Vis Comput (2019) 35:191–204
https://doi.org/10.1007/s00371-017-1461-y

ORIGINAL ARTICLE

GeoBrick: exploration of spatiotemporal data

Ji Hwan Park[1] · Saad Nadeem[1] · Arie Kaufman[1]

Published online: 30 November 2017

Abstract We present GeoBrick, an interactive technique for exploring spatiotemporal data. In GeoBrick, each region is comprised of multivariate data, which is encoded into simple shapes with colors. Additionally, users can adjust the resolution of data values to get an overview as well as details of the data. GeoBrick allows users to (1) juxtapose data and spatial profiles of discontiguous regions, (2) identify temporal patterns of user-defined classes of regions, and (3) comparatively evaluate across distinct configurations of regions. We demonstrate the effectiveness and efficacy of GeoBrick using two case studies.

Keywords Visual analytics · Spatiotemporal visualization · Multivariate data · Interactive data analysis

1 Introduction

A number of multivariate visualization techniques have been developed [7,12,13,17,19,37,39]. Traditional spatiotemporal visualization approaches cater to one or two of the following aspects but not all: (1) importance of neighboring regions/information, (2) inherent hierarchical structure between areas (e.g., state/county/city), (3) presence of time-dependent multiple variables. In the GeoBrick visual analytics framework, we cater to all these aforementioned aspects while allowing the user to visualize, explore, and analyze spatiotemporal data. We characterize the goals of users and derive design goals and tasks. To perform the tasks, GeoBrick allows different time periods and relationships between multiple variables to be visually examined concurrently (Fig. 1).

Broadly speaking, two distinct approaches for visualizing relationships between multiple categories/variables have been employed, clustering (e.g., [1]) and glyphs (e.g., [3]). Clustering-based approaches rely first on aggregating the variables and then assigning colors to the clusters. The advantage of this approach is that it can scope more variables than the glyph-based alternative. The disadvantage of clustering is that the attributes for individual data points cannot be visualized, since different attributes can be clustered together.

Glyph-based approaches, on the other hand, simply assign glyphs or visual representations to individual data points and thereby allow for visualizing at a much finer resolution than clustering-based approaches. In addition, glyph-based approaches can easily cater as many as 12 variables, which is also the upper limit of colors that humans can distinguish simultaneously [36]. Hence, in this paper, we use glyphs with 12 variables to underpin our GeoBrick platform.

The aim of our GeoBrick is to detect regions with similar or different attributes, to analyze local neighborhoods, and to identify relationships between multiple variables. GeoBrick provides several views to meet these aims. The details of these views will be discussed in Visualization section. The main contributions of this paper are summarized as follows:

- We provide users with visual encodings for comparing data points from regions, and for exploring temporal variation of selected variables in spatiotemporal data.
- We offer interactive tools within our platform to aid users in local neighborhood analysis and temporal analysis of selected regions.

✉ Ji Hwan Park
jihwan@cs.stonybrook.edu; jihwpark@cs.stonybrook.edu

Saad Nadeem
sanadeem@cs.stonybrook.edu

Arie Kaufman
ari@cs.stonybrook.edu

[1] Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

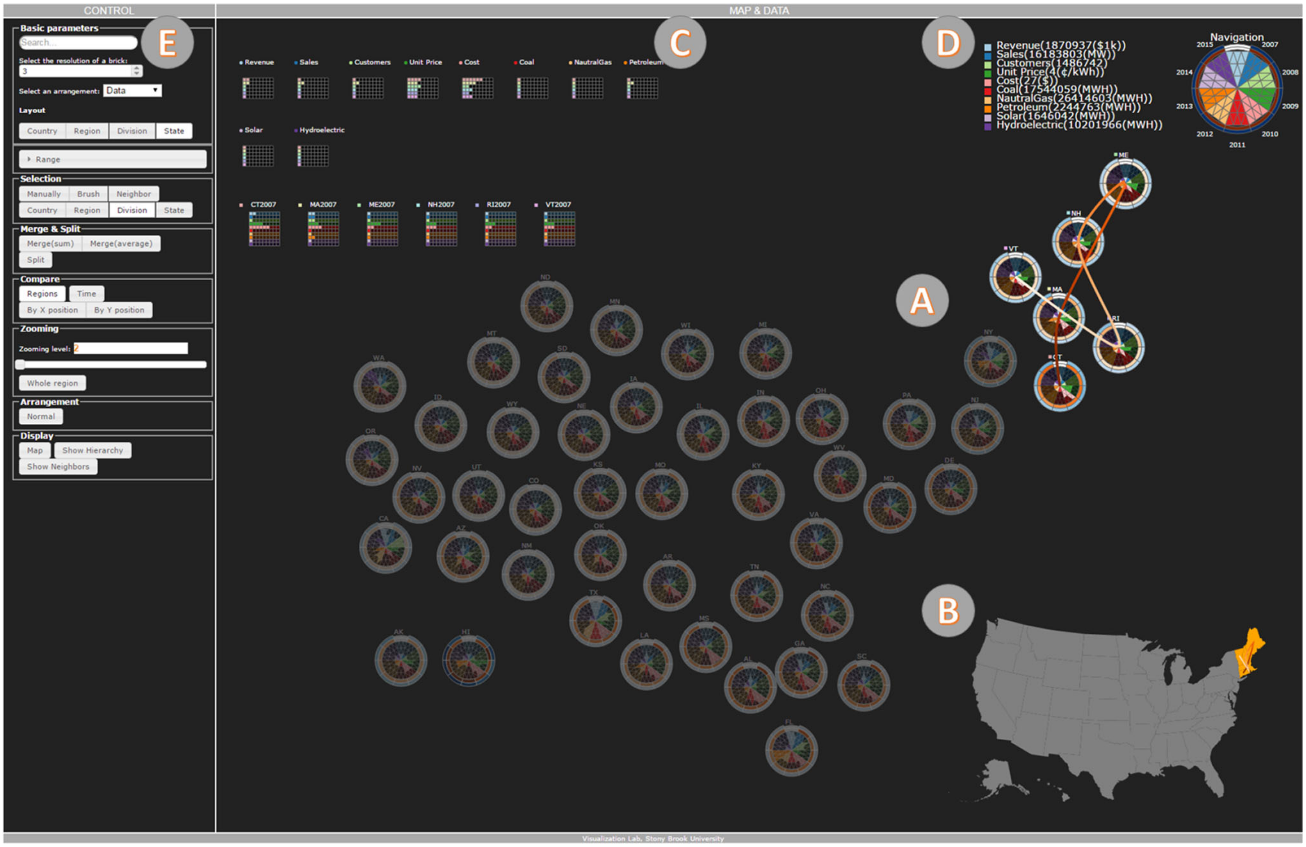


Fig. 1 An example of GeoBrick for local neighborhood analysis (the New England states) based on revenue from residential electricity sales. (A) The Abstract View displays abstracted spatiotemporal data, (B) the Map View shows the spatial relationship of the selected regions on a map, (C) the Comparison View allows users to compare among selected regions or time variation at a selected region, (D) navigation legends and polygon allow policy makers to select a variable and reordering variables across regions, and in (E) the control panel, policy makers can modify variable ranges, the resolution of polygons, the type of comparison and merging, and other basic operations for interaction. In the Abstract View (A) and Map View (B), the path depicts the order of regions based on a selected variable (with the darkest line depicting a region with the highest value and the brightest line representing a region with the lowest value)

– We illustrate the effectiveness of our platform with two use cases.

2 Related work

There are many previous approaches to efficiently visualize spatiotemporal data. Descartes [4] visualizes variables based on choropleth maps with other visual encodings such as shapes and sizes. It also allows users to interactively explore maps, where each map was displayed in a separate window. Andrienko et al. [2] have described several issues and recommendations for spatiotemporal visual analytics. STempo [31] has also visualized events to analyze them by offering several views to explore spatiotemporal events by using computational methods. Jern and Franzen [18] have visualized multivariate data by integrating several techniques such as parallel coordinates, time graph, time trend graph, and choropleth maps to analyze spatiotemporal data. Hoeber et al. [16] have introduced GTdiff to visualize spatial and temporal differences.

GTdiff consists of three views such as a temporal, difference, and geospatial views. The temporal view allows users to select a specific time period and generated temporal bins. In the difference view, differences between all the possible pairs of temporal bins are displayed. The geospatial view shows detailed information from a selected elements(s) of the other views. Slingsby et al. [29] have presented a technique to explore the Output Area Classification. It contained various techniques such as dot maps, rectangular hierarchical cartograms, bar charts, and parallel coordinates plots. Lastly, Andrienko et al. [1] have suggested a framework based on Self-Organizing Map to interactively analyze spatiotemporal data. The Great Wall of Space-Time [33] has visualized selected spatial regions by creating a line among them and add a temporal component by extending the line to a 3D surface. All the above work focused on comparing individual variables.

A few approaches similar to our GeoBrick have also been proposed. Guo et al. [14] have developed VIS-STAMP to visualize multivariate data. In their approach, multivariate data were clustered using a Self-Organizing Map (SOM) and the resultant clusters were visualized as a reorderable matrix and map matrix. Moreover, the patterns between variables were shown on parallel coordinates. The clustering is a limiting factor in this approach since users cannot find regions with exactly the same data values. Goodwin et al. [11] have focused on comparing variables with different scales and locality in only the spatial domain. There are spatiotemporal visual analytics frameworks for just one variable (i.e., trajectory data) by simplifying the complex data [20] and using 3D animation [5] and multiple views [6]. In contrast, GeoBrick can display multiple variable data with discrete spatial scales and temporal variation to understand the relationships between these variables.

3 Design goals and tasks

Based on discussions with potential users and prior art [27], we derived design goals of GeoBrick as follows:

- Help users interactively cluster geographical regions and show the corresponding spatial distributions.
- Explore relationships between temporal variables within a specific region.
- Perform local neighborhood analysis which allows selecting a region and comparing the selected region to its neighboring regions.

Spatiotemporal data can have different geographical units, time periods, and a hierarchy of geographical units (e.g., nation > regions > divisions > states in the USA). Based on these data characteristics and our objectives, GeoBrick allows the following tasks to be performed:

T1. Find regions with (dis)similar data attributes (e.g., *Which state(s) have similar residential electricity consumption to Connecticut?*).
T2. Identify patterns (similarity or difference) in neighboring regions (e.g., *Do neighboring regions of Massachusetts have similar types of energy sources? Do the New England states have similar education and household income patterns?*).
T3. Find region(s) with the highest/lowest metric value in selected regions and how these regions are spatially distributed (e.g., *Which state(s) have the highest residential electricity expenditure in the Northeast region? Are these neighboring states?*).
T4. Compare regions with same/different geographical unit (e.g., *Does the state of New York have similar data points to the states in New England?*).
T5. Find a variable that correlates with another variable within a region/across regions? (e.g., *Why did the residential electricity bill for Hawaii decrease from 2014 to 2015? Does a strong relationship exist between increased solar power electricity and the decreased cost of residential electricity in US?*).

4 Visualization

GeoBrick offers three linked views, namely the Abstract View, the Map View, and the Comparison View, to help users perform the tasks mentioned in Sect. 3.

4.1 Abstract View

4.1.1 Glyph

The Abstract View provides an overview of the data points for neighboring regions in a certain spatial locality (Fig. 1a). One of our goals is to compare data points for each region (**T1, T2**). Star glyph is a popular method to compare data points in small multiples [10]. Here, we use a glyph (visual representation) similar to the star glyph, which shows multiple variables for a region in a specific time period, as shown in Fig. 2. More specifically, when we have n variables, we create an n-polygon to represent each region and then divide the n-polygon into n sections uniformly, where each section represents a variable. We have selected n-polygon representation because it can easily be extended up to 12 variables based on our experiments, which is enough for our target cases such as electricity consumption and US census data. Additionally, in an n-polygon, each section can have the same shape, which is similar to a star glyph. Throughout the rest of this paper, we refer to n-polygon simply as a *polygon*. We assign a color to each section to represent a variable because color is one of the pre-attentive features [36] and a popular element to encode data in geographical visualization [4]. In order to successfully distinguish between variables, we deploy the ColorBrewer color schemes [15].

Once we have determined the color of each variable, the data value of each variable is represented as the number and the size of the most basic unit in our polygon representation, triangles. We have opted for a triangle to represent a data value because we can tightly divide each section of polygons which itself is a triangle. The number and the size of the symbols are determined by the resolution of the symbols. The users might want to perform our characterized tasks (**T1–T5**) at varying granularity. For example, a user may want to find states with higher residential electricity consumption

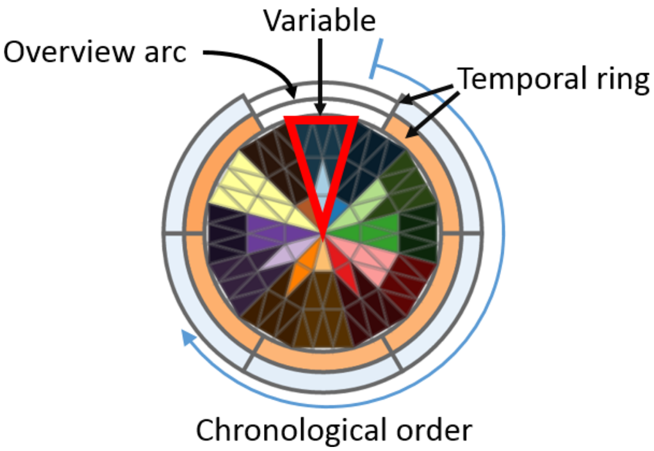


Fig. 2 An example of our glyph, including 12 variables and two selected variables for temporal rings. A polygon is divided into 12 sections, where each section represents a variable (example marked in red). Each temporal ring visualizes temporal variation of a selected variable. Each arc/subsection in a temporal ring denotes a specific time period; we arrange these in a clockwise direction with chronological ordering. The color of each arc represents data values. An overview arc allows users to see the difference of a region across time by selecting that arc

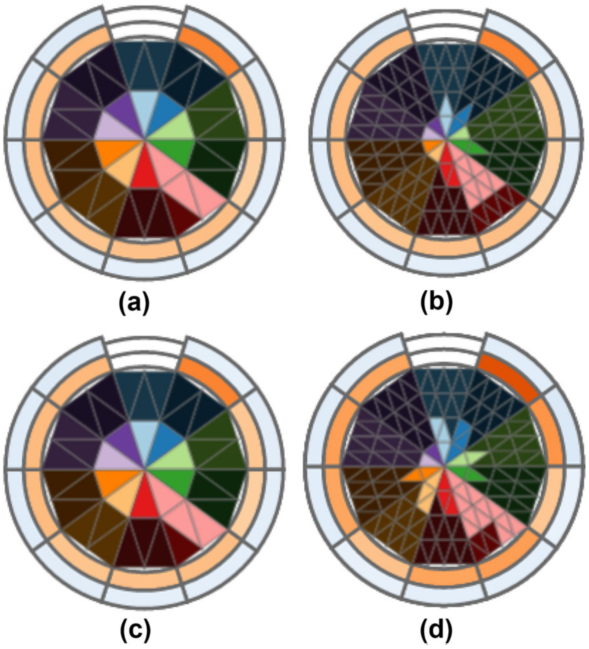


Fig. 3 An example of varying resolution to show data points from two regions—Region 1 (**a** and **b**), Region 2 (**c** and **d**)—with different levels of granularity. **a** and **c** show resolution 2, and **b** and **d** illustrate resolution 4. There are no visible differences at resolution 2, but when the resolution is increased to 4, these differences become more pronounced

compared to other states, and then find the state with the highest residential electricity consumption and the state with the lowest one. For this purpose, we allow users to change the resolution of symbols (the granularity of the data), where i resolution indicates i triangles on an edge of each section. If users want to see more details or abstract of datasets, they can increase or decrease the resolution (Fig. 3).

If the current resolution is i, the total number of symbols for each section or variable is i^2. After computing the number of symbols for each variable, we calculate a data value of each symbol, as follows. First, we set the range of each variable by computing the maximum ($\max_j$) of each variable (j) in all the regions and set zero as the minimum value ($\min_j$). The users can adjust the range manually to remove outliers or magnify the small difference of a variable across regions. Given the range of the variables, the data value of a symbol for each variable is $(\max_j - \min_j)/i^2$. We use two types of symbols, namely active and inactive symbols. Active symbols denote a data value of a variable while inactive symbols represent the difference between the defined maximum data value of the variable and the data value of the variable in the selected regions. Given the data value of each symbol, we compute how many active symbols (N_{active}) are required to show the data value of each variable, and then we define the number of inactive symbols (N_{inactive}) as $i^2 - N_{\text{active}}$. Active symbols are colored according to the method described previously. We note that in the current implementation, the background color for GeoBrick is black, and thus we opted for a darker shade for inactive symbols to make them unnoticeable.

After we calculate the number of active and inactive symbols, we arrange the symbols. The active symbols are placed from bottom to top and left to right of each section to compare variables within each polygon. When we arrange the symbols, we keep symbols as close as possible by switching the order of every leftmost symbol to their next symbols. We note that unlike other active symbols, the last active symbol of every variable in the arrangement indicates a value between 0 and $(\max_j - \min_j)/i^2$.

When the users change the resolution of polygons, the sizes of all the symbols are changed accordingly to maintain the size of polygons irrespective of resolution. If the current resolution is i and the size of a symbol is S_i, the size of the symbol will be $S_i \times \frac{i^2}{(i+1)^2}$ or $S_i \times \frac{i^2}{(i-1)^2}$ for a changed resolution $i+1$ or $i-1$, respectively.

In GeoBrick, the order of variables can be computed based on the correlation among variables. First, users select one variable. Next, we compute the correlation between the selected variable and the other variables by using Pearson's correlation coefficient. We then place variables with a higher correlation coefficient close to the selected variable while variables with a lower correlation coefficient are positioned on the opposite side of the selected variable. When comparing selected variables, the arrangement of variables in each polygon might be important because it enables users to view the pattern of selected data values. Thus, GeoBrick also allows users to reorder variables manually.

4.1.2 Temporal ring

In GeoBrick, data vary spatially as well as temporally. To visualize time-varied selected variables, we provide a temporal ring, where a series of concentric rings indicating selected variables are wrapped around a polygon. We chose a concentric ring because it can be tightly integrated into our polygon. In order to compare the difference between regions with different time periods (**T4**), we allow users to select a time period for each region by selecting an arc in the concentric ring. Moreover, we provide an overview of a time-varying selected variable to see the difference of a region across time. For the overview of time-varying data, we highlight the difference among time periods, where uncommon active symbols across time periods are brighter. We divide the ring into the number of time periods + 1 arcs (time arcs) to select time periods and present an overview (Fig. 2).

The color of each arc represents a normalized data value of a selected variable in a time period (**T5**). The normalized data value can be computed in two ways. The first method is global normalization, where each value is normalized based on the maximum and minimum data values of a selected variable over all regions. This normalization can help users find the difference among regions. The second method is local normalization, where the color of each arc shows the relative difference of a selected variable across time periods. This local normalization can be useful when the users analyze time variation of a selected variable in each region. Figure 4 illustrates our global and local normalization methods.

We assign white to an overview arc to distinguish it from arcs for other time periods because we do not assign white for the arcs. Moreover, we use the ColorBrewer color schemes for coloring the arcs. We make the overview arc thinner than other arcs to distinguish between these easily. Furthermore, we arrange the arcs in a clockwise direction based on their chronological order, and the arc for an overview is located at the top. When users select multiple variables, we can also add multiple temporal rings. Additionally, a temporal ring can visualize temporal variation of a sum/average of selected variables.

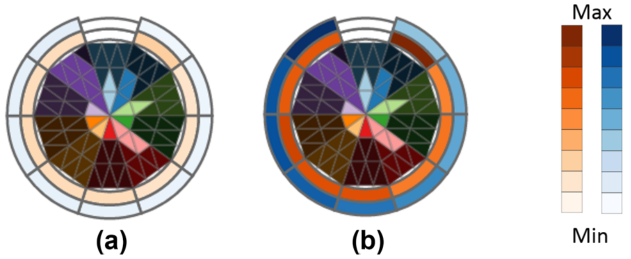


Fig. 4 An example of our **a** global normalization, and **b** local normalization for temporal rings. In our global and local normalizations, each value is normalized based on the maximum and minimum data values of a selected variable over all regions and in each region, respectively

4.1.3 Layout

As we mentioned earlier, one of our aims is local neighborhood analysis (**T2**), so preserving neighboring regions is important. Therefore, in the Abstract View, GeoBrick preserves neighboring regions in order for users to detect (1) a neighborhood with similar data points, and (2) a region with different data points within the local neighborhood. For this purpose, we can use an existing approach, as several layout algorithms have been proposed in the past. To save space and approximately preserve neighboring, we can easily apply algorithms such as [9,21,30] to polygons in the Abstract View. In this paper, we simply use a force-directed algorithm to preserve the neighborhood, where we consider each region as a node and create links between nodes if two regions share their border. If users want to see the data of discontiguous regions such as Hawaii and Alaska in the USA, the regions are placed close to the main layout (Fig. 5a).

4.2 Interactive operations

4.2.1 Clustering

One of our objectives is to identify regions similar to user-selected regions (**T1, T2**). For this, we use k-means clustering [38] to cluster regions based on selected variables. The k-means clustering algorithm requires users to set the number of clusters. In our approach, we start with a small number of clusters and interactively increase or decrease this number. Our experiments with several datasets show that 3–5 clusters are good to start with. Additionally, we observed that the adjustment of the number of clusters does not always result in

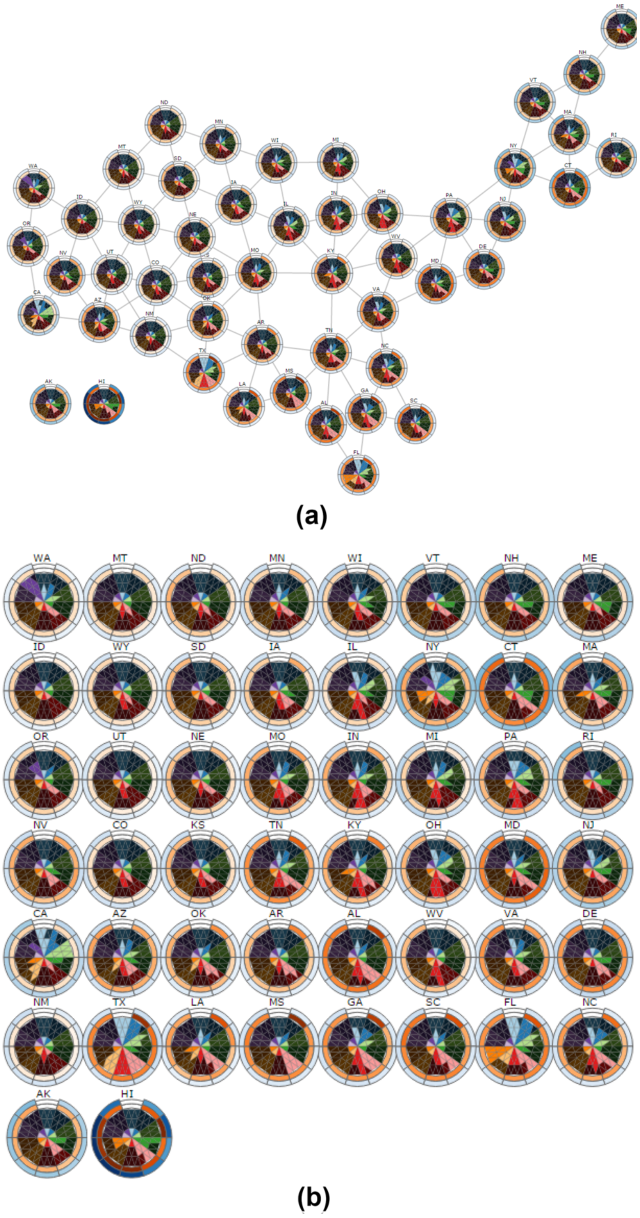


Fig. 5 An example of two different layouts for GeoBrick: **a** force-directed layout, and **b** CorrelatedMultiples [21]

better clustering. Therefore, we allow users to interactively include/exclude regions based on their glyphs.

4.2.2 Merge and split

Spatiotemporal data can contain various geographical units and time periods, which are linked to several variables in GeoBrick. In order to explore the data across those units and periods (**T4**), GeoBrick needs to support the change in geographical units and time periods, for example, merging regions to obtain a super region at a higher level in a hierarchy of geographical units; splitting a region into several subregions at a lower level in the hierarchy of geographical units; or merging data values from different time periods.

In GeoBrick, we offer methods to easily merge or split regions. For merging regions, if data regarding a super region of selected regions are available, we visualize it. In cases where these data are not available, we create this super region data based on selected regions. In this case, we use two types of operations for merging data: aggregation and averaging. For aggregating data from the selected regions, we compute the size of a merged polygon by calculating the maximum of merged variables. The size of each symbol does not change. We sum the data values of merged variables from the regions and then compute the number of active and inactive symbols for each merged variable. Averaging is done by deriving the mean for data values of variables from selected region. After creating a merged polygon by aggregating and/or averaging data, we average the center of the selected regions at the position of a merged polygon. We also compute neighboring regions of a merged region based on selected regions' neighboring regions. We then reapply a layout algorithm to all regions to avoid overlap between a merged region and existing regions. We only split a region into subregions when their data are available about subregions. In other words, if a selected region is a merged region, we simply remove the merged region and display subregions, where we already have the data available. In Fig. 6, we use the aggregation method to merge eight variables and the averaging method for two variables (pink and green).

In some cases, users need to compare between a region and one of its subregions. For this purpose, we also provide a method to visualize both a merged region and subregions, where subregions are displayed at their original positions, and a merged region is located at the Comparison View.

For merging time periods, users select time periods and then merge data values through either an aggregation or averaging method, as described above. When the data are merged, we also merge selected time arcs.

4.2.3 Comparison

In the Abstract View, polygons for two regions have the same size, and each polygon can show the same range of a data value for each variable if the polygons are not merged. Users can compare the regions by dragging one region onto another region (**T4**), which is similar to the approach used in OnSet [28]. When the distance between two regions reaches less than a certain threshold, we start comparing the regions. Similar to our temporal overview approach, the difference between the two regions is highlighted. We compute whether each active symbol in the regions is common in both regions (common active symbol) or not (uncommon active symbol). An uncommon inactive symbol is a symbol located at the same position as an uncommon active symbol in the

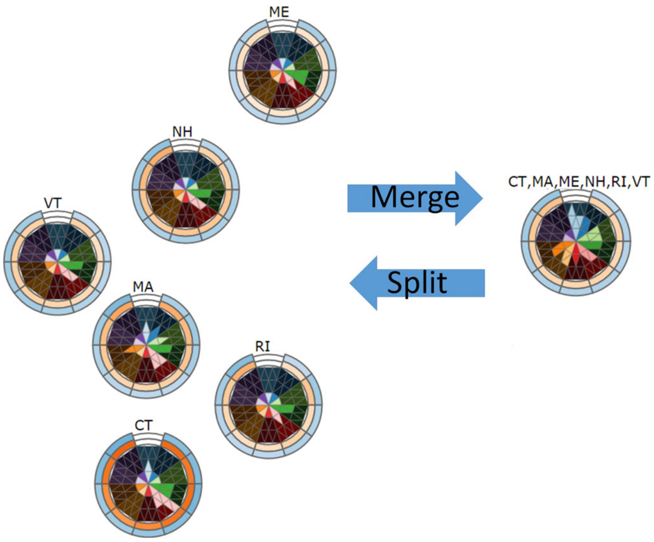


Fig. 6 An example of merging and splitting the New England states, including New Hampshire (NH), Maine (ME), Connecticut (CT), Massachusetts (MA), Vermont (VT), and Rode Island (RI), using averaging and aggregation. After the user selects the states, two variables (green and pink) are merged by using averaging, and the others are merged by using aggregation

other regions. We then make uncommon active symbols and uncommon inactive symbols brighter and darker, respectively.

If the two regions have different polygon sizes due to merging regions, all the common active symbols can be shown, but all the uncommon inactive symbols in a smaller polygon cannot be shown. However, since they have the same layout for the variables, we can focus on a bigger polygon to see the difference and similarity between the two regions by displaying all types of symbols while the other region illustrates only symbols, which can be illustrated within the region.

4.2.4 Labels and shapes

In order to assist users to identify each region more efficiently, we provide a label and a way to show the actual location and shape of each region. A label shows the abbreviation (e.g., NY for New York) or the name of each region (e.g., Northeast), and it is placed at the top of each region. In addition, we also add a symbol next to a label, where the color of a symbol is the same as that in the Comparison View and Map View.

In order to display the shape of each region, we normalize each region based on its area to focus on its shape instead of its size. We also blend the shapes of the regions and their corresponding polygons together to show both the shape and data of a region.

Lastly, to help users understand the range and position of each variable, we show navigation legends and the navigation polygon. All operations on the navigation polygon are applied to all the regions. Navigation legends present the color of each variable, its name, and the range of a symbol. Every polygon has the same layout, as shown by the navigation polygon in Fig. 1d.

4.3 Map View

The Abstract View shows local neighboring regions, but cannot show the geographical context, the exact location and size of each. Thus, we provide the Map View, which shows each region in an actual map (Fig. 1b). When the users select a region(s) in the Abstract View or the Map View, we highlight those regions in both views. When a user finds similar regions using our clustering method, these regions are also highlighted in the Map View (**T1**). Additionally, when we analyze the local neighborhood, the colors of selected regions in the Map View are the same as the colors of rectangles for the corresponding regions (**T3**). In the Map View, we can also overlay information such as the ordering path between regions, which will be described in the Comparison View.

4.4 Comparison View

The Abstract View shows only an overview of data points in regions and temporal variation of selected variables and provides very basic comparison capabilities. Thus, we present the Comparison View to analyze temporal variation in a region or local neighborhood, in a more detailed way (**T2–T5**).

In the Comparison View (Fig. 1c), we visualize selected information, similar to the Table Lens [26], which can help users visualize the relationships among selected regions or variables. For temporal analysis in a selection region, we visualize data values of variables in each period at the bottom, and the time variation of each variable at the top (Fig. 7). Each colored cell represents the data value as the same as a triangle in a polygon. The goal of this temporal analysis is to analyze the difference between each time period. Thus, we again utilize the global and local normalization methods to visualize different distributions (Fig. 7).

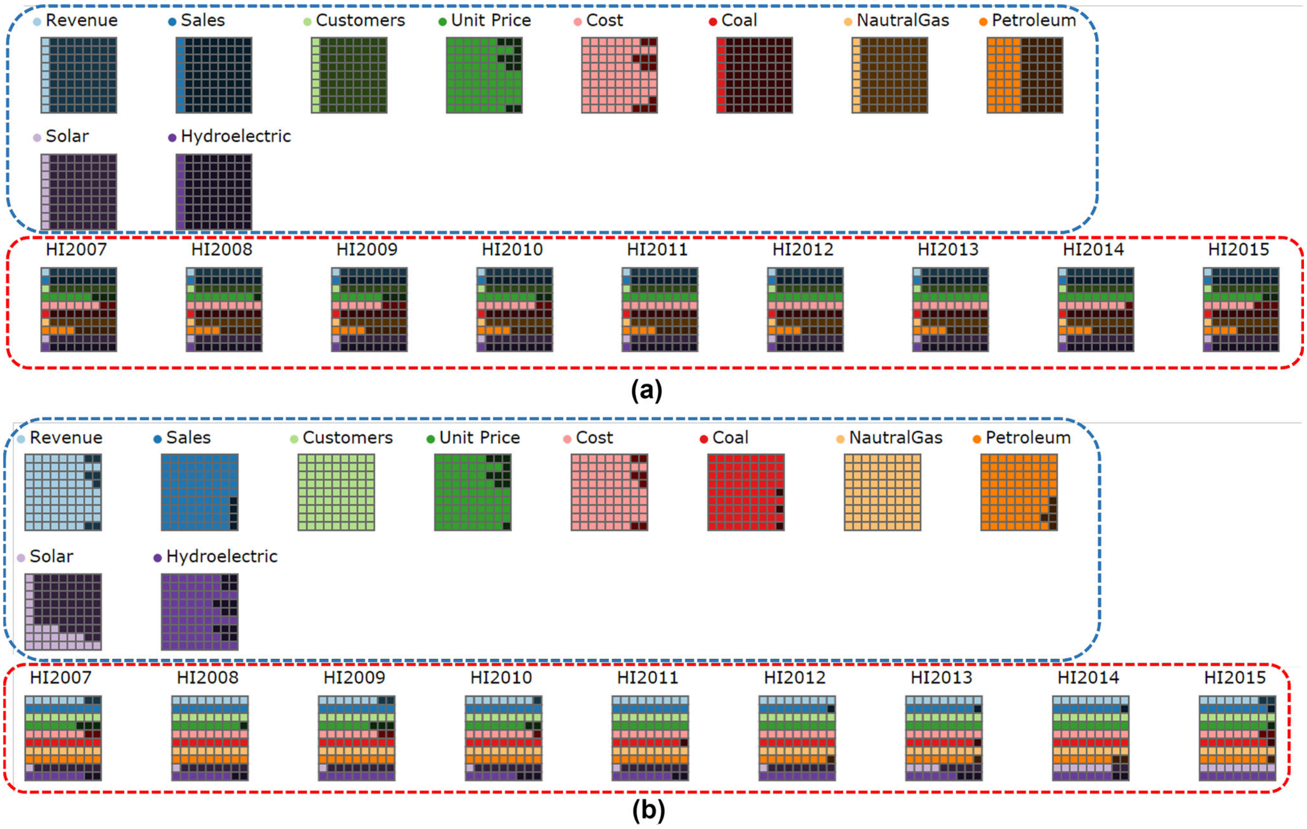


Fig. 7 An example of our temporal analysis from the data of Hawaii (HI) from 2007 to 2015: **a** global normalization, and **b** local normalization. In both normalization methods, the x-axis represents data values, and the y-axis indicates time (a dashed blue rounded rectangle: top—2007; bottom—2015) or each variable (in a dashed red rounded rectangle). In the top part (a blue rounded rectangle), each variable has a unique color, which is used in the bottom part of the view (a red rounded rectangle) to show data values of each variable in a region. In the global normalization, we cannot see the changes of solar power generation. However, we can clearly see the increase in solar power generation in the local normalization

For local neighborhood analysis, we also visualize two types of information as shown in Figs. 1c and 8. In the view for these two information, the x-axis indicates data value and the y-axis represents a region (top part) or a variable (bottom part). At the bottom of the view, we visualize selected regions and provide a colored square as a label for each region to find a corresponding region at the bottom of the display. Users can rearrange the regions by sorting them based on their x or y positions. At the top of the view, we group data points from the selected regions into variables to show the correlation between the variables. Users can sort regions based on a selected variable by clicking the name of a variable. In order to show the spatial relationship between sorted regions, a path between regions is drawn in both the Abstract View and Map View, where the color of the path indicates the order of each region based on a color scheme selected by users, e.g., the darkest line shows a region with the highest value while the brightest line indicates a region with the lowest value (Fig. 1a, b).

In temporal rings, users can select a specific time period for data points within a region. This is useful for comparing data points from two regions with different time periods. It is still, however, difficult to compare data points from multiple regions with different time periods. For this purpose, we provide a way to visualize data values of variables from selected regions in each time period. In this visualization, each row shows data values of variables from a given region for a specific time period (Fig. 9a).

4.5 Interaction

We provide several basic interactions to help users understand the overall data. As we described previously, users can change the resolution of all the polygons to obtain an overview of data. To assist users in exploring different levels of aggregation of regions, we allow users to choose an initial geographical unit of all the regions to be visualized. When users hover over a variable, we show the name of the pointed variable and the data value of the selected variable on the tooltip. Users can zoom into a specific region by increasing the size and resolution of the region. Additionally, we allow users to select regions by manual selection and brushing.

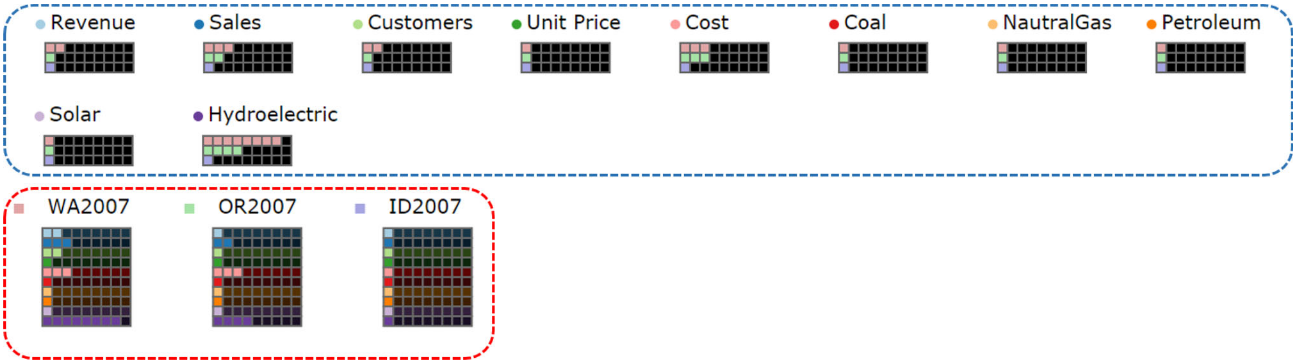


Fig. 8 A comparison of the local neighborhood of Washington in 2007. In the top of the Comparison View (a dashed blue rounded rectangle), we group data points from the selected regions into variables to show the correlation between the variables. Each variable has a unique color, which is used in the bottom part of the view (a dashed red rounded rectangle) to show data values of each variable in a region. In the bottom of the view (a dashed red rounded rectangle), we visualize selected regions and provide a colored rectangle as a label for each region (a dashed red rounded rectangle), which is used in the top view to represent data values of a corresponding region. In the top and bottom parts, the number of squares represents data values

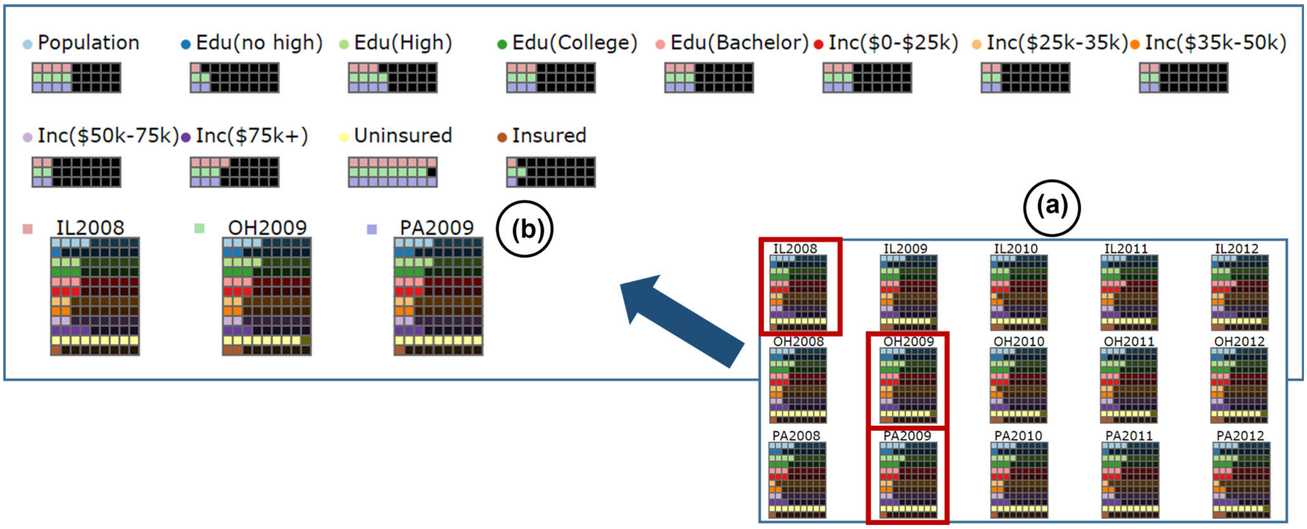


Fig. 9 An analysis of Ohio (OH), Pennsylvania (PA), and Illinois (IL): **a** comparison of overall temporal variation of three regions and selected time periods (outlined with red boxes), and **b** analysis of three regions with selected time periods

Once users have selected the region(s) of interest, the neighboring regions or regions belonging to the same selected geographical unit can be automatically selected.

5 Case studies

We describe two examples to demonstrate the benefits of GeoBrick. In the first case study, we use GeoBrick to understand the sales and generation of residential electricity (10 variables) in the USA. In the second example, GeoBrick is used to explore the US census data (12 variables). In both cases, we use a hierarchy of geographic units from US census [34], i.e., nation > regions > divisions > states. We also use our force-directed layout for both case studies because we focused on neighboring information of each region, and we have enough space to display all the data.

5.1 Residential electricity generation and sales in the USA

Electricity is one of the essential components of modern life. The US Energy Information Administration (EIA) releases data on annual residential electricity generation and sales every year [35]. From these datasets, we used information on the amount of sales, revenue from the sales, the number of customers, the unit cost, and average electricity bills from 2007 to 2015. We also used data regarding different energy sources such as coal, natural gas, petroleum, solar, and hydroelectric power for the period from 2007 to 2015.

Analyzing this dataset allows our potential user (e.g., a policy maker) to understand where obvious pressures exist and where optimization could potentially be introduced with new, local sources of power.

First, we wanted to compare regions in the New England states in terms of residential electricity consumption (**T1**). We selected five variables (revenue, sales, the number of customers, unit price, and electricity bills) to cluster regions based on these variables and select one of the New England states such as Rhode Island (RI). Massachusetts (MA) and Connecticut (CT) belonged to a different cluster because of the unit price and the electricity bills. We then chose these two variables and visualize their temporal variation through temporal rings (**T4**). Connecticut had higher electricity bills and unit cost than MA from 2007 to 2015.

Among states with the highest residential electricity bills, Hawaii (HI) and Delaware (DE) had relatively low electricity sales (**T3**). We found that HI had the highest unit price among all states. We then analyzed its temporal variation through the Comparison View (**T5**), as shown in Fig. 7b. In 2015, the electricity bills dropped significantly compared to 2014 due to the drop in unit cost. To see the effect of electricity generation in more detail, we explored and found an increase in solar and hydroelectric power generation.

Lastly, we determined that Washington (WA) had one of the lowest unit costs for electricity in the USA because of hydroelectric energy. We wanted to know whether its local neighborhood had similar electricity generation (**T2**). We found that Washington, Oregon, and Idaho had similar electric generation through our clustering approach. We then evaluated these regions in the Comparison View and found that they had similar unit costs and electricity generation (Fig. 8).

5.2 US census

The other dataset we explore using GeoBrick is US census data. We use data from the US Census Bureau [34] and National Center for Health Statistics [23]. In doing so, one can determine where new critical infrastructure and public services may be required in relation to population. There are various types of information related to census data. Among these, we first chose five variables: total population, education attainment (no high school, high school, college (< 4 years), bachelor or higher), household income ($0–$25k, $25k–$30k, $35k–$50k, $50k–$75k, > $75k), and health insurance status from 2008 to 2012.

First, we wanted to compare Ohio (OH), Pennsylvania (PA), and Illinois (IL), where IL and PA were the most similar states to OH in 2008. We visualized the overall time variation of OH, PA, and IL, and then compared data points for OH in 2008 to those of IL and PA in 2009 (Fig. 9). Neighboring states such as IL and OH had the same data values except with the insurance status, namely uninsured rate and insurance rate. One interesting point of note is that PA had a higher average household income ratio (> $75k) and lower no-high-school and only-high-school-degree rates than others. We further compared the neighboring regions of OH to identify whether the neighboring regions had similar data points in 2012 (**T2**). Overall, they had similar data points in 2012.

Next, we wanted to know which state(s) in the Northeast region had the highest average household income (> $75k) in 2008 (**T3**). We chose all states in the Northeast region by using a region selection method. We then sorted these based on average household income (> $75k) in the Comparison View. New Jersey had the highest average household income (> $75k), followed by New Hampshire (NH), Connecticut, Massachusetts, Rhode Island, and New York (NY). We also found that neighboring regions of CT had similar income levels. We then investigated neighboring regions of CT. To do so, we selected CT and chose all variables. We found that regions that were not adjacent to CT had more similar data points than its neighboring regions by using our clustering method (**T1**), as shown in Fig. 10.

Lastly, we merged all the states in the New England division and compared it to NY (**T5**). NY had lower average household income (> $75k) and uninsured rate than the averaged New England states.

5.3 Expert review

To evaluate our framework, we interviewed two potential users (policy makers working in the public policy domain). We first explained to them our system workflow and visual encoding, and then asked them to try GeoBrick out.

Overall, they expressed that GeoBrick could serve as a powerful and central tool for analysts and policy makers, allowing unusual integration of what is often highly discrete levels of analysis. They enjoyed the Abstract View because this is an unusual way to view (in)congruences among neighboring entities and regional profiles. One expert commented, “I can see a lot of value with its capacity to incorporate variables at a complex resolution.” The other expert said, “I like the Abstract View because it is compact visualization of multiple variables across time.” They especially liked our interactive clustering method, and merging and splitting operations because they can find regions with similar/different data points, where regions can have different geographical units. An interesting observation was that some experts changed the layout of our glyph based on their task, not dataset. For example, if a task did not require an exact neighborhood preservation, the expert preferred to use a grid layout. They expressed that the Comparison View was a powerful way to drill in and evaluate scenarios at a more granular level. Both experts appreciated showing the geographical context of selected regions in the linked Map View.

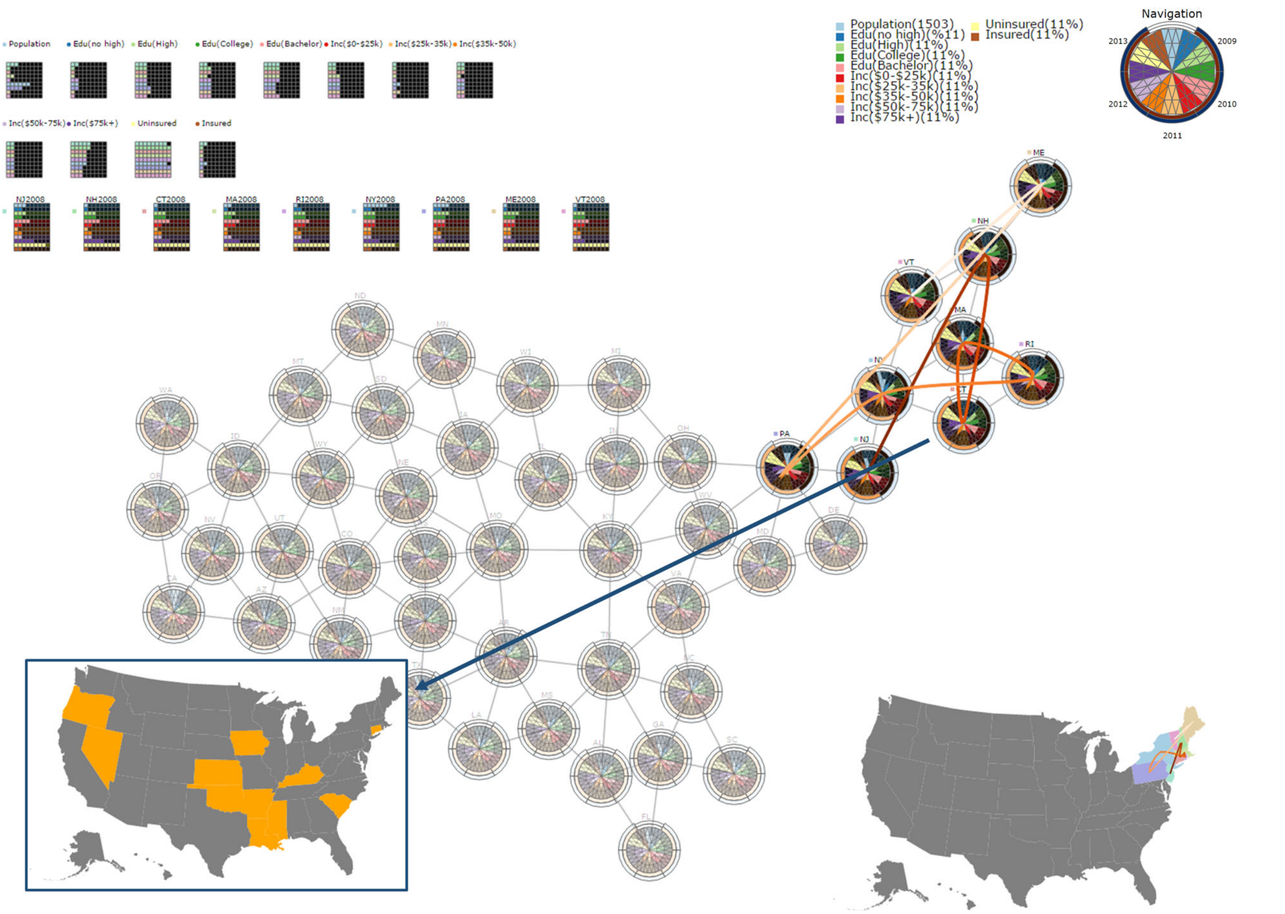


Fig. 10 Local neighborhood analysis in the Northeast region highlighted in the Abstract View and Map View (bottom right) by ordering states based on their averaged income (> $75k). States in the Northeast region are analyzed in the Comparison View. States with similar data points to Connecticut (CT) are highlighted in the Map View (bottom left)

One expert said that the Map View is useful to geographically orient users because users can select some states with this view and then move to the Comparison View to evaluate their profiles more fully. The other expert pointed out that the Map View is useful for spatial correlation compiling similarities and differences of states/regions against geographical units. Both experts really liked that they could interactively sort regions and that the sorted regions were shown as a colored path. They expressed that GeoBrick was not entirely intuitive for immediate use, so users should give themselves time to experiment; the experts were able to perform tasks easily (without any hindrance) after a brief 30-min tutorial. One expert suggested that it might be good to make unselected regions invisible in the Abstract View and Map View when users perform analysis in the Comparison View.

We also asked them to speculate about potential application areas. They proposed that it would be good for many areas such as the environment, education, and technology adoption analysis. One expert said, “GeoBrick would be good for public health assessments, tracking high-level regional similarities, and then drilling in for fuller socioeconomic correlations.” The other expert mentioned, “It would be a great tool for analyzing educational attainment based on census data, including test score averages, budgetary allocation, dropout rates, and information of other resources.”

5.4 Discussion and future work

Currently, GeoBrick is designed to deal with any spatiotemporal data (Fig. 11). Our technique can be extended to any multivariate data visualization without the geographical information constraint. For example, GeoBrick can be used in image analysis by exploring the graph of attributes of multiple connected components of a segmented image (or a segmented video frame). Another possible scenario is an exploration of planar graphs with nodes associated with multivariate geometric and temporal data. There are some limitations in applying GeoBrick to other spatiotemporal data.

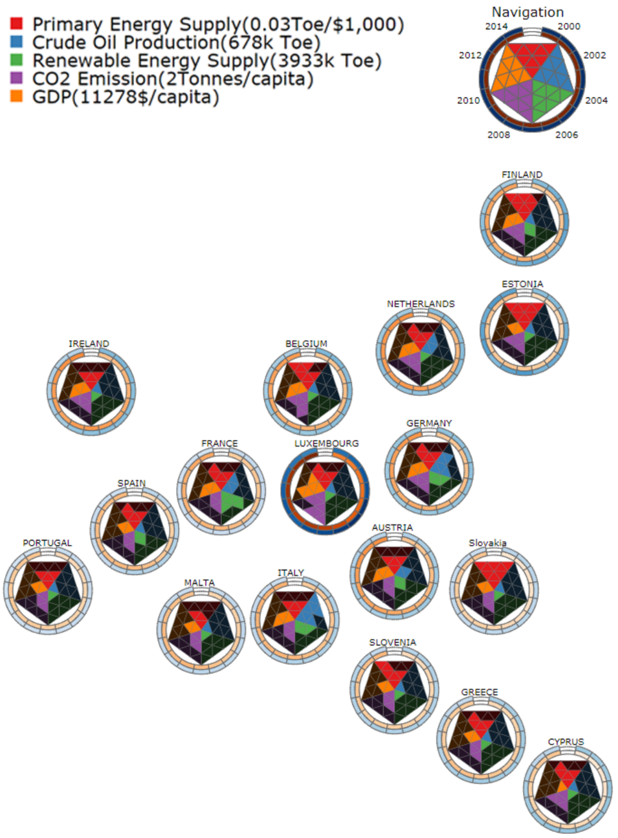


Fig. 11 An example of GeoBrick for the euro area data from 2000 to 2014 [24]. In the temporal rings, we visualize CO_2 emission and gross domestic product (GDP)

5.4.1 Scalability

In GeoBrick, we use color to encode each variable and assigning colors to more than twelve variables is a non-trivial task [36]. Moreover, when these variables have a hierarchical structure, the problem becomes even more challenging [32]. One possible extension to GeoBrick is combining color with other visual elements such as surface texture to represent more variables.

Another issue is that we can visualize temporal variation of only a certain number of variables, depending on the display size. However, policy makers we collaborated with expressed that comparing temporal variation of up to four variables was enough for most cases; on most displays, we can visualize temporal variation for these number of variables. Lastly, a traditional display (e.g., monitor, smartphone, tablet screen) has limited resolution, which is often not enough for displaying a large number of polygons and symbols. We will work on creating an optimal layout and conduct experiments to optimize the size and the number of polygons and symbols for different resolution displays from low-resolution displays to extremely high-resolution displays such as the Reality Deck [25].

5.4.2 Glyph design

In our glyph design, all triangles have the same size, so the glyph represents quantized values accurately. However, the position/order of each triangle in the glyph might not be intuitive to some users. Thus, although we had positive feedback from potential users on our glyph design, there are possible design alternatives depending on users' need. Users, wanting an intuitive order of data values with minimal training overhead, can use a glyph that emphasizes the order of data values such as a multi-level pie chart in which the number of levels is given by the level of resolution. However, every subsection of the pie chart does not have the same size, so people might misinterpret data values and difference between two regions. Thus, our design would be beneficial for users once they are familiar with our glyph ordering.

5.4.3 Spatial scale

In GeoBrick, we can show data from discrete spatial scales based on predefined geographical units. In the future, we will investigate approaches to select and show arbitrary-selected regions, for example, by drawing a region on the Map View, and extracting and visualizing data within that region.

5.4.4 Layout

In our use cases, we dealt with geographic entities at the state level by using our force-directed layout algorithm to maintain neighboring information accurately. However, one limitation of our force-directed layout is white space between regions, which limits the size of each region [22]. If users want to explore data at a finer level such as county, we can pack regions by applying an existing algorithm for small multiples (e.g., [8]). However, there is no existing algorithm to satisfy all factors for small multiples [22]. To solve this issue, we plan to further study this problem.

5.4.5 Correlation in the Comparison View

Lastly, in order to show a correlation between variables, we have used a visualization, which is similar to the Table Lens. However, it only shows a brief correlation among variables in selected regions. If the number of variables increases beyond a certain number, it can be difficult to understand the correlation of the variables. To remedy this issue, we can apply another technique [11], as a complementary method for our comparison method.

6 Conclusion

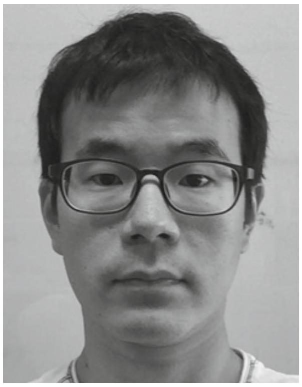
In this paper, we introduced GeoBrick, an interactive visual analytic tool for analyzing spatiotemporal data. We presented linked views such as the Abstract View, the Map View, and the Comparison View to facilitate this analysis. The Abstract View showed abstracted data points and regions to identify regions with similar/different data points. It also helped users analyze local neighborhoods. The main purpose of the Map View was to show the spatial distribution of selected regions. The Comparison View allowed users to analyze selected regions and temporal patterns of a selected region in more detail. We also offered interactions to aid experts in analyzing the data. Lastly, we demonstrated the effectiveness of our GeoBrick technique using two case studies and expert feedback from our potential users.

Acknowledgements This work has been partially supported by the National Science Foundation Grants IIP1069147, CNS1302246, IIS1-527200, NRT1633299, and CNS1650499.

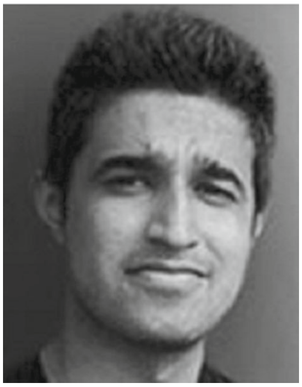
References

1. Andrienko, G., Andrienko, N., Bremm, S., Schreck, T., Von Landesberger, T., Bak, P., Keim, D.: Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. Comput. Graph. Forum **29**(3), 913–922 (2010)
2. Andrienko, G., Andrienko, N., Demsar, U., Dransch, D., Dykes, J., Fabrikant, S.I., Jern, M., Kraak, M.J., Schumann, H., Tominski, C.: Space, time and visual analytics. Int. J. Geogr. Inf. Sci. **24**(10), 1577–1600 (2010)
3. Andrienko, G., Andrienko, N., Fuchs, G., Wood, J.: Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. IEEE Trans. Vis. Comput. Graph. **23**(9), 2120–2136 (2017)
4. Andrienko, G.L., Andrienko, N.V.: Interactive maps for visual data exploration. Int. J. Geogr. Inf. Sci. **13**(4), 355–374 (1999)
5. Buschmann, S., Trapp, M., Döllner, J.: Animated visualization of spatial-temporal trajectory data for air-traffic analysis. Vis. Comput. **32**(3), 371–381 (2016)
6. Cibulski, L., Gračanin, D., Diehl, A., Splechtna, R., Elshehaly, M., Delrieux, C., Matković, K.: ITEA—interactive trajectories and events analysis: exploring sequences of spatio-temporal events in movement data. Vis. Comput. **32**(6), 847–857 (2016)
7. Claessen, J.H.T., van Wijk, J.J.: Flexible linked axes for multivariate data visualization. IEEE Trans. Vis. Comput. Graph. **17**(12), 2310–2316 (2011)
8. Dorling, D.: Area Cartograms: Their Use and Creation. University of East Anglia, Environmental Publications, Norwich (1996)
9. Eppstein, D., van Kreveld, M., Speckmann, B., Staals, F.: Improved grid map layout by point set matching. In: IEEE Symposium on Pacific Visualization, pp. 25–32 (2013)
10. Fuchs, J., Isenberg, P., Bezerianos, A., Fischer, F., Bertini, E.: The influence of contour on similarity perception of star glyphs. IEEE Trans. Vis. Comput. Graph. **20**(12), 2251–2260 (2014)
11. Goodwin, S., Dykes, J., Slingsby, A., Turkay, C.: Visualizing multiple variables across scale and geography. IEEE Trans. Vis. Comput. Graph. **22**(1), 599–608 (2016)
12. Gratzl, S., Gehlenborg, N., Lex, A., Pfister, H., Streit, M.: Domino: extracting, comparing, and manipulating subsets across multiple tabular datasets. IEEE Trans. Vis. Comput. Graph. **20**(12), 2023–2032 (2014)
13. Gratzl, S., Lex, A., Gehlenborg, N., Pfister, H., Streit, M.: LineUp: visual analysis of multi-attribute rankings. IEEE Trans. Vis. Comput. Graph. **19**(12), 2277–2286 (2013)
14. Guo, D., Chen, J., MacEachren, A.M., Liao, K.: A visualization system for space-time and multivariate patterns (VIS-STAMP). IEEE Trans. Vis. Comput. Graph. **12**(6), 1461–1474 (2006)
15. Harrower, M., Brewer, C.A.: Colorbrewer.org: an online tool for selecting colour schemes for maps. Cartogr. J. **40**(1), 27–37 (2003)
16. Hoeber, O., Wilson, G., Harding, S., Enguehard, R., Devillers, R.: Exploring geo-temporal differences using GTdiff. In: IEEE Symposium on Pacific Visualization, pp. 139–146 (2011)
17. Im, J.F., McGuffin, M.J., Leung, R.: GPLOM: the generalized plot matrix for visualizing multidimensional multivariate data. IEEE Trans. Vis. Comput. Graph. **19**(12), 2606–2614 (2013)
18. Jern, M., Franzen, J.: "GeoAnalytics": exploring spatio-temporal and multivariate data. In: Proceedings of the Tenth International Conference on Information Visualisation, pp. 25–31 (2006)
19. Kehrer, J., Piringer, H., Berger, W., Groller, M.E.: A model for structure-based comparison of many categories in small-multiple displays. IEEE Trans. Vis. Comput. Graph. **19**(12), 2287–2296 (2013)
20. von Landesberger, T., Brodkorb, F., Roskosch, P., Andrienko, N., Andrienko, G., Kerren, A.: Mobilitygraphs: visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. IEEE Trans. Vis. Comput. Graph. **22**(1), 11–20 (2016)
21. Liu, X., Hu, Y., North, S., Shen, H.W.: Correlatedmultiples: spatially coherent small multiples with constrained multi-dimensional scaling. In: Computer Graphics Forum (2015)
22. Meulemans, W., Dykes, J., Slingsby, A., Turkay, C., Wood, J.: Small multiples with gaps. IEEE Trans. Vis. Comput. Graph. **23**(1), 381–390 (2017)
23. National Center For Health Statistics. http://www.cdc.gov/nchs/ Accessed Jan 2015
24. OECD data. https://data.oecd.org/. Accessed Nov 2017
25. Papadopoulos, C., Petkov, K., Kaufman, A., Mueller, K.: The Reality Deck—an immersive gigapixel display. IEEE Comput. Graph. Appl. **35**(1), 33–45 (2015)
26. Rao, R., Card, S.K.: The Table Lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 318–322 (1994)
27. Roth, R.: An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. IEEE Trans. Vis. Comput. Graph. **19**(12), 2356–2365 (2013)
28. Sadana, R., Major, T., Dove, A., Stasko, J.: OnSet: a visualization technique for large-scale binary set data. IEEE Trans. Vis. Comput. Graph. **20**(12), 1993–2002 (2014)
29. Slingsby, A., Dykes, J., Wood, J.: Exploring uncertainty in geodemographics with interactive graphics. IEEE Trans. Vis. Comput. Graph. **17**(12), 2545–2554 (2011)
30. Speckmann, B., Verbeek, K.: Necklace maps. IEEE Trans. Vis. Comput. Graph. **16**(6), 881–889 (2010)
31. Swedberg, B., Robinson, A.C., Hardisty, F., Peuquet, D.J.: Geovisualization of spatio-temporal events in STempo. In: GeoVisual Analytics: Interactivity, Dynamics, and Scale Workshop at GIScience Conference (2014)
32. Tennekes, M., de Jonge, E.: Tree colors: color schemes for tree-structured data. IEEE Trans. Vis. Comput. Graph. **20**(12), 2072–2081 (2014)
33. Tominski, C., Schulz, H.J.: The great wall of space-time. In: Proceedings of the Workshop on Vision, Modeling and Visualization, pp. 199–206 (2012)
34. United States Census Bureau. http://www.census.gov/. Accessed Jan 2015

35. U.S. Energy Information Administration: State Energy Data System. http://www.eia.gov/state/seds/. Accessed Nov 2016
36. Ware, C.: Information Visualization: Perception for Design, 3rd edn. Morgan Kaufmann Publishers, Los Altos (2012)
37. Wickham, H., Hofmann, H.: Product plots. IEEE Trans. Vis. Comput. Graph. **17**(12), 2223–2230 (2011)
38. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Publishers, Los Altos (2005)
39. Xiaoru, Y., Peihong, G., He, X., Hong, Z., Huamin, Q.: Scattering points in parallel coordinates. IEEE Trans. Vis. Comput. Graph. **15**(6), 1001–1008 (2009)



Ji Hwan Park is a Ph.D. candidate in the Computer Science department, Stony Brook University. He received his B.S. degree in computer engineering from Hongik University, Seoul, South Korea, in 2004. His research interests include visual analytics, information visualization, scientific visualization, and human computer interaction.



Saad Nadeem is a Ph.D. candidate in the Computer Science department, Stony Brook University. He received his B.Sc. Honors in Computer Science and Mathematics from School of Science and Engineering, Lahore University of Management Sciences, Pakistan. His research interests include computer vision, computer graphics, and visualization.

Arie E. Kaufman is a Distinguished Professor and Chairman of the Computer Science Department, Director of the Center of Visual Computing (CVC), and Chief Scientist of the Center of Excellence in Wireless and Information Technology (CEWIT) at Stony Brook University. He has conducted research for over 40 years in visualization and graphics and their applications, has published more than 300 refereed papers, has presented more than 20 invited keynote talks, has been awarded/filed more than 40 patents, and has been PI/co-PI on more than 100 grants. He was the founding Editor-in-Chief of IEEE Transaction on Visualization and Computer Graphics (TVCG), 1995–1998. He is a Fellow of IEEE, a Fellow of ACM, the recipient of the IEEE Visualization Career Award (2005), and was inducted into the Long Island Technology Hall of Fame (2013). He received his Ph.D. in Computer Science from Ben-Gurion University, Israel, in 1977.