# Estimation of Deformable Object Properties from Shape and Force Measurements for Virtualized Reality Applications

Ana-Maria Cretu, Emil M. Petriu, Pierre Payeur and Fouad F. Khalil
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada

*Abstract*—**The paper addresses the topic of intelligent sensing and mapping of deformable objects' properties for virtualized reality applications. Shape information in form of contours of soft deformable objects tracked over a sequence of images is correlated to the interaction measurements collected at the level of the fingers of a robotic hand by means of neural networks. The proposed solution allows the automated and implicit modeling of the actual elastic behavior without *a priori* knowledge on the material of an object. It also provides the ability for the application to estimate the shape of an object for unrecorded interactions. Experimental results presented for several soft objects show the ability of the proposed solution to accurately capture and predict severe shape deformations in spite of slight changes in lighting, contrast and background.**

***Keywords-deformable object modeling; contour tracking; robotic hand manipulation; neural networks; virtualized reality.***

## I. Introduction

Virtualized reality environments differ from virtual reality environments in the manner the virtual world object models are constructed. While virtual reality is typically constructed using artificially-created, simplistic models that lack fine details and accurate description of objects' characteristics, virtualized reality starts from the real world scene and virtualizes it [1]. In other words, a virtualized environment contains conformal representations of real world objects based on information captured by a variety of sensors [2]. For soft objects, such representations have to preserve the visible detail of the described real-world scene [3] as well as accurately encompass the elasticity of the corresponding objects. While the problem of rigid objects has been solved, virtualized reality still needs to introduce accurate representations of deformable objects in order to fully reach its potential and to ensure its full usability and functionality. Such representations are critical for a plethora of applications such as medical robotics, interactive virtual environments for training and robotic assembly, and are highly desirable in other applications such as computer gaming.

This paper uses shape and force measurements for the acquisition and mapping of properties characterizing deformable objects. It is a continuation of our work on innovative approaches to design and implement an automated framework to acquire data and model 3D deformable objects for virtualized reality applications. In particular, the work in this paper is based on a previously proposed neuro-inspired algorithm for object segmentation and contour tracking of deformable objects in image sequences [4]. A series of unsupervised neural networks is employed to track the contour of the object over a series of images collected using a camera while the object deforms under the forces imposed by the fingers of a robotic hand. These networks are then associated to the interaction parameters collected at the level of the robotic hand to obtain a complete description of the object's shape deformation. As a significant improvement over the previously proposed solution, the work in this paper provides an original mapping of a deformable object contour and its elastic behavior. It does not only capture the contour of the object and its dynamics under deformation, but also predicts in real-time its shape and implicitly the behavior of an object under previously unrecorded interactions, without imposing particular material models. Such a description greatly enhances the accuracy of the models obtained and represents a significant advantage over existing deformable object models.

## II. Literature Review

Mass-spring models and finite-element methods are still the standard for virtual reality applications as the latest research on the topic proves [5-7]. Mass-spring models are constructed on well-understood dynamics, are relatively simple to build and can often be simulated in real-time. However their structure is often application-dependent and the resulting behavior varies dramatically with different spring configurations and different values of the spring constants. The values of constants are not easy to set or derive from measured material properties and the models obtained have in general low accuracy. Finite-element methods can obtain more accurate models, but they require a very high computation time. Because the force vectors, mass and stiffness matrices are computed by integration over the surface of the object, they have to be re-evaluated each time the object deforms [8]. Due to the fact that 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, leading undoubtedly to less accurate models. Because of these challenges, much of the current research on deformable object models is still based on simplistic computer-generated models with simulated elastic behavior [5, 6].

A natural way to provide more accurate models without assumptions and pre-computations is to interact with them in a controlled manner, observe and then try to mimic as accurately as possible the displayed object behavior. Neural networks, due to their intrinsic non-linearity, their computational simplicity and the ability to learn and generalize are natural candidates for such tasks. In the area of deformable object models, a neural-based solution to learn the behavior of an elastic object subject to an applied force is presented by Greminger *et al.* [9]. Their neural network has as inputs the coordinates of a point over a non-deformed body and the applied load on the body, and as outputs the coordinates of the same point in the deformed body. The non-deformed body point coordinates are obtained by a computer vision deformable body tracking algorithm based on boundary-element method that builds on the equations of the elasticity. A different deformable object modeling approach where the deformation is formulated as a dynamic cellular network that propagates the energy generated by an external force among an object's mass points following Poisson equation is presented in [10].

In the area of robotic grasping and manipulation, neural networks have also received considerable interest due to their capability to learn the complex functions that characterize the grasping and manipulation operations [11-15] and/or to achieve real-time interaction after training [14]. Pedreno-Molina *et al.* [11] integrate neural models to control the movement of a finger in a robotic manipulator based on information from force sensors. A neural network is used by Xia *et al.* [12] to approximate the dynamic system that describes the grasping force-optimization problem of multi-fingered robotic hands (the set of contact forces such that the object is held at the desired position and external forces are compensated). Howard and Bekey [13] represent the viscoelastic behavior of a deformable object according to the Kelvin model and train a neural network for extracting the minimum force required for lifting it. Chella *et al.* [14] use a neuro-genetic approach for solving the problem of three-finger grasp synthesis of planar objects. An object is first segmented from the background, fitted with superellipses and a neural network estimates an approximation of the grasp. This approximation is refined with a genetic algorithm. A hierarchical self-organizing neural network to select proper grasping points in 2D is proposed in [15].

In the context of this work, neural architectures are chosen for similar reasons to those mentioned above, namely their capability to store (offline) and predict (online, in real-time) the complex relationship between the deformation of the object and the interaction parameters at each robotic finger. Unlike the other neural network solutions, the proposed approach neither imposes a certain representation of the deformable object [13, 14], nor requires certain equations to model the elastic behavior [9, 10] or certain dynamic models at the points of contact [12]. The proposed solution innovatively combines neural architectures to identify an object of interest, to track its contour in visual data and to associate and predict its shape under a certain interaction exercised with a robotic hand.

## III. Proposed Framework

In order to map the elastic properties of a deformable object, its controlled interaction with a robotic hand is observed by means of shape and force data. Its elastic behavior is then learned as a complex relationship between the interaction parameters measured at the level of the robotic hand and the deformed contour of an object obtained from the visual data. Neural network approaches are used both to segment and track the object contour in the image sequence [4] and to capture implicitly the complex relationship between the object's contour deformation and the forces exercised on the object through the robotic fingers. The choice to use a supervised (feedforward neural network) architecture for capturing the relationship between the interaction parameters at the level of the robotic hand and the corresponding contours is justified by the capability of the neural network to eliminate the need for predefined elastic parameters of the object. Most of the objects under study are made of soft, highly deformable material whose elastic behavior is very difficult to be described in terms of standard elastic parameters. The choice of a neural-network approach also ensures the ability of the application to estimate the contour of an object for previously unseen and untrained combinations of interaction parameters.

### A. Experimental Setup

A setup composed of a Barrett robotic hand and a camera situated perpendicularly to the surface of an object manipulated with the hand is used to collect the shape deformation and the interaction parameters. As shown in Fig. 1, the robotic hand is positioned with the palm up in order to allow the observation of the interaction without unnecessary occlusions.

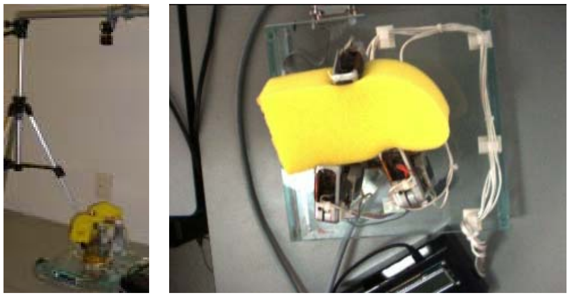

Figure 1. Setup with camera and Barrett hand.

Two interaction parameters are recorded at each finger. One corresponds to the position, $P_{ij}$, of each fingertip and it is represented by the number of pulses in the encoder that reads the angle of the motor that drives the finger. This measurement is referred to as "position" measurement to simplify the explanations and create a compact representation of the hand motion. It is equivalent to the Cartesian coordinates of the fingertip, using the Barrett hand kinematic model. The second parameter is a measure of the interaction force, $F_{ij}$, applied at each fingertip and obtained via strain gauges embedded in each finger. It will be called hereon the "force" measurement as the strain value can be converted to equivalent physical force measurements, in kg or N, through proper calibration. These interaction parameters are collected simultaneously with an image sequence of the object's shape deformation as captured by the camera. Measurements are collected for different force magnitudes applied on a set of test objects made of deformable materials. The force and position measurements are then associated with the tracked contour of the object in the image sequence using an innovative combination of neural network architectures.

### B. *Visual Data Processing*

The proposed solution for the acquisition and mapping of the elastic properties characterizing deformable objects is summarized in Fig. 2.

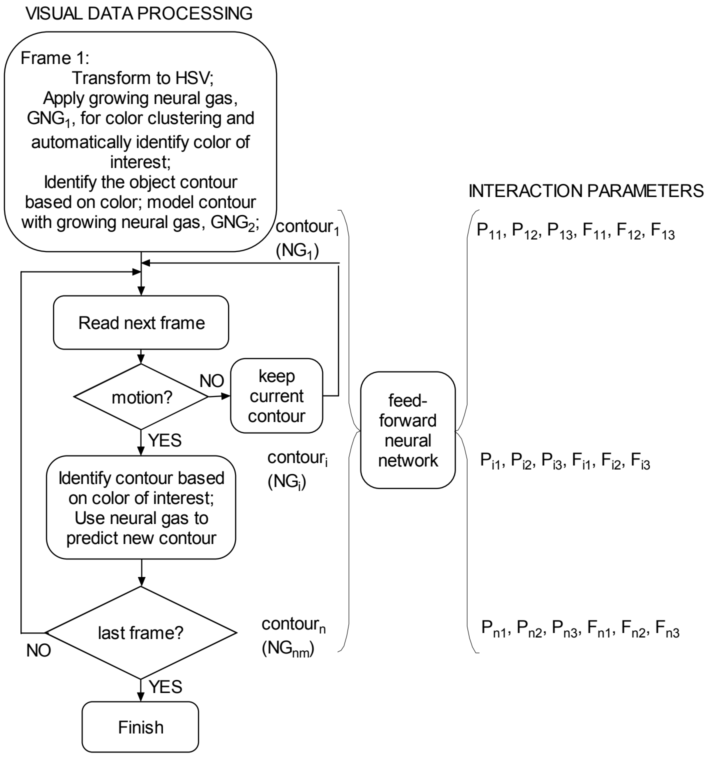
VISUAL DATA PROCESSING
Frame 1:
Transform to HSV;
Apply growing neural gas, GNG1, for color clustering and automatically identify color of interest;
Identify the object contour based on color; model contour with growing neural gas, GNG2;
contour1 (NG1)
Read next frame
motion?
NO
keep current contour
YES
Identify contour based on color of interest; Use neural gas to predict new contour
contouri (NGi)
last frame?
NO
YES
Finish
contourn (NGnm)
feed-forward neural network
INTERACTION PARAMETERS
P11, P12, P13, F11, F12, F13
Pi1, Pi2, Pi3, Fi1, Fi2, Fi3
Pn1, Pn2, Pn3, Fn1, Fn2, Fn3

Figure 2. Proposed framework for acquisition and mapping of properties characterizing deformable objects.

The segmentation and tracking algorithm, depicted in the left side of Fig. 2, can be briefly summarized as follows [4]: the object of interest is automatically segmented from the initial frame of the sequence of images collected by the camera. A growing neural gas, *$GNG_1$*, maps the color (HSV coding) and spatial components (*X, Y* coordinates) of each pixel of the initial frame in the video sequence. The clustering map obtained by *$GNG_1$* is split into two categories: object of interest and background. The color of interest is then automatically computed as the mean for all HSV values within the cluster representing the object of interest. The identified HSV color code is then searched over every subsequent frame in the sequence where some movement occurs. The contour of the object is identified after straightforward image processing with a Sobel edge detector.

A second growing neural gas, *$GNG_2$*, is employed to represent the position of each point over the contour with the main purpose to detect the optimum number of points, $c_n$, on the contour that accurately represent its geometry. This compact growing neural gas description of the contour is then used as an initial configuration for a sequence of neural gas networks, *$NG_i$*, that track the contour over each frame in the image sequence in which motion occurs.

A new neural gas network, initialized with the contour of the object in the previous frame, is used to predict the new position of its neurons and to readjust them to fit the new contour. This new contour will then be used iteratively to initialize the next neural gas network in the sequence. As illustrated in Fig. 2, the procedure is repeated until the last frame of the sequence, resulting in $n_m$ separate neural gas networks, as determined by the number of frames exhibiting motion. The full description of the object segmentation and contour tracking algorithm is presented in [4]. These networks will later be associated to the measured interaction parameters, as shown in the right side of Fig. 2, for a more comprehensive description of the object's deformation.

### C. *Mapping of the Contours with Interaction Parameters*

A feedforward neural network is employed to map the $n_m$ contours extracted from the sequence, as obtained in Section III.B, with the interaction parameters, as defined in Section III.A. The network modeling an object has six input neurons associated with the interaction parameters namely: the position of the three fingers ($P_{i1}, P_{i2}, P_{i3}$) and the force measurements at each fingertip ($F_{i1}, F_{i2}, F_{i3}$), as shown in Fig. 3.

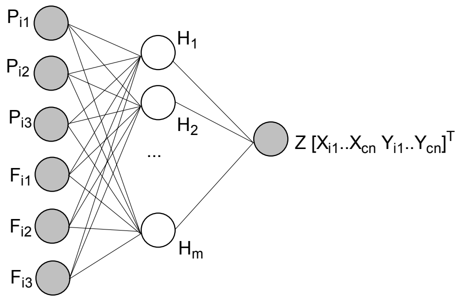

Figure 3. Neural architecture to store and predict the object contour shape based on the position of the robotic fingers and force measurements.

A number of 30 hidden neurons is used for all the objects under study and this number is identified such that it ensures a good compromise between the length of training and the accuracy of modeling. The output vector is the set of coordinates for the points on the contour. In order to contain the concatenated vectors of *X* and *Y* coordinates for each point, its size is the double of the number of points, $c_n$, in the contour. This is also the number of nodes in the second growing neural gas network, *$GNG_2$*, which defines those contours, and in the series of neural gas networks, *$NG_i$*.

The only preprocessing applied on the input is a normalization to the [0 1] interval prior to training. Three quarters of the data available (values for $P_{ij}$ and $F_{ij}$) is used for training and a quarter for testing. The network for each object under study is trained for 150,000 epochs using the batch version of scaled conjugate gradient backpropagation algorithm [16] with the learning rate set to 0.1. Once trained, the network takes as inputs the interaction parameters ($P_{i1}$, $P_{i2}$, $P_{i3}$, $F_{i1}$, $F_{i2}$, $F_{i3}$) and outputs the corresponding contour that the object should exhibit under the current interaction configuration.

## IV. Experimental Results

Experimental evaluation was conducted with the setup presented in Fig. 1 on a set of deformable objects with different shapes and colors, of which a limited set is presented here, namely a green rectangular foam sponge, an orange foam ball, and a smoothly curved yellow foam sponge. Fig. 4 summarizes the object segmentation and tracking stage for the rectangular sponge, as explained in Section III.B and illustrated in the left side of Fig. 2. It starts with the automated identification of the color of interest with $GNG_1$, continues with the modeling of the contour with an optimal number of nodes, $c_n$, using the second growing neural gas, $GNG_2$, over the initial frame, and concludes with the tracking over a series of frames using a sequence of neural gas networks, $NG_i$.

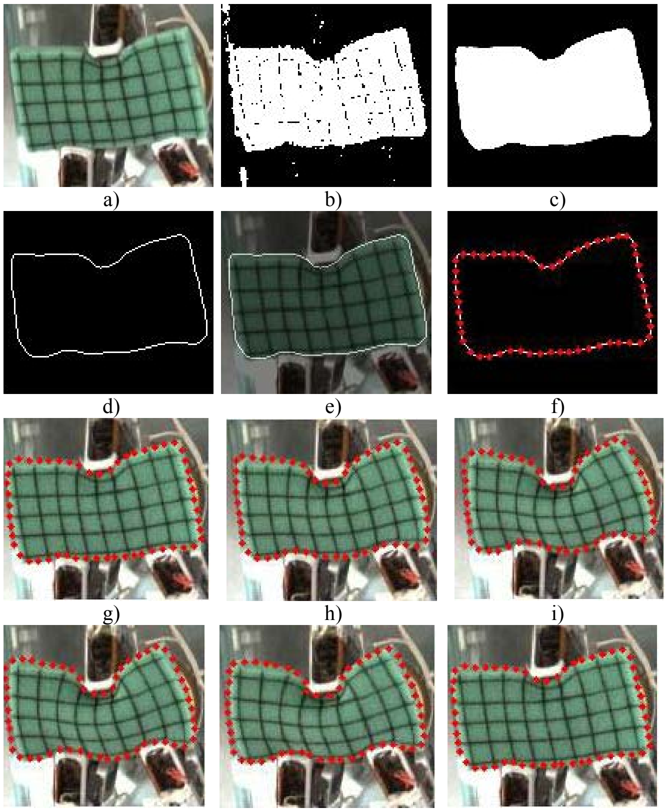

Figure 4. Object segmentation and tracking stage for the rectangular sponge: a) initial frame, b) identification of the color of interest, c) results when a tolerance level of 0.07 for H and S and 0.01 for V value is accepted and after a median filtering is applied, d) contour identification, e) contour overlapped on initial frame, f) growing neural gas model of contour in the initial frame and g) - l) contour tracking using a series of neural gas networks.

After initialization, the computation time required to track the objects is low, on average 0.35s per frame when running on the Matlab platform. The average error (measured as the Hausdorff distance from the points in the contour obtained with the Sobel edge detector and the modeled neural gas points) is of the order 0.215. Fig. 5 illustrates the average computation time per frame (in seconds) as well as the error incurred during tracking for each of the objects under study. The error is slightly higher for objects that deform rapidly from one frame to the other and/or roll during probing, as for the case of the ball.

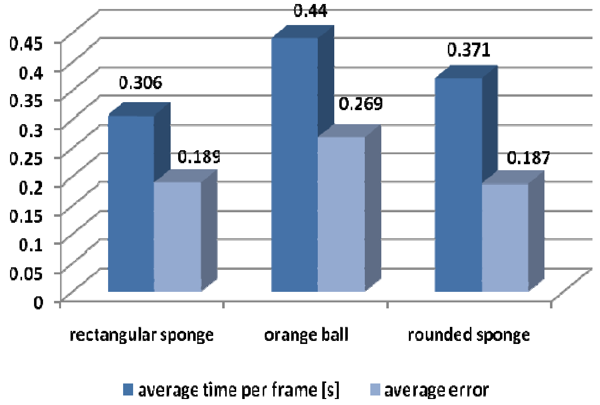

Figure 5. Average time per frame and average error for tracking points.

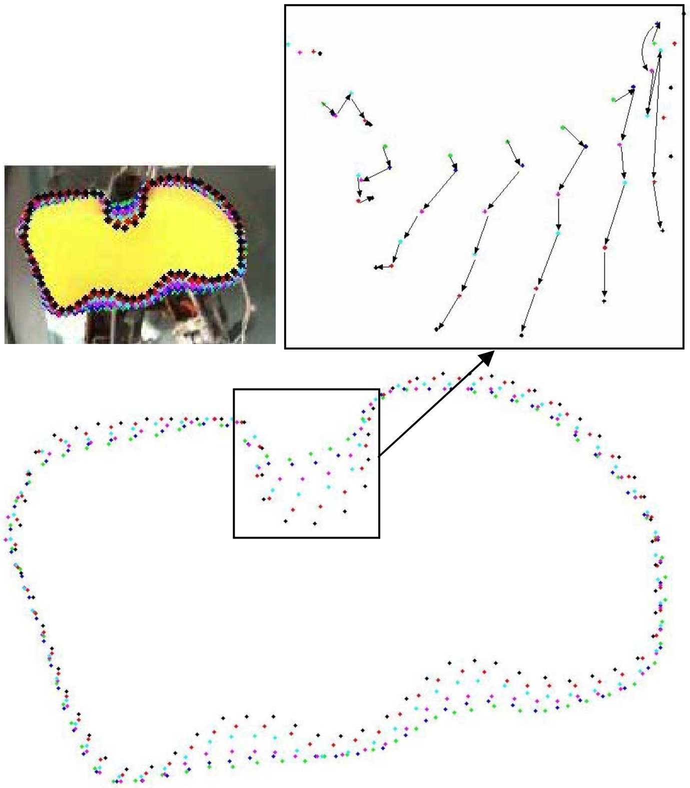

Figure 6. Continuous tracking of points during the compression of the yellow rounded sponge.

Fig. 6 illustrates a part of the complex trajectory that the points in the neural gas model of the yellow rounded sponge follow during the interaction with the robotic hand. It can be observed in the detailed image, where the trajectory is marked with arrows, that due to the choice of a fixed number of nodes in the neural gas network, $NG_i$, and with the proposed learning mechanism, the nodes in the contour retain their correspondence throughout the deformation. This one-to-one correspondence of the points in the trajectory helps to avoid their mismatch during deformation and ensures a unified description of the contour deformation throughout the entire sequence of images, a property that is not offered by any standard approach for tracking.

The sequence of contours is mapped with the corresponding interaction parameters at the level of the robotic hand by means of a feedforward neural network as described in Section III.C. Figs. 7 and 8 illustrate the performance of the neural network approach for the ball and for the rectangular sponge.

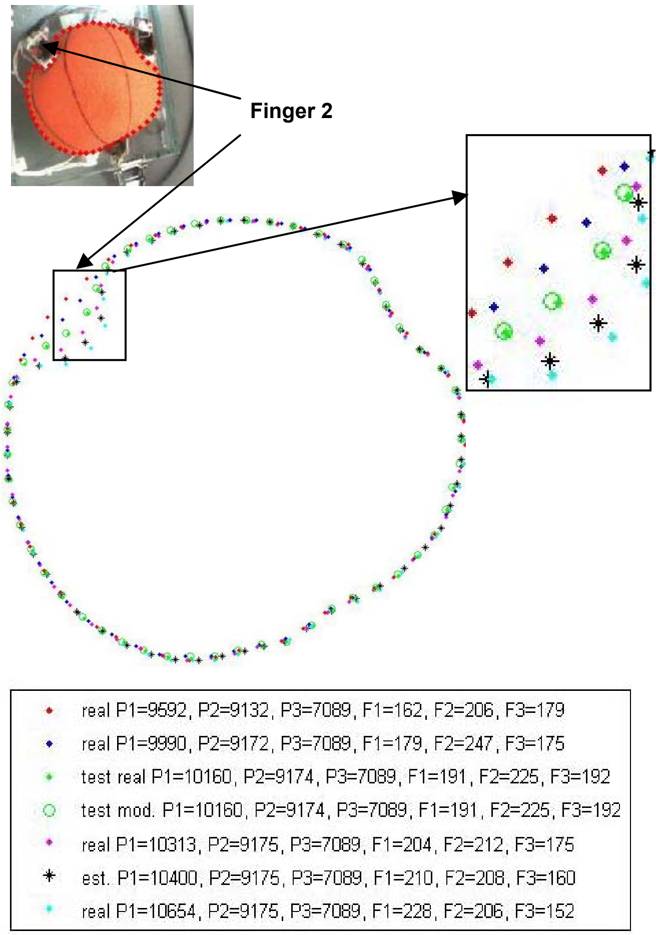

Figure 7. Real, modeled and estimated contour points and details for the ball.

The training/learning error for the neural networks corresponding to these two objects is of the order $5\times10^{-5}$, illustrating the capability of the network to accurately map the interaction parameters to the corresponding deformed contour. The testing error (the error obtained on the testing set) is of the order $4\times10^{-3}$.

Fig. 7 shows real data (marked with dots), results for testing data (green circles) and an estimated deformed contour for a set of forces that were not part of the training set, in order to test the prediction capability of the network. For example, in this experimental scenario, the finger positions are kept almost at the same position, but the force at second finger ($F_2$=208) is slightly decreased from the value in the magenta dot contour ($F_{2\_magenta}$=212), while kept above the value in the cyan curve ($F_{2\_cyan}$=206). The estimated profile depicted with black stars is placed, as expected, in between the magenta and cyan contours, but closer to the cyan one to which the force value is closer.

Another example is presented for the rectangular sponge in Fig. 8. In this example, the network provides an estimate for a value of the force at the third finger ($F_3$=168) that is in between the one of the cyan contour ($F_{3\ cyan}$=164) and the magenta contour ($F_{3\ magenta}$=171), associated as well with a slight movement in the position of the fingers.

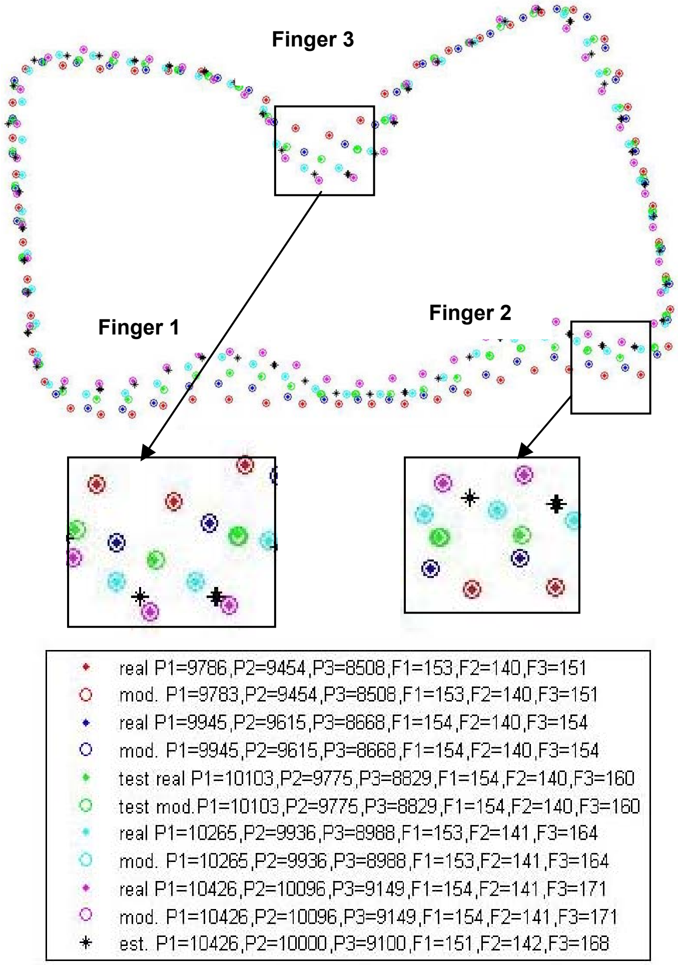

Figure 8. Real, modeled and estimated contour points and details for the rectangular sponge.

Moreover, one can notice multiple interactions and changes in the contour that occur due to increased forces at different fingers. For example, while the forces at Fingers 1 and 2 in Fig. 8 are almost unchanged, an increased force at Finger 3 creates a bending in the object that appears to be due the forces applied at Fingers 1 and 2.

A careful observation leads to the conclusion that the estimate is still correctly placed around the extra fingers under these conditions. The proposed approach is therefore able not only to capture the contour, but also to predict its shape in spite of the coupling between multiple finger interactions.

## V. Conclusion

This paper demonstrates the benefit of using neural networks approaches for mapping and predicting the shape of soft deformable objects from actual measurements collected by a robotic hand and a camera. The proposed combination of networks for segmentation and tracking runs fast, with low errors and guarantees the continuity of points in the tracked contours. The neural approach used for the modeling and prediction of contour shapes based on force measurements and the position of the robotic fingers ensures that the application handles properly the situations on which the system was not trained. As future work, the study will be expanded for different orientations of the robot fingers for a more extensive description of the interaction and for objects with diverse elastic properties.

## References

[1] T. Kanade, P.J. Narayanan and P.W. Rander, “Virtualized Reality: Concepts and Early Results”, *Proc. IEEE Workshop Representation of Visual Scenes*, pp. 69–76, 1995.

[2] E.M. Petriu, "Neural Networks for Measurement and Instrumentation in Virtual Environments," Neural Networks for Instrumentation, Measurement and Related Industrial Applications, S. Ablameyko, L. Goras, M. Gori, and V. Piuri (eds.), NATO Science Series, Series III: Computer and System Sciences, vol. 185, pp. 273-290, IOS Press, 2003.

[3] T. Kanade, P. Rander and P.J. Narayanan, “Virtualized Reality: Constructing Virtual Worlds from Real Scenes”, *Proc. IEEE Conf. Visualization*, pp. 277-283, 1997.

[4] A.-M. Cretu, E.M. Petriu, P. Payeur and F. F. Khalil, “Deformable Object Segmentation and Contour Tracking in Image Sequences Using Unsupervised Networks“, *Proc. Seventh Canadian Conference on Computer and Robot Vision*, pp. 277-284, Ottawa, Canada, May 2010.

[5] H. Wang, Y. Wang and H. Esen, “Modeling of Deformable Objects in Haptic Rendering System for Virtual Reality”, *Proc. IEEE Int. Conf. Mechatronics and Automation*, pp. 90-94, Changchun, China, 2009.

[6] Q. Luo and J. Xiao, “Modeling and Rendering Contact Torques and Twisting Effects of Deformable Objects in Haptic Interaction”, *Proc. Conf. Intel. Robots and Systems*, pp. 2095-2100, San Diego, USA, 2007.

[7] C. Duriez, C. Andriot and A. Kheddar, “Signorini’s Contact Model for Deformable Objects in Haptic Simulations”, *Proc. IEEE/IRSJ Int. Conf. Intelligent Robots and Systems*, pp. 3232-3237, Sendai, Japan, 2004.

[8] S.F.F. Gibson and B. Mirtich, “A Survey of Deformable Modeling in Computer Graphics”, Technical report, Mitsubishi Electric Research Laboratory, 1997.

[9] 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.

[10] Y. Zhong, B. Shirinzadeh, G. Alici and J. Smith, “Cellular Neural Network Based Deformation Simulation with Haptic Force Feedback”, *IEEE Workshop Advanced Motion Control*, Istanbul, Turkey, pp. 380-385, 2006.

[11] J. L. Pedreño-Molina, A. Guerrero-González , J. Calabozo-Moran, J. López-Coronado and P. Gorce, “A Neural Tactile Architecture Applied to Real-time Stiffness Estimation for a Large Scale of Robotic Grasping Systems”, *Intelligent Robot Systems*, vol. 49, pp. 311–323, 2007.

[12] Y. Xia, J. Wang, and L.-M. Fok, “Grasping-Force Optimization for Multifingered Robotic Hands Using a Recurrent Neural Network”, *IEEE Trans. Robotics and Automation*, vol. 26, no. 3, pp. 549-554, 2004.

[13] A.H. Howard and G. Bekey, “Intelligent Learning for Deformable Object Manipulation”, *Autonomous Robots*, vol. 9, pp. 51-58, 2000.

[14] A. Chella, H. Dindo, F. Matraxia, and R. Pirrone, “Real-Time Visual Grasp Synthesis Using Genetic Algorithms and Neural Networks”, R. Basili and M.T. Pazienza (Eds.), *Artificial Intelligence and Human-Oriented Computing,* Lecture Notes in Artificial Intelligence, vol. 4733, pp. 567–578, Springer, 2007.

[15] G.L. Foresti and F.A. Pellegrino, “Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation”, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 34, pp. 325-333, 2004.

[16] M. F. Moller, “A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning”, *Neural Networks*, vol. 6, pp. 525-533, 1993.