Soft Object Deformation Monitoring and Learning for Model-Based Robotic Hand Manipulation

Ana-Maria Cretu, *Member, IEEE*, Pierre Payeur, *Member, IEEE*, and Emil M. Petriu, *Fellow, IEEE*

***Abstract*—This paper discusses the design and implementation of a framework that automatically extracts and monitors the shape deformations of soft objects from a video sequence and maps them with force measurements with the goal of providing the necessary information to the controller of a robotic hand to ensure safe model-based deformable object manipulation. Measurements corresponding to the interaction force at the level of the fingertips and to the position of the fingertips of a three-finger robotic hand are associated with the contours of a deformed object tracked in a series of images using neural-network approaches. The resulting model captures the behavior of the object and is able to predict its behavior for previously unseen interactions without any assumption on the object's material. The availability of such models can contribute to the improvement of a robotic hand controller, therefore allowing more accurate and stable grasp while providing more elaborate manipulation capabilities for deformable objects. Experiments performed for different objects, made of various materials, reveal that the method accurately captures and predicts the object's shape deformation while the object is submitted to external forces applied by the robot fingers. The proposed method is also fast and insensitive to severe contour deformations, as well as to smooth changes in lighting, contrast, and background.**

***Index Terms*—Deformable object, neural networks, object deformation monitoring, object segmentation.**

I. Introduction

STUDIES have consistently shown that object grasping and manipulation are fundamental capabilities of autonomous robot systems. The current generation of assistive robotic arms is equipped with grippers to perform grasping and manipulation tasks. While grippers are very effective for grasping, they are generally unable to manipulate objects [1]. To address this issue, multifinger robotic hands have been developed that can be used, similar to human hands, for exploring, restraining, and/or manipulating objects with the aid of fingers. However, the presence of multiple fingers increases significantly the complexity of the information required in order to achieve safe manipulation of objects. When a task is performed with a robot hand, complete knowledge of the manipulated object, namely, whether the object is rigid or deformable, its location in the environment, and the precise action to be performed, is required. While well-established procedures exist for the manipulation of rigid objects, as well as several 1-D and 2-D solutions for the manipulation of soft objects, very few researchers have developed so far automated solutions for the modeling and manipulation of 3-D deformable objects [2]. Because a large part of the real-world objects are deformable, the automated handling of such soft flexible objects is a challenging research issue with potential use in several applications.

The technique proposed in this paper aims at providing the necessary deformable object model (e.g., shape and elastic behavior) to a robotic hand involved in the dexterous manipulation of the object in order to enable a stable grasp and a precise manipulation capability without *a priori* knowledge on the material of the object. The main goal of this research is therefore to design and implement an automated framework that performs the acquisition of data and the construction of 3-D deformable object models as a result from the interaction with a robotic hand. The work presented is a continuation of the research that the authors pursued on the topic of rigid and deformable object modeling [3]–[6] toward a more complete solution to allow the stable grasp and manipulation of different deformable objects using a seven-degree-of-freedom three-finger Barrett robotic hand. In particular, this paper addresses the modeling aspects that enhance the description of objects and provide accurate estimates of the objects' shape and elastic behavior to the controller of the robotic hand. The proposed solution has significant impact for medical robotics, for remote medical examination of patients, and also for other applications such as interactive virtual environments for training, robotic assembly, and the computer game industry.

This paper is structured as follows. Section II describes the proposed modeling framework, the experimental setup, and the proposed methodology in light of the existing solutions available in the literature. Section III presents the experimental results, their evaluation for different testing scenarios, and comparisons with classical solutions. Section IV derives the conclusion and presents future research directions.

II. Proposed Framework for Object Sensing and Modeling

As the behavior of contact points is the only characteristic that can be directly monitored by the fingers of a robot hand for deformable objects, the measurement of the elastic behavior of the manipulated material cannot rely solely on force and torque sensors. Therefore, incorporating visual data is essential in order to analyze the shape and deformation of the object over its entire surface during the manipulation. The addition

Manuscript received December 10, 2010; revised May 14, 2011 and October 4, 2011; accepted October 30, 2011. Date of publication December 27, 2011; date of current version May 16, 2012. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor W.-Y. Wang.

The authors are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: acretu@site.uottawa.ca; ppayeur@site.uottawa.ca; petriu@site.uottawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2011.2176115

of visual data allows the extraction of relevant information over the entire visible surface of the object and enables the robot to strategically position the hand while adjusting the effort applied on the contact points to ensure the stability and the integrity of the object [2].

To capture the behavior of an object, its interaction with a robot hand is observed by a camera. Neural-network approaches are used both to segment and monitor the deformation of the object in an image sequence and to capture implicitly the complex relationship between the deformation of the object's shape and the interaction parameters measured at the robotic fingers. The deformation of the object can be represented by its contour or by its contour augmented with a grid of lines printed on the object surface to better describe the internal object deformation. For segmentation and deformation monitoring, unsupervised neural networks are used. The motivation for using unsupervised neural architectures is to ensure an automated processing of the image sequences that is fast and insensitive to smooth changes in lighting, contrast, and background and, most importantly, to severe contour deformations. The proposed segmentation procedure also leads to lower errors than a classical segmentation method. The choice of an appropriate unsupervised architecture for deformation monitoring enables the study of the trajectory of every point on the contour of the object, therefore simplifying and unifying the description of the contour shape in all frames. This property usually requires sophisticated feature descriptors with most known tracking algorithms. On the other hand, a supervised architecture is chosen for capturing the relationship between the interaction parameters at the level of the robotic hand and the corresponding contours. This is justified by the possibility to eliminate the requirement to know *a priori* the elastic parameters of the object under study which is provided by a supervised neural network. In this research, most of the objects studied are made of soft highly deformable material whose elastic behavior is impossible to be properly described in terms of simple elastic parameters. The selected architecture also ensures that the application can handle unseen situations, therefore reliably providing critical information for the controller of a robot hand to achieve accurate and stable grasp and more elaborate manipulation capabilities.

A. Experimental Setup

The proposed setup is composed of a Barrett robotic hand and a camera situated perpendicularly to the surface of an object, as shown in Fig. 1. A Point Grey Research Flea 2 industrial camera provides uncompressed video streams via a FireWire connection, with a resolution up to 640×480 pixels. The hand is positioned with the palm up in order to allow the observation of the interaction by the camera without unnecessary occlusions. Two interaction parameters are recorded at each finger. The first one corresponds to the position of each fingertip. It is represented by the number of pulses in the encoder that reads the angle of the motor that drives the finger, and it is equivalent to the Cartesian coordinates of the fingertip, using the Barrett hand kinematic model. This measurement is referred to as "position" measurement to simplify the explanations. The second parameter is a measure of the interaction force applied

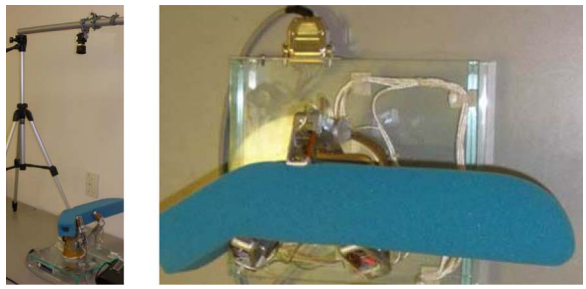

Fig. 1. Experimental setup (lateral and top views).

at each fingertip. It is obtained via strain gauges embedded in each finger. It will be called hereon the "force" measurement as the strain value can be converted into equivalent physical force measurements, in kilograms or newtons, through proper calibration. Additional details on the measurement procedure and data collection are available in [6].

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 forces applied on a set of test objects made from different materials. The force and position measurements are then associated with the tracked contour of the object in the image sequence using a feedforward neural-network architecture.

B. Deformable Object Segmentation and Tracking

Many approaches have been explored for both color image segmentation and object tracking. Among them, neural-network-based solutions have received considerable attention because they are usually "more effective and efficient than traditional ones" [7]. There are various neural-network-inspired solutions proposed for the segmentation of objects in images. One approach is to classify each image pixel into background and object of interest (or foreground) using self-organizing networks [8] or unsupervised competitive networks [9]. Other approaches for object segmentation are based on color information. A hierarchical two-stage self-organizing network is employed in [10] to segment images based on color, where, in the first stage, color image pixel values are grouped in an unsupervised way to identify groups of typical colors and, in the second stage, the dominant color is identified from each group. In [11], the image segmentation procedure is based on a self-organizing map with inputs representing the coordinates of a pixel and its red–green–blue values. Dong and Xie [12] perform color image segmentation based on a combination of unsupervised and supervised techniques. These solutions are limited because the network parameters (e.g., the network size) are determined heuristically, the number of clusters in which the image is divided is predefined, and it is not clear how to select such parameters for a certain image.

After the segmentation of the object of interest, the aim of an object tracker is to generate the trajectory of the object over time by locating its position in every frame of the video [13]. Several neural-network solutions have been proposed for tracking of rigid objects in image sequences. In [14], each rigid object in a scene is assigned to a neuron of a growing competitive network, and neurons are added or deleted when new objects

enter or exit the scene. In [15], a variation of growing neural gas is employed to track simple circular computer-generated rigid objects in video sequences. Rodriguez *et al.* [16] use the topology of a growing neural gas network to determine the posture of a hand in an image sequence and the adaptation dynamics of the same neural network in time to determine the gesture performed by the hand. In [17], contours of hands are described by an active growing neural gas network. An unsupervised solution based on a time-adaptive self-organizing map for tracking of the outer boundary of the lips in an image sequence is proposed in [18]. Apart from the last solution, all the other ones refer to rigid objects and cannot be adapted to deformable objects.

In order to enable the automatic recuperation of elastic parameters and the learning of elastic behavior of deformable objects that would allow for their manipulation using robotic hands, research can greatly benefit from the development of automated segmentation and shape deformation monitoring procedures for such nonrigid objects. It is important to mention that several papers were published in the literature, of which a few examples are [19]–[22], where the term "deformable object tracking" is used to define the tracking of objects that change appearance. These objects can take the shape of pedestrians [19], [20], hockey players [21], faces, and cars [22] that are tracked in video sequences mainly for surveillance applications. In these papers, authors are interested in the motion of the subject or object and less in the way its appearance or in the way its shape changes. This fact is proven by the common use of bounding boxes or circles around the tracked object. In this context, the term "tracking" refers to motion tracking or following of the object. The research topic addressed in this work is a completely different one, in the sense that, while the object moves under the interaction with the robot hand, it is not its motion that is of interest but rather the way that the object deforms under external forces and, therefore, how its shape changes due to the interaction. This is the reason why the term "shape deformation monitoring" is used in the context of this paper instead of "deformable object tracking" to avoid the confusion with the aforementioned literature. While there has been a lot of work on "deformable object tracking" in the sense of motion tracking, there are much fewer papers that address the topic of highly deformable soft object deformation monitoring [4], [23]. In [23], the shape of an object is represented by a small subset of angles between the facets of a mesh, and a set of potential shapes is built on these angles. A dimensionality reduction technique is then employed to produce low-dimensional 3-D deformation models that can capture object models from video sequences. The research work discussed in this paper goes beyond monitoring and capturing the behavior of a soft object by also addressing the learning of soft object behavior from real data and its correlation to the action applied on the object body.

Also, while addressing the problem of automatic deformable object segmentation and monitoring in image sequences, the proposed solution aims further than the classical segmentation/tracking solutions. The neural-network solution presented not only segments and monitors but also models the elastic behavior of an object. Such a description is useful to control a robotic hand for the manipulation of deformable objects. It is important to mention that the experimentation takes place in a relatively controlled environment. The accent is on a fast algorithm that is insensitive to smooth changes in lighting, contrast, and background. The solution does not have to deal with multiple moving objects, as separate object models are built for each object, and neither with severe changes in the environment. Instead, the emphasis is put on accurately tracking and capturing severe object deformations. The proposed approach builds on the solution introduced in [3] and [4]. In the current work, an extensive evaluation with more testing objects and scenarios and a thorough comparison with classical solutions of the segmentation and tracking algorithm are performed. Moreover, additional data are brought in the model to better describe the internal dynamics of deformation that cannot be captured by the contour only. The shape deformation, depicted by both contour and surface grid lines, is finally integrated with the force and position measurements to a more complete description that better supports the stable grasp and manipulation of the modeled objects.

TABLE I
PROPOSED SEGMENTATION AND MONITORING ALGORITHM

Step 1:
- Read initial frame
- Downsample in the window of interest and transform to HSV
- Apply GNG for color clustering:
  - input vector $\mathbf{P} = \{p_1,\ldots,p_n\}$, $p_i = [H_i, S_i, V_i, X_i, Y_i]$, i=1…n
  - neural map $\mathbf{M} = \{m_1,\ldots, m_r\}$, $m_j = [H_j, S_j, V_j, X_j, Y_j]$, with j = 1...r
- Split **M** into object of interest: $C_f = \{m_1,...,m_{rf}\}$, where $m_{rf} \in \mathbf{M}$ with f = 1...rf and background: $C_b = \{m_1,..,m_{rb}\}$, $m_b \in \mathbf{M}$, with b = 1...rb and rb + rf = r
- Identify color of interest; O_c= mean (H_k, S_k,V_k), k = 1…rf.
- Replace it with 1, background with 0 in frame within a tolerance level.
- Apply median filter.
- Identify the object contour using Sobel edge detector.
- Use GNG to find initial contour:
  - input $\mathbf{P_c}=\{x_{c1},\ldots,x_{cn}\}$, with $x_{ci}=[X_i,Y_i]$, i = 1…cn
  - neural map $\mathbf{M_{GNG}}$
- Set current contour = $\mathbf{M_{GNG}}$

Step 2:
- Read next frame
- If motion occurred then
  - Search O_c in the current frame. Replace it with 1 and background with 0.
  - Apply median filter. Identify contour using Sobel edge detector
  - Use NG to predict/model new contour:
    - input vectors $P_{ci} = [X_i, Y_i]$, i = 1…cn
    - initial map = current contour;
    - neural map $\mathbf{M_{NG}}$
  - Set current contour = $\mathbf{M_{NG}}$
  - If last frame then
    - Finish
  - Else
    - Return to Step 2
- Else
  - Keep current contour
  - Return to Step 2

The segmentation and shape deformation monitoring algorithm for the contour, illustrated in Table I, can be summarized as follows. The object of interest is automatically segmented from the initial frame of the sequence of images collected by

the camera. The segmentation is treated as a clustering problem based on color information (hue–saturation–value (HSV) color components) and spatial features (X and Y coordinates of each pixel). The HSV color space is chosen because it represents better the color similarities and is able to more accurately identify pixels on the same surface in spite of some differences between their colors due to nonuniform illumination or shading effects. An unsupervised ontogenic network, namely, growing neural gas [24], is employed to cluster each pixel of the initial frame. At that stage, an unsupervised architecture is preferred to a supervised one because the latter requires a set of training samples, which may not be available, particularly when the image features are unknown or when a certain degree of automation is desired. Growing neural gas is a network that builds by itself by incrementally adding node by node to a neural map, thus eliminating the constraint imposed by the fixed map size of standard unsupervised networks (e.g., self-organizing maps). Moreover, the adaptation parameters of the network are constant; therefore, the need for selecting them heuristically is eliminated. The full mathematical formulation is presented in [24]. In the context of this work, the growing neural gas network receives as input a set of given HSV color-coded pixels, together with their spatial coordinates (x, y) in an image. The input vector is of the form $\mathbf{P} = \{p_1, \ldots, p_n\}$, with p_i being a 5-D vector defined as $[H_i, S_i, V_i, X_i, Y_i]$, with $i = 1, \ldots, n$, where n represents the number of pixels in the image. The spatial coordinates of the pixels in the image are used together with the color information in order to improve the clustering results.

The addition of spatial information does not add to the computational burden, as the computation time is not proportional to the size of the input vectors but to the number of adaptation steps. However, it is worth mentioning that, generally, the clustering results depend more on the first components of the input vector. The addition of too many dimensions in the input vector will not bring improvements in the learning procedure; on the contrary, it might lead to larger errors in the clusters.

The growing neural gas maps the $\mathbf{P}$ input on a smaller array of nodes $\mathbf{M}$ of the same form as $\mathbf{P}$, $\mathbf{M} = \{m_1, \ldots, m_r\}$, with $m_j = [H_j, S_j, V_j, X_j, Y_j]$, with $j = 1, \ldots, r$, with $r < n$. $\mathbf{M}$ is structured in such a way that pixels with similar color and position in $\mathbf{P}$ are projected to nearby nodes in $\mathbf{M}$. Initially, $\mathbf{M}$ is empty. Nodes are added into the network structure at the position where the accumulated error is the highest and when the number of iterations performed is an integer multiple of a predefined value. The growth of the network is terminated when a predefined stopping criterion is met (e.g., a minimum error is reached). The clustering results obtained by the growing neural gas are then classified as one of two categories: object of interest or background. $\mathbf{M}$ is therefore split into two clusters, namely, foreground (object of interest) denoted as $C_f = \{m_1, \ldots, m_{rf}\}$, where $m_f \in \mathbf{M}$ with $f = 1, \ldots, rf$, and background denoted as $C_b = \{m_1, \ldots, m_{rb}\}$, where $m_b \in \mathbf{M}$ with $b = 1, \ldots, rb$, and $rb + rf = r$. The distinction is made based on the mean HSV value computed for the two clusters and based on knowledge that, generally, the background is darker in color than the object of interest. The latter assumption is generally satisfied due to the controlled environment in which the experiments are performed. If, however, this assumption does not hold, a swap of the two clusters can be performed to ensure that the algorithm maps the correct part of the image as foreground. It is worth mentioning that the color segmentation algorithm also works correctly to extract objects that do not have a single color, provided that the background color is different enough from the color of the object and that the colors of the object are somewhat similar (within the threshold). If the colors of an object are significantly different, the color segmentation algorithm can be readily expanded to deal with this situation by performing a multiple-class clustering (e.g., the number of clusters equal to the number of colors) on the results obtained by the growing neural gas instead of the proposed binary foreground–background classification.

In order to identify the color of interest, which is the color of the object to be segmented, the mean is computed for all HSV values in the foreground cluster C_f, $O_c = \mathrm{mean}(H_k, S_k, V_k)$, with $k = 1, \ldots, rf$. The procedure is applied only on the first frame in the sequence. The identified HSV color code O_c is then searched in the initial image and over all images in the sequence where movement occurs, and all pixels with this color code or a very similar code are replaced with one and the rest are replaced with zero in order to segment the object of interest in subsequent frames. The tolerance level allows for the fine tuning of areas on the object surface with different lighting conditions and around the object edges that are usually perceived darker because of the shadow effect. A median filter is finally applied on the result to reduce isolated patches of color. The contour of the object is identified based on the filtered image with the aid of the Sobel edge detector. In the case of objects made of different colors, aside from the search of each color of interest, an additional procedure is required to merge the areas of the object prior to the application of the Sobel edge detector.

A second growing neural gas is employed here to represent the position of each point over the contour $\mathbf{P_c} = \{p_{c1}, \ldots, p_{cn}\}$, with $p_{ci} = [X_i, Y_i]$, with $i = 1, \ldots, cn$. The main purpose of this second network is to detect a reduced number of points (stored in $\mathbf{M_{GNG}}$) that accurately represent the geometry of the contour. The adequate number of points is achieved due to the inherent property of neural gas [25] and growing neural gas networks [24] to find compact data representations based on feature vectors while preserving the topology of the input space. While one might argue that a standard uniform sampling could be used instead, a uniform distribution of points would not allow the monitoring of large local deformations, as not enough points will be allocated locally to accurately capture severe local deformation. By their nature, neural gas and growing neural gas networks preserve the topology of the input space that they are capturing, therefore coping with severe local deformations. For the interested reader, a visual comparison of the differences between uniform sampling, neural gas, and growing neural gas distributions for point clouds is available in [26].

The compact growing neural gas description of the contour is then used as an initial configuration for a sequence of neural gas networks [25] that monitor the contour deformation in the image sequence. Each neural gas network monitors the contour of an object by predicting and readjusting the position of the neurons to follow the contour. Similar to the second growing neural gas described earlier, it receives as input vectors p_{ci} of

the form $[X_i, Y_i]$, with $i = 1, \ldots, cn$. It starts with an initial configuration of nodes, which are, in this case, set to the initial contour points obtained using the growing neural gas (from $\mathbf{M_{GNG}}$). These nodes take the shape of the contour by moving over the data space during adaptation. The main difference between growing neural gas and neural gas is the fact that the number of nodes in the output map $\mathbf{M_{NG}}$ of the neural gas is predefined. In this case, it is set equal to the number of points in $\mathbf{M_{GNG}}$. As the optimum number of points to represent the contour is known from the initial frame analyzed with the growing neural gas, the proposed algorithm uses this number of nodes to track the contour over the image sequence. A fixed-size network with a predefined number of nodes ensures a unified representation of the contour in all frames and makes easier the study of the trajectory of specific points in the contour as each point can be iteratively referred to its previous position. Knowledge of these trajectories enables the description of the deformation behavior under the forces exercised with the robotic fingers. This representation provides, as it will be shown, a solid basis for the anticipation (prediction) of the deformation of the object under manipulation. As well, the capability of the network to start with a set of predefined positions allows the tracking algorithm to always continue from the shape of the contour in the previous frame, therefore increasing the speed of the training procedure given that the movement is, in most cases, smooth from one frame to another.

The training of a neural gas network takes place only for those frames in which a motion occurs with respect to the previous frame. At the time a new frame is presented to the algorithm, an intensity difference is computed between the gray-scale representation of the new frame and the previous one with the purpose of identifying if motion occurred in the scene. This motion can be either the result of the manipulation of the object, which leads to its deformation, or the result of some external factors, such as a change in the background. By computing intensity differences between frames, the proposed solution can handle smooth changes in the contrast and lighting conditions. Moreover, due to the fact that the algorithm searches for the color of interest in the frame each time a motion occurs, the proposed solution can also handle a cluttered and/or moving background as long as it does not interfere with the mean HSV color of the object.

Each frame where motion occurs is transformed to HSV color coding, and the object of interest and its contour are found based on the mean HSV color as identified from the first frame. A 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 be used to initialize the next neural gas network in the sequence when a new frame in which motion occurs is presented again to the algorithm. The procedure is repeated until the last frame of the sequence, as shown in Table I.

To further improve the description of the deformable object shape, a grid of black straight lines can be printed on the object surface to give a better image of the dynamics of deformation without being restricted to the external contour of the object. The points in the deformed grid are recuperated from each image sequence where movement occurs at the same time as the contour. Since the grid is known to be black within a tolerance level, it can be identified following the proposed color segmentation technique. In this case, the second growing neural gas employed to represent the shape of the deformed object will map not only the contour points $\mathbf{P_c} = \{p_{c1}, \ldots, p_{cn}\}$, with $p_{ci} = [X_i, Y_i]$, with $i = 1, \ldots, cn$, but also the grid line points $\mathbf{P_g} = \{p_{g1}, \ldots, p_{gn}\}$, with $p_{gi} = [X_i, Y_i]$, with $i = 1, \ldots, gn$. As a consequence, the sequence of neural gas networks will monitor the evolution of the contour together with the grid points $\mathbf{P_c} \cup \mathbf{P_g}$. This sequence of neural gas networks will be further on associated to the interaction parameters at the level of the hand where the applied forces determined the corresponding deformed contours.

In several manipulation tasks, the proposed algorithm will allow the manipulation of a 3-D object using only the available 2-D data from the sequence of images. If a more complete description of the object's deformation is desired, video data can be collected for different positions of the deformable objects from different sides. In this case, the proposed framework is repeated for each point of view and integrated with the interaction parameters. While being beyond the purpose of this paper, if a complete 3-D object model is desired for complicated manipulation tasks, the proposed method can be employed without changes to monitor 3-D deformations of the object surface as both neural gas and growing neural gas are multidimensional networks. Instead of the grid of black lines imprinted on the object surface, a calibrated stereoscopic camera system can be employed for live acquisition of 3-D data on the surface of the object and for real-time estimation of the surface shape of the manipulated deformable object. Such a framework is proposed in [6]. In this case, the second growing neural gas employed to represent the shape of the deformed object will map 3-D points $\mathbf{P_d} = \{p_{d1}, \ldots, p_{dn}\}$, with $p_{di} = [X_i, Y_i, Z_i]$, with $i = 1, \ldots, m$, with m being the number of reconstructed 3-D points on the object surface that will be monitored using the sequence of neural gas networks and further associated with the interaction parameters.

C. Learning the Mapping of Object Deformation to Force and Position Measurements

The topic of robotic grasping and manipulation of objects has been widely discussed in the literature. Interested readers can find a complete literature review on the topic in [2]. Among the many solutions offered, neural networks have received considerable interest due to their capability to learn the complex functions that characterize the grasping and manipulation operations [27]–[33] and/or to achieve real-time interaction after training [32]. Pedreño-Molina *et al.* [27] integrate three neural models to control the movement of an anthropomorphic finger in a two-finger robotic manipulator with no dependence on the nonlinear stiffness characteristic of objects, based on tactile information from force sensors. A neural network is used by Xia *et al.* [28] to approximate the dynamic system (the set of contact forces such that the object is held at the desired position and external forces are compensated) that describes the grasping force-optimization problem of multifinger robotic hands. To solve the same problem, Fok and Wang [29] use a

primal-dual neural network based on a quadratic formulation with linearized friction constraint. Valente *et al.* [30] propose a neural-network-based system to determine the three contact positions out of a set of fixed points defining a 2-D object contour. Howard and Bekey [31] 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 manipulating it. Chella *et al.* [32] use a neurogenetic approach to the problem of three-finger grasp synthesis of planar objects fitted with superellipses. A hierarchical self-organizing neural network to select proper grasping points in 2-D is proposed by Foresti and Pellegrino [33].

In the context of this paper, neural architectures are chosen for reasons similar to those mentioned earlier, namely, their capability to store (offline, in a compact manner) and predict (online, in real time) the complex relationship between the deformation of the object and the corresponding interaction parameters at each finger. Unlike the other neural-network solutions, the proposed approach neither makes any assumption on the material of the object by imposing a certain representation [31], [32] nor imposes certain dynamic models at the points of contact [28], [29]. A feedforward neural network is employed to map each contour in the sequence, as obtained in Section II-B, with the interaction parameters that are the corresponding position of the fingers of the robotic hand and the corresponding force measurement at each fingertip, as defined in Section II-A. After training, the network is capable of predicting the behavior of an object under manipulation in real time. By storing such networks, a database of object models is built. Such models, when used appropriately in the control loop of the robotic hand, can support more accurate and stable grasp and more elaborate manipulation capabilities for deformable objects. The network defined to map contours with interaction parameters on a given object has six input neurons associated with the interaction parameters, namely, the positions of the three fingers (P_1, P_2, P_3) and the force measurements at each fingertip (F_1, F_2, F_3). A number of 30 to 50 hidden neurons are generally used. This number is adjusted such that it ensures a good compromise between the training time and the accuracy of models. The output vector is the set of coordinates defining the points on the contour. Its size is double the number of points in the contour (the number of points in $\mathbf{M_{GNG}}$), as it contains the X and Y coordinates for each point in the contour. The only preprocessing applied on the input is a normalization to the [0 1] interval prior to training. In our experimentation, three-quarters of the data available are used for training, and a quarter are used for testing. The network for each object under study is trained for 120 000 epochs using the batch version of the scaled conjugate gradient backpropagation algorithm [33] with the learning rate set to 0.1. Once trained, the network takes as inputs the interaction parameters $(P_1, P_2, P_3, F_1, F_2, F_3)$ and outputs the corresponding object contour.

Similarly, the contour and the grid points can be mapped to the corresponding set of interaction parameters with a feedforward neural-network architecture identical to the one described earlier, i.e., with the same number of inputs, outputs, training parameters, and learning algorithm. The only difference is that the output vector will contain not only the X and Y coordinates for each point over the contour but also the X and Y coordinates for each point recuperated from the grid printed on the object. This model provides a comprehensive evaluation of the object deformation over its entire surface, i.e., not only around its contour where robot fingers are applying forces.

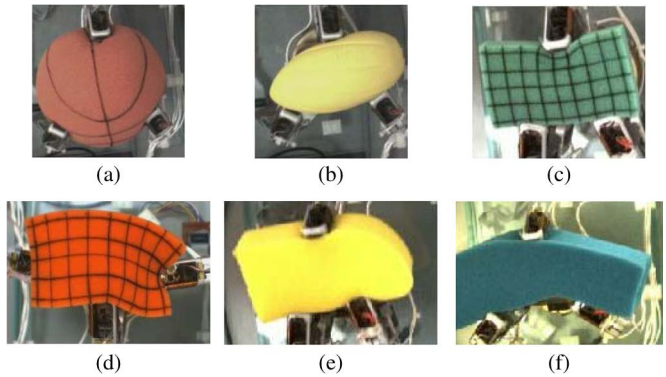

Fig. 2. Test objects: (a) Round foam ball, (b) oval rubber ball, (c) rectangular green sponge, (d) rectangular orange sponge, (e) yellow sponge, and (f) long blue sponge.

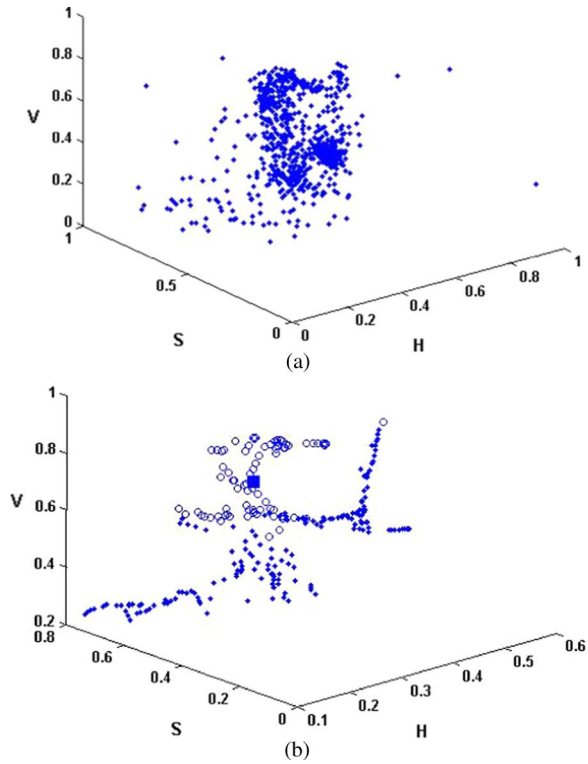

Fig. 3. (a) Distribution of HSV color values in the HSV color space for the long blue sponge and (b) the corresponding growing neural gas map, with the identified background shown in dots, the object marked by circles, and the mean HSV color of the object marked by a large square.

III. Experimental Results

In order to validate the proposed modeling framework, several tests were conducted on a set of deformable objects with different shapes, colors, and elasticity characteristics, of which a limited subset is shown in Fig. 2. The deformations of each object as a result of the interaction with the robotic hand's fingers are recorded in a series of image sequences collected with the vision system located perpendicularly over the object.

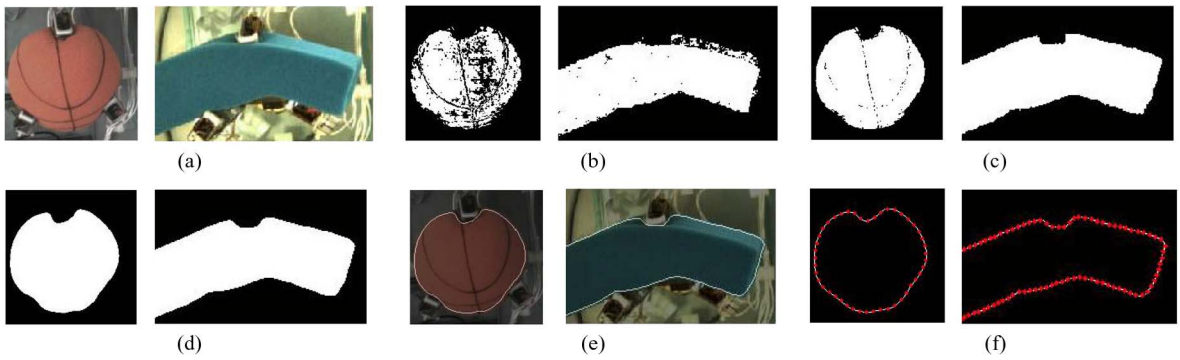

Fig. 4. (a) Downsampled images of the round foam ball and the long blue sponge. Results obtained (b) when replacing the color of interest with one and the background with zero, (c) when tolerance levels of 0.09 for H and S and 0.02 for V are accepted, and (d) after median filtering. (e) Contour detection with the Sobel edge detector on the filtered image (with the contour displayed over the initial frame). (f) Growing neural gas model of the contour.

Each object is tracked in the image sequence using the algorithm in Section II-B, and then, the contours, along with the grid line markers if available on the object, are associated with the position and force data, as defined in Section II-C.

A. Segmentation Results and Evaluation

The first steps for the proposed segmentation applied on the initial frame in the case of the long blue sponge from Fig. 2(f) are shown in Fig. 3. Fig. 3(a) shows the initial distribution of HSV color values in the HSV color space, while Fig. 3(b) shows the corresponding map $\mathbf{M}$ obtained when the color-coded HSV values [in Fig. 3(a)] and the spatial coordinates of the corresponding pixels are presented to a growing neural gas network. In order to segment the object in a selected image, $\mathbf{M}$ is split into two clusters.

Fig. 3(b) shows the foreground (denoted by circles) and background (denoted by dots) in the growing neural gas map $\mathbf{M}$. The color of interest, which is the color of the object to be segmented, is selected to be the mean for all HSV values in the foreground cluster. The mean HSV value (denoted as O_c in Table I) is marked with a larger square in Fig. 3(b).

To visually evaluate the results, Fig. 4(a) shows the images downsampled by 50% both for the round foam ball in Fig. 2(a), whose resolution goes from 160 × 160 pixels to 80 × 80 pixels, and for the long blue sponge in Fig. 2(f), whose resolution goes from 260 × 150 pixels to 130 × 75 pixels. Pixel areas considered are limited to the region of interest in the original images. The downsampling is performed to remove some unnecessary information in the background and therefore reduce the time required for the background/foreground classification.

Fig. 4(b) shows the results obtained by replacing the color of interest with one (white) and the rest of the image with zero (black) for the two objects. It can be observed that the results are rough at this stage. Several parts of the objects that are slightly darker due to nonuniform lighting conditions and shadow effects are erroneously classified as background. The inclusion of a tolerance brings an improvement in the results obtained, as shown in Fig. 4(c). In general, tolerance levels of less than 0.09 for H and S and less than 0.02 for V are used. These values can be adjusted according to each object. The role of this tolerance level is to compensate for slight shading and color variations resulting from the lighting conditions and/or shadow effects. A median filter is applied next to reduce isolated patches and smooth the edges. The filtered images are shown in Fig. 4(d).

TABLE II
COMPARISON WITH CLASSICAL COLOR-BASED SEGMENTATION

Object	Average Error	
	errHSV-NN	errHSV
Round ball	0.02	0.06
Oval ball	0.01	0.02
Rectangular green sponge	0.02	0.1
Rectangular orange sponge	0.07	0.2
Yellow sponge	0.01	0.3
Long blue sponge	0.01	0.025

In order to show the benefit of using an unsupervised neural network to segment an image, the proposed method is compared at this stage for all the objects under study with a standard color-based segmentation based on HSV color codes only. The latter is based on mean HSV values computed in a user-selected frame that samples the object color. To ensure a fair basis of comparison, the ground truth is obtained by a manual segmentation of the object of interest in a set of test frames. The same tolerance level and median filtering are applied on the ground-truth image, on the results obtained using the standard color-based segmentation, and on the ones obtained using the proposed solution. The average error rate is then computed for both the proposed color-based segmentation and the classical solution using the formula proposed in [34], against the ground-truth segmented image

$$E = \frac{\sum_{i=1}^{f} N_{ow}^{i} + N_{bw}^{i}}{N * f} \tag{1}$$

where N_{ow}^{i} is the number of pixels that belong to the foreground in the ground-truth image but are wrongly classified as background in the ith frame, N_{bw}^{i} is the number of pixels that belong to the background in the ground-truth image but are wrongly classified as foreground in the ith frame, N is the total number of pixels in a frame, and f is the number of frames in the image sequence. The results are presented comparatively in Table II.

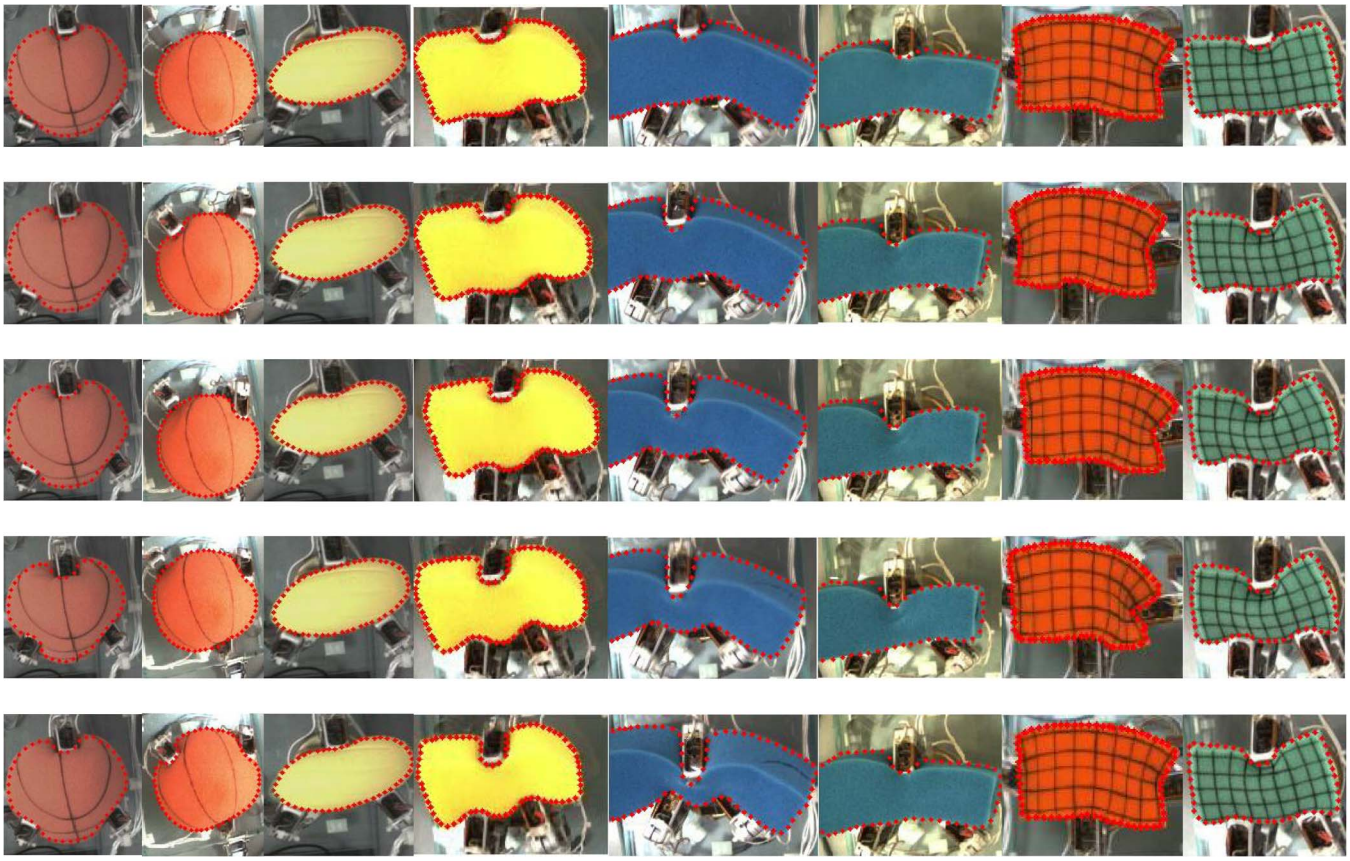

Fig. 5. Examples of contour tracking for the objects under study.

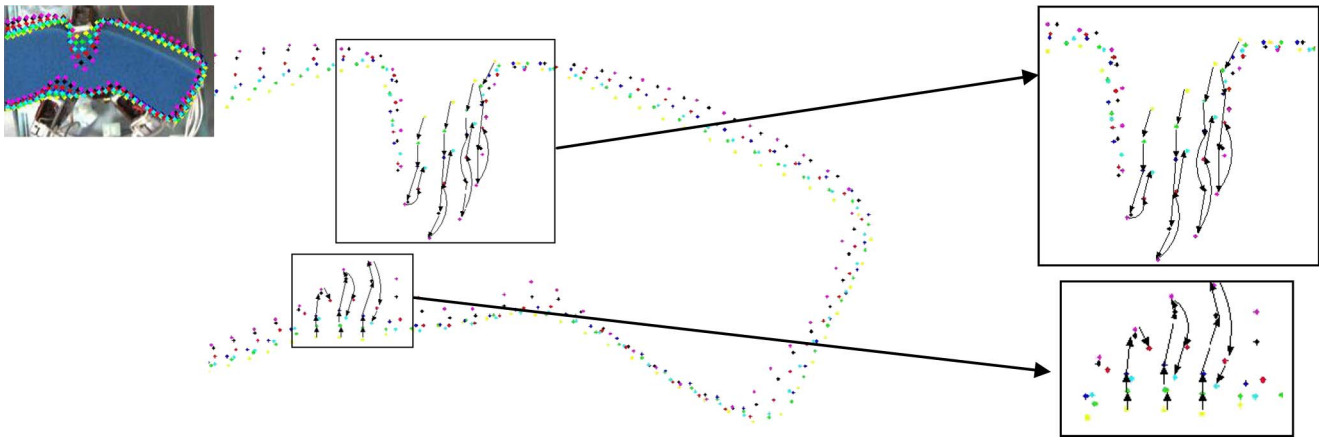

Fig. 6. Trajectory of nodes between successive configurations for the long blue sponge and details of the enlarged trajectory.

It can be observed that, apart from the fact that the proposed solution does not require the user to select an area of interest that samples the object color, it also reaches slightly lower errors $(errHSV\text{-}NN)$ than the standard color-based segmentation $(errHSV)$. The image obtained after median filtering [Fig. 4(d)] is then used to detect the contour of the object of interest, shown in Fig. 4(e), using the Sobel edge detector. The growing neural gas model trained to capture the initial contour, denoted as $\mathbf{M_{GNG}}$ in Table I, is shown with red dots in Fig. 4(f).

B. Deformation Monitoring Results and Evaluation

Starting from the growing neural gas model, a series of neural gas networks is employed to monitor the deformation of the contour of the object as it deforms due to the interactions with the robotic hand. Each neural gas network starts with the configuration of nodes from the previous frame in order to minimize the movement of nodes during adaptation.

Figs. 5 and 6 show various frames extracted at different times throughout the videos and the corresponding configuration of points in $\mathbf{M_{NG}}$ (shown with red dots) for each of the test

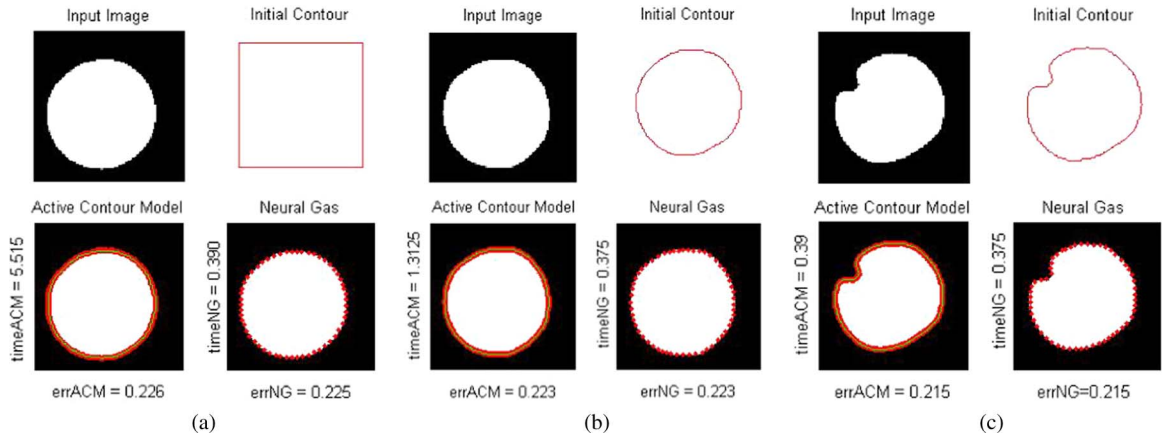

Fig. 7. Comparison of series of neural gas networks with active contour models for the round foam ball.

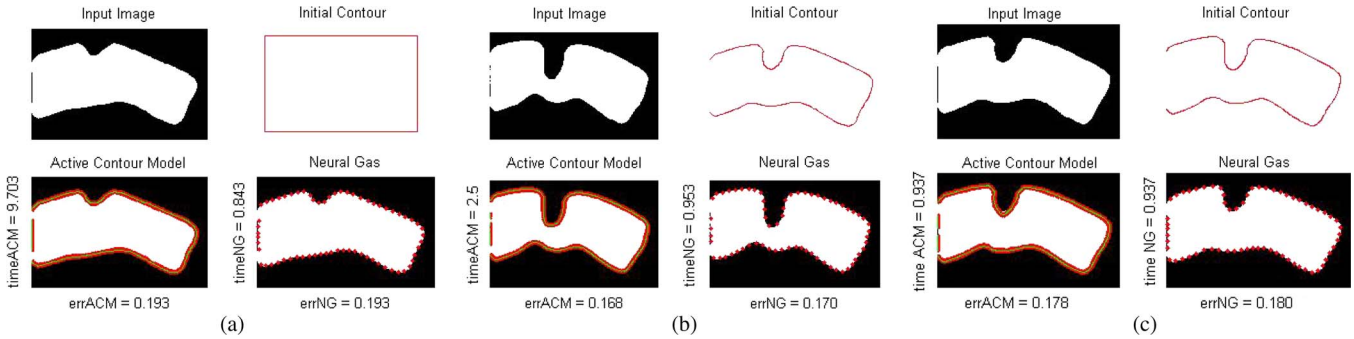

Fig. 8. Comparison of series of neural gas networks with active contour models for the long blue sponge.

objects. It can be seen that the monitor follows accurately the contour of each deformed object, a fact that is also demonstrated by the low error achieved during tracking, as will be illustrated in the next section. The results are robust against smooth changes in lighting and background as it can be observed when comparing the images for the orange ball between columns 1 and 2 in Fig. 5 and for the blue sponge between columns 5 and 6 in the same figure for different lighting conditions. Fig. 6 shows parts of the complex trajectory that the points in the contour of the long sponge follow during manipulation. The trajectory is marked with arrows for only a few nodes in order not to overload the figure. It can be observed that, due to the choice of a fixed number of nodes used in the neural gas network and to the proposed learning mechanism, the nodes in the contour retain their correspondence with specific points 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 frames. The same property ensures also the monitoring of the lateral motion (along the sides of the object) that occurs beyond the object's compression, as it is illustrated in the enlarged trajectories on the right side of Fig. 6. Finally, the fact that each tracked point is uniquely paired with a single well-defined correspondent in a successive contour and further learned and predicted using a feedforward neural network in Section III-E eliminates the need for implementing specific procedures for data association. The elimination of the need for techniques such as the nearest neighbor principle or Kalman filters to predict the trajectory is another great advantage of the proposed solution.

By imposing continuity in the displacement of shape points, the proposed solution also brings an additional geometrical constraint over the object deformation. Given that contour tracking is dedicated here to deformable objects, one cannot rely on other information, such as shape constraints, for example, that could be otherwise used as a basis for tracking. Finally, the proposed solution for shape deformation monitoring provides the basis for the estimation of the elastic behavior of an object based on its contour deformation when submitted to external forces at certain positions, which distinguishes it from classical tracking algorithms.

C. Comparison of Tracking With Active Contour Models

In order to further evaluate the proposed method for monitoring the shape contour deformation, it is compared with an active contour tracking method based on an improved version of classical active contour models, namely, the active contours without edges proposed by Chan and Vese [36].

The reason for choosing an improved active contour model for comparison is twofold. First, in spite of the fact that active contour models, also known as snakes, have been proposed over a decade ago and initially not specifically proposed for the tracking of deformable objects, they are considered to be one of the standards for tracking deformable objects, as revealed by the literature on the topic over the years up to now [20], [37]–[41]. The second reason is that active contour models are the closest in concept to the research proposed in this paper. Similar to the proposed solution, active contour models are proposed for a single image and applied sequentially for all

the images in the sequences. The main idea for active contour models is to evolve a curve, subject to constraints from a given image, in order to detect objects in that image [36], similar to the sequence of neural gas networks proposed in Section II-B whose nodes evolve in time to track the object's contour in the image sequence. Unlike neural gas networks, active contour models require an initialization curve around the object to be detected. This curve moves for a predefined number of steps toward its interior normal and stops on the boundary of the object. Since, in the context of the current application, the accent is on developing automated solutions, the initial contour for the active contour model in the initial frame of the sequence is automatically set to the same size as the input image (minus a 5-pixel frame around), as shown in Figs. 7(a) and 8(a) for the ball and the long sponge, respectively.

The active contour modeling, particularly the improved active contour model without edges proposed in [36] that incorporates techniques of curve evolution, Mumford–Shah functionals for segmentation, and level sets, is applied on the presegmented image, at the same stage as the sequence of neural gas networks in the proposed solution. There are two reasons for this. First, in this way, a common basis for comparison is ensured. The second reason is the incapability of the active contour model to capture the object alone in the color image. The segmentation obtained with the active contour model is correct, but it also contains the cables and the robotic fingers present around the ball. The algorithms are tested on the Matlab platform running on a Pentium 1.3-GHz machine with 512-MB memory. Figs. 7(a) and 8(a) show that, for the two examples presented, the time required by the active contour model to capture the shape of the object in the initial frame of the sequence (denoted as $timeACM$) is significantly larger than the one required by the neural gas (denoted as $timeNG$). This very large difference characterizes only the first frame. At the same time, the errors achieved by the two methods on this frame, $errACM$ and $errNG$, respectively, are very similar. The error is computed as the Hausdorff distance between each of the contours obtained by the two methods (C_{NG} and C_{ACM}, respectively) and the contour of the object obtained by manual segmentation of the object in the input image (C_m)

$$H(C_m, C_{\mathrm{NG/ACM}}) = \max\left(h(C_m, C_{\mathrm{NG/ACM}}), h(C_{\mathrm{NG/ACM}}, C_m)\right) \quad (2)$$

where

$$h(C_m, C_{\mathrm{NG/ACM}}) = \max_{a \in C_m} \min_{b \in C_{\mathrm{NG/ACM}}} \|a - b\| \quad (3)$$

and $\|\cdot\|$ denotes the Euclidean distance. In a similar manner to the proposed solution in Section II-B, each time a new frame is presented, the contour obtained in the previous step is used as the current initialization contour for the new active contour model, as shown in Figs. 7(b) and (c) and 8(b) and (c), respectively. This initialization reduces the time required by the active contour model for the subsequent frames in the sequence. Nevertheless, the computing time for the active contour model is generally higher than the one required by the neural gas, as it can be observed in Figs. 7(b) and 8(b), Figs. 7(c) and 8(c), and Table III that displays the average computing time and the average error for all the objects under study. The average time and error include the initial frame, but as the average length of videos is 900–1500 frames, the weight of this first frame is low. The overall error achieved by our method is similar, as it can be observed in Figs. 7(b) and (c) and 8(b) and (c) and Table III. The error is higher for both methods for the round ball when compared to the other objects due to the severe deformations and also due to the fact that the ball rolls in the hand during probing. The probing for this object occurs by repetitively contracting and relaxing the grip of the hand, but the movement of the three fingers is not synchronized, while for all the other objects, the contracting and relaxing occur simultaneously at the three fingers.

TABLE III
AVERAGE TIME (PER FRAME IN SECONDS) AND AVERAGE ERROR FOR MONITORING

Object	Average time		Average error	
	timeNG	timeACM	errNG	errACM
Round ball	0.444	1.06	0.26	0.271
Oval ball	0.482	1.254	0.15	0.154
Rectangular green sponge	0.306	0.691	0.18	0.188
Rectangular orange sponge	0.455	0.88	0.19	0.195
Yellow sponge	0.371	0.723	0.19	0.19
Long blue sponge	0.902	2.037	0.18	0.18

In general, the time is higher for the active contour model in those situations where the object moves under the interaction with the fingers during probing (e.g., the round ball in Fig. 7(b) rolls over the palm of the robotic hand) and when the object undergoes large deformations with respect to the previous frame [e.g., the long sponge in Fig. 8(b)]. The video sequences are made up of a large number of severe deformations, and this justifies the significantly longer time for the active contour model. When only small and smooth deformations occur from one frame to the other as those shown in Figs. 7(c) and 8(c), the times are almost equal for both methods. These observations lead to the conclusion that the proposed solution for deformation monitoring is faster than active contour models for severe deformations. It also offers a higher degree of automation due to the insensitivity to initial conditions, unlike the initialization curve that is required for active contour models, and to the parameter setting, unlike the preset number of iteration steps for active contour models.

D. Grid and Contour Tracking Results

To further improve the description of the deformable object shape over its entire surface, a grid of black straight lines is printed on some objects' surface, as shown in Fig. 2(c) and (d). Given that the grid has a defined color (within a tolerance level), it can be recuperated from the initial image in the sequence following the color segmentation algorithm detailed in Section II-B. Fig. 9(b) shows the grid points and the contour that are simultaneously recuperated for the orange

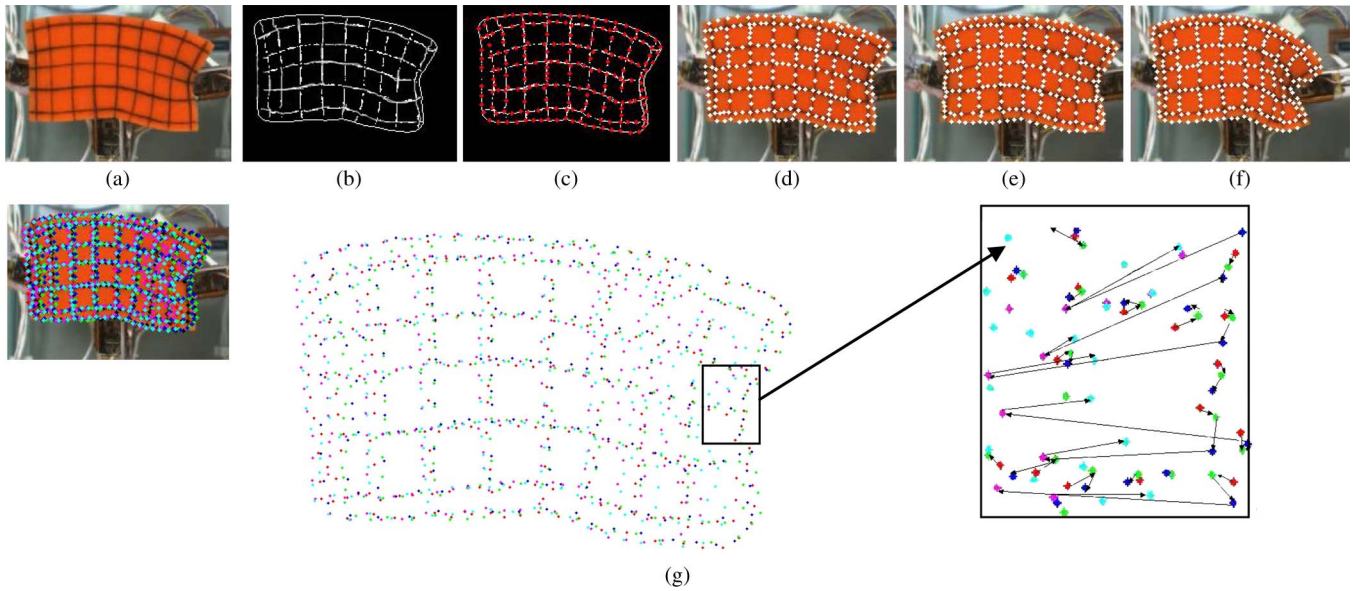

Fig. 9. Contour and grid point tracking for the orange sponge: (a) Initial frame, (b) grid and contour points, (c) initial growing neural gas, (d)–(f) tracking using a sequence of neural gas networks, and (g) trajectory of tracked points and detail of the one-to-one correspondence of tracked points.

Fig. 10. Contour and grid point tracking for the rectangular green sponge.

rectangular sponge shown in Fig. 9(a). It can be observed that the grid points are fairly well but not fully recuperated due to the change in the color of the grid that occurs during the interaction with the object. As the object's deformation is stronger, the number of lost points in the grid is larger, but nevertheless, the presence of grid points helps to better understand the internal deformation of the object that cannot be captured by tracking the contour only. Starting from a growing neural gas model that maps the contour points along with the grid points as shown in Fig. 9(c), a series of neural gas networks tracks the shape deformation that results on the object due to the interaction with the robotic hand. Several frames illustrating these neural gas networks (the points in $\mathbf{M_{NG}}$ are shown with white dots) are shown in Fig. 9(d)–(f). In this case, as for contour tracking, there is a one-to-one correspondence between the points, as shown in Fig. 9(g), in spite of a larger number of points tracked.

To demonstrate the robustness of the approach, an additional example of tracking of the contour together with the grid lines is shown in Fig. 10, where stronger deformations occur on the narrow side of the object. It can be observed by comparing these frames with the last two columns in Fig. 5 that the presence of grid points aside from the contour gives a better image of the dynamics of shape deformation. The contour, as obtained in Section III-B, or the contour together with the grid points, as obtained earlier, is further associated to force and position measurements to capture the object model in terms of its deformation behavior.

E. Evaluation of Learning and Prediction of Deformation

A feedforward neural network is used to store implicitly the behavior of an object under manipulation by mapping contours to the finger positions and force measurements at fingertips. Finally, an estimated deformed contour for a set $P_{i1}, P_{i2}, P_{i3}, F_{i1}, F_{i2}, F_{i3}$ of interaction parameters that were not part of neither the training set nor the testing set is presented in order to test the prediction capability of the network. The latter are marked with black stars and denoted as "est." in the legend. In this case, the network has 30 hidden nodes and 110 units at the output and is trained for 200 000 epochs. The learning error is, on average, of order 5×10^{-5}. The error on the testing set is slightly higher. This is shown by the dots representing real data being slightly off centered in the circle representing modeled data in the right side of Fig. 11. However, overall, the error remains low (on average, of order 3×10^{-3}).

To show the prediction capability in the testing scenario, the finger positions are kept almost at the same position, but the force at the first finger $(F_1 = 220)$ is increased from the value in the magenta dot contour $(F_{1\_\text{magenta}} = 204)$, while it is kept below the value in the cyan curve $(F_{1\_\text{cyan}} = 228)$. 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. Similar results were obtained for different testing scenarios and for objects made of different materials. The network was able to provide

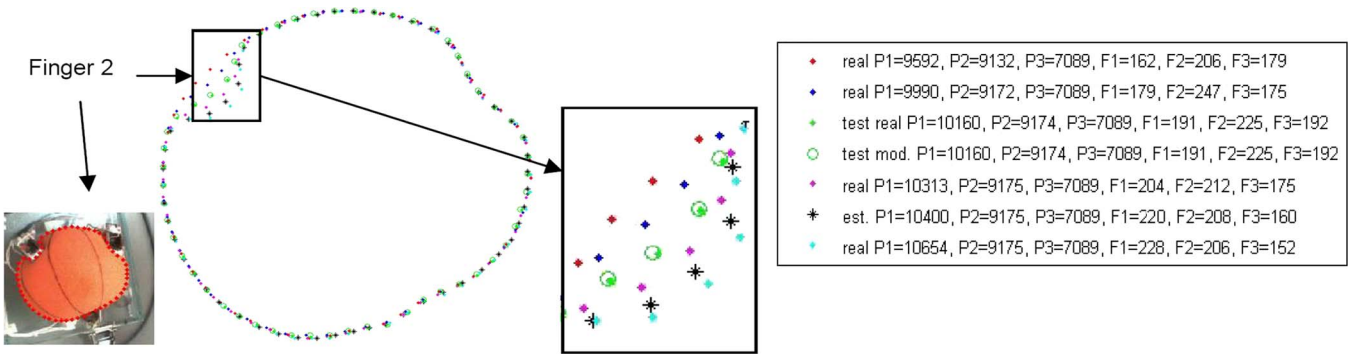

Fig. 11. Real, modeled, and estimated contour points for the ball.

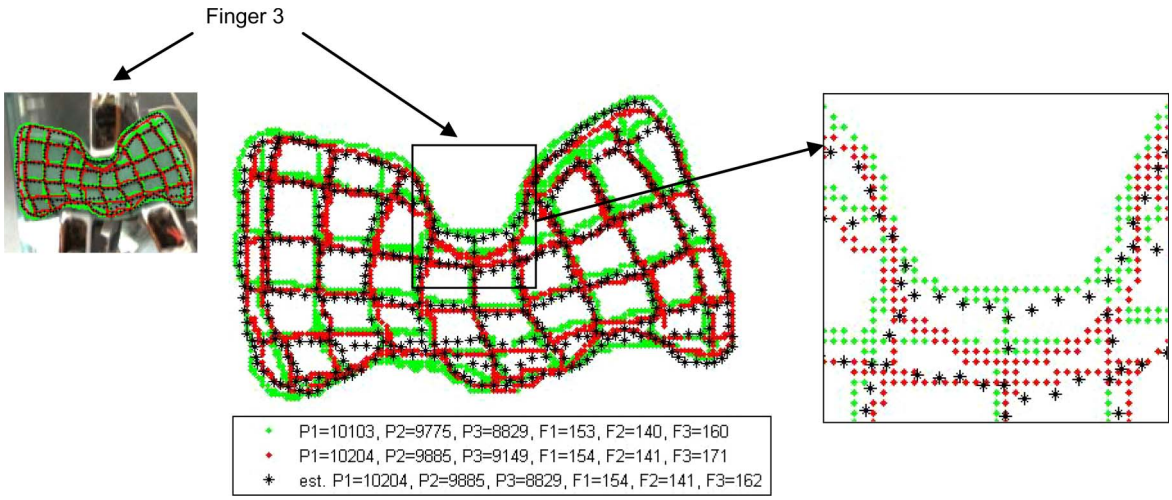

Fig. 12. (Dots) Real contour and grid data recuperated from the image sequence and (stars) estimated shape deformation when compressing the green sponge.

accurate estimates for data that were not part of the training or testing sets.

Similarly, a feedforward neural network is used to store implicitly the shape deformation of an object under manipulation by mapping contours together with the grid points to the finger positions and force measurements at fingertips. This case is exemplified for the green sponge in Fig. 2(c) and detailed in Fig. 12. To test the prediction capability, the network is tested for a simulated force at finger 3 ($F_3 = 162$) that is higher than the value of the measured force ($F_{3\_\mathrm{green}} = 160$) applied when the object takes the shape shown with green dots but lower than the measured one applied when the object deformation is illustrated with red dots ($F_{3\_\mathrm{red}} = 171$). The positions of the fingers are kept almost unchanged, and so are the forces applied at the other fingers, as shown in the legend. The estimation is shown in black stars, while depicted in green and red are the contours and grid points recuperated from the image sequence. It can be seen that the estimated shape, while not perfect, is close to what one would expect—namely, passing slightly below the green contour and closer to it as the value of the force is closer but above the red one around finger 3. The placement around the other two fingers is correct as well, and therefore, the network is able to provide estimates of the shape deformation of the object under previously unseen parameter configurations.

IV. Conclusion

This paper has demonstrated the benefits of using neural-network approaches for deformable object segmentation and monitoring in image sequences on one side and for modeling and predicting the behavior of soft deformable objects manipulated by a robotic hand on the other side.

The proposed combination of neural networks for segmentation and deformation monitoring runs fast, is robust, and eliminates the constraints of standard unsupervised networks and of active contour models with respect to the choice of parameters and the predetermined size of the network. The segmentation solution obtains lower errors than the classical color-based segmentation solution, and the tracking solution is faster than active contour models. It can be easily adapted to track grid points printed on a deformable object for a better description of the internal deformation of an object. The neural approach used for the modeling and prediction of deformation shapes based on force measurements and the positions of the fingers, when integrated with the control algorithms, can allow for more accurate and stable grasp and for more elaborate manipulation capabilities on deformable objects.

As future work, additional parameters of interaction will be collected at the level of the robot fingers for a more complete description of the interaction. The study will be expanded for different orientations of the robot fingers for a more extensive

description of the interaction and for objects with a broader range of elastic properties. Data collection will be performed for different situations, including those that lead to slippages or drops of the objects. An extensive study of the resulting data will be performed to allow for the characterization of the manipulation parameters for specific objects, materials, and grasping points and to identify configurations that ensure stable grasps and correct manipulation techniques. As well, additional experiments will be performed on higher density of data obtained from the stereoscopic vision system to achieve full 3-D surface shape estimation.

ACKNOWLEDGMENT

The authors would like to thank F. F. Khalil for his help in collecting the data sets used for the experimentation.

REFERENCES

[1] L. Birglen, T. Laliberte, and C. Gosselin, *Grasping vs. manipulating*, vol. 40, *Springer Tracts in Advanced Robotics*. Berlin, Germany: Springer-Verlag, 2008, ch. 2, pp. 7–31.

[2] F. F. Khalil and P. Payeur, "Dexterous robotic manipulation of deformable objects with multi-sensory feedback—A review," in *Robot Manipulators, Trends and Development*, A. Jimenez and B. M. Al Hadithi, Eds. Vukovar, Croatia: In-Tech, 2010, pp. 587–619.

[3] A.-M. Cretu, E. M. Petriu, P. Payeur, and F. Khalil, "Estimation of deformable object properties from visual data and robotic hand interaction measurements for virtualized reality applications," in *Proc. IEEE Int. Symp. Haptic Audio Vis. Environ. Appl.*, Phoenix, AZ, 2010, pp. 168–173.

[4] A.-M. Cretu, E. M. Petriu, P. Payeur, and F. Khalil, "Deformable object segmentation and contour tracking in image sequences using unsupervised networks," in *Proc. IEEE Canadian Conf. Comput. Robot Vis.*, Ottawa, ON, Canada, May 2010, pp. 277–284.

[5] A.-M. Cretu, E. M. Petriu, and P. Payeur, "Data acquisition and modeling of 3D deformable objects using neural networks," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, San Antonio, TX, 2009, pp. 3383–3388.

[6] F. F. Khalil, P. Payeur, and A. M. Cretu, "Integrated multisensory robotic hand system for deformable object manipulation," in *Proc. IASTED Int. Conf. Robot. Appl.*, Cambridge, MA, 2010, pp. 159–166.

[7] N. Yeo, K. Lee, Y. Venkatesh, and S. Ong, "Color image segmentation using the self-organizing map and adaptive resonance theory," *Image and Vision Computing*, vol. 23, pp. 1060–1079, 2005, Elsevier.

[8] L. Maddalena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," in *Adv. Brain, Vis. Artif. Intell.*, vol. 4729, *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2007, ch. 18, pp. 181–190.

[9] R. M. Luque, E. Dominguez, E. J. Palomo, and J. Munoz, "A neural network approach for video object segmentation in traffic surveillance," in *Int. Conf. Image Analysis and Recognition*, vol. 5112, *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2008, ch. 15, pp. 151–158.

[10] S. H. Ong, N. Yeo, K. Lee, Y. Venkatesh, and D. M. Cao, "Segmentation of color images using a two-stage self-organizing network," *Image Vis. Comput.*, vol. 20, no. 4, pp. 279–289, Apr. 2002.

[11] Y. Jiang, K.-J. Chen, and Z. H. Zhou, "SOM based image segmentation," in *Lecture Notes in Artificial Intelligence*, vol. 2639, G. Wang, Y. Yao, and A. Skowron, Eds. New York: Springer-Verlag, 2003, pp. 640–643.

[12] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 925–936, Jul. 2005.

[13] A. Yilmaz, O. Javed, and M. Shah, "Object tracking, a survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 13–45, Dec. 2006.

[14] R. M. Luque, J. O. L. Lobato, E. Rubio, and E. Palomo, "Object tracking in video sequences by unsupervised learning," in *Int. Conf. Comp. Anal. Images & Patterns*, vol. 5702, *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2009, ch. 130, pp. 1070–1077.

[15] G. A. Drumea and H. Frezza-Buet, "Tracking fast changing non-stationary distributions with topologically adaptive neural network," in *Proc. Symp. Artif. Neural Netw.*, Bruges, Belgium, 2007, pp. 43–48.

[16] J. G. Rodriguez, F. Flórez-Revuelta, and J. M. García-Chamizo, "Learning and comparing trajectories with a GNG-based architecture," in *Proc. Symp. Distrib. Comput. Artif. Intell.*, J. M. Corchado, S. Rodriguez, J. Linas, and J. M. Molina, Eds., 2009, pp. 644–652.

[17] A. Angelopoulou, A. Psarrou, G. Gupta, and J. G. Rodriguez, "Robust modeling and tracking of nonrigid objects using active-GNG," in *Proc. IEEE Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–7.

[18] M. K. Moghaddam and R. Safabakhsh, "TASOM-based lip tracking using the color and geometry of the face," in *Proc. Conf. Mach. Learn. Appl.*, 2005, pp. 63–68.

[19] C.-J. Li and S.-J. Wang, "Detection and tracking of a single deformable object on an active surveillance camera," in *Proc. 16th IPPR Conf. Comput. Vis., Graph. Image Process.*, Kinmen, China, 2003, pp. 96–103.

[20] M. S. Allili, "Effective object tracking by matching object and background models using active contours," in *Proc. Int. Conf. Image Process.*, 2009, pp. 873–876.

[21] H. Zhou, Y. Yuan, Y. Zhang, and C. Shi, "Non-rigid object tracking in complex scenes," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 98–102, Jan. 2009.

[22] A. Jacquot, P. Sturm, and O. Ruch, "Adaptive tracking of non-rigid objects based on color histograms and automatic parameter selection," in *Proc. IEEE Workshop Motion Video Comput.*, Jan. 2005, vol. 2, pp. 103–109.

[23] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, "Surface deformation models for non-rigid 3D shape recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1481–1487, Aug. 2007.

[24] B. Fritzke, "Unsupervised ontogenic networks," in *Handbook of Neural Computation*, E. Fiesler, Ed. London, U.K.: Oxford Univ. Press, 1997.

[25] T. M. Martinetz, "Competitive Hebbian learning rule forms perfectly topology preserving maps," in *Proc. Int. Conf. Artif. Neural Netw.*, Amsterdam, The Netherlands, 1993, pp. 427–434.

[26] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Neural gas and growing neural gas networks for selective 3D sensing: A comparative study," *Sens. Transducers J.*, vol. 5, pp. 119–134, Mar. 2009.

[27] 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," *J. Intell. Robot Syst.*, vol. 49, no. 4, pp. 311–323, Aug. 2007.

[28] Y. Xia, J. Wang, and L.-M. Fok, "Grasping-force optimization for multifingered robotic hands using a recurrent neural network," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 549–554, Jun. 2004.

[29] L.-M. Fok and J. Wang, "Two recurrent neural networks for grasping force optimization of multi-fingered robotic hands," in *Proc. IEEE Int. Joint. Conf. Neural Netw.*, 2002, pp. 35–40.

[30] C. M. O. Valente, A. Schammass, A. F. R. Araujo, and G. A. P. Caurin, "The role of the RBF training in a neural model for object grasping," in *Proc. IEEE Conf. Intell. Robots Syst.*, 1999, pp. 430–435.

[31] A. H. Howard and G. Bekey, "Intelligent learning for deformable object manipulation," *Autonomous Robots*, vol. 9, no. 1, pp. 51–58, Aug. 2000.

[32] A. Chella, H. Dindo, F. Matraxia, and R. Pirrone, "Real-time visual grasp synthesis using genetic algorithms and neural networks," in *Artificial Intelligence and Human-Oriented Computing*, vol. 4733, *LNAI*, R. Basili and M. T. Pazienza, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 567–578.

[33] G. L. Foresti and F. A. Pellegrino, "Automatic visual recognition of deformable objects for grasping and manipulation," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 325–333, Aug. 2004.

[34] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, 1993.

[35] E. P. Ong, W. Lin, B. J. Tye, and M. Etoh, "Fast automatic video object segmentation for content-based applications," in *Advances in Image and Video Segmentation*, Y.-J. Zhang, Ed. Hershey, PA: IRM Press, 2006, p. 156.

[36] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[37] G. Tsechpenakis, K. Rapantzikos, N. Tsapatsoulis, and S. Kollias, "A snake model for object tracking in natural sequences," *Signal Process. Image Commun.*, vol. 19, no. 3, pp. 219–238, Mar. 2004.

[38] J. Shin, H. Ki, and J. Paik, "Motion-based hierarchical active contour model for deformable object tracking," in *Proc. CAIP*, vol. 3691, *Lecture Notes in Computer Science*, A. Gagalowicz and W. Philips, Ed.s, 2005, pp. 806–813.

[39] H. Tan, X. Chen, and M. Jiang, "Object tracking based on snake and sequential Monte Carlo method," in *Proc. IEEE Int. Conf. Intell. Syst. Des. Appl.*, Oct. 2006, vol. 2, pp. 364–367.

[40] Q. Chen, Q.-S. Sun, P. A. Heng, and D.-S. Xia, "Two-stage object tracking method based on kernel and active contour," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 4, pp. 605–609, Apr. 2010.

[41] N. Bangalore, R. Young, P. Birch, and C. Chatwin, "Tracking moving objects using bandpass filter enhanced localisation and automated initialisation of active contour snakes," *Proc. ICGST-GVIP J.*, vol. 10, no. IV, pp. 1–8, 2010.

Ana-Maria Cretu (S'04–M'10) received the M.A.Sc. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada.

She is currently a Postdoctoral Fellow with the School of Electrical Engineering and Computer Science, University of Ottawa. Her research interests include machine intelligence, soft computing, biologically inspired computational models, tactile and vision sensing, and 3-D object sensing, modeling, and manipulation. She is the author of more than 35 technical papers.

Dr. Cretu is a member of the IEEE Instrumentation and Measurement Society and the IEEE Computational Intelligence Society. She served as Technical Committee Member for several international conferences and as Reviewer for journals and transactions.

Pierre Payeur (S'90–M'98) received the Ph.D. degree in electrical engineering from Université Laval, Quebec, QC, Canada, in 1999.

Since 1998, he has been with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada, where he is currently a Professor and the Director of the Sensing and Machine Vision for Automation and Robotic Intelligence Research Group. He is also a Founding Member of the Vision, Imaging, Video and Autonomous Systems Research Laboratory, University of Ottawa. His research interests include machine vision, motion capture, probabilistic 3-D modeling, range data processing, tactile sensing, robot guidance, and teleoperation. He also regularly serves as Reviewer for several journals and transactions in the same fields and as External Consultant for companies in industrial automation, sensing, and robotic applications.

Dr. Payeur is a member of the IEEE Robotics and Automation Society, the IEEE Instrumentation and Measurement Society, and the Ordre des Ingénieurs du Québec.

Emil M. Petriu (M'86–SM'88–F'01) received the Dipl.Eng. and Dr.Eng. degrees from the Politehnica University of Timişoara, Timişoara, Romania.

Since 1985, he has been on the faculty with the University of Ottawa, Ottawa, ON, Canada, where he is currently a Professor and the University Research Chair with the School of Electrical Engineering and Computer Science. During his career, he has published more than 300 technical papers, authored two books, edited two other books, and received two patents. His research interests include multisensor systems, soft computing, biology-inspired robot sensing, and human–computer symbiosis.

Dr. Petriu is a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He was a corecipient of the 2003 IEEE Donald G. Fink Prize Paper Award and a recipient of the 2009 IEEE Instrumentation and Measurement Society Distinguished Service Award.