Neural Networks: Modeling Applications

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

NEURAL NETWORK MODELS OF PHYSICAL PROCESSES

Modelling allows to simulate the behavior of a system for a variety of initial conditions,excitations and systems configurations - often in a much shorter time than would be required to physically build and test a prototype experimentally

- 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.

➡ A more relevant model might be one which provides results more rapidly - even if a degradation in a solution accuracy results.

Analog Computer vs. Neural Network Tools for Physical Processes Modelling

- 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.
As an interesting note, "because of the difficulties inherent in analog differentiation the [differential] equation is rearranged so that it can be solved by integration rather than differentiation." [A.S. Jackson, *Analog Computation*, McGraw-Hill Book Co., 1960].

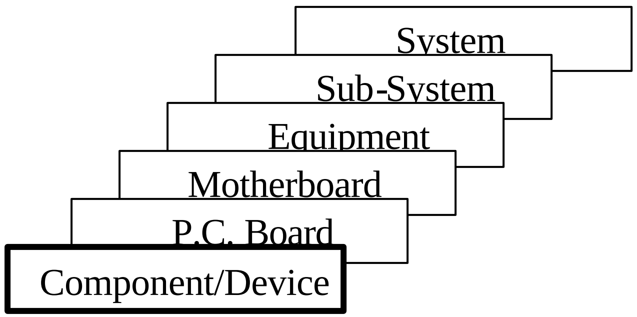
- The Neural Network (NN) doesn't require a prior mathematical model. A *learning algorithm* is used to adjust, sequentially by trail and error during the learning phase, the synaptic-weights/ coefficient-potentiometers of the neurons/computing-elements. As the AC, the NN don't follow a sequential computation, all its neuron performing simultaneously and continuously. The neurons are also integrative-type computing/processing elements.

NN Modelling of 3D Electromagnetic Fields for a Virtual Prototyping Environment

E.M. Petriu, M. Cordea, D.C. Petriu, Lou McNamee, "Modelling Issues in Virtual Prototyping Environments," Proc. VIMS'99, *IEEE Workshop Virtual and Intell. Meas. Syst.,* pp. 1-5, Venice, Italy, May 1999

EMC Modelling for Electronic Design Automation

EMC Design Levels



Optimum Approach to EMC Design

- {Design+Test+Analysis} **Synergy**
- **EMC_Behavior** = F (Design_Principle, Analysis&Modeling&Simulation_Tools, Test_Methodology&Instrumentation)

- Multiple PCBs can be integrated in any way as desired to define a complete electronic system, including mechanical parts.

- The final system can be ***interactively*** tested on an ***enhanced-reality virtual work-bench*** as a final product, by ***concurrently*** running what-if experiments in a ***multi-domain*** *(*mechanical, electrical, thermal)environment.

⇨ The design cycle is shortened, the cost of the tests is reduced, the quality of the product is improved, and the time-to-market is reduced.

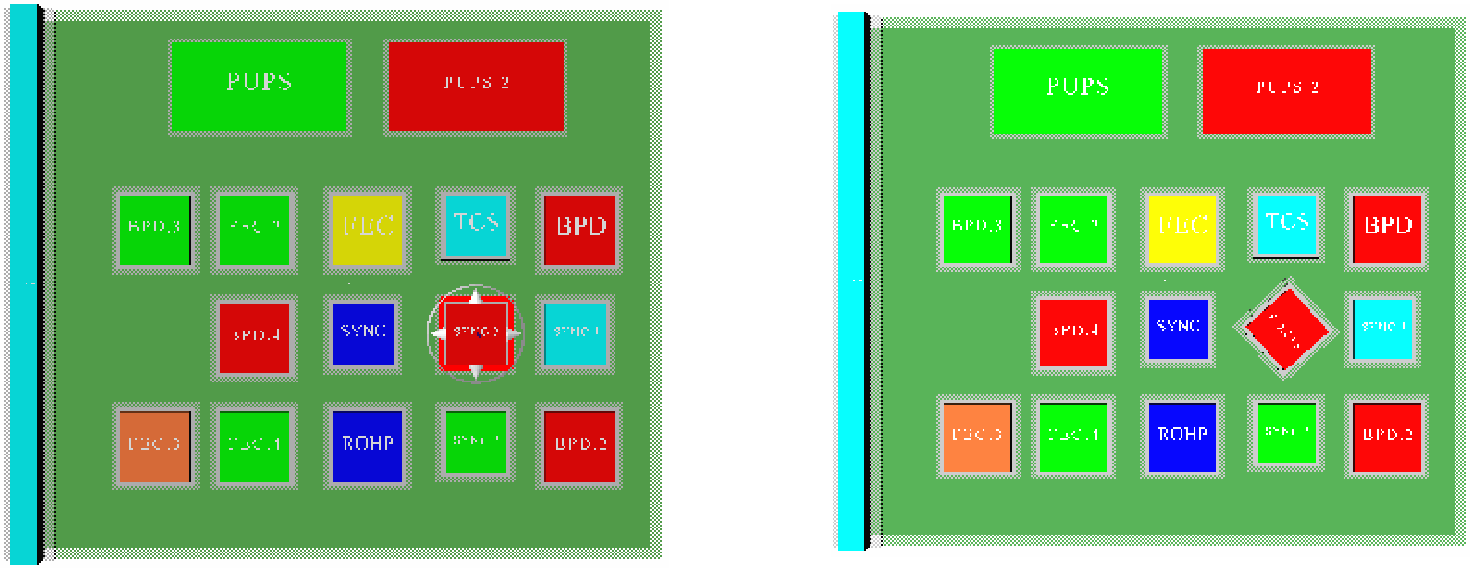
EM Virtual Prototyping Environment for the Interactive Design of Very High Speed Circuits

- **user-centered**, **task driven** point of view;

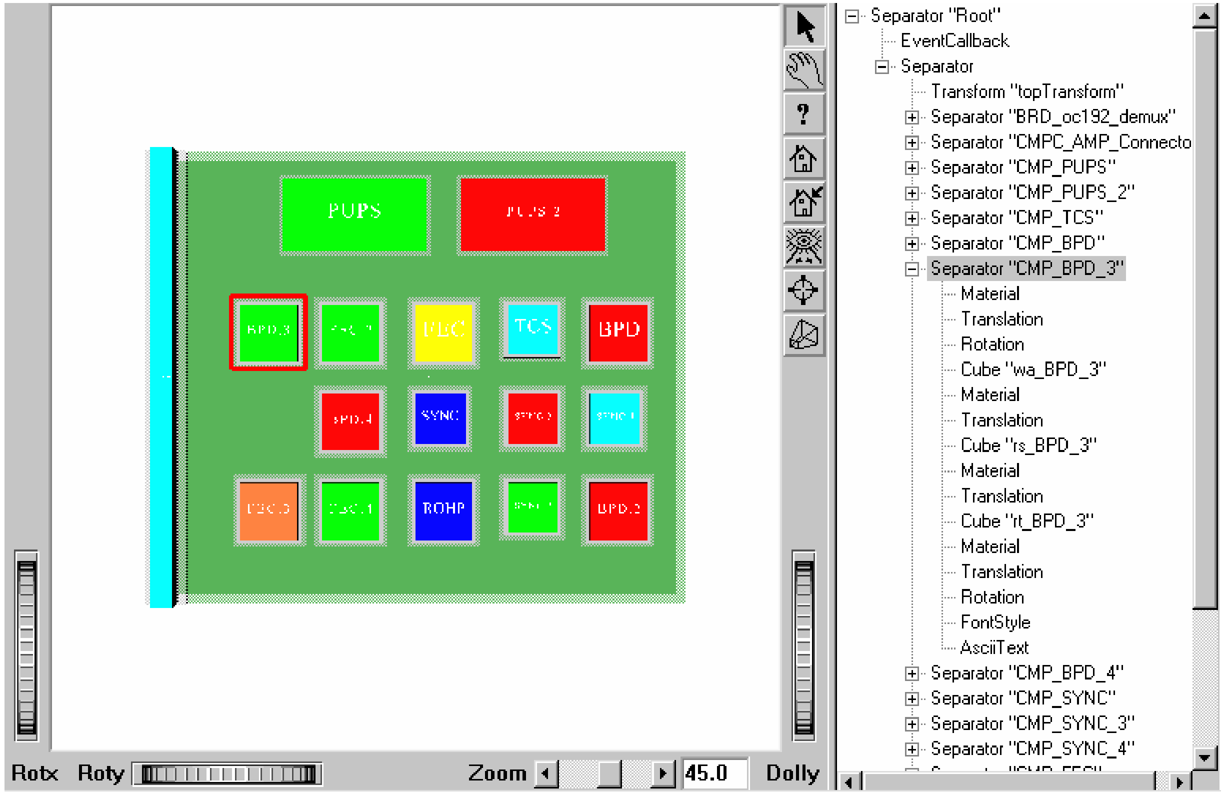
- **interactive functions**:
  (i) walk-through the 3D virtual world;
  (ii) specify material, electrical, and thermal specifications of circuit components;
  (iii) 3D manipulation of the position, shape, size, of the circuit components and layout;
  (iv) visualization the electrical wave forms, **3D Electromagnetic (EM) field** and thermal field effects in different regions of the electronic circuit.

- 3D scenes are composed of multiple objects: boards, components, connectors.
  - any object is characterized by its usual 3D geometric shape and *safety-envelopes* (the 3D geometric space points where the intensity of a given field radiated by that object becomes smaller than a specified threshold value), each type of field (EM, thermal) will have its own safety-envelope (the geometric safety-envelope being the object shape itself);
  - any object can be selected/becomes *active* by attaching a manipulator to it;

- The ***main objective*** is to detect a collision caused by a linear transformation (translation, rotation or scaling) between the selected object and the other objects in the scene.
  - for each transformation of the selected/active object, the program updates the 3D geometric parameters and the bounding box of the object;
  - then the program checks for collision between the safety-envelopes selected object and those of the other objects in the scene;
  - when a collision is detected, the active object returns to its position just before the collision

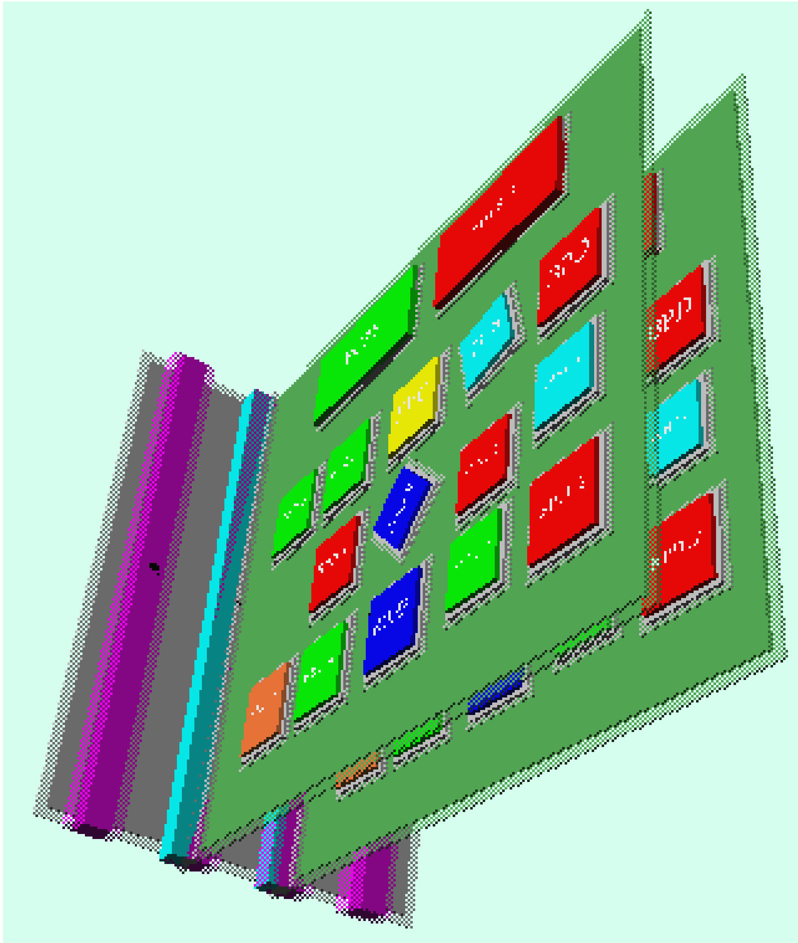


Rotation-translation manipulator dragger



Editing material properties

Assembling multiple PCBs



Electromagnetic Compatibility (EMC) Modelling Methods

- *circuit theory* to describe the conducted disturbances (such as overvoltages, voltage dips, voltage interruptions, harmonics, common ground coupling);
- *equivalent circuit* with either *distributed* or *lumped parameters* (such as in low frequency electromagnetic field coupling expressed in terms of mutual inductances and stray capacitances, field-to-line coupling using the transmission line approximation, and cable crosstalk);
- formal solutions to *Maxwell's equations* and the appropriate field boundary conditions (as for example in problems involving antenna scattering and radiation).

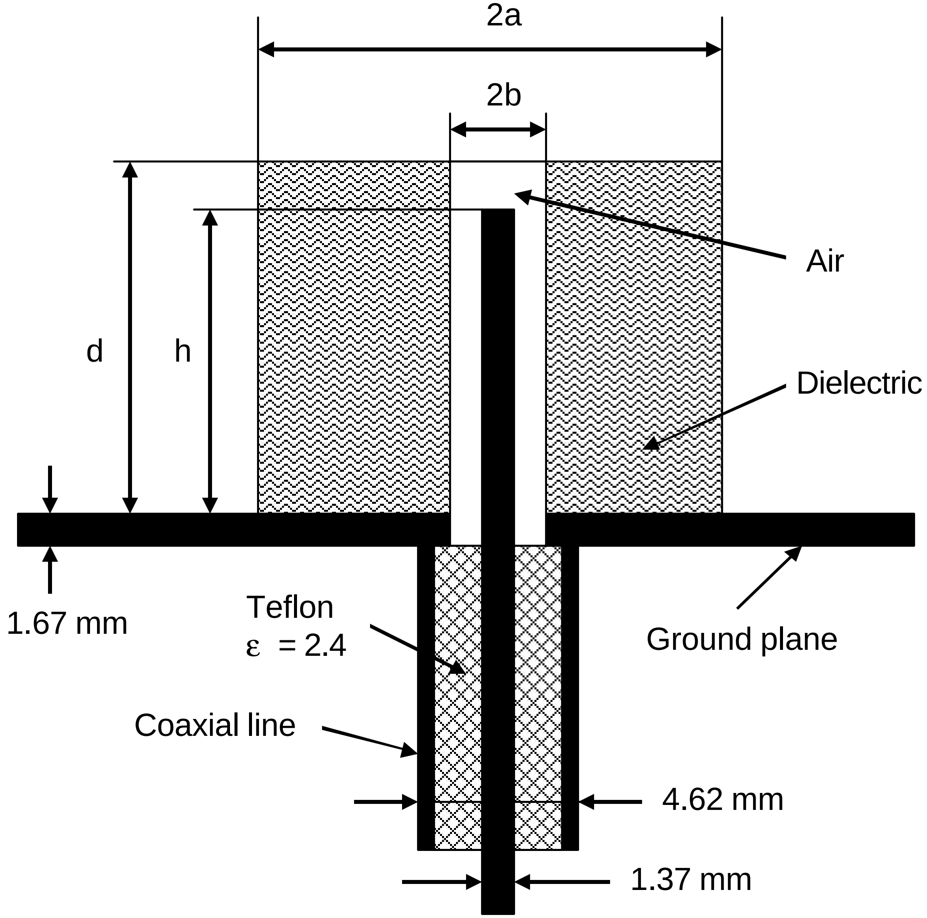
Parallel and Distributed Processing Techniques for Electromagnetic Field Solution

- **Classical numerical EM modelling** using <u>sequential algorithms</u> such as TLM (transmission-line matrix) or FEM (finite element method) is <u>computer intensive</u>, particularly as spatial discretization, geometry complexity, and domain size requirements become more demanding.

⇩

- More efficient **parallel and distributed computing** techniques must be developed to <u>reduce the execution time</u> for these methods so that they can <u>be used in commercial CAD software</u>. Speed of execution is particularly important when the field analysis is to be coupled with optimization, which may require several hundred analyses to be performed within a reasonable time. ⇨ **NN models**

NN modeling of the 3D EM field radiated by a dielectric-ring resonator antenna



- *I. Ratner, H.O. Ali, E.M. Petriu, "Neural Network Simulation of a Dielectric Ring Resonator Antenna," J. Systems Architecture, vol. 44, No. 8, pp. 569-581, 1998.*

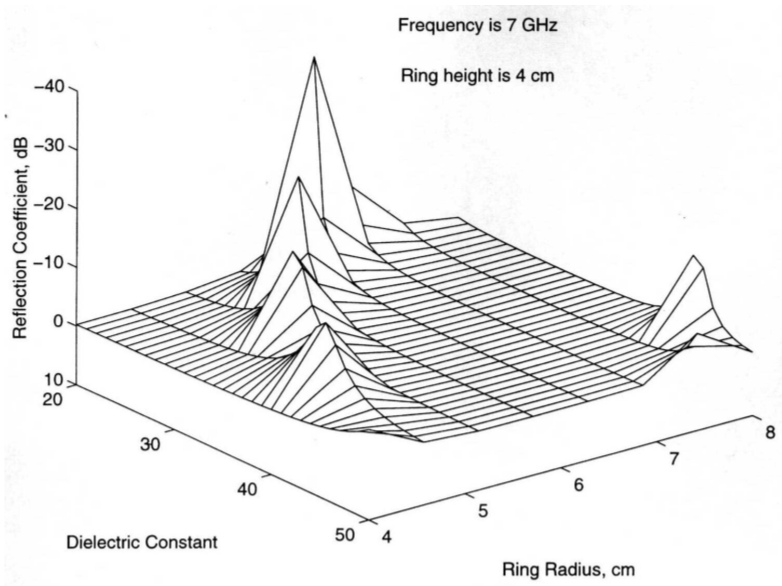
>> NN modeling of dielectric-ring resonator antenna EMF

Maxwell's equations: $\nabla \times \overline{H} = (\sigma + j\omega\varepsilon)\overline{E}$

$\nabla \times \overline{E} = -j\omega\mu\overline{H}$

$\nabla \times \nabla \times \overline{H} = -j\omega\mu(\sigma + j\omega\varepsilon)\overline{H}$

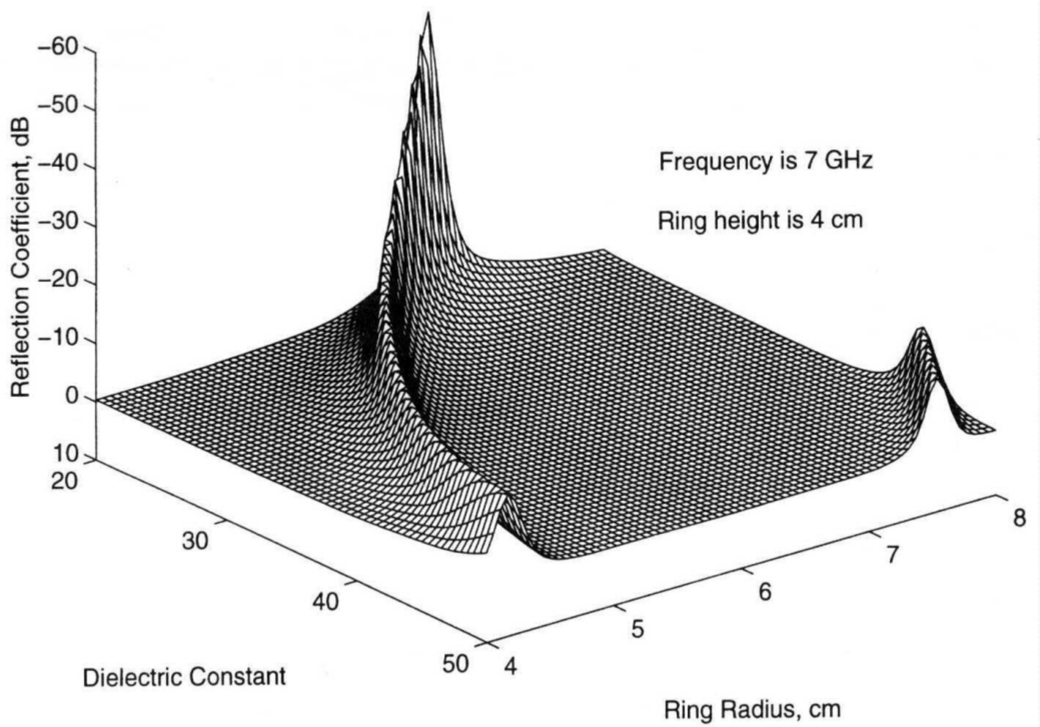
Finite Element Method (*FEM*)
1400 frequency steps 2-16 GHz;
31 dielectric constants; a = d = 5.14 mm



FEM numerical Solution => **1.3×10^5 s** on SPARC 10 UNIX

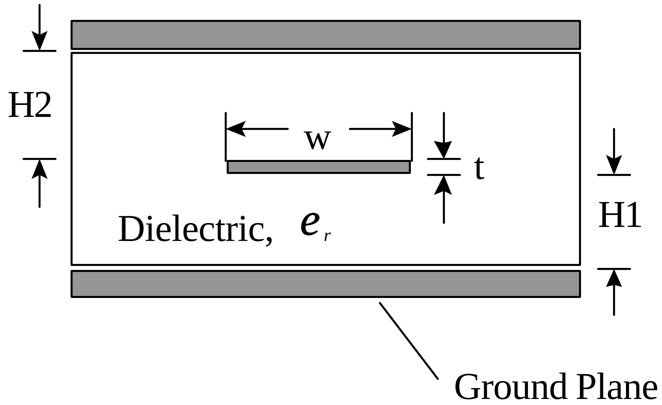
NEURAL NETWORK

- Two input neurons (frequency, dielectric constant) + Two hidden layers (5 neurons each, with hyperbolic tangent activation function) + One output linear neuron;
- Backpropagation using the Levenberg-Marquard algorithm;
- **55 s** /200 epochs to *train the NN off line* on SPARC 10 UNIX station;
- **0.5 s** to *render on line 5,000 points* of the EM field surface- model, SPARC 10 UNIX.



Modeling Single Stripline Interconnects

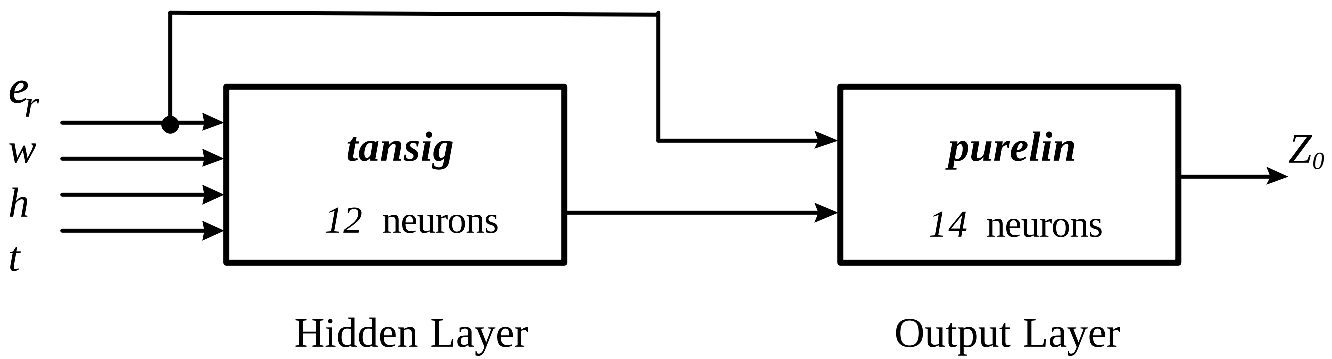
[Mao Jie, "NN Modeling of Single Stripline Interconnects," Technical Report, SMRLab, SITE, University of Ottawa, 1998

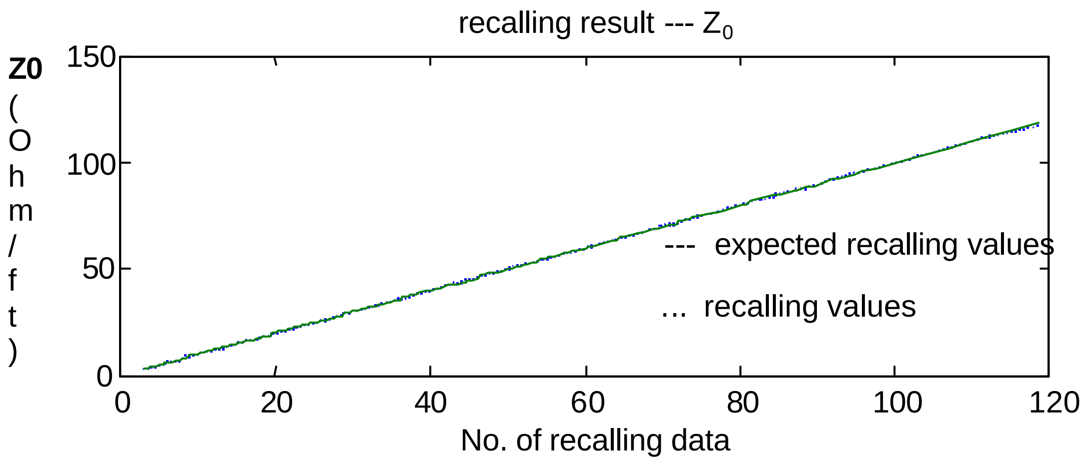


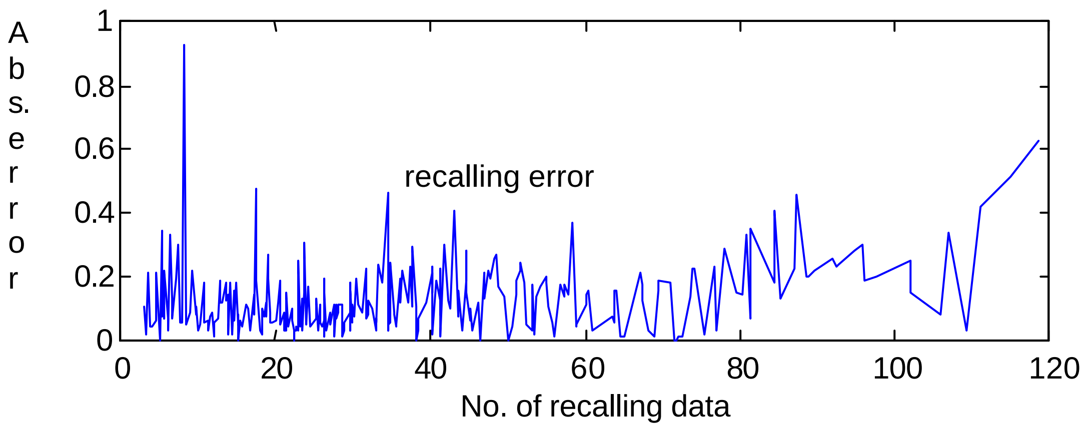
⇨ Model for $\mathbf{Z_0}$.

⇨ Model for $\mathbf{C_0}$ and $\mathbf{L_0}$.

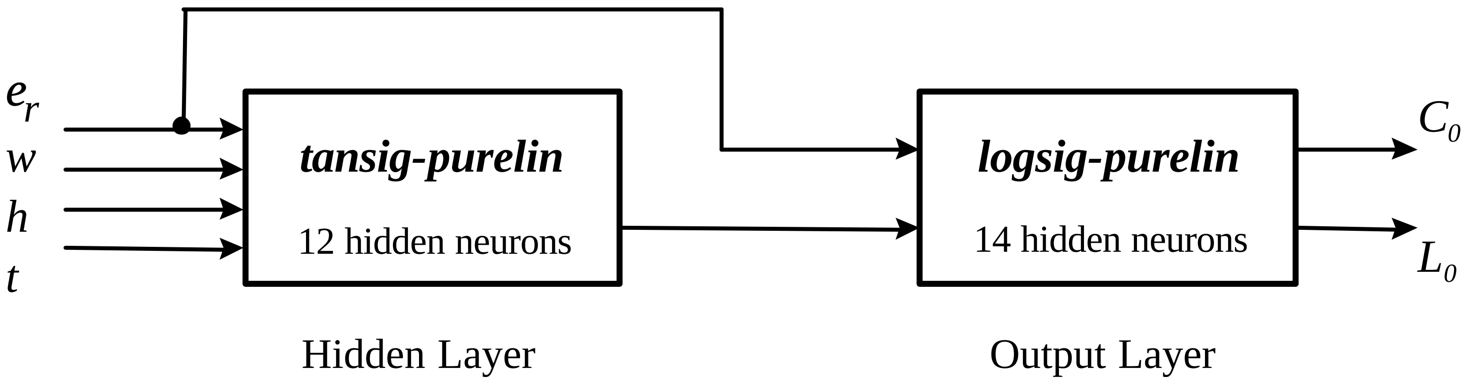
NN architecture modelling Z_0

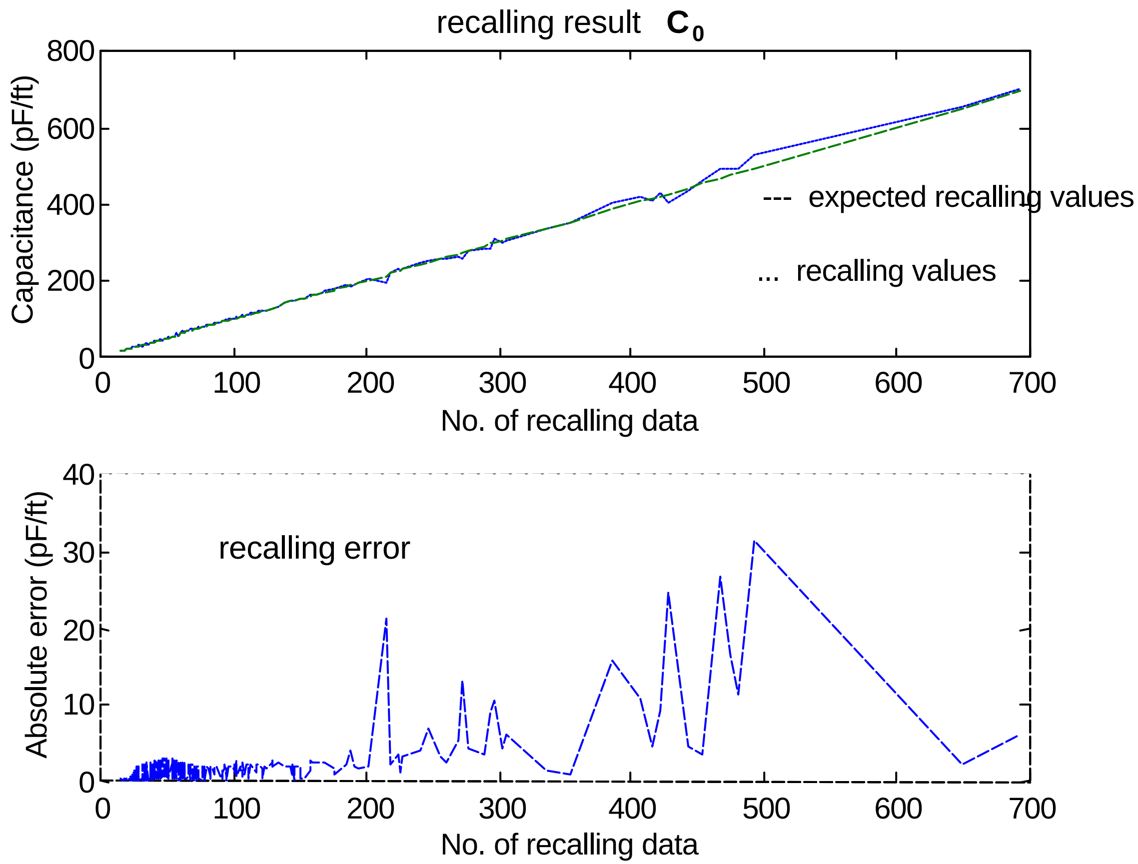



recalling result --- Z0
Z0 (Ohm/ft)
150
100
50
0
--- expected recalling values
... recalling values
0
20
40
60
80
100
120
No. of recalling data

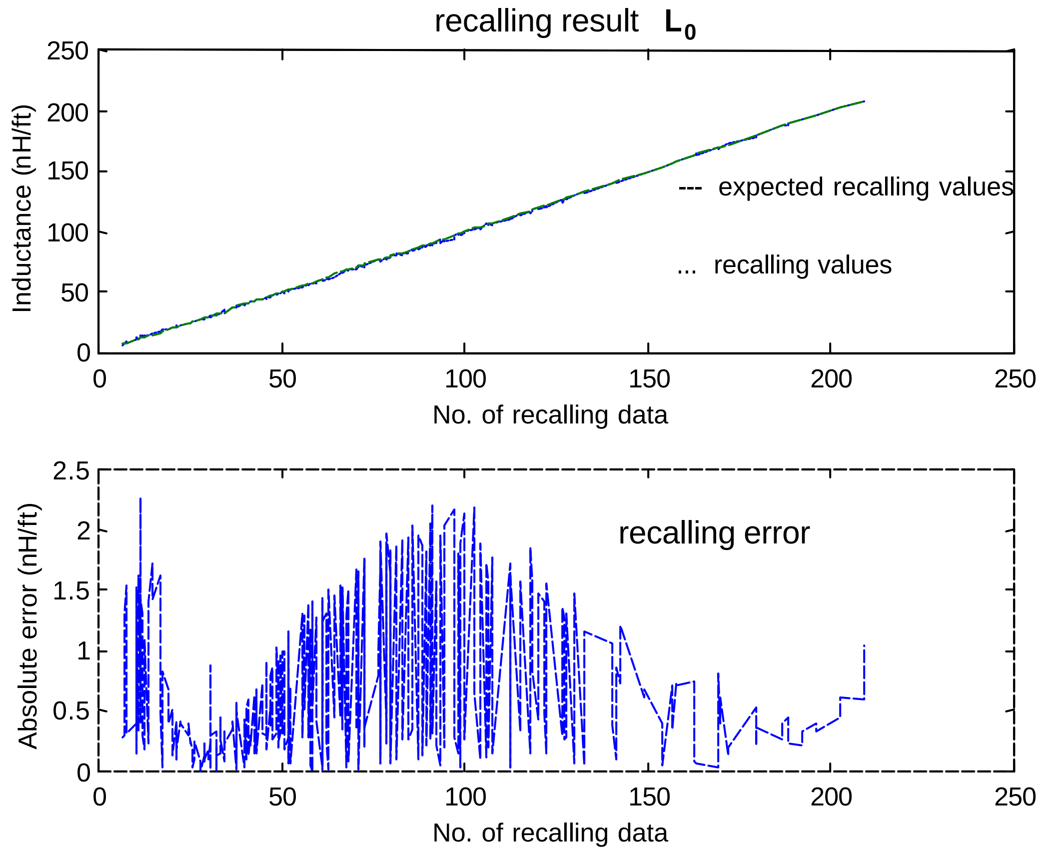

Abs. error
1
0.8
0.6
0.4
0.2
0
recalling error
0
20
40
60
80
100
120
No. of recalling data

NN architecture modelling C_0 and L_0




recalling result C_0
Capacitance (pF/ft)
800
600
400
200
0
0
100
200
300
400
500
600
700
No. of recalling data
--- expected recalling values
... recalling values
Absolute error (pF/ft)
40
30
20
10
0
recalling error
0
100
200
300
400
500
600
700
No. of recalling data

>> Modeling Single Stripline Interconnects



NEURAL NETWORK MODELLING OF PLAIN AND GROOVED MICROSTRIPS

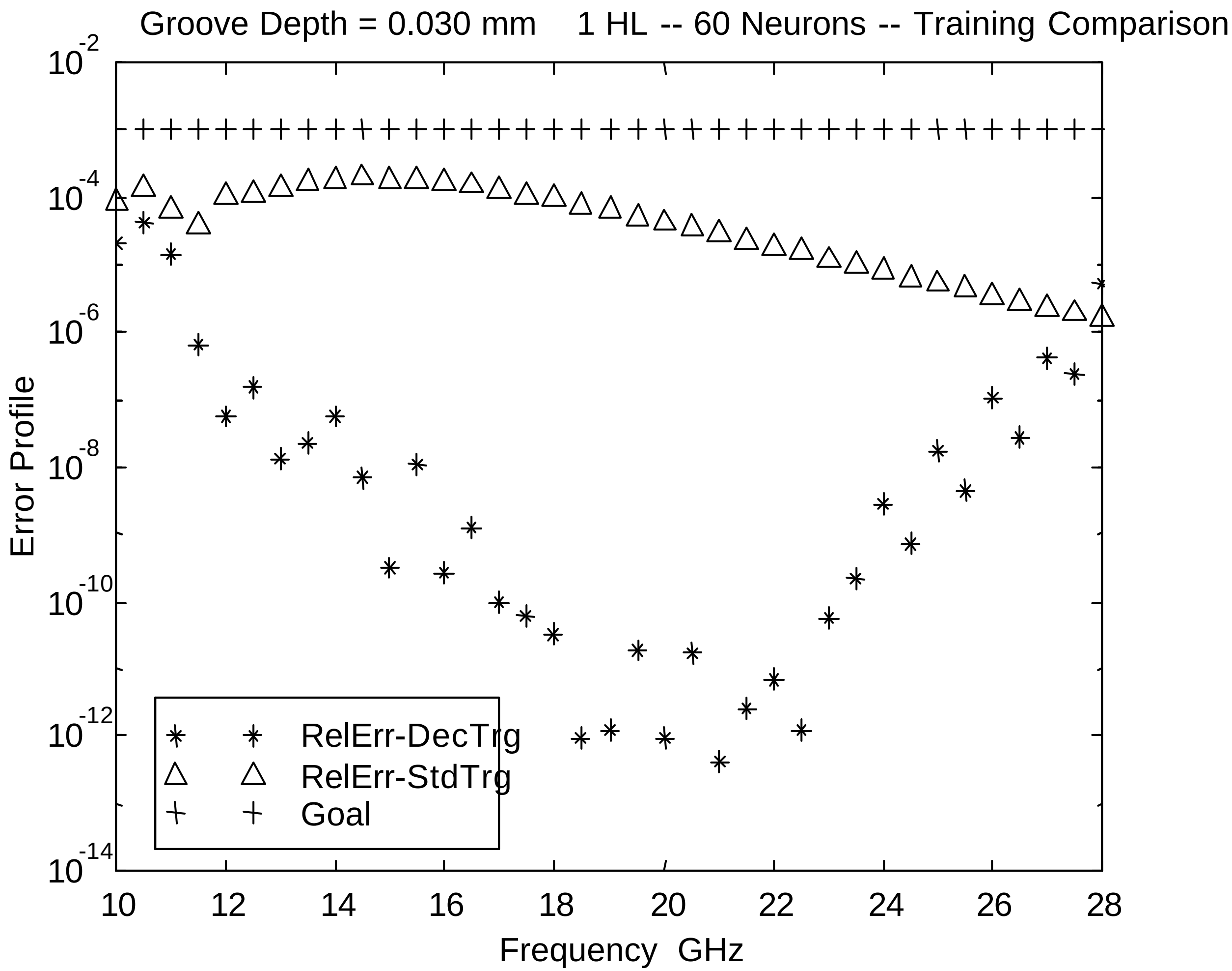
- *A. Chubukjan, "Computational Aspects in Modelling Electromagnetic Field Parameters in Microstrips, " Ph.D. Thesis, University of Ottawa, 2000*

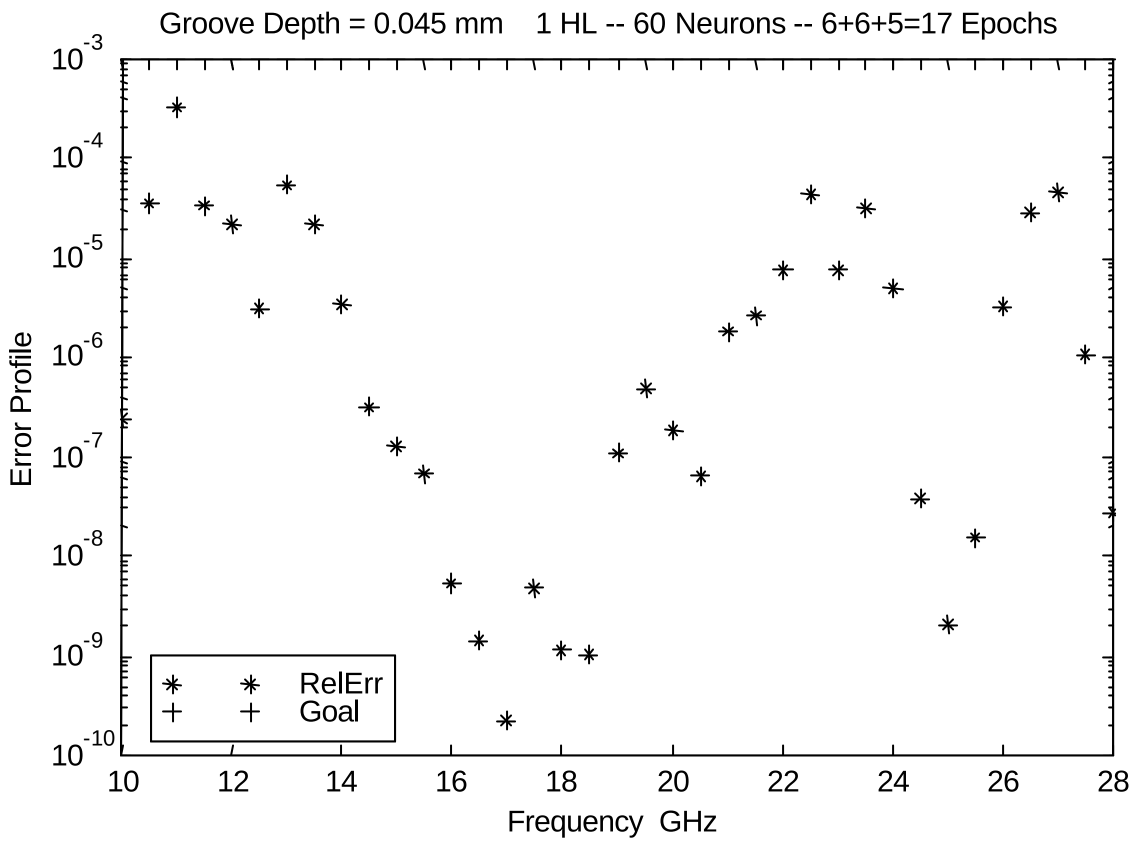
- The problem was solved by "Vector Finite Element Method" VFEM, and the values of the microstrip characteristic impedance for both plain and grooved geometries were obtained. These values describing both the frequency-dependent and/or groove-dependent behaviour of each microstrip geometry were used to train the NN models.

- A feedforward network with backpropagation, having one or more hidden layers with non-linear transfer functions and one output layer with a linear transfer function, is capable of approximating any function with a finite number of discontinuities with arbitrary accuracy. A two-layer sigmoid/linear NN can represent any functional relationship between inputs and outputs if the sigmoid layer has enough neurons.

❖ The grooved microstrip was modelled initially by two separate one hidden-layer NN architectures, having 50 and 60 hidden neurons respectively. These networks were trained both by decimation and by the standard way. The resulting error obtained by decimation was comparable to that obtained by standard training, and at times, was superior. The networks reached the desired error goal easily, with excellent sum-squared error figures. Nevertheless, the NN architecture with 60 neuron hidden-layer gave better results compared to the 50 neuron hidden-layer architecture, and it was selected for further modelling.

Error performance for standard and decimated training of a “60 neuron one hidden-layer” NN model of grooved microstrip.





Error performance for standard and decimated training of a "60 neuron one hidden-layer" NN model of grooved microstrip.

MODEL CALIBRATION

The whole idea of virtual prototyping relies on the ability to develop ***models conformable to the physical objects and phenomena*** which represent reality very closely.

There is a need for ***calibration techniques able to validate the conformance with the physical reality of the models*** incorporated in the new prototyping tools.

Experimental Measurements

- The EM field training data are conveniently obtained as analytical estimations of far-field values in 3D space and frequency from near-field data using the finite element method combined with method of integral absorbing boundary conditions.
- The near field data could be obtained analytically and/or by physically measuring EM field values at for given frequency values and 3D space locations.
- This approach allows to replace the usual cumbersome open site far-field measurement technique by anechoic chamber measurements.

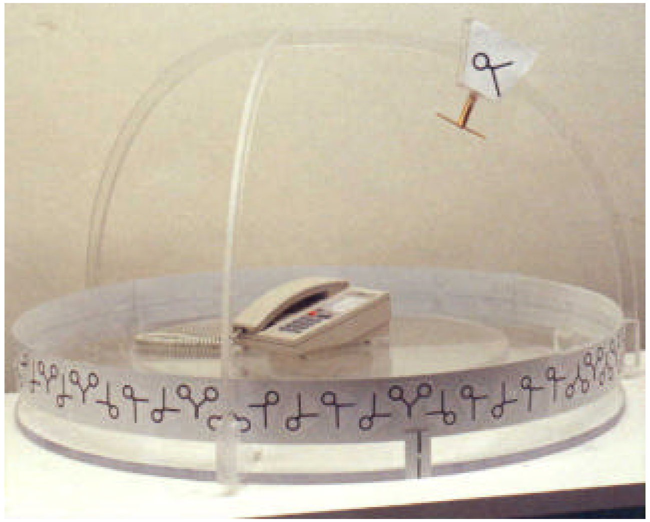
The amount and extent of the area of measurements is significantly reduced by collecting data in the near-field only and calculating then the far-field values using Poggio's equation:

$$H\left(r'\right) = \frac{1}{4\pi}\int_{s1}\left[G\left(r, r'\right)\frac{\partial H(r)}{\partial n} - H(r)\frac{\partial G\left(r, r'\right)}{\partial n}\right] dS_1$$

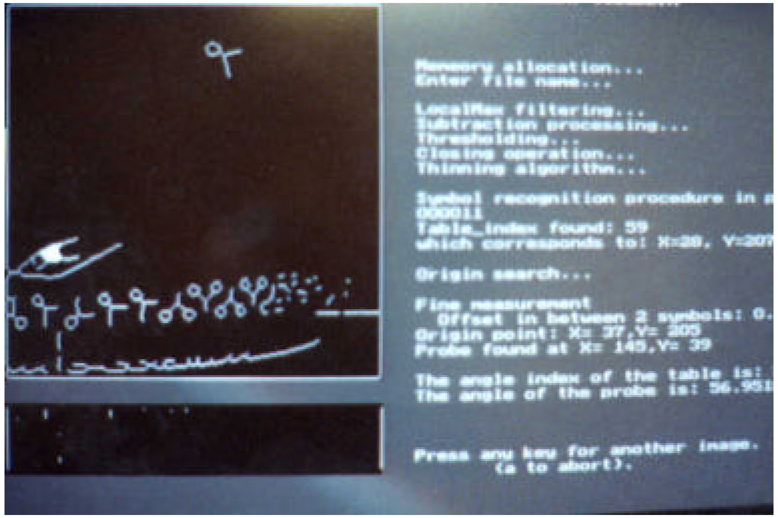
where:

- S_1 is the surface on which measurements are made, closed or made closed,
- n is the normal to S_1 and is the free space Green’s function.

- *This equation states that if the field values and their derivatives are known on a closed surface enclosing all inhomogeneities, then the field outside the surface can be calculated.*



Experimental setup for the noninvasive measurement of the 3D near field data



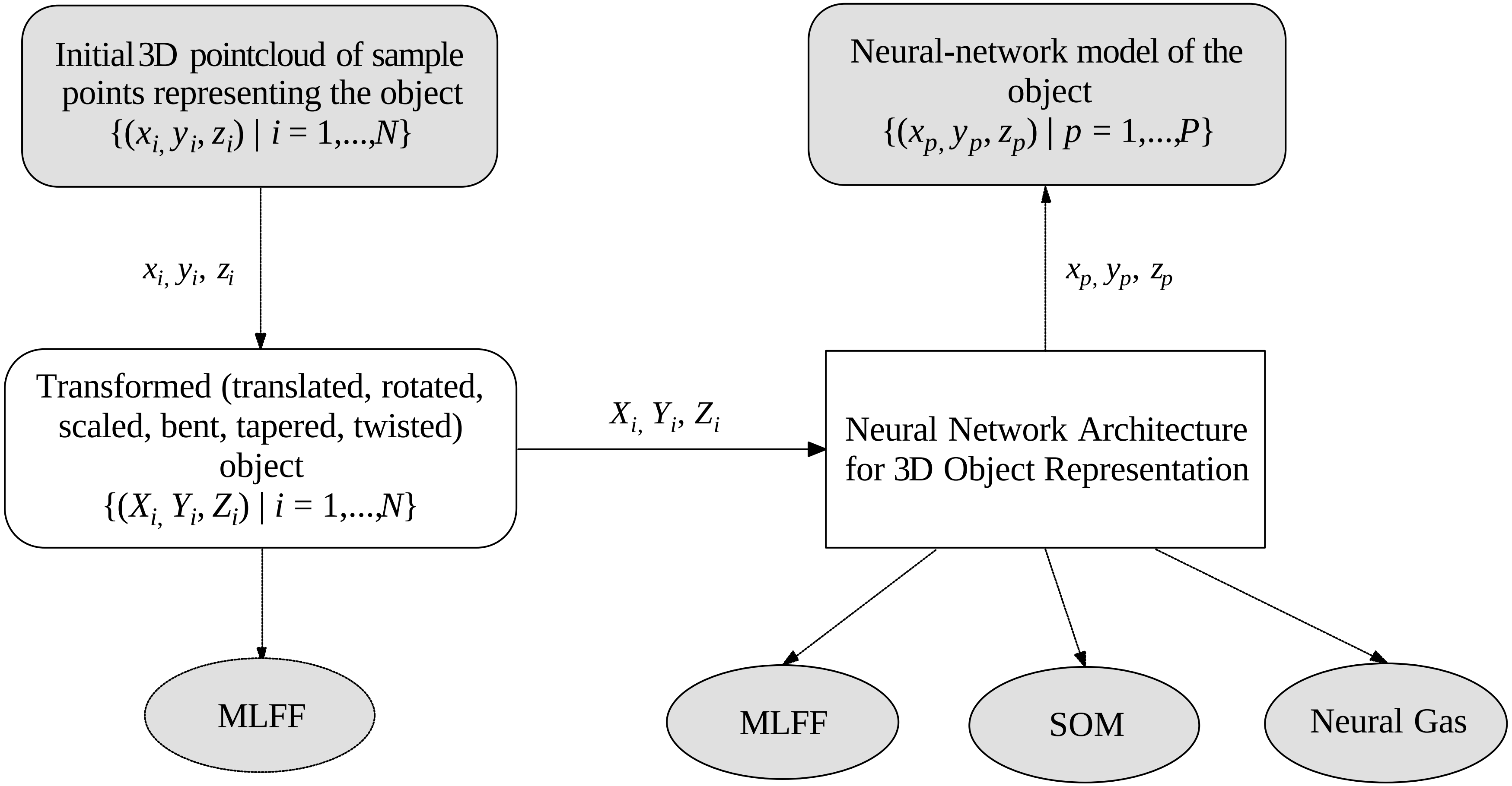
Computer vision recovery of the 3D position of the EM probe

Neural Network Modeling of 3D Objects

A-M. Cretu, *E.M. Petriu*, G.G. Patry, " A Comparison of Neural Networks Architectures for Geometric Modelling of 3D Objects," *Proc. CIMSA'04 IEEE Intl. Conference on Computational Intelligence for Measurement Systems and Applications*, pp. 155-160, Boston, MA, USA, July 2004

Compare the performance of three NN architectures used for 3D Object modelling:

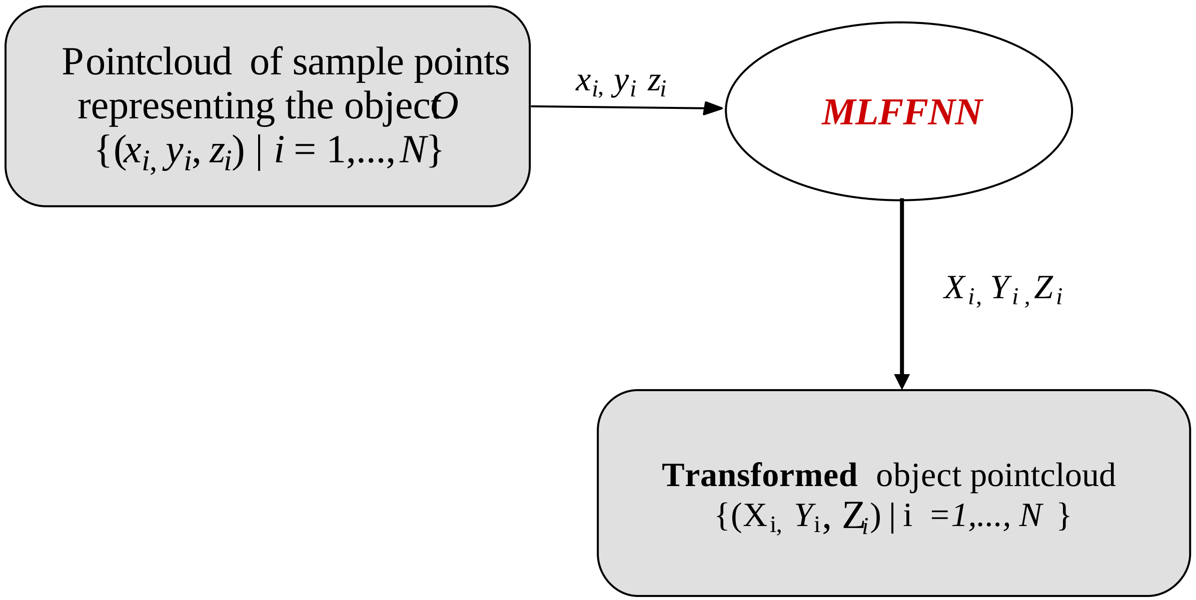
- Multilayer Feedforward Neural Network (*MLFFNN*)
- Self-Organizing Map (*SOM*)
- Neural Gas Network



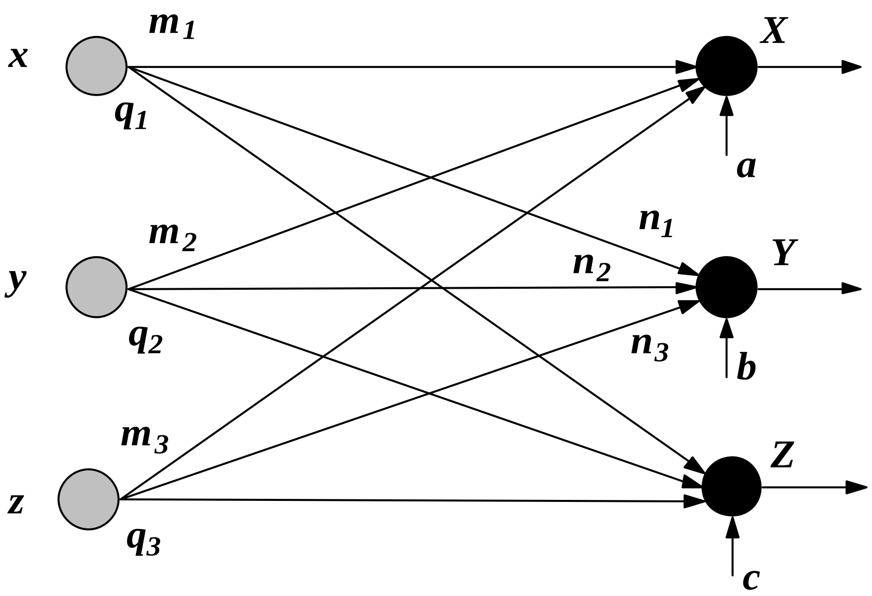
Transformation Function:

translation, rotation, scaling,

and deformations (bending, tapering, twisting)



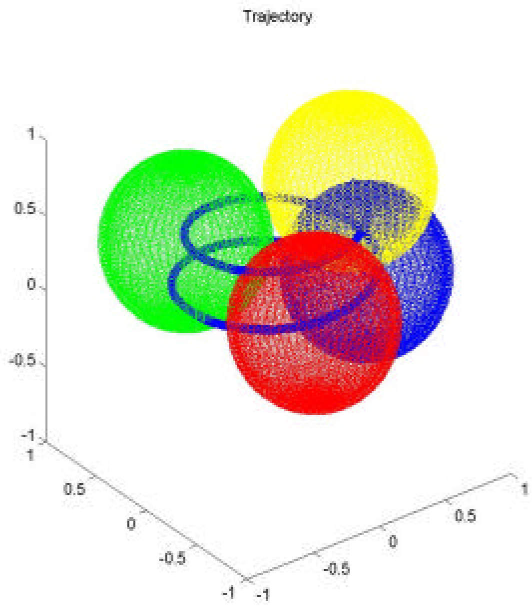
Transformation Function – NN Architecture

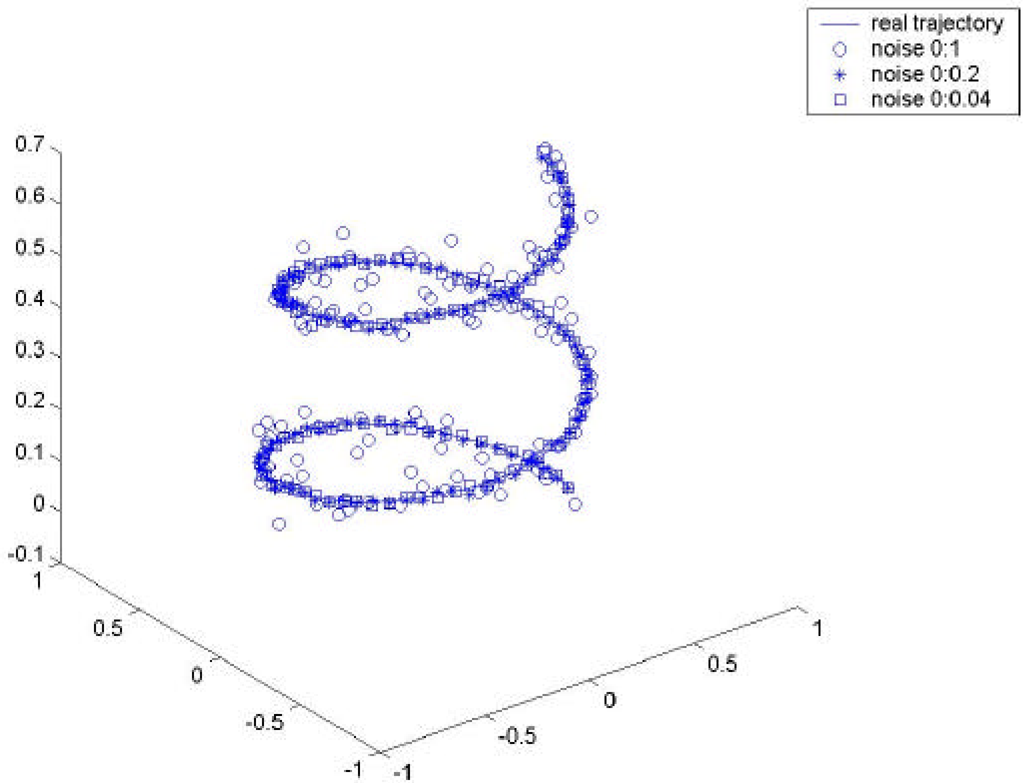


$$m_1 = a_1 \cos q \cos j$$
$$m_2 = a_1(\cos q \sin j \sin y - \sin q \cos y)$$
$$m_3 = a_1(\cos q \sin j \cos y + \sin q \sin y)$$
$$n_1 = a_2 \sin q \cos j$$
$$n_2 = a_2(\sin q \sin j \sin y - \cos q \cos y)$$
$$n_3 = a_2(\sin q \sin j \cos y - \cos q \sin y)$$
$$q_1 = -a_3 \sin j$$
$$q_2 = a_3 \cos j \sin y$$
$$q_3 = a_3 \cos j \cos y$$

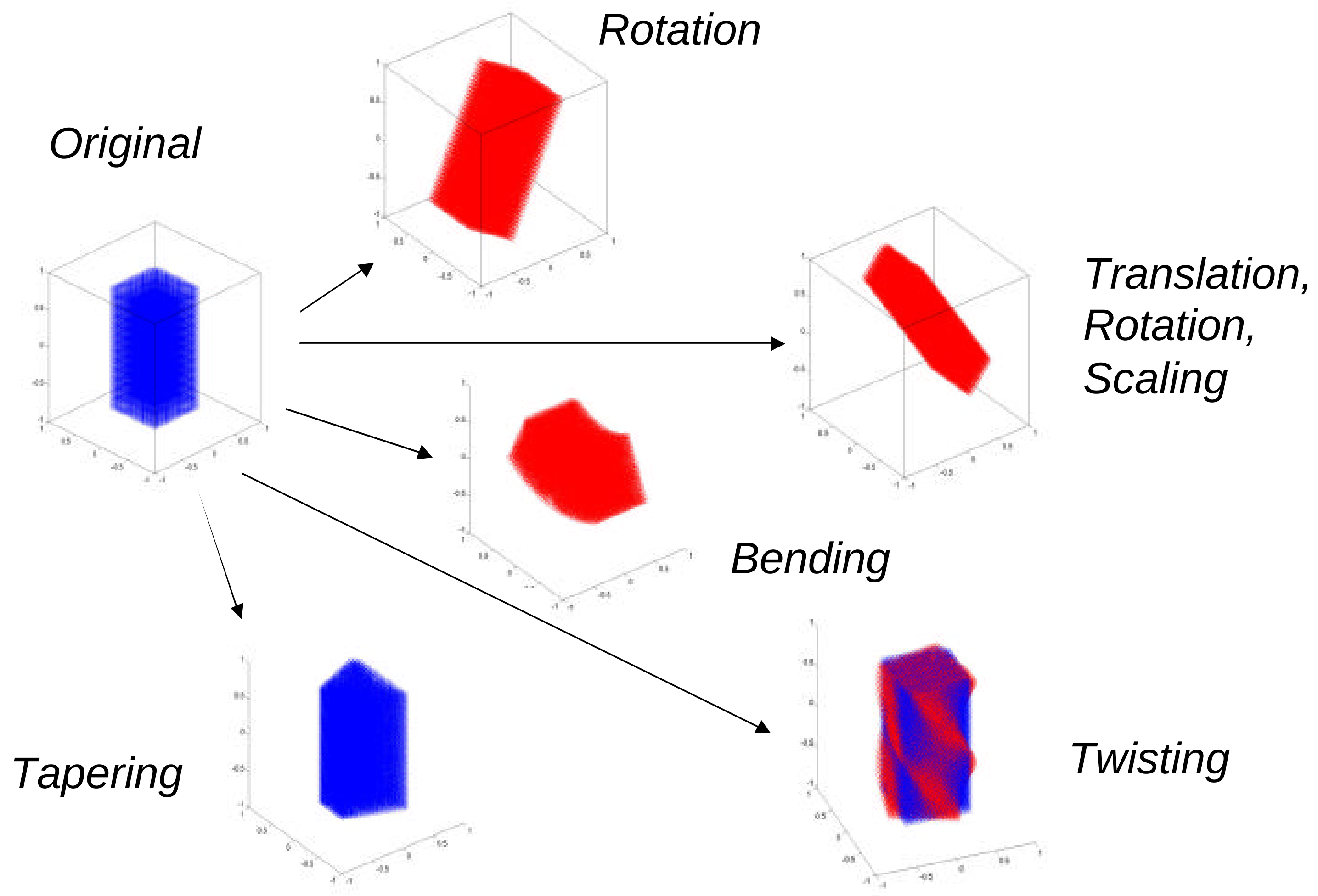
Transformation Function - Training Mode

Motion Estimation



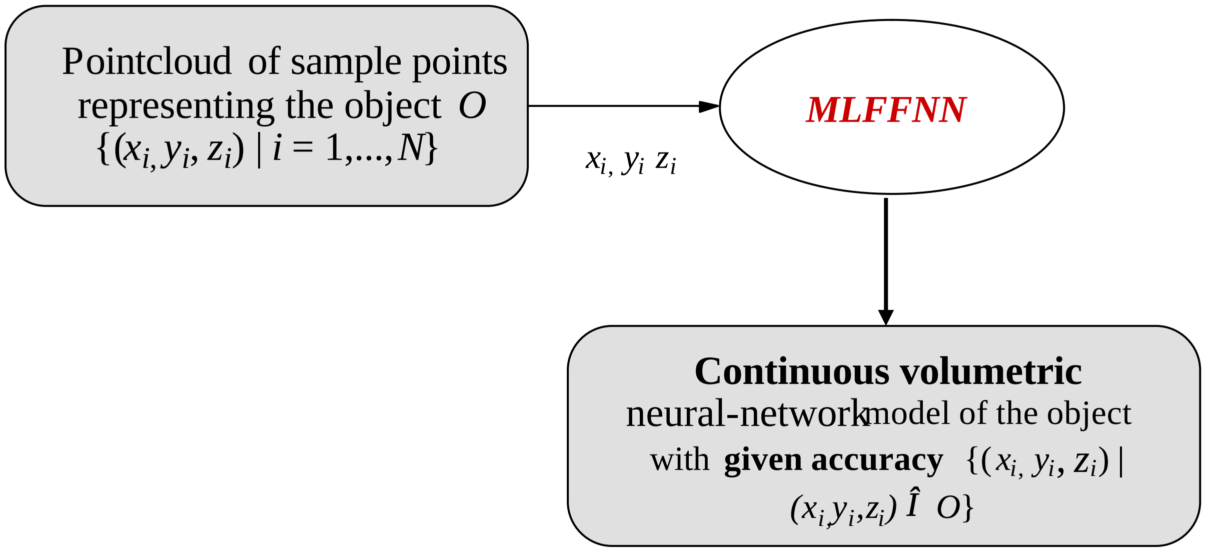


Transformation Function - Generation Mode

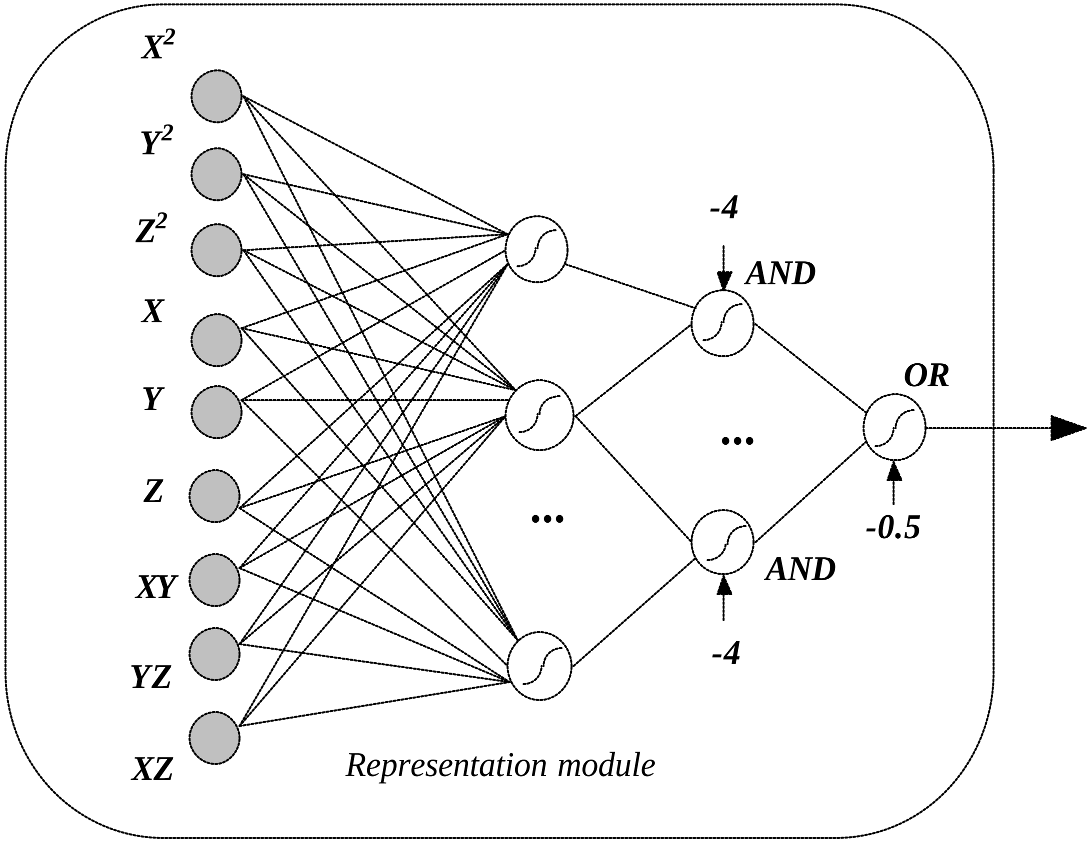


MLFF Representation

generates a value proportional to the distance between an input point and the modeled object surface

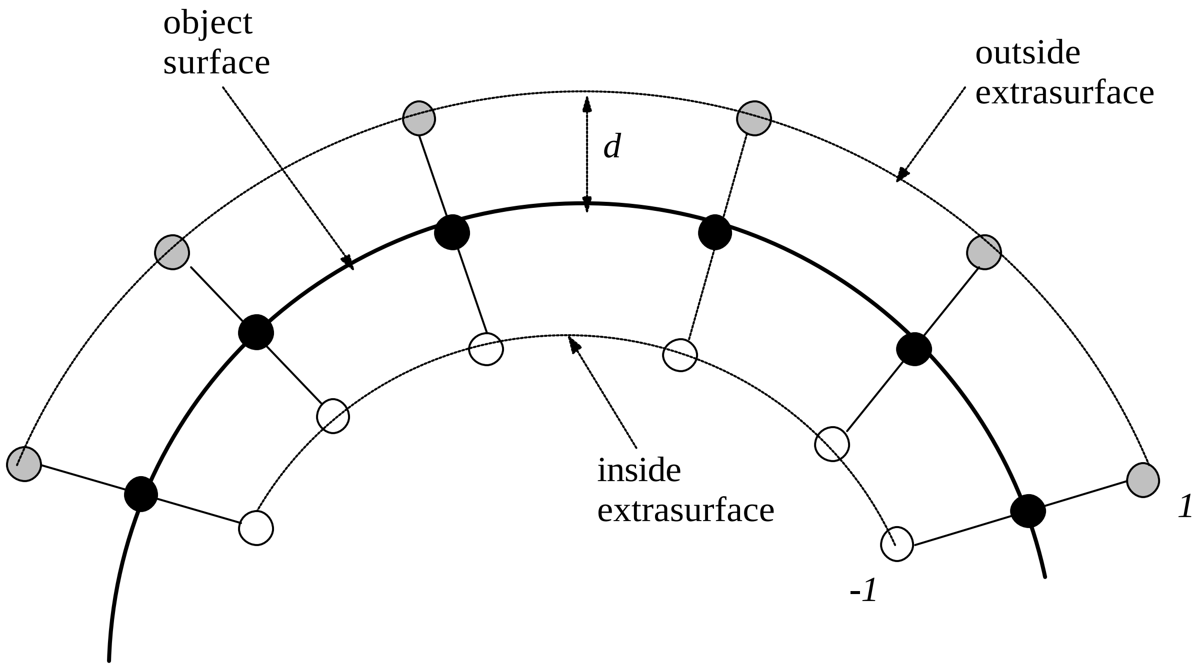


MLFFNN Representation – NN Architecure



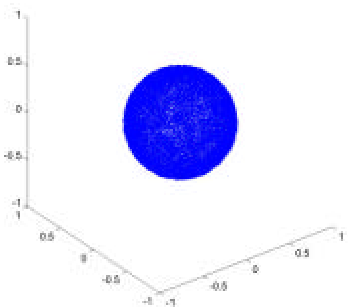
- **Activation Function**
  - sigmoid
- **Training/Testing Data**
  - normalized points in the [-1 1 –1 1 –1 1] cube
- **Learning**
  - supervised
  - scaled-gradient descent backpropagation

MLFFNN Representation - Training Mode

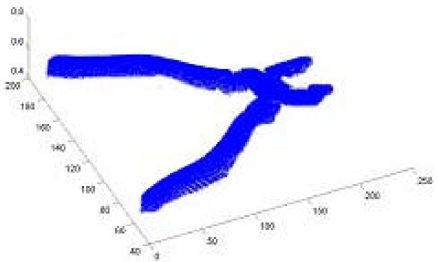


- Models objects given as pointclouds
- **Decisions**:
  - inputs to use
  - number of neurons in hidden layer
  - values for training parameters
  - number of extrasurfaces and distance

MLFFNN Modelling - Results

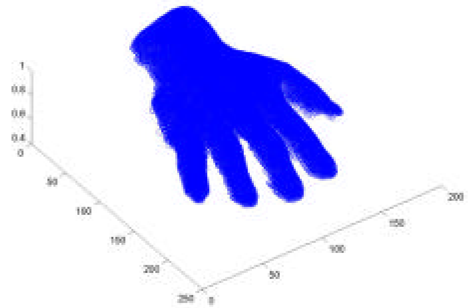


250 points, 6-3-1, 1 extrasurface, d=0.055, 550 epochs, mse: 0.14, 7 min.

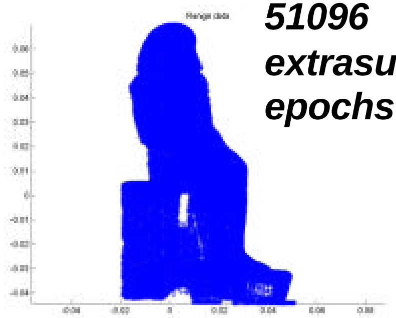


7440 points, 8-4-1, 5 extrasurfaces, d=0.055, 1100 epochs, mse: 0.24, 1 hr

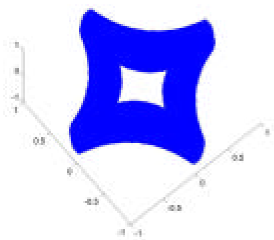
19080 points, 10-5-1, 5 extrasurfaces, d=0.055, 1200 epochs, mse: 0.35, 2.8 hrs.



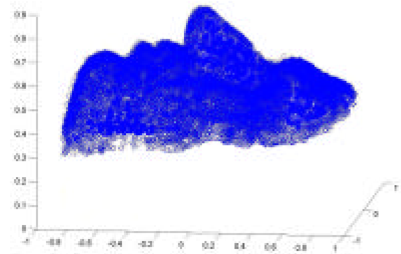
19000 points, 14-7-1, 4 extrasurfaces, d=0.055, 1100 epochs, mse: 0.4, 3.3 hrs



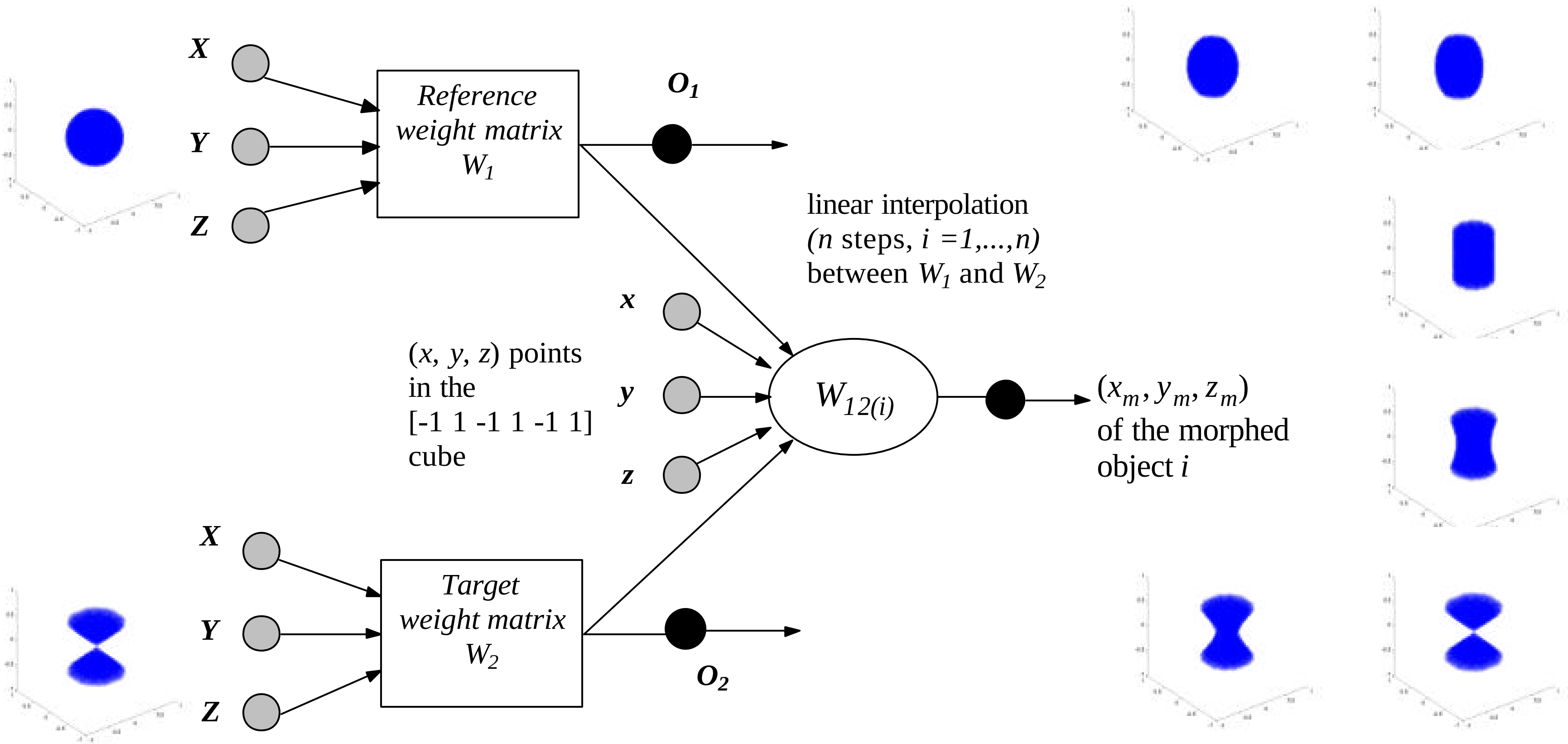
51096 points, 20-10-1, 5 extrasurfaces, d=0.055, 2000 epochs, mse: 0.67, 5.2 hrs.



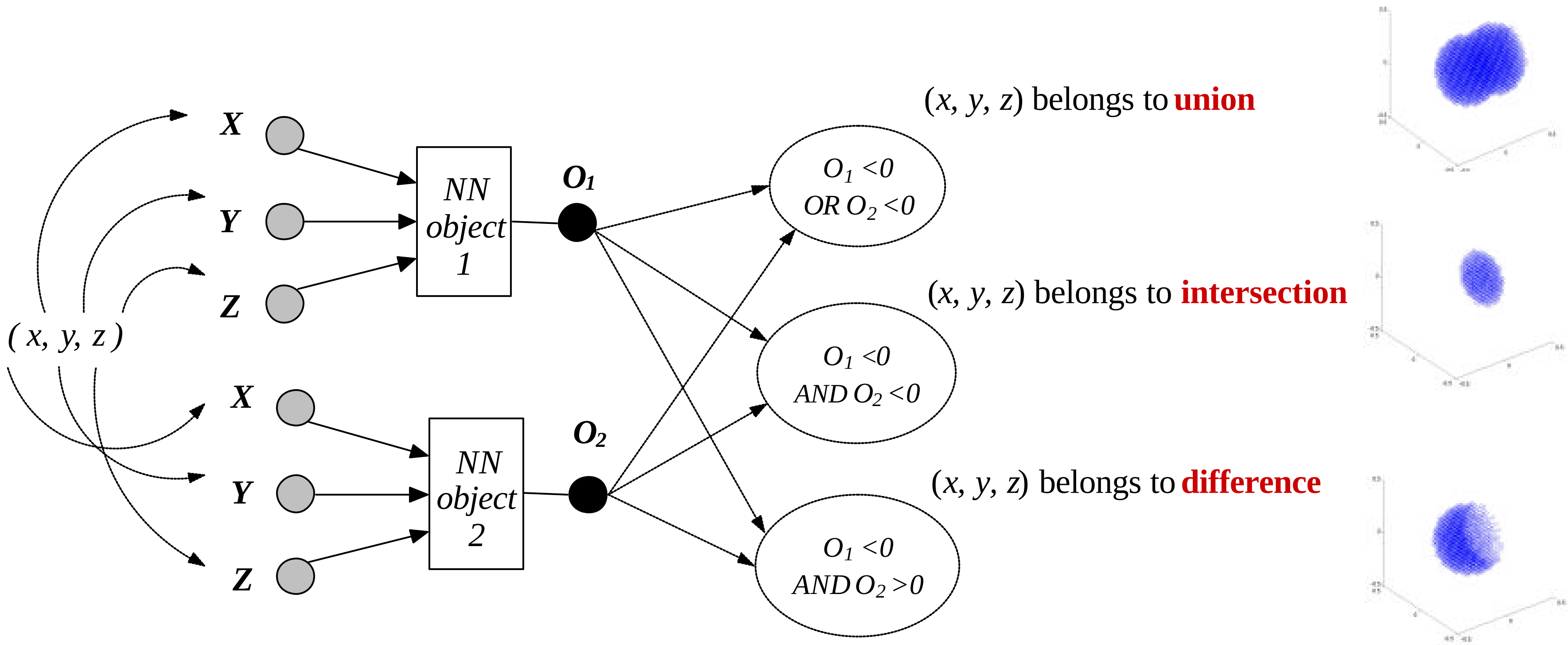
2500 points, 12-6-1, 2 extrasurfaces, d=0.06, 1020 epochs, mse: 0.39, 45 min.



MLFFNN Representation – Applications ⇨ *Object Morphing*

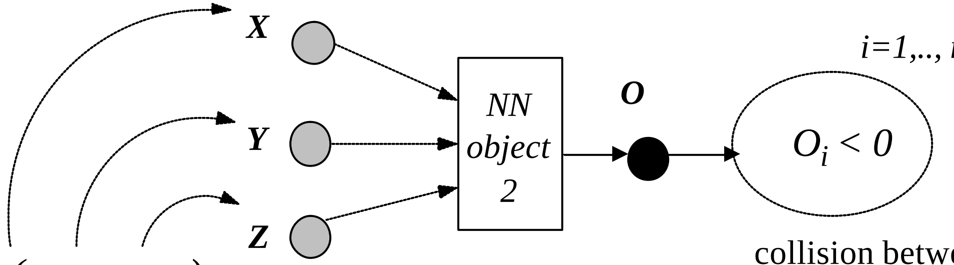


MLFFNN Representation – Applications ⇨ *Set Operations*



MLFFNN Representation – Applications

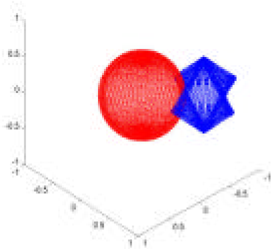
⇨ *Object Collision Detection*



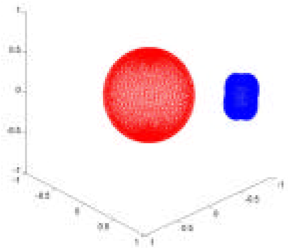
(x_1, y_1, z_1)
(x_2, y_2, z_2)
...
(x_n, y_n, z_n)

sampled points
object 1

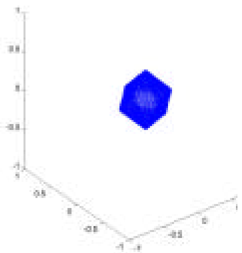
collision between
object 1 and
object 2



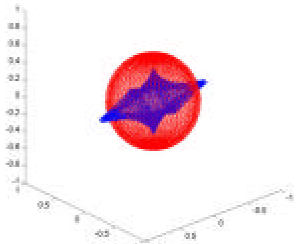
96.56%



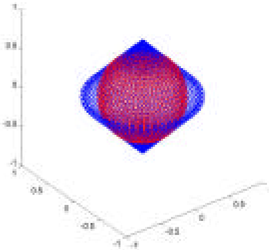
0%



100%

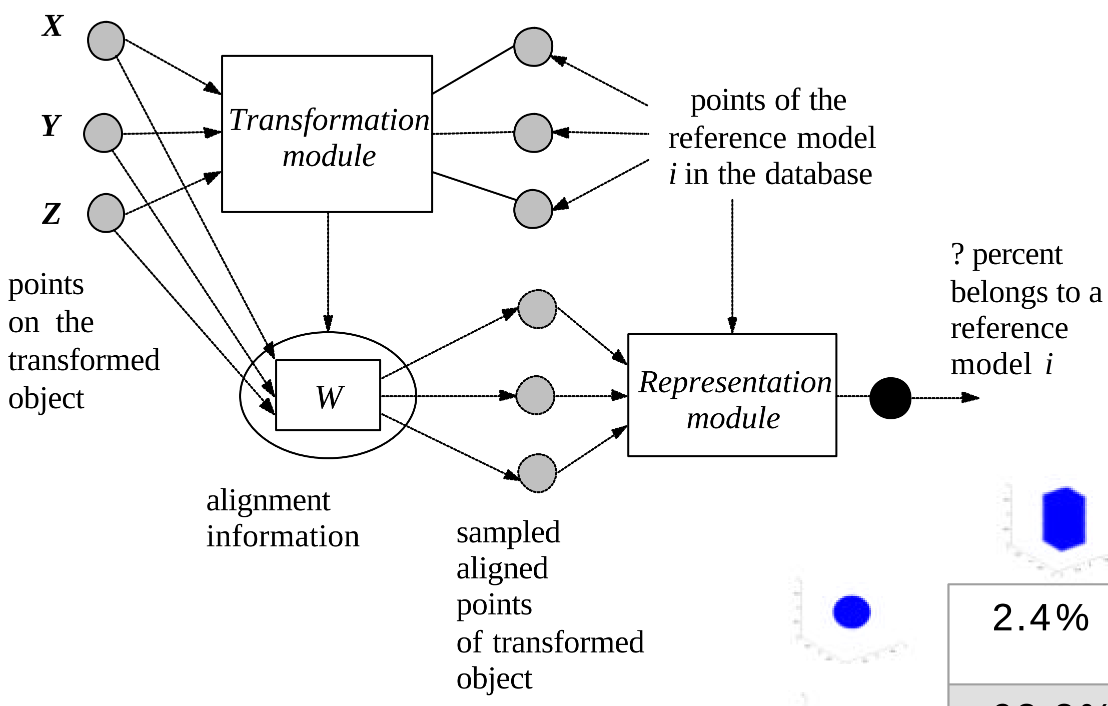


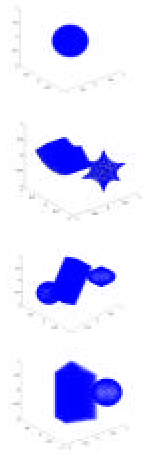
2.3%



97
%

MLFFNN Representation – Applications ⇨ *Object Recognition*





2.4%	4.9%	1.6%	99.1%
93.3%	3.3%	91.7%	3.1%
95%	99.1%	5.78%	98.3%
91.7%	6.6%	1.6%	92.5%

MLFFNN Modelling – Summary

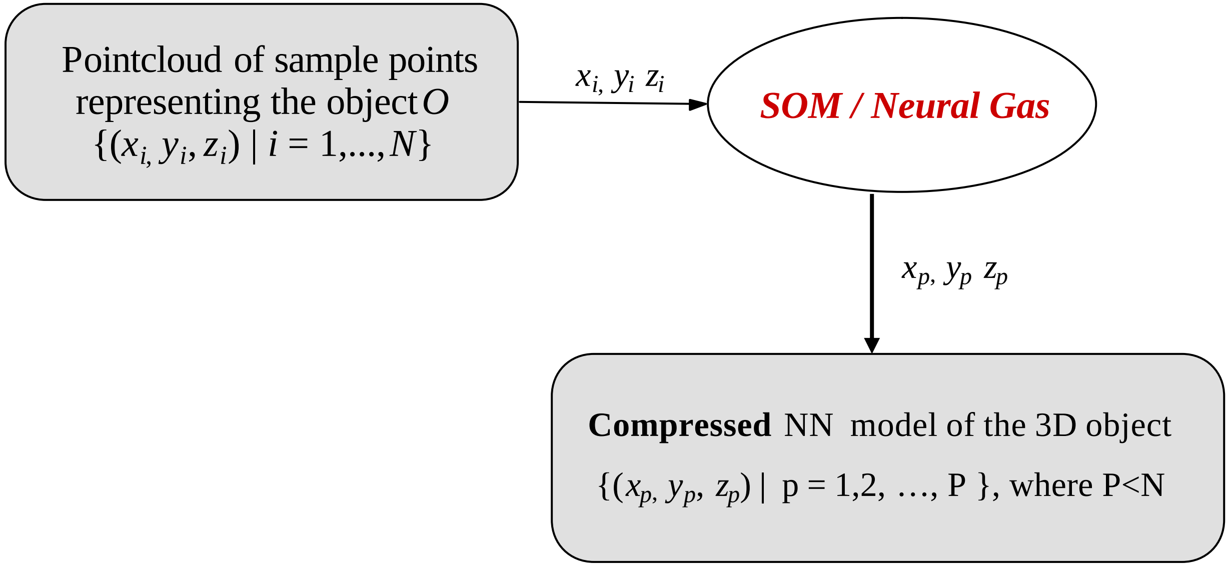
Advantages

- simple and compact (weights+architecture)
- less memory usage
- 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 (convenience)

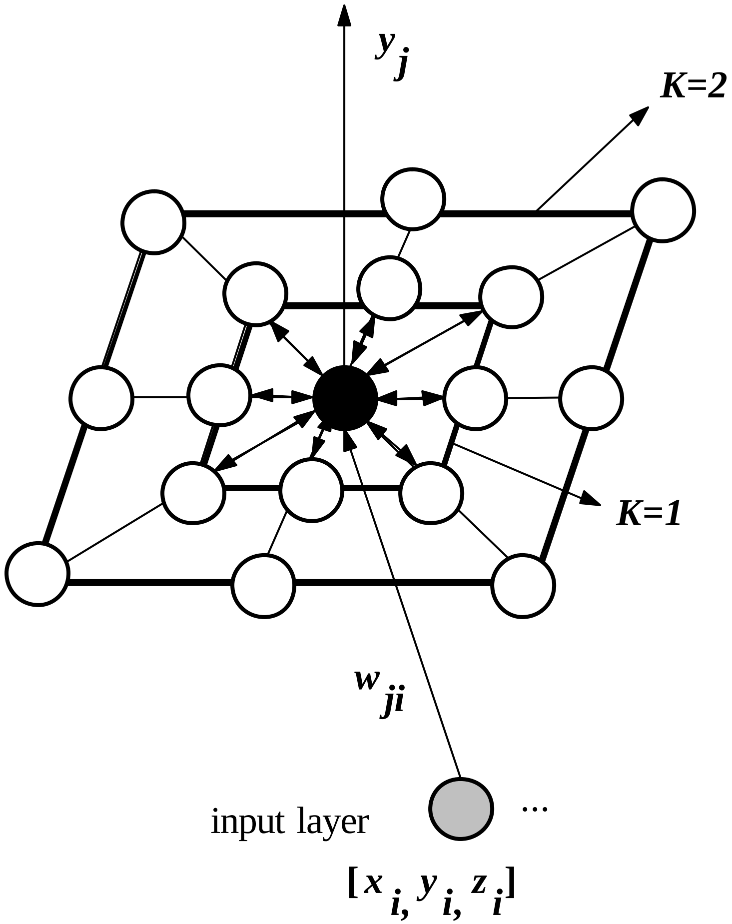
Disadvantages

- computationally expensive (for both learning and rendering)
- lack of local control of the object

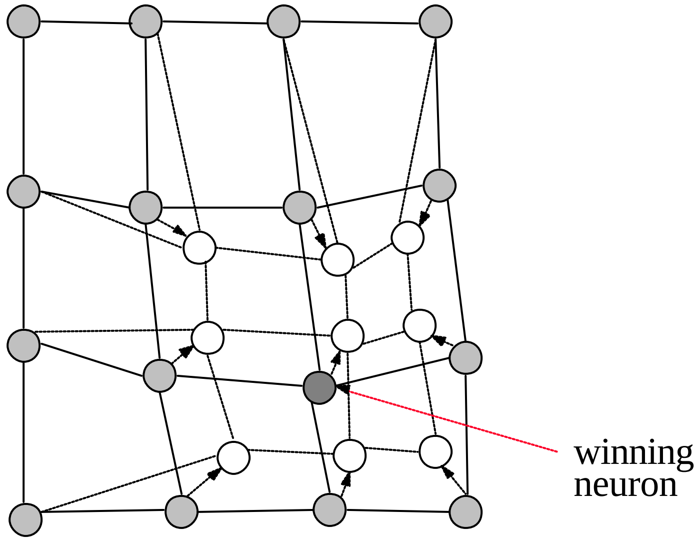
SOM and Neural Gas
- Compressed Representation Models



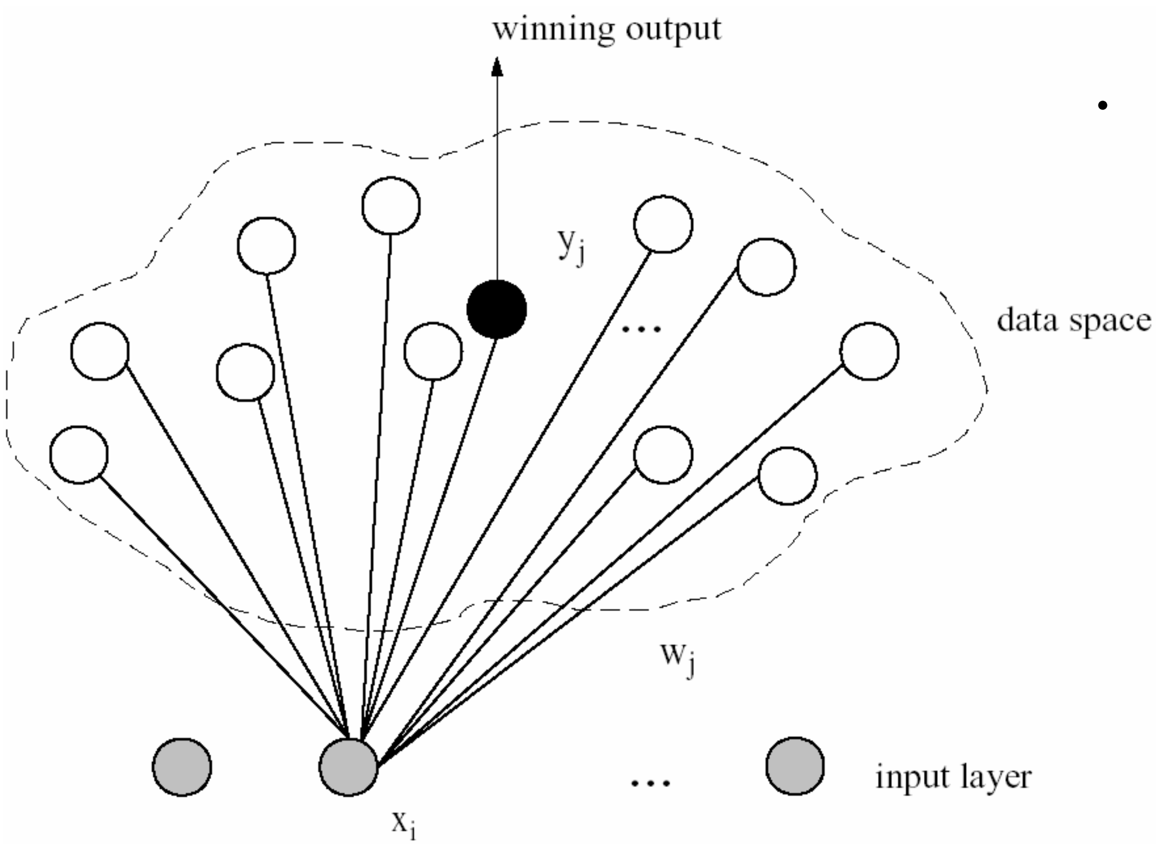
SOM Representation – NN Architecture



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



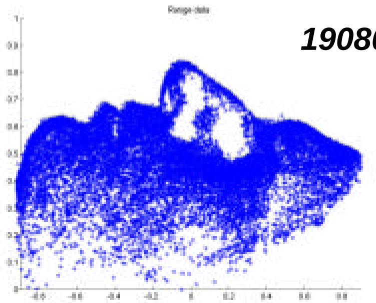
Neural Gas Representation – NN Architecture



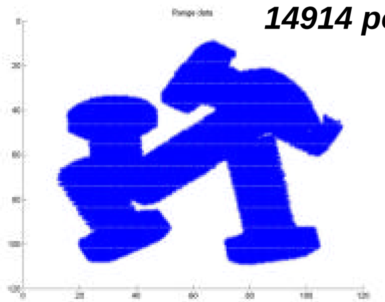
- **Activation Functions**:
  - soft competition
  - neighbourhood ranking
- **Learning**
  - unsupervised

SOM and Neural Gas Modelling - Results

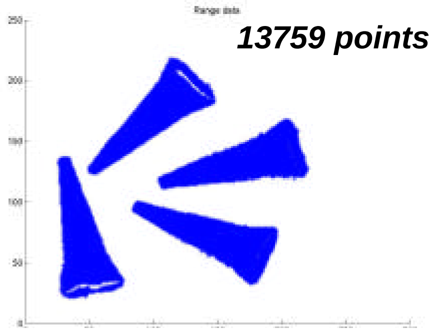
Initial pointcloud



19080 points

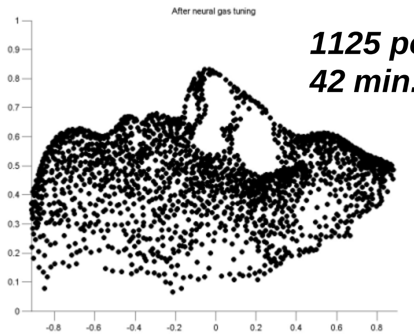


14914 points



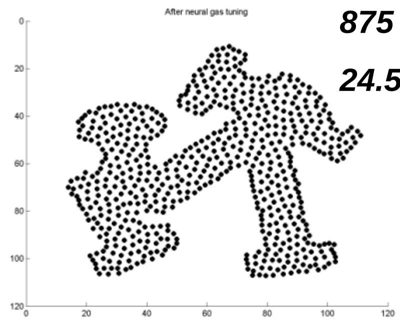
13759 points

Neural Gas



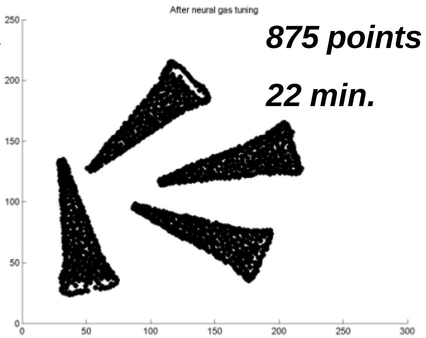
1125 points, 42 min.

er= 0.0098



875 points,

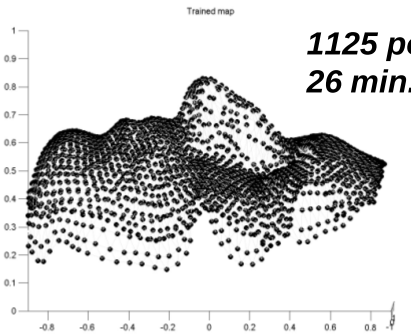
24.5 min.



875 points

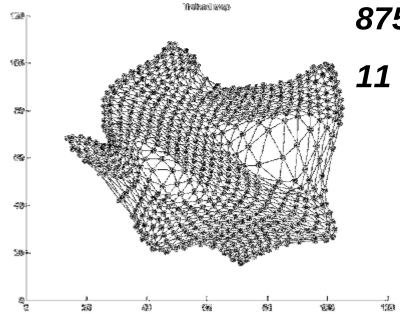
22 min.

SOM



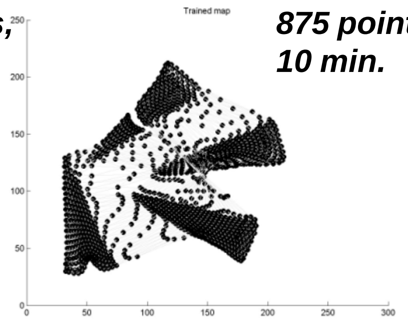
1125 points, 26 min.

er= 0.0125



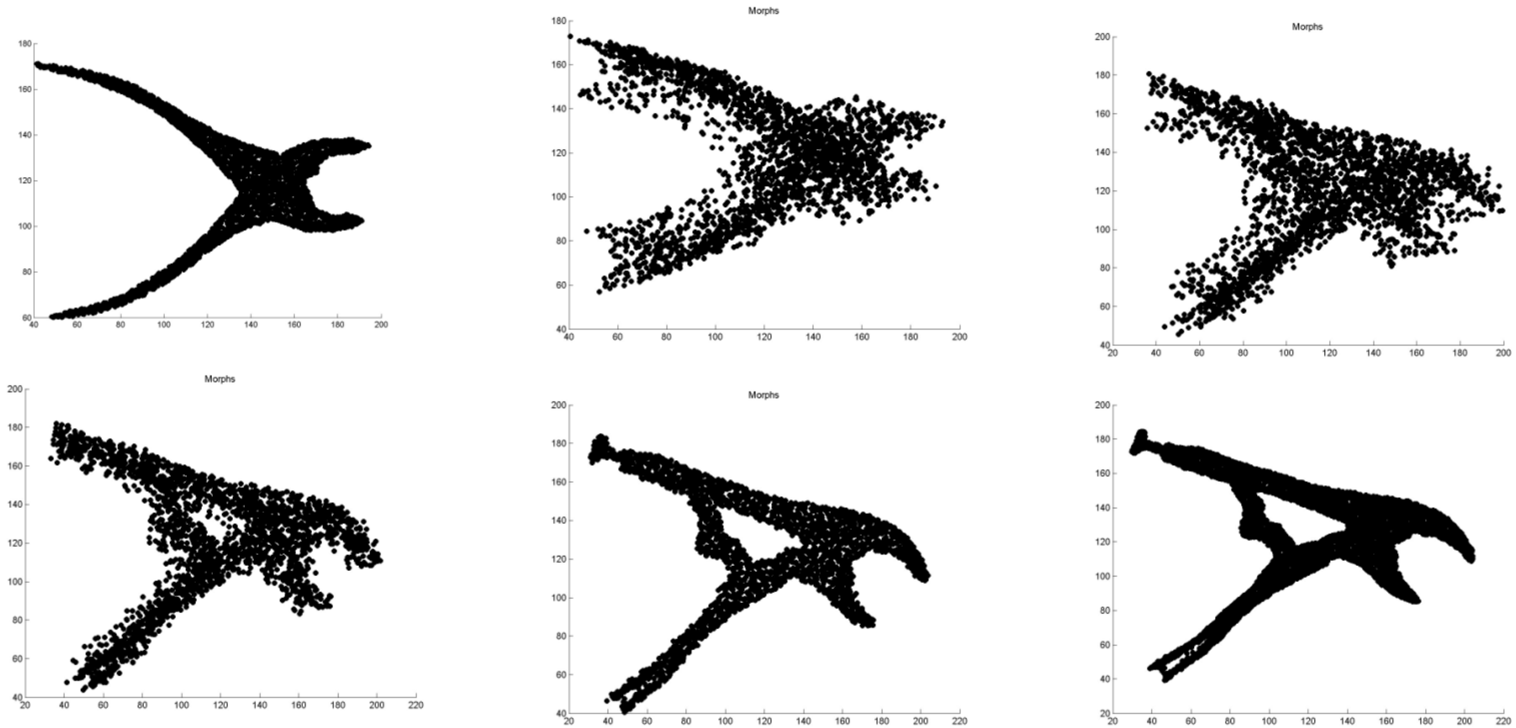
875 points,

11 min.

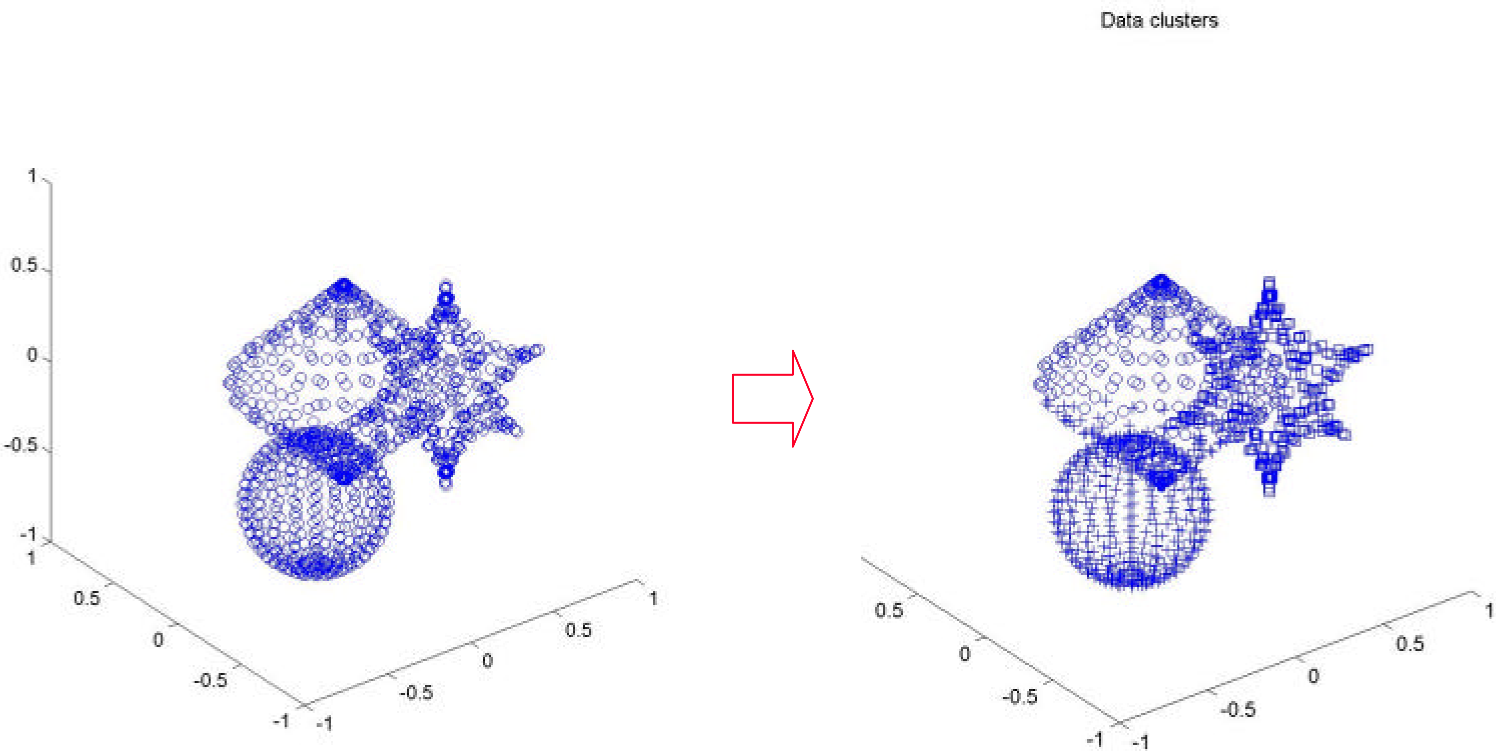


875 points, 10 min.

SOM and Neural Gas Modelling – Applications ⇒ *Object Morphing*



SOM and Neural Gas Modelling – Applications ➡ *Segmentation*



SOM and Neural Gas Modelling – Summary

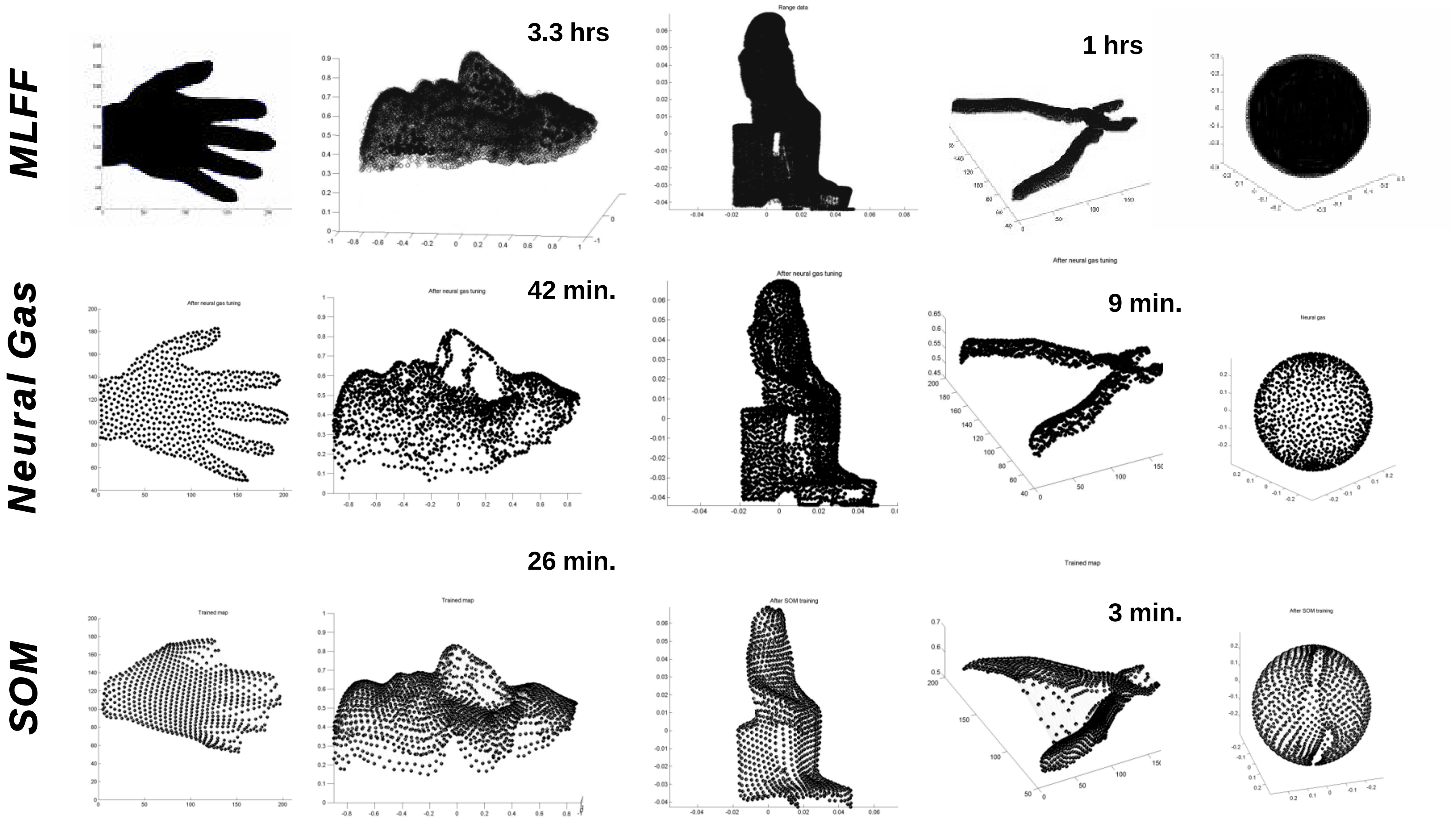
Advantages

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

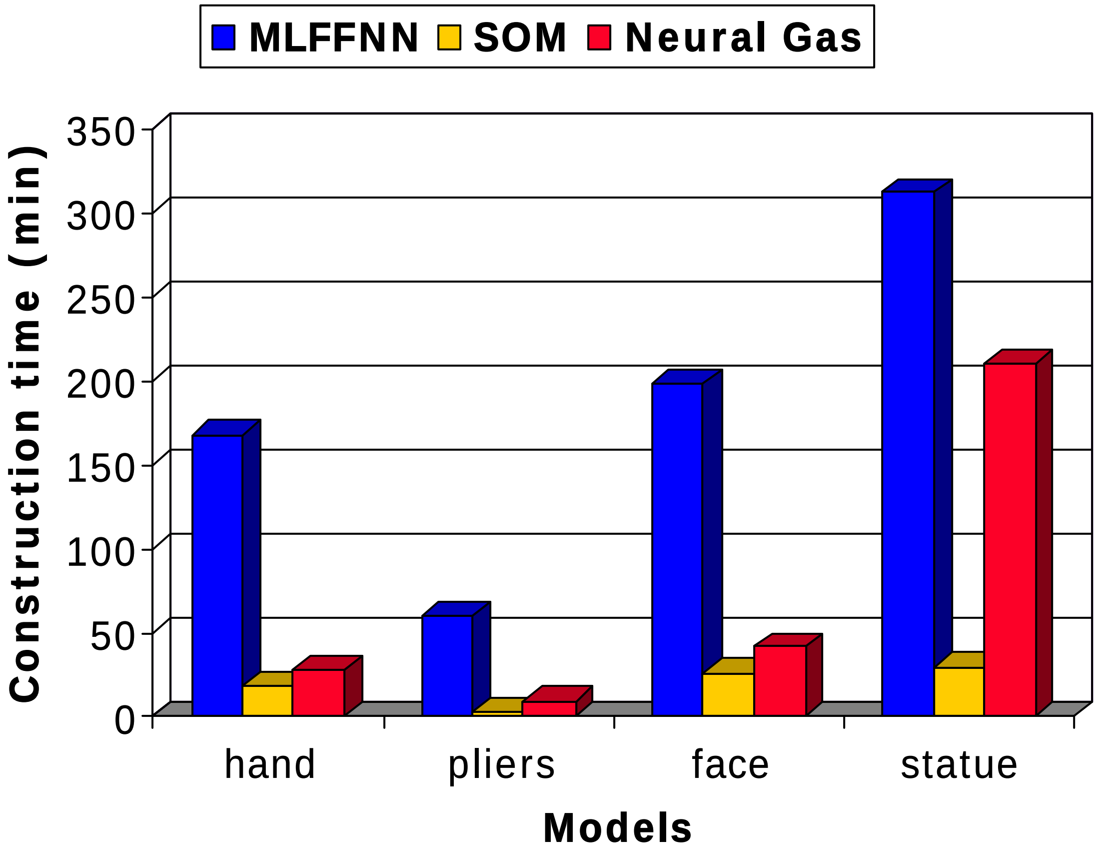
Disadvantages

- 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, SOM, and *Natural Gas* Modelling – Performance Comparison

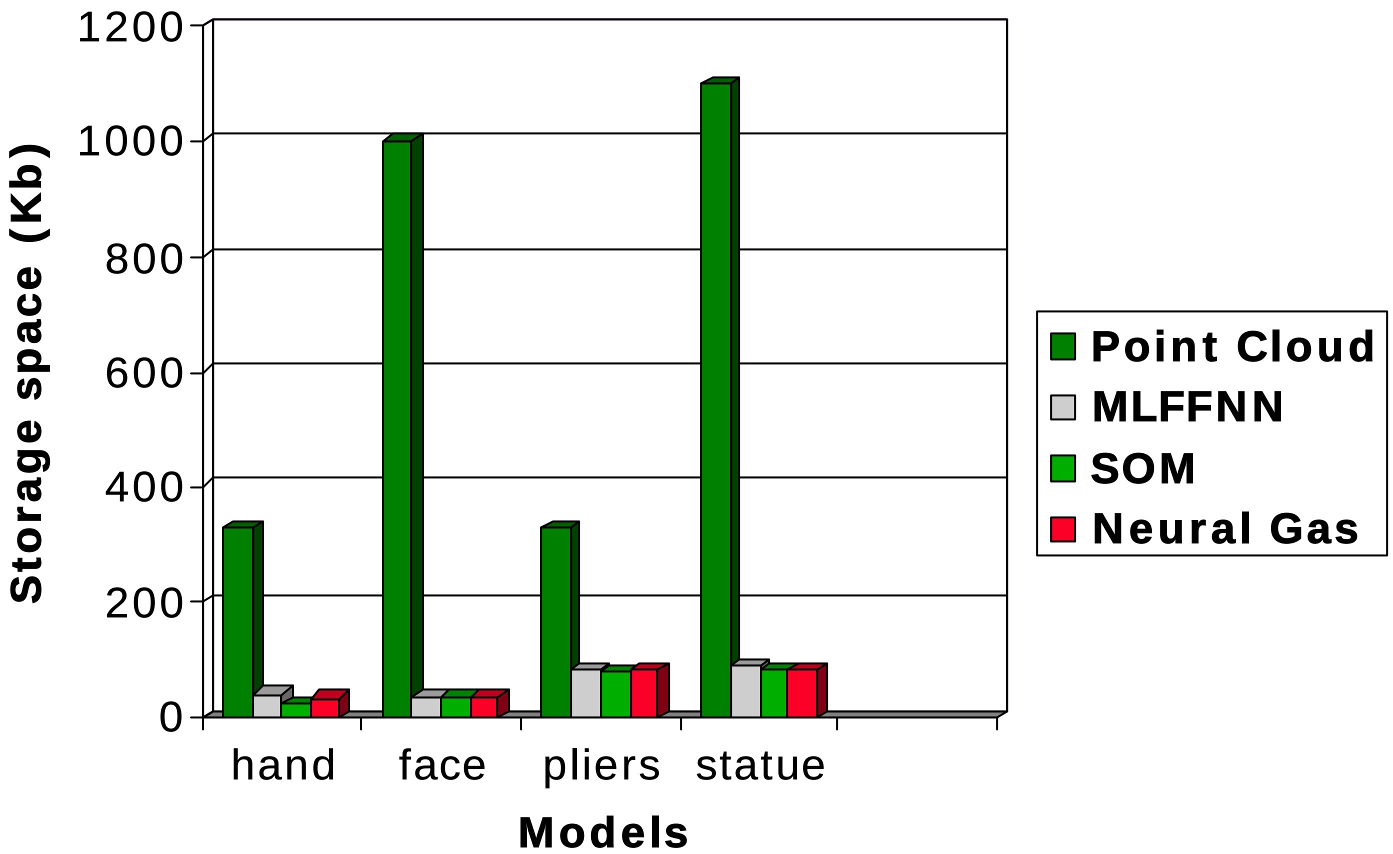


- **MLFNN**
  - computational time = construction time + generation time+rendering

- **SOM and Neural Gas**
  - computational time =

    construction time + rendering

MLFF, SOM, and *Natural Gas* Modelling – Performance Comparison
Compactness



CONCLUSIONS

- The use of neural network modeling advantageous mainly for simplicity and compactness
- ***MLFNN*** – continuous model, information on the entire object space, many applications, but time consuming
- ***SOM and Neural Gas*** – compressed model while maintaining the properties of the object, very good accuracy, less time consuming
- The use of different techniques depends on the application requirements.

Neural Network Adaptive Sampling of 3D Surface Elastic Properties

A-M. Cretu, *E.M. Petriu*, G.G. Patry, "Neural Network-Based Adaptive Sampling of 3D Object Surface Elastic Properties," *Proc. IMTC/2004, IEEE Instrum. Meas. Technol. Conf.*, pp. 285-290, Como, Italy, May 2004

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 of the touched points.

Tactile probing is a time consuming Sequential operation

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.

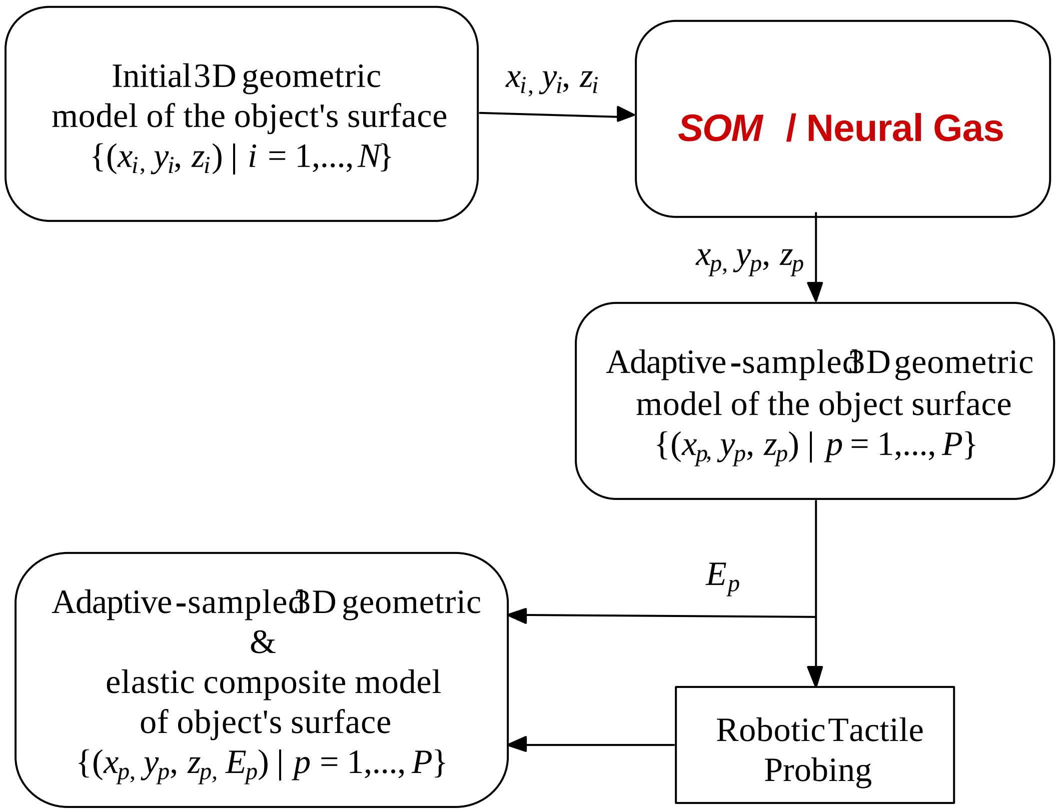
non-uniform adaptive sampling algorithm of the object's surface, which exploits the SOM (*self-organizing map) ability to find optimal finite quantization of the input space.*

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

$$\begin{cases} s_p = E_p \cdot e_p & \text{if } 0 \leq e_p \leq e_{p\max} \\ s_p = s_{p\max} & \text{if } e_{p\max} < e_p \end{cases}$$

where E_p is the modulus of elasticity, s_p is the stress, and e_p is the strain on the normal direction.

Adaptive Sampling Control of the Robotic Tactile Probing of Elastic Properties of 3D Object Surfaces



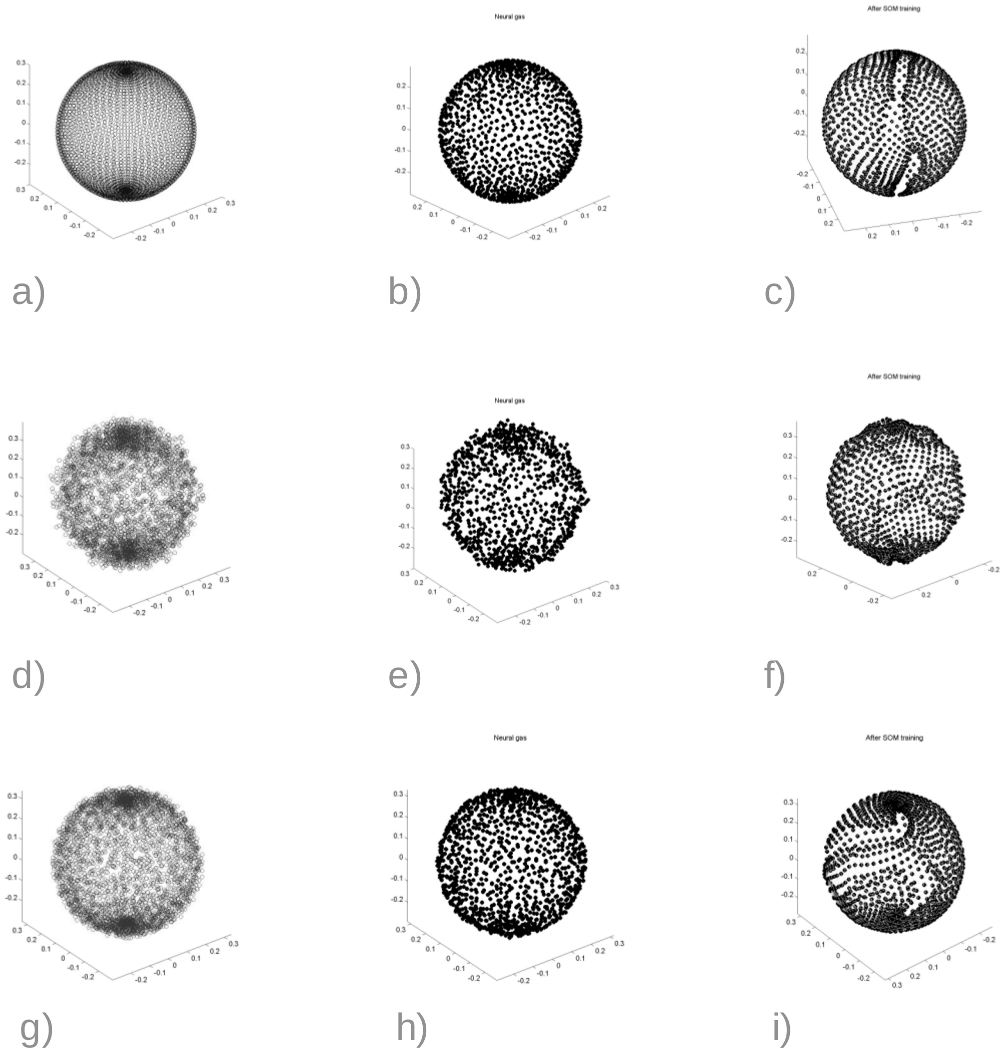
- ***SOM*** (Self Organizing Map) and ***Neural Gas*** NN architectures are both used to build compressed model of the 3D object originally defined as a point-cloud.

- 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 3721points
(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

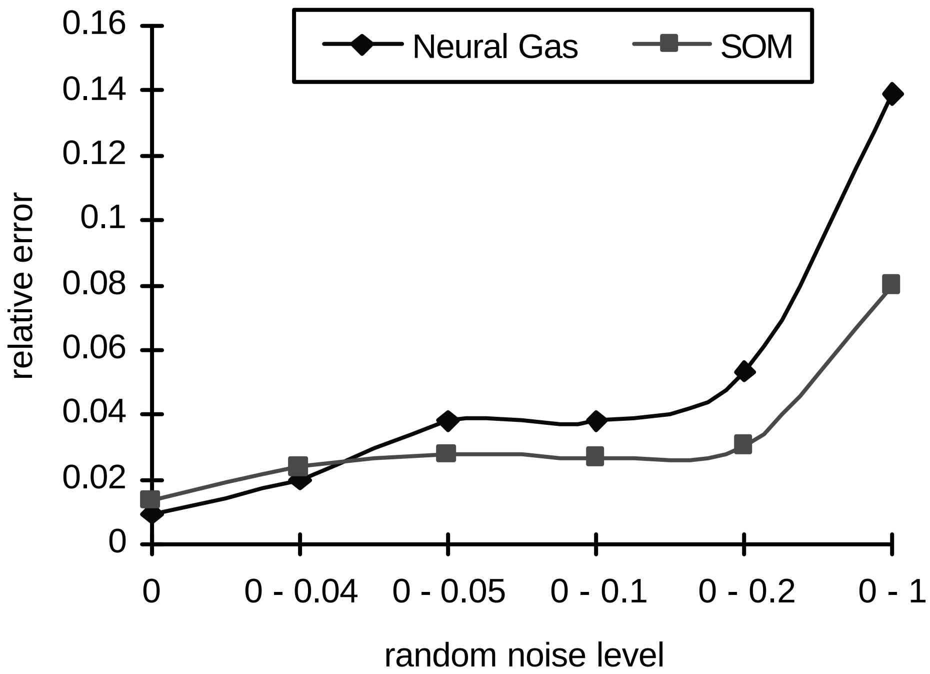
Training Neural Gas network

with a map size of 25×45 for 20 epochs, for a_0 = 0.5, l_0 = number of neurons/2 and a SOM, with the initial neighborhood radius s_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.

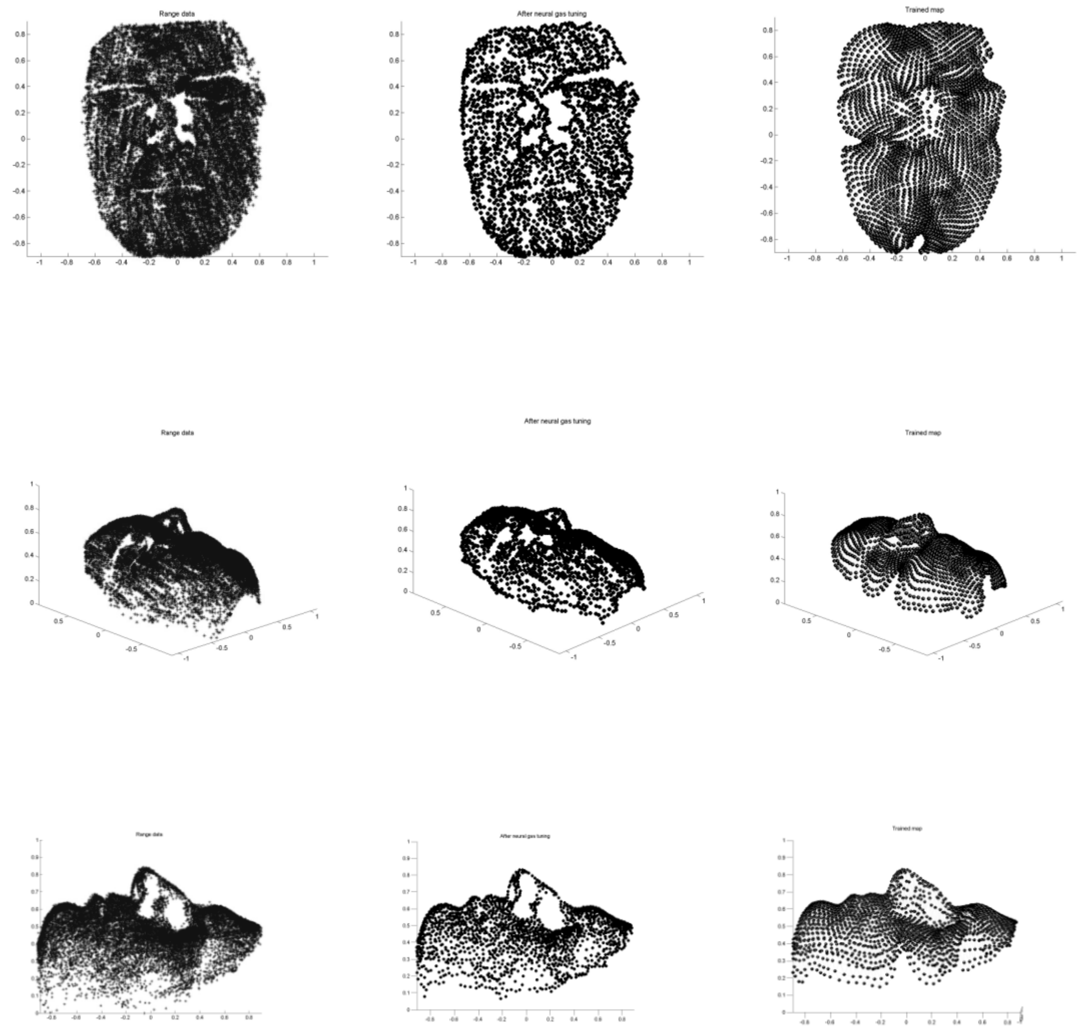
It takes approximately 250s for the SOM to build a model of a sphere, while it takes approximately double for the Neural Gas NN. However, even for a larger number of training epochs (5 times more) the SOM does not reach the same accuracy as the Neural Gas NN does, for data that is not very noisy (a random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

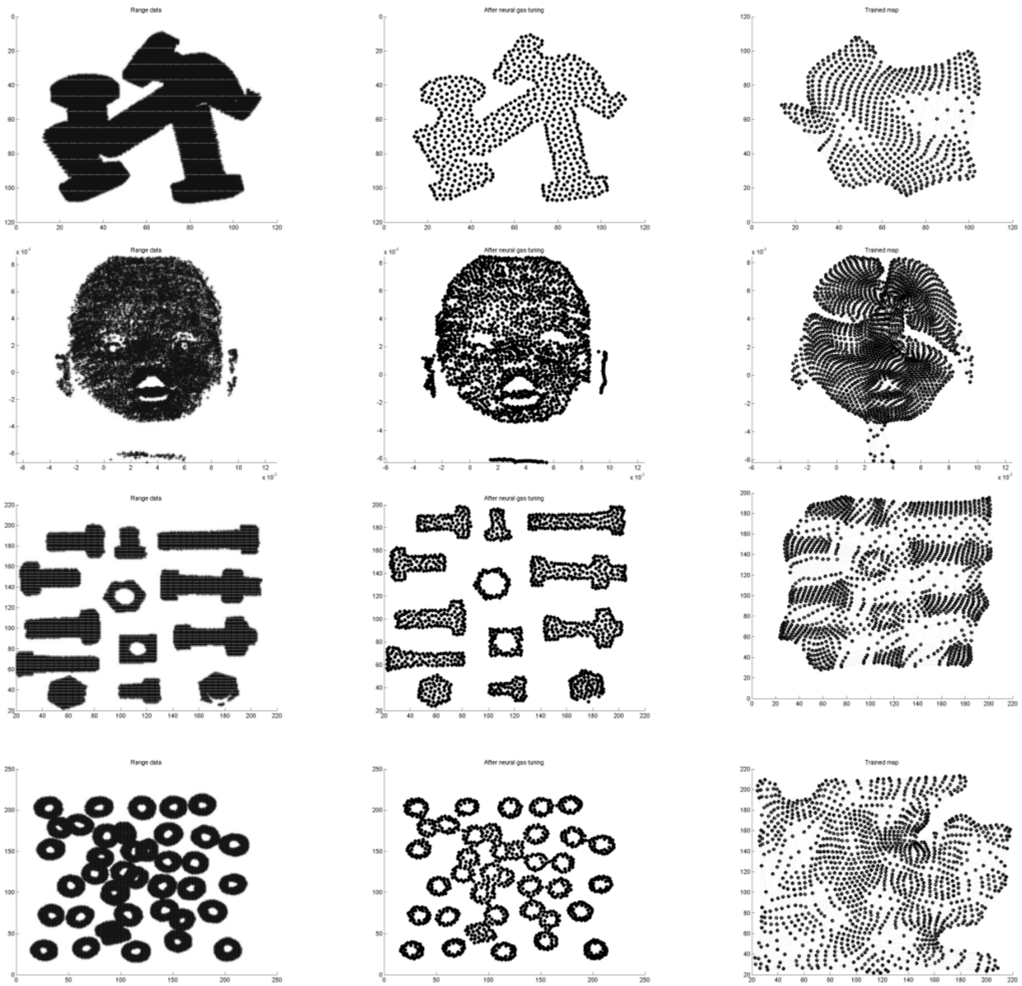
>>> Adaptive Sampling Probing of Elastic Properties of 3D Object Surfaces



• *The first column* presents three views of the original point-cloud of 19080 points representing a human face.

• *The second column* presents the compressed model of 1152 points obtained using The Neural Gas network.

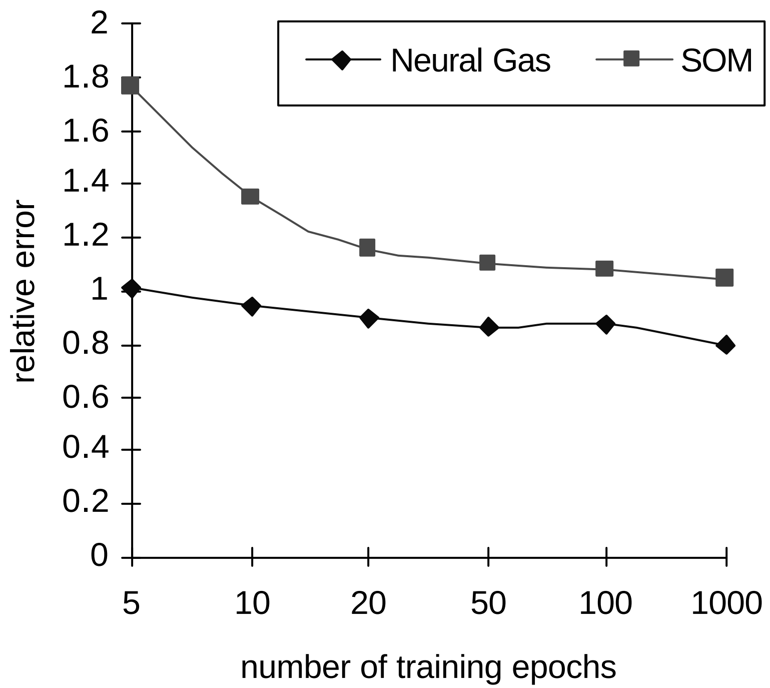
• *The third column* presents the compressed model of 1152 points obtained using SOM.



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 14914 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

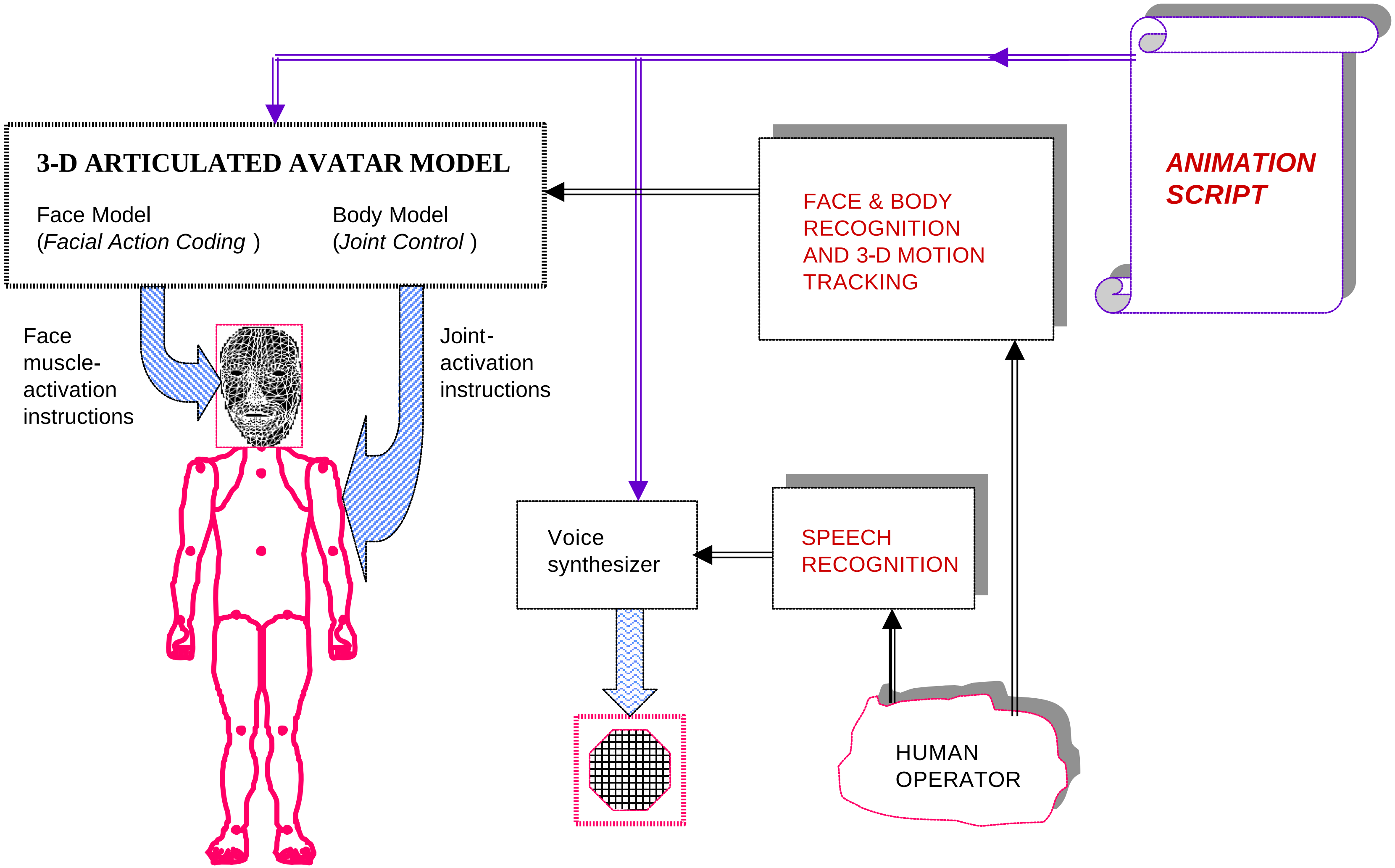
On the whole the quality of the Neural Gas models appears to be better. Because of the boundary problem, the SOM models are to be avoided for non-noisy data.

- 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.

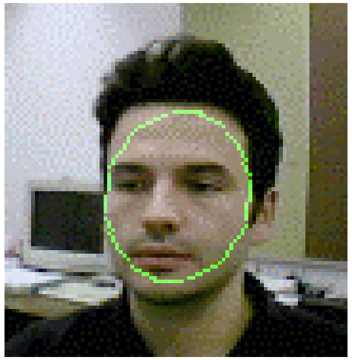
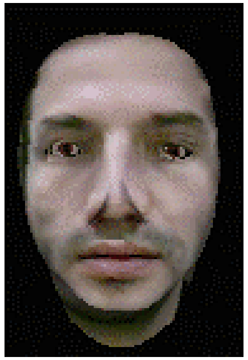
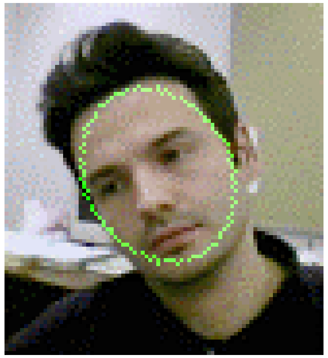
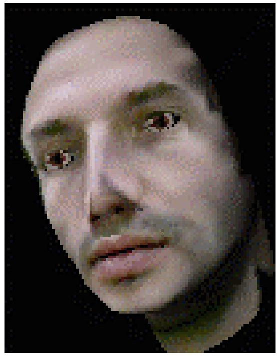
Modelling Avatar Behaviours in Interactive Virtual Environments.

T.E. Whalen, D.C. Petriu, L. Yang, E.M. Petriu, M.D. Cordea, "Capturing Behaviour for the Use of Avatars in Virtual Environments," *CyberPsychology & Behavior*, Vol. 6, No. 5, pp. 537-544, 2003

M. D. Bondy, *E. M. Petriu*, M. D. Cordea, N. D. Georganas, D. C. Petriu, and T. E. Whalen, "Model-based Face and Lip Animation for Interactive Virtual Reality Applications", *Proc. ACM Multimedia 2001*, pp. 559-563, Ottawa, ON, Canada, Sept. 2001


3-D ARTICULATED AVATAR MODEL
Face Model
(Facial Action Coding)
Body Model
(Joint Control)
Face muscle-activation instructions
Joint-activation instructions
FACE & BODY RECOGNITION AND 3-D MOTION TRACKING
ANIMATION SCRIPT
Voice synthesizer
SPEECH RECOGNITION
HUMAN OPERATOR

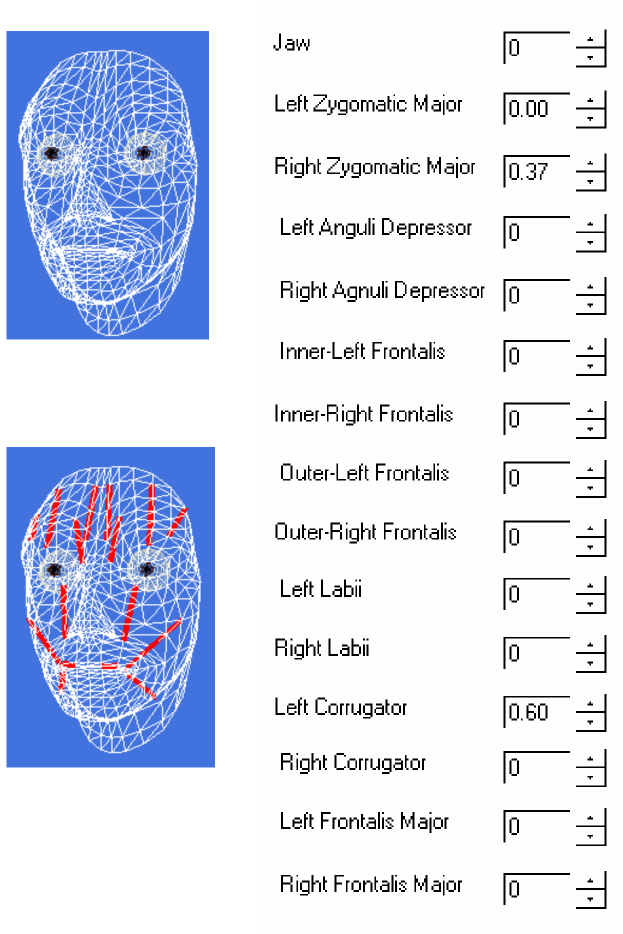
REAL-TIME TRACKING OF THE HEAD & BODY MOVEMENTS

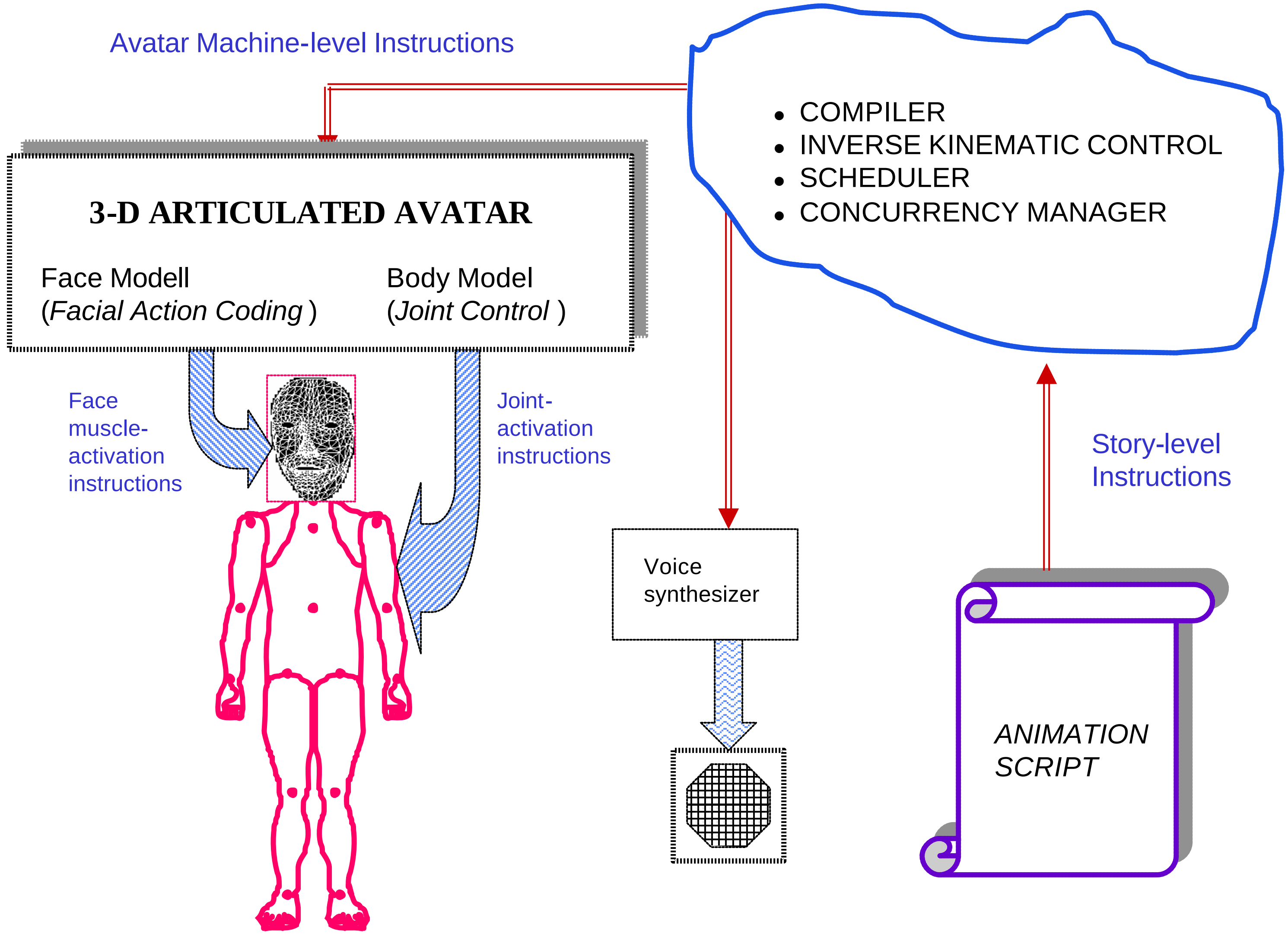
- *R. Rae and H.J. Ritter, "Recognition of Human Head Orientation Based on Artificial Neural Networks," IEEE Tr. Neural Networks, Vol. 9, No. 2, pp.257-265, March 1998]*
- *M.D. Cordea, "Real Time 3D Head Pose Recovery for Model BasedVideo Coding," M.A.Sc. Thesis, SITE/OCIECE, University of Ottawa, 2001*
- *C. Rigotti, P. Cerveri, G. Andreoni, A. Pedotti, and G. Ferrigno, "Modeling and Driving a Reduced Human Mannequin through Motion Captured Data: A Neural Network Approach," IEEE Tr. Syst. Man Cyber., Vol. 31, No. 3, pp. 187-193, May 2001*

REAL-TIME RECOGNITION OF FACIAL EXPRESSIONS

❖ Facial expressions can be described using the ***Facial Action Coding System*** , allowing to control the movements of specific facial muscles. It supports 46 ***Action Units*** – ***AU's*** (37 are muscle controlled and 11 do not involve facial muscles)



❑ *Y.-I. Tian, T. Kanade, and J.F. Cohn, "Recognizing Action Units for Facial Expression Analysis," IEEE Tr. Pattern Analysis and Machine Intelligence, Vol. 23, No. 2, pp. 97-115, Feb. 2001*


Avatar Machine-level Instructions
3-D ARTICULATED AVATAR
Face Modell
(Facial Action Coding)
Body Model
(Joint Control)
- COMPILER
- INVERSE KINEMATIC CONTROL
- SCHEDULER
- CONCURRENCY MANAGER
Face muscle-activation instructions
Joint-activation instructions
Voice synthesizer
Story-level Instructions
ANIMATION SCRIPT

Scripting Language: Abstraction Levels

- Three levels of abstraction for the avatar animation scripting language:
  - Highest: story-level description
    - constrained English-like description
    - syntactic and semantic analysis to extract information such as: main player(s), action, subject and object of the action, relative location, degree, etc.
    - translate in a set of skill-level instructions, that may be executed sequentially or concurrently
  - Middle: skill-level macro-instructions
    - describe basic body and facial skills (such as walk, smile, wave hand, etc.)
    - each skill involves a number of muscle/joint activation instructions that may be executed sequentially or concurrently
  - Lowest: muscle/joint activation instructions
    - activation of individual muscles or joints to control the face, body or hand movement

Personalizing Skills

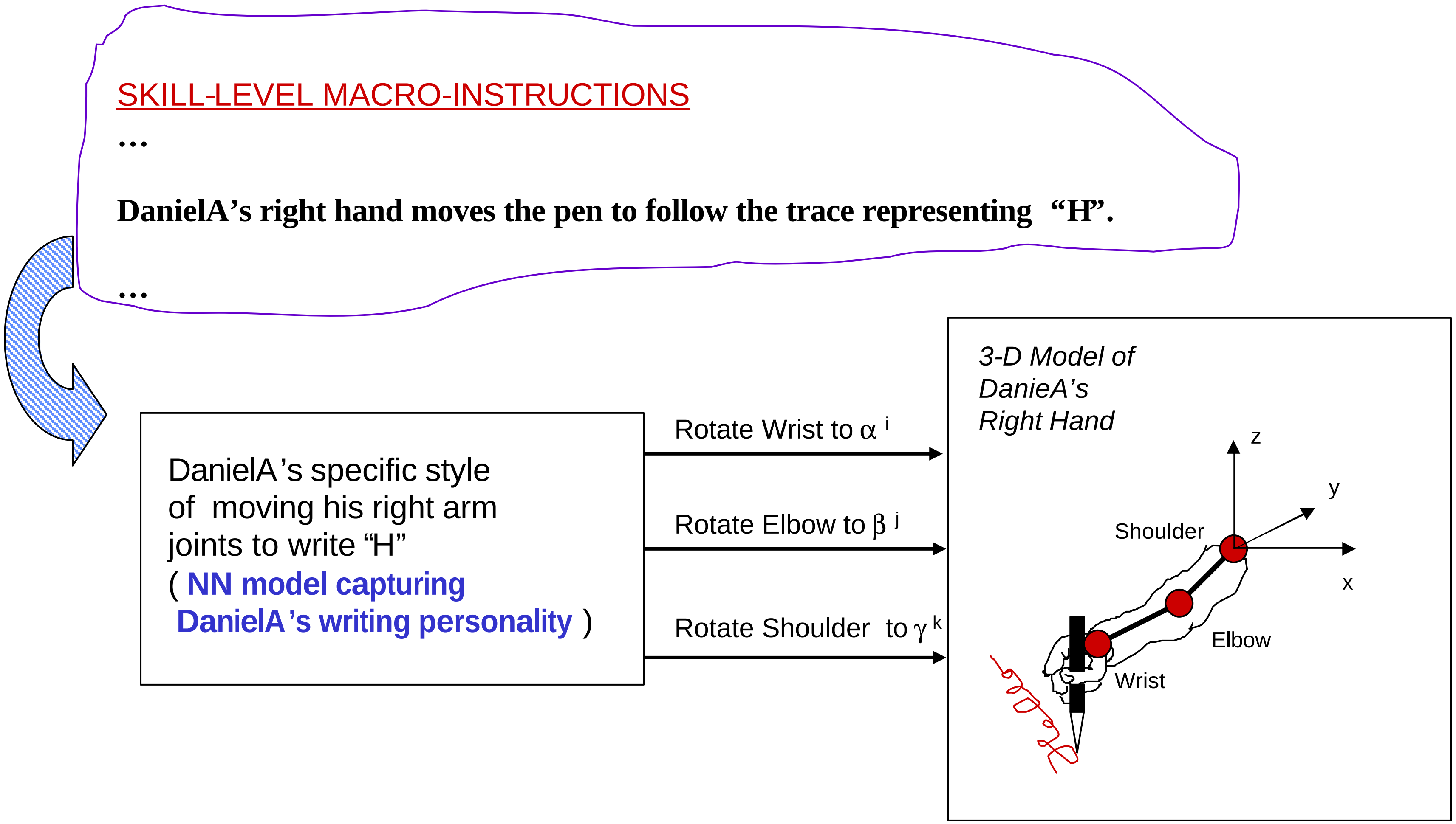
- Add “personality” to skill-level macro-instructions
  - different avatars may perform a certain skill in a “personalized” way
    - examples: “walk like Charlie Chaplin”
      “write like Emil”
  - there is a skill generalization/specialization relationship (similar to object-oriented systems) between
    - a generic skill
    - one or more specialized (or personalized) skills
- Personalizing skills
  - by using Neural Network models
    - off-line training
    - on-line rendering

STORY-LEVEL DESCRIPTION

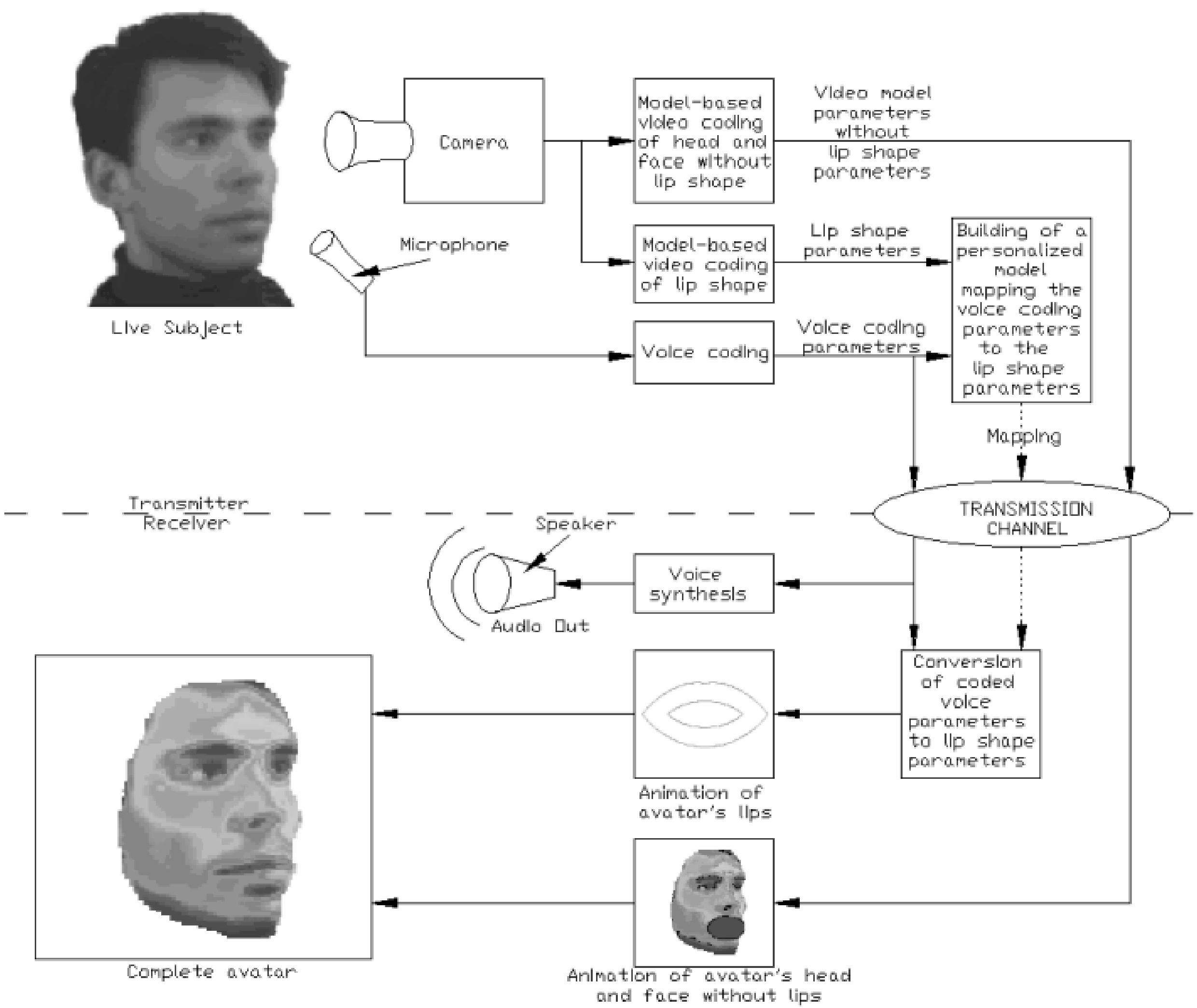
…..
DanielA sits on the red chair.
DanielA writes "Hello" on stationary.
DanielA sees HappyCat under the white table and starts smiling.
HappyCat grins back.
……

SKILL-LEVEL ("MACRO") INSTRUCTIONS

…..
DanielA's right hand moves the pen to follow the trace representing "H".
DanielA's right hand moves the pen to follow the trace representing "e".
DanielA's right hand moves the pen to follow the trace representing "l".
DanielA's right hand moves the pen to follow the trace representing "l".
DanielA's right hand moves the pen to follow the trace representing "o".
……



- M. Costa, P. Crispino, A. Hanomolo, and E. Pasero, "Artificial Neural Networks and the Simulation of Human Movements in CAD Environments", *International Conference on Neural Networks*, 1997, vol. 3, pp. 1781 -1784



- M. Bondy, “Voice Stream Based Lip Animation for Audio-Video Communication,” M.A.Sc. Thesis, 2001

References

- W. McCulloch and W. Pitts, “A Logical Calculus of the Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, 1943.
- D.O. Hebb, *The Organization Of Behavior*, Wiley, N.Y., 1949.
- 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.
- F. Rosenblat, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,” *Psychological Review*, Vol. 65, pp. 386-408, 1958.
- B. Widrow and M.E. Hoff, “Adaptive Switching Circuits,” *1960 IRE WESCON Convention Record, IRE Part 4*, pp. 94-104, 1960.
- M. Minski and S. Papert, *Perceptrons*, MIT Press, Cambridge, MA, 1969.
- J.S. Albus, “A Theory of Cerebellar Function,” *Mathematical Biosciences*, Vol. 10, pp. 25-61, 1971.
- T. Kohonen, “Correlation Matrix Memories,” *IEEE Tr. Comp*., Vol. 21, pp. 353-359, 1972.
- J. A. Anderson, “A Simple Neural Network Generating an Interactive Memory,” *Mathematical Biosciences*, Vol. 14, pp. 197-220, 1972.
- S. Grossberg, “Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors,” *Biological Cybernetics*, Vol. 23, pp.121-134, 1976.
- J.J. More, “The Levenberg-Marquardt Algorithm: Implementation and Theory,” in Numerical Analysis, pp. 105-116, Spriger Verlag, 1977.
- K. Fukushima, S. Miyake, and T. Ito, “Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition,” *IEEE Tr. Syst. Man Cyber*., Vol. 13, No. 5, pp. 826-834, 1983.

- D. E. Rumelhart, G.E. Hinton, and R.J. Willimas, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, (D.E. Rumelhart and J.L. McClelland, Eds.,) Vol.1, Ch. 8, MIT Press, 1986.
- D.W. Tankand J.J. Hopefield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Tr. Circuits Systems*, Vol. 33, No. 5, pp. 533-541, 1986,
- M.J.D. Powell, "Radial Basis Functions for Multivariable Interpolation" A Review," in *Algorithms for the Approximation of Functions and Data*, (J.C. Mason and M.G. Cox, Eds.), Clarendon Press, Oxford, UK, 1987.
- G.A. Carpenter and S. Grossberg, "ART2: Self-Organizing of Stable Category Recognition Codes for Analog Input Patterns," *Applied Optics*, Vol. 26, No. 23, pp. 4919-4930, 1987.
- B. Kosko, "Bidirectional Associative Memories," *IEEE Tr. Syst. Man Cyber*., Vol. 18, No. 1, pp. 49-60, 1988.
- T. Kohonen, *Self_Organization and Associative Memory*, Springer-Verlag, 1989.
- K. Hornic, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989.
- B. Widrow and M.A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proc. IEEE*, pp. 1415-1442, Sept. 1990.
- B. Kosko, *Neural Networks And Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 1992.
- E. Sanchez–Sinencio and C. Lau, (Eds.), *Artificial Neural Networks*, IEEE Press, 1992.
- A. Hamilton, A.F. Murray, D.J. Baxter, S. Churcher, H.M. Reekie, and L. Tarasenko, "Integrated Pulse Stream Neural Networks: Results, Issues, and Pointers," *IEEE Trans. Neural Networks*, vol. 3, no. 3, pp. 385-393, May 1992.
- S. Haykin, *Neural Networks: A Comprehensive Foundation*, MacMillan, New York, 1994.
- M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, NY, 1994.

- C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, NY, 1995
- M.T. Hagan, H.B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Co., 1995
- S. V. Kartalopoulos, *Understanding Neural and Fuzzy Logic:Basic Concepts and Applications*, IEEE Press, 1996.
- M. T. Hagan, H.B. Demuth, M. Beale, *Neural Network Design*, PWS Publishing Co., 1996.
- C.H. Chen (Editor), *Fuzzy Logic and Neural Network Handbook*, McGraw Hill, Inc., 1996.
- ***, "Special Issue on Artificial Neural Network Applications," *Proc. IEEE*, (E. Gelenbe and J. Barhen, Eds.), Vol. 84, No. 10, Oct. 1996.
- J.-S.R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, 1997.
- C. Alippi and V. Piuri, "Neural Methodology for Prediction and Identification of Non-linear Dynamic Systems, " in *Instrumentation and Measurement Technology and Applications*, (E.M. Petriu, Ed.), pp. 477-485, IEEE Technology Update Series, 1998.
- ***, "Special Issue on Pulse Coupled Neural Networks," *IEEE Tr. Neural Networks*, (J.L. Johnson, M.L. Padgett, and O. Omidvar, Eds.), Vol. 10, No. 3, May 1999.
- C. Citterio, A. Pelagotti, V. Piuri, and L. Roca, "Function Approximation – A Fast-Convergence Neural Approach Based on Spectral Analysis, *IEEE Tr. Neural Networks*, Vol. 10, No. 4, pp. 725-740, July 1999.
- ***, "Special Issue on Computational Intelligence," *Proc. IEEE*, (D.B. Fogel, T. Fukuda, and L. Guan, Eds.), Vol. 87, No. 9, Sept. 1999.
- L.I. Perlovsky, *Neural Networks and Intellect, Using Model-Based Concepts*, Oxford University Press, NY, 2001.

• T.M. Martinetz, S.G. Berkovich, and K.J. Schulten, “Neural-Gas Network for vector quantization and its application to time-series prediction”, *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp.558-568, 1993.
• ***, “SOM toolbox online documentation”, *http://www.cis.hut.fi /project /somtoolbox/documentation/*
• N. Davey, R.G. Adams, and S.J. George, “The architecture and performa nce of a stochastic competitive evolutionary neural tree network”, *Applied Intelligence* 12, pp. 75-93, 2000.
• B. Fritzke, “Unsupervised ontogenic networks”, *Handbook of Neural Computation*, Eds. E. Fiesler, R. Beale, IOP Publishing Ltd and Oxford University Press, C2.4, 1997.
• N. Kasabov, *Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*, Springer Verlag, 2003.
• N. Morgan and H. Bourlard, “Neural Networks for Statistical Recognition of Continuous Speech,” *Proc. IEEE, Vol 83, No. 5, pp.742-770, May 1995*
• E.M. Petriu, "Neural Networks for Measurement and Instrumentatio n in Virtual Environments,” in Neural Networks for Instrumentation, Measurement and Related Industrial Applications , (S. Ablameyko, L. Goras, M. Gori, V. Piuri - Eds.), NATO Science Series, Series III: Computer and System Sciences, Vol. 185, pp.273-290, IOS Press, 2003