uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Ana Maria Cretu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Experimental Data Acquisition and Modeling of 3D Deformable Objects using Neural Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Emil Petriu

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Gilles Patry

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Abdulmotaleb El Saddik

Rafik Goubran

George Giakos

Shervin Shrimohammadi

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Experimental Data Acquisition and Modeling of 3D Deformable Objects using Neural Networks

by

Ana-Maria Cretu

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the

Ph.D. degree in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-59546-6
Our file Notre référence
ISBN: 978-0-494-59546-6

To Pierre

To my dear mom and dad

Contents

List of Tables

List of Figures

Abstract

Nowadays there are many technologies and design tools available to accurately obtain and model the geometric shape and the color of objects. However, these methods are not able to provide any information about the elasticity of the objects. This thesis presents a general-purpose scheme for measuring, constructing and representing geometric and elastic behavior of deformable objects without *a priori* knowledge on the shape and the material that the objects under study are made of. The proposed solution is based on an advantageous combination of neural network architectures and an original force-deformation measurement procedure. An innovative non-uniform selective data acquisition algorithm based on self-organizing neural architectures (namely neural gas and growing neural gas) is developed to selectively and iteratively identify regions of interest and guide the acquisition of data only on those points that are relevant for both the geometric model and the mapping of the elastic behavior, starting from a sparse point-cloud of an object. Multi-resolution object models are obtained using the initial sparse model or the (growing or) neural gas map if a more compressed model is desired, and augmenting it with the higher resolution measurements selectively collected over the regions of interest. A feedforward neural network is then employed to capture the complex relationship between an applied force, its magnitude, its angle of application and its point of interaction, the object pose and the deformation stage of the object on one side, and the object surface deformation for each region with similar geometric and elastic behavior on the other side. The proposed framework works directly from raw range data and obtains compact point-based models. It can deal with different types of materials, distinguishes between the different stages of deformation of an object and models homogeneous and non-homogeneous objects as well. It also offers the desired degree of control to the user.

Acknowledgements

I would not have been able to complete this journey without the aid and support of several people and agencies over the past five years. I wish therefore to express my gratitude to my thesis supervisor Dr. Emil Petriu, for his continued encouragement and invaluable suggestions during these years. As well, I would like to thank my co-supervisor, Dr. Gilles Patry.

I am truly grateful to my beloved parents for their unquestioning faith in me and my abilities from the very beginning of my studies. Your love and support has helped to make all this possible and for that, and everything else, I thank you. I know now that if I can dream it, I can do it!

I owe my loving thanks to my dear husband for his untiring support and unlimited belief in me. Thank you for your unfailing patience and support in bringing this thesis to an end. Your unlimited love and understanding have been my greatest source of strength and courage to overcome all hardships throughout these years. Thank you for always being there for me!

My special thanks go as well to my dear friend Oana for her kind encouragements and to all my friends in the SMR and Discover labs.

I would like to thank Dr. G. Giakos. Dr. R. Goubran, Dr. A. El Saddik and Dr. S. Shirmohammadi for their valuable comments regarding the thesis work.

Special thanks go to Dr. Jochen Lang for his comments and help with the force/displacement data for the triceratops model, to Dr. Chad English from Neptec Design Group Inc. for providing access to the high resolution range data of the door.

I would like at the same time to thank the several agencies that made all this research work possible, particularly the Natural Sciences and Engineering Research Council of Canada, the Ontario Graduate Scholarship Program, the Communications and Information Technology Ontario Centre of Excellence and also the University of Ottawa that partially funded this endeavor.

1. Introduction

1.1. Statement of the Problem

Nowadays there are many technologies and design tools available to accurately obtain and model the geometric shape and the color of objects. However, these methods are not able to provide any information about the elasticity of the objects. Accurate deformable object modeling raises a complex combination of issues ranging from determining the required equipment to perform measurements, fusing different technologies available to improve the measurement process, interpreting and correlating these measurements and finding proper methods to model elastic data, to finding means to combine these measurements in composite geometric and elastic models and providing the user with ways to study the resulting models or interact with them.

Because of its complexity, modeling of elastic behavior of objects is of current interest to the international research community and also to Canadian research groups pursuing related studies on tactile and haptic modeling or on three-dimensional (3D) computer vision and solid modeling. These research efforts are demonstrated by a large number of papers studying different aspects of deformable objects over the past three decades. The fact that the majority of papers are relatively recent shows an increasing interest in the topic.

However, current research in deformable objects is mainly focusing on the modeling of inherently elastic objects for computer graphics and virtual environments and less on real measurements of elastic behavior. Most research concerns itself with simulations, and studies methods to respond to the computational cost of increasingly complex models and the requirements for real-time interaction with computer-generated deformable models. Difficulties are often encountered in conducting strain-stress relationship measurements for objects made of materials that exhibit nonlinear behavior, where a material model is unavailable or where a model is available for the object's material, but has limited accuracy. As a response to these challenges, the majority of applications leaves the choice for the selection of elastic parameters to the user or chooses some values for these parameters according to some *a priori* knowledge regarding the deformable object model. Such approaches are subjective and while they can be employed for certain types of applications that do not necessarily require physically-correct deformable models, but only some degree of physically plausible dynamic behavior, they cannot be employed where accuracy is expected.

In most of the cases where measurements of elasticity are performed, often a single probing of the object is used to elicit its elastic behavior. While this approach can produce satisfactory results for objects that are made of homogeneous elastic materials, it is unsuitable for heterogeneous or piecewise homogeneous objects that have varying elasticity in different parts of their bodies. On the other hand, the procedure to collect elastic data from each point is time consuming. These two aspects explain the considerable interest in finding fast sampling procedures for the measurement of the elastic properties of 3D object surfaces. Appropriate data acquisition control algorithms should be able to minimize the number of the sampled points by selecting only those points that are relevant to the elastic and geometric characteristics of the studied object. Additional problems in elasticity measurement arise from the need of relatively expensive equipment and complex procedures to measure or elicit elastic behavior. The results during measurements are not easily interpretable as long as no assumptions are made on the material of the object under study and the recuperation or modeling of elastic behavior poses additional challenges in the researched domain.

To sum up, the problem of data acquisition and modeling elastic objects is still and will still be under extensive research due on one side to the need to model or know elasticity in a plethora of applications, and due to its complexity, its sophisticated equipment requirement and to the difficulties encountered in conducting and interpreting strain-stress relationship measurements on the other side.

1.2. Objectives

The goal of this research is to develop a general-purpose scheme for measuring, constructing and representing geometric and elastic behavior of deformable objects without *a priori* knowledge on the material that the objects under study are made of.

The unique features that this research aims to achieve are:

- *Measurement of elastic behavior*: In the majority of recent publications, the selection of the physical constants is performed based on the visual appeal of the resulting deformable object or some values are chosen according to some *a priori* knowledge regarding the object under study. The proposed scheme uses an innovative in-house experimental setup to elicit and collect elastic behavior information. This procedure ensures that the resulting models are reflecting real world objects and that the methodology is repeatable. Several objects with different material properties are tested to validate the proposed approach.
- *Implicit modeling of elastic parameters with neural networks*: The majority of researchers assume the material of the objects under study to be linearly elastic and isotropic in order to simplify the computational load or make use of classical deformable object methods (e.g. finite-element method, boundary-element method) to recuperate the elasticity information and/or to model the objects. The implicit modeling of elasticity implies that the modeling of elasticity is inherent, without the need of recuperating elastic parameters explicitly. This avoids the heavy mathematical models and the difficulties encountered in recuperating elastic parameters for objects made of materials that exhibit nonlinear behavior or where a material model is unavailable. The proposed scheme is stand-alone and does not imply the use of heavy mathematical deformable objects representation techniques. The modeling of elasticity is done implicitly using neural networks. Moreover, the proposed modeling scheme allows for the construction of 2.5D object models, unlike some already existing 2D neural network models.
- *Selective acquisition of geometric and elastic data:* While a single probing point of the object might be enough for objects that have homogeneous elasticity throughout their bodies in order to elicit their elastic behavior, it is unsuitable for heterogeneous or piecewise homogeneous objects. The proposed scheme uses an innovative probing (and measuring) technique for the selective acquisition of geometric and elastic characteristics in a series of points on the object's surface. This ensures the capability of the application to deal not only with homogeneous objects from the elasticity point of view, but also with non-homogeneous, particularly piecewise-homogeneous objects as well. This approach is proven to

be appropriate in vision (for 3D geometry data acquisition) and tactile sensing (for elastic measurements acquisition).

- *Modeling of complex multi-resolution real objects*: Most of the existing models represent only objects that are made of homogeneous elastic materials or that are designed specifically for a certain application. The purpose of this research is to build a conformal representation of physical 3D objects based on sensor information about real world objects. The proposed modeling scheme makes use of innovative composite self-organizing and feedforward networks to describe the geometric and elastic behavior of deformable objects. The use of self-organizing architectures (that perform a sort of clustering) ensures that the resulting models are multi-resolution ones. This means that they can contain representations of the same object with different levels of detail, property that ensures important gains in computation and networking time and support of dynamic changes of the topology. The resulting objects are not specific for a certain application, but the system provides a representation for different real objects to be introduced in virtualized reality environments.

The ability to realistically measure and model the elasticity of objects comes into benefit and is applicable in the areas of CAD/CAM, quality control, and reverse engineering of elastic material components, interactive virtual environments for training, interactive virtual prototyping for industrial CAD-based process, digital evaluation and maintenance operation simulation, the computer games industry, as well as for many dexterous manipulation applications such as robotic assembly, tele-robotics, remote medical examination of patients, medical robotics.

In order to keep the area of research within manageable limits of complexity and sufficiently narrow, the work in this study is limited to the data acquisition and modeling areas in the context of 3D deformable objects. The framework presented here is a breach in a relatively unexplored area of modeling objects without assumptions on their material and it is meant to be a "proof of concept" work. The accent is on viable, innovative and easily applicable solutions to measure and model elastic behavior of real world objects using the available equipment and the experimental setup in SMRLab, SITE, University of Ottawa. These aspects will be further discussed in Chapter 3. While computation time is of concern to this study, the real-time behavior of the proposed scheme is beyond the purpose of the work. The emphasis is on accurate models as opposed to real-time behavior. In terms of interaction, the user is provided with a simple way of testing the elastic behavior of the modeled objects, but neither complicated interfaces, nor haptic feedback are provided at this stage. As well, no details are provided on the topic of robotic manipulation of the deformable objects under study.

1.3. Proposed Framework

This framework is the result of several experiments that have been performed and analyzed and it is a continuation of the research that we performed previously on rigid objects [1]. The novelty consists in the addition of the elastic behavior to the geometric models (with all the related issues such as: measurement equipment and procedure, result interpretation and modeling, and methods of constructing composite geometric and elastic models from the experimental data) and in an original modification of one of the methods that we proposed in [1] for a much broader purpose, in the context of selective data acquisition of relevant features of objects to improve the accuracy of the modeling results. The work here proposes and experimentally demonstrates that the use of neural networks is also appropriate as a means for the acquisition, construction and representation of geometric and elastic behavior of deformable 3D objects.

Neural networks have been employed in many applications related to object modeling, ranging from representation of data of different types (range, intensity) from different sources (structured light and shape from shading) to the modeling of meshes and isosurfaces, emulation of physical dynamics, modeling of global and local deformations of different materials and decimation of point-cloud data. Self-organizing architectures that perform clustering are a promising research direction to build multi-resolution models on one side, and to ensure a distinct treatment for different clusters of different elastic behaviors therefore improving the modeling behavior for non-homogeneous/piecewise-homogeneous objects, on the other side. The ability of feed-forward neural networks to model highly nonlinear mappings and provide estimates for data that were not part of the training set and their fault-tolerant nature make them a good candidate for modeling elastic behavior.

The scheme proposed in this thesis advantageously combines self-organizing and feedforward neural network architectures to achieve diversified tasks as required for data collection and modeling of elastic characteristics, as depicted in Fig. 1.1.

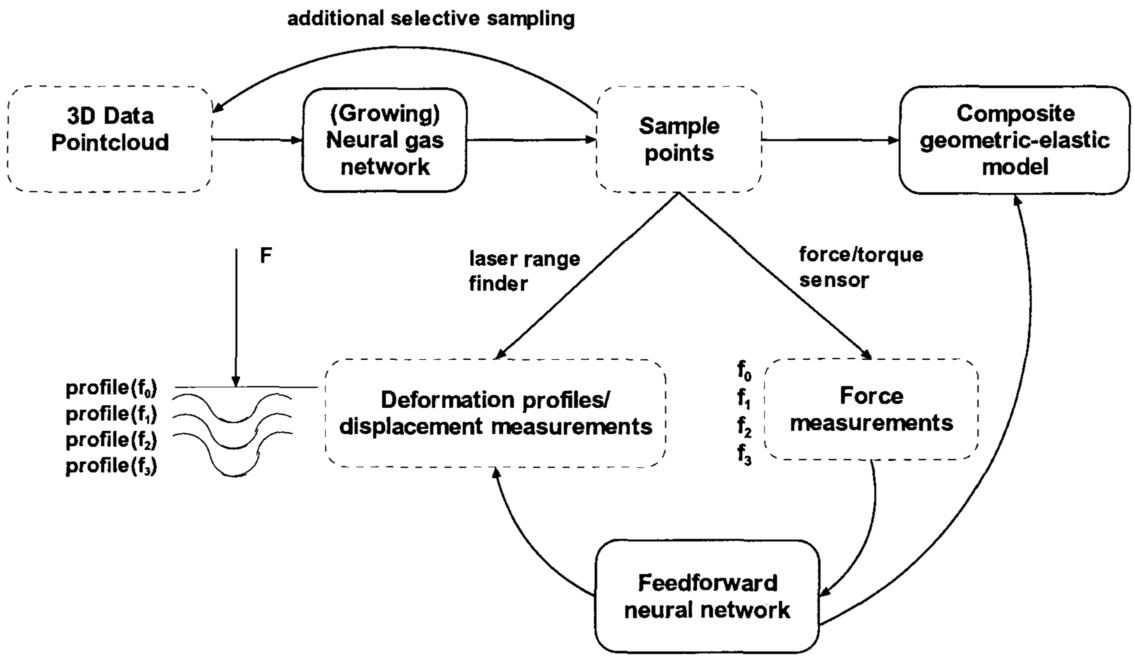


Fig. 1.1. The structure of the proposed neural-based sensing and mapping framework.

During the first phase, an original non-uniform selective sampling algorithm based on a self-organizing neural architecture is implemented to selectively collect data only on those points that are relevant for both the acquisition of the geometric model and the mapping of the elastic behavior of an object. Starting from a 3D point-cloud collected on an object or a scene of objects via an active range finder, a neural gas or a growing neural gas network obtains a compressed model for the dataset in which the weight vector consists of the 3D coordinates of the objects' points. During the learning procedure, the model contracts asymptotically towards the points in the input space, respecting their density and thus taking the shape of the objects encoded in the point-cloud. These modeling properties ensure that the density of the points is higher in the regions with more pronounced variations in the geometric shape and elastic behavior and therefore in regions rich in features. The advantage of such a modeling approach is not only to identify relevant sampling points, but also to allow for the determination of clusters of sampling points with similar geometric and elastic properties due to its ability to find an optimal finite set that quantizes the given input space. This provides a robust mechanism for modeling multi-resolution non-homogeneous/piecewise-homogeneous objects as well. The addition of a measure of elasticity as a fourth dimension in the input space is shown to enhance the modeling scheme in the case of objects where the elastic behavior is not correlated to the changes in the geometry. The growing neural gas is explored in order to avoid the limitation of neural gas that requires the user to decide on a map size, as the latter constrains the accuracy of the modeling results obtained by the network.

The resulting regions of interest identified by the growing neural gas or neural gas mapping are sampled using an original combination of vision and tactile probing. A feedforward neural network is then employed to model the force/displacement behavior of selected sample points that are probed simultaneously by a force/torque sensor and an active range finder. Such an approach allows not only to store inherently the elastic parameters at the sampled points by learning the mapping from force to displacements or deformation profiles respectively, but also to provide an estimate on the elastic behavior on surrounding points that are not part of the selected sampling point set. Moreover, the fault-tolerant nature of neural networks improves the behavior of the proposed modeling scheme in the presence of the noise inherent in any real measurement data. The same architecture with the same parameters can be used as long as the same measurement equipment is used (same number of sample points in the profiles and/or curves), eliminating the need for an expert user to make the necessary adjustments for each material to be modeled.

1.4. Contributions

The proposed, expected contributions of this thesis can be summarized as follows:

- An original approach to combine, synchronize and merge different sensing technologies for the acquisition of force-deformation measurements for 3D deformable objects.

- An innovative automated selective data acquisition algorithm to guide online the probing of only relevant points for the objects under study. The algorithm works directly from raw range data. It is appropriate for use both in vision for 3D geometry data acquisition and in tactile sensing for elastic measurements acquisition. It deals with homogeneous and non-homogeneous objects as well, allowing distinct treatment of regions with different elastic behaviors. Finally, it offers the desired degree of control to the user in that modeling parameters can be chosen automatically or they can be tuned by the user.
- An original feedforward neural network architecture to learn 2.5D profiles of elasticity from the collected experimental data. The approach works on objects made of materials with different elastic behavior, including rigid, elastic and plastic deformable objects, and/or distinguishes the different deformation stages of the same object.
- An original combination of neural architectures to model multi-resolution deformable object models.
- The integration of the previous elements into an innovative data acquisition and modeling framework for deformable objects that operates from noisy experimental measurements without *a priori* knowledge on the shape and the material of the object under study.

The results in this thesis were reported in five journal papers (of which three are published, one accepted and one is submitted), nine conference papers (six published, one accepted, and two submitted) and three presentations.

1.5. Thesis organization

The second chapter provides a brief introduction to elasticity issues and an overview of existing approaches for the measurement and modeling of elastic parameters, for the adaptive acquisition, sampling and decimation of geometric and elastic data, as well as for deformable objects models. The presentation is not meant to be exhaustive, but rather aims at putting in evidence the different trends, the interest and actuality of the topic, the fast recent advance of the research on this topic and the several issues and areas that are less covered or ignored by the available literature.

The third chapter discusses the modeling of elastic behavior from experimental measurements. It starts by showing how elasticity can be modeled from and in the form of force-displacement curves, knowing a mesh of an object, and how this representation compares with a classical method for modeling elasticity, namely the Three-element method. Then a new, more accurate method is proposed to elicit and model elastic behavior. This method allows for the measurement of elastic behavior directly from point-clouds and can generate 2.5D profiles of elasticity for the modeled objects. Several experiments are performed for each of the cases and results are presented to validate the proposed solutions.

The fourth chapter describes an innovative selective data acquisition algorithm that can be employed to collect both geometric data using vision sensors and elastic data using tactile sensors. It starts by describing the motivation driving the research into selective sampling procedures for data collection and justifying the suitability of self-organizing neural networks for this purpose. It then looks in depth at the neural gas architecture, its training algorithm and testing procedure and its use in the context of selective data acquisition. It continues by showing experimental results for vision and tactile sampling of 3D objects embedded in point-clouds obtained with various types of scanners. It finally describes the composite geometric and elastic object modeling framework and exemplifies it with a complete modeling scenario.

The fifth chapter tackles some of the limitations of the neural gas, particularly the one related to the need of the map size to be decided prior to learning, size that constrains the resulting mapping as well as the accuracy of the modeling results. Growing networks add additional nodes into the network structure to support the nodes with highest accumulated errors until a predefined stopping criterion is met, therefore eliminating the need of fixed dimensionality maps. The chapter starts by briefly presenting the general framework of growing networks, it then describes the growing neural gas architecture, its training and testing procedure, the influence of its parameters on the modeling results and its use in the context of selective sampling. It shows experimental results for vision and tactile sampling of 3D objects embedded in point-clouds obtained with various types of scanners and compares them with those obtained using the neural gas. It finally presents a complete modeling scenario for composite geometric and elastic objects.

The sixth chapter presents the conclusions, highlights the contributions and proposes some directions of further study.

2. Literature Review

The interest areas of the thesis cover three major directions of research: elasticity measurement (section 2.1), adaptive/selective sampling of points for geometric and elastic description of objects (section 2.2) and modeling of deformable objects (section 2.3). This survey provides a very brief and simplistic discussion of elasticity, as well as an overview of elastic measurement methods, of adaptive/selective sampling and decimation of elastic and geometric data, and of deformable object models encountered in the literature. It is not meant to be exhaustive but rather aims at revealing some of the existing research trends for capturing elastic behavior and modeling of deformable objects and some of the envisioned applications, highlighting the interest and actuality of the topic, the fast advance of the research on this topic in the latest years and the several issues and problems that are simplified or less covered by the existing literature.

2.1. Elasticity Measurement for Deformable Objects

2.1.1. Definition and Issues in Elasticity

Mechanics, one of the oldest and most important branches of engineering analysis, deals with the behavior of solids and fluids under mechanical disturbances. Mechanics of solids is divided into mechanics of rigid bodies and mechanics of deformable bodies [2]. Elasticity and its theory is a part of mechanics of deformable bodies and deals with the analysis of elastic bodies subjected to loads. A body or object is said to be *elastic* if it recovers its original shape when the forces causing its deformation are removed.

There are two types of loadings which can act on an elastic body – body forces and surface forces. Body forces, such as forces due to inertia and gravity, magnetic and centrifugal forces are distributed over the volume of the body, while surface (traction) forces are distributed over an area of the surface and are produced as a result of physical contact with another body/object.

Under the application of external loading, elastic objects deform. Stress is the distribution of forces inside an object due to an outside deformation force. Tension and compression occur when the force acts perpendicular to the object surface and shear stresses occur when the force acts parallel to the object surface. The normal stress is the stress perpendicular to a surface. Its value can be obtained when a body is axially loaded and the external force is divided by the surface area of application of that force [3]. The total stress state at any point within the object material should be given in terms of both normal and shear stresses. Strain is a measure of the relative deformation (extension) of the object due to the action of applied stress.

The stress and strain are related to each other through certain relations known as constitutive relations that depend on the properties of the elastic body under consideration. A body is said to be *non-homogeneous* or *heterogeneous* if the deformation under the external load varies from point to point. If the material is *homogeneous*, the elastic behavior does not vary spatially and thus elastic parameters are constant. Piecewise homogeneous bodies are bodies in which several regions exist having different but homogeneous properties [4].

Another fundamental material property, similar to homogeneity, is *isotropy*. Isotropy is related to the differences in material parameters with respect to orientation. A body is said to be composed of anisotropic material if the material properties are functions of the orientation, therefore implying the tendency of the material to react differently to stresses applied in different directions [5].

The most general linear relationship which connects stress to strain is known as the generalized Hooke's law and can be expressed mathematically as:

$$\sigma_{mn} = E_{mnpr}\gamma_{pr} \tag{2.1}$$

where the components of the forth-order tensor E_{mnpr} are the elastic constants, γ_{pr} is the strain tensor and σ_{mn} is the stress tensor [6]. As both the stress and the strain tensors are fourth-order tensors, there are 81 components of E_{mnpr} that have to be defined in order to link the strain to the stress tensor. Based on certain symmetries, the number of total independent components can be reduced at maximum to 21 in case of anisotropic materials. In the case of isotropic materials, the constants of E_{mnpr} reduce to two, the so-called Lamé constants (λ, μ):

$$E_{mnpr} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \tag{2.2}$$

and can be expressed in terms of two material properties– known as Young modulus (E) and Poisson ratio (ν):

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad \nu = \frac{\lambda}{2(\lambda + \mu)} \tag{2.3}$$

The Young modulus, also called the elastic constant is the slope of the stress-strain curve, and Poisson ratio is the ratio of lateral contraction to longitudinal extension that occurs in the deformable object due to tensile stresses.

In case of elastic materials, the stress is completely determined by the strain and vice versa. Fig. 2.1 shows a typical strain-stress curve that can be obtained through experimental testing by plotting measured strain-stress data.

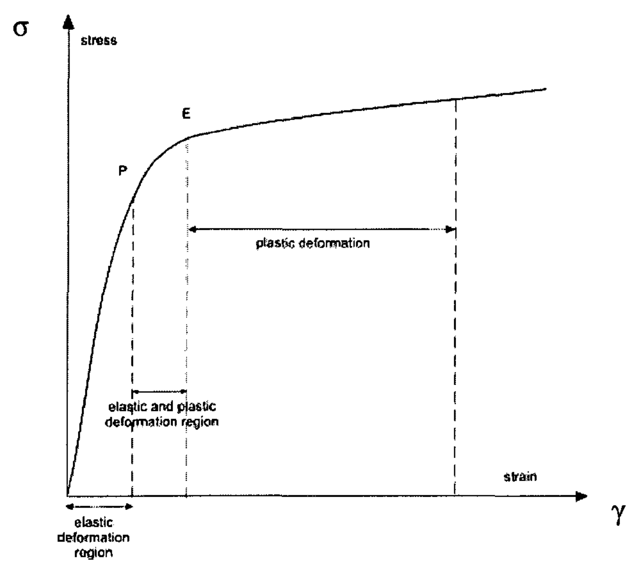


Fig. 2.1. Typical stress-stress curve.

The majority of materials exhibit an approximately linear initial strain-stress response for small deformations. This is followed by a change to a nonlinear behavior that can take various forms – for example the curve can continue to grow (hardening behavior) or it can decrease and finally the material may rupture (hardening and softening behavior) or the slope has a point of inflection after which it starts to rise again (stiffening behavior) [7]. For each material, the initial linear response ends at a point referred to as *proportional limit* (denoted P in Fig. 2.1). If the loading is removed, the material under study will return to its original shape and the strain disappears. The point at which the non-elastic behavior begins is called *elastic limit* (denoted E in Fig. 2.1). After this point the unloading will not bring the strain to 0 and some permanent plastic deformation will occur. Many times the value of elastic limit and proportional limit coincide.

These aspects lead to the discussion about elasto-plasticity and plasticity. All elastic bodies return to their original shape after the external force that determined their deformation is removed. It is said that elastic objects do not have residual deformation. A *plastic* object remains in exactly the same shape as when it was deformed. It is said that plastic objects do not have bouncing deformation. *Elasto-plastic* or *rheological* objects [4, 6] are objects that have residual and bouncing deformations as well. Their deformation is partially lost after the removal of the deformation force. A *viscoelastic* substance [6] has an elastic component and a viscous component and exhibits time dependent strain. There are several schematic models that represent such behaviors, consisting of mechanical elements combined into networks. Several examples are: a simple spring to model elastic behavior, the Kelvin-Voigt model, that consists of a damper and a spring connected in parallel, the Maxwell model consisting of a spring and a damper connected in series or the three-element model that consists of a damper and spring in parallel which are in series with another spring to model viscoelasticity [6]. Such models relate forces to corresponding displacements, provided that the material is assumed to have certain properties such as isotropy or linear elasticity.

As briefly discussed, the problems of elasticity are usually not easy to solve, unless certain assumptions and/or approximations are made on the material of the objects under the study. Serious difficulties are encountered both to conduct strain-stress relationship measurements and to model elasticity for objects made of materials that exhibit nonlinear behavior, where an accurate mathematical material model is not available or where a model is available for the object's material, but its parameters (material properties) are not known or are known with a low certainty.

2.1.2. Methods to Collect and Extract Elasticity Data

There are four main categories of solutions encountered in the literature for gathering elastic behavior of objects: indentation [8-11], vibration-based measurements [12], sound-based measurements [13, 14], and vision-based measurements [15-21]. Apart from these, some hybrid methods are also proposed [9, 22-26].

2.1.2.1. Indentation

The measurement of the elastic parameters of an object can be carried out based on the results of instrumented indentation tests. These tests usually imply the monitoring of the evolution of the force strength using a force-feedback sensor. In case of linear elasticity assumption, the elasticity coefficient can be computed from Hooke's law, where the strain is a function of the measured displacement.

A typical indentation test for elasto-plastic objects is composed of a loading and an unloading part, with the loading part used for the extraction of strength properties and the unloading portion for the elasticity properties of the indented material [8]. The elastic and viscosity coefficients can be determined in the case of linear elasticity assumption by observing the attenuation curve of contact force and the attenuation curve of displacement. However, the interpretation of the indentation load-displacement curve becomes complex and no closed-form solutions are readily available when the elasticity is not assumed to be linear due to large geometric and material nonlinearities as well as to contact problems. A neural-network solution is employed by Tho *et al.* [8] to describe the relationships between the curvatures and the areas under the loading and unloading curves, and the strain exponent on one side and the yield stress on the other side. Two backpropagation multi-layer feedforward neural networks are used and the training data is generated numerically through finite-element analysis for each type of indenter employed.

Charlton *et al.* [9] use backpropagation neural networks for the recuperation of contact parameters of applied force, both normal and tangential, from normal stress distribution. Using only the normal stress distribution and based on the plane strain assumption, a three-layer neural network trained with backpropagation is employed to recuperate the contact parameters of the applied force and the indenter width.

A simplified procedure is proposed by Monserrat *et al.* [10] to determine the elasticity parameters for elastic tissue. The underlying idea is to determine a range of displacements within which the tissue exhibits a quasi-linear elastic behavior. An increasing linear force, with a prescribed velocity is applied to identify a velocity range for which identical loads result in the same displacement (implying that the tissue exhibits linear elastic behavior). The tests are redone for the determined elastic range defining two control points close the load application point. The changing coordinates of these two control points are monitored and the elasticity and Poisson ratio are computed using an inverse method based on a Newton-Raphson technique to minimize the difference between the experimental and simulated displacements.

D'Aulignac *et al.* [11] use a probe mounted on a force sensor and on a robot arm, positioned perpendicular to the surface of a human thigh. Regularly distributed measurements are collected over the surface in order elicit elastic behavior. The parameters are then estimated using an optimization approach based on nonlinear least-squares estimation.

2.1.2.2. Vibration-Based Measurement

A vibration-based measurement method for elasticity is proposed by Okamura *et al.* [12]. Vibrations, forces and velocities are collected during various tasks executed with a stylus, such as tapping on materials (detects stiffness), stroking textures (reveals relative roughness), and puncturing membranes. The analysis of the vibration waveforms gathered results in parametric, empirical models describing the various tasks (e.g. exponentially decaying sinusoid with parameters depending on the material describes tapping waveforms) and a library of model parameters is compiled to be further reproduced.

2.1.2.3. Sound-based Measurement

While the direct assessment of simple object can be done with indentation procedures, there are situations when the displacement measurement is not easy to be performed. Such a situation is, for example, the assessment of tissue stiffness by monitoring internal tissue displacement responses to an externally applied force. Mainly designed for medical applications, sonoelasticity and ultrasound elastography are representatives of dynamic and static compression techniques respectively that avoid the indentation procedure.

Sonoelasticity is a dynamic compression technique in which a vibrator is used to propagate low-frequency waves into the tissue. Shear wave velocity or wavelength is estimated, and from it, the shear modulus can be estimated. However, problems associated with this technique are high image noise, low spatial resolution, and difficulty in propagating the shear wave energy across tissue boundaries [13].

In (quasi)static compression techniques, the elastic modulus distribution of the tissue is estimated from the tissue deformation and boundary pressure measurements. A representative of this category of techniques is ultrasound-based elastography [13, 14]. The setup is composed of a focused ultrasound transducer, adjusted perpendicular on the surface of the sample material, and that sends an ultrasonic beam into the sample. A localized shear wave is induced and another ultrasound transducer in transmitter-receiver mode is used to capture the RF (radio-frequency) echoes. The tissue is deformed and the RF echo waveforms before and after a deformation are recorded. The tissue displacement distribution is estimated by comparing these RF waveforms.

2.1.2.4. Vision-based Measurement

Another category of methods used to collect elastic data in an engineering context are based on vision. The principle employed here is to take a series of images before and after the deformation, analyze them individually and extract profiles in order to compute displacements that elicit elastic properties of the objects. A point-wise correspondence is also needed between the deformed and non-deformed profiles in order to compute the displacement. Several methods can be employed for this purpose. The objects can be directly marked using markers [15-17] or the correspondence can be established using speckle patterns [20, 21]. In the latter case, a thin film coating is applied on the surface of the object to generate the necessary speckle and the change of patterns with the deformation of the object is observed.

Wang *et al.* [15] use a grid of markers that also serve as nodes in a finite-element method (FEM) model. Using digital image processing (corner extraction), the coordinates of various feature points in the grid and their displacement are obtained. From this information a strain field and the corresponding work-conjugate stress field are constructed and the forces are computed by balancing the internal stresses at each node.

In Kamiyama *et al.* [16, 17], a tactile sensor which uses a transparent elastic body with markers and a color CCD camera are used to measure elastic behavior of objects. By taking images of a certain marker in the interior of an elastic body, when a force is applied to the surface of an object (considered linear elastic), the variation information of the interior is measured, and used to reconstruct the force distribution.

A neural-based solution without explicit computation of elasticity is presented by Greminger *et al.* [18], who take images before and after deformation to train a backpropagation neural network to learn the deformation of an elastic object subject to an applied force. The inputs are the coordinates of a point in a non-deformed body and the applied load on the body, and the outputs are the coordinates of the same point in the deformed body. The training data pairs are obtained directly from images of the object under known loads and a computer vision deformable body tracking algorithm based on boundary-element method that builds on the equations of the elasticity is used to measure the displacement of a point in the non-deformed body.

Vuskovic *et al.* [19] use a specially designed instrument to measure hyper-elastic, isotropic materials, such as living tissues. The measurement method, based on pipette tissue aspiration, consists in leaning a tube against the tissue and gradually reducing the pressure in the tube. As the organ remains fixed to the tube, there are well-defined boundary conditions and a complete description of the deformation is given by the profile of the aspirated tissue. The measurements are performed based on vision. An optical fibre illuminates the scene and the deformation is observed in a small mirror placed beside the aspiration hole and is captured using a camera. From the observed pictures, the material parameters are determined via the inverse finite-element method. A Levenberg-Marquart parameter identification algorithm performs a minimization of the difference between the measured load-deformation data and the data obtained using the finite-element method.

Ferrier *et al.* [20] and Hristu *et al.* [21] describe a deformable image-based tactile sensor consisting of a roughly elliptical membrane, filled with fluid-like gel and inscribed with a grid of dots at precisely computed locations of the inner surface of the membrane. A fibre optic cable illuminates the interior surface and images of the grid are taken as the membrane deforms. A solution for the 3D coordinates of the grid is obtained based on the assumption that the volume enclosed by the membrane remains constant, the boundary of membrane is fixed, and that the portion of the membrane which is not in contact will assume a shape that minimizes its elastic energy.

2.1.2.5. Hybrid Methods

Tanaka *et al.* [22, 23] propose and apply the novel paradigm of haptic vision (that refers to augmenting active vision with active touch) to elasticity estimation. The experimental setup is composed of a haptic vision sensor (a range finder and a CCD camera mounted on a robot arm) that provides information on the geometrical properties of the studied object, such as 3D shape, posture or texture and another robot arm equipped with a force-feedback sensor that makes contact to elicit different behaviours of objects. Using this setup, a top view of the object is first obtained from which the 2D mass center and the plane of symmetry are estimated. An external force is then applied downward, perpendicular to the support plane that passes through the object's center of gravity to avoid the movement of the object. The strength of the force is monitored using the force feedback sensor. The strain is measured by subtracting the observed height in an image provided under deformation by the CCD camera and the range finder sensor from the initial height. The elasticity modulus is computed from Hooke's law. The elasticity study is reduced to linear elastic behaviour of indoor objects, situated on horizontal planes, stable under gravity, with plane symmetry.

Pai *et al.* [24-26] use an expensive prototype and sophisticated robotics for interaction and behaviour measurement in the context of 3D deformable objects. Their ACME facility is composed of a 3 degree-of-freedom test bed for positioning of the test object, a field measurement system for measuring the light and sound fields around the test object, composed of a color video camera and a trinocular stereo vision system, a contact manipulation system which includes a robot arm with a wrist mounted force-torque sensor to measure properties such as friction and stiffness and to make controlled impacts to generate sounds, and a time-of-flight range finder. To elicit elastic properties of an object, a contact force is applied with the robot arm while monitoring the deformation with the field measurement system. Displacement information is recorded using the trinocular stereo system and a range-flow approach is employed to measure the difference between the non-deformed and deformed object states. The model built is a triangular mesh and the elasticity parameters, assuming that the material is linear-elastic, homogeneous and isotropic, are solved using an iterative method.

2.1.2.6. Elastic Parameters Reconstruction/Modeling

There are three methods encountered in the literature for recuperating Young's modulus from measurements: solving the differential equation of elasticity numerically based on given constraints, using variational techniques and using iterative techniques. The last category is the most popular one. As well, there are several proposed methods that do not recuperate the elastic parameters directly, but learn them implicitly.

Skovoroda *et al.* [27] solve numerically a second-order partial differential equation that describes a linear, anisotropic, incompressible medium under static deformation. Romano *et al.* [28] use variational techniques to reconstruct the modulus distribution.

Iterative techniques to reconstruct the modulus distribution are based on the minimization of the difference between displacement distribution (measured or computed by other methods) and simulated results. Displacement distribution can be described in terms of finite-difference analysis [29] and finite-element analysis [13, 30]. There are also several iterative approaches proposed in the literature for real measured data [10, 25, 26, 30].

Raghavan and Yagle [29] use finite-difference analysis to describe elasticity. The set of linear equations of elasticity is rearranged such that the modulus distribution becomes the unknown variable. The modulus distribution can then be solved analytically. In Kallel and Bertrand [30], the finite-element analysis algorithm receives as inputs the measured displacement field, the assumed boundary conditions and an initial value for the modulus distribution, and outputs an estimate of the distribution. The difference between the real and estimated value is then used to adjust the next initial value to be introduced in the algorithm. The process is repeated to minimize the error in the least squares sense by means of a Newton-Raphson technique.

Zhu *et al.* [13] rearrange the linear equation set such the modulus of elasticity distribution is the explicit variable and solve it iteratively based on a simplification of the boundary conditions.

Considering given a minimum and maximum value for the Young's modulus in case of vibrational displacement of living soft tissues, Fu *et al.* [31] use an iterative approach based on measured data to compute the modulus of elasticity. The algorithm starts from a value of the Young's modulus equal to the arithmetic mean of its maximum and minimum value and these values are then updated based on the error between the estimated and measured amplitude of the vibration waveform. If the estimated value of the amplitude is higher than the measured one, the maximum value is kept, while the minimum value becomes the current value of the Young's modulus. If the measured value of the amplitude is higher than the estimated one, the minimum value of the modulus is kept, while the maximum value becomes the current value. The procedure is repeated until a fixed number of iterations is reached or the error in the modulus estimation drops below a predetermined level.

Monserrat *et al.* [10] apply an iterative inverse method to recuperate the Young's modulus and Poisson's ratio for a set of real data. Two control nodes are defined on the tissue surface close to the load application point and the corresponding changing nodal coordinates are monitored, from which the respective displacements are calculated as a difference between the initial position of the control node and the changed position. With an arbitrary selection of Young's modulus and Poisson ratio, the real load conditions are simulated numerically and the displacements are obtained. The parameters are then updated iteratively, by minimizing the difference between experimental and numerical displacements using a Newton–Raphson technique until the updates of the parameters become sufficiently small. The elastic modulus along with the yield strength and the strain exponent are recuperated from the indentation load-displacement curve in case of elasto-plastic objects and the applied forces can be recovered from spatially discrete knowledge of the stress–field over the surface inside the sensor [10, 32].

Strain and stress vectors, are used as input and output vectors respectively in a neural network with 2 hidden layers by Shin *et al.* [33] to identify anisotropic material parameters from a single structural test. The network is trained on simulated data using finite-element method (FEM) to learn the mapping from strain to stress, corresponding to the elasticity matrix E_{mnpr}. The elastic constants are computed by comparing the elasticity matrix with the results provided by the neural network.

A different research direction does not attempt to recover the elastic constants since elastic constants are material-oriented parameters and do not describe many real-world objects [8, 9, 24-26]. Instead quantities that can be measured on the surface of an object without destroying the object are considered. The object is loaded by a force in a point and the displacement of the surface is observed. The resulting elastic parameters can then be recuperated, based on certain assumptions, from the relationship between the force and displacement [8, 10, 18, 24-26].

2.1.3. Elasticity Measurement Conclusions

As this study shows, there are on the whole, relatively few papers devoted to elasticity measurement. Less than 8% of all the reviewed publications on aspects of elastic and deformable objects imply elasticity measurements. From the four main categories of solutions encountered in the literature for gathering elastic behavior of objects, the indentation and vision-based measurements are the most commonly employed in engineering applications, while the sound-based measurements are almost in their entirety dedicated to medical applications.

Most of the models proposed are aimed at very specific applications (e.g. modeling of tissue based on pipette suction method) and cannot be transferable to another context. Further research in this direction would benefit from developing solutions that are not specific for one application, but rather can be relatively easily generalized or adapted to other contexts.

The majority of the proposed methods make the assumption that the material exhibits linear elastic behavior, that the modeled objects are homogeneous from the elasticity point of view, and/or make use of another method (e.g. finite-element method, boundary-element method) to recuperate the elasticity information. Critical advantage is expected in this direction by the implementation of modeling schemes able to cope with nonlinear elasticity and heterogeneous or piecewise homogeneous objects. In addition, promising results are expected from modeling/learning implicitly the elastic behavior, in order to avoid the recuperation of parameters using heavy and complex mathematical methods and/or the assumptions made on the material of the object under study.

2.2. Adaptive Data Acquisition and Compression Techniques for 3D Objects

Due to the high measurement speed of 3D data acquisition devices (e.g. laser scanners) on one side and to the lack of knowledge on appropriate accuracy levels for a correct description of shape and geometry on the other side, the data acquisition process is elaborate and often produces too many sample points. Reducing the complexity of such data sets is one of the key techniques required in order to operate subsequent applications on the resulting data at a reasonable computational cost. To tackle this issue, the most widely exploited trend in the literature implies the post-processing of large data sets obtained by acquisition devices. Frequently, the proposed algorithms rely on the user input for providing parameters such as the density of sampling, regularity of sampling, or the minimum distance between samples. This is a difficult task as the user is not always aware of the appropriate level of accuracy required for a model in order to be further processed and the adjustment of such parameters can be a lengthy trial-and-error process.

In general, there are three sampling policies proposed in the literature: uniform sampling, random sampling and stratified sampling. In uniform (regular or grid) sampling, samples are spread such that the probability of a surface point being sampled is equal for all surface points. The method is popular because it can be easily implemented and ensures complete coverage of the surface within the sensor's field of view. However the cost is high, because in order to achieve adequate sampling density at those regions requiring highest density, the sampling density must be uniformly high everywhere. In random sampling, each point of the object has an equal chance of being selected, but only a lower number of points are collected. As the percentage of sampled points increases, the cost gets higher to eventually reach the one of uniform sampling. As well, the risk is that samples randomly collected miss important features of the object under study.

Stratified sampling is a technique that generates evenly spaced samples by subdividing the sampling domain into non-overlapping partitions (clusters) and by sampling independently from each partition. The method ensures that an adequate sampling is applied to all partitions. This idea is often exploited in the context of post-processing of large point-clouds or meshes [34-40], where a subdivision of models occurs into grid cells and sample points that fall into the same cell are replaced by a common representative.

The 3D model proposed by Nehab *et al.* [34] is first voxelized with an octree and one sample is outputted for each voxel. The common representative for a voxel is selected according to a probability that decays as the distance of the sample to the center of the voxel increases. The user controls the sampling resolution, the regularity of the sampling, and the minimum distance between samples. In [35] the representative point is the measured point that is the closest to the average of points that fall into the same voxel.

The 3D grids proposed by Lee *et al.* [36] are octrees that voxelize 3D points constructed by registration and integration of multiple scanned data sets of an object or scene of objects. The method uses point normal values on the surface of the object (computed based on the knowledge that scanned lines and points are ordered due to the raster scanning procedure employed and based on a triangulation performed on two neighboring scan lines) from which non-uniform grids are generated using the standard deviation of normal values. The representative point for each grid is the point whose normal is closest to the average of the points in the same voxel.

The standard volumetric subdivision strategy cannot be adapted to non-uniformities in the sampling distribution and sometimes joins unconnected parts of a surface if the grid cells are too large. To alleviate these problems, Pauly *et al.* [37] perform surface-based clustering, where clusters are built by collecting neighboring samples while taking into account local sampling density. Points are incrementally added to a cluster until a maximum size and/or a maximum allowed variation is reached. The sample points of those clusters that do not reach the minimum size or variation bound are distributed to the neighboring clusters. Alternatively, clusters can be computed by recursively splitting the point-cloud using binary space partitions.

Uesu *et al.* [38] simplify large unstructured meshes by segmenting them into two parts: the boundary of the original domain and the interior samples. Each part is then simplified separately, considering proper error bounds. For the boundaries, a modified surface-simplification algorithm that takes into account the scalar field defined at the vertices is employed, while the interior points are sampled using a *k-d* tree partition of the mesh from which the samples that are outside the boundary, or closer than a certain minimum distance to the boundary are removed. Finally, the simplified domain boundary and scalar field are combined into a complete, simplified mesh using a Delaunay tetrahedralization.

A similar idea is employed by Song and Feng [39] whose point-cloud simplification algorithm starts by identifying and retaining edge boundaries (using a measure of the deviation of the normal vectors in the Voronoi neighborhood) and then removing least important points from the remaining data based on the contribution to the representation of local surface geometry. If the local geometry cannot be reliably reflected by the neighboring points of a point and their associated properties, that point is considered important for defining geometry, otherwise it can be removed. The removal procedure ends once the specified data reduction ratio is reached.

Kalaiah and Varshney [40] propose a scheme to compactly decimate and represent point-clouds using Principal Component Analysis (PCA). The input is pre-processed using an octree and PCA analysis is performed for each cell. Due to the fact that PCA parameters (orientation, frame, mean and variance) tend to be similar for coherent regions, the node parameters can be classified using clustering and quantization. At run-time, based on the viewpoint, a cut in the octree is determined and each node of the cut is visualized using a Gaussian random generator. Attributes like normals and color are also generated in the same manner.

Based on the fact that sampling algorithms are usually implying a measure of error, the idea of using error propagation neural networks that minimize an error measure seems a good choice to Fiore *et al.* [41], who propose a multilayer-feedforward network for non-uniform image sampling for robot motion control. The neural algorithm allows a controller to determine the robot's location within a structured environment based on a digital image sequence coming from a camera. The proposed sampling starts with a number of points (pixels in the image) uniformly distributed over the whole image that all become input points in a neural network. Network pruning is then performed on the basis of inputs relevance and therefore the number of inputs is reduced while the mean-squared error is kept below a pre-defined threshold. After pruning, the remaining sampling points are moved toward the high relevance areas based on a radial-basis-function sampling operator.

All these described methods and solutions are not meant to be incorporated in the actual sampling procedure, but rather post-process collected data. An approach to integrate the sampling procedure into the measurement procedure is proposed by Pai *et al.* [24, 25]. They use a probing procedure that considers known a mesh of the object under study, as well as a set of parameters such as the maximum force exerted on the object, the maximum probing depth and the number of steps for the deformation measurement. During probing, the designed algorithm generates the next position and orientation for the probe based on the specified set of parameters and the mesh of the object under test. It performs at the same time proximity checks and verifies the expected contact location of the probe with the mesh based on line intersection.

More recently, an adaptive scanning procedure was introduced by MacKinnon *et al.* [42] that uses a group of intermediate quality metrics to monitor several aspects of the scanning procedure, including outliers, aliasing, planarity, resolvability and reflectivity of the surface. This heuristic approach helps at focusing a laser scanner over an optimal field of view and guides the collection of 3D points over regions that will most likely maximize the visual quality of a final 3D model of an object while minimizing the number of measurements required. This approach is only evaluated in the context of high-resolution modeling of artifact objects, where the necessary information to compute the quality metrics is typically available and reliable.

Both of the two above mentioned methods are adaptive, but not selective. Further research on adaptive and selective sampling would benefit from automated selective procedures to determine regions of interest and perform only relevant measurements for both vision and elastic modeling applications.

2.3. Deformable Object Modeling and Applications

Several methods have been proposed to respond to the increasing complexity and realism of interaction requirements of deformable models, while trying to maintain reasonable processing time. All these aspects depend on the type of models used to describe them. Non-physical models are simpler representations and therefore easier to update in real-time, but provide less realistic simulations. Due the exclusive interest of this thesis into accurate deformable object models, non-physical object modeling will not be further discussed here.

Physically-based modeling techniques augment geometric objects with forces, torques, velocities, accelerations, kinetic and potential energies, heat, and other physical attributes such as mass, damping and stiffness distributions that are used to control the creation and evolution of models. They are governed by simulated physics laws and facilitate the creation of models capable of automatically synthesizing complex shapes and realistic motions to a higher degree of realism. Physically-based models are responsive to one another and to the simulated physical worlds that they inhabit [43]. The main categories of physically-based models are mass-spring models, finite-element and finite-differences methods, long and boundary-element methods, modal analysis, particle systems, implicit models and mesh-free methods. These are described further in the following sections.

2.3.1. Mass-Spring Models

A deformable object can be modeled as a collection of particles connected by springs in a lattice structure, creating a so-called mass-spring model (MSM) [44]. The most straightforward strategy is to consider each vertex of an object mesh as a particle and each edge as a spring. The spring rest length is set equal to the length of corresponding edge. The mass is assigned arbitrarily or it can be evenly distributed among the object's particles. If there is an uneven distribution of vertices in the object, then masses can be assigned to vertices in such a manner as to make a more even distribution of mass. Spring constants are also usually assigned uniformly in the object to a user-specified value. When external forces are applied to specific particles, either because of collisions, gravity, wind or other user specified forces, the particles are displaced and this displacement will induce spring forces in the object, which in turn will induce other spring forces. Therefore, the object will jiggle as a result of forces propagating inside it and it will finally come to rest. Spring forces are often linear, but nonlinear springs can be used to model inelastic behavior. The motion of a single particle in the object is governed by Newton's law of motion and the equations of motion for the whole mass-spring system are assembled for all the particles in the lattice:

$$M\ddot{x} + D\dot{x} + Kx = f \tag{2.4}$$

where x is a $3N$-dimensional position vector, M, D, K are the mass, damping, and stiffness matrices of size $3N\times3N$ and f represents the total external forces on the particles. M and D are diagonal matrices and K is banded as it encodes spring forces, which are functions of distances between points [45]. The solution of the system is obtained by expressing the equations of motion as a set of first-order differential equations and using one of the many available numerical integration techniques to solve it. Among the most popular ones are Euler's method, second order and fourth order Runge-Kutta and Verlet integration.

2.3.1.1. Advantages and Disadvantages of Mass-Spring Models

Mass-spring models are simple, as they are based on well-understood dynamics. They are easy to construct, and both interactive and real-time simulations of mass-spring systems are possible as they have low computational demands. Another advantage is their ability to handle both large displacements and large deformations. They are also well suited to parallel computation due to the local interaction between the nodes [45].

However, the structure of the mass spring is often application-dependent and hard to interpret. The animation results often vary dramatically with different 2D and 3D spring structures [46]. Since the model is tuned through the spring constants, good values for these constants are not always easy to derive from measured material properties. Furthermore, it is difficult to express certain constraints (like incompressibility and anisotropy) in a natural way. When all the springs in a system are set to the same stiffness, the mesh geometry may generate undesired anisotropy. Even though this behavior tends to disappear with an increase in mesh density, an extremely dense mesh would reduce the efficiency [47]. Another problem occurs when spring constants are large. Such large constants are used to model nearly rigid objects, or model non-penetration between deformable objects. This problem is referred to as stiffness. Stiff systems are problematic because of their poor stability, which requires small time steps for numerical integration resulting in slow simulation. And yet another problem encountered is that while most of the materials found in nature maintain a constant or quasi-constant volume during deformations, mass-spring models do not have this property [47].

2.3.1.2. Parameter Identification

Among the intrinsic limitations of mass-spring systems, one of the major problems is parameter setting. In the vast majority of applications, the selection of the physical constants is performed on the basis of the visual appeal of the resulting deformable object [48]. The user controls the spring constants, damping coefficients, and the mass properties of each element individually or of the model as a whole. This choice is justified by the fact that, for computer animation applications, accuracy and correct physical behavior is not a must. Some publications, like Brown *et al.* [49] initialize the parameter values using expert knowledge (biomechanical data models are tuned based on comments given by surgeons interacting with them).

More accurate methods for elastic constants computation rely on the use of genetic algorithms [50, 51], simulated annealing [52], fuzzy systems [53-57], neural networks mimicking the deformation of springs (the network takes the changes of length and velocity of springs as input and outputs the total spring force) [58] or direct measurement of elastic properties of objects that are discussed in the section 2.1.

Another problem related to mass-spring models is how to distribute the object mass over the particles of the system. In most cases, mass is assigned arbitrarily or is evenly distributed among the object's particles. In [50], the distribution of mass respects the inertial properties of the object. The object is decomposed in a set of tetrahedrons, and each tetrahedron is then replaced by a set of 5 particles (one at each vertex and one at the inertial center). The mass of the particles in the vertices is 1/20 of the total mass, while the mass in the inertial center represents 16/20 of the total mass of the object. In [52, 59] each mass is computed according to the volume of the Voronoi region around it. The distribution of the mass points over the deformable object surface varies in local density in proportion to the surface curvature in Cover *et al.* [60].

Stiff systems are problematic because of their poor stability, which requires small time steps for numerical integration resulting in slow simulation. Stability problems are avoided in Wagner *et al.* [61] by setting maximum values for acceleration and velocity of each node. If the maximum acceleration is exceeded by a certain percentage, the spring exerting the highest force on the node is deleted.

2.3.1.3. Anisotropy and Volume Preservation

When all the springs in a system are set to the same stiffness, the mesh geometry may generate undesired anisotropy. To tackle the anisotropic behavior problem, Bourguignon and Cani [47] let the user define mechanical characteristics of the material along a given number of axes corresponding to orientations of interest at each current location. All internal forces will be acting along these axes instead of acting along the mesh edges. During deformations of the material, the three axes of interest of given initial orientation, evolve with the volume element to which they belong. From the deformation of the local frame, the resulting forces on each intersection point are estimated and for a given face, the force value on each point mass belonging to this face is computed by inverse interpolation of the force value at the intersection point. The proposed method also ensures volume preservation using additional springs running from the center of the voxel to the corner masses.

To ensure volume preservation, Vassilev and Spanlang [62] use support springs. Some adjustable constraints between masses are added in Di Giacomo and Magnenat-Thalmann [63], which try to preserve initial volume of the objects by modifying positions relatively to the difference between initial and current distances.

2.3.1.4. Adaptive Refinements and Hierarchical Representations

Exploiting the fact that many deformations are local, hierarchical techniques approximate the deformation from sparse to fine scale to improve the response time of MSM and support dynamic changes of topology [45, 53, 55].

By propagating forces in an ordered fashion through an elastic mass-spring mesh, Brown *et al.* [49] limit computations to those portions of the objects that undergo significant deformations.

Hutchinson *et al.* [64] present a mechanism for adaptive refinement of a 2D mass-spring model, that starts from a coarse approximation and a measure indicating the accuracy required for a particular modeling case. If an inaccuracy is detected in the model, more masses are introduced around the crease point and the stiffness of the springs adjusted to prevent regions of increased masses to behave differently.

Montgomery *et al.* [65] track the region of deformation by breadth-first ordering the nodes from a point of interaction into levels and only process within this region in order to increase performance, while Paloc *et al.* [66] propose a dynamic multi-resolution volumetric mass-spring representation. A coarse mesh is created offline by forming an initial boundary constrained Delaunay tetrahedralization. During online simulation, the given mesh can be locally refined or simplified by inserting, respectively deleting, vertices.

Other improvements of MSM refer to building interfaces between MSM and rigid objects [67, 68]. The representation for a rigid body is the same as the one for a deformable body, except that the distances between its elements are fixed. As all physical interaction occurs between the mass elements only, there is no need for the explicit handling of interaction between rigid and deformable bodies or between rigid and rigid bodies.

2.3.1.5. Applications of Mass-Spring Models

Mass spring models are applied to model soft tissue and internal organs for virtual surgical training [51, 55-60, 61, 63, 65, 66, 69-71], for modeling and animation of human faces [72, 73], characters [64, 74, 75] and cloth [76-78], for building virtual sculpting tools [79, 80] and for interactive shape modeling [68, 81, 82]

2.3.2. Finite-Element Method

While the mass-spring method approximates an object as a finite mesh of points and discretizes the equilibrium equations at the mesh points, the finite-element method

(FEM) divides the object (solid body with masses and energy distributed throughout) into a set of elements and approximates the continuous equilibrium equation over each element. The desired function for each element is represented as a finite sum of element-specific interpolation functions. In the displacement-based FEM method, the equilibrium equation is derived from the goal of minimizing the potential energy with respect to the material displacement [45]. The steps involved in FEM [45, 83, 84] are:

1. The continuum of an object is discretized in a number of finite elements. These elements are assumed to be interconnected at a discrete number of nodal points situated on their boundaries.
2. Equations to approximate the solution for each element are developed. A set of interpolation functions is chosen to define uniquely the state of displacement (an equilibrium equation) for each element in terms of its nodal displacement. In other words, for each element, the components of the equilibrium equation are re-expressed in terms of the interpolation function and the element's node displacements. The coefficients of these functions are evaluated such that they approximate the solution in an optimal fashion.
3. After the individual element equations are derived, they are assembled to characterize the unified behavior of the system. Adjustments are made such that the system respects the boundary conditions.

The system representing the set of equilibrium equations is solved for the node displacements over the whole object. The node displacements and the interpolation functions of a particular element are then used to calculate displacements or other quantities of interest (such as internal stress or strain) for points within the element.

2.3.2.1. Elements and Element Equations

The best choice of elements and interpolation functions depends on the object shape, convergence requirements, degrees of freedom, and trade-offs between accuracy and computational resources [45]. As the complexity of an element increases, the total number of unknowns can be reduced in order to obtain the same degree of accuracy. However, the total computation and data preparation time is not reducing, since the time for element formulation increases for complex elements. Higher order interpolation functions require more sophisticated numerical integration techniques. As the element equations increase in degree, the convergence to the exact solution is more rapid [83]. Each element has an interpolation equation that describes how the quantities vary within the element. Because they are easy to manipulate mathematically, polynomials are often employed for this purpose [84]. For example, for a 2D triangular element its simplest alternative is the linear polynomial.

2.3.2.2. Equilibrium Equations

The object deformation is a function of forces acting on it and of the object's material properties. For dynamic deformation systems, the effect of inertia and velocity are considered, resulting in a second order differential equation of the form, similar to eq. (2.4) used for mass-spring models:

$$M\ddot{U} + D\dot{U} + KU = F \tag{2.5}$$

where *M, D, K* are the mass, damping and stiffness matrices, and *U* is the composite vector of displacements and *F* the composite vector of forces that act on the object. Because the force vectors, mass and stiffness matrices are computed by integration over the object, they should be re-evaluated as the object deforms. As this evaluation is very costly, the assumption that the body undergoes only small deformation is often made and offline pre-computations are used to minimize the intensive computational requirements.

2.3.2.3. Advantages and Disadvantages of Finite-Element Method

FEM deformable models are more accurate than mass-spring models, but the high computation time implied remains an obstacle for real-time applicability of the method.

2.3.2.4. Offline Pre-Computations of Finite-Element Method

The idea behind many developed improvements of FEM basic model is to maximize offline pre-computations so that minimal real-time effort is necessary. Elementary deformations are pre-computed by an iterative method to solve each linear system displacement in Cotin *et al.* [85]. In [86] two linear elastic models are combined: a quasi-static pre-computed model, that is efficient but does not allow topology changes and a dynamic tensor-mass elastic model, which requires more computation but authorizes topology changes in order to optimize the trade-off between computation time and visual realism of the simulation.

In Popescu and Compton [87], based on the assumption that the area of interaction of the external forces on the object is small and given a tetrahedral mesh representing an elastic object and its elasticity constants, the positive definite stiffness matrix and its inverse computation is done offline.

As a pre-computation step, a linear stiffness matrix is computed for the system in [88]. At each time step of the simulation, a tensor field that describes the local rotations of all the vertices in the mesh is computed. This field allows the computation of the elastic forces in a non-rotated reference frame while using the pre-computed stiffness matrix.

2.3.2.5. Adaptive Refinements

Adaptive refinements can be employed to provide sufficient detail only where required, while minimizing unnecessary computation. Astley and Hayward [89] restrict the computation to the regions surrounding the point of interaction between the virtual tool and the deformable body employing a multi-layer finite-element mesh. The body is decoupled into a hierarchical mesh structure consisting of high and low density regions. The top layer (parent) consists of a coarse mesh on the entire body. Child-meshes represent sub-regions of the coarse mesh, but have a much finer resolution. Using Norton equivalents to relate two meshes, it is possible to decouple the coarse and fine regions. Different regions of the mesh can be computed at different rates, and also parent and child meshes can be updated at different frequencies. The same idea of partitioning the underlying volume into one or more relatively dense child meshes and a sparse parent mesh and linking them by Norton equivalents is used in Audette *et al.* [90].

Capell *et al.* [91] combine FEM with a multi-resolution subdivision framework in order to produce animations. Deformations are represented using a hierarchical basis constructed by volumetric subdivision.

Debunne *et al.* [92] use an automatic space and time adaptive level of detail technique, in combination with a large displacement (Green) strain tensor formulation. The body is partitioned in a non-nested multi-resolution hierarchy of tetrahedral meshes.

The local resolution is determined by a quality condition that indicates where and when the resolution is too coarse. As the object moves and deforms, the sampling is refined to concentrate the computational load into the regions where more deformations occur.

A mixed finite-volume/finite-element formulation is employed to optimize the computational effort in [93]. Each sample point is assigned a specific region (its Voronoi region) such as these regions cover the whole object, and the object is described as a non-nested hierarchy of meshes. Different levels of the mesh hierarchy communicate through a ghost node technique. The nodes of a mesh represent a given region of space and their values are the average of the local material's properties within this region. When sampling becomes too coarse in a region, a more detailed level of detail is switched to, by adding sample points in that region to enhance the local description of the material. A coarse node is replaced by its child mesh, where the children are the nodes from the finer mesh which lie inside the Voronoi region of their parent.

Kang and Kak [94] perform interactive deformations of virtual 3D objects using a hierarchical implementation of FEM. During the first phase of interaction, the forces are used for the calculation of volumetric deformations using a coarse-resolution cubic implementation of a volumetric FEM. In a second phase, the surfaces of the object are re-meshed at higher resolution. Forces applied subsequently yield displacements at the old nodes and, via plate-theory based FEM, at newly created surface nodes.

2.3.2.6. Other Improvements of Finite-Element Method

Also in order to achieve real-time response, condensation can be applied to the stiffness matrix and the related computation of the stiffness matrix is only performed for those few positions of the force vector that are non-zero in [95].

The equations of motion of a deformable model are expressed in 3D vector form by a set of second-order ordinary differential equations, which is solved using modal analysis in Chen *et al.* [96]. By mathematical transformation, the base of the matrices is changed such that the equations of motion can be uncoupled.

Hirota *et al.* [97] focus on the simulation of mechanical contact between nonlinearly elastic objects, in form of tetrahedral meshes. Their penalty method for FEM simulation is based on the concept of material depth (the distance between a particle inside an object and the object's boundary). By linearly interpolating pre-computed material depths at node points, contact forces can be analytically integrated over contact surfaces without raising the computational cost.

The inverse problem of elastic reconstruction is stated in terms of an objective function to minimize the square error between measured and computed displacements estimated by a FEM model at the observation points in Han *et al.* [98]. As the displacement vector is a nonlinear implicit function, the defined minimization problem is a linearly constrained nonlinear optimization problem, which is solved using the Levenberg-Marquart method to determine Young's modulus.

Metaxas and Kakadiaris [99] present a technique for the automatic adaptation of a deformable model's elastic parameters within a Kalman filter for shape estimation applications. The modification of each of the FEM model's elastic parameters is based on ideas from the theory of proportional-derivative control. The variation of the elastic parameters depends on the distance of the model from the data and the rate of change of this distance. The algorithm uses physics-based modeling techniques to iteratively adjust both the geometric and the elastic degrees of freedom of the model in response to forces that are computed from the discrepancy between the model and the data.

2.3.2.7. Applications of Finite Element Method

Finite-element method is applied to surgical simulations [85, 86, 90, 92, 95-99] and character animation [91].

2.3.3. Finite-Differences Method

If a deformable object is sampled using a regular spatial grid, the equation of motion can be discretized using finite differences. Terzopoulos *et al.* [100, 101] construct differential equations that model the behavior of non-rigid curves, surfaces, and solids as a function of time, based on simplifications of elasticity theory. The shape of a body is determined by the Euclidean distances between nearby points. As the body deforms, these distances change. Potential energies of deformation are defined for elastic curves, surfaces, and solids. Energies restore deformed bodies to their natural shapes, while being neutral with respect to rigid body motion. The deformation energy is zero for rigid motions and grows as the model gets increasingly deformed away from its natural state. For simulating the dynamics of the model, the motion equations are discretized by applying finite-difference approximations.

Terzopoulos and Fleischer [102] extend the proposed framework to non-elastically deformable models with behaviors such as viscoelasticity, plasticity, and fracture. Their formulation introduces additional global deformation degrees of freedom inherited from superquadrics. Through dynamic simulation, the parameters of the deformable superquadric evolve until reaching a minimum energy state, fitting the deformable model to a given 2D or 3D dataset.

Advantages and Disadvantages of Finite-Differences Method

The method of finite differences is easier to implement than the FEM. However, it is difficult to approximate the boundary of an arbitrary object with a regular mesh. Also, local adaptations are only possible with irregular meshes.

2.3.4. Long-Elements Method

In the Long-Elements Method (LEM) proposed by Costa and Balaniuk [103], volumes of objects are discretized in long elements (LEs), which can be compared with a spring fixed in one extremity and having the other extremity attached to a point in the movable object surface. An LE has no mass and does not occupy a real space. The real space inside the solid is occupied by an incompressible fluid. The LE is affected by internal and external pressures on its surface. External pressure is composed of the atmospheric pressure, of the stress when an elongation exists, and of the forces due to the change in the surface area. The internal pressure is given by the pressure of the fluid, the effect of gravity and the contacts between the object and its environment. A set of equations is built on the static condition that the external pressure equals the internal one and is solved using standard numerical methods.

Applications of Long-Elements Method

LEM is almost in its entirety dedicated to medical applications. Costa and Balaniuk [103] use LEM to model soft tissues. Nakao *et al.* [104] achieve haptic reproduction and real-time visualization of a beating heart for cardiac surgery simulation.

Mendoza *et al.* [105] and Sundaraj *et al.* [106] use a thigh echographic simulator to test the corresponding LEM deformable model.

2.3.5. Boundary-Element Method

The boundary-element method (BEM) has been developed originally as an approximate method of solving differential equations defined for a certain domain when the exact solution cannot be obtained analytically. In 3D elasticity modeling, the domain is equivalent to an elastic body and the boundary corresponds to the body's surface, which is discretized into a finite number of elements. The unknowns are the boundary displacements and forces (tractions). BEM leads to a small but dense set of well-conditioned equations which are simple to solve. Numerous fast integral transform methods exist for Navier's equation satisfied by the displacements of linear elastostatic objects with isotropic and homogeneous material properties. Interactive speeds and accuracy are attained by exploiting the fact that a linear model allows many system responses (Green's functions) to be pre-computed, and then combined later in real-time using an incremental low-rank update solution reconstruction approach [107-109].

2.3.5.1. Advantages and Disadvantages of Boundary-Element Method

Contact forces are more accurate for computation since they are solved just like displacements, instead of being derived from displacements using difference formulas. As a result, BEM easily handles mixed boundary conditions and can use the same boundary discretization as used for rendering, meaning that no separate meshing is required. However, the fact that BEM uses the same boundary discretization as used for rendering, also means that BEM can handle only a class of materials with homogeneous material properties.

2.3.5.2. Applications of Boundary Element Method

James and Pai [108] use BEM for simulation of a kinematically constrained rigid object contacting a deformable object. Monserrat *et al.* [10] propose a deformable object model based on a boundary-element method for soft tissue simulation. Meier *et al.* [110] study cutting on BEM deformable models.

2.3.6. Modal Analysis

Modal analysis is the field of measuring and analyzing the dynamic response of structures and fluids when excited by an input. Its goal in structural mechanics is to determine the natural mode shapes and frequencies of an object or structure during free vibration. It is common to use the finite-element method to perform this analysis.

Pentland and Williams [111] use the finite-element method of modal analysis to define a set of polynomial deformations that has the advantage of allowing efficient approximations of physical behavior. Polynomial deformation mappings are used to couple a vibration-mode representation of object dynamics together with volumetric models of object geometry. The polynomials are derived from the eigenvectors of the finite-element equations. To obtain a simulation of the dynamics of an object one uses a linear superposition of these modes to determine how the object responds to given force. In order to improve efficiency only a limited set of modes are used and small amplitude, higher frequency modes are usually discarded. The level of detail in a simulation can be controlled by gradually introducing higher frequency modes where accuracy is important.

2.3.6.1. Advantages and Disadvantages of Modal Analysis

Modal analysis restricts the deformable object to fewer degrees of freedom, sacrificing generality for speed.

2.3.6.2. Applications of Modal Analysis

Pentland and Sclaroff [112] propose a physically based object recognition method using an approach based on FEM and modal analysis. James and Pai [113] simulate interactive, volumetric and dynamic deformation models using a Dynamic Response Texture simulation process that employs pre-computed modal vibration models excited by rigid body motions.

Hauser *et al.* [114] implement a deformable dynamics simulator for free bodies by combining modal simulation with a standard rigid body dynamics simulator. The two systems interact with each other through inertial effects. As the rigid body moves, the modal system experiences centrifugal and coriolis forces. Inertial moments of the rigid body change as the modal system deforms.

2.3.7. Particle Systems

A particle system is a collection of simple primitive objects (particles) that evolve over time and that are processed as a group in order to represent an object. The characteristics of these objects, such as shape, size, position, color, and the lifetime of the particle, can be changed dynamically using a set of parameters. In general, each parameter specifies a range in which a particle's value must lie. If these parameters of the particles are coordinated, the collection of particles can represent an object. The dynamic behavior of the particles can be determined by the solutions of sets of coupled differential equations or by simulating the behavior imposing forces and constraints.

2.3.7.1. Advantages and Disadvantages of Particle Systems

A particle is a much simpler primitive than a polygon, the simplest of the surface representations. Therefore, in the same amount of computation time one can process more of the basic primitives and produce a more complex image. The model definition is procedural and stochastic and therefore, obtaining a highly detailed model does not necessarily require a lengthy design time. Because it is procedural, a particle system can adjust its level of detail to suit a specific set of viewing parameters. Particle systems can be used to model dynamic objects that are more difficult to represent with other modeling techniques (e.g. surfaces).

However, particles of changing sizes can lead to performance penalties. Another disadvantage is the fact that particle systems are hard to integrate seamlessly into a complex scene and it is harder to achieve exact analytic control over the surface of the modeled objects.

2.3.7.2. Applications of Particle Systems

Particle systems are encountered in three areas of applications: elastic surface modeling [115], volumetric object modeling [116] and cloth modeling and animation [117-120]. Szeliski and Tonnesen [115] model elastic surfaces based on interacting particle systems. The particles have long-range attraction forces and short-range repulsion forces and follow Newtonian dynamics. To enable particles to model surface elements instead of point masses or volume elements, an orientation is added to each particle's state. In addition to having a position, an oriented particle also has its own local coordinate frame, which adds three new degrees of freedom to each particle's state. To force oriented particles to group themselves into surface-like arrangements, a collection of potential functions is devised. These potential functions are derived from the deformation energies of local triangular patches using finite-element analysis and favor locally planar or spherical arrangements.

An interactive, physically-based, elastically deformable model using a particle system to model surfaces with interior volumes that can be haptically felt is presented by Stahl *et al.* [116]. Oriented particles used in existing surface-only models, and non-oriented particles used in volume-only simulations are combined to form a bag-of-particles. Multiple species of surface and volume particles, coupled with predefined interspecies parameters, determine the elastic properties of a bag. Starting with an object represented as a 3D voxel bitmap of connected components, the gradient of its distance map gives a vector field that captures the static shape of an object and provides shape-maintaining forces. This field vector enables the user to define the geometry of the simulated objects, and provides feedback reaction forces allowing a user to feel the model.

Breen *et al.* [117] represent the microstructure of woven cloth with interacting particles. Cloth is modeled as a collection of particles that conceptually represents the crossing points in a plain weave fabric. The various thread-level structural constraints are represented with energy functions that capture simple geometric relationships between particles within a local neighborhood. These energy functions are meant to encapsulate four basic mechanical interactions: thread collision, thread stretching, out-of-plane bending, and trellising. Eberhardt *et al.*'s particle system [118] is extended to model air resistance in cloth modeling.

Volino *et al.* 's animated deformable cloth object [119] is represented as a particle system with vertices forming irregular triangles. The object is considered to be isotropic, of constant thickness and with known elastic parameters. Considering only the continuous constraints initially (such as wind and gravity), differential equations are solved using a time step that ensures mechanical stability for all vertices (computed from the acceleration of the vertices versus the length of their connected edges). Then, collisions are detected and direct correction of position and speed complying momentum transfer laws occurs. This way, the collision evaluation time is roughly proportional to the number of colliding elements, and independent of the total number of elements that compose the deforming surface.

Baraff and Witkin [120] describe a cloth simulation system that couples a technique for enforcing constraints on individual cloth particles with an implicit integration method. The cloth is modeled as a triangular mesh, with internal cloth forces derived using a simple continuum formulation that supports modeling operations such as local anisotropic stretch or compression.

2.3.8. Implicit Models

Instead of using parameters such as (u, v) to explicitly describe a point on the surface as in the case of curves and surfaces, an implicit function of the form:

$$f(x,y,z) = f(p) = 0 \tag{2.6}$$

can be used. In other words, a point p is on the implicit surface if the result is 0 when the coordinates of the point are inserted into the implicit function f.

2.3.8.1. Advantages and Disadvantages of Implicit Models

A main advantage of implicit representations over their parametric counterparts is that they comprise more information about their interior and exterior. Implicit surfaces are inherently manifold. They can be collided and deformed [121]. Moreover, for the same polynomial of degree n, implicit algebraic surfaces have more degrees of freedom, compared with parametric surfaces of the same degree. Therefore, implicit algebraic surfaces are more flexible to approximate a complicated structure with fewer pieces or to achieve a higher order of smoothness. Implicit surfaces can be extended to implicit volumes.

Despite their many advantages, implicit surfaces and volumes are difficult to render efficiently. Today's real-time graphics systems are mainly optimized for rendering triangles, implying that an implicit surface must be converted to a mesh of triangles before being rendered. The representation being multi-valued may cause the real zero-contour surface to have multiple sheet, self-interactions and several other undesirable singularities [121]. Their use is thus limited by the difficulty in identifying specific locations on a surface.

2.3.8.2. Applications of Implicit Models

Cani-Gascuel and Desbrun [122] design and animate complex deformable models with implicit surfaces, particularly rigid articulated bodies coated with implicit flesh and soft substances performing separation and fusion. Desbrun and Gascuel [123] present a hybrid model for animation of soft inelastic objects, that undergo topological changes (e.g. separation and fusion) and that fit with the objects they are in contact with. Du and Qin [124] couple partial derivatives equations with volumetric implicit functions, in order to achieve interactive shape representation, manipulation and deformation for sketch curve sculpting.

2.3.9. Mesh-Free Methods

Mesh free methods are used to establish a system of algebraic equations without the use of a predefined mesh as in the case of FEM or BEM. Even though they use a set of nodes scattered on the boundaries of the problem domain to represent it, these nodes do not form a mesh, which means that no information on the relationship between these nodes is required [125].

2.3.9.1. Advantages and Disadvantages of Mesh-Free Methods

The creation of a mesh that is prerequisite for FEM or BEM methods is eliminated here. These methods are a natural way to process readily available point-clouds provided by measuring equipment, especially laser scanners. Since a whole modeling pipeline starting from surface representation, digital processing, physics-based modeling and rendering is possible without the employment of meshes, mesh-free methods are a promising research direction. However, the framework is fairly new and therefore less explored.

2.3.9.2. Applications of Mesh-Free Methods

Liu [125] develops mesh free models for solving mechanics problems, such as mesh-free methods for shells, beam and plates. However, the widest application of mesh-free methods is in the area of computer graphics for animation of elastic, plastic and fluid objects [126-129].

Muller *et al.* use continuum elasticity theory and point-based models to simulate the animation of elastic, plastic and melting objects [126, 127]. After assigning a mass for each material point (all the simulation quantities, such as location, density, deformation, velocity, strain, stress and force are carried by the physically simulated points) and using the assumption that the material is linear and isotropic, the behavior of each point is modeled using an implicit time integration of Newton's second law of motion. The method is similar to the one employed for the simulation of particle systems. Plasticity can be modeled using a strain-state variable that is attached to each point and that stores the degree to which the force is elastic or plastic. In a similar manner, fluid simulation can be done by introducing hydrodynamic equations for each point as in the case of particle-based fluid simulation.

Chang and Zhang [128] model deformation on point-clouds by applying displacements (displacement boundary conditions) or tractions (force boundary conditions) on the surface, based on the superposition principle in elastic mechanics which states that any complex loading can be equivalently produced with a combination of simple loadings. Objects are embedded in infinite bodies. A set of points are created on the surface of the object (regularly or randomly distributed) to capture its geometry and boundary conditions, and another set of points are distributed around the object where virtual forces (concentrated forces in the infinite body) are exerted. Considering the elastic parameters known, the deformations that occur on the infinite body are reduced to a linear system of equations solved using a minimization algorithm that determines the strength of virtual forces for any point. These forces are then used to determine the deformation either for surface or boundaries of the object under study.

Muller *et al.* [129] employ methods similar to modal analysis for point-clouds. The input points are simulated as a simple particle system without inter-particle interaction but including collision with the environment and external forces such as gravity. At each time step the particles are pulled towards their goal position. To compute individual goal positions, the original configuration is matched with the actual configuration defined by actual positions of the particles. The method can be used to simulate rigid bodies, linear (shear, stretch) and quadratic deformations (twist and bend) as well as cluster based deformations.

Schmitt *et al.* [130] deform heterogeneous volumetric objects defined as point sets with attributes. A heterogeneous volumetric object has a number of attributes assigned at each point. These attributes are mathematical models of object properties of arbitrary nature (material, photometric, physical, statistical, etc). While function representation directly defines object geometry, for an attribute it specifies a space partition used to define the attribute function. The set of proposed deformations is point based, as arbitrary points in the space are moved to arbitrary positions. Traditional field functions are then used to define the deformation.

2.3.10. Neural Networks

Neural networks are designed similarly to the biological neural pathways and perform the same basic task: processing input data, arranging this data into some sort of meaningful order, and then passing the information to the outputs. The basic architecture of a neural network consists of a large number of simple processing elements, called neurons. Each neuron has an internal state, called activation function, which is a function of the inputs it receives. The neurons are connected to form a network. The way in which neurons are connected defines the network topology or the network architecture. Most neural networks have a layered topology consisting of an input layer (neurons in this layer receive external inputs from outside the network) and an output layer (neurons in this layer produce some of the outputs of the network). The inner layers of a network are referred to as "hidden layers" since they receive neither input nor output information from outside of the network. The neurons are capable of storing information, which determines their output for any given input. By convention this information is stored in the weights associated with each communication link. Neural networks can modify this information during the learning process. Learning consists in the adjustment or adaptation of the network weights or parameters over time, such as to permit the system to perform the task it has been designed for. There are three types of learning: supervised, unsupervised and reinforcement learning. In the supervised learning, training is accomplished by presenting a sequence of training vectors, each associated with a target output vector. An error is computed between the output and the target vector. This error provides an external feedback, which is used to adjust the network. In unsupervised learning a sequence of input vectors is provided, but no target vectors are specified. The net adjusts itself such that the most similar input vectors are assigned to the same output unit. Reinforcement learning is an intermediate between supervised and unsupervised learning. Instead of providing total feedback (the error in supervised learning) or no feedback (as in unsupervised learning), a moderate feedback is obtained [1].

2.3.10.1. Advantages and Disadvantages of Neural Networks

The strength of neural networks stems from their intrinsic non-linearity, their computational simplicity and ability to learn, and subsequently to generalize. A neural network has the capacity to learn high-dimensional, redundant mappings from a limited set of measured data, without any *a priori* mathematical model. They have the capability to provide fast estimations for output values that were not part of the training set, due to their continuous memory. Even though they may take a relatively long time to learn, especially in the case of large datasets or where accuracy is desired, the recall phase is done real-time, which is very important in the context described in this thesis.

However, the training parameters and the network size are not always easy to set in certain modeling scenarios, and in spite of certain existing guidelines of choosing a network size for a certain input configuration, the modeling with neural networks can sometimes become a long trial-and-error procedure.

2.3.10.2. Applications of Neural Networks

In general, neural networks are used for a wide variety of tasks, including function approximation, pattern recognition, clustering, feature extraction and control-related tasks, just to mention a few. In the areas of modeling and computer graphics, neural networks have been applied for data fusion and improvement of measured data using *a priori* knowledge, projection, visualization and exploration of high-dimensional data, reconstruction of surfaces and shapes from scattered data, range data and shape from shading, object representation and modeling, modeling of isosurfaces, segmentation, shape matching and volume matching, object classification and object motion [1].

For elasticity modeling, as it was shown in section 2.1.2, neural networks are employed to model the relationship between stress and strain [8], to recuperate the parameters of applied force from the stress distribution [9] and to learn the mapping between a point in a undeformed body and the same point in a deformed body [18].

In the context of deformable object models, Grzeszczuk *et al.* [131, 132] propose the NeuroAnimator, an approach to create physically realistic animation of objects, employing neural networks. NeuroAnimators are automatically trained to emulate physical dynamics through the observation of physics based models. The NeuroAnimator learns an approximation to the dynamic model by observing instances of state transitions, as well as control inputs and/or external forces that cause these transitions (state of the model, its control inputs, and the external forces acting on it at time) and emulates continuous animations.

Saito *et al.* [133] build a solution-specific neural-network-based deformation simulator for cutting soft material with a manipulator equipped with a blade. The proposed model is based on two networks that perform global and local deformation. The global deformation module simulates the propagation of distortion through the material and the local deformation module gives the local distortion. The global deformation module is a 2D grid of backpropagation networks. Each network communicates positional information to its four neighbors. The distortion transferred to neighbors is attenuated to avoid oscillations in case the interaction is too strong. The local deformations module, based on a 3-layered backpropagation network, predicts local distribution of material with data given by a controller or visual monitor. It outputs the shift of each lattice point. To obtain training data, a deformation detecting network is employed to extract shifting vectors for lattice points from an image sequence.

2.3.10.3. Final Remarks on Neural Networks

In spite of the fact that neural networks are used as modeling tools in the context of this thesis, no elaborate literature review is presented here on the existing different architectures, a field that is extremely wide. The thesis does not propose novel architectures and employs only existing ones. The classical neural networks, i.e. feed-forward architectures, are well-known and established and have been extensively discussed in literature and therefore are not elaborated here. The interested reader is invited to consult classical references on the topic [134, 135]. However, less known architectures employed in this thesis, such as neural gas and growing neural gas will be presented within the chapters where they are used.

2.4. Conclusions

The survey presents existing techniques and research trends in the deformable objects modeling context, with particular stress on different types of models. There is a vast collection of papers in the literature studying different aspects of deformable models. Although the domain has been studied for more than 3 decades, there is an increasing interest in the subject after the year 2000. Most of the methods have been proposed to respond to the increasing requirements of real-time interaction speed of deformable models in the detriment of accuracy and realism. Regarding the areas of applications, the vast majority of models are highly specialized and dedicated to a particular application (e.g. almost 50% of the total number of published papers is focused on the modeling of organs and soft tissues in surgical simulations).

Although significant advances have been made, the field of deformable object modeling is still open. Further research is required to respond to the increasing need to support realism and accuracy of complex real-world object models built from experimental data collected by sensors.

3. Neural Network Modeling of Elastic Behavior from Experimental Data

The chapter explores means to model elastic behavior from experimental measurements. It starts by showing how elastic behavior can be collected and modeled from force-displacement curves knowing a mesh of an object, and how the modeling results differ in quality from classical solutions, in this case the Three-element method. Then a novel method is proposed to elicit and model elastic behavior. This method combines vision and tactile probing and allows for the measurement of elastic behavior directly from point-clouds, without any assumption on the material of the objects under study. This novel method allows for the modeling of 2.5D deformation profiles of probed objects. Several experiments are performed for each of the cases and results are presented to validate the proposed solutions.

3.1. Modeling Framework

In the process of elastic behavior measurement and modeling, due to the fact that the elastic constants are material-related parameters and do not accurately describe real-world objects in general, the proposed approach will not attempt to recover explicitly neither the elastic constants, nor the stress and strain tensors for an object under loading. Instead, quantities of interest are observed at the surface of a real object, namely the displacement of the surface as the object is loaded by a measured external force at a given point. Therefore the forces of interest in the context of this work are surface (traction) forces produced as a result of the physical contact between the object under study and the probing equipment.

The mapping between the displacement and forces is learned inherently using neural networks. Such an approach avoids the complex and time consuming methods to recuperate elastic parameters from experimental data and any assumptions on the material under study that such methods imply. Moreover, neural networks are able, unlike any sort of analytical models, to provide estimates on the elastic behavior on surrounding points that were not probed and therefore were not part of the training set. Their fault-tolerant nature improves the behavior of the proposed modeling scheme in the presence of noise, which is inherent in any real measurement data. Furthermore, using a set of error measurements, the neural network system can be automatically tuned to a certain data set, eliminating the need of an expert user to make the necessary adjustments for each material to be modeled. The modeling approach is tested on two different forms of data that represent the elastic behavior of objects: force-displacement curves and deformation profiles correlated with force recordings.

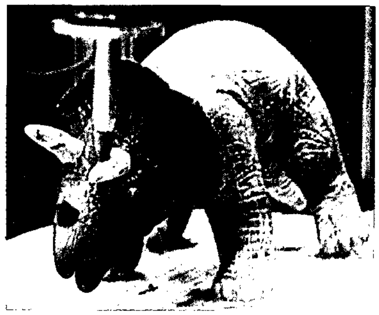
The proposed modeling framework based on deformation profiles is easily extensible to special cases of deformable objects, such as rigid objects, plastic or elasto-plastic objects.

3.2. Modeling of Elastic Behavior from Force-Displacement Curves

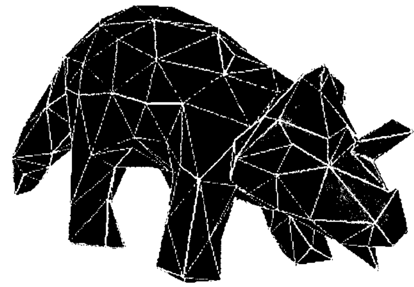
The experimental data used for modeling force-displacement curves is not collected in our lab, but was made available to us for testing purposes. The tactile probing is based on a mesh of an object. The collected data requires pre-processing prior to modeling. Also the modeling does not consider angles at which the probing is performed, which brings limitations in case of anisotropic and highly nonlinear materials. In spite of these drawbacks, due to the fact that the elastic parameters of material are assumed to be known, this case not only presents a good basis for comparisons of the proposed modeling solution with classical solutions for elastic behavior modeling, but also shows the capability of the elastic behavior modeling framework to work for different forms of input data.

3.2.1. Experimental Data

The experimental data that we have available has been collected by a force probe recording the 3D force at its tip and its position with respect to the surface of the object. The procedure is described in [136] and no detailed description will be provided here on the way the data was collected since it was not part of our experimental work. Briefly, samples were taken at approximately even spacing over the surface of the object at each vertex of a coarse triangular surface mesh with approximately equal size triangles. The test object was a toy triceratops depicted in Fig. 3.1a with the coarse mesh shown in Fig. 3.1b.



a)



b)

Fig. 3.1. a) Toy triceratops and b) coarse initial mesh.

The probe, an ATI force-torque sensor, applies forces at various angles. The force components and directions are recorded from the force-torque sensor and the corresponding displacement is computed based on the information provided by a stereo vision system [136]. Data in the form of force and surface displacement curves over time are collected at each vertex of the mesh, as depicted in Fig. 3.2.

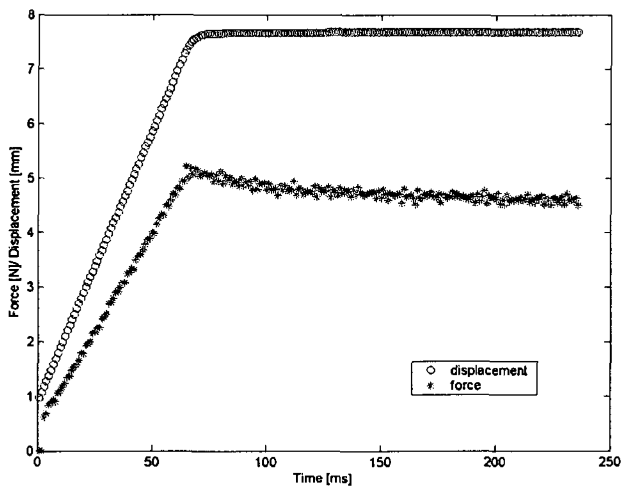


Fig. 3.2. Measured force-displacement curve over the period of force application.

Such curves can serve as a measure of apparent stiffness (compliance) of the surface. A surface compliance tensor is first estimated based on a linear approximation of the traction (force per area) and displacement curves. The surface tensor is then employed to scale the magnitude of the traction and displacement in the direction normal to the probe at the contact location. Therefore the angle at which the force is applied is not considered directly in this experiment. Measured force data is modeled instead of calculating traction in order to avoid errors due to the coarse linear approximation of the force distribution over the traction area.

3.2.2. Modeling Force-Displacement Curves

3.2.2.1. Data Set

The data set made available to us is noisy. This is mainly due to the imprecision of the placement of the probe. An imprecisely placed probe results in errors in the computation of the surface compliance tensor and causes variations in the force and displacement magnitude estimation. The force-torque sensor also collected some static noise in some measurements. Several raw force and magnitude measurements for the same vertex, scaled to normal directions, are depicted in Fig. 3.3.

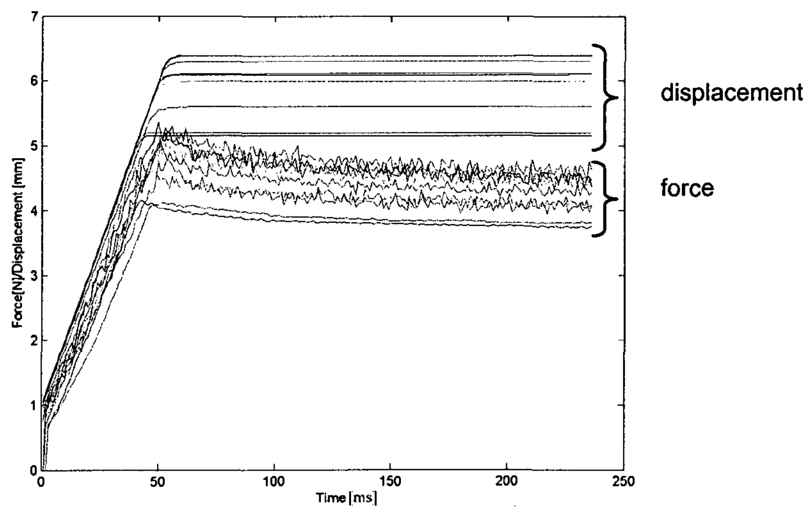


Fig. 3.3. Repeated measurement force-displacement curves.

In order to avoid the modeling of the initial contact error, the first three samples of each curve are removed from the data set. Overall, it can be noticed that the noise in the measurements is quite high. To alleviate its effect and eliminate biased measurements, median force and displacement curves are estimated for each vertex and are modeled as a representative of all the measurements performed for the respective vertex. However, with this procedure not only the measurement noise is reduced, but also the variation of elastic properties in a region. This implies that the obtained model is not very accurate, but still acceptable for many modeling applications.

Fig. 3.4 shows the median force and displacement curves for the vertex measurements presented in Fig. 3.3. The median curves are shown with stars.

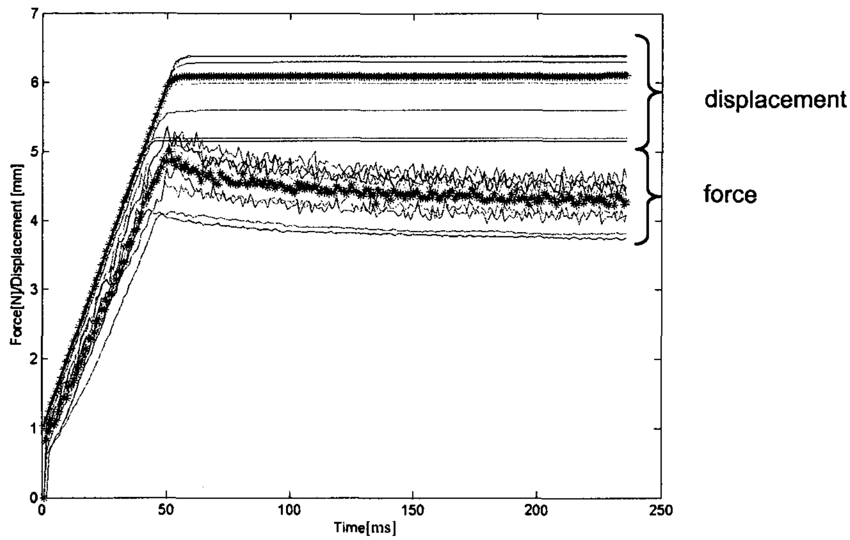


Fig. 3.4. Median force-displacement curves.

After the computation of median values, the data set is scaled such that its variance is unity as required by the neural network training procedure.

3.2.2.2. Neural Network Architecture

In spite of the fact that good modeling results were obtained with other architectures as well, such as the Elman network for example [137], only modeling results with a feedforward neural network are presented here to keep the approach consistent with the framework presented in Fig. 1.1.

The network takes as an input a force vector (F), corresponding to the force curve over time and outputs the corresponding displacement vector (displacement curve) (Z) at a given vertex. The same network is employed for all vertices. The network is therefore employed to model the generalized relationship or mapping from forces to displacements in order to avoid the need of recuperating the elastic parameters. The architecture with one input neuron, 15 hidden neurons with sigmoid activation functions and one output neuron depicted in Fig. 3.5 is used to model force-displacement curves.

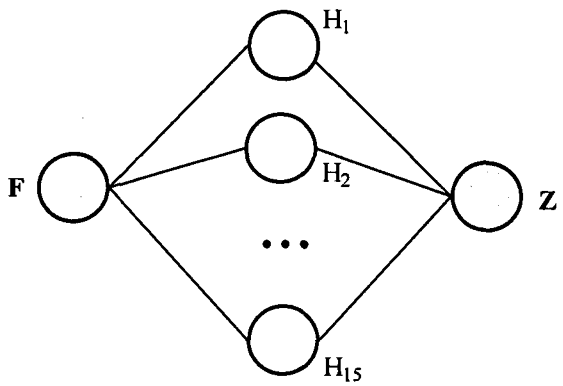


Fig. 3.5. Feedforward neural network to learn elastic behavior from force-displacement curves (F=force vector, $H_{1..45}$=hidden neurons, Z=displacement vector).

The training data set is split into two sets: a part for training the network and another part for testing. The networks are trained for 10000 epochs using backpropagation, with a learning rate of 0.1 and a training error goal of 0.0001 on a selection of 80% of the total number of scaled force-displacement curves. The rest of 20% of the measured data set is kept for testing. The training is based on Matlab code, running on a Pentium IV 1.3GHz machine with 512MB of memory. The choice of the number of neurons in the hidden layer and the choice of the training algorithms are further discussed.

3.2.2.3. The Number of Neurons in the Hidden Layer

When choosing the number of neurons in the hidden layer, one is searching for a best possible compromise between the learning accuracy and generalization capability versus the time of modeling (learning). These capabilities are reflected by two types of errors that can be computed based on the training and testing data sets. The learning or training error reflects the memorization capability of the network, meaning its capability to reproduce correct output data for input data that was part of the training set. It is computed as an average mean square error between the training and the modeled data over all the displacement curves in the training set. The generalization error reflects the capability of the network to correctly produce estimates of output data for input data that was not part of the training set. The testing set (in this case 20% of the measured data set) is built for this purpose. The testing error is computed as an average mean square error between the testing set and the corresponding modeled data over all the displacement curves in the testing data set. The training and generalization errors for the modeling of the force-displacement curve of the data set shown in Fig. 3.2 are depicted in Fig. 3.6 and Fig. 3.7.

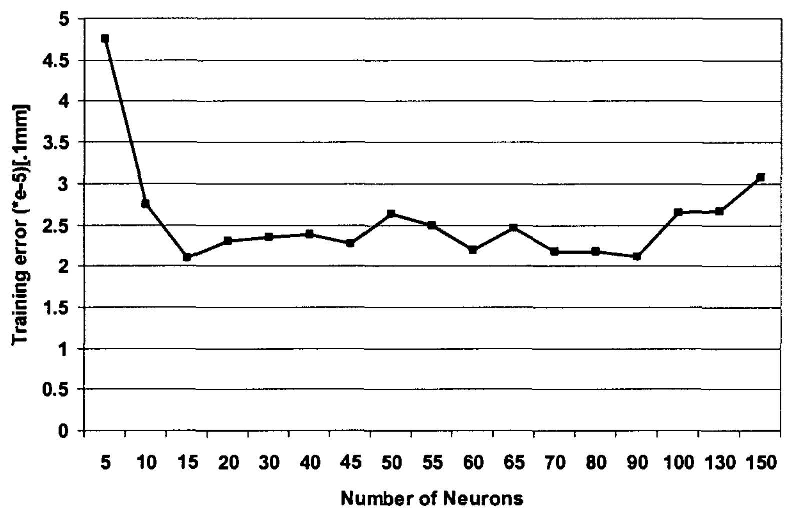


Fig. 3.6. Training error for different numbers of neurons in the hidden layer.

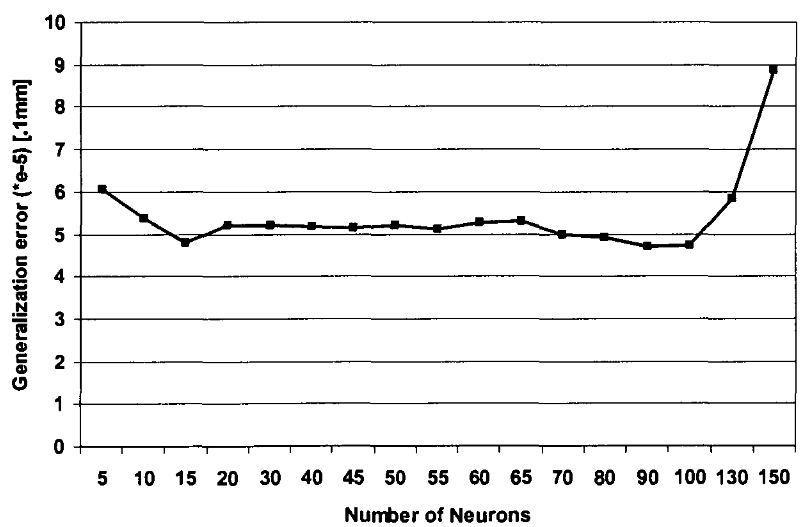


Fig. 3.7. Generalization error for different numbers of neurons in the hidden layer.

As it can be seen in Fig. 3.6 and Fig. 3.7, both errors are very low, of order 10^{-5}. Fig. 3.8 shows the corresponding training time, measured in seconds for different numbers of neurons in the hidden layer.

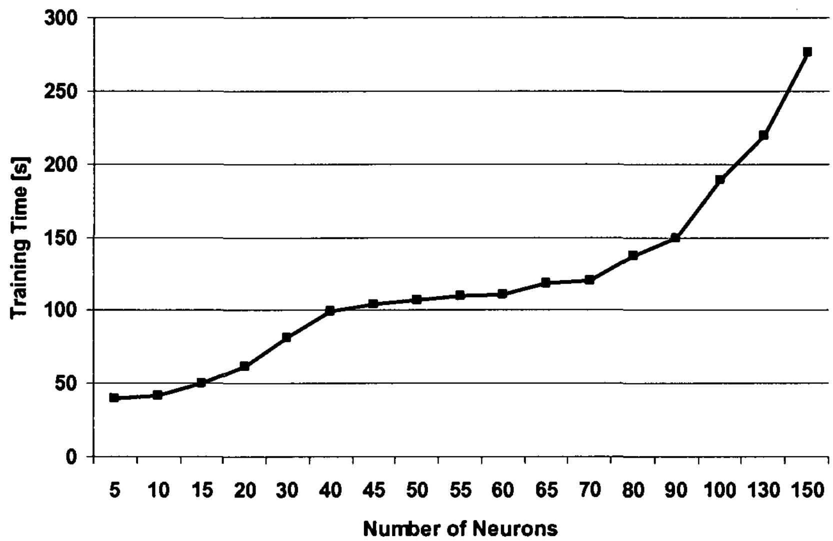


Fig. 3.8. Training time for different numbers of neurons in the hidden layer.

For this modeling case, it can be observed that any size of network having between 15 and 100 neurons would work, as both the generalization and training error are lower for this range of numbers of neurons in the hidden layer and tend to grow when the size of the network reaches out of this range.

For too few neurons, the network is under-fitting and does not learn well, an aspect that is reflected by the high training error in Fig. 3.6 for a number of neurons in the hidden layer less than 15. For too many neurons, the network memorizes quite well but does not generalize well, as shown in Fig. 3.7. For more than 100 neurons the generalization error grows exponentially. The training time grows with the number of hidden neurons, as depicted in Fig. 3.8. Considering all these aspects, it can be concluded that for this modeling case a number of 15 neurons in the hidden layer is a good compromise between the accuracy of modeling results and the training time.

This modeling case proves as well that the knowledge of training and generalization errors can be simply employed to automatically tune the neural network system to a certain data set based on the desired level of accuracy and the tolerable training time for a certain modeling case, without requiring an expert user to adjust the size of the network.

3.2.2.4. Training Algorithms

Several backpropagation algorithm variations were tested for the given modeling case, namely the Levenberg-Marquart (LM) method, the quasi-Newton BFGS (Broyden-Fletcher-Goldfarb-Shanno) method [138, 139] and the resilient propagation (Rprop) [140]. The Levenberg-Marquart and quasi-Newton BFGS methods are classical gradient-descent (steepest-descent) methods. Both are variations of Newton's method. They use the gradient of the error function to compute the search direction (the negative gradient direction points to the locally steepest direction). The error function is considered to be quadratic, and therefore the direction towards the minimum error is determined by the inverse of the Hessian, multiplied by the gradient, product which is known as the Newton step.

For each iteration of the algorithm, a quadratic function is used to approximate the global error function $E(x_{k+1})$ in the current point x_{k+1}:

$$E(x_{k+1}) \approx E(x_k) + g_k^T \Delta x_k + \frac{1}{2} \Delta x_k{}^T g_k^2 \Delta x = \frac{1}{2} x^T A x + d^T x + c \qquad (3.1)$$

where

$$g_k = \nabla E(x)\big|_{x=x_k} \qquad (3.2)$$

denotes the gradient evaluated at the precedent guess x_k and k denotes the iteration step.

In order to minimize the equation, the critical points have to be first found. They can be obtained as the solution of the linear system defined by the equation:

$$E''(x_{k+1}) = g_k + g_k^2 \Delta x_k = 0 \qquad (3.3)$$

Solving for Δx_k gives:

$$\Delta x_k = -(g_k^2)^{-1} g_k \qquad (3.4)$$

where the term $-(g_2^k)^{-1} g_k$ is called the *Newton step* and g_2 is the *Hessian.*

Therefore Newton's method uses the following iterative formula:

$$x_{k+1} = x_k - (g_k^2)^{-1} g_k \tag{3.5}$$

When the error function is indeed quadratic, the Newton's method will find the minimum in one step. If it is not quadratic, in most cases it will converge quickly since many analytical functions can be approximated accurately in a small neighborhood of a strong minimum by a quadratic function. However, the exact computation of the inverse of the Hessian is computationally demanding. Furthermore, when the Hessian is not positively definite, the Newton step can move towards a maximum, not a minimum.

The Levenberg-Marquart modification adds a term to the Hessian to ensure that the latter it is always positive definite. Therefore, the modified Newton method becomes:

$$x_{k+1} = x_k - (g_k^2 + \lambda I)^{-1} g_k \tag{3.6}$$

where I is the identity matrix and λ is a non-negative value.

To save the computation of the inverse of the Hessian, the BFGS method uses a series of approximations to the inverse of the Hessian $(g_2^k)^{-1}$ in the form of:

$$G_{k+1} = G_k + \frac{ww^T}{w^T v} - \frac{(G_k v)v^T G_k}{v^T G_k v} + (v^T G_k v)uu^T \tag{3.7}$$

where:

$$\begin{aligned} w &= x_{k+1} - x_k \\ v &= g_{k+1} - g_k \\ u &= \frac{w}{w^T v} - \frac{G_k v}{v^T G_k v} \end{aligned} \tag{3.8}$$

The update made by the BFGS method is then:

$$x_{k+1} = x_k + \alpha_k G_k g_k \tag{3.9}$$

Even with the Hessian approximation only, the method is still computationally demanding, and in those cases when the magnitude of the input is large, the derivative of the error function becomes very small and all steepest descent methods produce very small steps, sometimes never converging.

The resilient propagation method gives usually better results in significantly less time. In this method, the step size is not a function of the magnitude of the gradient, but rather depends on the sign of the derivative [140]. The update made by the resilient backpropagation is in form of:

$$x_{k+1} = x_k + \Delta x_k \tag{3.10}$$

where

$$\Delta x_k = \begin{cases} -\Delta_k, \ if \ g_k > 0 \\ \Delta_k, if \ g_k < 0 \\ 0, else \end{cases} \tag{3.11}$$

and

$$\Delta_k = \begin{cases} \eta^+ \Delta_{k-1}, \ if \ g_{k-1} g_k > 0 \\ \eta^- \Delta_{k-1}, if \ g_{k-1} g_k < 0 \end{cases}, \quad 0 < \eta^- < 1 < \eta^+ \tag{3.12}$$

In other words, if the sign of the derivative for a given weight remains the same over several iterations, then the magnitude of the size step is increased and if the sign oscillates, the magnitude of the size step is decreased. However, there is one exception. If the partial derivative changes sign, that is the previous step was too large and the minimum was missed, the previous weight update is reverted:

$$\Delta x_k = -\Delta x_{k-1}, \ \ if \, g_{k-1} g_k < 0 \tag{3.13}$$

The fact that only the sign of the derivative is used, leads to an efficient computation with respect to both time and storage consumption [141], even over long sets of deformation profiles with relatively smooth features.

In the present application, all variations of backpropagation succeeded to capture correctly the characteristics of the deformation profiles, but their computing requirements are different, as depicted in Fig. 3.9.

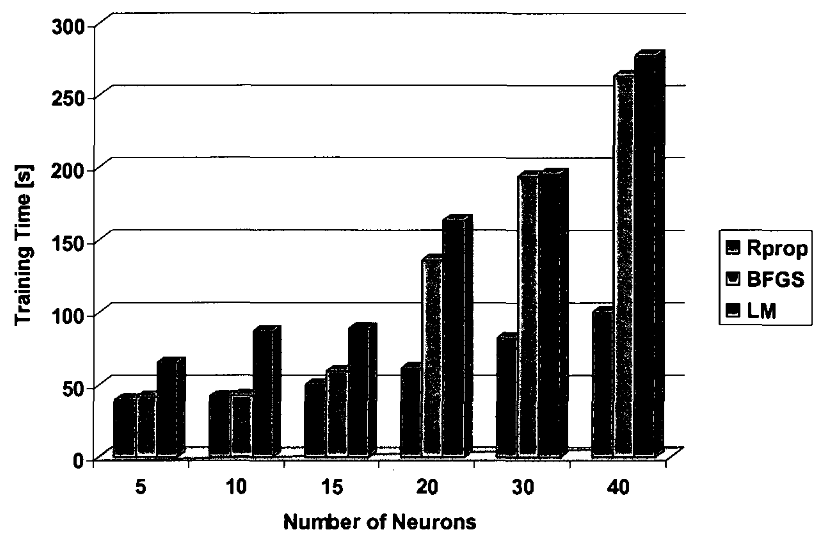


Fig. 3.9. Comparison in training time for different training algorithms and different numbers of neurons in the hidden layer.

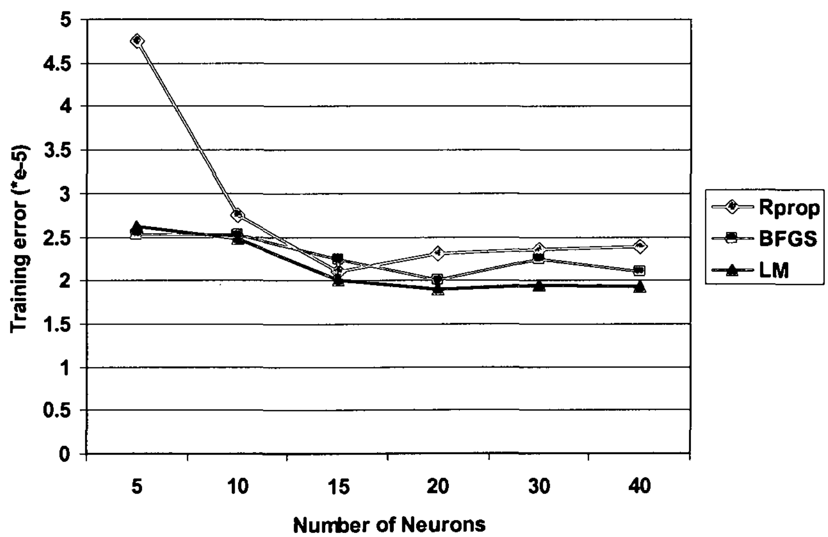


Fig. 3.10. Comparison in training error for different training algorithms and different numbers of neurons in the hidden layer.

Even if very good convergence is obtained for all algorithms, with errors of the order 0.00002 as seen in Fig. 3.10, the quality of the modeling results varies slightly with the algorithm used for training. It can be seen that, for this modeling case, the resilient backpropagation provides the best compromise between training time and training error for 15 neurons in the hidden layer.

Fig. 3.11 shows an example of training/target data produced by the network for the data set in Fig. 3.4, with the force depicted in green, the corresponding measured displacement in blue and the modeled displacement in red circles for 15 neurons and resilient propagation training algorithm. It can be seen that the network models quite accurately the displacement curve.

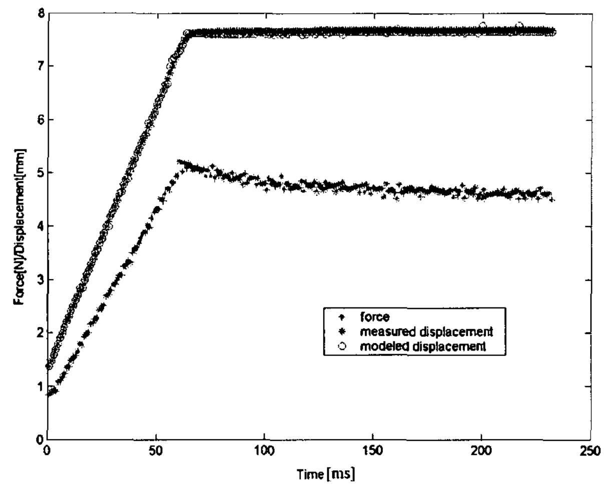


Fig. 3.11. Example of training results for median force-displacement curves in Fig. 3.4.

3.2.3. Comparison with the Three-Element Method

To validate the modeling results obtained using neural networks, a comparison is made with the Three-element method [142], whose model is shown in Fig. 3.12.

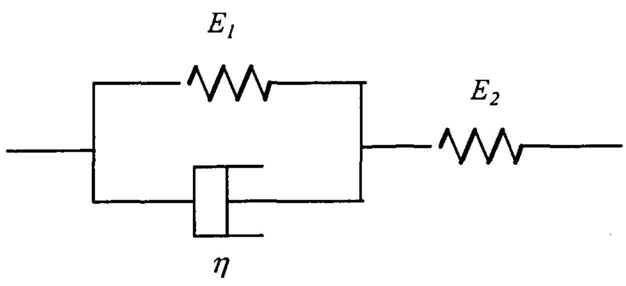


Fig. 3.12. Three-element model.

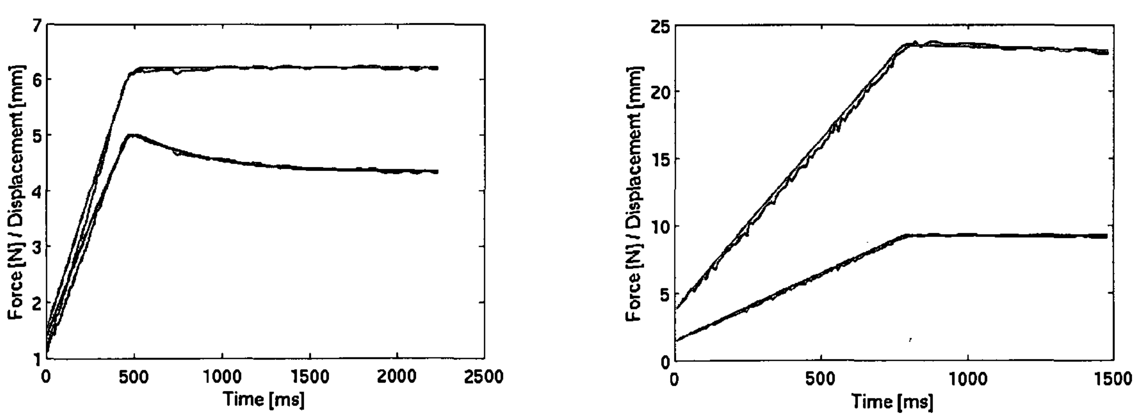
The Three-element method model is chosen for comparison since it is slightly more complicated and therefore more accurate than a simple spring model (based on the assumption of linear elastic behavior) or the simplest visco-elastic model, the Kelvin-Voigt, that consists of a damper and a spring connected in parallel. The Three-element model relates measured tractions and displacements by a linear visco-elastic dynamic system, shown in Fig. 3.12.

The system consists of a damper and spring in parallel which are connected in series with another spring. With this model, the one dimensional relationship between force and displacement is given by a differential equation in the form of:

$$F = E_2(u - x) = E_1 x + \eta \dot{x} \qquad (3.14)$$

where F is the force, u is the total displacement, x is the displacement of the damped spring, η is the damping coefficient and E_1 and E_2 are the elastic constants of the springs. The one dimensional relationship above is extended to the 3D case by replacing both the force and the displacement scalars by 3-by-3 tensors. The parameter fitting for the three-element model is performed by minimizing the sum of the weighted quadratic error in force given the displacement. The error is calculated between measurements and the results of a simulation with an implicit first-order finite difference scheme. The minimization is calculated with the Nelder-Mead simplex algorithm [142].

The results of a fitting with the three-element model, for two vertices with different elasticity characteristics collected on the triceratops model, are shown in Fig. 3.13a and Fig. 3.13b respectively.



a) b)

Fig. 3.13. Force/displacement curve fitting with Three-Element method on two vertices of the triceratops model.

The measured data displacement is shown in red, the measured force is depicted in blue and the output of the Three-Element model for both force and displacement is shown in black.

The Three-element model fits the median curves well in most cases. However, the case presented in Fig. 3.13a shows a derivation from the linear elastic behavior, and in this case, the fitting is not accurate over the early part of the curve (in the loading stage).

The results for the problematic measurement in Fig. 3.13a are compared with the proposed feedforward neural network solution in Fig. 3.14. The figure shows on the same graph the results obtained by the neural network simulation and the three-element fitting for the whole curve in Fig. 3.14a and then a detail of the loading stage in Fig. 3.14b.

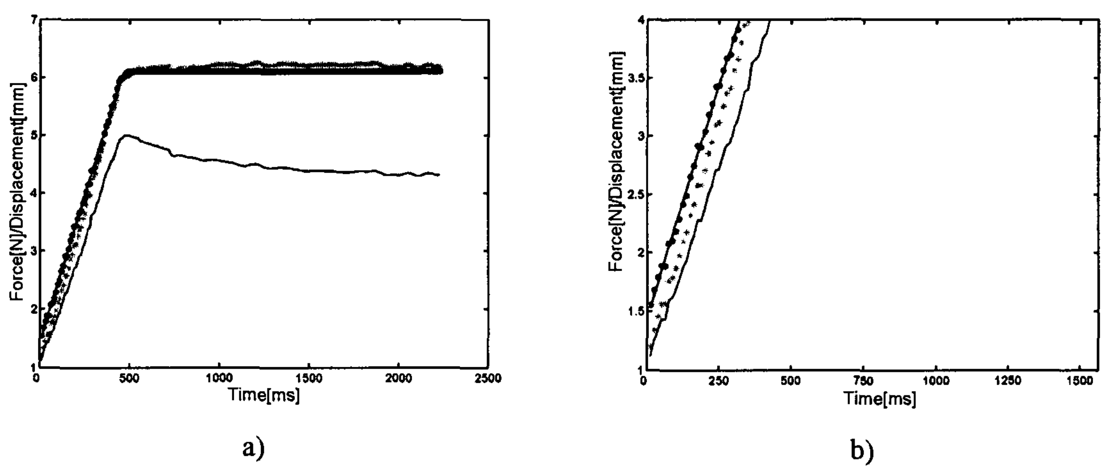


Fig. 3.14. Comparison between neural network result and the fit of the Three-Element model for (a) the entire force-displacement curve and (b) a detail of the loading stage.

The real displacement is shown with a continuous red line, the modeled displacement using neural networks is shown with red dots and the fitting provided for the real displacement curve by the Three-element method is shown with blue dots. As it can be seen, the neural network solution provides more accurate estimates of elastic behavior in case of objects that exhibit nonlinear elasticity.

3.2.4. Final Remarks

The same procedure described above can be used to model the elastic information collected at all vertices. In all studied cases, the proposed backpropagation network captures and estimates correctly the displacement curve. However, during this experiment, the measured data had to be heavily pre-processed. Therefore the result of the modeling is only approximately accurate. Measured forces are projected to normal direction only (the angle of interaction is not considered), while it is expected that the addition of the angle in modeling will improve the modeled behavior especially for objects exhibiting highly nonlinear characteristics. Moreover, the fact that the data was not collected in our lab limits the control and restricts the potential for testing in various conditions. All these motivated the development of a slightly different approach, based on an in-house experimental setup to elicit, record and model elastic behavior.

3.3. Modeling of Elastic Behavior from Deformation Profiles Correlated with Force Recordings

The experimental data used for modeling deformation profiles correlated with force recordings is collected in our lab. As in the previous case, the study is focused on the measurement of the displacement of the surface as an object is loaded by an external force, at a given point. However, the data collection is done over the whole surface visible to the sensor in each static loading case, and not at a single point position unlike those in section 3.2. As a result, the obtained data sets are not related to time, as in the previous modeling case.

The measurement is done starting directly from point-clouds, without the need of a mesh, and angles at which the probing is done are also recorded in order to improve the accuracy of the modeling. As well, in this case, the pre-processing of data is minimal.

3.3.1. Experimental Setup

The experimental setup used to collect force/displacement data is comprised of a multi-axis *ATI* force/torque sensor [143] attached to a console computer, and an active triangulation line-scanning Jupiter laser range finder [144] controlled via a second PC system, as depicted in Fig. 3.15a. The force/torque sensor is used to record the force components applied on the object while the range finder captures the deformation profile of the surface of the object under the given load. The range finder is positioned such that the scan-line intersects with the point where the external force is applied, as highlighted by the trace of the laser on the object in Fig. 3.15b.

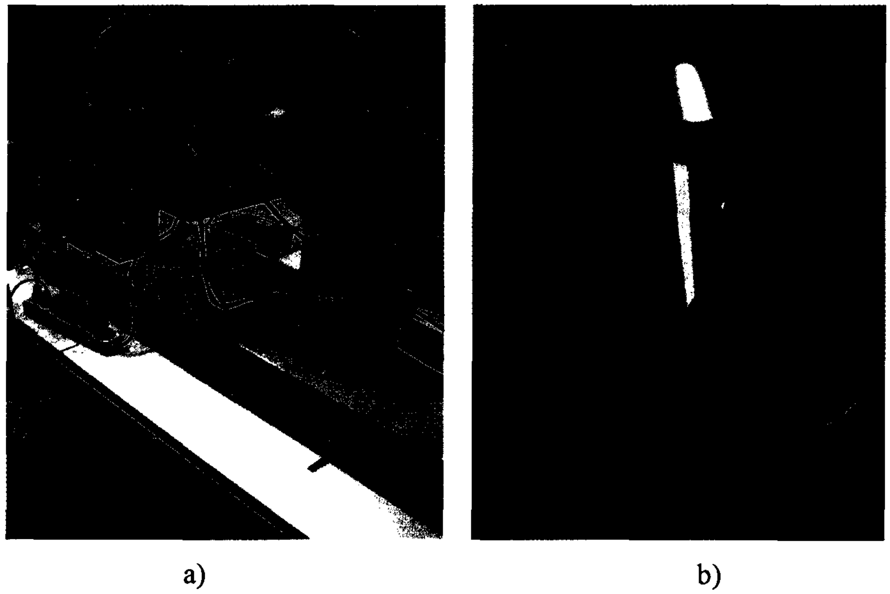


Fig. 3.15. a) Range sensor and force/torque setup producing b) a laser trace to capture the object deformation profile resulting from the applied force.

The raw deformation profiles are encoded under the form of 2D distributions of points in the *Y-Z* space, as shown in Fig. 3.16, where *Y* is the lateral displacement along the scan-line and *Z* the depth along the optical axis with respect to a back reference plane.

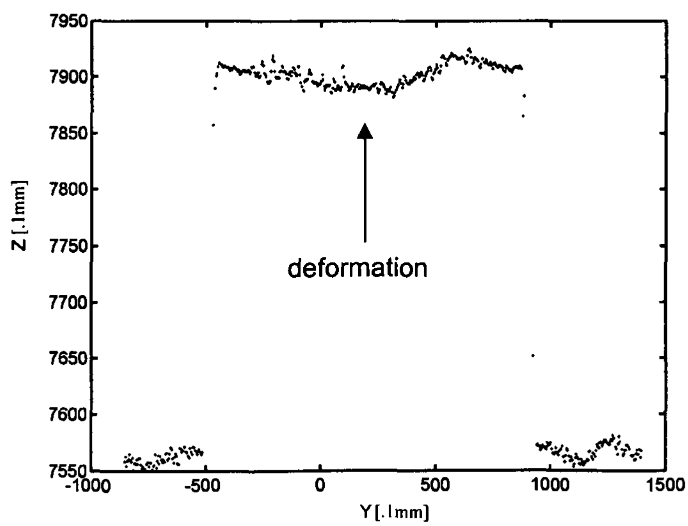


Fig. 3.16. Deformation profile mapped in the Y-Z space.

The laser range finder provides fast scans of 512 samples distributed along a straight line on the surface. A number of 75 to 100 scans of the same area are collected within a few seconds while the applied force and angle are kept constant. This technique provides efficient means to cope with the noise in the range data. Moreover, the success of the range data collection being highly sensitive to the texture and orientation of surfaces, missing measurements usually appear along the scan-line. The use of an iterative sampling procedure over a short period of time partially alleviates the impact of this constraint.

In order to filter out the average noise and include as many valid measurements as possible in areas where points could be missing in some of the scans, the mean value is computed on the depth (Z-axis) over all deformation profiles obtained under a given force. The resulting profiles are then saved for each magnitude and angle of force applied on the object.

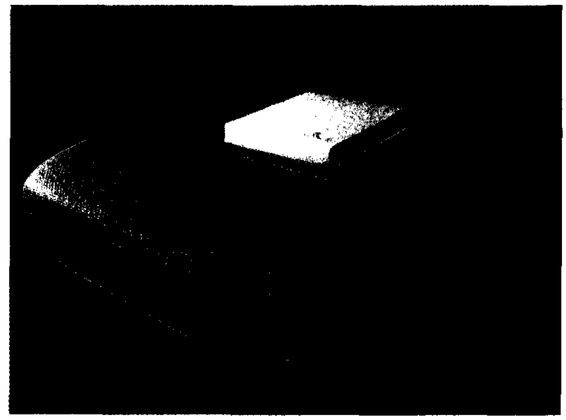
3.3.2. Elastic Data Modeling for One Scanning Direction

Using the proposed experimental setup, different magnitudes of forces are applied at different angles successively on selected sampling points using the probe attached on the *ATI* force/torque sensor and an average range profile is collected with the laser range finder for each force magnitude. As in the previous case, no explicit displacement information is recuperated from the range profiles. Instead the neural network models the raw range data mapped in the *Y-Z* space as a function of applied force, *F*, without explicitly defining values for the displacement.

3.3.2.1. Data Set

To validate the proposed modeling framework, experimentation was conducted on numerous deformable objects, of which a representative subset is reproduced here, that is a rubber ball and a simple composite object made of a square cardboard box mounted on the top of a covered foam pillow, as depicted in Fig. 3.17. Other examples of objects will be presented throughout the thesis.

a)



b)

Fig. 3.17. Objects used for experimentation: a) a rubber ball and b) a composite object.

These examples are chosen as they represent a significant combination of elastic materials, geometry and texture properties that are all critical factors that influence the performance of the proposed approach. From the geometry perspective some rounded surfaces (ball), edgy surfaces (cardboard box) and combinations of them (pillow with rounded edges) are presented, while from the elasticity perspective there are objects made of stiff material (cardboard), semi-soft materials (rubber) and highly nonlinear soft materials (foam). As well, some of the objects have light color (the rubber is yellow, the cardboard is light blue), others have dark color (the pillow is grey, with a highly scattered texture). All those parameters influence the accuracy of measurements collected with the laser scanner, as explained further on. The soft dark pillow is the most complex to be measured and modeled due to its nonlinear elastic behavior and non-uniform texture that makes the measurements using the laser range scanner difficult to collect. Therefore this example is shown to demonstrate the capability of the proposed method to cope with challenging modeling situations.

From the 3D point-cloud corresponding to the objects of interest, initially obtained with a complete scan of their entire surface with the laser range sensor, a random sampling algorithm returns a set of sampling points for the two objects under study. Force/deformation data is then collected for each of them.

Different magnitudes of forces at different angles are applied successively on the selected sampling points using the probe attached on the *ATI* force/torque sensor and a series of overlapping profiles are collected with the laser range finder for each force magnitude before being averaged to reduce noise. The force magnitude, F, is computed as the norm of the three orthogonal force components along the x, y, and z axes of the force/torque sensor's reference frame, as recorded by the device. The angle measurements are collected with the probe positioned in the plane parallel to the scan line that contains the probe tip, with the angle 0° starting on the far left and continuing up to the angle 180° on the far right, as depicted in Fig. 3.18.

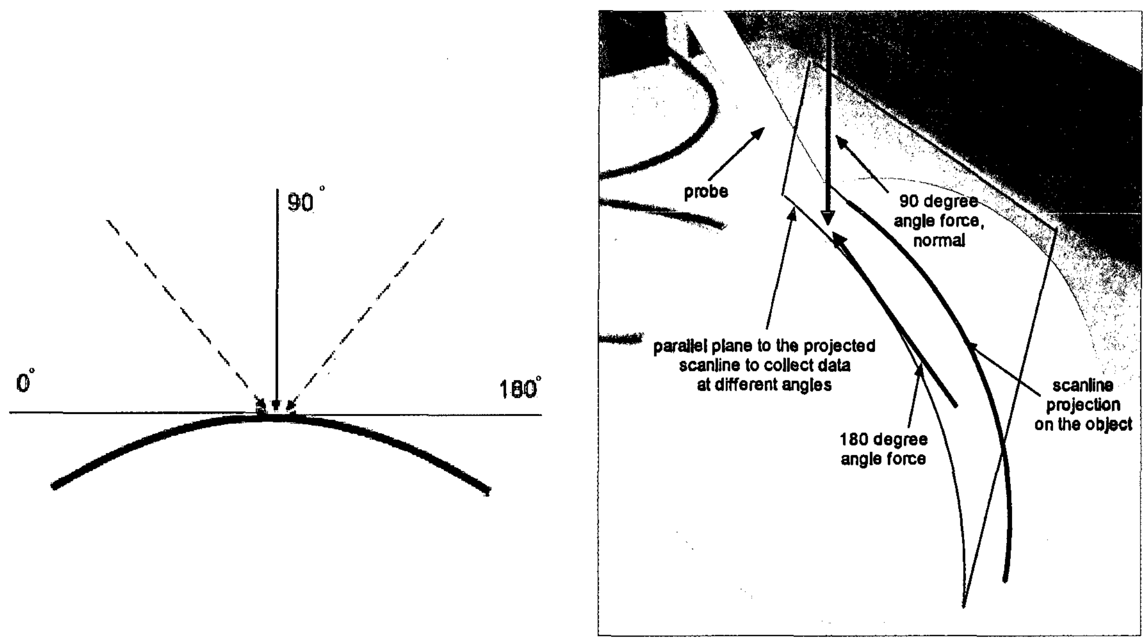


Fig. 3.18. Force measurement with different angles to the surface.

The normal force on the object's surface is considered at 90° with the tangent to the surface at the contact point. The point-wise correspondence required between the deformed and non-deformed range profiles is ensured by the fixed position of the range sensor and of the objects during data collection, therefore the parameter Y covers the same range for all measurements on an object (and material) during acquisition and is not considered as an input in the network, the Z values being uniformly indexed to the data vector.

Fig. 3.19, Fig. 3.20 and Fig. 3.21 show three examples of averaged deformation profiles resulting from the application of increasing forces along the normal direction to the surface of the object for the different materials under study.

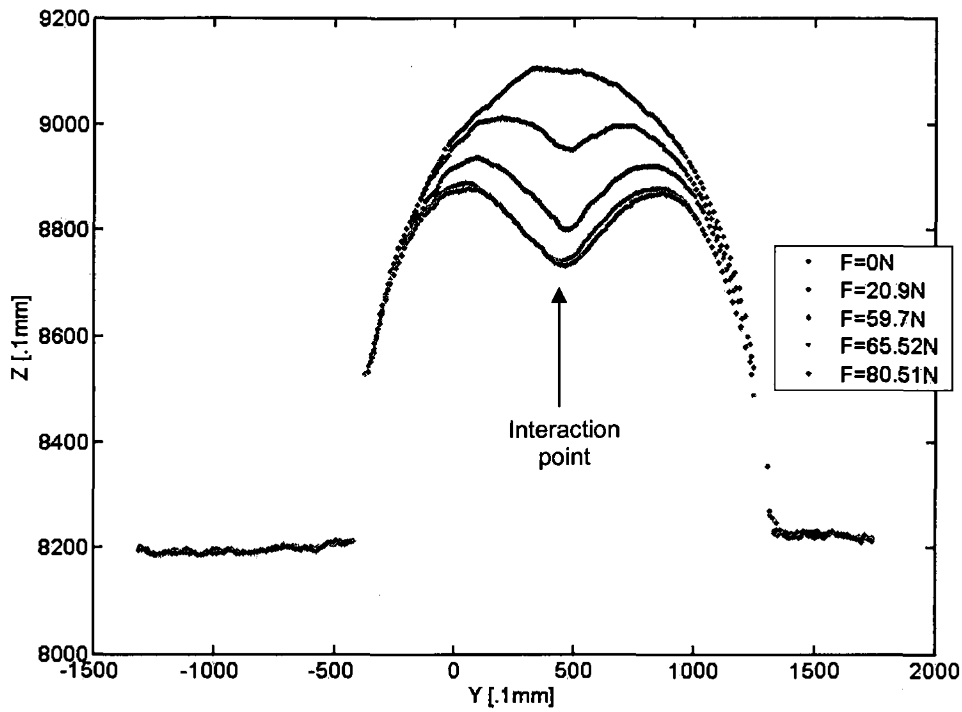


Fig. 3.19. Deformation profiles for the rubber ball under increasing normal force.

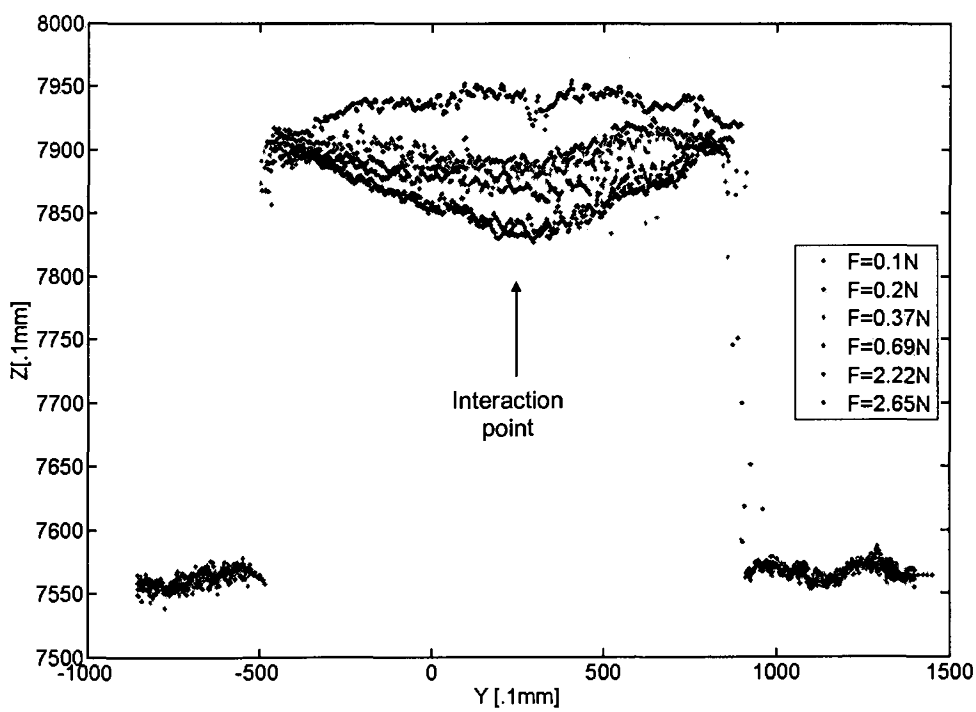


Fig. 3.20. Deformation profiles for the cardboard box under increasing normal force.

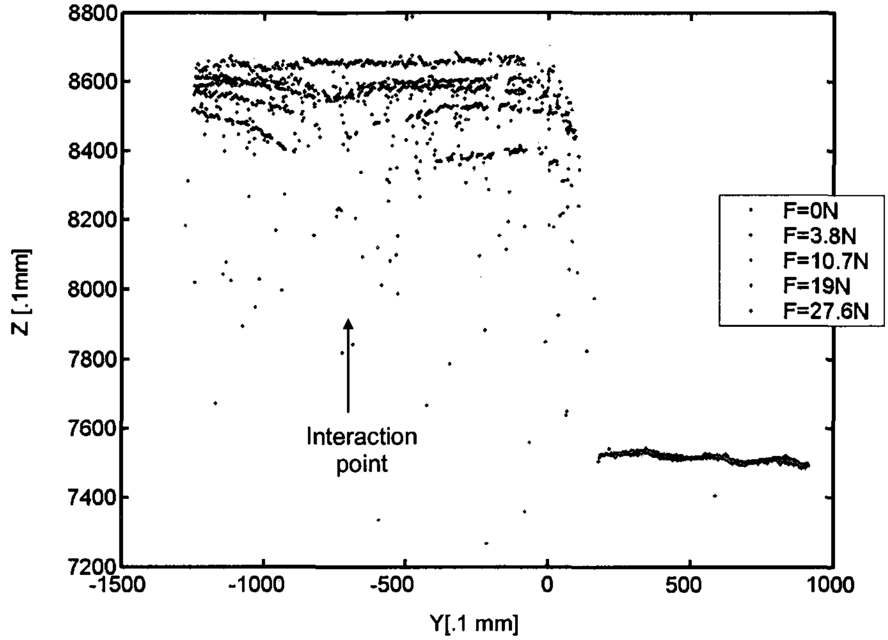


Fig. 3.21. Deformation profiles for the foam pillow under increasing normal force.

Fig. 3.22 and Fig. 3.23 show examples of averaged deformation profiles for the rubber and foam materials subject to different forces applied at different angles.

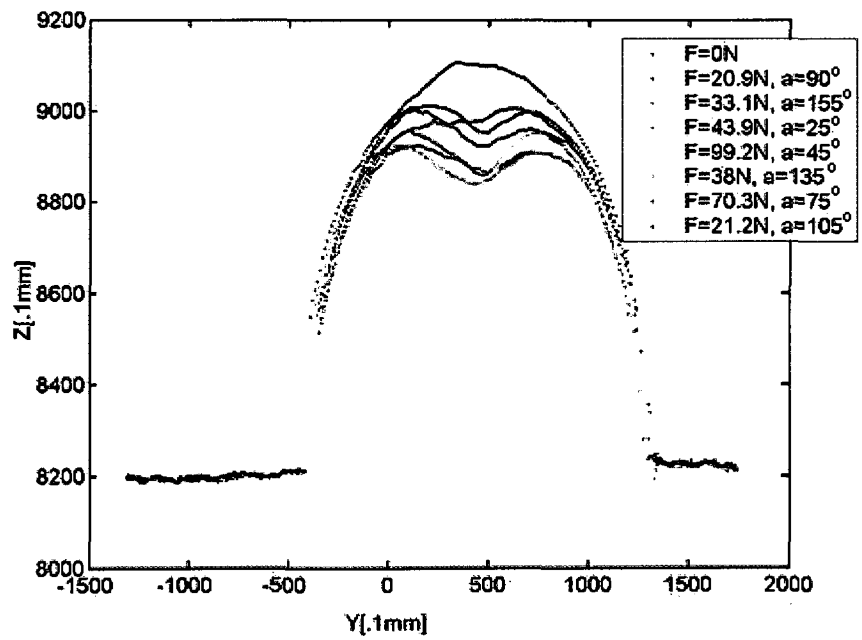


Fig. 3.22. Deformation profiles of the rubber ball under different magnitudes of force applied at different angles (measured in degrees).

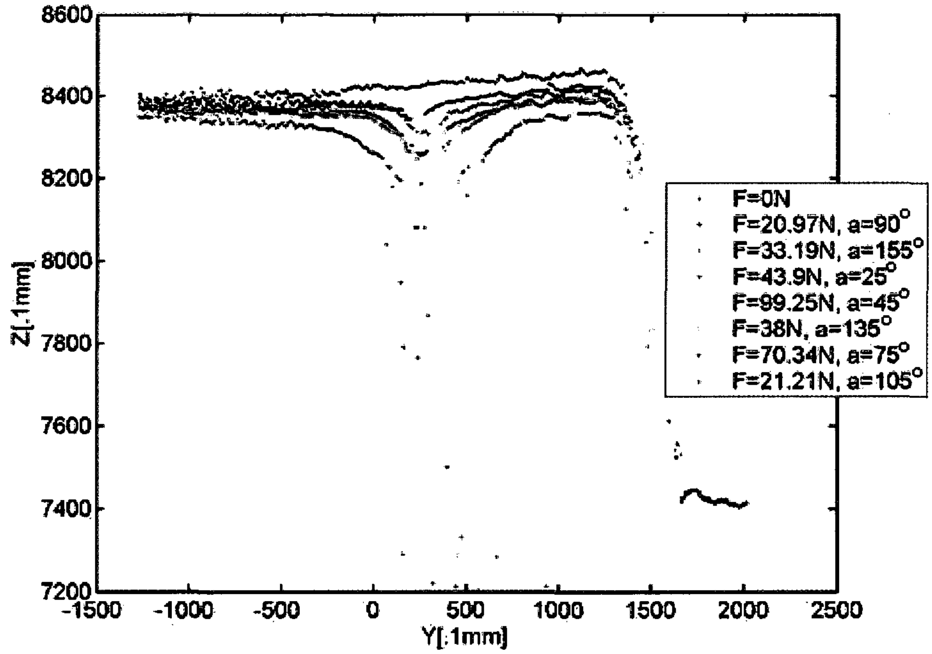


Fig. 3.23. Deformation profiles of the foam pillow under different magnitudes of force applied at different angles (measured in degrees).

The deformation profiles collected for angles under 90° present a deeper cavity around the probe's tip towards the right side, while the ones for angles larger than 90° have the cavity deeper towards the left side. As expected, angles closer to the normal to the surface usually lead to deeper deformation profiles, while tangential angles to the surface result in flatter profiles.

The quality of collected range data is directly related to the material under study. As it can be observed in Fig. 3.19 to Fig. 3.23, the range sensor data are more compact and less noisy for the rubber, which has a light color and a dense, smooth texture. The dataset is relatively compact for the cardboard, while measurements get scattered on a wider range of values and exhibit a larger number of gaps in the case of the foam. This effect is mainly due to the rough, dark texture of the fabric that covers the piece of foam. As the laser range finder operates on the principle of active triangulation, laser rays are expected to reflect on the object's surface before reaching back the sensor. When projected on a rough surface, laser rays are partially absorbed or diffused in all directions, depriving the range sensor from the expected echo or creating false reflections that introduce errors on range estimation. This phenomenon gets amplified as the external force applied on a soft object increases. The fact that the orientation of the local surface is significantly modified and is no longer perpendicular to the laser beam propagation direction leads to the formation of gaps around the probe's tip. These errors due to the measurement equipment are not systematic and cannot be removed from the data set without the risk of losing essential information. However, they are normally treated as noise by the neural network employed to learn the deformation profile, therefore do not pose significant problems for the modeling.

The deformation profiles are segmented to the regions of interest. Irrelevant information, such as the probing table on which the measurements are performed, depicted as straight lines around the value 8200 on the Z axis in Fig. 3.19 is removed, and only the part that contains the relevant deformation profile is provided in the training set.

3.3.2.2. Neural Network Architecture

A feedforward neural network with two input neurons associated with the magnitude and angle of the applied force (F, a), 45 hidden neurons (H_1-H_{45}) and an output vector with the same size as the length of the deformation profile (Z), as shown in Fig. 3.24, is employed to learn the relation between forces and angles and the corresponding deformation profiles provided by the range finder.

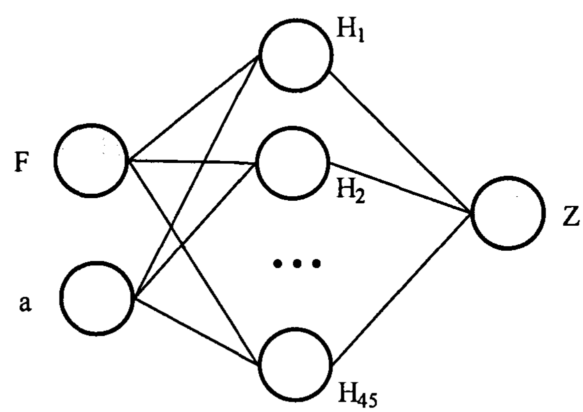


Fig. 3.24. Feedforward neural network to learn elastic behavior from deformation profiles under various force magnitudes, applied at various angles (F=force, a=angle, Z=deformation profile).

The length of the vector Z can be different for different objects, as the profiles are segmented to relevant deformation profiles only, as explained above. In this modeling case, forces of different magnitudes applied at different angles from the tangent to the surface are added as supplementary inputs in the network. This improves the modeled behavior especially for objects exhibiting highly nonlinear characteristics.

Even though the data is noisy and contains important gaps for large forces, the networks are trained directly with the averaged range profiles, to avoid losing essential information. The only data pre-processing applied is a normalization of the depth data contained in the deformation profiles to the [0 1] interval prior to training, as required by the neural network implementation.

From the several tests performed, a number of 45 hidden neurons gave a good compromise between the accuracy of modeling and the length of training, as depicted in Fig. 3.25 to Fig. 3.27 for the case of the rubber ball for normal profiles only. Excellent convergence is obtained in all cases, as shown by the very low average errors on all Z vectors of order 10^{-9} for training and 10^{-5} for generalization in Fig. 3.25 and Fig. 3.26 respectively. The training error (the error between the samples in the training set and the results provided for this set by the neural network) that reflects the memorization capability is relatively constant for any number of neurons larger than 40, as it can be seen in Fig. 3.25.

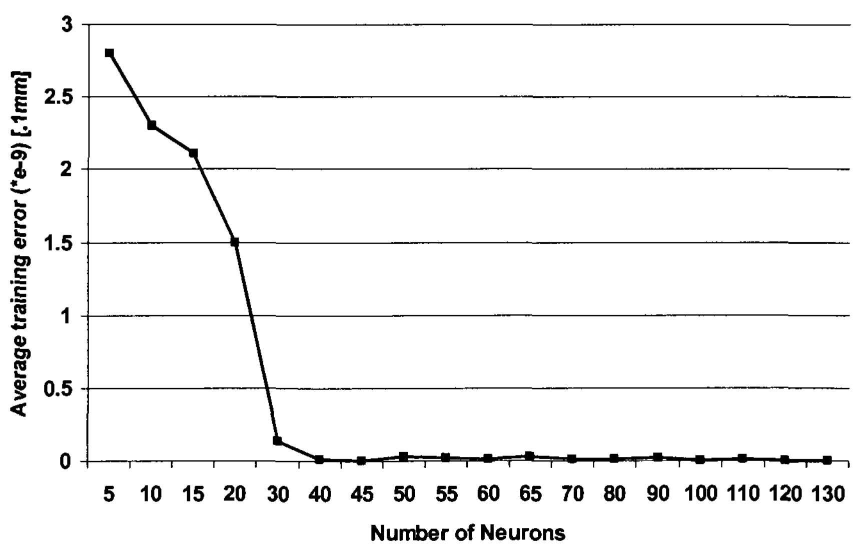


Fig. 3.25. Evolution of training error with the number of neurons.

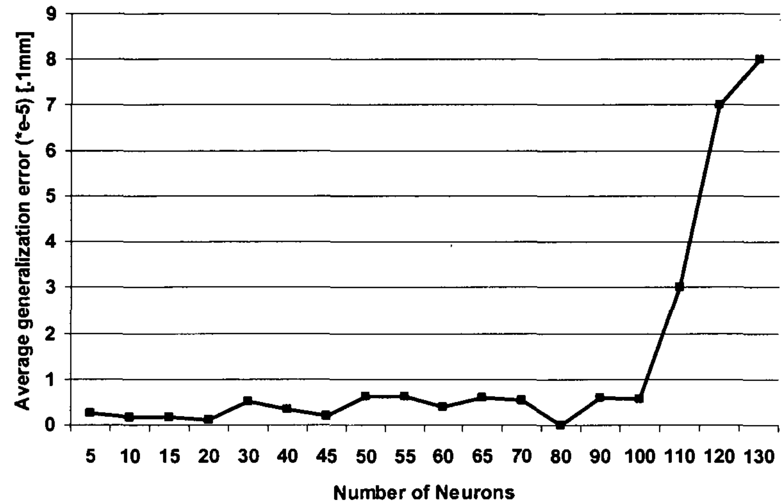


Fig.3.26. Evolution of generalization error with the number of neurons.

Fig. 3.26 shows the testing error (the error between the samples in the testing set and the results provided for this set by the neural network), which reflects the generalization capability of the network to provide estimates for data that was not part of the training set. In this figure, it can be seen that the number of neurons to keep the error relatively low is situated between 40 and 100 neurons, with the first low value at 45 neurons and the absolute minimum at 80 neurons. More than 100 neurons in hidden layer lead to a significant growth in this error. The training time is increasing with the number of neurons, as shown in Fig. 3.27.

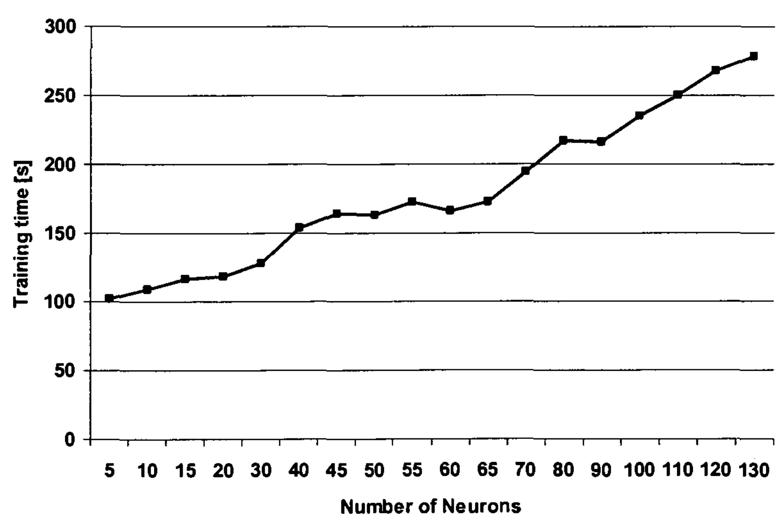


Fig. 3.27. Evolution of training time with the number of neurons.

It can be seen that 45 neurons is a good compromise between the accuracy of modeling results, the generalization capability and the training time. If the user is willing to pay the additional training time, a choice of 80 neurons in the hidden layer gives the most accurate results in this modeling case. As it was mentioned before, the knowledge of training and generalization errors can be employed to automatically tune the neural network system to a certain data set based on the desired level of accuracy and the tolerable training time for a certain modeling case

For the training, the same algorithms were used as for the modeling of force-displacement curves: Levenberg-Marquart, quasi-Newton BFGS and the resilient propagation. For this modeling case, neither Levenberg-Marquart, nor BFGS quasi-Newton method succeeded to capture correctly the characteristics of the deformation profiles. Due to the heavy memory loading implied, they were not able to model the entire deformation profile. Even though it was possible to model a compressed version of the deformation profile, obtained by the replacement of each 3 sampling points by their average value, this solution revealed to be insufficient to capture fine details in the profiles. The resilient propagation method gives better results in significantly less time. Fig. 3.25 to Fig. 3.27 above show the errors and training time for the resilient propagation algorithm.

Once trained, the network takes as inputs the force magnitude, F, and its corresponding angle of application, a, with respect to the tangent to the surface and outputs the corresponding deformation profile as an indexed vector of depth values, Z, along the scan-line.

3.3.2.3. Experimental Results

The force magnitudes and corresponding angles are provided as inputs to the network, and a part of the normalized segmented deformation profiles are used as training samples and another part as testing samples. The networks associated with each material are trained for 20000 epochs using the resilient propagation algorithm [140] implemented in Matlab code, with the learning rate set to 0.1. The training takes approximately 10 min. on a Pentium IV 1.3GHz machine with 512MB memory. For the rubber, the sum-squared error reached during training is 6.9×10^{-4}, for the cardboard is 3.5×10^{-3} while for the dark foam is 2.2×10^{-3}. As expected, the error is lower for the rubber where data is more compact and less noisy, while it remains slightly higher for the cardboard and even higher for the dark foam. But in all cases, excellent convergence is achieved.

The accuracy of the model is validated by the results obtained while testing the networks for all three materials. First, the networks are tested with samples from the training dataset. Fig. 3.28 shows the deformation characteristics for the rubber ball under three different force magnitudes applied at normal angle on the surface of the object, while Fig. 3.29 shows a sample of the real and modeled segmented deformation profiles obtained for the same forces applied at different angles on the same object. The real data is depicted by points and the modeled data, as estimated by the neural network for the same inputs, by circles.

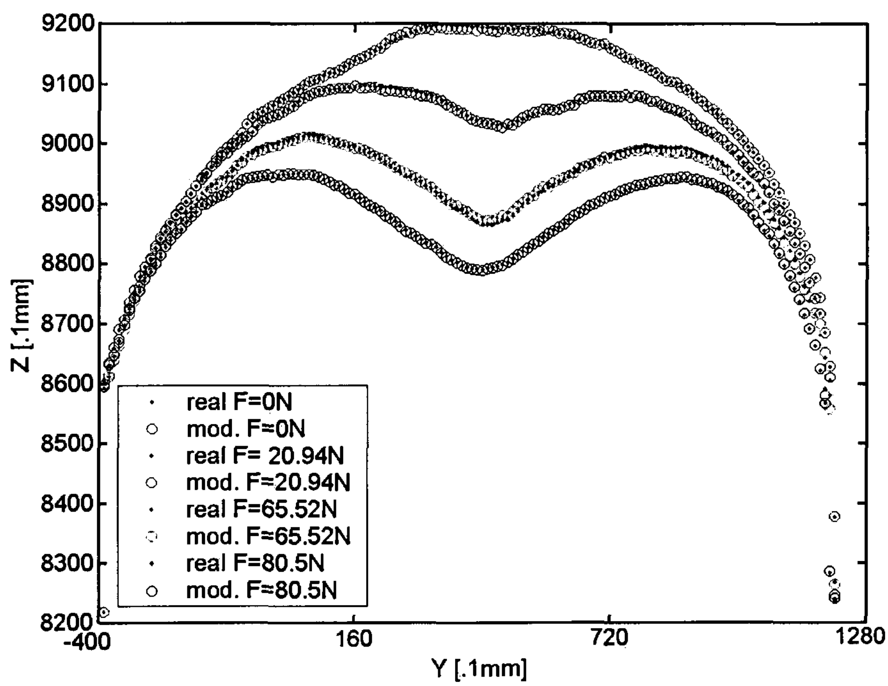
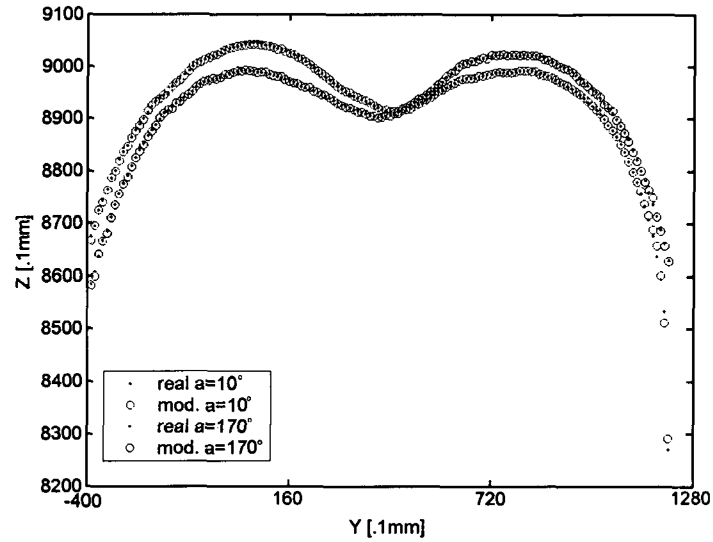
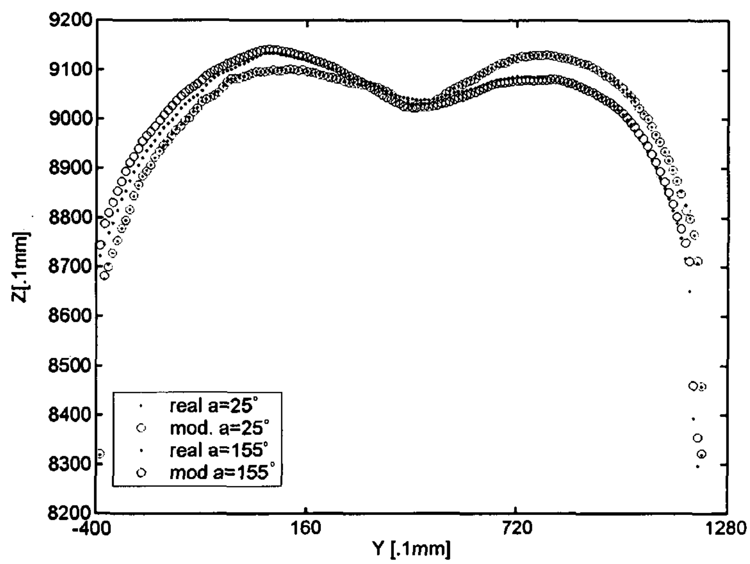


Fig. 3.28. Real and modeled deformation for the rubber ball under normal forces.

The modeled data is very close to the real one, and as expected, the modeled deformation profiles for mirror angles (10° and 170° and 25° and 155°) and for about the same force magnitude are roughly mirror profiles as well, as shown in Fig. 3.29.



a)



b)

Fig. 3.29. Real and modeled deformation for the rubber ball under forces applied at different angles: a) F=65N, a=10°, and F=65N, a=170° and b) F=36N, a=25°, and F=36N, a=155°.

Fig. 3.30 presents the real and modeled segmented deformation profiles for the cardboard material, submitted to the subset of force magnitudes shown in Fig. 3.20. The real data is shown with blue dots and the modeled profiles with red dots.

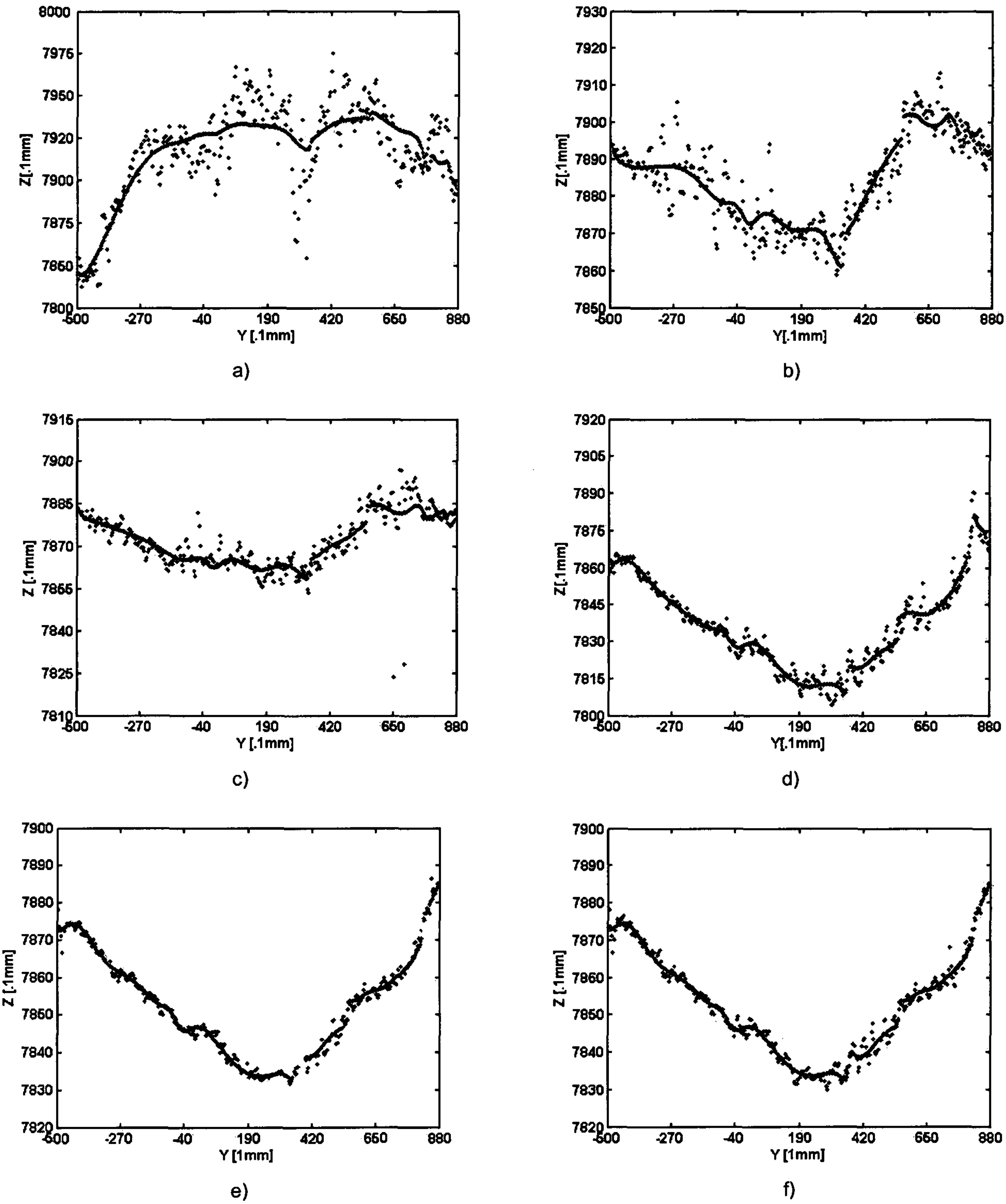
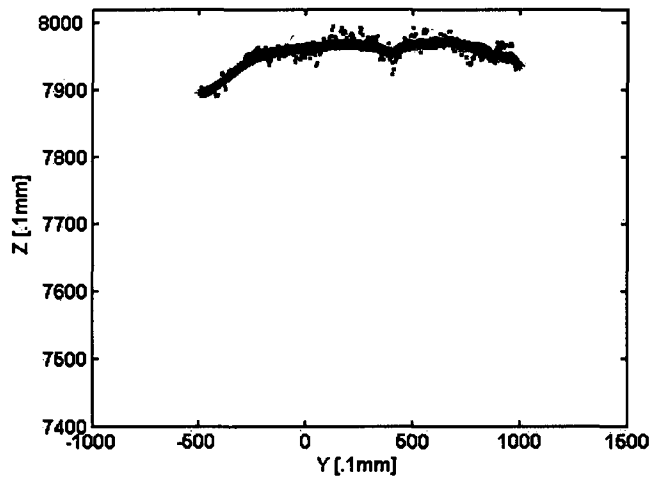
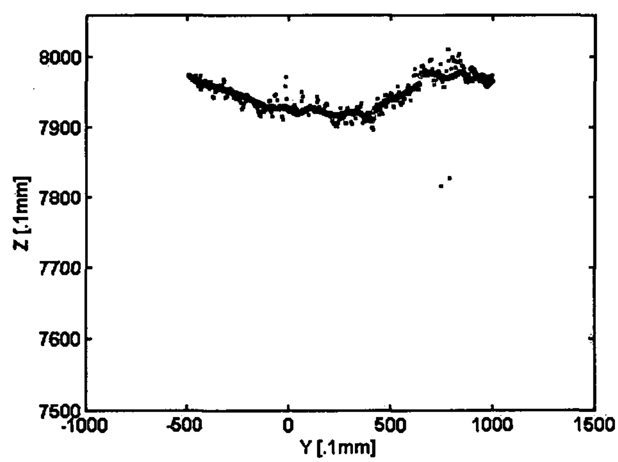


Fig. 3.30. Real and modeled deformation profiles for cardboard, under normal forces of: a) F=0.1N, b) F=0.9N, c) F=2.9N, d) F=8N, e) F=44.6N, and f) F=46.7N.

The data is noisier, but the network is still able to model the main shape of the deformation profile. Due to the scaling effect, the errors look large in Fig. 3.30, therefore to better show the modeling results, Fig. 3.31 depicts the deformation profiles to the scale used in the Fig. 3.19, for the profiles in Fig. 3.30a and Fig. 3.30c. Since there is no significant change in the deformation profile while probing the box at different angles, further results are not presented here.



a)



b)

Fig. 3.31. Scaled real and modeled deformation profiles for cardboard, under normal forces: a) F=0.1N, b) F=2.9N.

Fig. 3.32 shows real and modeled segmented deformation profiles obtained for the application of a normal force on the foam pillow, for the subset of forces in Fig. 3.21.

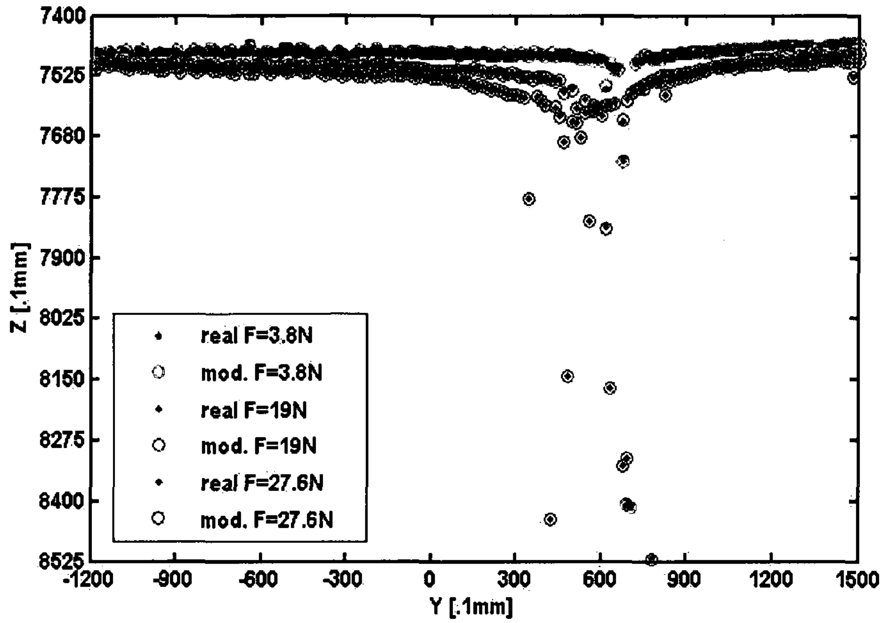


Fig. 3.32. Real and modeled profiles for the foam pillow under normal forces.

The network catches some of the noise in the curves, but models quite accurately the deformation profiles. This noise in the profiles appears due the interaction of the laser scan-line with the tip of the probing rod tip in certain measurement scenarios and it is not systematic. Since no assumption is made on the shape of the object under probing, any filtering procedure would imply the potential loss of important features, such as the peak (largest depth) of the deformation profile for example. Neural networks are fault-tolerant and therefore the noise in the resulting models is seldom evident and disturbing. Moreover, the measurement in the figure above refers to a single probing point. As it will be further shown in this work, a single probing point is not enough to characterize the elastic behavior of objects and several measurements will be taken at different points, therefore alleviating the effect of the noise in one measurement.

Fig. 3.33 shows the corresponding characteristics for foam submitted to different forces applied at different angles. Even if it is less obvious in this case due to the noise in the measurements, the mirror angles still result in mirrored profiles.

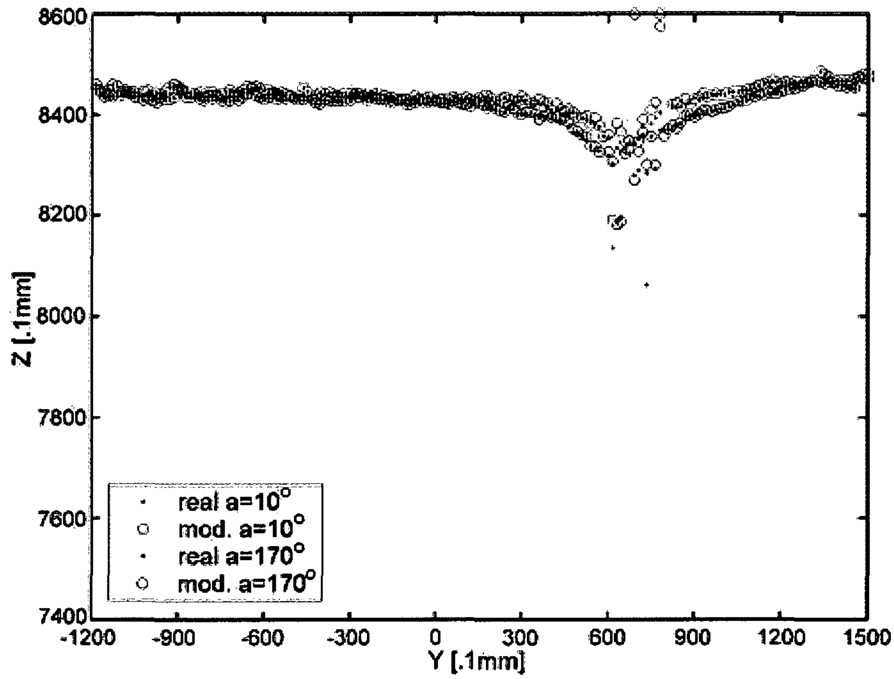


Fig. 3.33. Real and modeled deformation profiles for a foam pillow under forces applied at different angles: F=16N, a=10°, F=38N, a=170°.

All these experiments demonstrate the high potential of neural networks in extracting the main features of highly nonlinear datasets, while being fault-tolerant and quite insensitive to noise inherently present in any real data. Sparse errors in the deformation profiles are successfully eliminated by the network. But the neural model remains fully capable of capturing the peak of the deformation profile, even in areas where the profile does not contain numerous sampling points.

Moreover, our experimentation allowed the study of the generalization ability of the neural network modeling scheme. For this purpose, the network is tested for different forces and angles that were not part of the training set. Some results for the rubber ball are depicted in Fig. 3.34 and Fig. 3.35. Fig. 3.34 shows the entire set as well as a detail of the real, modeled and estimated deformation profiles for increasing forces applied at a constant angle (a=75°) on the rubber ball. The points denote the training data, the circles the modeled data, which consists of the estimation of the deformation when the network is presented with the training data, while stars represent a deformation profile estimated by the network for a magnitude of force that was not included in the training set. The network correctly interpolates the estimated profiles where expected. For example the estimated profile for a force of 68N is placed in between the real measured profiles of 64.4N and 70.34N, while the cavity in the profile, corresponding to the same angle used, is slightly moving to the right, as expected.

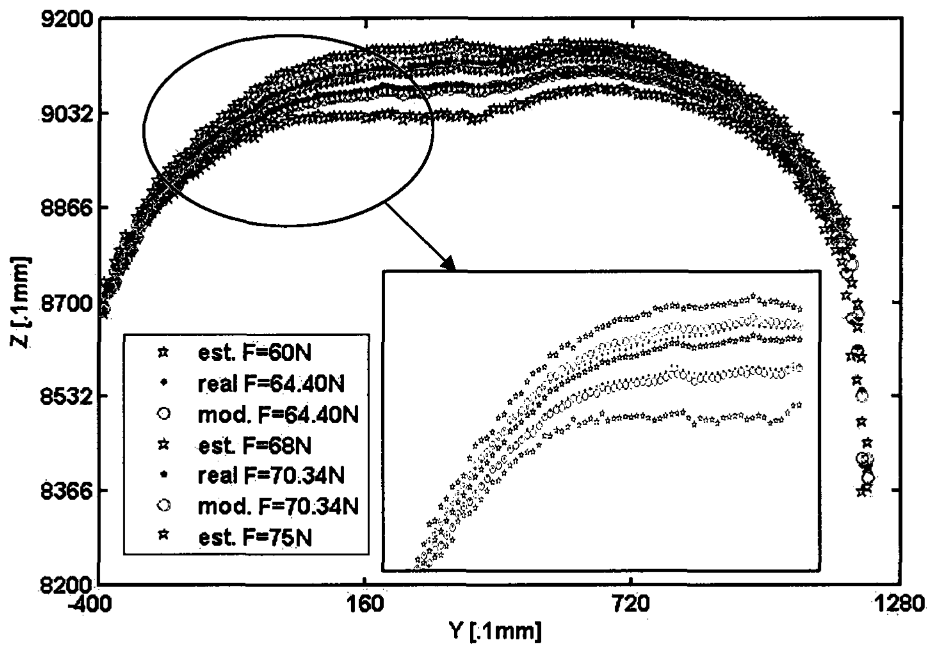


Fig. 3.34. Real, modeled and estimated deformation profiles and detail of estimated deformation for a rubber ball under increasing forces applied at a 75° angle.

Another test reveals the ability of the network to estimate deformation profiles for a constant force applied at different angles, force that was not part of the initial training set. Fig. 3.35 depicts the deformation profiles estimated for a constant force of 40N and increasing angles between 0° and 45° degrees, measured as in Fig. 3.18. None of the force-angle combination tested here was previously presented to the network during the training procedure. The position of the profiles in relation to each other and of the cavity of each profile corresponds to what one would expect it to be. As the force is the same, but the angle increases (valid for angles under 90°) the cavity becomes slightly deeper and shifted towards left. The network therefore generalizes well, being able to provide good estimates for both angles and forces that were not part of the training set.

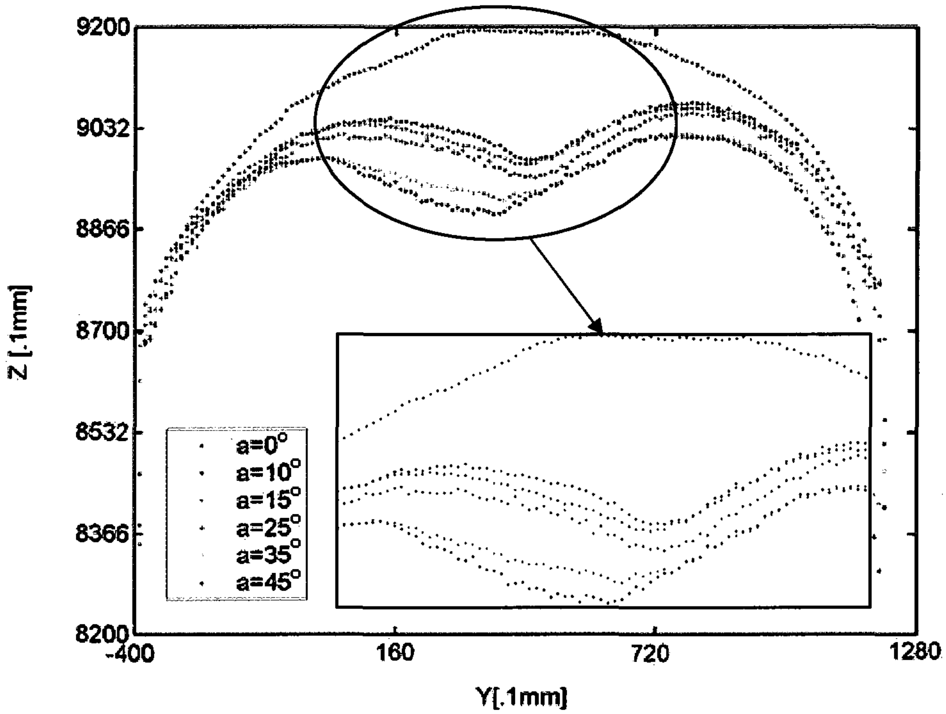


Fig. 3.35. Estimated deformation profiles and detail of estimated deformation profiles using neural network for rubber ball for a constant force F=40N applied at different angles.

3.3.3. Elastic Data Modeling for Multiple Scanning Directions

Additional experiments are performed in order to determine if an enhancement of the modeling procedure occurs by performing scans of the deformation profiles for different orientations of the object under study. Such an approach obtains a 2.5D deformation profile model for each interaction point, as it is further shown.

3.3.3.1. Data Set

Using the same experimental setup, different magnitudes of forces are applied at different angles successively on selected sampling points using the probe attached on the *ATI* force/torque sensor and an averaged range profile is collected with the laser range finder for each force magnitude. In this case measurements are also taken at different poses (*p*), encoded as angles in the table plane, as depicted in Fig. 3.36. Pose 0° is considered parallel to the orientation of the scanline.

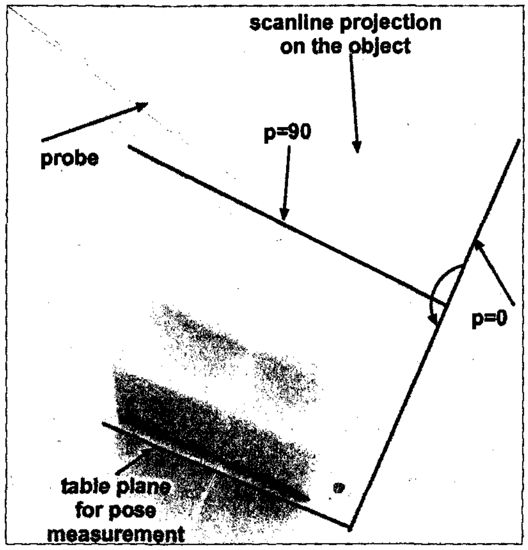


Fig. 3.36. Force measurement at different poses, denoted p.

Measurements are collected for increasing force magnitudes (F) and force angles (a) for different poses (p) using the equipment described in section 3.3.1. The study is exemplified here for the set of two foam sponges shown in Fig. 3.36. This is one of the most complex cases to be measured and modeled due to the highly nonlinear elastic behavior and non-uniform texture of foam that makes the measurement using the laser range scanner difficult, and it is presented to demonstrate the capability of the proposed method to cope with challenging modeling situations.

For a better evaluation of the modeling results, 7 control points are introduced for which measurements are collected. They are encoded in the Cartesian space (u, v, w), as O(0,0,0), A(2.5,-3.5,0), B(-2.5, -3.5, 0), C(0,-1.5, 0), D(0,1.5,0), E(2.5,3.5,0) and G(-2.5,3.5,0), depicted in Fig. 3.37.

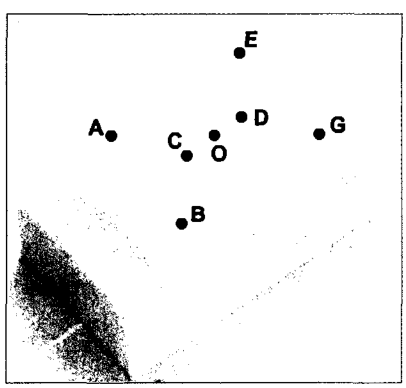


Fig. 3.37. Control points.

Fig. 3.38 shows examples of lateral deformation profiles for the set of sponges at different poses, when no force is applied to the surface.

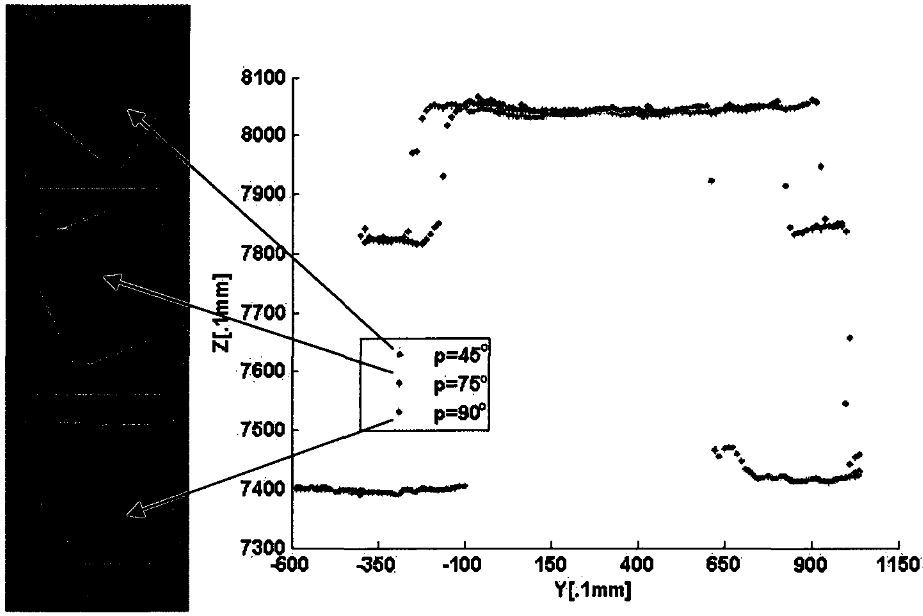


Fig. 3.38. Measured non-deformed profiles at different poses.

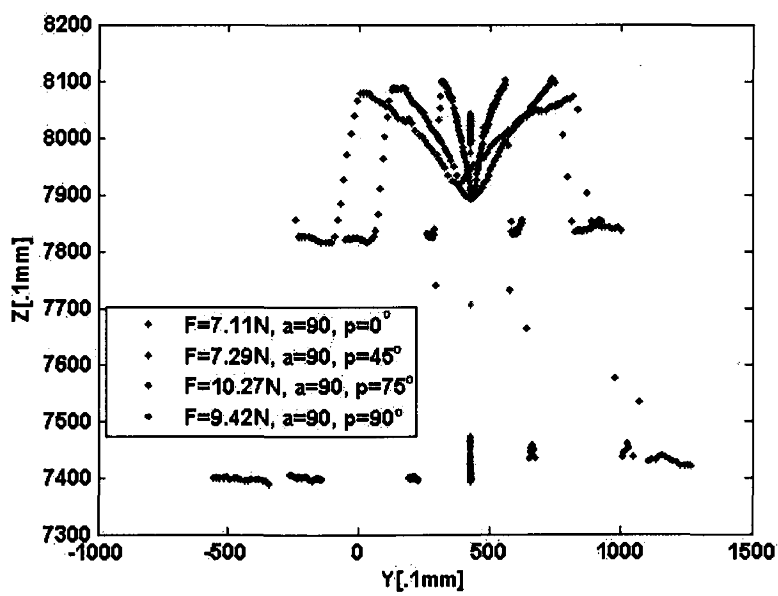


Fig. 3.39. Deformation profiles of the sponge set under different magnitudes of force applied on point O for fixed a=90° probing angle.

It can be easily seen that the profiles do not have the same length, as the sponges are rectangular. The profile depicted in Fig. 3.38 with blue at 45° (along the diagonal of the sponges) is longer than the profile in green at 90° (along the width of the sponges). In this case, the removal of the table line from the profiles (the line around the value 7400 on the Z axis) will imply the loss of significant information about the deformation behavior of the object. Therefore in order to obtain accurately modeled deformation profiles, the measured profiles will be modeled in their entirety.

Fig. 3.39 shows average deformation profiles for the set of sponges at different poses for different force magnitudes applied at normal direction on the point O, as defined in Fig. 3.37. Fig. 3.40 depicts measurements performed at a fixed pose for different force magnitudes applied at different angles on the same point O.

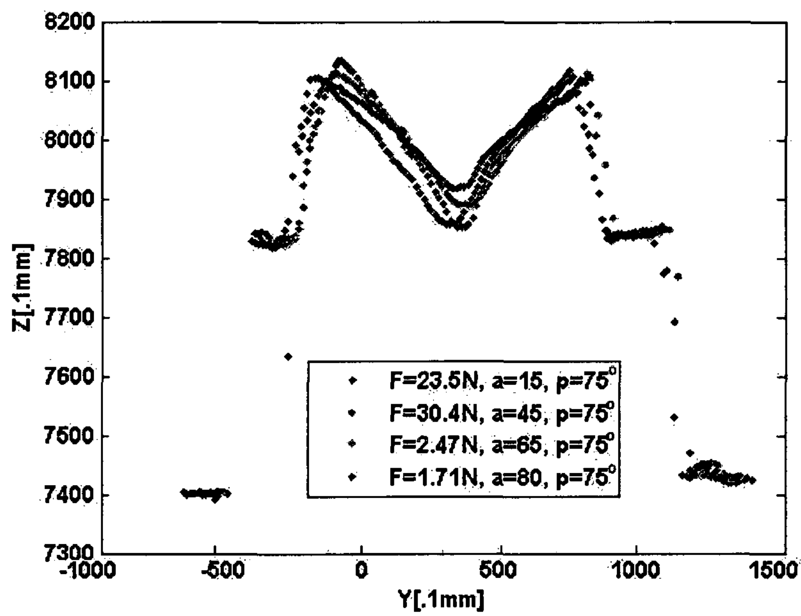


Fig. 3.40. Deformation profiles of the sponge set under different magnitudes of force applied at different angles on point O for fixed p=75° pose.

3.3.3.2. Neural Network Architecture

A feedforward neural network with six input neurons associated with the magnitude of force (F), the angle (a) of the applied force F, the pose of the object encoded as angle in the table plane (p), and the coordinates of the point (u, v, w) where the force is applied with respect to the object, 45 hidden neurons (H_1-H_{45}) and an output vector with the same size as the length of the deformation profile (Z), as shown in Fig.

3.41, is employed to learn the relation between forces and angles and the corresponding deformation profiles provided by the range finder.

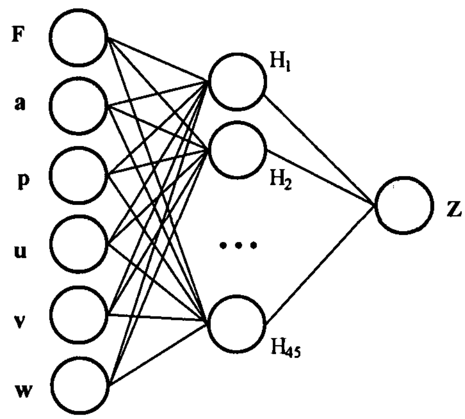


Fig. 3.41. Feedforward neural network used to learn elastic behavior from deformation profiles under various force magnitudes, applied at various angles and at various poses (F=force, a=angle of application of F, p=pose of object, (u,v,w)= coordinates of point on which the force is applied and Z=deformation profile).

The only data preprocessing applied is a normalization of the depth data contained in the deformation profiles to the [0 1] interval prior to training, as in the case of one scanning direction. Again, a number of 45 hidden neurons is a good compromise between the accuracy of modeling and the length of training. The employed procedure is similar to the one presented for one scanning direction. For the training, the resilient backpropagation algorithm is used as it was the only one that worked properly in the case of one direction of scanning. Once trained, the network takes as inputs the force magnitude, F, its corresponding angle of application, a, with respect to the tangent to the surface, the pose p of the object with respect to the table plane and the (u, v, w) coordinates of the point where the force is applied and outputs the corresponding deformation profile as an indexed vector of depth values, Z, along the scan-line.

3.3.3.3. Experimental Results

The network is trained for 20000 epochs as in the case of one scanning direction, with a learning rate of 0.1 and an error goal of 0.0001. The training takes approximately 5 min. The accuracy of the model is validated by the results obtained while testing the networks for all three materials. Fig. 3.42 depicts the real and modeled deformation profiles, using the proposed neural network architecture, obtained for the dataset presented in Fig. 3.39. The measured data is depicted with points and the neural network modeling results with circles.

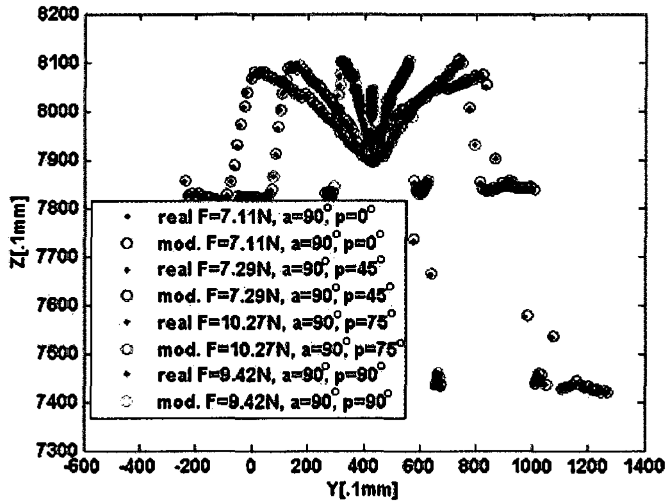


Fig. 3.42. Real and modeled deformation profiles for different magnitudes of force applied at different poses for fixed 90° probing angle at point O.

Fig. 3.43 shows the real and modeled deformation profiles obtained for the dataset shown in Fig. 3.40.

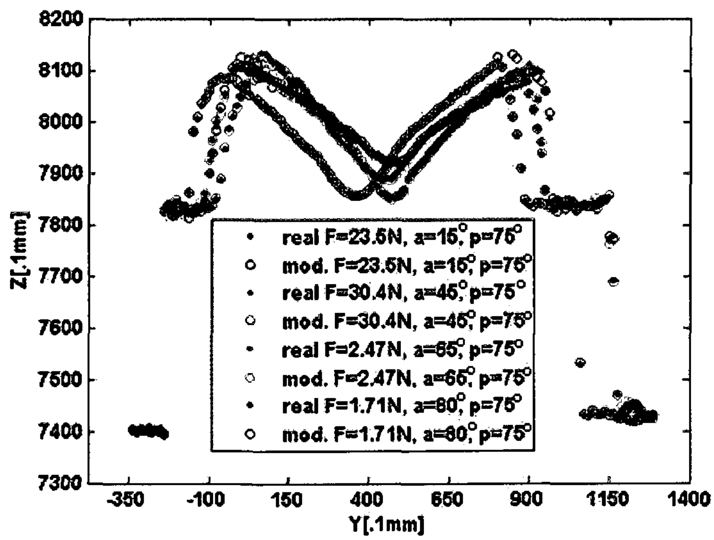


Fig. 3.43. Real and modeled deformation profiles under different magnitudes of force applied at point O, at different angles for a fixed 75° pose.

The modeling results are accurate in both cases, as the modeled data are very close to the real measured data. The resulting average mean squared training error, measured in mm, computed between the training and the modeled data over all the displacement curves in the training set, is 1.95×10^{-6}. The maximum mean square error encountered during tests is 2.44×10^{-5}. These values show that the modeling results are very close to the training data.

The network modeling results are also evaluated at different control points as defined in Fig. 3.37. Fig. 3.44 presents the results obtained for point *B* for different force magnitudes applied in the normal direction, for a fixed pose $p=0^{\circ}$.

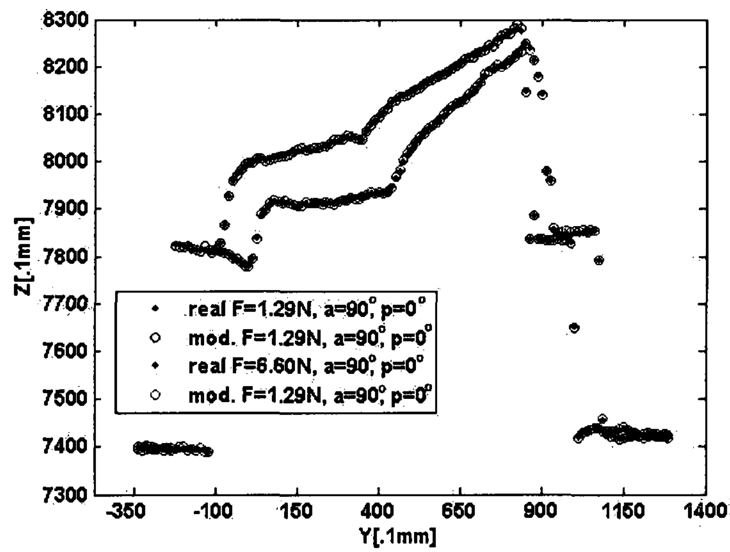


Fig. 3.44. Real and modeled deformation profiles for different force magnitudes applied in the normal direction, for fixed pose at point B.

Fig. 3.45 shows the real measurements and the modeled data using the proposed neural architecture at point *D* for different force magnitudes applied in the normal direction, for a fixed pose $p=0^{\circ}$.

In both figures, the real measurements are marked with dots and the neural network modeling results are shown with circles. As it can be seen, the network learns correctly the shape of the deformation profile for these control points and follows closely the measurement data even for large nonlinear deformations in the case of the foam.

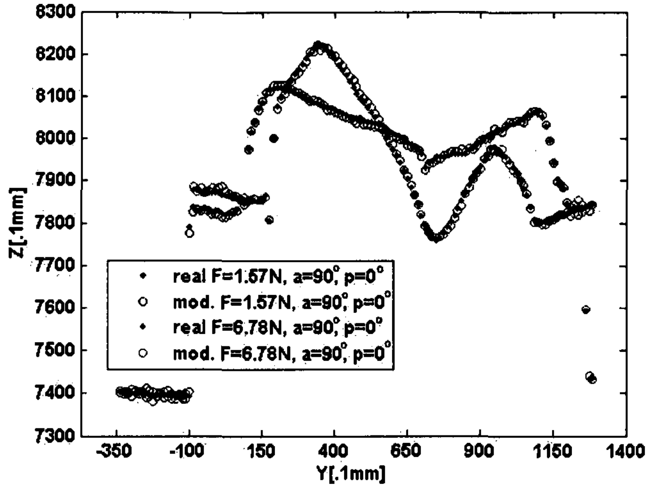


Fig. 3.45. Real and modeled deformation profiles for different force magnitudes applied in the normal direction, for fixed pose of 90° at point D.

Additional testing is performed to evaluate the generalization capabilities of the modeling scheme, by presenting to the neural network data that were not part of the training set. Fig. 3.46 shows real, modeled and estimated deformation profiles for a force of F=7.5N applied at normal direction (a=90°) on point O under different poses.

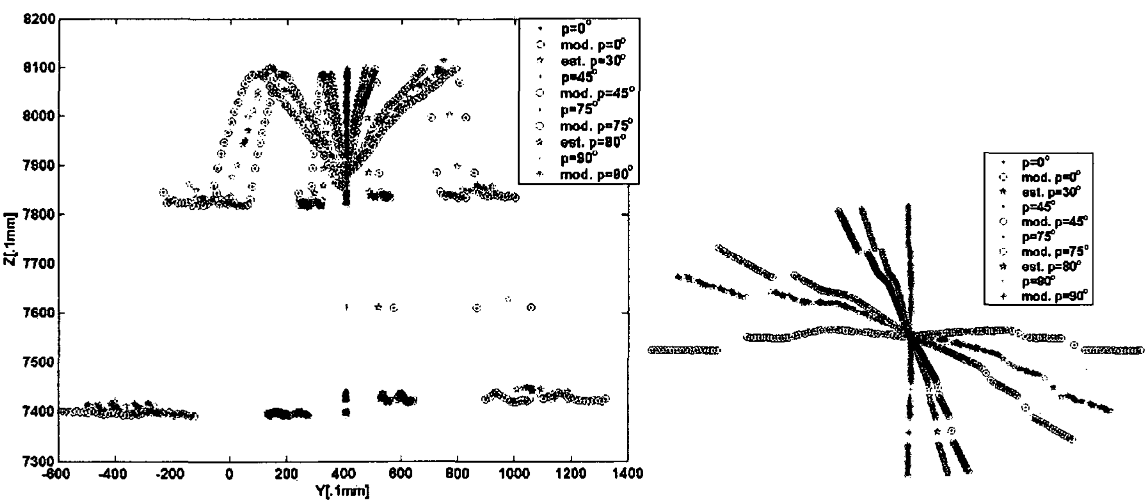


Fig. 3.46. Real, modeled and estimated deformation profiles at different poses for a force of F=7.5N, applied in the normal direction at point O.

The estimated profiles are placed as expected. For example, the estimation of the profile for the pose p=30°, which was not part of the training set, is placed in between the one for the pose p=0° and the one for pose p=45°, and closer to the deformation profile of pose 45°. The same for the estimation for pose p=80° that is placed in between 75° and 90° and closer to pose p=75°. Testing is also performed to evaluate the behavior at points there were not part of the training set. A small subset is presented here for the points depicted as P_1 to P_6 in Fig. 3.47.

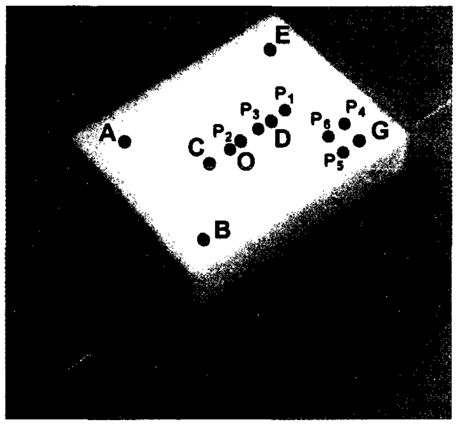


Fig. 3.47. Testing points

Fig. 3.48 shows estimated deformation profiles on the points P_1 (0,1.7,0), P_2 (0, -0.15,0) and P_3 (0,1.4,0) under a force F=6.5N, applied in the normal direction at p=0°.

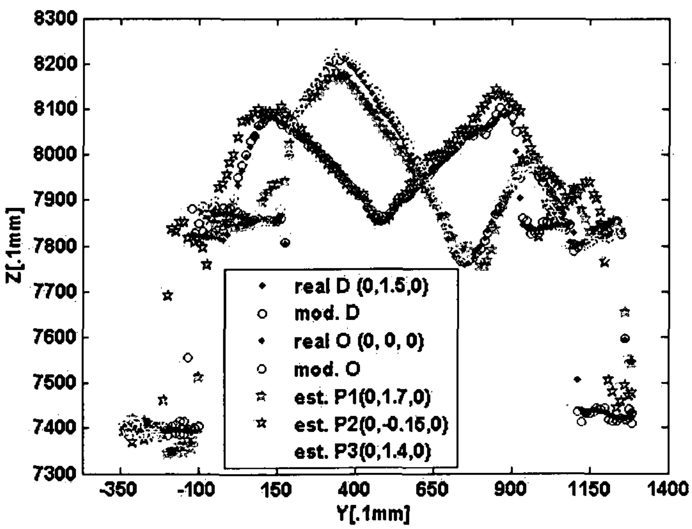


Fig. 3.48. Real, modeled and estimated deformation profiles for F=6.5N, a=90°, p=0° for probing points O(0,0,0), D(0,1.5,0), P_1 (0, 1.7, 0), P_2 (0, -0.15, 0) and P_3 (0, 1.4, 0).

To improve visualization, the real and estimated profiles at O (0, 0, 0) and D (0, 1.5, 0) are shown in red and blue respectively. It can be seen that the estimated profile for the point P_1, depicted in green, is close to the real measured profile at point D, with the lower peak slightly to the right, and the higher left peak below the one of point D as it would be expected. For the point P_3, the profile is closer to D, as P_3 is closer to D than P_1, slightly to the left and with the left peak slightly higher than the one of point D as expected. Point P_2 is closer to O and very slightly towards the left side therefore the profile is very slightly to the left and the higher left peak above the one of point O.

Results for testing of points around the down left corner of the sponge (points P_4, P_5, P_6 in Fig. 3.47), for a force F=12.78N applied in the normal direction and at a pose p=135° are shown in Fig. 3.49 and presented related to the profile for point G. Even if the profiles are a bit noisier due to a limited number of training samples, in this case again the profiles for P_4, P_5 and P_6 are placed correctly, as one would expect.

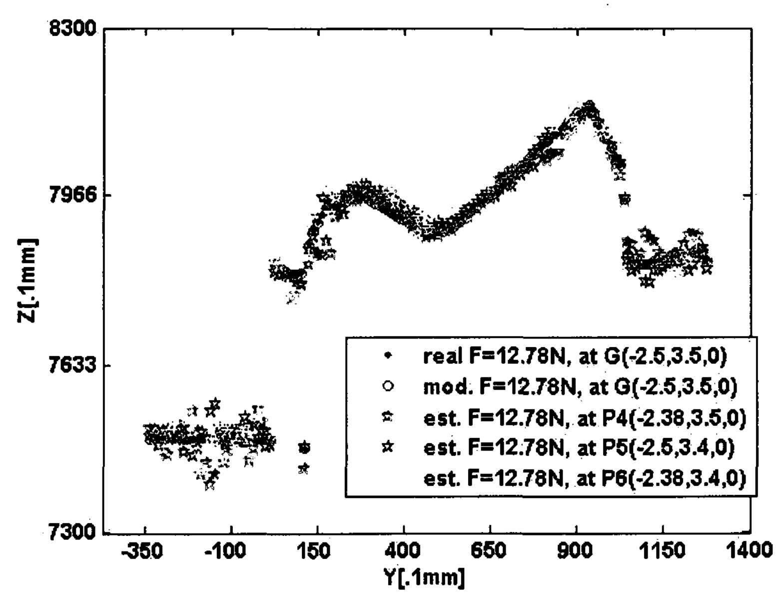
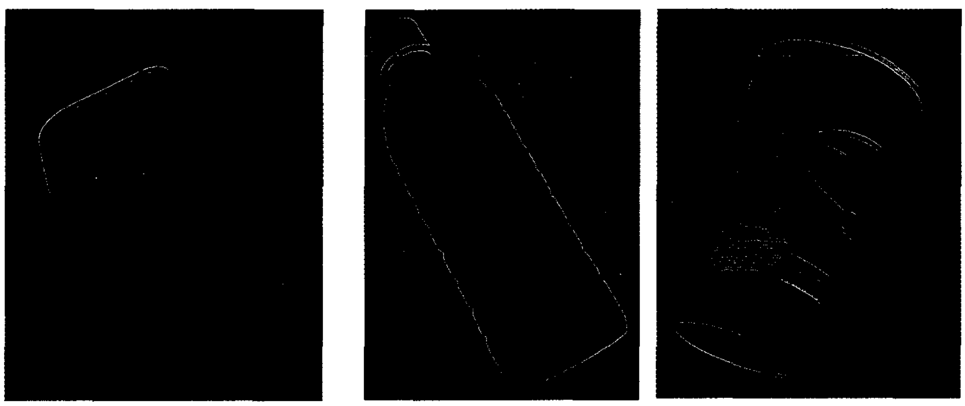


Fig. 3.49. Real, modeled and estimated deformation profiles for F=12.78N, a=90°, p=135° at point G (-2.5, 3.5, 0), P_4 (-2.38, 1.7, 0), P_5 (-2.5, 3.4, 0) and P_6 (-2.38, 3.4, 0).

All these examples show the modeling capability of neural networks, not only to capture correctly existing data, but also to provide good estimates for data that were not part of the training set.

3.4. Particular Modeling Cases

One of the main stated goals of this thesis is to propose a framework that operates from noisy experimental measurements without *a priori* knowledge on the material of the object under study. While the study up to now was restricted to elastic behavior, whether almost linear or highly nonlinear, this subchapter shows the ability of the proposed methodology to collect and model data belonging to particular cases of objects such as rigid, elasto-plastic and plastic deformable objects. A battery, a plastic bottle and a cardboard cup in Fig. 3.50 are used for experimentation.



a) b) c)

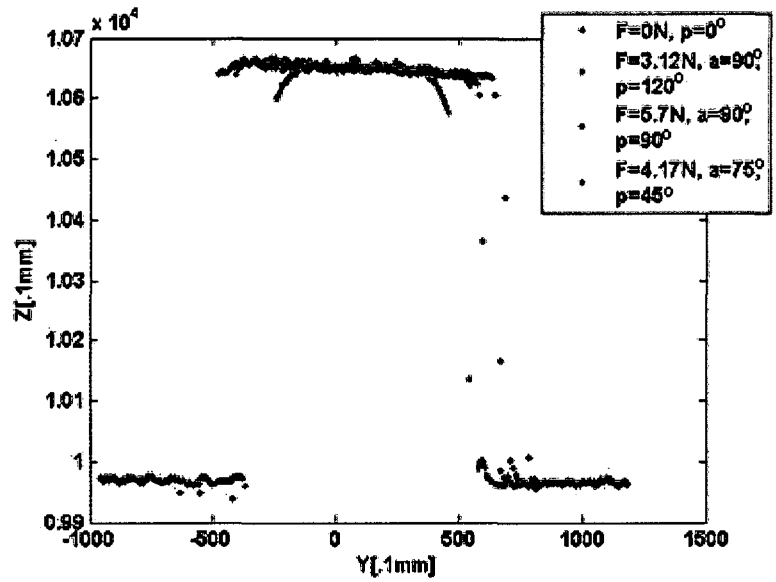
Fig. 3.50. Objects used for experimentation: a) battery, b) plastic bottle and c) cardboard cup.

3.4.1. Rigid Objects

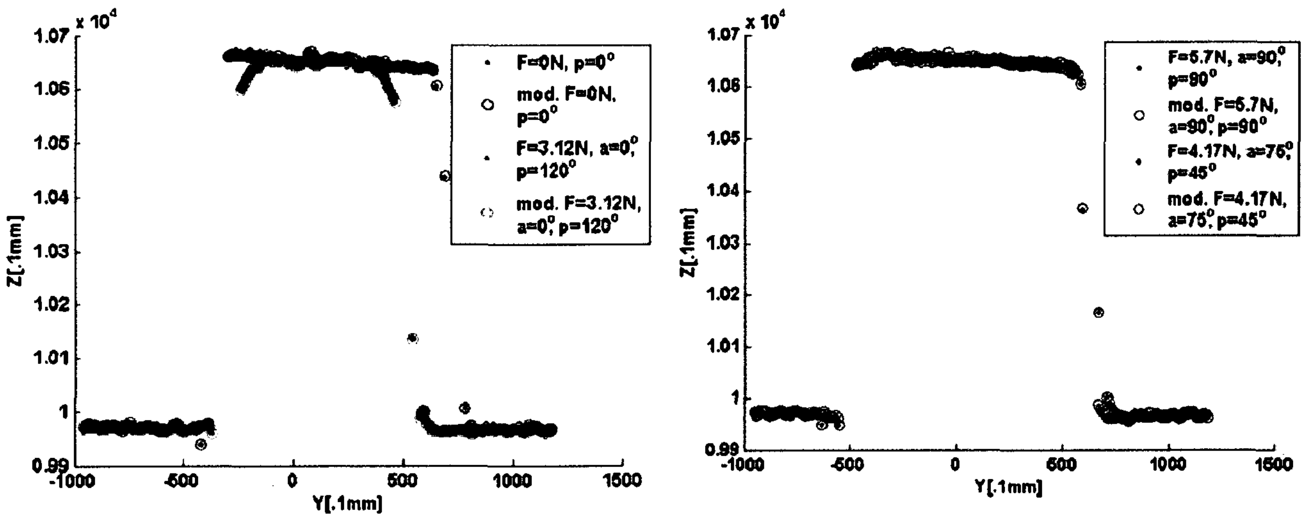
The case of rigid objects is exemplified for the battery in Fig. 3.50a whose measured deformation profiles and modeling results for different forces applied in the normal direction and for different poses are depicted in Fig. 3.51. Fig. 3.51a depicts the measurement data at point *O*. Point *O* is defined similar to Fig. 3.37 (the center of gravity of the object). As it can be observed, in this case the deformation profiles do not vary much, as it is expected for a rigid object.

Such deformation profiles are modeled with the same neural network architecture as the one depicted in Fig. 3.41. The modeling results for the measurement data in Fig.

3.51a are shown in Fig. 3.51b with the modeled data denoted by circles overlapped on the measurement data denoted by dots, at the same point *O*.



a)



b)

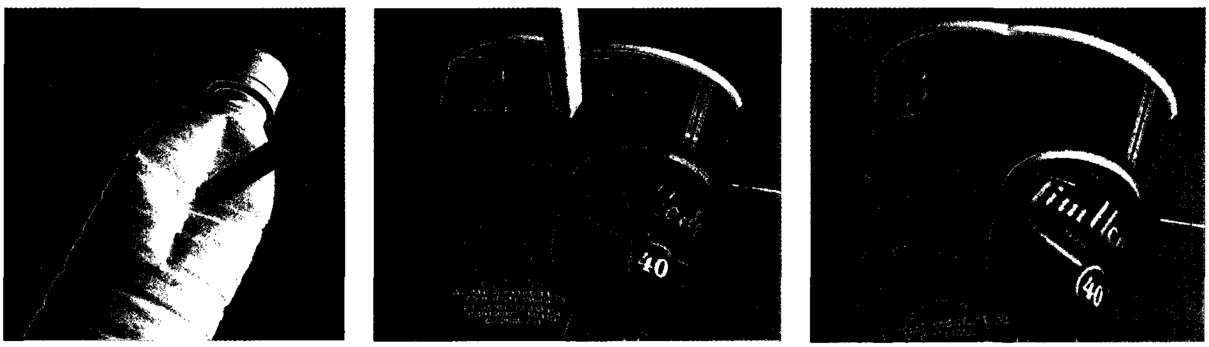
Fig. 3.51. Rigid object: (a) real data and (b) real and modeled data.

It can be seen that the fact that the object is rigid does not bring any changes in the modeling framework and the network is still able to capture correctly the behavior.

3.4.2. Plastic and Elasto-Plastic Objects

Unlike elastic objects, plastic objects remain in exactly the same shape as when they were deformed when the force is released, as it can be seen for the case of the plastic bottle in Fig. 3.52a.

Elasto-plastic or rheological objects are objects whose deformation is partially lost after the removal of the deformation force. Therefore a partial plastic deformation occurs, but the final shape of the object (Fig. 3.52c) will be different from the one in which the object was while submitted to the deformation force (Fig. 3.52b).



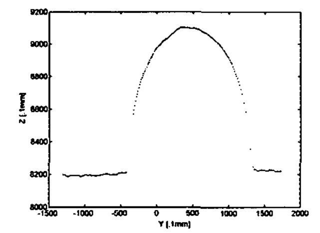
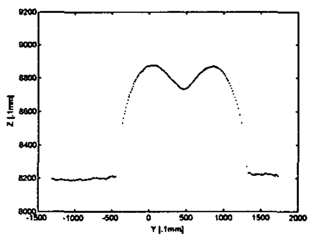
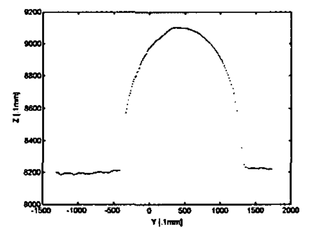
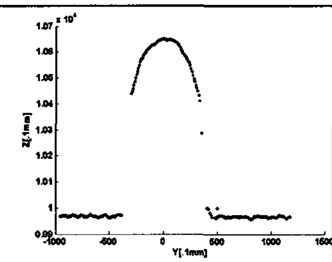
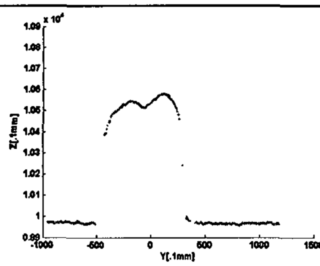
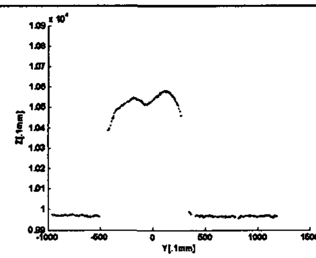
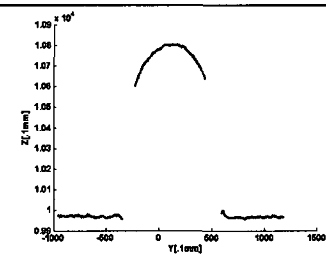
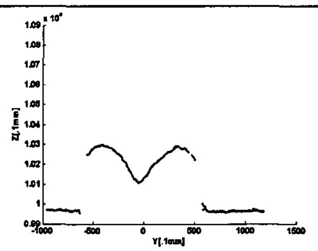
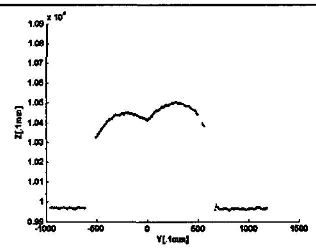
a) b) c)

Fig. 3.52. (a) Plastic deformation and elasto-plastic deformation: (b) deformation under latest force applied and (c) after latest force removed.

A simple way to detect the cases where a plastic or elasto-plastic deformation occurs is to collect a final deformation profile after all interacting forces are removed from the surface of an object. If this final deformation profile is almost identical to (a noisy variation of) the initial deformation profile (collected in the beginning of the measurement procedure, before any force is applied) it means that the object is made from elastic material or that the objects, under the force applied is in its elastic deformation stage (as defined in Fig. 2.1). If the final deformation profile is different from the initial one it means that a plastic or elasto-plastic deformation occurred.

Table 3.1 shows examples of an elastic, elasto-plastic and plastic deformation. The elastic material is exemplified by the rubber ball, the plastic material by the plastic bottle (in its plastic deformation stage) and the elasto-plastic material by the cardboard cup.

Table 3.1. Deformation profiles for elastic, plastic and elasto-plastic materials/stages.

	Initial deformation profile	***Deformation profile with the latest applied force***	***Final deformation profile after the force is removed***
Elastic material or elastic deformation stage			
Plastic material or plastic deformation stage			
Elasto-plastic material or elasto-plastic stage			

The first column of Table 3.1 shows the deformation profiles for the objects before any interaction occurs, the second column presents the deformation profiles that result from the latest interaction with the object and the third column shows the final deformation profile for each object collected after any force that was applied to it is removed. It can be seen by looking at the first row that for the elastic object the initial and final deformations are almost similar, therefore the shape of the object returns to its initial state as the latest force applied is removed. In the second row, a plastic deformation occurs, after a force F= 72.8N is applied to the plastic bottle body. In this case the initial and final deformations are different, but the shape of the object remains as when it was deformed (the deformation profile for the latest applied force and the final deformation profile are identical within noise limits). In the case of elasto-plastic materials or of objects in their elasto-plastic deformation stage shown in the third row, all the three profiles are different. The object restores its shape partially after the removal of the latest deformation force. These observations suggest that a simple way to detect if any of the plastic or elasto-plastic deformation occurs is to compare at first the initial and final deformation profiles. If these are different (more than a threshold that covers the noise in the measurements), it means that either a plastic or an elasto-plastic behavior occurred. The distinction between the plastic and elasto-plastic behaviors can be made by comparing the final deformation profile with the one resulting from the application of the latest force on the object. If they are almost identical (within noise limits), it implies that a plastic deformation occurred. If they are different, the material exhibits elasto-plastic properties or the object is within its elasto-plastic deformation stage.

To model such behaviors, an additional parameter is added to the input of the neural architecture in Fig. 3.41, parameter that refers to state of the material and is denoted by s. The elastic deformation region of any object as per Fig. 2.2 is characterized by s=0, the elasto-plastic region by s=0.5 and the plastic region by s=1. The neural architecture is shown in Fig. 3.53.

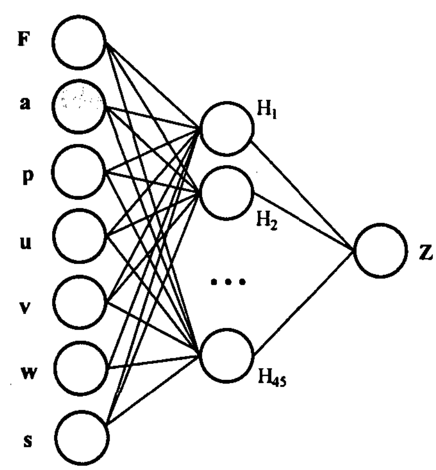


Fig. 3.53. Neural network to model elastic, elasto-plastic and plastic regions (F=force, a=angle of application of F, p=pose of object, (u,v,w)= coordinates of point on which the force is applied, s=deformation stage/material state and Z=deformation profile).

The tagging of data for the different deformation stages or different material states is done during the data collection procedure by continuously monitoring the differences between the initial profile, the final profile and the profile obtained when the latest force was applied. Results are evaluated for the bottle and the cardboard cup in Fig. 3.50b and Fig. 3.50c. Fig. 3.54 shows samples of real and modeled data for the plastic bottle.

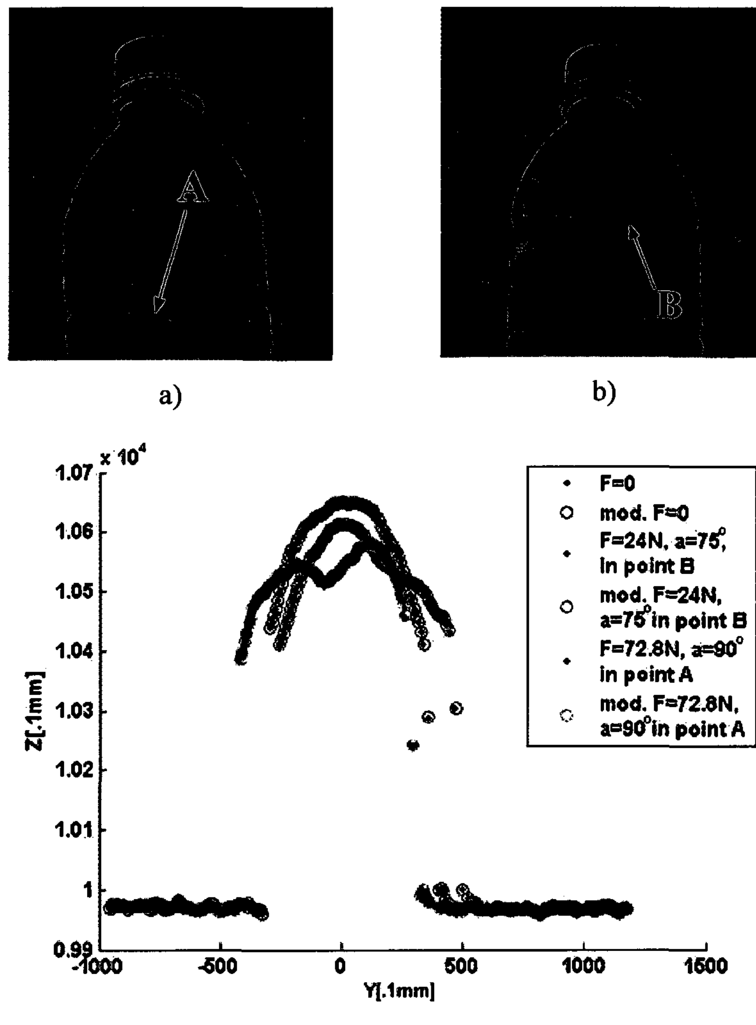
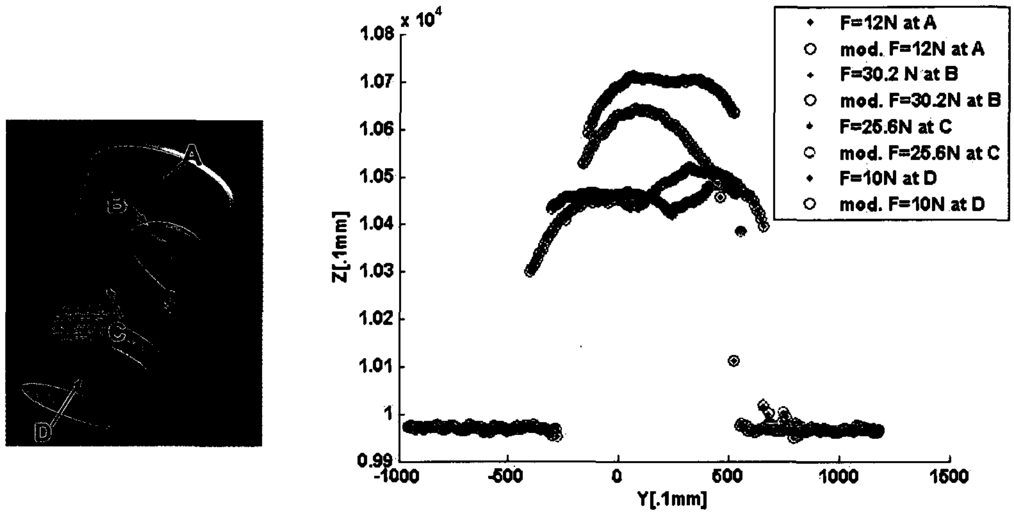


Fig. 3.54. Real and modeled data for plastic object for p=90° at different points and different angles.

The measured data of the non-deformed profile of the plastic bottle body and deformation profiles at points *A* and *B,* as depicted in Fig. 3.54a and Fig. 3.54b respectively, are shown in Fig. 3.54c with dots and the modeled data for the same profiles with circles. The mean squared error reached for training is 3.5×10^{-5}.

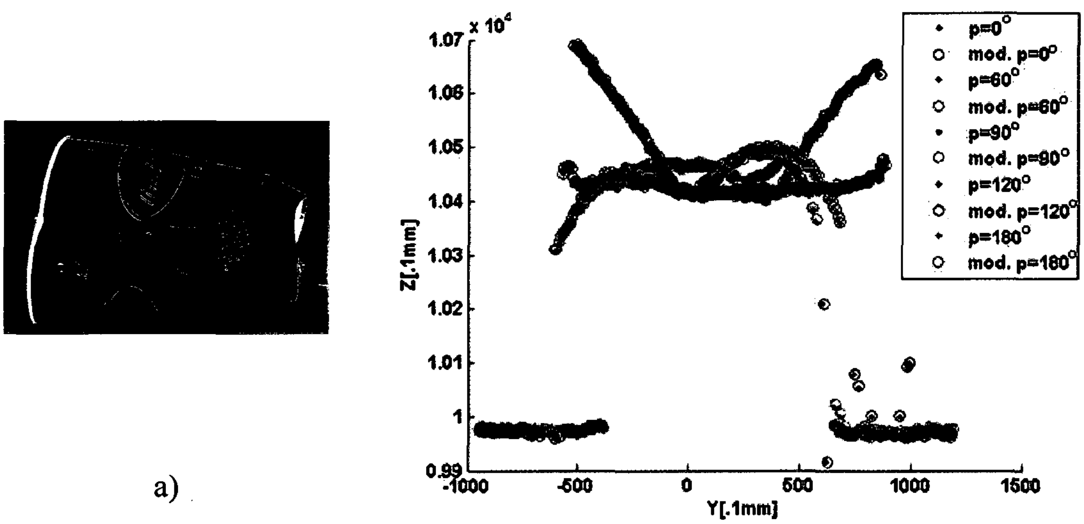
Fig. 3.55b depicts the real and modeled deformation profiles for the cardboard cup in Fig. 3.50c for a fixed pose $p=90^\circ$, at a fixed angle $a=90^\circ$ at the predefined control points *A, B, C* and *D* shown in Fig. 3.55a.



a) b)

Fig. 3.55. (a) Control points and (b) real and modeled data for elasto-plastic object for p=90 and a=90° at control points.

Fig. 3.56 shows measured and modeled deformation profiles for the same cup at different poses for a constant force of *F=25.6N,* applied at fixed angle *a=90°*.



a) b)

Fig. 3.56. Deformation profile at pose p=0° and b) real and modeled data for elasto-plastic object for F=25.6N, a=90° at point C at different poses.

It can be observed that the modeled data follow closely the measurement data in all these cases as well. The average mean squared error reached for training is 2.9×10^{-5}.

A final example is presented in Fig. 3.57, figure that shows modeling results at point *C*, at pose $p=0^{\circ}$ for increasing normal forces.

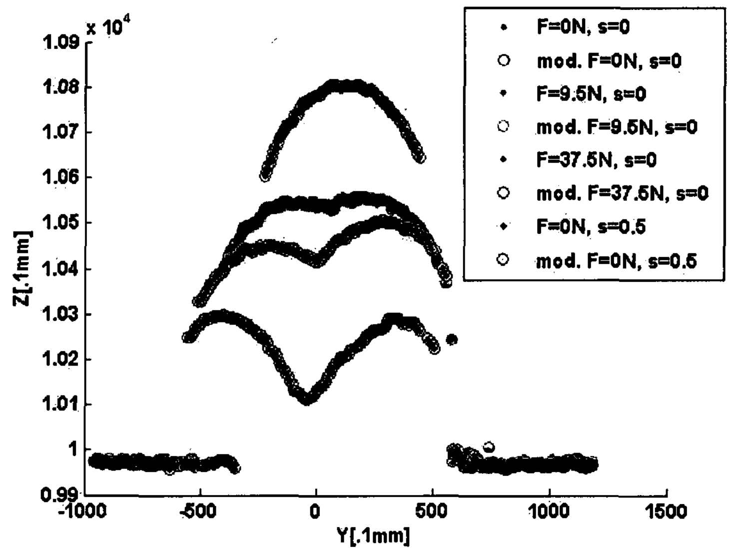


Fig. 3.57. Profile at pose p= 90°, a=90° at point C for different force magnitudes.

The red, green and blue profiles are belonging to the region where the object exhibits elastic behavior. After the application of a force F=37.5N (the deformation profile is shown in blue) the object enters the elasto-plastic region of deformation. The object surface does not return to the non-deformed state, but to an intermediate profile, shown in black between the non-deformed state (shown in red) and the state when the latest force was applied (shown in blue). For the deformation profile shown in black, s becomes 0.5. It can be seen that the network is able to capture the different deformation stages as well, fact proven by the very similar measured and modeled data profiles.

Another interesting aspect to be noticed in Fig. 3.57 is that the initial non-deformed profile, shown in red is narrower along the Y axis than any of the deformed profiles. This phenomenon occurs because of a volume expansion of the object under probing, due to the applied forces, expansion that is well captured in the modeled results. Therefore the proposed method of deformation measurement and modeling using neural networks captures naturally volume expansions which are a challenge to be modeled using other methods (e.g. mass-spring models, FEM, BEM, etc) as it was shown in chapter 2.

3.5. Conclusions

This investigation into means of modeling elastic behavior demonstrates that the benefit of using neural networks to model deformable objects is multi-folded. First, neural networks provide continuous output behavior, thus being able to provide the necessary nonlinear interpolation for estimates of data that was not part of the training set. When compared with most of the work found in the literature where *a priori* knowledge about the characteristics of the material is assumed available, this research work proposes a robust approach for modeling force/deformation relationships from realistic experimental data with noisy and incomplete measurements, the latter being exemplified here by missing values for some of the points along the scan line.

Second, the use of a neural network modeling scheme avoids the complicated and frequently impossible to solve problem of recuperating explicit elastic parameters, especially for highly nonlinear elastic materials.

Third, neural networks provide an accurate and fast response once they are trained. After the elastic behavior of an object is stored in a neural network, the latter can provide real-time information about the elasticity in any point of an object.

Finally, the neural network modeling of deformation profiles captures naturally the modification of the objects' volume due to applied forces.

Further research on this topic would benefit from automated selective procedures to determine regions of interest and perform only relevant measurements, as opposed to employing a simple random algorithm to select points to be probed as it was used in this chapter. Probing points randomly can create significant cost penalties. Even if only a lower number of points are collected than in the case of uniform sampling, as the percentage of sampled points increases, the cost gets higher to eventually reach the one of uniform sampling. The risk is also that samples randomly collected miss important features. From here stems the need of a selective sampling algorithm to determine those regions that are relevant for the geometry and/or behavior of an object, algorithm that is presented in the next chapter.

4. Selective Acquisition of Geometric and Elastic Data from 3D Point-Clouds Based on Neural Gas Networks

This chapter describes an innovative selective data acquisition algorithm that can be employed to collect both geometric data using vision sensors and elastic data using tactile sensors. It starts by describing the motivation driving the research into selective sampling procedures for data collection and justifying the suitability of a self-organizing neural network for this purpose. It then looks in depth at the neural gas self-organizing architecture, its training algorithm and testing procedure and its use in the context of selective data acquisition. It then presents experimental results for vision and tactile sampling of 3D objects embedded in point-clouds obtained with various types of scanners. It finally shows how composite-elastic models can be built, exemplifying the concepts with a complete modeling scenario.

4.1. Introduction and Context

Sensing systems are experiencing an unprecedented growth in numerous applications. Due to the high measurement speed of 3D data acquisition devices (e.g. laser scanners) on one side and to the lack of knowledge on appropriate accuracy levels for correct description of shape and geometry on the other side, the data acquisition process is elaborate and often produces too many sample points. Reducing the complexity of such datasets is one of the key techniques required in order to operate subsequent applications on the resulting data at a reasonable computational cost. To tackle this issue, the most widely exploited trend in the contemporary literature involves the post-processing of large datasets obtained by acquisition devices. Frequently, the proposed algorithms rely on the user input for providing parameters such as the desired density of sampling, the regularity of sampling, or the minimum distance between samples. This is a difficult task as the user is not always aware of the appropriate level of accuracy required for a model in order to be further processed and the adjustment of such parameters can be a lengthy trial-and-error procedure. Further research on adaptive sampling would benefit from intelligent automated selective procedures to determine regions of interest and collect only relevant measurements for modeling applications.

The objective of this work is the design of innovative approaches to achieve automatic selection of regions of interest for vision and tactile sensors to collect only relevant measurements without human guidance. The proposed algorithm is a selective sampling algorithm with the purpose of increasing the relevance and speed of probing by online progressive refinements. "Online" in this context refers to the use of the algorithm in each successive step of the measurement procedure as opposed to post-processing the data collected by a sensor. The algorithm is not adaptive in that it is not compliant - does not adjust the probing procedure as a result of the interaction with the object. It is selective, in that it picks out those features that are relevant for the object under study. An iterative use of the algorithm will result in multi-resolution object models, with higher resolution in the areas rich in features and/or with changes in elastic behavior.

The relevant regions of interest are extracted from 3D point-clouds during the acquisition procedure to prevent an avalanche of data and the related excessive processing load. Starting from an initial, fast, and sparse scan of an object, a neural gas network map is used to adaptively select areas of interest for further scanning in order to improve the accuracy of the model. The use of a self-organizing architecture is justified by its ability to quantize the given input space into clusters of points with similar properties. As it was presented in the literature review, clustering is an efficient way to compress data and retain significant information only. A similar approach is appropriate in vision to collect 3D geometry data and in tactile sensing as well. The final model obtained by augmenting the initial sparse model with the regions of interest is a multi-resolution model with a higher resolution in areas rich in features and/or in areas with changes in elastic behavior. The summary of the approach is depicted in Fig. 4.1.

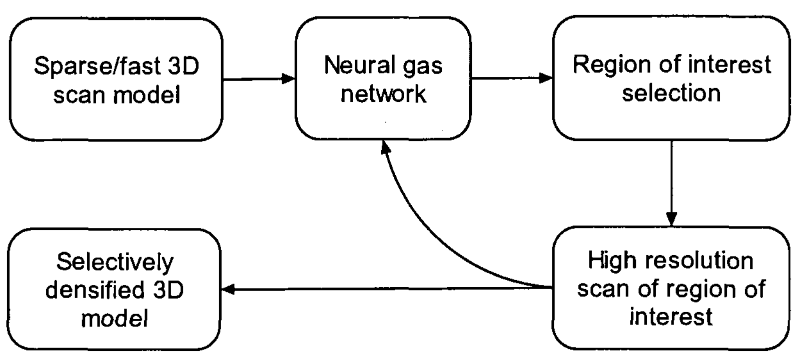


Fig. 4.1. Proposed framework for selective sampling.

The algorithm operates on a sparse, fast scan of the object under study. A neural gas map is built on this low resolution scan and the regions of interest are selected by searching areas with high density of points and/or areas of elasticity changes in this map. The identified regions are then scanned at higher resolution. Each of these regions can be iteratively mapped using neural gas models in order to identify additional features of interest, which can be scanned at even higher resolution. The final model is a selectively densified model, obtained by the addition of the high resolution scans over the initial sparse scan of the object. If the user is interested in a compressed model, a selectively densified model can be obtained as well by the addition of the high resolution scans over the neural gas map size.

4.2. Neural Gas Architecture and its Use for Selective Data Acquisition

4.2.1. Introduction and Notations

Clustering is an efficient way to compress data and retain significant information only. A data set is encoded using a finite set of reference vectors that constitute cluster centres. Data patterns are classified into clusters according to some similarity measure, which is usually the distance between the pattern and the nearest cluster centroid. The representative (winning) data vector is determined to be the closest one in distance from the input vector. There is a very close relationship existing between clustering methods and unsupervised learning neural networks. In unsupervised learning, the network adjusts itself using the input patterns only and groups similar sets of input patterns into clusters predicted upon a predetermined set of criteria relating the components of the data [139].

The most extreme competitive learning strategy is the "winner-takes all" where the activation of the neuron with the largest net input is the only one to have its weights updated. In this case, neurons which have weight vectors that are far from any input vector may never win and hence, never learn. These are like "dead neurons". Thus the new idea of soft competition appeared.

Soft competition involves updating all the nodes in the network to some degree as opposed to just the winning node. This movement of all nodes, although computationally expensive, helps to improve the performance in two ways. It helps to prevent dead units on one side, and on the other side, the change in the value of the error function is not just in the direction of the steepest gradient but may cause a jump to a new value somewhere else in the error space, thus improving the learning procedure [145]. Therefore self-organizing architectures based on soft competition are good candidates to perform clustering.

Even if these competitive self-organizing architectures are sometimes denominated with terms like "neural" and "networks", i.e. neural gas network, they are not what one usually understands as being a "neural network" [146]. Each network consists of a set S of N units:

$$S = \{c_1, c_2, \ldots, c_N\} \tag{4.1}$$

Each unit c has an associated n-dimensional reference vector that indicates its position in the input space:

$$w_c \in \Re^n \tag{4.2}$$

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

$$C \subset S \times S \tag{4.3}$$

of unweighted and symmetric neighborhood connections, used for the adaptation of the winner and the topological neighbors. The set of topological neighbors of a unit c is denoted:

$$N_c = \{i \in S \mid (c, i) \in C\} \tag{4.4}$$

The n-dimensional input signals are usually generated according to a continuous probability density function:

$$p(x), \quad x \in \Re^n \tag{4.5}$$

or chosen from a finite training data set

$$D = \{x_1, x_2, \ldots, x_M \mid x_i \in \Re^n\} \tag{4.6}$$

For a given input signal x, the winner $s(x)$ among the units in S is defined as the unit with the nearest reference vector, measured as Euclidean distance:

$$s(x) = \arg\min_{c \in S} \|x - w_c\| \tag{4.7}$$

where $\|.\|$ denotes the Euclidean vector norm. The other distant units of interest are denoted with s_i with i meaning the i-nearest unit (s_1 is the winner, s_2 the second-nearest unit, etc).

The most popular self-organizing architectures are the Kohonen self-organizing map (SOM) [150] and the neural gas network. Neural gas is selected in the context of this work after several experimental tests, due to its accuracy proven by small distortion errors and its ability to capture fine details as opposed to SOM architectures that result in larger distortions and tend to smooth fine details. The emphasis in this work is on accuracy as opposed to fastness. An extended comparison between neural gas and SOM can be found in [147].

4.2.2. Neural Gas Architecture

The main purpose of the neural gas network is to cluster multi-dimensional vectors. An initial large set of vectors is reduced to a smaller set of neurons of the same dimension, with a minimal mean distortion between each of the vectors in the initial set and its best-matching neuron. The best matching neuron is defined as the vector in the final set that is the closest to the input vector [148].

The neural gas network consists of nodes (cluster centers), which independently move over the data space while learning. It can be however seen as a feedforward unsupervised model having one n-dimensional layer of neurons and one layer of connections [148]. Each neuron is connected to the input vector of the same dimension, but there are no connections between units in the neuron layer.

4.2.3. Neural Gas Algorithm

The neural gas algorithm starts by initializing the set S of network nodes to contain N units c_i with the corresponding reference vectors $w_{c_i} \in \Re^n$ chosen randomly according to a probability density function $p(x)$ and having very small values, of the order of 10^{-5}.

Random input signals are generated according to $p(x)$ or chosen from a finite training data set. The neural gas network exploits the idea of soft competition. Eq. (4.7) is employed to identify the winning neuron that best matches the input vector x. The neurons to be adapted in the learning procedure are selected according to their rank. The rank of neurons is the rank they have in an ordered list of distances between their weights and the input vector. Each time a new input vector x is presented to the network, a neighborhood ranking indices list is built $(j_0, \ldots, j_{N-1})$, where w_{j_0} is the weight of the closest neuron to x, w_{j_1} the weight of the second-closest neuron, and w_{j_k} is the reference vector such that k vectors w_i exist with:

$$\|x - w_i\| \leq \|x - w_{j_k}\| \tag{4.8}$$

The neural gas is in general invariant to initialization. This is due to the fact that the distance in input space is only used for determining the rank order, and then the adaptations are done on the basis of this rank order [146].

In each training step, the best matching neuron $s(x)$ at time t is computed using the minimum Euclidean distance criterion (eq. (4.7)). All neurons are then ordered according to eq. (4.8). Based on this rank, a certain number of units is adapted.

The neurons' weights are updated according to:

$$w_j(t+1) = w_j(t) + \alpha(t) h_\lambda(k_j(x, w_j))[x(t) - w_j(t)] \tag{4.9}$$

where $\alpha(t) \in [0, 1]$ describes the overall extent of the modification, and h_λ is 1 for $k_j(x, w_j) = 0$ and decays to zero for higher values according to eq. (4.10):

$$h_\lambda(k_j(x, w_j)) = \exp\left(-\frac{k_j(x, w_j)}{\lambda(t)}\right) \tag{4.10}$$

As in Martinetz *et al.* [148], $k_j(x, w_j)$ is a function that represents the ranking of each weight vector w_j. If j is the closest to input x then $k = 0$, for the second closest $k = 1$ and so on. The learning rule is local, meaning that the operation at neuron level only requires the knowledge of the weight associated with that neuron and the input vector. However, a non-local function is needed to rank the neurons [145].

The name "neural gas" emphasizes the similarity to the movement of gas molecules in a closed container. Thus a neural gas network can be seen as using a temperature coefficient to affect the softness of the competition over time [149]. The network minimizes the following cost (error) function:

$$E(w,\lambda)=\frac{1}{2C(\lambda)}\sum_{j=1}^{N}\int P(x(n))h_\lambda(k_j(x,w))\cdot(x-w_j)^2 \tag{4.11}$$

where $P(x)$ is the distribution of input vectors $x(n)$ and

$$C(\lambda)=\sum_{k=0}^{N-1}h_\lambda(k) \tag{4.12}$$

The parameter λ can be seen as a temperature factor. At high temperatures many of the nodes in the network add to the cost of the network. As the temperature is reduced only the nodes close to the input contribute significantly to the error, and at very low temperatures only the winning node produces a cost. The corresponding update rule, that is applied to all nodes and which reduces the cost function in shown in eq. (4.9).

The learning rate α (t) and the function $\lambda(t)$ are both time-dependent. In order to ensure that the algorithm converges, it is important that these parameters are decreased slowly during the learning process. Usually the following time dependencies are used:

$$\alpha(t)=\alpha_0\left(\frac{\alpha_T}{\alpha_0}\right)^{t/T} \tag{4.13}$$

$$\lambda(t)=\lambda_0\left(\frac{\lambda_T}{\lambda_0}\right)^{t/T} \tag{4.14}$$

where the constants α_0 and λ_0 are the initial values for α (t) and λ (t), α_T and λ_T are the final values, t is the time step and T the training length. For the time-dependent variables, initial and final values have to be chosen. The algorithm continues to generate random input signals x while $t<T$.

Neural gas converges quickly, reaches after convergence lower distortion error than other self-organizing architectures and obeys a gradient descent on an energy function. However, the neural gas network can be computationally expensive. This cost is mainly due to the neighborhood ranking procedure. All nodes have to be ranked and updated for each input vector. In spite of the fact that the training procedure is longer in comparison with other self-organizing architectures as shown in [147], neural gas is chosen because the main goal in the context of this work is the accuracy of models as opposed to real-time behavior of simplified computer generated and simulated models that has already been extensively studied in the literature.

4.2.4. Data Sets

This neural architecture has been actively studied for various sources of simulated data [147]. In the context of this thesis, 3D range data measured point-clouds of objects and also sets of synthetic data obtained from the interpolation on a higher mesh of a sparse scan are used to train the network. Normalization is employed to remove redundant information from a data set, by a linear rescaling of the input vectors such that their variance is 1. As in [135], for each input vector x, its mean is computed as:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{4.15}$$

where n ($n = 1,\ldots, N$) denotes the dimension of the vector x, and the variance according to:

$$\sigma^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x}) \tag{4.16}$$

The set of rescaled variables are defined as:

$$\tilde{x} = \frac{x - \bar{x}}{\sigma} \tag{4.17}$$

This ensures that the transformed variables will have 0 mean and a variance of 1.

4.2.5. Testing

The mean quantization error is used as a measure to evaluate the quality of the obtained models. The quantization error (mapping precision) [150] describes how accurately the neurons respond to the given data set, and therefore how well the model built corresponds to the 3D object embedded in the input point-cloud. For example, if the reference vector of the winning neuron calculated for a given testing vector x_i is exactly the same x_i, the error in precision is 0. Normally, the number of data vectors exceeds the number of neurons and the precision error is thus always different from 0. It is computed as an average distance between each data vector and its winning neuron, as follows:

$$qe = \frac{1}{N}\sum_{j=1}^{N}\left\|x_j - w_i(x_j)\right\| \tag{4.18}$$

4.2.6. Use of Neural Gas for Selective Data Acquisition

In the context of this work, a neural gas architecture is employed to guide vision and tactile probing by clustering measurements that represent uniform geometry and/or elasticity regions and therefore direct the sensor toward areas of transitions where higher sampling density is required, based on the self-organizing neural network ability to find an optimal finite set that quantizes the given input space. This ensures the capability of the modeling scheme to deal not only with homogeneous objects from the elasticity point of view, but also with non-homogeneous, particularly piecewise-homogeneous, objects as well. Starting from the points collected during a fast scan of an object via an active range finder and an initial configuration of unconnected nodes, the latter move freely over the data space while learning and the model contracts asymptotically towards the points in the input space, respecting their density and thus taking the shape of the objects encoded in the point-cloud. This ensures that density of the probing points will be higher in the regions with more pronounced variations in the geometric shape and elastic behaviour.

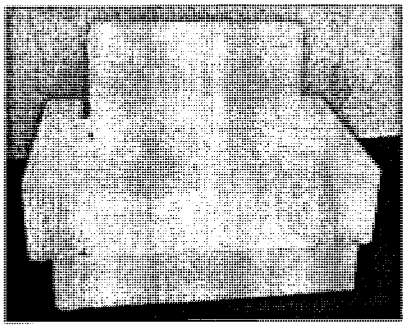
4.3. Experimental Results for Geometric Data Acquisition with Vision Sensors

In an initial stage of research, tests were performed to check the proposed modeling scheme for a full scan point-cloud of an object in order to identify those regions of interest where changes occur into the geometry of an object [147]. This testing scenario corresponds to the post-processing procedures for data compression explained in subchapter 2.2. Within a small margin of error, the proposed model was always able to identify regions of geometrical changes in an object point-cloud.

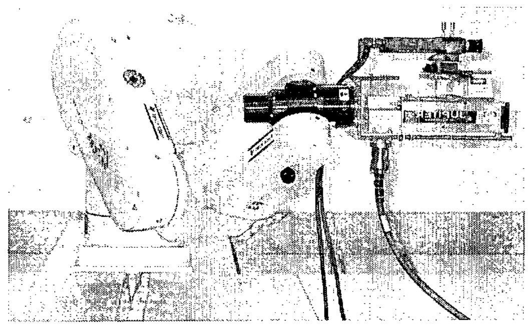
In the context of this thesis, the proposed solution is meant to be incorporated directly in the sampling procedure, to achieve automated selective scanning over large workspaces, by adaptively selecting regions of interest for further refinement from a cloud of 3D sparsely collected measurements. Starting from an initial low resolution scan of an object, a neural gas network is employed to model the resulting point-cloud. Those regions that are worth further sampling in order to ensure an accurate model are detected by finding higher density areas in the neural gas map. This is done by applying a Delaunay triangulation (discussed in detail in the following chapter) to the resulting neural gas output map and by subsequently removing all the triangles that are larger than a set threshold. The threshold is automatically computed based on the length of vertices for each triangle in the tessellation. Rescanning at higher resolution is performed for each identified region (regions marked by the remainder of small triangles in the tessellation) and a multi-resolution model is then built using the initial sparse model and augmenting it with the high resolution regions of interest. The rescanning procedure can be repeated in several steps to gradually improve the accuracy of the model. The number of steps in which the procedure is repeated depends on the accuracy needs of the application, but also on the type scanner employed (e.g. its level of granularity and the maximum allowed resolution).

4.3.1. Experimental Measurement Data

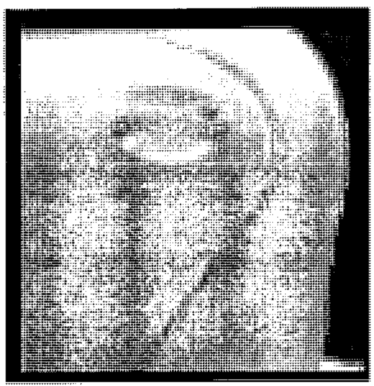
The proposed method is first tested in the context of range imaging where selective sampling of points is performed with range vision sensors in order to accurately describe a 3D object. Detailed results of experimentations are presented for three objects: the triceratops from section 3.2.1, a foam armchair and a mock-up car door, each presenting different sorts of regions of interest, as depicted in Fig. 4.2e, Fig. 4.2a and Fig. 4.2c respectively.



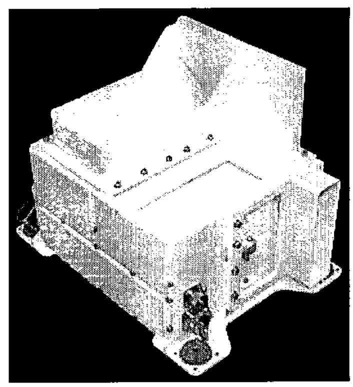
a)



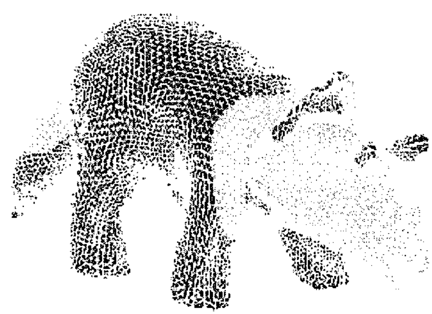
b)



c)



d)



e)

Fig. 4.2. a) Foam armchair, b) Jupiter laser scanner used to collect the data, and c) car door collected using d) Neptec's full image LMS range sensor and e) high resolution point-cloud of the triceratops.

For the triceratops model, the area of the head, the neck and around the horns should be regions of interest. The armchair should have as regions of interest the edges, while for the door model, the regions of interest should be depicted around the door knob and the door opening gap.

In order to create a full point-cloud for the triceratops model depicted in Fig. 4.2e, the initial coarse mesh model from Fig. 3.1b is interpolated on a high resolution mesh. The interpolation is based on a displaced subdivision mesh hierarchy. First, measurements are diffused on the coarse resolution mesh. The measurement values are kept fixed but vertices with no measurement are filled in by the diffusion process. The diffusion is solved using forward Euler steps. The interpolated measurements on a coarser level mesh are mapped on the next finer level mesh and the diffusion is calculated again. In case of the triceratops, refinements over four levels are executed [136, 137].

The point-cloud model of the armchair depicted in Fig. 4.2a is obtained using an automated Jupiter laser scanner mounted on a robot arm, as shown in Fig. 4.2b. The full resolution point-cloud of a mock-up car door shown in Fig. 4.2c is collected with the Neptec Design Group Inc.'s Laser Metrology System (LMS) range sensor, depicted in Fig. 4.2d. These three examples of objects are representative, since the objects are not only very different in terms of features, but come from different sources of data as well, showing the capability of the neural gas solution to cope with different data features and with data coming from various sensing mechanisms.

4.3.2. Experimental Results for Vision Data Acquisition

Starting from an initial fast, sparse scan of each object, a neural gas network is employed to model the data in the point-cloud, in the form of (x, y, z) coordinates. The data set is normalized prior to the neural gas mapping such that its variance is unity as required by the learning procedure and as described by eq. (4.15)-(4.17).

Testing is performed for several resolutions of the initial fast/sparse scan in order to identify how the resolution of this scan influences the modeling results and what would be the smallest scan to start with, that allows for the modeling of fine features in the modeled objects. Tests are run for an initial resolution of roughly 5%, 10%, 15%, 25%, 35% and 50% of the full resolution point-cloud. In each case, simulations are performed until the map size becomes larger than 100% of the initial scan. All simulations are performed for $\alpha_0 = 0.5$ and λ_0 equal to half of the number of neurons. The influence of the training parameters over the modeling results has been investigated and analyzed in [147]. The training is performed using Matlab code, on a Pentium IV 1.3GHz machine with 512MB of memory. Experiments are conducted for each of the three objects discussed in subchapter 4.3.1.

4.3.2.1. The Case of Objects with Roughly Uniform Distributed Sampling Points and Multiple Features

The point-cloud model of the triceratops is created, as explained in section 4.3.1, starting from a uniformly distributed mesh and results in a uniformly distributed point-cloud. It allows for the study of an object with multiple features and a roughly uniform distribution of points in the point-cloud.

The results for the triceratops are presented in Table 4.1 and Table 4.2 for different initial resolution scans and different neural gas map sizes. Table 4.1 presents the modeling results for an initial scan size of 5% to 15% of the full resolution model, while Table 4.2 depicts the modeling results for an initial size from 20% to 50% of the full resolution model. The tables also show, under each model, the approximate size of the neural gas map employed, as a percentage of the size of the initial scan. The size of the neural gas map is only considered until it reaches the one of the initial scan (100% of the initial scan size), and therefore some cells are empty as for those initial sizes the map of the neural gas reaches or surpasses the size of the initial scan. The modeling results for a neural gas network having as inputs these initial sparse point-clouds are compressed models for the dataset in which the weight vector consists of the 3D coordinates of the object's points. The high resolution areas in the resulting neural gas maps are obtained by removing large triangles in a Delaunay triangulation built on the neural gas map, for a constant threshold for all modeling cases to give a common comparison basis.

As it can be seen in Table 4.1, for an initial scan of less than 15% of the full resolution point-cloud, or less than roughly 2000 points, the model is able to capture the overall shape of the object embedded but is not capable to capture accurately fine details, as the area around the horns of the triceratops, for example, in spite of the increase in the map size.

Table 4.2 depicts the regions of interest resulting for an initial scan size from 20% to 50% of the full resolution model.

Table 4.1. *Evolution of regions of interest identification for different map sizes and for initials scan sizes between 5% and 15% of the full resolution triceratops point-cloud.*

Map size	***509 points = 4.16% of full resolution point-cloud***	***1222 points = 9.99% of full resolution point-cloud***	***1746 points = 14.28% of full resolution point-cloud***
15*25 (375 points)	 73.6% of initial scan	 30.6% of initial scan	 21.4% of initial scan
20*25 (500 points)	 98.2% of initial scan	 40.9% of initial scan	 28.6% of initial scan
25*30 (750 points)		 61.3% of initial scan	 42.9% of initial scan
35*40 (1400 points)		 71.6% of initial scan	 50.1% of initial scan
35*45 (1575 points)			 90.2% of initial scan

Table 4.2. *Evolution of regions of interest identification for initials scan sizes between roughly 20% and 50% of the full resolution triceratops point-cloud.*

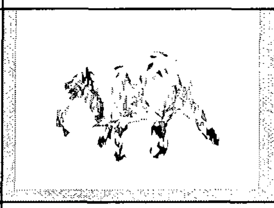
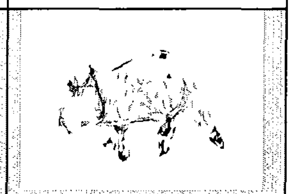
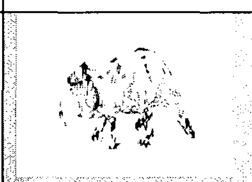
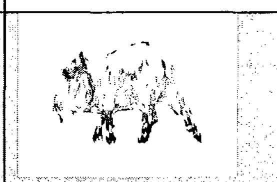
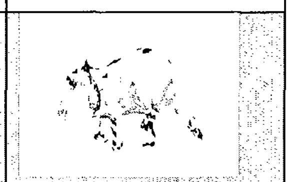
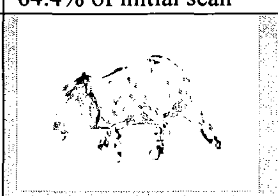
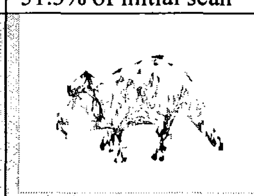
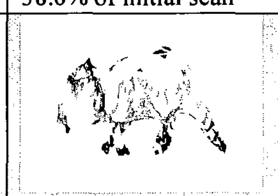
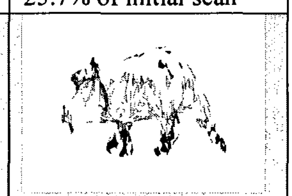
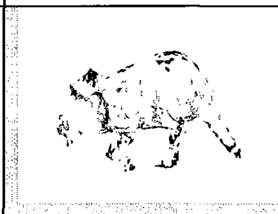
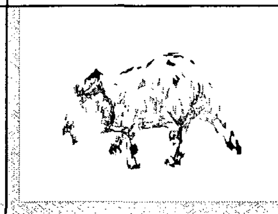
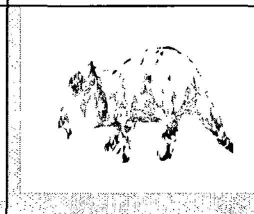
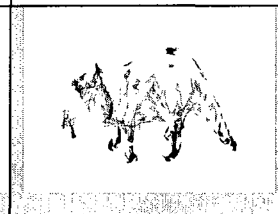
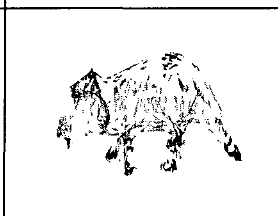
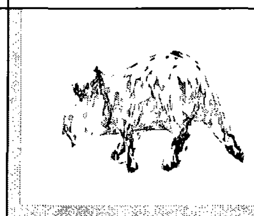
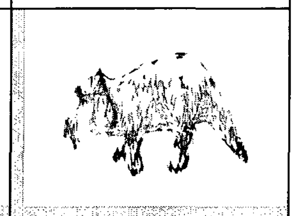
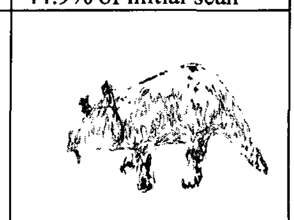
Map size	***2445 points=19.99% of full resolution point-cloud***	***3065 points = 25% of full resolution point-cloud***	***4075 points = 33% of full resolution point-cloud***	***6113 points = 50% of full resolution point-cloud***
15*25 (375 points)				
	15.3% of initial scan	12.2% of initial scan	9.2% of initial scan	3% of initial scan
20*25 (500 points)				
	20.4% of initial scan	16.3% of initial scan	12.2% of initial scan	4% of initial scan
25*30 (750 points)				
	30.6% of initial scan	24.4% of initial scan	18.4% of initial scan	12.2% of initial scan
35*40 (1400 points)				
	35.7% of initial scan	28.5% of initial scan	21.4% of initial scan	14.3% of initial scan
35*45 (1575 points)				
	64.4% of initial scan	51.3% of initial scan	38.6% of initial scan	25.7% of initial scan
40*45 (1800 points)				
	73.6% of initial scan	58.7% of initial scan	44.1% of initial scan	29.4% of initial scan

Table 4.2. (cont.) Evolution of regions of interest identification for initials scan sizes between roughly 5% and 15% of the full resolution triceratops point-cloud.

Map size	*2445 points=19.99% of full resolution point-cloud*	*3065 points = 25% of full resolution point-cloud*	*4075 points = 33% of full resolution point-cloud*	*6113 points = 50% of full resolution point-cloud*
*40*50 (2000 points)*	 81.7% of initial scan	 65.2% of initial scan	 49% of initial scan	 32.7% of initial scan
*50*55 (2750 points)*		 89.7% of initial scan	 67.4% of initial scan	 44.9% of initial scan
*60*70 (4200 points)*				 68.7% of initial scan

It can be seen that starting from roughly 20% of the full resolution scan, or about 2500 points in the initial sparse scan, the network is able to model correctly and identify fine features of the triceratops. The good modeling results are framed in light gray. In this case, it can be observed that a larger neural gas map size models better a lower resolution scan, while a smaller map size is enough for the higher resolution scan. Overall, the visual quality improves with the increase in the size of the initial scan.

Fig. 4.3 shows the detected regions of interest, enlarged, for the best adaptive threshold obtained using several trials, for a initial scan of 3065 points (25% of the full resolution scan), shown in Fig. 4.3a and for an initial scan of 6113 points (50% of the full resolution scan), presented in Fig. 4.3b.

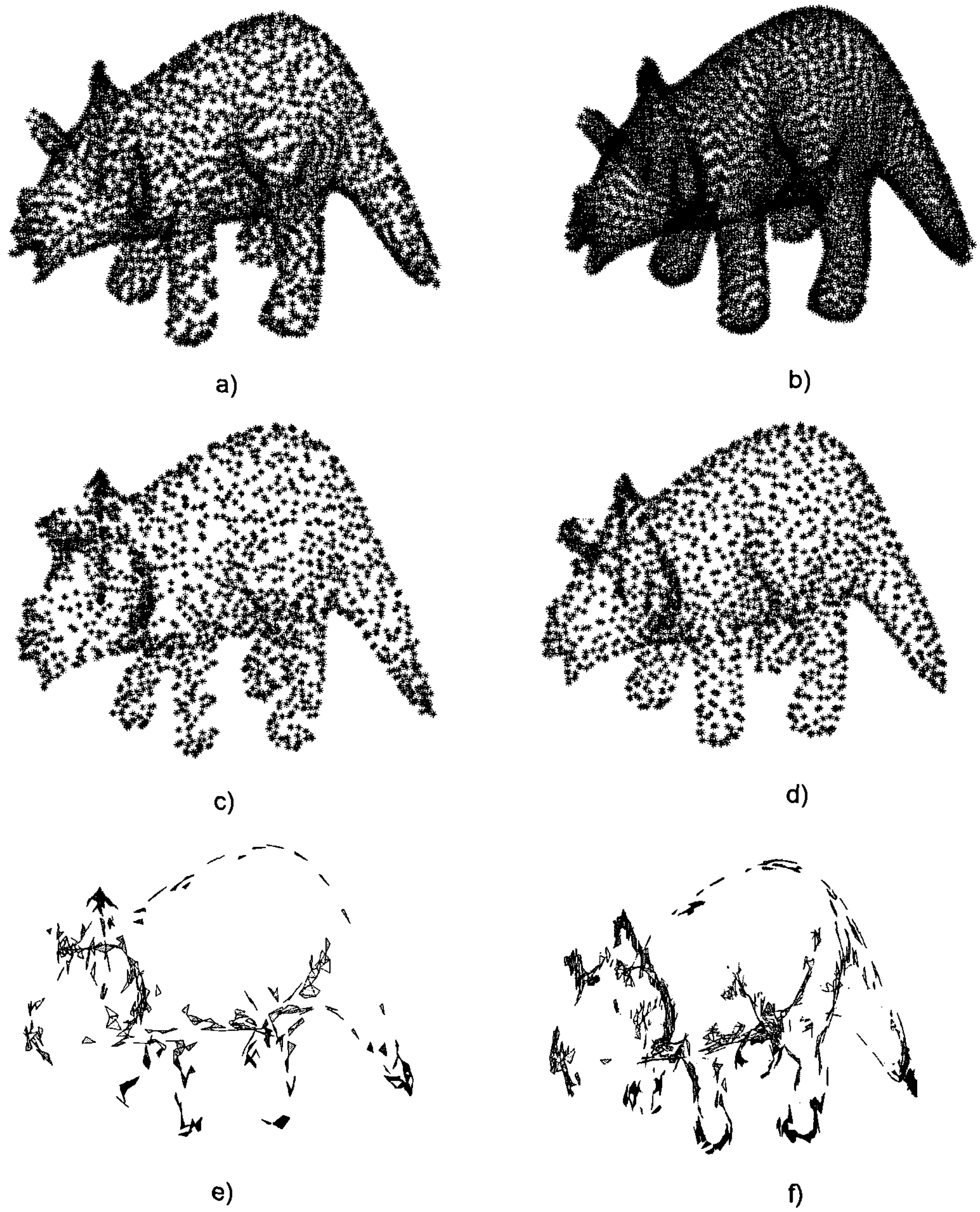


Fig. 4.3. Initial scan of the triceratops model of (a) 3065 points and (b) 6113 points, neural gas model for a map size of (c) 40×45 for the 3065 points initial scan and (d) 35×45 for the 6113 points initial scan and detected regions of interest for further sampling for (e) 3065 points initial scan and (f) 6113 points initial scan.

The neural gas network having as inputs these initial sparse point-clouds, for a neural gas map size of 40×45 (1800 points and therefore about 58% of the size of the initial scan) and 35×45 (1575 points and therefore a size of about 25% of the size of the initial scan) respectively are compressed models for the dataset in which the weight vector consists of the 3D coordinates of the object's points and are depicted in Fig. 4.3c and d. The high resolution areas are much better identified and contoured for the higher resolution point-cloud, as it can be seen in Fig. 4.3f. In spite of some noise, in both cases, the network is able to identify, apart from the contours of the model, the areas around the head and horns of the triceratops as areas for further scanning, as it is expected. These areas can be scanned at this point at a higher resolution in order to improve the accuracy of the model.

In this modeling case, a higher resolution of the initial sparse scan increases the accuracy of the sampling procedure for larger neural map sizes. This can be also seen by studying the evolution of the quantization error for the same modeling cases, depicted in Fig. 4.4.

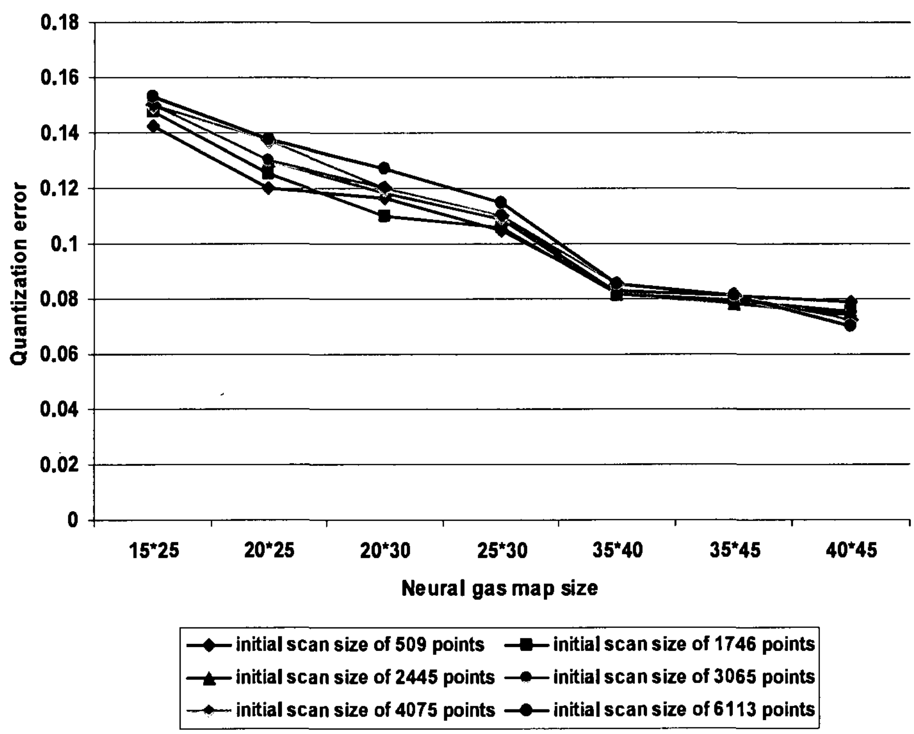


Fig. 4.4. Evolution of quantization error with neural gas map size for different initial scan sizes of the triceratops point-cloud.

The error tends to decrease with the size of the neural gas map and reaches lower values for higher resolution initial scans. However, the absolute values are not significantly different for neural gas map sizes larger than 25×30 (within 0.015). The training time however increases with both an increase in a neural gas map size and in the resolution of the initial scan, as depicted in Fig. 4.5. Therefore the choice of the right size is a compromise between the accuracy required by an application and the time the network takes to build this accurate model.

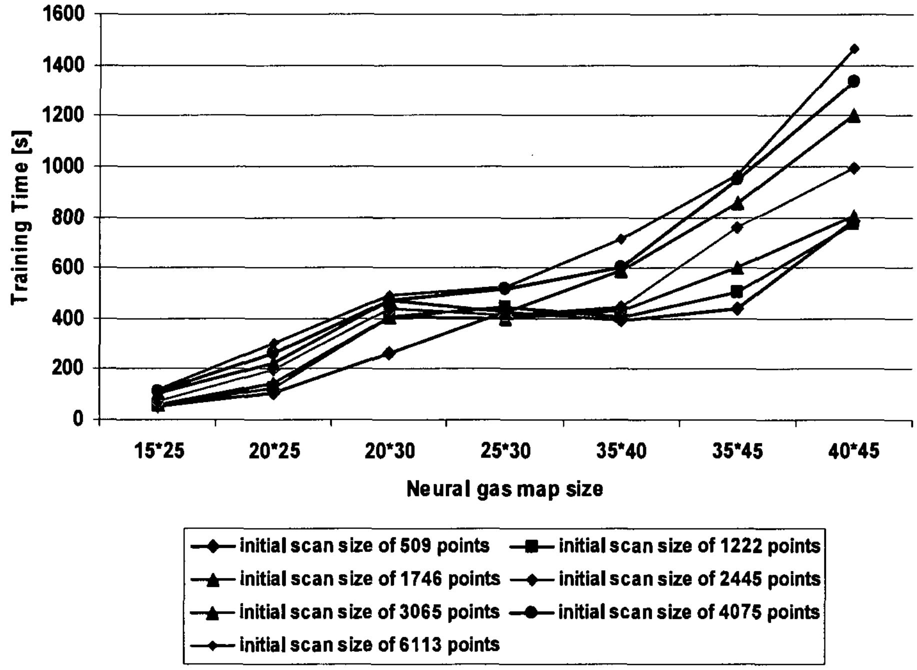


Fig. 4.5. Evolution of training time with the neural gas map size for different initial scan sizes of the triceratops point-cloud.

4.3.2.2. The Case of Objects with Raster-like Distribution of Sampling Points and Few Non-Localized Features

The case study of the chair is the representative of an object without many features, with raster-like distributed sampling points. The data are used as collected by the Jupiter laser scanner (without any filtering procedure) therefore some noise is present in the scans. The same training parameters are used as in the case of the triceratops and the results are evaluated for different neural gas map sizes and different sizes of the initial scan. Some results for the identification of the regions of interest are presented in Table 4.3 and Table 4.4.

Table 4.3. *Evolution of regions of interest identification for initials scan sizes between roughly 5% and 15% of the full resolution chair point-cloud.*

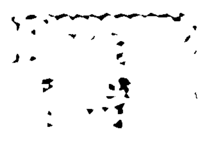
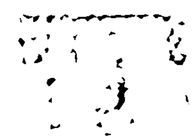
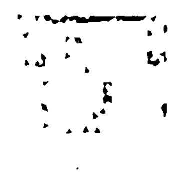
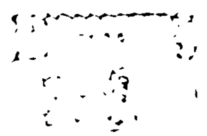
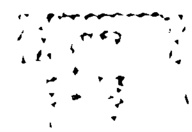
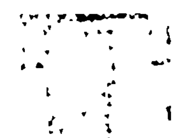
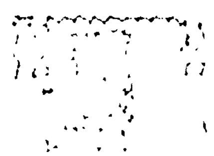
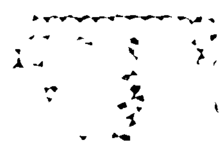
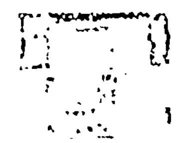
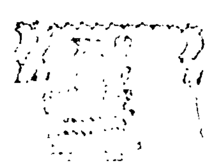
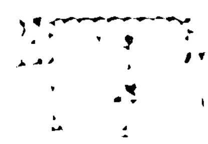
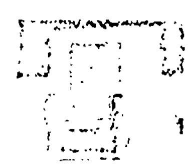
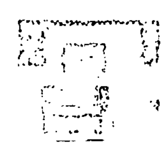
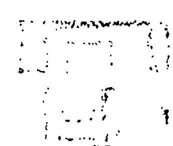
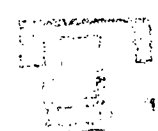
Map size	***769 points = 5% of full resolution point-cloud***	***1538 points = 10% of full resolution point-cloud***	***2197 points = 14.28% of full resolution point-cloud***
15*25 (375 points)	 48.7% of initial scan	 24.38% of initial scan	 17% of initial scan
20*25 (500 points)	 65% of initial scan	 32.5% of initial scan	 22.7% of initial scan
25*30 (750 points)	 97.5% of initial scan	 48.7% of initial scan	 34.1% of initial scan
35*40 (1400 points)	 113.7% of initial scan	 56.8% of initial scan	 39.8% of initial scan
35*45 (1575 points)		 117% of initial scan	 71.6% of initial scan
40*45 (1800 points)			 81.9% of initial scan

Table 4.4. *Evolution of regions of interest identification for initials scan sizes between roughly 25% and 50% of the full resolution chair point-cloud.*

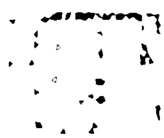
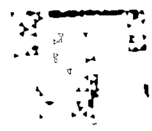
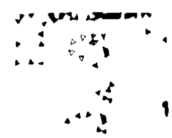
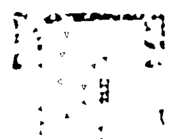
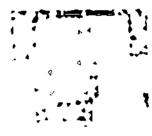
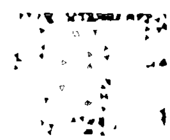
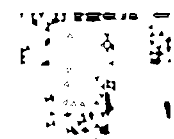
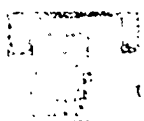
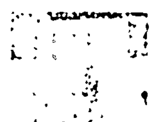
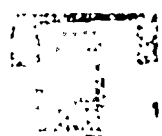
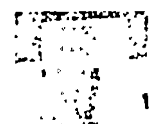
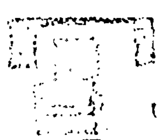
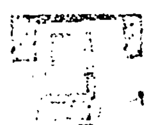
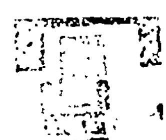
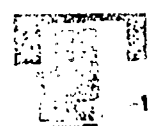
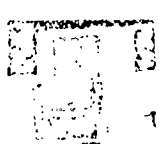
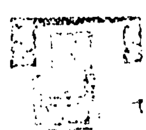
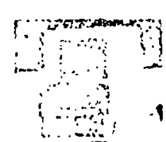
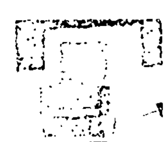
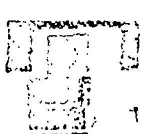
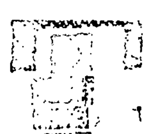
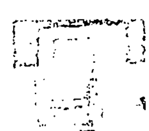
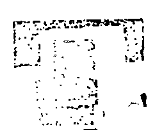
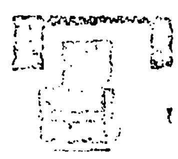
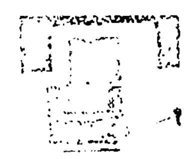
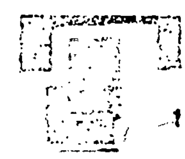
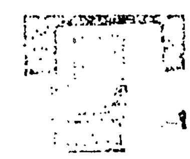
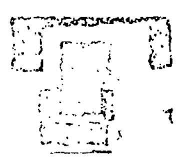
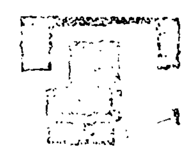
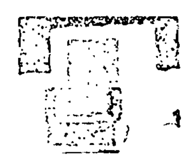
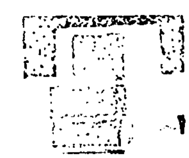
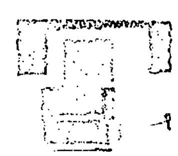
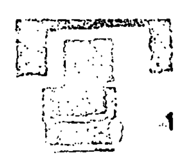
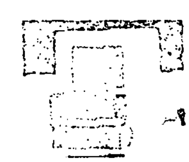
Map size	***3076 points=20% of full resolution point-cloud***	***3845 points = 25% of full resolution point-cloud***	***5127 points = 35% of full resolution point-cloud***	***7691 points = 50% of full resolution point-cloud***
15*25 (375 points)	 12.1% of initial scan	 9.7% of initial scan	 7.3% of initial scan	 4.8% of initial scan
20*25 (500 points)	 16.2% of initial scan	 13% of initial scan	 9.7% of initial scan	 6.5% of initial scan
25*30 (750 points)	 24.3% of initial scan	 19.5% of initial scan	 14.6% of initial scan	 9.7% of initial scan
35*40 (1400 points)	 28.4% of initial scan	 22.7% of initial scan	 17% of initial scan	 11.3% of initial scan
35*45 (1575 points)	 51.2% of initial scan	 40.9% of initial scan	 30.7% of initial scan	 20.4% of initial scan
40*45 (1800 points)	 58.5% of initial scan	 46.8% of initial scan	 35.1% of initial scan	 23.4% of initial scan

Table 4.4.(cont.) *Evolution of regions of interest identification for different neural gas map sizes and for initials scan sizes between roughly 25% and 50% of the full resolution chair point-cloud.*

Map size	***3076 points=20% of full resolution point-cloud***	***3845 points = 25% of full resolution point-cloud***	***5127 points = 35% of full resolution point-cloud***	***7691 points = 50% of full resolution point-cloud***
40*50 (2000 points)	 65% of initial scan	 52% of initial scan	 39% of initial scan	 26% of initial scan
50*55 (2750 points)	 89.4% of initial scan	 71% of initial scan	 53.6% of initial scan	 35.7% of initial scan
60*70 (4200 points)		 109% of initial scan	 81.9% of initial scan	 54.6% of initial scan

As it can be seen in Table 4.3, similar to the case of the triceratops, the network is not able to capture the model for an initial resolution scan with a number of points less than 2200 or 15% of the full resolution point-cloud. Starting from roughly 20% of the full resolution scan or roughly 3000 points in the initial scan, the network is able to model correctly and identify fine features of the chair. The good modeling results are again framed in light gray. It can be observed, as for the triceratops, that a larger neural gas map size models better a lower resolution scan, while a smaller map size is enough for the higher resolution scan. Overall, the visual quality improves with the increase in the size of the initial scan.

Fig. 4.6 and Fig. 4.7 show the evolution of the error and training time for different resolution initial scans and different neural gas map sizes.

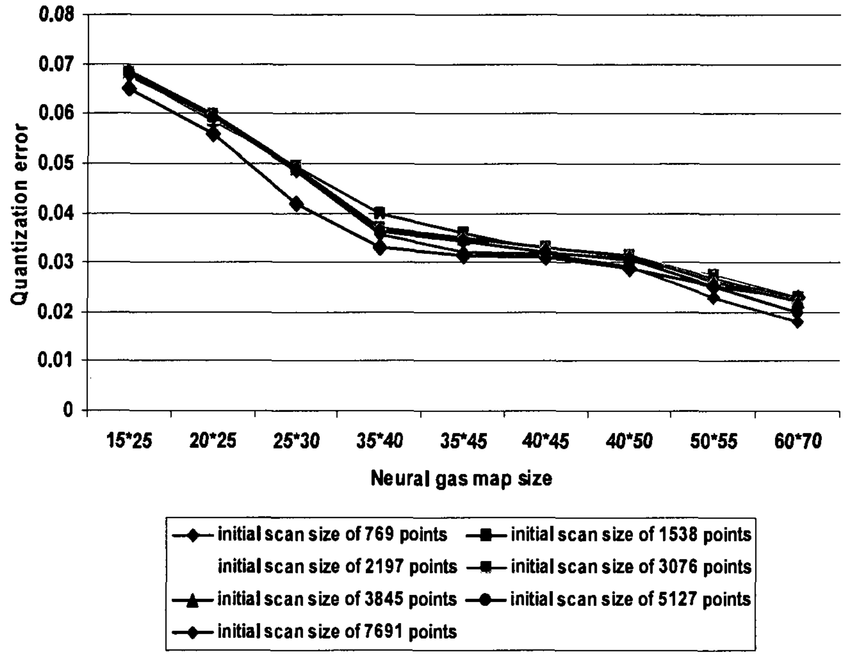


Fig. 4.6. Evolution of quantization error with neural gas map size for different initial scan sizes of the chair point-cloud.

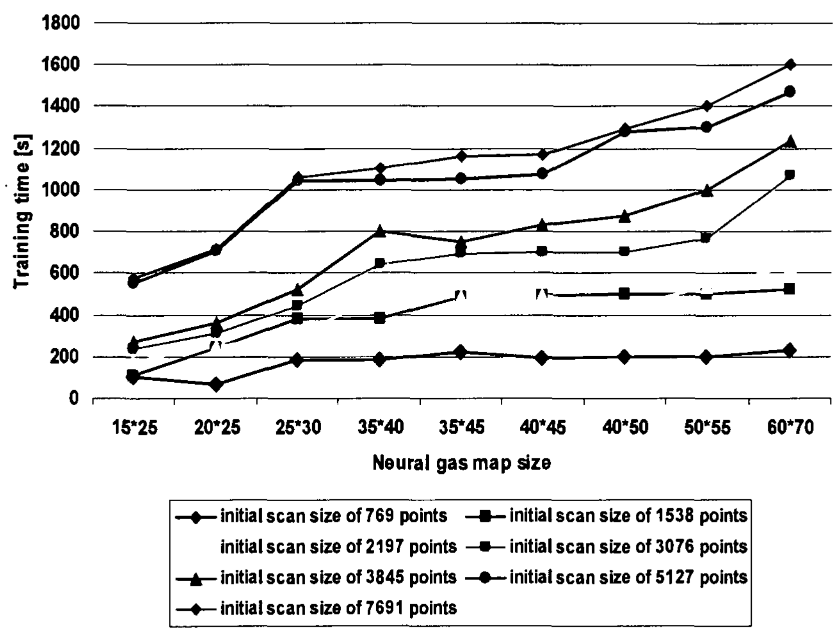


Fig. 4.7. Evolution of training time with neural gas map size for different initial scan sizes of the chair point-cloud.

Again, the error decreases with an increase in the map size for large neural map sizes, while the training time increases with an increase in the neural gas map size and in the size of the initial scan. The best enlarged modeling results, for initial scan of the chair of 3845 points (25% of the full resolution scan) and 7691 points (50% of the full resolution scan), are shown enlarged in Fig. 4.8a and b respectively.

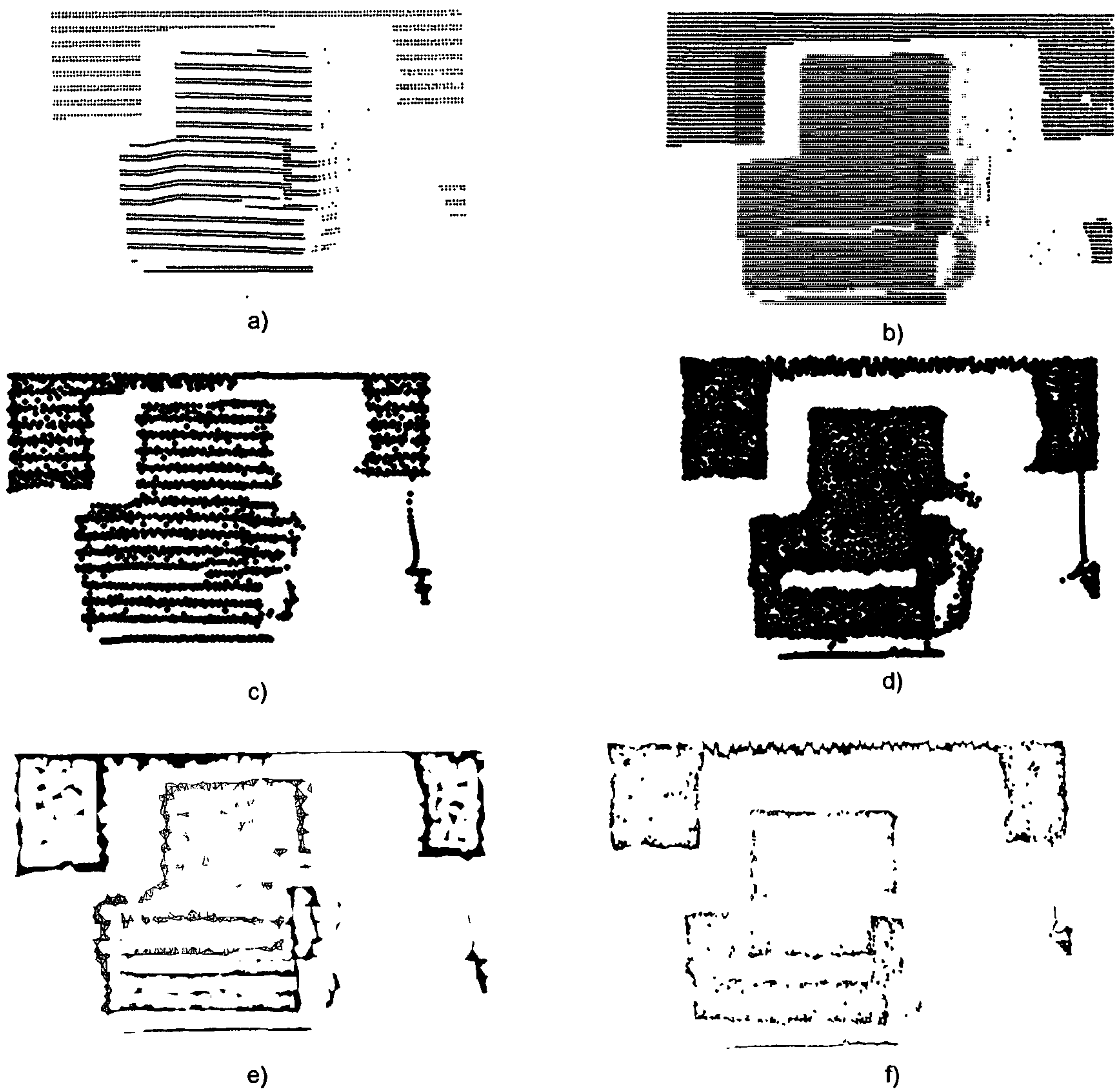


Fig. 4.8. Initial scan of a chair at a resolution of (a) 3845 points and (b) 7691 points, neural gas model for a map size of (c) 40×45 for the 3845 points initial scan and (d) 35×45 for 7691 points initial scan and detected regions of interest for further sampling for the initial scan of (e) 3845 points and (f) 7691 points.

These point-clouds are provided each as inputs to a neural gas network with the map size of 40×45 and 35×45 respectively and the obtained output maps are shown in Fig. 4.8c and d respectively. The artefacts in the neural gas map are due to the fact that the training is stopped early in order to avoid that the nodes become uniformly distributed instead of capturing details. As the point-clouds that are obtained using the laser sensors are raster-like models and their density is uniform, the neural gas, that respects the density in the input samples, after a long training, will tend to build uniformly dense maps as opposed to keeping nodes in the regions rich in features.

Since there are no other significant details in the model apart from the edges and the back plane, these are selected as regions of interest for additional sampling in order to improve the accuracy of the model. Even in the case of the low resolution initial dataset of 3851 points and in spite of some noise and artifacts visible in the regions of interest identification result (Fig. 4.8e), the proposed algorithm succeeds to identify the edges. When the initial dataset is denser, those features are more accurately located.

4.3.2.3. The Case of Objects with Raster-like Distribution of Sampling Points and Small Localized Features

This case is studied for the mock-up car door in Fig. 4.2c, that presents fine localized details. The same modeling scenario is followed as in the case of the triceratops and of the chair and the findings are summarized here. A fast scan is initially performed to obtain a very low resolution point-cloud of 4096 points shown in Fig. 4.9a (0.39% of the high-resolution full scan of 1048576 points) and a medium low resolution of 16384 points depicted in Fig. 4.10a (1.56% of the high-resolution full scan). The data provided by the Neptec's full image LMS range sensor is less noisy, due to its higher resolution when compared to the Jupiter laser scanner. Fig. 4.9b and Fig. 4.10b show the rendered mesh models embedded in each point-cloud. The region representing the door opening is very small in comparison with the whole model and so is the number of points representing it. This requirement poses additional challenges to the modeling. In case of the very low resolution model, a denser neural gas map is required in order to capture the details, again stopped early in the training as in the case of the chair.

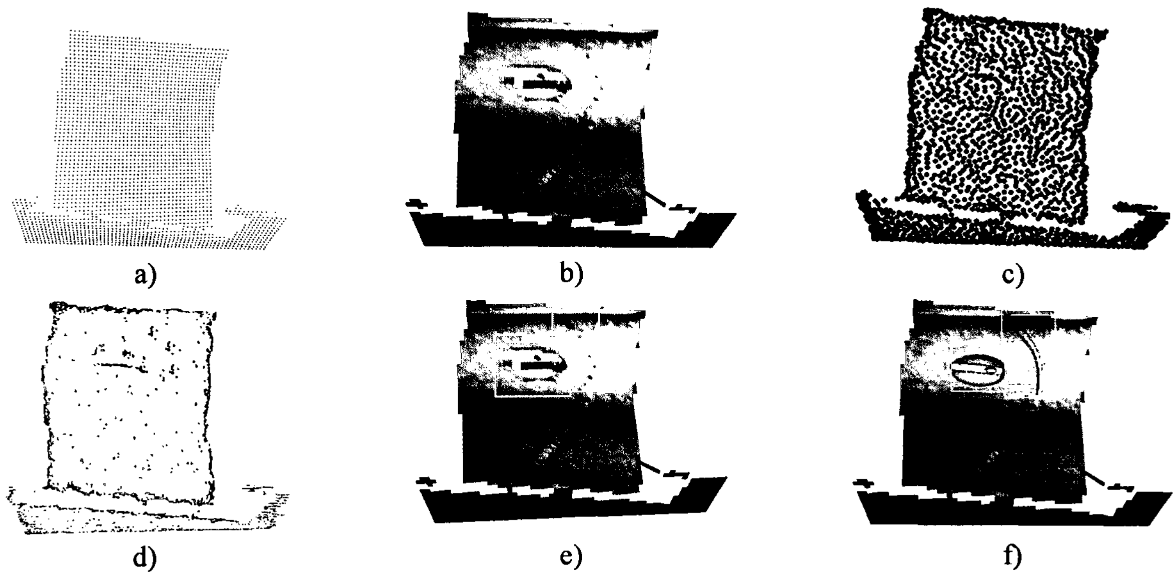


Fig. 4.9. Modeling steps for a very low resolution car door point-cloud: a) initial point-cloud, b) rendered mesh model, c) neural gas model, d) higher density areas in neural gas model, e) identified regions of interest, and f) selectively densified model.

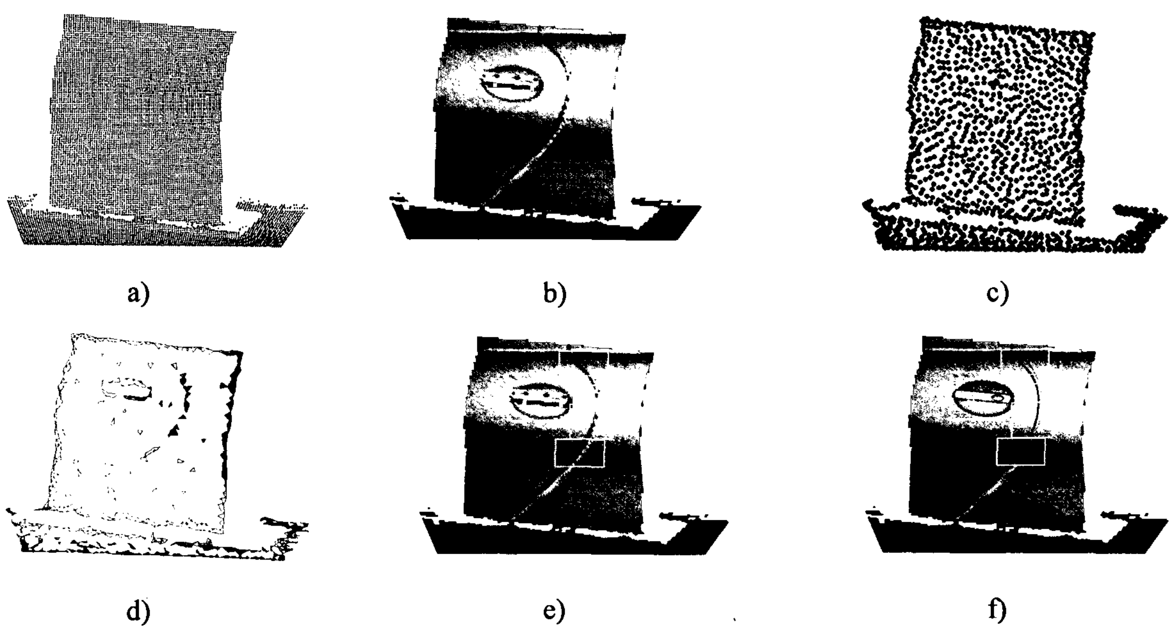


Fig. 4.10. Modeling steps for a medium low resolution car door point-cloud: a) initial point-cloud, b) rendered mesh model, c) neural gas model, d) higher density areas in neural gas model, e) identified regions of interest, and f) selectively densified model.

The normalized point-cloud of the 4096 points initial scan is provided as input to a neural gas network with a map size of 45×40 (1800 points which represents a map size of 43% of the number of points in the initial sparse scan) and the resulting model is depicted in Fig. 4.9c. A lower density map provides good results both in terms of accuracy and length of training for the medium low resolution model. The training result for a map size of 35×45 is shown in Fig. 4.10c. Those regions that are worth further sampling in order to ensure an accurate model are detected by finding high density areas in the neural gas model. The detected regions of high density in the neural gas output map are identified in Fig. 4.9d and Fig. 4.10d respectively, by building the Delaunay triangulation and removing large triangles from it. Higher density scans are performed within these bounded areas. The regions of interest superimposed on the rendered model are depicted in Fig. 4.9e and Fig. 4.10e respectively. The resulting augmented multi-resolution models for the sample of regions of interest are presented in Fig. 4.9f and Fig. 4.10f respectively. The selectively densified model in Fig. 4.9f contains 111596 points and it represents a reduction of about 90% in the number of points when compared to the full scan, while the one in Fig. 4.10f contains 173884 points (about 83% reduction from the full resolution scan). It would be expected that an even higher resolution initial scan would give better results in terms of regions of interest identification. However, this is not necessarily true. Additional points in the model make the door gap almost invisible, since the relative number of points representing it in the point-cloud is significantly reduced, especially for point-clouds with a larger number of points than 40% of the highest resolution scan.

The same procedure can be then repeated for each of the regions of interest detected in the previous step. Each region can be provided as input in a new neural gas network in order to further detect fine details that are worth to be scanned at a higher resolution. Fig. 4.11 presents the detail of the high resolution rendered model of the door, with the selected regions in the previous step. The point-cloud of each region has slightly over 5000 points. For each of the regions, the corresponding neural gas model for a map size of about 30% of the total number of the points in the region, and the selected regions for additional sampling, identified as high density areas in the neural gas model, are shown as well. The regions of interest are shown from different viewpoints in Fig. 4.12.

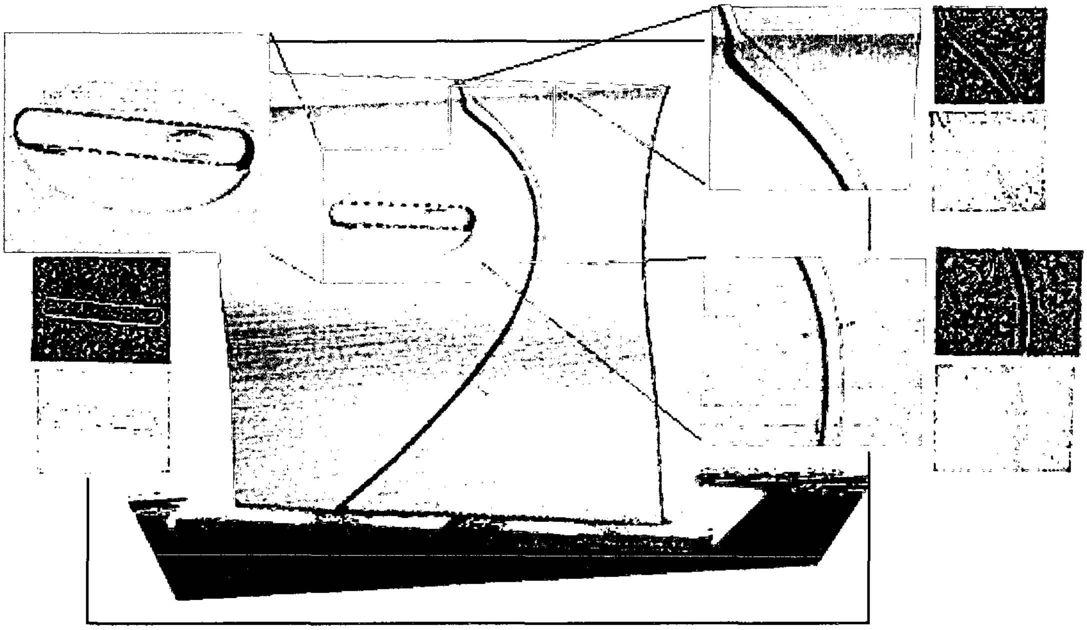


Fig. 4.11. Model of car door showing a sample of selected regions, their enlargement, their neural gas model and the areas for further sampling.

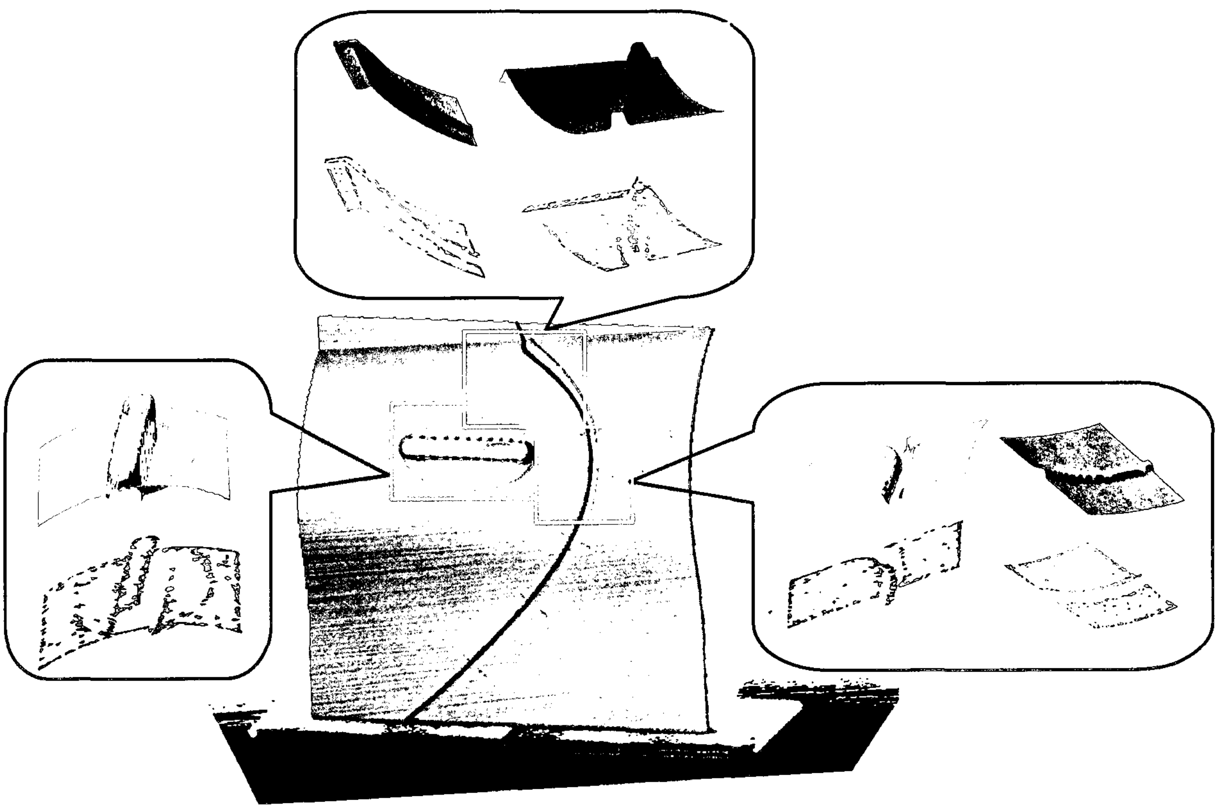


Fig. 4.12. Selected areas for further sampling from different viewpoints and the regions of interest identified in a neural gas map for each of them.

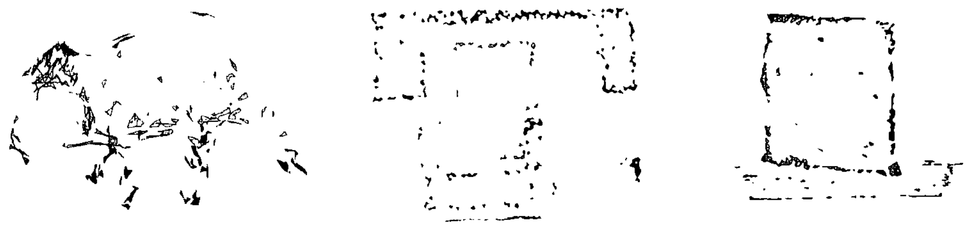
4.3.2.4. Guidelines to Choose an Appropriate Neural Gas Map Size

To fully benefit from the proposed method, research has been done in order to identify ways to select an appropriate neural gas map size for the objects under study that ensures a good identification of regions of interest. Particularly, the investigation looked into the correlation between the neural gas map size and the size of the initial scan point-cloud. These tests revealed that a neural gas network, regardless of its neural map size cannot capture enough features in an initial point-cloud with a size that is smaller than roughly 2000 points, as presented in Tables 4.1 and 4.3 and summarized in the first row of Fig. 4.13.

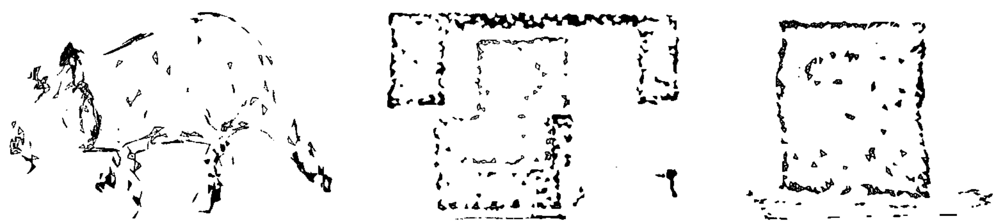
The second, third, fourth and fifth rows of Fig. 4.13 show some good modeling results obtained for 2500-3000-point, 4000-5000-point and 15000-16000-point and over 16000-point initial scan size, respectively. The "good" modeling results are identified as those compressed models with the highest compression rate that offer at the same time an optimal balance between the quality of the model (evaluated by a low relative error and good distribution of samples over the object surface, distribution that is evaluated visually) and the time required for training. The relative error is computed as an average Euclidean distance between each data vector and its winning neuron and therefore showing how close the modeled data is to the initial scan point-cloud.

As it can be observed in Tables 4.2 and 4.4 and summarized in the second and third row of Fig. 4.13, that starting from 2500 points in the initial sparse scan the network is able to capture the edges of the armchair, but captures only barely the triceratops and the door features. To capture correctly the area around the horns of the triceratops and the fine details of the door model, at least 4000 points are required. The same observations were made for other objects under study, leading to the conclusion that the initial scan should contain a minimum of 4000 points.

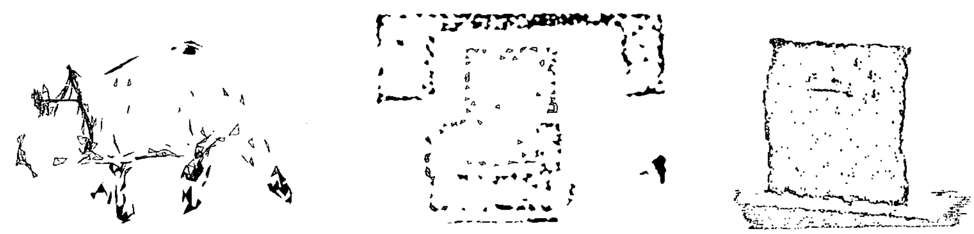
The same tests revealed that a larger neural gas map size models better a lower resolution initial scan, while a smaller map size is enough for a higher resolution initial scan. As shown in Tables 4.2 and 4.4 and Fig. 4.13 for example, a neural gas map of 40×50 is required to successfully capture some of the features in the object under study for the 1700-2000 initial scan size, while a map size of 25×30 is sufficient for initial point-clouds of over 16000 points.



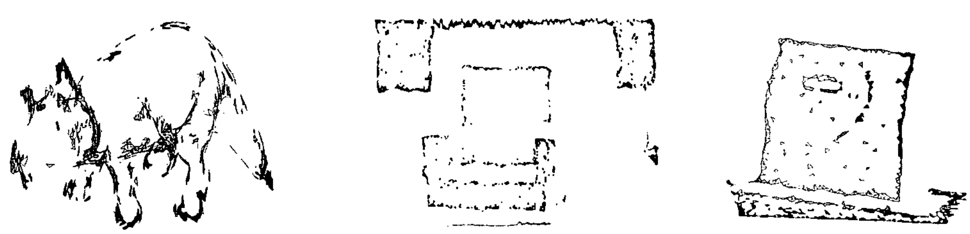
1700-2000 points initial scan, neural gas map size of 40×50



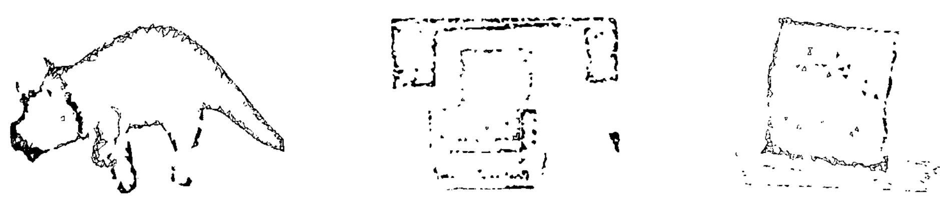
2500-3000 points initial scan, neural gas map size of 40×45



4000-5000 points initial scan, neural gas map size of 35×45



15000-16000 points initial scan, neural gas map size of 30×35



over 16000 points initial scan, neural gas map size of 25×30

Fig. 4.13. Examples of regions of interest detected for different sizes of initial scans and different neural gas map sizes.

Overall, the visual quality improves with the increase in the size of the initial scan. This fact is also proven by a decrease in the error with an increase in the map size as shown in Fig. 4.4 and Fig. 4.6. The error, even if it remains low overall, tends to decrease with the size of the neural gas map and reaches lower values for higher resolution initial scans. Therefore a higher resolution of the initial sparse scan increases the accuracy of the sampling procedure for larger map sizes. The training time however increases with both an increase in the map size and in the resolution of the initial scan, as depicted in Fig. 4.5 and Fig. 4.7. A compromise must then be identified on the initial scan and the neural gas map sizes.

On the other hand, one would expect that even higher resolution initial scans would give better results in terms of regions of interest identification. However, this is not necessarily true, as shown in the fifth row of Fig. 4.13. For initial scan sizes larger than 16000 points, for the triceratops model, the contour is better highlighted, but the model loses the fine details around the legs and the abdomen. In the model of the door, the additional points make the door gap almost invisible, since the relative number of points representing it is significantly reduced in comparison with the size of the point-cloud.

Table 4.5 summarizes the correlation that exists between the number of points in the initial scan and the map size that provides reasonably good modeling results for this initial scan.

Table 4.5. Approximate neural gas map sizes recommended for different sizes of initial scans.

Size of initial scan point-cloud (number of points, N)	***Approximate neural gas map size to provide reasonably good results***
2000-3000	60%-100%N
3000-4000	50%-90%N
4000-5000	30%-70%N
5000-6000	20%-60%N
6000-7000	15%-50%N
7000-16000	10%-40%N

The table shows that according to the number of points in the initial scan, *N,* one can identify a map size, as a percentage of this number of points that ensures good modeling results. The larger the initial scan is (within the upper limit of about 16000 points), the lower map size is required to capture the details. An increase in the percentage within the shown range brings better results, but in longer time. The geometrical distribution of nodes should be balanced in that a similar number of neurons should ideally be used along the two dimensions of the neural gas map. However, networks with non-square maps tend to perform better than those who employ square maps. Normally a difference of 5 to 10 neurons between the length and width of the neural gas map ensures good performance under most circumstances.

In spite of the fact that the ranges proposed for point-cloud sizes and the neural gas map sizes can vary slightly with the characteristics of the objects under study (size of objects, number and size of features), Table 4.5 can be employed as guidelines to choose an adequate neural gas map size according to the required quality of model (accuracy) and training time.

4.3.2.5. Conclusions on Vision Data Acquisition

All the examples considered during this investigation demonstrate the ability of a neural gas map to capture fine details in a sparsely collected point-cloud of objects independently of the sensing technology and the distribution of points. By finding the high density areas in the equivalent neural gas map model rather than in the original point- cloud, the proposed selective sampling procedure can be used to guide a vision sensor to collect only measurements in those regions that are dense in 3D features and are of interest for the improvement of accuracy of the obtained models, saving large amount of less relevant data in the scans. The existent correlation between the number of points in the initial scan and the size of neural gas map also guides the user in choosing an adequate neural gas map size according to the needs of the application, which greatly contributes to automate the scanning process. Further research on the topic is directed towards means to eliminate the constraints of a fixed map size imposed by the neural gas solution.

The study up to now was restricted to vision sensing. The next subchapter shows how a similar procedure can be employed to guide tactile sampling of object surfaces.

4.4. Experimental Results for Tactile Sampling with Force Sensors

A similar approach to the one presented for vision sensors can be used to exemplify the detection of regions for tactile sampling. Starting from an initial sparse scan, the proposed solution tries to identify those regions that are representative for the elastic behavior of objects. This ensures the capability of the application to deal not only with homogeneous objects from the elasticity point of view, but also with non-homogeneous, particularly piecewise-homogeneous objects, as well.

4.4.1. Experimental Measurement Data

The same set of deformable objects presented in chapter 3 are used here, that is: the rubber ball, an example of a homogeneous object; the simple composite object made of a square cardboard box mounted on the top of a covered foam pillow, and the toy triceratops, the latter two being representative examples of objects made of non-homogeneous/piecewise homogeneous elastic materials. For the pillow, the elastic behavior is correlated with a change in geometry, while for the triceratops it is not.

4.4.2. Experimental Results for Tactile Sampling Based on Geometry Only

The study starts by testing the neural gas maps presented for vision sampling, that take as input the geometric point-cloud (x, y, z) of an object, in the context of tactile sampling. A k-means clustering algorithm is applied on the resulting neural gas map to identify those regions that are geometrically different, and that are associated with different elastic behaviors provided that a change in the elastic behavior is correlated with a change in the object's geometry.

For the first two tested objects, the rubber ball and the composite object made from a cardboard box placed on a foam pillow, the 3D point-cloud initially obtained with a sparse scan of its entire surface with the Jupiter laser range finder is modeled by a neural gas network that returns a set of significant sampling points as shown in Fig. 4.14a for the ball for a map size of 4×6 and in Fig. 4.14b for the composite object for a map size of 5×6. Such small map sizes are enough to illustrate the concepts.

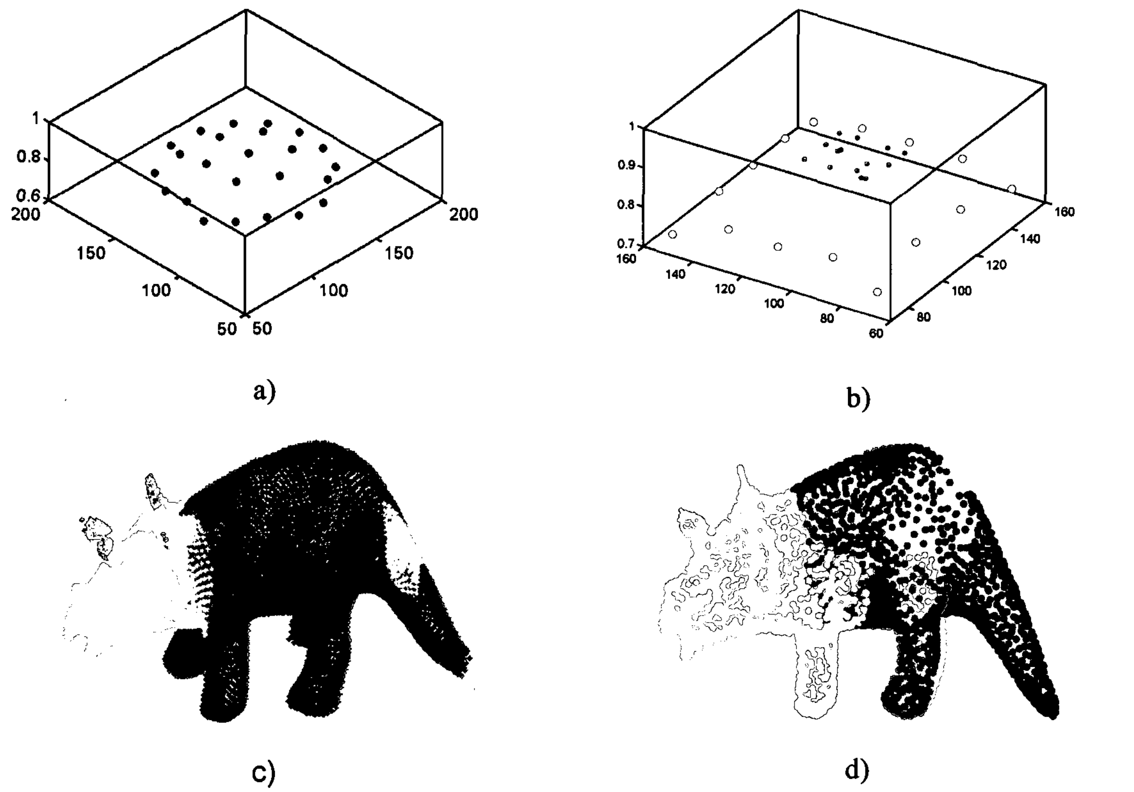


Fig. 4.14. (a) Rubber ball model, (b) composite deformable object model clusters, and (c) triceratops model c) with compliance and d) its clusters based on geometry for a map size of 40×45.

The case presented in Fig. 4.14a is the case of any homogeneous object, where the elastic behavior is constant over the surface of the object. In this case, all the points can be considered part of a single cluster that contains all the sampled points of the object. Several sampling points are selected in order to alleviate the effect of noise into the measurement procedure. The composite object in Fig. 4.14b is an example of object with piecewise homogeneous behavior. The selected sampling points are clustered using k-means clustering in two groups, corresponding to the two different geometric parts. Due to the fact that in this case the change in geometry is correlated with a change in the elastic behavior, the two groups also represent different elastic materials as well. In Fig.

4.14b, the blue points belong to the box, the red ones to the pillow. By extrapolation, in all those cases where the elastic behavior is correlated to changes in geometry (situation that occurs quite frequently), the tactile sampling procedure can be based on geometry only and therefore the same neural gas networks presented before for vision sampling can be used as basis for tactile sampling as well.

However, there are situations in which the elasticity is not correlated to the geometry of the object. The triceratops point-cloud with compliance information is depicted in Fig. 4.14c (each area with the same compliance has the same color), while the neural gas map obtained after training for the triceratops toy is depicted in Fig. 4.14d with the different clusters based on geometry marked in different colors. A denser map is used in this case in order to better visualize the clusters based on geometry. Each part of the triceratops body, being a different geometrical entity is clustered separately. While no error would be introduced in the model by sampling each of these clusters in order to collect tactile data, it is obvious that a penalty in terms of time will occur in this case.

4.4.3. Experimental Results after Addition of Elasticity Measure

To avoid the penalty in terms of time that occurs for those objects where a change in geometry is not necessarily correlated to a change in elastic behavior, it is interesting to explore if the addition of an elastic measure in the neural gas model would improve the behavior of the sampling procedure in such cases. This implies that the initial sparse sampling procedure would probe not only the geometry, but the elastic behavior as well at each of the points. This elasticity measure is computed in different manners for the two types of elasticity data collected in chapter 3, namely the force-displacement curves and deformation profiles, respectively.

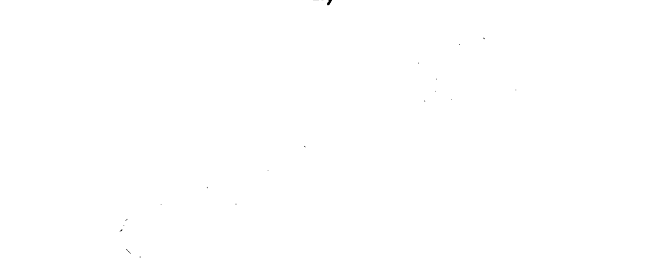
4.4.3.1. Elasticity Measure from Force-Displacement Curves

A simplistic measure of elastic behavior that can be derived directly from the force-displacement curves is the compliance at each probed point. The compliance is the inverse of stiffness and is a measure of the resistance of an object's body to the deformation resulting by applying a force. It corresponds to a constant ratio between force and displacement, similar to the behavior of a linear elastic material under small deformation or a linear spring and can be obtained by approximating a measured force-displacement curve by a line. Such an approximation is widely accepted in many computer graphic simulations where less accurate models can be tolerated.

In this case, the weight vector of the neural gas architecture that represents the geometry of an object and also embeds an elasticity measure consists of 4D coordinates (3D point locations (x, y, z) and the compliance at that point). To present the results, color is used to depict the compliance, the fourth dimension of the neural gas network.

The case of objects with homogeneous compliance is demonstrated for the french-key point-cloud depicted in Fig. 4.15a, with a uniform color representing the constant compliance on the surface of the object.

a)



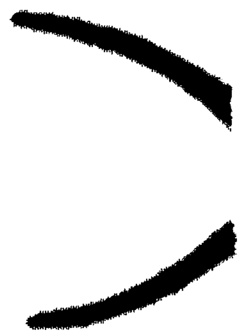
b)

Fig. 4.15. French key model with homogeneous compliance (a) point-cloud (b) neural gas model with compliance for a neural gas map of 15×25.

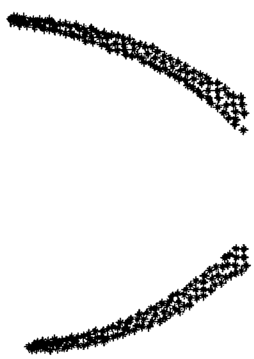
The map size chosen for the neural gas mapping can be relatively small, only large enough to sense subtle changes in the object geometry. The choice of the neural gas map size does not influence in this case the implicit modeling of the compliance, which is encoded in this case with 100% accuracy. Fig. 4.15b shows the result for a map size of

15×25 and 50 iterations of the training algorithm. The neural gas map reproduces correctly in this case both the compliance and the geometry of the object embedded in the point-cloud.

The case of non-homogeneous objects without significant localized changes in compliance is studied for the pliers depicted in Fig. 4.16.



a)



b)

Fig. 4.16. Pliers model with non-homogeneous compliance, without significant localized changes in compliance (a) point-cloud and (b) neural gas model with compliance for a map of 20×25.

The pliers have a plastic handle with a different compliance than the peak (the peak, in metal is denoted by a lighter blue shade; the handle, in plastic, that is slightly smoother is shown in darker blue shade). In this case the boundary line between different elasticity areas is not absolutely correctly delimited unless a sufficiently large point-cloud is available for the modeling. The boundary line is denoted by black points, in the Fig. 4.16b, incorrectly categorized within a quantization error of 0.01.

The case of a heterogeneous/piecewise-homogeneous object with significant localized changes in compliance is studied for the model of the triceratops. The model shows significant changes in compliance in the area of the head and horns of the triceratops. Areas of different elasticity are represented using colors (red – the stiffest region and dark blue - the smoothest region according to the compliance measurements) as shown in Fig. 4.17a.

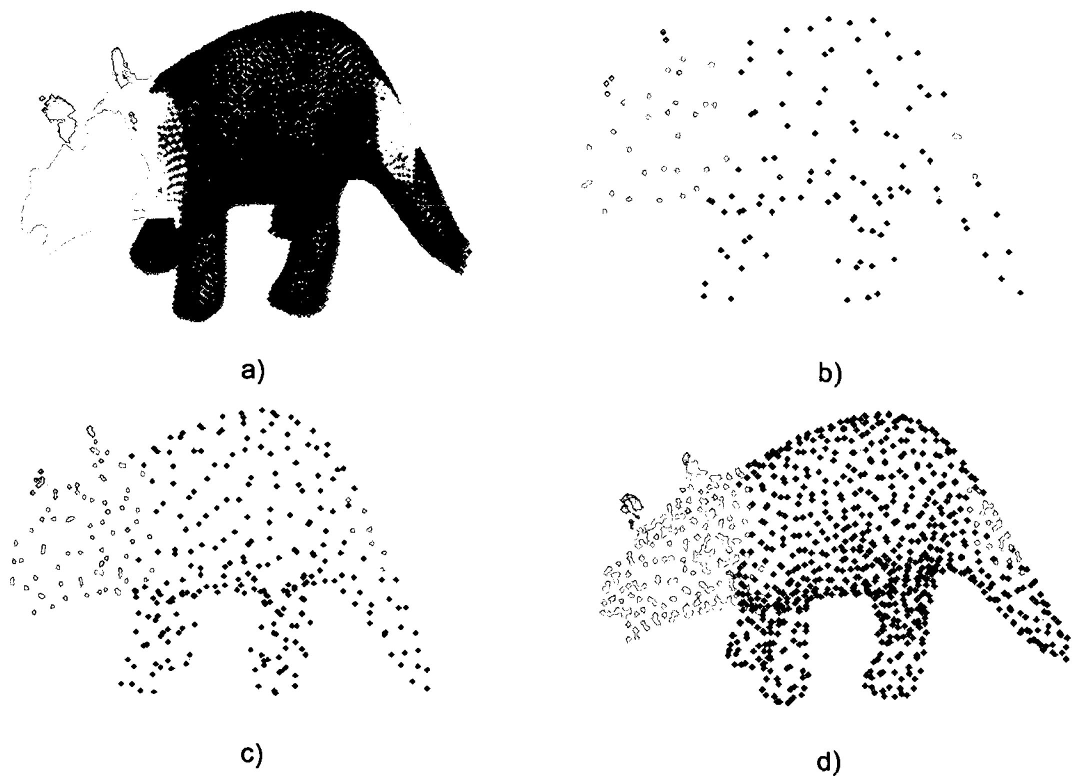


Fig. 4.17. Triceratops model with non-homogeneous/piecewise-homogeneous compliance, with significant localized changes in compliance: (a) point-cloud, and neural gas model with compliance and map size of (b) 10×15, (c) 15×25, (d) 30×45 respectively.

The neural gas map size plays an important role in the modeling procedure in this case, as it did for the accurate geometric modeling as well. While a relatively small size of the map as the one used in Fig. 4.17b is enough to show that there is a significant change of compliance in the area of the horns, it cannot describe accurately the compliance characteristics of the object under study. An increase in the map size reflects better both the geometric and elastic properties, as visually depicted in Fig. 4.17c-d. The increase in the map size comes at the price of computational cost as in the case of geometric modeling. The same guidelines as described in Table 4.5 for choosing the map size according to the number of points in the point-cloud can be employed here as well.

Comparing these results with those depicted in Fig. 4.14c, it can be observed that the model based solely on geometry, apart from considering each part of the body a distinct cluster, misses some details around the neck of the triceratops and in the tail, where the change of elastic behavior is not related to a change in the geometry. These experiments show that an improvement in the modeling resulting occurs when a simplistic measure of elasticity is added to the neural gas as the forth dimension in the input sequence for those objects where the changes in geometry are not necessarily correlated with changes in elastic behavior.

4.4.3.2. Elasticity Measure from Deformation Profiles

Since it has been shown that an improvement in the modeling results can be obtained by the addition of a measure of elasticity, such a measure should be found in the case of the deformation profiles as well. In this case, it is impossible to compute explicitly the stiffness, since there is no explicit value for the displacement available. However, a measure of the displacement can be computed by comparing the deformation profiles when no force is acting on the object and after a force acts in a point of an object, as depicted in Fig. 4.18.

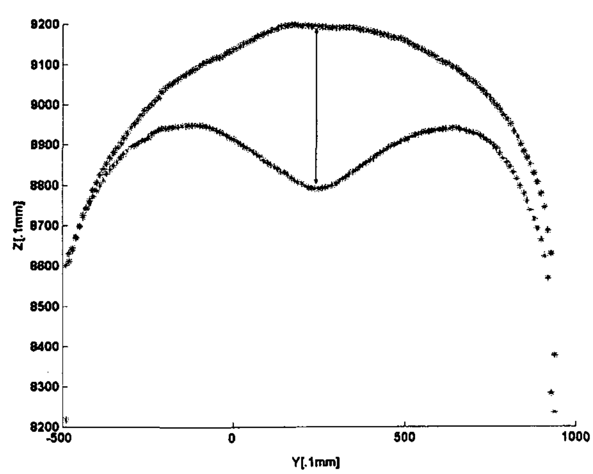


Fig. 4.18. Comparison of non-deformed profile with deformed profile.

The point where the force acts can be determined by computing the difference vector between the two profiles and looking for the largest component of this vector. This largest component can be considered a measure of displacement at the interaction point. The compliance is then computed as the ratio between the displacement and the force acting on the object that causes the displacement, and its average value is stored for each measured point/cluster. Using this compliance, it is easy to build the 4D neural gas map that better reflects the clusters of similar elastic behavior.

The case of objects with homogeneous compliance, represented by the rubber ball, is depicted in Fig. 4.19 with a uniform color representing the constant compliance. As it is expected, in this case the elasticity is correctly embedded in the neural gas map of 4×6. The case of non-homogeneous objects without significant localized changes in compliance is studied for the box-pillow composite object. A map size of 5×6 is used and the results are depicted in Fig. 4.19b, with red points denoting stiffer cardboard material (of which the box is made of) and blue denoting smoother foam material (of which the pillow is made of). In this case as well, the network learns correctly the two different compliances corresponding to the two materials.

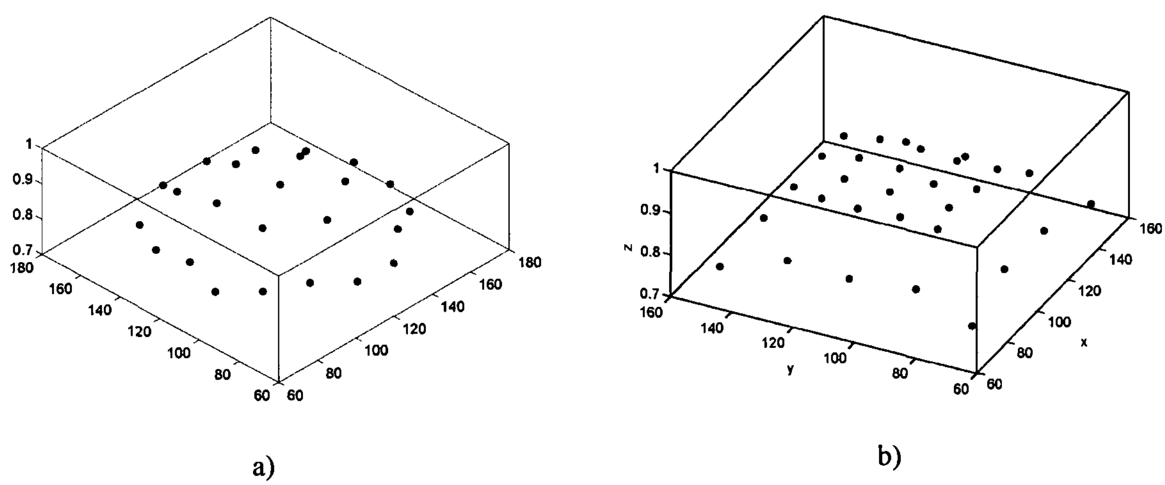


Fig. 4.19. (a) 4D neural gas to model (a) rubber ball and (b) box-pillow composite object.

4.4.4. Selective Tactile Probing

A similar procedure to the one employed for vision sampling can be employed for selective tactile probing. The neural gas map built based on geometrical information in form of (x, y, z) coordinates and on tactile information, in form of compliance, as presented in the previous section, can be employed to detect areas for additional tactile probing. The neural gas map is employed, as in the case of the vision sampling to detect areas of further interest, in this case from the tactile sampling perspective. A similar procedure is used as for vision sampling by employing a Delaunay triangulation to the resulting neural gas output map and by subsequently removing from it triangles larger than a set threshold. The threshold is automatically computed based on the length of vertices for each triangle in the tessellation. However, here additional triangles are removed for all those cases where the three vertices have the same elastic property (compliance). Only those triangles are kept that have different compliance for one or two of the vertices, therefore identifying the areas where changes in the elastic behavior occur.

The triceratops and the pliers objects are used for exemplification as they both present cases of objects where the elastic behavior is not related to the geometry. Fig. 4.20a shows the 6113 points initial point-cloud of the triceratops, the same as in Fig. 4.3, however here the areas of different elasticity are depicted with different colors. For the case of the triceratops, the neural map gas marked with blue dots together with the areas of elasticity change identified with red stars are shown in Fig. 4.20b. The areas where additional scanning is required are shown overlapped on the neural gas map in Fig. 4.20c. Rescanning at higher resolution is performed for each identified region and a multi-resolution elastic model can then be built using the initial sparse model and augmenting it with the high resolution regions of interest. The rescanning procedure can be repeated in several steps to gradually improve the accuracy of the model, as in the case of vision sampling.

In spite of the increase in the dimensionality of the neural map due to the addition of compliance, the same guidelines presented in Table 4.5 can be used to identify an appropriate neural map gas according to the size of the initial scan and similar conclusions can be drawn regarding the relationship between the training time, error and neural gas map size as for vision sampling, in section 4.3.2.4.

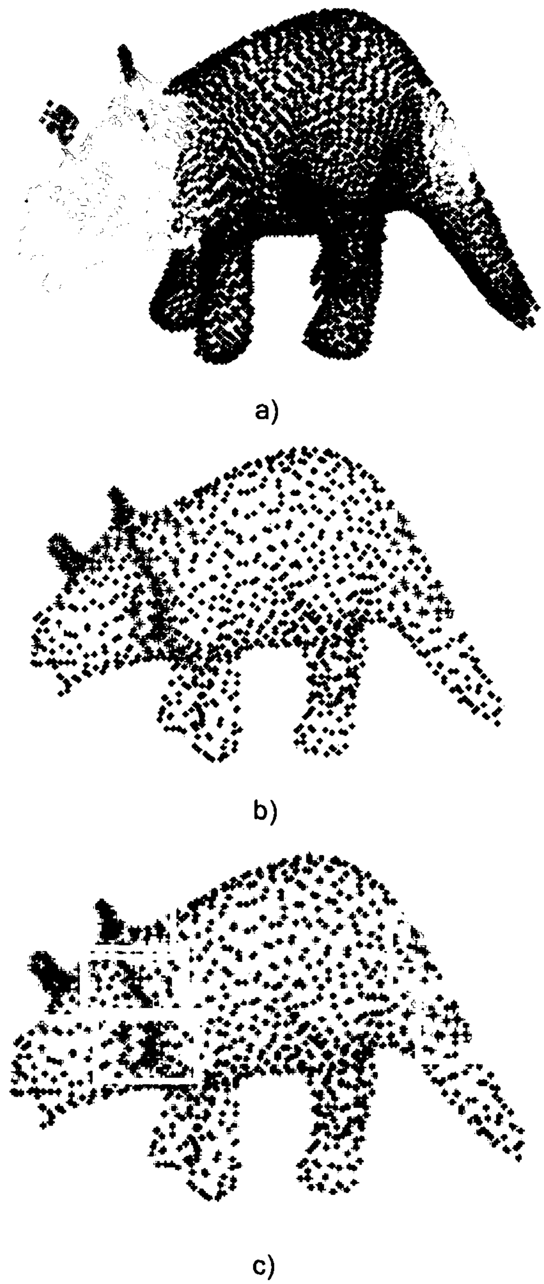
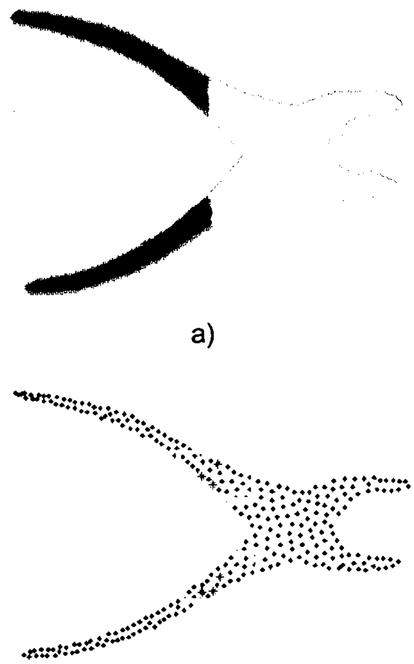


Fig. 4.20. (a) Initial scan of 6113 points of the triceratops with compliance, (b) neural gas model for a map size of 35×45 and (c) detected regions of interest in the map.

Fig. 4.21 presents the detected areas of interest for the pliers. It can be seen that in both cases, the areas that require additional scanning are correctly identified.



a)

b)

Fig. 4.21. (a) Initial scan of the pliers with compliance and (b) neural gas model and detected regions of interest for further sampling.

4.5. Composite Geometric and Elastic Models

4.5.1. Composite Models Framework

After showing how elastic behavior can be modeled in chapter 3 and after defining the selective sampling procedure to collect geometry and elastic data, the next step consists in defining a framework that connects the two parts into composite geometric and elastic models.

In summary, the procedure starts by collecting a fast/sparse scan of the object under study using a laser scanner. The sparse scan serves as input to the neural gas architecture, whose purpose is to further guide the sampling procedure towards the areas that contain relevant information for the geometric model and the elastic behavior of the object under study. The elasticity component of the probing is the compliance described in the chapter 3. This component is introduced as the forth-dimension input parameter in the neural gas architecture, as described in sections 4.4.3 and 4.4.4. The modeling properties of neural gas allow for an incremental re-scanning at higher resolution of different areas of interest as identified in a previous scanning step, by modeling each of the newly scanned areas with another neural gas architecture. The sparse point-cloud can therefore be selectively densified by the addition of increasingly clearer features. By storing the model in each phase of scanning, a multi-resolution object point-cloud is obtained, each re-scanning phase providing a higher point density in the regions rich in features.

The modeling properties of the neural gas network allow for the determination of clusters of similar geometric and elastic properties. These clusters can be imagined as "tactile patches". Therefore the object point-cloud can be now seen not only as a multi-resolution point-cloud, but also as a collection of non-connected tactile patches. There is a difference between what one normally understands by a patch and what tactile patches are. While a normal patch is a stand-alone entity connected to neighboring stand-alone entities, the tactile patch can be actually composed of several stand-alone entities throughout the body of an object that exhibit similar geometric and elastic properties but are not necessarily neighbors. This can be seen in Fig. 4.20a for the case of triceratops. Yellow areas for example are belonging to the same tactile patch.

To capture the elastic behavior of each of these patches, feedforward neural architectures as those proposed in chapter 3 are being used. The same architecture is employed for each cluster to capture the complex relationship between an applied force, its magnitude, its angle of application, its point of interaction, the object pose and its deformation stage/material state (if desired), and the object surface deformation. Each such neural network is associated with the location of the geometric points that belong to the corresponding cluster. According to the point where the interaction occurs, one of the neural networks will be selected to provide the deformation profile of the object under study based on the specific characteristics of the interaction, such as the magnitude and the angle of force applied, the application point, the object's pose and the deformation stage/material state. The application point is projected to the closest point from the model and the network that models the elastic behavior of the cluster to which this closest point belongs will be selected to provide the elastic behavior.

Due to its continuous output property, any of the networks will be able to provide an estimate of the elastic behavior even in those points where the behavior was not probed, therefore eliminating the need for any interpolation of values that normally occurs in any classical model for deformable objects.

To show how this framework can be applied, a complete modeling scenario is further presented.

4.5.2. Example of a Complete Modeling Scenario

The example presented in this context refers to the selective geometric and tactile sampling of an object based on its neural gas map and the modeling of elastic behavior in form of deformation profiles associated with force recordings. The same framework in section 4.5.1 is followed for the case of a piecewise-homogeneous object made of a rigid material (the bottle cap) and an elasto-plastic material (the bottle body), to show how the proposed framework fits for a generalized context.

The modeling scenario is presented for the plastic bottle shown in Fig. 3.50b. A point-cloud of the object is first collected in a raster-like manner to obtain a geometric model of the bottle, as shown in Fig. 4.22.

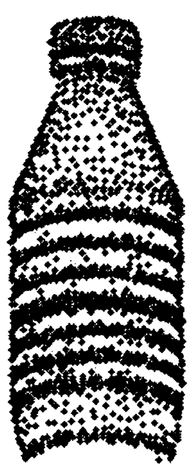


Fig. 4.22. Sparse point-cloud of a plastic bottle.

The sparse point-cloud contains 3420 points, and represents about 20% of the full resolution scan of 17100 points. This point-cloud is provided as input in a neural gas with a map size of 30×45 (selected as per Table 4.5) and the areas for further probing are identified as being those areas with higher density in this map. Fig. 4.23 shows the areas of interest detected in the bottle point-cloud.

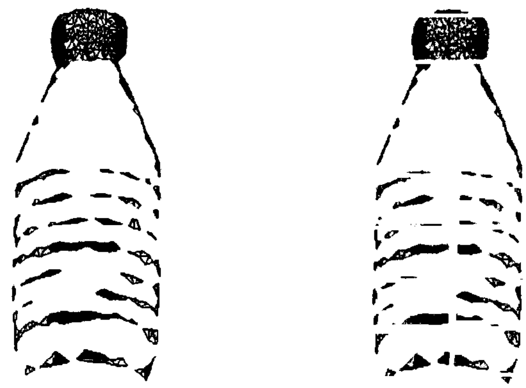
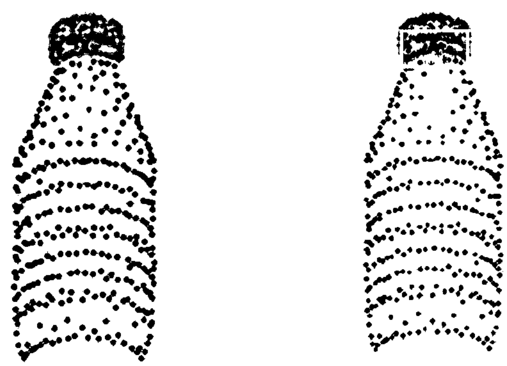


Fig. 4.23. Detected areas for further vision sampling in the 30×45 neural gas map.

If tactile data is also available, the neural gas model can be employed to detect areas of interest from the elastic behavior point of view, as shown in Fig. 4.24. Fig. 4.24b shows the area where changes in the elastic behavior occur.



a) b)

Fig. 4.24. (a) Neural gas model of bottle with compliance information and (b) detected areas for additional tactile sampling.

It can be observed in this case that the area identified for additional tactile probing has already been partially identified using the geometric model only, since the change in elastic behavior is associated with a change in the geometry around the base of the bottle cap. A compressed selectively densified model can be obtained by augmenting the neural map gas size with the areas of high density resulting in a 3215 point model, as shown in Fig. 4.25. Such a model represents roughly 18% of the full resolution scan, implying a reduction of 82% from the full scan.

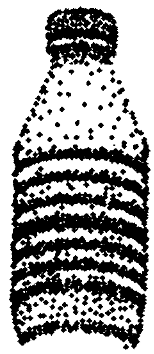


Fig. 4.25. Selectively densified bottle model.

Alternatively, a selectively densified model can be obtained by overlapping the higher resolution scanned areas on the initial pointcloud, resulting in a 5285 point model, that represents a reduction of 70% from the full resolution model. In both cases the reduction in size is significant.

To model the elastic behavior, data is collected separately for the two clusters depicted with red and blue respectively, in Fig. 4.24a. A sample set of collected data for the bottle cap (the cluster denoted with red points in Fig. 4.24a), centered on the point *O* is presented in Fig. 4.26.

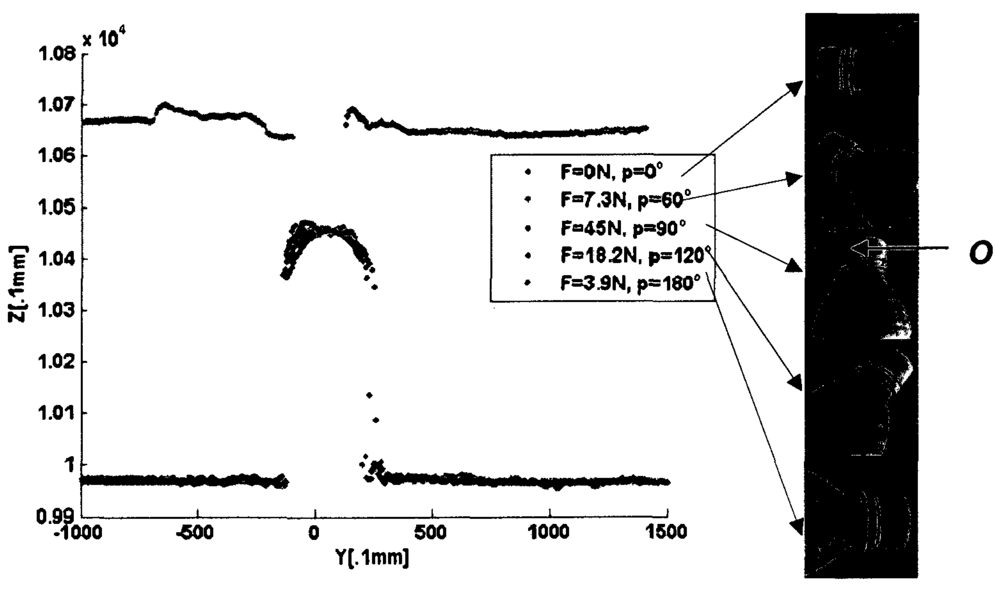


Fig. 4.26. Measurement data for bottle cap for different force magnitudes applied in normal direction (a=90°) for different poses.

The bottle cap is made of a rigid plastic material, therefore the profiles at different forces are similar. A neural network as the one in Fig. 3.41 is employed to model the deformation profiles. The average error reached is 1.66×10^{-5}. The network also generalizes well, as it can be seen in Fig. 4.27 that shows a subset of estimated and real measurement data and a detail of the curves for the bottle cap at different poses.

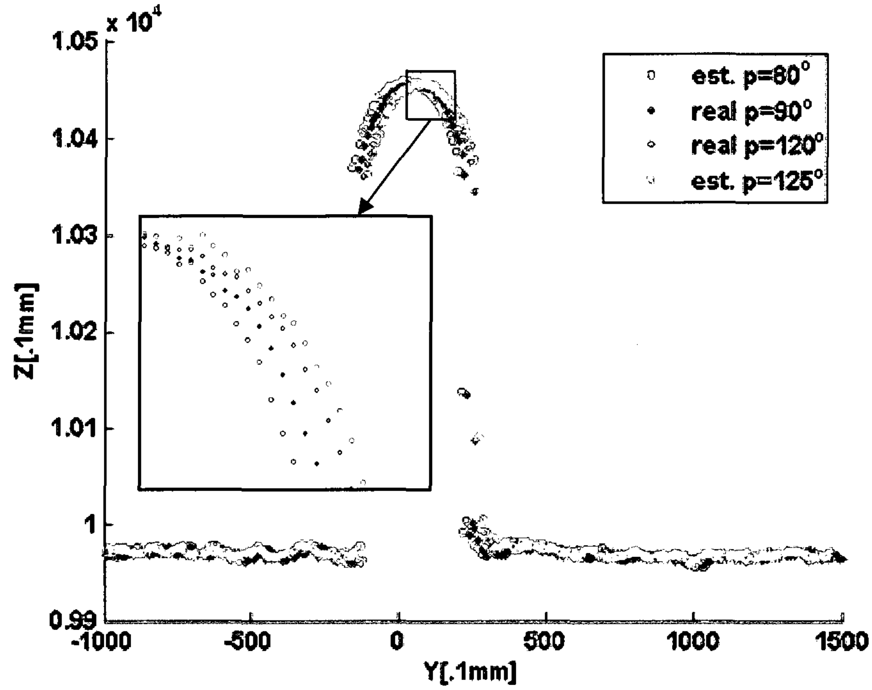


Fig. 4.27. Real and estimated data and detail for bottle cap.

The second cluster is the bottle body, denoted with blue points in Fig. 4.24a. Some results for modeling, together with the network architecture were shown in Fig. 3.54. The error reached for training is 3.5×10^{-5}. To evaluate the generalization capability, the network is tested with data samples that were not part of the training set. The average generalization error is 4.7×10^{-3}. A subset of real and estimated data using the neural network at a fixed pose of *90°*, is depicted in Fig. 4.28. It can be seen that the network places correctly the estimates, for all the stages of elastic, elasto-plastic and plastic deformation of the bottle. For example, in the elastic stage, the estimated displacement for a force F=8N, shown with magenta circles is closer to the non-deformed profile shown with blue dots that represents the non-deformed profile at the same pose, with a slight indentation around the middle point where the interaction occurs. The interaction in this example occurs in the point A (as depicted in Fig. 3.54a).

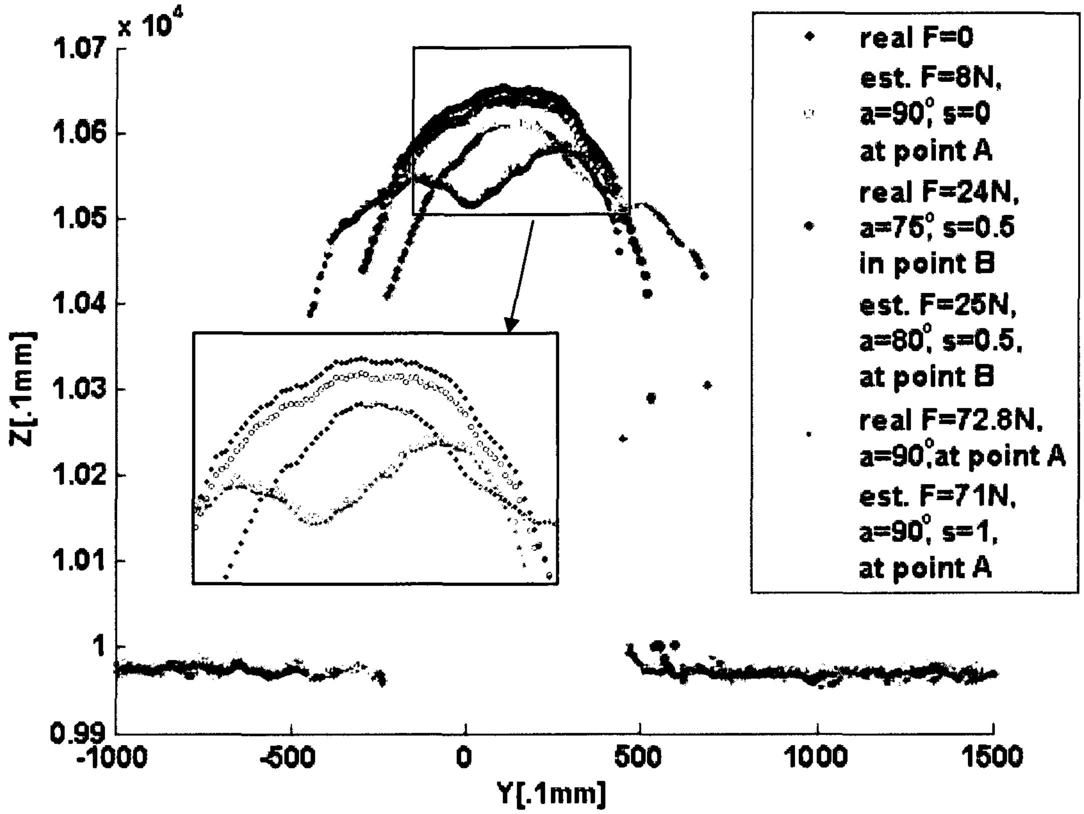


Fig. 4.28. Real and estimated data for bottle body for p=90°, for different force magnitudes and at different angles.

For the point B in Fig. 3.54b, during the elasto-plastic deformation stage, the estimated displacement curve for a force F=25N at a=80° angle (depicted with yellow circles) is turned slightly to the right of the real measurement at angle of 75° and the cavity is slightly deeper and towards the right side of the force of 24N (depicted with red dots). Finally, in the plastic deformation stage, that occurs for forces larger than F=70N, at point A the estimated curve for an applied normal force of 71N, shown with light blue dots is very close to the real measurement, depicted with green dots, of a normal force of 72.8N. In all cases, the network provides correct estimates for data that was not part of the training set.

The interaction with such models, as explained in section 4.5.1, implies the construction of a mapping between each of the two clusters of different elastic behavior and their corresponding network. Each time a point is probed with the force at an angle, for a defined pose, the closest point in the neural gas is determined and from there, the cluster to which it belongs is selected. Knowing the cluster, the corresponding feedforward network provides the estimated displacement profile for the given force.

4.6. Conclusions

The choice of using a self-organizing architecture for selective sampling is justified by its ability to quantize the given input space into clusters of points with similar properties. Neural gas is studied for this purpose and a similar approach is appropriate in vision to collect additional 3D geometry data and for tactile sensing as well. An enhancement is observed for tactile sampling when an additional measure of compliance is used in the mapping, especially for objects where there the change of elastic behavior is not related to a change in the geometry of the object. Composite geometric and elastic models can be built as multi-resolution models with tactile patches, with the behavior of each patch modeled with a feedforward neural network.

The next chapter looks at a slightly different self-organizing architecture to serve the same purpose, but that eliminates the fixed map size constraint imposed by neural gas and therefore the need of a user to interfere in the modeling procedure.

5. Selective Acquisition of Geometric and Elastic Data from 3D Point-Clouds Based on Growing Neural Gas Networks

This chapter tackles the limitation of the neural gas related to the need of the map size to be decided prior to learning, size that constrains the resulting mapping as well as the accuracy of the modeling results. Growing networks add additional nodes into the network structure to support the nodes with highest accumulated errors until a predefined stopping criterion is met, therefore eliminating the need of fixed dimensionality maps. The chapter starts by briefly presenting the general framework of growing networks, it then looks in depth at the growing neural gas architecture, its training algorithm and testing procedure, the influence of its parameters on the modeling results and its use in the context of selective sampling. It shows experimental results for vision and tactile sampling of 3D objects embedded in point-clouds obtained with various types of scanners and compares them with those obtained using neural gas. It finally presents a complete modeling scenario for a composite geometric and elastic object whose acquisition is guided based on a growing neural gas map.

5.1. Introduction

The major limitation of the neural gas network is that the network dimensionality (map size) must be decided prior to learning. As it was presented in the previous chapter, this size constrains the resulting mapping as well as the accuracy of the modeling results obtained by the network. While some guidelines to choose a suitable network size were presented in Table 4.5, the task of choosing a correct map size is not trivial especially when the shape of the object under study is unknown. Moreover, neural gas is unsuitable to continuous learning or the learning of non-stationary datasets [151] due to the fact that the capacity of the network is predefined (through a fixed map size).

Growing networks eliminate some of these limitations. Such networks add additional nodes into the network structure at the position where the accumulated error is the highest and when the number of learning iterations performed is an integer multiple of some predefined value, eliminating the constraint imposed by the fixed capacity. As a node is added another set of iterations is performed before another one is introduced in the network. Therefore the network grows always at the same rate regardless of the way the input distribution is changing. Such networks are also capable to track slowly moving non-stationary datasets. The growth of the network is terminated when a predefined stopping criterion is met (e.g. a minimum error or a network size limit is reached).

There are several growing self-organizing algorithms proposed in the literature. The most known are: the growing grid, the growing cell structures, the growing neural gas [146, 152] and the growing hierarchical SOM [153], just to mention a few. Several of the advantages of such architectures are that they only use constant parameters and that the training is faster when compared to other neural networks and other clustering algorithms. Their proposed applications are for pattern classification, data visualization, combinatorial optimization, image compression and clustering [152].

However, many of these growing architectures are still constrained by predefined network map shape, such as a rectangular grid in the case of the growing grid and the growing hierarchical SOM or *k*-dimensional hyper-tetrahedrons in the case of growing cell structures. Unlike these, a growing neural gas is a network without fixed dimensionality and has no predefined map. In this case the dimensionality of the network depends on the local dimensionality of the data and may vary within the input space. For this reason, growing neural gas seems a good choice for the topic under study in this thesis. As for the neural gas, the growing neural gas will be employed to detect areas rich in features starting from an initial fast scan of different objects under study.

5.2. Growing Neural Gas and Its Use for Selective Data Acquisition

5.2.1. Background Definitions

The growing neural gas is an unsupervised incremental clustering algorithm, whose purpose is to generate a graph structure that reflects the topology of the input data manifold [146]. Unlike neural gas that has no topology and just distributes a number of centers, growing neural gas generates a topology among these cluster centers as well.

There are two related concepts that are important in this context - the Voronoi region and the Delaunay triangulation. These are further explained in this section. Given a set of vectors:

$$\{w_1, w_2, \dots, w_N \mid w_i \in \Re^n\}$$

the Voronoi region V_i of a particular vector w_i is defined as the set of points for which w_i is the nearest vector:

$$V_i = \{x \in \Re^n \mid i = \arg\min_{j \in \{1,\dots,N\}} \|x - w_j\|\}$$

Each Voronoi region V_i is a convex area. The partitions formed by Voronoi polygons are called a Voronoi tessellation. For an arbitrary set of points shown in Fig. 5.1a, the Voronoi tessellation is depicted in Fig. 5.1b.

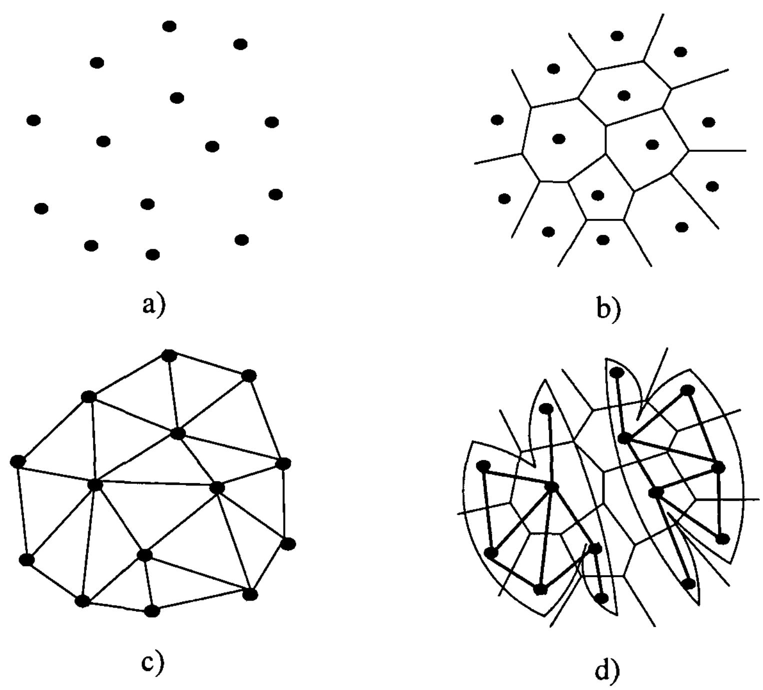


Fig. 5.1. (a) An initial point set, (b) its Voronoi tessellation, (c) the corresponding Delaunay triangulation and (d) induced Delaunay triangulation (adapted from [146]).

The Delaunay triangulation is obtained by connecting all pairs of points for which their respective Voronoi regions share an edge. Its main characteristic is the fact that it is a triangulation where a circle circumscribing any triangle does not contain any other input points in its interior. Fig. 5.1c shows the corresponding Delaunay triangulation for the same initial set of points in Fig. 5.1a.

The induced Delaunay triangulation is a subgraph of the Delaunay triangulation. It is obtained by masking the original Delaunay triangulation with a data distribution $p(x)$, shown in gray in Fig. 5.1d. Two centers are only connected if the common border of their Voronoi polygons lies at least partially in a region where $p(x)>0$ [146]. The resulting induced Delaunay triangulation is depicted in Fig. 5.1d by thick lines. The concept of induced Delaunay triangulation is important, as the growing neural gas builds such a graph as a result of learning (adaptation). The dimensionality of the graph depends on the dimensionality of the input data.

5.2.2. Growing Neural Gas Architecture and Algorithm

As it was already mentioned, growing neural gas builds a topology. This topology is generated using competitive Hebbian learning [154], which inserts an edge between the two closest nodes. The closeness is measured in Euclidian distance from an input signal/vector. There are no restrictions on the topology. Arbitrary edges are allowed and the topology can have different dimensionalities in different parts of the input space. The resulting graph is an induced Delaunay triangulation. This induced Delaunay triangulation has been shown to optimally preserve topology in a very general sense [154].

The growing neural gas algorithm can be described as follows [146, 152]: new nodes are added every λ iterations, to support the node with the highest local accumulated error. For each input signal/vector presented to the network, two best matching nodes are selected, whose weights are the closest to the input, based on Euclidean distance. A neighborhood connection is created between them if the connection does not already exist and its age is set to 0. The position of these nodes and the ones of the topological neighbors of the winner unit are moved such that they better fit the input. All edges that are not used increase in age and if the age exceeds a threshold (a_{max}), the corresponding edges are deleted. Any node that has no edge connections is removed as well. After λ iterations, a new node is added to support the node that has accumulated the highest error in the previous steps. The new node is placed between the node with the highest error and one of its neighbors that has the next highest error. The algorithm continues until some stopping criterion is met.

Based on the notations used in section 4.2.1, the growing neural gas algorithm can be described formally as it follows [146, 152]:

1. The network is initialized with two units c_1 and c_2:

$$S = \{c_1, c_2\}$$

with the corresponding reference vectors $w_{c_1}, w_{c_2} \in \Re^n$ chosen randomly according to a probability density function $p(x)$ or chosen from a finite set D and the connection set, C, is initialized to the empty set.

$$C \subset S \times S, C = \phi$$

2. A random input signal x is generated based on $p(x)$ or chosen from a finite set D and presented to the network.

3. The winner and second-nearest units s_1, $s_2 \in S$ are determined by:

$$s_1 = \arg\min_{c \in S} \|x - w_c\|$$

and

$$s_2 = \arg\min_{c \in S \setminus \{s_1\}} \|x - w_c\|$$

4. If s_1 are s_2 are not connected, a connection is created and added to the connection set:

$$C = C \bigcup \{s_1, s_2\}$$

and the age of the connection between them is set to 0.

$$age(s_1, s_2) = 0$$

5. The squared distance between the input signal and the winner is added to a local error variable:

$$\Delta E_{s_1} = \|x - w_{s_1}\|^2$$

6. The reference vectors of the winner and its direct topological neighbors are adapted by fractions ε_b and ε_n respectively of the distance to the input signal:

$$\Delta w_{s_1} = \varepsilon_b (x - w_{s_1})$$

$$\Delta w_{s_i} = \varepsilon_n (x - w_{s_{i1}}), \forall i \in N_{s_1}$$

where N_{s_1} is the set of direct topological neighbors of s_1.

7. The ages of all edges emanating from s_1 are incremented by 1:

$$age(s_1, i) = age(s_1, i) + 1, \forall i \in N_{s_1}$$

8. All edges with the age larger than a_{max} are removed. If this results in units having no more emanating edges, those units are removed as well.

9. If the number of the input signals generated so far is an integer multiple of λ, then a new unit is inserted as follows:

a) Determine the unit q with maximum accumulated error:

$$q = \arg\max_{c \in S} E_c$$

b) Determine the unit f among the neighbors of q with the maximum accumulated error:

$$f = \arg\max_{c \in N_q} E_c$$

c) Add a new unit r to the network and interpolate its reference vector for q and f.

$$S = S \cup \{r\}, \; w_r = (w_q + w_f)/2$$

d) Insert edges connecting q and f with r and remove the original connection between q and f:

$$C = C \cup \{(r,q), (r,f)\}, \qquad C = C \setminus \{(q,f)\}$$

e) Decrease the error variables of q and f by a fraction α:

$$\Delta E_q = -\alpha E_q, \qquad \Delta E_f = -\alpha E_f$$

f) Interpolate the error variable of r from the ones of q and f:

$$E_r = (E_q + E_f)/2$$

10. The error of all units is decreased:

$$\Delta E_c = -\beta E_c, \quad \forall c \in S$$

11. If a stopping criterion is not met, return to step 2.

5.2.3. Growing Neural Gas Mechanisms

5.2.3.1. Movement of Nodes

The adaptation step towards the input signals/vectors (the step 6 in section 5.2.2) leads to a general movement of all units towards those areas of the input space where signals come from (identified as those inputs x for which $p(x) > 0$) [146]. A similar principle is used for both the winner and its neighbors. The nodes are translated linearly along the scaled difference vector $(x - w_{s_1})$ with an amount specified by ε_b for the winner and ε_n for its nearest neighbors. The further the node is from the input signal, the further it will be translated. The constant ε_b is usually set to a value between [0 1] and ε_n is usually set to a value much smaller than the one of ε_b to ensure a much smoother behavior [155]. The effect of these parameters on the modeling results will be shown in detail in section 5.2.4.

5.2.3.2. Insertion and Removal of Edges

A single connection of the "induced Delaunay triangulation", with respect to the current position of all units [146] is created by the insertion of edges between the nearest and the second-nearest unit with respect to an input signal. Since nodes are moved, it can happen that the Delaunay triangulation becomes invalid. Therefore a mechanism is proposed based on edge "aging" to eliminate those edges that are not part of a valid induced Delaunay triangulation (step 8 in section 5.2.2) because their ending points have moved and other units are present in between them. The mechanism is implemented by resetting to 0 the age of those edges which already exist between the nearest and the second-nearest units (step 4 in section 5.2.2) and a local edge aging (step 7 in section 5.2.2) around the nearest neighbor. The edges that should no longer be part of the triangulation are those that are too old, identified with an age larger than a_{max} (step 8 in section 5.2.2). If the removal of edges results in dead nodes (nodes that have no more emanating edges) these are removed as well. With the insertion and removal of edges, the model makes sure to construct and then track the induced Delaunay triangulation which is a slowly moving target due to the adaptation of the reference vectors [146].

5.2.3.3. Insertion of Nodes

The process of node insertion is guided in the growing neural gas by a local accumulated error computed at each node in the network. The accumulation of squared distances (step 5 in section 5.2.2) during the adaptation helps to identify nodes lying in areas of the input space where the mapping from signals/vectors to nodes causes a large error. To reduce this error, new nodes are inserted in such regions. The insertion of nodes is done at a fixed interval controlled by the parameter λ. As a fixed insertion rate, this parameter has a significant impact on the performance of the algorithm. A too low value usually leads to poor initial distribution of nodes and also increases the risk of nodes to become inactive (they are not moved because they are too far from the inputs) [155]. A too high value on the other hand will result in slower growth. The effect of the parameter λ on the results will be shown in section 5.2.4.

After each λ iterations a new node is inserted between the node with the largest error and its nearest unit with the largest error (step 9a-c in section 5.2.2). This choice is justified by the fact that placing a new node in the median position will decrease the influence (Voronoi region) of each of the initial large error nodes and therefore will contribute to a minimization of their future errors. The errors of both the node with the largest error and its nearest node with the largest error are decreased (step 9e in section 5.2.2) in order to prevent that the next node insertion will always occur in the same region and also to reflect the new influence of the inserted node in the network (the error is spread among the nodes with the largest errors and the new inserted node). The error variable of the new inserted node is interpolated from the error values of the nodes it has been inserted between (step 9f in section 5.2.2). Also a global decrease of errors is performed (step 10 in section 5.2.2) to give recent errors greater influence [155] and keep the level of errors within a reasonable range. The influence of α and β on the modeling results will be shown in detail in section 5.2.4.

The mechanism of node insertion is exemplified for a simplified case in Fig. 5.2 for the steps 9 and 10 in the algorithm.

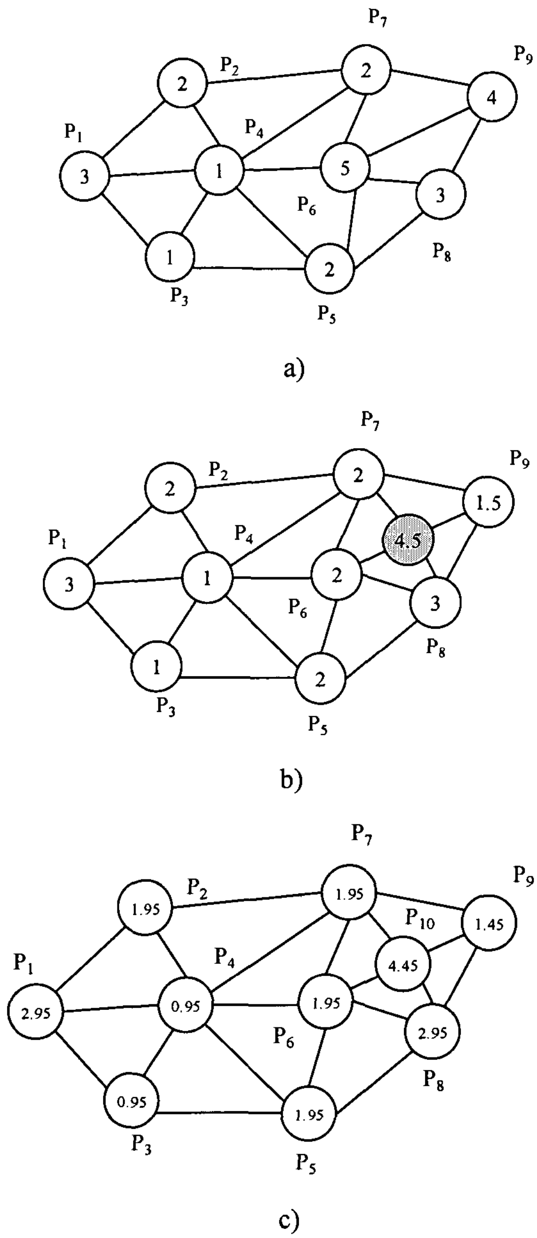


Fig. 5.2. (a) An initial node configuration, (b) insertion of a new node between the node with the largest error and its nearest unit with the largest error and (c) global decrease of errors.

The graph in Fig. 5.2a shows a possible configuration of nodes $\{P_1,...,P_9\}$ after λ iterations, where the values inscribed in each node represent the local accumulated error. In this case the node with the largest accumulated error is the node P_6 (the node q in step 9a, section 5.2.2) and its nearest neighbor with the largest error is P_9 (the node f in step 9b, section 5.2.2). Therefore the insertion of the new node (the node r in step 9c, section 5.2.2) is made between these two nodes and it is shown shaded in Fig. 5.2b. The local error of this node is interpolated from the errors of P_6 and P_9 (as in step 9f, section 5.2.2). The error variables of P_6 and P_9 are decreased (in this case α is arbitrarily set to 0.5). A global decrease of errors (step 10, section 5.2.2) is performed as well (in this case β=0.05) and the results are shown in Fig. 5.2c. The next insertion after other λ iterations will occur in this case between the nodes P_8 and P_{10}.

5.2.4. Influence of Training Parameters on Growing Neural Gas Performance

One of the main advantages of growing neural gas is the fact that all its parameters are constant. Even though several good choices for some of the parameters are suggested in the literature such as: ε_b=0.05, ε_n=0.0006, α=0.5, β=0.0005 [146, 152], a thorough analysis is performed to show an appropriate range of values for all these parameters for the modeling scenarios considered in this thesis.

5.2.4.1. Influence of Movement Parameters ε_b and ε_n

The ε_b and ε_n parameters control the amount the winner and its nearest neighbors are translated linearly along the scaled difference vector $(x-w_{s_1})$ as described in 5.2.3.1. The parameter ε_b is used for the winner node and ε_n for its neighboring nodes. While performing experiments, the parameters are set as follows: when ε_b is tested ε_n=0.0006, α=0.5, β=0.0005, λ=5 and a_{max}=10, and when ε_n is tested, ε_b=0.05, α=0.5, β=0.0005, λ=5 and a_{max}=10. The best choice for the constant ε_b is in the interval [0.025 0.4], as it can be seen in Fig. 5.3 that shows the influence of ε_b on the average error computed for all the objects presented in section 4.3.1. The average error is computed as an average local error as per step 10 in section 5.2.2. For values of ε_b out of this interval, the network is not able to capture the features of the objects under study.

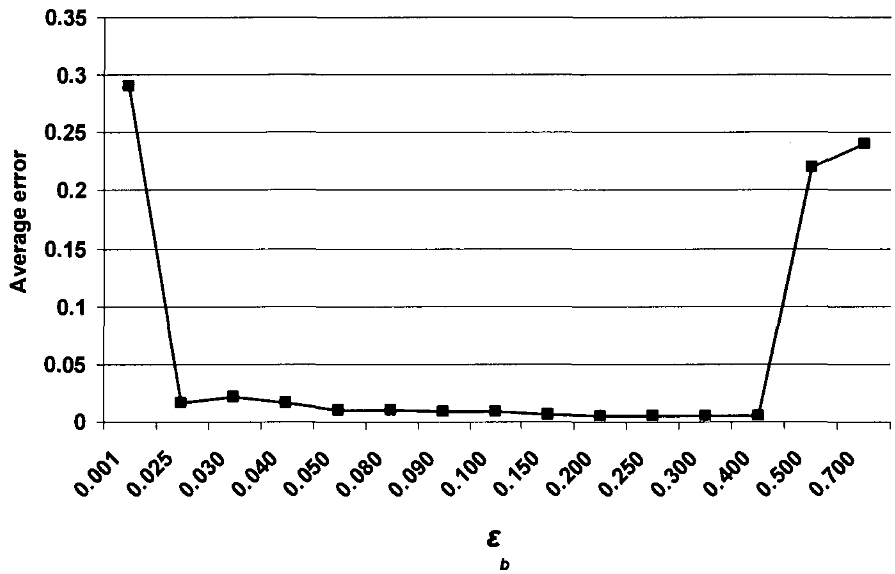


Fig. 5.3. Influence of ε_b on the average local error.

Fig. 5.4 shows that the value of ε_b, while kept in the interval [0.025 0.4] does not significantly affect neither the final map size, nor the time the algorithm takes to create the topology.

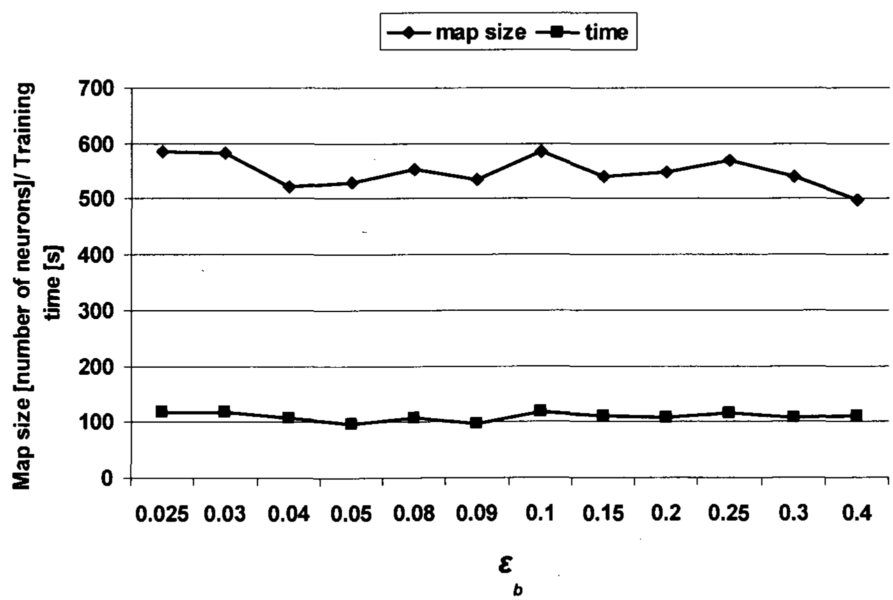


Fig. 5.4. Influence of ε_b on the map size and training time.

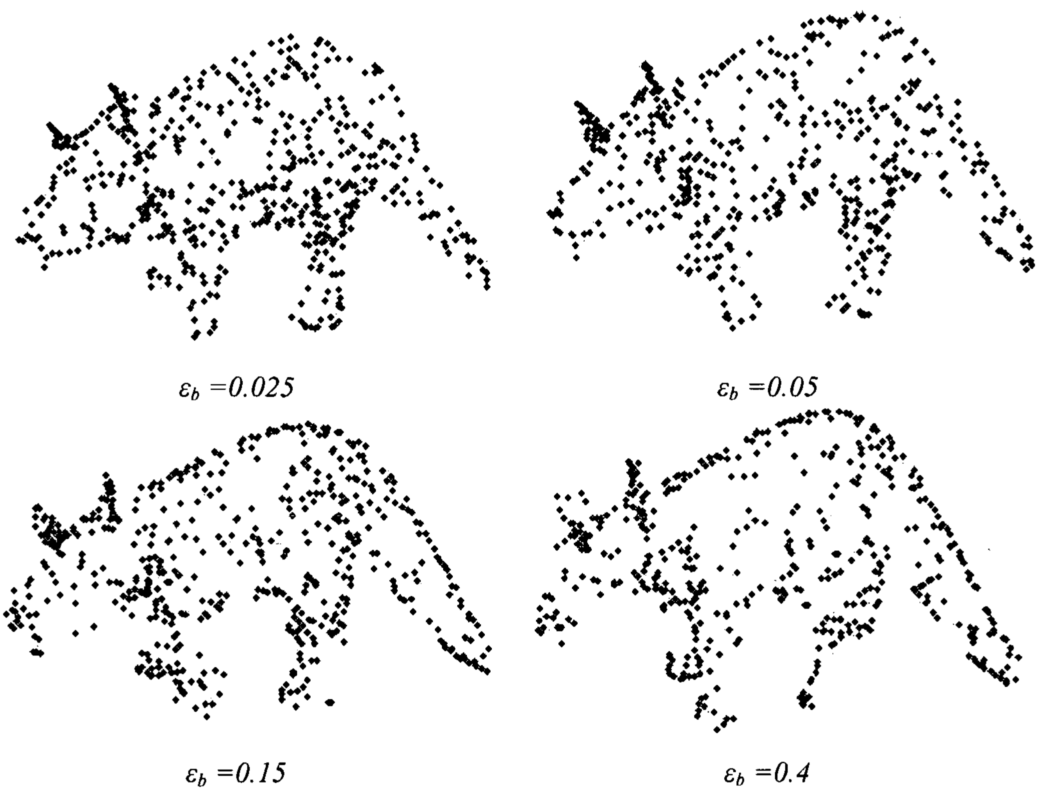


Fig. 5.5. Training results for different values of ε_b, for an initial 4075-point-cloud and λ=5, a_{max}=10, α = 0.5, β=0.0005, ε_n=0.0006.

Fig. 5.5 shows the modeling results for an initial fast scan of 4075 points of the toy triceratops, the same as in section 4.3.1. It can be noticed that the best modeling results are obtained for values of ε_b between 0.025 and 0.05, which is true for the other modeled objects as well, but not presented here for space concern. When compared to Fig. 5.4, the shortest training time combined with a relatively small map size are obtained around a value of ε_b =0.05. Therefore this value will be used for all the experiments.

For the constant ε_n a value much smaller than the one of ε_b is usually chosen. Values with an order of magnitude two to three times smaller work better. Fig. 5.6 depicts the evolution of the average error for different values of ε_n. The error is plotted only for the interval over which reasonable results in modeling occur.

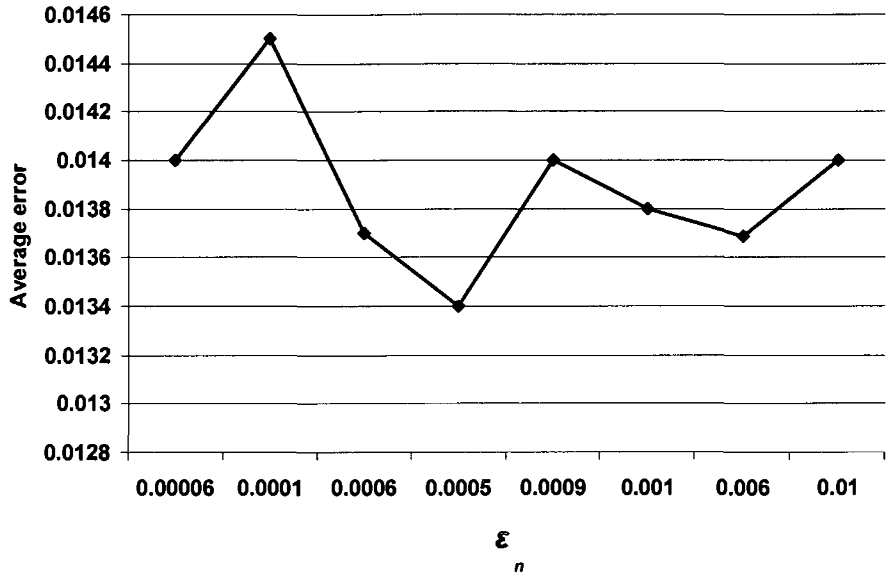


Fig. 5.6. Influence of ε_n *on the average local error.*

In terms of map size and training time, it can be noticed that there is no significant change in behavior for different values of ε_n, similar to the case of ε_b, as depicted in Fig. 5.7. However, both the map size and training time are slightly smaller around the same value ε_n=0.0005.

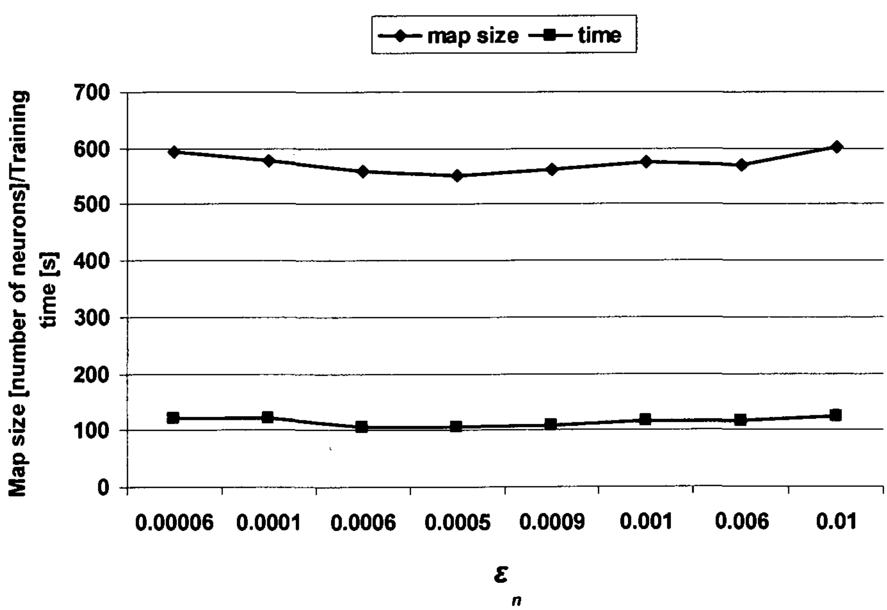


Fig. 5.7. Influence of ε_n *on the map size and training time.*

The modeling results for any value of ε_n between 0.00006 and 0.001 are visually good as demonstrated in Fig. 5.8. However, due to the lower error, map size and training time, a value of ε_n =0.0005 will be used in our experimentation.

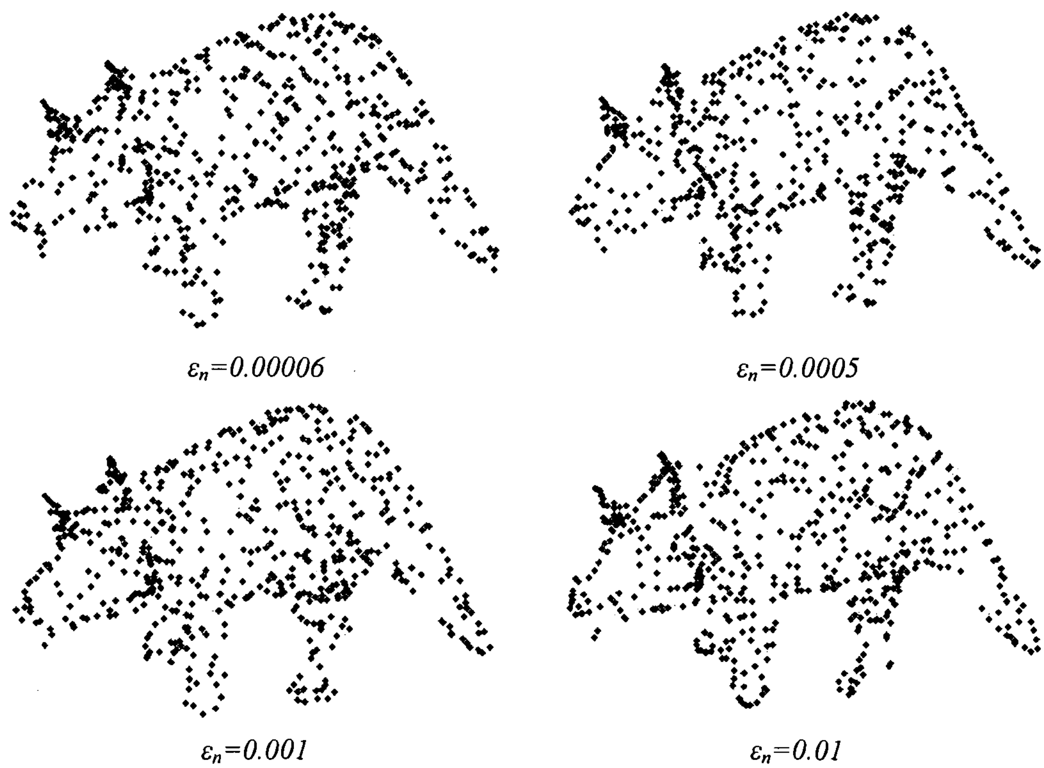


Fig. 5.8. Training results for different values of ε_n, for an initial 4075-point-cloud and λ=5, a_{max}=10, α = 0.5, β=0.0005, ε_b=0.05.

5.2.4.2. Influence of Error Decrease Parameters α and β

Parameters α and β control the proportion of decrease in the error of the node with the largest error and its nearest neighbor with the largest error (step 9e in section 5.2.2) and the global decrease of all errors variables (step 10 in section 5.2.2) respectively. The value of α is usually chosen two orders of magnitude larger than the value for β [146].

For the studied objects, the error decreases with an increase in the value of α and is constantly low for values of α between 0.5 and 0.6, as it can be noticed in Fig. 5.9.

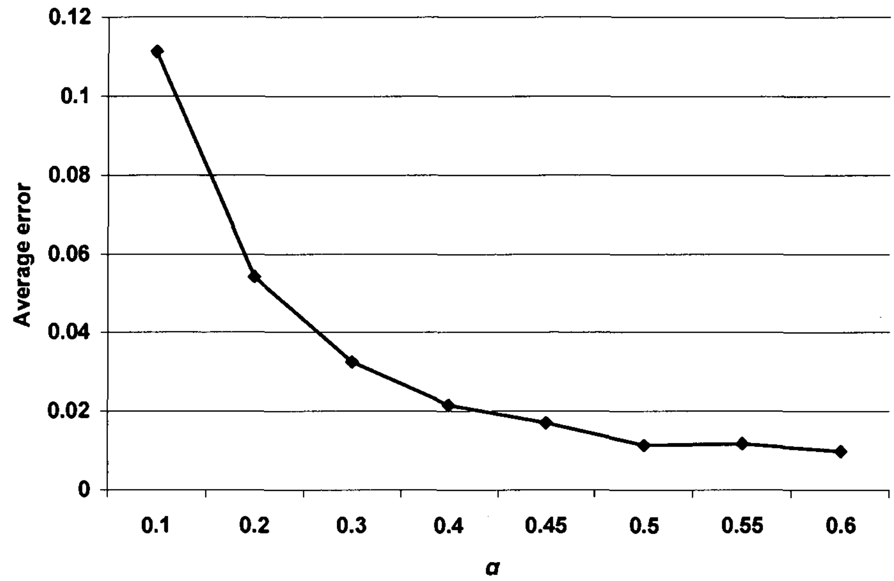


Fig. 5.9. Influence of α on the average local error.

Fig. 5.10 and Fig. 5.11 show the influence of α on the map size, training time and the visual quality of the modeling results.

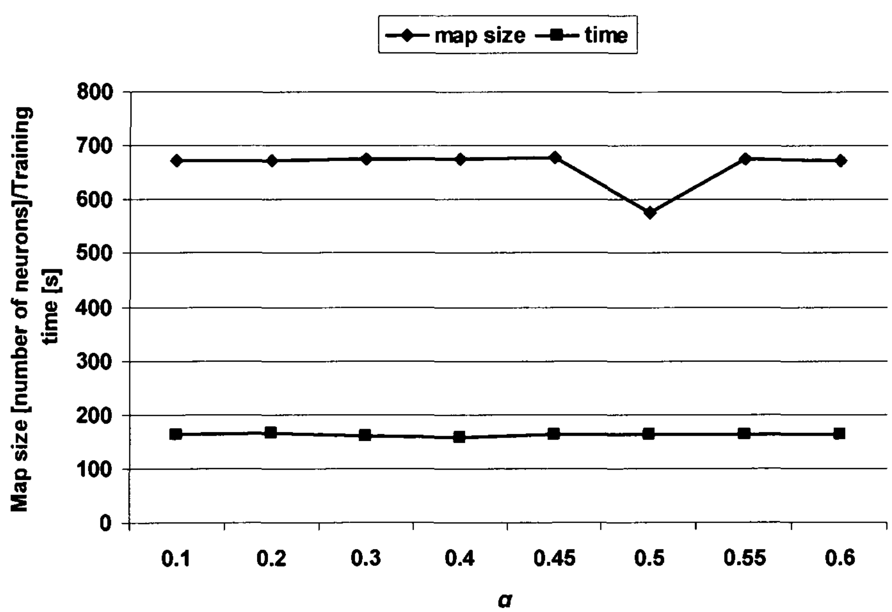


Fig. 5.10. Influence of α on the map size and training time.

While the training time is relatively constant for all the values in the range 0.1-0.6 as seen in Fig. 5.10 and there is no major visual difference between the obtained results in Fig. 5.11 (even though the model for α=0.5 is slightly better), it can be noticed a drop in the map size around α=0.5. All these considered, a value of α = *0.5* is chosen for modeling.

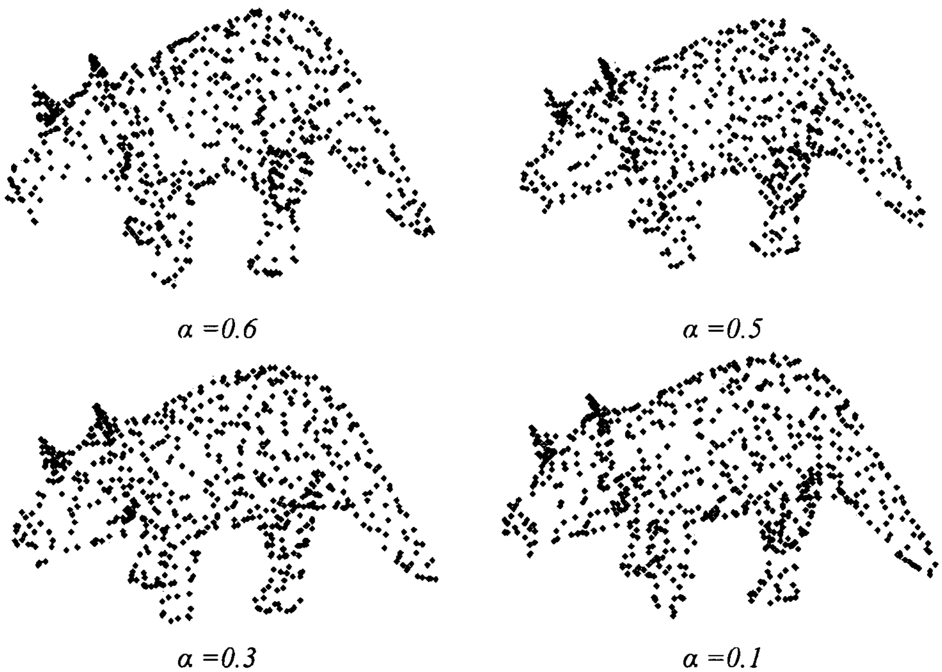


Fig. 5.11. Training results for different values of ε_n, for an initial 4075-point-cloud and

λ=5, a_{max}=10, β=0.0005, ε_b=0.05 ε_n=0.0005.

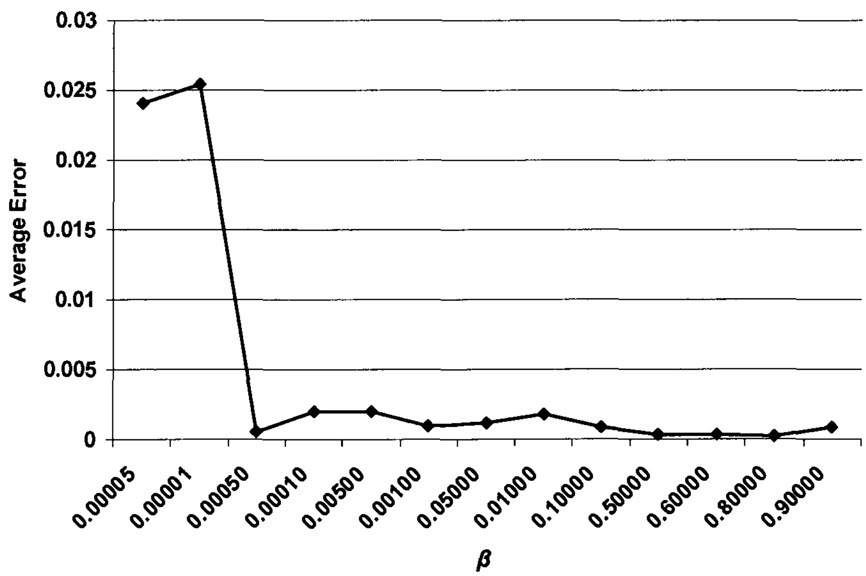


Fig. 5.12. Influence of β on the average local error.

Following the same procedure, the error, map size, training time and visual quality for different values of β are studied in Fig. 5.12 to Fig. 5.14. The study of average error in Fig. 5.12 shows no significant differences in behavior for any value between 0.0005 - 0.9. The value of the average error is smaller for 0.0005 and between 0.5 - 0.9.

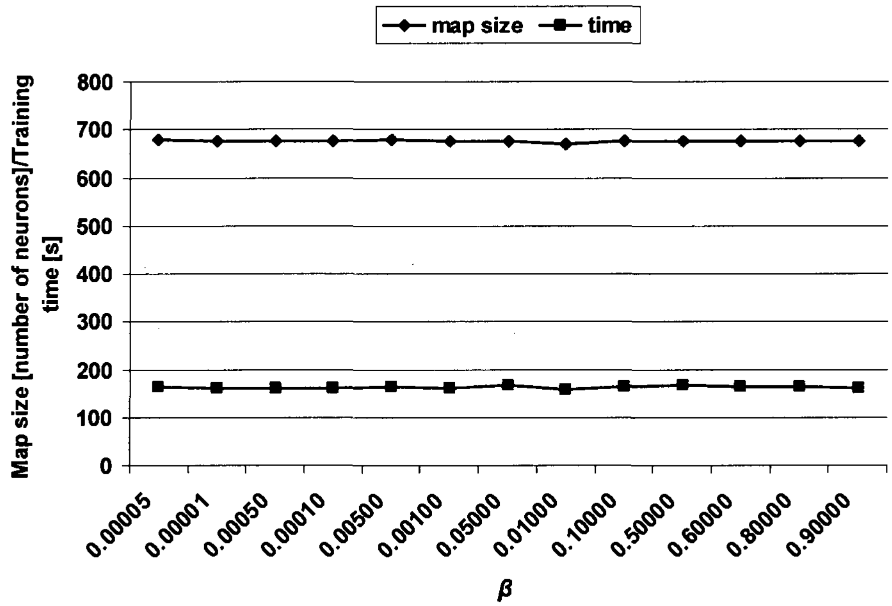


Fig. 5.13. Influence of β on the map size and training time.

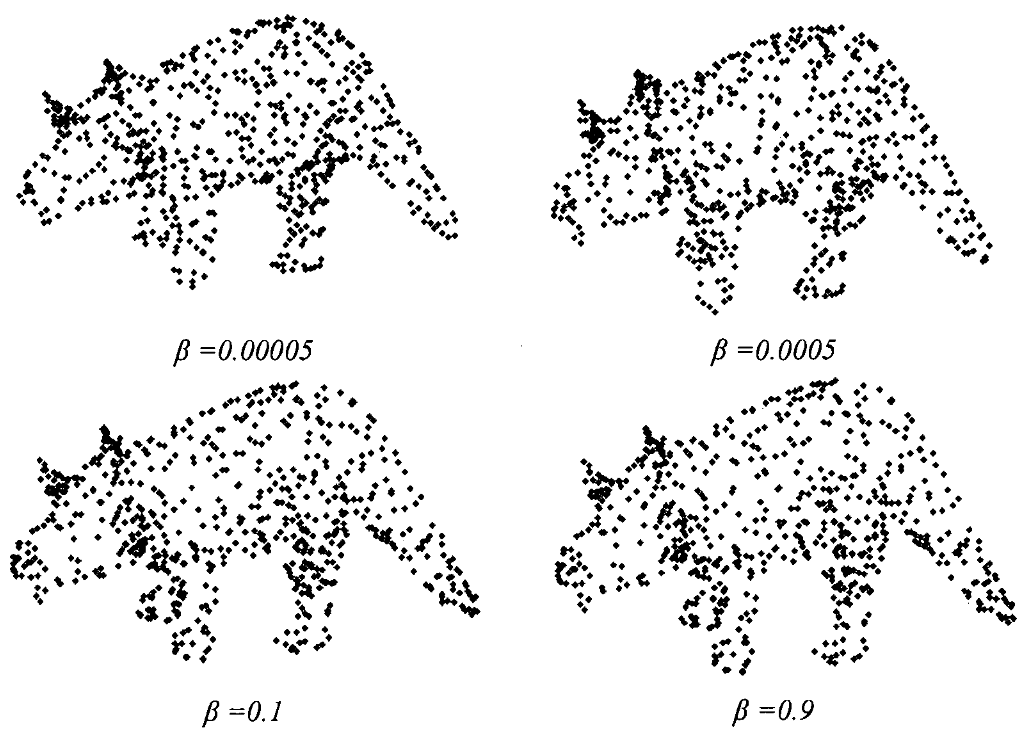


Fig. 5.14. Training results for different values of β, for an initial 4075-point-cloud and $\lambda=5$, $a_{max}=10$, $\alpha=0.5$, ε_b=0.05, ε_n=0.0005.

Fig. 5.13 and Fig. 5.14 demonstrate that no significant difference in map size, training time and visual quality appear neither. The value of β=0.0005 is selected because from the all the values that ensure minimal error, this is the one furthest in magnitude from the value of α and furthermore, it is the one suggested in the literature [146, 152]. Comparing the values of ε_b, ε_n, α and β obtained experimentally for the objects under study with the ones suggested in the literature, it can be concluded that the suggested values for parameters are the same or very close to the ones obtained experimentally.

5.2.4.3. Influence of Age Threshold Parameter a_{max} and of the Number of Iterations λ after which a new node is inserted

The parameter a_{max} is a threshold parameter that controls the removal of "old" edges in the induced Delaunay triangulation created during adaptation (step 8 in section 5.2.2). The study of the influence of a_{max} for fixed values of $\lambda=5$, $a_{max}=10$, $\alpha=0.5$, β=0.0005, ε_b=0.05 and ε_n=0.0005 is presented in Fig. 5.15 to Fig. 5.17.

Fig. 5.15 shows that the average error tends to decrease for a_{max} between 10 and 30, with the smallest reached error around 30 and that the value of error remains relatively constant for any tested values of a_{max} between 40 and 500.

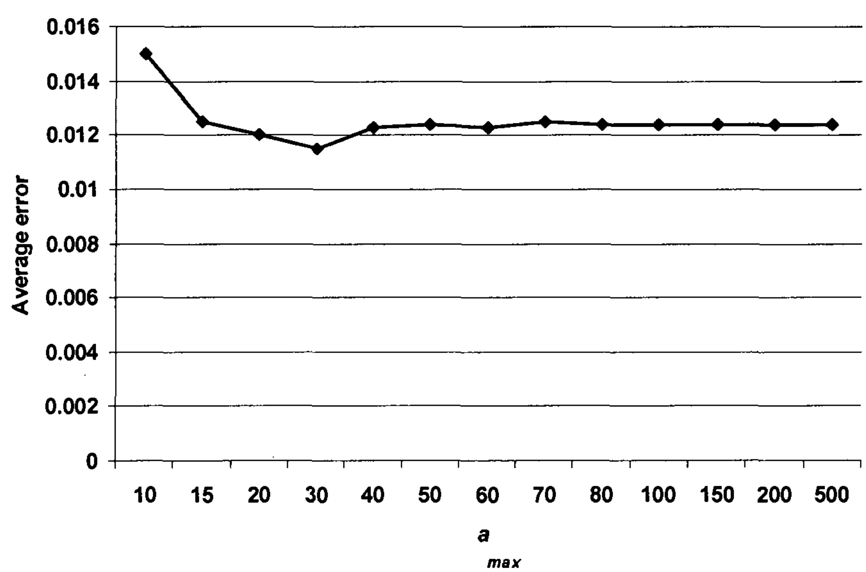


Fig. 5.15. Influence of a_{max} on the average local error.

When comparing the map size and the training time, it can be seen in Fig. 5.16 that both the map size and the training time increase for values between 10 and 30 and remain almost constant after 40 as well.

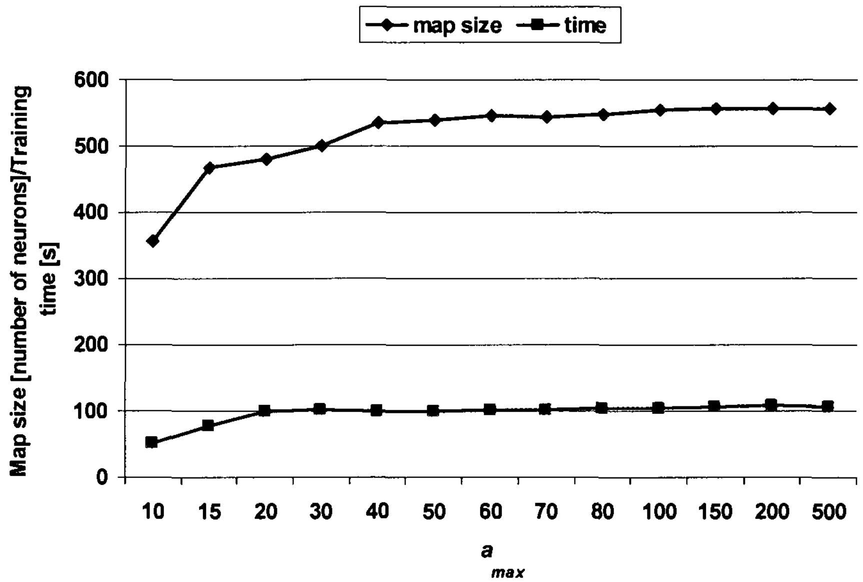


Fig. 5.16. Influence of a_{max} on the map size and training time.

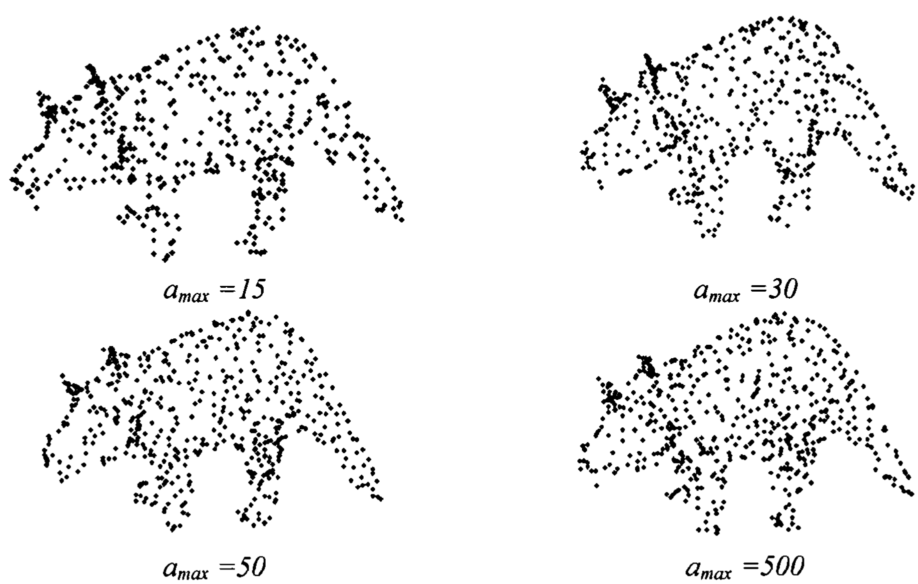


Fig. 5.17. Training results for different values of a_{max}, for an initial 4075- point-cloud and λ=5, α=0.5, β=0.0005, ε_b=0.05, ε_n=0.0005.

Fig. 5.17 depicts modeling results for the triceratops model for different values of a_{max}. As it can be seen, while there are no significant overall differences in the visual quality, the quality for a_{max} =30 is slightly better than the one for a_{max} =15, as the features are slightly better defined. Since for a_{max}=10 the error is quite high as seen in Fig. 5.15, experiments will be performed for values of a_{max} between 15 and 30, particularly a_{max}=15, 20 and 30. An increase of a_{max} will result in a more accurate model if the user is willing to pay the additional computational time.

The insertion of nodes is done at a fixed interval controlled by the parameter λ. As a fixed insertion rate, this parameter has a significant impact on the performance of the algorithm. A study of the parameter λ for different sizes of the initial fast scan is performed in order to determine suitable values for the modeling. Fig. 5.18 shows the evolution of the error for the triceratops model for different values of λ and different sizes of the initial fast scan.

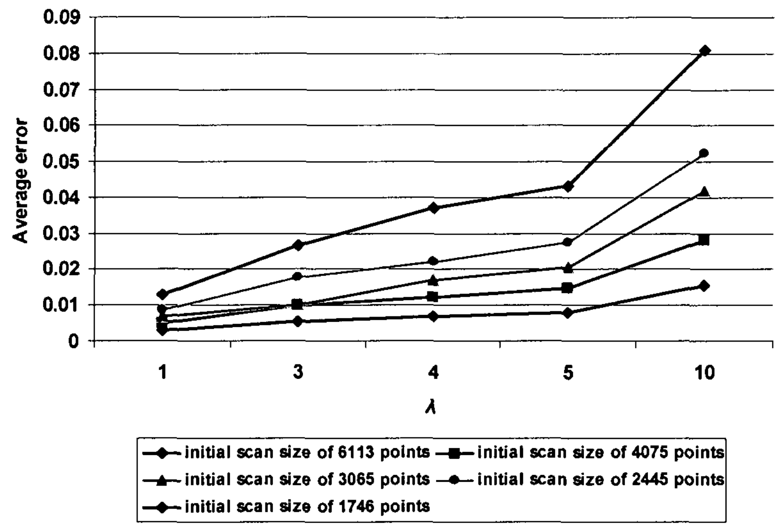


Fig. 5.18. Influence of λ on the average local error for different sizes of the initial fast scan.

Fig. 5.19 and Fig. 5.20 show the evolution of the map size and training time for the triceratops model for different values of λ and different sizes of the initial fast scan. Fig. 5.18 to Fig. 5.20 are plotted only for values of λ between 1 and 10, as for values out of this range, the growing neural gas is not able to capture the features in the point-cloud. The value of error decreases with an increase in the resolution of the initial scan and it decreases as well with a decrease in the value of λ, as shown in Fig. 5.18.

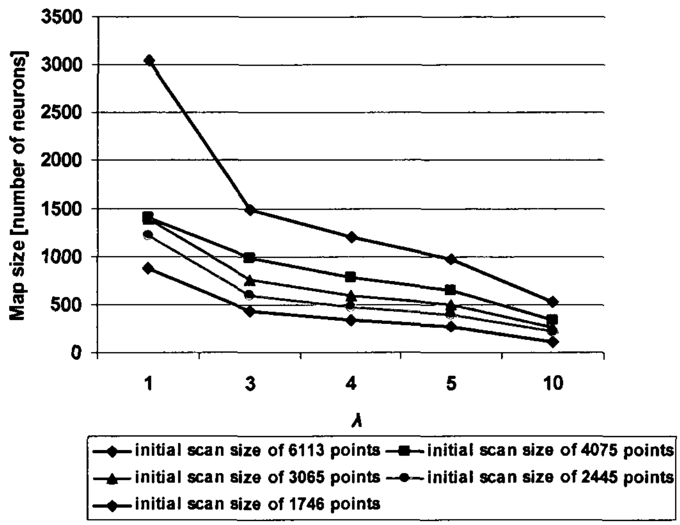


Fig. 5.19. Influence of λ on the map size for different sizes of the initial fast scan.

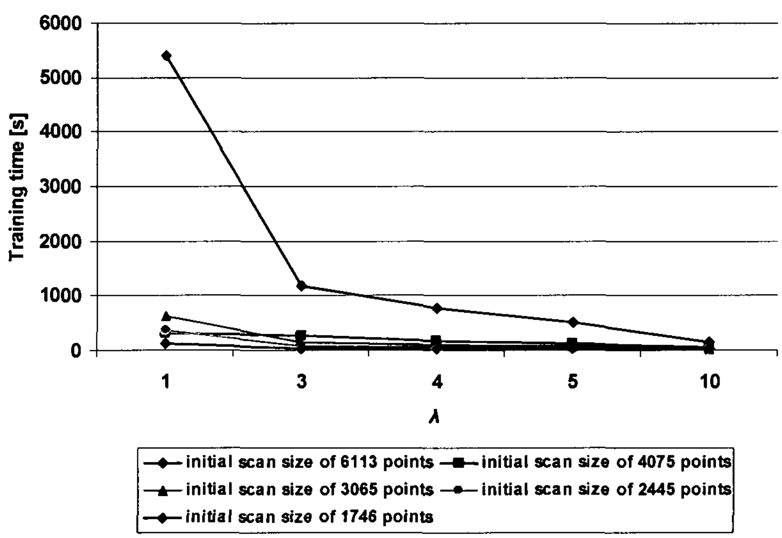
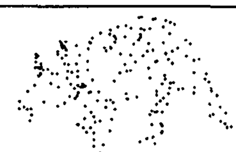
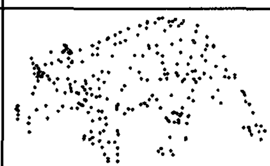
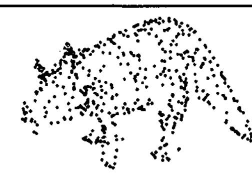
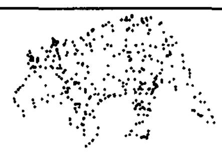
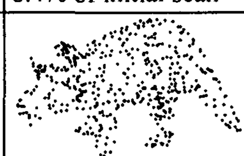
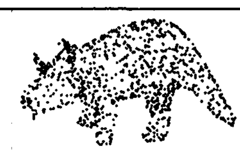
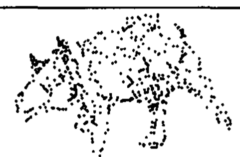
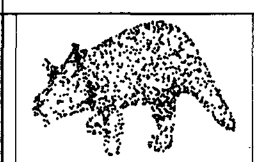
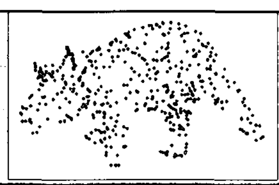
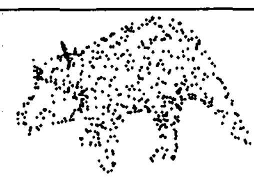
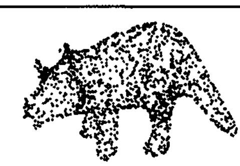
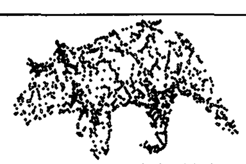
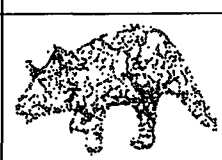
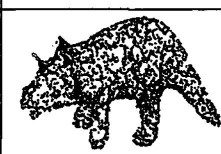


Fig. 5.20. Influence of λ on the training time for different sizes of the initial fast scan.

Therefore in order to build more accurate models, a lower value of λ is desired. However, accuracy means, as expected, larger map sizes and larger training times, as depicted in Fig. 5.19 and Fig. 5.20. It can also be observed in the same figures, that lower resolution initial scans lead to lower map sizes and lower training times. These observations lead to the conclusion that parameter λ can be used as a means to control the accuracy of the modeling results. A lower λ value will lead to more accurate results, but the algorithm will converge slower, resulting in longer training time. This can only be acceptable if the user is willing to pay the supplementary computational cost.

Growing neural gas is not able to capture features in initial scan with less than 2445 points, meaning less than roughly 20% of the full scan in the case considered here. This is similar to the behavior of the neural gas, as it was shown in Table 4.1. Table 5.1 shows that starting from roughly 20% of the full resolution scan, the network is able to model correctly and identify fine features of the triceratops.

Table 5.1. *Evolution of training results for different map sizes and for initial scan sizes between roughly 20% and 50% of the full resolution triceratops model and a_{max}=20.*

Lambda	***2445 points=19.99% of full resolution model***	***3065 points = 25% of full resolution model***	***6113 points = 50% of full resolution model***
λ=10	 8.46% of initial scan	 8.4% of initial scan	 8.5% of initial scan
λ=5	 16.2% of initial scan	 16% of initial scan	 15.9% of initial scan
λ=4	 19.15% of initial scan	 19.2% of initial scan	 19.6% of initial scan
λ=3	 24.6% of initial scan	 24.1% of initial scan	 24.2% of initial scan
λ=1	 45.3% of initial scan	 43% of initial scan	 49% of initial scan

The best modeling results are highlighted with dark grey frames, the good modeling results with light gray frames. It can be observed that for a lower value of λ the growing neural gas models better a lower resolution scan, while a medium low value for λ is enough for the higher resolution scan. Overall, the visual quality improves with the increase in the size of the initial scan. It can also be seen that in the case of lower resolution initial scans, for very low values of λ the resulting model becomes slightly distorted (last row of column 1 and 2). It can also be observed that for the same value of λ, the map size obtained is roughly about the same percentage of the initial scan size.

Another interesting aspect to study is if there is any correlation between a_{max} and λ. The results of this study are presented in Fig. 5.21 - Fig. 5.23, showing the evolution of error, map size and training time for different values of a_{max} and λ.

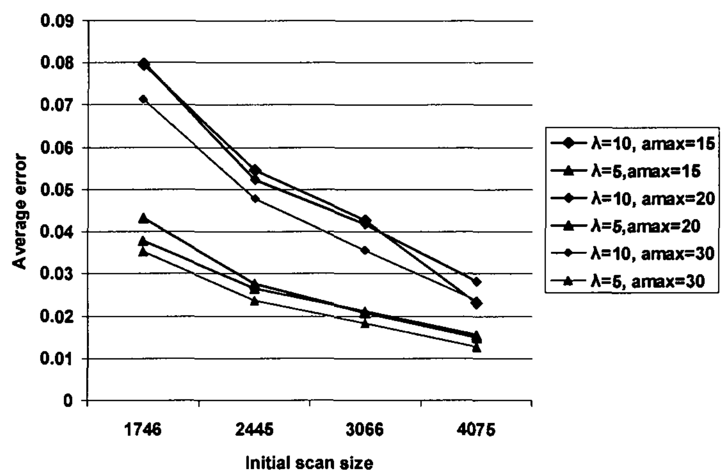


Fig. 5.21. Average error for different values of a_{max} and λ.

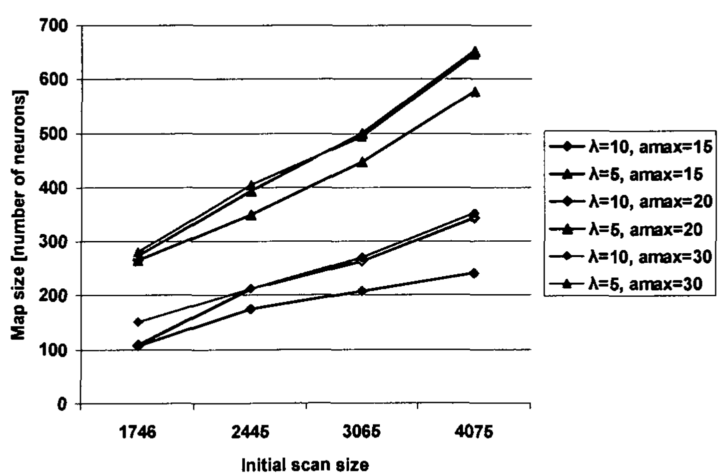


Fig. 5.22. Map size for different values of a_{max} and λ.

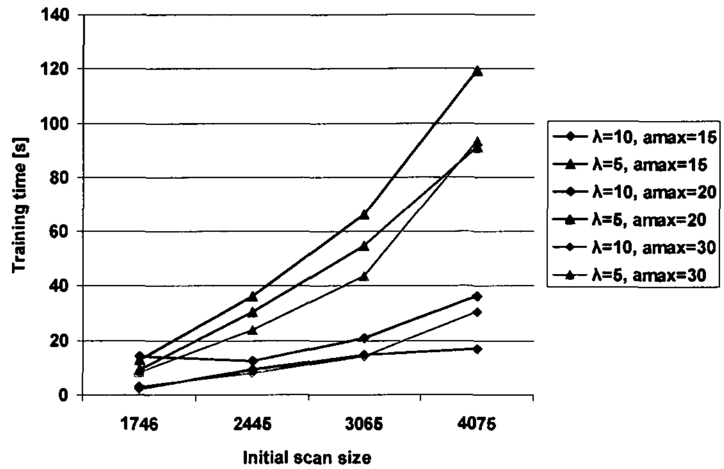


Fig. 5.23. Training time for different values of a_{max} and λ.

While it was already shown before that higher values of a_{max} and lower values of λ, lead to more accurate models, larger map sizes and longer training time, Fig. 5.20-Fig. 5.23 show that the influence of λ is stronger on the modeling results than the one of a_{max} as the curves of error, map size and time tend to follow more the decrease of λ than the increase in a_{max}.

5.2.5 Data Sets and Testing

In the context of this thesis, 3D range data point-clouds of objects and also sets of synthetic data obtained from the interpolation on a higher resolution mesh of a sparse scan are used to train the network. Normalization is employed to remove redundant information from the data set, by a linear rescaling of the input vectors such that their variance is 1. As in the case of neural gas, for each input vector x, its mean and variance are computed using eq. (4.15) and (4.16) and is rescaled according to eq. (4.17) to ensure that the data set presented to the network will have 0 mean and a variance of 1. The local error is used as a quantitative measure to evaluate the obtained results.

5.2.6. Use of Growing Neural Gas for Selective Data Acquisition

In the context of this work, a growing neural gas architecture is employed, as the neural gas in chapter 4, to guide vision and tactile probing by clustering measurements that represent uniform geometry and elasticity regions and therefore direct the sensor toward areas of transitions where higher sampling density is required. Starting from the points collected during a fast scan of an object via an active range finder, the algorithm adds incrementally nodes to a graph, respecting their density and taking the shape of the objects encoded in the point cloud.

5.3. Experimental Results for Geometric Data Acquisition with Vision Sensors

5.3.1. Experimental Measurement Data

The proposed method is first tested in the context of range imaging. Detailed results of experimentations are presented for the same three objects that the neural gas was tested on: the triceratops from section 3.1.1, presented in Fig. 3.1, a foam armchair and a mockup car door, each presenting different sorts of regions of interest, as depicted in Fig. 4.3a and Fig. 4.3c. The procedure to obtain the experimental data is presented in section 4.3.1. Starting from an initial fast, sparse scan of each object, a growing neural gas network is employed to model the data in the point-cloud, in form of (x, y, z) coordinates. Testing is performed for several resolutions of the initial fast/sparse scan in order to identify how the resolution of this scan influences the modeling results and what would be the smallest scan to start with that allows for the modeling of fine features in the modeled objects. All simulations are performed for the values of parameters as found in section 5.2.4: $\alpha=0.5$, $\beta=0.0005$, ε_b=0.05, ε_n=0.0005.

A similar procedure to the one used in the case of neural gas is employed to select areas of interest for scanning. The higher density regions in the growing neural gas map are detected by traversing the nodes successively and checking the length of vertices. An average mean for all these lengths is computed and a threshold is set to this value. All the nodes that are further from each other than the value of the threshold are removed from the model. The remaining points identify those regions that require additional sampling.

5.3.1. Experimental Results

5.3.2.1. The Case of Objects with Relatively Uniform Distributed Sampling Points and Multiple Features

The model of the triceratops is created, as explained in section 4.3.1, starting from a uniformly distributed mesh. It results in a relatively uniformly distributed point-cloud and represents an object with multiple features. Fig. 5.24 shows the enlarged best modeling results for a low resolution initial scan of 3065 points, shown in Fig. 5.24a and for a medium low resolution initial scan of 6113 points, presented in Fig. 5.24b, for $\lambda=3$ and $a_{max}=20$. The growing neural gas network, having as inputs these initial sparse point-clouds, builds a topology that respects the density of points in the initial point-cloud. The resulting mappings are depicted in Fig. 5.24c and 5.24d for the low and medium low resolution initial scans respectively. The high density areas found in these mappings are shown in Fig. 5.24e and 5.24f respectively. As it can be seen in Fig. 5.24f, the high density areas are much better identified and contoured for the higher resolution model. In both cases, the network is able to identify, apart from the contours of the model, the areas around the head and horns of the triceratops as areas worth further scanning, as it is expected.

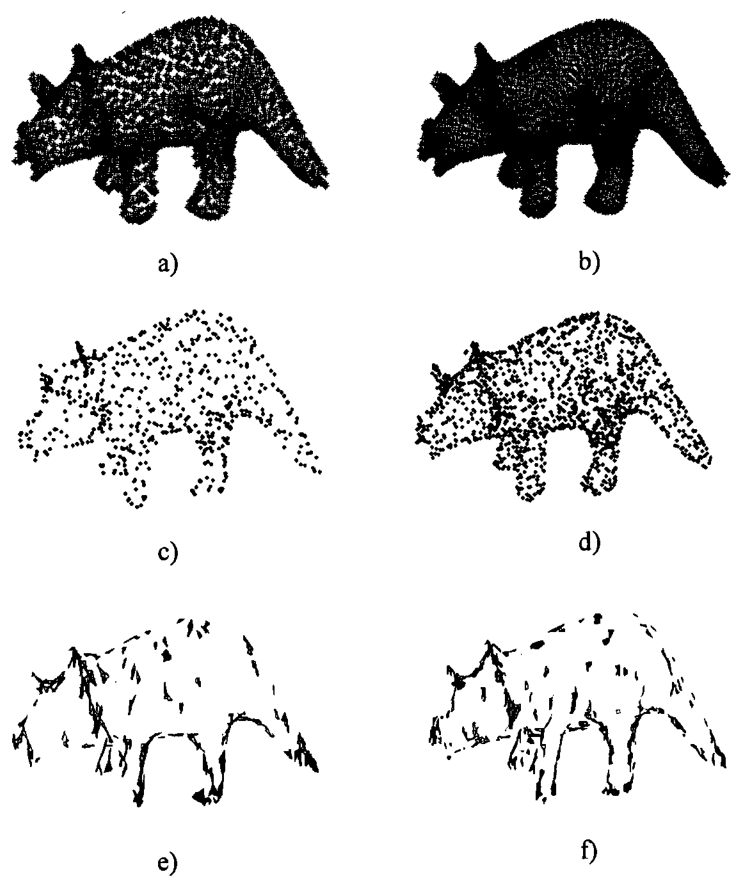


Fig. 5.24. Initial scan at (a) low resolution and (b) medium low resolution, growing neural gas model of (c) 757 points and (d) 1410 points and detected regions of interest for further sampling for (e) low resolution and (f) medium low resolution models.

5.3.2.2. Case of Objects with Raster-like Distribution of Sampling Points and Few Non-localized Features

The chair described in section 4.3.1 represents of an object without many features, with raster-like distributed sampling points. The data are used as collected by the Jupiter laser scanner, as in the case of neural gas, without any filtering procedure. Fig. 5.25 presents the best modeling results using growing neural gas for the low (3065 points) and medium (6113 points) resolution initial scans, obtained for $\lambda=3$ and $a_{max}=20$.

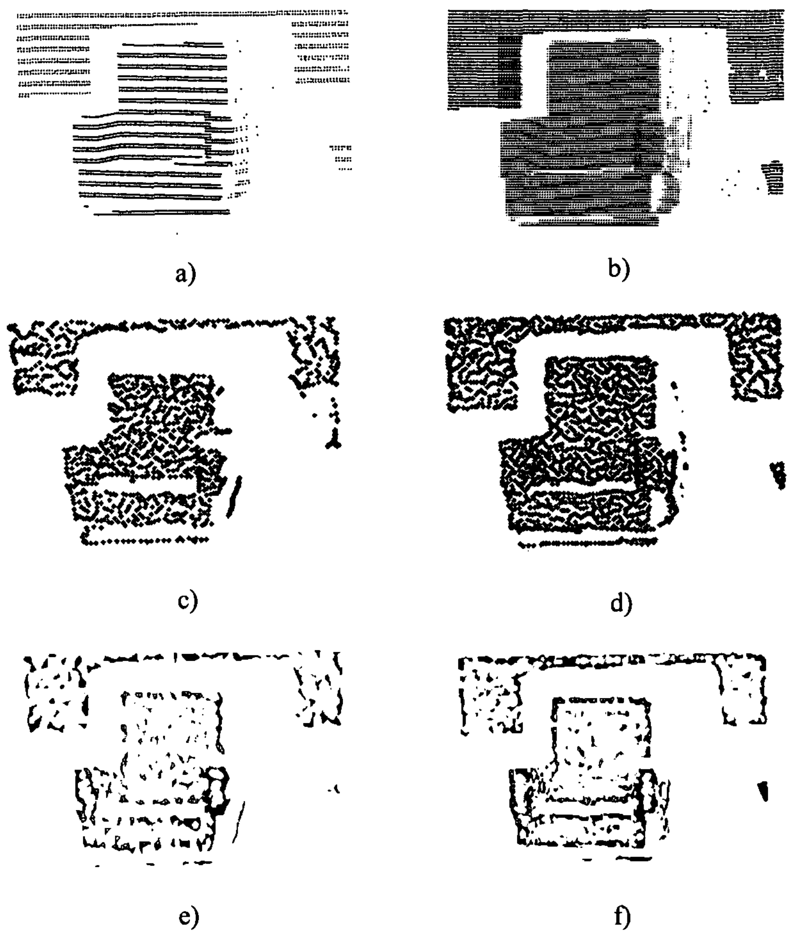


Fig. 5.25. Initial scan at (a) low resolution and (b) medium low resolution, growing neural gas model of (c) 1243 points and (d) 2826 points and detected regions of interest for further sampling for (e) low resolution and (f) medium low resolution models.

The same parameters are used as in the case of the triceratops and the results are evaluated for different sizes of the initial scan. For each case, the figure shows the detected regions of interest in the growing neural gas map (Fig. 5.25e for the low resolution and Fig. 5.25f for the medium resolution). As it can be seen, the network is able to detect the features in both cases in spite of some noise in the modeling results.

5.3.2.3. The Case of Objects with Raster-like Distribution of Sampling Points and Small Localized Features

This case is studied for the mockup car door which presents fine details. A fast scan of medium low resolution is initially performed to obtain the 16384 points medium resolution scan shown in Fig. 5.26a (the same as in Fig. 4.10b). The resulting growing neural gas mapping is depicted in Fig. 5.26b and the areas of high density in this mapping in Fig. 5.26c. The selectively densified point-cloud is depicted in Fig. 5.26d.

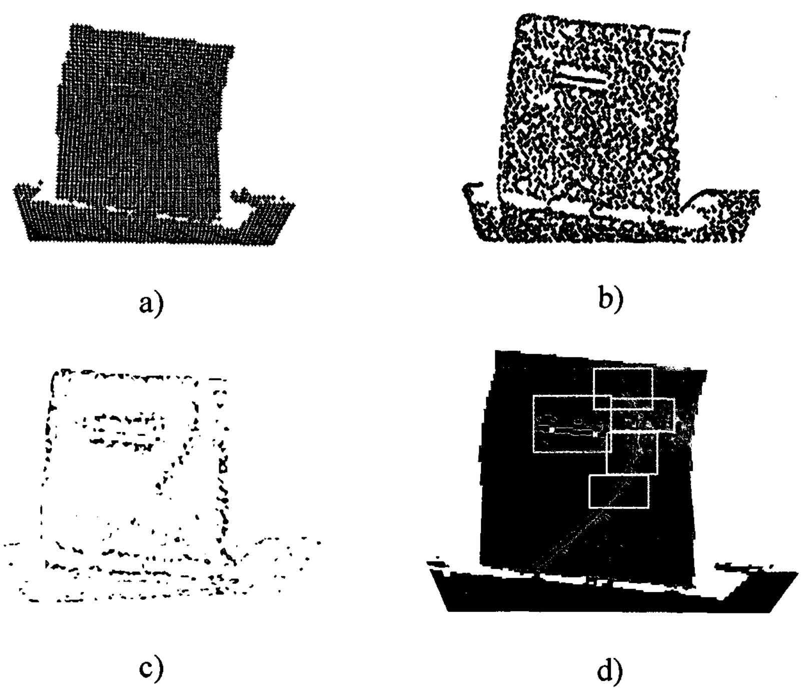


Fig. 5.26. The mock-up door (a) point-cloud, (b) growing neural gas mapping, (c) high density areas and the (d) detected regions of interest.

It can be seen, that in spite of some noise, the growing neural gas is able to detect the same regions for further scanning as the neural gas for the same initial resolution scan. As in the case of neural gas, the same procedure can be then repeated for each of the regions of interest detected in the previous step. Each region is provided as input in a growing neural gas network in order to further detect fine details that are worth to be scanned at a higher resolution. Fig. 5.27 presents the details of the high resolution rendered model of the door for the selected regions in the previous step. For each of the regions it shows the growing neural gas model for $\lambda=3$ and $a_{max}=20$ and the detected regions of interest for further scanning, identified as high density areas in the neural gas model. The average error is of order e^{-6} for each of the regions.

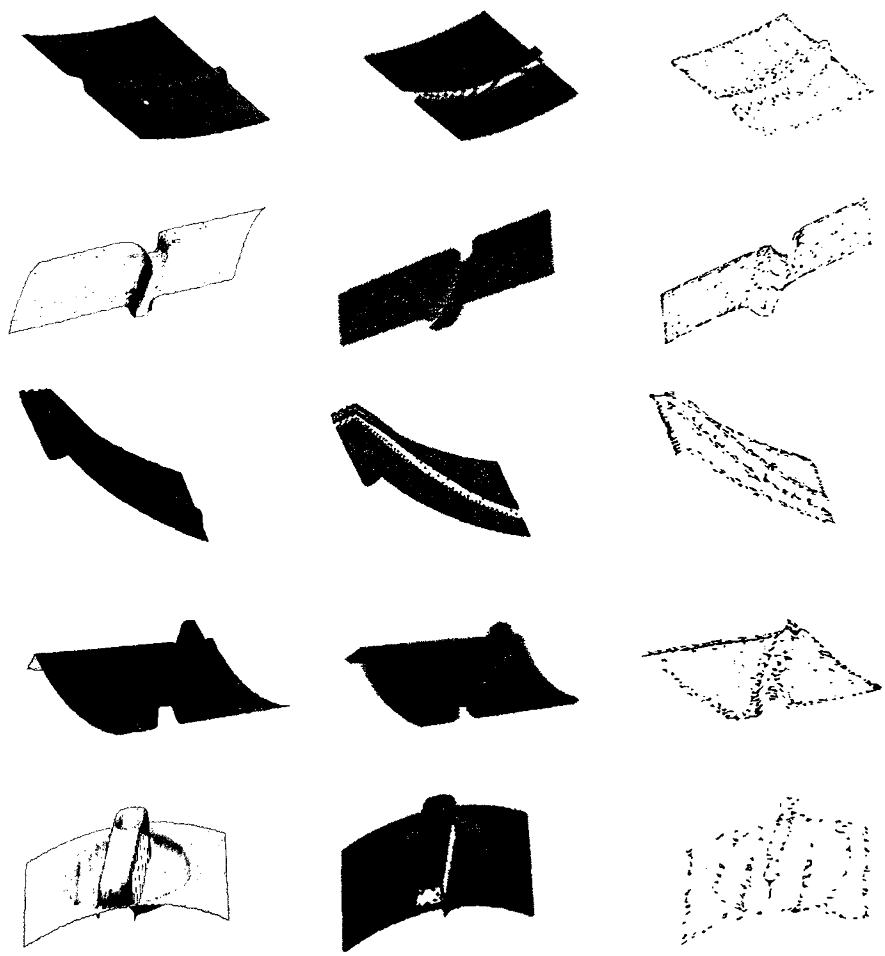


Fig. 5.27. Different views of rendered selected regions (first column), point-clouds of selected regions (second column) and detected regions of interest for each view (third column).

All the presented examples show the capability of the growing neural gas map to capture the fine details in the sparsely collected point-clouds of all the objects under study. By finding the high density areas in the growing neural gas map, the proposed selective sampling procedure is able to identify and guide the sensors to collect only measurements in those regions that are of interest for the improvement of accuracy of the obtained models. Moreover, it eliminates the constraint of a fixed map size imposed by neural gas.

5.4. Experimental Results for Tactile Data Acquisition with Force Sensors

As in the case of neural gas, a similar approach to the one presented for vision sensing can be used to exemplify the detection of regions for tactile sampling. Starting from an initial vision and tactile sparse scan, the proposed solution identifies those regions that are representative for the elastic behavior of objects in order to ensure the capability of the proposed solution to deal not only with homogeneous objects from the elasticity point of view, but also with non-homogeneous, particularly piecewise-homogeneous objects. Since it was already demonstrated in section 4.4.2 for the case of neural gas that the addition of a measure of compliance improves the modeling results, such a measure will be used for the modeling of all objects in this section.

The same set of deformable objects presented in chapter 4 for the neural gas (chapter 4.3.1) are used here: the rubber ball, the simple composite object made of a square cardboard box on the top of a covered foam pillow and the toy triceratops.

Fig. 5.28 shows some modeling results for the three objects under study, where different regions of elastic behavior are depicted in different colors. The rubber ball is depicted in Fig. 5.28a, with a uniform color representing the constant compliance throughout the body of the object. As it is expected, in this case the elasticity is correctly embedded in the growing neural gas map. The box-pillow composite object model depicted in Fig. 5.28b, with red points denoting the cardboard of which the box is made of and blue points denoting the foam of which the pillow is made of. In this case as well, the network learns correctly the two different compliances (computed from the collected tactile data for the ball and the pillow-box composite object and from the force-displacement curves in the case of triceratops, as explained in section 4.4.3.2 and 4.4.3.1 respectively) corresponding to the two materials. The triceratops model in Fig. 5.29b is composed of several regions of different elastic behavior, correctly embedded in the growing neural gas as well. Therefore, in all modeling cases, the growing neural gas map captures correctly the elastic parameter included as a fourth dimension in the input space.

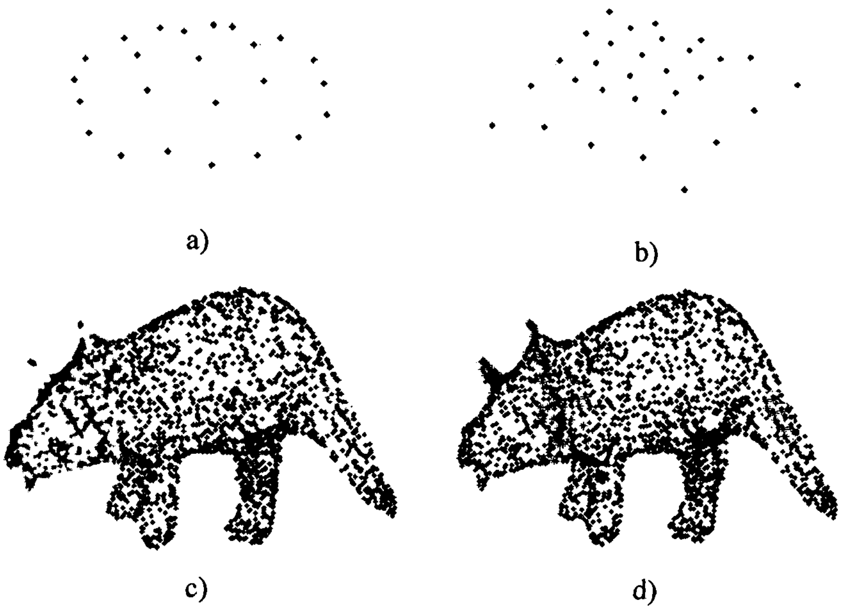


Fig. 5.28. *Growing neural gas with color representing compliance to model (a) rubber ball, (b) box-pillow composite object, (c) triceratops and d) areas where changes in compliance occur for the triceratops.*

Similar to the neural gas, the growing neural map gas can be used to identify areas of change in the elastic behavior for further probing. These regions are identified by triangles in the Delaunay triangulation whose vertices present different compliances, as explained in the case of neural gas in section 4.4.4. The identified regions where a change in compliance occurs in the growing neural gas map are depicted in Fig. 5.28d with red stars for the triceratops. It can be seen that the regions are correctly identified.

5.5. Comparison between Neural Gas and Growing Neural Gas

Although the neural gas and growing neural gas algorithms present both self-organizing attributes, are similar in approach and are based on the same ideas, there are differences in terms of required user intervention, accuracy of results, training time and obtained errors. These differences are summarized below.

To begin with, the growing neural gas eliminates the need of the map size to be decided prior to learning as in the case of neural gas, size that constrains the resulting mapping as well as the accuracy of the modeling results. Moreover, by comparing the final map size obtained by growing neural gas and the guidelines obtained for the neural gas in Chapter 4, it can be seen in Table 5.2 that generally the size of growing neural gas map (the number of nodes in the resulting graph) is smaller than the one required by neural gas in order to obtain reasonably good results.

Table 5.2. *Approximate map sizes for different sizes of initial scans for neural gas and growing neural gas.*

Size of initial scan point cloud (number of points, N)	***Approximate map size to provide reasonably good results for NG***	***Approximate map obtained by GNG to provide reasonably good results***
2000-3000	60%-100%N	20-45%N
3000-4000	50%-90%N	20-45%N
4000-5000	30%-70%N	20-45%N
5000-6000	20%-60%N	16-49%N
6000-7000	15%-50%N	16-49%N
over 7000	10%-40%N	10-40%N

The same conclusion can be derived from Fig. 5.29 that depicts the average network map sizes for different initial scans, computed as average value of map sizes for the range of values that provide reasonably good modeling results and for all the objects under study, as per Table 5.2.

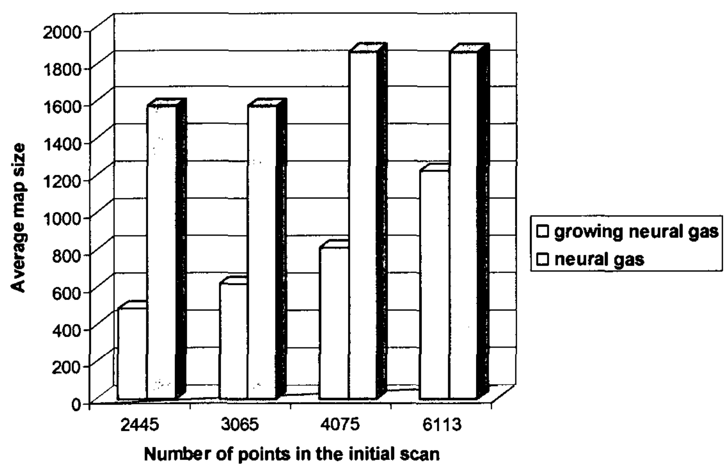


Fig. 5.29. Comparison of growing neural gas and neural gas based on map size.

In spite of the lower map size, the accuracy of growing neural gas models is higher when compared to neural gas. This fact proven by the lower average error limits reached, as shown in Fig. 5.30. The average error is computed as the average relative error for all objects under study, as in the case of the network map size.

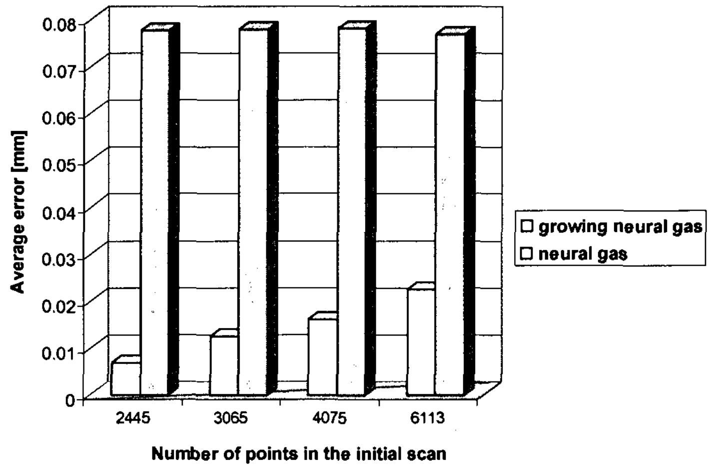


Fig. 5.30. Comparison of growing neural gas and neural gas based on local errors.

In terms of training time, the time required to build the model is also generally lower for growing neural gas, as shown in Fig. 5.31.

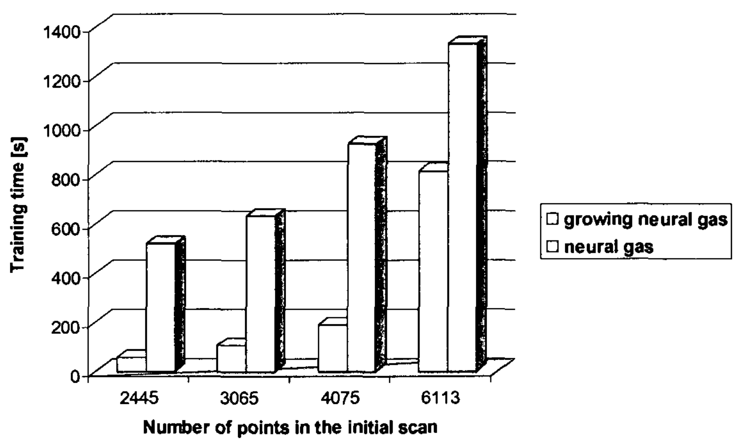


Fig. 5.31. Comparison of growing neural gas and neural gas based on training time.

However, since the map obtained by growing neural gas is evenly distributed over raster-like object point-clouds, the relevant areas for additional scanning are slightly harder to identify and they result in higher noise in the characterization of relevant features, as it can be seen for example, for the case of the medium resolution initial scan of the triceratops, the chair and the mock-up car door, in Fig. 5.32.

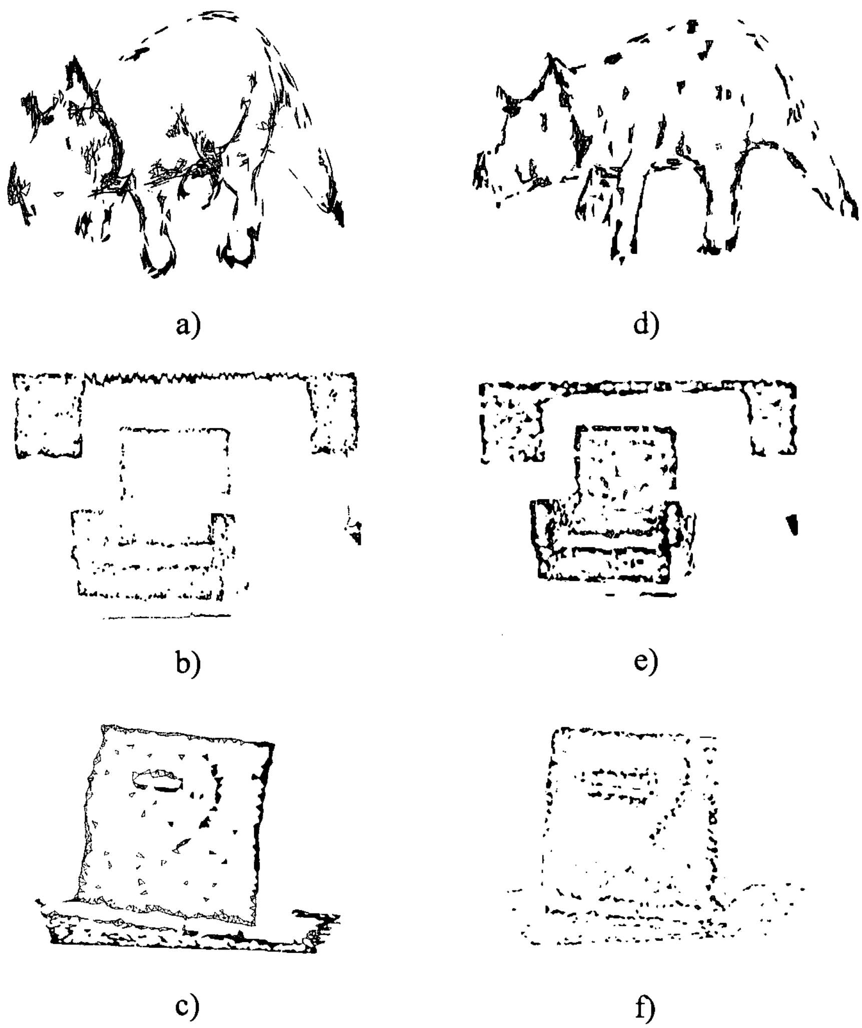


Fig. 5.32. Higher density areas identified in the neural gas map (first column) for (a) triceratops (b) chair and (c) door and in the growing neural gas map (second column) for (d) triceratops, (e) chair and (f) door.

The first column of Fig. 5.32 shows the best modeling cases for the neural gas while the second column for the growing neural gas. When comparing the results, it can be noticed that the edges are clearer contoured in Fig. 5.32c, representing the high density areas detected in the neural gas map than those in Fig. 5.32f that represent the high density areas in the growing neural gas map. Therefore the results in the second column are slightly noisier.

This phenomenon is alleviated in the case of neural gas model by slightly oversizing the map dimensions and stopping the adaptation early enough not to allow the output space to become evenly distributed. Such a mechanism is impossible to be established in the case of the growing neural gas as the points in the output space are added iteratively to support the node that has accumulated the highest error in the previous steps. The new node is placed between the node with the highest error and one of its neighbors with the next highest error, therefore the error between the input space and the map being built is always decreased. This implies necessarily that the model will respect the density of points in the point-cloud and cover evenly the input space.

When employed for tactile sampling guidance, both neural gas and growing neural gas are able to correctly capture the elastic parameter included as a fourth input parameter when used for tactile sampling and to identify areas of elasticity change where additional sampling is required, as it can be seen in Fig. 4.21 and Fig. 5.28c-d.

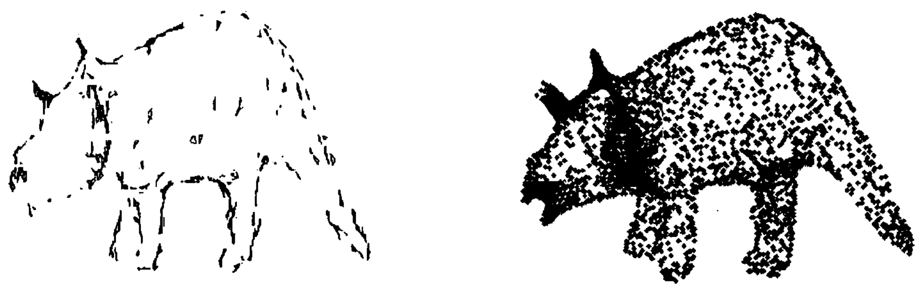
From this experimentation, the growing neural gas reveals to be faster, to lead to lower errors in mapping and to require less user interaction in the modeling procedure. But it gives slightly noisier results for geometrical feature detection, as required for further sampling regions selection. Therefore the growing neural gas is an appropriate choice when more compact object models are desired, and when the user is not willing to interfere at all in the modeling procedure. On the other hand, neural gas provides clearer definition of the regions for further scanning, especially on edges but requires some trial-and-error settings in order to provide the best results. Therefore, it is appropriate when the user is interested in accurately finding features over the objects.

5.6. Example of Complete Modeling Scenario

The example presented here refers to the selective geometric and tactile sampling of an object based on its growing neural gas map and the modeling of elastic behavior in form of force-displacement curves. It is meant to complement and complete the modeling scenario presented in section 4.5.2 that referred to the neural gas and elasticity measurements in form of deformation profiles associated with force recordings.

This modeling scenario is exemplified for the triceratops model. The modeling procedure starts by collecting data samples, both geometric and tactile, at approximately even spacing over the surface of the object at each vertex of a coarse triangular surface mesh with approximately equal sized triangles. Data is collected in form of force-displacement curves in time are collected at each vertex of the mesh. The *(x, y, z)* coordinates of the mesh and the compliance are used as an input in a growing neural gas architecture to identify regions of further interest and detect clusters of similar elasticity.

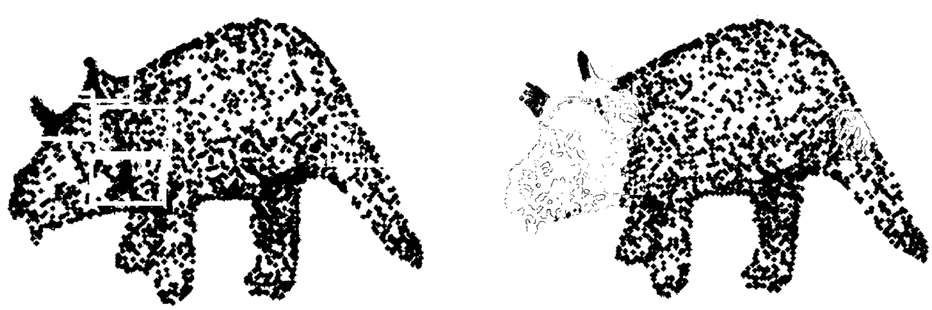
Fig. 5.33a shows the mapping obtained for an initial scan of 6113 points using a growing neural gas map of 2562 points and the identified areas of high density, neglecting the contour areas.



a) b)

Fig. 5.33. (a) High density areas detected in the growing neural gas map and b) the corresponding selectively densified triceratops point-cloud.

The selectively densified model is built by rescanning the areas determined in the previous step at higher resolution and by augmenting the growing neural map with the points belonging to these areas. In this way a multi-resolution compressed geometric model of 3528 points is obtained and shown in Fig. 5.33b.

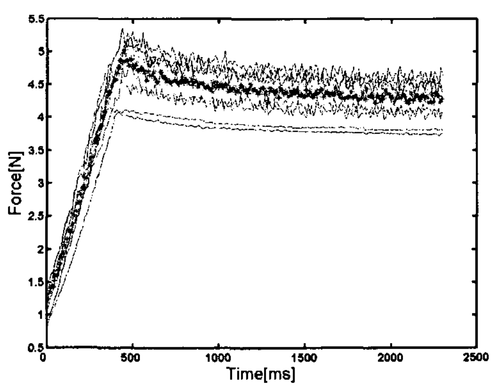


a) b)

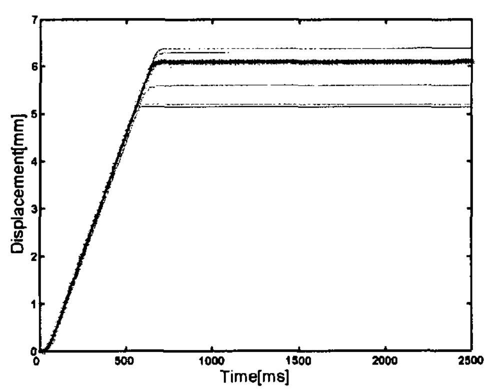
Fig. 5.34. (a) Areas of elasticity change detected in the growing neural gas map and (b) the corresponding selectively densified point-cloud.

Based on the compliance information in the same growing neural gas map, regions of changes in compliance are identified as shown in section 4.4.4 and are depicted in Fig. 5.34a. The selectively densified model, shown in Fig. 5.34b and obtained by the addition of the high resolution scans of the regions framed in Fig. 5.34a to the growing neural gas map, contains in this case 3584 points. It can be seen the areas around the horns and the neck of the triceratops are identified as areas of further sampling both by the geometry and tactile-based information, due to the fact that in these areas changes in elastic behavior is associated with changes in the local geometry. To achieve a full composite model, the union of the two selectively densified points is performed. The composite multi-resolution model contains 3615 points and represents about 30% of the full resolution scan of the triceratops. This implies the elimination of about 70% of the less relevant points by employing the proposed selective sampling framework.

To model the elastic behavior, the selectively densified model is first split into 7 clusters/tactile patches, corresponding to the different compliances, shown in different color in Fig. 4.24b. In order to train each of the feedforward networks that represent the elastic behavior of each cluster/tactile patch, the median force and displacement curves are estimated first, as presented in Chapter 3, this time for all of the data available for each cluster. Fig. 5.35 depicts the measured data set for a cluster with the apparent stiffness less than 0.08 (the least stiff region of the measured sample object, shown in dark blue) and the corresponding median curve, shown in bold.



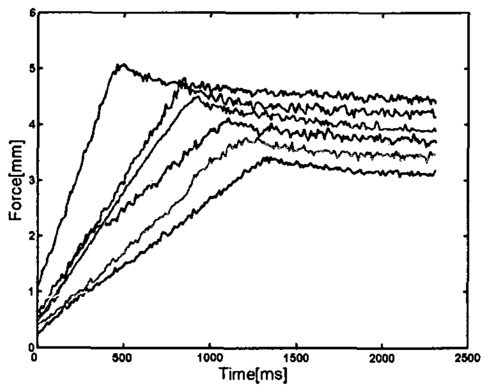
a)



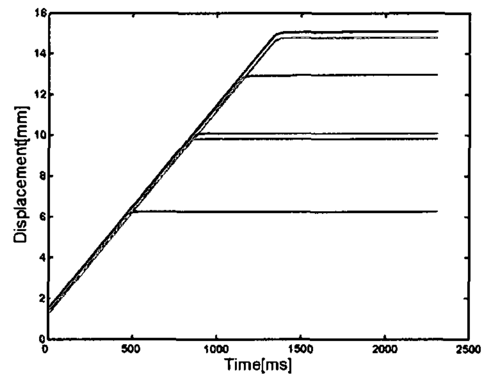
b)

Fig. 5.35. Measured data and the corresponding median curve a) force and b) displacement for cluster/tactile patch 1.

The relative error that is introduced in the elasticity modeling results by using this procedure depends mainly on the size of the chosen clusters and the number of sampled points measured in the corresponding cluster. Their corresponding median force and median displacement curves for all the clusters are presented in Fig. 5.36.



a)



b)

Fig. 5.36. Median (a) force and (b) displacement for the seven clusters/tactile patches.

The characteristics of the clusters/tactile patches: the apparent stiffness (ε), the median stiffness (ε_m), the mean relative error (introduced by using the median as representative for all force and displacement curves from a certain cluster and computed as the average distance from the median to each of the represented curves) (e_m), the mean learning errors for all the curves in the cluster (e_l) (computed as an average of the differences between the median curve and the modeling results obtained with the neural network) and the total cumulative error (e_t), computed as a sum of the mean relative error (e_m) and the mean learning error (e_l) are summarized in Table 5.3.

Table 5.3. Cluster/tactile patch characteristics.

	ε	ε_m	e_m	e_l	e_t
Cluster 1	$\varepsilon<0.08$	0.0440	0.0601	0.0019	0.062
Cluster 2	$0.08<\varepsilon<0.1$	0.0938	0.1416	0.0099	0.1515
Cluster 3	$0.1<\varepsilon<0.2$	0.1451	0.23	0.0057	0.2357
Cluster 4	$0.2<\varepsilon<0.3$	0.2901	0.1631	0.0102	0.1733
Cluster 5	$0.3<\varepsilon<0.5$	0.4492	0.1381	0.0114	0.1495
Cluster 6	$0.5<\varepsilon<0.6$	0.5269	0.1090	0.0063	0.1153
Cluster 7	$0.6<\varepsilon<1$	0.9464	0.0534	0.0052	0.0586

It can be observed that the use of the median as representative of a cluster introduces more error in the modeling than the learning procedure using neural networks. However, for most of the clusters, the total error is relatively low.

The same network architecture is used to model the median force-displacement of each of the clusters, with one input neuron, 15 hidden neurons and one output neuron, as the one depicted in Fig. 3.5, therefore a total of 7 networks are required in this case. All the networks are trained for 20000 epochs, with the learning rate 0.1 and an error goal of 0.0001. Fig. 5.37 shows samples of training and corresponding target data produced by the networks and the desired force-displacement curves, for example, for clusters 1 and 7.

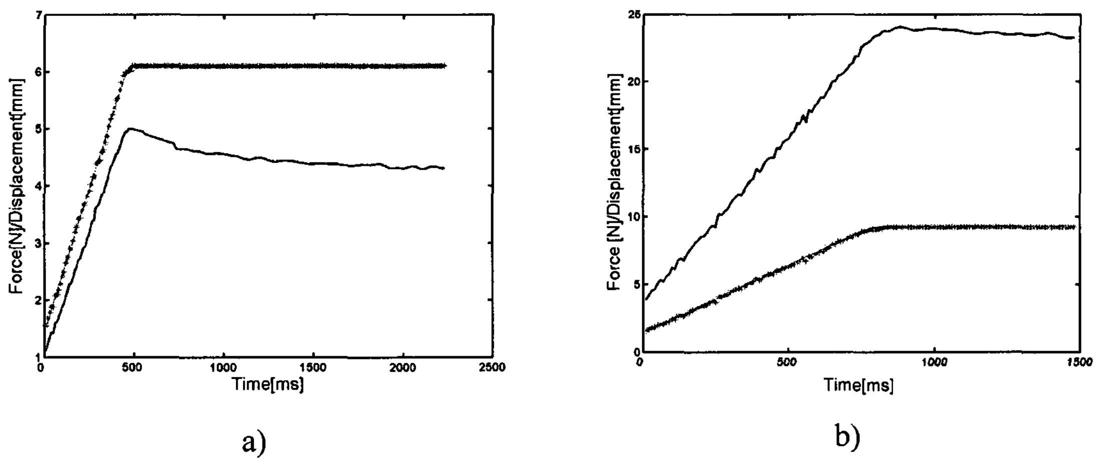


Fig. 5.37. Training results for a) cluster/tactile patch 1 and b) cluster/tactile patch 7.

In order to interact with such models, as explained in section 4.5, a mapping is built between each cluster and its corresponding network. Each time a point is probed with the force at an angle, the closest point in the neural gas is determined and from there, the cluster to which it belongs is selected. Knowing the cluster, the corresponding feedforward network provides the estimated displacement for the given force.

5.7. Conclusions

To conclude, the growing neural gas eliminates the need of the map size to be decided prior to learning. During adaptation the network adds iteratively nodes to the structure to better fit the data provided as input and is able, in all the cases under study to identify the regions of further interest for scanning. All the examples show the ability of the growing neural gas map to capture the fine details in the sparsely collected point clouds of the objects under study. By finding the high density areas in the growing neural gas map, the proposed selective sampling procedure is able to automatically identify and guide the vision and tactile sensors to collect only measurements in those regions that are of interest for the improvement of accuracy of the obtained models, saving at the same time large amount of less relevant data in the scans. The growing neural gas solution is faster, reaches lower errors and does not require user intervention for the selection of an appropriate map size when compared to the previously proposed neural gas solution. However the features in the growing neural gas map are slightly less clearly defined.

6. Conclusion and Future Work

6.1. Summary of Work

The thesis presents the results of more than four years of research performed on the topic of data acquisition and modeling of deformable objects based on experimental data using soft computing techniques, particularly neural networks and neural clustering algorithms.

The problem of accurately modeling deformable objects is not trivial. The area of expertise required covers a vast range of topics such as: the determination of proper equipment to perform measurements, the fusion of different technologies and of data available from different sources to improve the measurement process, the interpretation, analysis and correlation of data, the proper selection of methods to model elastic data and to join these data in coherent composite geometric and elastic models and finally the means of interacting with the models. While an intensive research has been performed on several of these topics in the past decades, the thorough literature review conducted during this research led to the conclusion that there were certain limitations and several neglected aspects related to the measurement and modeling of deformable objects.

The main limitation encountered is related to the focus of current researchers on simulations based on assumptions on the material of deformable object and on means to interact with the increasingly complex models in real-time. Much less attention has been given to real measurements of elastic behavior, in the detriment of accuracy and realism. Overall, there are few papers that imply any sort of measurement to elicit the object's elastic behavior, and in most cases when a measurement is performed, the probing is limited to a single point. Such an approach can only produce satisfactory results for objects that are made of homogeneous elastic materials and is unsuitable for heterogeneous or piecewise homogeneous objects. Last but not least, many of the proposed solutions are highly specialized and dedicated to a particular application and cannot be employed in a different context. These are some of the reasons that drove the research work in this thesis.

In particular, the work presented here is a breach in a relatively unexplored area of modeling objects without assumptions on their material, with accent on viable, innovative and easily applicable solutions to measure and model elastic behavior of real world objects. The emphasis is on accurate models as opposed to real-time behavior. The final goal is to develop a general-purpose scheme for measuring, constructing and representing geometric and elastic behavior of deformable objects without *a priori* knowledge on the material that the objects under study are made of.

The first aspect that was investigated is related to the means of representing elastic behavior of objects from experimental data. The proposed approach does not attempt to recover explicitly neither the elastic constants, nor the stress and strain tensors. The procedure of recovery of elastic parameters and stress and strain tensors implies complicated mathematical models based on assumptions. Such models are frequently impossible to define and solve for highly nonlinear elastic materials. The focus in this work is on the observation of the displacements of the surface of an object when loaded by a measured external force. Particularly, data in form of force-displacement curves collected using a force/toque sensor and a stereo vision system, and deformation profiles sensed with a laser scanner correlated with force components measurements collected by a force/torque sensor, are used as a representation of elastic behavior of objects.

The choice of using neural networks to learn the highly nonlinear mapping from force to displacements, therefore encoding the elastic behavior data, proved to bring multiple advantages over classical solutions: neural networks deal easily with real experimental data with noisy and incomplete measurements, they avoid the complicated and frequently impossible to solve problem of recuperating explicit elastic parameters, they provide continuous output behavior thus being able to provide the necessary nonlinear interpolation for estimates of data that was not part of the training set, they provide an accurate and fast response once they are trained, and finally the neural network modeling of deformation profiles captures naturally the modification of the objects' volume due to applied forces. The framework can be easily adapted to model different types of materials and/or the different stages of deformation (elastic, elasto-plastic and plastic stages).

The second aspect of interest to this thesis is the means to selectively probe an object in order to only collect relevant data from the geometric and elastic points of view. The proposed scheme uses an innovative probing and measuring technique for the selective data acquisition of geometric and elastic characteristics on a series of points over the object's surface. The choice of using a self-organizing architecture for selective sampling is justified by its ability to quantize the given input space into clusters of points with similar properties. These modeling properties ensure on one side the capability of the application to deal not only with homogeneous objects from the elasticity point of view, but also with non-homogeneous, particularly piecewise-homogeneous objects. On the other side, they ensure that the density of the points is higher in the regions with more pronounced variations in the geometric shape and therefore such information can be used to guide the probing procedure. While a similar approach proved to be appropriate in vision to collect 3D geometry data and for tactile sensing as well, an enhancement was observed for tactile sampling when an additional measure of compliance is used in the mapping, especially for objects where the change of elastic behavior is not related to a change in the geometry of the object. From the self-organizing architectures, neural gas and growing neural gas were studied as a basis for the selective sampling procedure.

The regions that are worth further sampling in order to ensure an accurate geometric and elastic model are detected by finding higher density areas in the (growing or) neural gas map based on an initial sparse scan of the object. In order to ensure an accurate elastic behavior model, areas where changes in the elastic behavior occur in the neural map are identified. This means that the proposed selective sampling procedure is able to automatically identify and guide the vision and tactile sensors to collect only measurements in those regions that are of interest for the improvement of accuracy of the obtained models, saving at the same time large amount of less relevant data in the scans. Rescanning at higher resolution can be performed for each identified region and a multi-resolution geometric and elastic model can then be built using the initial sparse model or the (growing or) neural gas map if a more compressed model is desired, and by augmenting it with the high resolution measurements in the regions of interest. The rescanning procedure can be repeated in several steps to gradually improve the accuracy of the model. By storing the model in each phase of scanning, a multi-resolution point-based object model is obtained, each re-scanning phase providing a higher point density in the regions rich in features and/or with changes in elastic behavior.

The comparison between the two self-organizing architectures revealed that the growing neural gas eliminates the need of the map size to be decided prior to learning, as in the case of neural gas, is faster and reaches lower errors when compared to the initially proposed neural gas solution. However it gives slightly noisier results for geometrical feature detection, as required for the selection of further sampling regions. Therefore the growing neural gas is an appropriate choice when more compact object models are desired, and when the user is not willing to interfere at all in the modeling procedure. On the other hand, neural gas provides clearer definition of the regions for further scanning, especially on edges but requires some trial-and-error settings in order to provide the best results. Therefore, it is appropriate when the user is interested in accurately finding features over the objects.

The last area of research is related to means of constructing multi-resolution composite geometric and elastic models. As mentioned before, the modeling properties of growing neural gas and neural gas allow for an incremental rescanning at higher resolution of different areas of interest as identified in a previous scanning step. Moreover, the clustering properties of the (growing and) neural gas network allow for the determination of clusters of similar geometric and elastic properties. These clusters can be imagined as "tactile patches". Therefore objects can be built as multi-resolution models with tactile patches, with the behavior of each patch being modeled with a feedforward neural network. The number of neural networks implied is equal to the number of tactile patches. The same architecture is employed for each cluster to capture the complex relationship between an applied force, its magnitude, its angle of application and its point of interaction, the object pose and the deformation stage/material state (if desired), and the object surface deformation. Each such neural network is associated with the location of the geometric points that belong to the corresponding cluster. When an interaction occurs with an object model at a certain point, the point of interaction is projected to the closest point from the model and the proper neural network (that models the elastic behavior of the cluster to which this closest point belongs) is selected to provide the deformation profile of the object under study based on the specific characteristics of the interaction.

The work in this thesis comes into benefit and is applicable in the areas of CAD/CAM, quality control, and reverse engineering of elastic material components, interactive virtual environments for training, interactive virtual prototyping for industrial CAD-based processes, digital evaluation and maintenance operation simulation, the computer games industry, as well as for many dexterous manipulation applications such as robotic assembly, tele-robotics, remote medical examination of patients and robotic surgery. As well, some other applications that would benefit from the proposed selective vision sampling scheme are envisioned for 3D modeling for remote robotic exploration, space vehicles inspection, metrology, aerial/satellite imaging, and image-based virtual environments data collection.

6.2. Contributions

The contributions of this thesis are therefore related to aspects of measurement and modeling of 3D deformable objects and can be summarized as follows:

- An original approach to collect force-deformation measurements (in form of deformation profiles correlated with force recordings) on deformable objects by combining, synchronizing and merging data acquired using a laser range sensor and an industrial force-torque sensor.
- An innovative automated selective sampling algorithm to guide online the probing of only relevant points for the objects under study, including:
    - The adaptation of neural gas and growing neural gas clustering algorithms for this purpose
    - An experimental evaluation and comparison of the two clustering algorithms for this purpose
    - The application of the proposed selective sampling for the acquisition of both the geometry and the elastic characteristics of deformable objects.

Some of the advantages of the proposed algorithm are the fact that it works directly from raw range data (3D vision), it deals with homogeneous and non-homogeneous objects as well, allowing distinct treatment of regions with different elastic behaviors, and it offers the desired degree of control to the user in that the modeling parameters can be chosen automatically or they can be tuned by the user.

- An original use of a feedforward neural network to capture the complex relationship between an applied force, its magnitude, its angle of application, the point of interaction, the object pose and the deformation stage/material state on one side, and the object surface deformation on the other side. The network learns implicitly the possibly highly nonlinear elasticity in form of 2.5D profiles of deformations of an object. It works on materials with different elastic behavior (including rigid, elastic and plastic deformable objects) and can correctly capture the different deformation stages of an object. Last but not least, it deals naturally with the expansion of the objects' body due to an application of a load.
- An original combination of neural architectures to drive the acquisition and to encode compact multi-resolution models of deformable objects, which leads to the merge of geometry and elasticity data.
- The integration of the previous elements into an innovative sampling and modeling framework for deformable objects that operates from noisy experimental measurements without *a priori* knowledge on the shape and the material of the object under study.

The proposed solution presents several advantages over the classical deformable object models. It eliminates all the guesses for tuning the spring constants and their configurations in mass-spring models or the requirements for small deformations imposed by FEM in order to avoid the computation of the integration when the object deforms. It also avoids the complicated models to recuperate the elastic parameters as well all the assumptions on objects' homogeneity used by BEM. There is no restriction on the degrees of freedom on object as imposed by modal analysis, while the topology of the deformed surface is available unlike the cases of particle systems or implicit surfaces. As opposed to any of these models, the proposed solution is able to estimate the elastic behavior for data that was not measured and modeled. Last but not least, our solution also deals naturally with the object's volume expansion, which few of the classical solutions are able to do.

The results in this thesis were reported in 4 journal papers of which 3 are published [147, 156, 157], and two in press [158, 167], 9 conference papers [137, 159-162, 166-169] (of which [161] received the Best Student Paper Award, at *IEEE Instrumentation and Measurement Technology Conference*, Victoria, BC, May 2008) and three presentations [163-165] (of which [163] won the 3rd place Research Poster Prize, at the *Graduate and Research Day*, University of Ottawa, Ottawa, February 2007). The areas of coverage of these publications can be summarized as follows: the selective sampling theory is discussed in [147], exemplified for tactile sampling in [157, 159, 160, 163, 166] and for vision sampling in [161, 162, 164, 165]. The acquisition setup and modeling for deformation profiles are presented and tested in [157, 160] and for force-displacement curves in [137]. The idea of multiple scanning directions for the characterization of elastic behavior, particularly of a perpendicular scanning direction appeared in [156]. The additional dimension of elastic compliance for neural gas is proposed in [159]. The growing neural gas and the comparison with neural gas for selective vision sampling is presented in [162, 167]. The adaptation of the feedforward architecture for rigid, elastic and elasto-plastic objects appeared in [168] and the description of composite geometric-elastic modeling framework in [169].

6.3. Future Work

The framework proposed in the thesis proved its potential for the sampling and modeling of deformable objects for various sources of experimental data, for various materials and for different object shapes.

Further developments of the work presented refer to the rendering of the resulting composite geometric and elastic models to be incorporated in a 3D deformable object engine. While several solutions have been proposed in the literature for point-cloud based rendering, the technologies are still in their infancy and additional research has to be performed to create realistic objects from their point-based representation. Alternatively a mesh of the object can be built for the same purpose. None of these are trivial or fast to implement. Another aspect that requires attention in the same context of rendering is the integration of elasticity measurements or of the deformation profiles with the geometric model for a seamless fitting of the deformed area within the whole model.

The proposed model can be further integrated in complex deformable objects environments by programming protocols for interaction and/or feedback. The integration with force feedback equipment is another complicated issue to solve mainly due to the challenges in recreating the experience caused by the contact between an interaction tool and the object under study.

Several improvements towards faster training of neural networks are envisioned by code optimization and transfer to a different software platform or by a combined hardware-software implementation of the neural network training. Last but not least, a speed up of the processing can be obtained using additional computing power, which is not a limitation anymore, at least for the rendering part, due to the continuous increase in the computing capacity of modern systems.

7. References

[1] A.-M. Cretu, *Neural Network Modeling of 3D Objects for Virtualized Reality Applications*, M.A.Sc. Thesis, University of Ottawa, Ottawa, Canada, 2003.

[2] M. Ameen, *Computational Elasticity*, Alpha-Science, India, 2005, pp. 3-11.

[3] A.C. Fischer-Cripps, *Introduction to Contact Mechanics*, Mechanical Eng. Series, Springer, New York, 2000.

[4] M.H. Saad, *Elasticity – Theory, Applications and Numerics*, Elsevier, USA, 2005.

[5] O. Rand and V. Rovenski, *Analytical Methods in Anisotropic Elasticity*, Birkhauser, 2005.

[6] R.L. Bisplinghoff, J.W. Marr and T.H.H. Pian, *Statics of Deformable Solids*, Dover Publications, 1990.

[7] W. Han, and B. Daya Reddy, *Plasticity – Mathematical Theory and Numerical Analysis*, Springer, 1999.

[8] K.K. Tho, S. Swaddiwudhipong, Z.S. Liu, and J. Hua, "Artificial Neural Network Model for Material Characterization by Indentation", *Modeling and Simulation in Material Science and Engineering*, no. 12, pp. 1055-1062, 2004.

[9] S. Charlton, P. Sikka and H. Zhang, "Extracting Contact Parameters from Tactile Data Using Artificial Neural Networks", *IEEE Int. Conf. Systems, Man and Cybernetics*, vol.5, pp. 3954-3959, 1995.

[10] C. Monserrat, U. Meier, M. Alcañiz, F. Chinesta, and M.C. Juan, "A New Approach for the Real-Time Simulation of Tissue Deformations in Surgery Simulation", *Computer Methods and Programs in Biomedicine*, vol. 64, issue 2, pp. 77-85, 2001.

[11] D. Aulignac, C. Laugier, and M.C. Cavusoglu, "Towards a Realistic Echographic Simulator with Force Feedback", *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, pp. 727-732, Kyongju, Korea, 1999.

[12] A.M. Okamura, J. T. Dennerlein and R.D. Howe, "Vibration Feedback Models for Virtual Environments", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1-6, 1998.

[13] Y. Zhu, T.J. Hall, and J. Jiang, "A Finite-Element Approach for Young's Modulus Reconstruction", *IEEE Trans. Med. Imaging*, vol. 22, no.7, pp. 890-901, 2003.

[14] J. Li, M. Wan, M. Qian, J. Cheng, "Elasticity Measurement of Soft Tissue Using Shear Acoustic Wave and Real-Time Movement Estimating Based on PCA Neural Network", *Proc. IEEE Int. Conf. Engineering in Medicine and Biology Society*, vol. 3, pp. 1395-1396, 1998.

[15] X. Wang, G.K. Ananthasuresh, and J.P. Ostrowski, "Vision-Based Sensing of Forces in Elastic Objects", *Sensors and Actuators A*, vol.94, pp.146-156, 2001.

[16] K. Kamiyama, H. Kajimoto, M. Inami, N. Kawakami, and S. Tachi, "A Vision-Based Tactile Sensor", *Proc. Int. Conf. Artificial Reality and Telexistence*, pp. 127-134, 2001.

[17] K. Kamiyama, H. Kajimoto, N. Kawakami, and S. Tachi, "Evaluation of a Vision-Based Tactile Sensor", *Proc. IEEE Int. Conf. Robotics and Automation*, vol.2, pp. 1542-1547, 2004.

[18] M. Greminger and B.J. Nelson, "Modeling Elastic Objects with Neural Networks for Vision-Based Force Measurement", *IEEE Int. Conf. Intelligent Robots and Systems*, pp. 1278-1283, Las Vegas, USA, 2003.

[19] V. Vuskovic, M. Krauer, G. Szekely and M. Reidy, "Realistic Force Feedback for Virtual reality Based Diagnostic Surgery Simulators", *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 1592-1598, April 2000.

[20] N. J. Ferrier and R. W. Brockett, "Reconstructing the Shape of a Deformable Membrane from Image Data," *Int. Journal Robotics Research*, vol. 19, no. 9, pp. 795-816, 2000.

[21] D. Hristu, N. Ferrier and R. W. Brockett, "The Performance of a Deformable-Membrane Tactile Sensor: Basic Results on Geometrically-Defined Tasks," *Proc. IEEE Int. Conf. Robotics & Automation*, vol.1, pp. 508-513, 2000.

[22] H.T. Tanaka, K. Kushihama, and N. Ueda, "A Vision-Based Haptic Exploration", *Proc. IEEE Int. Conf. Robotics & Automation*, Taipei, Taiwan, pp. 3441-3448, 2003.

[23] N. Ueda, S. Hirai, and H.T. Tanaka, "Extracting Rheological Properties of Deformable Objects with Haptic Vision", *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 3902-3907, 2004.

[24] D. K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau, "Scanning Physical Interaction Behavior of 3D Objects", *Proc. Computer Graphics and Interactive Techniques,* pp. 87-96, 2001.

[25] J. Lang, D.K. Pai, and R. J. Woodham, "Acquisition of Elastic Models for Interactive Simulation", *Int. Journal of Robotics Research*, vol. 21, no. 8, pp. 713-733, 2002.

[26] D. L. James and D. K. Pai, "ArtDefo: Accurate Real Time Deformable Objects", *Proc. ACM SIGGRAPH*, pp. 65-72, 1999.

[27] A. R. Skovoroda, S.Y. Emelianov, and M. O'Donnell, "Tissue Elasticity Reconstruction Based on Ultrasonic Displacement and Strain Images", *IEEE Trans. Ultrason., Ferroelect. Freq. Constr.*, vol. 42, pp. 747-765, 1995.

[28] A.J. Romano, J. J. Shirron, and J. A. Bucaro, "On the Noninvasive Determination of Material Parameters from a Knowledge of Elastic Displacements: Theory and Numerical Simulation", *IEEE Trans. Ultrason., Ferroelect. Freq. Constr.*, vol. 45, pp. 751-759, 1998.

[29] K.R. Raghavan and A. E. Yagle, "Forward and Inverse Problems in Elasticity Imaging of Soft-Tissue", *IEEE Trans. Nucl. Sci.,* vol. 41, pp. 1639-1648, 1994.

[30] F. Kallel, and M. Bertrand, "Tissue Elasticity Reconstruction Using Linear Perturbation Method", *IEEE Trans. Med. Imaging*, vol.15, pp. 299-313, 1996.

[31] D. Fu, S.F. Levinson, S.M. Gracewski, and K.J. Parker, "Non-Invasive Quantitative Reconstruction of Tissue Elasticity Using an Iterative Forward Approach", *Phys. Med. Biol.*, vol. 45, pp. 1495-1509, 2000.

[32] G. Canepa, M. Morabito, D. De Rossi, A. Caiti, and T. Parisini, "Shape From Touch by a Neural Net", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2075-2080, 1992.

[33] H.S. Shin and G.N. Pande, "Identification of Elastic Constants for Orthotropic Materials from a Structural Test", *Computers and Geotechnics*, vol. 30, pp. 571–577, 2003.

[34] D. Nehab and P. Shilane, "Stratified Point Sampling of 3D Models", *Eurographics Symposium on Point-Based Graphics*, M. Alexa, S. Rusinkiewicz (Eds.), pp. 49-56, 2004.

[35] R. Dillmann, S. Vogt and A. Zilker, "Data Reduction for Optical 3D Inspection in Automotive Applications", *IEEE Int. Conf. Multisensor Fusion and Integration for intelligent Systems,* pp. 159-164, 1999.

[36] K.H. Lee, H. Woo and T. Suk, "Point Data Reduction Using 3D Grids", *Int. Journal of Adv. Manuf. Technol,* vol. 18, pp. 201-210, Springer, 2001.

[37] M. Pauly, M. Gross and L.P. Kobbelt, "Efficient Simplification of Point-Sampled Surfaces", *IEEE Conf. Visualization*, pp. 163- 170, 2002.

[38] D. Uesu, L. Bavoil, S. Fleishman, J. Shepherd and C. T. Silva, "Simplification of Unstructured Tetrahedral Meshes by Point Sampling", *IEEE Int. Workshop Volume Graphics*, E. Gröller, I. Fujishiro (Eds.), pp. 157- 238, 2005.

[39] H. Song and H.-Y. Feng, "A Point Cloud Simplification Algorithm for Mechanical Part Inspection", *Information Technology for Balanced Manufacting Systems*, W. Shen (Ed.), Springer, pp. 461-468, 2006.

[40] A. Kalaiah and A. Varshney, "Statistical Point Geometry", Eurographics Symp. Geometry Processing, K. Kobbelt, P. Schroder and H.Hoppe (Eds.), pp. 107-115, 2003.

[41] S. Fiore, A. Faustini, and P. Burrascano, "Non-Uniform Sampling for Robot Motion Control by the GFS Neural Algorithm", *IEEE Int. Conf. Neural Networks*, pp. 2057-2060, 1999.

[42] D. MacKinnon, V. Aitken, and F. Blais, "Adaptive Laser Range Scanning using Quality Metrics", *Proc. IEEE Int. Instr. and Meas. Technology Conf.*, pp. 348-353, Victoria, Canada, May 2008.

[43] D. Terzopoulos, J. Platt, A. Barr, D. Zeltzer, A. Witkin, and J. Blinn, "Physically-based modeling: past, present, and future", *ACM SIGGRAPH Computer Graphics*, vol. 23, issue 5, pp. 191-209, 1989.

[44] R. Parent, *Computer Animation: Algorithms and Techniques*, Morgan Kaufmann, 2002.

[45] S.F.F. Gibson and B. Mirtich, "A Survey of Deformable Modeling in Computer Graphics", *Technical report*, Mitsubishi Electric Research Laboratory, 1997.

[46] Y. Zhuang and J. Canny, "Haptic Interaction with Global Deformations", *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, pp: 2428 –2433, Apr. 2000.

[47] D. Bourguignon and M.P Cani, "Controlling Anisotropy in Mass-Spring Systems", *Computer Animation and Simulation*, pp. 113-123, 2000.

[48] J. E. Chadwick, D. R. Haumann, and R. E. Parent, "Layered construction for deformable animated characters", *ACM SIGGRAPH Computer Graphics*, vol. 23, issue 3, pp. 243-252, 1989.

[49] J. Brown, S. Sorkin, C. Bruyns, J.-C. Latombe, K. Montgomery, and M. Stephanides, "Real-Time Simulation of Deformable Objects: Tools and Applications", *Computer Animation*, Seoul, 2001, available online at http://ai.stanford.edu/~latombe/papers/ca01/ca01.pdf.

[50] A. Joukhadar, F. Garat, and C. Laugier, "Parameter identification for dynamic simulation", *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, pp. 1928 – 1933, April 1999.

[51] G. Bianchi, M. Harders and G. Székely, "Mesh Topology Identification for Mass-Spring Models", *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003,* Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, Germany, pp. 50–58, 2003.

[52] O. Deussen, L. Kobbelt, and P. Tuck, "Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Objects", *Proc. Eurographics Workshop on Animation and Simulation*, pp. 30–43, Springer-Verlag, 1995.

[53] A. Nürnberger, A. Radetzky and R. Kruse, "Modeling and Simulating a Time-Dependent Physical System Using Fuzzy Techniques and a Recurrent Neural Network", *Proc. Int. Workshop Fuzzy- Neuro Systems*, pp. 306-313, Munchen, 1998.

[54] A. Nurnberger, A. Radetzky, and R. Kruse, "A Problem Specific Recurrent Neural Network for the Description and Simulation of Dynamic Spring Models", *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, pp. 468-473, May 1998.

[55] A. Radetzky, A. Nurnberger, and D.P. Pretschner, "Simulation of Elastic Tissues in Virtual Medicine Using Neuro-Fuzzy Systems", *Proc. of SPIE Medical Imaging: Image Display*, vol. 3335, pp. 399-409, 1998.

[56] A. Radetzky, A. Nurnberger, M. Teistler, D.P. Pretschner, and R. Kruse, "The Simulation of Elastic Tissues in Virtual Laparoscopy using Neural Networks", *Proc. Neural Networks in Applications*, pp. 167-174, Magdeburg, Germany, 1998.

[57] A. Radetzky, A. Nurnberger, M. Teistler, and D.P. Pretschner, "Elastodynamic shape modeling in virtual medicine", *Proc. Int. Conf. Shape Modeling and Applications*, pp. 172 –178, March 1999.

[58] A. Duysak, J. Zhang, and V. Ilankovan, "Efficient Modeling and Simulation of Soft Tissue Deformation Using Mass-Spring Systems", *Int. Congress Series,* vol. 1256, pp. 337-342, June 2003.

[59] W. Mollemans, F. Schutyser, J. Van Cleynenbreugel, and P. Suetens, "Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation", *Lecture Notes in Computer Science*, Springer-Verlag, vol. 2673, pp. 145–154, August 2003.

[60] S.A. Cover, N.F. Ezquerra, J.F.O'Brien, R. Rowe, T. Gadacz, and E. Palm, "Interactively Deformable Models for Surgery Simulation", *IEEE Trans. Computer Graphics and Applications*, vol. 13, issue 6, pp. 68–75, Nov. 1993.

[61] C. Wagner, M. A. Schill, and R. Männer, "Collision Detection and Tissue Modeling in a VR-Simulator for Eye Surgery", S. Müller and W. Stürzlinger, (Eds.), 8th *Eurographics Workshop on Virtual Environments*, pp. 27-36, Barcelona, May 2002.

[62] T. Vassilev and B. Spanlang, "A Mass-spring Model for Real Time Deformable Solids", *East-West-Vision,* 2002, available online at http://www.cs.ucl.ac.uk/staff/b.spanlang/AMSMFRTDS.pdf.

[63] T. Di Giacomo and N. Magnenat-Thalmann, "Bi-Layered Mass-Spring Model for Fast Deformations of Flexible Linear Bodies", *Proc. Int. Conf. Computer Animation and Social Agents*, pp. 48-54, May 2003.

[64] D. Hutchinson, M. Preston, and T. Hewitt, "Adaptive Refinement for Mass/Spring Simulation", *Proc. Eurographics Workshop Animation and Simulation*, pp. 31-45, Poitiers, France, Sept. 1996.

[65] K. Montgomery, C. Bruyns, J. Brown, G. Thonier, F. Mazzella, S. Wildermuth, S. Sorkin, A. Tellier, B. Lerman, B. Beedu, and J.C. Latombe, “Spring: A General Framework for Collaborative, Real-time Surgical Simulation”, *Medicine Meets Virtual Reality*, IOS Press, Newport Beach, USA, 2002.

[66] C. Paloc, F. Bello, R. I. Kitney, and A. Darzi, "Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation”, *Proc. Medical Image Computing and Computer-Assisted Intervention*, Tokyo, Japan, pp. 219–226, 2002.

[67] J. Jansson and J.S.M. Vergeest, “Combining Deformable and Rigid-body Mechanics Simulation”, *The Visual Computer*, vol. 19, no. 5, pp. 280-290, Aug. 2003.

[68] J. Jansson, and J.S.M. Vergeest, “A Discrete Mechanics Model for Deformable Bodies”, *Journal of Computer-Aided Design*, vol. 34, no. 12, 2002.

[69] G. Coppini, R. Poli, and G. Valli, “ Recovery of the 3-D Shape of the Left Ventricle from Echocardiographic Images”, *IEEE Trans. Medical Imaging*, vol. 14, issue 2, pp. 301–317, June 1995.

[70] U. Kuhnapfel, H.K. Cakmak and H. Maass, “Endoscopic Surgery Training using Virtual Reality and Deformable Tissue Simulation”, *Computers & Graphics*, vol. 24, pp. 671-682, Elsevier, 2000.

[71] K. S. Choi, H. Sun, and P. A. Heng, “Interactive Deformation of Soft Tissues with Haptic Feedback for Medical Learning”, *IEEE Trans. Information Technology in Biomedicine*, vol. 7, issue 4, pp. 358 – 363, Dec. 2003.

[72] T. Ishikawa, S. Morishima and D. Terzopoulos, “3D Face Expression Estimation and Generation from 2D Image Based on a Physically Constraint Model”, IEICE Transactions on Information and Systems, vol. E83-D, no. 2, 2000.

[73] E. Keeve, S. Girod, R. Kikinis, and B. Girod, “Deformable Modeling of Facial Tissue for Craniofacial Surgery Simulation”, *Computer Aided Surgery*, vol. 3, issue 5, pp. 228 – 238, 1998.

[74] J. Christensen, J. Marks, and J. Ngo, “Automatic Motion Synthesis for 3D Mass-Spring Models”, *The Visual Computer*, vol. 13, pp. 20-28, 1997.

[75] X. Tu and D. Terzopoulos, “Artificial Fishes: Physics, Locomotion, Perception, Behavior”, *Computer Graphics*, vol. 28, pp. 43-50, 1994.

[76] X. Provot, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior", *Graphics Interface*, pp. 147-154, May 1995.

[77] M. Meyer, G. Debunne, M. Desbrun, and A. H. Barr, "Interactive Animation of Cloth-like Objects in Virtual Reality", *The Journal of Visualization and Computer Animation*, vol.12, issue 1, pp. 1-12, May 2001.

[78] Y.-M. Kang, J.-H. Choi, H.-G. Cho and D. Lee, "An Efficient Animation of Wrinkled Cloth with Approximate Implicit Integration, *The Visual Computer Journal*, vol. 17, no. 3, pp. 147-157, 2001.

[79] F. Dachille, H. Qin, A. Kaufman, and J. El-Sana, "Haptic Sculpting of Dynamic Surfaces", *Proc. Symp. Interactive 3D Graphics*, pp. 103-110, 1999.

[80] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk, "Virtual clay: Haptics-Based Deformable Solids of Arbitrary Topology", *Articulated Motion and Deformable Objects*, H.-H. Nagel, F.J. Perales Lopez (Eds.), Lecture Notes in Computer Science, Springer, pp. 1–20, 2002.

[81] S.-W. Chen, G.C. Stockman, and K.-E. Chang, "SO Dynamic Deformation for Building of 3-D Models", *IEEE Trans. Neural Networks*, vol. 7, issue 2, pp. 374–387, 1996.

[82] F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno, "A Multiresolution Model for Soft Objects Supporting Interactive Cuts and Lacerations", *Computer Graphics Forum*, vol. 19, issue 3, pp. 271- 282, 2000.

[83] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method*, McGraw Hill, vol. 1, 1988.

[84] S.C. Chapra and R.P. Canale, *Numerical Methods for Engineers with Software and Programming Applications,* 4th edition, McGraw-Hill, 2002, pp. 858-875, pp. 681-723, pp. 726-750.

[85] S. Cotin, H. Delingette, and N. Ayache, "Real-time Elastic Deformations of Soft Tissues for Surgery Simulation", *IEEE Trans. Visualization and Computer Graphics*, vol. 5, issue 1, pp. 62-73, 1999.

[86] S. Cotin, H. Delingette, and N. Ayache, "A Hybrid Elastic Model Allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation", *The Visual Computer*, vol. 16, issue 8, pp. 437-452, 2000.

[87] D. C. Popescu and M. Compton, "Interaction and VR: A Model for Efficient and Accurate Interaction with Elastic Objects in Haptic Virtual Environments", *Proc. Int. Conf. Computer Graphics and Interactive Techniques in Austalasia and South East Asia*, Melbourne, Australia, pp. 245–250, Feb. 2003.

[88] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable Real-Time Deformations", *Proc. Eurographics Symposium on Computer animation*, pp. 49–54, 2002.

[89] O.R. Astley and V. Hayward, "Multirate Haptic Simulation Achieved by Coupling Finite-Element Meshes through Norton Equivalents", *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 989 – 994, May 1998.

[90] M. A. Audette, A. Fuchs, O. Astley, Y. Koseki, and K. Chinzei, "Towards Patient-Specific Anatomical Model Generation for Finite Element-Based Surgical Simulation", *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, vol. 2673, pp. 332–339, August 2003.

[91] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović, "A Multiresolution Framework for Dynamic Deformations", *Proc. Eurographics Symp. Computer Animation,* pp. 41–47, San Antonio, Texas, July 2002.

[92] G. Debunne, M. Desbrun, A. Barr, and M.P. Cani, "Interactive Multiresolution Animation of Deformable Models", *Eurographics Workshop Computer Animation and Simulation*, Milan, Italy, September 1999.

[93] G. Debunne, M. Desbrun, M.-P. Cani, and A. Barr, "Adaptive Simulation of Soft Bodies in Real-Time", *Computer Animation* 2000, pp. 15-20, May 2000.

[94] H. Kang and A. Kak, "Deforming Virtual Objects Interactively in Accordance with an Elastic Model", *Computer-Aided Design*, vol. 28, issue 4, pp. 251-262, April 1996.

[95] M. Bro-Nielsen, "Finite Element Modeling in Surgery Simulation", *IEEE Surgery Simulation*, pp. 490-503, 1998.

[96] Y. Chen, Q. Zhu and A. Kaufman, "Physically-Based Animation of Volumetric Objects", *Proc. IEEE Computer Animation*, pp. 154-160, 1998.

[97] G. Hirota, "An Improved Finite Element Contact Model for Anatomical Simulations", PhD thesis, University of North Carolina, 2002, available online at http://www.cs.unc.edu/~hirota/fem/.

[98] L. Han, A. Noble, M. Burcher, "The Elastic Reconstruction of Soft Tissues", *Proc. IEEE Int. Symp. Biomedical Imaging*, pp. 1035 –1038, Jul. 2002.

[99] D. Metaxas, and I. A. Kakadiaris, "Elastically Adaptive Deformable Models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1310-1321, 2002.

[100] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically Deformable Models", *Proc. Conf. Computer Graphics and Interactive Techniques*, vol. 21, issue 4, Aug. 1987.

[101] D. Terzopoulos and A. Witkin, "Physically Based Models with Rigid and Deformable Components", *IEEE Computer Graphics and Applications*, vol. 8, no. 6, pp. 41-51, Nov. 1988.

[102] D. Terzopoulos and K. Fleischer, "Modeling Inelastic Deformation: Viscolelasticity, Plasticity, Fracture", *Proc. ACM SIGGRAPH Computer Graphics*, vol. 22, issue 4, pp. 269–278, 1988.

[103] F. Costa and R. Balaniuk, "LEM - An Approach for Real Time Physically Based Soft Tissue Simulation", *Proc. IEEE Int. Conf. Robotics and Automation*, vol.3, pp. 2337 – 2343, 2001.

[104] M. Nakao, H. Oyama, M. Komori, T. Matsuda, G. Sakaguchi, M. Komeda, and T.Takahashi, "Haptic Reproduction and Interactive Visualization of a Beating Heart for Cardiovascular Surgery Simulation", *Int. Journal of Medical Informatics*, vol. 68, issue 1-3, pp. 155-163, Dec. 2002.

[105] C. Mendoza, K. Sundaraj, and C.Laugier, "Faithful Haptic Feedback in Medical Simulators", *Springer Tracts in Advanced Robotics*, vol. 5, pp. 414-423, 2003.

[106] K. Sundaraj, C. Mendoza and C. Laugier, "A Fast Method to Simulate Virtual Deformable Objects with Force Feedback", *IEEE Int. Conf. Automation, Robotics, Control and Vision,* Singapore, vol. 1, pp. 413-418, Dec. 2002.

[107] J. Lang, D.K. Pai, and R. J. Woodham, "Acquisition of Elastic Models for Interactive Simulation", *Int. Journal of Robotics Research*, vol. 21, no. 8, pp. 713-733, 2002.

[108] D. L. James and D. K. Pai, "A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics", *Haptics-e, The Electronic Journal of Haptics Research*, vol. 2, nr. 1, pp. 1-13, Sep. 2001.

[109] D. L. James and D. K. Pai, "Multiresolution Green's Function Methods for Interactive Simulation of Large-scale Elastostatic Objects", *ACM Transactions on Graphics*, vol. 22, issue 1, pp. 47-82, 2002.

[110] U. Meier, C. Monserrat, N.-C. Parr, F. J. García, and J. A. Gil, "Real-Time Simulation of Minimally-Invasive Surgery with Cutting Based on Boundary Element Methods", *Proc. Medical Image Computing and Computer-Assisted Intervention*, W.J. Niessen, M.A. Viergever, (Eds.), Lecture Notes in Computer Science, vol. 2208, p. 1263, 2001.

[111] A. Pentland and J. Williams, "Good Vibrations: Modal Dynamics for Graphics and Animation", *Computer Graphics*, vol. 23, no.3, pp. 207-214, 1989.

[112] A. Pentland and S. Sclaroff, "Closed-Form Solutions for Physically-Based Modeling and Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 715-729, 1991.

[113] D. L. James and D. K. Pai, "DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware", *Proc. ACM SIGGRAPH*, pp. 582-585, San Antonio, Texas, July 2002.

[114] K.K. Hauser, C. Shen, and J.F. O'Brien, "Interactive Deformation Using Modal Analysis with Constraints", *Proc. Graphics Interface*, pp. 247-256, 2003.

[115] R. Szeliski and D. Tonnesen, "Surface Modeling with Oriented Particle Systems", *Proc. ACM SIGGRAPH Computer Graphics*, vol. 26, issue 2, pp. 185–194, 1992.

[116] D. Stahl, N. Ezquerra, and G. Turk, "Bag-of-Particles as a Deformable Model", *Proc. ACM Symp. Data Visualization*, Barcelona, Spain, pp.141–150, 2002.

[117] D.E. Breen, D.H. House, and M.J. Wozny, "Predicting the Drape of Woven Cloth Using Interacting Particles", *Proc. SIGGRAPH'94*, A. Glassner, (Ed.), pp. 365-372, Jul. 1994.

[118] B. Eberhardt, A. Weber, and W. Straßer, "A Fast, Flexible, Particle-System Model for Cloth Draping", *IEEE Computer Graphics and Applications*, vol. 16, no. 5, pp. 52-59, Sep. 1996.

[119] P. Volino, M. Courchesne, and N. Magnenat Thalmann, "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects", *Proc. Conf. Computer Graphics and Interactive Techniques*, pp. 137-144, Sep. 1995.

[120] D. Baraff, A. Witkin, "Large steps in cloth simulation", *Proc. Conf. Computer Graphics and Interactive Techniques*, pp. 43-54, Jul. 1998.

[121] G. Yngve, and G. Turk, "Creating Smooth Implicit Surfaces from Polygonal Meshes", *Technical Report GIT-GVU-99-42*, Georgia Institute of Technology, 1999.

[122] M.Cani-Gascuel and M. Desbrun, "Animation of Deformable Models Using Implicit Surfaces", *IEEE Trans. Visualization and Computer Graphics*, vol. 3, issue 1, pp. 39–50, 1997.

[123] M. Desbrun and M.P. Gascuel, "Animating Soft Substances with Implicit Surfaces", *Proc. Conf. Computer Graphics and Interactive Techniques*, pp. 287–290, 1995.

[124] H. Du, H. Qin, "Interactive Shape Design Using Volumetric Implicit PDEs", *Proc. ACM Symp. Solid Modeling and Applications,* pp. 235–246, Seattle, Washington, USA, 2003.

[125] G.R.Liu, *Mesh Free Methods – Moving beyond the Finite Element Method*, CRC Press, USA, 2003.

[126] M. Muller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, "Point Based Animation of Elastic, Plastic and Melting Objects", *ACM Siggraph Symp. Computer Animation*, R. Boulic, D.K.Pai, (Eds.), pp. 141-150, 2004.

[127] M. Muller-Fischer, R. Keiser, M. Pauly, M. Wicke and M. Gross, "Physics-Based Animation", *Point-Based Graphics*, M. Gross, H. Pfister, (Eds.), pp. 341-387, Morgan-Kaufmann, 2007.

[128] J. Chang and J. Zhang, "Mesh-free Deformations", *Computer Animation and Virtual Worlds*, vol. 15, pp. 211-218, 2004.

[129] M. Muller, B. Heidelberger, M. Teschner, M.Gross, "Meshless Deformations Based on Shape Matching", *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 471-478, 2005.

[130] B. Schmitt, A. Pasko, and C. Schlick, "Shape-Driven Deformations of Functionally Defined Heterogeneous Volumetric Objects", *Proc. Int. Conf. Computer Graphics and Interactive Techniques in Austalasia and South East Asia*, Melbourne, Australia, pp. 127, 2003.

[131] R. Grzeszczuk, D. Terzopoulos, and G. Hinton, "NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models", *Proc. ACM SIGGRAPH*, Orlando, pp. 9-20, 1998.

[132] R. Grzeszczuk, "NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models", PhD Thesis, University of Toronto, 1998.

[133] K. Saito, M. Sase, and Y. Kosugi, "Neural Networks for Simulating the Deformation of Soft Materials", *Proc. Int. Joint Conf. Neural Networks*, pp. 1853 – 1856, vol. 2, 1993.

[134] *Handbook of Neural Computation*, Eds. E. Fiesler, R. Beale, B3, IOP Publishing Ltd and Oxford University Press, 1997.

[135] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1994.

[136] J. Lang, D.K.Pai, and R.J. Woodham, "Acquisition of Elastic Models for Interactive Simulation", *Int. Journal of Robotics Research*, vol. 21, no. 8, pp. 713 - 733, 2002.

[137] A.-M. Cretu, J. Lang, and E.M. Petriu, "A Composite Neural Gas-Elman Network that Captures Real-World Elastic Behavior of 3D Objects", *Proc. IEEE Int. Instr. and Meas. Technology Conf.*, pp. 1063-1067, Sorrento, Italy, April 2006.

[138] J.R. Jang, C.T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, USA, 1997.

[139] J. N. Noyes, "Neural Network Training", in *Handbook of Neural Computation*, E. Fiesler, R. Beale, (Eds.), B3, IOP Publishing Ltd and Oxford University Press, 1997.

[140] M. Riedmiller, H. Braun, "A Direct Adaptive Method for Fast Backpropagation Learning: The RPROP Algorithm", *Proc. IEEE Int. Conf. Neural Networks*, pp. 586-591, San Francisco, CA, 1993.

[141] M.T. Hagan, H.B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Company, USA, 1996.

[142] J.L. Schoner J. Lang, and H.-P. Seidel, "Measurement-Based Interactive Simulation of Viscoelastic Solids", *Computer Graphics Forum*, vol. 23, no. 3, pp. 547-556, 2004.

[143] ATI Industrial Automation Inc, Installation and Operations Manual for Stand-Alone F/T Sensor Systems, Garner, NC, 1997.

[144] Servo-Robot Inc, Jupiter 3-D Laser Vision Camera Installation and Operation Manual, St-Bruno, QC, 1996.

[145] N. Davey, R.G. Adams, and S.J. George, "The Architecture and Performance of a Stochastic Competitive Evolutionary Neural Tree Network", *Applied Intelligence*, vol. 12, pp.75-93, 2000.

[146] B. Fritzke, "Unsupervised Ontogenic Networks", in *Handbook of Neural Computation*, C2.4, E. Fiesler, R. Beale, (Eds.), IOP Publishing Ltd and Oxford University Press, 1997.

[147] A.-M. Cretu and E.M. Petriu, "Neural Network-Based Adaptive Sampling of 3D Object Surface Elastic Properties", *IEEE Trans. Instr. and Meas.*, vol. 55, no. 2, pp. 483-492, 2006.

[148] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten, "Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction", *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558-568, 1993.

[149] M. Daszykowski, B. Walczak, and D.L. Massart, "On the Optimal Partitioning of Data with K-means, Growing K-means, Neural Gas and Growing Neural Gas", *Journal of Chemical Information and Computer Sciences*, vol. 42, pp. 1378-1389, 2002.

[150] "SOM Toolbox Online Documentation", [Online], available online at http://www.cis.hut.fi /projects /somtoolbox/documentation/.

[151] S. Marsland, J. Shapiro, U. Nehmzow, "A Self-Organising Network that Grows When Required", *Neural Networks*, vol. 15, pp. 1041–1058, 2002.

[152] B. Fritzke, "Some Competitive Learning Methods", draft, 1997, available online http://www.neuroinformatik.ruhr-unibochum.de/ini/VDM/research/gsn/ JavaPaper/

[153] M. Dittenbach, D. Merkl and A. Rauber, "The Growing Hierarchical Self-Organizing Map", *Proc. Int. Joint Conf. Neural Networks*, pp. VI-15-VI-19, Como, Italy, 2000.

[154] T.M. Martinetz, "Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps", *Proc. of Int. Conf. Artificial Neural Networks,* Springer, Amsterdam, pp. 427-434, 1993.

[155] J. Holmstrom, "Growing Neural Gas - Experiments with GNG, GNG with Utility and Supervised GNG", Uppsala Master's Thesis in Computer Science, Uppsala University, Sweden, 2002.

[156] E.-M. Petriu, P. Payeur, and A.-M. Cretu, "Haptic Human Interfaces for Robotic Telemanipulation", *Int. Journal of Computing*, vol. 6, issue 2, pp. 81-88, 2007.

[157] A.-M. Cretu, E.M. Petriu and P. Payeur, "Neural Network Mapping and Clustering of Elastic Behavior from Tactile and Range Imaging for Virtualized Reality Applications", *IEEE Trans. Instrumentation and Measurement,* vol. 57, no. 9, pp. 1918-1928, 2008.

[158] A.-M. Cretu, E.M. Petriu and P. Payeur, "Selective Range Data Acquisition Driven by Neural Gas Networks", *IEEE Trans. Instrumentation and Measurement,* in press.

[159] A.-M. Cretu, J. Lang and E.M. Petriu, "Adaptive Acquisition of Virtualized Deformable Objects with a Neural Gas Network", *Proc. IEEE Int. Workshop Haptic Audio Visual Environments and their Applications*, Ottawa, Canada, pp. 165-170, Oct. 2005.

[160] A.-M. Cretu, E.M. Petriu, P. Payeur, "Neural Network Mapping and Clustering of Elastic Behavior from Tactile and Range Imaging for Virtualized Reality Applications", *Int. Workshop Imaging Systems and Techniques*, Italy, pp. 17-22, Apr. 2006.

[161] A.-M. Cretu, P. Payeur and E.M. Petriu, "Selective Vision Sensing with Neural Gas Networks", *Proc. IEEE Int. Instr. and Meas. Technology Conf.*, pp. 478-483, Victoria, Canada, May 2008 (Best Student Paper award).

[162] A.-M. Cretu, E.M. Petriu and P. Payeur, "Evaluation of Growing Neural Gas Networks for Selective 3D Scanning", *Proc. IEEE Int. Workshop on Robotic and Sensors Environments,* pp. 108-113, Ottawa, Canada, Oct. 2008.

[163] A.-M. Cretu, E.M. Petriu and P. Payeur, *"Neural Network Mapping of Elastic Behavior from Tactile and Range Imaging for Virtualized Reality Applications"*, Graduate and Research Day, University of Ottawa, Feb. 2007 (3rd place Research Poster Prize).

[164] A.-M. Cretu, P. Payeur and E.M. Petriu, *"Automated Selective Sensing with Neural Gas Network"*, Discovery 2007, OCE, Toronto, May 2007.

[165] A.-M. Cretu, P. Payeur and E.M. Petriu, *"Selective Vision Sensing with Neural Gas Network"*, Graduate and Research Day, University of Ottawa, Feb. 2008.

[166] A.-M. Cretu, P. Payeur and E.M. Petriu, "Selective Tactile Data Acquisition on 3D Deformable Objects for Virtualized Reality Applications", accepted for publication in *Proc. IEEE Workshop Computational Intelligence in Virtual Environments,* Nashville, TN, USA, May 2009

[167] A.-M. Cretu, P. Payeur and E.M. Petriu, "Comparison of Neural Gas and Growing Neural Gas Networks for Selective 3D Sensing", *Sensors & Transducers*, in press.

[168] A.-M. Cretu, P. Payeur and E.M. Petriu, "Modeling of Elastic Behavior of 3D Deformable Objects from Range and Tactile Imaging", submitted to *Int. Conf. on Image Analysis and Recognition*, Halifax, NS, Canada, Jul. 2009.

[169] A.-M. Cretu, E.M. Petriu and P. Payeur, "Data Acquisition and Modeling of 3D Deformable Objects using Neural Networks", submitted to *IEEE Int. Conf. on Systems, Man, and Cybernetics*, San Antonio, TX, US, Oct. 2009