A New Approach to Cluster Visualization Methods Based on Self-Organizing Maps

Marcin Z.
Department of Computer Science Chemnitz, University of Technology Chemnitz, Germany
marcin.zimniak@cs.tu-chemnitz.de

Johannes F.
Department of Computer Science Chemnitz, University of Technology Chemnitz, Germany
johannes.fliege@cs.tu-chemnitz.de

Wolfgang B.
Department of Computer Science Chemnitz, University of Technology Chemnitz, Germany
wolfgang.benn@cs.tu-chemnitz.de

Abstract
The Self-Organizing Map (SOM) is one of the artificial neural networks that perform vector quantization and vector projection simultaneously. Due to this characteristic, a SOM can be visualized twice: through the output space, which means considering the vector projection perspective, and through the input data space, emphasizing the vector quantization process. This paper aims at the idea of presenting high-dimensional clusters that are 'disjoint objects' as groups of pairwise disjoint simple geometrical objects – like 3D-spheres for instance. We expand current cluster visualization methods to gain better overview and insight into the existing clusters. We analyze the classical SOM model, insisting on the topographic product as a measure of degree of topology preservation and treat that measure as a judge tool for admissible neural net dimension in dimension reduction process. To achieve better performance and more precise results we use the SOM batch algorithm with toroidal topology. Finally, a software solution of the approach for mobile devices like iPad is presented.

Keywords: *Self- organizing maps (SOM); topology preservation; clustering; data-visualisation; dimension reduction; data-mining.*

1 INTRODUCTION

Neural maps are biologically inspired data representations that combine aspects of vector quantization with the property of function continuity. Self-Organizing Maps (SOMs) have been successfully applied as a tool for visualization, for clustering of multidimensional datasets, for image compression, and for speech and face recognition.

A SOM is basically a method of vector quantization, i.e. this technique is obligatory in a SOM. Regarding dimensionality reduction, a SOM models data in a nonlinear and discrete way by representing it in a deformed *lattice*. The mapping, however, is given explicitly and well defined only for the prototypes and in most cases only offline algorithms implement SOMs. For our purpose we concern the so-called 'batch' version of the SOM which can easily be derived from the basic model: instead of updating prototypes one by one, they are all moved simultaneously at the end of each run, as in a standard gradient descent. In order to reduce border effects in the neural network we use a *toroidal topology*. For more details concerning the *degree of organization* we refer the reader to [1]. Applying this approach, we work with a so-called *well-organized* neural grid. One of our main tasks concerning the application of Self-Organizing Maps is to implement a suitable mapping procedure that should result in a topology preserving projection of high-dimensional data onto a low dimensional lattice.

In our project we consider only three admissible dimensions of output space, namely $d_A = 1,2,3$ for a

given neuronal grid A. However, in general, the choice of the dimension for the neural net does not guarantee to produce a topology-preserving mapping. Thus, the interpretation of the resulting map may fail. Therefore, we introduce the very important concept of a topologically preserving mapping, which means that similar data vectors are mapped onto the same or neighbored locations in the lattice and vice versa.

In this paper we propose a new concept of cluster visualization; we illustrate clusters as disjoint objects in pairs of simple geometrical objects like spheres in 3D centered at best matching units (BMUs) coordinates within a neural network of admissible dimension.

Our paper is organized as follows: in section 2 we give a precise mathematical description of SOM including the topology preservation measure (topographic product) as a measure for an admissible dimension of the output space. In section 3 we present existing methods of cluster visualization followed by the extension of a graphical visualization method for providing a new solution. In section 4 we demonstrate a software realization approach for our new visualization concept. Finally, we outline our conclusion and emerging further work in section 5.

2 MATHEMATICAL BACKGROUND OF THE SOM

One of the powerful approaches to adopt our cluster considerations within SOM is the application of Self-Organizing Maps to implement a suitable mapping procedure, which should result in a topology-preserving projection of the high-dimensional data onto a low dimensional lattice. In most applications a two- or three-dimensional SOM lattice is the common choice of lattice structure because of its easy visualization. However, in general, this choice does not guarantee to produce a topology-preserving mapping. Thus, the interpretation of the resulting map may fail. Topology preserving mapping means that similar data vectors are mapped onto the same or neighbored locations in the lattice and vice versa.

A. SOM Algorithm and Toplogy Preservation

Within the framework of dimensionality reduction, SOM can be interpreted intuitively as a kind of nonlinear but discrete PCA. Formally, *Self-organizing maps* (SOM) as a special kind of artificial neural network map project data from some (possibly high-dimensional) input space $V \subseteq \Re^{D_V}$ onto a position in some output space (neural map) A, such that a continuous change of a parameter of the input data should lead to a continuous change of the position of a localized excitation in the neural map. This property of *neighborhood preservation* depends on an important feature of the SOM, its output space topology, which has to be predefined before the learning process to be started. If the topology of A (i.e. its dimensionality and its edge length ratios) does not match that of the data shape, neighborhood violations will occur. This can be written in a formal way by defining the output space positions as $r = (i_1, i_2, i_3, .., i_{n_m}),\ 1 < i_k < n_n$ with $N = n_1 \times n_2 \times n_3 .. \times n_m$ where $n_k, k = 1..m$ represents the dimension of A (i.e. length of the edge of the lattice) in k^{th}-direction. In general, other arrangements are possible, e.g. the definition of a connectivity matrix. Nevertheless, we consider hypercubes in our project. We associate a weight vector or *pointer* w_r with each neuron $r \in A$ in V.

The mapping $\Psi_{V \to A}$ is realized by rule: *the winner takes it all* (WTA). It updates only one prototype (the BMU) at each presentation of a datum. WTA is the simplest rule and includes the classical competitive learning as well as the frequency-sensitive competitive learning

$$\Psi_{V \to A}: v \mapsto s = \arg\min_{r \epsilon A} \|v - w_r\| \qquad (1)$$

where the corresponding reverse mapping is defined as $\Psi_{A \to V}: r \mapsto w_r$. These two functions together determine the map

$$M = (\Psi_{\mathrm{V} \to \mathrm{A}}, \Psi_{\mathrm{A} \to \mathrm{V}}) \qquad (2)$$

realized by the SOM network. All data points $v \in \Re^n$ that are mapped onto the neuron r make up its receptive field $\Omega_r^{'}$. The masked receptive field of neuron r is defined as the intersection of its receptive field with V namely

$$\Omega_r = \{v \in V: r = \Psi_{V \to A}(v)\} \qquad (3)$$

Therefore, the masked receptive fields Ω_r are closed sets. All masked receptive fields form the Voronoi tessellation (diagram) of V. If the intersection of two masked receptive fields $\Omega_r, \Omega_r^{'}$ is non-vanishing ($\Omega_r \cap \Omega_r^{'} \neq \emptyset$), we call both of them $\Omega_r \Omega_r^{'}$ *neighbored*. The neighborhood relations form a corresponding graph structure in G_V in A: two neurons are connected in G_V if and only if their masked

receptive fields are neighbored. The graph G_V is called the *induced Delaunay-graph.* For further details we refer the reader to [2]. Due to the bijective relation between neurons and weight vectors, G_V also represents the Delaunay graph of the weights (Figure. 1).

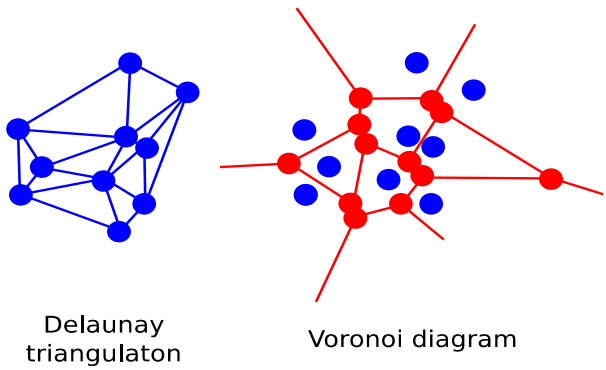

Figure 1: The Delaunay triangulation and Voronoi diagram are dual to each other in the graph theoretical sense.

To achieve the map M, the SOM adapts the pointer positions during the presentation of a sequence of data points $v \in V$ selected from a data distribution $P(V)$ as follows:

$$\Delta w_r = \varepsilon \cdot h_{rs}(v - w_r) \qquad (4)$$

where $0 \leq \varepsilon \leq 1$ denotes learning rate, and h_{rs} is the neighborhood function, usually chosen to be of Gaussian shape:

$$h_{rs} = \exp\left(-\frac{\|r-s\|^2}{2\sigma^2}\right) \qquad (5)$$

We note that h_{rs} depends on the best matching neuron defined in (1).

Topology preservation in SOMs is defined as the preservation of the continuity of the mapping from the input space onto the output space. More precisely, it is equivalent to the *continuity* of M (in the mathematical topological sense) between the *topological spaces* with a properly chosen metric in both A and V. Thus, to indicate the topographic violation we need metric and topological conditions, e.g. in Figure. 2 a) a perfect topographic map is indicated, whereas in 2 b) topography is violated. The pair of nearest neighbors w_1, w_3 is mapped onto the neurons 1 and 3, which are not nearest neighbors. The distance relation between both is inverted as well: $d_V(w_1, w_2) > d_V(w_1, w_3)$ but $d_A(1,2) < d_A(1,3)$. Thus, topological and metric conditions are violated. For detailed considerations we refer to [3]. The topology preserving property can be used for immediate evaluations of the resulting map, e.g. for interpretation as a color space which we applied in Sec. 3.

As we already pointed out in the introduction, violations of the topographic mapping may raise false interpretations. Several approaches were developed to measure the degree of topology preservation for a given map. We chose the topographic product P, which relates the sequence of input space neighbors to the sequence of output space neighbors for each neuron. Instead of using the Euclidean distances between the weight vectors, this measure applies the respective distances $d^{G_V}(w_r, w_{r'})$ of minimal path lengths in the induced Delaunay graph G_V of w_r. During the computation of P the sequences $n_m^A(r)$ of the m$^{\text{th}}$ neighbors of r in A and $n_m^V(r)$, describing the m$^{\text{th}}$ neighbor of w_r have to be determined for each node r. These sequences and further averaging over neighborhood orders m and nodes r finally lead to

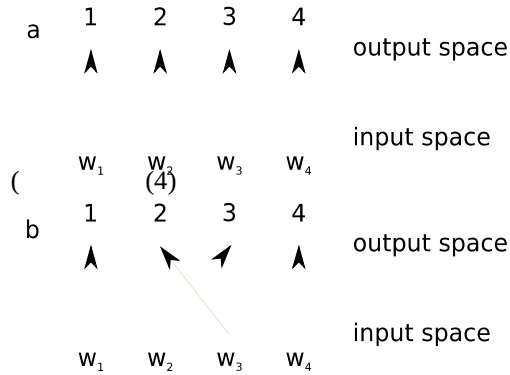

Figure 2 : Metric vs. topological conditions for map topography.

$$P = \frac{1}{N^2 - N} \sum_r \sum_{m=1}^{N-1} \frac{1}{2m} \log\left(\prod_{l=1}^{m} \frac{d^{G_V}\left(w_{n_l^A(r)}\right)}{d^{G_V}\left(w_{n_l^V(r)}\right)} \cdot \frac{d_V\left(r, n_l^A(r)\right)}{d_V\left(r, n_l^V(r)\right)} \right) \qquad (6)$$

The sign of P approximately indicates the relation between the input and output space topology whereas $P < 0$ corresponds to a too low-dimensional input space, $P \approx 0$ indicates an approximate match, and $P > 0$ corresponds to a too high-dimensional input space.

In the definition of P, topological and metric properties of a map are mixed. This mixture provides a simple mathematical characterization of what P actually measures. However, for the case of perfect preservation of an order relation, identical sequences $n_m^A(m)$ and $n_m^V(m)$ result in P taking on the value $P = 0$.

Application of SOMs to very high-dimensional data can produce difficulties that may result from the so-called '*curse of dimensionality*': the problem of sparse data caused by the high data dimensionality. We refer to approach proposed by KASKI in [4].

B. *Application of the Topographic Product involving real-world Data*

Data set in case of speech feature vectors ($D_V = 19$, dimension of input space) obtained from several speakers uttering the German numerals[1]. We see (Fig. 3) in that case topographic product single out $d_A \approx 3$.

C. *Batch Version of Kohonen's Self-Organizing Map*

Depending on the application, data observations may arrive consecutively or alternatively, the whole data set may be available at once. In the first case, an online algorithm is applied. In the second case, an offline algorithm suffices. More precisely, offline or *batch* algorithms cannot work until the whole set of observations is known. On the contrary, online algorithms typically work with no more than a single observation at a time.

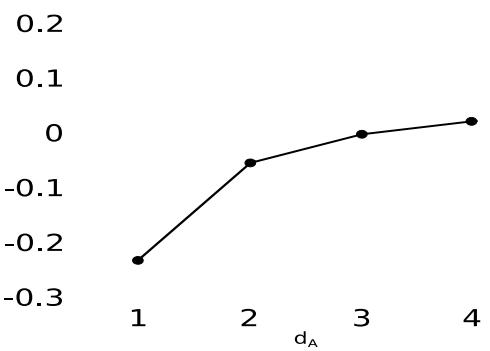

Figure 3 : Values of the topographic product for the speech data.

[1] The data is available at III. Physikalisches Institut Goettingen; previously investigated in [8], [9].

For most methods the choice of the model largely orients the implementation towards one or the other type of algorithm. Generally, the simpler the model, the more freedom is left to the implementation. In our project we apply the batch version of the SOM described in the following algorithm:

1) Define the lattice by assigning the low-dimensional coordinates of the prototypes in the embedding space.

2) Initialize the coordinates of the prototypes in the data space.

3) Assign to ε and to the neighborhood function h_{rs} their scheduled values for epoch q.

4) For all points v in the data set, compute all prototypes as in (1) and update them according to (4).

5) Continue with step 3 until convergence is reached (i.e. updates of the prototypes become negligible).

3 DATA MINING WITH SOM

If a proper SOM is trained according to the above mentioned criteria several methods for representation and post-processing can be applied. In case of a two dimensional lattice of neurons many visualization approaches are known. The most common method for visualization of SOMs is to project the weight vectors in the first dimension of the space spanned by the principle components of the data and connecting these units to the respective nodes in the lattice that are *neighbored*. However, if the shape of the SOM lattice is hypercubical there are several more ways to visualize the properties of the map. For our purpose we focus only on those that are of interest in our application. An extensive overview can be found in [6].

A. *Current Cluster Visualization Methods of SOM*

An interesting evaluation is the so-called U-matrix introduced by [5] (Figure. 4).

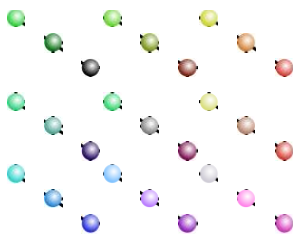

Figure 4 : Representation of positions of neurons in the three-dimensional neuron lattice A as a vector c=(r,g,b) in the color space C, where r, g, b denote the intensity of the colors red green and blue. Thus, colors are assigned to categories (winner neurons).

The elements of the matrix U represent the distances between the respective weight vectors and are neighbors in the neural network A. Matrix U can be used to determine clusters within the weight vector set and, hence, within the data space. Assuming that the map is topology preserving, large values indicate cluster boundaries. If the lattice is a two-dimensional array the U-matrix can easily be viewed and gives a powerful tool for cluster analysis. Another visualization technique can be used if the lattice is three-dimensional. The data points then can be mapped onto neuron *r* which can be identified by the color combination *red, green* and *blue* (Figure. 5) assigned to the location *r*. In such a way we are able to assign a color to each data point according to equation (1) and similar colors will encode groups of input patterns that were mapped close to one another in the lattice A.

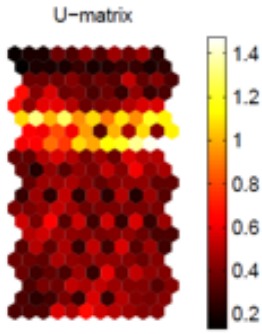

Figure 5: Cluster visualization via U-Matrix.

It should be emphasized that for a proper interpretation of this color visualization, as well as for the analysis of the U-matrix, topology preservation of the map M is a strict requirement. Furthermore, we should pay regard to the fact that the topology preserving property of M must be proven prior to any evaluation of the map.

B. A new Concept for Cluster Visualization

We provide a new idea in order to get insight of visualizing clusters as disjoint objects in pairs of simple objects like 3D spheres, independently of the resulting admissible output space. In this manner, additionally to existing visualization methods, we are able to distinguish and illustrate the "volume" of each cluster obtained by the radius of the constructed spheres.

In the following steps we describe our visualization approach in further detail. At the very beginning the input data set is predefined as clustered data set after the GNG [11] learning process is finished. Afterwards the batch version of the SOM algorithm is performed whereas all BMUs are computed for all input clusters respectively. Finally, the dimension reduction of the input space is achieved by utilizing the topographic product as a judgment tool for an admissible output space.

Affine spaces provide a better framework for doing geometry. In particular, it is possible to deal with points, curves, surfaces, etc., in an intrinsic manner, i.e., independently of any specific choice of a coordinate system. Naturally, coordinate systems have to be chosen to finally carry out computations, but one should learn to resist the temptation to resort to coordinate systems until it becomes necessary. So, we treat the admissible output space as an affine space in intrinsic manner where no special origin is predefined. We set the origin neuron numbered with 1 (Figure. 6). For simplicity, in the neuronal grid, distances between all directly neighboring neurons are set to 1.

Let $|C_x|$ denote the power of a cluster C_x (the number of entities for a given C_x). We are aiming to construct a presentation space in homogenous form in the sense of space dimension for any case of d_A. We calculate the radius of spheres[2] centered on corresponding BMUs as follows:

$$\mathbf{r}_i = \mathbf{0.5} \cdot \left(\mathbf{1} - \frac{|C_i|}{\sum_j |C_j|} \right) \qquad (7)$$

[2] In our considerations we use the term of spheres for all cases of d_A regarding the topology amongst them.

Obviously, spheres constructed in that manner in the output space of dimension d_A do not have any point in common. In our calculations we apply a parametric equation of a sphere. In order to keep the presentation space homogenous to dimension 3 (Fig. 7), with no *relative topology* at presence, we extend the output space as described below.

In case of $d_A = 3$ we perform no operation, since no extension is needed (identity map). In case of $d_A = 2$

$$(r\cos x, r\sin x) \mapsto \left(r\cos x, r\sin x, \pm\sqrt{r_i^2 - r^2}\right), \quad (8)$$

where $0 \le x < 2\pi, 0 \le r \le r_i$, needs to be applied. Finally, in case of $d_A = 1$ (functions composition) the application of

$$r \mapsto (r\cos x, r\sin x) \mapsto \left(r\cos x, r\sin x, \pm\sqrt{r_i^2 - r^2}\right), \quad (9)$$

where $0 \le x < 2\pi, 0 \le r \le r_i$, becomes necessary. In our method we propose to describe clusters as disjoint spheres' centers located at every BMUs position respectively after the batch SOM algorithm is finished. In any cases of topology preservation criterion results (1, 2 or 3 - admissible dimension of neuronal net, after dimension reduction process) we are able to construct a group of disjoint spheres in 3D.

C. Comments

The novelty of our approach is to present clusters via suitable separated object – spheres in ous 3D presentation space. In contrast to the k-clustering concept [12] we apply modern Growing Neuronal Gas unsupervised learning process returning separated objects in form of a clustered probability distribution for a given input data set of possibly high dimension. Finally, we link this concept with Self-Organizing Maps framework in order to illustrate clusters in space of admissible reduced sion. For comprehensive source on dimension reduction of high-dimensional data the reader is referred to [13]

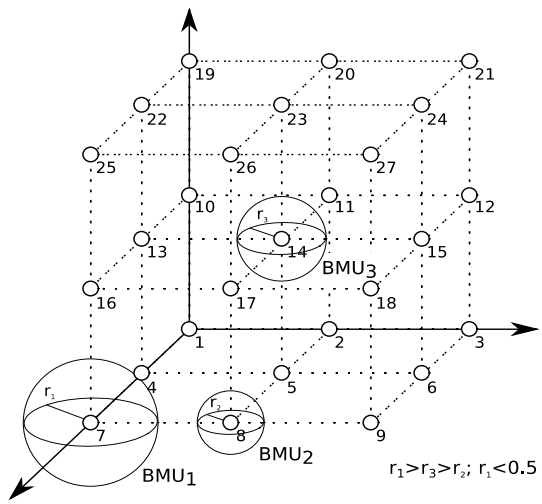

Figure 6 : Neurons and best matching units in a chosen admissible output space with the origin neuron intrinsically numbered with 1.

4 VISUALIZING CLUSTER INFORMATION VIA SOM ON MOBILE DEVICES

The following example will describe a realization of a SOM-based cluster visualization technique for information visualization, thus, displaying a semantic-based database index cluster structure on mobile platforms. The aim was to visually represent the internal database index organization structure intuitively to a user. Our realization had to focus on different requirements.

A. Requirements

The implementation of a SOM-based cluster visualization platform to display a database index' cluster data on mobile entities had to fulfill certain requirements. First of all, the requirement to run our application on mobile devices with potentially low computational power was a challenge. Second, the functionality of our application had to be ensured using any type of network connection provided by the mobile device also including mobile networks with low bandwidth. As a functional requirement, it was requested to visualize clusters as spheres, where the number of data tuples contained in each cluster should be presented implicitly.

B. Requirements Analysis

Due to computational limitations of mobile platforms, the possibility of running SOM transformations on a mobile device could not be regarded as feasible. Thus, a separation of our desired application into a client and a server part was regarded as the most promising solution. Based on the result of the analysis of our first requirement, we did not regard it as suitable to transmit all cluster data required for SOM computations. We decided to transmit only the results of the SOM process since this also seemed to guarantee a smaller data amount compared to the SOM's input data. Furthermore, we intended to reduce possible error causes with this decision regarding the possible necessity of different implementations for different mobile platforms. Finally, the requirements analyses led us to centralize computational effort, thus, utilizing the application on a mobile device only as interface for visualization and user interaction.

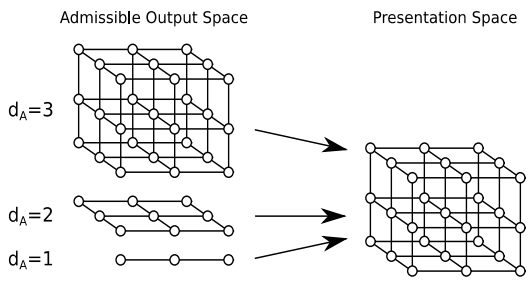

Figure 7 : Expansion of output space A to presentation space depending on admissible output space dimension d_A.

C. Realization

We separated our application into two parts: a server application, and a client application for mobile devices. As described in our requirements analysis we decided to centralize computation effort on the server side, thus, realizing SOM computations there. For realizing the SOM computations we made use of SOM Toolbox contained in Matlab® by building a bridge to C++ for enabling our server application to run the necessary SOM transformations easily. Using this tool chain allowed us to prepare the cluster data for visualization by dimension reduction through SOM efficiently.

The mobile application was designed to run on mobile platforms with touch interfaces but comparably low computational resources. An example screen shot of our user interface is given in Fig. 8 showing clusters, i.e. spheres, that were transformed from n-dimensional space to 3-dimensional output space using SOM.

As shown in Figure. 8, the spheres are of different size. We decided to use a spheres size to implicitly visualize the number of data tuples contained in its according cluster. For determining a sphere's actual size we put the number of data tuples in a cluster into relation to the number of data tuples contained in all clusters. To prevent the spheres from intersecting each other we decided to limit their size by regarding the minimum Euclidian distance δ_{min} of each pair of spheres amongst all spheres into consideration. At a first glance we took the radius of a sphere into consideration for determining its size by making the radius proportionally dependent of the number of data tuples contained in the underlying cluster. Nevertheless, data is *contained* in a cluster, which leads us to the volume of spheres. Therefore, we decided to represent the number of tuples in a cluster by making a sphere's volume dependent on these. Thus, we were able to implicitly represent the data amount contained in a cluster.

Our example was based on a data set with 998 dimensions in input space.

D. Capabilities of our Example

The software system presented in our example is capable of visualizing information on the clustering state of a semantic based database index allowing the user to navigate through the index' cluster structure. This may be performed either by using the visualization feature of the index' hierarchy or by utilizing the realized SOM-based visualization feature. In future development our aim is to present more detailed information and to increase user interaction possibilities potentially influencing the clustering process.

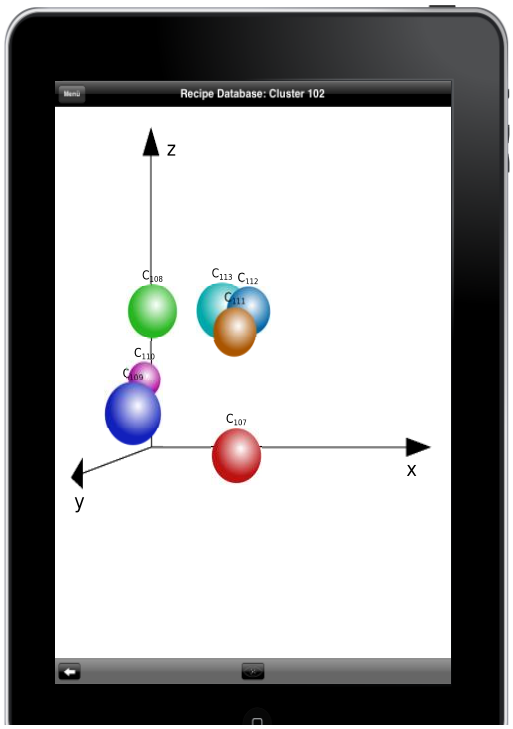

Figure 8 : Visualization of clusters in three-dimensional output space after applying SOM.

5 Conclusion and further Work

In our paper we have deeply described SOM from the mathematical point of view, giving precise description for that kind of neuronal nets, emphasizing the role of topographic product as a criterion for admissible neuronal net dimensions in dimension reduction process.

We have proposed a new illustration method for cluster visualization, linking existing visualization methods of colors (RGB) with methods of separated objects like 3D-spheres, providing better understanding of clusters as joint objects. Finally the software realization approach has been presented.

In our further research we will consider a data-driven version of SOM, so called growing SOM (GSOM). Its output is a structure adapted hypercube A, produced by adaptation of both the dimensions and the respective edge length ratios of A during the learning, in addition to the usual adaptation of the weights. In comparison to the standard SOM, the overall dimensionality and the dimensions along the individual directions in A are variables that evolve into the hypercube structure most suitable for the input space topology.

REFERENCES

[1] G. Andreu, A. Crespo, and J. M. Valiente. Selecting the toroidal self-organizing feature maps (tsofm) best organized to object recognition. In Proceedings of International Conference on Artificial Neural Networks, Houston (USA), volume 1327 of Lecture Notes in Computer Science, pages 1341–1346, June 1997.

[2] T. Martinetz and K. Schulten, "Topology representing networks". *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.

[3] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology Preservation in Self–Organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[4] S. Kaski, J. Nikkilä, and T. Kohonen. Methods for interpreting a self-organized map in data analysis. In *Proc. Of European Symposium on Arti cial Neural Networks (ESANN'98)*, pages 185–190, Brussels, Belgium, 1998. D facto publications.

[5] A. Ultsch. Self organized feature maps for monitoring and knowledge aquisition of a chemical process. In S. Gielen and B. Kappen, editors, *Proc. ICANN'93, Int. Conf. on Artifcial Neural Networks*, pages 864–867, London, UK, 1993. Springer.

[6] J. Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(7):123–456, 1999.

[7] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[8] H.U. Bauer, and K.Pawlzik, Quantifying the neighborhood preservation of self-organizing feature maps, IEE Trans. Of Neur. Netw. 3 (4), 570-579 (1992)

[9] T. Gramss, H.W. Strube, Recognition of Isolated Words Based on Psychoacoustics and Neurobiology. Speech. Comm. 9, 35-40, 1990.

[10] T. Kohonen, "Self organization and associative Memory", 2nd Edition, Berlin, Germany: Springer-Verlag, 1988.

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

[12] Preparata and Shamos, "Computational geometry, an introduction", Springer-Verlag, 1985.

[13] John A. Lee, Michel Verleysen, Nonlinear Dimensionality Reduction, Springer, 2007