Interactive Virtual Environments

Modelling 3D Geometric and Elastic Properties

Emil M. Petriu, Dr. Eng., FIEEE
Professor, School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada
http://www.site.uottawa.ca/~petriu

May 2008
Interactive Model-Based Hapto-Visual Teleoperation - a human operator equipped with haptic HCI can telemanipulate physical objects with the help of a robotic equipped with haptic sensors.
Model-based approach, based on the kinematics and dynamics of the object handled with the fingertips, provides a convenient representation of the dexterous manipulation. Quoting Salisbury et al.’s recent survey of haptic rendering, “improved accuracy and richness in object modeling and haptic rendering will require advances in our understanding of how to represent and render psychophysically and cognitively germane attributes of objects, as well as algorithms and perhaps specialty hardware (such as haptic or physics engines) to perform real-time computations”.

-------------
Modelling allows to simulate the behavior of a system for a variety of initial conditions, excitations and systems configurations.

The quality and the degree of the approximation of the model can be determined only by a validation against experimental measurements.

The convenience of the model means that it is capable of performing extensive parametric studies, in which independent parameters describing the model can be varied over a specified range in order to gain a global understanding of the response.

Discreet vs. Continuous Modelling of Physical Objects and Processes

DISCREET MODEL
• sampling => INTERPOLATION COST
  \[ y(j) = y(A) + \frac{[x(j)-x(B)] \cdot [y(B)-y(A)]}{[x(A)-x(B)]} \]

CONTINUOUS MODEL
• NO sampling => NO INTERPOLATION COST
Both the Analog Computers and the Neural Networks are *continuous modelling devices*.

The Analog Computer (AC) allows to *solve the linear or nonlinear differential and/or integral equations representing mathematical model* of a given physical process. The coefficients of these equations must be exactly known as they are used to program/adjust the coefficient-potentiometers of the AC’s computing-elements (OpAmps). The AC doesn’t follow a sequential computation, all its computing elements perform simultaneously and continuously. An interesting note, “because of the difficulties inherent in analog differentiation the [differential] equation is rearranged so that it can solved by integration rather than differentiation.” [A.S. Jackson, *Analog Computation*, McGraw-Hill Book Co., 1960].
The Neural Network (NN) doesn’t require a prior mathematical model. A learning algorithm is used to adjust, sequentially by trial and error during the learning phase, the synaptic-weights/coefficient-potentiometers of the neurons/computing-elements.

Similarly to the analog computer, a NN doesn’t follow a sequential computation algorithm, all its neurons performing simultaneously and continuously. The neurons are also integrative-type processing elements.
Compare the performance of three NN architectures used for 3D object shape modelling:

- Multilayer Feedforward (MLFF)
- Self-Organizing Map (SOM)
- Neural Gas Network

Starting from a pointcloud of sample points capturing the shape of the object to be modeled, the NN produces a volumetric representation of the object.

The surface of an object is described by a set of zeros in the output response of the NN, while the interior or exterior regions of the object are described by negative or positive value, respectively.
The NN model provides a continuous representation of the object surface and permits an extensive study not only of the modeled object surface, but of the entire object volume. Thus it inherits the advantages of the polygonal and surface models, without inheriting their drawbacks.

It can be used to perform simple operations such as
• object morphing,
• set operations, and
• object collision detection
Initial 3D pointcloud of sample points representing the object \( \{(x_i, y_i, z_i) \mid i = 1, \ldots, N\} \)

Transformed (translated, rotated, scaled, bent, tapered, twisted) object \( \{(X_i, Y_i, Z_i) \mid i = 1, \ldots, N\} \)

Neural-Netwrok model of the object \( \{(x_p, y_p, z_p) \mid p = 1, \ldots, P\} \)

Neural Network Architecture for 3D Object Representation

MLFF
MLFF
SOM
Neural Gas
Transformation Module:

*translation, rotation, scaling, and deformations (bending, tapering, twisting)*

Pointcloud of sample points representing the object $O$

\[
\{(x_i, y_i, z_i) \mid i = 1, \ldots, N\}
\]

$\text{MLFF Neural Network}$

$X_i, Y_i, Z_i$

Transformed object pointcloud

\[
\{(X_i, Y_i, Z_i) \mid i = 1, \ldots, N\}
\]
The transformation module implements:
  * rotation,
  * translation,
  * scaling, and
  * deformations such as:
    - bending,
    - tapering, and
    - twisting.
Transformation Module - Generation Mode

Original

Rotation

Translation, Rotation, Scaling

Tapering

Bending

Twisting
The transformation module has no hidden layer and the outputs have linear activation functions. Knowing the equations, the network weights will be set such that the desired transformation is performed.

The transformation module can learn and store in its weights information on how these points have been transformed and/or deformed.

The translation, rotation, scaling, and tapering will be encoded in the weights implementing the generalized 3D transformation matrix.
Rotation and Translation

Rotation

\[ m_1 = a_1 \cos \theta \cos \phi \]
\[ m_2 = a_1 (\cos \theta \sin \phi \sin \psi - \sin \theta \cos \psi) \]
\[ m_3 = a_1 (\cos \theta \sin \phi \cos \psi + \sin \theta \sin \psi) \]
\[ n_1 = a_2 \sin \theta \cos \phi \]
\[ n_2 = a_2 (\sin \theta \sin \phi \sin \psi - \cos \theta \cos \psi) \]
\[ n_3 = a_2 (\sin \theta \sin \phi \cos \psi - \cos \theta \sin \psi) \]
\[ q_1 = -a_3 \sin \phi \]
\[ q_2 = a_3 \cos \phi \sin \psi \]
\[ q_3 = a_3 \cos \phi \cos \psi \]
MLFF Representation - Results

250 points, 6-3-1, 1 extra surface, $d=0.055$, 550 epochs, 7 min.

19080 points, 10-5-1, 5 extra surfaces, $d=0.055$, 1200 epochs, 2.8 hrs.

7440 points, 8-4-1, 5 extra surfaces, $d=0.055$, 1100 epochs, 1 hr
MLFF Representation - Results

51096 points, 20-10-1, 5 extra surfaces, d=0.055, 2000 epochs, 5.2 hrs.

19000 points, 14-7-1, 4 extra surfaces, d=0.055, 1100 epochs, 3.3 hrs

2500 points, 12-6-1, 2 extra surfaces, d=0.06, 1020 epochs, 45 min.
MLFF Representation – Applications

Object Morphing

Reference weight matrix $W_1$

$W_{12(i)}$

Target weight matrix $W_2$

$(x, y, z)$ points

linear interpolation $(n \text{ steps}, i = 1, \ldots, n)$ between $W_1$ and $W_2$

$O_1$

$O_2$

$(x_m, y_m, z_m)$ of the morphed object
MLFF Representation – Applications

Set Operations

(x, y, z) belongs to union

(x, y, z) belongs to intersection

(x, y, z) belongs to difference
MLFF Representation – Applications

Object Collision Detection

$NN_{\text{object 2}}$ $O_{i} < 0$

collision between object 1 and object 2

sampled points object 1

50% $96.56%$ $97%$ $2.3%$ $0%$
The model-based recognition is done by first aligning the transformed given object with the reference model stored in the database.

The NN of the transformation module is trained using the reference model points as training points and the points of the given object as targets. In this way, the transformation information is stored in the NN weights.

After the alignment, a set of points is sampled from the transformed given object and then tested to see the degree of belonging to the reference model.
### MLFF Representation – Applications

#### Object Recognition

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4%</td>
<td>4.9%</td>
<td>1.6%</td>
<td>99.1%</td>
<td></td>
</tr>
<tr>
<td>93.3%</td>
<td>3.3%</td>
<td>91.7%</td>
<td>3.1%</td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td>99.1%</td>
<td>5.78%</td>
<td>98.3%</td>
<td></td>
</tr>
<tr>
<td>91.7%</td>
<td>6.6%</td>
<td>1.6%</td>
<td>92.5%</td>
<td></td>
</tr>
</tbody>
</table>
## MLFF Neural Network Modelling – Summary

### Advantages
- Simple and compact architecture
- Continuous volumetric model (though trained with surface)
- Information about the entire object space
- Provides desired accuracy
- Represents objects of varied complexity
- Preserves details
- Morphing, set operations, recognition, collision detection

### Disadvantages
- Computationally intensive (for both learning and rendering)
- Lack of local control of the object
**SOM and Neural Gas - Compressed Representation Models**

Pointcloud of sample points representing the object \( O \)
\[ \{(x_i, y_i, z_i) | i = 1, ..., N\} \]

\( x, y, z \)

\( x_{p}, y_{p}, z_{p} \)

Compressed NN model of the 3D object
\[ \{(x_{p}, y_{p}, z_{p}) | p = 1, 2, ..., P\}, \text{ where } P < N \]
**SOM Representation – NN Architecture**

- **Activation Function**
  - soft competition
- **Learning**
  - unsupervised

**Diagram Notes:**
- Input layer: \([x_i, y_i, z_i]\)
- Weight: \(w_{ji}\)
- Winning neuron

**Diagram:**
- SOM architecture with input layer and connected neurons.
A SOM has an input layer, where vector containing the $x, y, z$ coordinates of points will be presented. Computations are feedforward in the first layer and bidirectional in the connection (output layer). The purpose of the lateral connection (represented by a Gaussian) is to reinforce the activation values of strong neurons and decrease the activation of weak ones. The output neurons are arranged on a grid, usually 2D. The network is based on competition - soft competition. The winning neurons is the one closest to the input vector. Its value and the value of a given neighbourhood are updated during learning. After learning, two vectors belonging to the same cluster will be projected on two close neurons in the output space.
Neural Gas Representation – NN Architecture

- Activation Functions:
  - soft competition
  - neighbourhood ranking

- Learning
  - unsupervised
The NG releases the neurons from the output grid. There are no connections between units in the connection layer. The nodes move independently over the data space. It also exploits the idea of soft competition, but the neurons to be updated are not selected according to topological relation, but rather according to rank. The rank is the rank the neurons have in an ordered list of distances between their weights and the input vector. NG converges quickly and to a lower distortion error.
**SOM and Neural Gas**  Modelling - Results

**Initial pointcloud**

- 19080 points
- 14914 points
- 13759 points

**Neural Gas**

- 1125 points, 42 min.
- 875 points, 24.5 min.
- 875 points, 22 min.

**er = 0.0098**

**SOM**

- 1125 points, 26 min.
- 875 points, 11 min.
- 875 points, 10 min.

**er = 0.0125**
SOM and Neural Gas Modelling – Applications ➚ Object Morphing
SOM and Neural Gas Modelling – Applications ➞ Segmentation

Data clusters
**SOM and Neural Gas Modelling – Summary**

**Advantages**
- simple and compact (weights)
- compressed
- less memory usage
- desired accuracy
- objects of varied complexity
- details
- morphing, motion detection, segmentation

**Disadvantages**
- computational expensive for high accuracy
- no information about the object space
- no direct surface representation
MLFF, SOM, and Natural Gas Modelling – Performance Comparison

Training Time

MLFF: 3.3 hrs
Neural Gas: 42 min.
SOM: 26 min.
MLFF, SOM, and Natural Gas Modelling – Performance Comparison

MLFNN
- computational time = construction time + generation time + rendering

SOM and Neural Gas
- computational time = construction time + rendering

Models
- hand
- pliers
- face
- statue
As it can be seen, the MLFFNN is the most computational expensive, followed by NG. This time is only the construction time. However, the MLFFNN requires a generation time as well to reconstruct the model given the weights and self-organizing architectures require the rendering time. For the point based rendering technique, the self-organizing architectures are still less computationally expensive than the MLFFNN.
MLFF, SOM, and Natural Gas Modelling – Performance Comparison Compactness
The use of neural network modeling is advantageous from the point of view simplicity and compactness.

MLFNN – provide continuous models, information on the entire object space, convenient for many applications, however they are time consuming.

SOM and Neural Gas – provide compressed models while maintaining the properties of the objects, have very good accuracy, and they are less time consuming.

The use of any specific techniques depends on the application requirements.
Neural Network Adaptive Sampling of 3D Object Elastic Properties

Recovery of the elastic material properties requires touching each point of interest on the explored object surface and then conducting a strain-stress relation measurement on each point.

Tactile probing is a time consuming
Sequential operation

\[ \begin{align*}
\text{if } E_p \varepsilon_p & \leq \varepsilon_{p,\text{max}} \\
\sigma_p &= E_p \cdot \varepsilon_p \\
\text{if } \varepsilon_{p,\text{max}} < \varepsilon_p \\
\sigma_p &= \sigma_{p,\text{max}}
\end{align*} \]

The elastic behaviour at any given point \((x_p, y_p, z_p)\) on the object surface is described by the Hooke's law:

where \(E_p\) is the modulus of elasticity, \(s_p\) is the stress, and \(\varepsilon_p\) is the strain on the normal direction.

**Find fast sampling procedures able to minimize the number of the sampling points by selecting only those points that are relevant to the elastic characteristics.**

**non-uniform adaptive sampling algorithm of the object’s surface,**
which exploits the SOM (self-organizing map) ability to find optimal finite quantization of the input space.
Adaptive Sampling Control of the Robotic Tactile Probing of Elastic Properties of 3D Object Surfaces

Initial 3D geometric model of the object's surface \[ \{(x_i, y_i, z_i) \mid i = 1, \ldots, N\} \]

\( x_i, y_i, z_i \)

SOM / Neural Gas

Adaptive-sampled 3D geometric model of the object surface \[ \{(x_p, y_p, z_p) \mid p = 1, \ldots, P\} \]

\( x_p, y_p, z_p \)

Adaptive-sampled 3D geometric & elastic composite model of object's surface \[ \{(x_p, y_p, z_p, E_p) \mid p = 1, \ldots, P\} \]

\( E_p \)

Robotic Tactile Probing
SOM (Self Organizing Map) and Neural Gas NN architectures are both used to build compressed model of the 3D object originally defined as a point-cloud. The weight vector will consist of the 3D coordinates of the object’s points.

During the learning procedure, the model will contract asymptotically towards the points in the input space, respecting their density and thus taking the shape of the object encoded in the point-cloud.

Data point-clouds obtained with a range scanner are used to train the network. Normalization is employed to remove redundant information from a data set, by a linear rescaling of the input vectors such that their variance is 1.

In order to evaluate the quality of the models, a straightforward measure of the precision is used. The precision is estimated as the average distance between each data vector and its winning neuron.
Robustness to noisy training data

(a) Training data set of 3721 points
(b) Neural Gas network, error=0.0112,
(c) SOM, error=0.0133
(d) Noisy data set, random 0 – 0.1,
(e) Neural Gas network error=0.0383,
(f) SOM, error=0.0266
(g) Noisy data set, random 0 – 0.04,
(h) Neural Gas network error=0.0224,
(i) SOM, error=0.0241
Training Neural Gas network
with a map size of 25×45 for 20 epochs, for \( \alpha_0 = 0.5 \), \( \lambda_0 = \) number of neurons/2 and a SOM, with the initial neighborhood radius \( \sigma_0 = 5 \), and a map size of 25×45, trained for 100 epochs, with data corrupted by different levels of noise.

The initial set of 3721 points is reduced to 1125 points.

It takes approximately 250s for the SOM to build a model of a sphere, while it takes approximately double for the Neural Gas NN. However, even for a larger number of training epochs (5 times more) the SOM does not reach the same accuracy as the Neural Gas NN does, for data that is not very noisy (a random noise level below 0.1).

SOM suffers from the boundary problem. The models obtained look as if they contain cavities.
For low levels of noise the Neural Gas network performs better than SOM. For higher level of noise, SOM tends to smooth the effect of noise, while the Neural Gas network, which has high sensitivity, follows the noisy patterns.
• The first column presents three views of the original point-cloud of 19,080 points representing a human face.

• The second column presents the compressed model of 1,152 points obtained using The Neural Gas network.

• The third column presents the compressed model of 1,152 points obtained using SOM.
Qualitative comparison between the Neural Gas and the SOM adaptive sampled models.

- The map sizes are equal for both networks.
- The *first column* represents the original point-cloud,
- The *second column* represents the Neural Gas model.
- The *third column* represents the SOM model.
For the 14,914 points of the original point-cloud model given in the first figure, it takes 24 min. to build the Neural Gas model shown in the second figure and 11 min. to build the SOM model shown in the third figure (for the same map size of 25x35 in both cases).
For both, Neural Gas and SOM, networks the quality is improving with the number of training epochs.
On the whole the quality of the Neural Gas models appears to be better. Because of the boundary problem, the SOM models are to be avoided for non-noisy data.

- Neural Gas and SOM neural networks are both able to compress the initial model with the desired degree of accuracy.

- The number of points can be further reduced by reducing the map size. However, there is a compromise to be made between the quality of the resulting compressed model and the map size.

- Neural Gas networks are able to model an entire scene of objects while the SOM networks are not able of such a performance.
Starting from a 3D point-cloud, a neural gas NN yields a reduced set of points on the 3D object’s surface which are relevant for the tactile probing. The density of these tactile probing points is higher in the regions with more pronounced variations in the geometric shape. A feedforward NN is then employed to model the force/displacement behavior of selected sampled points that are probed simultaneously by a force/torque sensor and the active range finder.
Variable elasticity object used for experimentation.

Sampling points selected with the neural gas network.

Elastic ball used for experimentation.

Sampling points selected with the neural gas network for the ball.

Different magnitudes of a normal force are applied successively on the selected sampling points using the probe attached on the force/torque sensor and a range profile is collected with the laser range finder for each force magnitude.

There is no need to recover the explicit displacement information from the range profiles. Instead the NN models use the raw range data as a function of applied force, $F$, without explicitly defining values for the displacement. For each cluster of similar elasticity, a feed-forward NN with two input neurons ($F$ and $a$), 45 hidden neurons ($H_1$-$H_{45}$) and one output neuron ($Z$), is used to learn the relation between forces and the corresponding geometric profiles provided by the range finder.
The NN associated with each material were trained for 10,000 epochs using the Levenberg-Marquardt variation backpropagation algorithm with the learning rate set to 0.1. The whole data set is used for training in order to provide enough samples. The training takes approximately 10 min. on a Pentium IV 1.3GHz machine with 512MB memory. For the rubber, the sum-squared error reached during training is $3.7 \times 10^{-3}$, for cardboard is $3.5 \times 10^{-2}$ while for the foam is $2.2 \times 10^{-2}$. As expected, the error is lower for the rubber where data is more compact and less noisy, while it remains slightly higher for the cardboard and even higher for the foam. But in all cases, excellent convergence is achieved.

Deformation profiles for semi-stiff material (cardboard).

Deformation profiles for smooth material (foam).

Real and modeled deformation curves using neural network for semi-stiff material (cardboard) under a normal force of: a) $F=0.1\text{N}$, b) $F=0.37\text{N}$, and c) $F=2.65\text{N}$.

Real and modeled deformation curves using neural network for smooth material (foam) under a normal force of: a) $F=0\text{N}$, b) $F=0.93\text{N}$, and c) $F=3.37\text{N}$.

Real and modeled deformation curves using neural network for rubber under a normal force of: a) $F=0\text{N}$, b) $F=65.52\text{N}$, and c) $F=80.5\text{N}$.

Real and modeled deformation curves using neural network for rubber under forces applied at different angles:

a) \( F=65\text{N}, a_1=10^\circ \) and \( F=65\text{N}, a_2=170^\circ \),

b) \( F=36\text{N}, a_1=25^\circ \), and \( F=36\text{N}, a_2=155^\circ \)

Real, modeled and estimated deformation profiles detail of estimated deformation profiles using neural network for rubber ball for increasing forces applied at 75-degree angle.

Hardware Neural Network Architectures

ANNs / Neurocomputers => architectures optimized for neuron model implementation

- *general-purpose*, able to emulate a wide range of NN models;
- *special-purpose*, dedicated to a specific NN model.

Hardware NNs consisting of a collection of simple neuron circuits provide the massive computational parallelism allowing for a higher Model rendering speed.
ANN VLSI Architectures:
- analog ==> compact, high speed, asynchronous, no quantization errors, convenient weight “+” and “X”;
- digital ==> more efficient VLSI technology, robust, convenient weight storage;

Pulse Data Representation:
- Pulse Amplitude Modulation (PAM) - not satisfactory for NN processing;
- Pulse Width Modulation (PWM);
- Pulse Frequency Modulation (PFM).

Pulse Stream ANNs: combination of different pulse data representation methods and opportunistic use of both analog and digital implementation techniques.
Looking for a model to prove that algebraic operations with analog variables can be performed by logical gates, von Neuman advanced in 1956 the idea of representing analog variables by the mean rate of random-pulse streams [J. von Neuman, “Probabilistic logics and the synthesis of reliable organisms from unreliable components,” in Automata Studies, (C.E. Shannon, Ed.), Princeton, NJ, Princeton University Press, 1956].

The “random-pulse machine” concept, [S.T. Ribeiro, “Random-pulse machines,” IEEE Trans. Electron. Comp., vol. EC-16, no. 3, pp. 261-276, 1967], a.k.a. "noise computer", "stochastic computing", “dithering” deals with analog variables represented by the mean rate of random-pulse streams allowing to use digital circuits to perform arithmetic operations. This concept presents a good tradeoff between the electronic circuit complexity and the computational accuracy. The resulting neural network architecture has a high packing density and is well suited for very large scale integration (VLSI).
Generalized $b$-bit analog/random-data conversion and its quantization characteristics

Moving Average ‘Random Pulse -to- Digital” Conversion
2-bit random-data NN architecture of an auto-associative memory

Training set

Recovery of 30% occluded patterns

\[ a = \text{hardlim}(W * P) \]
Conclusions

- Model-based approach, based on the kinematics and dynamics of the object handled with the fingertips, provides a convenient representation of the dexterous manipulation. “Improved accuracy and richness in object modeling and haptic rendering will require advances in our understanding of how to represent and render psychophysically and cognitively germane attributes of objects, as well as algorithms and perhaps specialty hardware (such as haptic or physics engines) to perform real-time computations” [K. Salisbury, F. Conti, F. Barbagli, “Haptic Rendering: Introductory Concepts,” IEEE Computer Graphics and Applications, Vol. 24, No. 2, pp. 24 – 32, 2004].

- Neural Networks which are able to learn nonlinear behaviors from a limited set of measurement data can provide efficient and compact multi-media object modeling solutions. Due to their continuous, analog-like, memory behavior, NNs are able to provide instantaneously an estimation of the output value for input values that were not part of the initial training set.

- NNs consisting of a collection of simple neuron circuits provide the massive computational parallelism offering efficient storage, model transformation, and real-time rendering capabilities for large numbers of composite geometric & haptic object models involved in the model-based interactive telemanipulation.
Ottawa “U” Research Group – Relevant Graduate Theses

Ottawa “U” Research Group - Publications in Modelling 3D Object Geometry and Elastic Properties