

**VIS-STAMP:
A Visualization System for Space-Time and Multivariate Patterns**

**User Manual
Version 1.0**

Author:
Diansheng Guo

Department of Geography
University of South Carolina

Website:
www.SpatialDataMining.org

June 17, 2009

Launching VIS-STAMP:

To run this program, it is recommended to have the latest version of Java installed on your machine. You can visit <http://java.com/en/download/index.jsp> and test if you have the latest version. .

Then you simply start VIS-STAMP using one of the following options:

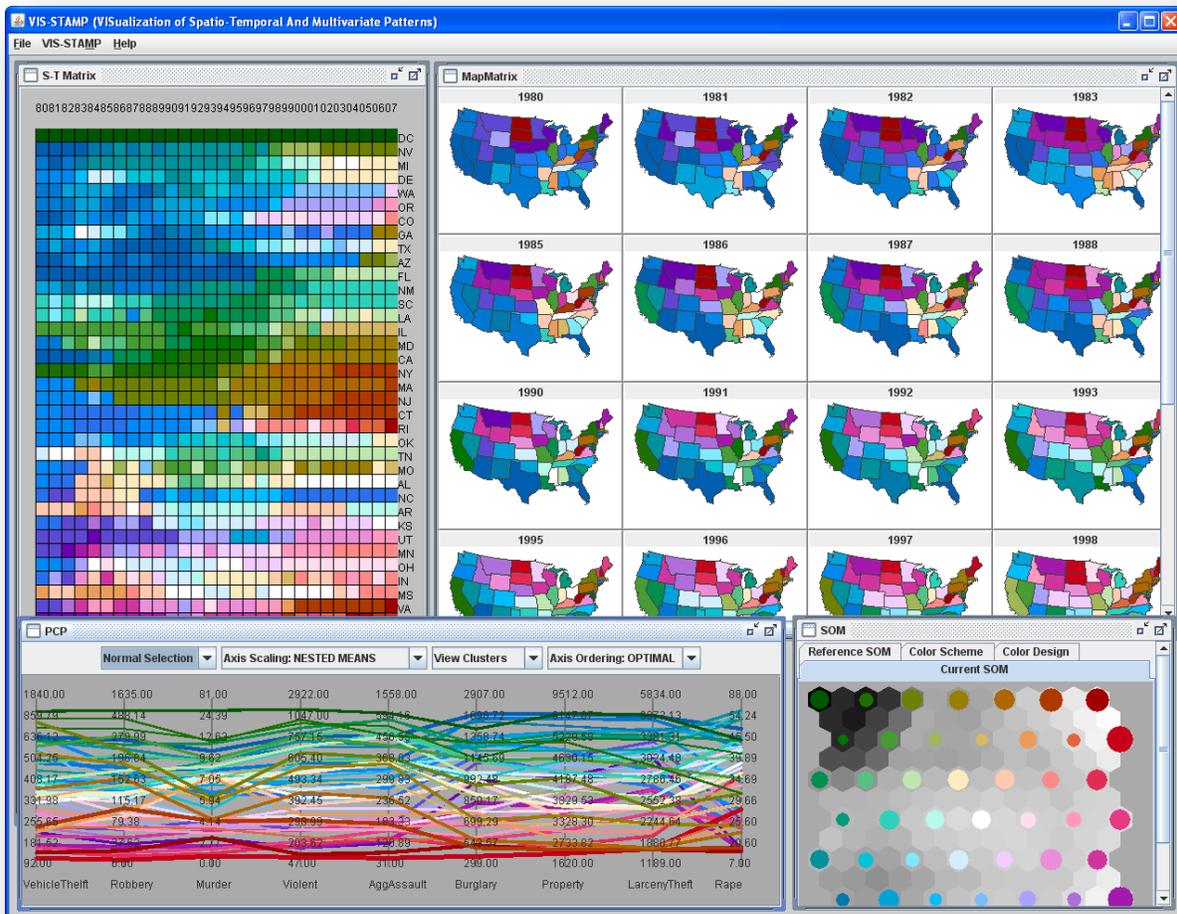
- Double click the file **visstamp.jar**; **OR**
- Right click on **visstamp.jar** → open with... → Java ...; **OR**
- Execute in a command window with "**java -jar visstamp.jar**". This approach allows you to review output messages when unexpected error occurs.

(If for some reason, your downloading software actually saves the file as "**visstamp.ZIP**", you need to rename the file back to "**visstamp.JAR**" before running it as instructed above.)

INTRODUCTION

VIS-STAMP is a software package to explore multivariate spatio-temporal data, discover interesting and unknown complex patterns, and present them in an easy-to-understand form to support human interpretation, analytical reasoning, and/or decision-making. The software tool integrates computational, visual, and cartographic methods together to detect and visualize **multivariate spatio-temporal** patterns. It is able to:

- perform multivariate clustering and abstraction (including time-series clustering) with a Self Organizing Map (SOM);
- encode SOM result with colors derived from a two-dimensional diverging-diverging color scheme;
- visualize the multivariate patterns with an enhanced Parallel Coordinate Plot (PCP) display, which serves as a multivariate “legend” in the integrated system;
- visualize the spatio-temporal variations of multivariate patterns, or the space-variable variations of temporal patterns in a hierarchical, computationally sortable matrix and a temporally or geographically ordered map matrix;
- support human interactions to explore patterns from different perspectives and at different detail levels.



Related Publication:

Guo, D., J. Chen, A. M. MacEachren, and K. Liao (2006), "A Visualization System for Spatio- Temporal and Multivariate Patterns (VIS-STAMP)", IEEE Transactions on Visualization and Computer Graphics, 12(6), pp. 1461-1474.

Guo, D. and M. Gahegan (2006). "Spatial Ordering and Encoding for Geographic Data Mining and Visualization", Journal of Intelligent Information Systems, 27(3), pp.243-266.

DATA FORMAT

There are three data files involved:

(1) a shape file (*.shp), which contains the boundaries of spatial objects.

(2) a csv file as the attribute table for the shape file, which contains all attributes for each spatial object.

The order of records in this file must match the order of shapes in the shape file. You can simply save the dbf file of a shape layer to a csv file. The only column that is useful in this csv file is the name field, which is the first column that has "name" in its name (e.g., "STATE_NAME").

For example, a record in this csv file may read as follows, where the first column is the name for the shape. The names of shapes will be used to match the spatial objects in your spatio-temporal data file.

CA, California, 39030, 32000, ...

The shape file and its csv file will be opened together. They must have the same name but with different extensions (e.g., States.shp and States.csv)

(3) The spatio-temporal and multivariate data (a CSV file---see the "sample_stv_data.csv" file).

For example, a record in your spatio-temporal data file may be:

CA, 1998, 1.2, 3.8, 10

The first column must be the spatial object name. The second column must be the time field (e.g., year). The rest of columns are variables. The order of records can be different than the order of shapes in the shape file. The program will automatically match names.

Missing data is assumed to be ZERO! So, better not have missing values.

It is better to have all values positive (if not, please transform the data, say, by adding a large enough number to all numbers).

LOADING DATA

File → Load Data ...

Choose either the .csv file or the shape file (the other one will be automatically loaded). Then, choose the spatio-temporal csv file.

Only the numerical fields (columns) will be used by VIS-STAMP. Other columns will be ignored.

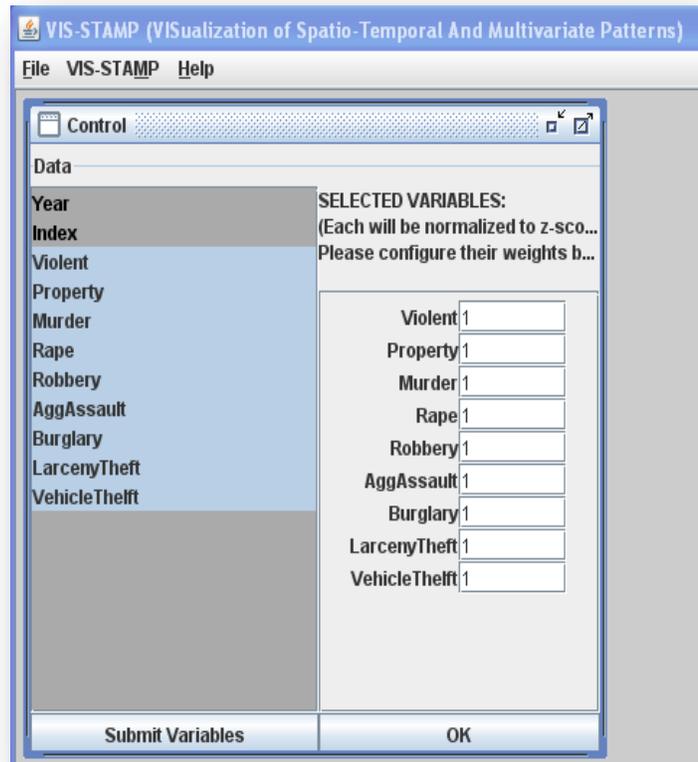
Select multiple variables from the list.

Click “Submit Variables”.

The values of each variable will then be normalized to Z-scores (i.e., zero mean and unit variance).

You may give each variable a different weight (which will be multiplied to the z-score).

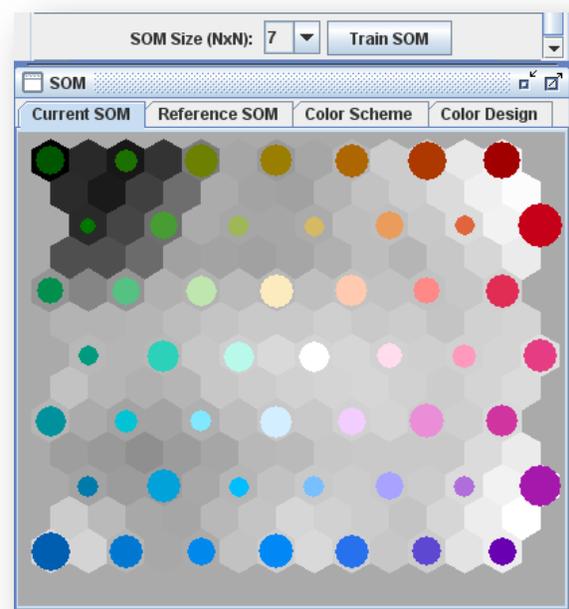
Click “OK”.



SELF-ORGANIZING MAP CLUSTERING

The normalized (and optionally weighted) multivariate data is processed by a Self-Organizing Map (SOM) to derive clusters of spatial objects based on their multivariate similarity. The SOM uses the Euclidean distance to assess multivariate similarity between spatial objects.

The default size for SOM is 7x7 (i.e., 49 clusters). Depending on the data set size, one may choose a smaller or larger size. Then click “Train SOM”. The SOM clustering result is visualized as depicted in the snapshot



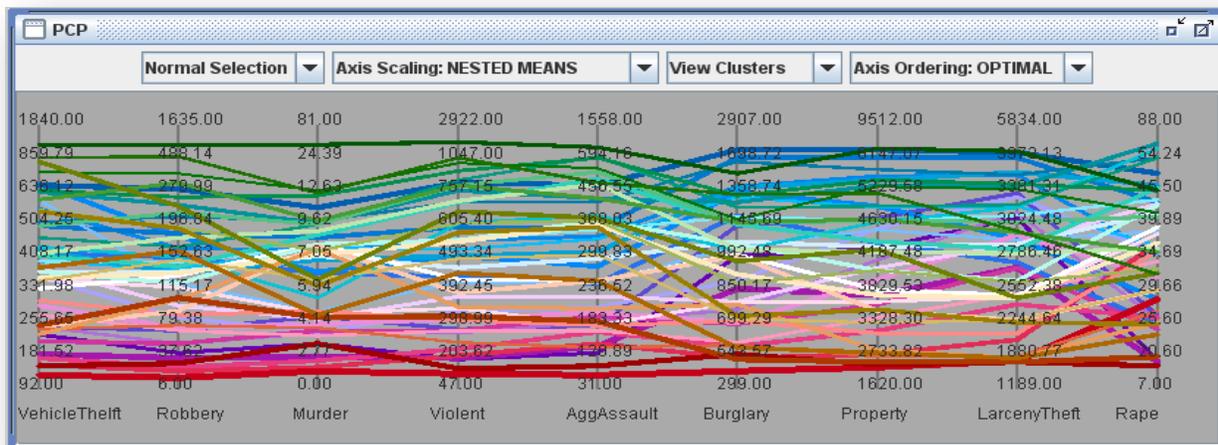
to the right, which uses two different types of hexagons: (1) *node hexagons*, each of which contains a circle that is scaled to depict the number of data items in the node (cluster); and (2) *distance hexagons*, each of which is shaded to represent the multivariate dissimilarity between two neighboring nodes (i.e., two codebook vectors). This kind of graphic display is called the U-matrix. For methodological details about SOM, readers are referred to book:

Kohonen, T. (2001). Self-organizing maps, Berlin ; New York : Springer.

A data item is assigned to a node (cluster) if that node’s codebook vector is the closest to the data item. A node can have more than one data item assigned or may have no data item assigned (in which case it is an empty node). The area of the circles inside node hexagons represents the number of items contained in each node. Each circle is filled with a color from a 2D color scheme so that similar clusters (i.e., nearby nodes) have similar colors. See Guo et al. 2005 for detailed explanation.

The user can rotate or flip the 2D color scheme in case a certain corner is desired to have a certain color. These functions are enabled at the “Color Scheme” tab. The user may also change the 2D color design by going to the “Color Design” tab. See Guo et al. 2005 for detailed explanation on these functionalities.

MULTIVARIATE VISUALIZATION WITH PARALLEL COORDINATE PLOT (PCP)



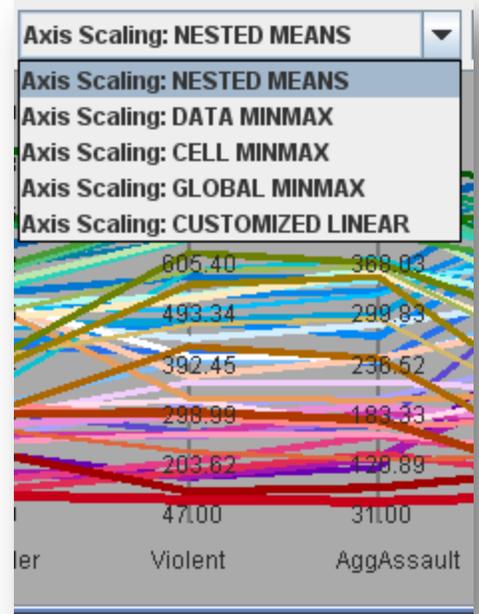
The PCP supports two detail levels:

- **View Clusters:** Each string represents a cluster with its mean vector. The thickness of each string represents the cluster size (i.e., the number of data items in the cluster).
- **View Data Items:** Each string represents a data item with its multivariate vector.

- For either choice, each cluster (or data item) has the same color as it does in the SOM view.

The PCP supports five axis scaling methods:

- **Nested-Means:** scaling on each axis using nested means and thus adjust the spacing of intervals according to data distribution. This method can alleviate the overlapping problem in PCP for skewed data distribution. Specifically, nested-means is a *non-linear scaling* method that recursively calculates a number of mean values (and sub-means) and uses these values as break points to divide each axis into equal-length segments. Therefore, nested-means scaling always puts the mean value at the center of each axis and thus makes axes defined by different units and data ranges comparable.
- **Data Min-Max:** each axis is linearly scaled using its min and max values.
- **Cluster Min-Max** (or Cell Min-Max): each axis is linearly scaled using cluster centroid min and max values.
- **Global Min-Max:** this option is only useful when all the variables are comparable to each other, for example percentage values. Axes will be scaled linearly using the global min and max values (for all variables).
- **Customized Linear:** this option is only useful when all the variables are comparable to each other, for example percentage values. The user will define the min and max (same for all variables) to linearly scale each axis. In future versions, the user may be able to define the min/max differently for each axis.



The PCP can *optimally* order dimensions

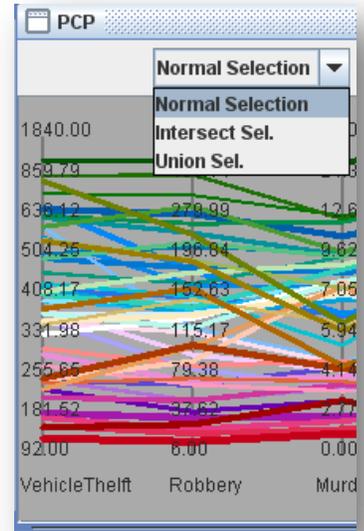
- **Optimal Ordering of Axes:** dimensions are ordered using an optimal hierarchical ordering method based on the mutual correlations among dimensions

Guo, D. and M. Gahegan (2006). "Spatial ordering and encoding for geographic data mining and visualization." *Journal of Intelligent Information Systems* 27(3): 243-266.

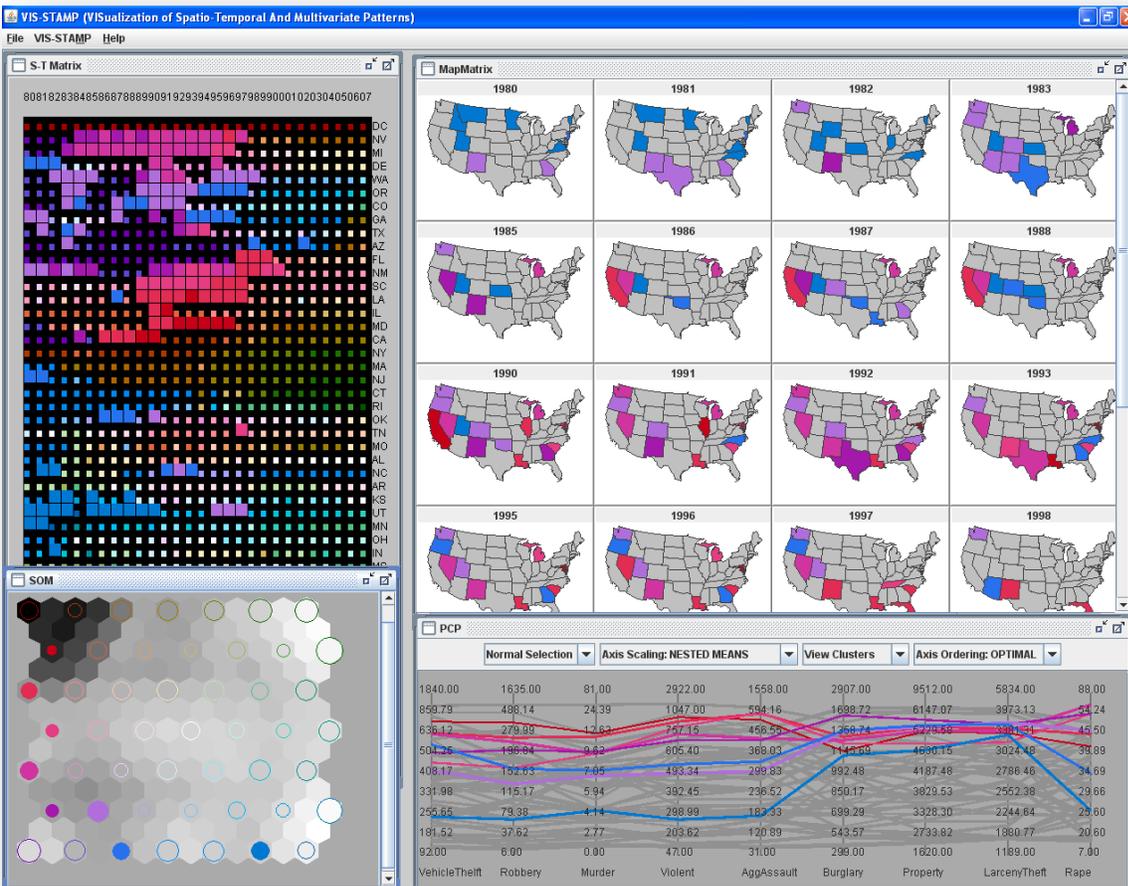
- **Original Ordering of Axes:** dimensions are in their original order as in the data file.

The PCP supports different types of selection at different levels

- The PCP at the cluster level presents a global view of the overall patterns. A user can select one or more clusters in the PCP (or in the SOM), then switch to the data item level (instead of the cluster level), and examine all the data items in the cluster(s).
- Selection can also be made at the data item level. For example, one can show data at the item level in the PCP and then select a single data item to read its exact variable values. One can also switch back to the cluster level and see which cluster the selected item belongs to. If that cluster also contains other items, its circle will become a wedge to show the partial selection.
- By selection “Intersect Sel.” or “Union Sel.”, the user may also combine two different selections or select within a selection.

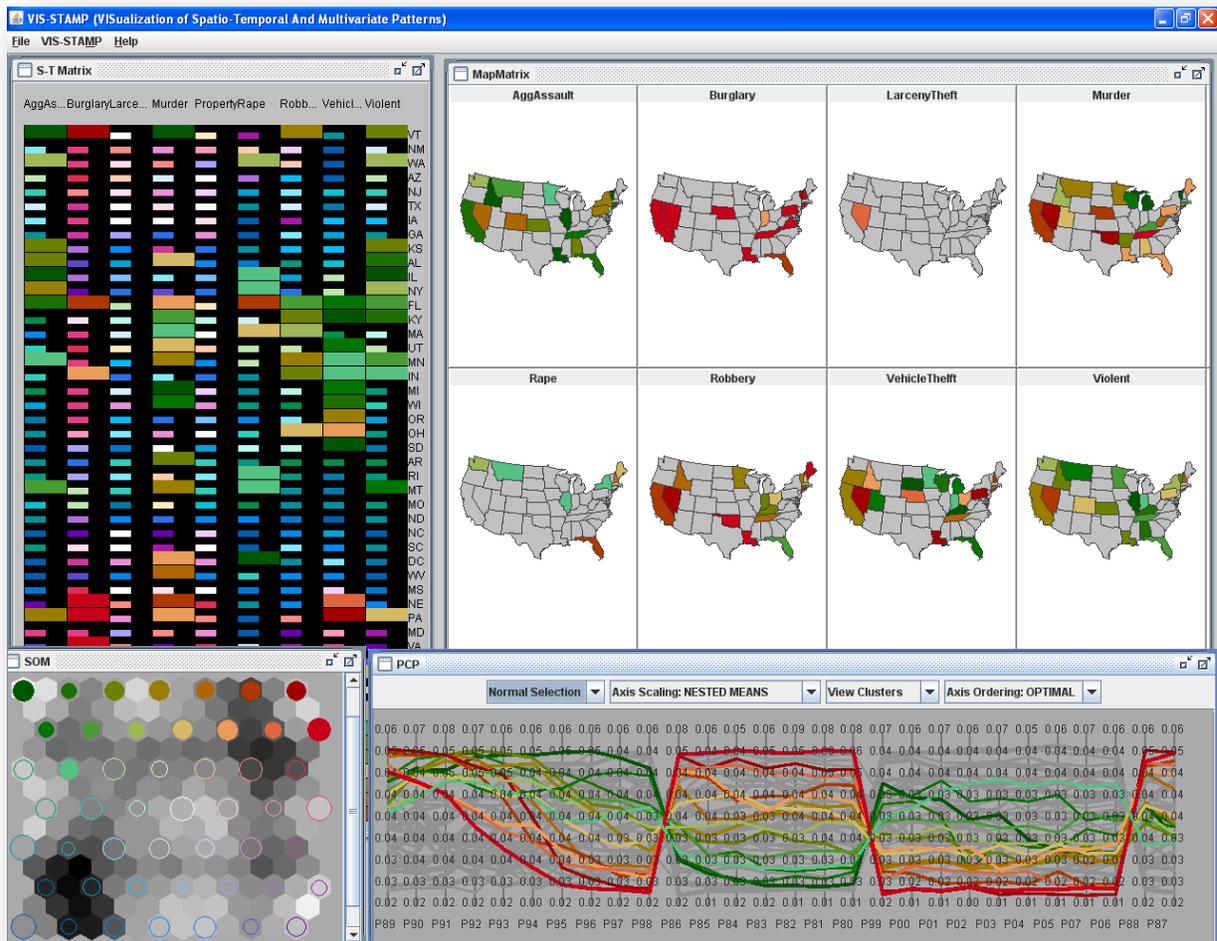


Spatio-Temporal Visualization of Multivariate Patterns



A form of reorderable matrix (top-left in the figure above) is developed to organize multivariate patterns (represented with colors) across space and time. This reorderable matrix, when one of the two dimensions represents geography, will be accompanied with a reorderable map matrix (top-right in the figure above). In the application shown in the figure above, columns represent time (years) and rows represent places (states of the US). Ordering of time is fixed (for these applications) in normal temporal order. Ordering of places is computationally derived with several cluster-based ordering methods. After the reordering, states that have similar crime patterns over time are next to each other in the matrix and form homogeneous spatio-temporal “regions”. The reorderable map matrix essentially converts each column in the reorderable matrix to a map and these maps are arranged in the same order as that of the columns.

Space-variable Visualization of Temporal Patterns

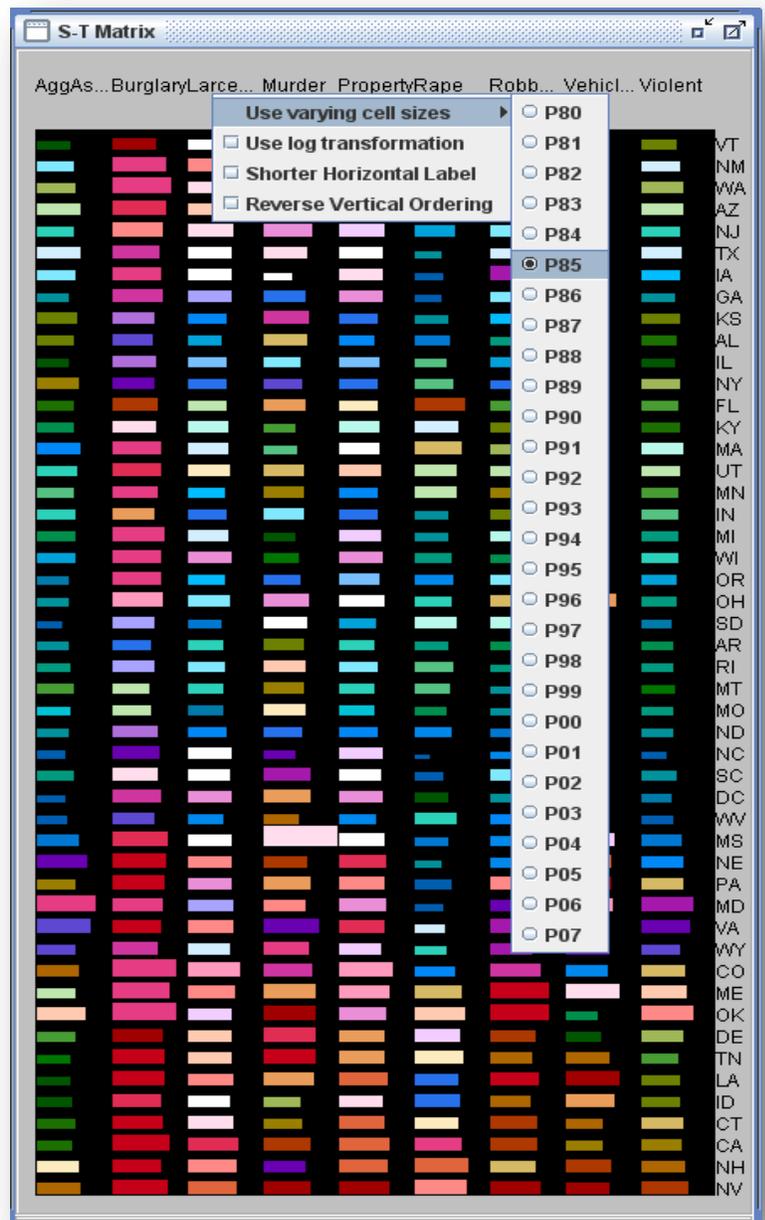


Our system has the flexibility to present patterns in the space-time-multivariate data cube form different perspectives. For example, we can use crime types and states (instead of years and states) to organize the reorderable matrix (called space-variable matrix). In other words, each column represents a crime type and each row represents a state. Temporal series will be treated as multivariate vectors and clustered (and colored) by the SOM. To characterize a temporal trend, we convert each time series (one for each state/crime combination) to percentages. For example, in the sample data “1980_2007_pct_crime_rates.csv”, the column “P80” is the percentage of 1980’s crime rate against the total crime rates for all years for that specific state and crime. Now, the variation of temporal patterns can be examined across geography and multiple categories (e.g. crime types).

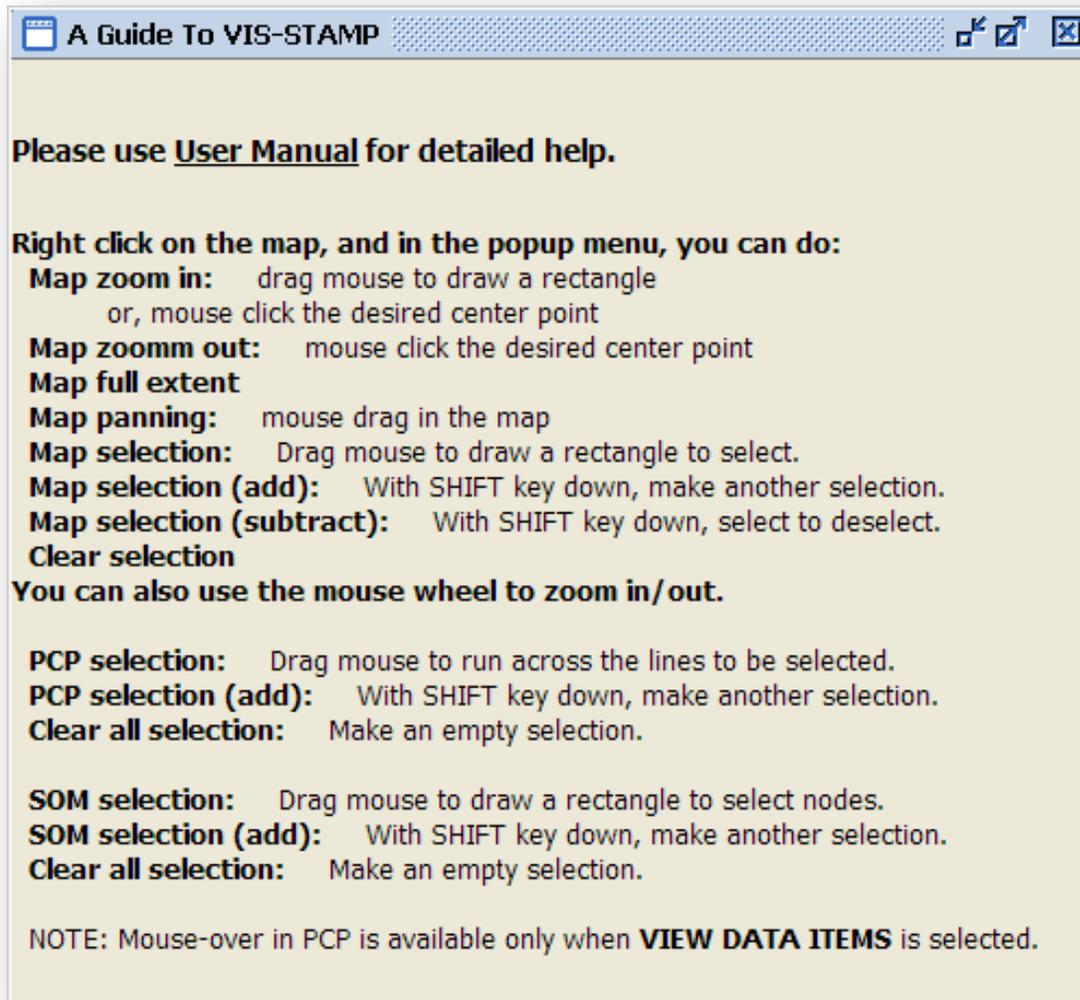
INTERACTIVE USER EXPLORATION AND INTERPRETATION

In addition to constructing a holistic view of patterns in the spatio-temporal and multivariate data cube, VIS-STAMP also supports a variety of user interactions that allow the analyst to examine patterns in detail. Each view is able to support user selections and the selection made in one view will be highlighted in other views. The user can make a selection in one component and then refine that selection in the same or another visual component by adding or subtracting new selection(s). Each component should be able to respond to selections made at different levels (i.e., data elements or clusters).

The user can right click the reorderable matrix, and use varying-size cells based on the values of a specific variable. Other settings such as “map matrix layout” and “background color” can also be changed through the menu “VIS-STAMP”.



To facilitate efficient user interaction, a set of short keys are supported (see below for details).



ACKNOWLEDGEMENTS

This software took was initially started and supported, in part, by NSF grant #9983445, NSF grant #EIA-9983451, grant # TS 1125, ATPM/CDC and grant CA95949 from the National Cancer Institute (NCI).

Thanks to Dr. Masahiro Takatsuka (masa@vislab.net) for providing the initial SOM implementation and Dr. Frank Hardisty (hardisty@sc.edu) for providing the component for automatically coordinating events and listeners.