Adaptive Hierarchical Incremental Grid Growing: An architecture for high-dimensional data visualization

Dieter Merkl[1,2], Shao Hui He[3], Michael Dittenbach[2], and Andreas Rauber[3]

[1] Institut für Rechnergestützte Automation, Technische Universität Wien
http://www.rise.tuwien.ac.at/

[2] Electronic Commerce Competence Center–EC3, Wien
http://www.ec3.at/

[3] Institut für Softwaretechnik und Interaktive Systeme, Technische Universität Wien
http://www.ifs.tuwien.ac.at/

Keywords: Hierarchical clustering, adaptive architecture, text mining

Abstract— Based on the principles of the *self-organizing map*, we have designed a novel neural network model with a highly adaptive hierarchically structured architecture, the *adaptive hierarchical incremental grid growing*. This feature allows it to capture the unknown data topology in terms of hierarchical relationships and cluster structures in a highly accurate way. In particular, unevenly distributed real-world data is represented in a suitable network structure according to its specific requirements during the unsupervised training process. The resulting three-dimensional arrangement of mutually independent maps reveals a precise view of the inherent topology of the data set.

1 Introduction

The *self-organizing map* (SOM) [9] is an artificial neural network model that proved to be exceptionally successful for data visualization applications where the mapping from an usually very high-dimensional data space into a two-dimensional representation space is required. The remarkable benefit of SOMs in this kind of applications is that the similarity between the input data as measured in the input data space is preserved as faithfully as possible within the representation space. Thus, the similarity of the input data is mirrored to a very large extend in terms of geographical vicinity within the representation space.

However, some difficulties in SOM utilization remained largely untouched despite the huge number of research reports on applications of the SOM. First, the SOM uses a fixed network architecture in terms of number and arrangement of neural processing elements which has to be defined prior to training. Obviously, in case of largely unknown input data characteristics it remains far from trivial to determine the network architecture that allows for satisfying results. Thus, it certainly is worth considering neural network models that determine the number and arrangement of units during their unsupervised training process. We refer to [1, 2, 7, 8] for recently proposed models that are based on the SOM, yet allow for adaptation of the network architecture during training. Second, hierarchical relations between the input data are not mirrored in a straight-forward manner. Such relations are rather shown in the same representation space and are thus hard to identify. Hierarchical relations, however, may be observed in a wide spectrum of application domains, thus their proper identification remains a highly important data mining task that cannot be addressed conveniently within the framework of the SOM. The *hierarchical feature map* (HFM) as proposed in [13], i.e. a neural network model with hierarchical structure composed from independent SOMs, is capable of representing the hierarchical relations between the input data. In this model, however, the sizes of the various SOMs that build the hierarchy as well as the depth of the hierarchy have to be defined prior to training. Thus, considerable insight into the structure of the input data is necessary to obtain satisfying results.

Only recently, we have proposed the *growing hierarchical self-organizing map* (GHSOM) as an artificial neural network architecture designed to address both limitations within a uniform framework [4, 5, 6]. Similar to the HFM, a hierarchical layout of the architecture is chosen. In case of the GHSOM, however, this hierarchical layout is determined during the unsupervised training process guided by the peculiarities of the input data.

In this work, we describe an alternative model, the *adaptive hierarchical incremental grid growing* (AHIGG), where the individual layers of the hierarchical architecture are variants of the *incemental grid growing* (IGG) network as originally proposed in [2, 3]. The major difference of the AHIGG as compared to the GHSOM is that maps on individual layers may grow irregularly in shape and may remove connections between neighboring units. In this way a better understanding of the underlying input data can be gained which lends itself for easy visual exploration.

The remainder of this paper is organized as follows.

Section 2 contains a brief review of the *incremental grid growing* neural network, an adapted version of which will be used as building blocks for the AHIGG. In Section 3 we provide an outline of architecture and training process of the AHIGG. Section 4 contains the description of an application scenario for the AHIGG, namely the organization of document archives. Finally, we present our conclusions in Section 5.

2 A quick review of *Incremental Grid Growing*

The *incremental grid growing* (IGG) model as proposed in [2, 3] combines the topology preserving nature of SOMs with a flexible and adaptive architecture that represents cluster structures during an unsupervised training process. Initially, the IGG network consists of four connected units, each of which is assigned an initially random weight vector of the same feature space as the training data. During the training process, the network is dynamically changing its structure and its connectivity to resemble the topology of the input data.

The training process consists of a sequence of iterations where each cycle consists of the following three phases:

(1) *The SOM training phase*: The SOM algorithm is applied to train the current map. The weight vectors of the units are adapted to the high-dimensional relations in the input data.
(2) *The expansion phase*: New units are added to that region at the perimeter of the current map that are responsible for the largest quantization error.
(3) *The adaptation of connections phase*: Connections between neighboring units are added or removed from the network depending on the metric distance between the units' weight vectors. Thus, cluster boundaries and discontinuities in the input data become explicitly visible.

A note on the three phases is in order. During the *SOM training phase* the quantization error of the various units is cumulated as detailed in Eq. (1) with i being the index of the unit in question, m_i that unit's weight vector, x an input vector, and t referring to the current SOM training iteration.

$$E_i(t+1) = E_i(t) + \sum_k (x_k - m_{ik})^2 \qquad (1)$$

After a fixed number of SOM training iterations the unit with the largest cumulated quantization error is selected as the *error unit*, as symbolized in Figures 1(a) and 1(c). When given a rectangular network layout, each unit may have four neighbors. During the *expansion phase* new units are generated at the unoccupied neighboring grid positions of the *error node*, as symbolized in Figures 1(b) and 1(d).

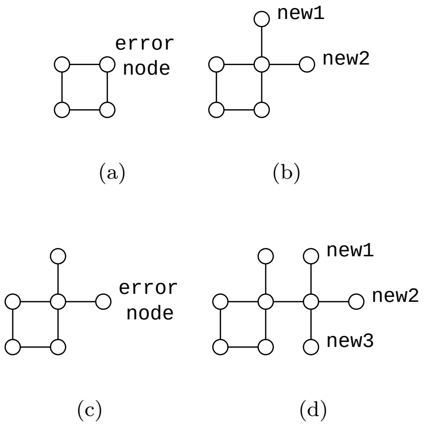

Figure 1: IGG–Expansion phase

Finally, during the *adaptation of connections phase* the metric distance of weight vectors at neighboring grid positions are analyzed. A connection between the respective units is established if the distance of their weight vectors is below a particular treshold value $\tau_{connect}$, as symbolized in Figures 2(a) and 2(b). However, if this distance is larger than a second threshold parameter $\tau_{disconnect}$ then a possible existing connection between the units is removed, as symbolized in Figures 2(c) and 2(d). The thus established connections play a critical role during the then following next *SOM training phase* because only connected nodes in the neighborhood of the *winner* are adapted.

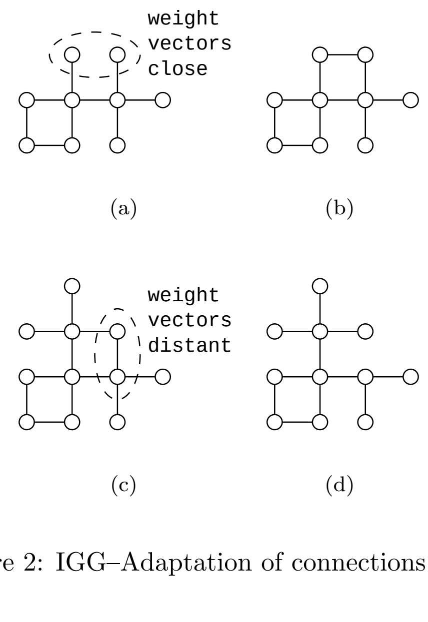

Figure 2: IGG–Adaptation of connections phase

3 A hierarchically growing IGG network

Basically, the AHIGG is composed of a hierarchical arrangement of independent IGG networks on each of its layers. Each layer is resposible for input data representation at a specific level of granularity. Pragmatically speaking, a rough idea of the similarities in the input data is represented in the first layer of the AHIGG. Each unit of this first layer map may be expanded to an individual map on the second layer of the hierarchy if the desired level of granularity in data representation is not reached yet. Thus, the layers further down the hierarchy give a more detailed picture of subsets of the input data. Consider Figure 3 as a simple pictorial representation of an AHIGG consisting of three layers.

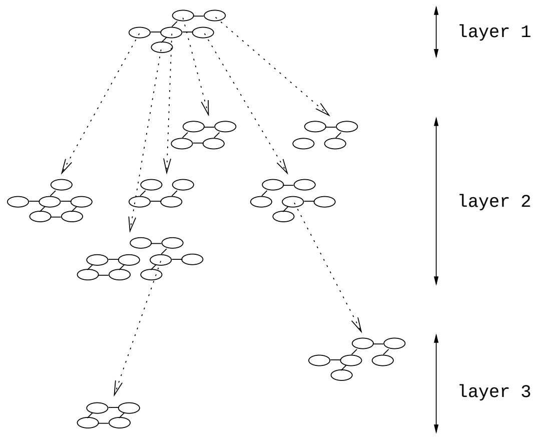

Figure 3: Architecture of the AHIGG

At the beginning of training the weight vector of a single unit map at layer 0 is initialized as the statistical mean of the input data. The *mean quantization error* of this unit as given in Eq. (2) will play a crucial role during the training process of the AHIGG. In this formula, $\mathcal{I}$ refers to the set of input data, n is the cardinal number of $\mathcal{I}$, x is an input data, and m_0 is the weight vector of the single unit at layer 0.

$$mqe_0 = \frac{1}{n} \sum_{x \in \mathcal{I}} ||m_0 - x|| \qquad (2)$$

In the next step of training, a map at layer 1 is created that consists of a small number of units, e.g. four units arranged in a square. The weight vectors of these units are initialized randomly but taking into account the weight vector of its 'parent' unit in the preceding layer (m_{parent}) together with the mean quantization error of that unit (mqe_{parent}). The initialization scheme is given in Eq. (3), with v_{rand} denoting a random vector of length 1.

$$m_i = m_{parent} + mqe_{parent} \cdot v_{rand} \qquad (3)$$

Please note this initialization scheme is different to the one proposed originally for the IGG network in [2]. We have chosen this scheme because it allows for weight vectors being roughly aligned within the input data space.

After initialization, the network is training according the the SOM algorithm for a fixed number λ of input vector presentations. Then, the border unit with the largest mean quantization error is selected and new neighboring units are added to the network. Finally, the weight vectors of neighboring units are checked for possible adaptation of connections. This training process follows the description as given in Section 2. This training process is repeated until the mean quantization error of the map falls below a certain fraction τ_1, $0 < \tau_1 < 1$, of the mean quantization error of it's parent unit. A fine-tuning phase is then performed where only the winner is adapted and no further units are added to the network. After this fine-tuning phase, each unit is checked for possible hierarchical expansion. More precisely, the mean quantization error of each unit is computed and units with too high a mean quantization error are expanded on the next layer of the hierarchy, i.e. for those units a new map on the next layer of the hierarchy is established. The mean quantization error of a units is compared to the mean quantization error of the unit at layer 0 and a simple threshold logic is used for the decision of hierarchical expansion. Each unit for which Eq. (4) holds true is further expanded. In this formular, τ_2 represents the threshold, $0 < \tau_2 < 1$.

$$mqe_i > \tau_2 \cdot mqe_0 \qquad (4)$$

A difference to the original IGG model can be found in the initialization of weight vectors of newly added units. In [2] an intialization strategy is proposed the preserves the local topology by taking into account statistical means. More precisely, the new weight vectors are initialized such that the error node's weight vector is the statistical mean of its neighbors. Apart from the fact that in some cases the new weight vectors may lie beyond the data space, this scheme may produce isolated units at the perimeter of the map. The reason for this phenomenon is explained by the location of the growth process. We believe that a topology preserving initialization works well in the interior of the map where the extent of the interpolation is given by the enclosure. Such a strategy can be found for example in the *growing grid* network [8]. However, steady continuation into the open area at the perimeter of the IGG may be fatal. In the most pathological case, the connections to the new units are immediately removed after each growing step. They thus become isolated and are not likely to contribute to the share of input patterns. We therefore suggest to initialize the new weight vectors randomly with a vector from the ϵ-environment of the error node, as given in Eq. (5).

In this formular, m_{new} refers to the weight vector of a newly added unit, m_{error} is the weight vector of the error unit, v_{rand} is a random vector of length 1, and ϵ is a small constant, i.e. $0 < \epsilon << 1$.

$$m_{new} = m_{error} + \epsilon \cdot v_{rand} \quad (5)$$

Due to the fact that neighboring units are not necessarily connected, we slightly adapted the notion of neighborhood range in the SOM algorithm for training. Instead of measuring the distance between two units in Euclidean map space, we rather take their connectivity into account for determining those units that are subject to adaptation apart from the winner. Simply the length of the shortest path between two units is determining the strength of adaptation. Please note that adaptation is thus no longer symmetrical around the position of the winner. Figure 4 gives a simple graphical representation of neighborhood adaptation in AHIGG; the darker the shading of a node in this Figure, the stronger is it's adaptation.

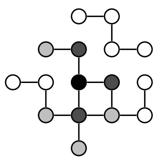

Figure 4: Neighborhood adaptation in the AHIGG

4 The *TIME* Collection

In the experiments presented hereafter we use the *TIME Magazine* article collection as a reference document archive. The collection comprises 420 documents from the *TIME Magazine* of the early 1960's. The documents can be thought of as forming topical clusters in the high-dimensional feature space spanned by the words contained in the documents. The articles cover the typical range of subject matters in a news magazine, i.e. ranging from foreign politics and world economics to fashion and gossip. The goal is to map and identify the topical clusters on the 2-dimensional map display. Thus, we use full-text indexing to represent the various documents according to the vector space model of information retrieval. The indexing process identified 5,923 content terms, i.e. terms used for document representation, by omitting words that appear in more than 90% or less than 1% of the documents. The terms are roughly stemmed and weighted according to a $tf \times idf$, i.e. term frequency times inverse document frequency, weighting scheme [14], which assigns high values to terms that are considered important in describing the contents of a document. Following the feature extraction process we end up with 420 vectors describing the documents in the 5,923-dimensional document space, which are further used for neural network training.

To get a better understanding of the results of the training process, we performed an automatic unit labeling technique after training has completed. The goal of this labeling technique is to make those features of the input data explicit that have high impact on the clustering result. In case of our application, obviously, the features are terms extracted from the various documents. The idea behind the labeling technique is to select those terms that are highly important for the documents represented by a particular unit of the AHIGG network, i.e. terms with high average $tf \times idf$ values yet low standard deviation. More formally, we describe the importance of a particular feature, i.e. imp_i, for the labeling process as given in Eq. (6). In this formula, ξ_i refers to the i-th feature, $\overline{\xi_i}$ is its statistical mean and σ_i its statistical standard deviation.

$$imp_i = \frac{\overline{\xi_i}}{\sqrt{1 + \sigma_i}} \quad (6)$$

Please note this labeling strategy is comparable to our *LabelSOM* method, descibed in [12]. The major difference is that we reduced the originally proposed feature selection function such that only one parameter remains instead of two in the *LabelSOM* method.

The resulting AHIGG represents a quite intuitively interpretable representation of the subjects in the news articles. Due to space restrictions we cannot present the complete topic hierarchy of the *TIME Magazine*, we will rather focus on a few sample maps. The top-layer map is depicted in Figure 5.

At first glance, we can instantly detect an isolated cluster in the lower left corner of the map which refers to articles about Vietnam. Please note that disconnected units are visualized by means of a black bar in between them. This might be regarded as a rather clumsy visualization technique but at least it works with most web browsers. Documents related to China, the Kashmir conflict between India and Pakistan or the Malaysian independence are located to the left on unit (2/1)[1] and (2/2). These two together with unit (3/1) form more or less the Asia cluster of the *TIME Magazine*. Various European topics are concentrated in the middle of the map on the units (3/2), (3/3), (2/3) and (1/3). The range of subjects goes from articles about the Cold War and the NATO to documents concerning the French-German relationship and the political situation in Great Britain. Relevant names of involved politicians such as *Nikita Khrushchev*, *Charles de Gaulle*, *Konrad Adenauer* or *Harold Wilson* have been extracted as labels of these units. The remaining

[1] We will use the notation (x/y) to refer to the unit located at row x and column y, starting with (1/1) in the upper left corner of the table, i.e. the grey colored cells, which do not represent any units of the map, also get a two-dimensional location vector for the sake of simplicity.

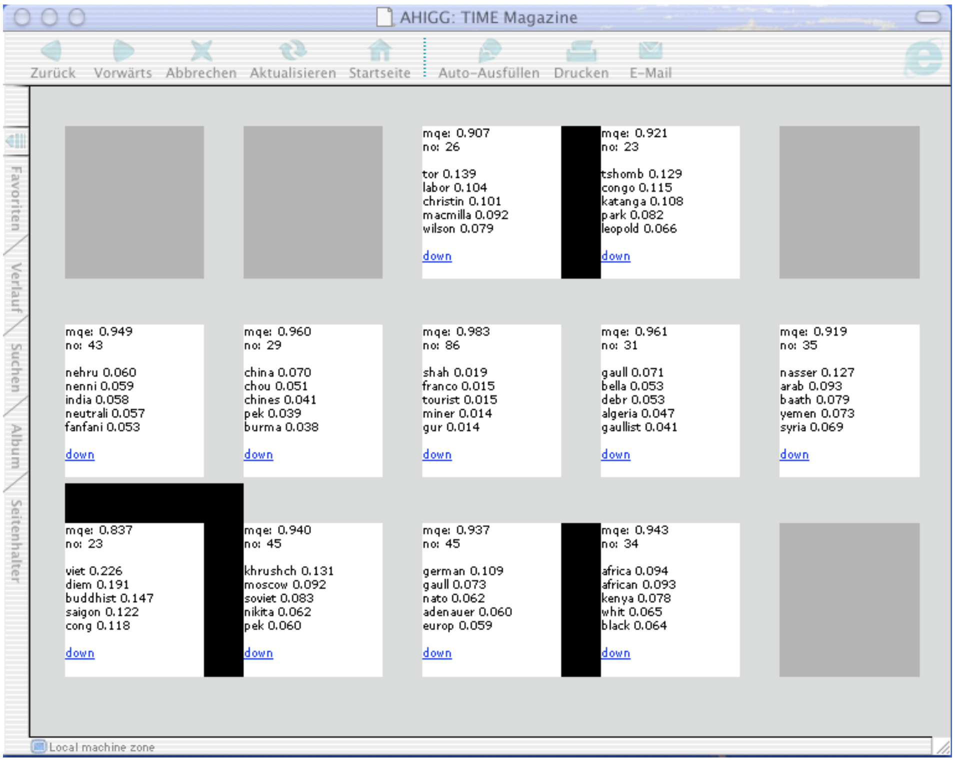

Figure 5: *TIME Magazine*: Top-layer map

four units on the right-hand side of the map mainly deal with Middle East affairs and African related subjects. For example, we find articles about the *Congo* conflict in the province *Katanga* on unit (1/4) or the situation in *Algeria* after the independence war on (2/4).

If we take a closer look at the Vietnam cluster in the second layer of the hierarchy as shown in Figure 6, we can identify three units on the lower left corner onto which documents about the religious problems in Vietnam have been mapped. The labels buddhist and diem refer to the crackdown on the buddhist population and monks by the regime of the roman catholic president Ngo Dinh Diem. The other units represent the Vietnam War cluster. The isolated unit (1/2) is a kind of 'technical report' covering the rocket carrying chopper *Huey* of US Airforce. Unit (1/4) contains articles about the US helicopter attacks against the viet cong guerillas. The neighboring units (2/3) and (2/4) are all concerned with peace conferences and meetings of vietnamese officials with US representatives.

5 Conclusions

In this paper we have presented a novel neural network model, i.e. the *adaptive hierarchical incremental grid growing* (AHIGG). The distinctive features of this model are its hierarchical architecture, where the depth of the hierarchy is determined during the unsupervised training process. Each layer in the hierarchy consists of a number of independently developing maps which determine their size and arrangement of units also during the unsupervised training process. Thus, this model is especially well suited for applications which require hierarchical clustering of the input data space.

We have shown the usefulness of this model by using an application scenario from the information retrieval area, namely the organization of document archives. Such an application scenario is especially well suited to demonstrate the capabilities of an artificial neural network because, first, the documents are represented in a very high dimensional feature space by nature, and, second, document archives are inherently structured hierarchically according to the different subject matters dealt with in the documents.

In summary, we can state that the *adaptive hierarchical incremental grid growing* has successfully proven its applicability in data mining problems which require an accurate representation of hierarchical relations and cluster structures in high-dimensional data. The intuitively interpretable hierarchical organization built during the unsupervised training process offers the user a convenient interface for interactive data analysis of large amounts of complex data.

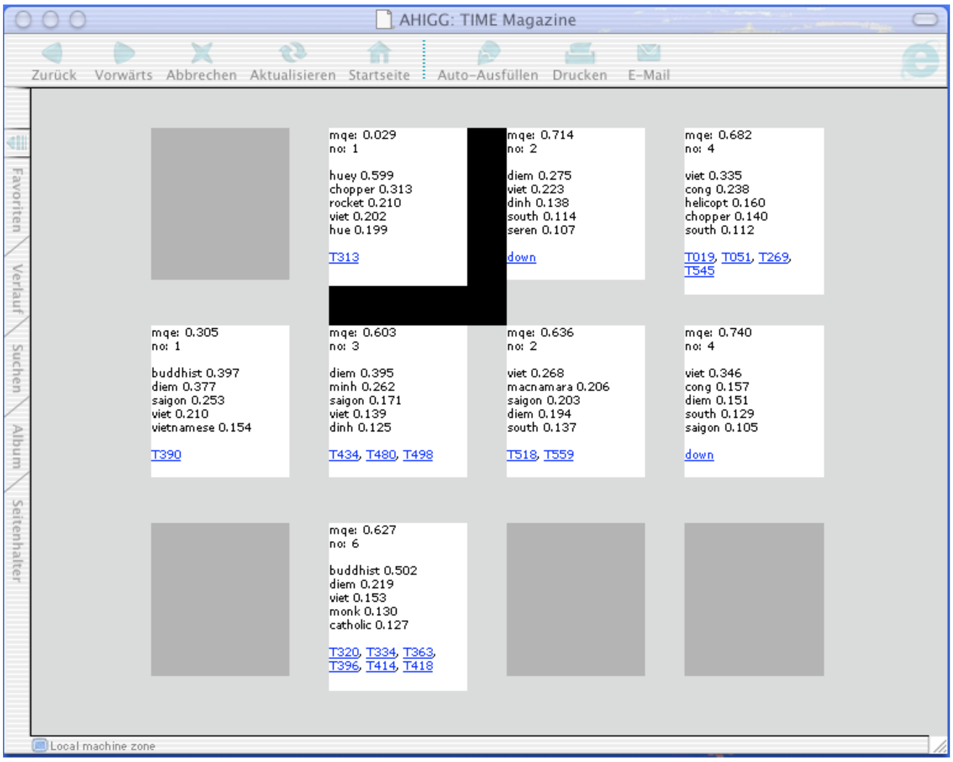

Figure 6: *TIME Magazine*: Vietnam map

References

[1] D. Alahakoon, S. H. Halgamuge, and B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery", *IEEE Trans Neural Networks*, 11(3), 2000.

[2] J. Blackmore and R. Miikkulainen, "Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map", In *Proc IEEE Int'l Conf on Neural Networks* (ICNN'93), San Francisco, CA, 1993

[3] J. Blackmore and R. Miikkulainen, "Visualizing high-dimensional structure with the incremental grid growing neural network", In *Proc Int'l Conf on Machine Learning* (ICML'95), Tahoe City, CA, 1995.

[4] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map", In *Proc Int'l Joint Conf on Neural Networks* (IJCNN'2000), Como, Italy, 2000.

[5] M. Dittenbach, A. Rauber, and D. Merkl, "Recent advances with the growing hierarchical self-organizing map", In *Proc Workshop on Self-Organizing Maps*, Lincoln, UK, 2001.

[6] M. Dittenbach, A. Rauber, and D. Merkl, "Uncovering hierarchical structure in data using the growing hierarchical self-organizing map", *Neurocomputing*, 48(1–4), 2002.

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

[8] B. Fritzke, "Growing grid: A self-organizing network with constant neighborhood range and adaptation strength", *Neural Processing Letters*, 2(5), 1995.

[9] T. Kohonen, "Self-Organized formation of topologically correct feature maps", *Biological Cybernetics*, 43, 1982.

[10] T. Kohonen, *Self-Organizing Maps*, 2nd Ed., Springer–Verlag, Berlin, Germany, 1997.

[11] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection", *IEEE Trans Neural Networks*, 11(3), 2000.

[12] D. Merkl and A. Rauber, "Automatic labeling of self-organizing maps for information retrieval", In *Proc Int'l Conf. on Neural Information Processing* (ICONIP'99), Perth, WA, 1999.

[13] R. Miikkulainen, "Script recognition with hierarchical feature maps", *Connection Science*, 2(1), 1990.

[14] G. Salton, *Automatic Text Processing: The transformation, analysis, and retrieval of information by computer*, Addison-Wesley, Reading, MA, 1989.