A Design Space For Data Visualisation Transformations Between 2D And 3D In Mixed-Reality Environments

Benjamin Lee
Monash University
Melbourne, Victoria, Australia
benjamin.lee1@monash.edu

Maxime Cordeil
Monash University
Melbourne, Victoria, Australia
University of Queensland
Brisbane, Queensland, Australia
max.cordeil@monash.edu

Arnaud Prouzeau
Inria & LaBRI (University of
Bordeaux, CNRS, Bordeaux-INP)
Bordeaux, France
arnaud.prouzeau@inria.fr

Bernhard Jenny
Monash University
Melbourne, Victoria, Australia
bernie.jenny@monash.edu

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

Figure 1: Conceptual future scenarios of 2D and 3D transformations and mixed-reality data visualisation. (a) A user at a desktop extrudes a 3D bar chart by grabbing and pulling from the monitor. (b) A remote worker on the side of a street extruding a 3D scatterplot from a tablet with a virtual surface aiding organisation of a faceted bar chart. (c) Two collaborators visualising data on a wall without the need for large high-resolution displays.

ABSTRACT

As mixed-reality (MR) technologies become more mainstream, the delineation between data visualisations displayed on screens or other surfaces and those floating in space becomes increasingly blurred. Rather than the choice of using either a 2D surface or the 3D space for visualising data being a dichotomy, we argue that users should have the freedom to transform visualisations seamlessly between the two as needed. However, the design space for such transformations is large, and practically uncharted. To explore this, we first establish an overview of the different states that a data visualisation can take in MR, followed by how transformations between these states can facilitate common visualisation tasks. We then describe a design space of how these transformations function, in terms of the different stages throughout the transformation, and the user interactions and input parameters that affect it. This design space is then demonstrated with multiple exemplary techniques based in MR.

CCS CONCEPTS

• **Human-centered computing → Visualization theory, concepts and paradigms**; **Mixed / augmented reality**.

KEYWORDS

visualisation, mixed reality, Immersive Analytics, animated transitions, direct manipulation

1 INTRODUCTION

Prior to computer graphics, abstract data was mostly represented in 2D for publication in reports, books, or posters. Meanwhile 3D representation was limited to physical constructs of spatial data like geographic globes, chemical, medical, or architectural models. The first wave of desktop computers with reasonable graphics capability led to a proliferation of representations of 3D data projected onto 2D screens. This arguably resulted in the overuse of 3D graphics—such as the classic gratuitous 3D charts so hated by Tufte

and others—and early studies of 3D visualisations on 2D screens demonstrated their limitations. This has since led to a long period in the information visualisation research community of consolidation of the information visualisation design space around 2D representations, optimally arranged for 2D screens.

With the emergence of mixed reality (MR) technologies in recent years, we need to reconsider some of our assumptions about the "natural habitat" of data visualisations. Mixed-reality (MR) headsets, such as the Microsoft HoloLens 2, are finally achieving tetherless, robust spatial tracking and high-resolution stereoscopic rendering with reasonable field-of-view. These headsets now also have an understanding of their environment, mapping surfaces in the room and tracking the hand gestures of their user. We can render 2D-like graphics that are visibly projected on any surface in the environment, 2.5D-like graphics that visibly extrude out from said surfaces, or suspend them in the 3D space around us—all with equal ease and fidelity (Figure 2).

This new capability presents us with new design choices and possibilities for data visualisation in an immersive environment—also known as *Immersive Analytics* [10, 41]. We should of course continue to visualise data in the best manner possible, whether it be on a 2D surface or in 3D space. With the flexibility that MR provides however, we can consider how any given visualisation can freely move between these two environments—a surface or the space—to suit a user's needs. Imagine, the ability to temporarily extrude a 2D visualisation out from a monitor into 3D just by grabbing and pulling it with your hand to encode some data to the third spatial dimension (Figure 1a), or to extrude out visualisations from a tablet and place them suspended in the space in front of you (Figure 1b). These visualisations can also be placed flush against arbitrary surfaces, mimicking a large wall-sized 2D display while retaining the flexibility of 3D (Figure 1c). In contrast to these, we can also flatten a 3D visualisation down to 2D on a surface, such as by applying a projection or creating a cross-sectional view. Supporting these transitions between surfaces and spaces in immersive environments has been identified as one of the grand challenges of Immersive Analytics [17].

While recent work (Section 2) has demonstrated applications involving the use of 2D surfaces and displays in conjunction with MR for data visualisation, we specifically focus on how visualisations can be transformed between different states. We provide a more formal exploration of how transformations between these states can facilitate common visualisation tasks (Section 3). We then develop a design space and conceptual framework that helps to define and describe these transformations in terms of their visual state changes and the user interaction(s) that accompany it (Section 4). We then demonstrate instantiations of visualisation transformation techniques that we created in MR (Section 5 and our supplemental video). Lastly, we provide some high-level guidelines and considerations for how best to design transformations, as well as a discussion of limitations and possible future work (Section 6).

2 RELATED WORK

2.1 Animated Transitions and User Control

Perhaps most relevant to our work is the study of animated transitions between related statistical graphics. Animation has been

Figure 2: A matrix of how visualisations can exist in mixed-reality. *Visualisation Dimensionality* **refers to the number of spatial encodings a visualisation has.** *Environment* **refers to where a visualisation is placed within the user's surroundings. Native visualisations have a number of spatial encodings equal to the number of spatial dimensions in their display environment.**

shown to be useful in numerous ways, such as to keep track of changes between visual states [24, 46], improve decision making [22], or increase viewer engagement [2, 24]. Various grammars and toolkits have been developed to aid visualisation designers in creating effective and engaging animated transitions (e.g., [32, 58]). These all focus on conventional 2D displays however, and therefore there has been little consideration of how transitions should be authored for immersive 3D environments. Moreover, there is a preference to use keyframe animation for creating transitions [57], which is an interpolation between a start and end state across some time period (and potentially within this, staging and staggering [11, 24]). While this paradigm is convenient for the designer, it does not take into account how users may control the behaviour of the animation itself. A good example of the utility of user-controlled transitions in 2D visualisation is the *DimpVis* interaction technique by Kondo and Collins [33], which uses direct manipulation on graphical marks to progress through a time-varying visualisation (or in other words, an animation).

As MR better facilitates multi-modal input and direct manipulation with hand tracking, it is natural to expect a higher degree of user control for these animations, such as to alter which variables are being aggregated or to control the progress of a transition. While standard animated transitions are effective for presentation purposes, it has been shown that user control of transitions is required to support data analysis [47]. However, the input space is more complex in immersive environments, opening up many more design options and considerations for designing interactive transformations. In this work, we aim to establish a design space that maps out these opportunities and more.

2.2 Transitions between 2D and 3D on Flat Displays

While data visualisation has traditionally been performed in 2D on flat displays, some previous research has demonstrated careful use of 3D to enhance some aspect of users' exploration—even without the use of stereoscopic displays. An early use of 3D transitions to help navigate 2D hierarchical charts can be found in the work of

Robertson et al. on *Polyarchies* [46]. 3D rotations were later used in other 2D visualisation techniques to help transition between 2D scatterplots with the *ScatterDice* technique [16], 2D graphs with the *GraphDice* technique [6] and a similar technique used between 2D aircraft trajectories views [28]. These transform a visualisation between 2D and 3D by shifting between orthographic and perspective projections [42] at the beginning and end of each rotation. Such 3D rotation transitions were later studied to understand how they can convey information in dense visualisations [15]. *Matrix Cubes* by Bach et al. extends this by visualising multiple adjacency matrices of different time steps as a single 3D space time cube, which can then be partitioned into multiple 2D slices and so forth [3]. 3D rotation transitions to 2.5D perspective views have also been used to improve visual links between 2D visualisations [12] or as a transitional view between parallel coordinates and radar charts [21].

While the focus of the user's exploration in these examples is still on 2D visualisations, there is clear value in the use of 3D to supplement certain aspects such as transition awareness and context preserving. These are, of course, all within the confines of a 2D display. In this work, we aim to fill this gap by exploring further possibilities of these transitions in an immersive 3D space.

2.3 Transitions between 2D and 3D in Immersive Analytics

Considerable research in Immersive Analytics has explored techniques to improve the perception and understandability of 3D visualisations, especially as a result of common 3D pitfalls such as occlusion and perspective distortion [23, 42]. For example, Kraus et al. investigated the role of immersion on cluster identification tasks [34], and Prouzeau et al. proposed the use of haptic feedback and highlight planes with handheld controllers to help find occluded features in 3D scatterplots [43]. On the other hand, some research has, whether intentionally or not, devised ways of side-stepping these 3D challenges by allowing for 3D visualisations to temporarily be transformed into 2D. For example, *FiberClay* by Hurter et al. allows users to project 3D trajectories down into 2D at orthogonal angles using bimanual rotational and scaling operations [27], in a manner reminiscent of *Scatterdice*. Most closely related to our work, *Tilt Map* by Yang et al. demonstrates transitions between a 2D choropleth map, 3D prism map, and 2D bar chart by simply tilting a handheld controller [64]. As the map is tilted, it smoothly transforms between a 2D choropleth, 3D prism, and 2D bar chart at set angular intervals. This simple interaction allows the user to control both the transition and the visual state depending on their given needs. In contrast, Reipschlager et al. use *Augmented Reality Visualisation Layers* to enhance 2D visualisations on wall-sized displays with superimposed 3D visualisation through simple touch gestures [45], simulating a transformation into 3D. Other works provide visualisation authoring tools that make it easy to create both 2D and 3D visualisations, allowing users to quickly swap between them at will. Most notably, *ImAxes* by Cordeil et al. enables the construction of 3D visualisations simply by placing three embodied virtual axis objects at orthogonal angles in VR [14]. Using the same ImAxes grammar, Smiley et al. demonstrate how the *MADE-Axis*—a tangible controller for data visualisation—can

be used to physically construct both 2D and 3D visualisations by leveraging its composability in a collaborative MR prototype [53].

While some of these works share striking similarities with our work, our goal is to devise a higher level design space that can encapsulate these techniques and others like them. We also explore how these transformations can be activated and controlled by users, rather than solely focusing on their visual design.

2.4 Physical and Virtual Surfaces in Immersive Analytics

Physical surfaces, such as wall-sized displays, tabletops, and tablets, are very commonly used in Immersive Analytics applications for varying purposes. By default, surfaces can be used to display conventional 2D visualisations, while separate standalone 3D visualisations are rendered as floating objects using an MR headset (e.g., [8, 60]), allowing for a wider range of visualisations—both 2D and 3D—to be displayed simultaneously in the workspace. A physical surface can also act as a shared area that is suitable for collaboration, with MR headsets providing private spaces for each user to work individually [31, 55]. They can also facilitate touch interaction that can be used in tandem with MR, such as through a multi-touch display (e.g., [7, 26, 50]), infrared touch sensor (e.g., [54]), or even using the headset's hand-tracking capabilities [63]. By leveraging touch input, it is common for visualisations on 2D displays to be directly "augmented" with superimposed 3D graphics in MR. These include graphical marks which spatially encode data using the third dimension protruding from the surface [35, 40, 45], visual links [26], or even entire visualisations that are contextually placed close to the surface [35, 40, 45]. Such surface augmentations have also been demonstrated to be used with real-world objects in the form of embedded visualisations using MR headsets [62]. In contrast to physical surfaces, *virtual* surfaces may act as substitutes in cases where access to physical surfaces is not possible—such as in highly mobile settings [19]—or when tangibility is not strictly required—such as when only used for organisation and presentation [36].

These works motivate the combination of a 2D surface with the 3D virtual space. Few however consider how visualisations may *transition* between the two and in what manner. Moreover, due to the close connection that 2D content has with surfaces (e.g., [9, 20, 38, 45, 60]) and 3D content with space (e.g., [7, 54, 59]), we believe that surfaces present a strong signifier that 2D visualisations can somehow be "brought into" space. As a result, we focus on this transformation between 2D surfaces and 3D spaces in our work.

3 THE PURPOSE OF VISUALISATION TRANSFORMATIONS

As previously discussed in Section 2, we see visualisation transformations in immersive environments as animations which are initiated and controlled by user interaction for some specific purpose. Before we can identify the types of tasks to support however, it is first important to understand the types of transformations that are possible in an immersive 3D environment.
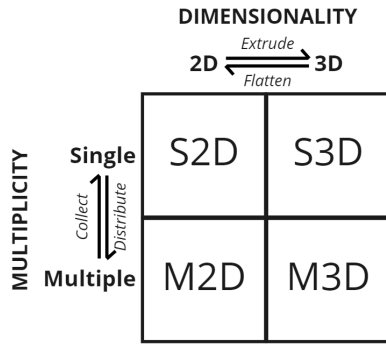
Figure 3: A matrix of the different states that a visualisation can be in while in an immersive environment. Visualisations can exist in any one of the four states in the matrix. Transformations between 2D and 3D visualisations are referred to as extrude or flatten actions; between single and multiple are referred to as distribute or collect actions.
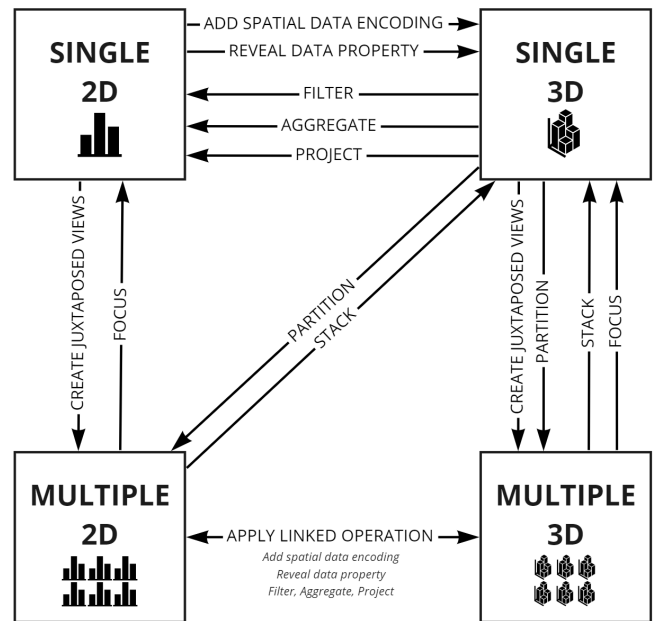

Figure 4: A mapping of possible visualisation tasks to the different transitions between the four visualisation states. A transformation technique is generally associated with one of these tasks for it to carry practical significance.

3.1 Visualisation states and transformations between them

A visualisation transformation involves a change from one visual state to another. In immersive environments, visualisations can exist either in space around the user or on a flat surface. We therefore map out how 2D and 3D visualisations can exist in these two environments in a matrix, as seen in Figure 2. This matrix describes a visualisation with two parameters: its *Dimensionality* (i.e., 2D vs 3D), and its *Environment* (i.e., surface vs space). 2D visualisations are intrinsically native to surfaces, with 3D native to space. It is also possible for a 2D visualisation to exist suspended in space, or a 3D visualisation to be projected onto a surface. While the manner in which the user perceives these from their native formats is different, the visualisation fundamentally still encodes the same data in the same visual aesthetics.

However, we find that a frequent purpose for the transformation of visualisations, especially when moving between two and three dimensions, is to change the *multiplicity* of the views and to facilitate comparison between them. For example, side-by-side views in 2D may be transitioned to a directly overlaid stack in 3D for more direct comparison, and vice-versa. We therefore rely on the following definitions for our work:

- By **Dimensionality**, we refer to whether or not a visualisation encodes meaningful information in only two dimensions on a flat plane, or in all three dimensions in a volume. This information can either be mapped directly from source data, or based on calculated/derived value(s) or model(s). For example, we consider a scatterplot that maps data along the x and y axes to be 2D, regardless if the scatterplot itself is rendered on a flat 2D surface or a virtual object floating in 3D space. In addition, the use of 3D primitives such as spheres or cones will still be considered as 2D if no data is encoded along the third spatial dimension.
- By **Multiplicity**, we refer to whether or not a visualisation is comprised of a single view or of multiple views. Certain operations or transformations can be applied to one or more

views in the visualisation simultaneously. In an immersive environment, Gestalt principles such as similarity and proximity can be used to identify multi-view visualisations [61].

These two dimensions form the four main states in Figure 3 that a visualisation may be in both before and after a transformation: single 2D (S2D), single 3D (S3D), multiple 2D (M2D), and multiple 3D (M3D). We also define a set of common terminology that help us better classify and explain visualisation transformations. Along the dimensionality axis, transforming a visualisation from 2D to 3D is **extrusion**, and the opposite direction is **flattening**. Along the multiplicity axis, from single to multiple is **distribution**, and the opposite direction is **collection**.

3.2 Mapping tasks to visualisation transformations

We now associate tasks to the different transitions between these visualisation states. As interactivity is vital in the visual analytics process [56], we focus on its use for data exploration as opposed to presentation [2]. Note however that we are not focused on visualisation authoring [48], but instead how existing visualisations may be interacted and manipulated to fulfil different purposes.

There exist many visualisation task taxonomies in the literature (e.g., [1, 25, 37, 42, 49]). However, many of these are primarily based on desktop applications that do not support both 2D and 3D visualisations simultaneously. While we initially tried to map these tasks to the different transitions in Figure 3, we soon found that numerous tasks either do not require these transformations or are not appropriate for them (e.g., record, annotate). We therefore derived and mapped a new set of tasks that are relevant to each

transition, as shown in Figure 4. As our work is inherently limited to transformations between 2D and 3D, the list of tasks we describe is not exhaustive as compared to those in related works. It comprises of 14 tasks, some of which are mirrored between states. We detail these tasks in this section.

3.2.1   Single 2D ↔ Single 3D Transformations. S2D to S3D transformations all encode some new information along the third unused dimension, causing data points to visibly "extrude" from the surface. **Add Spatial Data Encoding** encodes an additional data dimension from the original source using the third spatial dimension. This is equivalent to adding a third depth dimension to a 2D visualisation. **Reveal Data Property** is similar in that it also encodes information along the third spatial dimension, but the extent to which the third spatial dimension is used is dependent on some calculated value or property of the data (e.g., mean, node centrality).

S3D to S2D transformations involve the visualisation being "flattened" down from a 3D to a 2D object. They primarily reduce the amount of information that the visualisation encodes, usually to manage visual complexity. **Aggregate** and **Filter** are two main approaches in doing so, and our mapping reflects this. Note that we treat slicing and cutting, operations commonly used for spatial and volumetric data sets, as filter operations due to them effectively hiding portions of the visualisation. **Project** on the other hand encompasses transformations which project a 3D visualisation down to 2D. As Munzner describes [42], this is differentiated between orthographic and perspective projection. Orthographic projections simply exclude values for a given data dimension that is to be dropped in the transition from 3D to 2D, and are functionally the opposite operation to **Add Spatial Data Encoding**. Examples of these can be seen in related work, with 3D visualisations being orthographically projected down into 2D in rotational transitions [6, 16, 28]. Perspective projections on the other hand have greater implications due to the use of head-mounted displays in MR, as users can control how this projection occurs by moving their own head position or changing the orientation of the visualisation.

3.2.2   Single 2D ↔ Multiple 2D Transformations. S2D to M2D transformations all involve **Creating Juxtaposed Views**. Views are created and "distributed" from an initial 2D visualisation. While these transformations can be performed entirely on a flat 2D display, there are opportunities to distribute and arrange the views in 3D spatial layouts [39].

M2D to S2D transformations on the other hand involve "collecting" multiple visualisations together into a single one. **Isolate View** allows users to focus their attention on a single view by isolating it from the rest. As with above, single views can exist on both a surface or elsewhere in space.

3.2.3   Single 3D ↔ 2D/3D Multiple Transformations. S3D to either of the multiple visualisation states (M2D and M3D) involves the **Partitioning** of the 3D visualisation into several smaller visualisations, which get distributed in some manner. For when the target state is M2D, each partition is further reduced down to two dimensions. This would be the case when each individual partition has no benefit being in 3D. Alternatively, the partitions can remain in M3D, mainly in instances where they still encode 3D information (such as for volumetric visualisation).

Multiple visualisations to S3D involves **Stacking** them together to form a single visualisation. This can be considered as the reverse operation to Partitioning. In the specific case of S3D ↔ M3D, the same operations that exist for S2D ↔ M2D still apply.

3.2.4   Multiple 2D ↔ Multiple 3D Transformations. M2D to M3D can be seen as a mirror image of S2D to S3D. That is, any of the tasks performed to single visualisations can also be applied as a **Linked Operation** across multiple visualisations at the same time. The selection of which visualisations should be transformed may be performed either implicitly, usually due to some semantic grouping (e.g., small multiples), or explicitly defined by the user.

4   A DESIGN SPACE FOR VISUALISATION TRANSFORMATIONS

We now consider how visualisation transformations between different states can be designed by crafting the design space shown in Figure 5. It was created through an iterative semi-systematic process, where we first brainstormed and developed prototype transformation techniques (shown in Section 5.2), then updated the design space to best describe these, and repeat. We took inspiration both from related work (e.g., [27, 35, 45, 64]) and our own previous work on 2D and 3D visualisation [36, 53] during this process. The design space is meant to be read from the perspective of a designer that is creating an immersive visual analytics system for data analysts to use. The designer needs to consider both visualisation and interaction elements to make proper and effective transformations. The transformations they create can then be implemented into the system which users can then leverage for their own analysis.

The design space loosely follows the general flow of a visualisation transformation: the initial state of the visualisation that is required for the transformation to be applicable; the user interaction(s) that enable the transformation; the transformation itself; and the resulting state of the visualisation after the transformation is concluded. These roughly fall under three colour-coded categories: the visual representations of the data (in yellow), the interaction(s) that the user is performing (in blue), and the transformation itself (in green). Several design dimensions are not mutually exclusive (marked by an asterisk), meaning that multiple elements from it can be considered in the design. The design space is not prescriptive, and is intended more as a guide to aid designers in considering the various options and aspects of visualisation transformations. In this section, we describe each part of the design space in turn and their design dimensions.

4.1   Initial Visualisation State

All *Visualisation State* sections refer to the various properties of the visualisation object. The *Initial Visualisation State* is the state of the visualisation prior to the transformation occurring. In the context of the state transitions described in Section 3, this section can be viewed as a starting node in Figure 4. As our transformations are applied by the user onto existing visualisations, the Initial Visualisation State is effectively the list of conditions that needs to be met by any given visualisation for the transformation to be usable. For example, a particular transformation may only make sense to be used on a scatterplot, and not for any other types of visualisations. This visual state is then affected by the Transformation section, as
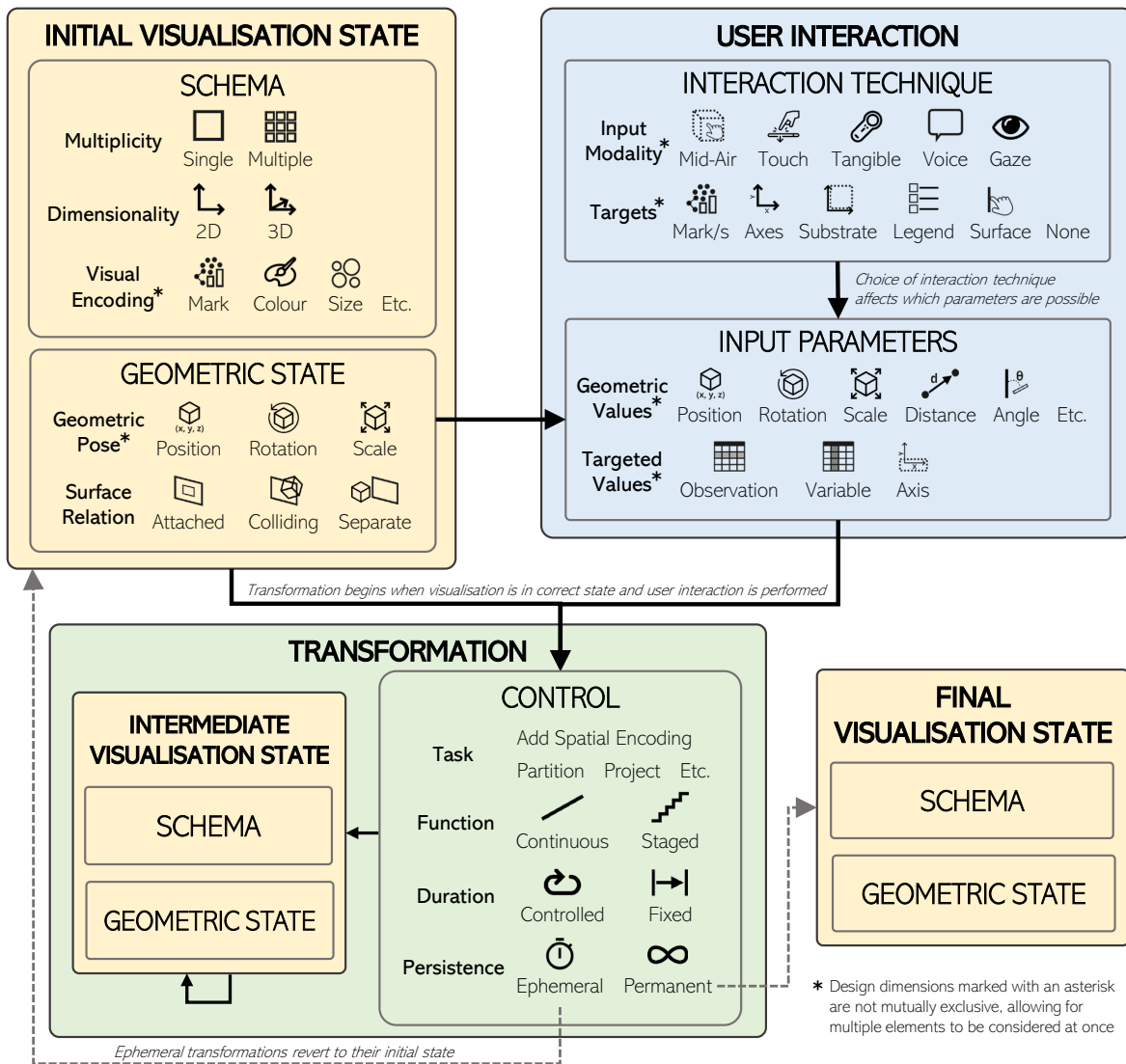
Figure 5: A design space for visualisation transformations between 2D and 3D. The *Initial Visualisation State* acts as requirements for the transformation to be available to the user. *User Interaction* is the actions the user needs to take to perform the transformation. *Transformation* refers to how the transformation progresses once triggered by the user.

denoted by the arrow. We separate this section into two categories: *Schema* and the *Geometric State*.

4.1.1 Schema. This refers to the manner in which the visualisation maps data to graphics. It has three design dimensions:

Multiplicity. This is whether or not the visualisation needs to be **Single** □ or **Multiple** ⊞ -view for the transformation to be valid. It is the same concept as discussed in Section 3.

Dimensionality. This is whether or not the visualisation needs to be **2D** ⌐ or **3D** ⌐ for the transformation to be valid. It is the same concept as discussed in Section 3.

Visual Encoding. This is the set of one or more required visual encodings for the transformation to be valid. These encodings are similar in concept to those found in other visualisation taxonomies (e.g., [42]), such as **Marks** , **Colour** , **Size** , and so forth.

4.1.2 Geometric State. This refers to the properties of the visualisation as a virtual object in the immersive three-dimensional space. It has two design dimensions:

Geometric Pose. This is the physical attributes of the visualisation in regards to its **Position** , **Rotation** , and **Scale** in the immersive three-dimensional space. As an example, a 2D visualisation may need to be facing in a direction perpendicular

to the ground, or it may need to be at a certain height from the ground, or even both conditions at the same time. Note that scale in this context refers to the overall size of the visualisation, and not its graphical marks.

Surface Relation. This is the geometric relation between the visualisation and any particular surface in the environment (e.g., walls, tables). The choice of relationship matters since it can be a strong signifier as to whether or not certain transformations are available to the user. For example, a 2D visualisation suspended in space affords different ways of interacting with it compared to one that is displayed against a surface. There are three different possibilities for this:

- **Attached** ⬚ is when a visualisation is bound to a surface. This can either be a visualisation that is physically rendered on a display, or a virtual visualisation rendered in MR that is aligned parallel to a surface.
- **Colliding** ⬚ is when a visualisation intersects with a surface, generally at a non-orthogonal angle. This is possible in MR as virtual objects can be rendered in 3D space that do not conform to real-world physics or collisions.
- **Separate** ⬚ is when a visualisation has no association with a surface. This is the case for visualisations that are suspended in space with no use of surfaces (e.g., [27, 64].

The geometric state of a visualisation can be used as input parameters which then affect how the transformation functions, as indicated by the arrow. These input parameters are explained later.

4.2 User Interaction

The *User Interaction* section refers to the interactions with the system that the user performs in order to trigger (and optionally control) the visualisation transformation. As previously mentioned, this greater focus on interactivity distinguishes our transformations from conventional animated transitions (see Section 2.1). Note that all design dimensions in this section allow for multiple elements to be used at once, allowing for possibilities such as multi-modal interaction. This section is separated into two categories with a one-way relationship: *Interaction Technique* and the Input *Parameters*.

4.2.1 Interaction Technique. This is the action that the user needs to perform in order to trigger and/or control the transformation. It has two design dimensions:

Input Modality. This is the input mode(s) that is being used in the interaction. While the set of possible modalities is dependent on the input devices available to the user, we consider a number of modalities that are commonly available on consumer MR devices such as the Microsoft HoloLens 2 in our design space:

- **Mid-air** ✍ inputs generally involve hand-tracking to detect user movements and gestures in space. Such gestures may use either direct manipulation (e.g., pinch on a virtual object) or indirect manipulation (e.g., hand wave in front of user).
- **Touch** ✍ inputs are those which are performed on a tangible surface. These inputs may be detected by a touch-enabled display, an infrared touch tracking device, or simulated via the device's hand-tracking capabilities [63].

- **Tangible** 🔑 inputs are from a physical object, such as a virtual reality controller or mouse. This may involve a button press or some movement of the object.
- **Voice** 🗩 inputs are voice commands similar to those readily supported on MR headsets.
- **Gaze** 👁 refers to the eye or head gaze of the user. Inputs can be performed using dwell-based interactions or eye gestures [29].

Targets. This refers to any specific visual element that the user is interacting with, particularly in the case of deictic techniques. It is necessary for two reasons. First, it can help distinguish between several transformations that share the same input modality by requiring the action to be performed on different targets. Second, it is an inherent selection operation that can then affect the transformation itself (explained later). We present a non-exhaustive list of possible targets for transformations:

- **Marks** ⠿ are the graphical elements on the visualisation that encode data. For our purposes, this can be any graphical primitive or higher order graphic that is perceived as being interactable. Selecting a mark may require precise selection directly on it, or instead select the closest mark to the interact point.
- **Axes** ↱ are the visible axes objects that represent the coordinate system. All components of an axis (e.g., line, ticks, labels) can be encapsulated into a single target, or each one can be individually targetable for higher granularity.
- **Substrate** ⌐ is the spatial region which the graphical marks exist in. As compared to marks, substrate refers to the broader area whereby there is no requirement to interact with any specific mark.
- **Legend** ▤ is the legend that is associated with the visualisation, allowing for selection of a specific categorical item or value in a continuous range.
- **Surface** ✍ is any physical or virtual surface in the environment. This can be one which a visualisation is displayed on, or a different surface entirely.
- **None** is simply the case where there is no target involved in the interaction technique. This would be for non-deictic interactions, such as hand waves and gestures in a general area in front of the user.

The notion of coupling input modality and target together is particularly useful when considering how embodied interaction can be used for these transformations in MR. Related work in Immersive Analytics has used embodied interaction principles and paradigms to "reach through" and manipulate data visualisations in meaningful ways (e.g., [4, 14, 64]), and it is important our design space also captures this. Use of embodied interaction can also be beneficial as it may carry certain connotations or metaphors that aid in the use of the transformation technique. For example, directly interacting with a specific mark on a visualisation may imply that the transformation affects the data point in some fashion, as opposed to something more abstract like a command line interface.

4.2.2 Input Parameters. These are the values which are derived from the manner which the user performs the aforementioned interaction technique(s), hence the one-way arrow. These input

parameters are then used as part of the transformation in order for the user to control how it behaves and/or how it progresses. As the possibilities here are virtually endless, we opt for a more abstract view of these parameters in terms of how they are derived. Note that the Geometric Pose of the visualisation are also valid input parameters. We describe two broad forms of input parameters:

Geometric Values. These are values that are derived from the geometric properties of both the interaction technique(s) and the visualisation. Certain values may need to be calculated on-the-fly based on certain mathematical functions or interaction metaphors. These are useful when simple numerical values are needed to control the transformation, such as a time variable that progresses through a transformation (e.g., [33]). Examples of these values are:

- **Position** ⬡, **Rotation** ⬡, and **Scale** ⬡, which can either be directly from the visualisation's Geometric Pose, or in some cases from a tracked tangible object or controller.
- **Distance** ⬈ is a numerical value between some start and end point. Examples are the distance between both the user's hands, the distance between a visualisation from a surface, or the distance which the user has dragged when performing a drag-and-drop interaction.
- **Angle** ⬈ is a numerical value comparing the rotation of one object to another. Examples are the angle between a visualisation and a surface, or the angle between a visualisation and the floor plane.

Targeted Values. These are values that are derived from the selected Target(s) as part of the Interaction Technique. These relate to various aspects of the underlying dataset, which may be necessary for the transformation to function. For example, a transformation may aggregate itself based on a particular data variable, which is defined through the targeting process. Examples of values are:

- **Observation** ▦ and **Variable** ▦ respectively refer to one or more elements or data dimensions in the dataset. Observations may be selected by targeting specific marks on the substrate or categories in the legend, whereas variables may be selected by targeting an axis (and therefore its corresponding variable).
- **Axis** ⬚ refers specifically to a spatial axis (i.e., x, y, or z) and not any corresponding variable. This may be used in instances where the data variable is not important, and only the spatial axis that the user is interested in.

4.3 Transformation

The *Transformation* section is how the transformation itself progresses. When a visualisation is in the correct state (based on the Initial Visualisation State) and the user is performing the necessary inputs (based on the User Interaction), only then does the transformation occur. At its core, the transformation constantly updates the visualisation to an *Intermediary Visualisation State*, changing its Schema and/or its Geometric State. This constant changing of properties is, in essence, the output of the transformation before it fully terminates into a *Final Visualisation State* (or back to the initial state, see below).

A core component of this section that is not explicitly described in the design space of Figure 5 is the underlying operations that map input parameters and data to the intermediate visualisation state and its visual encodings. This is due to the overwhelming number of ways that this mapping may be performed, especially when considering factors such as data types and transformations, visualisation idioms, and any other bespoke visualisation customisations. In order to still sufficiently describe the design of the transformation process, we take a generalised, high-level approach in an overall category which we refer to as *Control*.

4.3.1 Control. As described, this is the manner in which the transformation itself behaves. It is comprised of four design dimensions:

Task. This is the intended purpose of the transformation. For example, the transformation may add a particular spatial encoding, partition the visualisation into several segments, or project a visualisation down to 2D. As this inherently applies a constraint on the possible Multiplicity and Dimensionality choices (as per Section 3), choosing this task would be one of the first decisions made when designing transformations.

Function. This is the manner and frequency in which the Intermediate Visualisation State is visibly changed as transformation progresses. We identify two main types of transformations:

- **Continuous** ⟋ transformations have smooth visual changes throughout the entire transformation. This is similar in concept to the typical approach of keyframe animation.
- **Staged** ⬈ transformations have staged visual changes throughout the entire transformation. This is similar in concept to staggering approaches in conventional animated transitions [11, 24].

Duration. This refers to how long the transformation lasts and when it terminates. There are two possible options:

- **Controlled** ⟳ transformations are those in which the user has direct control over how the visualisation changes throughout the transformation process. The intermediate visualisation state is procedurally generated based on the input parameters, which in turn are based on the user's input. As a result, the transformation will terminate whenever the user stops interacting. This can be particularly useful when the intermediate visualisation state sufficiently reveals information to the point where reaching an "optimal" final state is not necessary, allowing the user to cut short the transformation and move on.
- **Fixed** ⊢→⊣ transformations are akin to conventional animated transitions in that they change over a certain period of time. While the user may still influence the final state of the visualisation via the input parameters, they otherwise have no control over how the transformation progresses once it has started. The duration of the transformation may be predefined by the designer, or chosen by the user via an input parameter. The duration may also be zero in the case of jump cuts.

Persistence. This refers to whether or not the effects of the transformation persist on the visualisation after the transformation has terminated, regardless of whether or not its duration is Controlled or Fixed. Two main possibilities exist:

- **Ephemeral** ⏲ transformations reset back to their Initial Visualisation State. This reset may be instantaneous, or a

smooth animation that effectively plays the transformation in reverse. This is mostly relevant for Controlled transformations, as the intermediate visualisation state may no longer be desirable to retain, hence the need for an automatic reset.

- **Permanent ∞** transformations are just that: the visualisation permanently remains in the Final Visualisation State until it is modified further, or the user does some action to reverse the transformation. Note that when terminating Controlled transformations, the visualisation may skip to a predefined Final Visualisation State, or it may be a freeze-frame of the Intermediate State at that point in time. In the latter case, we do not consider the visualisation to have actually reached the Final State yet from a design standpoint.

4.4 Final Visualisation State

Lastly, the *Final Visualisation State* section refers to the properties of the visualisation once the transformation has concluded. We consider this to be the destination node in any visual state transition in Figure 4. The design dimensions and properties are the same as in the Initial State, except these no longer act as conditionals, but instead as a set of changes that have occurred throughout the transformation.

5 USING THE DESIGN SPACE

In this section, we demonstrate how our design space can be used to describe existing examples of visualisation transformations found in related work in immersive analytics. We then demonstrate how the design space may be used to create new visualisation transformations by presenting a set of techniques that we believe best showcases its use.

5.1 Transformations in Related Work

As described in Section 2.3, related work has shown examples of transforming visualisations between 2D and 3D through user interaction in order to serve various purposes.

In *FiberClay* [27], the user can **Project** a **Single □** -view, **3D ⌞** trajectory visualisation of flights down into **2D ⌞** . This corresponds with a S3D to S2D transition. Besides its dimensionality and multiplicity, no specific visual encoding or geometric state is necessary. With two **Tangible** 🔑 handheld controllers, the user presses down a button on both controllers in the general space around them (i.e., no target) to initiate the transformation. Depending on the **Distance** and **Angle** of the two controllers in relation to each other, the 3D visualisation's **Scale** along a single axis is modified in a **Continuous** fashion, effectively projecting it down to 2D as this value approaches zero. The transformation is **Controlled ↻** by the user at all times, with the visualisation being left as a **Permanent ∞** S2D visualisation upon completion.

Tilt Map [64] actually consists of two separate transformations, each occurring at separate intervals. The first is a S2D to an S3D transition as it goes from **2D ⌞** choropleth to a **3D ⌞** prism map. This **Adds a Spatial Encoding**, adjusting the height of each prism along the new spatial axis. The height is dependent on the **Rotation** of the visualisation's geometric pose relative to the horizontal plane. Note that the interaction technique here is irrelevant, as the rotation of the visualisation may be controlled by any means.

Within the transformation, the height adjustment is handled in a **Continuous** fashion and is **Controlled ↻** by the user. The second is a S3D to S2D transition where the **3D ⌞** prism map is **Projected** into a **2D ⌞** bar chart. In a similar fashion, the **Rotation** of the visualisation is used to affect the progress of the transition into the bar chart. In both cases, the results of the transformation are **Permanent ∞** , until of course the user performs the reverse action.

5.2 Example Visualisation Transformations

We now showcase a number of visualisation transformation techniques in the context of the design space, which were developed using the Unity3D game engine and the *Immersive Analytics Toolkit* [13]. The code can be found on a publicly available GitHub repository[1]. The application was deployed on a Microsoft HoloLens 2 headset, which also allowed us to leverage its hand and surface detection capabilities. All of our techniques use a regular vertical wall as the surface, and primarily make use of mid-air interaction.

We describe one of these techniques in detail from a design perspective as though we are creating one such transformation from scratch. We then briefly describe our remaining techniques to demonstrate the range of possibilities when designing visualisation transitions. For readability, these latter techniques are separated into three categories roughly based on Figure 3: *extrusion*-based transformations from 2D to 3D, *flatten*-based transformations from 3D to 2D, and other generic transformation methods that can be adapted to any visualisation type. We include images in three general phases: the visualisation before the transformation, it during the transformation, and it after the transformation, with pictograms of their respective design space dimensions shown below. Please see the supplemental material for a video version of these techniques.

5.2.1 Example Usage of the Design Space. Scatterplots are commonly used to visualise bivariate data. They however are subject to overplotting, particularly when the data is densely packed. To allow users to get around this, we want to somehow allow them to **Reveal a Data Property**, which in this case is the degree that each point is overplotted. From Figure 4, we see that this is suited for extrusion transformations between S2D and S3D visualisations. Therefore, we first require that our visualisation be a **Single □** -view **2D ⌞** scatterplot in the Initial Visualisation State for the transformation to be available. As 2D visualisations can be considered native to surfaces and 3D to space (Figure 2), we want to emulate our data being extruded into 3D space. Therefore, we also require that the Initial Visualisation be **Attached** to any surface.

Through this, we can leverage the additional dimension to encode our calculated overplotting property, translating each point in a smooth **Continuous** function. The geometric length of this dimension as it extrudes outwards can either be Fixed or Controlled by the user. In this case, we choose **Controlled ↻** as we want our system to be as interactive as possible.

While we have effectively worked backwards by considering our transformation first, we now know what inputs we need from the user to control this geometric length. As we need a linear value to map to length, we leverage a "pinch-and-drag" metaphor using

---

[1]https://github.com/benjaminchlee/2D-3D-MR-Transformations

Mid-air 🖐 interaction, allowing us to derive a **Distance** 🖈 value between the initial and current interaction points. This initial pinch can be anywhere on the scatterplot's **Substrate** ⬓ , as we do not require any specific data values to be selected. While the exact declaration of how the input parameter(s) affects the Intermediate Visaul State is not part of the design space, we choose to **Scale** ⬡ the third dimension's length based on this new Distance value.

Lastly, we consider whether or not we want the effects of the transformation to be permanent or not. As overplotting is something which analysts should be aware of, but not necessarily focus their exploration around, we can safely have the transformation be **Ephemeral** Ō , as its state should automatically reset once the user lets go of their pinch gesture.

Overplotting Extrusion
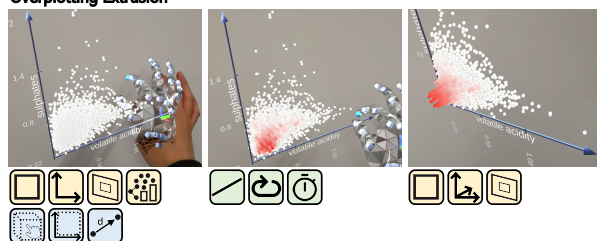
5.2.2 Extrusion-based Techniques.
Extrusion transformations transform a visualisation from 2D to 3D. All of our extrusion techniques use a "grab and pull" metaphor with a mid-air pinch gesture, which simulates the visualisation being stretched or extruded outwards from the surface.

Shortest Paths Extrusion is used to reveal a data property, in this case the the shortest paths between nodes in a single 2D network. It requires any single 2D network visualisation that is attached to a surface. The user needs to grab onto two different nodes using mid-air interaction, one with each hand, which then act as input parameters to the transformation. Using a continuous function, the nodes are all extruded along the third spatial dimension based on the shortest path distance between the two selected nodes calculated using the Floyd-Warshall algorithm. This extrusion distance is also scaled depending on the minimum distance of either hand from the surface.

Shortest Paths Extrusion
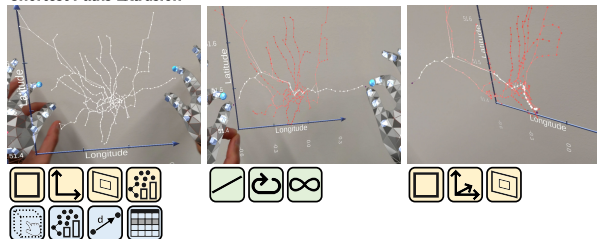
Histogram Extrusion is used to reveal the distribution within each categorical value in a single 2D bar chart. By grabbing onto any part of the visualisation substrate, each bar extrudes outwards into 3D, splitting into multiple histograms that are read perpendicular to the surface. This transformation follows a staged function, as the number of bins in the extruded histograms is controlled by the user based on the distance from their hand to the surface.

Histogram Extrusion
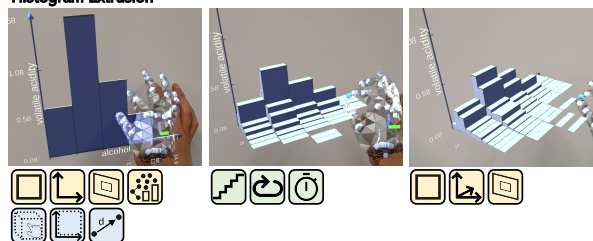
Parallel Coordinates Plot Extrusion creates juxtaposed views stemming from a single 2D scatterplot, which is initiated by grabbing onto either of the two axes. As the user pulls away from the surface, new 2D scatterplots are created at different stages depending on the distance from the surface to their hand. The axis that was grabbed is an input parameter, which sets the fixed dimension across all created scatterplots, with the other axis dimension set to the next most correlated dimension. The geometric position of each view is also set by the transformation to be equidistant to each other. The 2D scatterplots are each linked together, forming a 3D parallel coordinates plot.

Parallel Coordinates Plot Extrusion
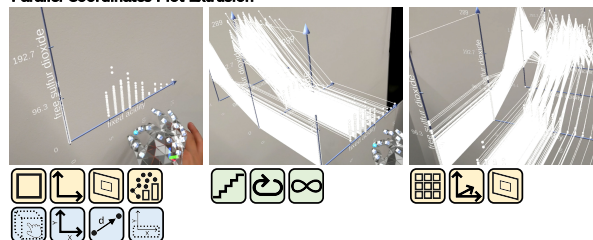
5.2.3 Flatten-Based Techniques.
Flatten transformations transform a visualisation from 3D to 2D. All of our flattening techniques rely on a 'push into surface' metaphor, whereby the user holds onto and collides a 3D visualisation into a surface to squish or flatten it.

3D Scatterplot Projection projects a single 3D scatterplot that is colliding with a surface down to two dimensions. The initial placement of the scatterplot is performed to get it to collide with the surface is outside of the transformation process. When the movement operation is finished however (in this case the grab action), the transformation instantly projects all data points from 3D to 2D depending on the rotation of the scatterplot and the user's viewpoint.

3D Scatterplot Projection
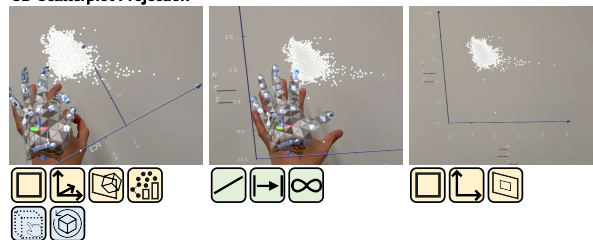
3D Bar Chart Partitioning partitions a single 3D bar chart into multiple 2D bar charts. As the 3D bar chart collides with a surface, a staged transformation occurs whereby rows of bars are partitioned off from the 3D bar chart and animate towards the surface. The

number of partitions is dependent on the controlled position and rotation of the bar chart to the surface.

3D Bar Chart Partitioning

Volumetric Surface Slicing filters out all but a slice of a single 3D visualisation that is colliding with a surface. As the user controls the distance and rotation of the volume in relation to the surface by moving it around, the transformation continuously filters out all parts of the volume that are not touching the surface, creating a single 2D slice that is visibly attached to the surface.

Volumetric Surface Slicing

5.2.4 Generic Techniques. These techniques do not strictly relate to any specific visualisation type and therefore can be adapted to many different situations.
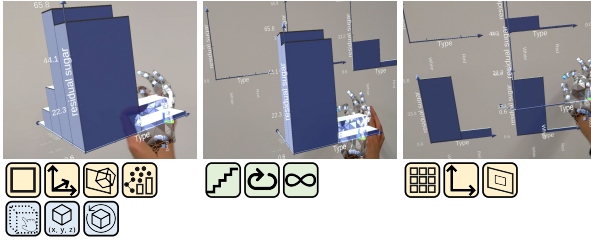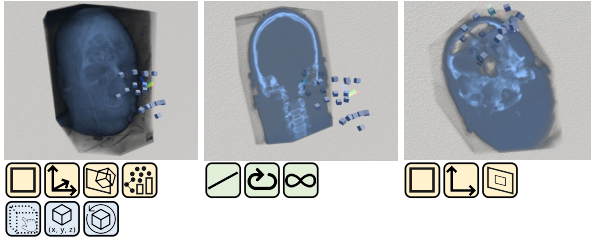
Apply Linked Operation applies a linked extrusion or reduction operation on multiple 2D or 3D visualisations at the same time, turning an M2D into M3D or vice versa. In this specific example, an Overplotting Extrusion transformation is applied to all 2D scatterplots in a matrix using the same interaction technique. As the user may wish to compare different views with each other, the transformation effects are permanent.

Apply Linked Operation
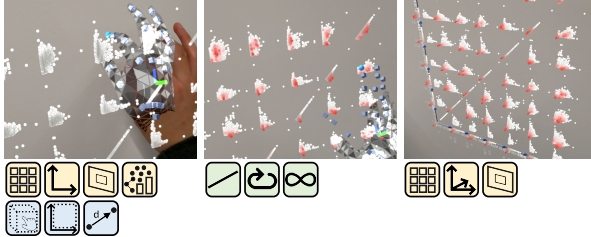
Matrix Extrusion creates juxtaposed views from a single attached 2D scatterplot or other visualisation by grabbing onto the top-right corner of its substrate. As the user drags their hand diagonally towards the top-right, a matrix of 2D charts is created and expanded in stages, based on the distance between the start grab point and the current position of the user's hand.

Matrix Extrusion
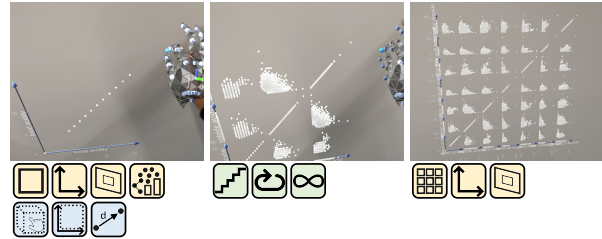
6 DISCUSSION, LIMITATIONS, AND FUTURE WORK

We believe our design space provides a rich framework for the design and creation of visualisation transformations between 2D and 3D. During the process of creating both the design space and our exemplary techniques, we identified some important considerations and questions that merit future research, as these are beyond the scope of this paper. We reflect on some of these in this section.

Using the design space and design principles. Our proposed design space can be used to help guide the creation of visualisation transformation in immersive environments. In contrast to prior work in animated transitions (e.g., [32, 58]), we do not provide a grammar with a strict syntax. It relies on the designer to take into consideration the sorts of input modalities, interactions, data types, and contexts in which they are designing these visualisation transformations for. As we have not conducted a user study or an evaluation with experts however, we cannot give concrete design principles on how best to make these decisions. We can however derive some general guidelines based off prior knowledge in both visualisation and HCI research. First, visualisation transformations should still aim to minimise the use of 3D as much as possible [42]. This may either be through flattening 3D visualisations down into 2D, or by using more ephemeral transformations such that 3D is only used sparingly and in short bursts. Second, following direct manipulation principles [52], the intermediary visual state during a transformation should closely match the user interaction as much as possible, such as the "grab-and-pull" metaphor of our extrusion techniques, or the tilt metaphor found in *Tilt Map* [64]. Third, the number of visual changes that occur throughout the transformation should be kept to a minimum. While regular animated transitions may undergo several visual encoding changes at once, necessitating the need for staggering to minimise visual complexity [11, 24], transformations between 2D and 3D can bypass this issue by simply not modifying as many visual variables at the same time. For example, our overplotting extrusion technique does not need rank the frequency of each overplotted x-y coordinate, as the depth dimension (and colour) is sufficient enough for seeing this information.

Use of 3D visualisations and the 'peeking' metaphor. Some of our exemplary techniques use ephemeral transformations because, as described in Section 4.3, the information they reveal does not serve any purpose in the long term. For example, once a user is able to identify which data points are overplotted, the effects of the transformation can be discarded as this information is now known to the user. This act of taking a 'peek' at the data in 3D is similar to techniques such as *ScatterDice* [16] and *GraphDice* [6] for maintaining transition awareness. In both cases, 3D is used sparingly and

only in short bursts, rather than relying on it for an extended period of time. While it is possible to achieve similar results by changing visualisation encodings or creating different views of the data on 2D visualisations, we believe that these peeking transformations do not involve a lot of cognitive processes to perform such that the peeking metaphor can provide much timelier access to the required information. This also adheres to our aforementioned guideline as to avoid relying too heavily on 3D.

Discoverability and configuration of visualisation transformations. The notion of extruding objects from a 2D surface into a 3D MR environment is not new in the literature. However, previous work had typically consisted only of a single technique that applies to the entire object of interest, such as a hand motion anywhere on the surface (e.g., [5, 44]), picking up a physical object (e.g., [18]), pulling on a visible widget (e.g., [51]), or pressing a button on a handheld controller (e.g., [27, 30]). With our work, we demonstrated techniques that can coexist simultaneously on any given visualisation (i.e., overplotting, parallel coordinates plot, and matrix extrusions) that are accessed through direct interaction. This naturally raises concerns of discoverability. As our design space was crafted with Immersive Analytics system designers in mind, they will need to consider how users might learn how to access these transformations. It is clear that some form of instruction, guidance, or affordance needs to be given to the user, especially when so many configurations of data representations are possible—and therefore a combinatorial explosion of different possible transformations. Future work may seek to provide further structure and standardisation to the design space, such that any given interaction technique will result in a consistent form of transformation no matter the visualisation schema or data type. Moreover, future work might circumvent the issue altogether by exposing the capability of creating and configuring visualisation transformations to the end-user directly, allowing them to tailor transformations in a manner which they find most useful for their given task.

Chaining and branching transformations. While we demonstrated our design space as being capable of describing existing transformations in the literature (Section 5.1), it still assumes that each visualisation transformation is a discrete standalone action with a clear initial and final state. What this does not fully capture however is the chaining and/or branching of different transformations together. This is made evident when considering *Tilt Map* by Yang et al. [64]. While we considered its transformations (choropleth to prism and prism to bar) as two distinct transformations that share the same interaction modalities and parameters, it can also be considered as one larger transformation but with two main stages—all performed with the same action. Moreover, there may be the possibility for branching transformations that change course depending on how this action is performed. In the case of our extrusion techniques, for example, depending on whether the user pulls away from the surface orthogonally or at an angle, a different follow-up transformation can be applied. We believe that some higher level view of how these different transformations are linked with each other is necessary to adequately understand and design such techniques.

Transformations involving different surfaces. For the purposes of generalisability, our design space uses an abstract notion of "surface". In reality, there exist many different types of surfaces (e.g., tabletops, tablets, phones) that each have their own considerations to take into account, such as orientation, size, and portability. Despite all of our techniques being demonstrated on a single vertical flat wall, it is to be expected that not all would be practical or effective when applied to other types of surfaces. It is highly likely that this distinction needs to be recognised fully in future work. Furthermore, while we specified these surfaces to be flat planes, there may be instances where visualisations can be attached to or even rendered on non-Euclidean surfaces. Such possibilities lie firmly outside the scope of this work however.

7 CONCLUSION

In this paper, we investigated the design of visualisation transformations between 2D and 3D in mixed-reality environments. We first discussed the relationship between 2D and 3D visualisations with surfaces and spaces, followed by describing the possible tasks that these transformations can fulfil when transitioning between four main states: single-view 2D, single-view 3D, multi-view 2D, and multi-view 3D. We then proposed a design space which designers of Immersive Analytics tools can use to create these transformations, which their end users may then make use of in their own analysis. This considers numerous factors such as the form of user input required, whether or not the transformation is directly controlled by the user throughout its animation, and if the effects of the transformation are permanent or not. We then demonstrated this design space with a set of example transformation techniques that were developed for the Microsoft HoloLens 2 headset. We believe that the use of 2D and 3D visualisations in this fashion will help inspire a different paradigm of Immersive Analytics, where data visualisations can freely flow between surfaces and space depending on the needs of the user.

ACKNOWLEDGMENTS

REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko. 2005. Low-Level Components of Analytic Activity in Information Visualization. In *Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05)*. IEEE, Minneapolis, MN, USA, 15–15. https://doi.org/10.1109/INFOVIS.2005.24

[2] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Jason Leboe-McGowan, and Pourang Irani. 2018. Hooked on Data Videos: Assessing the Effect of Animation and Pictographs on Viewer Engagement. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. ACM, Castiglione della Pescaia Grosseto Italy, 1–9. https://doi.org/10.1145/3206505.3206552

[3] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2014. Visualizing Dynamic Networks with Matrix Cubes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 877–886. https://doi.org/10.1145/2556288.2557010

[4] Andrea Batch, Andrew Cunningham, Maxime Cordeil, Niklas Elmqvist, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2020. There Is No Spoon: Evaluating Performance, Space Use, and Presence with Expert Domain Users in Immersive Analytics. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 536–546. https://doi.org/10.1109/TVCG.2019.2934803

[5] H. Benko, E.W. Ishak, and S. Feiner. 2005. Cross-Dimensional Gestural Interaction Techniques for Hybrid Immersive Environments. In *IEEE Proceedings. VR 2005.*

Virtual Reality, 2005. IEEE, Bonn, Germany, 209–327. https://doi.org/10.1109/VR.2005.1492776

[6] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. D. Fekete. 2010. Graphdice: A System for Exploring Multivariate Social Networks. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization* (Bordeaux, France) *(EuroVis'10)*. The Eurographs Association & John Wiley & Sons, Ltd., Chichester, GBR, 863–872. https://doi.org/10.1111/j.1467-8659.2009.01687.x

[7] 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* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 90, 12 pages. https://doi.org/10.1145/3173574.3173664

[8] Marco Cavallo, Mishal Dholakia, Matous Havlena, Kenneth Ocheltree, and Mark Podlaseck. 2019. Dataspace: A Reconfigurable Hybrid Reality Environment for Collaborative Information Analysis. arXiv:1903.03700 [cs.HC]

[9] Marco Cavallo, Mishal Dolakia, Matous Havlena, Kenneth Ocheltree, and Mark Podlaseck. 2019. Immersive Insights: A Hybrid Analytics System ForCollaborative Exploratory Data Analysis. In *25th ACM Symposium on Virtual Reality Software and Technology* (Parramatta, NSW, Australia) *(VRST '19)*. Association for Computing Machinery, New York, NY, USA, Article 9, 12 pages. https://doi.org/10.1145/3359996.3364242

[10] Tom Chandler, Maxime Cordeil, Tobias Czauderna, Tim Dwyer, Jaroslaw Glowacki, Cagatay Goncu, Matthias Klapperstueck, Karsten Klein, Kim Marriott, Falk Schreiber, and Elliot Wilson. 2015. Immersive Analytics. In *2015 Big Data Visual Analytics (BDVA)*. IEEE, Hobart, Australia, 1–8. https://doi.org/10.1109/BDVA.2015.7314296

[11] Fanny Chevalier, Pierre Dragicevic, and Steven Franconeri. 2014. The Not-so-Staggering Effect of Staggered Animated Transitions on Visual Tracking. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 2241–2250. https://doi.org/10.1109/TVCG.2014.2346424

[12] Christopher Collins and Sheelagh Carpendale. 2007. VisLink: Revealing Relationships Amongst Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1192–1199. https://doi.org/10.1109/TVCG.2007.70521

[13] Maxime Cordeil, Andrew Cunningham, Benjamin Bach, Christophe Hurter, Bruce H. Thomas, Kim Marriott, and Tim Dwyer. 2019. IATK: An Immersive Analytics Toolkit. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Osaka, Japan, 200–209. https://doi.org/10.1109/VR.2019.8797978

[14] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017. 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* (Québec City, QC, Canada) *(UIST '17)*. ACM, New York, NY, USA, 71–83. https://doi.org/10.1145/3126594.3126613

[15] Maxime Cordeil, Christophe Hurter, Stéphane Conversy, and Mickaël Causse. 2013. Assessing and Improving 3D Rotation Transition in Dense Visualizations. In *Proceedings of the 27th International BCS Human Computer Interaction Conference* (London, UK) *(BCS-HCI '13)*. BCS Learning & Development Ltd., Swindon, GBR, Article 7, 10 pages.

[16] N. Elmqvist, P. Dragicevic, and J. Fekete. 2008. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov 2008), 1539–1148. https://doi.org/10.1109/TVCG.2008.153

[17] Barrett Ens, Benjamin Bach, Maxime Cordeil, Ulrich Engelke, Marcos Serrano, Wesley Willett, Arnaud Prouzeau, Christoph Anthes, Wolfgang Büschel, Cody Dunne, Tim Dwyer, Jens Grubert, Jason H. Haga, Nurit Kirshenbaum, Dylan Kobayashi, Tica Lin, Monsurat Olaosebikan, Fabian Pointecker, David Saffo, Nazmus Saquib, Dieter Schmalstieg, Danielle Albers Szafir, Matt Whitlock, and Yalong Yang. 2021. *Grand Challenges in Immersive Analytics*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3411764.3446866

[18] B. Ens, S. Goodwin, A. Prouzeau, F. Anderson, F. Y. Wang, S. Gratzl, Z. Lucarelli, B. Moyle, J. Smiley, and T. Dwyer. 2021. Uplift: A Tangible and Immersive Tabletop System for Casual Collaborative Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1193–1203. https://doi.org/10.1109/TVCG.2020.3030334

[19] Barrett Ens and Pourang Irani. 2017. Spatial Analytic Interfaces: Spatial User Interfaces for In Situ Visual Analytics. *IEEE Computer Graphics and Applications* 37, 2 (March 2017), 66–79. https://doi.org/10.1109/MCG.2016.38

[20] Barrett Ens, Eyal Ofek, Neil D. B. Bruce, and Pourang Irani. 2016. Shared Façades: Surface-Embedded Layout Management for Ad Hoc Collaboration Using Head-Worn Displays. In *Collaboration Meets Interactive Spaces*, Craig Anslow, Pedro F. Campos, and Joaquim A. Jorge (Eds.). Springer, Cham, 153–176. https://doi.org/10.1007/978-3-319-45853-3_8

[21] E. Fanea, S. Carpendale, and T. Isenberg. 2005. An interactive 3D integration of parallel coordinates and star glyphs. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, Minneapolis, MN, USA, 149–156. https://doi.org/10.1109/INFVIS.2005.1532141

[22] Cleotilde Gonzalez. 1996. Does Animation in User Interfaces Improve Decision Making?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Common Ground - CHI '96*. ACM Press, Vancouver, British Columbia, Canada, 27–34. https://doi.org/10.1145/238386.238396

[23] Antonio Gracia, Santiago González, Víctor Robles, Ernestina Menasalvas, and Tatiana von Landesberger. 2016. New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. *Information Visualization* 15, 1 (2016), 3–30. https://doi.org/10.1177/1473871614556393 arXiv:https://doi.org/10.1177/1473871614556393

[24] Jeffrey Heer and George Robertson. 2007. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1240–1247. https://doi.org/10.1109/TVCG.2007.70539

[25] Jeffrey Heer and Ben Shneiderman. 2012. Interactive Dynamics for Visual Analysis: A taxonomy of tools that support the fluent and flexible use of visualizations. *Queue* 10, 2 (Feb. 2012), 30–55. https://doi.org/10.1145/2133416.2146416

[26] Sebastian Hubenschmid, Johannes Zagermann, Simon Butscher, and Harald Reiterer. 2021. *STREAM: Exploring the Combination of Spatially-Aware Tablets with Augmented Reality Head-Mounted Displays for Immersive Analytics*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445298

[27] 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. https://doi.org/10.1109/TVCG.2018.2865191

[28] C. Hurter, B. Tissoires, and S. Conversy. 2009. FromDaDy: Spreading Aircraft Trajectories Across Views to Support Iterative Queries. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1017–1024. https://doi.org/10.1109/TVCG.2009.145

[29] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. 2012. Gaze Gestures or Dwell-Based Interaction?. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 229–232. https://doi.org/10.1145/2168556.2168602

[30] Bret Jackson and Daniel F. Keefe. 2016. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (April 2016), 1442–1451. https://doi.org/10.1109/TVCG.2016.2518099

[31] Raphaël James, Anastasia Bezerianos, Olivier Chapuis, Maxime Cordeil, Tim Dwyer, and Arnaud Prouzeau. 2020. Personal+Context navigation: combining AR and shared displays in Network Path-following. In *Proceedings of Graphics Interface 2020* (University of Toronto) *(GI 2020)*. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, Toronto, Canada, 267 – 278. https://doi.org/10.20380/GI2020.27

[32] Younghoon Kim and Jeffrey Heer. 2021. Gemini: A Grammar and Recommender System for Animated Transitions in Statistical Graphics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (Feb. 2021), 485–494. https://doi.org/10.1109/TVCG.2020.3030360

[33] Brittany Kondo and Christopher Collins. 2014. DimpVis: Exploring Time-Varying Information Visualizations by Direct Manipulation. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 2003–2012. https://doi.org/10.1109/TVCG.2014.2346250

[34] M. Kraus, N. Weiler, D. Oelke, J. Kehrer, D. A. Keim, and J. Fuchs. 2020. The Impact of Immersion on Cluster Identification Tasks. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (Jan 2020), 525–535. https://doi.org/10.1109/TVCG.2019.2934395

[35] Ricardo Langner, Marc Satkowski, Wolfgang Büschel, and Raimund Dachselt. 2021. *MARVIS: Combining Mobile Devices and Augmented Reality for Visual Data Analysis*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3411764.3445593

[36] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. 2021. Shared Surfaces and Spaces: Collaborative Data Visualisation in a Co-located Immersive Environment. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1171–1181. https://doi.org/10.1109/TVCG.2020.3030450

[37] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors novel evaluation methods for information visualization - BELIV '06*. ACM Press, Venice, Italy, 1. https://doi.org/10.1145/1168149.1168168

[38] Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae. 2018. Projective Windows: Bringing Windows in Space to the Fingertip. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Montreal QC, Canada, 1–8. https://doi.org/10.1145/3173574.3173792

[39] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. 2020. Design and Evaluation of Interactive Small Multiples Data Visualisation in Immersive Spaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Atlanta, GA, USA, 588–597. https://doi.org/10.1109/VR46266.2020.00081

[40] Tahir Mahmood, Erik Butler, Nicholas Davis, Jian Huang, and Aidong 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)*. IEEE, Hobart, Australia, 1–11. https://doi.org/10.1109/BDVA.2018.8533893

[41] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H. Thomas (Eds.). 2018. *Immersive Analytics*. Lecture Notes in Computer Science, Vol. 11190. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-01388-2

[42] Tamara Munzner. 2014. *Visualization Analysis and Design* (0 ed.). A K Peters/CRC Press, New York. https://doi.org/10.1201/b17511

[43] Arnaud Prouzeau, Maxime Cordeil, Clement Robin, Barrett Ens, Bruce H. Thomas, and Tim Dwyer. 2019. Scaptics and Highlight-Planes: Immersive Interaction Techniques for Finding Occluded Features in 3D Scatterplots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300555

[44] Patrick Reipschläger and Raimund Dachselt. 2019. DesignAR: Immersive 3D-Modeling Combining Augmented Reality with Interactive Displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (Daejeon, Republic of Korea) *(ISS '19)*. Association for Computing Machinery, New York, NY, USA, 29–41. https://doi.org/10.1145/3343055.3359718

[45] P. Reipschlager, T. Flemisch, and R. Dachselt. 2021. Personal Augmented Reality for Information Visualization on Large Interactive Displays. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1182–1192. https://doi.org/10.1109/TVCG.2020.3030460

[46] George Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. 2002. Animated visualization of multiple intersecting hierarchies. *Information Visualization* 1, 1 (2002), 50–65. https://doi.org/10.1057/palgrave.ivs.9500002

[47] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. 2008. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov. 2008), 1325–1332. https://doi.org/10.1109/TVCG.2008.125

[48] Arvind Satyanarayan, Bongshin Lee, Donghao Ren, Jeffrey Heer, John Stasko, John Thompson, Matthew Brehmer, and Zhicheng Liu. 2020. Critical Reflections on Visualization Authoring Systems. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 461–471. https://doi.org/10.1109/TVCG.2019.2934281

[49] Hans-Jorg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. 2013. A Design Space of Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2366–2375. https://doi.org/10.1109/TVCG.2013.120

[50] Mickael Sereno, Lonni Besançon, and Tobias Isenberg. 2019. Supporting Volumetric Data Visualization and Analysis by Combining Augmented Reality Visuals with Multi-Touch Input. In *EuroVis 2019 - Posters*, João Madeiras Pereira and Renata Georgia Raidou (Eds.). The Eurographics Association, Porto, Portugal, 21–23. https://doi.org/10.2312/eurp.20191136

[51] Wu Shengzhi. 2019. Pulling a 3D model from an editor display (e.g. Maya, C4D) to a real-world environment, as if the screen is a drawer [Tweet]. https://twitter.com/Wu_Shengzhi/status/1155725029310521344/. Accessed: 2021-11-25.

[52] Ben Shneiderman. 1982. The Future of Interactive Systems and the Emergence of Direct Manipulation†. *Behaviour & Information Technology* 1, 3 (July 1982), 237–256. https://doi.org/10.1080/01449298208914450

[53] Jim Smiley, Benjamin Lee, Siddhant Tandon, Maxime Cordeil, Lonni Besançon, Jarrod Knibbe, Bernhard Jenny, and Tim Dwyer. 2021. The MADE-Axis: A Modular Actuated Device to Embody the Axis of a Data Dimension. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 501 (nov 2021), 23 pages. https://doi.org/10.1145/3488546

[54] Maurício Sousa, Daniel Mendes, Soraia Paulo, Nuno Matela, Joaquim Jorge, and Daniel Simões Lopes. 2017. VRRRRoom: Virtual Reality for Radiologists in the Reading Room. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, Denver, Colorado, USA, 4057–4062. https://doi.org/10.1145/3025453.3025566

[55] T. Sun, Y. Ye, I. Fujishiro, and K. Ma. 2019. Collaborative Visual Analysis with Multi-level Information Sharing Using a Wall-Size Display and See-Through HMDs. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, Bangkok, Thailand, 11–20. https://doi.org/10.1109/PacificVis.2019.00010

[56] James J. Thomas and IEEE Computer Society (Eds.). 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Soc, Los Alamitos, Calif.

[57] J. Thompson, Z. Liu, W. Li, and J. Stasko. 2020. Understanding the Design Space and Authoring Paradigms for Animated Data Graphics. *Computer Graphics Forum* 39, 3 (June 2020), 207–218. https://doi.org/10.1111/cgf.13974

[58] John R Thompson, Zhicheng Liu, and John Stasko. 2021. Data Animator: Authoring Expressive Animated Data Graphics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–18. https://doi.org/10.1145/3411764.3445747

[59] J. A. Wagner Filho, C.M.D.S. Freitas, and L. Nedel. 2018. VirtualDesk: A Comfortable and Efficient Immersive Information Visualization Approach. *Computer Graphics Forum* 37, 3 (June 2018), 415–426. https://doi.org/10.1111/cgf.13430

[60] Xiyao Wang, Lonni Besançon, David Rousseau, Mickael Sereno, Mehdi Ammi, and Tobias Isenberg. 2020. Towards an Understanding of Augmented Reality Extensions for Existing 3D Data Analysis Tools. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, Honolulu, HI, USA, 1–13. https://doi.org/10.1145/3313831.3376657

[61] Colin Ware. 2020. *Information Visualization: Perception for Design* (fourth ed.). Elsevier, Inc, Philadelphia.

[62] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. 2017. Embedded Data Representations. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 461–470. https://doi.org/10.1109/TVCG.2016.2598608

[63] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (April 2018), 1653–1660. https://doi.org/10.1109/TVCG.2018.2794222

[64] Y. Yang, T. Dwyer, K. Marriott, B. Jenny, and S. Goodwin. 2021. Tilt Map: Interactive Transitions Between Choropleth Map, Prism Map and Bar Chart in Immersive Environments. *IEEE Transactions on Visualization & Computer Graphics* 27, 12 (dec 2021), 4507–4519. https://doi.org/10.1109/TVCG.2020.3004137