Data Visualization Final Project

Brandon Hosley

UIN: 676399238

Yanhui Guo, Ph.D

Data Visualization Final Project

Introduction

> *The final project is used to evaluate the understanding of the knowledge we learned from class, and ability to apply the knowledge into real academic and industry areas.*

This objective will be met through a critique of the paper Automatic generation of puzzle tile maps for spatial-temporal data visualization by Shih-Syun Lina, Juo-Yu Yanga, Huang-Sin Syub, Chao-Hung Linb, and Tun-Wen Pai.

The Data

This paper is about an algorithm for visualizing spatial-temporal data. In order to demonstrate a general utility the authors used a multitude of data that necessarily had a spatial-temporal aspect, that is, data with both geographic and time variables.

In summary, the paper shows visualizations of Air pollution and Resident transfers in Taiwan, Cancer Deaths in the United States, and Population in Japan; each by the country's Administrative divisions.

Figure:	Spatial Aspect	Temporal Aspect	Qualitative Aspect
Figure 1	Taiwan by Districts	Unlabeled	Air Pollution
Figure 1	Taiwan by Districts	Unlabeled	Residential Transfers
Figure 8	United States by State	Years	Cancer Death Rates
Figure 9	United States by State	Years	Cancer Death Rates
Figure 12	United States by State	Years	Cancer Death Rates
Figure 12	Japan by Prefectures	5 Years	Population

The Algorithm

This data visualization technique has two parts to its algorithm. There is the tile-map generating function and the Puzzle-Piece Tile generating function.

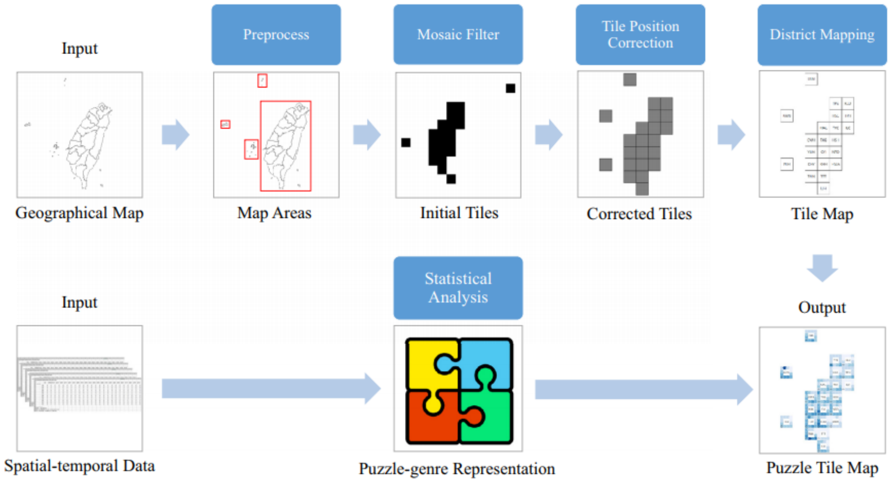


Figure 1. Algorithm Structure as described by Lin, Yang, Syu, Lin, and Pai (2019)

Tile-Map

Preprocess. First, a map of the region to be used is input using a GeoJSON format. The borders are simplified using the Ramer–Douglas–Peucker algorithm in the Turf.js library. Simplifying the borders improves performance for the future steps. Contiguous regions are determined so that they can be binned separately when drawing the tiles, this will prevent unintended interference from geographically separated regions.

Mosaic Filter. Each region is then processed by a mosaic filter. The authors specify that future implementations can use any filter that advantages the user but they describe their own method. The relevant regions are colored black and the remained area is set white. The filter divides the image into square tiles then sets the tile's color to the same as the majority of the pixels it contains. The number of colored tiles is then counted. This process is performed iteratively with increasing numbers of divisions until the number of colored tiles matches the number of districts to be measured.

Tile Position Corrections. Tiles are moved so that 1) Tiles from different region bins have at least one blank tile between them and any tiles of another bin, and 2) Tiles in the same region bin are contiguous; as defined by edges, not corners.

District Mapping. A matrix is formed with axes or dimensions representing the tiles and map location. Each value in the matrix is formed first from vectors that represent the distance from the center of the tile and the centroid of the district. The user may define an α to increase the relative value of distance. The next matrix is filled with an objective function based on distance and direction. The Hungarian algorithm is employed to solve for an optimized solution to this second matrix and districts are assigned to tiles accordingly.

Tile Generation

Now the tiles are placed and assigned to districts the algorithm must fill each of the tiles appropriately.

Statistical Analysis. The data must be processed into an appropriate form. The data for this visualization method must have factors representing district matchable to the first part of the algorithm, and each of those must have time series data with a time divisions equal to a multiple of four. While they only specify 8, 12, 16, and 20 this method could be arbitrarily extended to any multiple of four, though anything greater than 20 will likely be lost to the viewer.

Puzzle Pieces. The data is then given a color scale in a similar manner to a heat map. Each piece is wrapped around the outside of the tile in a counter-clockwise order. Connection points may be semi-circular, triangular, or rectangular and their position along the line is dependent on the rate of change from their parent piece to the next piece in line. The middle represents unchanged data, the top of the line is a 100% increase, the bottom a 100% decrease. Lastly, the name of the district, or an abbreviated name is placed in the center of the tile.

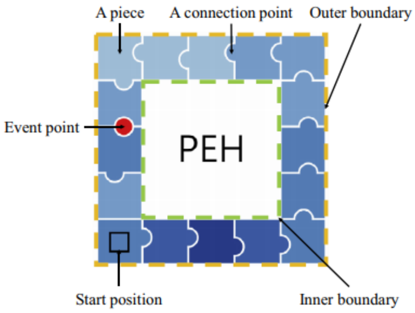


Figure 2. Example of a Tile

Optionally, the user can specify event points, which are a date and district, and this method will place a red dot on the appropriate puzzle piece.

Output

After each tile is built from the data it may be placed on the map in the area designated for it during the tile-mapping part of the algorithm.

This final map may be output as a static image or a custom interface built by the authors. The custom interface has additional functionality wherein a viewer may click on a tile to view a larger version of the tile. The user may also pick one of the time divisions and the view will highlight that time for each district, giving a transparency to all other times.

Results

As a new technique for data-visualization it is important to attempt to find an objective measure of value and effectiveness. For this, the authors designed a short survey about the data and measured two results; first they measured the accuracy which participants could answer questions about the data; second they measured the time taken to answer the questions. The results are reproduced below.

Quantitative evaluation for the generated puzzle tile maps.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
· Selection accuracy	95%	95%	100%	100%	100%	100%	100%	95%	100%
· Average time spent (s)	55.45	27.13	25.09	39.54	45.57	30.71	39.61	25.68	26.35
· Standard deviation of time spent (s)	26.5	12.9	15.04	22.89	18.25	13.9	12.28	12.24	10.9

Quantitative evaluation for the related visualization method.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
· Selection accuracy	75%	65%	65%	90%	100%	100%	65%	65%	65%
· Average time spent (s)	48.63	29.15	37.23	42.97	39.64	43.79	38.75	32.83	28.37
· Standard deviation of time spent (s)	38.4	15.2	37.19	29.59	21.16	32.32	26.53	20.97	21.31

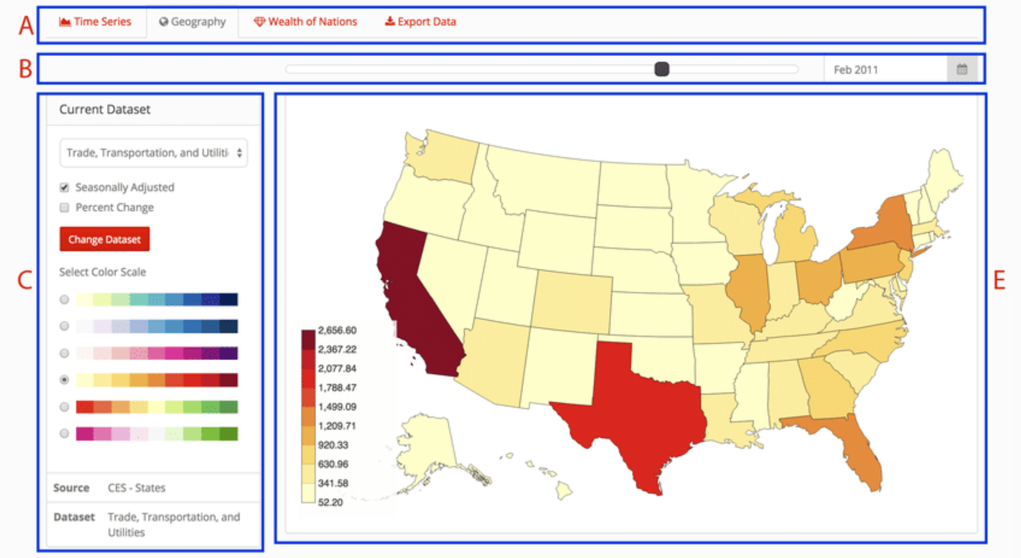
The control Visualization method used was a map with each district linked to bar charts showing the time series data.

Based on the reported results, both methods were very effective at conveying general spatial information, such as which districts had overall larger numbers in the given category and generally this information was conveyed faster than in the control method. The puzzle tile method was significantly better at accurately demonstrating temporal data and trends than the control method. For example, questions 7 through 9 asked about the effects of changes that occurred at specific times, with the puzzle tile method providing much better results.

Advantages of the Method

The puzzle tile map method seeks to add an additional dimensionality to the data that is being represented. In the most common examples of thematic mapping we see representations of scalar data with one or two additional dimensions being shown. For example, in Demers and Dorling cartograms the size of each area corresponds to one dimension of data being shown; in a Choropleth, shades of color represent the data. These two types of visualization can be combined to show two dimensions of data whilst correlating spatial information in a relatively intuitive manner.

The puzzle tile map retains the intuitive representation of the choropleth, but allows the user to represent time series data in a static image. The most common time series representation before this is a series of choropleths each representing a different moment in time.



Feb 2011

Current Dataset

Trade, Transportation, and Utiliti

☑ Seasonally Adjusted
□ Percent Change

Source CES - States

Dataset Trade, Transportation, and Utilities

Figure 3. Slider Example from Mancini, Bengfort, and Shneiderman (2016)
*Slider shown at label **B***

A single static image has several advantages over the series of images. If the series is condensed into a single pane then the user must define a length of time to progress through the slides if they wish to use a common file format (such as a gif or video format). A more interactive format involves a slider and the freedom for the viewer to control which time period they wish to observe, but while this is a simple enough function to deliver with a webpage, there are no common filetypes that allow this function. Additionally, neither of these mentioned methods function in a static environment, neither can be printed to paper in any of its forms. For this it is necessary to print the series separately, whether side-by-side or in a matrix. This should effectively show trends over a longer period of time but will not be as effective as the puzzle tile method for observing trends in single administrative divisions.

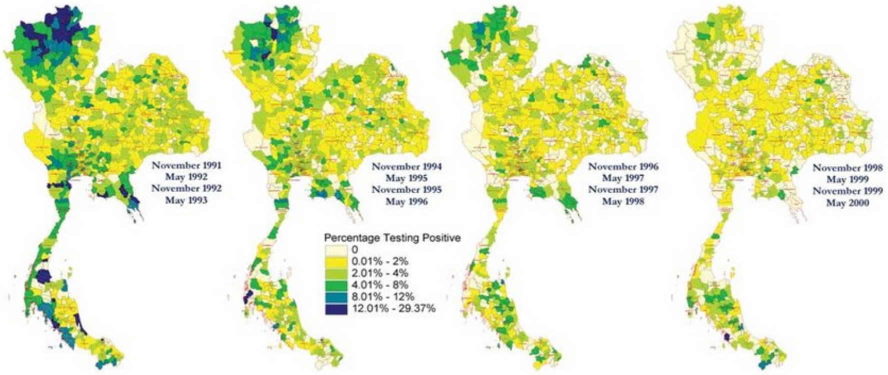
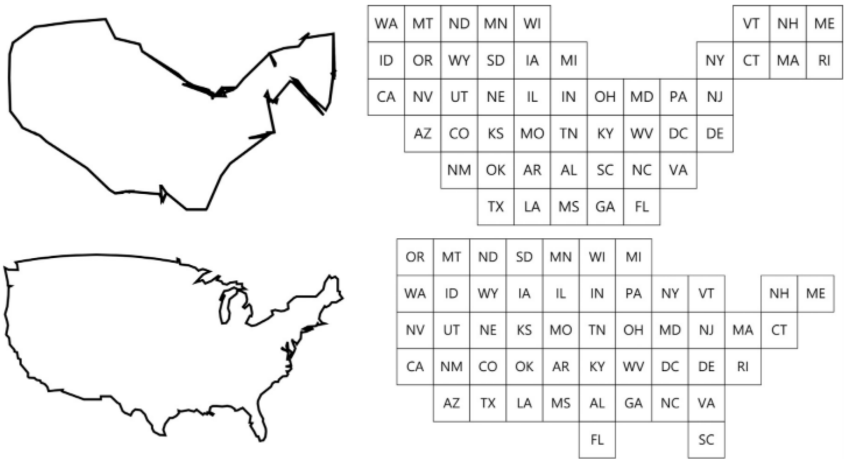


Figure 4. Time Series Example from Torugsa et al. (2003)

Disadvantages of the Method

The placement of the tiles appears to be a somewhat novel method with questionable results. Taken as a whole the final mosaic does appear to be effective at representing the general shape of the map that it represents. However, the placement of the tiles can often be counter-intuitive. Major difficulties in employing this tiling algorithm result from fine features such as peninsulas and bays, and when land areas of different district have a high or multi-modal variance. The first option can be resolved by deforming the map of features before applying the mosaic filter.



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

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

Figure 5. Map deformation option for tile placement. Lin et al. (2019)

The result without the deformation shows a mosaic with a more distinct silhouette. Once the mosaic's silhouette loses the distinct shape there does not seem to be a reason to leave it in that shape; perhaps at that point it would be better to deform the mosaic even further to better approximate the location of districts. The examples produce system level optimization that results in small areas of counter-intuitive placement. Example: Idaho placed between Washington and California, neither sharing a border with Oregon; Florida moved to the Texan peninsula; Washington and Oregon inverted; etc.

Another disadvantage comes from the scale being wrapped around the tile, the four sides give the restriction that the time series must come in a multiple of four. The corners themselves represent an interruption in the axes that the data is presented along. Further, the act of wrapping the data along the outside of each tile causes first quartile data to be adjacent to its neighbors 3rd quartile; second quartile next to the neighbor's fourth; etc. At these borders the direction of time is plotted in opposing directions and for each tile the direction of this plot progresses in each of the four orthogonal directions at different times. Without the puzzle piece ends there is no other intuitive way to know how the time series is supposed to progress. Starting from the bottom right and moving counter-clockwise is opposite of the most common measures of time.

Suggestions for Improvement

Instead of deforming the map to decrease the size of irregular features, the algorithm would be improved by some manner of normalizing the area of the districts. The Unites States is a great example of problems cause by uneven distribution. The majority of small states exist on the east side of the map, and most of the western states occupy relatively large areas. The Hungarian optimization in this case favors a shift of most states toward the west and causes some jitter that may look odd to people familiar with the region. Likewise it places the New England states along what would normally be the Appalachian region and assigns states to the peninsular landmarks that are geographically so far away as to not even border those states normally. This is likely to be a problem for any region that does not have a relatively even distribution of similarly sized districts. By investing the compute time to normalize the area of these regions before the mosaic tiling the tile map can more resemble something similar to the following example, which does a better job of showing the placement of the districts.

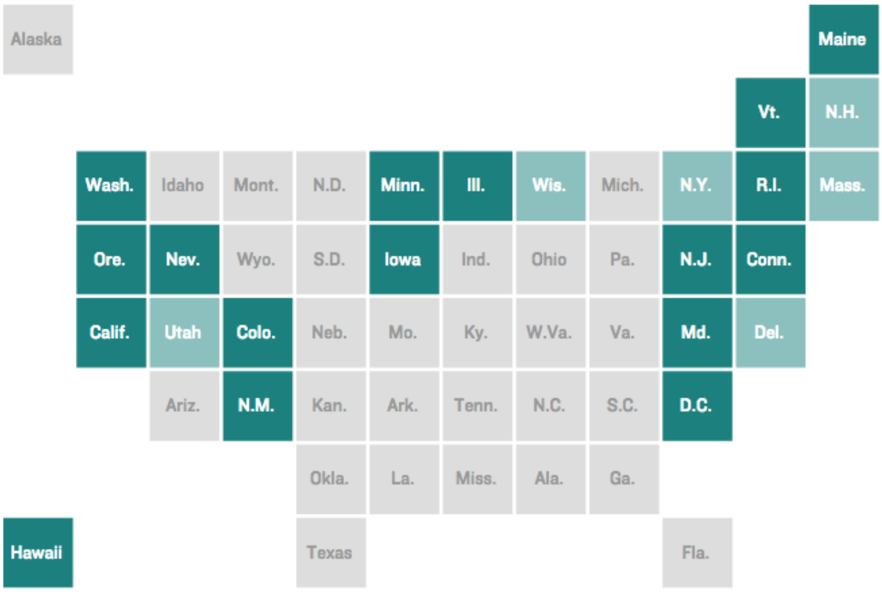


Figure 6. Example from NPR.

To improve the issues with the wrapped scale, the bends should be removed. Three ways to fix this are the vase method mentioned in the introduction of the paper, a facet grid method, or a polar coordinates method.

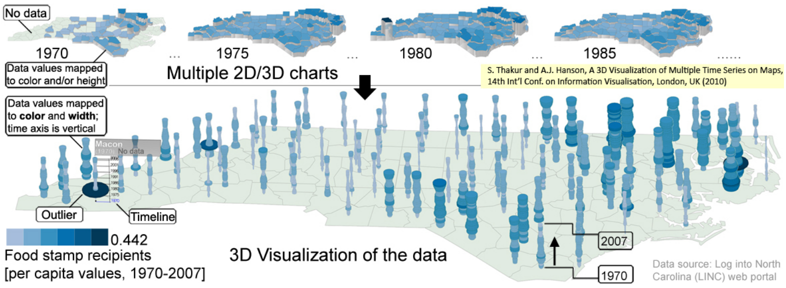

No data
1970
...
1975
...
1980
...
1985
......
Data values mapped to color and/or height
Multiple 2D/3D charts
S. Thakur and A.J. Hanson, A 3D Visualization of Multiple Time Series on Maps, 14th Int'l Conf. on Information Visualisation, London, UK (2010)
Data values mapped to color and width; time axis is vertical
Macon
No data
Outlier
Timeline
0.442
Food stamp recipients [per capita values, 1970-2007]
3D Visualization of the data
2007
1970
Data source: Log into North Carolina (LINC) web portal

Figure 7. Example from Thakur and Rhyne (2009)

The Puzzle-Tile method was designed to convey similar data as the Data-Vase model but in an easier to read manner. While the Puzzle-Tile method is neater and has fewer additional features the wrapped tiles are not easier to read than the 3-D Violin chart "Vases". Likewise, the geography being preserved makes for a quicker identification of districts within the region for anyone familiar with the region, with further benefit to those familiar with the geography but not the naming of the districts.

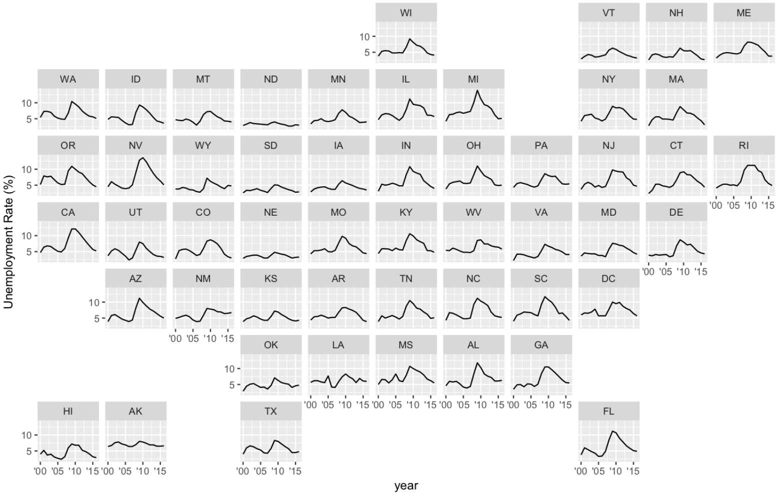


Figure 8. GoeFacet: Hafen (2018)

The GeoFacet is the method that I personally envisioned to solve the problem as described by the authors of this paper. This implemented an extension of GGPlot2's facet grid. It has nearly all of the benefits of the Puzzle-Tile method, and addresses all of the draw backs that I have previously mentioned. This implementation does not have its own grid generating algorithm, but rather uses grid that have been pre-built. The benefit is that the available grids are generally mature iterations, but there will be greater difficulties in implementing new or non-standard regions.

References

Hafen, R. (2018, Mar). *Ryan hafen.* Retrieved from `https://ryanhafen.com/blog/geofacet/`

Lin, S.-S., Yang, J.-Y., Syu, H.-S., Lin, C.-H., & Pai, T.-W. (2019). Automatic generation of puzzle tile maps for spatial-temporal data visualization. *Computers and Graphics (Pergamon).* doi: 10.1016/j.cag.2019.05.002

Mancini, P., Bengfort, B., & Shneiderman, B. (2016, 08). Interactive exploration of the employment situation report: From fixed tables to dynamic discovery.

Thakur, S., & Rhyne, T. M. (2009). Data vases: 2d and 3d plots for visualizing multiple time series. *Proceedings of the fifth international symposium on advances in visual computing: part II*, 929-938.

Torugsa, K., Anderson, S., Thongsen, N., Sirisopana, N., Jugsudee, A., Junlananto, P., . . . Brown, A. (2003, 08). Hiv epidemic among young thai men, 1991-2000. *Emerging infectious diseases*, *9*, 881-3. doi: 10.3201/eid0907.020653