Computers & Graphics 82 (2019) 1–12

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Automatic generation of puzzle tile maps for spatial-temporal data visualization

Shih-Syun Lin[a,*], Juo-Yu Yang[a], Huang-Sin Syu[b], Chao-Hung Lin[b], Tun-Wen Pai[c,a]

[a] *Department of Computer Science and Engineering, National Taiwan Ocean University, No.2, Peining Rd., Jhongjheng District, Keelung City 20224, Taiwan*
[b] *Department of Geomatics, National Cheng-Kung University, No.1, University Road, Tainan City 70101, Taiwan*
[c] *Department of Computer Science and Information Engineering, National Taipei University of Technology, No.1, Sec. 3, Zhongxiao E. Rd., Taipei City 10608, Taiwan*

A R T I C L E I N F O

Article history:
Received 20 August 2018
Revised 31 March 2019
Accepted 3 May 2019
Available online 15 May 2019

Keywords:
Tile map
Choropleth map
Tile layout optimization

A B S T R A C T

Tile maps are a visualization tool to display geographic data without the accurate representation of geographic boundaries. Each region in a tile map is represented as a tile of identical shape and size. The tiles are fit in a regular grid at positions that approximate their geographic positions such that large regions do not dominate the map visualization, and information in small regions can be enhanced. In this study, the automatic generation of a tile map composed of puzzle tiles is proposed for spatial-temporal data visualization. A puzzle tile is an extension of a standard square tile. A sequence of connected and directional pieces in a puzzle tile is used to represent time-varying quantities in a geographic region. To generate a puzzle tile map, the proposed method includes algorithms for optimizing district-to-tile mapping according to not only geographic positions but also region orientations and for placing puzzle pieces in a tile. The proposed puzzle tile map can serve as a choropleth map in which the ordered pieces in a tile are shaded in proportion to the measurements of a statistical time variable, such as a time sequence of fertility rates, air pollution (PM2.5), or transfer of residential property, being displayed on a 2D map. Experimental demonstrations of various cases show that the proposed methods for district-to-tile mapping optimization and puzzle generation are feasible for automatic puzzle tile map generation. User studies show the capabilities of the puzzle tile map in terms of usability, readability, and comparability of spatial-temporal data visualization.

1. Introduction

Maps can be classified into two categories, namely, *topographic maps* and *thematic maps*. Topographic maps accurately display the location of an object in a space, and thematic maps focus on combining specific themes with maps [1]. With the development of information visualization techniques, thematic maps have elicited increasing attention in visualization and computer graphics. Tile maps are important tools in thematic cartography, and their characteristics are distinct from those of well-known tools, such as choropleths, cartograms, and symbol maps [2]. The tiles in a tile map are placed in a grid at positions that approximate their geographic positions so that each tile has an identical shape and size. Tile maps provide an easy means to visualize the level of variability within a region and how a measurement varies across a geographic area. Furthermore, complex data can be shown on tiles in a consistent format that allows users to intuitively and easily compare data among tiles, as shown in Fig. 1. Compared with choropleths, cartograms, and symbol maps, tile maps have the clear and simple appearances.

Large amounts of time-series data exist in our daily life, and these data are generally visualized by diagrams, such as line charts and heat maps [3]. Most of these time-series data contain geographic information, which is referred to as spatial-temporal data. Spatial-temporal data cannot be exposed by diagrams. Geographic information plays an important role in this type of data. Thus, how to intuitively and clearly display spatial-temporal data is a challenge that requires urgent attention. 2D choropleths with time sliders [4] and 3D choropleths [5,6] are common means to visualize spatial-temporal data with the aid of a geographical map. Time-series data are displayed in a geographical map with the aid of time sliders to manually display time-varying quantities or 3D graphical widgets and to represent the profiles of time-varying quantities in 3D space. In their study on information availability in 2D and 3D displays [7], Smallman et al. concluded that the readability of 2D displays generally outperforms that of 3D

* Corresponding author.
E-mail address: linss@mail.ntou.edu.tw (S.-S. Lin).

https://doi.org/10.1016/j.cag.2019.05.002
0097-8493/

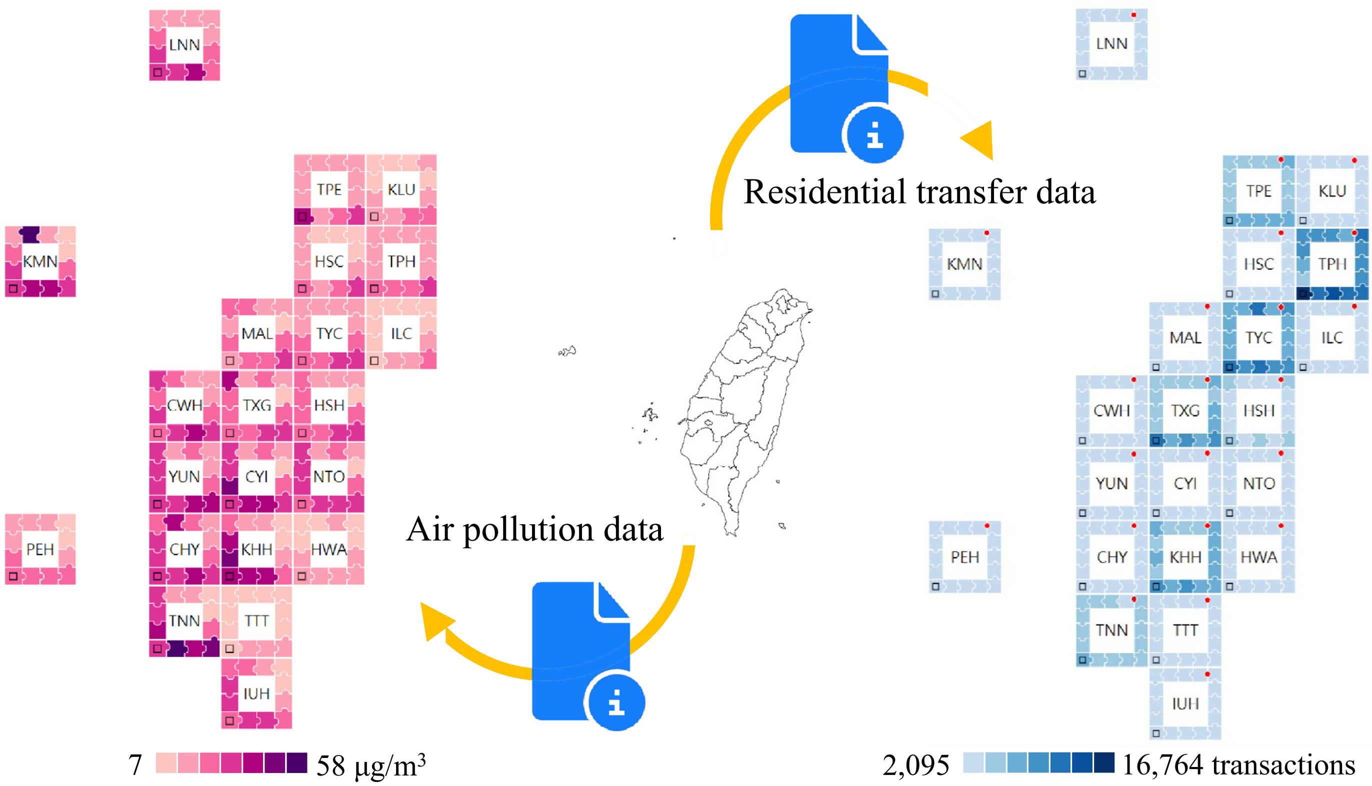

Fig. 1. Results of puzzle tile maps (Taiwan). Middle: geographic map; left and right: puzzle tile maps for air pollution and residential transfer data, respectively.

displays because of the simplicity of 2D displays and the occlusion problem in 3D displays. Therefore, in this study, 2D graphical widgets, that is, 2D puzzle pieces, are used in tile maps instead of time sliders or 3D widgets in a geographic map. A tile with puzzle pieces in a map is called a puzzle tile map, which can show spatial-temporal data in a single and static 2D diagram.

The proposed method for generating a puzzle tile map includes algorithms to optimize district-to-tile mapping according to geographic positions and orientations and to place puzzle pieces in a tile for time-series data visualization. The resulting puzzle tiles can display spatial-temporal data in a 2D map, reveal time-varying quantities in a sequence of puzzle pieces, and facilitate comparisons of time-varying quantities among puzzle tiles. Compared with related tile map generation study [2,8,9], this study proposes the use of puzzle pieces to display time-series data of a region and establishes an algorithm to optimize district-to-tile mapping. The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the overview of puzzle tile map. Section 4 describes the tile map generation. Section 5 demonstrates the puzzle representation. Section 6 discusses the experimental results and user studies. Section 7 presents the conclusions and future work.

2. Related work

Several spatial-temporal data visualization methods have been proposed, and they can be classified into three categories, namely, *2D static*, *2D dynamic*, and *3D visualization methods*, on the basis of display dimensionality and data representation. 2D static methods were developed for temporal data visualization. For example, the heat map proposed by Suematsu et al. [3] visualizes multiple time-series data by using multiple color belts. A belt is filled with colors to represent time-sequence quantities. The data vases proposed by Thakur and Rhyne [10] make up another approach that yields a pictorial display for multiple time-series data. The profile of a single time-sequence quantity is represented by a data vase and plotted along a vertical temporal axis with time increasing from bottom to top. All data vases displayed in a rectangular canvas are arranged according to the order of their geographical positions. These 2D static methods can achieve simple and compact data visualizations, but they are limited to temporal information visualization. When data contain geographic information, 2D static methods cannot easily display the geographical relation and distribution.

In 2D dynamic techniques, the use of choropleths with animation is a common means to visualize time-series data in a geographical map [4]. The method employs color schemes to represent the data for each district or region at a time slice and changes colors over time. This method can visualize spatial and temporal data, but providing an overview of time-series data is difficult. Compared with 2D methods [3,4], 3D visualization methods [5,11,12] have more spaces to display spatial-temporal data. Most 3D methods display time-series data on a geographic map, which allows users to overview and explore data. A classic example is the space-time cube (STC) [11,12] which was proposed to visualize spatial movements, such as traffic incidents and earthquakes, in a geographical space and detect spatial-temporal patterns in event occurrences. Although the STC approach is applicable for visualizing event-based data, displaying general spatial-temporal data in a cube may make it difficult for users to observe detailed geographic and temporal information. To improve STC visualization, researchers proposed 3D glyphs [5] that extend the representation of 2D data vases [10] to 3D and combine the 3D data vases with maps. Specifically, the profiles of time-varying quantities are plotted as 3D glyphs on a geographic map to reveal the spatial-temporal distributions of data. However, reduced readability is a common problem in 3D visualization because of object occlusion even when interactive viewing transformation is provided [7].

The tile maps, namely, *2D visualization*, still face a huge demand to present spatial-temporal data due to the readability problem in 3D visualization. Many interesting websites discuss tile map construction [13–15]. However, only a few methods have been proposed for automatic tile map generation [2,8,9,16–18]. Cano et al. [16] proposed a specific type of cartogram, namely, *a mosaic map*, is to represent each region by using multiple tiles, and the types of regular tiles are square and hexagon. The entire shape of the mosaic cartogram is similar to the original map. Wu et al. [17] proposed Mobiseg system that supported the exploration of peoples movement activities to segment the urban area into regions sharing similar activity patterns. Hexagonal tessellations with stacked elliptical activity glyphs are used to visualize spatial-temporal data. Chen et al. [18] proposed a visual analytics method to analyze information diffusion processes in social media, and the D-Map+ they proposed is generated by hexagonal tessellations. Each node represents one person or a group of people with similar behaviors. The researches in [17,18] applied the mosaic map for visualization since their data is that a district consists of multiple nodes. However, the data in our study is that the amount of each district is uniform, so that a district consists of only one node. For the reason, the one-to-one mapping method is more suitable for tile map generation in this study. Eppstein et al. [9] proposed a styled map, namely, *a grid map*, is generated with the regular cells and in line with the data type of this study. The map can allow users to quickly find a district in a grid. Meulemans et al. [8] extended the study of Eppstein et al. [9] to design several criteria that allow users to create a tile map by optimizing an objective function. However, these two different methods lack abilities for handling empty cells. Therefore, McNeill and Hale [2] proposed an approach to generate tile maps considering the region-to-tile assignments with a specified tile shape, such as square, hexagon, and triangle. Although this method is suitable for tile map generation, but only semi-optimal one-to-one mapping is obtained. Furthermore, the method in [2] is designed for contiguous geographical data. Although the method in [8] supports for non-contiguous (or contiguous) geographical data, the generated cells of a layout possibly have a slight defect, which is inappropriately non-connected (or connected) relation in contrast to original geographical data. On the other hand, handling the boundaries of a map for much proper map shape is another issue in this study. Auber et al. [19] proposed an algorithm for creating treemap-like representations of trees that have irregular shapes. Tong et al. [20] proposed a cartographic treemap to integrate a modified representation of the UK based on the geospatial information of CCG regions combined with a modified treemap to present the multivariate NHS data. Although the layouts in these two studies belong to the category of tile maps that have similar shapes with the original maps, the methods are not suitable for this study. Because of the demand of comparing and observing time-series data in tiles, the uniform and neat layout is necessary. In this study, a puzzle tile map is generated, in which optimized district-to-tile mapping is obtained according to the geographic positions and orientations. The proposed method also allows tile map generation for non-contiguous geographical data, such as geographical data containing multiple islands.

3. The overview of puzzle tile map

The tile map generation, which is illustrated in Fig. 2, consists of four main steps, namely, *data preprocessing*, *mosaic filtering*, *tile position correction*, and *mapping optimization*. It represents spatial information. The input to the system is spatial-temporal data with a corresponding local map. In the first step, preprocessing is performed to simplify the input geodata and separate each connected object from the input map. This step accelerates the map generation and deals with maps containing multiple connected objects, such as countries containing multiple islands. In the second step, initial tiles are generated through mosaic filtering, in which the outer boundaries of initial tiles approximate the geographic boundaries because the initial tiles from different connected objects may have different gridding systems in different grid sizes. Therefore, in the third step, the tiles from different connected objects are rearranged in such a manner that the tiles have an identical size and correct positions. In the last step, an objective function is utilized to optimize the mapping between districts/cities and tiles. Puzzle pieces in each tile are generated and placed to represent the input time-varying data. Tile map generation is described in Section 4, followed by puzzle representation in Section 5.

4. Tile map generation

4.1. Data preprocessing

The input geodata format is GeoJSON, which supports the following geometry types: points, line segments, polygons, and collections of features. Text-based GeoJSON may lead to poor performance because of the duplicated region boundaries. To consider system performance, the input GeoJSON data is simplified. Buttenfield [21] identified the line simplification problem as a part of linear features, and there are several classic algorithms for the manipulations of lines, such as Peucker's study [22]. In this study, we employ Turf.js [23] that performs boundary simplification using the Ramer–Douglas–Peucker algorithm [24].

District connectivity is determined in preprocessing to automatically process geographical data containing multiple components, such as countries with multiple islands. A region growing strategy is adopted to check the connectivity of districts. Then, the input geographical data are separated into several connected regions. For example, the US map is divided into three connected regions, that is, 48 states in North America, Alaska, and Hawaii.

4.2. Mosaic filtering

Mosaic filtering is an image processing method to segment an image into several squares or polygons and draw the divided squares and polygons by colors, such that the appearance approximates the original image [25]. The purpose of mosaic filtering is to determine initial tiles for the input geographic map, in which the number of tiles is equal to that of desired districts/cities, and the outer boundaries of the initial tiles approximate the original geographical map. The standard mosaicking method is adopted to determine the initial tiles. Any advantaged mosaicking methods can be adopted and integrated into the proposed system. Before mosaicking, the map image is transformed to a binary image. The land regions are set to black, and the others are set to white. The mosaic filter is then applied to the binary image to generate initial tiles of the same size. If the number of black pixels is more than that of white pixels in each tile, then this tile is set to black and vice versa. The initial tiles generated using mosaic filtering can force the layout to be as similar as possible to the original geographic map. A large filter kernel results in a few tiles, and a small filter kernel generates many tiles. An iterative tuning strategy is adopted for filter size to obtain the desired number of tiles. Specifically, the mosaic filter size is gradually modified such that the number of obtained tiles matches the required number of districts/cities.

Although mosaic filtering allows for the generation of tiles with a shape that fits the original geographic map well, the tile positions may not match that of districts/cities in the mapping process because of the inconsistent area between tiles and districts/cities. The method proposed by McNeill and Hale [2] is adopted to alleviate this problem. With this approach, the boundaries of the input map are deformed based on the density of districts/cities. In the

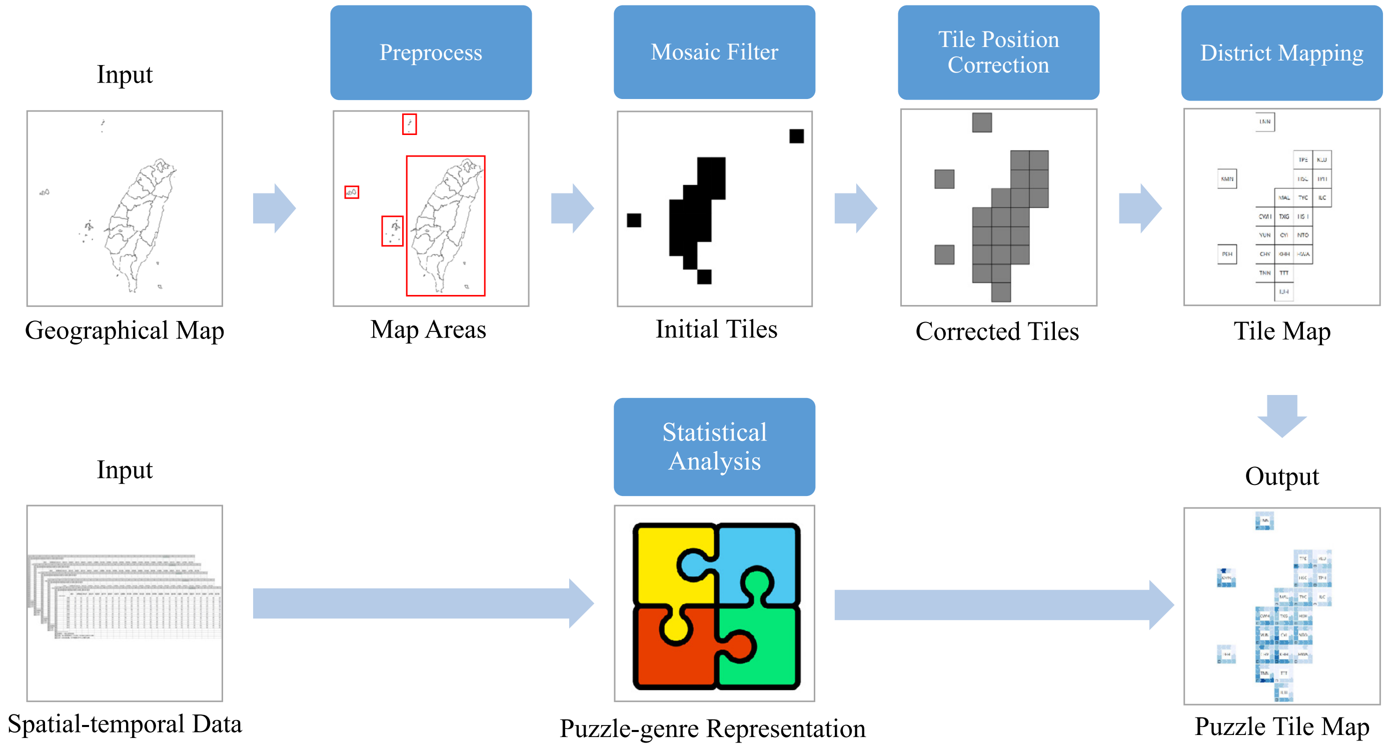

Fig. 2. System workflow. The system consists of four steps, namely, *data preprocessing*, *mosaic filtering*, *tile position correction*, and *mapping optimization* for generating a tile map. The puzzle tile map is composed of a tile map and puzzle pieces.

other words, the map region containing dense districts is enlarged, and the region containing sparse districts is shrunk. This algorithm is briefly described as follows. First, the centroid of each district is moved so that the distances from the centroid to its neighboring districts are the same while the relative orientations of neighboring districts are preserved. This process is formulated as

$$c_i^t = \frac{1}{|N(i)|} \sum_{j \in N(i)} \left(c_j^{t-1} + s \cdot \frac{g_i^{t-1} - c_j^{t-1}}{\|g_i^{t-1} - c_j^{t-1}\|} \right), \tag{1}$$

where $N(i)$ and c_i^t represent the set of neighbors and the centroid of district i in the t-th iteration, respectively; g_i^t denotes the mean of the centroids of the districts in $N(i)$ in the tth iteration; and s is the estimated grid size that is set to the square root of the current area divided by the number of districts. In each iteration, the position of the centroid c_i^t is updated by its neighbors $(c_j^{t-1}, j \in N(i))$ and the mean of the centroids g_i^{t-1}. Then, the boundaries of the geographic map, i.e., the boundary points, are pushed or pulled toward the new centroids, by using the equation

$$b_i^t = \frac{\sum_{l \in M(i,k)} c_l^t}{k} + \sqrt{\frac{s}{\|c_l^t - b_i^{t-1}\|}}\,(c_l^t - b_i^{t-1}), \tag{2}$$

where $M(i, k)$ is the index set of the k nearest centroids c_l^t of the boundary point b_i^{t-1}. Initially, b_i^0 is set to the original boundary point of the map. The default value of k is set to 3.

4.3. Tile position correction

Two problems may arise after the initial tile generation using mosaic filtering. First, mosaic filtering is performed with the shape of the map; however, the method does not consider the case of multiple connected regions. Second, to generate the desired number of tiles, the initial tiles for each connected region are generally generated with different grid sizes depending on the region of filter size. Therefore, the obtained tiles may have different sizes in different gridding systems.

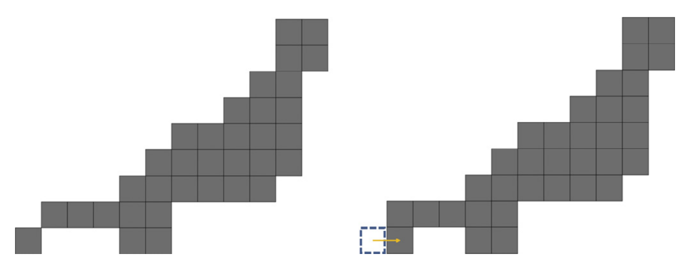

Fig. 3. Example of connected region. The tile marked by a blue dotted line is shifted to the nearest vacancy such that the region becomes connected, in terms of first-order neighborhood. The tested datum is the map of Honshu, Japan.

To ensure that all tiles have a correct connection, the first-order neighborhood, i.e., 4-connectivity, is defined as a contiguous area, and the others are defined as non-contiguous areas. If a tile belongs to a region, then this tile will be connected with that region based on the first-order neighborhood. Otherwise, as illustrated in Fig. 3, the position of that tile is rearranged to the nearest vacancy that connects with that region.

Next step, we propose an iterative strategy for correcting orientation and economizing the space of the layout. The region with the most number of districts in a geographical map is defined as the main region and denoted as $\mathbf{B} = \{b_1, b_{n_b}\}$, where n_b represents the number of districts; and the others are defined as non-main regions and a set $\mathbf{U}^r = \{u_1^r, u_{n_r}^r\}$ is created for rth non-main region in a geographical map, where n_r represents the number of districts. In accordance with a geographical map, the region with the most number of tiles in an initial tile map is defined as the main region and denoted as $\mathbf{W} = \{w_1, w_{n_b}\}$, and a non-main region is denoted as $\mathbf{P}^r = \{p_1^r, p_{n_r}^r\}$.

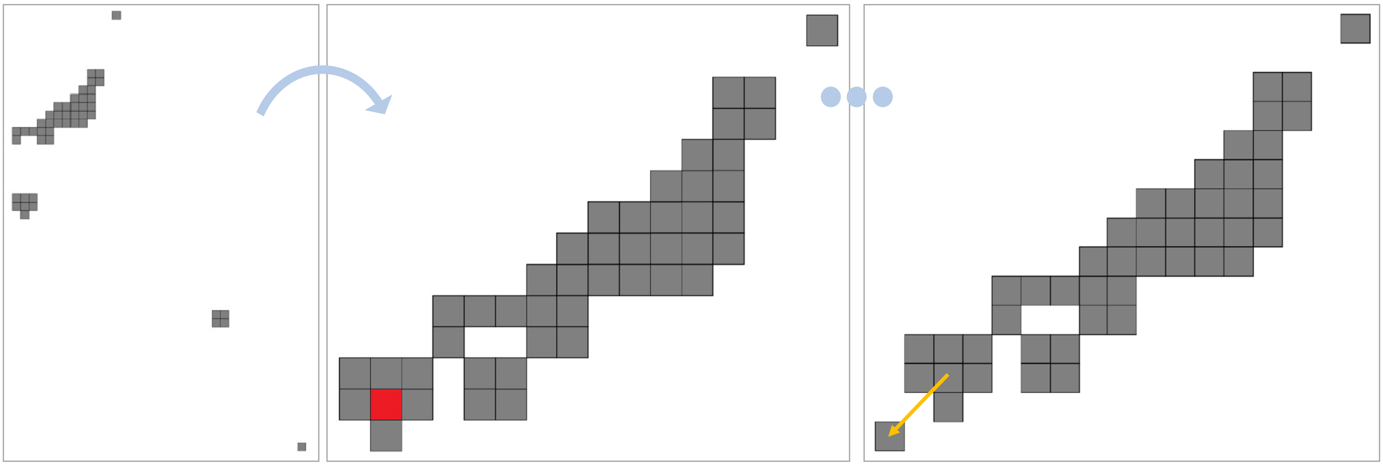

Fig. 4. Corrected result of tile position. Left: mosaic filtering result. The tiles have wrong placement. Middle: a tile map after the first shifting process, and the red one is overlapped on the other tile. Right: adjusted result. The tested datum is the map of Japan.

$\mathbf{B}^{'}$ and $\mathbf{U}^{r'}$ are the centroids of the main region $\mathbf{B}$ and rth non-main region $\mathbf{U}^{r}$ in a geographical map, respectively. Then, $\mathbf{W}^{'}$ and $\mathbf{P}^{r'}$ are the centroids of the main region $\mathbf{W}$ and a non-main region $\mathbf{P}^{r}$ in an initial tile map, respectively. To shift a non-main region of an initial tile map into a suitable position according to the relative orientation of a geographical map, the equation can be formulated as follows:

$$\hat{\mathbf{P}}^{r} = \mathbf{W}^{'} + \frac{\mathbf{U}^{r'} - \mathbf{B}^{'}}{h}, \tag{3}$$

where h is the tile size of the main region and $\hat{\mathbf{P}}^{r}$ is the suitable position in the tile map. Although the suitable position of the tile map is similar to its correct position of the geographical map, how to economize the space of the layout is a problem. Then, we have to make the gap between the main and non-main regions as close as possible. The two nearest tiles are $\hat{p}_i^r$ and w_j in non-main region $\mathbf{P}^{r}$ and in main region $\mathbf{W}$, respectively. The equation can be formulated as follows:

$$\underset{i \in n_r, j \in n_b}{\arg\min} \| \hat{p}_i^r - w_j \|^2. \tag{4}$$

The shifting operation is formulated as follows:

$$\tilde{p} = w_j + \frac{\hat{p}_i^r - w_j}{\| \hat{p}_i^r - w_j \|} \cdot k \cdot h, \tag{5}$$

where k is the number of iterations. $\tilde{p}$ is the corrected position of the tile in the non-main region and it is updated iteratively until no overlap exists in the entire tile map. The tile position correction result is shown in Fig. 4.

4.4. Mapping between districts and tiles

Given n districts and n tiles, the problem to be addressed is how to search for the optimal one-to-one mapping between the districts and tiles. The goal of this mapping is to minimize the distances between tile positions and their corresponding geographic positions. Each distance between city i and tile j is utilized to construct the set $\mathbf{Q} = \{q_{ij}\}_{n \times n}$. The set $\mathbf{X} = \{x_1, x_n\}$ is the initial district position assigned to a tile. Initially, the adjacency set $\mathbf{D} = \{d_{ij}\}_{n \times n}$ that describes the adjacency of districts in the geographic map is constructed based on the second-order neighborhood. This adjacency array is presented as

$$\mathbf{D} = \{d_{ij}\}_{n \times n}, \; d_{ij} \in \{0, \ldots, 9\}, \tag{6}$$

where each entry in set $\mathbf{D}$ is represented by an index from 0 to 9. With the definition of directions, denoted as $Dir(d_{ij})$, in Eq. (7), the directions between districts in a tile can be defined. $Dir(d_{ij})$ is formulated as

$$Dir(d_{ij}) = [\nwarrow, \uparrow, \nearrow, \longleftarrow, \vec{0}, \longrightarrow, \swarrow, \downarrow, \searrow], \tag{7}$$

where " $\vec{0}$ " represents the zero vector, " $\nwarrow$ " represents the vector $(-1, -1)$, " $\uparrow$ " represents the vector $(0, -1)$, and so on.

To obtain district-to-tile mapping in the tile, an objective function that consists of two energy terms, namely, *distance* and *orientation*, is defined. The distance term is formulated by measuring the distance between the tile positions and their corresponding geographic positions. The orientation term is defined as the difference in arbitrary directions in two districts in the geographic and tile maps. By combining these two energy terms, the objective function $e(i, j)$ for districts i and tile j is defined as

$$e(i, j) = \alpha \times q(i, j) + (1 - \alpha) \times \| Dir(d_{ij}) - (x_j - x_i) \|, \tag{8}$$

where $q(i, j)$ represents the distance between district i of the geographic map and tile j. $Dir(d_{ij})$ and $(x_j - x_i)$ in the second term represent the vector from the centroid of district i to that of district j in the tile and geographic map, respectively. The defined objective function considers not only the positions but also the orientations of districts. Thus, appropriate one-to-one mapping between the city and the corresponding tile can be obtained. α is the parameter used to balance the contributions of the distance and orientation terms. A tradeoff exists between the preservation of district position and orientation in the mapping optimization. To consider both, α is set to 0.5 as the default value in the system.

The minimization of the objective function can be formulated as

$$\min \quad \sum_i^n \sum_j^n e(i, j) \cdot v_{ij}^{'}, \tag{9}$$

$$\text{subject to} \quad \sum_i^n v_{ij}^{'} = 1, \; \sum_j^n v_{ij}^{'} = 1, \tag{10}$$

where $v_{ij}^{'}$ denotes the assignment of district i to tile j subject to the hard constraints in Eq. (10). This bipartite problem is solved by using the Hungarian algorithm [26], which provides an efficient means to determine the optimal solution.

5. Puzzle representation of tile

To visualize time-series data on tile maps, an aesthetic pattern, namely, *a puzzle*, is adopted to represent time-varying data. The input time-varying data can be sliced according to the users demand,

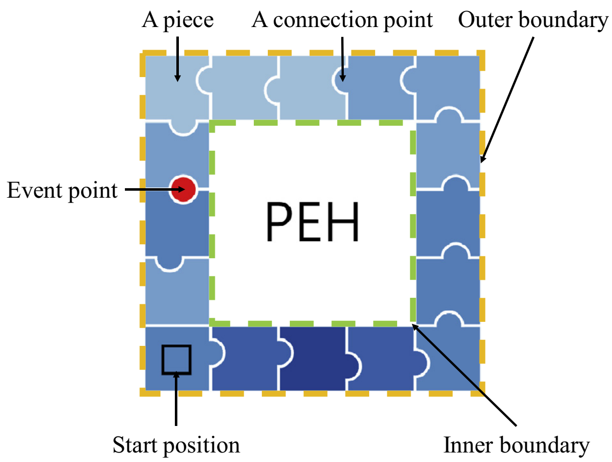

Fig. 5. Puzzle tile example.

and the appropriate time slices are 8, 12, 16, and 20. A puzzle consists of five characteristics, including tile position, puzzle pieces, piece color, puzzle connection point, and event point, as shown in Fig. 5. The proposed map based on puzzle tiles can visualize the information of geographical position, time-varying quantities, and change in quantities. Borgo et al. [27] mentioned glyph design criteria according to the study [28], which is a glyph-based sorting framework. For good observation, the puzzle glyph design is based on visual orderability, searchability, attention-balance, and focus-plus-context design criteria. In the following text, each detail of the puzzle tile is introduced. The text also shows how the proposed visualization approach can be used to observe spatial-temporal data.

District name. The district name is placed at the center of a puzzle, and it represents the district to which the puzzle belongs to in the geographical map. It can display the geographic position of this district in a global view.

Puzzle pieces. A puzzle is a square pattern surrounded by puzzle pieces. For consistency, observation, and comparison, the default piece placement is in counterclockwise order, and the default start position of the piece sequence, denoted by black square, is at the bottom left corner of the square. The starting location of each puzzle can be changed without damaging its representation because of the circular arrangement of the puzzle pieces. In addition, the pieces are shaded in proportion to the measurements of a statistical time variable. The task-driven color coding proposed by Tominski et al. [29] and the tool for selecting a color scheme [30] are adopted to ensure that the colors are suitable for various tasks.

Puzzle connection points. The puzzle connection points make up a particular pattern. The default connection points represent by round shape, which can represent not only the order of time-series data but also the change (e.g., increasing or decreasing) between the quantities of two adjacent pieces, as shown in Fig. 6. There is no limitation to puzzle form. Fig. 7 represents other types of puzzle form, such as triangle and square. The information of quantity change rate is effective for numeric data visualization designed with color schemes. A color scheme represents the visualized intensity of numeric data. In their study [30], Harrower and Brewer mentioned that if a color scheme is used to display a dataset with a wide dynamic range of data quantity, then visualization may be perceptually unreliable. Therefore, a default sequential scheme set in our study is 7 color classes. However, the perceptual reliability of only 7 color classes may cause a problem since it is difficult to display the variations between adjacent pieces. For example, two

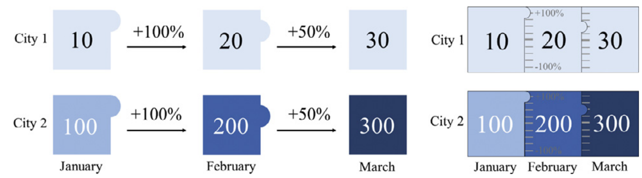

Fig. 6. A bad case of visualization using a color scheme for a dataset with high dynamic range on data quantities. The increasing trend in City 1 is imperceptible from the piece colors even the quantities are increased by 100% and 50%. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

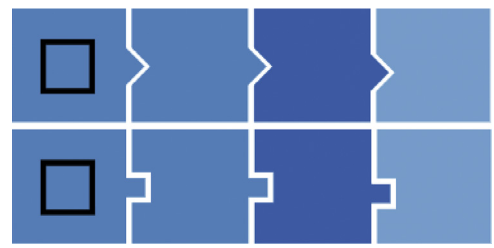

Fig. 7. Other types of puzzle form.

cities with different populations are displayed using puzzle pieces in Fig. 6. The numbers in the pieces represent the numbers of patients with a cold in specific months. The colors of the pieces are normalized according to the maximal quantity. The number of patients with a cold shows an increase in this example. However, in the visualization, observing the increasing trend in City 1 based on the displayed colors is difficult because the colors in three puzzle pieces are the same. Even the quantities in the second and third pieces are increased by 100% and 50%, respectively, compared with the quantities in the first and second pieces.

In this study, this problem is addressed by the piece connection pattern. If the data quantity in the current piece is greater (smaller) than that in the previous piece, then the piece connection pattern will be located close to the inner (outer) boundary. The locations of piece connection patterns allow users to observe the change rate of data quantities between two adjacent puzzle pieces.

Event point. When an event occurs in the displayed data sequence, an event point pattern designed as a red dot is placed in the puzzle tile to represent the event time. With the event point, the influence of that event on the time-series data can be visualized. A dot with a different hue is used in the puzzle tile to highlight the event and its occurrence time.

6. Experimental results and user study

The proposed method is tested on a PC with an Intel Core i7 (1.7GHz) processor and 8 GB of memory. The code is written in JavaScript, and the one-to-one mapping problem is solved with the Hungarian algorithm. The computational time for puzzle tile map generation is 19.19 milliseconds. In addition, we provide several sub-panels to show the details of the selected tile (i.e., a summary statistic and focus tile).

A screenshot of the interface is shown in Fig. 8 and it is an overview-plus-detail structure. Five panels in our interface include puzzle tile map in the blue box, focus district in the yellow box, the summary of the focus district in the green box, the legend in the red box, and simple descriptions of the data in the orange box, respectively. The sub-window of puzzle tile map shows the overview of the data, and if users select one district of puzzle tile map, the sub-window of focus district will show the details and context, and it further demonstrates the change of data quantity in

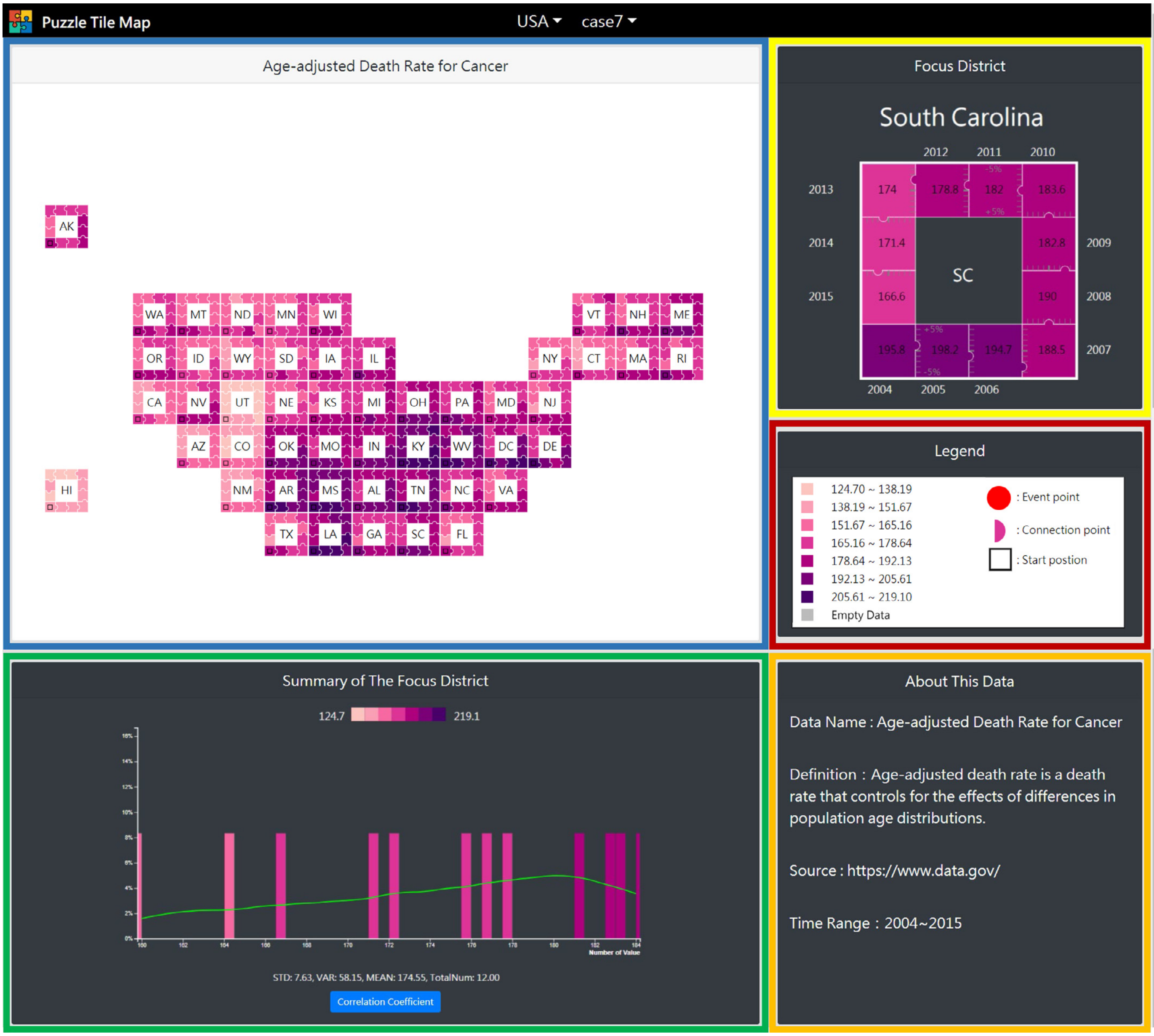

Fig. 8. Interface screenshot. Overview of puzzle tile map, focus district, the summary of the focus district, the legend, and simple introduction to the data are shown in the blue, yellow, green, red, and orange boxes, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

SC. Although the difference between colors is small, users can still observe the varying by puzzle connection points. The summary of the focus district shows the distribution of the data and the correlation coefficients between each pair of districts. It's worth noting that to make the data between different districts to easily compare at the same time, our interface supports a highlight mode, as shown in Fig. 9.

A demonstration of deforming the shape from an original geographic map is given in Section 4.2. Although map deformation decreases the similarity between the initial tiles and the original geographical map, the deformed result can improve the accuracy of mapping between districts and tiles. The example shown in Fig. 10 indicates that the shape of the generated tile map with shape deformation (figure at the right-top part) is better than that without shape deformation (figure at the right-bottom part).

Comparisons of district-to-tile mapping results using different parameter settings of α in the objective function (Eq. (8)) are shown in Fig. 11. For the geographic map of Kaohsiung, KHH (Fig. 11 left), the spatial neighbors of Kaohsiung are Nantou (NTO), Hualien (HWA), Taitung (TTT), Pingtung (IUH), Tainan (TNN), and Chiayi City (CHY). The weights of these two parameters would both effect the mapping result. If only considering the orientation parameter, it may lead to a poor layout with good neighbor relationship. When the objective function only adopts the energy term of distance, which is similar to a related method [2], i.e., $\alpha = 1.0$ in the Eq. (8), several cities cannot satisfy relative relations, such as NTO, HWA, and TTT (the red double-headed arrows in Fig. 11 in the middle). Setting α to 0.5 means that the energy terms of distance and orientation are included in the objective function. The orientation constraint is defined based on contiguity. As shown in Fig. 11 in the right, the number of incorrect neighbors is reduced. This example demonstrates that the cities/districts are placed in tiles under the consideration of not only geographic positions but region orientations.

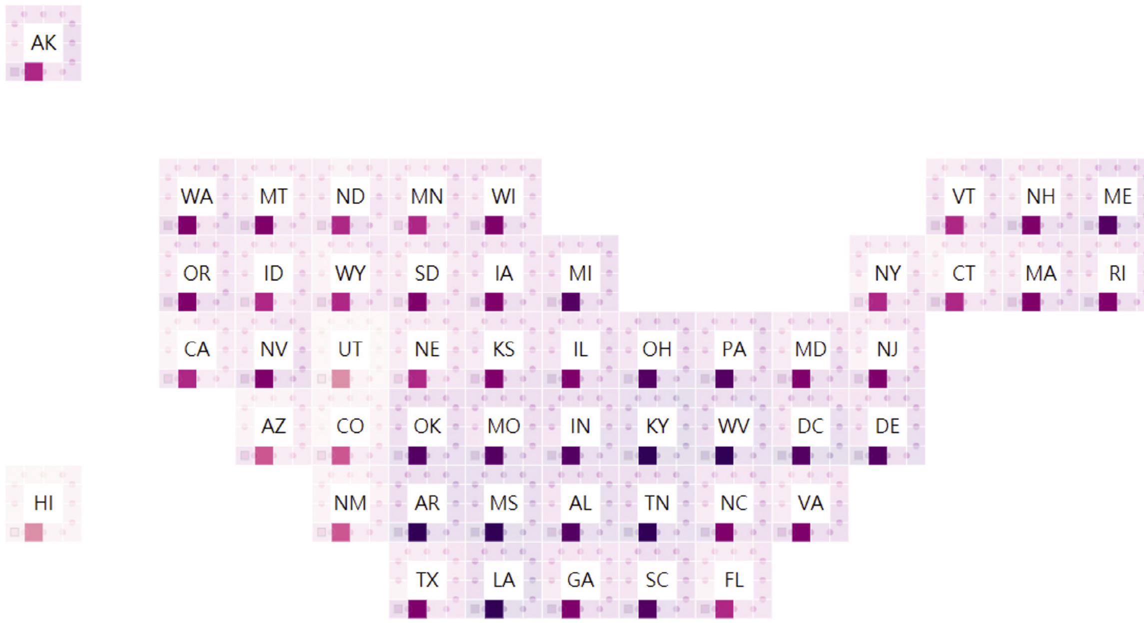

Fig. 9. The highlight mode of our interface. Highlighting the data at the same time makes comparisons between different districts easier.

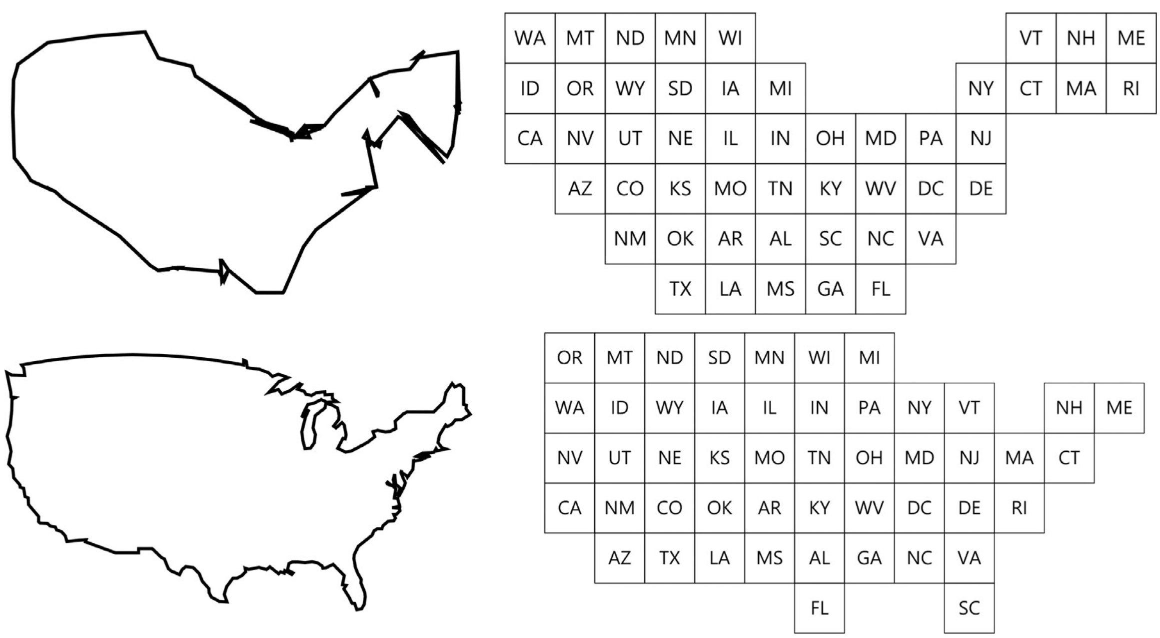

Fig. 10. Comparison of tile map generation with (right-top) and without (right-bottom) map deformation. The deformed map and original geographical map of the United States are shown at left-top and left-bottom parts, respectively.

The simplicity and regularity of puzzle tile maps facilitate the visualization of spatial-temporal data. The puzzle tile map is appropriate and effective for many cases in our daily life. Two examples are demonstrated using the proposed puzzle tile map. The first example is 2016 air pollution data in Taiwan, as shown in Fig. 1 (left). The data contain monthly air pollution (PM2.5) quantities of each city in Taiwan. A dark color means poor air quality. In this case, seasonal and regional air quality can be observed from the puzzle tile map. The air quality in summer is better (i.e., lighter color) than that in winter in many cities, such as TYC and TNN. The air quality in central and southern Taiwan is worse than that in other regions in Taiwan. The second example is the transfer of residential property in Taiwan from the fourth quarter of 2013 to the third quarter of 2016. This example is an event in which the government revised a real estate taxation regulation, that is, consolidated housing and land taxes, which took effect in the first quarter of 2015. This regulation encourages people to purchase houses for self-use rather than investment purpose and caused a negative consequence on the real estate market. In Fig. 1 (right), the red point means the event time. Most of the dark pieces are located in large cities, such as TPH and TYC. In other words, most of the transfers occur in large cities. However, analysis of the influence of the regulation on transfers in large and small cities alike is difficult when only the color scheme. In this situation, the positions of puzzle connection points show increasing and decreasing rates. The transfers of residential properties in all cities decrease obviously, that is, this regulation has been affecting the real estate market since the first quarter of 2016 in Taiwan.

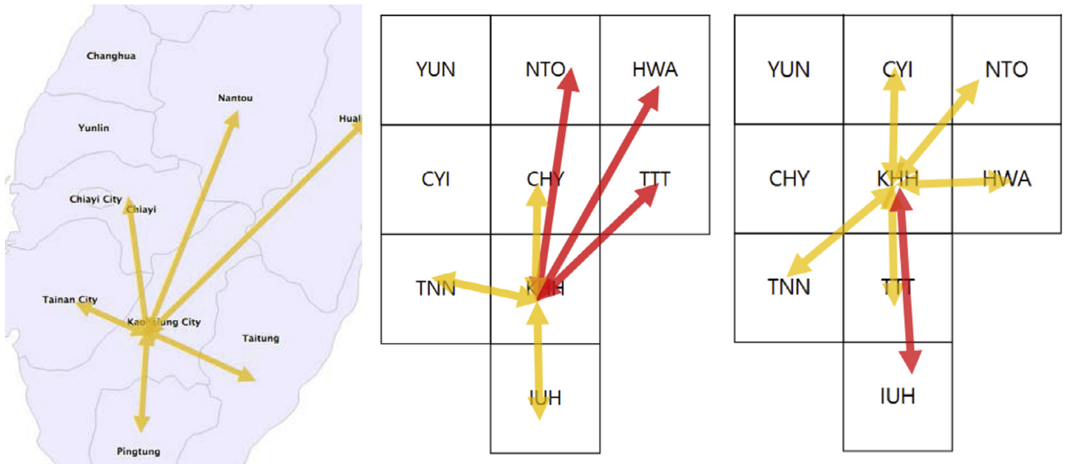

Fig. 11. Comparison of county-to-tile mapping results (Kaohsiung) between the settings of $\alpha = 1.0$ (middle) and $\alpha = 0.5$ (right). The geographic map is shown on the left.

Table 1
Different cases with several factors.

Case\Factor	Temporal information	Spatial information	Difference between two time periods	Data trend	Event time
Total fertility rate	C1F1	C1F2	C1F3	C1F4	C1F5
Air pollution (PM2.5)	C2F1	C2F2	C2F3	C2F4	C2F5
Transfer of residential property	C3F1	C3F2	C3F3	C3F4	C3F5

To demonstrate the feasibility and robustness of the proposed method, various countries with spatial-temporal data are tested. The top and bottom figures in Fig. 12 show the age-adjusted death rate for cancer in the U.S.A. and the population in Japan, respectively.

User study. A user study involving 40 participants aged 18–30 years old is conducted to evaluate the information visualization utility of the generated puzzle tile maps. The participants are shown several generated puzzle tile maps and asked to evaluate each of the tile maps, in terms of temporal information, spatial information, amounts of data, trend of time-varying quantities, and event time. The purpose is to discover if users can obtain spatial-temporal information from the puzzle patterns easier than by using a choropleth map with a time slider. The studies contain three cases, including general fertility rate, air pollution (PM 2.5), and transfer of residential property. All datasets are free to access from an open government data platform (https://data.gov.tw/). We demonstrate the feasibility of our puzzle tile map visualization by conducting three case studies in Taiwan. A discussion is provided about usability, readability, and comparability. Usability is defined as whether this puzzle tiled map is helpful or informative in representing spatial-temporal information, such as time-series data, within a specific region. Readability and comparability are defined as whether this puzzle tile map is more readable than a related method for users to understand exactly what information is shown. Five main factors of this user study are identified with the different cases shown in Table 1. For examples, C1F1 represents the factor of temporal information for the total fertility rate case in our questionnaire, C1F2 denotes the factor of spatial information for the total fertility rate case in our questionnaire, and so on.

A brief introduction to the tasks in our questionnaire is provided. Case 1 involves three questions about evaluating the impact of maternity allowance on the total fertility rate. According to Chinese astrology, children born in the Year of the Dragon, i.e., 2012, are more auspicious than those born in other years. In addition, the fertility rate decreases in the Year of Tiger. We expect that users will come to the conclusion that the effect of the policy of maternity allowance is insignificant. In case 2, the air quality in most of southern Taiwan is worse in summer, which coincides with the fact that the air pollution of southern Taiwan is worse than northern Taiwan. The last case is about the transfer of residential property in Taiwan. Recently, the Ministry of Finance revised real estate taxation rules to introduce the consolidated housing and land tax system. This new system assesses tax based on real transaction prices and gains, which encourages people to buy houses for self-use rather than investment purposes. In the last case, we expect users to come up with the conclusion that the transaction amounts exhibited a sharp decrease in the market after the policy of the consolidated housing and land tax system was implemented in 2016.

During the evaluation phase, we record the timestamp information for each section that contains only one question. The different tasks are used to ask our participants to answer listed questions on a screen. For each section in the different tasks, as listed in Table 2, we record how long a user spent on each question.

Table 3 shows the accuracy, average time spent, and standard deviation of time spent for each question about the generated puzzle tile maps. The selection accuracies are greater than 95%. The average amount of time spent on the questions ranges between 25 and 56 s. The standard deviation of time spent ranges between 10 and 23 s. To provide a comparison with related work, the visualization method called dynamic queries and brushing on choropleth maps [4] with an additional bar chart is implemented. Then, the results of Dang et al. [4] are compared with our method. The average amount and standard deviation of the time spent for each question and the accuracy of each question are provided. The results are shown in Table 4. In the comparisons, the standard deviation of the time spent for each question in Table 3 is lower than that in Table 4. The selection accuracies in Table 3 are much higher than those in Table 4. From the statistical results in Tables 3 and 4, we draw the conclusion that the puzzle tile map generated by the proposed method has the characteristics of usability, readability, and comparability. In addition, a few users commented that questions Q1–Q3 and Q7–Q9 are difficult to decide on based on the

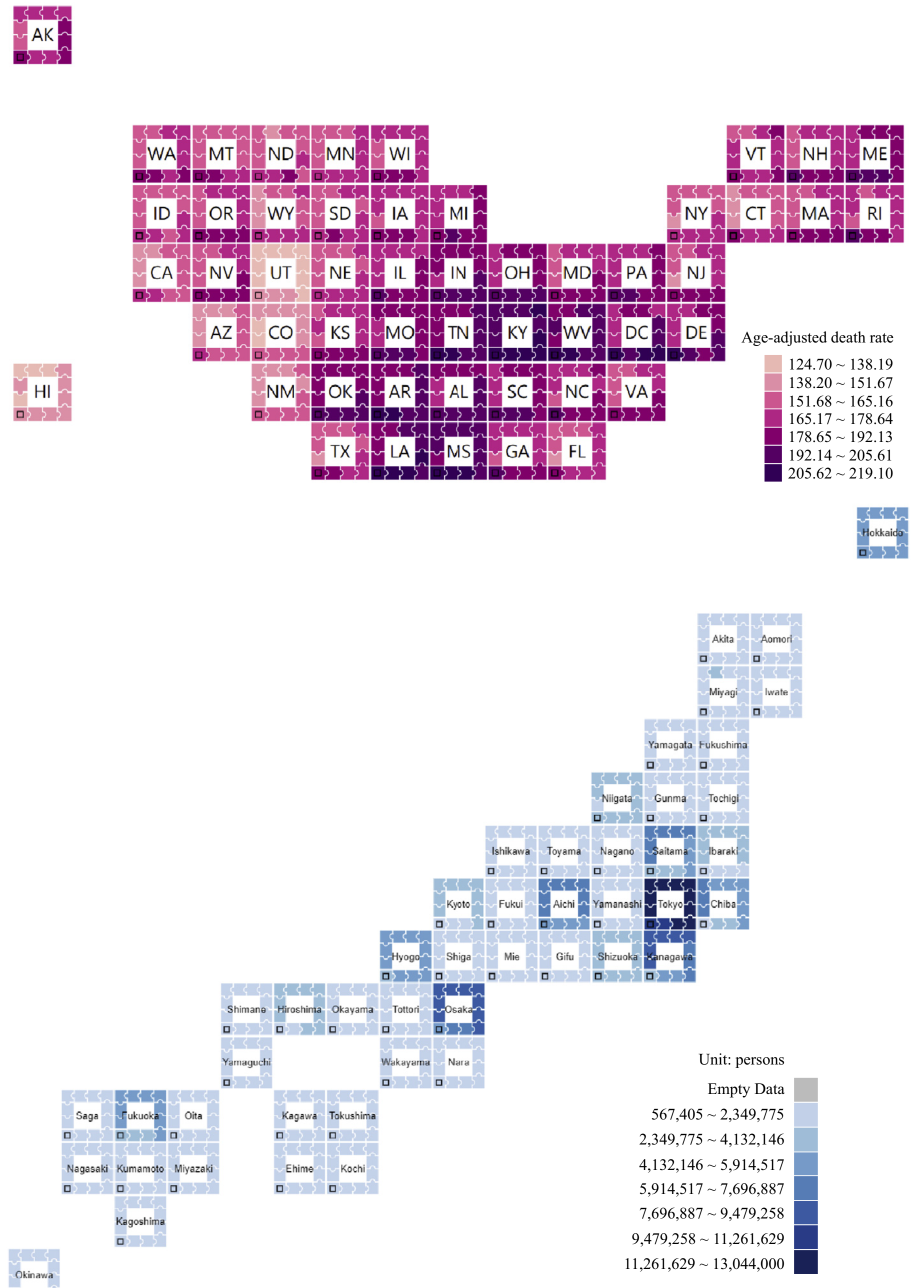

Fig. 12. Results of puzzle tile maps. Top: age-adjusted death rate for cancer in the U.S.A. from 2004 to 2015. Bottom: population in Japan every 5 years from 1960 to 2014.

Table 2
Task questionnaire.

Case	Questions	Affecting factors
C1: Total fertility rate	• As shown in the diagrams, YUN announced the promulgation of the maternity allowance policy. How did the policy affect the total fertility rate for 2010?	C1F1, C1F3, C1F4, C1F5
	• After announcing the promulgation of maternity allowance policy in TPE, how did the policy affect the total fertility rate?	C1F1, C1F3, C1F4, C1F5
	• As shown in the diagrams, which year has the best fertility rate in NTO?	C1F1, C1F2, C1F4
C2: Air pollution	• As shown in the diagrams, which season has the best air quality in KHH?	C2F1, C2F2, C2F4
	• As shown in the diagrams, which city had the worst air quality in Oct. 2016?	C2F1, C2F3, C2F5
	• As shown in the diagrams, which city has a larger impact than other city on air quality in Taiwan?	C2F1, C2F3, C2F5
C3: Transfer of residential property	• Which city had the largest amount of transfer of residential property in Q3 2014?	C3F1, C3F2, C3F4
	• After revising the consolidated housing and land tax, what is the extent of the increase or decrease in the transfer of residential property in TXG?	C3F1, C3F4, C3F5
	• Overall, is the impact of the consolidated housing and land tax on the transfer of residential property significant?	C3F1, C3F2, C3F4, C3F5

Table 3
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

Table 4
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

results generated by the visualization method [4] with a bar chart. This difficulty results in a large standard deviation of time spent in Table 4. Therefore, we conclude that the puzzle tile map can help users make a decision easily and quickly for spatial-temporal domains.

7. Conclusions, limitations, and future work

The puzzle tile map is a novel visualization technique that combines various characteristics into 2D space without sacrificing one of the characteristics of spatial-temporal data. A tile is composed of puzzle pieces and can integrate the temporal information into the spatial data. The proposed spatial-temporal visualization represented by 2D diagram can focus on time-series data in a specific region from the geographic perspective. Puzzle tile map has the ability to visualize numeric data in most cases. Users can observe data from several patterns on puzzle tile. Pieces, connection points, and event point patterns represent the data amount, data changing quantities, and event time (if necessary), respectively. The spatial-temporal data can be several cities with a data length of 16 days/weeks or several districts with a data length of 12 months, and users can slice data into any length they want. Although data in this study include geographical information, puzzle tile map can also extend to other data types. For examples, a seating plan with students grades and the number of passengers in each compartment at every station can be visualized by the puzzle tile maps, and they show in the supplemental result. Actually, numeric data with positional and temporal information can be visualized by the proposed approach. The suitable time scale of puzzle tile map depends on the layout container size, and in this study, it is between 8 and 20.

The puzzle pattern of visualization used to represent the amount of information in a 2D tile map is a double-edged sword. If massive data visualization is required, then observing several characteristics from the generated results may be difficult, resulting in considerable loss of details on the tile map. The proposed method is also not a sustainable visualization to gain significant details from large volumes of time-series data.

In the future, we plan to extend the puzzle tile map to various platforms with different screen aspect ratios by using responsive layout techniques. Furthermore, we plan to extend the method to be able for non-uniform layout. In this manner, the blank spaces of the whole layout will be reduced while the shape of the layout can be preserved as much as possible. Then, a hierarchical visualization strategy with different levels and numbers of tiles is a possible solution, and with a zoom in/out function, it will allow user to change timescale of observing view. This method can be used effectively in advertising applications, including poster making.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the manuscript. This research was supported in part by the Ministry of Science and Technology (contracts MOST-106-2221-E-019-069-MY2, MOST-107-2119-M-006-027, and MOST-104-2321-B-019-009), Taiwan.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cag.2019.05.002.

References

[1] Lin S-S, Lin C-H, Hu Y-J, Lee T-Y. Drawing road networks with mental maps. IEEE Trans Vis Comput Gr 2014;20(9):1241–52.
[2] McNeill G, Hale SA. Generating tile maps. Comput Gr Forum 2017;36(3):435–45.
[3] Suematsu H, Yagi S, Itoh T, Motohashi Y, Aoki K, Morinaga S. A heatmap-based time-varying multi-variate data visualization unifying numeric and categorical variables. In: Proceedings of the eighteenth international conference on information visualization; 2014. p. 84–7.
[4] Dang G, North C, Shneiderman B. Dynamic queries and brushing on choropleth maps. In: Proceedings of the fifth international conference on information visualization; 2001. p. 757–64.
[5] Thakur S, Hanson AJ. A 3d visualization of multiple time series on maps. In: Proceedings of the fourteenth international conference on information visualization; 2010. p. 336–43.
[6] Tominski C, Schulze-Wollgast P, Schumann H. 3d information visualization for time dependent data on maps. In: Proceedings of the ninth international conference on information visualization; 2005. p. 175–81.
[7] Smallman HS, St John M, Oonk HM, Cowen MB. Information availability in 2d and 3d displays. IEEE Comput Gr Appl 2001;21(5):51–7.
[8] Meulemans W, Dykes J, Slingsby A, Turkay C, Wood J. Small multiples with gaps. IEEE Trans Vis Comput Gr 2017;23(1):381–90.
[9] Eppstein D, van Kreveld MJ, Speckmann B, Staals F. Improved grid map layout by point set matching. Int J Comput Geom Appl 2015;25(2):101–22.
[10] Thakur S, Rhyne T-M. Data vases: 2d and 3d plots for visualizing multiple time series. In: Proceedings of the fifth international symposium on advances in visual computing: part II; 2009. p. 929–38.
[11] Högerstrand T. What about people in regional science? Pap Reg Sci Assoc 1970;24(1):7–21.
[12] Gatalsky P, Andrienko N, Andrienko G. Interactive analysis of event data using space-time cube. In: Proceedings of the eighth international conference on information visualization; 2004. p. 145–52.
[13] Zazhary R. Data visualization strategies using tile grid maps; 2015. https://www.gislounge.com/data-visualization-strategies-using-tile-grid-maps/.
[14] Wongsuphasawat K. A semi-automatic way to create your own grid map; 2016. https://medium.freecodecamp.com/creating-grid-map-for-thailand-397b53a4ecf.
[15] Shaw T. Good data visualization practice: tile grid maps; 2016. https://forumone.com/ideas/good-data-visualization-practice-tile-grid-maps-0.
[16] Cano RG, Buchin K, Castermans T, Pieterse A, Sonke W, Speckmann B. Mosaic drawings and cartograms. Comput Gr Forum 2015;34(3):361–70.
[17] Wu W, Zheng Y, Cao N, Zeng H, Ni B, Qu H, et al. Mobiseg: Interactive region segmentation using heterogeneous mobility data. In: Proceedings of the IEEE Pacific visualization symposium; 2017. p. 91–100.
[18] Chen S, Chen S, Wang Z, Liang J, Wu Y, Yuan X. D-map+: interactive visual analysis and exploration of ego-centric and event-centric information diffusion patterns in social media. ACM Trans Intell Syst Technol 2018;10(1):11:1–11:26.

[19] Auber D, Huet C, Lambert A, Renoust B, Sallaberry A, Saulnier A. Gospermap: using a Gosper curve for laying out hierarchical data. IEEE Trans Vis Comput Gr 2013;19(11):1820–32.
[20] Tong C, Roberts R, Laramee RS, Berridge D, Thayer D. Cartographic treemaps for visualization of public healthcare data. In: Proceedings of the computer graphics and visual computing (CGVC); 2017.
[21] Buttenfield BP. Treatment of the cartographic line. Cartographica 1985;22(2):1–26.
[22] K Peucker T. A theory of the cartographic line. Int Yearb Cartogr 1976;16:508–18.
[23] Turf.js http://turfjs.org/; Accessed date 30 March 2017.
[24] Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Canad Cartogr 1973;10(2):112–22.
[25] Brown J, Ashlock D, Orth J, Houghten S. Autogeneration of fractal photographic mosaic images. In: Proceedings of the IEEE congress of evolutionary computation; 2011. p. 1116–23.
[26] Kuhn H.W.. The hungarian method for the assignment problem. 50 Years of Integer Programming 1958–20082010;:29–47.
[27] Borgo R, Kehrer J, Chung DHS, Maguire E, Laramee RS, Hauser H, et al. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In: Eurographics (STARs); 2013. p. 39–63.
[28] Chung D, Legg P, Parry M, Bown R, Griffiths I, Laramee R, et al. Glyph sorting: Interactive visualization for multi-dimensional data. Inf Vis 2013;14.
[29] Tominski C, Fuchs G, Schumann H. Task-driven color coding. In: Proceedings of the twelfth international conference on information visualization; 2008. p. 373–80.
[30] Harrower M, Brewer CA. Colorbrewer.org: an online tool for selecting colour schemes for maps. Cartogr J 2003;40(1):27–37.