TOWARDS HIGH-RESOLUTION SELF-ORGANIZING MAPS OF GEOGRAPHIC FEATURES

André Skupin

Department of Geography
San Diego State University
San Diego, California, 92182
USA

skupin@mail.sdsu.edu

Aude Esperbé

Department of Geography
San Diego State University
San Diego, California, 92182
USA

esperbe@rohan.sdsu.edu

ABSTRACT

This chapter introduces the use of high-resolution self-organizing maps (SOM) to represent a large number of geographic features on the basis of their attributes. Until now, the SOM method has been applied to geographic data for both clustering and visualization purposes. However, the granularity of the resulting attribute space representations has been far below the resolution at which geographic space is typically represented. We propose to construct SOMs consisting of several hundred thousand neurons, trained with attributes of an equally large number of geographic features, and finally visualized in standard GIS software. This is demonstrated for a data set consisting of climate attributes attached to 200,000+ U.S. census block groups. Further, overlays of point, line, and area features onto such a high-resolution SOM are shown.

INTRODUCTION

This volume demonstrates the range of approaches currently pursued in the field of geographic visualization. Geographic visualization has clearly captured the public's imagination. Evolutionary changes in creation, distribution, and interaction with cartographic depictions have powerfully converged in early realizations of the *digital earth* concept (see chapter by Goodchild in this volume). Further convergence of various technologies and methodologies is likely, including trends towards high-resolution imagery (see preceding chapter by Orford) and locations captured using GPS or cell phones (see chapters by Wright et al and Kraak). One example of such convergence is the extension of space-time paths to *n*-dimensional attribute space (Skupin, in press).

Geographic visualization does not merely represent the reemergence of cartography, to who GIS was once thought to have delivered a deadly blow. Instead, the confluence of various academic and market forces towards formation of such new disciplinary categories as information visualization and visual data mining and the success of products and services like MapQuest and Google Earth suggests the emergence of a *cartographic imperative*, a *need to map*, aimed at making sense of voluminous, multi-facetted data. This imperative delivers a powerful impulse to create meaning-bearing visualizations, even of non-georeferenced data and of the non-spatial elements of geographic data. In order to achieve this, one is however required to shed notions of cartography as being essentially about attaching symbols to geometry in order to communicate geographic reality. On a much more fundamental level, such approaches as spatialization remind us that cartography is all about transformation (Tobler, 1979) and that the impact of visualization often derives from novel combinations of transformative processes.

As reflected in this volume, geographic visualization at its core grew out of the cartographic tradition of representing geographic objects in a representational space derived through projection of locations from a curved two-dimensional space (i.e., *latitude* and *longitude*) into a planar map space (i.e., *x* and *y*). New techniques of representation have emerged, such as parallel coordinate plots (PCP) and self-organizing maps (Kohonen, 2001), and are now being linked to form increasingly powerful means for discovering interesting patterns and relationships in large, multidimensional geographic databases. However, the geographic map tends to be the element binding it all together, the one with which all other representations interact and are bound to, and to which users will ultimately refer. One major reason for this is the well-deserved recognition given to the possible effects of spatial autocorrelation – or the First Law of Geography (Tobler, 1970) – in any such geographic investigation. In fact, effects of spatial autocorrelation, such as the appearance of spatial clusters, are the very subject of many investigations. However, apart from such effects, one might argue that an additional impetus for referring back to the geographic map is the sheer richness provided by it. Ultimately, this richness derives not simply from a choice of symbols for point, line, area, and text objects – because that would apply to many non-geospace representations as well – but from the finely grained geometric base to which such symbols become attached. The inherent geometric detail or resolution provided by a geographic map tends to remain unmatched by alternative representations.

Out of these considerations the theme of this chapter then emerges, to advance the convergence of intense computation with the cartographic tradition, to spatialize an *n*-dimensional geographic attribute space with a resolution detailed enough to mimic geographic maps, and to apply a range of transformations towards eventual visualization. This is demonstrated through a visualization derived from climate attributes associated with more then 200,000 U.S. census block groups.

SELF-ORGANIZING MAPS

First introduced a quarter-century ago, the self-organizing map (SOM) has become a popular method for visual modeling of complex, *n*-dimensional data. A number of excellent overviews of the method, edited volumes, as well as a comprehensive monograph on the subject exist (Deboeck and Kohonen, 1998, Oja and Kaski, 1999, Kohonen, 1982, Kohonen, 1990, Kohonen, 2001); so in this chapter the method is introduced in only the briefest terms. A SOM is an artificial neural network in which neurons are arranged as a low-dimensional, typically two-dimensional, lattice such that each neuron has either four or six neighbors (i.e., square or hexagonal neighborhood). Each neuron is associated with an *n*-dimensional vector of weights. Input data presented to the neuron lattice are of the same dimensionality. For example, 30 population attributes associated with 200,000 census enumeration units would correspond to a training data set consisting of 200,000 eleven-dimensional vectors.

During training, one input vector at a time is presented to all the neurons and the most similar neuron is determined, typically based on a Euclidean similarity coefficient. The *n* weights of that so-called best-matching unit (BMU) then get adjusted towards an even better match. More important – and essential for the self-organizing nature of a SOM – is that weights of neighboring neurons around the BMU are likewise adjusted, up to a certain neighborhood size and with a diminishing magnitude best described as distance decay. Over the course of many such training runs, the low-dimensional lattice of neuron vectors begins to replicate major topological structures existing in the *n*-dimensional input space.

A trained two-dimensional SOM can itself be visualized in various forms, including the display of weights for a particular variable as color shading across the neuron lattice. This is also known as component plane display and an example is included later in this chapter. One could also opt for a display based on multi-dimensional computation, such as clustering of neuron vectors using hierarchical or *k*-means clustering (Skupin, 2004). A very popular choice has been to visualize *n*-dimensional differences among neighboring neurons using the so-called U-Matrix method (Ultsch, 1993). Finally, the original input vectors or other vectors, if they contain the same variables and underwent identical preprocessing, could also be visualized. This involves finding for each the BMU from among the trained neurons and placing point symbols and text labels at that respective BMU's location.

Further computational and visual transformations may be desired, but existing SOM software is in fact severely limited in that respect. The vast majority of examples of SOM – at least when used for visualization purposes – make a choice among a limited number of available SOM software solutions. Extremely popular has been SOM software created by the Neural Networks Research Center at the Helsinki University of Technology. One important reason for this popularity is that the software is freely available, including access to the source code. *SOM_PAK* (Kohonen et al., 1996a) is a collection of programs written in C, which can be compiled for different platforms, though Windows executables are also available. It implements the standard SOM training algorithm and was used for all examples presented in this chapter. Its visualization functionality is however rudimentary. This was a major reason for our implementation of GIS-based storage and visualization of a trained SOM. From the same source as *SOM_PAK* comes the equally free *SOM Toolbox for Matlab* (though it requires Matlab to already be installed), which includes various visualization options. However, compared to graphic design or GIS software it is much harder letting a user's imagination drive the control and transformation of these visualizations. That is why the majority of visual examples of *SOM Toolbox* applications found in the literature have a fairly uniform appearance. That is also the case for most commercial SOM software, like *Viscovery SOMine* (http://www.eudaptics.de).

HIGH-RESOLUTION SOM

This section first presents the main arguments for wanting to build SOMs consisting of a very large number of neurons. Some examples for large SOM already exist and are discussed in the context of finally introducing our proposed high-resolution SOM, most notably using GIS-based storage and visualization of neuron geometry.

Rationale

The rationale for creating high-resolution SOMs derives from the desire to: (a) represent macro and micro structures existing in *n*-dimensional data; (b) use a trained SOM as a base map; and (c) leverage GIS technology.

When strictly used for clustering, a SOM will typically consist of only up to a few dozen neurons, especially if the attribute space portion occupied by an individual neuron is interpreted as a single cluster. Thus, a three-by-three neuron SOM trained with demographic attributes for fifty-one geographic objects – 50 states and the District of Columbia – would come to represent the attribute space in nine clusters, which in the case of attributes of geographic features can by readily visualized in *geographic space* (Figure 1). Such a SOM can also inform color design (not visible in the grayscale version) and provide a convenient, logical legend layout (see bottom right of Figure 1). However, such an extremely low-resolution SOM is barely useful for visualization of input features in *attribute space*. In fact, SOM software tends to have problems with multiple input features being mapped onto the same neuron, due to overplotting of symbols and labels. For example, in the *SOM Toolbox* only one feature vector can be labeled at each neuron location and any further input vectors at that location remain basically invisible. One solution is to map features randomly near the respective best-matching neuron (Skupin, 2002), as seen in Figure 2. Fifty-one geographic objects are here mapped onto the same nine-neuron SOM. However, this is a purely graphic solution and, similar to the mixed-pixel problem known in raster GIS, the model provides no means to actually distinguish *n*-dimensional differences among vectors assigned to the same neuron. In more general terms, one can say that a low-resolution SOM only allows visualizing global or macro structures existing in n-dimensional data, while finer structures remain hidden.

Akin to resolution effects in raster GIS, the approach proposed here is to provide a larger number of map units (i.e., neurons). For example, a 20-by-20 neuron SOM provides 400 different map units onto which the 51 geographic objects can be mapped (Figure 3). It is important to note that such a SOM at this point stops functioning as a clustering method, because a 400-cluster solution for 51 objects is not very useful. Instead, the SOM allows detailed two-dimensional layout of the geographic objects. Those states still assigned to a single neuron (e.g., Louisiana (LA) and Mississippi (MS)) are too similar to be distinguishable even at this level of granularity.

The detail provided in high-resolution SOMs makes it possible to have them play the role of a base map onto which various other data could be mapped. This is particularly true due to the fact that a SOM does not directly represent the input vectors as such, in contrast to such methods as multidimensional scaling (MDS) or spring models. Instead it creates a low-dimensional output *model* of the *n*-dimensional input space. That model can be applied to other data, as long as they have the same dimensionality. Once those data are mapped onto the SOM, other features can be attached. For example, if a SOM is constructed from multi-temporal demographic attributes of geographic objects, one could link individual temporal vertices to form trajectories and then visualize previously unrelated attributes onto those trajectories (Skupin and Hagelman, 2005). From clustering to labeling of neuron regions, a number of transformations have been proposed that all depend on a view of high-resolution SOMs as base maps (Skupin, 2004).

Most SOM software solutions provide limited support for effectively storing and transforming large neuron lattices and derived data, such as trajectories and surfaces. One alternative is to leverage the ability of GIS to dealing with large, low-dimensional geometric data sets. Within GIS one can first chose among the various geometric data models, has access to various database solutions, and can perform a wide array of transformations, from interpolation to overlay operations. Finally, in the hands of a cartographer, GIS can produce attractive visualizations with a large degree of automation (for example for the complex task of feature labeling), while still performing data-driven visualization. Use of GIS can thus make high-resolution SOM a much less daunting proposition on many levels.

Examples of high-resolution SOM

Most SOM implementations are based on lattices of no more than a few hundred neurons, and typically much less than that. A few examples for large SOMs exist though. Most of these were in fact created by the research group around the method's inventor, Teuvo Kohonen. In the mid-1990s they mapped more than 130,000 newsgroup postings onto a SOM that eventually consisted of 49,152 neurons, though in a two-stage process that began with a much smaller SOM of 768 units, from which the larger SOM was interpolated and further training was then applied (Kohonen et al., 1996b). By far the largest SOM known was created from the text of almost seven million patent applications (Kohonen, 2001). Training was a three-step process, during which progressively finer SOMs were created beginning with a 435-neuron SOM and eventually leading to a model consisting of 1,002,240 map units. Training took six weeks on a six-processor computer system.

Training speed is not merely a function of the number of neurons, but also of the model's dimensionality. Text documents tend to be represented with much longer vectors than other data. The demographic data visualized in Figures 1-3 includes 32 attributes, while Skupin's visualization of AAG conference abstracts represented each abstract as a 741-dimensional vector (Skupin, 2002). At that time, training of a 4800-neuron SOM with the conference abstracts took three hours. Training of a much higher-resolution, yet very low-dimensional, SOM can be quite fast. An extreme example is probably the projection of geographic coordinates (without consideration of any other attributes) into a SOM space consisting of 125,000 neurons (Skupin, 2003), which took 48 hours on an 800 MHz Pentium III PC and resulted on an odd new form of map projection (Figure 4). Note that there are many more factors influencing the speed of SOM training, including the number of training cycles and the specific SOM algorithm used (e.g., the later stages of training for the patent SOM used a variation known as the *Batch Map*).

Proposition

The core idea advocated in this chapter is to use the SOM method to project geographic objects into a finely grained display space in order to provide a different, yet equally rich and holistic perspective on geographic phenomena as that provided in traditional map space. While the latter is based on location given in geographic coordinates, the former will be constructed from the objects' attributes.

When dealing with non-georeferenced data, such as text documents, a high-resolution spatialization can become the center of a visualization system because it is often the first and only such visual depiction and has the potential for becoming the central access mechanism for complex data and, with wide-spread acceptance, even developing iconic and reference status for a large user group, for example in the visualization of scientific knowledge domains (Shiffrin and Börner, 2004). This is different for georeferenced data, where the geographic map naturally maintains a central role, due to the already discussed spatial autocorrelation effects. However, a detailed visualization derived from just the non-spatial attributes can provide an alternative perspective on geographic phenomena. This point is driven home by another aspect of our proposal, which is to juxtapose geographic and attribute space depictions while deliberately applying uniform designs and thus allowing the data to 'speak' about commonalities and differences between the two visualizations.

GIS can play a central role in implementing high-resolution SOMs. While SOM training functionality is generally not provided in GIS (with the exception of some functions available in *IDRISI* software), the low-dimensional neuron lattice of a trained SOM can readily be represented using GIS data models. There is also plenty of flexibility when it comes to using multiple data models, like vector and raster. For example, the vector data model may be used to represent the locations of vectors mapped on the SOM or the trajectories of features over time. Component planes – each representing a single variable retrieved for all neurons – could be represented using a polygon structure, but for very large SOMs it becomes more efficient to represent component planes as rasters interpolated from neuron centroids. From the interpolation of these component landscapes to the dissolving of boundaries during cluster visualization, GIS provides a large toolset readily usable for spatialization. Another advantageous aspect is the traditional integration of spatial and non-spatial attributes in GIS databases.

HIGH-RESOLUTION SOM FOR CLIMATE ATTRIBUTES

The high-resolution SOM demonstrated here is related to one presented by Skupin (in press), in which demographic attributes for all 200,000+ census block groups were spatialized in a SOM. Space-time paths captured in different cities and based on different modes of transport were then projected onto that SOM by tracing the sequence of transected block groups. One future direction of that effort is to combine demographic attributes with other human and physical attributes towards a rich, attribute-based model of geographic space. Ultimately, we would like to create a single SOM combining all of these attributes, which requires integrating attributes originating in very different domains and stored in different formats and attach them to identical geographic features. The latter could be of uniform shape and size, like raster cells, or one could use varied features, like polygons of different shape and size. To that end, the implementation described in this chapter demonstrates how a set of physical attributes – specifically climate attributes – are summarized for census block groups, which are then spatialized. This allows experimentation with a number of interesting aspects, including performing complex overlay procedures for transferring attributes from raster grids to several hundred thousand polygon features.

Due to the relatively smooth variation of climate attributes across space, using only the climate data allows for more detailed observation of differences between the geographic and attribute space visualization. With dominant spatial autocorrelation effects, neighboring regions in geographic space will tend to have similar climate and they should thus remain in close proximity in the spatialization. Where that is not the case, one is either dealing with pronounced geographic structures, characterized by rapid change of attribute values across space, or with distortion caused by the dimensionality reduction technique.

Climate Source Data and Preprocessing

The data chosen for this study consisted of 11 climate attribute attached to point locations in the contiguous states of the U.S. (48 states plus the District of Columbia). Data were obtained from the Web site of the National Climatic Data Center (http://www.ncdc.noaa.gov/oa/ncdc.html). The point attributes used included annual averages of:

- the numbers of days classified as cloudy, clear, or sunny
- humidity
- precipitation

- snowfall
- average, minimum, and maximum temperature
- average and maximum wind speeds

Note that the attribute data obtained consisted of only just over 200 points distributed across the contiguous United States. Future studies will include a much larger point data set, thus providing a better match between the granularity of these data and that of the block groups and the high-resolution self-organizing map.

The methodology then called for interpolation of all attributes to continuous raster grids, followed by a zonal average computed for each block group. This created unexpected challenges to the creation of appropriate source data before SOM training could even occur. Note that block groups represent a detailed tessellation of geographic space into areas of varied shape and size. Each block group is an aggregation of several census blocks into units containing around 1,500 persons, with a range of around 600 to 3000 persons. Thus, block groups in rural areas can be quite large, while urban block groups literally consist of only a few city blocks. Given this potentially very small size, the interpolation method and method-specific settings have to be carefully chosen. To that end, all attributes underwent a rigorous process of cross validation, where one point observation at a time is removed, interpolation is performed and predicted and known values are compared. When this is done for all points, a summary measure based on a root mean square error (RMS) can be computed. This was performed for several dozen combinations of interpolators and settings. In all cases, some variation of kriging produced the best result, though with different models (e.g., spherical, Gaussian, etc.) reflecting different patterns of spatial variation for the various attributes.

The most difficult lessons learned in the preprocessing of SOM training data related to the limits of current commercial off-the-shelf (COTS) GIS software in performing spatial analysis on very large data sets. Throughout this process, *ArcGIS 9.1* was used, including the *Geostatistical Analyst* extension. Given the potentially very small size of block groups, attributes were at first interpolated at a pixel resolution of 1 km^2. Then, a zonal average was attempted to be computed for each of the 200,000+ block groups. However, even at a pixel size of 1x1 km, standard zonal operators fail for a large number of urban block groups, because they do not contain a single pixel centroid. This was circumnavigated by converting pixel centroids to points, resulting in very large point files. There are numerous ways to then implemented point-to-polygon transfer of zonal attributes, most of which work reasonably well for small subsets, like for a single city or county, consisting of up to a few thousand block groups. However, even for those subsets, overlay operations take a significant amount of time. For larger data sets, execution would theoretically take several days, but overlays quickly run into limits determined by available RAM. Resolution of interpolated raster grids was eventually reduced to 10x10 km, which made overlay operations feasible. Failure of block groups to contain any pixel centroids was solved by assigning attributes of the nearest point features.

Every block group thus had a set of eleven climate attributes associated with it. These were then normalized to a 0-1 range and the order of block groups was randomized. This accounts for the fact that the standard SOM training algorithm takes one input vector at a time, presents it to the lattice of SOM neurons, finds the most similar neuron, and then updates weights of that neuron and its neighborhood. The size of that neighborhood is larger early during training and then begins to shrink and the magnitude of changes made to neuron weights likewise decreases over time. Input vectors presented early thus have more influence on the training process than later vectors.

SOM Training and Transformation

An ASCII text file containing normalized climate attributes for block groups, with random order of block groups, was presented as input to SOM training. The number of block groups was the principal factor in determining the number of neurons. Ideally, we would like to obtain a unique two-dimensional point location for each of the 200,000 block groups. Therefore there should be at least as many neurons as block groups. Despite the expected density effects, with denser attribute space regions being represented with more neurons, there will still be a fair number of neurons capturing multiple input vectors. This is mostly due to the fact that some neurons will have to represent empty portions of the input space, although in highly contracted form. In addition, there is the problem of edge neurons, where the expanded/contracted representation of input space tends to be less reliable and many input vectors tend to get captured. Given these concerns and computational resources, it was decided to train a SOM consisting of 250,000 neurons (500x500). This will not translate into a square SOM, but into a rectangular two-dimensional lattice, due to the use of a hexagonal neighborhood, with rows dropping into the “gaps” below (see also Figure 5).

Using *SOM_PAK*, initial weights of neurons were set randomly and training then proceeded in two stages. During the first stage, 20,000 training runs were performed, with an initial neighborhood size of 250. This stage serves to represent major, “global” structures among the input data. The second stage then aims at shaping the representation of regional and local structures. It consisted of 100,000 training runs, with a starting neighborhood size of 25 neurons. Training took 47 minutes for the first stage and 313 minutes for the second stage (wall clock time), on a single-CPU 2.3 GHz Xeon processor system.

In retrospect, given the number of neurons and the number of input vectors, a larger number of runs might have been preferable, as seen in some of the visualizations below. However, one important lesson to be conveyed in this chapter is that in order to visualize *n*-dimensional data one will often proceed in an iterative manner, especially when it comes to detecting anomalies in the data and for setting training parameters. Experience shows that visualization is in fact uniquely suited to not only generating knowledge about the mapped domain as such, but that is also a powerful tool for testing and refining visualization methods and data. For example, note below the discussion on the effects of the wind speed variable on the trained SOM.

SOM_PAK was also used to "map" input vectors onto the trained SOM. Every block group climate vector is compared to all neuron vectors to find the most similar neuron. *SOM_PAK* thus produced two output files. One is the trained SOM, also known as codebook file and consisting of a list of all neurons and their final weights for all variables. The other output contains information about the best-matching neuron found for each input vector.

While *SOM_PAK* has only rudimentary visualization capability in the form of PostScript output, its codebook format has become a standard read by a number of SOM software solutions, including the *SOM Toolbox for Matlab* and *Viscovery SOMine*, where one can then perform such operations as display of individual component planes, U-Matrix, and sometimes limited clustering. However, there tends to be virtually no user control over design specifics, like color schemes and other symbol choices and the display of features mapped onto the SOM is extremely limited and virtually useless when dealing with large numbers of features. This lack of control over the visualization is arguably the main reason for the widespread uniformity and lack of visual appeal associated with most SOM-based visualizations described in the literature, with output from the *SOM Toolbox* being particularly prevalent. On the other hand, transformation of SOM output into a form agreeable with standard GIS allows tight control over visual appearance and has the added advantage – compared to most graphic design software – of still being data driven and thus quickly adaptable to SOMs of different size and type. We decided to convert the codebook file of neuron vectors into the *ESRI Shapefile* format, with neurons represented as hexagonal polygons and neuron weights for all attributes placed in the associated dbase file. *SOM_PAK*'s initial output of the best-matching neuron for each input vector was transformed into a *Shapefile* containing a unique point location for each vector, based on random placement inside the respective best matching neuron.

Visualizing the SOM

The use of climate data in this experiment is specifically aimed at observing and understanding issues arising with a high-resolution SOM. This must precede future studies, in which such SOM can then become the basis for discovery of patterns and relationships that are simultaneously “valid, novel, potentially useful, and ultimately understandable” (Fayyad et al., 1996). The *Shapefiles* generated from *SOM_PAK* code book files can be visualized in an uncomplicated manner in standard GIS software, as shown in Figure 6. Each of the eleven component planes is simply shown through gray-scale shading of all 250,000 neurons (lighter shading indicates higher weight for an attribute). Side-by-side display allows observing relationships between variables. For example, in the upper right corner of the SOM one observes high values for humidity, precipitation, and temperature, but low amounts of snowfall, and a medium number of days with sunshine. Equally straightforward as the display of component planes is the point symbol display of all 200,000+ block groups (lower right corner of Figure 6).

That point display is an example for the suggestive power of visualization and the danger inherent in it. Notice how the apparent arrangement of densely occupied and empty portions in the spatialization suggests the existence of linear separation regions in attribute space, which separate “clusters” from each other. While one might find esthetic value and even beauty in this display of “data mangroves,” they in fact turn out to be largely artifacts caused by data preprocessing. Some in-depth explanation of this issue is in order, less for what it may teach us about SOM, but more as an example of how visualization has the power to not only deceive us, but also enable us to uncover and explain those deceptions.

Compare the block group display to each of the component planes and try to find some correspondence, which might indicate that the corresponding variable possibly had overriding influence on SOM training! It would appear that patterns in block group point locations are most closely related to patterns in the average wind speed variable. In that component plane one observes sudden changes in wind speed, with some wind speed classes forming narrow bands, even though all classes have equal width for that attribute. Those narrow bands correspond to the dominant linear separation features in the point display. None of this would be problematic, *if* the wind speed variable, as observed in nature, indeed shows such a staggered structure as opposed to the pattern mostly being an artifact of some data transformation process. Unfortunately, the latter turned out to be largely the case here. The correspondence between block group locations and wind speed prompted us to look more closely into that variable. It turned out that average wind speed in this data set has a very small absolute range of values when compared to other variables. That should not cause any problems, since all variables were scaled to a 0-1 range. However, in the process of interpolation all eleven attributes had become represented as integer raster grids in order to limit data volumes for rasters spanning all of the contiguous United States at a 1 km^2 resolution. For attributes with small absolute range the combination of integer storage and 0-1 scaling meant the introduction of wide gaps along that attribute dimension, when compared with attributes with larger absolute range. Effectively, the SOM represents these gaps existing in the source data correctly (see lower right corner of Figure 6), though they were introduced through preprocessing rather than being representative of an actual climatic phenomenon.

Juxtaposition of Visualizations in Geographic Space and Attribute Space

One argument raised earlier in this chapter was that high-resolution spatialization of geographic features based on their attributes may provide a useful alternative perspective on geographic phenomena. For instance, one would be able to juxtapose geographic and attribute space visualizations in a more equitable manner. Synchronization of symbology takes semiotic choices out of the equation and let's the two-dimensional layout speak for itself. However, such synchronization may first require further transformation of the spatialization geometry. This is where the rich set of out-of-the-box tools available in GIS software becomes especially handy. For example, when the goal is to create an alternative map of the lower 48 states, one can start with point locations for census block groups (Figure 6), create a two-dimensional Voronoi region for each block group, and then dissolve boundaries between Voronoi regions of block groups within the same state. With identical symbology attached to state polygons, the two maps can now juxtaposed (Figure 7).

Given the regular, predictable geographic patterns of climate derived from a coarse set of geographic samples, one would expect the SOM map (bottom of Figure 7) to mirror many of the structures found in the geographic map (top of Figure 7). Due to the nature of the SOM update rule – with similar training vectors being attracted to and updating nearby neuron regions – major topological relationships will tend to get replicated in the low-dimensional neuron lattice. One would thus expect that states are represented as contiguous polygons in SOM space. In many cases, this does in fact come true in the SOM map, as in the case of such states as Florida (FL), Louisiana (LA), or California (CA). As another consequence of the preservation of topology, one would expect that states sharing a boundary in geographic space will also be neighbors in the SOM space. For example, notice how the clock-wise order of neighbors of the state of Indiana (IN) is replicated in the SOM (KY-IL-MI-OH). Topology preservation is sometimes even able to bridge the earlier mentioned chasms caused by the preprocessing of data. For example, the large polygon depicting most of the state of Ohio (OH) combines block groups whose mapping had included distinct gaps (see region corresponding to Ohio in the block group visualization in Figure 6).

There are also a number of examples where the expected preservation of topological relationships does not occur, and illustration of this is also made easier by a GIS-based representation (Figure 8). Some states are represented by a number of disjoint polygons. For example, Pennsylvania (PA) becomes represented by four main polygons in the spatialization of states (left side of Figure 8). Notice however the distinct geographic organization of these four parts as representing mainly the western and eastern portion of the state (parts numbered 1 and 2). All four portions of Pennsylvania are themselves positioned next to neighboring states (parts numbered 1, 2, 3, and 4 positioned next to Ohio, New Jersey, New York, and Maryland, respectively).

There are also some decidedly odd neighbors in the SOM-based map of states. For example, notice the polygons representing Kansas and Nebraska appearing as neighbors of California (Figure 7). Zooming in on a part of that SOM region shows that there is a large gap between block groups in Northern California / Southern Oregon and those in Southeastern Nebraska / Northeastern Kansas (right side of Figure 8). Clearly, creating and using a high-resolution SOM is a difficult proposition and much remains to be learned about strategies for training such a SOM and for how to identify artifacts introduced by the computational process.

This is greatly helped by specific knowledge of the computational method used for dimensionality reduction. For example, the SOM method is known to preserve the density of input vectors. During the course of SOM training, the occurrence of multiple input vectors from a one region of attribute space will cause that region to be represented by a large number of neurons, i.e., the region will be represented in expanded form. The opposite is true also, so that thinly “populated” attribute space regions become represented by few neurons, i.e., attribute space appears compacted. In the case of census block groups this has pronounced effects. Due to the role played by total population numbers in the delineation of census block groups (aiming at a total population of around 1,500, as mentioned earlier), geographic areas exhibiting high population density will be represented by more block groups and therefore more neurons, compared with regions with lower population density. In our experiment, where state polygons are constructed from block group polygons, the SOM thus acts as a type of area cartogram! That is why high-population states like California (CA) and high-density states like Connecticut (CT) are represented as relatively large polygons, while low-density states like North Dakota (ND) or Idaho (ID) remain small (Figure 7). The reliability of this density preservation effect is however tempered by edge effects.

Edge neurons capture relatively large portions of attribute space and the size of state polygons near the edges and especially corners (like Arizona (AZ) in the lower right corner of Figure 7) should thus be treated with caution.

Mapping *n*-Dimensional Clusters onto the Climate SOM

Readers will at this point appreciate the difficulty of judging whether two-dimensional patterns and relationships visually observed in the SOM actually correspond to *n*-dimensional structures. For instance, our climate experiment led to multiple examples of the SOM generating shared borders between states that are not neighbors in traditional geographic space. It would be nice to more directly operate on *n*-dimensional data, while being able to project such operations onto the geographic map and spatialization. Among such operations are the various clustering methods commonly applied in multivariate analysis.

In order to demonstrate this this, we computed *k*-means clustering on all 200,000+ block groups, based on the very same input vectors used in SOM training. Given the typically compact nature of *k*-means solutions and the SOM's tendency towards preservation of neighborhood relationships, one would expect that clusters computed with the *k*-means method will tend to form contiguous shapes after being projected onto the SOM. To test this, block groups in both the geographic map and the SOM-based map were joined into larger polygons based on shared cluster membership and overlaid on top of states (Figure 9). This direct comparison of the mapping of identical cluster solutions onto two different "base maps" provides support for the need to perform further training on the high-resolution SOM. Clusters appear almost completely contiguous when mapped onto the geographic map (top of Figure 9), while some clusters appear broken into several parts in the SOM-based map (bottom of Figure 9). That is surprising, since the SOM and the *k*-means clusters (*k*=25) were computed from exactly the same source vectors.

One would further expect that climate clusters and state boundaries should show virtually no correlation (except for such cases where a state boundary coincides with a physical feature effecting climate, like the ridge line of a major mountain range). Indeed, that is the case in the geographic map. It is true in many parts of the SOM as well. For example, notice how the progression of climate clusters as parallel bands in the Southeastern U.S. (sequence of clusters 2-6-18-14-1-16) gets represented in both geographic and SOM-based visualization, virtually independent of state boundaries. However, there are cases in the SOM-based map where cluster and state boundaries coincide and these correspond to the same oddly neighboring states mentioned earlier. For example, the "northern" and "western" boundaries of California in the SOM completely coincide with cluster boundaries. The large gap among two-dimensional block group locations along the same boundaries therefore seems to indeed be justified (see also bottom-right portion of Figure 8). The earlier mentioned break-up of Pennsylvania (left side of Figure 8) is supported by the association of the large western portion of Pennsylvania (#1 in Figure 8) with Ohio in cluster 15 and of the eastern portion (#2 in Figure 8) with New Jersey in cluster 17.

Mapping Extreme Weather Events onto the Climate SOM

All visual transformations illustrated so far were based on the very same input vectors used to train the SOM. However, it is also possible to map other data onto the SOM that were not part of the training process, based on two different approaches. One consists of finding best-matching neurons for non-training vectors, as long as those vectors consist of the same dimensions as the training vectors and identical preprocessing has been applied (e.g., scaling to 0-1 range based on the same minimum/maximum values). In our case, one could map climate data for geographic areas outside of the U.S. onto the SOM to identify global similarities. For example, areas along the Mediterranean coast would likely end up inside of cluster 24 “in” Southern California (see Figure 9).

Another option is to use a currently mapped geographic feature as a socket through which another geography feature can be mapped onto the SOM, based on shared geographic location. Skupin and Hagelman (2005) demonstrated this by first training a SOM with multi-temporal census data and then using single-time vectors as temporal vertices that define the trajectory of a geography feature. Another proposed trajectory mapping technique takes space-time paths, such as those captured by GPS, and projects them onto a spatialization based on the sequence of geographic features traversed (Skupin, in press). We demonstrate this here for hurricanes that made landfall in the continental U.S. during the 2005 hurricane seasons (Figure 10). This involves determining the sequence of block groups traversed by each hurricane. While conceptually a straightforward overlay operation of hurricanes (represented as uni-directional routes) with block groups, it turned out to be extremely challenging in commercial GIS software, due to the very large number of block group polygons (200,000+). Mapping of point features onto the SOM is much easier, as that the block group sockets can be accessed via a simple point-in-polygon overlay. In this manner we mapped another data set of extreme weather events onto the SOM, locations at which tornadoes touched down in 2005 (Figure 10).

While this mapping of hurricanes and tornadoes illustrates the technical principles driving this type of *n*-dimensional overlay, it does not generate any particular insight, due to the regular spatial pattern of climate in the continental U.S. The technique becomes much more interesting when complex *n*-dimensional patterns are encountered across geographic space. Demographic data, such as those captured by the U.S. Census Bureau, contain many examples of such complex patterns. Skupin (in press) describes a SOM of equal resolution as the climate SOM described here, but generated from demographic attribute for 200,000+ block groups. Actual geographic movement captured by GPS is then mapped onto the SOM. For example, the author's commute from his previous residence in the Mid-City neighborhood of New Orleans to the University of New Orleans located near Lake Pontchartrain is mapped onto the SOM generated from nationwide data (Figure 11). Among the interesting patterns observed in this visualization are the relative compactness of neighborhoods in attribute space (e.g., *Mid-City*, *French Quarter*, *New Marigny*), and that movement between neighborhoods is accomplished either via large hyperjumps (e.g., movement between block groups #12 and #13) or via bridge block groups that are transitional in terms of both geographic and attribute space location (e.g., block group #23).

SUMMARY AND OUTLOOK

This chapter proposes the creation of high-resolution spatializations from the attributes of geographic features. Its main aim is to generate an alternative to the standard approach in geovisualization, where the geographic map tends to be the only stable element, while other visualizations are characterized by fluid geometry, topology, and visual appearance. It is argued that the self-organizing map could be one method able to generate more stable base maps onto which various types of other data could be mapped through a series of geometric and attribute-based transformations. This approach is decidedly different from the current use of such tools as scatter plots, parallel coordinate plots, and even of the SOM method itself.

We demonstrated this approach by first training a SOM consisting of 250,000 neurons with climate data generated for more than 200,000 geographic features, performing various transformations, and finally juxtaposing SOM-base maps with geographic maps. Numerous challenges were encountered, beginning with the difficulty of performing geographic overlays involving several hundred thousand objects in standard GIS software. Training using *SOM_PAK* was unproblematic, but a number of artifacts in the visualization – in particular the break-up of *k*-means clusters after projection onto the SOM – suggest that more training cycles should be applied, beyond the 120,000 cycles used in our experiment (20,000 and 100,000 in the two training stages, respectively).

Future work will include a more formal investigation of the distortions incurred by the training of a SOM consisting of several hundred thousand neurons with an equally large number of training vectors. On one hand, there is a need to develop recommendations for how to use standard SOM tools (e.g., *SOM_PAK*) in the context of high-resolution SOM. This must occur in recognition of the fact that geometric distortion as such is unavoidable when high-dimensional data are represented by low-dimensional geometry and that the SOM method is in fact able to bridge large dimensional gaps between source data and display space through density-driven effects of expansion and compression. On the other hand, overall distortion characteristics can be addressed by recent variations of the SOM method. For example, edge effect distortions are caused by topological heterogeneity in the neuron structure, with edge neurons having fewer neighbors than neurons further inside of the SOM. This source of distortions could be diminished by arranging neurons on a closed surface. A number of spherical SOM approaches have been proposed (Sangole and Knopf, 2002, Wu and Takatsuka, 2005), but have not yet been used much in practical SOM applications and have not involved large numbers of neurons. Another interesting question in the context of high-resolution SOMs is how the total number of neurons is to be determined. The standard SOM algorithm used in our climate experiment as well as spherical and other variants take the number of neurons as an input parameter, which therefore allows user control of SOM granularity. A very different, alternative approach involves the deletion or addition of neurons in response to certain threshold functions. This leads to so-called *growing SOMs* (Fritzke, 1999). Apart from these approaches addressing problems observed in both low- and high-resolution SOMs, there may be issues arising principally with SOMs of high and extremely high resolution that are as of yet undefined. Systematic studies should be undertaken to explore this, which may in particular call for well-controlled synthetic data sets.

The use of climate data in a high-resolution SOM and the use of existing geographic objects as place holders of climate attributes was informed by the desire to extent the notion of *attribute space travel* (Skupin, in press). One of the overarching goals of that research direction is to provide a methodological framework for dealing with the experience of geographic space in a computational manner. Socio-economic characteristics of a geographic place have an effect on one’s experience with and interaction in that place, but the physical attributes, such as temperature and humidity, obviously play an important role as well. The experimental work presented in this chapter is meant as a first step towards an integrated visual modeling of physical and social attributes, in this case by attaching climate attributes to demographic enumeration units. Future work will include the actual combination of very different types of attributes, including climate, demographic, land use / land cover, and many others, and thus create rich spatializations of geographic objects. Such work may include irregular enumeration units – therefore incurring the cartogram effects described in this chapter – as well as regular tessellations of geographic space.

ACKNOWLEDGEMENTS

We gratefully acknowledge the assistance of Charles Schmidt in the pre-processing of climate data and of Martin Lacayo-Emery in creating geometric and attribute base data for the climate SOM.

6. REFERENCES

DEBOECK, G. & KOHONEN, T. (Eds.) (1998) *Visual Explorations in Finance with Self-Organizing Maps,* Berlin; Heidelberg; New York, Springer-Verlag.

FAYYAD, U. M., PIATETSKY-SHAPIRO, G., SMYTH, P. & UTHURUSAMY, R. (Eds.) (1996) *Advances in knowledge discovery and data mining,* Menlo Park, CA, AAAI/MIT Press.

FRITZKE, B. (1999) Growing self-organizing networks - history, status quo, and perspectives. IN OJA, E. & KASKI, S. (Eds.) *Kohonen Maps.* Amsterdam, Elsevier.

KOHONEN, T. (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics,* 43**,** 59-69.

KOHONEN, T. (1990) The self-organizing map. *Proceedings of the IEEE,* 78**,** 1464-1480.

KOHONEN, T. (2001) *Self-Organizing Maps,* Berlin, Springer-Verlag.

KOHONEN, T., HYNNINEN, J., KANGAS, J. & LAAKSONEN, J. (1996a) *SOM_PAK: The Self-Organizing Map Program Package,* Espoo, Finland, Helsinki University of Technology, Laboratory of Computer and Information Science.

KOHONEN, T., KASKI, S., LAGUS, K. & HONKELA, T. (1996b) Very Large Two-Level SOM for the Browsing of Newsgroups. *1996 International Conference on Artificial Neural Networks.* Springer, Berlin.

OJA, E. & KASKI, S. (Eds.) (1999) *Kohonen Maps,* Amsterdam, Elsevier.

SANGOLE, A. & KNOPF, G. K. (2002) Representing high-dimensional data sets as closed surfaces. *Information Visualization,* 1**,** 111 - 119.

SHIFFRIN, R. M. & BÖRNER, K. (2004) Mapping Knowledge Domains. *Proceedings of the National Academy of Sciences,* 101**,** 5183-5185.

SKUPIN, A. (2002) A cartographic approach to visualizing conference abstracts. *IEEE Computer Graphics and Applications,* 22**,** 50-58.

SKUPIN, A. (2003) A novel map projection using an artificial neural network. *21st International Cartographic Conference.* Durban, South Africa.

SKUPIN, A. (2004) The World of Geography: Visualizing a knowledge domain with cartographic means. *Proceedings of the National Academy of Sciences*, 101**,** 5274-5278.

SKUPIN, A. (in press) Where do you want to go today [in attribute space]? IN MILLER, H. J. (Ed.) *Societies and Cities in the Age of Instant Access.* Springer.

SKUPIN, A. & HAGELMAN, R. (2005) Visualizing Demographic Trajectories with Self-Organizing Maps. *GeoInformatica*, 9**,** 159-179.

TOBLER, W. R. (1970) A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46**,** 234-240.

TOBLER, W. R. (1979) A Transformational View of Cartography. *The American Cartographer*, 6**,** 101-106.

ULTSCH, A. (1993) Self-organizing neural networks for visualization and classification. IN OPITZ, O., LAUSEN, B. & KLAR, R. (Eds.) *Information and Classification: Concepts, Methods, and Applications.* Springer-Verlag.

WU, Y. & TAKATSUKA, M. (2005) Geodesic Self-Organizing Map. IN ERBACHER, R. F., ROBERTS, J. C., GROHN, M. T. & BÖRNER, K. (Eds.) *Conference on Visualization and Data Analysis 2005 / Proceedings of SPIE Vol. #5669.* San Jose, CA, January 17-18, 2005.

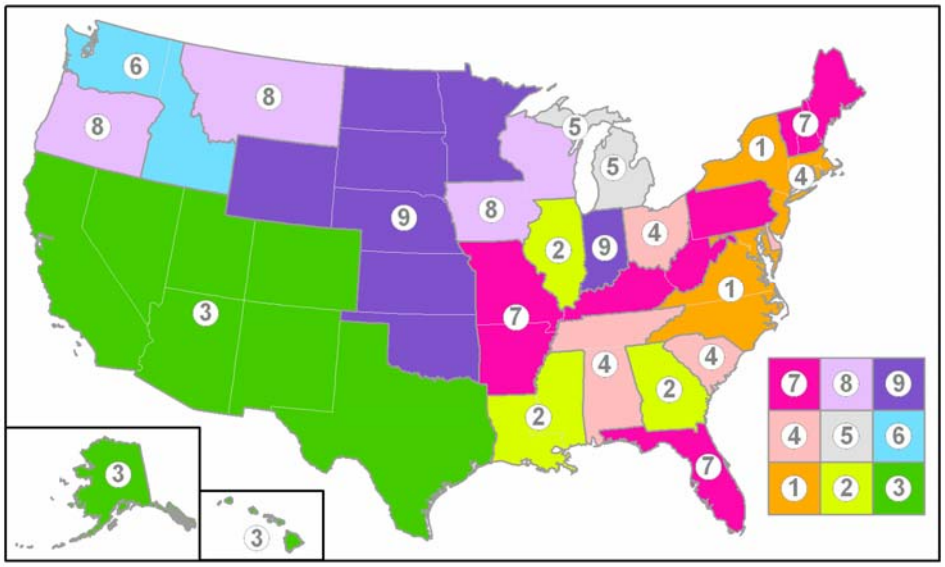

Figure 1. Use of a low-resolution SOM for clustering and geographic visualization. A three-by-three neuron SOM is trained with demographic data for U.S. States.

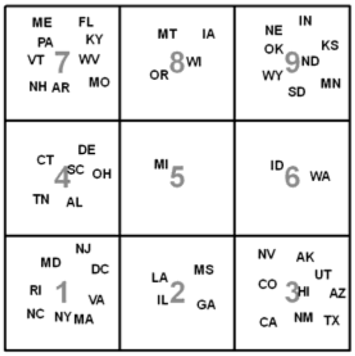
ME FL PA KY VT 7 WV NH AR MO
MT IA 8 WI OR
NE IN OK KS 9 ND WY MN SD
DE CT 4 SC OH TN AL
MI 5
ID 6 WA
NJ MD DC RI 1 VA NC NY MA
LA MS IL 2 GA
NV AK UT CO 3 HI AZ CA NM TX

Figure 2. Mapping of 51 features onto low-resolution SOM (nine neurons), including disambiguated geometry through random placement inside winning neuron polygons.

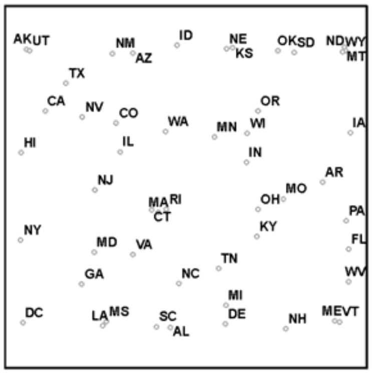

Figure 3. Mapping of 51 features onto higher-resolution SOM (400 neurons), with much reduced need for disambiguation of geometry.

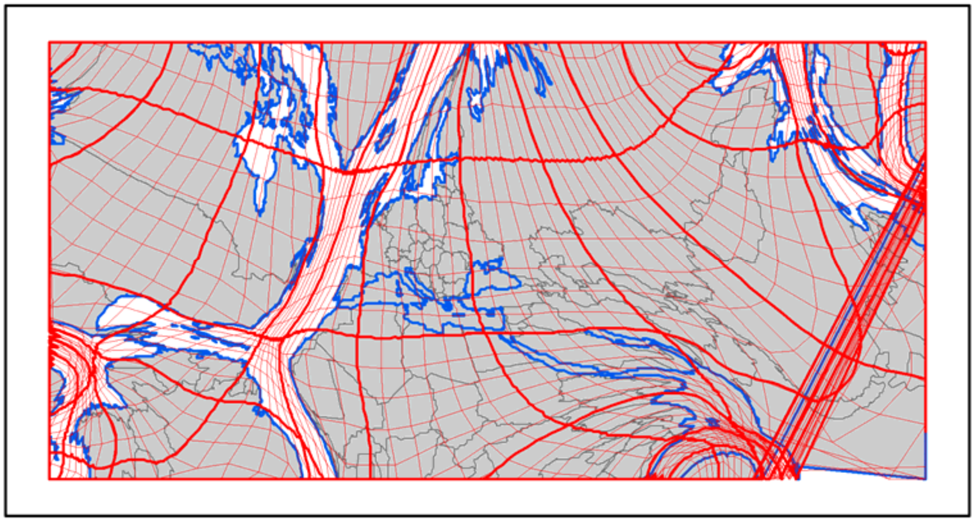

Figure 4. Multiple layers projected onto a high-resolution SOM (125,000 neurons) trained with geographic coordinates (from Skupin 2003).

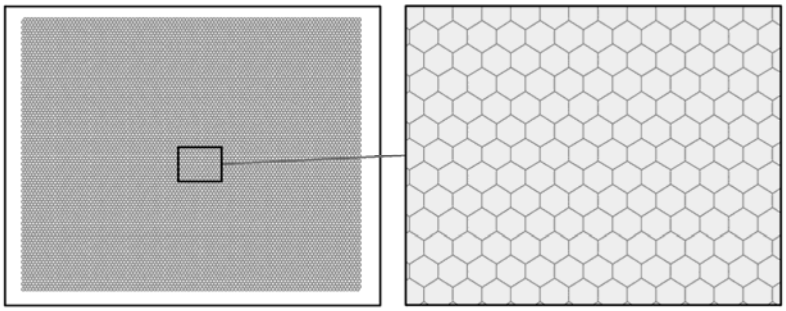

Figure 5. Neuron geometry in a high-resolution SOM with hexagonal neighborhood (10,000 neurons).

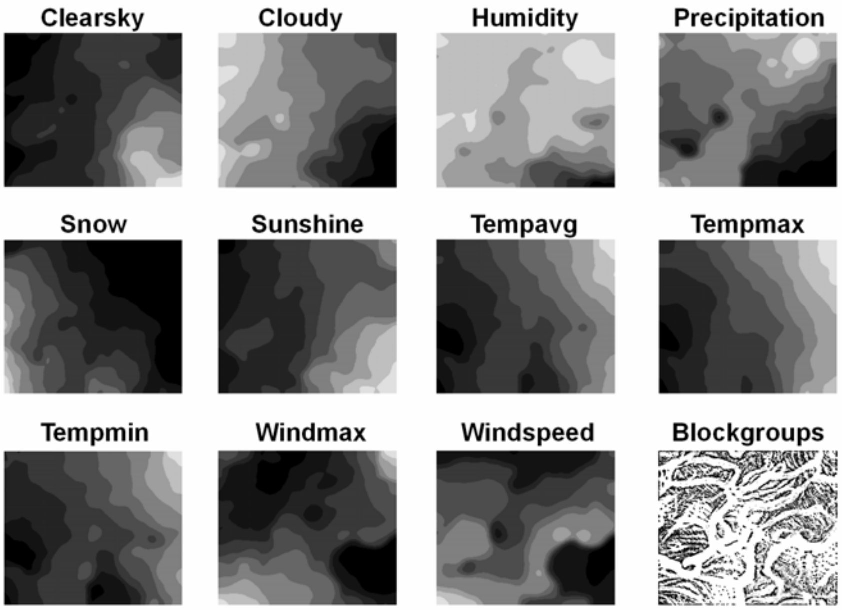

Figure 6. Component planes of a high-resolution SOM of 250,000 neurons constructed from climate data for 200,000+ census block groups. Block groups mapped onto SOM (bottom right).

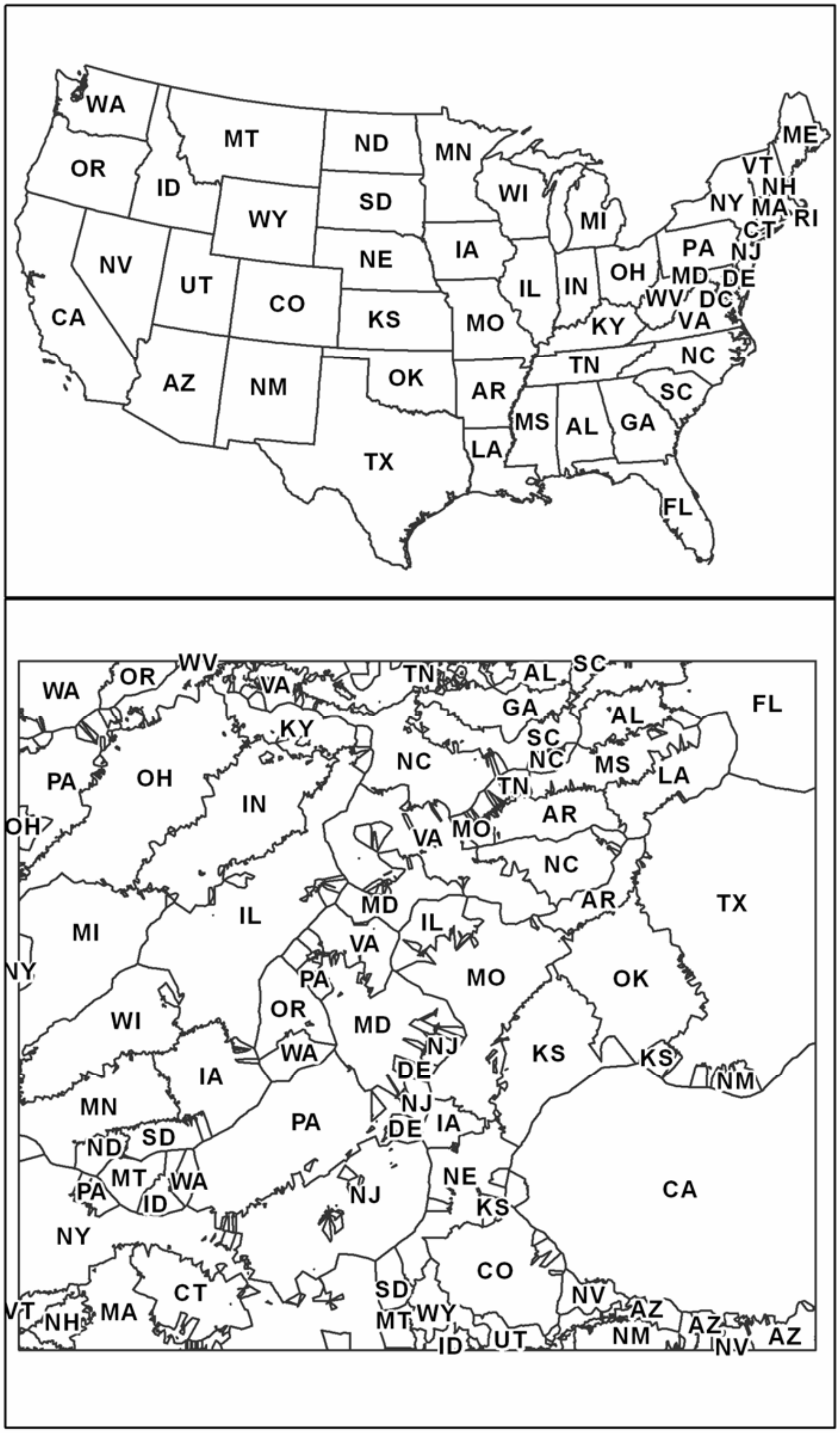

Figure 7. U.S. states in geographic map (top) and spatialized based on a high-resolution SOM of 250,000 neurons trained with climate data for 200,000+ census block groups (bottom).

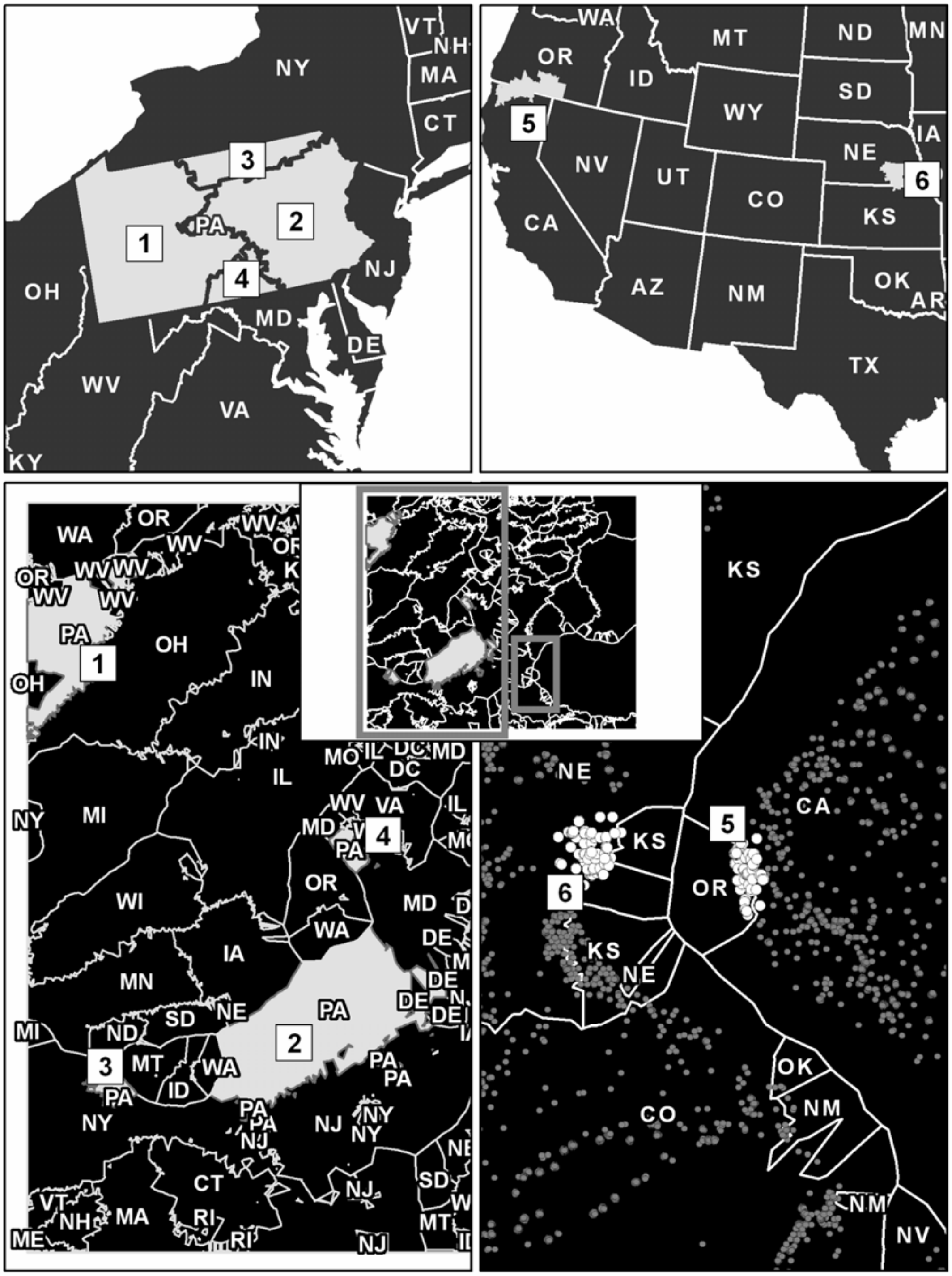

Figure 8. Investigating regions of interest in geographic map and spatialization. Break-up of Pennsylvania into several regions (left) and different geographic regions appearing in relative proximity in the spatialization (right).

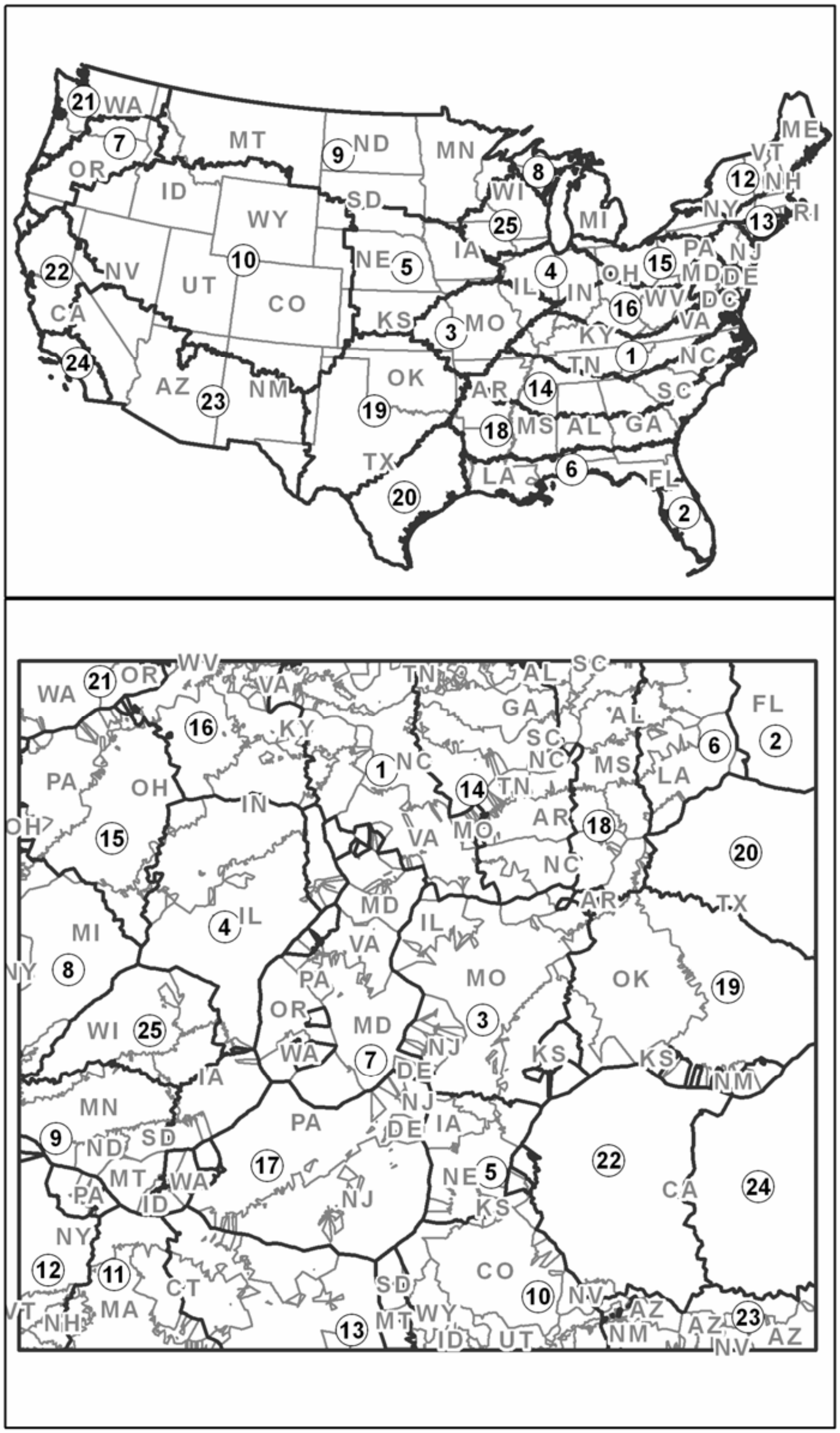

Figure 9. k-means clustering of climate attributes for 200,000+ census block groups visualized in geographic map and spatialization (k=25).

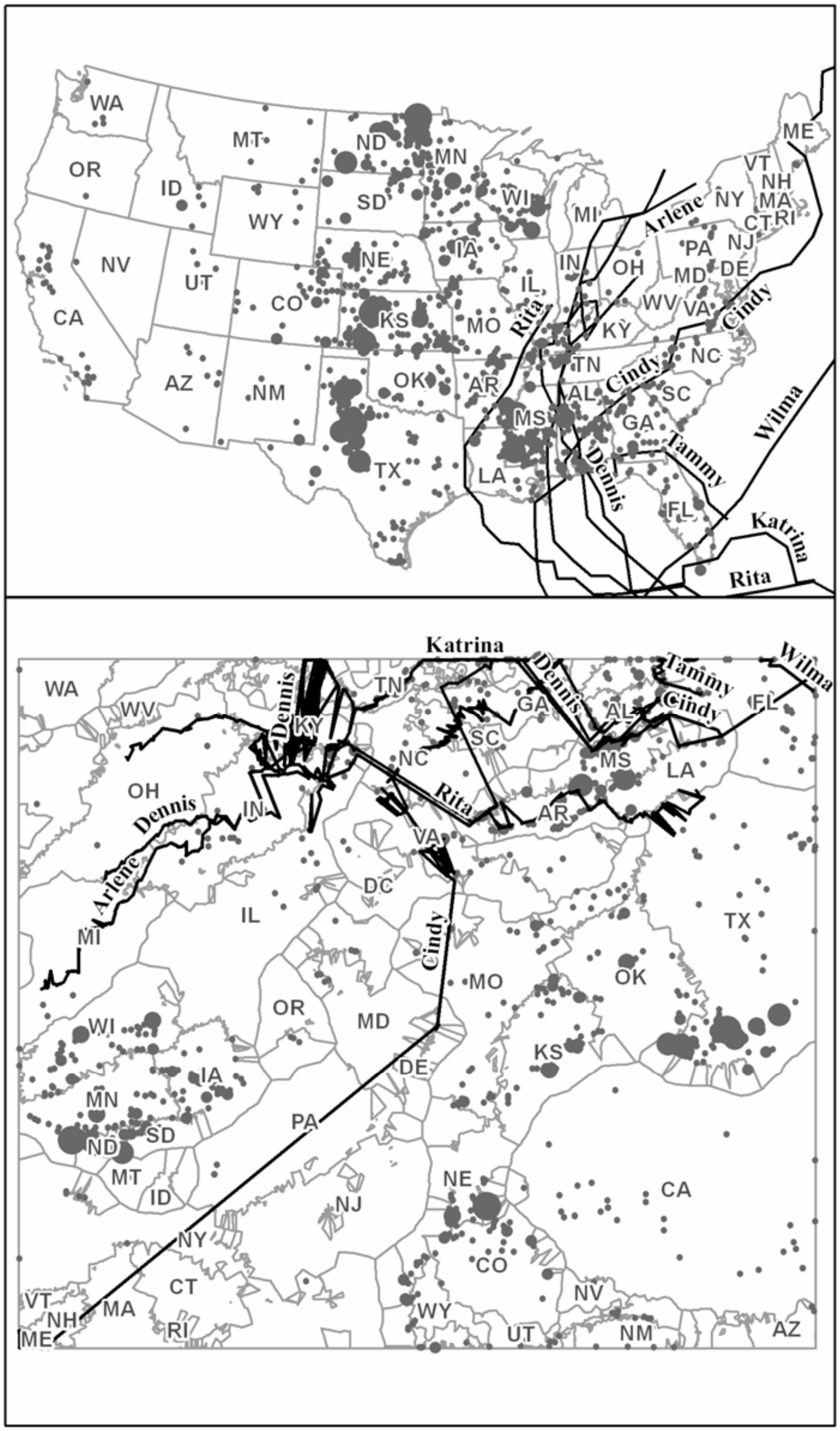

Figure 10. Tornado touchdowns and hurricane paths observed during 2005 mapped onto geographic map and climate-driven spatialization.

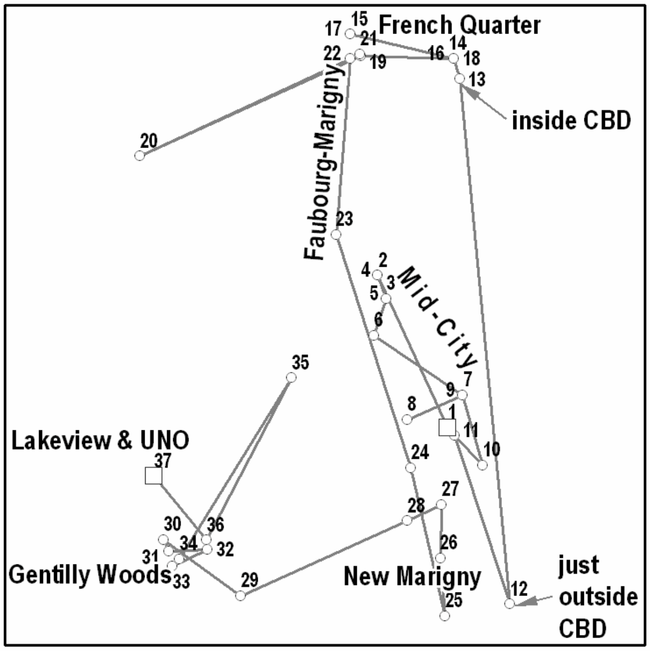

Figure 11. Traveling in attribute space in New Orleans. Spatialization derived from demographic attributes for all 200,000+ U.S. census block groups (modified after Skupin in press).