Visual Link Routing in Immersive Visualisation

Arnaud Prouzeau
Monash University
Melbourne, Australia
arnaud.prouzeau@monash.edu

Antoine Lhuillier
University of Stuttgart
Stuttgart, Germany
antoine.lhuillier@gmail.com

Barrett Ens
Monash University
Melbourne, Australia
barrett.ens@monash.edu

Daniel Weiskopf
University of Stuttgart
Stuttgart, Germany
daniel.weiskopf@visus.uni-stuttgart.de

Tim Dwyer
Monash University
Melbourne, Australia
tim.dwyer@monash.edu

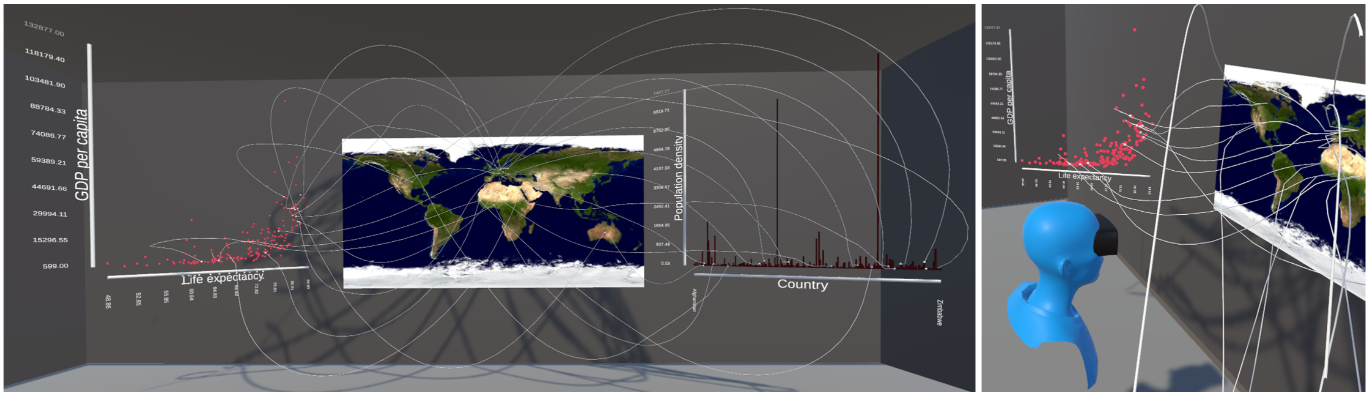

Figure 1. Visual links can show associations between data points in coordinated visualisations. The routing of such links in immersive environments presents challenges such as considering the viewpoints of multiple users

ABSTRACT

In immersive display environments, such as virtual or augmented reality, we can make explicit the connections between data points in visualisations and their context in the world, or in other visualisations. This paper considers the requirements and design space for drawing such links in order to minimise occlusion and clutter. A novel possibility in immersive environments is to optimise the link layout with respect to a particular point of view. In collaborative scenarios there is the need to do this for multiple points of view. We present an algorithm to achieve such link layouts and demonstrate its applicability in a variety of practical use cases.

ACM Classification Keywords

H.5.m. Information Visualization; Virtual Reality; Augmented Realty.

Author Keywords

Visual Links; Information Visualization; Immersive Analytics; Virtual Reality; Link Routing.

INTRODUCTION

Human beings live and work surrounded by complex systems of technological, social and economic structures, and webs of information. Normally these constructs are invisible in our environment but technologies like data visualisation seek to make them visible. Until recently this has been done on computer screens and sometimes in elaborate visualisation centres but, increasingly, we look to the emerging promise of augmented-reality technologies to overlay visual representations of these hidden systems into the space around us: at home, in the conference room, on the factory floor, out in the field, in the operating theatre, and so on. The goal in such situated analytics [43] scenarios is to provide access to these hidden layers of complexity when and where people need it.

DOI: `10.1145/1235`

But how do we make explicit the links between these information visuals floating in space, and the physical objects in our environment to which they relate? Similarly, in either augmented or virtual reality, how do we show the relationships between multiple data visualisations, perhaps floating on different panels in the space around us? In this paper, we explore the design space for rendering visual links between physical and virtual objects to show these connections. In particular, we explore the possibilities and requirements for routing such visible links. We need to balance sometimes conflicting goals, such as the need for each link to be easily visible with the need to avoid excessive clutter from such links. Such clutter would otherwise hide the very objects to which links are intended to draw attention.

The routing of such links has precedent in 2D data visualisation. For example, quite sophisticated routing algorithms have been developed for readable drawings of complex node-link network diagrams (see Henry Riche et al. [35] for a survey). However, immersive environments bring new and interesting challenges which we incorporate into our design space. In particular, we consider opportunities to route the links taking into consideration a user's (possibly moving) view point, or in collaborative scenarios, the view points of multiple users.

In summary, the contributions of this paper are:

- A design space for visual link routing, including the novel possibility in immersive environments for optimising specifically for the point(s) of view of one or more users.
- An algorithm for real-time, interactive routing of such links that is parameterised to support optimisation with respect to the various dimensions of our design space.
- A prototype immersive visualisation environment supporting visual links for multiple users.
- Use cases demonstrating visual links in various applications in virtual reality (VR) and augmented reality (AR).

RELATED WORK

In this section, we review prior work on visual links in 2D displays and in immersive environments. We also consider relevant literature on link routing in network visualisation.

Visual Linking in 2D Displays

Visual links have been used extensively in the field of information visualisation but with varying purposes. Within a single visualisation they can associate two different points. Diaconis and Friedman [11] plot a 4-dimensional dataset using two scatterplots by linking points associated with the same data point across the two plots. Parallel Coordinates Plot (PCP) visualisations follow a similar idea, linking an arbitrary number of axes, each axis representing a dimension in a multivariate dataset [23, 55]. While traditional PCPs have axes that are parallel, Claessen and Van Wijk propose Flexible Axes [7], an interactive tool where the axes can be composed into linked scatterplots and parallel coordinates in arbitrary ways to create infinite combinations of composite visualisations.

Another use of visual links is to connect associated elements in different views in visualisations composed into Multiple Coordinated Views (MCVs); or more commonly 'dashboard displays'. Such associations can also be shown using transient interactive brushing or coordinated highlighting [4], however, this is not the focus of our paper. Nelson shows associations between several text documents [33]. Weaver uses them to show which views are coordinated in MCVs [54] and Shneiderman and Aris use them to visualise multi-variate networks [40]. Waldner et al. [50] show how links can be used between different applications on a desktop. Jiang et al. [24] describe a similar use of links, but in a large display setup. Finally, Tobiasz et al. [44] use them to link two views of the same visualisation at different times.

In all of the above situations, links can create occlusion; if they are too numerous, they can significantly hinder the readability of a visualisation. One solution is to display the links only for elements currently selected by the user, as is done in FlowVizMenu [48]. With ConnectedCharts, Viau and McGuffin [47] propose to transfer the links to the axes of the visualisations instead of the data points when they are too numerous. Another solution proposed by Steinberger et al. [42], is to automatically detect important areas of a visualisation using a visual saliency detection algorithm and re-route the links to avoid these areas. A study shows that with such links users are more performant than by using colour highlighting. On a related topic, Alper et al. [2] focus on how to draw links that are best understood by users. Their studies show that the links have to be smooth and linear.

Visual links are also used between 2D visualisations in 3D space on regular displays. Dwyer et al. explore network centrality with PCPs [12]. The authors investigate different renderings for the links including variations of opacity and size. Collins and Carpendale [8] generalize this concept in their VisLinks design, with visual links between any configuration of coordinated 2D visualisations in 3D space, represented on a large display screen. To avoid excessive clutter in the space around and between visualisations they bundle the links using a simple scheme of Bezier curves with shared control points. Waldner et al. use visual links to guide a user's attention between displays in a multi-display environment [49]. Note that all of the above examples of visual linking rely on very simple curves or straight lines for the links. Algorithms are not used to provide link routing that is optimal with respect to clutter or any other attributes for link curvature that might be desirable.

Visual Links in Immersive Environments

An important use of visual links in immersive environments is to associate virtual or real objects that are somehow related. One early example is the work of Rekimoto and Saitoh [34], which shows links from projected data elements (e.g. text documents or images) to the real objects to which they relate (e.g. the computer or other device on which it is stored). Systems by Sandor et al. [37] and Serrano et al. [39] use AR representations of links to show connections between real devices for input/output redirection. Using screen-based AR, the Reality Editor [18] allows users to dynamically create links between physical smart objects. Similarly, Ivy [13] allows link authoring in VR to program complex behaviour. These links are also used to visualise data flow between objects when a program is running.

In immersive environments, visual links can be used to associate points in data visualisations. Following Flexible Axis, Cordeil et al. propose ImAxes [9], a system in which the user can directly manipulate axes to create complex composite visualisations in VR. In this system, links reveal associations between PCP axes or between coordinated scatterplot visualisations. In ART (Augmented Reality above the Tabletop) [6], Butscher et al. apply links in the context of Parallel Coordinate Scatterplots in AR above an interactive tabletop. Finally, links are used also to associate elements from different views. Ens and Irani [14] show how to use them to show relationships between points in different spatially situated view configurations in AR. Similarly, Mahmood et al. [32] visualise links between an AR headset and visualisations shown on a large wall display. Modif: Yang et al. [57] used links to represent flows between regions of the world. the authors showed that some parameters of the routing of the link, like the maximal height, can be efficiently used to represent specific values.

Another case where visual links are used in immersive environments is to associate labels with specific objects in the scene. This is not always easy, as the labels should not occlude the scene but neither should they overlap. Additionally, the layout of labels is view-dependent (depend on the position of the user), and should be recalculated each time the user moves. Methods of calculating this layout include identifying empty regions in the scene when projected on the user's point-of-view-plane [5], using force-based algorithms [17], and using image driven measures like visual saliency to avoid areas which seem important [17]. However, to our knowledge, no work has tried to optimise the routing of links in such a case.

We can see that visual links have myriad application both on 2D displays and in immersive environments. However, while there has been work investigating intelligent routing of such links in 2D [42], the examples from immersive environments described above all use very straightforward link routing schemes (i.e. simply straightlines or Bezier bundles). In the domain of network visualisation, however, there has been greater exploration of the potential benefits of more intelligently routed links (edges) to facilitate path following between nodes.

Link/Edge Routing in Networks

Much research has focused on the layout of node-link network visualisations (*graphs* in the sequel) to improve their readability. Following convention, we refer to links in this context as *edges*. Ware et al. [53] explain that the three main characteristics that influence graph readabilty are edge lengths, their continuity and their crossings. Additionally, Huang et al. [20] show that the angle at which edges cross is important and that the larger an angle is, the easier it is to read.

However, as the number of edges increases, a graph becomes harder to read even with a very good layout. One solution is to act globally on the routing of the edges by aggregating them into bundles. Such edge bundling has been extensively studied (see [30] for a review). While bundling improves the readability of a graph, it also removes information regarding individual links. Other research focuses on more local techniques using lenses: in EdgeLens [56] links inside the lens are bent without moving the nodes in order to disambiguate node and link relationships; Local Edge Lens [46] shows in the lens only the links connected to nodes inside the lens; PushLens [38] pushes links that would transit through the lens away instead of hiding them; MoleView [22] can hide links depending on specific attributes, or bundle and unbundle them. Finally, most of the lenses presented above are combined in MultiLens [26]. Riche et al. [35] propose a design space for the interactive manipulation of link curvature which spans most of the previous solutions.

Ware and Franck [51] show that visualisation of graphs in immersive environments with stereoscopy and head-tracking is beneficial to readability. A similar study by Ware and Mitchell [52] demonstrates similar results with higher resolution displays. Alper et al. [1] use stereoscopy on a 2D graph layout (bringing selected nodes forward) and show that it is beneficial for highlighting when coupled with colour. Halpin et al. [16] show that the use of virtual reality leads to better insight regarding social network exploration. Finally, FiberClay [21] Modif: and NeuroCave [25] uses bundling in immersive environments. However, in most of these cases, users are looking at the graph from the outside (i.e. from an exocentric point of view) such that most of the graph is within the user's field of view. Another solution is to draw the graph around the user (i.e. from an egocentric point of view), Kwon et al. [29] propose the use of spherical user-centred layout and used edge bundling. Their study shows that participants were faster and encountered fewer errors with a large graph compared to a 2D layout.

Overall, a high-quality routing of edges in network visualisation is a routing with short edges with as few crossings as possible. When crossing cannot be avoided, edges should cross at maximal angles (as close to orthogonal as possible). Rather than trying to achieve such a routing across the entire graph (which can be intractable), more localised methods can be used to improve graph readability, like lenses to bundle, spread and remove edges. The transition to immersive environments allows more options for edge routing (i.e. in three dimensions instead of two), however occlusion from a specific view angle remains a problem. The use of a routing centred on the user's point of view can be a solution. In the next section we present our design space inspired by these characteristics.

A DESIGN SPACE FOR IMMERSIVE VISUAL LINKS

Visual links are important in visualisations involving multiple views since they highlight relationships between related pieces of information across views [42]. The most basic task that users must perform is following a link from one end to another. This can be difficult with a high number of links. The previous section shows that it is possible to modify the routing of the links themselves to facilitate such navigation.

First, it is possible to decrease separation between links, as proposed in the Moleview system [22]. Alternately, when there are too many links to significantly increase space between them, separation between links with nearby end points can instead be decreased (i.e. bundling) to increase the overall

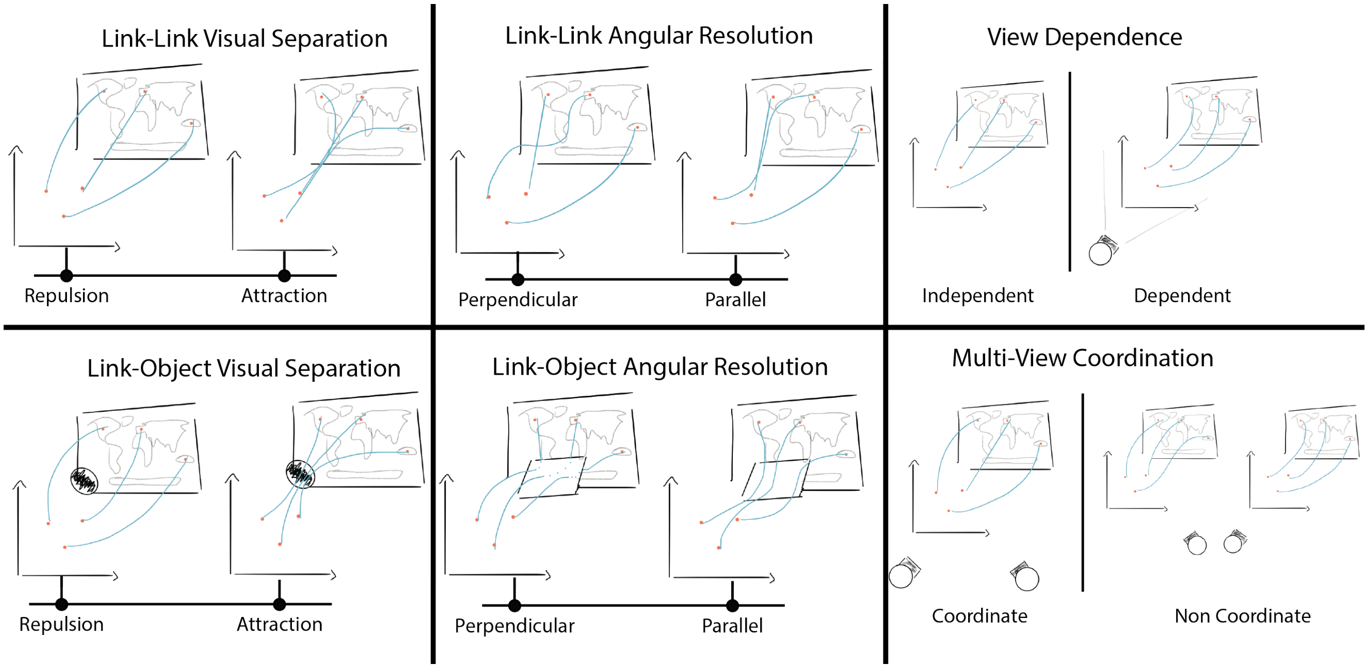

Figure 2. Design space dimensions: *Visual Separation* (link-link and link-object); *Angular Resolution* (link-link and link-object); *View Dependence*; and *Multi-View Coordination*

saliency of the scene [30]. Links can also obscure other important objects in the scene, it is then possible for links to avoid these objects, as has been done for graphs with EdgeLens [56] and PushLens [38], and by Steinberger et al. for multiview visualisations [42]. Second, it is possible to change how links cross. Huang et al. showed that near-perpendicular crossings are more readable [20]. By contrast, Viau and McGuffin propose to group links by making them parallel [47]. The systems above all consider 2D links in network diagrams. In immersive environments there is very little work on link routing optimisation. Hurter et al. propose to use an exocentric (the user's position is not taken into account) 3D edge bundling in FiberClay [21]. Kwon et al. use an egocentric spherical layout (surrounding the user's view point) to visualise a graph [29]. Both of these techniques focus on a single user context and do not take into account the possibility for multiple users in the scene, which is a compelling use-case in immersive environments.

Informed by this classification of past work, we propose a design space for the routing of visual links in augmented and virtual reality. For routing links we have the following dimensions: *Visual Separation*, *Angular Resolution*, *View Dependence*, and *Multi-View Coordination* (Figure 2). *Visual Separation* and *Angular Resolution* are 3D equivalents of design goals that have been considered in past work on 2D link routing in network layout and view linking. The difference is that immersive environments can involve a richer set of obstacles and attachments which need to be considered by the routing, such as physical objects in the environment in an augmented reality scenario. *View Dependence* and *Multi-View Coordination*, however, are unique to head-tracked immersive environments and are, to the best of our knowledge, entirely new considerations for link routing. This section presents each dimension of the design space and their attributes.

Visual Separation

The degree to which links are kept apart from other links and objects in the environment, along their entire length and in all three spatial dimensions, we refer to generally as *Visual Separation*. We consider two cases, link-link separation and link-object separation.

Link-Link Separation

The goal of increasing link separation is to make it easier to follow individual links, as there is less chance to confuse them. However, such behaviour can not scale with a large number of links. Either the links will still be close to each other, or they will became very long and thus harder to follow. With a large number of links, it may be preferable to highlight patterns of connectivity through bundled 'highways'.

Link-Object Separation

In addition to keeping links separate from one another, it is important to keep them well separated from other objects in the scene, for instance collaborators or other important objects in the background of an AR application [5]. In general, links should not pass through objects to which they are not explicitly connected. Rather, they should route around them at some distance sufficient to avoid any ambiguity.

Angular Resolution

Angular Resolution is the visual angle with which a link crosses another link or an object. The extremes are *perpendicular* and *parallel*. Similarly to *Visual Separation*, there are two cases, link-link separation and link-object separation.

Link-Link Angular Resolution

The angular resolution of a crossing between two links can favour either the identification of individual links (with perpendicular crossings) or the identification of trajectory patterns in the links (with parallel crossings). While perpendicular crossings may help avoid confusion at a crossing when the user follows a link, near-parallel crossings will contribute to aggregate links and thus reduce clutter.

Link-Object Angular Resolution

Just as poor angular resolution between links can make them difficult to follow when they cross, angular resolution between links and objects' surface can help to clarify whether the link is connected to the surface rather than merely 'grazing' it at a near-tangential angle. However, there may also be situations where it is desirable for a link to be close to a surface or even run parallel to the surface. For example, links between visualisations might need to meet up with continuations of those links as 2D paths within a flat visualisation.

View Dependence

View Dependence is the degree to which the routing depends on the user's viewpoint. The routing can either be *independent* of, or *dependent* on the viewpoint. This aspect is important in immersive environments that track the user's head position, as the appearance of sets of links routed through space varies greatly depending on the position of the viewer in 3D space.

In a view-independent link routing, the trajectory of the links is invariant to the position of user (or, potentially, multiple users). The quality metric of the link routing in the view-independent case depends on the absolute distance between links and their crossings in three dimensions. However, links can still appear to cross from a user's viewpoint (i.e. when projected onto the 2D viewing plane), even when they do not cross in the 3D space. To optimise the visualisation of links, we can make the routing view-dependent, so that the link routing dynamically adapts according to the position of the user.

Visual Separation between links has a different definition in a view-dependent 'space' compared to a view-independent one. In the view-dependent space, two links can appear visually close for the user, but not be physically close. The choice of view-dependence will then impact on the effect each link has on others. It is possible to apply complex behaviour to the links by using two different separation criteria in the two different spaces. For example, we can have links attracting each other in the view-independent space to create bundles, but have these bundles repulse each other in the view-dependent space to make them easier to visually follow (Figure 4-bottom).

Multi-View Coordination

Multi-view coordination considers whether or not multiple users view the same link routing. Routes can either be *coordinated* or *non coordinated* across multiple users' viewpoints.

In single-user applications, the quality of a view-dependent link routing relies only on a satisfactory configuration of the links from the point of view of one user. However, if multiple users are involved, then finding a configuration of links that is at-once satisfactory (coordinated) for all of the users' different points of view adds another layer of complexity. Since the routing depends on the point of view of the user, the optimal routing will be different for each user.

A multi-view coordinated routing attempts to find a compromise between the different views and then present the same 3D routing to all users. An uncoordinated routing simply optimises the routes for the viewpoint of each individual user. In this case, each user may view a link routing that is different in absolute 3D space from what the others see.

Multi-view coordination raises the question of common ground. By providing a different routing to each user, we are reducing the common picture they have of the scene, which could thus hinder collaboration. In such case, users would lose the ability to enhance their communication by pointing towards a particular link, as its position might be completely different in the routings seen by others (although this is true only for the links, and not the objects that the links are connecting). In contrast, by providing a coordinated routing, we also provide a sub-optimal view for each user, which could then hinder performance. It is also possible to prioritise the view of one or more users over others, in which case the appearance of the routing would be of better quality from the prioritised viewpoints.

Discussion

This design space defines our objectives for *high-quality routing* of visual links. The definition of a *high-quality* is of course heavily reliant on the goal of the visual links in the application. If the goal is to favour the identification of individual links, then a 'good' routing will favour separation and perpendicular crossing between links. On the other hand, if the goal of the visual links is to highlight patterns in the trajectory of the links, then a good routing will aggregate individual links to create bundles showing the general trend of connectivity. There are probably other definitions of a good routing, and the purpose here is not to show why one is better than the others, but rather to propose a framework which provides a parameter space in which to explore.

The main goal of this paper is to show how thoughtful consideration towards the routing of links can improve readability of the visual links in immersive environments. As such, we focus on the primary dimensions of our design space above. Modif: Whereas analytic methods such as Zwicky's Morphological Analysis often intend for such design dimensions to be orthogonal [36], there are some potential dependencies introduced in this design space. For instance, combining a parallel link-link angular resolution with repulsive visual separation may introduce instability in the layout. Nonetheless, we believe the identified dimensions are conceptually useful in practice, as demonstrated by the force-based model discussed in the following section. We also note there are several other variables that could be used to increase (or decrease) the visual saliency of links to manage clutter:

Colour — Links can be coloured or textured to increase their contrast with the background [28]. Colour could also potentially be used to make links blend in with the background if desired [41].

Opacity — In our examples links are completely opaque. However, a degree of transparency can make the links less obtrusive when they cannot be rerouted around important content [42]. Transparency could also give an additive behaviour to links when they congregate; links added to a bundle will make it more and more opaque, hinting to users about the number of links inside the bundle.

Gradient — The choices for colour and opacity of links described above do not need to be applied uniformly along their length. For example, colour at either end of the link can encode information about features at the opposite end [13]. To reduce clutter, links could be opaque near their connections to data points, but fade into transparency towards their midsections. Relatively short opaque ends may even be sufficient to give an adequate cue of the direction of their other end, such that most of the rest of the link can be completely invisible.

Animation, Glow effects, etc. — Other possibilities for managing the visual saliency of links include animations (e.g. similar links could vibrate, wave or pulse in synchrony), lighting effects (e.g. glowing or pulsating) or others. The effectiveness of these techniques generally is well known from gestalt principles and perceptual psychology [27]. How well they would work in this particular application is worthy of further study.

Following our design space, we derive a framework which allows for the realisation of various link routing options. Our technical framework mirrors our design space: *visual separation* and *angular resolution* dimensions are controlled through a force-based model. Moreover, the forces are *view dependent* through their frame of reference (i.e. forces are expressed either in the world space or the local user space). Finally, the *multi-view* coordination aspect is achieved through a multi-player synchronisation process.

Force-based Model

In our technical framework, we use a force-directed approach that has proven successful in similar problems like graph routing [15], edge bundling [19], but also label positioning [17], to model our link-link and link-object behaviour. Following [30], we define a path-set P as an ordered tuple (V,E), where $V \in \mathbb{R}^3$ is a finite set of vertices and $E \subseteq V \times V$ a set of edges. In our specific case, we assume that a link $l \in E$ is an ordered couple (u,v), with $u,v \in V, u \neq v$. For the purpose of drawing and changing the link's shape, we sample links as a set of equidistant points $e = \{e_i\}_{0 \leq i \leq k}$, with $k \in \mathbb{N}$ (Figure 3). To ensure that the start and end points e_0 and e_k do not move during our force-directed placement, we only apply the forces on the sampled points e_j in between the start and end points ($\{e_j\}_{0<j<k}$).

Our force-based model takes into account several types of forces that can be applied to each point of a given link: the internal link force; the visual separation forces and the angular separation forces. Detailed in Table 1, these forces tightly mirror our definition of the design space for visual and angular separation. Both the visual and angular separation forces can be either exerted on a link by another link or any object. We note that all our forces are expressed as a function of a constant (k), a distance function (d) and a direction ($\hat{v}$). These parameters can change depending on their defined frame of reference (i.e. view-dependency level), see the next section for more details.

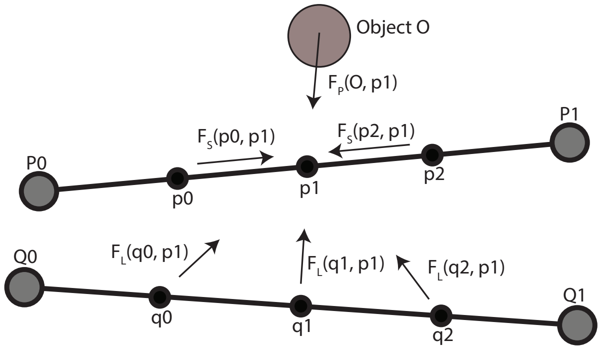

Figure 3. Force model applied on the point p1. Internal forces are coming from p0 and p2, and external visual separation forces from the points of the link Q and from the object O

First, we implement an internal link force which aims at preserving the integrity and linearity of the link with a zero-length Hooke's Law. Then, we model the visual separation forces as an electrostatic force that acts between either sampling points of different links or a sampling point and an object. In the visual separation forces, the sign of the constant coefficient (k_s) defines whether two entities should attract or repulse each other. Finally, we model the angular separation forces as a Lorentz force (i.e. a magnetic and electric force): Given the force exerted on a sampling point of a link e_i from another point p, we achieve perpendicular angular resolution by creating a magnetic force that moves e_i towards a position orthogonal to p's direction (i.e. the direction for a link and the normal to the surface for an object). Typically, this force is a function of the distance from the closest orthogonal position and the direction and magnitude of the vector, the projection vector for an object and rejection vector (i.e. the component of the direction that is orthogonal to the projection) for a link. Conversely, to achieve parallel angular resolution, we model an electrical force that steers the point e_i towards a position parallel to the other point's direction.

Frames of Reference

As detailed in our design space, the routing of the links can be done in a view-dependent space. As such, our force model must take into account the position of users in relation to the world-space frame of reference.

Projection Space

The view-dependency aspect of our design space raises the question of defining the correct view-dependent projection. Indeed, the view-dependent space requires us to define a specific projection space that preserves the principle of user-perspective aligned points, i.e. it requires a specific geometric projection space where two points aligned in reference to the origin are also visually aligned for the user. During our technical implementation of our visual link routing method, we tried three projection approaches: orthographic, perspective and gnomonic projections.

Table 1. Force-based model: classes of forces and their respective generic expression used in our visual link routing method

Internal Force			$k_l \left[d(e_i,e_{i+1}).\hat{\mathbf{v}}_{e_i} - d(e_i,e_{i-1}).\hat{\mathbf{v}}_{e_{i-1}}\right]$	k_l: Spring constant ($k_l \leq 0$).
Separation Forces	Link to Link		$\frac{k_s}{d(e_i,p)^2}.\hat{\mathbf{u}}_{e_i p}$	$k_s > 0$: attraction constant, $k_s < 0$: repulsion constant, $\hat{\mathbf{u}}_{e_i p}$: unit vector from e_i to p, where p is a link e_j or an object.
	Link to Object			
Angular Separation Forces	Link to Link	Perpendicular	$\frac{k_a}{d(e_i,e_j)^2}.\mathbf{v}_{e_i \perp \hat{v}_{e_j}}$	k_a: separation constant, $\mathbf{v}_{e_i \perp \hat{v}_{e_j}}$: rejection vector along $\hat{v}_{e_j}$, $\mathbf{v}_{e_i \| \hat{v}_{e_j}}$: projection vector along $\hat{v}_{e_j}$, where $\mathbf{v}_{e_i} = \mathbf{u}_{e_i e_{i+1}}$.
		Parallel	$\frac{k_a}{d(e_i,e_j)^2}.\mathbf{v}_{e_i \| \hat{v}_{e_j}}$	
	Link to Object	Perpendicular	$\frac{k_a}{d(e_i,p)^2}.\mathbf{v}_{e_i \| \hat{n}}$	$\hat{n}$ the normal to the object surface, $\mathbf{v}_{e_i \perp \hat{n}}$: rejection vector along $\hat{n}$, $\mathbf{v}_{e_i \| \hat{n}}$: projection vector along $\hat{n}$, where $\mathbf{v}_{e_i} = \mathbf{u}_{e_i e_{i+1}}$.
		Parallel	$\frac{k_a}{d(e_i,p)^2}.\mathbf{v}_{e_i \perp \hat{n}}$	

Overall, inspired by previous work [56] on general graph layout and from the results of both our own experimentation, we settled for a more human-like projection: the spherical gnomonic projection centred on the user's head. This projection is rotation-invariant and conserves angular resolution. This means that two points aligned from the user point of view will still be aligned with the origin after the projection. Moreover, we can express the distance between two points as the angle between them. We chose to do the projection on the unit sphere and thus disregard the distance between the projected object and the head of the user. Movements in this direction (i.e. bringing the object further or closer from the user) do not impact the separation or angular resolution between two objects. However, stereoscopy has been shown to be beneficial to highlighting edges in graphs, and this could be an interesting direction to explore in future work [1].

As such, in the view-independent space, we choose to use the regular 3D Euclidean space and a fixed frame of reference. Conversely, our view dependent space is modelled as a 2D sphere centred around the user's head in a moving frame of reference.

Forces in Different Frames of Reference

In our model, the position of each of these points is impacted by different forces and the points move accordingly. The direction of the impact of the forces differs based on the classes of object interacting with the point (e.g. point, line, plane) and on the space in which we apply the force. As such, in the view-independent routing, we apply the force in world space, while in the view-dependent one, we apply it in the spherical projection centred on the head of the user. Here, we explicitly set the direction and magnitude of the different forces as a function of these two parameters. Next, we detail how to express them based on their frame of reference.

In the world-space (or view-independent space), we encode their magnitude using the Euclidean distance and the direction as being towards either: a point for the internal and visual separation forces; or, a line or plane for the angular resolution forces. Using Euclidean linear algebra, these forces can be expressed easily as direction vectors and cross-products.

However, in the spherical space (or view-dependent space), the forces are expressed using different distance and vectors. In the spherical projection space, all the movements take place on the surface of the sphere centred on the user's head (i.e. in 2D). Here, there are only two possible types of movements: towards a point or towards a line. Moreover, expressing both distance and vectors in spherical geometry is non-trivial [45]. In the spherical space, the direction from two points s and p is the vector tangent to the sphere at the position of s going towards p (i.e. $\hat{\mathbf{s}} \times (\hat{\mathbf{u}}_{sp} \times \hat{\mathbf{s}})$). Their distance is expressed as the path along the sphere between them (*i.e.* the great circle distance: $R * atan2(||\mathbf{s} \times \mathbf{p}||, \mathbf{s} \cdot \mathbf{p})$ with R the radius of the sphere). Similarly, the direction between a point s and a line $(p, \hat{\mathbf{n}})$: $\hat{\mathbf{n}}$ is non-trivial on the sphere. First, we note that $\hat{\mathbf{n}}$ changes along the sphere. To solve this, it is accepted that the "normal" to a line in spherical geometry is the vector orthogonal to the circle drawn by the line ($\hat{\mathbf{c}} = \hat{\mathbf{p}} \times \hat{\mathbf{n}}$). Following this, the vector from s to the line is defined as the vector tangent to the sphere at p oriented such that the intersection with the line is orthogonal (i.e. $(\hat{\mathbf{s}} \times (\hat{\mathbf{p}} \times \hat{\mathbf{n}})) \times \hat{\mathbf{s}}$). Moreover, the distance between them is called the cross-track distance and defined as $R * (acos((\hat{\mathbf{p}} \times \hat{\mathbf{n}}) \cdot \hat{\mathbf{s}}) - \pi/2) = R * asin((\hat{\mathbf{n}} \wedge \hat{\mathbf{p}}) \cdot \hat{\mathbf{s}})$.

Overall, depending on the frame of reference in which we express a force, the direction vector of the generic forces presented in Table 1 is going to change. In particular, if we define the link-to-link separation forces as being user's-view dependent, the direction of the link-to-link unit vector is going to be defined as $\hat{\mathbf{u}}_{\mathbf{e_i e_j}|\mathbf{Spherical}} = \hat{\mathbf{e}}_{\mathbf{i}} \times (\hat{\mathbf{u}}_{e_i e_j} \times \hat{\mathbf{e}}_{\mathbf{i}})$.

Multi-user Synchronisation

When the routing is not coordinated between users, our implementation simply allows for each user to locally compute the forces applied to each link. Links moves accordingly, independently in each user's view. Conversely, when the routing needs to be coordinated between several users, our implementation uses a single server approach. The server – incorporating the position of all the users – computes separately for each user all the forces on each link point. Thus, for each user, the system knows the displacement that should be applied to each link point. It then simply computes the sum of all forces applied to each point and solves the simulation using Newton's second law. Finally, the new point positions are streamed back to the client.

This simple single "master-server" approach is straight forward to implement but can, of course, lead to a sub-optimal result for all users in terms of computational complexity and speed. Other approaches could be used to coordinate the views. For example, we could compute the displacement of the links for one user and apply the same to the other users. Future work should study the impact of such behaviour.

Implementation

In our implementation of our technique, we used a standard explicit Euler integration process. At each time step t we compute the forces acting on each point and then apply the displacement as δ. While this integration process could diverge, our experiments showed it was sufficient for our application. It could be improved for stability in the future using the Runge-Kutta method or a Semi-implicit Euler integration with a convergence criteria. Modif: The visual appearance of the links could also be improved by using for instance a Gaussian kernel to smooth the curve [19].

Overall, the model was developed in compute shader using unity. We used a shader kernel for each link, meaning that in each kernel we explore all subpoints and for all subpoints we compute the impact of all the other points. Again, this was sufficiently fast for the application, but it could be made to scale to examples involving more links or more link points using an approximation method such as Barnes-Hut simulation [3]. Our implementation can handle small data-sets of up to 100 links ($\simeq 3200$ points with a sampling of 32 points per link) with a run-time of 16 ms per iteration. This allows us to reach interactive framerates above the reprojection threshold of current HMDs (i.e. ≥ 45 fps).

USE CASES

In this section, we provide scenarios that demonstrate the use of our model for visual links in VR.

Use Case 1: An Individual User

Alice is a data scientist at the United Nations. She wants to analyse the relationship between life expectancy and GDP per capita for each country. Using data coming from the Gap Minder dataset and a VR data visualisation toolkit (e.g. IATK [31]), she can visualise a scatterplot of these measures. She would like to add context to this visualisation, to see if there are any geographical patterns, so she adds a 2D map to her environment. She then selects a few points of interest in the scatterplot and creates links between these points and the corresponding positions of the country in the map. Figure 4-top shows the result of the visual linking. Alice is not satisfied as it is actually difficult to follow each link. She uses our model to optimise the link routing as a function of her point of view. She also makes sure that the links do not overlap with the label of the axis by using separation between a link and an object (Figure 1). Finally, to improve readability, the model also has the link follow a trajectory perpendicular to both visualisations.

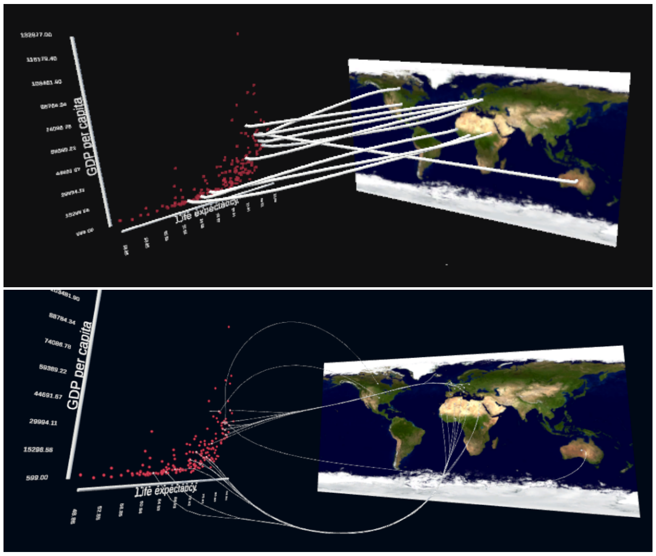

Figure 4. Links before adding a repulsive force, and after bundling

Satisfied with this first visualisation, she wants to explore the dataset more deeply. Using a brushing tool with one of her controllers, she selects all the points in the first quarter of the scatterplot (i.e. low life expectancy and low GPD per capita). Links are drawn using a view-dependent routing to increase separation and angular resolution of the links, which improves Alice's ability to identify and follow individual links. She is also interested to detect high-level geographic patterns. Using her other controller, she selects the links coming to the same geographical region and bundles them together (Figure 4-bottom). The bundling in this case is done in the view-independent space (i.e. 3D space) as she wants the links entering each country close to where they join the bundle.

Finally, she wants to compare population across countries, to see if there is any interesting correlation. She adds a barchart visualisation of population for each country. Each country linked to the scatterplot is now also linked to the correct bar of the barchart and Alice can test the hypothesis that the countries in the first quarter of the scatterplot also have high population (Figure 1).

After finishing her analysis, Alice wants to present her findings to executives in a board meeting. She selects a few countries

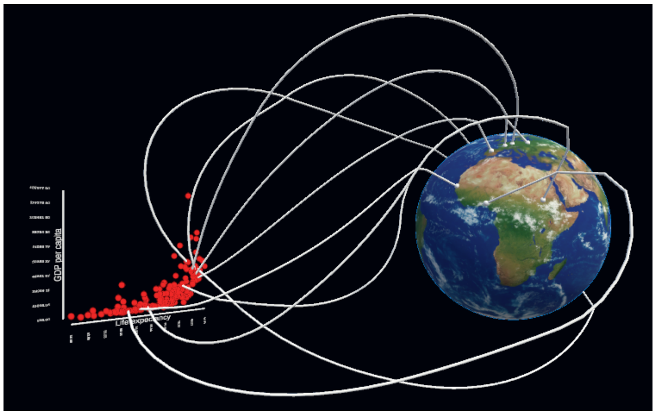

Figure 5. Optimisation with a 3D object. Links approach the globe surface from perpendicular angles, and avoid crossing the globe from the user's viewpoint

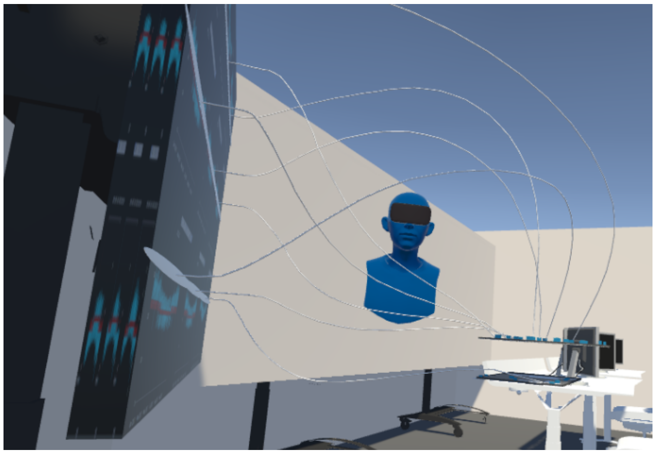

Figure 6. Links are rerouted to avoid occluding a collaborator's face

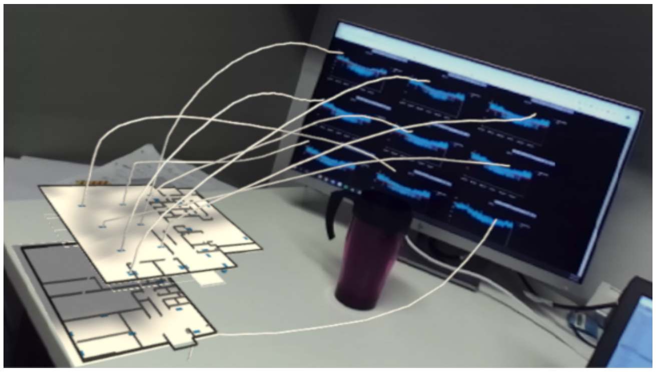

Figure 7. Links in AR approach real and virtual surfaces at perpendicular angles. The links are routed away from the model surface and avoid the user's view of her coffee cup

of interest and adds visual links for these countries between the scatterplots and the map. To have a more engaging presentation, she replaces the map with a 3D globe, which slowly rotates. She identifies the 3D globe as an obstacle in the model to make the links avoid passing through or behind it (Figure 5).

Use Case 2: A Multi-User Discussion

Bob and Arthur are two building managers of a university. Because they work at different campuses they often need to meet remotely and, for which they use a virtual reality meeting room. They are currently meeting to talk about a specific building whose faulty air-conditioning system has become the subject of several complaints from students and university staff. On a 2D virtual screen, they visualise the temporal evolution of the inside temperature in each room over the year. They also have a 3D 'exploded' model of the building and use visual links to connect each temporal evolution graph to the room location in the building.

At first, they decide to explore the data in parallel, with each working on their own to get a feeling of what could be wrong with the air conditioning system in the building. They use individual view dependent layouts for the links, which provides the best possible view for each of them without interference with the routing of the other. When Bob and Arthur are ready to discuss their findings, they want to be able to point to individual links during their discussion, with confidence that the other will see precisely the link to which they are referring. For this they switch to a multi-view coordinated, view-dependent layout. To favour collaboration, they also ask the model to increase the separation between the links and the head of each collaborator's avatar as viewed by the other (Figure 6).

Use Case 3: Virtual Links in Augmented Reality

Natalie is a data analyst at Melbourne International Airport. She wants to analyse the usage patterns of the boarding gates in a small wing of the airport. She uses AR to display on her desk a 3D model of departure lounge. The location, in the 3D model, of each gate is linked to the 2D desktop display showing the frequency of flights departing from that gate. By following the links Natalie can determine the relationship between the spatial information in the model and the abstract data on her screen.

Tho minimise occlusion of the data at both ends, Natalie uses an perpendicular angular resolution between the links and the virtual and physical surfaces. The virtual model also has a repulsive spatial separation to keep the links away from its surface as they bend towards the screen. Repulsion can also prevent links from occluding physical objects in her surroundings. In this case, Natalie has previously used a training application to familiarise the AR computer vision system with her coffee cup. She has assigned the cup a repulsive object-link force, so that the links do not occlude it and she can easily grasp it by reaching through a gap in the visible links (Figure 7).

Discussion

We present a model for routing of visual links which, to our knowledge, has not been seen in previous research. In the first use case, we describe different ways an individual user can interact with and use the links. The most basic way is by moving, the user will impact the positions of the links. This movement is continuous and stable enough to avoid noisy movement due to potential noise in the tracking. The user can also move different objects that have been defined as attractive or repulsive to links. Finally, the user can group links together (using selection), and have them bundle or unbundle interactively. While this model allows for a lot of flexibility, it has some limitations. First, with a high number of interactive objects, it can become difficult for the user to understand how the links are impacted by the forces. This can lead to the user performing unintentional actions. Alternately, such a model can also prevent the user to do specific actions if some forces forbid it.

The second use case focuses on a multi-user scenario. As stated before, a user-centred model makes sure that routing is optimised for user position. However, with several users, it may be undesirable for the model to be optimised for only one person, as it can provide a bad layout for the others. The two solutions we choose are to either show an optimised routing to each user, thus they do not have the same routing, or to show to them a compromised, but coordinated, routing which is suboptimal for all users. The best solution will depend on the situation. For example, when users are not collaborating, they are working in parallel, the first solution may be better. Users do not need to know what the others are doing, and with a coordinated routing, the others moving can actually be disturbing as it will impact the link positions. On the other hand, when users are closely collaborating, they need to have information about what the others are doing, and to not have the same routing could negatively impact communication between users by preventing the use of deictic gestures (i.e. pointing at an object, like a link, while talking about it). One limitation of the coordinated routing is that each new user will add constraints on the layout, and thus limit its flexibility. The routing quality will degrade as the number of users increases.

Finally, the last use case shows the use of visual links in an augmented reality scenario. With the use of efficient tracking system (e.g. VICON, Image recognition), it is possible to track real objects and integrate them in the force model. this can lead to interesting behaviours, like the links avoiding collaborators' faces to not hinder communication. By detecting salient regions in the real environment, links could automatically avoid areas of interest, as has been done by Steinburger et al. [42]. More research needs to be done to explore the behaviour of links in augmented reality contexts.

CONCLUSION, LIMITATIONS AND FUTURE WORK

We have presented a design space for routing visual links in immersive visualisations. Based on the design space, we define a technical framework for routing visual links in 3D space, with allowances for optimising layouts for the viewpoint(s) or one or more users. We implemented this model using a force-directed approach. Finally, we present 3 example scenarios that demonstrate the features of our implementation. This work extends a sophisticated work on 2D link routing into the area of immersive environments. This move opens the door to new possibilities for visual links, such as optimising their routings for moving viewpoints, and dynamically shifting to prevent occlusion of important real-world objects.

We have demonstrated that our implementation reliably produces many intended behaviours of our design space and technical framework in many circumstance. However, there are several areas for improvement and further investigation of this work. We have observed that the stability of the algorithm can be an issue when parameters are at extreme values, for example when links are too close to surfaces. We have also identified several possibilites to optimise our approach. Our algorithm currently produces interactive framerates (>60fps) on a standard desktop VR machine when there are fewer than 90 links (i.e. <3000 points). This is adequate for the use cases shown where pairs of points are linked across visualisations, but for many to many correspondences there could easily be thousands of links. The force-directed approach could be made to scale to these scenarios using standard techniques like spatial decompositions to aggregate long-range interactions.

There are also many opportunities to make the visual links and routing layouts more interactive. For instance, we have currently shown links connected to visualisations and objects that remain stationary. It would also be interesting to study how users could move visualisations and objects to improve the link layout, it could produce interesting unforeseen behaviours [10]. Further work is also required to explore alternate behaviours of links, for instance changing colour or opacity when near salient objects. Further research also remains to be done in the evaluation of link behaviours, especially the user reception to dynamically adapting routings as shown in our view dependent layouts.

REFERENCES

1. B. Alper, T. Hollerer, J. Kuchera-Morin, and A. Forbes. 2011. Stereoscopic Highlighting: 2D Graph Visualization on Stereo Displays. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2325–2333. DOI:`http://dx.doi.org/10.1109/TVCG.2011.234`

2. Basak Alper, Nathalie Riche, Gonzalo Ramos, and Mary Czerwinski. 2011. Design study of linesets, a novel set visualization technique. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2259–2267.

3. Josh Barnes and Piet Hut. 1986. A hierarchical O (N log N) force-calculation algorithm. *nature* 324, 6096 (1986), 446.

4. Richard A. Becker and William S. Cleveland. 1987. Brushing Scatterplots. *Technometrics* 29, 2 (1987), 127–142. `http://www.jstor.org/stable/1269768`

5. Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. ACM, New York, NY, USA, 101–110. DOI: `http://dx.doi.org/10.1145/502348.502363`

6. Simon Butscher, Sebastian Hubenschmid, Jens Müller, Johannes Fuchs, and Harald Reiterer. 2018. Clusters, trends, and outliers: How immersive technologies can facilitate the collaborative analysis of multidimensional data. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 90.

7. J. H. T. Claessen and J. J. van Wijk. 2011. Flexible Linked Axes for Multivariate Data Visualization. *IEEE*

Transactions on Visualization and Computer Graphics 17, 12 (Dec 2011), 2310–2316. DOI: http://dx.doi.org/10.1109/TVCG.2011.201

8. Christopher Collins and Sheelagh Carpendale. 2007. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1192–1199.
9. Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017a. ImAxes: Immersive Axes As Embodied Affordances for Interactive Multivariate Data Visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 71–83. DOI: http://dx.doi.org/10.1145/3126594.3126613
10. Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017b. ImAxes: Immersive Axes As Embodied Affordances for Interactive Multivariate Data Visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 71–83. DOI: http://dx.doi.org/10.1145/3126594.3126613
11. Persi Diaconis and Jerome H. Friedman. 1983. M AND N PLOTS11Work partially supported by the Department of Energy under contract number DE-AC03-76SF00515. Work partially supported by National Science Foundation under grant MCS77-16974. In *Recent Advances in Statistics*, M. Haseeb Rizvi, Jagdish S. Rustagi, and David Siegmund (Eds.). Academic Press, 425 – 447. DOI: http://dx.doi.org/https://doi.org/10.1016/B978-0-12-589320-6.50024-1
12. Tim Dwyer, Seok-Hee Hong, Dirk Koschützki, Falk Schreiber, and Kai Xu. 2006. Visual Analysis of Network Centralities. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Volume 60 (APVis '06)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 189–197. http://dl.acm.org.ezproxy.lib.monash.edu.au/citation.cfm?id=1151903.1151931
13. Barrett Ens, Fraser Anderson, Tovi Grossman, Michelle Annett, Pourang Irani, and George Fitzmaurice. 2017. Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments. In *Proceedings of the 43rd Graphics Interface Conference*. Canadian Human-Computer Communications Society, 156–162.
14. Barrett Ens and Pourang Irani. 2017. Spatial Analytic Interfaces: Spatial User Interfaces for In Situ Visual Analytics. *IEEE computer graphics and applications* 37, 2 (2017), 66–79.
15. Thomas M. J. Fruchterman and Edward M. Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and Experience* 21, 11 (1991), 1129–1164. DOI: http://dx.doi.org/10.1002/spe.4380211102
16. Harry Halpin, David J. Zielinski, Rachael Brady, and Glenda Kelly. 2008. Exploring Semantic Social Networks Using Virtual Reality. In *The Semantic Web - ISWC 2008*, Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 599–614.
17. Knut Hartmann, Kamran Ali, and Thomas Strothotte. 2004. Floating Labels: Applying Dynamic Potential Fields for Label Layout. In *Smart Graphics*, Andreas Butz, Antonio Krüger, and Patrick Olivier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 101–113.
18. Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 307–310.
19. Danny Holten and Jarke J. Van Wijk. 2009. Force-Directed Edge Bundling for Graph Visualization. *Computer Graphics Forum* 28, 3 (2009), 983–990. DOI: http://dx.doi.org/10.1111/j.1467-8659.2009.01450.x
20. Weidong Huang, Peter Eades, and Seok-Hee Hong. 2014. Larger crossing angles make graphs easier to read. *Journal of Visual Languages & Computing* 25, 4 (2014), 452 – 465. DOI:http://dx.doi.org/https://doi.org/10.1016/j.jvlc.2014.03.001
21. C. Hurter, N. H. Riche, S. M. Drucker, M. Cordeil, R. Alligier, and R. Vuillemot. 2019. FiberClay: Sculpting Three Dimensional Trajectories to Reveal Structural Insights. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (Jan 2019), 704–714. DOI: http://dx.doi.org/10.1109/TVCG.2018.2865191
22. C. Hurter, A. Telea, and O. Ersoy. 2011. MoleView: An Attribute and Structure-Based Semantic Lens for Large Element-Based Plots. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2600–2609. DOI:http://dx.doi.org/10.1109/TVCG.2011.223
23. Alfred Inselberg. 1985. The plane with parallel coordinates. *The Visual Computer* 1, 2 (01 Aug 1985), 69–91. DOI:http://dx.doi.org/10.1007/BF01898350
24. Hao Jiang, Daniel Wigdor, Clifton Forlines, and Chia Shen. 2008. System design for the WeSpace: Linking personal devices to a table-centered multi-user, multi-surface environment. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. IEEE, 97–104.
25. Johnson J. G. Keiriz, Liang Zhan, Olusola Ajilore, Alex D. Leow, and Angus G. Forbes. 2018. NeuroCave: A web-based immersive visualization platform for exploring connectome datasets. *Network Neuroscience* 2, 3 (2018), 344–361. DOI: http://dx.doi.org/10.1162/netn_a_00044

26. Ulrike Kister, Patrick Reipschläger, and Raimund Dachselt. 2016. MultiLens: Fluent Interaction with Multi-Functional Multi-Touch Lenses for Information Visualization. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 139–148. DOI:http://dx.doi.org/10.1145/2992154.2992168

27. Kurt Koffka. 2013. *Principles of Gestalt psychology*. Routledge.

28. Ernst Kruijff, J Edward Swan, and Steven Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 3–12.

29. O. Kwon, C. Muelder, K. Lee, and K. Ma. 2016. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 7 (July 2016), 1802–1815. DOI: http://dx.doi.org/10.1109/TVCG.2016.2520921

30. A. Lhuillier, C. Hurter, and A. Telea. 2017. State of the Art in Edge and Trail Bundling Techniques. *Computer Graphics Forum* 36, 3 (2017), 619–645. DOI: http://dx.doi.org/10.1111/cgf.13213

31. B. Bach C. Hurter B.H. Thomas K. Marriott T. Dwyer M. Cordeil, A. Cunningham. 2019. IATK: An Immersive Analytics Toolkit. In *Proceedings of IEEE VR*.

32. T. Mahmood, E. Butler, N. Davis, J. Huang, and A. Lu. 2018. Building Multiple Coordinated Spaces for Effective Immersive Analytics through Distributed Cognition. In *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*. 1–11. DOI: http://dx.doi.org/10.1109/BDVA.2018.8533893

33. Theodor Holm Nelson. 1999. Xanalogical Structure, Needed Now More Than Ever: Parallel Documents, Deep Links to Content, Deep Versioning, and Deep Re-use. *ACM Comput. Surv.* 31, 4es, Article 33 (Dec. 1999). DOI: http://dx.doi.org/10.1145/345966.346033

34. Jun Rekimoto and Masanori Saitoh. 1999. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 378–385. DOI: http://dx.doi.org/10.1145/302979.303113

35. Nathalie Henry Riche, Tim Dwyer, Bongshin Lee, and Sheelagh Carpendale. 2012. Exploring the design space of interactive link curvature in network diagrams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 506–513.

36. Tom Ritchey. 1998. General morphological analysis. In *16th euro conference on operational analysis*.

37. Christian Sandor, Alex Olwal, Blaine Bell, and Steven Feiner. 2005. Immersive mixed-reality configuration of hybrid user interfaces. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*. IEEE, 110–113.

38. Sebastian Schmidt, Miguel A. Nacenta, Raimund Dachselt, and Sheelagh Carpendale. 2010. A Set of Multi-touch Graph Interaction Techniques. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 113–116. DOI:http://dx.doi.org/10.1145/1936652.1936673

39. Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Gluey: Developing a head-worn display interface to unify the interaction experience in distributed display environments. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 161–171.

40. Ben Shneiderman and Aleks Aris. 2006. Network visualization by semantic substrates. *IEEE transactions on visualization and computer graphics* 12, 5 (2006), 733–740.

41. Srikanth Kirshnamachari Sridharan, Juan David Hincapié-Ramos, David R Flatla, and Pourang Irani. 2013. Color correction for optical see-through displays using display color profiles. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*. ACM, 231–240.

42. M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. 2011. Context-Preserving Visual Links. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2249–2258. DOI: http://dx.doi.org/10.1109/TVCG.2011.183

43. B.H. Thomas, G. F. Welch, P. Dragicevic, N. Elmqvist, P. Irani, Y. Jansen, D. Schmalstieg, A. Tabard, N.A.M. ElSayed, R.T. Smith, , and W. Willett. 2019. Situated Analytics. In *Immersive Analytics*, K. Marriott, F. Schreiber, T. Dwyer, K. Klein, T. Itoh, W. Stuerzlinger, and B.H. Thomas (Eds.). 185–220.

44. M. Tobiasz, P. Isenberg, and S. Carpendale. 2009. Lark: Coordinating Co-located Collaboration with Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1065–1072. DOI: http://dx.doi.org/10.1109/TVCG.2009.162

45. Isaac Todhunter. 1863. *Spherical trigonometry, for the use of colleges and schools: with numerous examples*. Macmillan.

46. C. Tominski, J. Abello, F. van Ham, and H. Schumann. 2006. Fisheye Tree Views and Lenses for Graph Visualization. In *Tenth International Conference on Information Visualisation (IV'06)*. 17–24. DOI: http://dx.doi.org/10.1109/IV.2006.54

47. C. Viau and M. J. McGuffin. 2012. ConnectedCharts: Explicit Visualization of Relationships between Data Graphics. *Computer Graphics Forum* 31, 3pt4 (2012), 1285–1294. DOI: http://dx.doi.org/10.1111/j.1467-8659.2012.03121.x

48. C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica. 2010. The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (Nov 2010), 1100–1108. `DOI:` **`http://dx.doi.org/10.1109/TVCG.2010.205`**

49. Manuela Waldner, Alexander Lex, Marc Streit, and Dieter Schmalstieg. 2009. Design Considerations for Collaborative Information Workspaces in Multi-Display Environments.

50. Manuela Waldner, Werner Puff, Alexander Lex, Marc Streit, and Dieter Schmalstieg. 2010. Visual Links Across Applications. In *Proceedings of Graphics Interface 2010 (GI '10)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 129–136. **`http://dl.acm.org/citation.cfm?id=1839214.1839238`**

51. Colin Ware and Glenn Franck. 1994. Viewing a graph in a virtual reality display is three times as good as a 2D diagram. In *Proceedings of 1994 IEEE Symposium on Visual Languages*. IEEE, 182–183.

52. Colin Ware and Peter Mitchell. 2008. Visualizing Graphs in Three Dimensions. *ACM Trans. Appl. Percept.* 5, 1, Article 2 (Jan. 2008), 15 pages. `DOI:` **`http://dx.doi.org/10.1145/1279640.1279642`**

53. Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. 2002. Cognitive Measurements of Graph Aesthetics. *Information Visualization* 1, 2 (June 2002), 103–110. `DOI:` **`http://dx.doi.org/10.1057/palgrave.ivs.9500013`**

54. C. Weaver. 2005. Visualizing coordination in situ. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* 165–172. `DOI:` **`http://dx.doi.org/10.1109/INFVIS.2005.1532143`**

55. Edward J. Wegman. 1990. Hyperdimensional Data Analysis Using Parallel Coordinates. *J. Amer. Statist. Assoc.* 85, 411 (1990), 664–675. **`http://www.jstor.org/stable/2290001`**

56. N. Wong, S. Carpendale, and S. Greenberg. 2003. Edgelens: an interactive method for managing edge congestion in graphs. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*. 51–58. `DOI:`**`http://dx.doi.org/10.1109/INFVIS.2003.1249008`**

57. Y. Yang, T. Dwyer, B. Jenny, K. Marriott, M. Cordeil, and H. Chen. 2019. Origin-Destination Flow Maps in Immersive Environments. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (Jan 2019), 693–703. `DOI:` **`http://dx.doi.org/10.1109/TVCG.2018.2865192`**