framework. *Progress in Pattern Recognition, Image Analysis and Applications, 11th Iberoamerican Congress in Pattern Recognition, CIARP 2006*, 4225:306–315, 2006.

[145] P. M. Roth. *On-line* Conservative Learning. PhD Thesis, Graz University of Technology, 2008.

[146] S. Scalroff and A. Pentland. Modal Matching for Correspondence and Recognition. *IEEE Trans. on Pattern Aanalysis and Machine Intelligence*, 17(6):545–561, 1995.

[147] S. Schaal. Nonparametric Regression for Learning. *In Proc. of the Conference on Prerational Intelligence-Adaptive and Learning Behavior*, 1994.

[148] H. Schnack, P. Hulshoff, W. Baare, M. Viergever, and R. Kahn. Automatic segmentation of the ventricular system from mr images of the human brain. *NeuroImage*, 14:95–104, 2001.

[149] K. Schwerdt and J. Crowley. Robust Face Tracking using Color. *In Proc. of the International Conference on Automatic Face and Gesture Recognition*, pages 90–95, 2000.

[150] G. Scott. The Alternative Snake - and Other Animals. *In Proc. of the 3rd Alvey Vision Conference*, pages 341–347, 1987.

[151] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *In Proc. of the Royal Society of London*, B244:21–26, 1991.

[152] L. S. Shapiro and M. J. Brady. A Modal Approach to Feature-based Correspondence. *In Proc. of the* $2^{nd}$ *British Machine Vision Conference*, pages 78–85, 1991.

[153] J. Shlens. *A* T*utorial on* P*rinicpal* C*omponent* A*nalysis:* D*eriviation,* D*iscussion and* S*ingular* V*alue* D*ecomposition*. Tutorial on Principal Component Analysis, 2003.

[154] B. Simon. *Functional* I*ntegration and* Q*uantum* P*hysics*. Academic Press, 1979.

[155] J. Sivic, M. Everingham, and A. Zisserman. Person Spotting: Video Shot Retrieval for Face Sets. *In International Conference on Image and Video Retrieval*, pages 226–236, 2005.

[156] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. CL-Engineering, 2nd edition, 1998.

[157] A. Souza and J. Udupa. Automatic Landmark Selection for Active Shape Models. *Proccedings of SPIE*, 2005.

[158] L. H. Staib and J. S. Duncan. Parametrically deformable contour models. *In IEEE conference on Computer Vision and Pattern Recognition*, pages 427–430, 1989.

[159] Z. L. Stan and K. J. Anil. *Handbook of* F*ace* R*ecognition*. Springer Verlag, 2005.

[160] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, II:246–252, 1999.

[161] M. Strring. *Computer* V*ision and* H*uman* S*kin* C*olour*. PhD Thesis, Aalborg University, 2004.

[162] S. Stefan. Nonparametric Regression for Learning. *In Proc. of the Conference on Adaptive Behavior and Learning*, pages 123–133, 1994.

[163] M. B. Stegmann and D. D. Gomez. *A* B*rief* I*ntroduction To* S*tatistical* S*hape* A*nalysis*. Tech. report on Statistical Shape Models, 2002.

[164] E. Stergiopoulou and N. Papamarkos. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8):1141–1158, 2009.

[165] E. Stergiopoulou, N. Papamarkos, and A. Atsalakis. Hand Gesture Recognition via a New Self-organised Neural Network. *Progress in Pattern Recognition, Image Analysis and Applications*, 3773:891–904, 2005.

[166] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Distributed Occlusion Reasoning for Tracking with Nonparametric Belief Propagation. *In Advances in Neural Information Processing Systems*, 17:1369–1376, 2005.

[167] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual Hand Tracking Using Nonparametric Belief Propagation. *CVPRW '04 Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, 12:189, 2005.

[168] C. Sui. *Appearance-based hand gesture identification*. Master of Engineering, University of New South Wales, 2011.

[169] H. D. Tagare. Shape-Based Nonrigid Correspondence with Application to Heart Motion Analysis. *IEEE Trans. on Medical Imaging*, 18(7):570–579, 1999.

[170] H. D. Tagare, D. Shea, and A. Rangarajan. A Geometric Criterion for Shape-based Non-rigid Correspondence. *In Proc. of the* $5^{th}$ *Internationl Conference on Computer Vision*, pages 434–439, 1995.

[171] P. Vamplew and A. Adams. Recognition of Sign Language Gestures using Neural Networks. *Australian Journal of Intelligent Information Processing Systems*, 5(2):94–102, 1998.

[172] D. Vasquez and T. Fraichard. A Novel Self Organizing Network to perform Fast Moving Object Extraction from Video Streams. *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4857–4862, 2006.

[173] V. Vezhnevets, V. Sazonov, and A. Andreeva. A Survey on Pixel-Based Skin Color Detection Techniques. *In Proc. of the International Conference on Automatic Face and Gesture Recognition*, pages 90–95, 2000.

[174] T. Villman, M. Herrman, and T. Martinetz. Topology preservation in self organising feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256266, 1997.

[175] M. Walter. *Automatic* Model A*cquisition and* R*ecognition of* H*uman* G*estures*. PhD Thesis, University of Westminster, 2002.

[176] 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.

[177] H. Wang, H. Zheng, and F. Azuaje. Poisson-Based Self-Organizing Feature Maps and Hierarchical Clustering for Serial Analysis of Gene Expression

Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(2):163–175, 2007.

[178] A. Webb. *Statistical* Pattern Recognition, *2nd* Edition. Wiley-Blackwell, 2002.

[179] S. Widz, K. Revett, and D. Slezak. An Automated Multi-spectral MRI Segmentation Algorithm Using Approximate Reducts. *Rough Sets and Current Trends in Computing*, pages 815–824, 2004.

[180] S. Wong and S. Ranganath. Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 873–891, 2005.

[181] C. Wren, A. Azarbayejani, D. T., and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[182] Y. Wu, Q. Liu, and T. Huang. An adaptive Self-Organising Colour Segmentation Algorithm with Application to Robust Real-time Human Hand Localisation. *In Proc. of the IEEE Asian Conference on Computer Vision (ACCV)*, pages 1106–1111, 2000.

[183] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 8(3):359–369, 1998.

[184] J. Yang, W. Bang, E. Choi, S. Cho, J. Oh, J. Cho, S. Kim, E. Ki, and D. Kim. A 3D Hand-drawn Gesture Input Device Using Fuzzy ARTMAP-based Recognizer. *Journal of Systemics, Cybernetics and Informatics*, 4(3):1–7, 2009.

[185] M. Yang and N. Ahuja. Gaussian Mixture Model for Human Skin Color and Its Applications in Image and Video Databases. *In Proc. of SPIE99*, pages 458–466, 1999.

[186] Z. Yang and K. Chou. Mining Biological Data Using Self-Organizing Map. *J. Chem. Inf. Comput. Sci. 2003*, 43:1748–1753, 2003.

[187] H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19:780–784, 2006.

[188] A. L. Yuille, D. S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *Int. J. Computer Vision*, (8):99–112, 1992.

[189] M. Zaki Shireen and H. Yiu. Semi-supervised Growing Neural Gas for Face Recognition. *Intelligent Data Engineering and Automated Learning*, 5326:525–532, 2008.

[190] Q. Zhou and J. Aggarwal. Tracking and Classifying Moving Objects from Videos. *In Proc. of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 157–164, 2001.

[191] P. Zhu and P. M. Chirlian. On Critical Point Detection of Digital Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.

*"For the things we have to learn before we can do, we learn by doing"*

Aristotle (Greek philosopher, 384-322 BC)