RESEARCH ARTICLE

Tiled vector data model for the geographical features of symbolized maps

Lin Li^{1,2,3*}, Wei Hu¹, Haihong Zhu^{1*}, You Li¹, Hang Zhang¹

1 School of Resource and Environmental Sciences, Wuhan University, Wuhan, China, **2** Geo Spatial Information Science Collaborative Innovation Center of Wuhan University, Wuhan, China, **3** The Key Laboratory for Geographical Information Systems, Ministry of Education, Wuhan, China

* lilin@whu.edu.cn (LL); hhzhu@whu.edu.cn (HHZ)

Abstract

Electronic maps (E-maps) provide people with convenience in real-world space. Although web map services can display maps on screens, a more important function is their ability to access geographical features. An E-map that is based on raster tiles is inferior to vector tiles in terms of interactive ability because vector maps provide a convenient and effective method to access and manipulate web map features. However, the critical issue regarding rendering tiled vector maps is that geographical features that are rendered in the form of map symbols via vector tiles may cause visual discontinuities, such as graphic conflicts and losses of data around the borders of tiles, which likely represent the main obstacles to exploring vector map tiles on the web. This paper proposes a tiled vector data model for geographical features in symbolized maps that considers the relationships among geographical features, symbol representations and map renderings. This model presents a method to tailor geographical features in terms of map symbols and ‘addition’ (join) operations on the following two levels: geographical features and map features. Thus, these maps can resolve the visual discontinuity problem based on the proposed model without weakening the interactivity of vector maps. The proposed model is validated by two map data sets, and the results demonstrate that the rendered (symbolized) web maps present smooth visual continuity.

OPEN ACCESS

Citation: Li L, Hu W, Zhu H, Li Y, Zhang H (2017) Tiled vector data model for the geographical features of symbolized maps. PLoS ONE 12(5): e0176387. <https://doi.org/10.1371/journal.pone.0176387>

Editor: George-John Nychas, Agricultural University of Athens, GREECE

Received: October 13, 2016

Accepted: April 10, 2017

Published: May 5, 2017

Copyright: © 2017 Li et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the Supporting Information files.

Funding: The article was supported by the national Natural Science Foundation of China under grant No.41271453 and url is <http://www.nsf.gov.cn/>, The Scientific and Technological Leading Talent Fund of National Administration of Surveying, mapping and geo-information (2014), Wuhan ‘Yellow Crane Excellence’ (Science and Technology) program (2014) and the Doctoral Independent Scientific Research Project in Wuhan University (2012205020211). The funders had no

Introduction

Web electronic maps (E-maps) have been extensively exploited and widely applied with the rapid development of technologies such as wireless communication, computer technology and geographic space information technology [1]. Many successful applications of E-maps have been implemented in daily life [2]. E-maps can be used as information inquiry services by providing users with specific information, such as POIs (points of interest) and tourist information [3,4]. E-maps present bus routes and real-time traffic information and provide data that support vehicle navigation [5,6]. E-maps are also useful for teaching geography [7] and urban planning [8]. The emergence of social media (such as Twitter, Facebook, and the Sina microblog) has inspired a considerable amount of research that has used check-in data and

role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

additional data regarding the social ties between users, which are collected from various sources [9]. Street views [10], indoor maps [11] and WEBGIS/VR [12–15] further extend the applications of E-maps.

The traditional applications of E-maps are implemented based on web map services (WMS) [16]. However, the spatial parameters in WMS requests are not constrained, which forces images to be generated on the fly each time a request is received, and the processes involved in the image generation by the WMS server cause delays typically on the order of several seconds [17]. Furthermore, WMS is not scalable to large numbers of users due to a lack of ability to cache requests and this is implicit in the way the standards are written [18]. Therefore, Google Maps, Yahoo Maps, Virtual Earth and other popular E-maps use a “pyramid technique” that is based on tiles in the form of raster data to improve transmission performance [19]. A new map service standard called Web Map Tile Service (WMTS) has been developed based on WMS and the “pyramid technique”, which have been adopted by most large-scale web mapping systems [17,18,20]. Under the WMTS framework, the map is divided into images called “tiles” that are transmitted to the client side according to the requested region. The tiles are re-combined by using their respective coordinates on the client side [21–24].

Another critical issue for E-maps is interactivity, which is also a hallmark of E-maps [25]. E-maps with broader interactive functions can enable users to use the map data for spatial analyses and temporal-spatial phenomena investigations [26–30]. Interactive E-maps can help strengthen spatial cognition, enhance expressive abilities and improve map availability [26,31–33].

Compared with raster data maps, which are limited in their interactivity, maps that use vector data provide excellent and unique advantages in accessing geographical features, although the uneven data amounts of vector maps may cause transmission efficiency problems. The Open Geospatial Consortium provides a Web Feature Service Interface Standard (WFS), which enables requests for geographical features across the web. To overcome the lower vector data transmission rates, many studies have proposed progressive transmission solutions [34–39] and data compression to increase transmission efficiency [40–43]. Meanwhile, the emergence of GeoJSON has provided a better web data encoding format to improve transmission efficiency [44]. Compared to XML-based web data encoding formats (GML/KML), GeoJSON can be more easily and quickly parsed by a computer, and can describe complex data structures [43]. GeoJSON also provides a light-weight data encoding format that can be easily transferred. These efforts have indeed improved the performance of vector map transmissions over the Internet. However, transmitting all vector data online is still time consuming and likely intolerable for interactive operations, and the pyramid technique remains an effective organizing method for vector maps, namely vector tile map.

When a vector map is presented on the client side, individual map tiles are sequentially transmitted through the Internet and then reassembled into a map. To decrease the amount of transmitted data, a vector map can be divided into two data sets: geographical features and cartographic representation. Only geographical features are transmitted, and cartographical representation is completed locally on the client side. Cartographical representation includes the rules for portraying geographical features and map symbols [45]. For example, an ArcGIS vector map is represented by geographic features and *.style files [46,47]. Map legibility largely depends on the rendering of map features with different symbols [48]. However, this organizational structure of vector maps is challenged when geographical features are symbolized in individual ‘vector tiles’ and joined with neighboring features. Such a map rendering method produces visual discontinuities and conflicts at the tile borders, which may break the spatial distribution characteristics of features [49] and reduce the readability and applicability of the

map [48,50]. Therefore, a critical issue for tiled vector E-maps is to join neighboring features with their symbol representation to maintain visual continuity and avoid visual conflicts.

To eliminate visual breakage, this study proposes a tiled vector data model for the geographical features that define the additivity of map features and geographical features, partition vector geographical features, and implement map symbolizations to graphically match joined symbolized partitioned features without causing graphic conflicts and losses.

This paper is organized as follows: Section 2 describes the symbol representation method. Section 3 introduces the tiled vector data model, verifies the feasibility of the data model, and then proposes different data organization methods for geographical features based on the tiled vector data model. Section 4 reports an experiment that uses the proposed model and discusses the results of the experiment. Finally, Section 5 presents the conclusions and future work.

Symbol representation

Point symbols, linear symbols and area symbols are the three basic symbol representation methods for vector data in cartography and geographic information system [48]. The symbol representation for point features renders the specified point symbol at a location point. The symbol representation for area features can be summarized as the rendered filling of different graphic cells based on the scan line algorithm or its improved algorithms. Compared to the symbol representation for point symbols and area symbols, the symbol representation for linear features is much more complicated. A linear feature is generally symbolized by repeatedly drawing the corresponding symbol along its path. Fig 1(a) shows the symbol of a railway and the map feature of a railway geographical feature (representing the linear feature), and Fig 1(b) shows the symbol of grassland and the map feature of a grassland geographical feature (representing the area feature).

Any symbol q in the symbol library Q is an ordered set that consists of geometric graphics (such as polylines, polygons and ellipses) with properties that include color and line width. The circumscribing polygon of the symbol can be expressed as a rectangle; the width of the rectangle is referred to as the width of the symbol, and the length of the rectangle is referred to as the length of the symbol. λ is a symbol parameter: when the feature is a linear feature, the symbol parameter λ denotes the length of the linear symbol l , and when the feature is an area feature, λ denotes the width of the symbol w and the length l of the symbol [51]. The symbol

Fig 1. (a) Railway symbol and its map feature, and (b) grassland symbol and its map feature.

<https://doi.org/10.1371/journal.pone.0176387.g001>

parameter λ can thus be expressed as follows:

$$\lambda = \begin{cases} l, & \text{the symbol is a linear symbol} \\ (w, l), & \text{the symbol is an area symbol} \end{cases} \quad (1)$$

Tiled vector data model for geographical features

Mapping geographical feature space on map feature space

Geographical feature space is the abstract expression of spatial entities and geographical phenomena in the real world [52]. Geographical feature space can be converted to map feature space by symbol representation. This conversion process can be expressed as follows [51]:

$$M = S(G, Q) \quad (2)$$

where M represents the map feature space, G represents the geographical feature space, Q represents the symbol library (set) that corresponds to the geographical feature space G , and S represents the common function for symbolizing the geographical feature space. The conversion process for an individual map feature in the map feature space can be expressed as follows:

$$m = S(g, q) (m \in M, g \in G, q \in Q) \quad (3)$$

In eq (3), the connection between geographic features and map features is fixed for a given scale, this is because the symbol is fixed. However, a feature may have different symbols at different scales.

Graphic conflicts in symbolization and operators for tiled vector data model

Graphic conflicts and losses may occur along the borders of tiles when symbolizing geographical features on each tile and connecting tiles according to their respective coordinates. Examples of the three basic types of problems that arise are shown in Fig 2. As shown in Fig 2(a), when the points at the borders of the tiles are symbolized, the components of the map features inside the corresponding tile are easily rendered. Nonetheless, we cannot omit the components of map features that are outside these tiles because such deletions would create additional issues, such as a loss of certain components and the non-conformity of map features along the

Fig 2. Graphic conflicts and losses.

<https://doi.org/10.1371/journal.pone.0176387.g002>

tile borders. Fig 2(b) shows an example of symbolizing a linear feature along the borders of these tiles, and the problems are marked. The loss of map features from Mark 1 and Mark 3 is obvious, and graphic conflicts between two adjacent tiles are shown at Mark 2. Fig 2(c) shows graphic conflicts in area map features that are caused by filling graphics cells at neighboring tiles. The reason for the graphic conflicts in Fig 2 is that geographical features are clipped into different vector tiles: when symbolizing the clipped features on each tile and connecting tiles according to their respective coordinates, the symbolized features (map features) cannot match well at the tile borders. Thus, obvious graphic conflicts occur between map features in neighboring tiles. The key to resolving these problems is to design or construct an operator for map features with ‘additivity’, which means that two separate symbolizations can yield the same graphic presentation as a single symbolization.

Several basic concepts and signs are introduced here to illustrate the above problems and clarify the principle of the proposed model. The “addition” sign $\oplus$ denotes joining two features, $\oplus_M$ denotes joining two map features and $\oplus_G$ denotes joining two geographical features.

Definition 1. The premise of the “addition” operator for geographical features: The operation of $g_1 \oplus_G g_2$ is tenable if EITHER (1) both g_1 and g_2 are linear features, g_1 and g_2 are connected at their ends, or $g_1 = g_2$; OR (2) both g_1 and g_2 are area features and g_1 intersects, touches or contains g_2 .

Examples for Definition 1 are shown in Fig 3, where (a), (d) and (e) satisfy the premise of the “addition” operator for geographical features and (b), (c) and (f) do not satisfy the premise of the “addition” operator.

Definition 2. The “accumulation” operator for geographical features: An “accumulation” operator for geographical features is defined as $\wedge$. $\wedge_{i=1}^n g_i$ can be expressed as $\wedge_{i=1}^n g_i = g_1 \oplus_G g_2 \oplus_G \dots \oplus_G g_n$ ($g_i \in G, n \geq 1, g_1, g_2, \dots, g_n$ satisfy Definition 1).

The difference between the “addition” sign for geographical features and the “accumulation” operator in this study is similar to the difference between the plus sign “+” and the sum operator “ Σ ” in mathematics. To avoid cognitive conflicts with common mathematical operators, we use “ $\oplus_G$ ” and “ $\wedge$ ” to replace “+” and “ Σ ”, respectively. The “accumulation” operator for geographical features is used in this article to simplify this formula. As shown in Fig 4, g is a symbol, and the linear feature A-B is divided into 5 components (g_1, g_2, g_3, g_4 and Δg) by this symbol. Thus, the linear feature can be written as $g_1 \oplus_G g_2 \oplus_G g_3 \oplus_G g_4 \oplus_G \Delta g$. However, expressing all the geographical features in this manuscript similarly to this formula is tedious. Therefore, we use the “accumulation” operator to simplify this formula to $\wedge_{i=1}^4 g_i \oplus_G \Delta g$.

Fig 3. Examples of the premise of the “addition” operator for geographical features.

<https://doi.org/10.1371/journal.pone.0176387.g003>

Fig 4. Examples of the “accumulation” operator for geographical features.

<https://doi.org/10.1371/journal.pone.0176387.g004>

Definition 3. The premise of the “addition” operator for map features: The geographical features g_1 and g_2 have the same symbol q and a corresponding symbol parameter λ . By using these symbols, we can obtain the map features $m_1 = S(g_1, q)$ and $m_2 = S(g_2, q)$. The formula $m_1 \oplus_M m_2 = m_3$ holds true when (1) feature g_1 can be divided into n components ($g_1^1, g_1^2, \dots, g_1^n$, where n is an integer) by the symbol parameter λ and (2) g_1 and g_2 satisfy Definition 1. In this formula, m_3 can be expressed as $m_3 = S(g_1 \oplus_G g_2, q)$.

If the features satisfy Definition 3, the map features can match well. Some examples are shown in Fig 5 to explain Definition 3. In Fig 5(a), both g_1 and g_2 are the railway feature, and q is the railway symbol. $S(g_1, q)$ and $S(g_2, q)$ do not satisfy the premise of the “addition” operator for map features in Fig 5(a) because feature g_1 cannot be divided into n components (n is an integer) by the length of the symbol q . $S(g_1, q) \oplus_M S(g_2, q)$ is obviously not equal to $S(g_1 \oplus_G g_2, q)$. In Fig 5(b), feature g_1 can be divided into n components (n is an integer) by the length of the symbol q , and $S(g_1, q) \oplus_M S(g_2, q)$ is equal to $S(g_1 \oplus_G g_2, q)$.

Definition 4. The invariance of addition by itself: For any feature $\varphi (\varphi \in M \parallel \varphi \in G)$, $\varphi \oplus \varphi = \varphi$ is tenable.

The railway symbol is chosen as an example to illustrate why the traditional vector data model is already inapplicable to vector map tiles based on Definitions 1–4 but applicable for other linear symbols. For example, Fig 6 shows two tiles called T_1 and T_2 ; a railway feature g crosses the tiles and is clipped by the tiles into g_1 and g_2 . Thus, we can obtain the following formula:

$$g = g_1 \oplus_G g_2 \quad (4)$$

Fig 5. Examples of the premise of the “addition” operator for map features.

<https://doi.org/10.1371/journal.pone.0176387.g005>

Fig 6. Example of linear features.

<https://doi.org/10.1371/journal.pone.0176387.g006>

The map feature that corresponds to g can be expressed as follows:

$$m = S(g, q) \quad (5)$$

where q is the railway symbol and m denotes the map feature of g . Similarly, the map features of g_1 and g_2 can be represented by the following formulas:

$$m_1 = S(g_1, q) \quad (6)$$

$$m_2 = S(g_2, q) \quad (7)$$

If $m_1 \oplus_M m_2$ is equal to m , then the map features match well at the borders of the tiles. As shown in Fig 6, we introduce λ to represent the length of the symbol q , and then feature g can be divided into n components ($g_1^\lambda, g_2^\lambda, \dots, g_n^\lambda$, where n is an integer) by the corresponding symbol parameter λ . As shown in Fig 6, we assume that feature g intersects tiles at the component k of the feature and that the tiles divide g_k^λ into Δg_1 and Δg_2 . Δg_1 and Δg_2 are located in T_1 and T_2 , respectively. Feature g can then be rewritten as follows:

$$g = \wedge_{i=1}^n g_i^\lambda \oplus_G \Delta g \quad (8)$$

The expression of feature g_1 then becomes:

$$g_1 = \wedge_{i=1}^{k-1} g_i^\lambda \oplus_G \Delta g_1 \quad (9)$$

The expression of feature g_2 then becomes:

$$g_2 = \Delta g_2 \oplus_G \wedge_{i=k+1}^n g_i^\lambda \oplus_G \Delta g \quad (10)$$

In Formula (9), Δg_1 is not equal to g_k^λ , as shown in Fig 6, so g_1 cannot be divided into n components (n is an integer) by the length of the symbol q . Therefore, $m_1 \oplus_M m_2$ does not satisfy Definition 3. As a result, $m_1 \oplus_M m_2$ is not equal to m and the map features of feature g_1 and feature g_2 do not match well at the borders of the tiles, an illustrative diagram is shown as Fig 7.

Fig 7. Example of linear features based on the traditional data model.

<https://doi.org/10.1371/journal.pone.0176387.g007>

Both linear features (railway) and other types of features encounter this problem. Therefore, the key to implementing the vector map tile technique is proposing a tiled vector data model for geographical features so that tiles can comprise an entire map with accurate and complete cartographic representation. Using the proposed data model can make the cartographic representation results render more favorably for features along the borders of tiles. Meanwhile, the rationality of the data model is also verified in this section.

Tiled vector data models for point, linear and area features

We designed a novel tiled vector data model for all types of features to match map features well along the borders of tiles.

Tiled vector data model for linear features. A tiled vector data model for linear features is proposed based on Definition 3 as follows:

$$g = \begin{cases} \bigwedge_{i=1}^n g_i^j \oplus_g \Delta g \oplus_g \sigma, & \text{Case 1 : The end point of the linear feature is the intersection point.} \\ \sigma \oplus_g \Delta g_1 \oplus_g \bigwedge_{i=1}^n g_i^j \oplus_g \Delta g, & \text{Case 2 : The start point of the linear feature is the intersection point.} \\ \sigma_1 \oplus_g \Delta g_1 \oplus_g \bigwedge_{i=1}^n g_i^j \oplus_g \Delta g_2 \oplus_g \sigma_2, & \text{Case 3 : The start point and end point of the linear feature are the intersection points.} \end{cases} \quad (11)$$

In Formula (11), σ is an expansion feature and $\Delta g \oplus_g \sigma = g^k$. Fig 8 shows examples for Formula (11). In Fig 8(a), the linear feature A-B intersects the tile at point P. The end point for the linear feature A-P is the intersection point, therefore, the data model for feature A-P is case 1 in Formula (11). In Fig 8(b), the data model for feature P-B is case 2 in Formula (11). In Fig 8(c), the data model for feature P1-P2 is case 3 in Formula (11). An illustrative diagram that is based on the proposed tiled vector data model for linear features in Fig 6 is shown in Fig 9. Based on

Fig 8. Examples of linear features for the tiled vector data model.

<https://doi.org/10.1371/journal.pone.0176387.g008>

Formula (11), the feature g_1 in Fig 6 satisfies case 1 and thus can be changed as follows:

$$g_1 = \wedge_{i=1}^{k-1} g_i^j \oplus_G \Delta g_1 \oplus_G \Delta g_2 = \wedge_{i=1}^{k-1} g_i^j \oplus_G g_k^j \quad (12)$$

The feature g_2 in Fig 6 satisfies case 2 and can be rewritten as follows:

$$g_2 = \Delta g_1 \oplus_G \Delta g_2 \oplus_G \wedge_{i=k+1}^n g_i^j \oplus_G \Delta g = g_k^j \oplus_G \wedge_{i=k+1}^n g_i^j \oplus_G \Delta g \quad (13)$$

According to Formula (12) and Formula (13), the expansion features of g_1 and g_2 are Δg_2 and

Fig 9. Example of linear features based on the tiled vector data model.

<https://doi.org/10.1371/journal.pone.0176387.g009>

Δg_1 , respectively, and $m_1 \oplus_M m_2$ can be rewritten as follows:

$$m_1 \oplus_M m_2 = (S(\wedge_{i=1}^{k-1} g_i^{\lambda}, q) \oplus_M S(g_k^{\lambda}, q)) \oplus_M (S(g_k^{\lambda}, q) \oplus_M S(\wedge_{i=k+1}^n g_i^{\lambda}, q) \oplus_M S(\Delta g, q)) \quad (14)$$

Formula (14) can be simplified as follows based on Definition 3 and Definition 4:

$$\begin{aligned} m_1 \oplus_M m_2 &= S(\wedge_{i=1}^{k-1} g_i^{\lambda}, q) \oplus_M S(g_k^{\lambda}, q) \oplus_M S(\wedge_{i=k+1}^n g_i^{\lambda}, q) \oplus_M S(\Delta g, q) \\ &= S(\wedge_{i=1}^n g_i^{\lambda}, q) \oplus_M S(\Delta g, q) \end{aligned} \quad (15)$$

In Formula (15), $m_1 \oplus_M m_2$ is equal to m , which means that using the data model for linear features in Formula (11) can solve the problems of graphic conflicts and losses at the borders of tiles. This relationship also proves that expanding a feature that is clipped by a tile to a complete symbol can resolve these problems.

Tiled vector data model for point features. When using the method of linear features for reference and by considering the simplicity of point features, the data model for point features is as follows:

$$g = \begin{cases} g, & \text{Case 1 : The tile contains point.} \\ \sigma, & \text{Case 2 : The tile does not contains point, the map feature intersects the tile.} \\ \emptyset, & \text{Case 3 : The map feature does not intersect the tile.} \end{cases} \quad (16)$$

In the data model, σ is an expansion feature and $\sigma = g$. According to Formula (16), case 1 means that if the tile contains a point feature, this point feature is added to the feature set of the tile. In case 2, when the tile does not contain a point feature but the map feature of a point intersects the tile, we also add the point feature to the feature set of the tile. In case 3, when the map feature of a point does not intersect the tile, then the model skips the point feature.

Tiled vector data model for area features. Research on the rendering of area symbols can be summarized as the filling of different graphic cells [48]. Graphic cells have two patterns: color filling and symbol filling. When the filling pattern is color filling, the map features of an area feature can match well along the borders of tiles. However, when the filling pattern is symbol filling, as shown in Fig 2(c), then obvious graphic conflicts from area map features occur within neighboring tiles. Generally, a line is considered a one-dimensional object and an area is considered a two-dimensional object. Area features are more complicated than linear features; specifically, the tiled vector data model for linear features cannot adequately handle area features. We now introduce several basic concepts and signs prior to introducing the tiled vector data model for area features.

Definition 5. The symbolization starting point of area features: The upper-left corner of the circumscribing rectangle of an area feature is called the symbolization starting point of the area feature.

As shown in Fig 10, the upper-left corner of the circumscribing rectangle of area feature g (polygon A-B-C-D-E) is vertex P, so vertex P is called the symbolization starting point of feature g .

Definition 6. The premise of the “addition” operator for area map features: For area features g_1 and g_2 , the symbolization starting points of g_1 and g_2 are located at $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, respectively. These features have the same symbol q , and the corresponding symbol parameter λ contains the width of the symbol w and the length of the symbol l . We can obtain the area map features $m_1 = S(g_1, q)$ and $m_2 = S(g_2, q)$. The formula $m_1 \oplus_M m_2 = m_3$ holds true; therefore, m_1 and m_2 can be properly matched when the following criterias are met:

Fig 10. Examples of symbolization starting points for area features.

<https://doi.org/10.1371/journal.pone.0176387.g010>

(1) $|x_1 - x_2| = k \times w$ (k is an integer), (2) $|y_1 - y_2| = n \times l$ (n is an integer), and (3) g_1 and g_2 satisfy Definition 1. In this formula, m_3 can be expressed as $m_3 = S(g_1 \oplus g_2, q)$.

Some examples are shown in Figs 11 and 12 to explain Definition 6. As shown in Fig 11, the area features are g_1 (polygon A-B-C-D) and g_2 (polygon G-E-F). The symbol is q , and the symbol parameter is $\lambda(w, l)$. The map features of g_1 and g_2 are $m_1 = S(g_1, q)$ and $m_2 = S(g_2, q)$, respectively. The circumscribing rectangle of g_1 and g_2 can be divided by the corresponding symbol as the dashed boxes shown in Fig 11. Obviously, the areas g_1 and g_2 do not satisfy (1) $|x_1 - x_2| = k \times w$ (k is an integer) and (2) $|y_1 - y_2| = n \times l$ (n is an integer). Therefore, as shown in Fig 11, $S(g_1, q) \oplus_M S(g_2, q)$ is not equal to $S(g_1 \oplus g_2, q)$. In Fig 12, the areas g_1 and g_2 satisfy Definition 6 and $S(g_1, q) \oplus_M S(g_2, q)$ is equal to $S(g_1 \oplus g_2, q)$, as shown in Fig 12.

Based on Definition 5 and Definition 6, the tiled vector data model for area features is as follows:

$$g = \begin{cases} \Delta g, & \text{the fill pattern is color filling} \\ \Delta g \oplus_G \sigma, & \text{the fill pattern is color filling} \end{cases} \quad (17)$$

In Formula (17), Δg is the result of the area feature that is clipped by tiles and σ is an expansion feature of Δg that allows g to satisfy the data model. The details of the expansion feature σ are illustrated in Section “Data organization method for area features”.

Data organization based on the tiled vector data model

We now propose data organization methods for vector features based on the “Tiled vector data model for point, linear and area features” section. The results can avoid graphic conflicts and losses when they are symbolized. Moreover, the results can ensure the correctness and integrity of the geographical features. These features can be applied to user interactions and spatial analyses.

Data organization method for point features. Based on the data model for point features, the data organization method for point features consists of the following steps:

1. Obtain the circumscribing polygon of the map feature. If the circumscribing polygon does not intersect the target tile, skip the point.

Fig 11. Example that does not satisfy Definition 6: (a) area geographical features g_1 and g_2 , (b) grassland symbol q , (c) map features of g_1 , (d) map features of g_2 , (e) map feature $m_1 \oplus m_2$, and (f) map feature $S(g_1 \oplus g_2, q)$.

<https://doi.org/10.1371/journal.pone.0176387.g011>

2. If the circumscribing polygon of the map feature intersects the target tile, add the point to the feature set of the vector tile.

An example of point feature data organization is shown in Fig 13. In the traditional data model for point features, only tile 1 contains Point A, which may cause graphic losses, as illustrated in Fig 2. Based on the proposed tiled vector data model for point features, tile 1 satisfies case 1, and tiles 2–4 satisfy case 2. Therefore, we added Point A to the feature set of all of the tiles (tiles 1–4).

Data organization method for linear features. The preparation for processing linear features mainly includes the following steps. Step one is to obtain the corresponding symbol for the linear feature. Then, the linear feature is divided into many segments based on the length symbol. We fully exploit the rectangular characteristic of the tile. The Liang-Barsky [53] algorithm is appropriate for organizing lines. The Liang-Barsky algorithm is a parametric line

Fig 12. Example that satisfies Definition 6: (a) area geographical features g_1 and g_2 , (b) grassland symbol q , (c) map features of g_1 , (d) map features of g_2 , (e) map features $m_1 \oplus_M m_2$ and $m_1 \oplus_M m_2 = S(g_1 \oplus_G g_2, q)$.

<https://doi.org/10.1371/journal.pone.0176387.g012>

clipping algorithm that clips straight lines against a standard rectangle. The algorithm operation process is simple because it only considers parameters. Coordinate operations are only performed when necessary. The Liang-Barsky algorithm can meet the requirements of the organization method and improves the computational efficiency. As shown in Fig 14, a linear feature $P_1 - P_2$ crosses the tile, and the steps for organizing linear features are as follows:

1. Divide a line into parts according to the length of the symbol. Uniform nodes are the hollow nodes in Fig 14.
2. Obtain the intersection points of the linear feature $P_1 - P_2$ and tile, called V_1 and V_2 , respectively. In the traditional model, the saved feature in the tile is $V_1 - V_2$. However, this feature may cause graphic conflicts and losses along the borders of tiles.

Fig 13. Example of organized point features.

<https://doi.org/10.1371/journal.pone.0176387.g013>

Fig 14. Example of organized linear features.

<https://doi.org/10.1371/journal.pone.0176387.g014>

- Based on the tiled vector data model for linear features (Formula (11)), the model of saved features is $g = \sigma_1 \oplus_G \Delta g_1 \oplus_G \bigwedge_{i=1}^n g_i^j \oplus_G \Delta g_2 \oplus_G \sigma_2$. $\Delta g_1 \oplus_G \bigwedge_{i=1}^n g_i^j \oplus_G \Delta g_2$ is the linear feature $V_1 - V_2$ in this case, and we must add the expansions σ_1 and σ_2 to $V_1 - V_2$. Finally, $A - V_1$ and $V_2 - B$ are σ_1 and σ_2 , and the saved feature in the tile is $A - V_1 - V_2 - B$.

Another case that should be considered is a linear feature that does not intersect the tile alongside a map feature that does intersect the tile. This case may result in the loss of the map feature. Concrete details are shown in Fig 15. A method to manage such situations is as follows:

- Divide the linear feature $P_1 - P_2$ into sections according to the length of the symbol. Uniform nodes are the hollow nodes in Fig 15.
- According to the width of the feature symbol, move the feature to the direction of the tile and obtain new features, such as $P'_1 - P'_2$ in Fig 15.
- Find the intersection points of $P'_1 - P'_2$ and the tile, called V'_1 and V'_2 , respectively, and then measure the perpendiculars of V'_1 and V'_2 with regard to P_1P_2 . The corresponding foot points are called V_1 and V_2 , which can be regarded as the intersections of the features and tiles.

Fig 15. Special case of organizing linear features.

<https://doi.org/10.1371/journal.pone.0176387.g015>

4. Combined with the equal nodes of the symbols, expand the feature based on the tiled vector data model for linear features (Formula (11)). Finally, the saved feature in the tile is $A - V_1 - V_2 - B$.

Data organization method for area features. In this section, the grassland symbol was chosen as an example to illustrate data organization for area features. As shown in Fig 16, two tiles are marked T_1 and T_2 . An area feature g (polygon A-B-C-D-E) intersects the tiles and is clipped by the tiles into g_1 (polygon F-B-C-G) and g_2 (polygon A-F-G-D-E). The symbol for grassland is q , and the symbol parameter is $\lambda(w, l)$. The map features of g , g_1 and g_2 are $m = S(g, q)$, $m_1 = S(g_1, q)$, $m_2 = S(g_2, q)$ respectively. The symbolization starting points of g , g_1 and g_2 are located in $P(x, y)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$, respectively. The circumscribing rectangle can be divided by the corresponding symbol into the dashed boxes in Fig 16. The areas g_1 and g_2 do not satisfy Definition 6; therefore, $m_1 \oplus_M m_2$ is not equal to m and the map features cannot be matched along the borders of the tiles.

Fig 16. Example of organized area features.

<https://doi.org/10.1371/journal.pone.0176387.g016>

Based on the tiled vector data model for area features in Section “Tiled vector data model for area features”, the key of this data organization is constructing the corresponding expansion features σ_1 and σ_2 for g_1 and g_2 . However, for the feature g_1 , $x = x_1$, $y - y_1 \neq n \times l$ (n is an integer). Therefore, we must find a point $P'_1(x'_1, y'_1)$ from g_2 so that P'_1 satisfies the following formula:

$$|y - y'_1| = n \times l \quad (n \text{ is an integer}) \quad (18)$$

The expansion feature for g_1 is constructed by P'_1 . In this case, the vertex A can satisfy the formula and minimize the area of the expansion feature. Therefore, the expansion feature σ_1 is the polygon A-F-G. Finally, we add the polygon A-F-B-C-G to the feature set of tile T_1 . Similarly, we must find a point $P'_2(x'_2, y'_2)$ for the feature g_2 from g_1 so that P'_2 satisfies the following formula:

$$|x - x'_2| = n \times w \quad (n \text{ is an integer}) \quad (19)$$

The vertex H can satisfy the formula, and the expansion feature σ_2 is the polygon F-H-G. Finally, we add the polygon A-F-H-G-D-E to the feature set of tile T_2 .

Experiments

The proposed tiled vector data model are illustrated experimentally by using the data organization methods in Section “Data organization based on the tiled vector data model”.

Experiment setting and implementation

The experimental platform of this article is a personal computer (PC) with an Intel i5-4670T central processing unit (CPU) with 2.3GHz, 4 processor counts, and 8.00 GB random access memory (RAM) running the Microsoft Windows 7 Ultimate x64 operating system. All the

Fig 17. Distribution form of map explorer tiles.

<https://doi.org/10.1371/journal.pone.0176387.g017>

experiments were implemented in Desktop as our team is experienced in representing symbols with this software. The generating tile program used to implement the proposed tiled vector data model was realized using Visual C#.NET and ArcEngine of ArcGIS. ArcEngine was used for accessing spatial data as well as a tool for geometric calculation. The map explorer program was implemented using Visual C#.NET and a graphics rendering library (graphics device interface plus (GDI+)).

The experimental data were obtained from a 1:10,000 scale electronic map, including various types of geographical features, such as “main street”, “subway”, “railway”, “building”, “canal”, “vegetation”, “school” and “hospital”. As the objective of the experiment was to test the proposed tiled vector data model, all the features are not involve in map generalization. All of the features were saved as shapefile (*.shp). Then, we used the data model proposed in Section “Tiled vector data model for point, linear and area features” and organized the data using the methods outlined in Section “Data organization based on the tiled vector data model”. The organization results for each tile can be saved in any format for decoding (we saved the vector tile in the form of shapefile in the article). Naming rules were based on the rows and columns of corresponding tiles. Thus, the name of the tile in row 5, column 4 was “5_4”. The advantage of this convention is its convenience for querying and its ability to effectively and accurately finding the target tiles.

The map explorer consisted of 16 tiles. The distribution form of the tiles on the map explorer is shown in Fig 17. All the functions of the map explorer, such as initializing the symbol library, symbolizing the features and viewing the map, were implemented in this experiment. Importantly, the map explorer had to initialize the symbol library before the tiles were loaded [54]. Then, tiles could be loaded into the map explorer and the map browser could render the map features in each tile as E-maps.

Fig 18. Rendering results based the proposed data model for feature (a) points, (b) lines and (c) areas.

<https://doi.org/10.1371/journal.pone.0176387.g018>

Experimental results

Map visualization. Examples of the three basic types of features were used to verify the feasibility of the proposed tiled vector data model presented in Section “Tiled vector data model for point, linear, area features”. As shown in Fig 18, the results for each type of feature showed good performances, demonstrating the proposed data model was able to effectively solve the problems of map feature matching at the borders of tiles that are shown in Fig 2.

To further evaluate the strength of the proposed model, we generated and visualized vector tiles using different data models in different scales. Figs 19 and 20 show the experimental results that using the traditional data model and the proposed data model, respectively, based on the same experimental data at the 1:10,000 scale. The red dotted lines in Figs 19 and 20 are the frames of each tile. The graphic conflicts and losses that are caused by the traditional data model are marked in Fig 19, whereas a perfect effect is displayed in Fig 20, which does not show the problems that are associated with graphic conflicts and losses. Detailed comparisons of typical examples with serial numbers are shown in Figs 21–23. Fig 24 shows the results at the 1:20,000 scale based on the proposed data model. Figs 20 and 24 show that the map features at the borders of the vector tiles are well matched. The good visualization results demonstrate that the tiled vector data model for geographical features is feasible for different map data sets and at different scales.

Generating efficiency comparison. The running times of generating tiles for raster tiles and vector tiles based on the traditional model and the proposed model are shown in Table 1. The experiments assess different sizes of data and all the experiments were performed 10 times to obtain the average time cost to reduce the randomness of the experiments. From Table 1, we can see that generating vector tiles costs more than generating raster tiles. Because the method of generating raster tiles is only image clip while generating vector tiles needs lots of geometric computations. However, compared with raster tiles, vector tiles allow users to interact with the system and perform spatial analysis. The efficiency of generating vector tiles based on the traditional model was better than that based on the proposed model. The result is due to the structure of the proposed model, which is more complicated than the traditional model and requires additional geometric computations. Nevertheless, the main focus of this study is to eliminate graphic conflicts and losses, and such elimination is the unique advantage of the proposed data model compared with the traditional data model. Furthermore, the vector tiles can be pre-generated at the server-side. Therefore, this loss in efficiency of generating tiles is acceptable.

Fig 19. E-map based on vector tiles at the 1:10,000 scale based on the traditional data model.

<https://doi.org/10.1371/journal.pone.0176387.g019>

Rendering efficiency comparison. Table 2 tile types (raster tiles, vector tiles based on the traditional model and vector tiles based on the proposed model). The time of rendering raster tiles can be ignored because raster tiles are images. Due to the complex structure of proposed model, rendering vector tiles based on this model required slightly more time than that required by the traditional model, but this increase in time cost had no effect on the performance of the browsing map.

As this article focuses on proposing a tiled vector data model for geographical features, optimized methods of generating tiles and rendering tiles are beyond the scope of this paper. The running times in Tables 1 and 2 are only reference values.

Fig 20. E-map based on vector tiles at the 1:10,000 scale based on the proposed data model.

<https://doi.org/10.1371/journal.pone.0176387.g020>

Discussion and conclusions

E-maps are widely applied in many fields. Most current applications of E-maps are implemented based on raster tiles or vector tiles that are generated based on the traditional data model. This article summarizes previous literatures and explains why the traditional data model is inappropriate for vector tiles. It then proposes a novel tiled vector data model for geographical features. The proposed data model is superior to traditional methods in eliminating visual discontinuities and conflicts at the tile borders. In addition, we verify the proposed model is theoretically correct and feasible by formula derivation and several experiments. The conclusions can be summarized as follows:

Fig 21. Detailed comparisons of the point features: (a) and (c) are from Fig 19, and (b) and (d) are from Fig 20.

<https://doi.org/10.1371/journal.pone.0176387.g021>

1. This paper proposed a theoretical framework for vector tile technology. We define some operators and definitions for geographical features and map features. Model derivations based on mathematics are used to illustrate the shortcomings of the traditional data model and testify the feasibility and validity of the proposed tiled data model. The proposed tiled data model is an advancement in vector tile technology of E-maps.
2. The proposed model is tested by performing several experiments using ground-truth data in different scales on a desktop platform. The experimental results indicate that the proposed vector data model can solve visual conflicts and discontinuities for all types of geographical features that is the greatest weakness of the traditional data model. The proposed

Fig 22. Detailed comparisons of the linear features: (a) and (c) are from Fig 19, and (b) and (d) are from Fig 20.

<https://doi.org/10.1371/journal.pone.0176387.g022>

Fig 23. Detailed comparisons of the area features: (a) and (c) are from Fig 19, and (b) and (d) are from Fig 20.

<https://doi.org/10.1371/journal.pone.0176387.g023>

model can be used as reference to solve similar problems for E-maps of mobile and web clients.

- Although the efficiency of generating and rendering tiles is lower for the proposed model than for the traditional model, the efficiency is acceptable considering the proposed model's complicated structure. Moreover, the cost of generating tiles can be offset by pre-generating all of the vector tiles before rendering the tiles. For rendering tiles, the time costs for the proposed model and the traditional model are almost equal, although the proposed model has a unique advantage. The efficiency comparisons reveal that the proposed model solves the conflicts problem while sacrificing very little time cost, which is significant for applications of vector tile technology.

This work has a good potential for contributing to enlarge the application scope of E-maps but still suffers from several limitations, which could be addressed in future research. First, the proposed model is only valid for maps with static symbols. The design of map symbols is open, and certain symbols that are cognitively effective for communication but might be realized by special procedures, such as dynamic symbols, which are especially useful for E-maps. Most dynamic symbols are dynamic or Flash images, and their symbol parameters (the length or the width of symbols) cannot be obtained directly. In future work, we will create an extension for the library to support dynamic symbols so that we can obtain the symbol parameter easily. Second, the proposed data model is more complicated than the traditional model, and the data size of vector tiles based on the proposed model is slightly larger. A balance between data transmission and cartographical rendering is critical for implementing the model. We will use "Protocolbuffer Binary Format" (PBF) as the encoding format for vector tiles rather than GeoJSON in future work. PBF is a high-compact format, and files encoded in PBF are much smaller than those encoded in GeoJSON. Using this technology will improve the transmission performance. Finally, real-time interactivity and spatial analysis have not been considered in this article. The geographical features are clipped into several vector fragments and stored in different vector tiles. These vector fragments cannot be used for interaction and spatial analysis, which are two merits of vector data and depend on reconstructing vector fragments and storing entire geographical features. The key to reconstruction is to seek out fragments and obtain the vertex series according to the chained

Fig 24. E-map based on vector tiles at the 1:20,000 scale based on the proposed data model.

<https://doi.org/10.1371/journal.pone.0176387.g024>

Table 1. Time cost analysis of generating tiles for the three models (in seconds).

Features (Tiles)	Raster tiles			Vector tiles based on the traditional model			Vector tiles based on the proposed model		
	Time cost	Visual continuity	Support interactivity	Time cost	Visual continuity	Support interactivity	Time cost	Visual continuity	Support interactivity
97074 (1320)	65.8	Yes	No	3740.78	No	Yes	5601.17	Yes	Yes
5962(42)	0.7	Yes	No	53.478	No	Yes	74.87	Yes	Yes
2098(16)	0.2	Yes	No	22.13	No	Yes	30.15	Yes	Yes

<https://doi.org/10.1371/journal.pone.0176387.t001>

Table 2. Time cost analysis of rendering tiles for the three models (in seconds).

Features (Tiles)	Raster tiles			Vector tiles based on the traditional model			Vector tiles based on the proposed model		
	Time cost	Visual continuity	Support interactivity	Time cost	Visual continuity	Support interactivity	Time cost	Visual continuity	Support interactivity
5962(42)	-	Yes	No	1.49	No	Yes	1.78	Yes	Yes
2098(16)	-	Yes	No	0.51	No	Yes	0.58	Yes	Yes

<https://doi.org/10.1371/journal.pone.0176387.t002>

relationships among vector tiles. These aforementioned limitations are the focus of our future work.

Supporting information

S1 File. Symbol library.

(XML)

S2 File. Experimental dataset.

(ZIP)

Acknowledgments

This study is funded by the National Natural Science Foundation of China (41271453); the Scientific and Technological Leading Talent Fund of National Administration of Surveying, mapping and geo-information (2014); the Wuhan ‘Yellow Crane Excellence’ (Science and Technology) program (2014); and the Doctoral Independent Scientific Research Project in Wuhan University (2012205020211).

Author Contributions

Conceptualization: LL WH.

Data curation: WH HZ.

Formal analysis: WH.

Investigation: WH.

Methodology: LL WH.

Resources: WH.

Software: WH.

Supervision: LL HHZ YL.

Validation: HHZ YL.

Visualization: WH.

Writing – original draft: LL WH.

Writing – review & editing: LL WH.

References

1. Mac Aoidh E, Bertolotto M, Wilson DC. Understanding geospatial interests by visualizing map interaction behavior. *Inf Vis.* 2008; 7: 275–286.

2. Kraak M. The role of the map in a Web-GIS environment. *J Geogr Syst.* 2004; 6: 83–93.
3. Pan B, Crotts JC, Muller B. Developing web-based tourist information tools using Google map. *Inf Commun Technol Tour.* 2007; 2007: 503–512.
4. Zhu X, Zhou C. POI inquiries and data update based on LBS. In: *IEEE International Symposium on Information Engineering and Electronic Commerce, IEEEC'09.* IEEE; 2009. pp. 730–734.
5. Sukaphat S. Creating of mobile search system for traffic inquiry. In: *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed computing, SNPD'09.* IEEE; 2009.
6. Chadil N, Russameesawang A, Keeratiwintakorn P. Real-time tracking management system using GPS, GPRS and Google earth. In: *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008, Vol. 1.* IEEE; 2008. pp. 393–396.
7. Patterson TC. Google Earth as a (not just) geography education tool. *J Geog.* 2007; 106: 145–152.
8. Deng W. WebGIS—A new way for public participation in city planning. In: *GIS2000, Toronto, Ontario;* 2000.
9. Grajales FJ III, Sheps S, Ho K, Novak-Lauscher H, Eysenbach G. Social media: a review and tutorial of applications in medicine and health care. *J Med Internet Res.* 2014; 16: e13. <https://doi.org/10.2196/jmir.2912> PMID: 24518354
10. Anguelov D, Dulong C, Filip D, Frueh C, Lafon S, Lyon R, et al. Google street view: capturing the world at street level. *Computer.* 2010; 43: 32–38.
11. Nossum AS. IndoorTubes a novel design for indoor maps. *Cartogr Geogr Inf Sci.* 2011; 38: 192–200.
12. Guo H, Li X, Wang W, Lv Z, Wu C, Xu W. An event-driven dynamic updating method for 3D geodatabases. *Geo-Spatial Inform Sci.* 2016; 19: 140–147.
13. Lu Z, Guerrero P, Mitra NJ, Steed A, editors. *Open3D: crowd-sourced distributed curation of city models.* In: *Web3D Symposium Proceedings;* 2016.
14. Lu Z, Réhman SU, Chen G, editors. *Webvrgis: Webgis based interactive online 3D virtual community.* In: *2013 International Conference on Virtual Reality and Visualization, ICVRV.* IEEE; 2013.
15. Lv Z, Li X, Zhang B, Wang W, Zhu Y, Hu, et al. Managing big city information based on WebVRGIS. *IEEE Access.* 2016; 4: 407–415.
16. Wenjue J, Yumin C, Jianya G. Implementation of OGC web map service based on web service. *Geo-Spatial Inform Sci.* 2004; 7: 148–152.
17. García Martín R, de Castro Fernández JP, Verdú Pérez E, Verdú Pérez MJ, Regueras Santos LM. An OLS regression model for context-aware tile prefetching in a web map cache. *Int J Geogr Inform Sci.* 2013; 27: 614–632.
18. Batty M, Hudson-Smith A, Milton R, Crooks A. Map mashups, Web 2.0 and the GIS revolution. *Ann GIS.* 2010; 16: 1–13.
19. Přidal MP, Žabička P. Tiles as an approach to on-line publishing of scanned old maps, vedute and other historical documents. *e-Perimetreon.* 2008; 3: 10–21.
20. Maso J, Pomakis K, Julia N. *OpenGIS web map tile service implementation standard.* Open Geospatial Consortium Inc.; 2010. pp. 4–6.
21. Wei ZK, Oh Y-H, Lee J-D, Kim J-H, Park D-S, Lee Y-G, et al. Efficient spatial data transmission in Web-based GIS. In: *Proceedings of the 2nd International Workshop on Web Information and Data Management;* 1999.
22. Coors V, Flick S. Integrating Levels of detail in a Web-based 3D-GIS. In: *Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems;* 1998.
23. Tu S, He X, Li X, Ratcliff JJ. A systematic approach to reduction of user-perceived response time for GIS web services. In: *Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems;* 2001.
24. Yang C, Wong DW, Yang R, Kafatos M, Li Q. Performance-improving techniques in web-based GIS. *Int J Geogr Inf Sci.* 2005; 19: 319–342.
25. Edsall R, Andrienko G, Andrienko N, Buttenfield B. Interactive maps for exploring spatial data. In: *Madden M, editor. ASPRS Manual of GIS.* Bethesda: American Society for Photogrammetry and Remote Sensing; 2008.
26. Li X, Di L, Han W, Zhao P, Dadi U. Sharing geoscience algorithms in a web service-oriented environment (GRASS GIS example). *Comput Geosci.* 2010; 36: 1060–1068.
27. Wartell ZJ, Kang E, Wasilewski A, Ribarsky W, Faust NL. Rendering vector data over global, multi-resolution 3D terrain. *GVU Center. Tech Rep.* 2003.

28. Zhou M, Chen J, Gong J. A virtual globe-based vector data model: quaternary quadrangle vector tile model. *Int J Digit Earth*. 2016; 9: 230–251.
29. Kang Y, Kim K, Kim Y. Probability-based tile pre-fetching and cache replacement algorithms for web geographical information systems. In: *East European Conference on Advances in Databases and Information Systems*. Berlin Heidelberg: Springer; 2001. pp. 127–140.
30. Huang H, Li Y, Gartner G, Wang Y. An SVG-based method to support spatial analysis in XML/GML/SVG-based WebGIS. *Int J Geogr Inf Sci*. 2011; 25: 1561–1574.
31. Thomas JJ. *Illuminating the path: [the Research and Development agenda for visual analytics]*. IEEE Computer Society; 2005.
32. Andrienko GL, Andrienko NV. Interactive maps for visual data exploration. *Int J Geogr Inf Sci*. 1999; 13: 355–374.
33. Poplin A. How user-friendly are online interactive maps? Survey based on experiments with heterogeneous users. *Cartogr Geogr Inf Sci*. 2015; 42: 358–376.
34. Yang B. A multi-resolution model of vector map data for rapid transmission over the internet. *Comput Geosci*. 2005; 31: 569–578.
35. Han H, Tao V, Wu H. Progressive vector data transmission. In: *Proceedings of the 6th AGILE, Lyon, France; 2003*. pp. 103–113.
36. Bertolotto M, Egenhofer MJ. Progressive transmission of vector map data over the World Wide Web. *GeoInformatica*. 2001; 5: 345–373.
37. Paiva AC, Rodrigues da Silva E, Leite FL Jr, de Souza Baptista C. A multiresolution approach for internet gis applications. In: *Proceedings from the 15th International Workshop on Database and Expert Systems Applications; 2004*. pp. 809–813.
38. Huang Y, Su L. A fast algorithm for constructing multi-resolution representations of linear features to support the progressive transmission of Vector map data. *GISci Remote Sens*. 2006; 43: 197–217.
39. Liu P, Li X, Liu W, Ai T. Fourier-based multi-scale representation and progressive transmission of cartographic curves on the internet. *Cartogr Geogr Inform Sci*. 2016; 43: 454–468.
40. Yang B, Li Q. Efficient compression of vector data map based on a clustering model. *Geo-Spatial Inf Sci*. 2009; 12: 13–17.
41. Jang B, Lee S, Lim S, Kwon K. Progressive vector compression for high-accuracy vector map data. *Int J Geogr Inf Sci*. 2014; 28: 763–779.
42. Shekhar S, Huang Y, Djughash J, Zhou C. Vector map compression: a clustering approach. In: *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*. ACM; 2002. pp. 74–80.
43. Li W, Song M, Zhou B, Cao K, Gao S. Performance improvement techniques for geospatial web services in a cyberinfrastructure environment—a case study with a disaster management portal. *Comput Environ Urban Syst*. 2015; 54: 314–325.
44. Butler H, Daly M, Doyle A, Gillies S, Schaub T, Schmidt C. The GeoJSON format specification. *Rapport Technique*. 2008: 67.
45. Draft, NWIP. ISO 19117rev. Draft New work item proposal, Revision of ISO 19117: 2005. Gleason, Greasley, editor. DGIWG; 2006.
46. Ormsby T, Napoleon E, Burke R, Groessl C, Bowden L. *Getting to know ArcGIS desktop*. Redlands: ESRI Press; 2010.
47. Robinson AC, Pezanowski S, Troedson S, Bianchetti R, Blanford J, Stevens J, et al. Symbol Store: sharing map symbols for emergency management. *Cartogr Geogr Inform Sci*. 2013; 40: 415–426.
48. Yue S, Yang J, Chen M, Lu G, Zhu A, Wen Y. A function-based linear map symbol building and rendering method using shader language. *Int J Geogr Inf Sci*. 2016; 30: 143–167.
49. Ting-Hua AI. A displacement of building cluster based on field analysis. *Review Acta Geodaetica et Cartographica Sinica*. 2004; 1: 016.
50. Goodchild MF, Yuan M, Cova TJ. Towards a general theory of geographic representation in GIS. *Int J Geogr Inf Sci*. 2007; 21: 239–260.
51. Lin LI, Zhu H, Biao HE, Wang H, Qiu J. Cartographic model for topographic maps based on algebraic structure. *Review Acta Geodaetica et Cartographica Sinica*. 2011; 40: 373–372.
52. Tang AY, Adams TM, Usery EL. A spatial data model design for feature-based geographical information systems. *Int J Geogr Inf Syst*. 1996; 10: 643–659.
53. Liang Y-D, Barsky BA. A new concept and method for line clipping. *ACM Trans Graph*. 1984; 3: 1–22.
54. Zhang J, Zhu Y. A method based on graphic entity for visualizing complex map symbols on the web. *Cartogr Geogr Inform Sci*. 2014; 42: 44–53.