The British Computer Society 2017.
All rights reserved. For permissions, please e-mail: journals.permissions@oup.com
Advance Access publication on 11 December 2017 doi:10.1093/comjnl/bxx117

Visual Exploration of 3D Geospatial Networks in a Virtual Reality Environment

MENG-JIA ZHANG[1], KANG ZHANG[2,3], JIE LI[1,4*] AND YI-NA LI[5]

[1]*School of Computer Software, Tianjin University, Peiyang Park Campus: No.135 Yaguan Road, Haihe Education Park, Tianjin 300350, P.R. China*
[2]*Department of Computer Science, The University of Texas at Dallas, 800 W. Campbell Road, Richardson, TX 75080-3021, USA*
[3]*Faculty of Information Technology, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau, P.R. China*
[4]*State Key Laboratory for Novel Software Technology, Nanjing University, 22 Hankou Rd, Gulou Qu, Nanjing, Jiangsu 210008, P.R. China*
[5]*School of Management, University of Science and Technology of China, No.96, JinZhai Road Baohe District, Hefei, Anhui 230026, P.R. China*
[*]*Corresponding author: jie.li@tju.edu.cn*

Classic geospatial network visualization tends to limit itself to 2D representation by organizing edges and nodes on a 2D map or the external surface of a traditional 3D globe model. Visual clutters and occlusions due to edge crossings and node-edge overlaps make efficient and effective exploration of geospatial networks a challenge. This paper proposes an interactive visualization approach for the intuitive exploration of geospatial networks inside a spherical virtual reality environment. To reduce visual clutter and reveal network patterns, we also propose a parameterized 5-step 3D edge-bundling algorithm and a set of techniques to avoid collision of network edges with the viewpoint. Our spherical interaction and 3D edge-bundling approach have been implemented in an Oculus Rift VR system. We demonstrate the usefulness of our approach with two case studies on real-world network data and usability experiments.

Keywords: geospatial network visualization; head mounted display; immersive interaction; virtual reality

Received 3 June 2017; revised 25 September 2017; editorial decision 3 November 2017
Handling editor: Suchi Bhandarkar

1. INTRODUCTION

Information visualization has been widely used to map big data to a screen space, with interactive control over how the data are presented. Most of the information visualization techniques are restricted to 2D mappings. Visual cluttering is inevitable when projecting a 3D visualization onto a 2D display. Switching between the global overview or focus on a specific piece of information, the observer tends to spend much time on changing the viewpoint to find the best position using a traditional interactive tool, such as keyboard or mouse.

With the recent advances of consumer-level stereoscopic and head mounted displays (HMD) technology, the effectiveness of 3D information visualization could be significantly improved. A growing number of techniques exploring 3D visualization have since been proposed [1]. According to a recent study, 3D visualization is superior to 2D visualization when applied to representing spatiotemporal data [2]. The extra dimension in a 3D visualization helps users to observe the holistic and individual features of data in different aspects of time, space and attributes.

Virtual reality (VR) is a simulated 3D environment, providing the user with the experience of being present in that environment. Scientific visualization has been a driving force behind the rapid development of the VR technology. In addition, the embracing cyber-world breaks the real-world boundaries and gives the user the immersive views. One of the key characteristics of VR is a new form of human–computer interaction, beyond mouse, keyboard or even touch screen. As a

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

result, users can interact with full visual immersion to look around the virtual world in the manner they would in the real world [3].

Being present in a VR environment, the user could potentially make full sense of the context information. In addition, VR devices could easily detect users' observation movements and behavior, which provide clues for reasoning about their intention. Hence, the viewpoint could be automatically changed to improve the effectiveness and efficiency of observation.

Recent research on immersive 3D graph visualization, however, tends to limit the graph layout within a rectangular display area [4, 5]. In an immersive VR environment, the visible scene is arranged omni-directionally around the user's viewpoint, raising the issue of inefficient and uncomfortable observation due to the rotation range of the user's neck. The benefits of stereoscopic graph visualization have been demonstrated in multiple user studies [6–9].

Geospatial network data typically describe associations between pairs of known geographic locations. The associations include movement of goods, knowledge [10], people [11] and airline or network traffic, etc. Traditional techniques that represent such associations include flow density maps [12], origin-destination (OD) matrices [13] and OD Maps [14]. Notably, the principle of direct flow mapping is to plot each trajectory as a line from the origin to the destination on a plane projected by the geographic space. However, too many trajectories make it difficult for users to recognize the directions of flows due to the overwhelming amounts of clutters and occlusions.

This paper presents a technique for interactive exploration of geospatial network datasets within a VR environment. First, different from the existing observation methods, we apply a new approach to observing network visualization inside a virtual sphere, accommodating human observing habits. Second, the positions of the network nodes are assumed unchangeable as they carry useful geographical information. Finally, based on the position and direction of network edges, we propose a 3D edge-bundling method to reduce occlusions and to avoid blind areas resulted from the edges being too close to the central viewpoint.

The primary contributions of our work are:

(1) Effective VR interaction design and a 3D edge-bundling method for enhancing clarity of the network skeleton;
(2) An algorithm that routes edges to bypass the viewpoint within the spherical VR environment; and
(3) Two case studies that demonstrate the feasibility of the above approaches, and an empirical experiment on the effectiveness of the approaches.

The remainder of this paper is structured as follows. Section 2 reviews the previous work on VR and stereoscopic techniques. Section 3 presents the immersive observation design of geospatial network visualization, a VR interaction paradigm for observation, followed by Section 4 on a 5-step 3D edge-bundling method. Section 5 demonstrates the case studies on two datasets and Section 6 concludes the paper.

2. RELATED WORK

A number of geospatial network visualization approaches have recently been proposed, such as flow density maps [12], OD matrices [13] and OD Maps [14]. They visualize a great amount of various geospatial data, such as migration [11], commuting behavior [15] and movements of knowledge [10]. For example, existing flow maps use lines to connect the flow sources and the destinations on 2D map, causing unnecessary clutter [16]. Recent research has found that 3D visualization techniques are superior to 2D visualization techniques when applied to spatiotemporal data [2]. 3D techniques have long been attempted in information visualization [17, 18].

Most traditional 3D information visualizations are developed on 2D displays. Different from monocular depth cues used in those existing 3D information visualizations [19], stereoscopic 3D displays offer a significant depth cue: binocular disparity, enabling the brain to extract depth information from the 2D retinal images in stereopsis. Moreover, immersive VR (IVR) and stereoscopic techniques in scientific visualization have demonstrated their usefulness in applications [20–22], such as medical imaging [23], Geographic Information System [24] and volume data [25]. Architectural walkthrough systems were the first attempt to apply VR as a visualization tool [26]. Recent research on immersive graph visualization has proved the potential of VR as a powerful tool for visualizing large network datasets [5].

Compared with the Euclidean space, a spherical surface offers a natural fisheye effect, which is a popular technique for visualizing large graphs [27]. The fisheye effect enlarges the focus point and shows other portions of the image with successively fewer details. Munzner discusses the advantages of hyperbolic layout over the fisheye lens [28]. With the user at the center, immersive environments can provide a natural model with a spherical paradigm where the user's viewpoint is equidistant to most of the display area. A majority of graph layout algorithms work in the standard Euclidean space instead of a spherical space [5, 28–32]. Munzner explores large graphs in a spherical space with nodes/edges inside the sphere [28], but the edges are straight and would go through the user's viewpoint in the center. Wu and Takatsuka [32] visualize multivariate network by grouping nodes of similar attributes to adjacent areas on a sphere, but the nodes' positions carry no useful information. Kwon *et al.* [5] introduce the immersive graph visualization, but the edges are routed outside a transparent sphere.

Graphs tend to suffer from much more visual clutters when laid in a 3D space than in a 2D space, making the exploration and data analysis inefficient. Therefore, one challenge of graph visualization is how to effectively manage a large number of edges, thus avoiding visual clutter. Edge bundling techniques are the common methods applied to graph visualization [17, 18, 33, 34]

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

and also in parallel coordinates and flow map visualization, etc [26, 35, 36]. Recent studies classify 2D edge-bundling algorithms into three categories [37]: geometry-based [38], force-directed [39] and image-based [40] techniques. A challenge for any 3D edge-bundling techniques is the additional dimension, which cannot be handled directly by existing 2D edge-bundling algorithms. Holten *et al.* [39, 41] present an approach for bundling geometrically close edges, with the objective of reducing edge clutter for general graphs, which is generalizable to 3D. However, this algorithm is not scalable for graphs of large numbers of edges. Pupyrev *et al.* [38] introduce another class of edge-bundling methods which implicitly bundle edges together by routing them through nearby, static control points. The efficiency of this method depends much on the density of the control points. When extended to 3D, many more possible routes make the algorithm highly complex.

Lambert *et al.* [42] present a generalization of a novel 2D edge-bundling method to reduce cluttering in a 3D representation [43], but the edges are routed on the surface of a sphere. Zielasko *et al.* [44] introduce an interactive 3D clustering-based edge-bundling algorithm and demonstrate it with a graph of simulating the function of a macaque brain. It is parallelized and scales with the graph's size in terms of rendering time. Nevertheless, the authors stated that the clusters are drawn independently, which could be an issue with directed graphs and antiparallel bundles.

Different from any existing work, we draw edges inside a sphere in true 3D to reduce unnecessary clutter and apply an approach to adjust the positions of control points to prevent network edges from being too close to the user's viewpoint.

3. MODEL AND INTERACTION

We proposed a spherical immersive model (SIM) that contains a hollow sphere [45] and sets the viewpoint at the center of the sphere. The world map texture could be horizontally flipped before being mapped on the internal surface of the sphere. The camera is set at the center of the sphere as the viewpoint. With the HMD on, the user can experience being present in the center of the hollow sphere. Through arbitrary head movement, users are given a continuous view of the interior globe space.

Wearing the HMD device, the user is isolated from the outside world and unable to see his/her hands. The use of traditional interactive devices, such as mouse and keyboard, becomes awkward.

To design a comfortable and effective observation mechanism, understanding the range of human's neck motion (cervical spine) plays an essential role. In particular, when seated, the user tends to look around in a small range without moving his/her torso. The movement range through which a person can actively move the joint without any assistance is referred as the active range of motion. Several studies [5, 46] illustrate that the vertical range of motion is around 50° upwards (extension) and

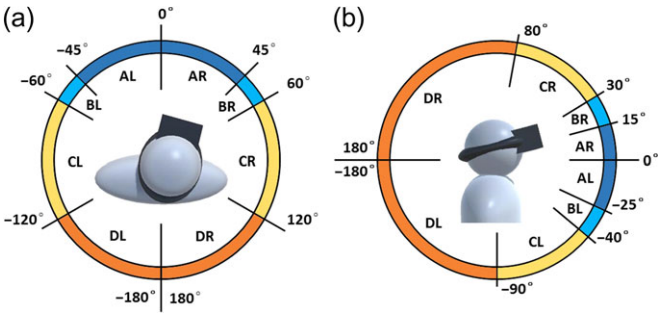

FIGURE 1. Magnetization as a function of applied field.

60° downwards (flexion), and the horizontal range to the left and right is around 80° each. Through preliminary experiments, we found that users were uncomfortable near or over these limits. We therefore propose the following interaction design.

When observing the geospatial data inside the virtual spherical space (i.e. a hollow sphere), the user would move the head by rotating the neck to change the viewing direction (VD). The horizontal and vertical components of the VD are denoted as VD_h and VD_v, respectively. As shown in Fig. 1a, the horizontal area between −45° (left) and 45° (right) is marked as AL and AR (the minus sign represents the left side), which is called the *horizontal observing area* (HOA). When VD_h is in the HOA, the hollow sphere remains stationary and the user changes the VD by slightly turning the head. When VD_h is outside the HOA, we assume that the user intends to observe the area outside the observing area. Therefore, to reduce the horizontal turning range of the user's head, the sphere begins to rotate in the opposite direction of head rotation at a certain angular velocity that is proportional to the rotation angle of VD.

We denote the angular velocity of the sphere as $\mathbb{W}$ and the horizontal and vertical components of $\mathbb{W}$ as $\mathbb{W}_h$ and $\mathbb{W}_v$, respectively. As shown in Fig. 1b, when VD_h arrives at the BR or BL, $\mathbb{W}_h$ increases at a small rate as VD_h increases. Hence, the user can sense the sphere's movement and easily adjust the VD to an ideal angle. Moreover, when VD_h arrives at the CR or CL, we assume that the user intends to observe an area far away from the observing area. Hence $\mathbb{W}_h$ increases at a high rate so that the user can observe the destination area immediately.

To minimize dizziness and discomfort potentially caused by the rising rate of the sphere's rotation, we keep the rotation speed the same as the maximum speed generated in CL or CR when the horizontal turning range reaches at DR or DL. This is because the user seldom rotates his/her head beyond DR and DL.

4. 3D EDGE BUNDLING

Effective visualization of complex or dense node-link diagrams is a great challenge, as the display tends to suffer from

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

visual clutter induced by node overlaps and edge crossings. Moreover, fixed node positions make this issue even more prominent, as the positions are represented by their geographical coordinates. Edge bundling techniques [35, 39, 41] are frequently used to solve this problem. The main steps of these techniques include computing each edge's curvature. To route edges along control points, these techniques draw curves such as Bezier curves, B-splines or Catmull–Rom splines.

Different from any traditional spatial edge-bundling techniques that route edges around the surface of a sphere [43], we display edges completely inside a sphere. According to the distribution characteristics of data, we visually cluster edges based on their positions and directions, thus improving the legibility and visibility when viewed from the inside of the sphere. In addition, to avoid edges passing through the user's viewpoint, we optimize the edge-bundling algorithm by adjusting the positions of control points and routing edges to bypass the viewpoint. We start with an unbundled and undirected spatial graph $G = (V; E)$ with a set of nodes V and a set of edges E. Our approach consists of five major steps:

(1) Temporarily removing all short edges under a predefined length threshold l, to avoid their interference with the decisions in the subsequent steps, similar to our previous 2D edge-bundling approach [47].
(2) Grouping edges based on their positions and directions (Section 4.1).
(3) Computing the average of all the edges in each group to obtain a 'central' edge (see Section 4.2), and determining a set of control points along this edge. We can then use the Bézier curve algorithm to route all the edges in the same group along these control points.
(4) Adjusting the positions of the control points to avoid blind areas resulted from the edges being too close to the user's viewpoint (Section 4.3).
(5) Adding back the temporarily removed short edges (Section 4.4).

The five steps, detailed in the following subsections, work together to bundle the edges, aiming at maximizing the legibility and visibility when viewed from the inside of the sphere.

4.1. Grouping edges

Edges are grouped according to thezier directions and positions. In other words, the edges in the same area are grouped if their directions are also similar, as detailed below.

To group edges according to similar directions, we use vector $\boldsymbol{V}$ to represent each edge denoted by its start point $S(x, y, z)$ and endpoint $E(a, b, c)$, i.e. $\boldsymbol{V} = (a - x,\ b - y,\ c - z)$. We first randomly select an edge, $\boldsymbol{V_1} = (a_1 - x_1,\ b_1 - y_1,\ c_1 - z_1)$, to be a *referential edge*, and define two spherical domains around S_1 and E_1:

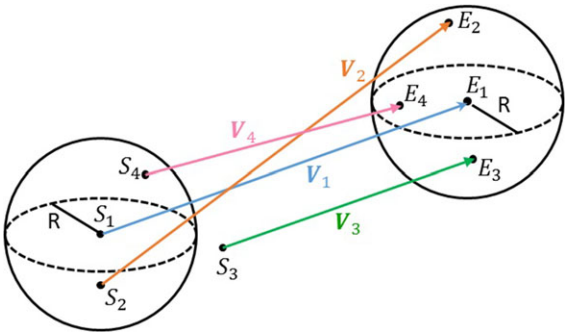

FIGURE 2. Grouping edges according to their positions and directions: Edges $\boldsymbol{V_1}$, $\boldsymbol{V_2}$ and $\boldsymbol{V_4}$ considered in the same area, and $\boldsymbol{V_1}$, $\boldsymbol{V_3}$ and $\boldsymbol{V_4}$ in similar directions.

$$\begin{cases}(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \le R^2 \\ (x - a_1)^2 + (y - b_1)^2 + (z - c_1)^2 \le R^2,\end{cases} \tag{1}$$

where R is a predefined distance threshold. The edge whose start and endpoints fall in the two respective domains is considered within the same area as the referential edge. In Fig. 2, only edges starting at S_2 and S_4 are in the same area with the edge starting at S_1. We have performed experiments using five different referential edges, but generated slightly different bundled results. The differences are around the middle part of the edges, due to different orders of edge rendering. They therefore do not make any difference in terms of visualization objectives.

The dot product of two edges $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$

$$\boldsymbol{V}_1 \cdot \boldsymbol{V}_2 = \|\boldsymbol{V}_1\|\|\boldsymbol{V}_2\|\cos\theta \quad \theta \in [0°, 180°], \tag{2}$$

where θ is the angle between $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, could be used to identify whether the two edges have similar directions [33, 48].

This follows that:

$$\cos\theta = \frac{\boldsymbol{V}_1 \cdot \boldsymbol{V}_2}{\|\boldsymbol{V}_1\|\|\boldsymbol{V}_2\|} = \boldsymbol{v}_1 \cdot \boldsymbol{v}_2, \tag{3}$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the unit vectors of $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$. Its computation complexity is much less than computing the angle directly. We predefine an angle threshold α and if two edges' unit vectors, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, satisfy:

$$\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 > \cos\alpha, \tag{4}$$

we consider them to be in similar directions. In Fig. 2, only $\boldsymbol{V}_2$ and $\boldsymbol{V}_3$ have similar directions with $\boldsymbol{V}_1$.

Edges in the same group are bundled if their directions are similar. In the example of Fig. 2, only the edge starting at S_4 is grouped with the referential edge starting at S_1. We can therefore cluster any set of edges into several groups.

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

4.2. Generating control points and routing edges

Having grouped all the edges, we now compute the average of all the edges in each group, and call the averaged edge the *central edge*. The central edge provides the routing direction and reference for the other edges in the group.

Assume a group of n edges, $\boldsymbol{V}_i(vx_i, vy_i, vz)(i = 1, 2, \cdots, n)$ at start points of $S_i(x_i, y_i, z_i)(i = 1, 2, \cdots, n)$. We obtain the group's central edge (i.e. vector $\boldsymbol{V}(vx, vy, vz)$ starting at $S(sx, sy, sz)$) by averaging all the edges' vectors and start points in the group:

$$\boldsymbol{V} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{V}_i = \frac{1}{n}\sum_{i=1}^{n}(vx_i, vy, vz_i) = \frac{1}{n}\left(\sum_{i}^{n}vx_i, \sum_{i}^{n}xy_i, \sum_{i}^{n}vz_i\right) \tag{5}$$

$$\mathrm{S} = \frac{1}{n}\sum_{i=1}^{n}S_i = \frac{1}{n}\sum_{i=1}^{n}(x_i, y_i, z_i) = \frac{1}{n}\left(\sum_{i}^{n}x_i, \sum_{i}^{n}y_i, \sum_{i}^{n}z_i\right). \tag{6}$$

The next step identifies control points on the central edge and uses them to route all the other edges in each group. Assume that we wish to select two intermediate control points $C_1(cx_1, cy_1, cz_1)$ *and* $C_2(cx_2, cy_2, cz_2)$, we obtain the points at the 1/4 and 3/4 of the central edge using the following equations:

$$C_1(cx_1, cy_1, cz_1) = S(sx, sy, sz) + \frac{1}{4}\boldsymbol{V}(vx, vy, vz) \tag{7}$$

$$C_2(cx_2, cy_2, cz_2) = S(sx, sy, sz) + \frac{3}{4}\boldsymbol{V}(vx, vy, vz) \tag{8}$$

The values of 1/4 and 3/4 are determined through several experiments with various pairs of values. Having identified the control points for every group, we route edges along their corresponding control points using the Bézier Curve algorithm.

4.3. Avoiding collision of viewpoint with control points

Recall that the user's viewpoint is located in the center of the sphere when observing. Therefore, edges going through the central part of the sphere would lead to uncomfortable observation. This subsection introduces the main steps for routing edges away from the central part of the sphere by properly positioning control points.

We consider the central part of the sphere as *safe observation area* (SOA), a much smaller sphere of radius r. Figure 3a and b shows the changes between the original bundled edges and edges after adjusting the positions of control points of edges in SOA (shown as a white sphere). One may notice that many edges passing through SOA (thus invisible in Fig. 3a) have now been rerouted to bypass SOA (some become visible in Fig. 3b).

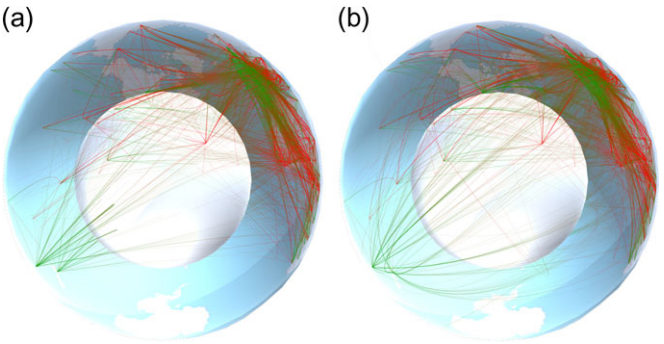

FIGURE 3. The results of avoiding collision of viewpoint with control points: (**a**), before adjusting; (**b**) after adjusting.

To keep edges away from the user's viewpoint, we need to allocate the control points outside SOA. This implies that the central edge of every group should not pass through SOA.

In Fig. 4a, the distance between the central edge and the user's viewpoint is d, denoted by OD. We consider the ratio of r and d:

$$\text{rate} = \frac{r}{d} \tag{9}$$

If rate < 1, the central edge of the group is outside SOA and thus no need to adjust the positions of the control points. If rate ≥ 1, the central edge of the group is inside SOA, edges of this group would be too close to the user's viewpoint after being routed along the control points. We have developed and experimented with several approaches to address this issue, and select one that is the fastest while producing the best results.

As demonstrated in Fig. 4a, we move the central edge along $\overrightarrow{OD}$ to make it tangent to the surface of SOA. Then, we extend the lines $\overrightarrow{C_1C_1'}$ and $\overrightarrow{C_2C_2'}$ to intersect the central edge at C_1' and C_2'. As the points O, C_1, C_2, C_1' and C_2' are on the same plane, we can obtain the coordinates of C_1' and C_2' via the following:

$$C_1' = \text{rate} * C_1 \tag{10}$$

$$C_2' = \text{rate} * C_2. \tag{11}$$

Simply using C_1' and C_2' as the control points would generate unsatisfactory results. Many curved lines are far away from the viewpoint and cannot be observed. We therefore choose the positions at C_1'' and C_2'', on the same line passing C_1' and C_2', as shown in Fig 4a. We mark C as the center of the new central edge, also as the center of the line $C_1''C_2''$ and the line $C_1'C_2'$. The relationship between the lengths of $C_1'C_2'$ and $C_1''C_2''$ is shown as the following:

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

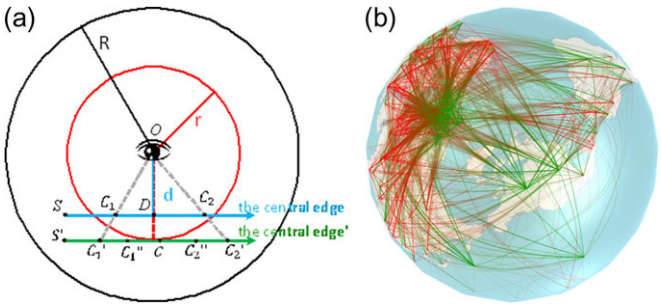

FIGURE 4. Illustration and results of adjusting the positions of control points $\overrightarrow{C_1C_1'}$ and $\overrightarrow{C_2C_2'}$, where C_1 and C_2 are on 1/4 and 3/4 of the central edge and C_1'' and C_2'' are on $\frac{\text{rate}-1}{2*\text{rate}}$ and $\frac{\text{rate}+1}{2*\text{rate}}$ of the line $C_1'C_2'$.

$$\frac{\left\|\overrightarrow{C_1'C_2'}\right\|}{\left\|\overrightarrow{C_1''C_2''}\right\|} = \text{rate} \tag{12}$$

C_1'' and C_2'' can then be obtained via:

$$C_1'' = C_1' + \frac{\text{rate}-1}{2*\text{rate}} * \overrightarrow{C_1'C_2'} \tag{13}$$

$$C_2'' = C_1' + \frac{\text{rate}+1}{2*\text{rate}} * \overrightarrow{C_1'C_2'} \tag{14}$$

The result in Fig. 4b shows that the no curved edge is outside SIM. However, after adjusting the positions of the control points, some points may still go outside of SIM or inside SOA. To solve this problem, we forcefully put those points on the boundary of SIM or SOA along the negative or positive side of the viewing direction. As Fig. 5a shows B_1 and B_2 are the original endpoints of Bezier curves. B_1 is outside of SIM, so we move B_1 to the B_1', which is the intersection of OB_1 and SIM. On the other hand, B_2 is inside the SOA, so we move B_2 to B_2', which is the intersection of the extension of OB_2 and SOA. Figure 5b shows the result.

4.4. Handling short edges

In practice, edge lengths usually vary greatly and many edges are short. Short edges disturb the bundling process. We therefore filter them out before the second step and then render them as straight edges in the final and fifth step.

5. TWO CASE STUDIES

We have implemented the aforementioned SIM and 3D edge-bundling algorithm using Unity in an Oculus Rift VR environment. This section reports our experiences in using the implemented system to explore two real-world datasets and demonstrates the effectiveness of our approaches. The two datasets are the Global Flights Dataset (http://openflights.org/data.html) and the UNHCR (The UN Refugee Agency) Refugee Dataset (http://data.un.org/) [16].

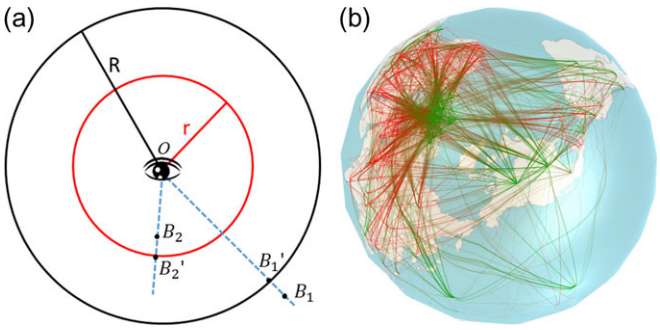

FIGURE 5. Illustration and results of adjusting the positions of the endpoints of Bezier curve, from B_1 to B_1' and B_2 to B_2'.

5.1. Global flights dataset

As Fig. 6 shows, different colors represent different airlines and small black points represent airports. Users can highlight specific airlines they wish to view via interactions. It is thus easy to observe the backbone of each airline's flight routes in each area. The results in Fig. 6 clearly demonstrate that our 3D edge bundling is able to reveal the backbone structures of the flight patterns.

Figure 7 demonstrates the effects of three parameters: length threshold l (Section 4), angle threshold α (Section 4.1) and distance threshold R (Section 4.1). When l is set to 2.0 (Fig. 7a) or above, i.e. longer than the edges in this dataset, then no edge is bundled. When decreasing l (Fig. 7a–c and j), more edges are bundled and main bundles become distinguishable. Moreover, when α is set to 0 (Fig. 7d), the results show that no edge is bundled since no pair of edges satisfies conditions (1) and (4) (Section 3.1) simultaneously. When increasing α (Fig. 7d–f and j), a bundle may contain more edges and the skeleton becomes clearer. When R is set to 0 (Fig. 7 g), no pair of edges satisfies condition (1), thus no edge is bundled. When increasing R (Fig. 7g–j), more edges may add to the same bundle and the backbone structure of the graph becomes more traceable. For example, one can clearly see five main routes with Ryanair flights.

5.2. UNHCR refugee dataset

Figure 8a–g visualizes the results of 2008 UNHCR refugee dataset with global and local views. The opacity of every edge encodes the number of migrants from one city to another. More refugees moving from one city to another, more clearly the edges show.

Figure 8 g and h shows the influence of r, i.e. the radius of SOA. With the increasing r (Fig. 8 g– h), the edges bend

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

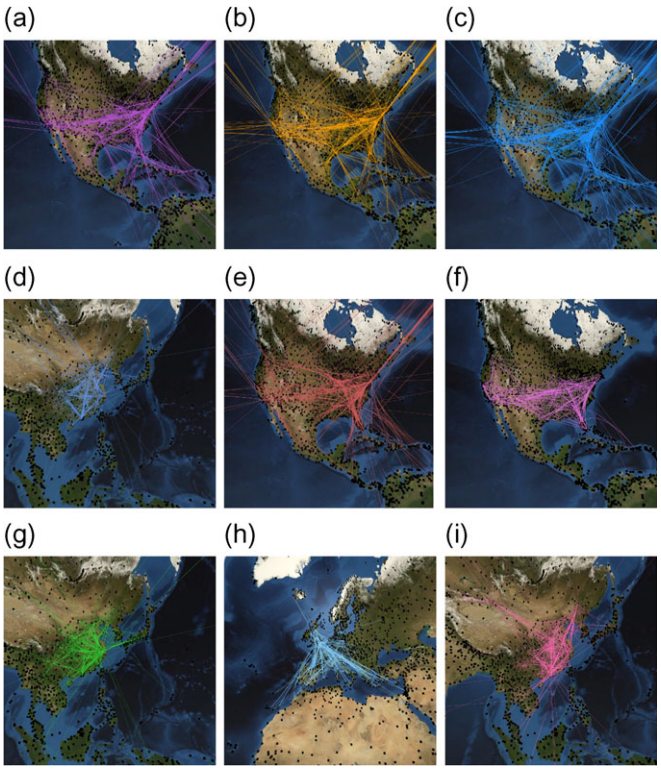

FIGURE 6. Bundling results for nine different airlines' datasets. (**a**) US Airways, (**b**) United Airlines, (**c**) American Airline, (**d**) Air China, (**e**) Delta Air Lines, (**f**) Southwest Airlines, (**g**) China Eastern Airlines, (**h**) EasyJet, (**i**) China Southern Airlines.

toward the internal surface of the sphere to keep away from the viewpoint. Therefore, the edges bypass, instead of passing through, the viewpoint.

Figure 8a–c is the zoom-in views inside the black dashed box of Fig. 8 g. Similarly, Fig. 8d–f is the zoom-in views of the area inside the black dashed box of Fig 8 h. Obviously, the zoom-in view with field of view (FOV) = 89 (Fig. 8 f) provides a larger range of view than that with FOV = 65 (Fig. 8c). The user views a wider range of areas and sees more edges when FOV = 89. On the other hand, the zoom-in view with FOV = 65 (Fig. 8c) offers more detailed information than that with FOV = 89 (Fig. 8f). The user can adjust the FOV of the HMD to obtain an ideal zoom-in view.

Figure 8a–c and g layouts straight lines without bundling and thus show serious clutter. In the figures, the red side of an edge represents the source of the movement and green side represents the destination. Figure 8d–f and h demonstrates the effectiveness of 3D edge bundling and reveal the main moving trend without clutter. Figure 8h shows that western and 'developed' countries are completely green, and other countries are mostly red. People tend to move from Africa and Asia to Europe, Oceania and North America in 2008.

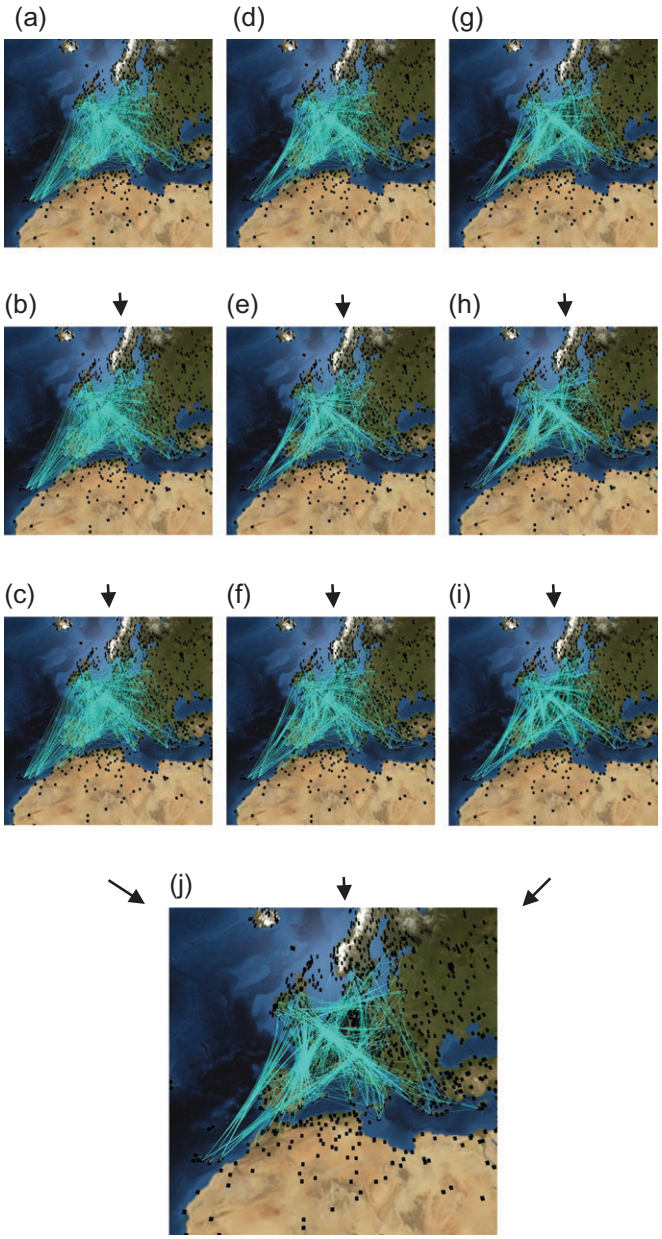

FIGURE 7. Bundled results for Ryanair flights dataset: (**a**)–(**c**) and (**j**) under different length threshold values of l; (**d**)–(**f**) and (j) under different angle threshold values of α; (**g**)–(**i**) and (j) under different distance threshold values of R. (**a**) $l = 2.0$, $\alpha = 50$, $R = 0.6$ (**b**) $l = 1.3$, $\alpha = 50$, $R = 0.6$ (**c**) $l = 0.9$, $\alpha = 50$, $R = 0.6$ (**d**) $l = 0.5$, $\alpha = 0$, $R = 0.6$ (**e**) $l = 0.5$, $\alpha = 15$, $R = 0.6$ (**f**) $l = 0.5$, $\alpha = 30$, $R = 0.6$ (**g**) $l = 0.5$, $\alpha = 50$, $R = 0$, (**h**) $l = 0.5$, $\alpha = 50$, $R = 0.2$, (**i**) $l = 0.5$, $\alpha = 50$, $R = 0.4$, (**j**) $l = 0.5$, $\alpha = 50$, $R = 0.6$.

6. USABILITY STUDY

As discussed earlier, 3D edge bundling should be able to reduce visual clutter, enhance the clarity of the network skeleton and thus provide an optimal exploration experience. We have

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

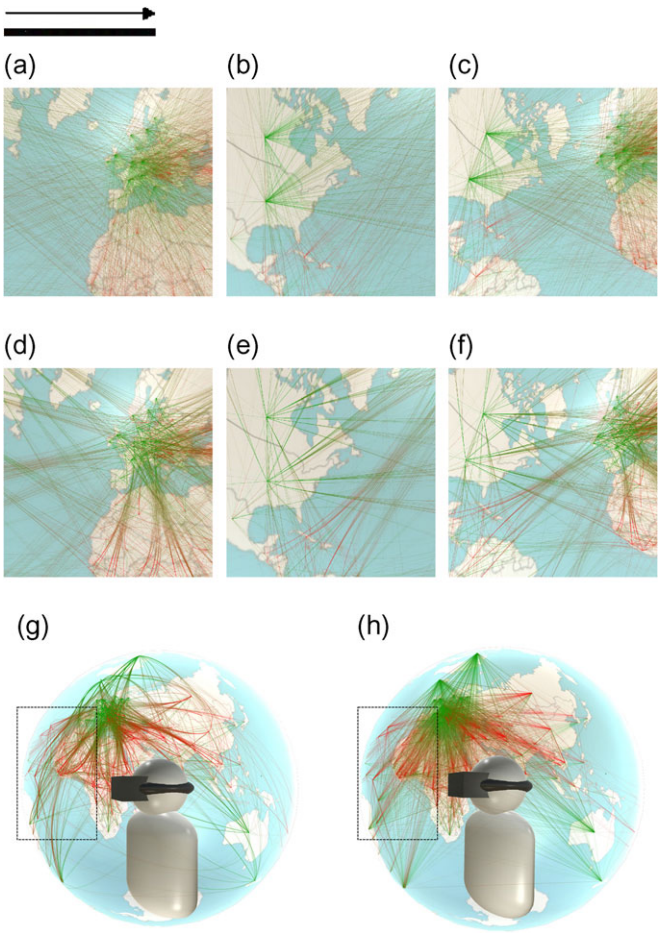

FIGURE 8. Zoom-in views (**a**)–(**f**) of UNHCR refugee dataset; The effects of bundled edges with $r = 3.5$ (**d**)–(**f**) and unbundled edges (**a**)–(**c**); the observation view with FOV = 89 (c and f) and FOV = 65 (**a**, **b**, **d** and **e**). Global views of UNHCR refugee dataset with the effects of bundled edges with $r = 3.5$ (**h**) and unbundled edges (**g**). (**a**) unbundled, FOV = 65, (**b**) unbundled, FOV = 65, (**c**) unbundled, FOV = 89, (**d**) $r = 3.5$, bundled, FOV = 65, (**e**) $r = 3.5$, bundled, FOV = 65 and (**f**) $r = 3.5$, bundled, FOV = 89.

conducted a user study to evaluate the usability and effectiveness of our 3D edge-bundling algorithm in an Oculus Rift VR environment.

6.1. Experiment design

We design between-group experiments in two settings (spherical immersive visualization with and without 3D edge bundling) with seven tasks to complete. Both settings use 2008 *UNHCR refugee dataset*, the same rendering method, and the same world map and labeling. The continents (Asia, Europe, Africa, the North America, the South America and Oceania) are labeled on the world map, and their boundaries are highlighted with black curves. One hundred eighty-six countries are labeled using numbers of the same font and size and labels are displayed toward users. All participants use a HMD, i.e. an Oculus Rift DK2.

To obtain the empirical evidences of the effects with and without 3D edge bundling, we collected the *task completion time* and *accuracy*. The task completion time refers to the duration spent on completing each task, excluding the time on listening to the descriptions and questions. The accuracy pertains to the percentage of correct answers.

6.2. Participants

We recruited 48 participants (24 males and 24 females, between 20 and 30 years of age) for our experiment. Those participants have at least finished 12 years of education, majoring in various fields in Tianjin University, China. All participants are qualified in their abilities of color discrimination. They are randomly assigned to Group C1 (spherical immersive visualization with 3D edge bundling) and Group C2 (spherical immersive visualization without 3D edge bundling). We balanced the genders of participants in the two groups.

6.3. Apparatus and implementation

First, we calibrate HMD for each subject. The distance between the pupils of two eyes, the distance between the HMD lens and the cornea and the tightness of the headband are adjusted appropriately to make sure the labels are perceived clearly, even for the participants with vision correction.

Each subject was required to stand in the middle of a 2 m^2 space and wear Oculus Rift DK2 (https://www.oculus.com), which has a 1920 × 1080 pixels (split to 960 × 1080 pixels per eye) organic light-emitting diode panel with a 75 Hz refresh rate. AMD Radeon HD 7479 graphics card was used to render the visualization. The positional tracking was enabled for both settings. To be able to move freely, the participants were required to stand up while completing the entire experiment.

6.4. Procedure

We started with a brief introduction to the basic principle of HMD, the immersive environment and the idea of 3D edge bundling (including the meanings of the colors, labels and lines). Participants were told to complete each task by answering the corresponding question based on what they see, rather than their knowledge.

Then we trained participants to operate the equipment, and get used to the immersive environment by performing the following initial tasks:

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

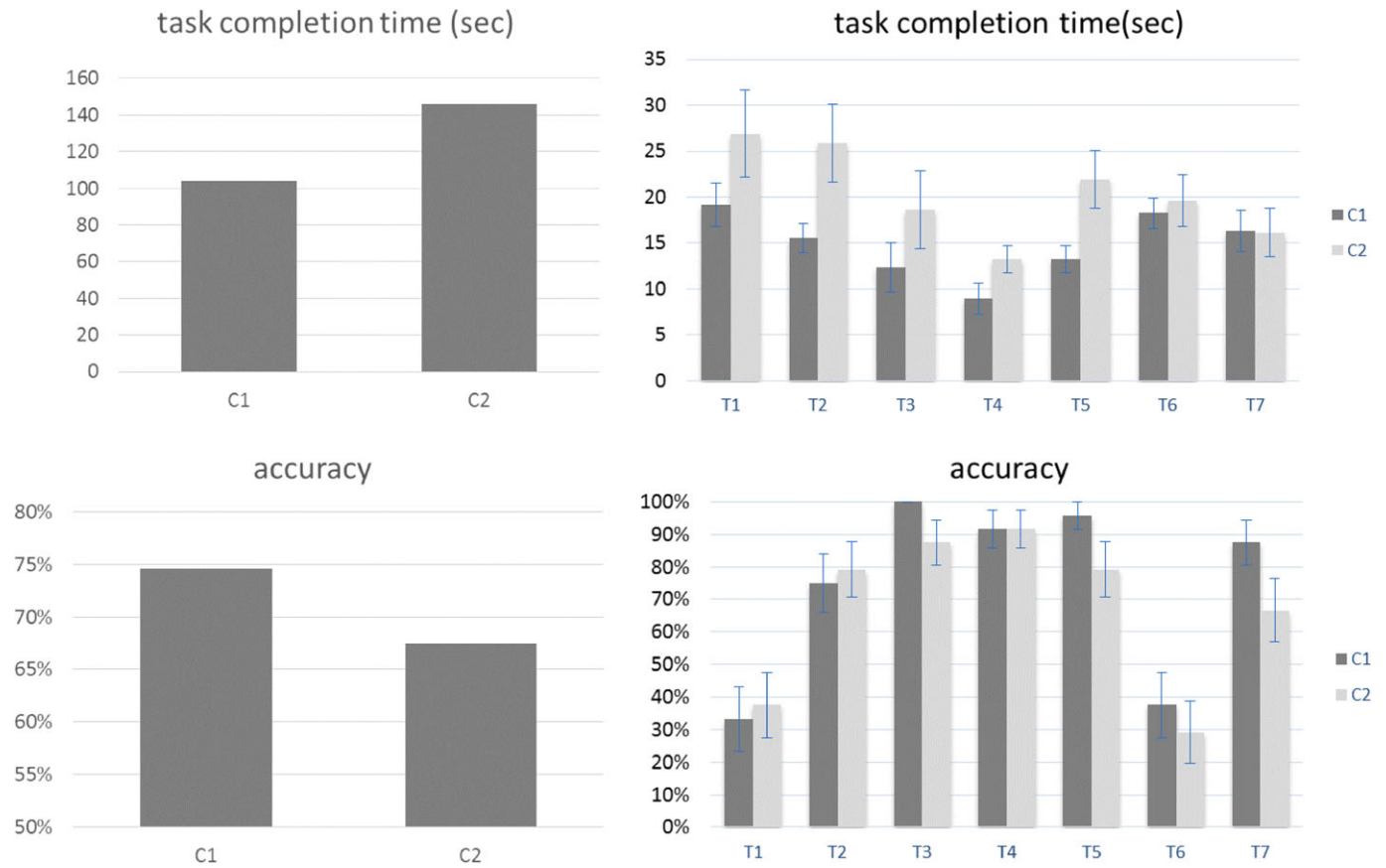

FIGURE 9. Results of the experiments for the two settings (C1 and C2). (**a**) Overall task completion times. (**b**) Individual task completion times. (**c**) Overall accuracy. (**d**) Individual task accuracy.

- Where is country numbered 167, which represents China?
- Does China have more migrants entering or exiting?
- Which country has the largest number of migrants to China? (Indicate the number for that country).

During training, participants were clearly informed of whether they answered the question correctly. We explained in detail and provided necessary support for participants who failed to answer questions or encountered any difficulties in performing the tasks. We also used similar questions referring to different countries to test the participants who failed to answer the above questions. By performing those tasks, participants learned the role of visualization and the operations of Oculus Rift DK2.

The participants were then asked to orally answer seven questions as the formal experiment, that are read out by the same experimental staff in the same order. We did not randomize the order of the questions since we attempt to control the potential influences of the GUI. The participants were not told whether they answered each question correctly during the experiment. In addition, they were required to return to the initial standing position, and their viewing direction in the virtual worlds was reset to eliminate undesirable effect of the previous task.

The seven questions are:

T1: Which continent has the largest group of immigrants to the USA.
T2: Which continent has the largest group of migrants to New Zealand.
T3: Which continent is the most popular destination for Mongolia migrants.
T4: Which continent is the most popular destination for Russia migrants.
T5: Except Asia, which continent is the most popular destination for Asia migrants?
T6: Except America, which continent has the largest group of migrants to North America?
T7: Which continent has the largest group of migrants to Oceania (Except Oceania).

The seven questions essentially cover how the information visualization on migrants is used. T1–T4 inquire the information between countries, while T5–T7 involve information on

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

continents. In particular, T1 and T6 test the effect of long-edge tracking. In contrast, the other questions focus on short distances.

Before each question was asked, the participants were shown the labels pertaining to the locations of the continents and countries which would be mentioned in the tasks. Having ensured that the participants could find the destinations, we then hide the labels of the countries to avoid occlusion.

6.5. Results and analysis

Overall, the 3D edge-bundling approach helps to effectively improve the task completion time.

6.5.1. *Task completion time*

The Group C1 participants spent 103.88 s (SD = 32.90) on average to answer each question, by contrast, Group C2 spent 146.0 s (SD = 94.44). An independent *T*-test analysis indicates that the Group C1 participants spent significantly less time to answer each question than those in Group C2 ($F = 14.422$, $P = 0.045$) (Fig. 9a), indicating the advantage of the 3D edge-bundling approach in speeding up users' information processing. The task completion durations on each question of the two groups are consistent with the pattern of the average durations, except for T7 (Fig. 9b). In particular, the Group C1 participants spent significantly less time to answer T2 ($F = 5.211$, $P = 0.027$) and T5 ($F = 6.403$, $P = 0.015$) than those in Group C2.

6.5.2. *Accuracy*

The Group C1 participants answered 74.40% questions (SD = 11.90) correctly and the Group C2 participants did 67.26% (SD = 19.97) (Fig. 9c). However, the differences in accuracy between the two groups are insignificant ($F = 2.79$, $P = 0.14$).

The Group C1 participants have significantly high accuracy in answering T5 ($P = 0.019$) and T7 ($P = 0.014$) than those in Group C2 (Fig. 9d).

The accuracy patterns of the questions are inconsistent (as shown in Fig. 9d). The patterns of T1 and T2 conflict to those of T3 and T5–T7. The participants in both groups have almost the same accuracy on T4. However, the differences between the two groups on T1, T2 and T4 are insignificant. It might suggest the need for further investigation in the influence of the 3D edge-bundling approach on accuracy.

6.5.3. *The influences of tracking distances*

We also evaluated the influences of tracking distances on users' task completion time and accuracy, to examine boundary conditions of the influences of the 3D edge-bundling approach. The result of logistic regression confirms that for the participants in both groups could process information in tracking long distance more accurately ($P = 0.000$) than tracking short ones. An independent *T*-test indicates that the Group C1 participants spent less time on average to complete tasks when the tracking distance is long ($P = 0.002$). However, the Group C2 participants did not show any time difference in completing tasks in tracking long/short distances. We did not find the interactive effect of tracking distance and groups on completion time or accuracy.

7. CONCLUSION

With the popularization of head-mounted displays and other stereoscopic devices, effective immersive exploration and visualization approaches become crucial. This paper has presented a SIM, a novel visual approach to the exploration of geospatial network datasets, specifically designed for IVR interaction. To reveal the backbone structure of any geospatial network structure, we have presented a parameterizable 5-step 3D edge-bundling approach. For effective exploration of 3D visualization, we optimize the algorithm to prevent edges from being too close to the viewpoint. Our experiments with the global fights dataset and UNHCR refugee dataset have shown that our approach is effective for visual exploration of large 3D graphs. We acknowledge the difficulty for experimental comparison with existing 3D edge-bundling approaches, none of which is in an immersive sphere environment. They would have to be reimplemented in our SIM environment for a meaningful comparison.

Although we have developed several VR interaction methods for effective large-scale exploration, there are many more interaction techniques to be explored and developed. To enhance the depth cue in the 3D visualization, we have also experimented with different color shades for the lines of the same group. The visualization results are somehow improved. We plan to further explore various color-coding strategies and conduct usability studies to evaluate their effectiveness. An important part of our future work is to further improve our algorithms based on the usability evaluation results.

This paper has assumed the viewpoint to be positioned in the center of the sphere. In fact, the viewpoint could be repositioned flexibly, e.g. upon a user interaction (request) or adaptive automatically to the distance or the size of the object being viewed. This, together with its usability study, will be part of further extension of our work. We also plan to apply our work to visualize and interact with other types of data.

FUNDING

National NSFC project (Grant nos. 61602340, 61572348 and 71702080); National High-tech R&D Program (863 Grant no. 215AA020506) and Humanity and Social Sciences Fund of Ministry of Education (17YJC630071).

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

REFERENCES

[1] Tominski, C., Schumann, H., Andrienko, G. and Andrienko, N. (2012) Stacking-based visualization of trajectory attribute data. *IEEE. Trans. Vis. Comput. Graph.*, **18**(12), 2565–2574.

[2] Kjellin, A., Pettersson, L.W., Seipel, S. and Lind, M. (2010) Evaluating 2D and 3D visualizations of spatiotemporal information. *ACM. Trans. Appl. Percept.*, **7**(3), 19.

[3] Desai, P.R., Desai, P.N., Ajmera, K.D. and Mehta, K. (2014) A review paper on oculus rift-a virtual reality headset. *arXiv preprint arXiv:1408.1173*.

[4] Kwon, O.H., Muelder, C., Lee, K. and Ma, K.L. (2015) Spherical Layout and Rendering Methods for Immersive graph visualization. In *Proceedings of 2015 IEEE PacificVis*, Hangzhou, China, 14–17 April, 63–67.

[5] Kwon, O.H., Muelder, C., Lee, K. and Ma, K.L. (2016) A study of layout, rendering, and interaction methods for immersive graph visualization. *IEEE. Trans. Vis. Comput. Graph.*, **22** (7), 1802–1815.

[6] Alper, B., Hollerer, T., Kuchera-Morin, J. and Forbes, A. (2011) Stereoscopic highlighting: 2D graph visualization on stereo displays. *IEEE. Trans. Vis. Comput. Graph.*, **17**(12), 2325–2333.

[7] Greffard, N., Picarougne, F. and Kuntz, P. (2014) Beyond the Classical Monoscopic 3D in Graph Analytics: An Experimental Study of the Impact of Stereoscopy. In *Proceedings of IEEE VIS International Workshop on 3DVis*, Paris, France, 9-9 November, 19–24.

[8] Ware, C. and Mitchell, P. (2005) Reevaluating Stereo and Motion Cues for Visualizing Graphs in Three Dimensions. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization*, A Coruña, Spain, 26–28 Aug, 51–58.

[9] Ware, C. and Mitchell, P. (2008) Visualizing graphs in three dimensions. *ACM Trans Appl Percept (TAP)*, **5**(1), 2.

[10] Paci, R. and Usai, S. (2009) Knowledge flows across European regions. *Ann. Reg. Sci.*, **43**(3), 669–690.

[11] Tobler, W.R. (1987) Experiments in migration mapping by computer. *Am Cartogr*, **14**(2), 155–163.

[12] Rae, A. (2009) From spatial interaction data to spatial interaction information? Geovisualisation and spatial structures of migration from the 2001 UK census. *Comput. Environ. Urban. Syst.*, **33**(3), 161–178.

[13] Voorhees, A.M. (2013) A general theory of traffic movement. *Transportation*, **40**(6), 1105.

[14] Wood, J., Dykes, J. and Slingsby, A. (2010) Visualisation of origins, destinations and flows with OD maps. *Cartogr. J.*, **47** (2), 117–129.

[15] Chiricota, Y., Melancon, G., Quang, T.T.P. and Tissandier, P. (2008) Visual Exploration of (French) Commuter Networks. In *Proceedings of Geovisualization of Dynamics, Movement, and Change AGILE'08 Satellite Workshop,* May, Spain.

[16] Boyandin, I., Bertini, E. and Lalanne, D. (2010) Using Flow Maps to Explore Migrations Over Time. In *Proceedings of Geospatial Visual Analytics Workshop in conjunction with the 13th AGILE International Conference on Geographic Information Science*, Guimarães, Portugal, 11–14, May, 2(3).

[17] Brath, R. (2014) 3D Infovis is Here to Stay: Deal with It. In *Proceedings of IEEE VIS International Workshop on 3DVis*, Paris, France, 9-9 November, 25–31.

[18] Teyseyre, A.R. and Campo, M.R. (2009) An overview of 3D software visualization. *IEEE. Trans. Vis. Comput. Graph.*, **15** (1), 87–105.

[19] Ware, C. (2012) *Information Visualization: Perception for Design*. Elsevier, San Francisco, CA.

[20] Brooks, F.P. (1999) What's real about virtual reality? *IEEE. Comput. Graph. Appl.*, **19**(6), 16–27.

[21] Bryson, S. (1996) Virtual reality in scientific visualization. *Commun. ACM.*, **39**(5), 62–71.

[22] Van Dam, A., Forsberg, A.S., Laidlaw, D.H., LaViola, J.J. and Simpson, R.M. (2000) Immersive VR for scientific visualization: A progress report. *IEEE. Comput. Graph. Appl.*, **20**(6), 26–52.

[23] Hänel, C., Weyers, B., Hentschel, B. and Kuhlen, T.W. (2014) Interactive Volume Rendering for Immersive Virtual Environments. In *Proceedings of IEEE VIS International Workshop on 3DVis, Paris*, France, 9-9 November, 73–74.

[24] Bennett, R., Zielinski, D.J. and Kopper, R. (2014) Comparison of Interactive Environments for the Archaeological Exploration of 3D Landscape Data. In *Proceedings of IEEE VIS International Workshop on 3DVis*, Paris, France, 9-9 November, 67–71.

[25] Mirhosseini, K., Sun, Q., Gurijala, K.C., Laha, B. and Kaufman, A.E. (2014) Benefits of 3D Immersion for Virtual Colonoscopy. In *Proceedings of IEEE VIS International Workshop on 3DVis*, Paris, France, 9-9 November, 75–79.

[26] Brooks, F.P. Jr (1987) Walkthrough—a Dynamic Graphics System for Simulating Virtual Buildings. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, North Carolina, USA, 23-24 October, 9–21.

[27] Herman, I., Melançon, G. and Marshall, M.S. (2000) Graph visualization and navigation in information visualization: A survey. *IEEE. Trans. Vis. Comput. Graph.*, **6**(1), 24–43.

[28] Munzner, T. (1998) Exploring large graphs in 3D hyperbolic space. *IEEE. Comput. Graph. Appl.*, **18**(4), 18–23.

[29] Hughes, T., Hyun, Y. and Liberles, D.A. (2004) Visualizing very large phylogenetic trees in three dimensional hyperbolic space. *BMC Bioinform*, **5**(1), 1.

[30] Kobourov, S.G. and Wampler, K. (2005) Non-Eeuclidean spring embedders. *IEEE. Trans. Vis. Comput. Graph.*, **11**(6), 757–767.

[31] Sprenger, T.C., Gross, M.H., Eggenberger, A. and Kaufmann, M. (1997) A Framework for Physically-Based Information Visualization. In *Proceedings of EuroGraphics Workshop on Visualization in Scientific Computing*, Boulogne-sur-Mer, France, 28–30 April, 71–83.

[32] Wu, Y. and Takatsuka, M. (2006) Visualizing Multivariate Network on the Surface of a Sphere. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, Tokyo, Japan, 1–3 February, 77–83.

[33] Spiegel, M. and Lipschutz, S. (2009) *Schaum's Outline of Vector Analysis* (2nd edn). McGraw Hill Professional, Columbus, OH.

[34] Ware, C. and Franck, G. (1996) Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans Graph*, **15**(2), 121–140.

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018

[35] Ersoy, O., Hurter, C., Paulovich, F., Cantareiro, G. and Telea, A. (2011) Skeleton-based edge bundling for graph visualization. *IEEE. Trans. Vis. Comput. Graph.*, **17**(12), 2364–2373.

[36] Lipson, M. and Lipschutz, S. (2009) *Schaum's Outline of Linear Algebra (eBook), 4th edn*. McGraw-Hill, Columbus, OH.

[37] Zhou, H., Xu, P., Yuan, X. and Qu, H. (2013) Edge bundling in information visualization. *Tsinghua Sci Technol.*, **18**(2), 145–156.

[38] Pupyrev, S., Nachmanson, L., Bereg, S. and Holroyd, A.E. (2011) Edge Routing with Ordered Bundles. *Computational Geometry*, **52**, 18–33.

[39] Holten, D. and Van Wijk, J.J. (2009) Force–directed edge bundling for graph visualization. *Comput Graph Forum*, **28**(3), 983–990. Wiley Online Library.

[40] Telea, A. and Ersoy, O. (2010) Image–Based Edge Bundles: Simplified Visualization of Large Graphs. *Comput Graph Forum*, **29**(3), 843–852. Wiley Online Library.

[41] Holten, D. (2006) Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE. Trans. Vis. Comput. Graph.*, **12**(5), 741–748.

[42] Lambert, A., Bourqui, R. and Auber, D. (2010b) Winding roads: Routing edges into bundles. *Comput Graph Forum*, **29** (3), 853–862. Wiley Online Library.

[43] Lambert, A., Bourqui, R. and Auber, D. (2010a) 3D Edge Bundling for Geographical Data Visualization. In *Proceedings of the 14th International Conference on Information Visualisation*, London, United Kingdom, 26–29 July, 329–335.

[44] Zielasko, D., Weyers, B., Hentschel, B. and Torsten, W.K. (2016) Interactive 3D force-directed edge bundling. *Comput Graph Forum*, **35**(3), 51–60.

[45] Zhang, M.J., Li, J. and Zhang, K. (2016) An Immersive Approach to the Visual Exploration of Geospatial Network Datasets. In *Proceedings of the 15th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI'2016)*, Zhuhai, China, 3-4 December, 381–390.

[46] Youdas, J.W., Garrett, T.R., Suman, V.J., Bogard, C.L., Hallman, H.O. and Carey, J.R. (1992) Normal range of motion of the cervical spine: an initial goniometric study. *Phys. Ther.*, **72**(11), 770–780.

[47] Cai, Z., Zhang, K., Hu, D.N. and Zheng, X.J. (2016) Visualizing Large Graphs by Layering and Bundling Graph Edges, Technical Report 02-16, Department of Computer Science, The University of Texas at Dallas.

[48] Borisenko, A.I. (1979) Vector and Tensor Analysis with Applications. Courier Corporation, New York.

Downloaded from https://academic.oup.com/comjnl/article-abstract/61/3/447/4725106
by University of Texas at Dallas user
on 22 April 2018