Learning and Prediction of Soft Object Deformation Using Visual Analysis of Robot Interactions

Ana-Maria Cretu, Pierre Payeur, and Emil M. Petriu

School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada

Abstract. The paper discusses an innovative approach to acquire and learn deformable objects' properties to allow the incorporation of soft objects in virtualized reality applications or the control of dexterous manipulators. Contours of deformable objects are tracked in a sequence of images collected from a camera and correlated to the interaction measurements gathered at the fingers of a robotic hand using a combination of unsupervised and supervised neural network architectures. The advantage of the proposed methodology is that it not only automatically and implicitly captures the real elastic behavior of an object regardless of its material, but it is also able to predict the shape of its contour for previously unseen interactions. The results obtained show that the proposed approach is fast, insensitive to slight changes in contrast and lighting, and able to model accurately and predict severe contour deformations.

1 Introduction

Unlike virtual reality environments that are typically made of simplistic CAD models, a virtualized environment contains models that constitute conformal representations to real world objects [1]. Such representations have to preserve the visible details of the described real-world object and to accurately capture its properties. While several approaches exist to the modeling of the behavior of rigid objects, virtualized reality still needs to introduce accurate representations of deformable objects in order to fully reach its usability and functionality. Such representations are highly desirable in applications such as computer gaming and interactive virtual environments for training, and critical for other applications such as robotic assembly and medical robotics.

Inspired by the human experience with object manipulation where the ability of vision interacts with the servo-muscular and touch sensory systems for every day manipulation tasks, the work in this paper uses visual information and interaction parameters measured at the level of the fingers of a robotic hand for the acquisition and mapping of properties characterizing soft deformable objects. The solution extends a previously proposed algorithm that tracks deformable object contours in image sequences captured by a camera while the object deforms under the forces imposed by the fingers of a robotic hand [2]. Each contour is associated to the corresponding measured interaction parameters to characterize the object's shape deformation and implicitly describe its elastic behavior without knowledge on the material that the object is made of. Due to the choice of neuro-inspired approach used for

G. Bebis et al. (Eds.): ISVC 2010, Part II, LNCS 6454, pp. 232–241, 2010.

modeling, the solution not only captures the dynamics of the deformation, but is also capable to predict in real-time the behavior of an object under previously unrecorded interactions. Such a description enhances the accuracy of the models obtained and represents a significant advantage over existing deformable object models.

2 Related Work

As the latest research proves [3, 4], much of the current research on deformable models is still based on computer-generated models. The classical mass-spring models and finite-element methods still constitute the standard for virtual reality applications. In spite of their simplicity and their real-time simulation ability, mass-spring models are application-dependent, their behavior varies dramatically according to the choice of spring constants and their configuration. Also the models obtained have in general low accuracy. Finite-element methods can obtain more accurate models, but they are more complex and the very high computational time incurred as the object deforms (force vectors, mass and stiffness matrices are re-evaluated each time the object deforms) is a serious obstacle for their use in real-time applications.

An alternative solution to ensure better conformance to the reality of the deformable object model without requiring the pre-selection of elastic parameters or material properties (as in the case of mass-spring models) or requiring increased complexity and excessive computation times (as in the case of finite-element methods), is to interact with the objects in a controlled manner, observe and then try to mimic as accurately as possible the displayed object behavior. Neural networks are well-fitted for such tasks, as they are computationally simple, have the ability to map complex and non-linear data relationships and have the ability to learn and then predict in real-time the displayed behavior. This explains the interest of researchers from both the deformable object modeling [5, 6] and the grasping and manipulation research fields [7-11] into such techniques.

In the area of deformable object models, object 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 in [5]. Greminger *et al.* [6] learn the behavior of an elastic object subject to an applied force, by means of a neural network which has as inputs the coordinates of a point over a non-deformed body (obtained by a computer vision tracking algorithm based on boundary-element method that builds on the equations of the elasticity) and the applied load on the body, and as outputs the coordinates of the same point in the deformed body.

In the area of robotic grasping and manipulation, neural networks have been employed to learn the complex functions that characterize the grasping and manipulation operations [7-11] and to achieve real-time interaction after training [10]. A neural network is used in [7] 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). Pedreno-Molina *et al.* [8] integrate neural models to control the movement of a finger in a robotic manipulator based on information from force sensors. Howard and

Bekey [9] 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. A hierarchical self-organizing neural network to select proper grasping points in 2D is proposed in [10]. Chella *et al.* [11] use a neuro-genetic approach for solving the problem of three-finger grasp synthesis of planar objects.

Neural network architectures are chosen in the context of this work for reasons similar to those mentioned above, namely their capability to store (offline) and predict (online) the complex relationship between the deformation of the object and the interaction parameters at each robotic finger. However, unlike the other neural network approaches encountered in the literature, the proposed approach neither imposes certain equations to model the elastic behavior [5, 6] or certain dynamic models at the points of contact [7], nor requires a certain representation of the deformable object [9, 11]. The proposed solution combines in an original manner neural architectures to identify an object of interest, to track its contour in visual data and to associate and predict its shape under interactions exercised with a robotic hand.

3 Proposed Solution

In order to map the properties of an elastic object, its controlled interaction with a robotic hand is monitored by means of visual data collected with a camera, while additional measurements are collected using sensors installed in the three fingers of a Barrett robotic hand. The elastic properties are then mapped and learned as a complex relationship between the deformed contour of an object obtained from the visual data and the corresponding interaction parameters, as illustrated in Fig. 1 that summarizes the proposed approach. Neural network approaches are used both to segment and track the object contour in the image sequence [2] and to capture implicitly the complex relationship between the object's contour deformation and the force exercised on the object through the robotic fingers at defined finger positions. The choice to use a supervised (feedforward neural network) architecture for mapping the relationship between the interaction parameters and the corresponding contours is justified by the capability of the neural network to eliminate the need for predefined elastic parameters of the object or predefined object models. This aspect is essential in the proposed application, as most of the objects used for experimentations 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 combinations of interaction parameters.

It is important to mention that the experimentation takes place in a relatively controlled environment, as data is collected separately for each modeled object. The solution does not have to deal with multiple moving objects and severe changes in the environment, but rather focuses on accurately tracking severe contour deformations that describe the object's behavior. For a proper identification of the object's elastic properties, it is considered that the object has already been grasped by the three robot fingers. Balanced grasping forces are initially applied to maintain the object within

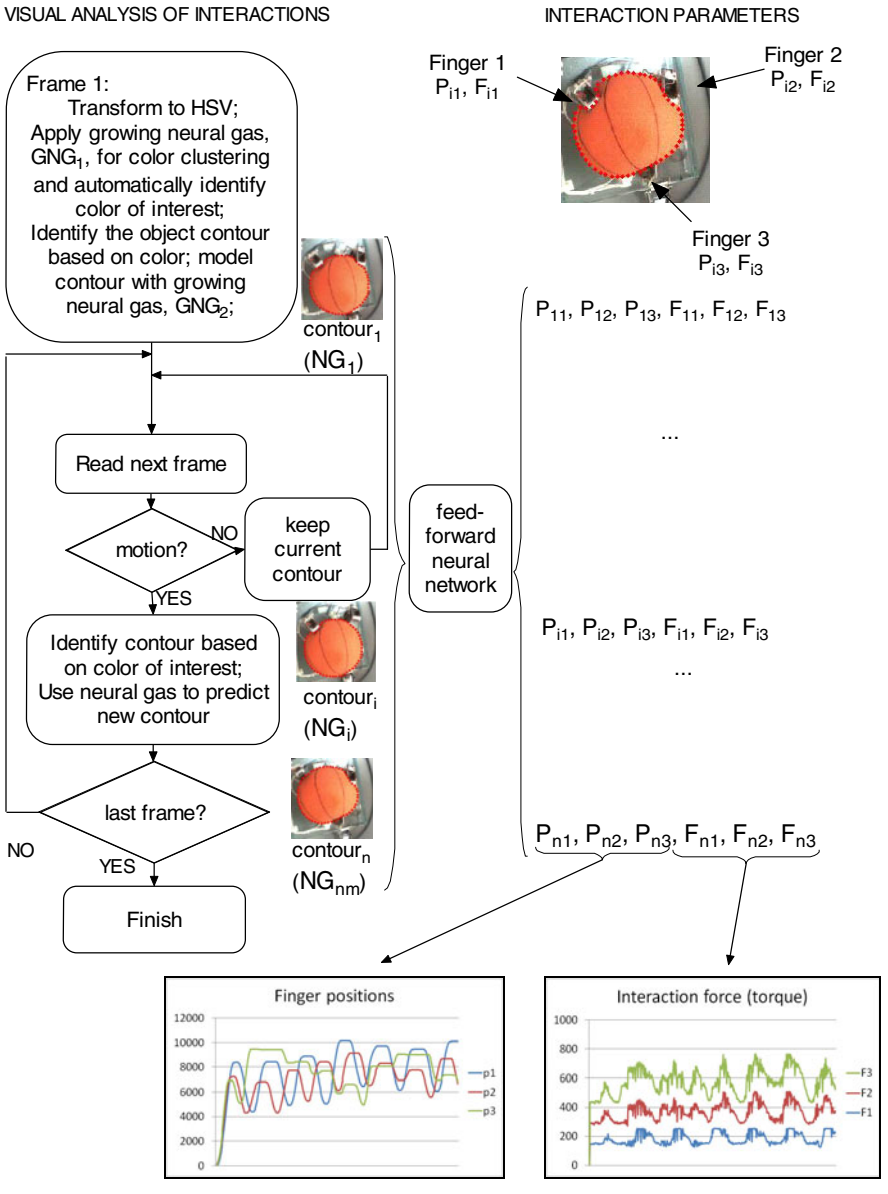

Fig. 1. Proposed framework to acquire, capture and predict the object contour shape based on the position of the robotic fingers and force measurements at level of the robotic fingers

the hand grip without slippage. The hand then compresses the object repetitively by contracting and relaxing its fingers over a period of several sampling periods.

The test is carried out using Barrett hand's real-time mode which allows the input fed to the hand system to generate a periodic movement of the fingers in order to

properly excite the object and extract its dynamic characteristics. The motion of the hand's three fingers is defined to follow a sinusoidal profile defined in encoder pulses. Two interaction parameters are recorded at each finger at every sampling instant, as illustrated in the lower part of Fig. 1. One corresponds to the position, P_{ij}, of each fingertip and 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. It is equivalent to the Cartesian coordinates of the fingertip using the Barrett hand's kinematic model. The second parameter is a measure of the interaction force (torque), F_{ij}, applied at each fingertip and obtained via strain gauges embedded in each of the three fingers. It will be called hereon the "force" measurement, as the strain value can be converted to equivalent physical force measurements through proper calibration. These interaction parameters are collected simultaneously with an image sequence of the object's contour deformation as captured by the camera. Measurements are collected on a set of test objects made of soft 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.

3.1 Visual Analysis of Interaction

The segmentation and tracking algorithm applied on visual data is illustrated in the left side of Fig. 1 in form of a flowchart and can be summarized as follows [2]: the initial frame of the sequence of images collected by the camera is used to identify automatically the object of interest by clustering the color (HSV coding) and spatial components (*X, Y* coordinates) of each pixel in this frame in two categories: object and background. The clustering is based on an unsupervised neural architecture, a growing neural gas, denoted GNG_1. The reason for choosing an unsupervised network is the fact that, beyond being automated, it results in lower error rates when compared with a standard segmentation technique based on mean HSV values computed in a user selected frame that samples the object color. 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 subsequently searched only over frames in the sequence where movement occurs to speed up the processing. The motion is detected based on intensity difference between the grayscale representations of the current and previous frames. The contour of the object is identified after straightforward image processing with a Sobel edge detector.

A second growing neural gas, GNG_2, is used to map and represent the position of each point over the contour. Its main purpose is 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, whose purpose is to 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 and adjust the position of its neurons to fit the new contour. This new contour is used iteratively to initialize the next neural gas network in the sequence. The procedure is repeated until the last frame of the sequence, as illustrated in the flowchart of Fig. 1, 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 [2]. The n_m contours representing each neural gas network are further associated to the measured interaction parameters for a comprehensive description of the object's deformation.

3.2 Mapping of Contours with Interaction Parameters

The n_m contours extracted from the sequence of images, as obtained in Section 3.1 are mapped with the interaction parameters using a feedforward neural network. The network capturing the behavior of each deformable 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 (F_{i1}, F_{i2}, F_{i3}) at each fingertip, as shown in the right side of Fig. 1. A number of 30 hidden neurons was experimentally selected to ensure a good compromise between the length of training and the accuracy of modeling for all the objects used in the experimentation. The output vector is the set of coordinates for the points on the contour as obtained by tracking in the image sequence. It contains concatenated vectors of X and Y coordinates for each point in the contour and therefore 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 input vectors (sets of P_{ij} and F_{ij} values) are normalized prior to training, and three quarters of the data available is used for training and a quarter for testing. Each network is trained using the batch version of scaled conjugate gradient backpropagation algorithm with the learning rate of 0.1 for 150,000 epochs. 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 configuration of interaction parameters.

4 Experimental Results

The proposed method has been applied on a set of deformable objects with different shapes and colors, of which a limited set is presented here, namely an orange foam ball and a green rectangular foam sponge. Fig. 2 illustrates the object segmentation and tracking algorithm for the ball, as described in Section 3.2 and illustrated on the left side of Fig. 1. The procedure starts with the automated identification of the color of interest (Fig. 2b) by clustering with GNG_1 the initial frame (Fig. 2a). A tolerance level (defined as the maximum distance for each component of the HSV coding from the mean HSV code that identifies the color of interest) is allowed to eliminate the effect of non-uniform illumination and shadow effects, e.g. color reflected by the shining fingers (Fig. 2c). The contour is then identified using Sobel edge detector (Fig. 2d and 2e) and modeled with an optimal number of nodes, c_n, using the second growing neural gas, GNG_2 (Fig. 2f). Finally the object is tracked over the series of frames using a sequence of neural gas networks. The computation time required to track the objects is low, on average 0.35s per frame on the Matlab platform. The average error, measured as the Hausdorff distance between the points in the contour

obtained with the Sobel edge detector and the modeled neural gas points is of the order 0.215. The Hausdorff distance is chosen because it allows the comparison of curves with different number of points, as it is the case between the neural gas model and the Sobel contour.

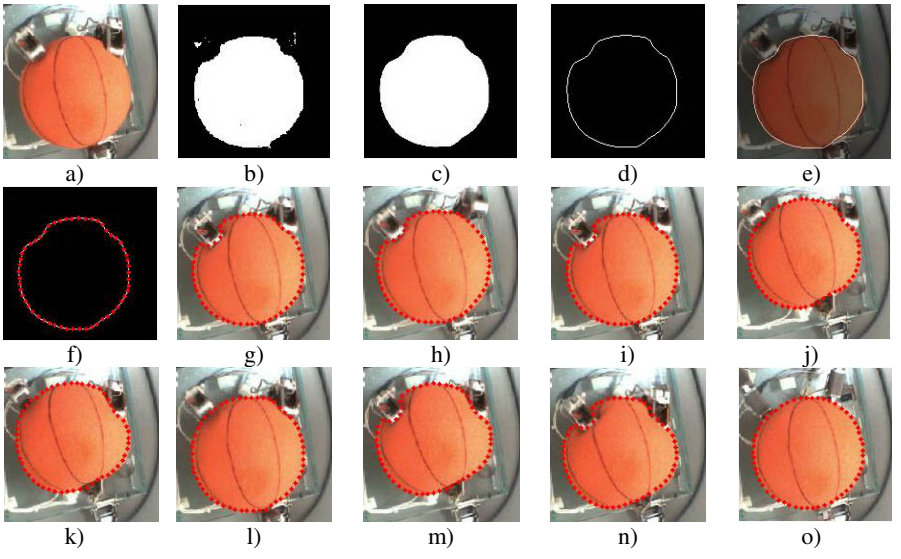

Fig. 2. Object segmentation and tracking based on unsupervised architectures: a) initial frame, b) identification of the color of interest, c) results when a tolerance level is accepted, d) contour identification, e) contour overlapped on initial frame, f)-g) growing neural gas model of contour in the initial frame and g) - o) contour tracking using a series of neural gas networks

Table 1 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 or roll during probing, as in the case of the ball illustrated in Fig. 2. It can be observed that the tracking procedure is fast and follows closely the contours of the objects. A typical characteristic of the proposed tracking algorithm is the fact that the nodes in the contour retain their correspondence throughout the deformation. This behavior is due to the choice of a fixed number of nodes in the neural gas network, NG_i, and to the proposed learning mechanism. Fig. 3a details a part of the trajectory (marked with arrows) that the points in the neural gas model of the ball follow as a result of the interaction with the robotic hand.

Table 1. Average error and average time per frame for the tracking algorithm

Object	Average error	Average time per frame [s]
Round ball	0.269	0.44
Rectangular sponge	0.189	0.306
Yellow curved sponge	0.187	0.371

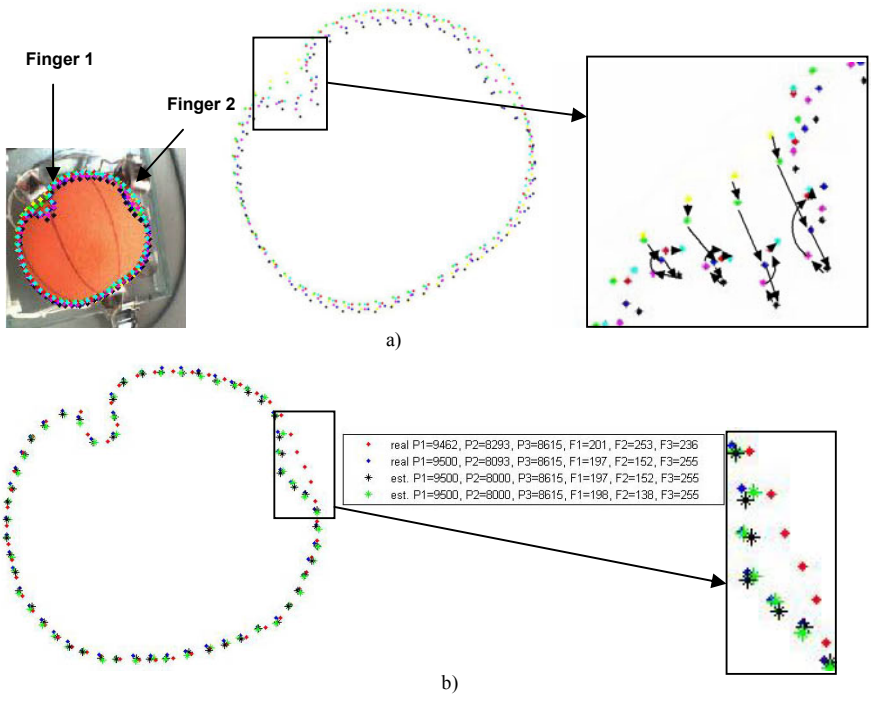

Fig. 3. a) Trajectory of nodes during compression and slight depression of ball and b) real (*dot*) and estimated (*star*) contour points and detail for the ball

It can be seen in the detailed image, that there is a one-to-one correspondence of the points in each contour. This correspondence ensures a unified description of the contour deformation throughout the entire sequence of images and at the same time avoids the mismatch of points during deformation. Such a property usually requires sophisticated feature descriptors to be defined in most tracking algorithms. The sequence of contours is finally mapped with the corresponding interaction parameters at the level of the robotic hand by means of a feedforward neural network as described in Section 3.2. The performance of the neural network solution is illustrated in Figs. 3b and 4. The training/learning error for the neural networks corresponding to these two objects is of the order 5×10^{-5}, illustrating the capability of the network to accurately map the interaction parameters to the corresponding deformed contour. The testing error is of the order 4×10^{-3}. To validate the prediction abilities of the network, tests are conducted on two datasets that were not part of the training or the testing sets. In the first example depicted in Fig. 3b, the position of the Finger 2 is moved slightly lower (from $P_{2\ blue}$=8093 to P_2=8000) while the other input parameters are kept unchanged with respect to the blue contour (parameters P_1, P_3, F_1, F_2, F_3 are the same as those in the blue-dot profile extracted from real measurements). The estimated contour marked with black stars passes slightly under the one at $P_{2\ blue}$=8093 marked with blue dots, with the peak slightly lower, as it is expected because the finger was moved lower. In the second example, also shown in Fig. 3b, the force at

the first finger is slightly increased (F_1=198 from F_{1blue}=197), while the one applied at the second finger is slightly decreased (F_2=138 from $F_{2\ blue}$ =152) from the values in the measured blue dot contour. As a result of the increased forced at first finger, the estimated contour denoted by green stars goes slightly deeper than the contours marked in blue and black (as they both have the same position and force applied at this finger). As well, as it is expected, under the slightly reduced force at second finger, the estimated contour goes below (more to the right) both the blue dot and black star contour and gets slightly closer to the red dot one, as it can be observed in the detailed image. Another example is presented for the rectangular sponge in Fig. 4.

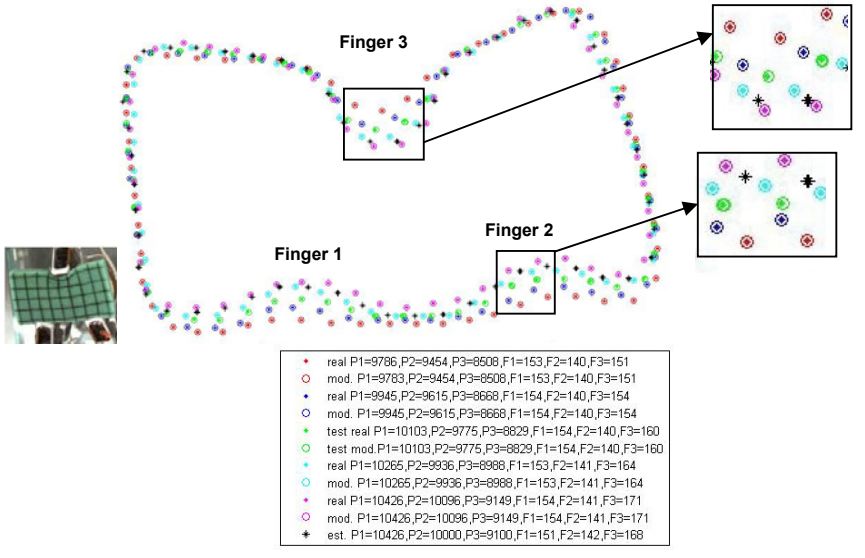

Fig. 4. Real (*dot*), modeled (*circle*) and estimated (*star*) contour points for the green sponge

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. Such a testing scenario where the force value is close to modeled values is chosen as an example to allow the observation and quantification of the results in spite of the complexity of the object behavior. 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. 4 are almost unchanged, an increased force at Finger 3 creates a bending in the object that appears to be due to 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.

5 Conclusion

This paper discusses an original approach to merge visual and force measurements to achieve models of deformable objects relying on experimental manipulation of actual objects. It demonstrates the advantage of using neural networks solutions to map and predict the contour shape of soft deformable objects from real measurements collected by a robotic hand and a camera. The neural network inspired algorithm for segmentation and tracking runs fast, with low errors and guarantees the continuity of points in the tracked contours unlike any classical tracking solutions. The neural network approach 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 previously unseen situations on which the system was not trained. The study will be expanded in future work for different orientations of the robot fingers and for additional parameters in order to achieve a more extensive description of the interaction.

References

1. Petriu, E.M.: Neural Networks for Measurement and Instrumentation in Virtual Environments. In: Ablameyko, S., Goras, L., Gori, M., Piuri, V. (eds.) Neural Networks for Instrumentation, Measurement and Related Industrial Applications. NATO Science Series III: Computer and System Sciences, vol. 185, pp. 273–290. IOS Press, Amsterdam (2003)
2. Cretu, A.-M., Payeur, P., Petriu, E.M., Khalil, F.: Deformable Object Segmentation and Contour Tracking in Image Sequences Using Unsupervised Networks. In: Canadian Conference Computer and Robot Vision, pp. 277–284. IEEE Press, Ottawa (2010)
3. Wang, H., Wang, Y., Esen, H.: Modeling of Deformable Objects in Haptic Rendering System for Virtual Reality. In: IEEE Conference on Mechatronics and Automation, pp. 90–94. IEEE Press, Changchun (2009)
4. Luo, Q., Xiao, J.: Modeling and Rendering Contact Torques and Twisting Effects on Deformable Objects in Haptic Interaction. In: IEEE International Conference on Intelligent Robots and Systems, pp. 2095–2100. IEEE Press, San Diego (2007)
5. Zhong, Y., Shirinzadeh, B., Alici, G., Smith, J.: Cellular Neural Network Based Deformation Simulation with Haptic Force Feedback. In: Workshop Advanced Motion Control, pp. 380–385. IEEE Press, Turkey (2006)
6. Greminger, M., Nelson, B.J.: Modeling Elastic Objects with Neural Networks for Vision-Based Force Measurement. In: International Conference on Intelligent Robots and Systems, pp. 1278–1283. IEEE Press, Las Vegas (2003)
7. Xia, Y., Wang, J., Fok, L.M.: Grasping-Force Optimization for Multifingered Robotic Hands Using Recurrent Neural Network. IEEE Trans. Robotics Automation 26(9), 549–554 (2004)
8. Pedreno-Molina, J., González, A.G., Moran, J.C., Gorce, P.: A Neural Tactile Architecture Applied to Real-time Stiffness Estimation for a Large Scale of Robotic Grasping Systems. Intelligent Robot Systems 49(4), 311–323 (2007)
9. Howard, A.H., Bekey, G.: Intelligent Learning for Deformable Object Manipulation. Autonomous Robots 9(1), 51–58 (2000)
10. Foresti, G.L., Pellegrino, F.A.: Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation. IEEE Trans. Systems, Man Cybernetics 34(3), 325–333 (2004)
11. Chella, A., Dindo, H., Matraxia, F., Pirrone, R.: Real-Time Visual Grasp Synthesis Using Genetic Algorithms and Neural Networks. In: Basili, R., Pazienza, M.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 567–578. Springer, Heidelberg (2007)