APPLICATIONS OF DIFFERENT SELF-ORGANIZING MAP VARIANTS TO GEOGRAPHICAL INFORMATION SCIENCE PROBLEMS

Fernando Bação[1], Victor Lobo[1,2], Marco Painho[1]

[1] ISEGI/UNL, Campus de Campolide, 1070-312 LISBOA, Portugal, bacao@isegi.unl.pt
[2] Portuguese Naval Academy, Alfeite, 2810-001 ALMADA, Portugal

Abstract

In this chapter, an overview of different variants of the original Self-Organizing Map algorithm is presented, with an emphasis on those that can integrate spatial reasoning or otherwise be applied in the context of Geographic Information Science. A few of the most relevant variants are discussed, together with a recently developed variant proposed by the authors. An example of the application of some SOM variants to an artificial dataset is presented, and their main advantages pointed out. Finally, these variants are used to solve a real world problem, and conclusions and recommendations for future work are given.

Keywords: SOM, SOM variants, Geo-SOM, GIScience

INTRODUCTION

The development and sophistication of today's geo-referenced data collection technologies has created a continuous stream of data which has been flooding our databases. This trend will build up in the next years as already available technologies find their way into our daily lives. Data collection through location-aware devices, high resolution remote sensing systems, decennial census operations, among other connected with Geographic Information Systems (GIS) made available huge volumes of geo-referenced data. This created opportunities to improve the available knowledge on a number of socio-economic phenomena that are at the heart of Geographical Information Science (GIScience).

These trends have shaped new challenges and raised unexpected difficulties on the analysis of high dimensional multivariate spatially referenced data. Today, the availability of methods able to perform intelligent data reduction is a central issue in science generically and GIScience is no exception. The need to transform into knowledge the massive digital geo-referenced databases has stimulated work in a number of research areas. It also led GIScientists to search for new tools, which are able to make sense of such complexity. The field of knowledge discovery constitutes one of the most relevant stakes is GIScience research to deal with this problem. Although knowledge discovery and data mining have put forward numerous methodologies and tools, the need to adequate those tools to the specific context of GIScience remains a research challenge (Openshaw and Openshaw, 1997, Openshaw, 1999).

Self-Organizing Maps have been proposed as a step forward in the improvement the data reduction task (Openshaw and Wymer, 1995) and have been used, with good results, to address different GIScience problems as we shall see later in this chapter. In general, these applications are based on the original SOM algorithm, but the most relevant research problem is to seek ways to adapt the algorithm to the specific difficulties caused by the special nature of the geographic data. In fact, the possibilities of altering the original SOM original algorithm are almost endless, so this flexibility can be used take into account the special features of geoinformation.

The main objective of this paper is to present a structured view of the possible modifications of the original SOM algorithm to develop SOM variants which are relevant to GIScience.

We shall start by overviewing some important parameterizations of the SOM algorithm, and then proceed to analyze how the different steps of the algorithm may be changed. We will then overview some of the most relevant variants. Finally we will explain in detail the Geo-SOM variant, together with an example of its application to an artificial data problem and a real world problem.

SOME IMPORTANT PARAMETRIZATIONS OF THE ORIGINAL SELF-ORGANIZING MAP ALGORITHM

Although they do not constitute true variants of the original algorithm, some parameterization choices can radically change the way the SOM may be used. We will discuss three of them, namely:

1. Size of the map
2. Output space dimension
3. Training schedule

Size of the map

There are three major options in terms of the size of the SOM. The first one consists on building a very big SOM in which the number of neurons is greater to the number of input patterns (Ultsch and Siemon, 1990, Ultsch and Li, 1993). The second and by far most common option is to build a medium size map, smaller than the number of input patterns, but still large enough to have a few units representing each cluster of data (Kohonen, 2001). Finally, there is also the possibility of building small maps where the number of units is drastically smaller than the number of input vectors, usually with only one unit for each expected cluster (Bação *et al.*, 2004a). The relevance of these choices is so big that it can be argued that they lead to completely different tasks.

When opting for a big map the underlying assumption is that we whish to explore the structural context of the data. By using more units than input patterns it is possible to obtain very large U-Matrices (Ultsch and Siemon, 1990) on which distances between input patterns can easily be identified. This can be seen as a strictly exploratory exercise, providing an augmented space which will only be helpful if the user is able to interpret the results. Clearly, the data reduction is solely based in the squashing of the *n*-dimensional space into a 1, 2 or 3-dimensional space.

The decision to build a medium size map can be seen as a compromise, in the sense that although reducing the number of dimensions and creating clusters, it still enables the user to understand the basic structure of the data, eventually informing further and more severe reductions.

Finally, small maps are used when the user is interested in clustering the data without concerns about the analysis of the structure. Here the objective is to form clusters of input patterns which are as similar as possible, aiming at a one step substantial data reduction. In this context the U-matrix is of little value, and component planes (Kohonen, 2001) become more relevant as they allow a simple characterization of the resulting clusters.

Output space dimension

The output space can have as many dimensions as the input space (in fact it can have even more). Nevertheless the output space seldom has more than 2-dimensions, because of visualization restrictions. Theoretically, the appropriate dimension of the map should be defined by the intrinsic dimension of the data (Fukunaga and Olsen, 1971, Camastra, 2001). This constitutes a problem since we rarely have such information. The possibility of visualizing the results, which is very relevant, together with the incapacity to confirm the true dimensionality of the map usually leads to opt for 1 or 2 dimensional map.

When the objective is to cluster based on very small SOM's the best approach is to use 1-D SOMs. This is due to the fact that the plasticity of the 1-D map is much greater than a 2-D SOM (Bação *et al.*, 2004a). This fact is apparent in the application to the traveling salesman problem (Maenou et al., 1997) where 1-D SOMs are preferred to 2-D SOMs. The need to closely represent a number of points that can form complex geometric shapes render inefficient the use of 2-D SOMs. On the other hand, if the objective is to obtain a comprehensive visualization of the input space then a 2-D SOM is to be preferred. The rationale is that the higher level of connectivity will yield a better coverage of the input space.

Finally, it is important to refer that any SOM will produce a bias in the representation of the input space. In fact, the distribution of the classification resources (units) will be more than proportional in lower density areas. This effect is usually designated as “magnification effect” (Cottrell *et al.*, 1998, Claussen, 2003). The quantification of this effect has proven to be elusive and is still an unresolved issue.

Training Schedule

The first step in building a SOM usually involves giving initial values to the units. This may be done using completely random values (which usually leads to slow convergence towards the general area of the data), or using values obtained from randomly selected input patterns. This type of initialization will usually produce maps which take a long time to unfold, or may not unfold at all (Kohonen, 2001). Better maps are usually obtained if the units are laid out on a 2-dimensional plane and centred near the mean of the input patterns. The plane may be defined, for example, by the two first eigenvectors of the input patterns.

The counting of training iterations may also vary from one implementation to another. While the well known SOM-PAK implementation (Kohonen et al., 1995) counts each presentation of an input pattern as an iteration, and adjusts the learning parameters after each iteration, most implementations count “epochs” that present the whole training data set, and only then do they adjust the training parameters. In the latter case, the parameters of the units may be updated after each input pattern is presented, in what is called on-line or sequential mode, or the changes may be stored and applied only after the whole training set is presented, in what is called the batch mode.

The way the learning rate and neighbourhood radius varies during training may also be done in different ways. For the SOM to converge to a stable configuration it is necessary to decrease the learning rate to 0. This may however be done in many ways, although we do not know of any conclusive analysis of its impact. Some authors have proposed dynamically changing learning rules that compensate the magnification effect of the standard SOM (Cottrell *et al.*, 1998). In some applications, particularly in on-line system monitoring, it is desirable to maintain some plasticity when using the SOM, and so the learning rate does not converge to 0. The final value of the neighbourhood radius can have a dramatic effect on the final map. If this radius is allowed to decrease to 0, the final stages of training will be equivalent to a *k*-means algorithm, and thus locally optimal. If instead the radius decreases to 1, then the units will always be pulled away from the local minima, and on the borders of the map they will be pulled towards the centre because there are no units pulling them outside de map. Both approaches make sense in different contexts, so care must be taken when choosing these values.

WAYS TO CHANGE THE ORIGINAL SELF-ORGANIZING MAP ALGORITHM

Quite a few reviews have been made of the different variants to the standard SOM algorithm (Kangas et al., 1990, Vesanto, 1999, Vesanto, 2000). So as to structure the different ways in which the basic SOM can be changed, we identified 3 main areas where those changes can occur:

1. Topology and connections between units
2. Matching and voting mechanism (calculation and voting phases)
3. Learning rule (update phase)

We shall now proceed to analyze each of these main areas separately, bearing in mind that any particular implementation of a SOM will probably include a combination of these different changes.

Topology and connection between units

In the standard SOM the units form some of type of regular grid. Units that are neighbors in this grid are influenced by each other, and thus we may consider that a connection exists between them. Some variants of SOM change these connections, or eliminate them altogether.

In the Neural Gas Architecture (NGA) (Martinetz et al., 1993) each unit is completely independent of the others, and thus these units do not form a distinct output space. Since there is no output space, this variant cannot be used for mapping or projection purposes, but it may and has been used for sampling or clustering. The lack of output space forces neighborhoods to be calculated in the input space. This means that during the update phase of the training algorithm the units are ordered by their distance (in the original input space) to the winning neuron, and updated accordingly.

Another approach is to maintain connections between units, but relax the constrain that they form a regular grid, such as happens with the Growing Cell networks (Fritzke, 1991). In this type of neural network units are inserted, one at a time, during the training phase, according to some established criteria. The resulting network may be quite irregular, and since the number of connections each unit has depends on how many units were inserted next to it, no simple output space will be formed. This network gave rise to a family of related architectures, namely the Growing Neural Gas (Fritzke, 1994) and the Gowing Grid (Fritzke, 1995) that allow connections between units to be established or broken during training.

Even when the units do form a regular grid, some SOM variants allow growth in the number of units (Almeida and Rodrigues, 1991) or in the number of dimensions (Bauer and Villmann, 1997).

Another form of interaction between units, even stronger than that imposed by the neighborhood effect, is to allow units to receive as inputs the outputs of other units. This happens in SOMs with feedback used in temporal data analysis (Guimarães et al., 2002), where units receive as inputs the delayed outputs. It also happens in hierarchical SOMs which we will discuss later.

Matching and voting mechanism (calculation and voting phases)

A large number of variants of the basic SOM change the way the matching between input patterns and units is made, and how the best matching unit is selected. A trivial way to change the matching mechanism is to use metrics other than the standard Euclidean distance, and many such metrics have been used (Kohonen, 2001).

More interesting variants of the basic SOM can be obtained if the units are allowed to have an internal structure that is different from the input patterns. In this case the units cease to be points in the input space. One such variant is the Adaptive Subspace SOM (ASSOM) (Kohonen, 2001), where the units represent subspace, and the matching is done by calculating the distance from the input pattern to the subspace. In temporal SOMs it is relatively common to find delay elements associated with the map units, and matching is done using those delays or pass activations of the units (Guimarães et al., 2002). The matching may also be done by finding the fitness of the input pattern to some given criteria that may be stored in the map units.

In some approaches, only a sub-set of units are searched to find the best match. This happens when spatial or temporal restrictions are imposed (Kangas, 1990, Kangas, 1992, Chandrasekaran and Liu, 1998, George, 2000), when tree structures are used to accelerate the search, and when certain supervised versions of SOM are used (Ritter *et al.*, 1992, Buessler *et al.*, 2002).

Learning rule (update phase)

In the basic SOM, each unit is updated according to its distance (in the output space) to the best matching unit, and to its distance in the input space to the input pattern. The final distances in the input space of units that are neighbours in the output space may vary widely.

In particular, if there are large differences in the density of input patterns throughout the input space, there will be regions where neighbouring units are close and others where they are far away. While this is a desirable feature when trying to obtain good U-Matrices (Ultsch *et al.*, 1993) it causes the map to concentrate or even “collapse” on the areas with greater density. This will not be a desirable feature if we want to avoid overspecialization and whish to keep some units available to detect new features or outliers. This line of thought led to the Visualization Induced SOM (ViSom) (Yin, 2001), where a repulsion force is introduced between units, forcing a certain minimum distance between them. It is argued that this approach will provide a “broader view” of the input space. Clusters will still be detectable in a ViSom by analysing variations in the number of patters that are mapped to different units.

The direction in which each of the units is updated is usually that of the input pattern. One alternative is to move the unit in the direction of the nearest unit, as was proposed by (Lee *et al.*, 2001). This resembles the way nodes are pulled in a fisherman’s net, and thus this update rule was dubbed fisherman’s rule. It has been shown (Lee *et al.*, 2001) that this will improve the convergence speed in the first iterations of the learning phase. The main reason for this is that, at each learning step, the different units will not be attracted in exactly the same direction (the direction of the input pattern), but will instead be pulled in a direction that depends on their immediate neighbours. It is suggested (Lee *et al.*, 2001) that the fisherman’s rule be used in the first iterations of the training algorithm, being replaced by the standard rule in the last iterations.

Although the standard SOM is an unsupervised learning algorithm, a number of supervised variants exist. It can be argued that the calibration mechanism (Kohonen, 2001) is in fact a form of supervised learning, but this does not change the way the SOM is trainined. The most common way of introducing supervised learning in the SOM training is to change the way units are updated according to the known class of the input pattern. The well known LVQ algorithm (Kohonen, 2001) is one such case, where units are attracted to the input pattern if they have the same class, or repelled if it is different. Other supervised versions of SOM have also been proposed (Buessler *et al.*, 2002).

Finally, the update rule may also be changed because of the particularities of the input space or the metric used. Such is the case when binary features are used (Gioiello *et al.*, 1992, Tanomaru, 1995, Lobo, 2002). In this case, since the only acceptable values for each attribute are 0 and 1, the smooth adaptation required by the standard rule in not possible. In (Lobo, 2002) two update rules are proposed, that require ‘correcting’ a number of bits proportional to the desired learning rate. A similar adaptation of the learning rate is required whenever categorical data is used, as shown in (Lourenço *et al.*, 2004).

GEO-VARIANTS – INCLUDING GEOGRAPHIC REASONING INTO THE SOM

In this section we overview some of the applications of SOMs is GIScience problems, and will then proceed to cover a selected number of SOM variants which allow for the inclusion of geography within the workings of the SOM.

In (Villmann and Merényi, 2001) and (Villmann *et al.*, 2003) SOM variants are used to analyze satellite images. Two different variants are used: the GrowingSOM (Bauer and Villmann, 1997), where the SOM is allowed to grow into a *n*-dimensional hypercube topology so as to accurately match the intrinsic dimensionality of the data; and the SOM with magnification control (Bauer *et al.*, 1996), which tries to distribute the input patterns evenly amongst the map units. An interesting feature of these papers is their use of color-coding to extract information from 3-dimensional SOMs.

Most other utilizations of SOMs in GISience have used the standard SOM algorithm, so only a general overview is provided here. Standard SOMs have been used for classification or residential areas (Openshaw et al., 1995), for typification in cartography (Sester and Brenner, 2000), to visualize the evolution of geodemographic variables (Skupin and Hagelman, 2003), and to select cartographic obects (Jiang and Harrie, 2004). Three-dimensional SOMs have been used to analyze census data (Takatsuka, 2001). The main contribution of this paper is the 3-D visualization software (GeoVista studio) that allows a very interesting exploration of the data. Parallel implementations of SOMs have also been used (Openshaw and Turton, 1996) to enable very large spatial datasets to be analyzed.

Standard SOMs have been used for spatialization of various systems (Skupin, 2001, Skupin and Fabrikant, 2003), including the World-Wide-Web (Girardin, 1996) .

Although not an application in GIScience itself, in (Mark *et al.*, 2001) standard SOMs are used to explore onthological disctinctions of geographic terms, as they are perceived by a sample of ‘ordinary’ citizens.

The variants presented next, although in some cases developed for other types of problems, allow for the possibility to explicitly incorporate geo-references and take into account the special features of spatial information into the SOM algorithm. One of the fundamental ideas consists in embedding the first law of Geography (Tobler, 1970) into the SOM. This can be translated into classifying geographical neighbors in similar areas of the output space. A balance between geographical proximity and attribute proximity should be achieved. There are different ways to accomplish this objective as we will show.

Hierarchical SOMs

Hierarchical SOMs change the normal interconnections between units. They are often used in application fields where a structured decomposition into smaller and layered problems is convenient. One or more than one SOMs are located at each layer, usually operating on different thematic variables (see Figure 2.1).

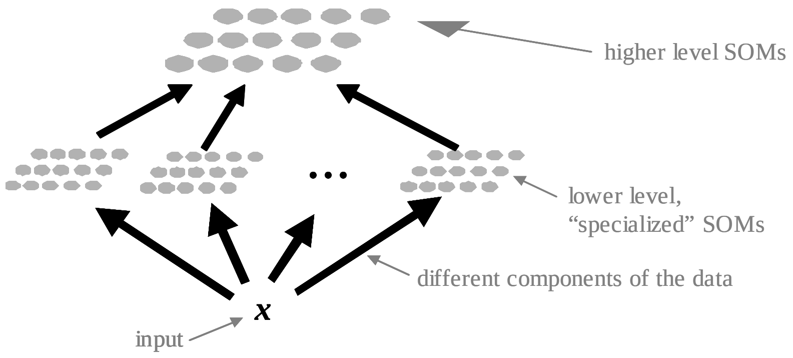


Figure 2.1 - Structure of a hierarchical SOM.

Hierarchical SOMs were first introduced in (Ichiki *et al.*, 1991), and were extensively used in speech recognition, where each layer deals with higher units of speech, such as phonemes, syllables, and word parts (Kempke and Wichert, 1993, Behme *et al.*, 1993, Jiang *et al.*, 1994). Hierarchical SOMs can have several lower level partial maps that cluster the data according to different characteristics and then pass the results to a upper level SOM, or they may have a lower level global SOM, that acts as a gating mechanism to activate one of several higher level SOM that specialize in a certain area of the input space.

In terms of GIScience one can envision the usefulness of hierarchical SOM in applications like geodemographics. Hierarchical SOMs allow the creation of purpose-specific or thematic classifications at lower layers which are then composed into a single one. This can constitute a major advantage as it has been noted that the single purpose geodemographic classifications constitute more powerful tools than general purpose classifications (Openshaw and Wymer, 1995). Additionally, the exploration of different SOMs at lower levels can be very valuable, especially if done in a computational environment where dynamic linking between SOMs can be set up. This way the interactive exploration of the different classifications can provide major insights. The idea is to take one of the SOMs, select specific units and study the distribution of the input patterns classified in that particular unit in the other SOMs and in the geographic space.

Geo-enforced

One simple way of producing quasi-variants and testing spatial effects is through pre-processing. Instead of altering the basic SOM algorithm the idea is to include spatial-relevant variables which are computed as any other socio-economic variable (Lobo *et al.*, 2004). This way there are two major operational decisions to be made. The first has to do with the choice of the spatial variables to use. The possibilities are endless, and dependent on the objectives pursued. The second one has to do with the weight that should be attributed to the geographic variables. Once all the variables are normalized the user has the possibility of deciding how much weight each of the variables will have in the calculations, thus giving more or less importance to the geographic information. As this approach cannot be considered a variant, it will not be discussed here.

Geographical Hypermap

In this approach, we change the matching and voting phase of the SOM algorithm. The Hypermap architecture was originally proposed in (Kohonen, 1991) for the recognition of phonemes in the context of cepstral features. In this architecture the input vector is decomposed into two distinct parts, a “context” vector and a “pattern” vector. The basic idea lies on treating both parts in different ways. The most common way is to use the context part to select the best match, and then adapt the weights using both parts, separately or together. However many other variants exist. In a geographic context using a Hypermap implies that the classification of a specific input vector is learned in the context of its geographic best match. In other words, the Geographical Hypermap will force the classification of input patterns based solely on their geographic positioning. This way each unit in the SOM will be a simple average of non-geographic attributes in the geographic area it covers. The smoothing effect of this averaging depends on the number of units and the density of input patterns.

Spatial-Kangas Map

This approach is yet another way of changing the matching and voting phase of the basic SOM. The Spatial-Kangas map introduced in (Lobo et al., 2004) is based on the temporal SOM first presented in (Kangas, 1992) and commonly known as Kangas map. The Spatial-Kangas map extends the underlying principles of the Hypermap, in the sense that the BMU is required to be in the geographical neighborhood of the input pattern. However in this approach the requirement that the BMU be geographically closest unit is relaxed, requiring only that it be close (within a certain radius named “geographical tolerance”). This is done by dividing the search for the BMU in two phases: first establish a geographical neighborhood where it is admissible to search for the BMU (see Figure 2.2), and then perform the final search using the non-geographical components of the input pattern. This can be seen as the separation of the vector into two different parts; one that carries the geographic context of the pattern, and a second one that provides information for the definition of the BMU within the context.

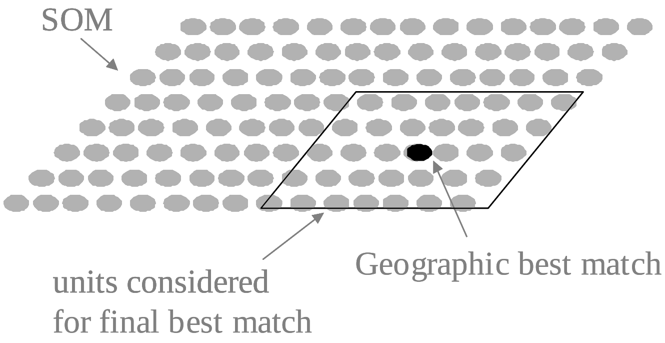


Figure 2.2 - Spatial-Kangas map structure (Geo-SOM with *k*>0)

Geo-SOM

One again, this variant changes the matching and voting phase of the basic SOM. The Geo-SOM constitutes our own conceptualization (Bação *et al.*, 2004b) of the Hypermap and the

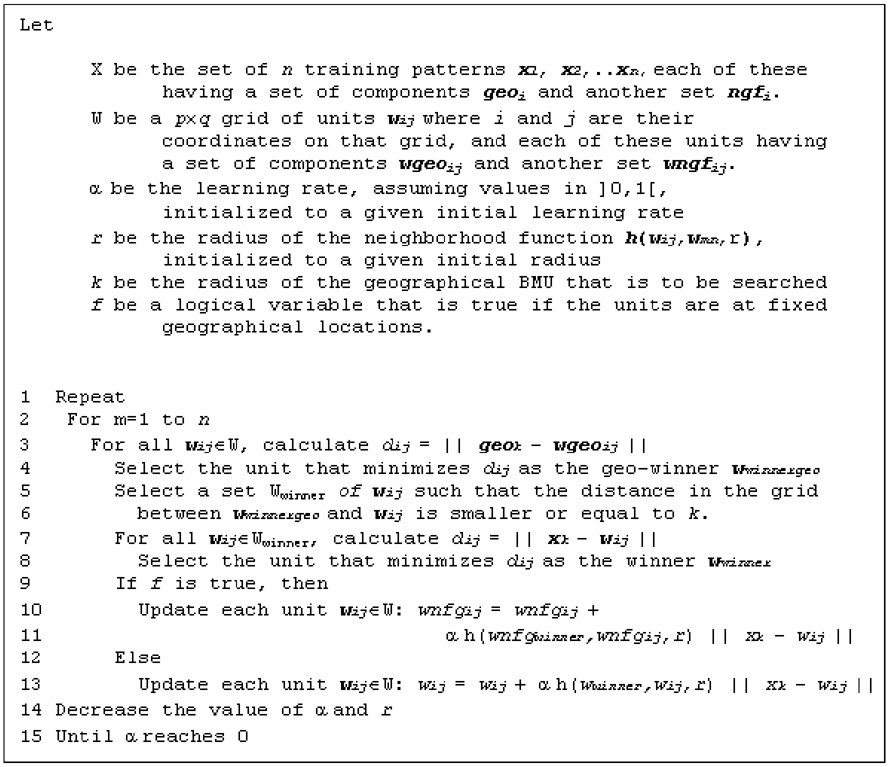
Spatial-Kangas Map, and can be viewed as a generalization of the concepts presented in the three previous points.

The geographic neighborhood where we search for the BMU can be controlled by a parameter *k*, defined in the output space, and called geographical tolerance. If we choose *k*=0, then the BMU will necessarily be the unit geographically closer, which corresponds to a Hypermap. If we allow *k* to grow up to the size of the map then the geographical coordinates will be ignored, corresponding to the original SOM. For *k* between 0 and the size of the map a Spatial-Kangas map will be obtained.

When *k*=0, the final locations in the input space of the units will be a quasi-proportional representation of the geographical locations of the training patterns (the proportionality is not exact due to the already discussed magnification effect), and thus the units will have local averages of the training vectors. Exactly the same final result may be obtained by training a standard SOM with only the geographical locations, and then using each unit as a low pass filter of the non-geographic features. The exact transfer function (or kernel function) of these filters depends on the training parameters of the SOM, and is not relevant for this discussion.

As *k* (the geographic tolerance) increases, the unit locations will no longer be quasi-proportional to the locations of the training patterns, and the “equivalent filter” functions of the units will become more and more skewed, eventually ceasing to be useful as models.

Formally, the Geo-SOM may be described by the following algorithm:



```
Let

      X be the set of n training patterns $x_1$, $x_2$,..$x_n$, each of these
            having a set of components $geo_i$ and another set $ngf_i$.
      W be a $p \times q$ grid of units $w_{ij}$ where $i$ and $j$ are their
            coordinates on that grid, and each of these units having
            a set of components $wgeo_{ij}$ and another set $wngf_{ij}$.
      $\alpha$ be the learning rate, assuming values in ]0,1[,
            initialized to a given initial learning rate
      $r$ be the radius of the neighborhood function $h(w_{ij}, w_{mn}, r)$,
            initialized to a given initial radius
      $k$ be the radius of the geographical BMU that is to be searched
      $f$ be a logical variable that is true if the units are at fixed
            geographical locations.


Repeat
 For m=1 to n
   For all $w_{ij} \in W$, calculate $d_{ij}$ = || $geo_k - wgeo_{ij}$ ||
     Select the unit that minimizes $d_{ij}$ as the geo-winner $W_{winnergeo}$
     Select a set $W_{winner}$ of $w_{ij}$ such that the distance in the grid
       between $W_{winnergeo}$ and $w_{ij}$ is smaller or equal to $k$.
     For all $w_{ij} \in W_{winner}$, calculate $d_{ij}$ = || $x_k - w_{ij}$ ||
       Select the unit that minimizes $d_{ij}$ as the winner $W_{winner}$
     If $f$ is true, then
       Update each unit $w_{ij} \in W$: $wnfg_{ij} = wnfg_{ij} +$
                                   $\alpha\, h(wnfg_{winner}, wnfg_{ij}, r)$ || $x_k - w_{ij}$ ||
     Else
       Update each unit $w_{ij} \in W$: $w_{ij} = w_{ij} + \alpha\, h(w_{winner}, w_{ij}, r)$ || $x_k - w_{ij}$ ||
Decrease the value of $\alpha$ and $r$
Until $\alpha$ reaches 0
```

The Geo-SOM has the potential to organize the SOM output space according to the geographic proximities of the input patterns. This way, areas of the geographic map with similar characteristics will warrant a smaller number of units than the areas of the map where characteristics differ a lot. One of the potential applications for the Geo-SOM is to develop homogeneous zones. Contrary to most zone design algorithms (Horn, 1995, Mehrotra et al., 1998, Macmillan and Pierce, 1994, Alvanides and Openshaw, 1999), in which the number of zones is pre-defined, the Geo-SOM can be viewed as an exploratory technique to build zones, as will be shown in the Application section of this chapter.

EXPERIMENTAL RESULTS WITH ARTIFICIAL DATA

In order to assist the comprehension of the major characteristics and properties of the some SOM variants, we carry out a set of tests based on artificial data. The objective of using artificial data is to produce a controlled environment where certain features of the variants can easily be understood. We constructed a toy problem comprising 5000 data points, each of which has geographical coordinates (*x* and *y*), and a third variable *z* that represents non-geographical data. The points follow a uniform distribution in the geographical coordinates, within the rectangle limited by [(0,0), (20, 5)] (see Figure 2.3). In the non-geographical dimension there are three zones of high spatial autocorrelation, where the values of *z* are very similar among neighboring points, with a uniform in the interval [90, 91] in two zones, and in the interval [10,11] in another. There is also one area of 'negative autocorrelation', where half the data points have z ≈10 and the other half have z ≈90. In the rest of the input space *z* has a uniform distribution in [0,100].

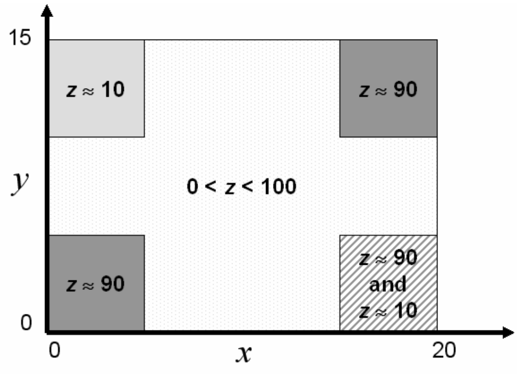


Figure 2.3 - Artificial dataset.

Five different SOM were used to process the data: a standard SOM, a Geo-SOM with *k*=0 (which is similar to the Hypermap), a Geo-SOM with *k*=1, *k*=2 and *k*=4. The *k* parameter, which we call geographic tolerance, refers to the size of the neighborhood amongst which the BMU will be searched.

In order to get a clear image of the error produced by each one of the tested variants we decided to separate the error in geographic error and quantization error. The geographic error computes the average distance between each input pattern and the unit to which it was mapped. This gives a notion of the geographic displacement of the units relatively to the input patterns they represent. The quantization error provides an assessment of the distances between input patterns and the unit to which they are mapped in the attribute space, in this case the *z* variable. The quantization error provides a measure of the quality of the representation of *z* (non-geographical attribute) achieved by each variant.

The results are quite elucidative in the sense that they allow a very clear distinction between the behaviors of the different variants. Clearly, the restrictions imposed by Geo-SOM tend to degrade the quantization error and improve the geographic error. In terms of quantization error the highest value is observed, as would be expected, in the Geo-SOM with the smallest geographic tolerance, and decays as *k* increases, until reaching the minimum with the standard SOM. Conversely, the geographic error decreases as *k* increases. The actual values may be seen in Table 2.1.

Type of map *vs* Type of error	GeoSOM *k*=0	GeoSOM *k* =1	GeoSOM *k* =2	GeoSOM *k* =4	Standard SOM
Geographical error	0.4193	0.8507	1.1800	1.5055	1.6713
Quantization error	21.0690	12.5902	7.1130	2.4440	0.9030

Table 2.1 – Average geographical and quantization errors for the artificial dataset

The quantization errors shown in the table are averages for all data patterns, and the individual values vary quite a lot. A close inspection of the way this quantization varies allows us to identify different clusters, which is one of the main purposes of using these techniques. If we calculate the average quantization error of the input patterns that are mapped to each individual unit and plot these values in a contour plot, we obtain the results presented in Figure 2.4. In this figure we plot the quantization error as a function of the geographical coordinates when using Geo-SOMs with *k*=0 and *k*=2, and using the standard SOM.

With *k*=0 the Geo-SOM is basically performing local averages. The points where those averages are calculated follow the geographical distribution of the input patterns, which in this case means they are evenly distributed. Areas where “natural” clusters exist are clearly shown by white areas, where the quantization error is low. Areas where there is less spatial autocorrelation are represented in progressively darker shades of gray, corresponding to increasing quantization errors. From this map little can be inferred about how to define regions in those areas. Thus, the choice of *k*=0 allows us to identify only the clearly homogeneous areas.

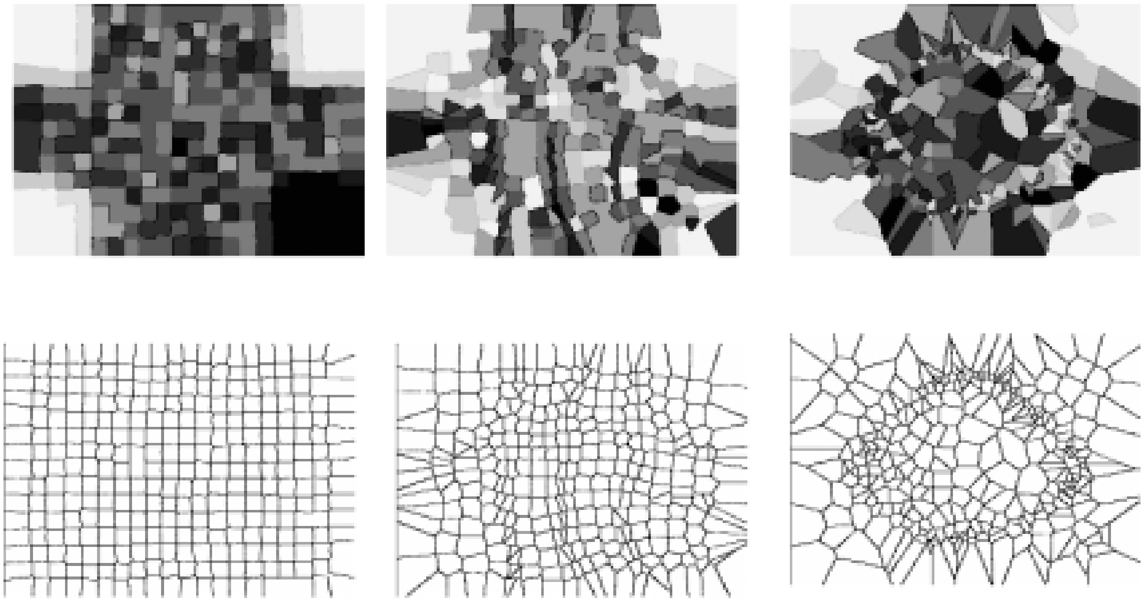


Figure 2.4 - Maps with the average quantization error per unit (top), and geographical coverage of those units (bottom), using the Geo-SOM with *k*=0 (left), *k*=2 (center) and the standard SOM (right)

With *k*=2 the Geo-SOM provides interesting insights into the data. Clearly homogenous areas are still very evident, but some new areas with low quantization error appear throughout the map. The lower right corner, where the data follows two distinct behaviors is divided (approximately along its diagonal) into two homogenous areas, one containing each type of data. These are separated by another area that serves as border, where the quantization error is quite large. A careful inspection of the remaining area shows stripes of low quantization error. These areas of low quantization are in general surrounded by sharp frontiers, which can be easily identified by the presence of smaller than average Thiessen polygons. As a conclusion, this map allows us to gain insight into less well structured areas of the data.

Finally, when using the standard SOM, the map has little information about the geographical organization of clusters. Since these are defined mostly by non-geographical attributes, their geographical location is basically meaningless and may lead to errors. The lower left corner of the map has basically the same configuration as the other corners even though the data in that corner are significantly different. We may thus conclude that while the standard SOM may be a good clustering tool, it naturally fails to single out the geographical information contained in it.

APPLICATION IN A REAL WORLD PROBLEM

For this study we used data form the INE (Portuguese bureau of statistics), referring to Lisbon’s Metropolitan Area. This data is at the Enumeration District (ED) level, and refers to 65 socio-demographic variables. These variables describe EDs based on 6 main topics: information about buildings, families, households, age structure, education levels, and economic activities. Additionally, we introduced two explicitly geographic variables, representing the *x* and *y* coordinates of the geometric centroids of the EDs.

All variables used were made to be invariant to size, by calculating ratios whenever necessary. Further more, all were normalized to be in the [0, 1] interval. In the particular case of the (*x*, *y*) coordinates, and in order to preserve the original form of the region, the *y* coordinate was allowed to have a different range so as to keep proportionality with the *x* coordinate.

As with the artificial dataset problem, we used the standard SOM algorithm and compared it with the Geo-SOM with *k*=0, 1, 2, and 4. In Table 2.2 the average geographical and quantization errors are presented, and they follow exactly the behavior explained for the artificial dataset.

Type of map *vs* Type of error	GeoSOM *K*=0	GeoSOM *k* =1	GeoSOM *k* =2	GeoSOM *k* =4	Standard SOM
Geographical error	0.0096	0.0195	0.0291	0.0457	0.1110
Quantization error	0.9446	0.8763	0.8265	0.7786	0.6488

Table 2.2 - Average geographical and quantization errors for Lisbon's EDs dataset.

Figure 2.5 shows the results obtained. The standard SOM (presented on the right hand side) provides no relevant insight into the geographical organization of the data. The map units geographical coordinates are mostly close to each other in the center of the map, and the areas with low quantization error do not correspond to any meaningful regions. A careful inspection of the EDs that are clustered together shows that these are in fact quite far away from each other, and the geographical centers of these clusters are meaningless. This is due to the fact that in order to maximally reduce the quantization error the best positioning of the units is achieved in central areas.

The Geo-SOM with *k*=0, although merely calculating local averages, has some advantages over other methods. These averages are calculated with more resolution in areas where there are more EDs, following the geographic density of the input patterns, which is a desirable effect. From observing the map we can easily identify areas with strong spatial autocorrelation (depicted in light grey), namely the central area of Lisbon (centre of the map), the area behind the south-side beaches, etc. We can also identify areas where the quantization error is large (dark grey), indicating that there are large variations in the socio-demographic characterization.

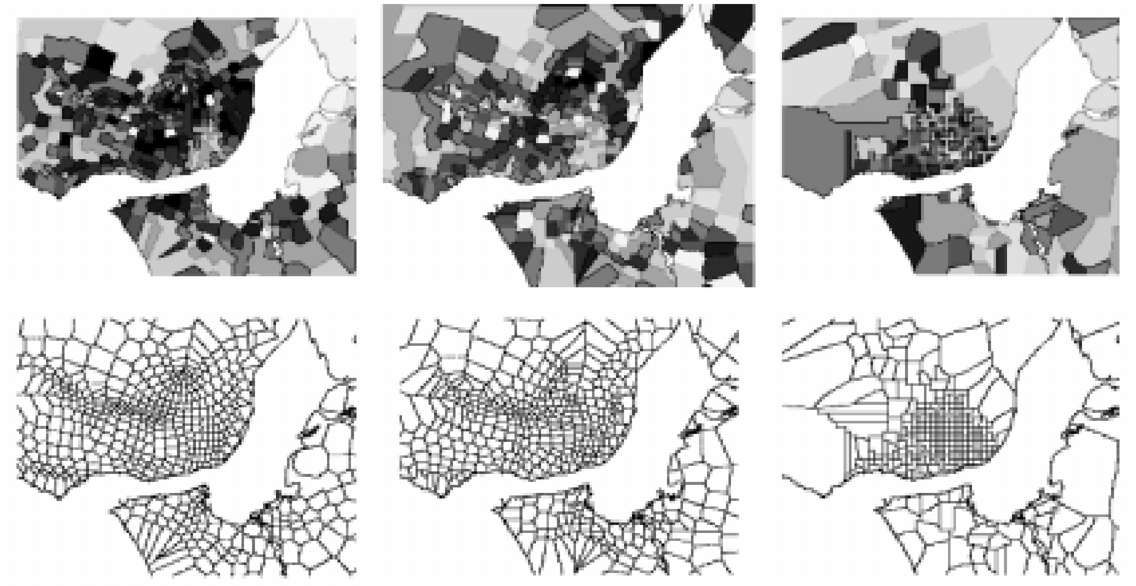


Figure 2.5 - Maps with the average quantization error per unit (top), and geographical coverage of those units (bottom), using the Geo-SOM with *k*=0 (left), *k*=2 (center) and the standard SOM (right). The data refers to socio-demographic variables of Lisbon are EDs.

The Geo-SOM with *k*=2 provides further insight on how to define homogenous zones. Some of the previously identified zones increase in size or shift slightly, and a number of new clusters appear on this map. These new regions can be identified with smaller scale areas, which in fact correspond to distinctive areas of Lisbon.

After processing the data with the Geo-SOM, the user can explore the results and build homogenous zones. The procedure for this is quite simple. The basic premise is that areas of low quantization error (represented in the maps by light shades of grey) indicate homogenous areas, and as quantization error increases the homogeneity decreases. An interesting way of approaching the problem is to produce a surface based on the quantization error (Figure 2.6) and flood the surface at different levels. With small amounts of water only very similar areas will be joined together (areas below water level) but, as the amount of water in the surface increases, less similar areas will be joined together. Eventually, only ridges (boundaries formed by areas of high quantization error) will be above water level. A parallel can be drawn between this approach to zone design and hierarchical clustering. As in hierarchical clustering the Geo-SOM starts with very small homogenous areas (only areas with a very small quantization error) and as the surface is flooded the number of homogenous areas decreases but their dimension increases.

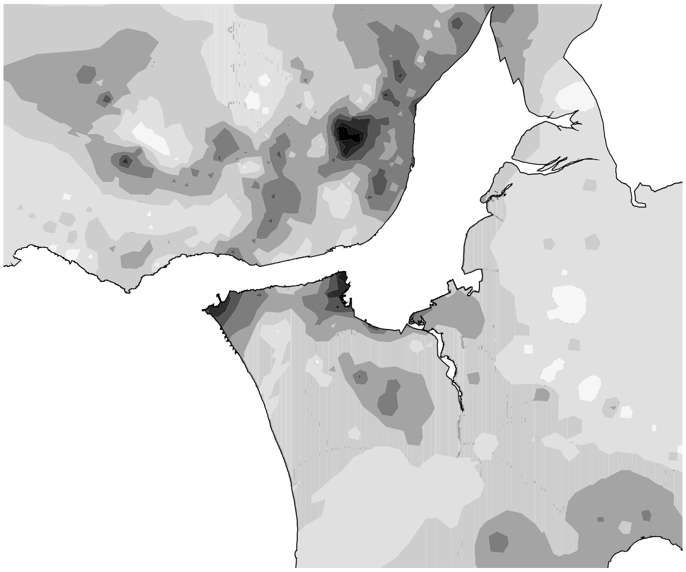


Figure 2.6 - A contour plot of the quanization error for the Lisbon Metropolitan Area

Clearly, different *k* values can be tested in the Geo-SOM and the results compared, enabling the user to understand the major spatial trends in the data. One of the major advantages of this methodology is that it addresses the problem of fuzzy classification, providing an alternative to “clear-cut” classifications which present some major drawbacks (Feng and Flowerdew, 1998, Openshaw and Wymer, 1995). While not completely inclusive (in the sense that it doesn’t force all areas to be included in clusters) this approach identifies both the homogenous zones and areas which are not included in any zone because of their dissimilarity with their neighbors.

CONCLUSIONS

There are many ways in which the standard SOM can be used in Geographical Science problems, and there are many variants that explicitly take into account their spatial nature. An overview of the different SOM variants was presented. A more detailed explanation of one of the geographical oriented variants was given together with an example of its application to an artificial problem and a real world problem. It was shown that in the latter case, this approach can provide a meaningful insight to the spatial data structure.

The practical application showed that it is possible to use the Geo-SOM to approach the zone design problem in an exploratory perspective. Instead of defining the number of zones at the beginning of the procedure, the Geo-SOM “lets data speak for themselves” (Gould, 1981), allowing an insightful analysis. The Geo-SOM can be thought of as a method which projects multidimensional data into the geographic space. The amount of geographic error and quantization error is controlled by the *k* parameter. Thus, as *k* increases the geographic error also increases and the quantization error decreases. The user must experiment with different *k* values in order to strike an adequate trade-off.

There are a number of issues that remain to be explored in the Geo-SOM. The effect that the relation between the density of the input patterns (in the geographic space) and the distance between them (in the variable space) has on the distribution of the units is still an open problem. Another interesting issue to address in future developments is the possibility of using dynamical *k* values. The idea is to adequate the *k* parameter according to the specific spatial autocorrelation index of the area of the input pattern.

Looking at the number of different SOM variants proposed in other research fields, it is quite probable that in the future GIScience will also “create” its own variants. In fact, the flexibility of the algorithm and its wide range of application can be seen as invitations to researchers to adapt the original SOM to specific paradigms and problems of GIScience.

All the programming routines for the architectures presented here are available at: www.isegi.unl.pt/docentes/vlobo/projectos/programas/programas.html

REFERENCES

Almeida, L. B. and Rodrigues, J. S. (1991) *Neural Networks, Elsevier*, 63-78.

Alvanides, S. and Openshaw, S. (1999) In *Geographical Information and Planning*(Eds, Stillwell, J. C. H., Geertman, S. and Openshaw, S.) Springer-Verlag, pp. 299-315.

Bação, F., Lobo, V. and Painho, M. (2004a) In *KD-Net Symposium 2004 - Knowledge-based services for the public sector*Bonn, Germany.

Bação, F., Lobo, V. and Painho, M. (2004b) In *GisScience 2004*.

Bauer, H.-U., Der, R. and Herrmann, M. (1996) *Neural Computation,* **8,** 757-771.

Bauer, H.-U. and Villmann, T. (1997) *IEEE Transactions on Neural Networks,* **8,** 218 - 226.

Behme, H., Brandt, W. D. and Strube, H. W. (1993) In *ICANN 93*Springer, pp. 416-419.

Buessler, J.-L., Urban, J.-P. and Gresser, J. (2002) *Neural Processing Letters,* **15,** 9-20.

Camastra, F. (2001) *Neural Processing Letters,* **14,** 27-34.

Chandrasekaran, V. and Liu, Z.-Q. (1998) *IEEE Transactions on Neural Networks,* **9,** 483-502.

Claussen, J. C. (2003) *Complexity (Wiley),* **8,** 15 - 22.

Cottrell, M., Fort, J. C. and Pages, G. (1998) *Neurocomputing, Elsevier,* **21,** 119-138.

Feng, Z. and Flowerdew, R. (1998) In *Innovations in GIS 5*(Ed, Carver, S.) Taylor & Francis, London, pp. 119-127.

Fritzke, B. (1991) In *ICANN-91*Elsevier Science Publ., Helsinki.

Fritzke, B. (1994) In *NIPS*Denver, USA.

Fritzke, B. (1995) *Neural Processing Letters,* **2,** 9-13.

Fukunaga, K. and Olsen, D. R. (1971) *IEEE Transactions on Computers,* **c-20,** 176-183.

George, S. (2000) *Knowledge and Information Systems (Springer-Verlag),* **2,** 359-372.

Gioiello, M., Vassallo, G. and Sorbello, F. (1992) In *International Conference on Signal Processing Applications and Technology*Boston.

Girardin, L. (1996) In *Fifth International World Wide Web Conference*Paris, France.

Gould, P. (1981) *Annals of the Association of American Geographers,* **71,** 166-176.

Guimarães, G., Lobo, V. and Moura-Pires, F. (2002) *Intelligent Data Analysis,* **7**.

Horn, M. E. T. (1995) *Geographical Analysis,* **27,** 230-248.

Ichiki, H., Hagiwara, M. and Nakagawa, N. (1991) In *IJCNN'91, International Conference on Neural Networks*(Ed, Center, I. S.), pp. 357-360.

Jiang, B. and Harrie, L. (2004) *Transactions in GIS,* **8,** 335-350.

Jiang, X., Gong, Z., Sun, F. and Chi, H. (1994) In *WCNN'93 - World Conference on Neural Networks*Lawrence Erlbaum, Hillsdale.

Kangas, J. (1990) In *Proc. IJCNN-90, International Joint Conference on Neural Networks, San Diego*, Vol. II IEEE Computer Society Press, Los Alamitos, CA, pp. 331--336.

Kangas, J. (1992) In *Artificial Neural Networks*, Vol. 2 (Ed, I. Aleksander, J. T.) Elsevier Science Publisher, pp. 117-120.

Kangas, J. A., Kohonen, T. K. and Laaksonem, J. T. (1990) *IEEE Transactions on Neural Networks,* **1,** 93-99.

Kempke, C. and Wichert, A. (1993) In *WCNN'93 - World Conference on Neural Networks*Lawrence Erlbaum, Hillsdale.

Kohonen, T. (1991) In *Artificial Neural Networks,* Vol. 1 (Eds, Kohonen, T., Mäkisara, K., Simula, O. and Kangas, J.) Elsevier Science Publishers, pp. 1357-1360.

Kohonen, T. (2001) *Self-Organizing Maps,* Springer.

Kohonen, T., Hynninen, J., Kangas, J. and Laaksonen, J. (1995) Helsinki University of Technology.

Lee, J. A., Donckers, N. and Verleysen, M. (2001) In *Advances in Self-Organizing Maps*(Eds, Allinson, N., Yin, H., Allinson, L. and Slack, J.) Spinger.

Lobo, V. (2002) In *Departamento de Informatica*Universidade Nova de Lisboa, Lisbon.

Lobo, V., Bação, F. and Painho, M. (2004) In *AGILE 2004*Heraclion, Greece.

Lourenço, F., Lobo, V. and Bação, F. (2004) In *JOCLAD 2004 - XI Jornadas de Classificação e Análise de Dados*Lisbon.

Macmillan, W. D. and Pierce, T. (1994) In *Spatial Analysis and GIS*(Eds, Fotheringham, S. and Rogerson, P.) Taylor and Francis, London.

Maenou, T., Fujimura, K. and Kishida, S. (1997) In *Progress in Connectionsist-Based Information Systems. Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, Vol. 2 (Eds, Kasabov, N., Kozma, R., Ko, K., O'Shea, R., Coghill, G. and Gedeon, T.) Springer, Singapore, pp. 1013-1016.

Mark, D. M., Skupin, A. and Smith, B. (2001) In *Lecture Notes in Computer Science*, Vol. 2205 Springer-Verlag, Heidelberg.

Martinetz, T. M., Berkovich, S. G. and Schulten, K. J. (1993) *IEEE Transactions on Neural Networks,* **4,** 558-569.

Mehrotra, A., Johnson, E. L. and Nemhauser, G. L. (1998) *Management Science,* **44,** 1100.

Openshaw, S. (1999) In *GeoComputation '99*.

Openshaw, S., Blake, M. and Wymer, C. (1995) In *Innovations in GIS*, Vol. 2 (Ed, Fisher, P.) Taylor and Francis, pp. 97-111.

Openshaw, S. and Openshaw, C. (1997) *Artificial intelligence in geography,* John Wiley & Sons, Chichester.

Openshaw, S. and Turton, I. (1996) *Computers & Geosciences,* **22,** 1019-1026.

Openshaw, S. and Wymer, C. (1995) In *Census users handbook*(Ed, Openshaw, S.) GeoInformation International, Cambrige, UK, pp. 239-268.

Ritter, H., Martinetz, T. M. and Schulten, K. (1992) *Neural Computation and Self-Organizing Maps: an introduction,* Addison-Wesley.

Sester, M. and Brenner, C. (2000) In *GisScience 2000*.

Skupin, A. (2001) In *Workshop on Visual Interfaces to Digital Libraries*Roanoke, Virginia.

Skupin, A. and Fabrikant, S. I. (2003) *Cartography and Geographic Information Science,* **30,** 95-115.

Skupin, A. and Hagelman, R. (2003) In *eleventh ACM international symposium on Advances in geographic information systems*New Orleans, Louisiana, USA, pp. 56 - 62.

Takatsuka, M. (2001) In *GeoComputation (6th International Conference on)*Brisbane, Australia.

Tanomaru, J. I., A. (1995) In *IEEE International Conference on Neural Networks*, Vol. 5 Perth, WA, Australia, pp. 2432 -2437.

Tobler, W. (1970) *Economic Geography,* **46,** 234-240.

Ultsch, A., Guimarães, G., Korus, D. and Li, H. (1993) In *Transputer-Anwender-Treffen*Springer Verlag, Aachen, pp. 194-203.

Ultsch, A. and Li, H. (1993) In *International Conference on Signal Processing, Peking, 1993*Peking.

Ultsch, A. and Siemon, H. P. (1990) In *Intl. Neural Network Conf. INNC90*Paris, pp. 305-308.

Vesanto, J. (1999) *Intelligent Data Analysis,* **3,** 111-126.

Vesanto, J. (2000) HUT, Finland.

Villmann, T. and Merényi, E. (2001) In *Self-Organizing Maps: Recent Advances and Applications*(Eds, U.Seiffert and Jain, L. C.) Springer-Verlag, pp. 121-145.

Villmann, T., Merenyi, E. and Hammer, B. (2003) *Neural Networks,* **16,** 389-403.

Yin, H. (2001) In *Advances in Self-Organizing Maps*(Eds, Allinson, N., Yin, H., Allinson, L. and Slack, J.) Springer, pp. 81-88.