

# GeoPeels: Deformation-Based Technique for Exploration of Geo-Referenced Networks

<p>Alberto Debiasi Fondazione Graphitech Via alla Cascata 56/C 38123, Trento, Italy alberto.debiasi@graphitech.it</p>	<p>Bruno Simões Fondazione Graphitech Via alla Cascata 56/C 38123, Trento, Italy bruno.simoies@graphitech.it</p>	<p>Raffaele De Amicis Fondazione Graphitech Via alla Cascata 56/C 38123, Trento, Italy raffaele.de.amicis@graphitech.it</p>
---	--	---

## ABSTRACT

Spatial relations between geographical entities are typically visualized as a node-link diagram that is depicted over a geographical layout. The node-link diagram representation is well-suitable for tasks that require the analysis of the topology and properties of a spatial graph because it is more intuitive with respect to the matrix representation. However, the readability of node-link diagrams is, especially for large sized graphs, one of the issues studied in literature. In this paper we describe an interactive technique for virtual globes, called *GeoPeels*, which uses spatial deformations to reveal spatial information otherwise hidden behind the geographical layout. The deformed geometries are rendered around the globe, therefore, occupying space that otherwise would be unused. Additionally, our approach facilitates graph-related tasks by reducing the overall number of actions required to visualize and understand the spatial relations. The evaluation of our approach is based on the analysis of the graphical performance and on a comparison between virtual globes and 2D flat maps.

## Keywords

3D geovisualization, edge congestion, interactive visualization, graph layout, geographical distortion, geo-referenced data.

## 1 INTRODUCTION

Large-scale map visualization systems of geographical datasets retain their traditional role of providing insight into geospatial patterns and relations, see Figure 1. On the one hand, they provide geospatial contextualization for each data point. On the other hand, they can be used to enrich data points with graphical elements like straight lines, circular arcs and curves, to give emphasis to specific spatial relations.

Virtual globes are considered mature visualization systems for the exploration of geo-referenced networks. In the past, they were used by Cox et al. [9] to visualize Internet traffic over the NFN/ANSnet backbone and by Munzenr et al. [27] to map the Mbone routers. In climate research domain, the analysis of climate data as complex network theory is considered a powerful approach [13]. In such context the globe is commonly used [35, 29]. Globes are also used in frameworks for management and monitoring of large scale distributed systems [23].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

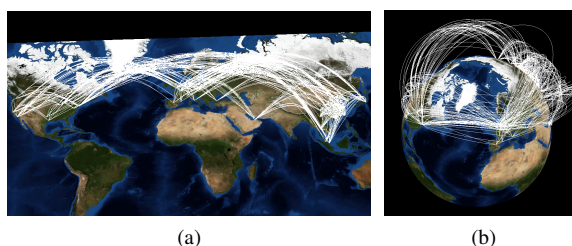


Figure 1: Geo-referenced network over imposed (a) on a flat map, and (b) on a virtual globe.

In literature, many approaches have been proposed to improve the visualization during network analysis. Brandes et al. [5] minimize clutter at important nodes by adjusting the angle of edges. A few techniques [22, 10] were provided to bundle edges around the globe. Alper et al. [1] present a visual technique to better understand the relations of the network. The globe is deformed to place the nodes connected with each other to appear closer. Debiasi et al. [11] present 3DArcLens, a focus+context technique to reveal areas occluded by undesired edges.

It is possible, through visual exploration, to interactively browse different portions of a dataset to better understand the data. On the one hand, visual exploration helps with hypothesis confirmation. On the other hand, it facilitates the discover of unexpected patterns, which can raise new questions, without the requirement

of having a complete knowledge about the data [37]. However, in virtual globes, we can visualize only one of the hemispheres, i.e. the globe occludes the nodes and the links that can be considered important for exploratory tasks.

In this paper, we propose *GeoPeels*, a distortion-based technique for the visualization of spatial relations in geographical data, where such interactive techniques remain unexplored. In particular, our technique is optimized for graphs-related tasks, such as explore node connections. Our approach is based on a virtual globe that automatically deforms the terrain surface to avoid the occlusion of key data nodes and links, i.e. based on the camera view position, the focus area, and relevant nodes and links. The terrain deformation is automatically adjusted to avoid rendering the geometries outside the screen view. In this way relevant nodes and connections are always visible.

In the exploration of spatial relations, our technique can be used in combination to zoom and pan, which by themselves are considered tedious, potentially disorienting, and often ineffective [16, 17]. Furthermore, *GeoPeels* does not require the operator to define visual node-link attributes, e.g. we do not use attributes like color and size of a node. Hence, these unused node-link attributes can be used to encode additional information. Consequently, our technique can be combined with other works that make use of such visual attributes, for example, to reduce the visual clutter.

This paper is structured as follows. Section 2 describes previous related work. Section 3 introduces our approach. Section 4 explains the list of parameters as well how to use them. Section 5 describes a use case that uses our technique. Section 6 provides an analysis of the graphical performance. Section 7 describes the advantages of our approach and makes a comparison between virtual globes and 2D flat maps. We conclude with some remarks about the usability of the proposed technique and future work in Section 8.

## 2 RELATED WORK

**Visualizations of Geo-referenced Networks.** There are two main visualization techniques to represent geo-referenced networks: the matrix-based representation and the node-link layout. Although the first visualization is free from occlusion, the latter allows layouts to encode the geographical information of nodes. Geographic node-link layouts can be classified in three categories. In the first category, both data and maps are depicted over a flat surface. The first system to automatically generate 2D visualization with arrows imposed on a map was presented by Tobler et. al. [33]. ArcMap [9] applies the same concept on a 3D space, where 3D arcs connect nodes on flat surface. In the second category everything is projected over a spherical (or ellipsoidal)

surface [9, 27, 6, 23]; and in the third category we have a combination of the previous ideas depicted in multiple views [4, 8]. VisLink [8] links two or more maps using straight lines. Flowstrates [4] uses a map for the origins and another one for the destinations to represent spatio-temporal data. For an overview of visualizations used to depict geo-referenced networks see [31, 40].

**Navigation to elements occluded by geographical surface.** In all aforementioned categories the nodes and links may have visibility problems that may affect the exploration of the network. In 3D visualizations, where the geographical surface is a volume, graph elements may be occluded. Different focus+context techniques can be implemented to alleviate this problem. Magic Lenses [2] are lenses that are able to modify the rendering of objects seen through them. The use of these lenses was also extended to 3D graphics [39, 32]. Cignoni et al. [7] presented MagicSphere, a magic lens metaphor to volumetric lenses. Cut-away viewing [30, 12], is another approach to reveal what is occluded by volumetric objects. Exploded views have been investigated in the context of architectural visualization by Niedauer et al. [28] and for volume visualization by McGuffin et al. [25]. Their approach facilitates interactive browsing of volume data partitioned into several layers. Tory et al. [36] provide a framework for combined visualization of context and detail in volumetric data sets. For a comprehensive overview of techniques for the occlusion managements see the survey by Elmquist et al. [15].

**Navigation to Off-screen elements.** Large graphs can not be entirely visualized in the screen view. However, many approaches are available in literature to overcome such problem. Pan in 2D layouts and the camera view relocation in 3D layouts are traditional tools to reveal off-screen elements. More recent techniques addressed such problem, both in terms of their visibility [18] and in navigating to them [19]. They provide visual indication of locations of off-screen objects using a simple representation at the viewport's edge. Also, they provide mechanisms to select items of interest, and to navigate to them quickly. With Bring Neighbors Lens [34], when a user moves the lens (using the mouse) towards a vertex, the neighboring vertices should be brought to the lens in a continuous fashion. To guarantee the visibility of multiple focus regions, Elmquist et al. [14] provide a distortion technique that folds the intervening space. The context is preserved of the folds themselves.

**Map Projections.** The performance of the exploration of geo-referenced networks is affected by the projection that is used. The major problem of map projection is the distortion. For example, the Mercator projection, which was originally designed to display compass bearings for sea travel, depicts distortions for large areas. Azimuthal projections allows the directions to

be correct from the center of the projection to any other point. Moreover, great circles of the globe are projected to straight lines. The azimuthal orthographic projection is mainly used for illustration purposes, since it shows the Earth as seen from space infinitely far away. The availability of high-quality imagery of the Earth and other planets as seen from orbit, has caused a recent revival of interest for this projection. Virtual globes use such projection enabling a wide variety of interactive pan and zoom operations. In Azimuthal equidistant projection the distances and the bearing from the center of the map to any point are correct. The projection is frequently used to show air-route distances. The Lambert azimuthal equal-area projection preserves areas while simultaneously maintaining a true direction from the center. Recommended to the European Commission for statistical analysis and display. The latter two projections cover the entire world. Van Wijk presented Myriahedral projections [38], a class of methods for mapping the earth on a polyhedron with a large number of faces. The polyhedron is then cut open and unfolded. In such way the maps are (almost) conformal and conserve areas, but they are characterized by a large number of discontinuities. In literature, different works change dynamically the map projection [20, 3, 21]. For navigation purposes, Möser et. al. [26] propose a technique that blends geo and egocentric perspectives to combine the advantages of both viewpoints while maintaining the ease of perception of a single view. Lorenz et. al. [24] implement an interactive multi-perspective views for 3D geographic environments. It automatically configures the deformation in a view-dependent way to maintain the multi-perspective view in an interactive environment.

### 3 PROPOSED APPROACH

In this work we define a deformation-based technique that enables the interactive exploration of spatial relations between geographical entities; in particular to facilitate graph-related tasks, e.g. exploring the node connections.

We propose a technique that enriches the idea of a virtual globe with the following features: a list of *important nodes*, i.e. nodes that are connected to nodes *in focus* should be always occlusion-free. We define *in focus* the list of nodes that are inside the user-defined area. Remaining nodes are said to be *in context*, see Figure 2.

In order to relocate nodes *in focus* that are occluded by the globe's hemisphere, without losing their background map context, we defined the concept of *slice*, which corresponds to a portion of terrain that is occluded by the globe and that contains at least one *important node*.

In the next sections, we use the terminology *deformed slice* to describe the geometry of a slice that was peeled

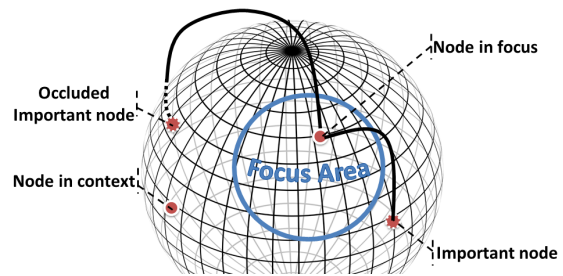


Figure 2: Nodes are classified accordingly with their position and the position of their connected nodes.

from the globe surface – which is represented as a continuous concave surface with respect to the camera view. The concept of *deformed slice* can be demonstrated with an orange: we can peel an orange to reveal its hidden parts, see Figure 3.

This approach keeps the globe's properties, i.e. nodes in the hidden hemisphere that are not *important* are intrinsically filtered out, even if located in the deformed slice.

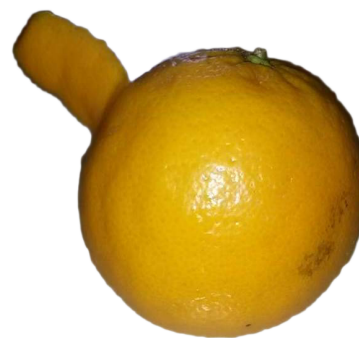


Figure 3: The orange peel is the metaphor that inspired this work.

The process to create and deform a slice is transparent to the operator. It entails the creation of *slices* for *important nodes* that are not visible. The first node that is added to a *slice* is called its *primary node*. Slices are created only if an occluded *important node* cannot be moved into an existent slice, see Figure 4. This strategy ensures that nodes are never occluded by the globe and the number of *deformed slices* is minimized.

The shape of each *slice* is defined as a rectangular geometry with the *primary node* located at the center with respect to the horizontal axis, and at a fixed position with respect to the vertical axis. During the interactive exploration, the *deformed slice* is bended to ensure the visibility of the geometry within the screen view, see Figure 5.

The focus area is represented as a circle in screen space with a radius  $r$  in pixels. The circular element remains still during the user navigation, e.g. tilt, pan and zoom,

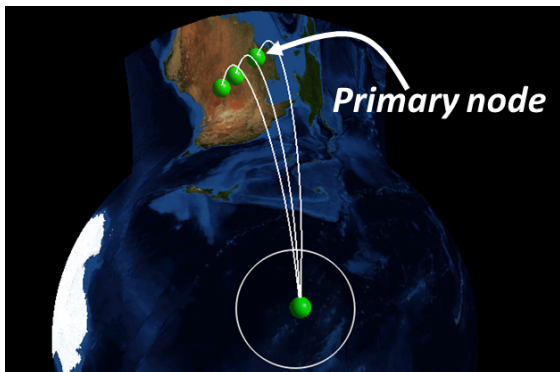


Figure 4: In this example the *deformed slice* has three nodes: the *primary node* and two nodes assigned to the existing *slice*.

hence revealing the hidden information that was initially covered by the circle. In fact, the circle is snapped to the terrain to improve the system usability; the camera movement affects the circle position in screen coordinates but not its geographical position.

*GeoPeels* was developed on top of *NASA WorldWind*, an open source virtual globe that approximates the earth as an ellipsoid. The proposed technique makes use of a reference system, which is described in Subsection 3.1. The overall procedure of *GeoPeels* is summarized in Subsection 3.2. In Subsection 3.3, we describe how nodes are relocated if there are multiple *slices* as candidates. Finally, in Subsection 3.4, we describe the function to manage the surface deformation. The parameters introduced in this section are described in Section 4.

### 3.1 Slice Reference System

The algorithm to check whether a node is inside a slice, to relocate a node, and to create a slice is based on a reference system  $R_S$ , specific to a *slice*  $S$ .  $R_S$  is defined by the planes  $P_d$  and  $P_\alpha$ . The first plane,  $P_d$ , is identified by three points: the node  $n$  that corresponds to the point we want to reveal, the center of the earth  $o$ , and  $c$  that is the camera view point  $l$  projected over the globe, see Figure 6(a).

For every point  $p$  on the globe,  $\gamma$  defines the angle between the vector  $lp$  and the normal of the globe surface on  $p$ . If  $\gamma > \pi/2$  then the point  $p$  is visible, otherwise it is occluded by the globe. Thus, we define  $x$  as the point on the great circle, between  $c$  and  $n$  with angle  $\gamma = (\pi/2) + 0.4$ . Consider a second plane,  $P_\alpha$ , to be perpendicular to  $P_d$  and to intersect  $o$  and  $x$ , see Figure 6(b). Then, we have that the position of  $p$  is defined by  $R_S(\alpha, d)$ , where  $d$  is the perpendicular distance between  $p$  and  $P_d$ , and  $\alpha$  is the angle between the plane  $P_\alpha$  and the plane perpendicular to  $P_d$ , which intersects the points  $o$  and  $p$ . See, respectively, Figure 6(c) and Figure 6(d).

### 3.2 GeoPeels

Our approach can be summarized with the following procedure, which is executed every frame before the rendering phase:

**procedure** GEOPEELS

(1) List< Node >  $N := \text{getNodesToRelocate}()$ ;

List< Slice >  $S := \emptyset$ ;

**for each**  $n \in N$  **do**

(3) bool hasSlice := *hasCompatibleSlice*( $n, S$ )

**if** hasSlice = false **then**

(2) Slice  $s := \text{createSlice}(n)$

$S.add(s)$

**end if**

**end for**

**for each**  $n \in N$  **do**

(3) Slice  $s := \text{getChosenCompatibleSlice}(n, S)$

(4) *relocateNode*( $n, s$ )

**end for**

(5) *drawArcs*()

**for each**  $s \in S$  **do**

(4) *deformSlice*( $s$ )

**end for**

**end procedure**

(1) *Retrieve the list of important nodes to relocate.*

The list of important nodes to relocate, is computed by checking their angle  $\gamma$ ; a node is added to the list if  $\gamma < (\pi/2) + 0.4$ . The list considers the occluded nodes together with the nodes that are placed near to the occluded hemisphere. Afterwards, the list is sorted in descending order accordingly to each node distance to the camera view projected over the globe. This operation is useful to minimize the number of *slices*.

(2) *Slice creation procedure.* Let  $R_S(\alpha', 0)$  be the position of a primary node, and  $w$  and  $k$  the parameters to define respectively the width and the length of its *slice*. Then, the *slice* is simply the set of all the points  $R_S(\alpha, d)$  such that  $d > -w$ ,  $d < w$ ,  $\alpha > 0$ , and  $\alpha < \alpha_S$ , where  $\alpha_S = \alpha' \cdot k$ , see Figure 7(a).

(3) *Retrieve compatible slices for a given node.* A *slice* is compatible with a node if the latter is inside the geographical area over the globe to be deformed, see Figure 7(b). If we have more than a compatible slice for a node, then we need to use a strategy to choose the best *slice*. Such strategy is described in Subsection 3.3.

(4) *Deformation of a slice and relocation of a node.* Each *deformed slice* is a grid of points where we map the texture of the globe surface that corresponds to the area of interest. Both, the nodes and the grid points are relocated over the globe and consequently in altitude using the relocation function, explained in details in Subsection 3.4.

(5) *Arc creation procedure.* Arcs are splines composed of a sequence of control points along the great circle

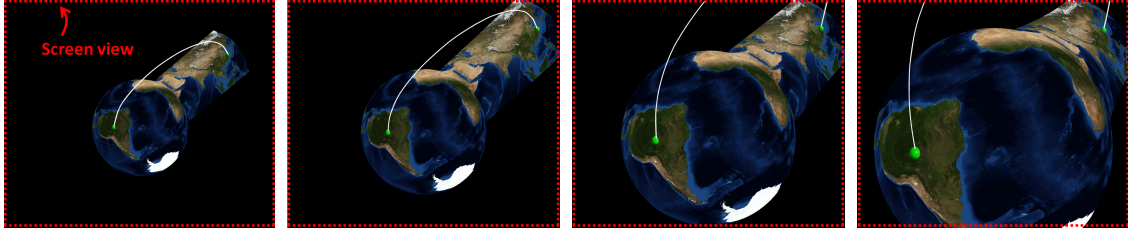


Figure 5: The *deformed slice* is bended to fit inside the screen view.

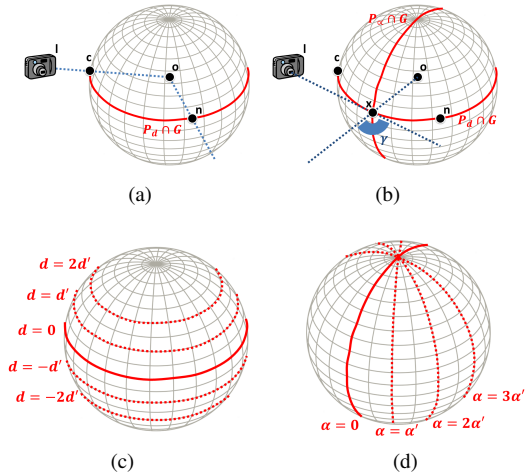


Figure 6: The planes  $P_\alpha, P_d$  are used to define the reference system for each *slice*.  $\alpha'$  and  $d'$  are values arbitrarily defined. Red lines are the intersection between the globe  $G$  and the planes.

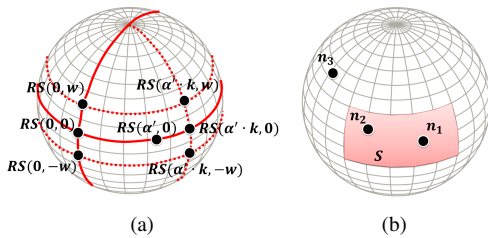


Figure 7: (a) Points used to define the *slice*. (b)  $n_1$  is the primary node of  $S$ .  $S$  is compatible for  $n_2$  but not for  $n_3$ .

that connects their two endpoints. Each control point has the following altitude:

$$h = 1 + h_{max} \sin(\pi t) \quad (1)$$

The parameter  $t$  ranges from 0 to 1, along the arc path.  $h_{max}$  is equal to a quarter of the distance between the endpoints. Arcs that connect nodes that were relocated are composed of control points with the following altitudes:

$$h^* = h + (h_s + (h_e - h_s)t) \quad (2)$$

The parameters  $h_e$  and  $h_s$  are respectively the altitudes of the two endpoints  $e$  and  $s$ . The altitude is applied with respect to the normal vector of the globe surface, in the point along the great circle. As shown in Figure 8, arcs that connect relocated nodes maintain an aesthetically pleasing shape.

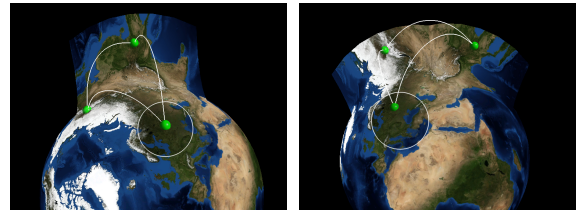


Figure 8: Examples of different arcs: arcs connecting nodes over the globe and arcs connecting relocated nodes.

### 3.3 Slice Selection Function

*Deformed slices* can occlude arcs, nodes and other slices. It is important to minimize the number of deformed slices as well as to ensure that each node is not occluded by its *compatible slices*. In this section, we describe the approach that satisfy these requirements.

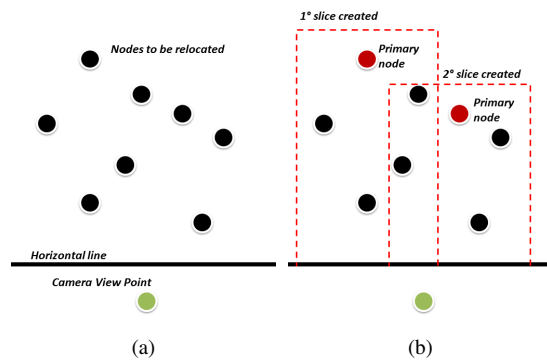


Figure 9: Our strategy to create slices and to choose the best one for a node.

Figure 9(a) shows our strategy in a projected 2D plane, for simplicity. The steps are the following:

The nodes are selected in descending order with respect to the distance with the camera view position projected over the globe. If the node has a compatible slice then

it is marked as *to relocate*. Otherwise, we create the *slice* and the node became its *primary node*, see red nodes in Figure 9(b). In a second round, we relocate the nodes. If a node has more than one compatible slice, then we choose the last created slice, because it is the most visible.

### 3.4 Function to Relocate Points

In this section, we describe how we relocate a point from  $R_S(\alpha, d)$  to  $R_S(\beta, d)$ . Such point can be a node or a grid point of the deformed slice.

First, we defined the altitude of the point to relocate with the following formula:

$$h' = 1 + h'_{max} \sin(\pi((\alpha - \beta)/2\alpha)) \quad (3)$$

where  $h'_{max}$  is equal to the great-circle distance between the original point and the point  $R_S(0, d)$ . Moreover, for a given point  $R_S(\alpha, d)$  that is relocated to  $R_S(\beta, d)$ , the corresponding vector to move the point in altitude, is the normal vector of the surface globe at the point  $R_S(\beta, 0)$ , see Figure 10(a). Thus, the deformed slice maintains the same width for every curvature.

The angle  $\beta$  is calculated with the following formula:

$$\beta = -2\alpha f(v) + \alpha \quad (4)$$

where  $f(v) \in [0, 1]$ . The function  $f(v)$  allow us to relocate points over a circular arc, see Figure 10(b).  $v$  is defined for each slice. If  $f(v)$  is the identity function, the *deformed slices* will be created as shown in Figure 10(c). However, the nodes relocated on the upper area of the deformed slice may be occluded by the surface itself. For such reason, as shown in Figure 10(d), we update it as follows:

$$f(v) = v(\alpha/\alpha_S)^2 \quad (5)$$

The parameter  $v$  is initially set to 1. However, if the *primary node* is outside a defined *sub-region* of the screen view,  $v$  is decremented until the *primary node* is considered visible (i.e. inside the *sub-region*), see Figure 11.

## 4 INPUT PARAMETERS

The list of user customizable parameters is the following.

1.  $r$  is the radius of the circle, expressed in pixels. We recommend the use of a small radius if the dataset contains many nodes in the circumscribed areas, in order to reduce the number of nodes that have to be relocated. The default value is 150 *pixels*, which can be interactively changed by the user.

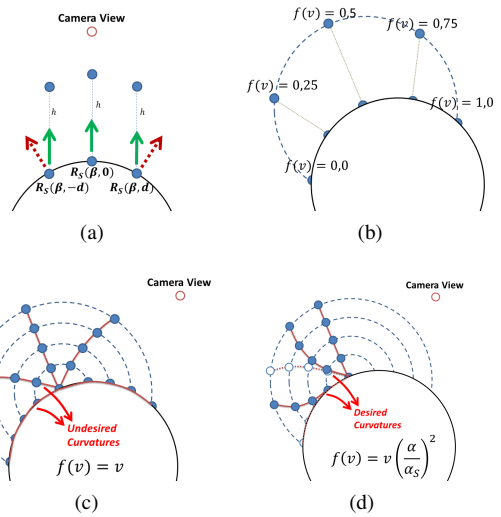


Figure 10: (a) Red arrows represent normal vectors over the globe surface. Green arrows represent the vectors used to move the points. (b) Each point is relocated over an imaginary arc. (c) The curvatures of a *deformed slice* modeled with the identity function, (d) and with our improved function.

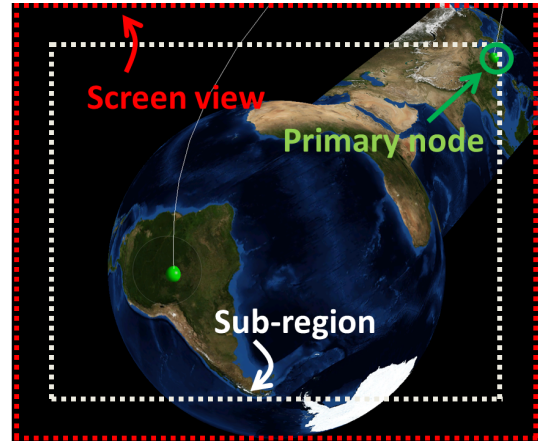


Figure 11: The *deformed slice* is bended until the *primary node* is inside the *sub-region* of the screen view.

2.  $X$  and  $Y$  are used to define the number of points that compose the grid of a *slice*. We do not need a dense grid to give the impression of a curved surface. Hence, heuristically we set  $X, Y = 7$ .
3.  $w$  defines the half width of the slice. The default value is 3.500 *km* that is a good balance between area to visualize and the possibility to intersect other slices.
4.  $k$  defines the position of the relocated node with respect to the y-axis of the deformed slice. The default value is 1.2. It allows the near surface of the node to be visible. However, if the area is too circumscribed, the cognitive process to identify the location of the node can be problematic for the operator.

- The dimension of the *sub-region* defines whether a node is considered visible or not. We set such parameter to be equal to 200 *pixels* less than the height and the width of the screen view. Since our strategy to decide the visibility of the nodes that are outside the screen view takes into account only the primary node of the *deformed slice*, other solutions can be adopted. For example, the visibility check can be performed on the two upper vertices of the geometry, and not on the *primary node*.

## 5 USE CASE

We used a dataset describing the connections between the main airports<sup>1</sup>, composed of 50 nodes and 677 edges. The dataset is composed of groups of near nodes connected with long distance nodes. Hence, the circle element is appropriate to explore such dataset, see Figure 12. Due to the high arcs density, the links connected with the nodes in focus are colored in red and the remaining are colored in semi-transparent white, see Figure 13. With this combination, the *important nodes* together with their connected arcs can be easily identified.

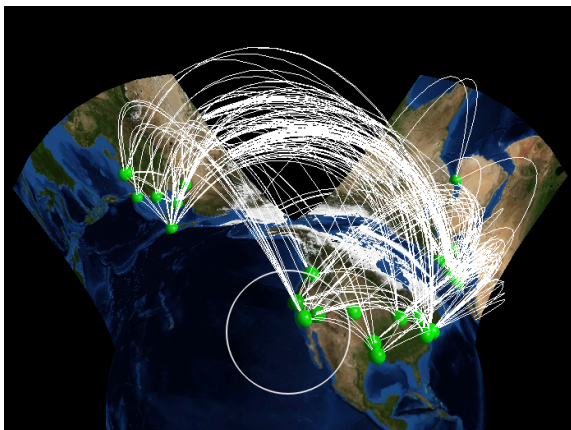


Figure 12: Airport connections dataset explored using *GeoPeels*.

## 6 EVALUATION

To measure the performance of our technique we take as metric the frame rate. *GeoPeels* was tested on a single processor Intel Xeon 2.26 GHz equipped with an NVIDIA GeForce GTX 280 with 1024 MB dedicated video memory.

Table 1 shows the *fps* (frame rate per second) of *GeoPeels* taking into account two datasets respectively with 154 and 4777 nodes. *#def. nodes* indicates the number of nodes that are relocated and *#slices* indicates the number of slices that are created. In this benchmark we used the values defined in Subsection 4.

<sup>1</sup> <http://openflights.org>

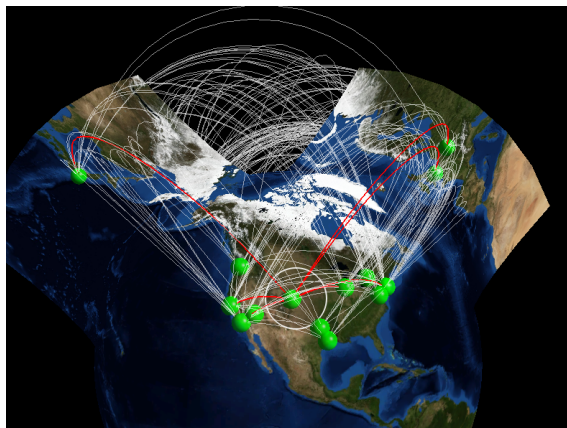


Figure 13: *GeoPeels* in combination with a color technique.

Our results take into account, for each frame, the time that is required to generate the grids and to render the geometries. In our tests we used spheres over the terrain to represent the nodes.

Table 1: Performance of *GeoPeels* for different datasets.

#nodes	#def. nodes	#slices	fps
154	0	0	120
	1	1	54
	4	1	50
	11	2	34
	9	3	28
	11	3	26
	7	4	23
4777	0	0	16
	2	1	14
	1	1	13
	6	2	12
	9	3	11
	3	3	11
	15	2	10
	22	5	8

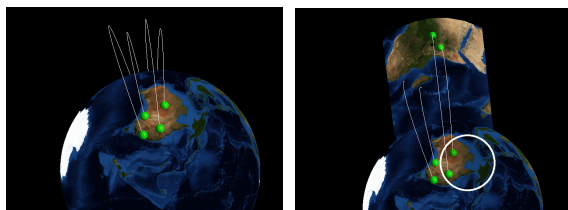
After analyzing the evaluation results, we noticed that the number of created slices and the number of relocated nodes have a considerable impact on the number of *fps*. However, for datasets with hundreds of nodes our approach gives acceptable performance.

## 7 DISCUSSION

In this section, we summarize our findings and the overall strength and weakness of our technique.

**Advantages with respect to other virtual globes.** An advantage of the proposed technique is that nodes that are considered important for a specific analysis are

never occluded by the surrounding map, see Figure 14. Additionally, we present an alternative solution to the approach of manually moving the camera view, which is a time consuming operation and may cause the loss of a previously established visual context.



(a) Virtual Globe. (b) GeoPeels.  
Figure 14: Nodes occluded behind the globe.

**Advantages with respect to 2D flat maps.** The limitation of 2D flat maps is on their spatial boundaries. It is difficult to get insight about nodes located on the borders of the map, especially on nodes that connect with nodes on the opposite side of the map, see Figure 15. Moreover the zooming out can make difficult the analysis of the spatial relations; the visual clutter caused by unoptimized long arcs can cause visualization problems. On possible solution is a 2D map projected over an imaginary infinite plan. However, such strategy may cause redundant information related to the nodes and the arcs.

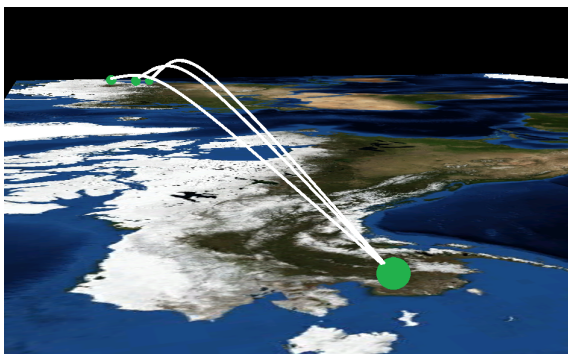


Figure 15: Fake sense of distance. These points are in reality quite close to each other.

**Limitations.** We identified few elements that require further investigation. First, some visual elements can be misleading for the operator because only nodes connected to nodes in the focus area are physically moved. But, our goal is to put more emphasis on a pre-selected data property, e.g. set of nodes and links.

Second, it might happen that arcs connecting visible nodes and nodes occluded by the globe's hemisphere intersect the newly deformed surfaces, e.g. when an occluded node is located behind the surface. Hence, it can mislead the operator to think that arcs end in the new deformed surface. However, this aspect can be alleviated by applying a level of transparency on the texture

of each deformed geometry, see Figure 16, or applying a different color to the arcs that end over the deformed surface, see Figure 13.

Third, in a classical virtual globe approach, we can look at the arcs shape to gain insight on the distance between its endpoints - farther the endpoints are higher the arc is. In our approach, this property does not hold in general.

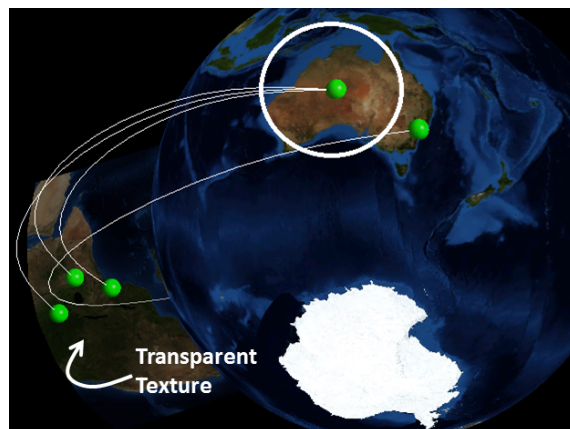


Figure 16: A transparent deformed slice. Hence, it is possible to keep trace of its intersecting arcs.

Four, as far as it concerns other deformation-based approaches, the main problem is the cognitive process associated with the identification of the deformed area. To improve this process, we can relocate together with the nodes auxiliary information, e.g. map labels.

Finally, *GeoPeels* is a deformation-based technique that requires the user to take care of the circle object.

**Extensions.** To improve the effectiveness of *GeoPeels*, it can be redesigned in terms of interaction. For example, it introduces an user defined selection of the nodes *in focus* as an alternative to the proximity-based selection of the circle element, see Figure 17.

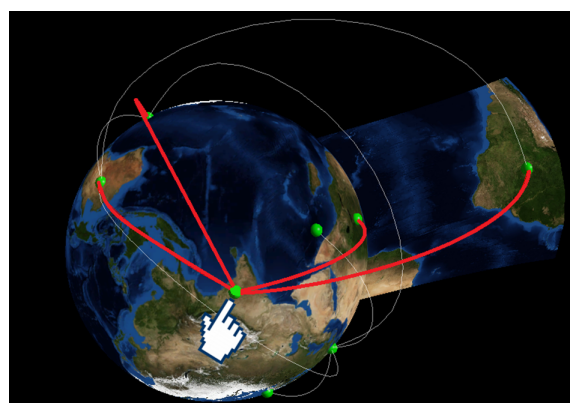


Figure 17: Extension based on node selection.

## 8 CONCLUSIONS AND FUTURE WORK

In this work, we have described an interactive deformation-based technique that capitalizes on dy-



dynamic globe deformations for an effective exploratory visual analysis of geographic networks.

Our technique provides an alternative approach to relocation of the camera view, which is a time consuming operation and may cause the loss of a previously established visual context. Another advantage of our technique is that it does not require the operator to define visual node-link attributes, e.g. we do not use attributes like color and size of a node, hence, they can be used to encode additional information.

Consequently, our technique can be combined with other works that make use of such visual attributes, for example, to reduce the visual clutter or combined with focus+context techniques applied on virtual globes, e.g. *important nodes* can be colored with red and the rest in semi-transparent white.

As future work, we plan to empirically measure the performance of the presented technique on a more comprehensive set of tasks. Moreover, a comparison has to be made with other approaches that are able to solve the raised issues. We also want to extend the orange peeling metaphor to other contexts such as the visualization of points of interest represented as icons and 3D models, or even to more complex scenarios such as the visualization of moving objects in real-time.

## 9 ACKNOWLEDGEMENTS

*GeoPeels* was applied to visually explore a geo-referenced network describing the wood chain. Such data was collected from the SLOPE project.

The work presented in this paper has received funding from the EC through the 7th Framework Programme under the Grant Agreement n. 604129 (project "SLOPE"). The authors are solely responsible for this work which does not represent the opinion of the EC. The EC is not responsible for any use that might be made of information contained in this paper.

## 10 REFERENCES

- [1] B. Alper. *Dynamic visualization of geographic networks using surface deformations*. PhD thesis, Citeseer, 2006.
- [2] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.
- [3] J. Böttger, M. Preiser, M. Balzer, and O. Deussen. Detail-in-context visualization for satellite imagery. In *Computer Graphics Forum*, volume 27, pages 587–596. Wiley Online Library, 2008.
- [4] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. In *Computer Graphics Forum*, volume 30, pages 971–980. Wiley Online Library, 2011.
- [5] U. Brandes, G. Shubina, and R. Tamassia. *Improving angular resolution in visualizations of geographic networks*. Springer, 2000.
- [6] S. Buschmann, T. Nocke, C. Tominski, and J. Döllner. Towards visualizing geo-referenced climate networks. In *Proceedings of Workshop GeoViz Hamburg 2013*, 0 2013.
- [7] P. Cignoni, C. Montani, and R. Scopigno. Magicsphere: an insight tool for 3d data visualization. In *Computer Graphics Forum*, volume 13, pages 317–328. Wiley Online Library, 1994.
- [8] C. Collins and S. Carpendale. Vislink: Revealing relationships amongst visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1192–1199, 2007.
- [9] K. C. Cox, S. G. Eick, and T. He. 3d geographic network displays. *ACM Sigmod Record*, 25(4):50–54, 1996.
- [10] A. Debiasi, B. Simões, and R. De Amicis. Supervised force directed algorithm for the generation of flow maps. In *Proceedings of the WSCG 2014 - 22nd International Conference on Computer Graphics*, 2014.
- [11] A. Debiasi, B. Simões, and R. De Amicis. 3darcLens: Interactive network analysis on geographic surfaces. In *Proceedings of the IVAPP - 6th International Conference on Information Visualization Theory and Applications*, 2015.
- [12] J. Diepstraten, D. Weiskopf, and T. Ertl. Interactive cutaway illustrations. In *Computer Graphics Forum*, volume 22, pages 523–532. Wiley Online Library, 2003.
- [13] J. F. Donges, Y. Zou, N. Marwan, and J. Kurths. The backbone of the climate network. *EPL (Europhysics Letters)*, 87(4):48007, 2009.
- [14] N. Elmqvist, Y. Riche, N. Henry-Riche, and J.-D. Fekete. Mélange: Space folding for visual exploration. *Visualization and Computer Graphics, IEEE Transactions on*, 16(3):468–483, 2010.
- [15] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(5):1095–1109, 2008.
- [16] G. W. Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.
- [17] G. W. Furnas. A fisheye follow-up: further reflections on focus+ context. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 999–1008. ACM, 2006.
- [18] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani.

- Wedge: clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 787–796. ACM, 2008.
- [19] P. Irani, C. Gutwin, and X. D. Yang. Improving selection of off-screen targets with hopping. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 299–308. ACM, 2006.
- [20] B. Jenny. Adaptive composite map projections. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2575–2582, 2012.
- [21] H. Jenny, B. Jenny, and L. Hurni. Interactive design of 3d maps with progressive projection. *The Cartographic Journal*, 47(3):211–221, 2010.
- [22] A. Lambert, R. Bourqui, and D. Auber. 3d edge bundling for geographical data visualization. In *Information Visualisation (IV), 2010 14th International Conference*, pages 329–335. IEEE, 2010.
- [23] I. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. Dobre, A. Muraru, A. Costan, M. Dediu, and C. Stratan. Monalisa: An agent based, dynamic service system to monitor, control and optimize distributed systems. *Computer Physics Communications*, 180(12):2472–2498, 2009.
- [24] H. Lorenz, M. Trapp, J. Döllner, and M. Jobst. Interactive multi-perspective views of virtual 3d landscape and city models. In *The European Information Society*, pages 301–321. Springer, 2008.
- [25] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Visualization, 2003. VIS 2003. IEEE*, pages 401–408. IEEE, 2003.
- [26] S. Möser, P. Degener, R. Wahl, and R. Klein. Context aware terrain visualization for wayfinding and navigation. In *Computer Graphics Forum*, volume 27, pages 1853–1860. Wiley Online Library, 2008.
- [27] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. Visualizing the global topology of the mbone. In *Information Visualization '96, Proceedings IEEE Symposium on*, pages 85–92. IEEE, 1996.
- [28] C. Niederauer, M. Houston, M. Agrawala, and G. Humphreys. Non-invasive interactive visualization of dynamic architectural environments. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 55–58. ACM, 2003.
- [29] T. Nocke. Basics and visual analytics of climate networks. In *Proceedings of the 2011 workshop on Climate knowledge discovery*, pages 2–2. ACM, 2011.
- [30] B. Pflesser, U. Tiede, and K. H. Höhne. Towards realistic visualization for surgery rehearsal. In *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 487–491. Springer, 1995.
- [31] P. Rodgers. Graph drawing techniques for geographic visualization. *Exploring geovisualization*, pages 143–158, 2005.
- [32] T. Ropinski and K. Hinrichs. Real-time rendering of 3d magic lenses having arbitrary convex shapes. 2004.
- [33] W. R. Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.
- [34] C. Tominski, J. Abello, F. Van Ham, and H. Schumann. Fisheye tree views and lenses for graph visualization. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 17–24. IEEE, 2006.
- [35] C. Tominski, J. F. Donges, and T. Nocke. Information visualization in climate research. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 298–305. IEEE, 2011.
- [36] M. Tory and C. Swindells. Comparing exovis, orientation icon, and in-place 3d visualization techniques. In *Graphics Interface*, volume 3, pages 57–64. Citeseer, 2003.
- [37] J. W. Tukey. *Exploratory data analysis*. 1977.
- [38] J. J. Van Wijk. Unfolding the earth: Myriahedral projections. *The Cartographic Journal*, 45(1):32–42, 2008.
- [39] J. Viega, M. J. Conway, G. Williams, and R. Pausch. 3d magic lenses. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 51–58. ACM, 1996.
- [40] A. Wolff. Graph drawing and cartography. In R. Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 23, pages 697–736. CRC Press, Boca Raton, FL, 2013.