

Modeling Elastic Objects with Neural Networks for Vision-Based Force Measurement

Michael A. Greminger

Department of Mechanical Engineering
University of Minnesota
Minneapolis, MN 55455
grem@me.umn.edu

Bradley J. Nelson

Institute of Robotics and Intelligent Systems
Swiss Federal Institute of Technology (ETH)
CH-8092 Zurich, Switzerland
brad.nelson@iris.mavt.ethz.ch

Abstract

This paper presents a method to model the deformation of an elastic object with an artificial neural network. The neural network is trained directly from images of the elastic object deforming under known loads. Using this process, models can be created for objects such as biological tissues that cannot be modeled by existing techniques. The neural network elastic model is used in conjunction with a deformable template matching algorithm to perform vision-based force measurement (VBFM). We demonstrate this learning method on objects with both linear and non-linear elastic properties.

1. Introduction

The ability to manipulate deformable objects has many important application areas particularly in the bioengineering and biomedical domains. Examples include the manipulation of biological cells and robotic surgery. When manipulating deformable objects, it is often useful to have knowledge of the force applied to the object in order to prevent damage. If vision feedback is available for observing object deformations, this feedback can be used to estimate the force applied to these objects. This technique is referred to as vision-based force measurement (VBFM).

It has been shown that if an accurate model is available that defines the elastic behavior of an object, then the force applied to that object can be measured using computer vision [2][10]. Three cases exist in which this explicit model based approach is not appropriate. The first case is for materials that exhibit nonlinear elastic properties in which it is computationally prohibitive to calculate the model in real-time. This situation occurs when there are large deflections or the stress-strain relationship for the material is nonlinear. A second case occurs when an accurate material model is unavailable for an object. Some examples are biological structures such as cells and organs. The third situation is

when a model is available for the object, but the parameters that define the model, such as material properties or geometry, are not known or are known to a low certainty.

For all three of these cases a trainable neural network model approach to VBFM is preferable. Figure 1 illustrates our neural network modeling approach to VBFM. A sequence of images is taken under various loading conditions. This sequence of images is then passed to the neural network training algorithm. Once the neural network model is trained, it can then be used to measure forces from new images.

2. Neural Network Elastic Material Model

Feed-forward two layer neural networks with the layout shown in Figure 2 are used. Each hidden node has a logistic sigmoid activation function of the form

$$g(a) = \frac{1}{1 + \exp(-a)} \quad (1)$$

where a is the sum of all of the weighted inputs to the node. The output nodes have a linear activation function that returns the sum of all of the node's weighted inputs. The neural network is represented in equation form by

$$y_i = w_{iP+1} + \sum_{j=1}^P \left[g \left(v_{jD+1} + \sum_{k=1}^D [v_{jk}x_k] \right) w_{ij} \right] \quad (2)$$

where the neural network has D inputs, P hidden nodes, and N outputs. The hidden layer weights are stored in the v matrix and the output layer weights are stored in the w matrix. Neural networks of this type have the property that they are universal approximators, meaning that the neural network can approximate a general nonlinear function to arbitrary accuracy provided that there are a sufficient number of hidden nodes [6].

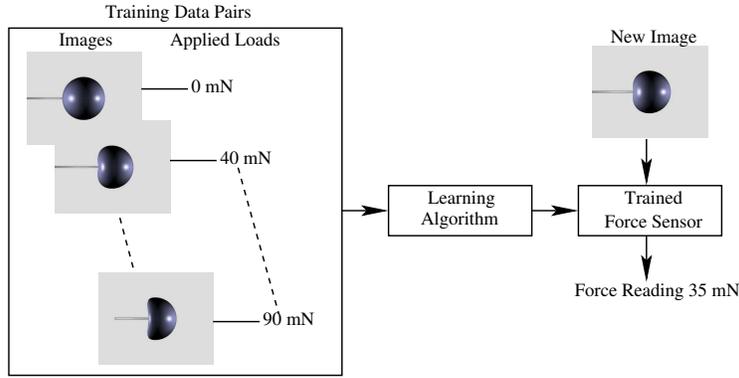


Figure 1: Diagram illustrating the learning approach to vision-based force measurement.

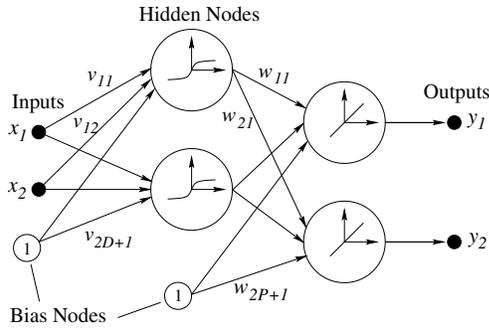


Figure 2: Neural network diagram.

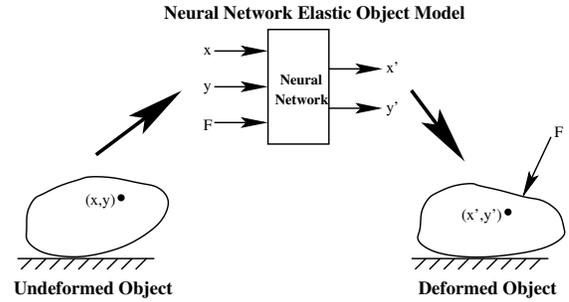


Figure 3: Process of using a neural network elastic model to calculate an object's deformation due to an applied load F .

The neural network elastic model presented in this paper inputs a point (x, y) from within the object and the load F applied to the object and returns the deformed location (x', y') of the input point. This process is shown Figure 3 for the two dimensional case. For this case, the neural network has three inputs and two outputs. If more than one load is being applied to the object, additional inputs can be added to the neural network. A neural network model constructed in this manner completely defines the deformation of an elastic object subject to an applied load F .

3. Neural Network Based Deformable Template Matching Algorithm

In order for the neural network elastic model to be useful for computer vision applications, it must be incorporated into a deformable template matching algorithm. The deformable template models both the rigid body portion of an object's motion and the object's deformation due to applied loads.

The deformable template is registered to a binary edge image [1] using a least square error measure. The template

is represented by a list of 2D vertices \mathbf{T}_i and the edge pixels in the current image are represented by the list of 2D vertices \mathbf{I}_i . The registration algorithm minimizes the distance squared between the transformed template vertices \mathbf{T}'_i and the nearest image edge vertices \mathbf{I}_i where the template vertices are transformed by an elastic transformation and a rigid body transformation.

3.1. The Rigid Body Portion of the Template Matching Algorithm

The rigid body portion of the transformation of the template vertices is simply an affine transformation given by

$$\mathbf{T}'_i = \mathbf{A}(\mathbf{T}_i) \quad (3)$$

\mathbf{A} is defined by

$$\mathbf{A}(\mathbf{T}) = \mathbf{X} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mathbf{T} \quad (4)$$

where θ is the angle of rotation and \mathbf{X} is the translation vector. The error function between the transformed tem-

plate vertices \mathbf{T}'_i and the image vertices \mathbf{I}_i can be written as

$$E(\theta, \mathbf{X}) = \sum_{j=1}^M \|\mathbf{T}'_j - \mathbf{I}_j\|^2 \quad (5)$$

where \mathbf{T}'_j is the position vector of the i th edge pixel of the template transformed by (3); \mathbf{I}_j is the position vector of the edge pixel in the image that is closest to the point \mathbf{T}'_j ; and M is the number of edge pixels in the template. This error function sums the square of the distance between each template vertex and the nearest image edge pixel. Since the transformed template vertices \mathbf{T}'_j were transformed by the affine transform \mathbf{A} , E will be a function of θ and \mathbf{X} . By minimizing E , the values of θ and \mathbf{X} that best match the image in a least squares sense will be found.

The error function E is minimized by a gradient-based multi-variable minimization technique called the Broydon-Fletcher-Goldfarb-Shanno (BFGS) method [9].

3.2. Incorporating Deformations into the Template Matching Algorithm

Minimizing (5) determines the rigid body motion of the object. We would also like to determine the non-rigid portion of the object's motion. To do this, the template is deformed according to the neural network elastic model before performing the affine transformation. The transformation becomes

$$\mathbf{T}' = \mathbf{A}(\text{NeuralNetwork}(\mathbf{T}, F)) \quad (6)$$

where $\text{NeuralNetwork}(\mathbf{T}, F)$ represents the neural network material model which returns the displaced location for the template edge pixel \mathbf{T} due to the applied force F on the object. The error function (5) becomes

$$E(\theta, \mathbf{X}, F) = \sum_{i=1}^M \|\mathbf{T}'_i - \mathbf{I}_i\|^2 \quad (7)$$

Since the error function (6) has an additional parameter F , minimizing the error function gives the applied force F in addition to the position and orientation of the object. This algorithm tracks the deformable object by finding the applied force F that, when applied to the template, causes the template to match the image.

4. Acquisition of Training Data and Network Training

As previously shown, the neural network elastic model has three inputs and two outputs. The inputs are the x-y coordinates of a point in the undeformed body and the applied load on the body. The outputs are the x-y coordinates of

the same point in the deformed body. Therefore, in order to train the neural network, it is necessary to obtain training pairs that consist of the x-y coordinate of a point in the undeformed body and the force applied to the body and the x-y coordinate of the same point in the deformed body. Many such points are needed to adequately train the neural network. These training pairs also need to be obtained for many force values over the range of expected loads.

4.1. Obtaining Training Data

The training data pairs are obtained directly from images of the object under known loads. Using this approach, a neural network model of the object is created without using an explicit material model for the object.

A computer vision algorithm is required that can measure the displacement of a point within an elastic body. A deformable body tracking algorithm is used for this purpose. There are many available deformable body tracking algorithms [4][7][12]. In order for a tracking algorithm to be useful for acquiring training data, it must be invariant to rigid body motions and it must provide a unique tracking solution. Many of the available deformable body tracking algorithms cannot be used here because they do not provide the ability to uniquely determine where a given point moves within a body. Active contour models (snakes) [5] provide a way to track deformable objects but they do not provide a way to uniquely determine where a point in an object moves in a subsequent image. Snakes also have the problem that the rigid body motions are not separated from the deformation of the object. The non-uniqueness problem occurs because the snakes algorithm is not based on an actual material model. A unique tracking solution can be obtained if we use a deformable object tracking algorithm that has knowledge of how materials deform. Two approaches that make use of material models include finite element method (FEM) based algorithms [7] and boundary element method (BEM) based algorithms [3]. We use the BEM deformable body tracking algorithm described in [3], because this algorithm is invariant to rigid body motions and provides a unique solution to the deformable body tracking problem.

The BEM method differs from the FEM method in the way the body is partitioned to solve the partial differential equations that define the elasticity problem. For the FEM method the entire domain that defines the object is partitioned into elements (see Figure 4). The BEM method only requires that the boundary of the object is partitioned. Because the BEM method only partitions the boundary of the object, it is more natural for use in computer vision problems where the boundary of the object is often all that is visible. Because the BEM deformable object tracking algorithm makes use of the equations of elasticity, it tracks a deformable contour uniquely. The left half of Figure 5

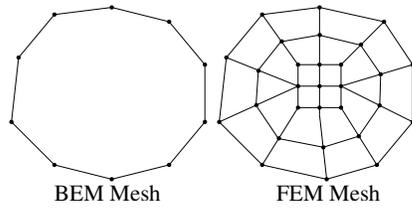


Figure 4: FEM and BEM meshes for a two dimensional object.

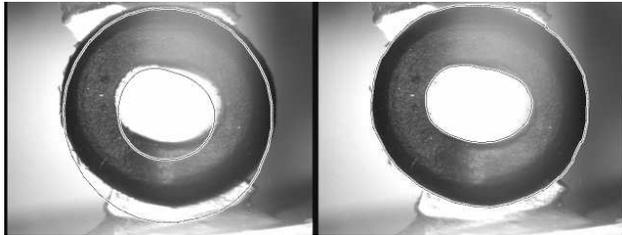


Figure 5: BEM deformable tracking algorithm tracking the rubber torus object. Undeformed template on left and template fitted to deformed torus on right.

shows a deformed rubber torus under a known load along with an undeformed BEM deformable object template. The right half of this figure shows the BEM template matched to the deformed rubber torus. This process is repeated for all of the images used to train the neural network elastic model. The training images should represent the range of loads that are likely to be encountered by the object. Figure 6 shows the training data obtained using the BEM tracking algorithm for the rubber torus object. In the figure, the same edge points are shown for the undeformed torus, the torus under a 0.613 N load and the torus under a 1.222 N load. When these point deformations are obtained for a sufficient number of different loads, they are used to train the neural network elastic model.

4.2. Training the Neural Network

The neural network is trained using the error back-propagation method [6]. The success of the training process depends on the number of training pairs that are used and the number of hidden nodes in the neural network model. If there are not a sufficient number of training data pairs, the network may model the training data well, but it will not be able to perform well for new loads that were not part of the training data. Also, if there are not a sufficient number of hidden nodes, the neural network will not be able to model the elastic object to a high degree of accuracy. However, if there are too many hidden nodes, the neural network will be able to model the training pairs very

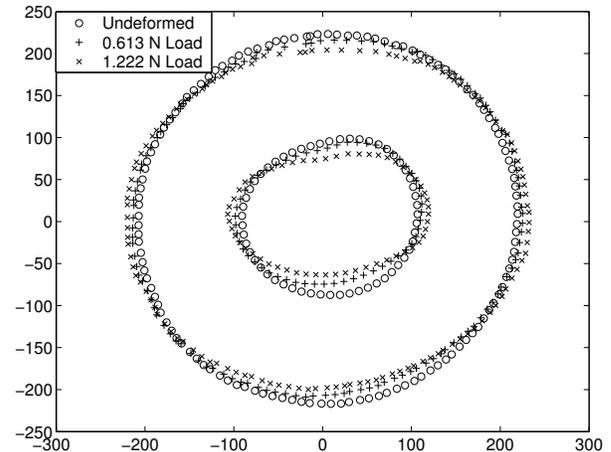


Figure 6: Training data obtained for the rubber torus object under two different applied loads.

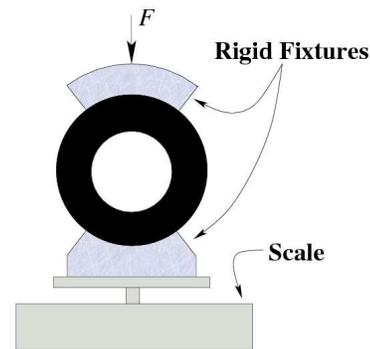


Figure 7: Rubber torus loading condition.

well but it may be very inaccurate for loads that were not part of the training data. This last phenomenon is known as over-fitting [6]. In general, the more training pairs that can be used the more accurate the model. The number of hidden nodes used should be the minimum number that still achieves adequate training.

5. Experimental Results

5.1. Rubber Torus Application

The neural network elastic model was first applied to the rubber torus object shown in Figure 5 with a force applied as shown in Figure 7. The torus, because it is constructed of rubber, has nonlinear elastic behavior. Therefore, it is computationally infeasible to model the elastic behavior of this object accurately in real-time. A neural network was trained directly from 39 images of the torus under loads between 0 and 0.84 N. There were 70 hidden nodes in the neural network model. Once the training process was com-

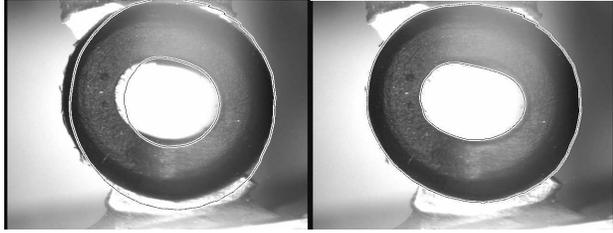


Figure 8: Neural network based deformable tracking algorithm tracking the rubber torus object. Undeformed template on left and template fitted to deformed torus on right.

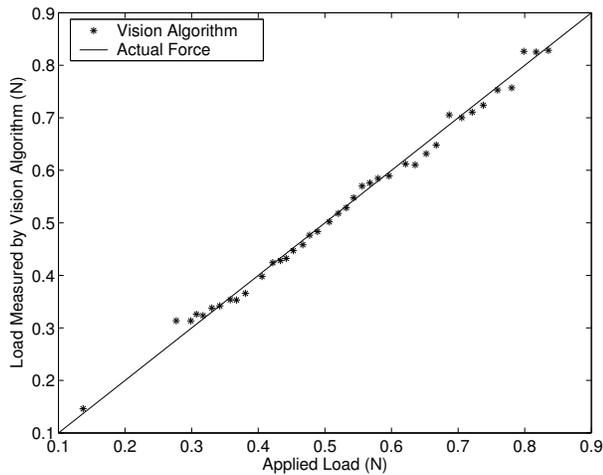


Figure 9: Neural network elastic model force measurement results for the rubber torus object.

pleted, the neural network elastic model was used to predict the force applied to the object visually as shown in Figure 8. This figure shows the torus being tracked with the deformable template based on the neural network model.

This trained model was applied to 39 images that were not used in the training process. Figure 9 shows a plot of vision-based force measurement versus applied load. The average error in the vision-based force measurement was 10.7 mN.

5.2. Microgripper Application

The neural network modeling method was also applied to the microgripper shown in Figure 10. The microgripper was cut by micro-wire EDM from spring steel 254 microns thick. A tube is slid over the microgripper to force the jaws of the gripper together in order to grasp an object [11].

When this microgripper is used for a microassembly task it is important to measure the gripping force that the microgripper applies to the object being held. Previously,

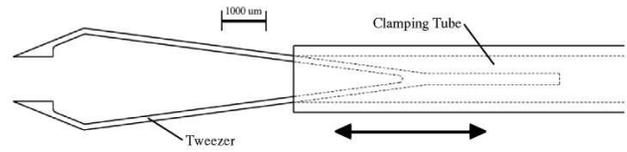


Figure 10: Microgripper.

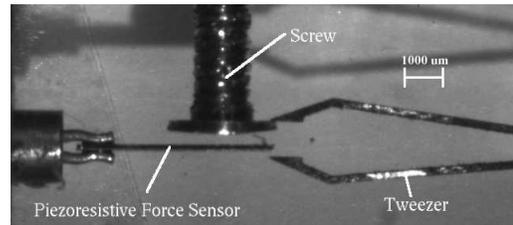


Figure 11: Experimental setup for measuring microgripper gripping force.

VBFM was applied to this gripper by modeling each jaw of the microgripper as a cantilever beam [2]. This model based approach required precise knowledge of the material properties and geometric parameters of the microgripper. It also required a precise calibration of the camera system so that deflections in pixel space can be converted into world space deflections. The experimental setup shown in Figure 11 makes use of a piezoresistive force sensor to measure the clamping force for the microgripper. Figure 12 shows a plot of force versus clamping tube position comparing the cantilever beam model based VBFM method to the piezoresistive force sensor. The average error with this model based approach was 6.0 mN.

The neural network modeling approach can be applied to the microgripper avoiding the need for precise knowledge about the parameters that define the microgripper or precise camera calibration. Figure 13 shows a plot of force versus tube position comparing the neural network based force measurement to the piezoresistive force sensor. The average error is 3.4 mN. Not only does the neural network modeling approach avoid having to model the microgripper, but it also provides a more accurate result. The neural network approach is more accurate in this case because it is not necessary to assume values for the material properties and geometry of the microgripper.

6. Conclusions

We have presented a method to model the deformation of elastic objects through the use of artificial neural networks. The neural network model can be incorporated into a deformable template matching algorithm to perform vision-

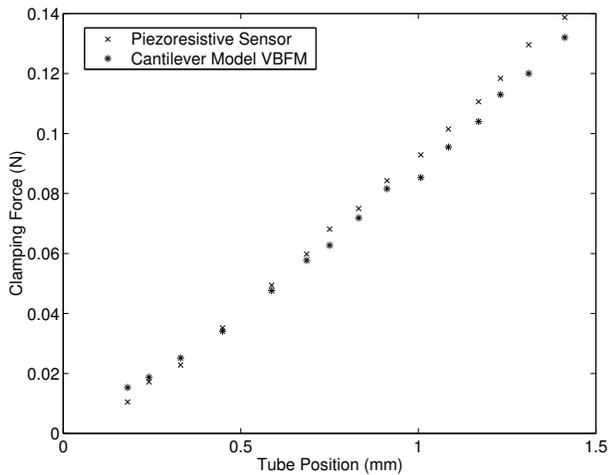


Figure 12: Cantilever beam model based VBFM results for microgripper showing an average error of 6.0 mN.

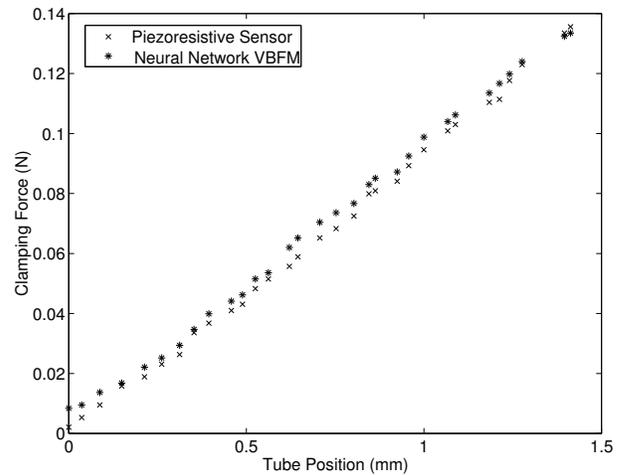


Figure 13: Neural network model based VBFM results for microgripper showing an average error of 3.4 mN.

based force measurements. This technique is useful for objects that have complex material models or that can not be accurately modeled with existing modeling techniques. It was also shown that this method can be useful even when there is an available model for the object because the neural network based method does not require knowledge of material properties or geometry. A precisely calibrated camera system is also not needed with the neural network approach.

This learning by seeing method is particularly useful for the manipulation of biological tissues or cells because it is difficult to accurately model such objects. These objects also tend to be easily damaged if excessive loads are applied, making force feedback essential.

Acknowledgments

This research was supported in part by the National Science Foundation through Grant Numbers IIS-9996061 and IIS-0208564. Michael Greminger is supported by the Computational Science Graduate Fellowship (CSGF) from the Department of Energy.

References

- [1] Canny, J., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, November 1986.
- [2] Greminger, M.A., Yang, G., Nelson, B.J., "Sensing nanonewton level forces by visually tracking structural deformations," *IEEE Int. Conf. on Robotics and Automation (ICRA2002)*, May, 2002.

- [3] Greminger, M.A., Nelson, B.J., "Deformable Object Tracking Using the Boundary Element Method," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2003)*, June 2003.
- [4] Jain, A. K., Zhong, Y., Dubuisson-Jolly, M.P., "Deformable template models: A review," *Signal Processing*, Vol. 71, pp. 109-129, 1998.
- [5] Kass, M., Witkin, A., Terzopoulos, D., "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 321-331, 1988.
- [6] Kecman, V., *Learning and Soft Computing*, MIT Press, Cambridge, Mass, 2001.
- [7] Metaxas, D., *Physics-Based Deformable Models*, Kluwer Academic Publishers, Boston, Mass, 1997.
- [8] Tsap, L., Goldgof, D., Sarkar, S., "Efficient Nonlinear Finite Element Modeling of Nonrigid Objects via Optimization of Mesh Models," *Computer Vision and Image Understanding*, Vol. 69, pp. 330-350, March 1998.
- [9] Vanderplaats, G., *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, 1984.
- [10] Wang, X., Ananthasuresh, G. K., Ostrowski, J. P., "Vision-based sensing of forces in elastic objects," *Sensors & Actuators A-Physical*, Vol. 94, No. 3, pp. 142-156, November 2001.
- [11] Yang, G., Gaines, J. A., Nelson, B. J., "A Flexible Experimental Workcell for Efficient and Reliable Wafer-Level 3D Microassembly," *Proc. 2001 IEEE Int. Conf. of Robotics and Automation (ICRA2001)*, Seoul, Korea, May 2001.
- [12] Yuille, A., Cohen, D., Hallinan, W., "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, Vol. 8, No. 2, pp. 99-111, 1992.