EXTRANOEMATIC ARTIFACTS

Neural systems in space and topology

Martin Kaftan

This dissertation is submitted in partial fulfillment of the requirements for the degree of
Master of Science in Adaptive Architecture & Computation
from the University of London.

Bartlett School of Graduate Studies
University College of London
September 2007

+ABSTRACT

During the past several decades, the evolution in architecture and engineering went through several stages of exploration of form. While the procedures of generating the form have varied from using physical analogous form-finding computation to engaging the form with simulated dynamic forces in digital environment, the self-generation and organization of form has always been the goal. this thesis further intend to contribute to self-organizational capacity in Architecture.

The subject of investigation is the rationalizing of geometry from an unorganized point cloud by using learning neural networks. Furthermore, the focus is oriented upon aspects of efficient construction of generated topology. Neural network is connected with constraining properties, which adjust the members of the topology into predefined number of sizes while minimizing the error of deviation from the original form. The resulted algorithm is applied in several different scenarios of construction, highlighting the possibilities and versatility of this method.

+TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank to Alasdair Turner for introducing me into the world of computation, to Ava Fatah for her guidance and help when needed, to Sean Hanna for sharing his knowledge on the subject and helping with direction during the research, and Chiron Mottram for helping to solve many computational problems.

Also thanks to my family for their support, and .... *dedovi*.

1+ INTRODUCTION

Unorganized point clouds can contain millions of points in space to represent geometry. Even though they demonstrate the shape in overall, each point has no relation to the others, containing only knowledge of its own spatial position. Therefore, just to reconstruct precise geometry from unorganized point clouds is a very difficult problem, even more if we add other requirements, such as optimization of the geometry for manufacturing purposes. A common example of unorganized point clouds derives from 3D laser scanners or photogrammetric image measurements, which are often incomplete, sparse and noisy data. To construct from an unorganized point cloud requires highly efficient algorithm with adequate processing speed. The solution to this task could lay in inspiration from other field of science.

Over a century, architects and engineers have been pursuing self-generation through different techniques and forms. Part of the research process has been engagement with generative models inspired by fields, such as biology.

An example of inspiration is the research work of D'Arcy Thompson. In his book On Growth and Form, Thompson developed understanding of living forms in terms of physics, studying affection of dynamics on organisms and their transformation through growth and movement. Thompson demonstrates that form of organisms is obeyed by engineering principles of continuous variety in biological materials such as structure of bones where some parts of the bone's structure deal with tension while other with compression (Thompson, p.230-238).

Architects inspired by Thompson and others tried to apply the knowledge into generating architecture in different ways. In the early 1990's, German architect Frei Otto and his team at the Institute for Lightweight Structures explored self-organization of design by using analogous form-finding machines. Otto used transition between different phases of materials such as soap film, wool thread, and glue or varnish to let the materials to find the form themselves through complex material's behavior (Otto,p.56-p.71). He used the imitation of nature in favor of working directly with the materials to produce models that were both natural and artificial. On the other hand, another architect Greg Lynn, inspired beside other by Thompson's analyses of 'variations in the morphology of animals using deformable grids' (Lynn,p.27), operated fully inside virtual environment to create self-generative architecture. Lynn used animating software capable of simulating dynamic forces, which he applied on the form to shape it.

And recently, architects have become interested with new possibilities of generative exploration in design through emergent self organizing algorithms, such as are Boids, Cellular Automaton or Genetic Algorithms. These emergent systems produce complex global behavior derived from local interaction between large amount of simple parts. One of the interesting computational emergent systems are the artificial neural networks.

The era of computational research in artificial intelligence was stimulated in 1943 when Walter Pitts and Warren S. McCulloch published their paper "A logical calculus of the ideas immanent in nervous activity" (McCulloch,p.115-133) . The thesis demonstrated the possibility to construct artificial neural networks by mathematical functions and algorithms. Their neural network was composed by a set of nodes called "binary decision units", which had the ability to be programed to solve any task that could be computed.

The term Artificial Neural Network in generally describes apparatus that consists number of interconnected neurons. The Artificial Neural Network reflects some apparent similarities with the way the neural cells are interconnected in the human brain. The neurons communicate with the neighboring neurons, receiving and transmitting information with connections called synapses. Each neuron of the network has capacity to store and process some information. Often, the synaptic connections also store some information, usually in the form of scalar weights. Neural networks are in practice most often simulated by a software, even though a neural network can be implemented as hardware, such as attempts on neural processors.

In the last twenty years neural networks were applied in numerous of applications in different science fields, such as pattern recognition, finance, bio-informatics and others. A typical application for a neural network, as for example the hand-writing recognition, is a simple problem which the human brain almost effortlessly solves, while the computer, despite its computational power, faces unexpected difficulties and requires sophisticated software for sometimes not very satisfactory results.

Acknowledging that the neural networks are capable to operate with large amount of data they seem to be suitable to solve the problems of construction from unorganized point clouds.

The focus in this thesis is on the following research questions:

1) Suitability of neural networks to generate geometry from unorganized point clouds

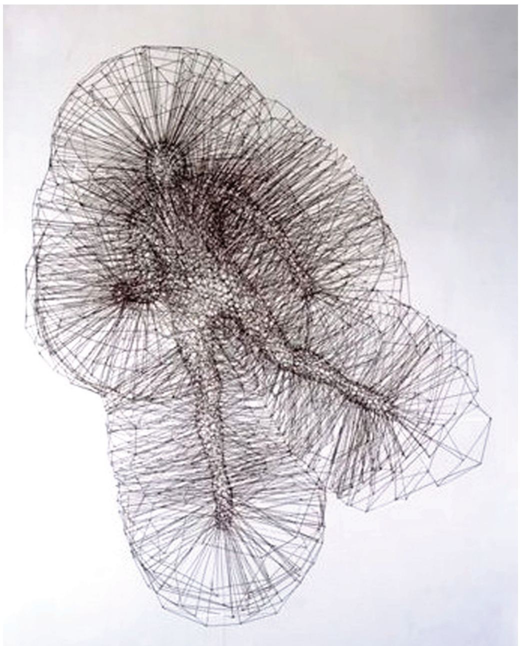
2) Adaptation of neural networks algorithm to adjust the structural members of the geometry to predefined standard sizes for manufacturing purposes

3) Versatility of neural networks algorithm to solve different geometrical tasks

2+ MOTIVATION

The renowned British sculptor Anthony Gormley centers his work on expressing the human body or as he says "the space of the body". One of Gormley's ongoing projects called "Matrices and Expansions" involves experimenting with creation of different types of geometrically based topologies from 3D body scan. For example one of the intriguing sculptures is based on combination of 2.5D and 3D Voronoi geometry. During the visit of Gormley's atelier in London, he mentioned several difficulties he faces during working with body scan data. In the process of construction of Matrices, Gormley first creates a cast of the body. This cast is afterwards scanned by 3D scanner. From the outputting point cloud is generated desired geometry in the computer. Here already appears to be problematic to correctly recreate from the point cloud desired geometry, as well as manipulate with the geometry's construction. When the desired geometry is finally created, however, in the next step this by software precisely generated geometry is used only as a guide for the welder. It is either printed on paper or a 3d model is generated from 3-D CNC printer. For the fabrication process the geometry is marked on the cast as a guide for the welder. Since each member of the geometry has a different size, the welding process is tedious and takes several months, depending on the complexity of the form. The welder has to approximate by visually checking and estimating each member size.

Gormley has been already collaborating on solving construction problems by generating computational models of Matrices with Sean Hanna who is a research engineer at University College London. Hanna applied on unorganized point cloud several algorithms to receive required performative geometry. He successfully used algorithms such as close packing and cellular automaton to receive topologically correct structure for the construction (Hanna). However, the problem of increasing efficiency of constructing from unorganized point clouds still remains.
 Hence here comes the desire to investigate the possibilities of manipulation and construction from an unorganized point cloud.



Anthony Gormley "Matrices"

3+ POINT CLOUDS FOR THE EXPERIMENTS

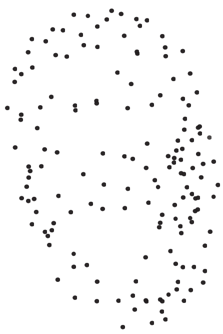
The experiments are mostly first tested in 2D environment and if successful they are proceeded into 3D. To receive comparable results, the experiments were mostly performed on standard models.

The 2D environment consists of input data randomly selected for each iteration from inside of a rectangular boundary. The amount of points is therefore immensely large, and the point cloud evenly covers that area. This provides good comparison of influence of good distribution of input data to other used models with lower data resolution.

For the 3D environment the point clouds were obtained from Antony Gormley; they are text files with xyz coordinates received from the 3D scanner, the same data format, that Gormley uses for generating his sculptures. The scanned objects are human bodies. The point clouds comprise two resolutions of the identical body and position; one is 2, 000 points resolution and the other 20, 000 points. Extracted parts of the point clouds such as only the head part are also used.

Point cloud from a body scan 20000 points resolution

Point cloud from a body scan - 2000 points resolution

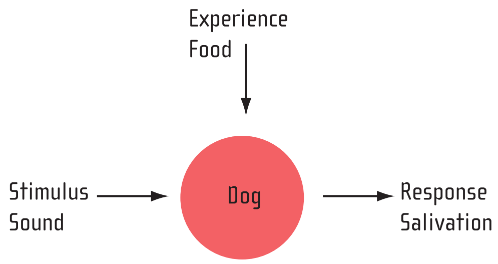


Point cloud from a head scan - 157 points resolution

4. INTRODUCTION TO NEURAL NETWORKS

4.1. ASSOCIATIVE LEARNING OF BIOLOGICAL NEURAL NETWORKS

Artificial neural networks similarly to biological ones respond to stimulus inputs by learning the conditions, also called the conditioning. Classical example of biological stimulus learning was described by the Russian scientist Ivan Pavlov (1849-1936). Pavlov trained dogs to associate sound with a food (Rescorla,p.266). The experiment involved several weeks of consistently switching on a sound during the feeding time. The dog's changes of behavior progressed:
a) the dogs at the beginning indicate no response to the sound
b) the dogs respond to food with salivation
c) after several weeks when the sound has been repeatedly switched on during the feeding time, the dogs begin to salivate as a response to the sound with absence of food.



The function of the response has been augmented; at the beginning dogs did not react to the stimuli (sound) , now the stimulus evokes a response (salivation). This change has occurred because of an experience (food). To describe the associative function in the terms of neural network apparatus:
Before the testing, synapses existed between the neurons detecting the presence of food and the neurons operating the salivation. There was no connection between the neurons detecting sound and the neurons for salivation. After the testing, the neurons triggers salivation as a response to the sound now without presence of food, indicating that dog's brain created and strengthened new synaptic connections. For this to happen, the sound and the food had to be presented simultaneously, meaning that the dogs went through an associative learning process.

4.2.+ ARTIFICIAL NEURAL NETWORKS

An artificial neural network can process the input data in several ways. In the first category of neural networks, the network can directly perform some computations, by receiving an input signal, processing it on its nodes, and outputting the result. An example of a direct neural network are the Braitenberg Vehicles.

Valentino Braitenberg in his book "Vehicles: experiments in synthetic psychology." (Braitenberg) used thought experiments to explore psychological ideas and the principles of intelligence in which an emergent behavior emerge from interaction of simple component parts. In his vehicles he used inhibitory and excitatory influences by connecting light sensors directly to motors. The resulting motion It is a reflex behavior called the "positive phototropism", similar to behavior of insect such as moths that are attracted to light. In this process each side of the insect is exited or inhibited, depending on the side from which they perceive a strong source of light, having the effect of steering the insect towards the light source. Thus, The steering derives from competition between two neurons where each is stimulated by a light intensity. The output, the velocity, is then linearly proportional to the input.

The second category of neural networks, which are the subject of studying for this thesis, can be trained in a learning process as the above described biological networks. In this case the neural network processes the input signal by adjusting itself, and the state of the network is considered the output. The adjustment of the network during the learning process can include changes in the information stored on the nodes and the connections, as well as changes in the architecture of the network. That is, the network adapts its state to the input signals by creating new nodes and connections, or/and by adjusting the existing ones.

One of the most widely used in practical applications is the Kohonen's Self Organizing Map .A Self Organizing Map adjusts itself to the input signals, learning gradually the space of the input data. The networks generates mappings from high-dimensional signal spaces to lower dimensional topological structures. It can reveal patterns of that input data space that the human brain cannot detect, either because of the nature of these data, like high dimensionality, or because of their size and complexity.

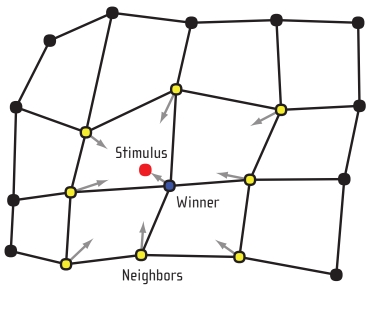
4.3.+ KOHONEN NEURAL NETWORK

Kohonen Self –organizing neural network model were first proposed by Kohonen in 1982 (Oja,p.V). The main idea behind Kohonen neural network is to leave the network organize itself through learning process. To do this the patterns must be presented continuously and randomly until stability is reached.

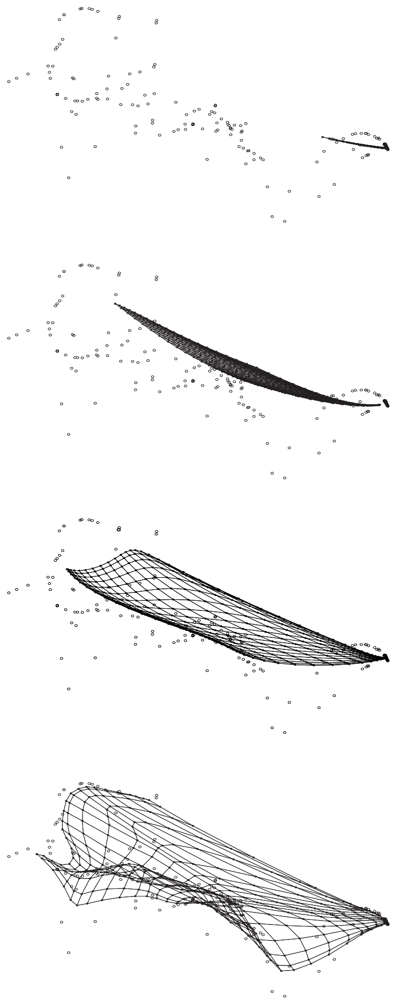
The network is composed of a predefined 2 dimensional array of geometric points in space which make up their members or neurons. This is a rectangular shape with equal number of rows in x and y dimension. Each neuron communicates, thus is connected to every other node in a topological rather than topographical manner. The self-organizing map preserves its topology by respecting the initially laid out structure of the array. The neural network receives input in the form of system external 3 dimensional points which are the array points from the point cloud. These vectors (further referred to as stimuli) are read by all the neurons. Whichever neuron is the has the closest position to any of the stimuli organizes his neighbors according to the signal from the stimuli. When all of the neurons are doing that simultaneously, the result of such an input reading is the adaptation of the neural network to the mapped space - it has learned the input. The learning is a mathematical function applied by the neurons that find similar input stimuli to all the nodes in the network. That way all the neurons read all of the input stimuli, but only some of them are 'winners' who get to excerpt feedback onto the network is called competition. At the end of each main loop run, the learning parameters are updated through monotonously decreasing their values. This subroutine returns the winners. Other neurons may be either inhibited or excited, depending upon their distances from the winner neuron. The feedback between the neighbors starts with the call to the winner's neighbors. Every time this subroutine is called within the main loop, the learning parameters have diminished. Thus, slowly the whole network is reorganizing itself in parallel matter.

Their weight is updated by dividing Eulerian (e) distance (dist) between neurons (n) with consistently decreasing learning factor (factor). This method is also called the "Mexican Hat", deriving from the shape of the wavelet.

$$n[k,l] = e^{(-\,dist(k,l)^2)\,/\,(2\,factor)}$$

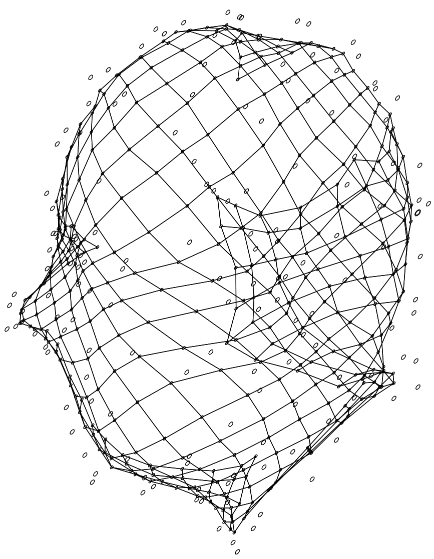


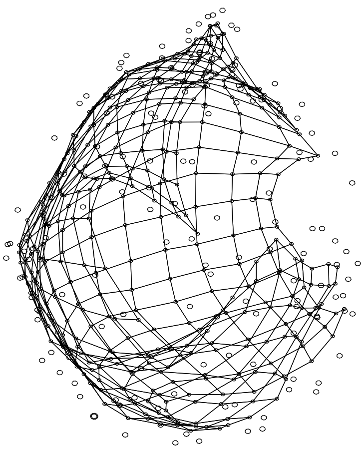
Adjustment of neuron's positions



Demonstrating the Kohonen network in 3D environment, a subtle spatially distributed random point cloud of 180 stimuli is first mapped. Here the Kohonen network correctly approximates the position of net's neurons among the stimuli.

Secondary the Kohonen network is used with the point cloud of the head from the scan. Because the point cloud represent semi-closed volume, the network is no longer being able to adapt its position correctly by being restricted with its predefined topology.





The Kohonen network demonstrated well mapping of a unorganized point cloud, however, it is restricted by its topology. As can be seen on the experiment with fairly vertically symmetrical semi-open geometry (the 3d scan of a head), the network mapped only on 1/2 of the point cloud. Its suitability is thus limited to mapping open surfaces with subtle convex and concave curvature. In this matter could be used to find a minimum surface, simulating so Frei Otto's soap film form finding analogous machines.

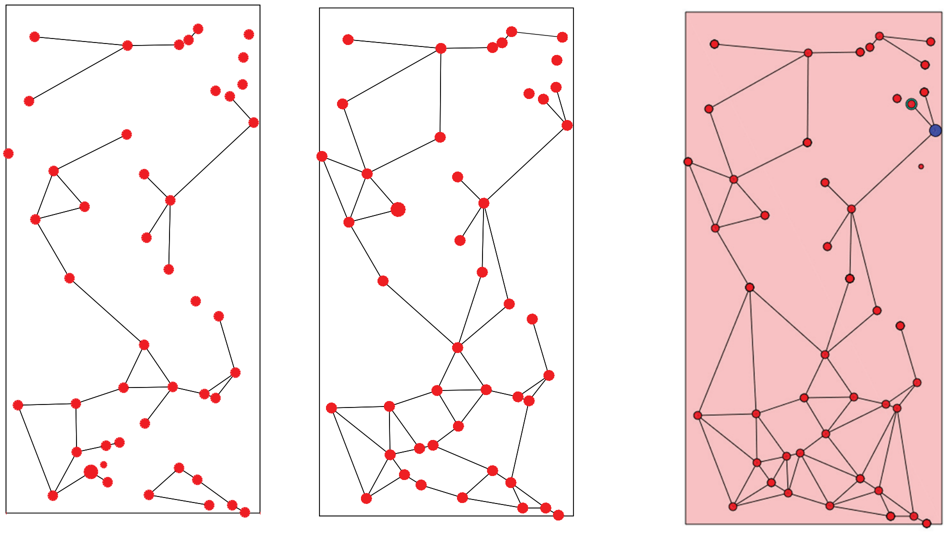
4.4+ SELF-ORGANIZING NEURAL TOPOLOGY

As mentioned above, in the case of Kohonen's SOM the problem was that the predefined rectangular mesh was not suitable for needed flexibility and precise mapping of complex shape point clouds. To obtain the required flexibility, a topologically self-organizing geometry is required, such as is Delaunay triangulation. Among neural networks exists a network called the Hebbian Learning Rule which synoptic connections are topologically related to Delaunay triangulation.

In 1949, Donald Hebb formulated what became the basis of the idea of hebbian Learning "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." (Gerstner,p.1). Hebbian Learning Rule or also called the Competitive Hebbian Learning is common way to calculate changes in connection strengths in a neural network, in which a change in the strength of a connection is a function of the pre – and postsynaptic neural activities. The learning process is associative because both the pre-synaptic n and post-synaptic neurons have to be activated at the same time to change weight in their synaptic connection (Gerstner,p.1).

The pseudocode for the Hebbian learning:

0) Connection is set to 0
1) Random stimulus is chosen from the point cloud
2) Closest two neurons are determined by Euclidian distance between stimulus and neurons
3) If there is no connection between these two neurons, the connection is created
4) repeating again from step 1



8 Iterations   26 Iterations   108 Iterations

4.5.+ INCREMENTAL GROWING NEURAL NETWORK

The combination of advantages of Hebbian's constantly self-organizing topology and Kohonen's adaptation to the input combines Growing Cell Structure algorithm (Fritzke, p.93). It is an incremental neural networks that don't have predefined structure. Instead, the structure is generated by adding or deleting neurons in the network during the learning process. Each network consists of a number of neurons and number of synapses and each neuron has associated position with a stimulus. Upon receiving signal from inputs (point cloud), the nearest neurons (winners) are repositioned toward the signal and after each repositioning error factor is calculated for each winning neuron . The accumulated error determines after predefined number of iterations where to insert new neurons.

GNG only uses the parameters that are constant in time. It employs a modified Kohonen Learning rule (unsupervised) combine with competitive Hebbian learning. The kohonen type learning rule serves to adjust the synaptic weight vectors and Hebbian learning establishes a dynamic lateral connection structure between the units. The new model of growing neural gas can be said the extension of Kohonen's feature map with respect to various important criteria (Fritzke).

The neural network generates topology is the modification of the Delaunay triangulation which is the result of the embedded Competitive Hebbian Learning algorithm. The principle of this method is similar to the method in 4.4. for finding the two closest neuron and its neighbor to the stimuli with the Euclidian distance, now with the addition of finding also other neighbors of the by winning neuron and connecting them with an edge.

A neural network implements aging process to remove connections in the network that is not part of the Delaunay triangulation. After a connection is updated or a new connection is created by the Hebbian, the edge of the connection is set back to 0, signaling to the network that the connection is still active part of the network. If a connection reaches certain predefined value of its age, it indicates to the network that the connection is inactive and is removed from the network. If the removal of connection results in neuron without connection and the neuron is not reconnected after certain number of iterations, the neuron is proclaimed as inactive as if removed from the network.

The growing neural network shows two basic differences from the Kohonen neural network:
1 Only the best-matching unit and its direct topological neighbors are adapted.
2 The adaptation strength is constant over time. For adjusting strength of connection are used two constant adaptation parameters , one for the best matching neuron and the second for the neighboring cells.

The algorithm has also utility factor which if switched on has the ability to adapt to changing distribution by changing location of less useful neurons. As in the aging process for the connections, the algorithm saves an utility factor for each neuron which determine its success in the global growth process. If the factor reaches predefined maximum value the neuron is not deleted, but its repositioned in place with the highest error factor and reconnected by the Hebbian back to the network.

The overall operation progress of the growth can be described in following steps (Oja, p.131-140):

1) First two connected neurons are positioned randomly in the environment. Their error value initialized as 0.

2} From the point cloud is randomly selected a point - the stimulus.

3) By measuring the Euclidian distance between stimulus and each of the neurons is determined the closest neuron - the winner and the second closest neurons - the neighbors.

4) The algorithm checks if the winner and the neighbor has already connection, if not, it creates it and sets age of the connection to 0.

5) The error value of the winner is updated by adding squared distance between the stimulus and the winner to the already existing error value.

6) The position of the winner and its topological neighbors is adapted by fraction of the distance to the stimulus.

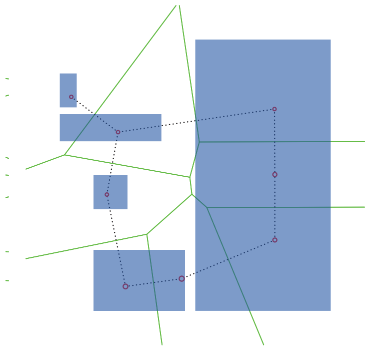
7) The age of all connecting synapses between the winner and the neighbors is updated incrementally.

8) If there are synapses with an age higher then the maximum allowed value, those synapses are removed. If there are any neurons without synapses, these neurons are removed

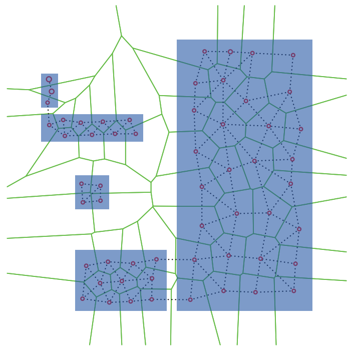
9) If the number of neurons is less than the allowed maximum of neurons a new neuron is inserted by:
a) A neuron with the largest error is found in the network
b) Neighbors of this neuron are located and the neighbor with its largest error is selected
c) New neuron is position between those two selected neurons and new edges are created between the new neuron and the selected neurons; the old edge is removed. The error of selected neurons is decreased by a fraction and error value is given to the new neuron.
d) The error of all neurons is decreased by a fraction.

5+ TESTING THE GROWING NEURAL NETWORK

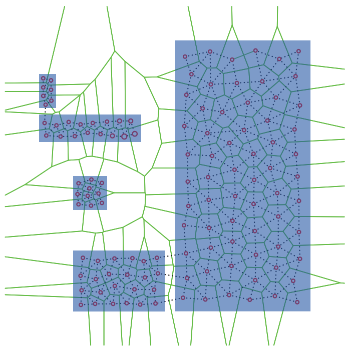
The constructed algorithm from the pseudocode is tested first in 2D dimension. The network samples stimuli from the rectangles. As already demonstrated by Fritzke, the network begins with positioning neuron to each of the rectangle and connecting all of them with the synapse. As the computation progresses, the algorithm adds more neurons inside the rectangles. Eventually, after number of iterations the neural network disconnects the synapses between the rectangle. The images also demonstrate non-induced Voronoi geometry.



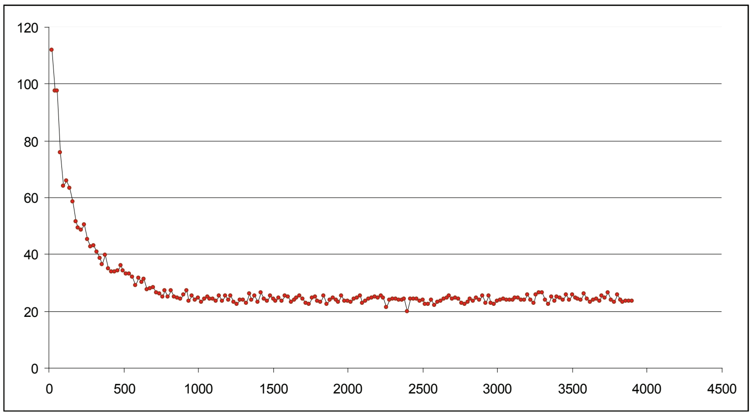
30 Iterations



160 Iterations

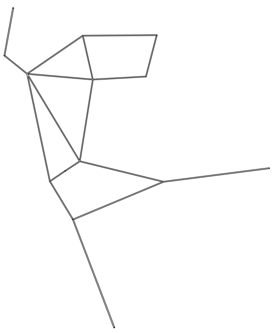


380 Iterations

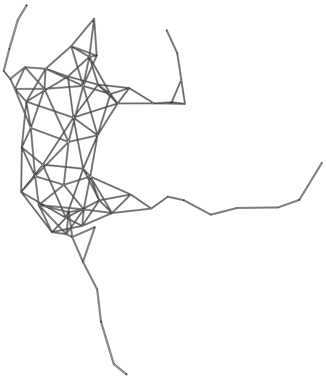


The graph demonstrates of learning curve of the network in the 2D. The curve is represented by decreasing average error value, which means that the network successfully adjusts its position close to the stimuli.

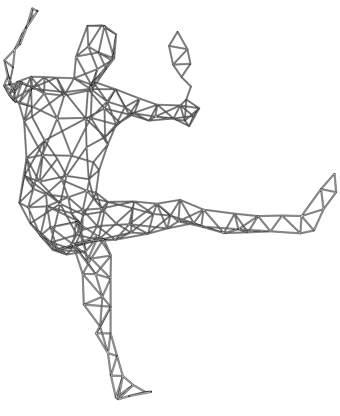
In the 3D environment the neural network is tested on learning topology from the imported point cloud of 20000 points body scan. The network successfully repeats the growing procedure as in the previous 2D model. The network first position neurons inside the body and after number of iteration the growing and adapting process repositions the topology correctly on the surface. Due to the large error are firstly generated large surface area as the torso and afterwards the grow continues to other smaller parts as legs and hands.



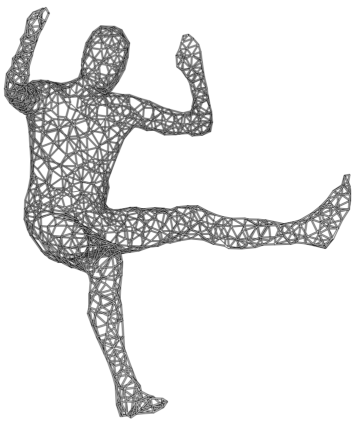
150 Iterations



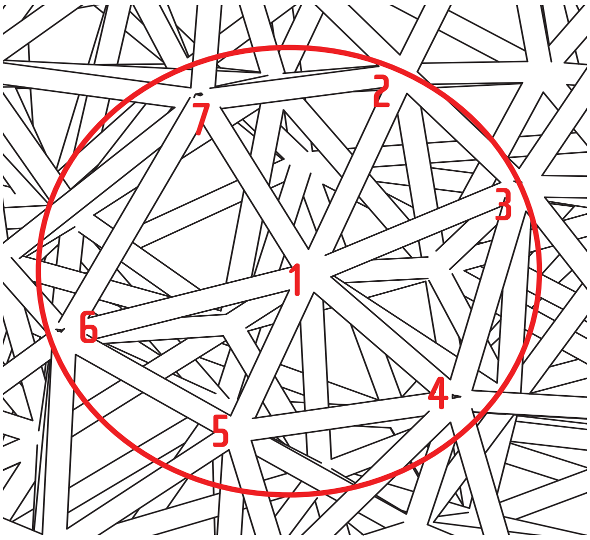
800 Iterations



3000 Iterations



10000 Iterations



Well distributed geometry
If 1 is the winner (closest to the stimuli), then 2,3,4,5,6,7 are the adjusted neighbors . The number of edges/members that meet at the vertex/ neuron are the valence. A single tetrahedron has a valence of 3 for each vertex. In this case the #1 is related to the neighbors with valence 6 while each neighbor is related to the winner and other selected neighbors with valence 3.

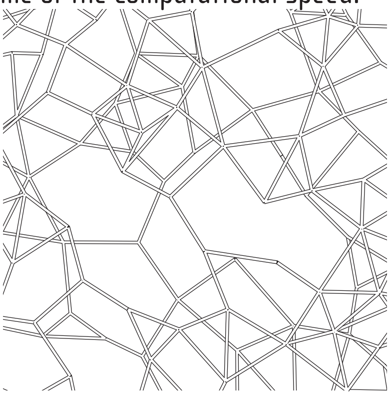
6.4 COMPARING MESH QUALITY FROM DIFFERENT POINT CLOUD RESOLUTION

One of the biggest advantages of the growing neural networks against other algorithms is the way it calculates its node placement. The neural network only samples the point cloud, using one randomly or sequentially selected point at a time. Therefore, its processing speed is not affected by the amount of input points.

In the chapter 5 was demonstrated 2D model, where the space was sampled evenly, and the neuron's distribution was also accordingly even, creating well distributed topology.

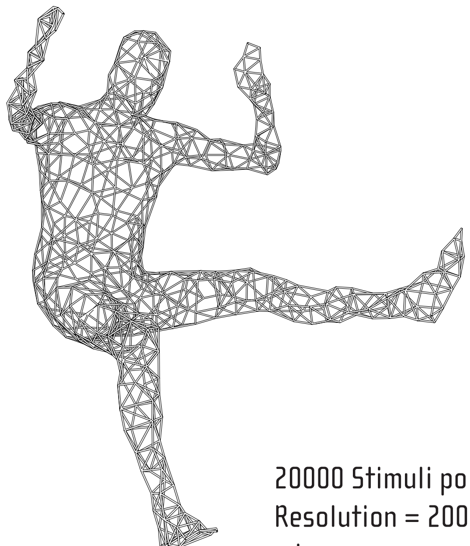
However, in the case of 3D model, the point cloud has uneven distances. Therefore, also in the neural mesh are appearing poor connections and negative artifacts in geometry definition. When comparing 2,000 and 20,000 nodes model, for the same number of neurons, the mesh from the 20,000 model shows much better quality then the 2,000 model.

As is visible, for the tested algorithm it is desirable to have as many points as possible, which improves the quality of mesh. Further on are demonstrated some thought on possible avoidance of those negative artifacts and creating more even mesh distribution from a low resolution point cloud, however, those adjustments of the neural network algorithm are for the price of reduction of some of the computational speed.



Artifacts in mesh generated from 2000 Stimuli points

2000 Stimuli points
Resolution = 20000 edges



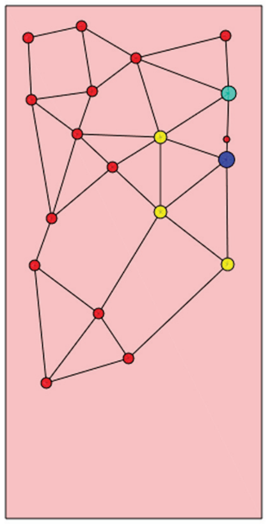
20000 Stimuli points
Resolution = 20000 edges

7+ ADJUSTING GEOMETRY INTO PREDEFINED SIZES

Because the nature of the incremental neural network is to constantly operate with an error, each member of the structure has a different length even during dense stimuli distribution. However, for the construction purposes we want to have a limited number of different member sizes. Therefore, we have to add constrains to adjust each member into one of the allowed sizes.

The main problem of using any constraining parameters with growing neural network is that it directly affects the node placement and the mapping precision of the algorithm . As already described, the algorithm functions on placing neurons based on selecting the highest error factor, in this case the Euclidian distance between the neuron and the stimulus. If any constraining parameters are added during the incremental process, it limits the network's flexibility needed for correct learning and leads to fatal deviations and wrong node placement.

As is seen in this example, the neural network was constrained to lock each of the members into one of the five predefined positions during the growth. The restricted flexibility of the neural network resulted in partial and poor mapping of the stimuli area, for the algorithm is being unable to readjust the neurons correctly.

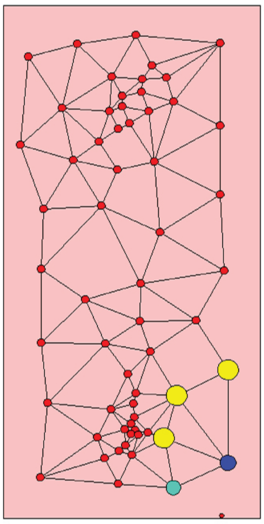


Thus, it is necessary to grow the geometry first to reach the desirable number of neurons or the average length and then continue the algorithm by replacing the operating error of the distance between neuron and stimulus with the error of the distance of current position from the wanted position of the neuron to reach the desired length.

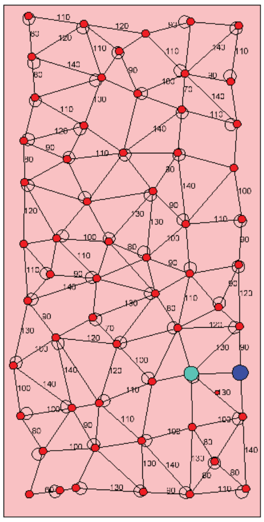
7.1+ THE TUNING PROCESS

The adjustment of the lengths referring further to as the "Tuning Process" consists of two main stages:

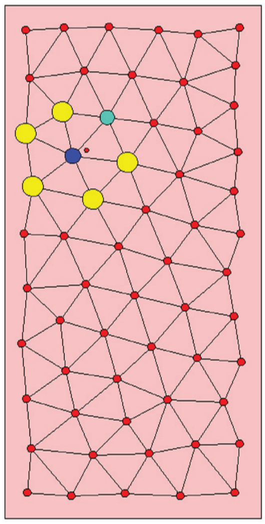
#1 - Setting the travel distance of adjustment of a neuron towards the input affects how well the networks distributes the neurons. If the travel distance is too small the network does not spread well and the neurons are close to each other. However, if the travel distance is set high, the network distributes well, but in the case of a well distributed point cloud the range from its average length is greater; the neuron does not stabilize itself, but travel constantly between surrounding signals. For our purpose is desirable to switch from high to low travel distance during the process. The algorithm starts with high distance and when it reaches the desirable maximum amount of neurons, it switches to low distance. Because at that point the network is already well distributed on the point cloud, the high travel distance is no longer necessary and the low distance travel causes the network to spread more uniformly its existing neurons. That narrows the ratio from the average length.



Poor distribution
low travel distance



Correct distribution
before tuning



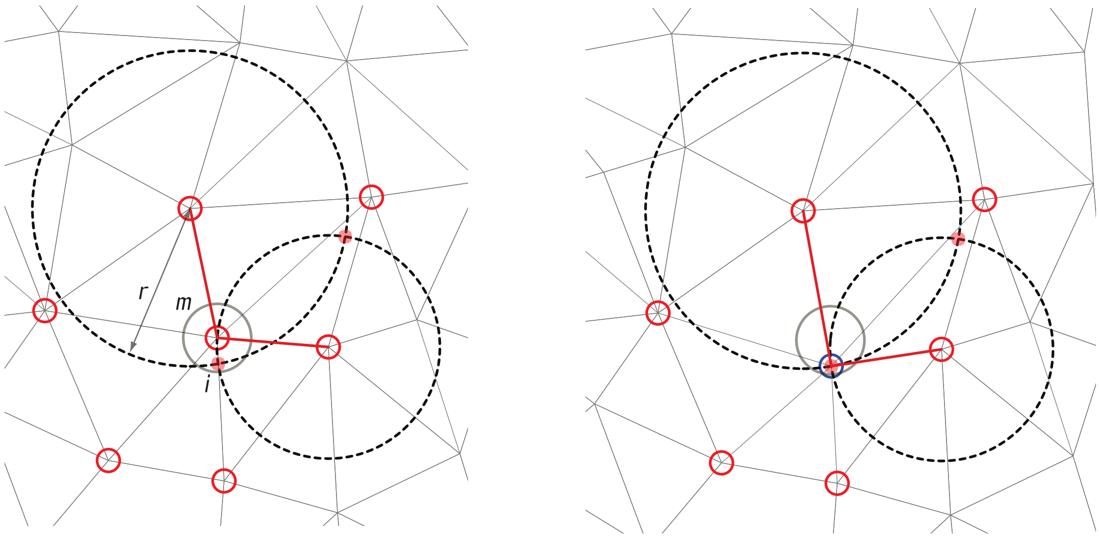
Even distribution
after tuning phase 1

II 2-- Locking into desired lengths

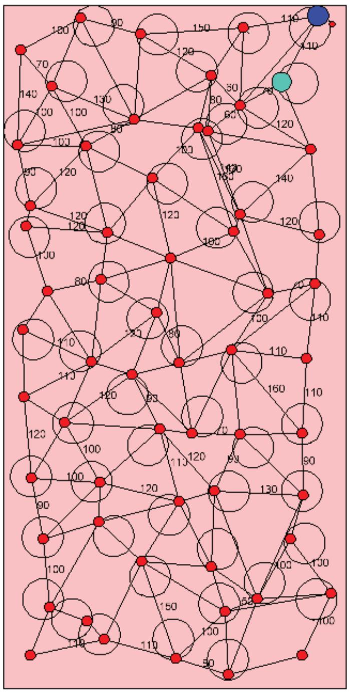
In the second phase the structure changes the adapting of its members/synopses into predefined members of standard sizes. The algorithm works in following steps:

a) The reaction of neural network to the stimuli is interrupted, therefore the geometry no longer reacts to the point cloud. The current position of all the neurons is stored in an array.
b) For the best results and the smallest deviation of the mesh, the average length of all of the member sizes is calculated which is considered to be 100 units. If the stopping mechanism of the growth was based on reaching the wanted size of the average length and not the maximum number of neurons, the lengths of all the members were already defined before the growth.
c) Each of the structural members are categorized into predefined dimensional ranges, based on the closest deviation from one of the desirable lengths. Each category are based on the range between the middle value of the closest lower length and the middle value of the closest higher length.
d) The standard maximum allowed deviation is calculated, that is how much is a neuron allowed to distant itself from the original position. The value of maximum error deviation is determined by the percentage from the average length. In 2D the deviation represents the radius for the circular area of allowable movement of neuron from its original position, in 3D the neurons allowable area is given by spherical volume with its center on the origin.
e) A sweeping algorithm selects a random neuron (the winner) and finds its closest connecting neurons in the valence (neighbors). From the neighbors are chosen randomly two neurons and their distance from the winner is checked. If both of the lengths have already wanted size,

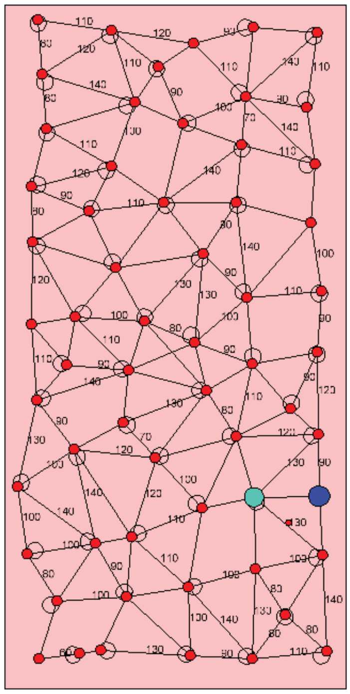
another neuron in the valence is selected and checked again. IF one the lengths deviates from the determined length, its closest locking length is selected from the catalog
f) Algorithm calculates the displacement of the center of the valence in such a way that only one member adjusts its length into desired one while the length of the second member remains unchanged. The calculation consists of finding intersections of two circles in the 2D, two spheres in the 3D scenario. One of the radius is unchanged member length, the second is the locking length.
g) algorithm checks the error deviation between the winner's stored origin and the closest found intersection - if the needed change of position distances neuron from its origin less than maximum allowed deviation, the neuron is positioned
- if the needed position is more than the allowed deviation, the algorithm returns back to f) and selects the lower cataloged length, and repeats finding the intersection with the new length. if this point of intersection is in the allowed deviation range, the neuron is positioned by adjusting the length randomly through out the structure, the off-lengths are better distributed.



r = radius of the closest desired member size
m = area of maximum allowable movement from origin
i = successful closest intersection < m radius



Deformation caused by too large maximum error (negative folds)



Optimal maximum error

By implementing the maximum allowed movement of the neuron the error distance is controlled. In the experiment it proved that by not controlling the error distance resulted in destroying the original shape with little significant further improvement of length adjustment.

Importantly, during the tuning the aging process as well as the Hebbian learning is left turned on. This removes negative artifacts - folds, which appear during the tuning process. Since the average length is stored before the tuning, its value remains unchanged.

7.2+ EFFICIENCY

To see the efficiency of the tuning, the process is repeated under same conditions with several different length sizes :
The maximum number of neurons = 60
Number of samples = 50

Lengths (#)	Maximum Allowed Distance	Adjusted members(#)	Remained members(#)	Success (%)
25	15	101	38	72.7
13	25	84	55	60.4
7	30	71	68	51
5	35	67	72	48.2
3	45	61	78	43.9

As can be seen from the table, the efficiency increases with more given lengths. Also the deviation from the original shape decreases. With 25 different length option, the mesh adjusts at around 70% of success. The maximum allowable distance was adequately changed according to number of lengths because less number of sizes need to travel greater distance to lock into positions than higher number of lengths.

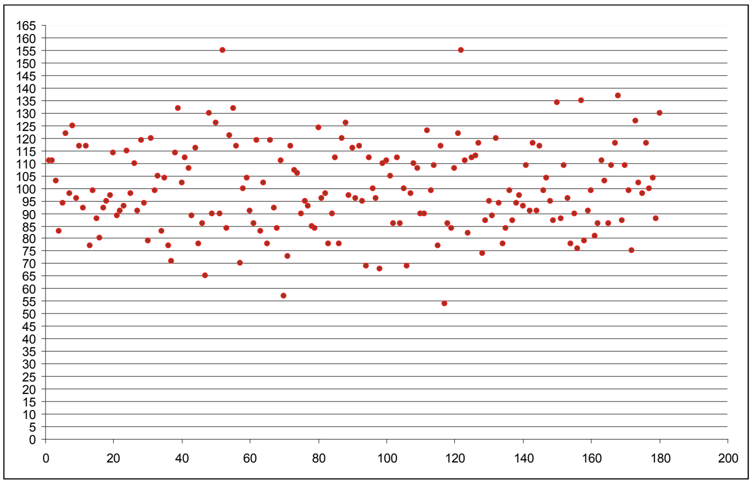
The experiment is repeated in 3D environment to compare the tuning success.
As a point cloud is used the head from Anthony's Gormley's body scan. The point cloud has 1546 points with maximum distance between two points and of minimum distance between two points.

The maximum number of neurons = 60
Number of samples = 50

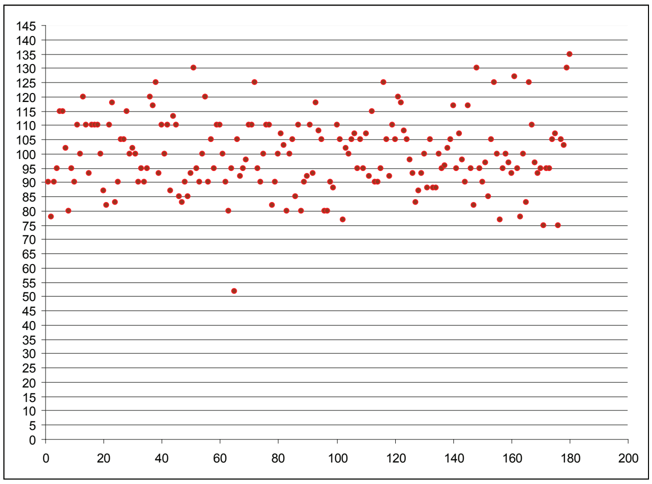
Lengths (#)	Maximum Allowed Distance	Adjusted members(#)	Remained members(#)	Success (%)
25	15	109	49	69
13	25	76	79	49
7	30	69	89	43.6
5	35	65	93	41
3	45	61	97	38.4

The results show lower success than in the 2D model. This is caused by less even neuron distribution over the point cloud, and its more difficult for the algorithm to lock into the positions. The 25 member sizes show still around 70 % while the other, less member sizes drop rapidly; the 13 member sizes adjusts only around 50 % of the total lengths.

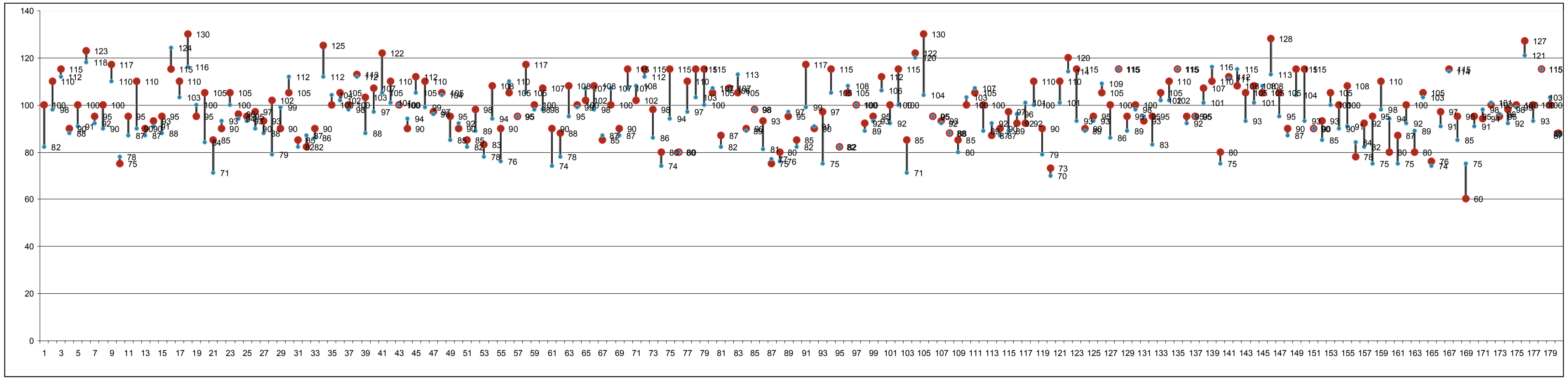
Comparing lengths sizes before and after tuning, the adjusted sizes show more even distribution of the point cloud.



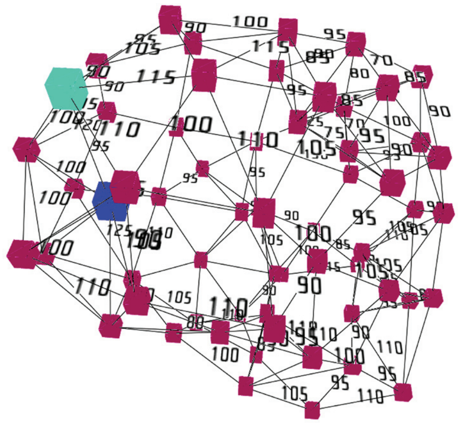
Member Lengths before tuning (25 sizes)



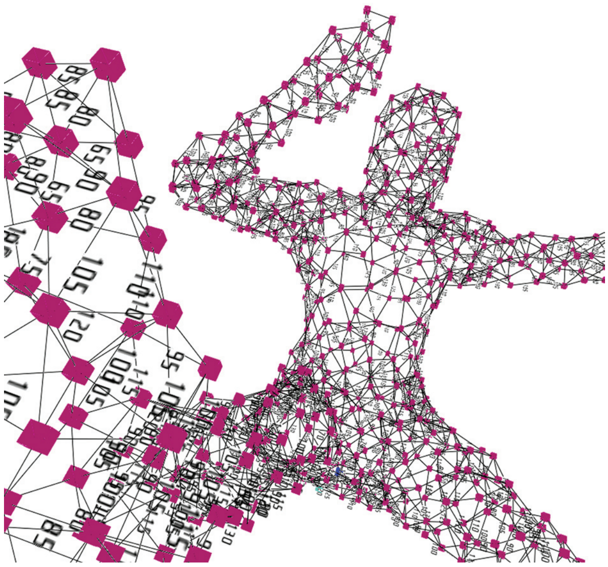
Member Lengths after tuning (25 sizes, 66% successful adjustment)



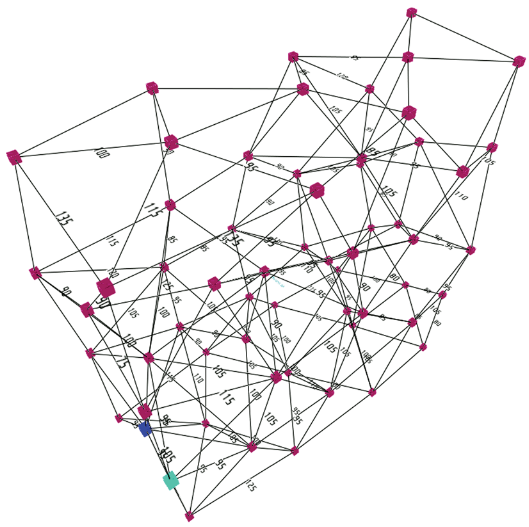
Graph represents deviation of neurons from its original positions. The x axe is the number of neurons, the y axe is the length of the members. The Blue dots are the original position of the neurons and the Red dots are the positions of neurons after the adjustment of sizes



Resulted Adjusted Geometry (60 neurons, 25 standard members adjustment)



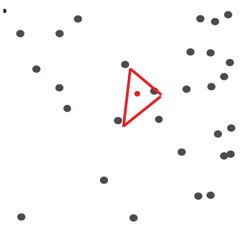
Using the whole body, with maximum of neurons 800, with 25 different member sizes the number of adjusted lengths reaches 63%.



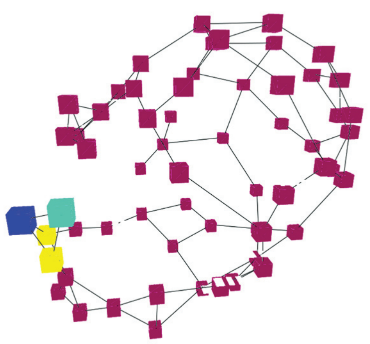
In the experiment of filling a volume, the stimuli have the same rectangular base as in the 2D model, but with added height dimension. the network fills cubical space. The maximum neurons is 60. With 25 different length sizes the number of adjusted members reaches 64%.

8+ IMPROVING MESH DISTRIBUTION IN UNEVEN INPUT POINT CLOUD

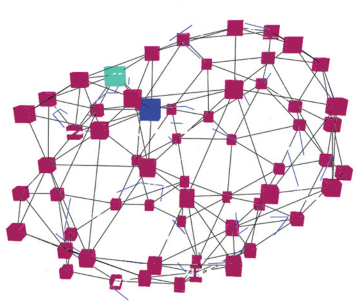
In chapter 6 was compared mesh quality for different densities of point clouds. The results demonstrated of problematic artifacts occurring with the lower resolution (2000 stimuli) point cloud. This modification of the neural algorithm tests the possibility of better distribution for neurons. When a stimulus is selected, the algorithm uses now the same procedure as it uses for the Hebbian learning to find the two closest neighboring stimuli. By connecting these three neurons with edges, it creates triangulated boundary. Next, the winning stimulus is randomly repositioned inside the area. Now the neural network operates with higher constant error which would result with better distribution of the geometry. In addition, because the stimulus is constantly moving on the surface are, the neural network still correctly creates the geometry from the point cloud. The negative side effect of this adjustment to the algorithm is that the algorithm must sweep through all the points to find the neighbors, which of course causes loosing some processing speed. Possibility would be that the algorithm check the distances of each of the point its neighbors at the beginning of the grow and selects only those which are too distant from the others, finds the centers and adds the points to the array of original points.



As seen the effectiveness in the visual comparison, with maximum of 60 neurons, first the regular algorithm was used. Afterwards with the same amount of neurons was created geometry with the adjusted algorithm. The unadjusted algorithm poorly created the mesh with only 91 members. The adjusted algorithm performed significantly better with 173 connecting members. However, comparing the adjustment of members to 5 different defined sized the poor distributed mesh adjusted successfully 51% of its member while the other mesh only 44%. The reason for this is that the poor distributed networks contains disconnected members, which adjustment does not affect the others as in the better distributed network.



Mesh distribution with original algorithm



Mesh distribution with modified algorithm

9. VERSATILITY OF ALGORITHM

The last part of the thesis elaborates on testing flexible possibilities of the growing neural network algorithm and testing it with the locking algorithm, several experiments are demonstrated:

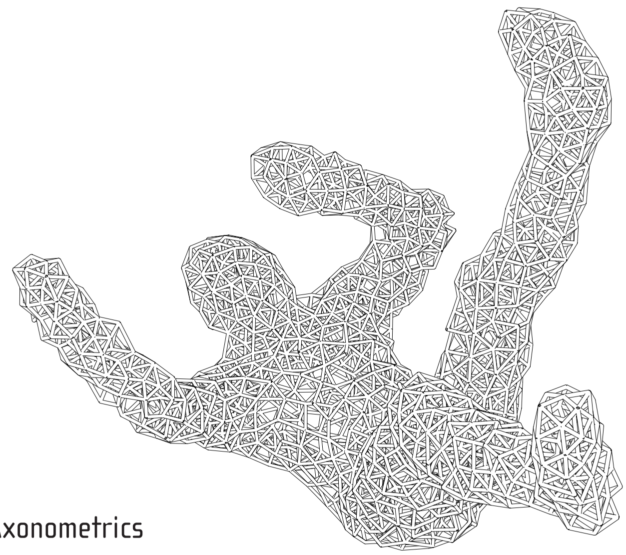
1) Surface Offset
2) Body Volume
3) combination of intersecting point clouds
4) moving neural network

9.1+ SURFACE OFFSET

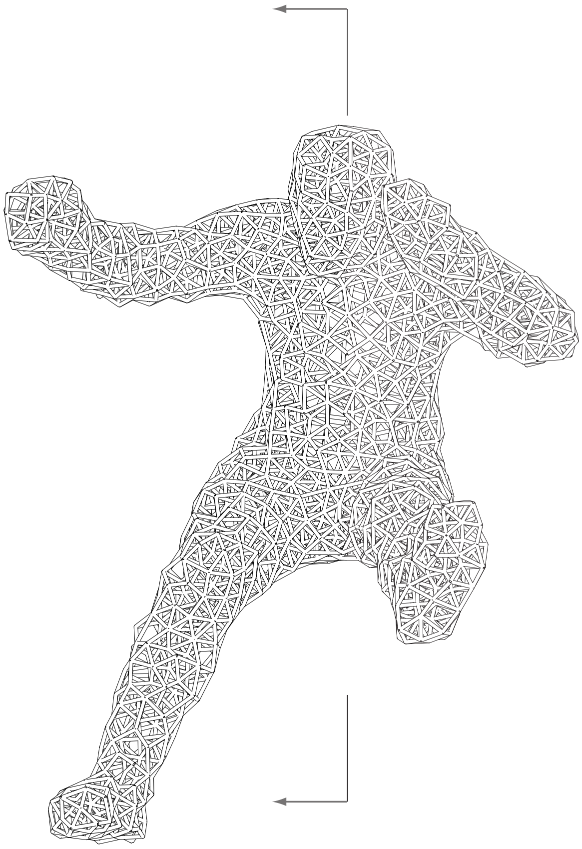
This example demonstrates the ability of incremental neural network to build a layered surface. As seen before, the neural network successfully created single surface mesh from the point cloud, and now the task is to give the point cloud offset. This results into creating additional layers of points evenly outside and inside of the original point cloud. In the demonstrated case this error is constant for each point. The solution is to give each of the sampled points a volumetric error, that means that the stimuli is no longer static, but its position changes inside of a constraining volume.   However, it opens up a possibility to vary this error from point to point, based on some construction aspects, such as local stiffness.

In the process of adaptation, the neural network firstly creates a single layer surface, as in the previous examples. However, when the skin is evenly distributed and the error distance between neurons is less than the error distance of the maximum offset, the networks starts to create additional layers of neurons. The layer starts this to be built up from the hand and legs, which were the last to be built for the first layer.

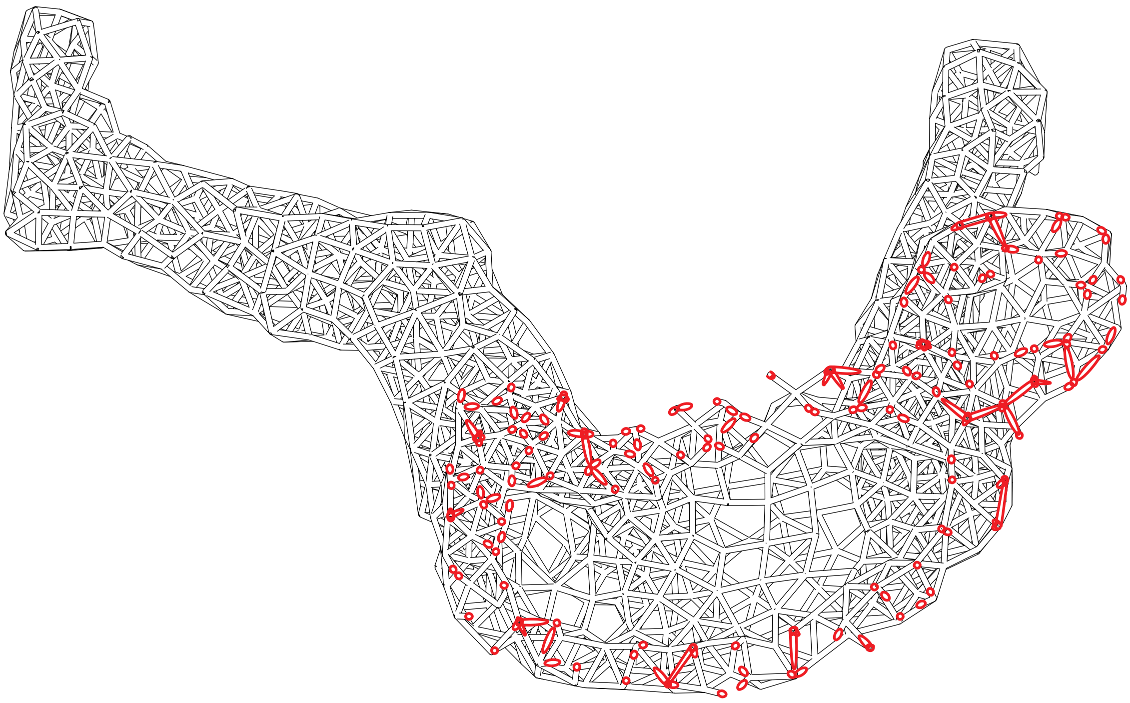
The images shows of resulted network of 2000. After adjusting its member sizes in 25 different length the adjustment demonstrated 64% success. This is given by even distribution of neurons and because only two layers were created, the network has enough flexibility for the adjustment .



Axonometrics



Elevation

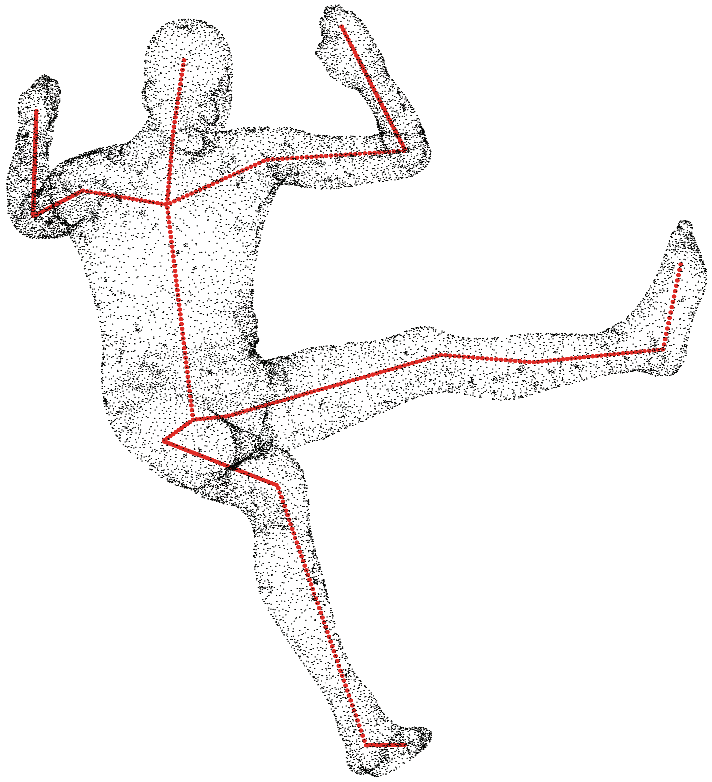


Section of the Resulted Geometry

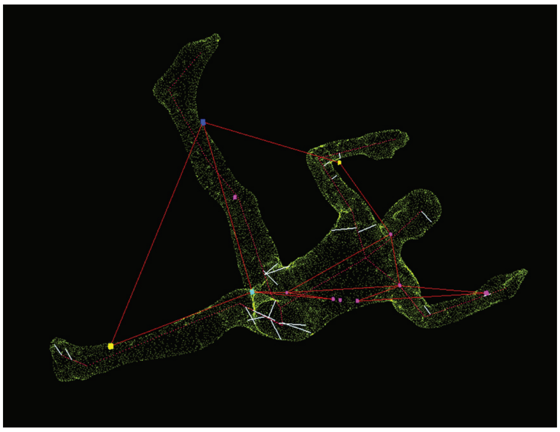
9.2+ VOLUME

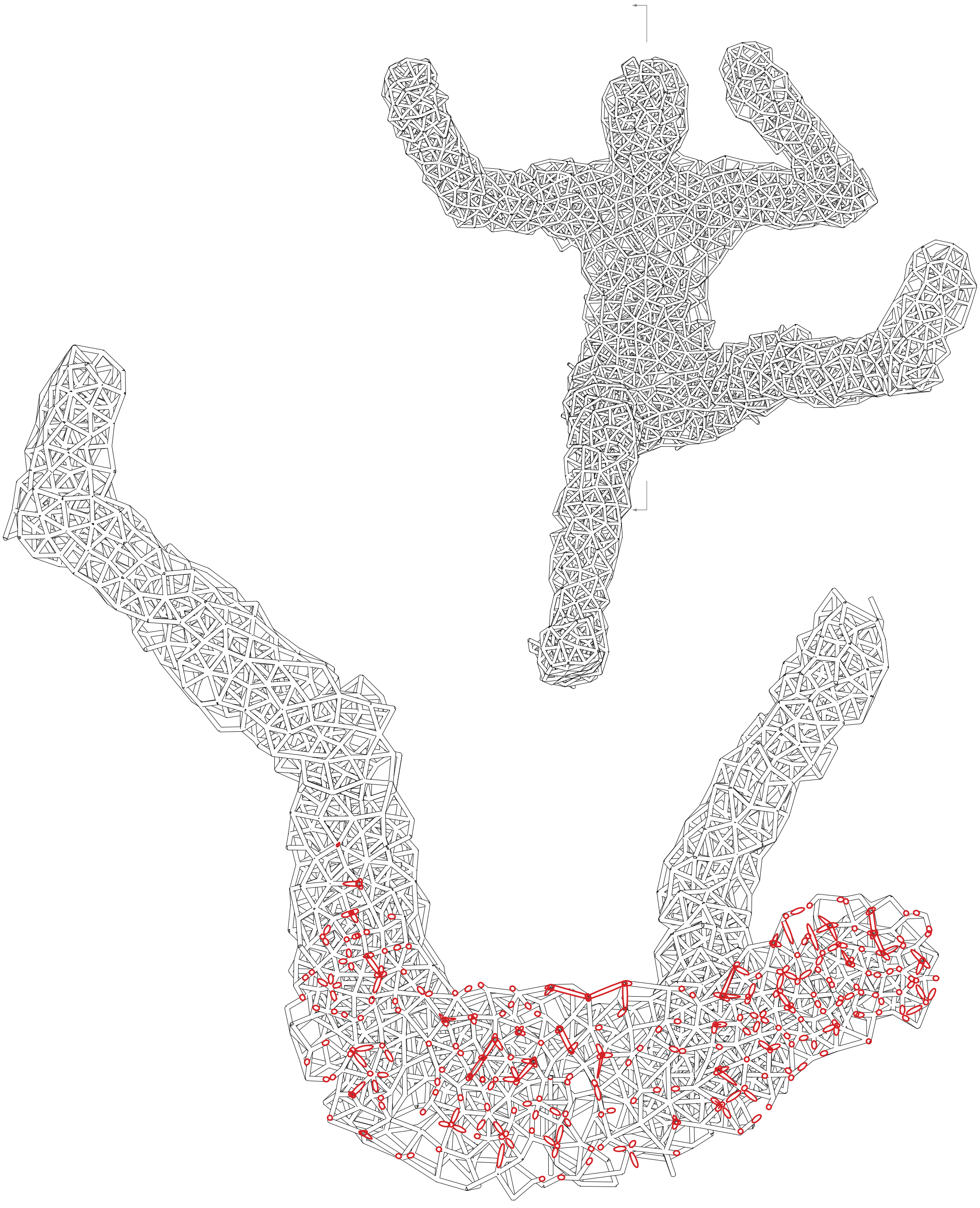
This example demonstrates neural network to fill volume of the obtained body scan. Because only surface points are known it is a rather difficult task. Therefore, the network needs additional method how to find the inside space. This can be done with several methods; however, they all involve inserting some kind of scanning mechanism, which results in fatal loosing processing speed, if the volume is calculated simultaneously with the growth. To calculate inside points during the growth without significant lost of speed tests the following method. The tested body scan has not inserted orientation lines, serious of lines running in the center of the torso, legs and arms, similar to the way of creating animation; each line is evenly divided into segments. When a random surface point is selected the same way as before, however, now the closest segment on the skeleton is found. These two points are connected by a tubular boundaries for the stimulus. A random point is selected from the inside of the tube which is now the stimulus. Because only several points are needed to form the skeleton, this process of finding the closest point does not significantly slow the growth process.

The demonstrated network consists of 4000 neurons. After adjusting its member sizes in 25 different length the adjustment demonstrated 53% success because the adjustment of the network is now restricted by large interconnected number of neurons.



Axonometrics of point cloud with inserted skeleton points



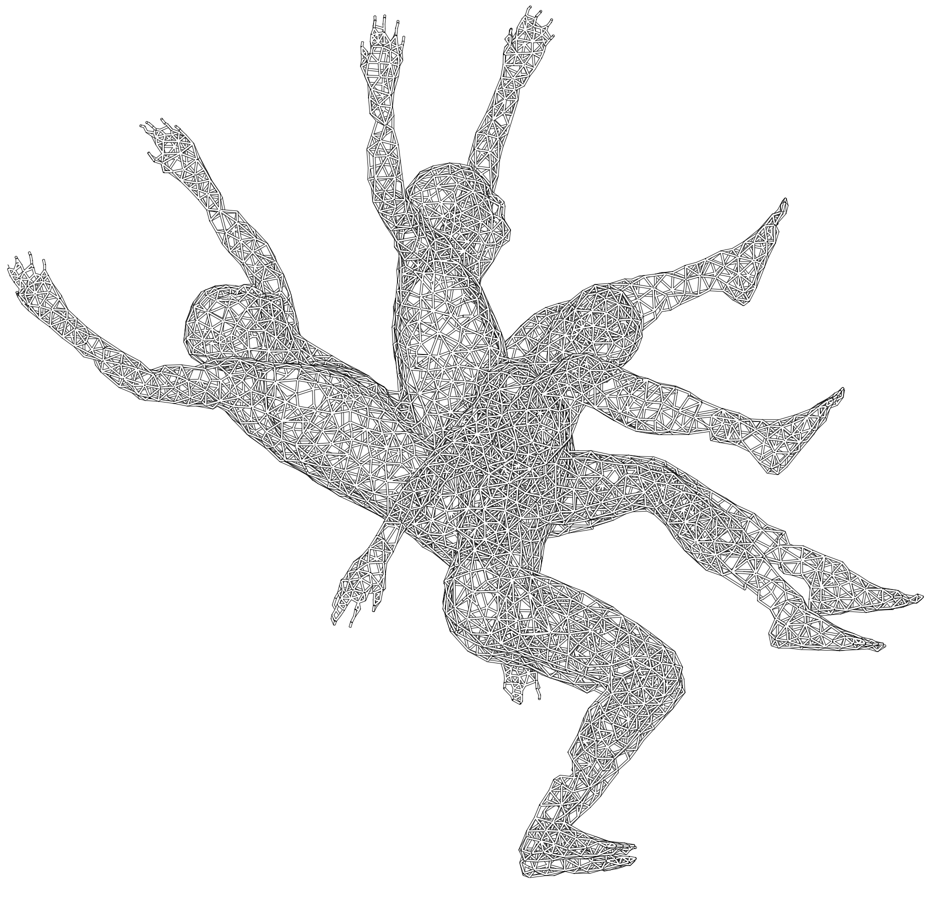


Section of the Resulted Geometry

9.3+ INTERSECTING SEVERAL POINT CLOUDS

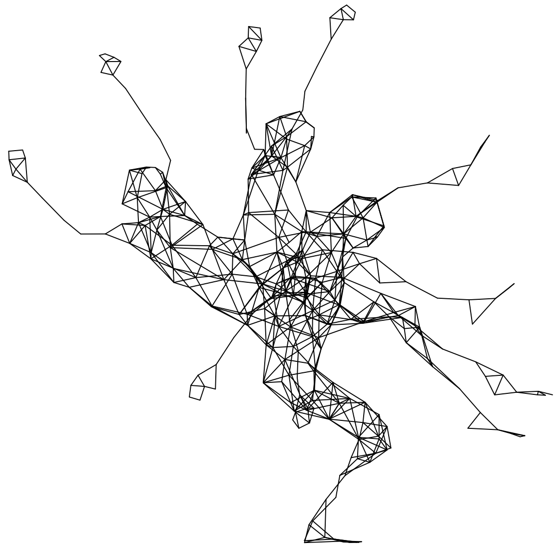
This example demonstrates testing of resulting mesh from combination of intersecting different point clouds. For the lack of variety of body scans were used point clouds created from a 3d model. The model of a male body was create inside of modeling software and animated. The polygonal geometry of several sequencing positions of the models was converted into point cloud. The neural network samples as previously random points from all of the point clouds at once. The resulting mesh demonstrates correct mapping to each of the individual point clouds. However, the interesting results appear at the intersections between the point clouds. Here the neural network fluidly and evenly connects the point clouds. This is a remarkable result, for it is very difficult to connect protruding geometries in such way manually or with some techniques in modeling software as is the boolean.

The demonstrated network consists of 3000 neurons. After adjusting its member sizes in 25 different length the adjustment demonstrated 66% success.

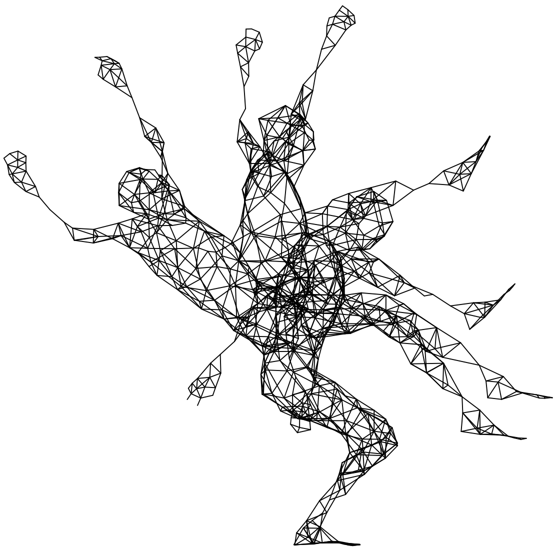


Elevation

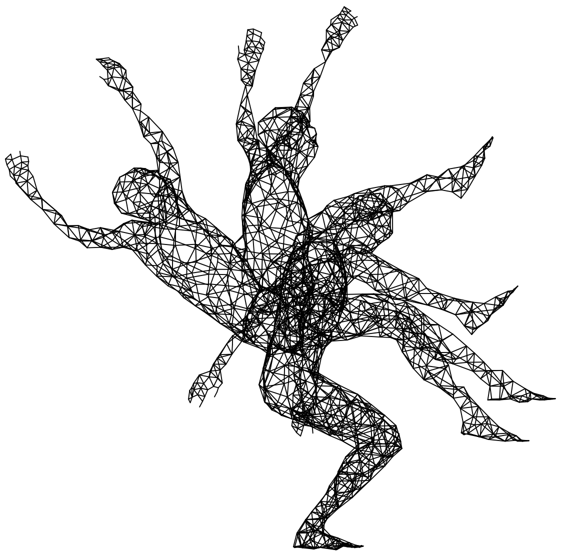
Elevation
300 iterations



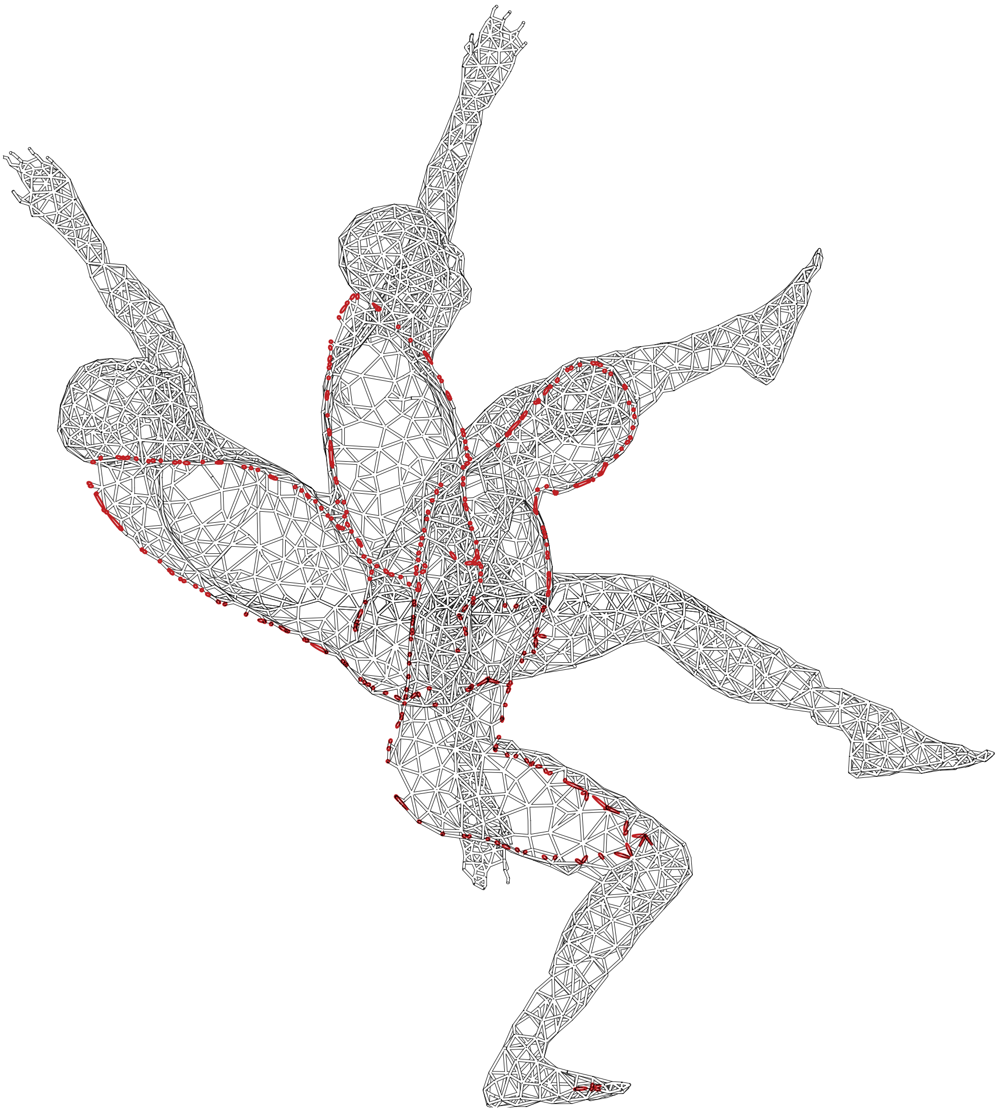
Elevation
1200 iterations



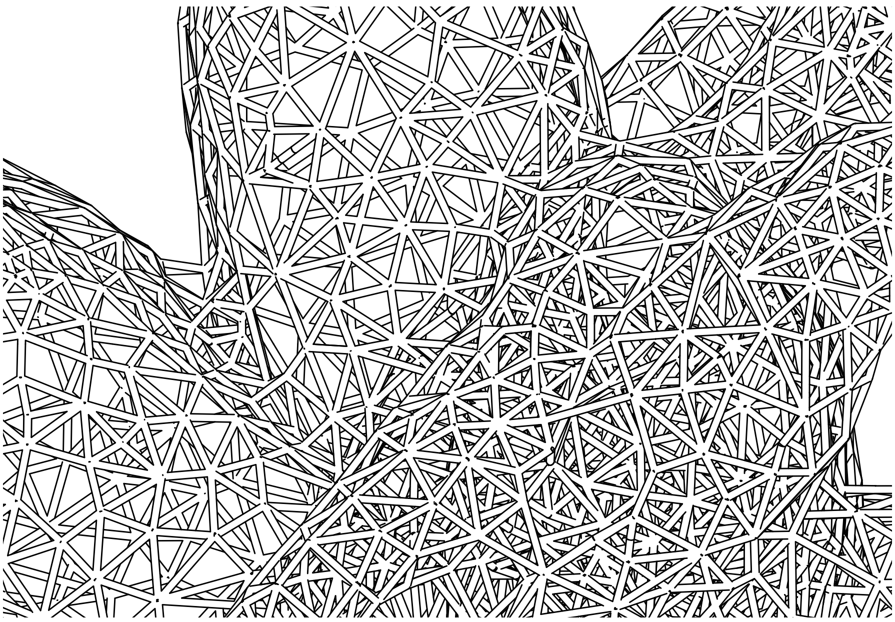
Elevation
4000 iterations



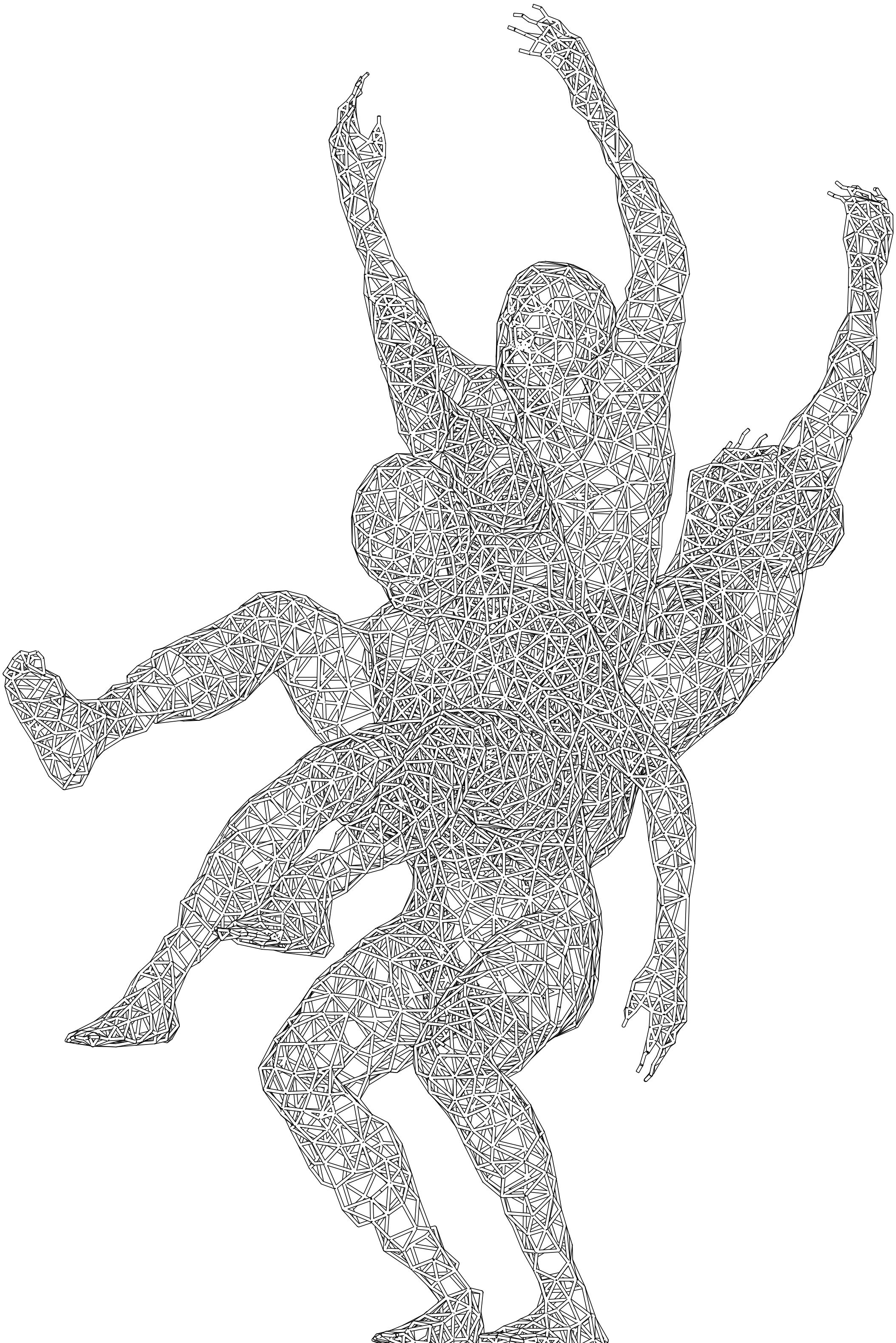
Elevation
15000 iterations



Section of the Resulted Geometry



Detail of the Intersections



Axonometrics of neural network structure

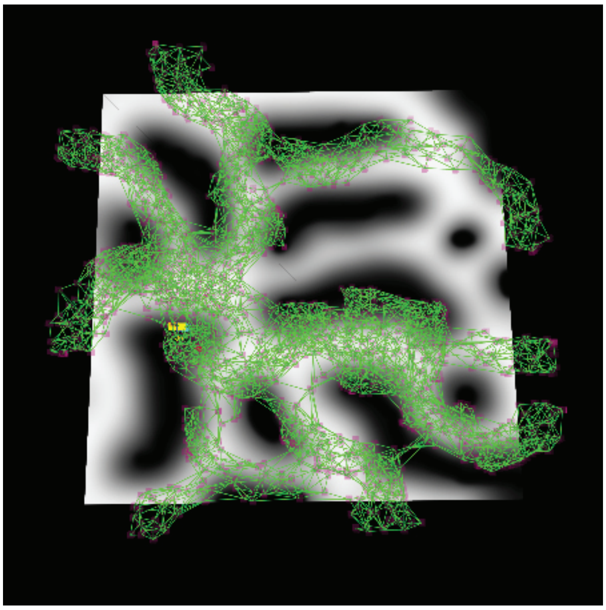
4000 neurons

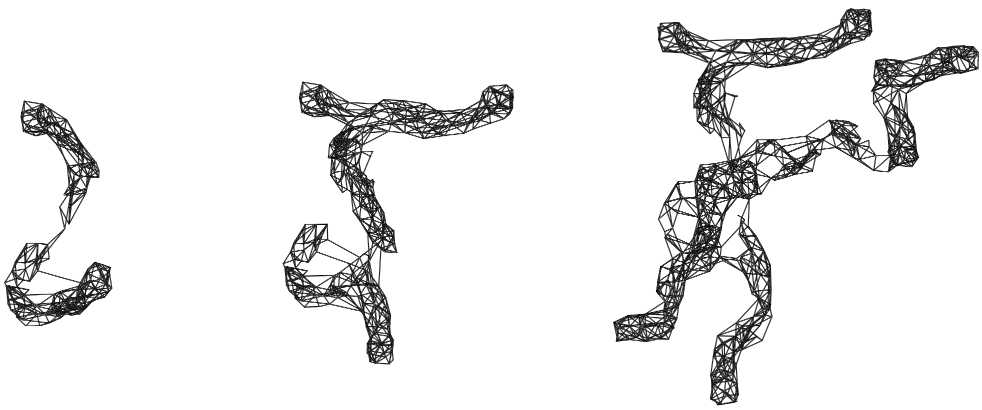
Length adjustment of members to 25 different sizes = 64% success

9.4 NEURAL NETWORK AND WALKING AGENT

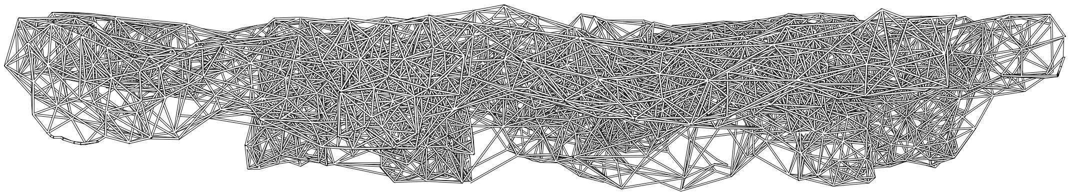
This last experiment demonstrates possibility of combination of the tested neural algorithm with other self-organizing systems. In the model the neural network reads stimuli from volumetric area - a box. The box itself represents Braitenberg vehicle/agent. The agent by itself receives stimuli from the environment in the form of changing light intensity, in this case represented by the value of the pixels of the image bellow the agent. The neural network has been constantly being stimulated by the travelling agent. Because its position changes the created neural network moves out of possible adjustment for the algorithm, therefore leaving a trail behind the agent. However, when the agent intersects its former path, the neural network in that area reconnects the network to the one currently created. Over a period of time, the agent creates geometry based on its own movement. The density of the neural network is affected also by the agent's constantly changing speed.

Resulted neural network driven by the agent. The agent's speed and direction was a responce from stimulation by the pixel's intensity of the image, which also affected the distribution of the neural network in the area.

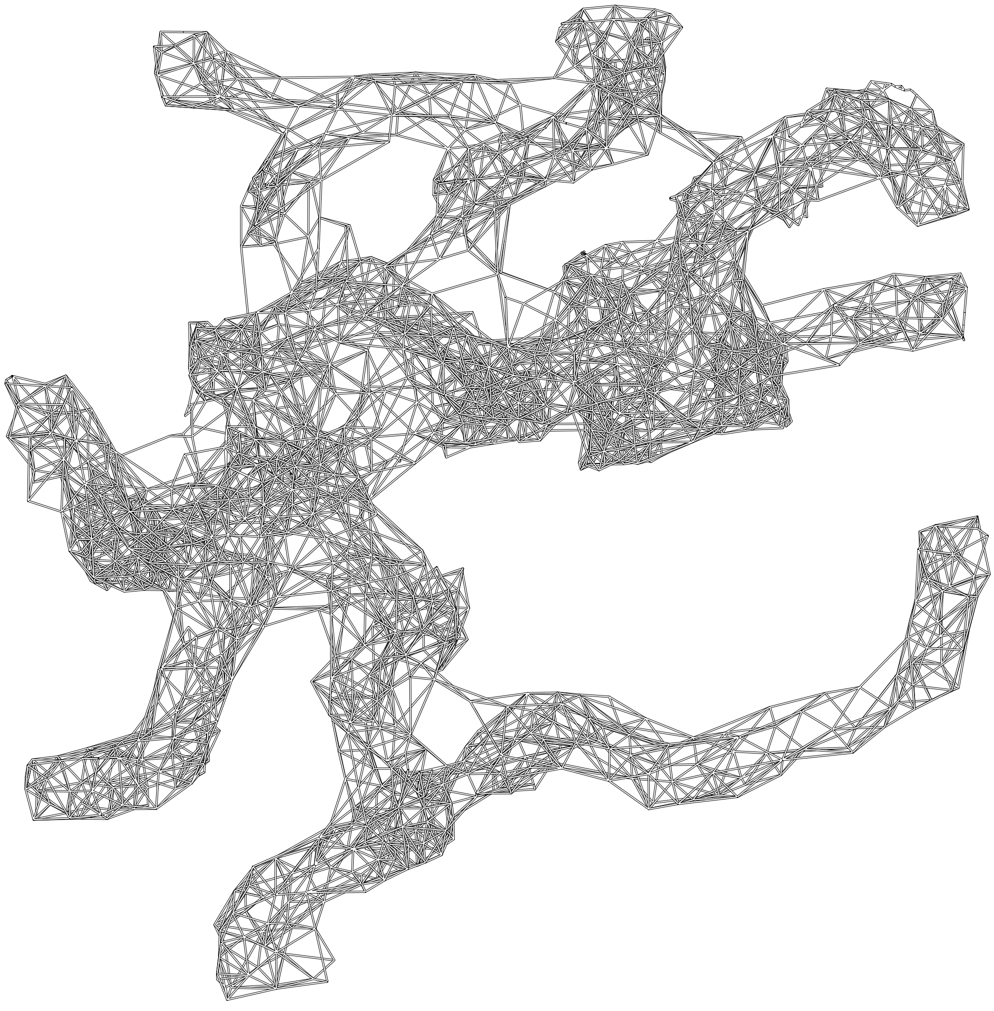




200 iterations 600 iterations 2000 iterations



Elevation



Axonometrics

10+ DISCUSSION

After creating and testing the algorithm for rationalizing the unorganized point clouds, the results can be now reviewed in overall spectrum.

10.1+ RATIONALIZING GEOMETRY FROM UNORGANIZED POINT CLOUD

Regarding the first question of the hypothesis concerning the rationalizing of unorganized point clouds such as 3D scans, the results show to be satisfactory. The adjusted incrementally growing neural network correctly recreates the scanned geometry. The speed of the algorithm proved to be very satisfactory, having practically no upper limit from the size of the input data. Even more, the higher is the resolution of the scan, the better and more even is the resulted geometry of the neural network. Being able to process unlimited amount of data without a bottleneck is an important knowledge, as presently 3D scans produce very high resolution of millions of points. On the other hand a poor resolution results in poor mesh geometry. One demonstrated possibility of successful solution to this problem was to adjust the algorithm to find additional stimuli points on the surface. However, this method has a set back of loosing processing speed, value which would be adequate to the size of the point cloud.

10.2 ADJUSTMENT OF MEMBER SIZES

The second task to improve construction possibilities from the point cloud proved also successful. The additional algorithm to the neural network demonstrated between 50-70 percents success in the adjustment of members, depending on the chosen amount of standard lengths and the form of the geometry. The results were more successful in the cases of surface geometry then in the volumetric. The reason for this is the more constraining topology of layered network containing more connections. As a tool for the manufacturing purposes, the designer can chose the number of different amount of lengths which affects the deviation error from the unadjusted geometry. More lengths results for less needed readjustment of original topology and better results of adjustment. For the practicality of the use for the construction the fabricator can export the sorted lengths as a text file and the geometry with correlated numbered members as a DXF file.

10.3 VERSATILITY OF ALGORITHM

Finally, the complete algorithm had been tested for different construction tasks. Besides recreating the surface geometry, the neural network also demonstrated successfully offset capabilities. With some adjustment to the algorithm, the neural network was also able to recognize and create volume without adequately loosing computational speed. When testing creating geometry from several intersecting point clouds simultaneously, the neural network revealed remarkable ability to weld the intersections, a task practically very difficult to obtain through any other process. Furthermore, the algorithm proved to be capable of connection with other mapping methods, such as the demonstrated connection with the walking agent.

11+ FURTHER DEVELOPMENT

The concern for the future development is to improve furthermore the efficiency of the adjustment of the geometry. The adjustment didn't exceed 70% of the success from all the members. This was given by inability of the algorithm to calculate more than one step ahead for the next member selection for the adjustment. This results into setting already adjusted members out of alignment. Because the operation of the alignment is constant, one of the possible solutions would be to implement a constrain solver, which would calculate several steps of optimal selection in advance.

12+ CONCLUSION

In the overall research for a suitable neural algorithm, first was tested the Kohonen network. The Kohonen self-organizing map demonstrated suitability with its ability to adapt its form depending on the shape of the point cloud, however, its possible usage was constrained by the predefined topology. On the contrary, the after tested Hebbian learning network showed self-organizing characteristics of its topology, but had the inability to adjusts position of its neurons. Finally, In the incrementally growing network were found positive characteristics of both neural networks. The growing neural network showed very positive results in reconstruction of geometry from the point cloud. The neural network was tested with different point clouds of different densities, revealing dependency of well articulated mesh upon high density of input point. Later was demonstrated a method of adjustment of the algorithm to better distribute the network in case of inputs from uneven point clouds. The algorithm was further adapted to adjust the sizes of the members of the geometry into predefined limited number of different sizes. Several different amounts of sizes were tested with results of up to 70% of successful adjustment of members of the neural network. The neural network were also tested for its versatility to solve different problems of constructions, requiring some adjustments in the algorithm for each task. Demonstrated was generation of offset from the surface, filling volume of enclosed point cloud, creating geometry from several intersecting point clouds and connection of neural network with walking agent.

13+ BIBLIOGRAPHY

Fritzke,B. *A growing neural gas network learns topologies*. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems 7, pages 625-632. MIT Press, Cambridge MA, 1995.

Fritzke,B. *Growing cell structures - a self-organizing network for unsupervised and supervised learning.* Neural Networks, 7(9):-1460, 1994.

Gerstner,W. *Mathematical formulations of Hebbian learning.* Swiss Federal Institute of Technology Lausanne, Laboratory of Computational Neuroscience,2002.

Hanna,S. *Body / Space / Frame*. http://www.sean.hanna.net/bodyspaceframe.htm.

Kohonen,T. *Self-Organization and Associative Memory*.Springer-Verlag, New York, 1984.

McCulloch, W., and Pitts, W. *A logical calculus of the ideas immanent in nervous activity*.
Bulletin for Mathematical Biophysics, 5, 115-133. 1943.

Oja,E. Samuel,K. *Kohonen Maps*. Elsevier, New York, 2003.

Otto,F. Rasch,B., *Finding Form.* Deutscher Werkbund Bayern, Germany,2001.

Rescorla,R. *Behavioral Studies of Associative Learning in Animals*. Department of Psychology, University of Pennsylvania, Philadelphia,Pennsylvania,1982.

Thompson, D.W., On *Growth and Form*, Cambridge University Press, Cambridge, UK, 1942.