Volume xx (20yy), Number z, pp. 1–12

Design Considerations for Visualization Transitions of 3D Spatial Data in Hybrid AR-Desktop Environments

Yucheng Lu (陆禹承)[1,†] Tobias Rau[1,2,†] Benjamin Lee[2] Andreas Köhn[2] Michael Sedlmair[2] Christian Sandor[1] Tobias Isenberg[1]

[1] Université Paris-Saclay, CNRS, Inria, LISN, France [2] University of Stuttgart, Germany [†] Y. Lu and T. Rau share first authorship.

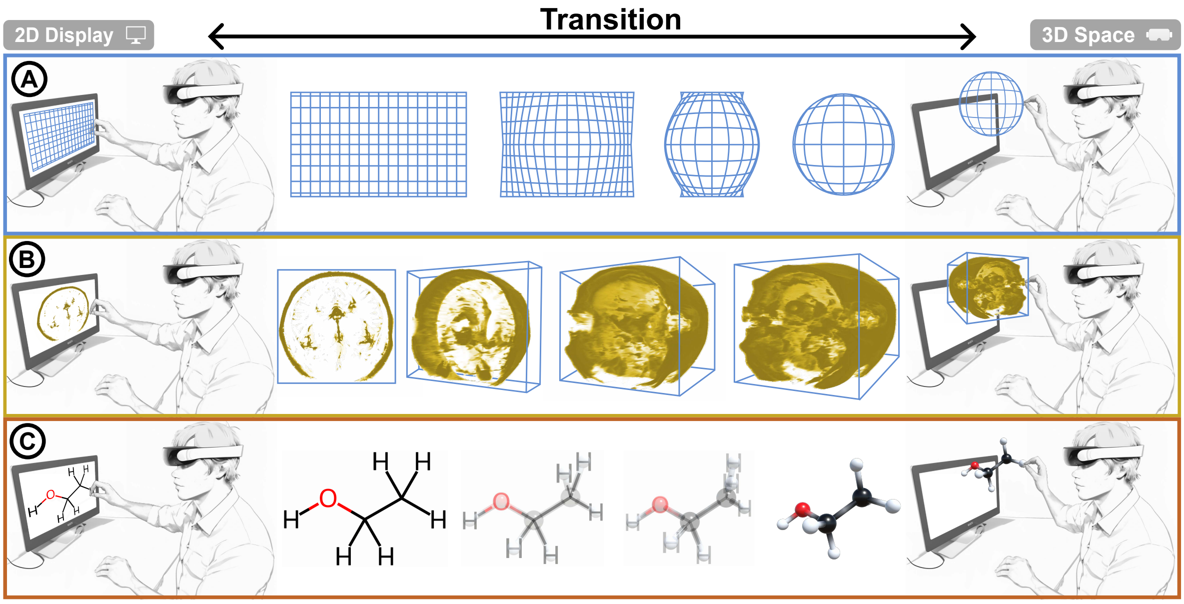

Figure 1: *Schematic views of XR transitions between 2D and 3D representations, applied to (A) a planar projection and its intermediate states, (B) an MRI slice and its full volumetric representation, and (C) a structure formula of a molecule and its 3D ball-and-stick representation.*

Abstract

We present design considerations for animated transitions of the appearance of 3D spatial datasets in a hybrid Augmented Reality-desktop context. Such hybrid interfaces combine both traditional and immersive displays to facilitate the exploration of 2D and 3D data representations in the environment in which they are best displayed. One key aspect is to introduce suitable transitional animations that change between the different dimensionalities to illustrate the connection between the different representations and to reduce the potential cognitive load on the user. The specific transitions to be used depend on data type, needs of the application domain, and other factors. We summarize these as design considerations to simplify the decision-making process and provide inspiration for future designs. We apply our concept to three case studies: sparse 3D point data, MRI scan data, and molecular data. Finally, we give some practical guidance for the prescriptive use of our design considerations.

CCS Concepts

• ***Human-centered computing*** → *Visualization theory, concepts and paradigms;*

1. Introduction

A myriad of domains and applications depend on 3D spatial data for their investigations. When 3D spatial data is presented in stereoscopic 3D views, no loss of structural information is induced, which is not the case for a projection onto a 2D screen. Instead of a pinhole projection of 3D data onto a 2D screen, designated 2D representations are commonly used, e. g., map projections [Jen12] or the surface of biomolecules [KFS*17]. Such representations, however, always bring drawbacks with them, primarily distortion and filtering information for the sake of readability.

Hybrid (mixed reality) environments [FS91, BBK*06] allow users to interact with multiple systems that reside at different points on the reality-virtuality continuum (RVC) [MTUK95] in a single environment. Visualizations in different manifestations of the RVC that the AR-plus-desktop setup offers us (also called actualities [AGF*23]), when simultaneously shown in a juxtaposed fashion, can provide a better data overview and improve the data-understanding process [WBR*20]—compared to a desktop-only setup. Generally speaking, the *mental effort* to cognitively connect a 2D with a 3D representation of the same data can be high and requires designated time to get used to it [RKP*24]. One possible way to aid with the mental connection between 2D and 3D representations in a single visualization is the use of animations [TMB02], which directly reveal this connection and can thus reduce cognitive load.

While there are existing showcases of systems that transition visualizations across different environments while changing the representation as well, a clear definition of important design considerations, especially concerning the animation, is missing in the literature. Therefore, in this work, we take a close look at all the important dimensions for a simultaneous transition between environments and 2D and 3D representations via animation and formulate them into *design considerations*. Generally, we aim to inspire the creation process of hybrid systems with animated environment adaptive visualization transitions. We extend on previous work [LCP*22, HHS*24] with considerations for 3D spatial data and animation design. Such transitions for spatial 3D data are special because—unlike for abstract datasets that are typically depicted in the form of 2D representations and for which the specific mappings even to 2D space are flexible (i. e., a free choice of visual variables)—for 3D data we are typically bound to map the data positions proportionally to the three dimensions of space.

We first explored possible datasets and domains that provide data with high complexity in order to benefit from the proposed approach. The datasets have to provide at least one 2D and one 3D representation that is well-known in their respective domains. We then developed a prototype that implements the transition process for these representations—focusing on 2D and 3D visualizations of spatial datasets in a hybrid AR-desktop environment. This prototype helped us to refine our design dimensions iteratively and to verify and showcase the design considerations with three example datasets (Exoplanet, Brain, Molecule). Then, we provide a brief overview of 3D spatial visualization design, since our considerations strongly depend on the design of the initial and target visualizations. Next, we describe possible design considerations when animating the interpolation between two visualization setups. Further, we discuss other potential design directions and considerations that focus on adopting our framework to a wide range of 3D spatial data (Section 6).

To illustrate the utility of our design considerations, we use our prototype to demonstrate this system as well as our guidelines with three case studies (Section 4): (i) astrophysics data with star systems that have exoplanets; (ii) magnetic resonance imaging (MRI) scan of a human brain; and (iii) molecular structures of different complexity.

In summary, we contribute an extension of the current literature of visualization transition design spaces to considerations specifically targeted at 3D spatial data and a discussion of practical implications and lessons learned from our three case studies.

2. Related work

Our work describes transitions of visualizations that generally—but not necessarily—change their dimensionality, corresponding to the environment.

2.1. Hybrid environments

The combination of an immersive (e. g., AR, VR) and a non-immersive (e. g., desktop workstation) environment can display both 3D and 2D visualizations in the space for which they were originally designed: e. g., 2D views on a desktop alongside visualizations of 3D spatial data in AR. Such hybrid systems support established desktop workflows that involve visual data analysis and even data manipulation on 3D spatial data via 3D input and stereoscopic rendering [CDH*19, FAP*22, RSK24, LCKP25], while extending the available workspace (e. g., Figure 1) and without losing features of specialized desktop applications and the familiar input.

A design space of single users interacting with a hybrid system is proposed by Wang and Maurer [WM22]. In one of their scenarios, a user moves a visualization from one point in the RVC to another, which involves transforming the visualization from 2D to 3D. A similar scenario is implemented and studied by Schwajda et al. [SFP*23] using graph data that transitions from a large-scale display into an AR environment. Another implementation by McDade et al. [MJC25] renders a 3D model as an exploded view on the PC and as a merged model in AR. While these approaches [GFM*22] focus on aspects that support users in forming a joint mental model of both visualizations, Lee et al. [LCP*22] formulate a general design space for animated data visualization transitions from 2D to 3D and vice versa within a single environment. Their showcases include an MRI data set, but the work has a clear focus on the visualization of abstract data. We aim to extend on this design space with considerations for (1) a wide range of 3D spatial data and (2) animation options that enhance the mental model building. Our extension, however, does not include considerations about interactions, instead we focus on visualization and animation.

We also note that, in an AR-desktop hybrid environment, augmenting 2D content in desktop monitors with 3D AR content is a common approach. Gall et al. [GHFH23], e. g., enhance the uncertainty visualization of the distribution by providing extra 3D AR visualization and facilitating gestural interaction. They and others apply this approach to tomography visualization [GFM*22, GHW*25, MNT*24]. Other studies focus on the interaction techniques for transferring virtual objects from desktop monitors into the AR space. Cools et al. [CGS*22] propose a framework that expands the desktop monitor to the AR space through an arced virtual screen. Rau et al. [RIK*25] explore various forms of gestural interaction that enable users to bring desktop monitor content into AR space. All of these works also demonstrate that an AR-desktop hybrid generally provides extra engagement and the two environments complement each other, in particular w.r.t. the depiction of different types of data.

2.2. Projections and mappings

Visualizations of 3D spatial data or descriptions based on the 3D physical world [CMS99] facilitate a recall of the physical spatial understanding. Hurter et al. [HRD*19], e. g., compare 3D visuals projected on a 2D screen with VR, and suggest that the immersive visualization “fosters the discovery of many additional insights.”

Geometric and abstract projections are commonly used to simplify the depiction of 3D spatial data, such as Earth map projection methods in geography [Sny87, Jac05, Jen12]. While distortion is inevitable due to non-isometric transformations, projections retain specific visual characteristics. The Azimuthal Equidistant projection [And74], e. g., keeps the distance undistorted when projecting the globe onto a plane. In medicine, conformal mapping is widely used to locally preserve small visual shapes by keeping internal angles invariant [KMM*18]. Physical 2D screens are considered precise and fast for 2D content, but limiting for 3D content [BSB*18]. Although it is debated whether 2D outperforms 3D or vice versa [TKAM06, HMK*19, LNP*23], hybrid interfaces are a valuable option that benefits from both, 2D and 3D [DRST14, HHS*24]. Users tend to make use of both 2D and 3D visualizations to complete their tasks depending on the circumstances [NVV*06], thus combining their advantages [MVB*17]. Building on these foundations, we develop our hybrid AR–desktop environment and the design considerations.

2.3. Animation and transition

Animated transitions have been widely adopted in various applications, not only in desktop applications but also in immersive environments. Thompson et al. [TLS21] present a variety of design considerations for animated transitions in data visualization. They highlight the importance of "coordinating objects in transition" and present an authoring system for the creation of such animations. Yang et al. [YDM*21] introduce *Tilt Map*, which transitions a 2D map into a 2D bar chart, but with a 3D intermediate region in which the bar chart is applied to the features of the map. Tominski et al. [TAA*21] discuss animated transitions within a visual analytics framework between the multivariate views. Thompson et al. [TLLS20] study people's preferences for creating animations. They show that among keyframe animation, procedural animation, and presets & templates, keyframe animation is preferred.

Other applications implement visualization transitions between two different environments to facilitate spatial understanding during the transition process [PFKJA24, LCKP25] or to help users track changes between visualizations [EDF08]. Liao et al. [LCKP25], e. g., propose an approach to smoothly morph 3D models from a 2D to a 3D representation between AR space and the monitor. They demonstrate that animated transitions enhance user engagement and facilitate perceptual tracking of objects across different representations. Also, Heer and Robertson [HR07] provide considerations about congruence and apprehension in the design of animated visualization transitions. Congruence refers to the presence of cognitive correspondences in the changes of the visualization, whereas apprehension for animation states that "animations must be slow and clear enough for observers to perceive movements, changes, and their timing, and to understand the changes in relations between the parts and the sequence of events." [TMB02] In our work, transitions occur between AR and the desktop environment, accompanied by changes in the visualization itself. We focus on transitions that adapt to the AR space or physical monitor displays as well as on animation designs to show changes in visual encoding.

3. Design considerations

We begin by discussing the challenges and important dimensions that have to be considered when designing a transition of spatial data that adapts to the current working environment.

3.1. Development process

To develop our proposed design considerations, we centered the transition design around the concept of spatial information. We surveyed relevant work through the IEEE Xplore database with the keywords "Cross-Virtuality/VR/AR/Immersive", "Transition", and "Visualization" to identify relevant prototype and literature review papers. This process covered application domains such as medicine, geography, and chemistry; three areas that feature 3D data with distinct 2D representations (e. g., projections, slices, structural diagrams). Prior surveys of 2D/3D visualization design [HHS*24], transformation design spaces for MR [LCP*22], and geometric mapping techniques [KMM*18, KFS*17, MC01, Sny87] further informed our framing. These domains collectively span volumetric, point-based, and topological data, offering a representative basis for spatial transition scenarios. We also examined existing hybrid systems that implement transitions between representations [HRD*19, MDLI*18, HMK*20, MAB*18, YJD*18, SFP*23]. We deliberately exclude device-specific interaction dimensions since they are covered by the design space of Lee et al. [LCP*22].

3.2. Challenges and opportunities

In addition to the challenges designers of animated visualization transitions have to face, we are switching the environment in which the visualization is embedded, which adds further geometric and contextual constraints as well as presents us with some opportunities for improved visual exploration as follows:

Tracking helps users maintain a stable mental correspondence between the source and target visualizations. In this case, users may tolerate a higher degree of geometric distortion, as long as the motion remains easily perceivable. Maintaining good visibility, however, is crucial to ensure that elements can be reliably followed throughout the transition.

Explorable intermediate visualization aims to improve the sense-making process. The animation is expected to preserve high congruence and maintain visual semantics at every increment of the transformation. The distortion tolerance is extremely low because users need to make sense of the intermediate visualizations and inspect meaningful states along the transition.

Explanatory transformations reveal the logic behind a complex mapping, enhancing users' understanding of how the transformation operates. Specifically, finding a balance between distortion and clarity to reveal the structure of the mapping is crucial.

Context highlights differences between desktop and AR. The monitor's bounded 2D frame constrains placement, while AR offers a larger working space. A transition across environments can disrupt the congruence of the context. Transitions, therefore, should be designed to smooth these contextual shifts.

3.3. Visualization pipeline

To properly introduce our design considerations for transitioning visualizations, we examine the components required to build a visualization for spatial data. This *visualization pipeline* (Figure 2) is inspired by Bruckner et al.'s [BIRW19] *Visual Mapping*,

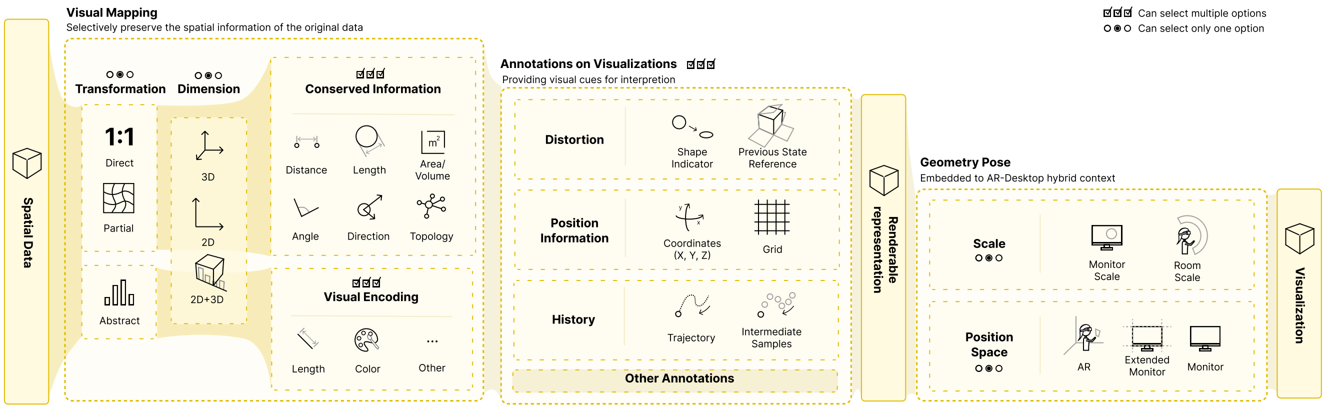

Figure 2: *Our design considerations for the visualization pipeline consist of three stages: The visual mapping, the additional annotations, and the geometry pose of the visualization embedded within the AR-desktop space. From left to right, the designer can make the design choices at each stage and create the target visualization. After several visualizations emerge through this pipeline with slightly different design choices, we can determine the design for transitioning between them, as illustrated in Figure 3.*

other design spaces [HHS*24, LCP*22], and other visualization pipelines [CMS99, SM05]. We consider the input data a conditioning factor, since possible choices arise from making decisions about how to filter and represent the data. Generally speaking, each transition consists of three visualizations: (1) an *initial visualization*, which describes the visualization before the transition, (2) one or more *intermediate visualization*, which are the state of the visualization during the transition and generally non-static, and (3) a *target visualization*—the visualization after a complete transition. The intermediate visualization changes depending on the progress of the animation and typically includes both the initial and target visualizations, either unaltered, transformed, or partially altered. In addition, the intermediate visualization can introduce elements that are not part of the initial or target visualizations, for example, by showing the trajectory of elements within a visualization.

As we show in Figure 2, our input data goes through the *Visual Mapping* block, which defines a *Transformation* and inherently a *Dimension* from that, as well as *Conserved Information*. A *Visual Encoding* such as color is commonly used, but generally optional. Also optional is the addition of *Annotations* on the visualization, like coordinate axis indicators, grids, or trajectories. As indicated by the *Geometry Pose* block, the visualizations can then be placed and scaled inside the AR-desktop hybrid environment.

We consider the blocks *Annotations of Visualizations* [TLLS20] and *Geometry Pose* as an extension to the design space of Lee et al. [LCP*22], since the dimensions of the first block help users to follow the animation, and the latter is specific for an AR-desktop setup. The *Position Space* dimension is the hybrid pendant to Lee et al.'s "Surface Relation" design dimension.

3.4. Transitions

For the transition itself, we mainly focus on animation-specific design considerations, which are mainly inspired by Thompson et al. [TLS21, TLLS20] and Heer and Robertson [HR07]. Some

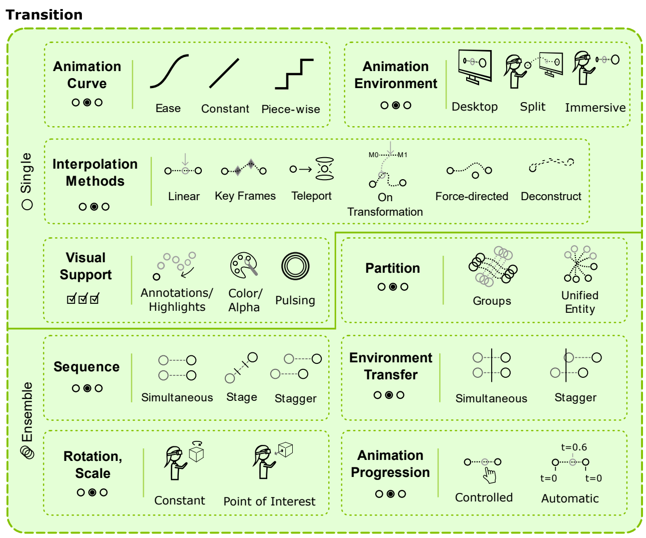

Figure 3: *This figure shows the possibilities for the transition phase of the visualization, which consist of dimensions for single visual elements (e. g., glyph or voxel) and ensembles of visual elements.*

dimensions, however, are specific to a multi-environment (hybrid) setup or to the immersive space (AR).

Figure 3 illustrates our design considerations, which we divided into two parts: considerations applicable to *single* visual elements (e. g., glyph, voxel) and those that apply to parts or the entire *ensemble* of visual elements.

First, the *Animation Curves* [DBJ*11, ARES13, LCP*22] dimension determines the animation speed, which usually includes three different behavior curves: *Constant*, *Ease-In/Out*, and *Piece-wise*.

In the *Animation Environment* dimension, our general assumption for each transition is that the visualization switches from the *Desktop* environment to the *Immersive* environment or vice versa, which includes some kind of animated movement or interaction like pushing [RIK*25] a visualization into the screen. The animation of the transformation is coupled to the environment switch, either by distance or time. The animation of the transformation can thus also be *Split* between the environments. An MRI volume, for example,—when transitioning from AR to the 2D desktop—first aligns with the screen such that the slices are parallel to the screen's plane; then each slice transitions individually and gets rearranged in the target environment as a set of independent slice images.

The next dimension describes *Interpolation Methods* [CSCM21]. The most common methods to interpolate between the initial and target state are either *Linear*, use of *Key Frames*, or to *Teleport* visual elements directly to the target position/state. In most cases, however, physical or mathematical constraints exist, especially when interpolating positions. When unfolding a cube from its 3D shape to its flat cross shape, for instance, a linear interpolation of the vertices would shorten the length of the edges. Animations that ignore these constraints could become uncanny. Therefore, we propose the *On Transformation* option in our design considerations. In the case of, e. g., node-link diagrams, the transformation to 3D or 2D can introduce unwanted crossings of branches, for which a *Force-directed* interpolation could provide a clearer transition. Sometimes, however, the crossing of visual elements cannot be avoided. For such situations we added the *Deconstruct* method. Elements can then be deconstructed in one position and reconstructed at another while using, e. g., a particle effect to link the two states.

We can use the *Visual Support* [HR07] dimension to provide additional visual elements to guide the user's attention during the transition or to, e. g., blend elements (*Color/Alpha*) that have unrelated visuals in the initial and target visualization. *Annotations* (Figure 2) can highlight certain elements or regions during a transition to support the sense-making process. The trajectory of each visual element, e. g., can be rendered as a line, but only during a transition. Alternatively, visual elements can be emphasized using a *Pulsing* or similar single element animations.

The first dimension in the *Ensemble* region is inspired by the discussion of apprehension by Heer and Robertson [HR07]. The *Partition*, which defines if the whole visualization as *Unified Entity* should be affected by further dimensions in this region or just specified *Groups*. These *Groups* can also be used to realize features such as trajectory bundling [DCZL15, LGS*25].

The next two dimensions, *Sequence* [CDF14, CSCM21] and *Environment Transfer*, both describe a temporal component. While *Sequence* describes in which order transformations are applied to the visual elements, *Environment Transfer* describes the order in which the visual elements are transitioned to the other environment. We consider both dimensions as being independent of each other. The *Simultaneous* sequence transforms groups or the entire visualization simultaneously. Respectively, *Simultaneous* transfers the groups or the whole visualization into the other environment. The *Stagger* sequence starts with one or a group of visual elements and subsequently animates others. This approach can be responsive to user interaction, providing means to explore intermediate visualizations interactively based on the context. The *Stage* sequence can be combined with the other two, as it manages transition changes consisting of several independent steps during the animation, facilitating the separation of changes that otherwise occur simultaneously. Moving visual elements and changing their color, e. g., can be handled in series—in different stages—or in parallel. In an *Environment Transfer* with the method *Stagger*, visual elements or groups are transferred in a sequence, e. g., individual slices of an MRI volume.

The *Rotation, Scale* [HR07] dimension considers additional rotating and scaling—independent of the target visualization, of groups, or the whole entity. For instance, a visualization can spin multiple times before reaching its target state during the transition (*Constant*), allowing the user to get a better view of the visualization as a whole, potentially reducing occlusion. Similarly, the visualization can be scaled and rotated, guiding the user's attention to a *Point of Interest*. The *Point of Interest* method can also keep the perspective on the visualization constant, e. g., relative to the user's viewing direction, ensuring optimal visibility of the transition, even when the user and the visualization are moving within the immersive environment.

Finally, the *Animation Progression* [LCP*22] dimension defines if the group or entity is animated via user input (*Controlled*)—e. g., dragging the visualization—or runs *Automatic*ally—the latter requiring some kind of trigger. A more complex transition could, for example, animate the most important group with the *Controlled* method, while the other visual elements undergo a *Automatic* animation, which could be triggered when the visualization touches the screen's surface. Again, this has the potential to draw the user's attention to the controllable part of the animation.

With these considerations for animated transitions of 3D spatial data we thus provide a plethora of options for effectively designing the change from one environment to another in hybrid environments, all of which have not been discussed in this context before—with the sole exception of the *Animation Curve* and *Animation Progression* dimensions previously mentioned by Lee et al. [LCP*22].

4. Case studies

To illustrate the utility of our design considerations, we now discuss three case studies: an exoplanet dataset, medical MRI data, and chemical molecule data. We believe that these case studies from different application domains of scientific data analysis showcase a representative amount of design choices possible in our design considerations. For this purpose, we implemented the characteristics of our design considerations using Unity and the Mixed-Reality Toolkit (MRTK). We deployed the application on the Oculus Quest 3 and HoloLens 2 headsets for AR and a Windows desktop to build an AR-desktop hybrid environment. We synchronize the virtual object transformation, state, and user interaction behavior between the two devices through a local Wifi network connection. In this prototypical implementation, we have realized various state conditions and transitions that we discussed, which now allow us to showcase a seamless screen extension that enables the user to drag a visualization off the screen using either mouse or gesture input. We refer the reader to the accompanying video (see our additional material at doi `10.17605/osf.io/ve52m`) for a visual explanation.

4.1. Implementation

To implement the prototypes, we relied on Unity and the Mixed Reality Toolkit (MRTK) to develop two distinct applications, targeting both the desktop platform (Windows) and AR headsets. We

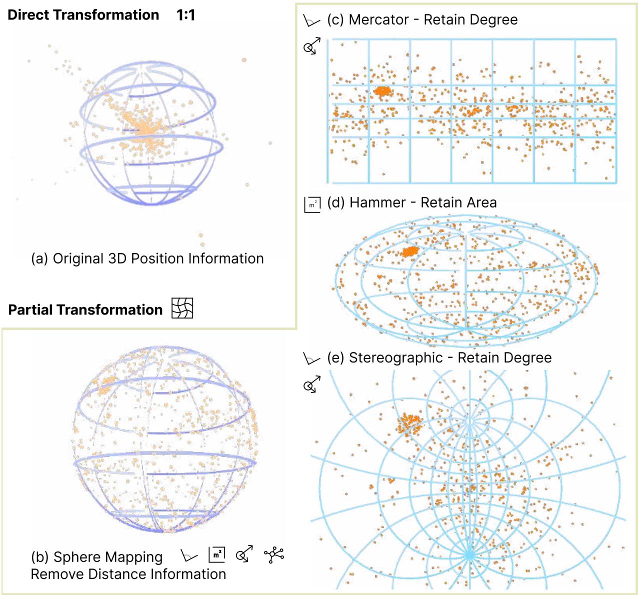

Figure 4: *Visual mapping examples for the exoplanet dataset. 3D: (a) direct mapping of the dataset with ICRS spherical coordinates as annotation; (b) location of the exoplanets orthogonally projected onto the ICRS surface. 2D: (c) Mercator, (d) Hammer, and (e) stereographic projection. All projections use a grid as an annotation.*

deployed the AR application on either the Oculus Quest 3 or the HoloLens 2, and ran the desktop component on a Windows desktop, together forming our AR-desktop environment. To set up this environment, we first calibrate the physical monitor by pointing to three of its corners in AR space with the index finger. We also establish a local Wi-Fi network to synchronize user input and virtual object data between the AR and desktop platforms in real time.

Our prototype features a basic interaction mechanism that facilitates seamless transitions across devices. Users can drag visualizations from desktop monitors into the AR space and vice versa. This drag operation can be performed using hand gestures or a mouse. A typical interaction involves dragging a visualization off the desktop screen, triggering an automatic transition from the 2D to the 3D visual representation. Based on this implementation, we now describe the three case studies to illustrate our design considerations and showcase its descriptive power.

4.2. Exoplanets

As a representative of sparse 3D point data, we selected NASA's dataset of stellar systems with known exoplanets [Cal25], which records information about the respective stellar system and its planets, e. g., spherical coordinates (IRCS [MAE*98]) of systems. Here, we visualize the locations of the respective stars.

Our general goal with this use case is to allow users to make a connection from the view of an observer who stands in the center of the stellar system and looks up to the firmament to various 2D map projections of the star positions on the sky. In addition, our goal is to support viewers in gaining insight into how these projections distort the view compared to the observer's view as well as to connect these views back to the original distance weighted data.

To support viewers in understanding the various projections and the distortion, we mostly use the *On Transformation Interpolation Method* for this dataset. To map the data to the actual visualization, we first define its appearance in 3D (AR) and 2D (screen) spaces (see Figure 4 for all representations we used for this case study).

For the animations, we found that an *Ease Animation Curve* with a *Simultaneous Sequence* on the *United Entity* is sufficient to follow the animation. Since the animation directly starts with the wrap of the 2D projection into a 3D object, we use the *Animation Environment Immersive*, so the animation starts after the 2D object has been transferred to the immersive space. We employ the *Controlled Animation Progression* because it allows the users to play the animation at will (forward, backward, slow, fast, or pause).

We begin with a 2D projection (Hammer) of the data with a *Grid* annotation on the 2D screen (Figure 5(A)), which gets dragged into the immersive space and transforms into a sphere with the stars projected on it (Figure 5(A; c)). Alternatively, we map the 2D projection to actual 3D locations, as shown in Figure 5(A; d). To embody the observer's view, the visualization can be scaled up to room-scale, providing an *inside-out view* of the data (Figure 5(B)), therefore, mimicking a scenario in which the stars are at great distances. *Staggering* the animation, as we show in Figure 5(C) for the Mercator projection, first forms a cylinder and then transforms the cylinder into a sphere. Hence, it reveals the steps behind this projection method. In the case of the Stereographic projection (Figure 5(D)), we can observe how the loose ends of the grid wrap around the sphere and tie back together when the animation finishes.

For all these animated map projections, we can add further *Annotations* (Figure 5(D; b))—particularly *Shape Indicators*—during the transition to highlight regions of excessive distortion, which can pose challenges for the explainability and the sense-making of intermediate visualizations. It is thus particularly useful to identify these early-on during the design phase of the animation.

As we show in Figure 5(D; a), we also employ the *Annotations* to render the *Trajectories* of the individual stars. We combine them with a *Staggered Animation Sequence*, such that the trajectories to the target visualization's position as well as the trajectories to the projected positions on the sphere are shown, revealing details of this particular projection.

Overall, the combination of the outlined transition strategies allows us to seamlessly animate from a 2D (map) representation to a 3D spherical mapping and, further, to a full 3D point cloud, while maintaining the unique characteristics of the 3D datasets and, at all points of this transition, showing to the viewer how the different mappings relate to each other.

4.3. Brain MRI data

The second data type in our case study is 3D volumetric data, which is sampled on regular lattices. We employ the MRI dataset from the IEEE VIS 2010 contest to demonstrate volumetric data transitions that reveal internal structure, as schematically shown in Figure 1(B).

The general goal using this dataset is to connect the volume back

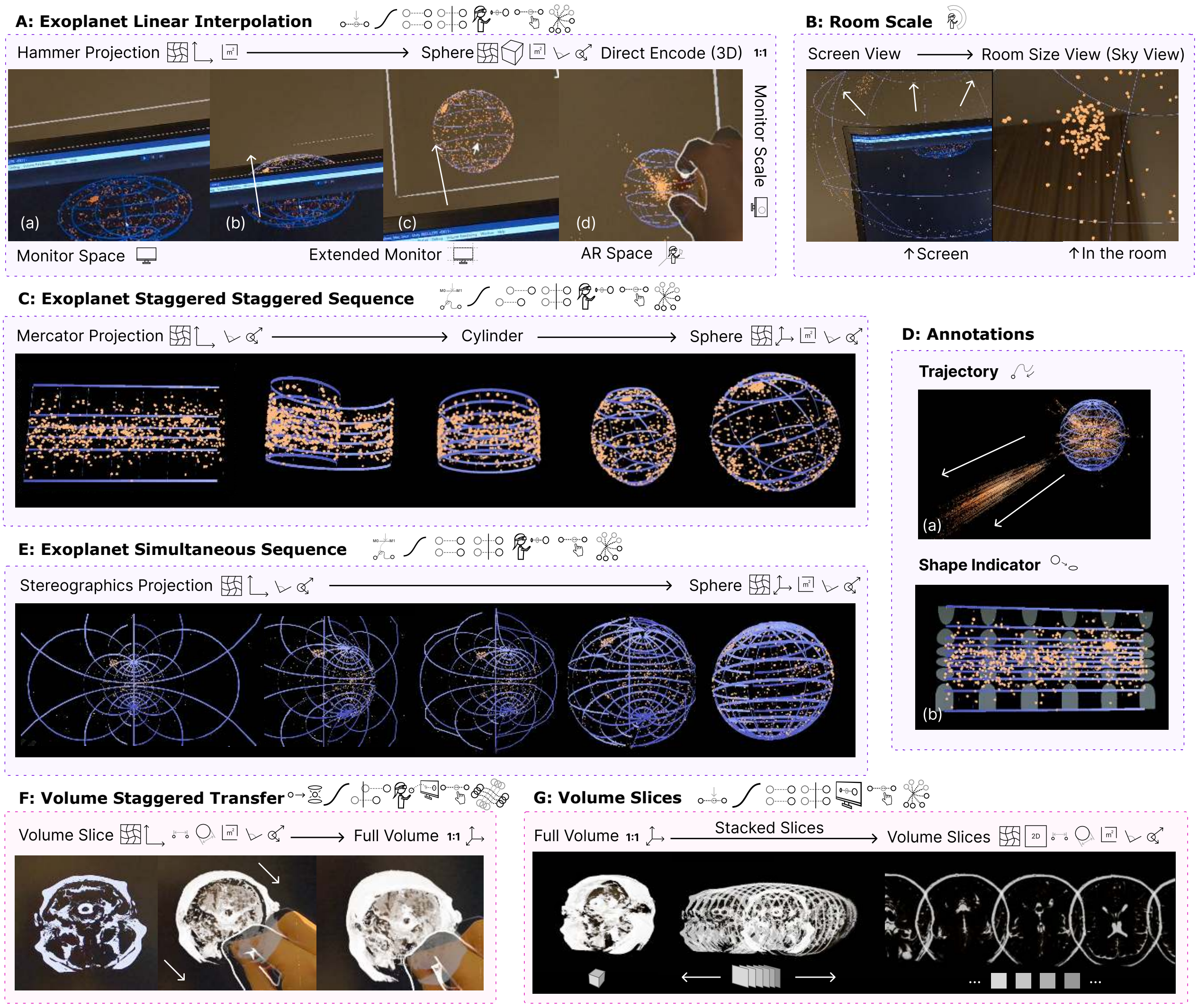

Figure 5: *Illustration of a series of transition examples from the exoplanet dataset (A–E) and the brain MRI dataset (F, G). In the figure we also show an example of a room sized representation (B) and of annotations to highlight distortion (D).*

to the raw input data—the slices. Since volume visualizations usually suffer from heavy occlusion, slicing into the dataset or spreading slices in a way that reconnects their individual positions within the volume helps reveal the inner structure of the dataset.

We start with a common visualization mapping to explore the volume data's inner structure, which shows individual slices. Figure 5(F) shows the process of transitioning from a 2D slice (*Partial* transformation) to a *Direct* (volume) representation, in which the slices are sequentially added as the user is "pulling the data out of the screen." In this animation we employ the *Split Animation Environment* in combination with a *Staggered Environment Transfer*, growing the volume in the immersive environment while we simultaneously flip through the slices on the desktop. For this purpose the scale in the immersive environment has to roughly match the *Monitor Scale*. During this transition, we can still observe the intermediate visualization, where the visualization presents a subset of the original visualization—maintaining both its integrity and semantic meaning without distorting the spatial information. This example demonstrates that the data's spatial information is not manipulated, only selectively filtered. Instead of flipping through the slices directly at the position where the volume touches the screen, we could also spread the slices on the desktop as the volume is pushed in, increasing the visibility of the 2D representation.

We can also transition the volume representation to a set of slice-based 2D representations (Figure 5(G)), where each slice is one that was originally recorded in the MRI data. Here, the *Animation Environment* is *Desktop* since we first transition the volume to the desktop (*Simultaneous Environment Transfer* on the *Unified Entity*) an then spread the slices in a *Simultaneous Sequence*. We break the *Topology* of the dataset and remove part of its global structure by animating the slices so that they end up being shown side-by-side as traditionally on a surgeon's light box. The slices are fanned out along the x-axis, utilizing the *On Transformation* interpolation method and *Simultaneous* animation sequence, while keeping the currently selected slice (on the desktop) in the center. This side-by-side representation does not introduce distortion within the slices, but does, for example, not conserve *Distance* between the slices.

4.4. Molecules

Finally, we present a use case of structured 3D data representing molecular structures. Graphical representations of molecules have a longstanding tradition in chemistry and biology, as clearly not only the composition but also the relative spatial arrangement of the atoms determines the properties of a chemical compound. Pertinent to our discussion of the design considerations, the structure of molecules is not only expressed in 3D but also frequently and uniquely in 2D. In contrast to the first case study, here the transformations are not described by a formula but rather by an algorithm that implements a defined set of rules [Bre08].

Traditional representations are known as structural or skeletal formulas (e. g., Figure 6(a)). They indicate which atoms are chemically bound to which others and may contain additional qualifiers. Connecting lines often express the bond type (single, double) and may be related to details of the electron structure (in Lewis formulas, each line represents an electron pair). Although generally used as a 2D representation, they can qualitatively express spatial information via, e. g., wedges instead of lines. Traditional structural formulas thus primarily express molecular topology, but also provide—for the experienced chemist—a wealth of information about possible reactivity and properties. Naturally, a classical 2D projection of the 3D assembly onto a 2D screen cannot represent all the subtle details of a molecular conformation, and the spatial extent of certain parts of the molecule may not be obvious.

The molecule in our example (Figure 6) contains two double bonds in one of the side chains and another double bond in the other side chain. There is thus the possibility of an intramolecular reaction. A reaction can only take place, however, if the spatial arrangement allows it, which is not visible in the 2D representation. Our general goal with a transitional visualization is thus to connect 3D molecular structures with their textbook representation, since textbook representations fail to indicate how close fragments on one side of the molecule are to fragments on the other side of the molecule. Depending on the complexity of the molecule, making this connection is even difficult for experienced chemists.

We start by applying spatial coordination to transition from structure formula to a 3D ball-and-stick representation (Figure 6(f)), i. e., we move the letter and line elements to their correct 3D positions as we transition. Yet, such a *Simultaneous Sequenced Linear* interpolation (Figure 6(b, c)), when done on a desktop-only setup, still requires interactive camera movement to fully grasp the alignment of the two side chains. In an immersive environment, in contrast, chemists can fully concentrate on tracking the location of the double bonds. The visibility and tracking can further be enhanced by a *staged* animation *Sequence* (Figure 6(d, e)). In the first stage, we move the atoms of the structure formula to their respective 3D positions and, afterward, blend them into the 3D balls-and-sticks model using *Ease Animation Curves*. In both cases, we use a representation at *Monitor Scale*, but the *Room Scale* can be beneficial to investigate, e. g., datasets of crystals. Here, we chose the *Animation Environment* as *Immersive* because problems in chemistry often require a certain view angle on the molecules for occlusion-free vision on the region of interest, which requires interaction to rotate the molecule. We use a *Controlled Animation Progression*, which consists of pulling the molecule away from the screen, since it is easier when done close to the user than next to the monitor.

A *Staggered* animation *Sequence* is a possible alternative to enhance the spatial sense-making during the transition. For example, each side chain can be put into a *Groups Partition*, which, together with the *Staggered Sequence*, allows the chemist to concentrate only on one part of the molecule at a time. The trade-off here is that this would create several intermediate visualizations that are chemically "incorrect." Adding *Visual Support*, however, could mitigate this effect because it reduces the opacity (*Color/Alpha*) of the side chains that are currently not animated or in the target visualization. Further, *Annotations/Highlights* could guide the chemist's attention during the transition animation onto the potential reactive fragments of the molecule. For molecules with many side chains, the *Force-directed Interpolation Method* potentially becomes important.

While the appearance of molecules indicates similarities to network graphs [CSCM21], making the connection to an inherent 3D structure is a different problem not discussed in the literature yet. Overall, our applied design considerations, especially the use of *Staged* animation *Sequences*, facilitate tracking reaction sites, which are easier to identify in the 2D representation.

5. Practical guidance

Since the selection of dimensions from our design considerations heavily depends on the content of the initial and target visualizations, we discuss the most promising combinations and *rules of thumb* we came across during our development process.

Complexity dictates if and how many steps are required for a given transition. It affects especially choices in the dimensions *Sequence*, *Environment Transfer*, and *Partition*. If a visualization is rather simple, a *Simultaneous Sequence* and *Environment Transfer* on the entire visualization is often sufficient. For visualizations with higher complexity, each small motion can potentially be singled out or emphasized using the *Visual Support* dimension.

Giving a user **Control** over the animation is advantageous, since they can (1) repeat the transition as often as they desire, (2) adjust the animation speed dynamically, (3) only play a small part of the animation, and (4) adjust the view on the visualization in between. Only visual elements that are not crucial for the sense-making process can be animated using the *Automatic Animation Progression*.

If an animation can be separated into a 2D and 3D part, the *Split* **Animation Environment** can be employed. However, 2D animation parts, e. g., spreading MRI slices, can also be achieved in the *Immersive Animation Environment* if the space is required. All animations that include a transform between 2D and 3D should use the *Immersive Animation Environment.* For spatial data, the *Desktop* environment will be used only rarely exclusively.

Depending on the **Experience** of the audience, additional aids can

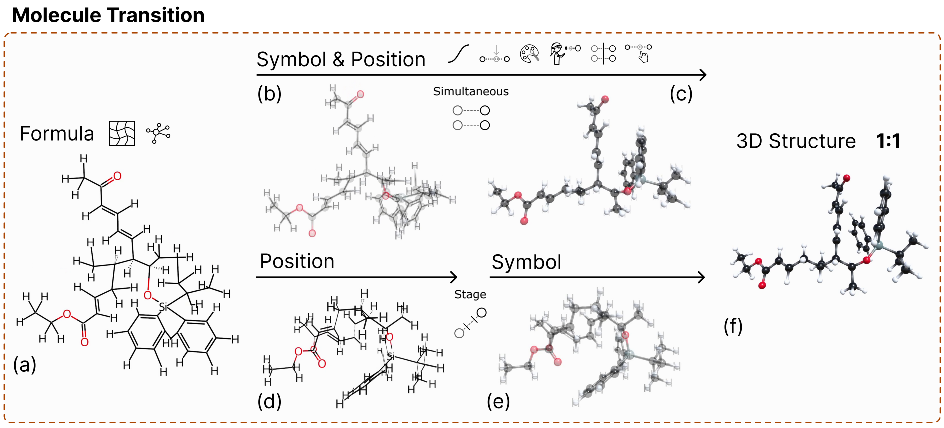

Figure 6: *Case study on molecular data. The atom positions are interpolated linearly between the 2D formula and the 3D structure. Upper animation path:* Simultaneous *interpolation; lower animation path:* Staged *interpolation between the two visualizations.*

be implemented. For example, a transitional visualization system may get employed in an exhibition. Here we have to assume that the user does not have any experience with the data and the system, so we could pick *Point of Interest* from the *Rotation, Scale* dimension or the *Visual Support* dimension to ensure the attention of the user is guided to the important regions inside the visualizations. In this situation, the *Key Frames Interpolation Method* can provide further means to curate a visualization transition.

Further general guidance about the design of animation visualization transitions can be found in the considerations about congruence and apprehension by Heer and Robertson [HR07].

6. Discussion

While our case studies demonstrate the potential application of our design considerations, we also want to raise several points that affect their application and future use as well as a potential extension.

Notion of scale. In most cases, 3D spatial data inherently provides a literal scale of the data. Whether or not it is feasible to present the data on the literal scale, especially in cases where the literal scale is between monitor and room scale, is up for further investigation. In the examples we presented in our case studies, only the MRI dataset can be presented at its literal scale, which is, depending on the required analysis process, probably not the preferred choice by domain scientists. The question of how to scale the data during the transition between environments thus strongly depends on the task at hand. Since the scale is difficult to generalize, we believe that giving users control over the size is the best solution.

Composed visualizations. Visualizations of 3D spatial data that are combined with 2D annotations or visualizations [HHS*24] pose a whole new set of challenges in the context of AR-desktop hybrid environments. In a transition, for example, the 2D parts of the visualization could stay on the desktop PC, and only (annotation) lines could be provided to link between the representations and environments [SFP*23]. But this process could also include a different set of transformation operations to reach a different final state of the 2D visualization. Therefore, we could separate the 2D and 3D parts of the visualization into two categories and regard the information separately during the animated transition. Our design considerations provides some options to separately process the visualizations, but would require further dimensions to provide comprehensive options for this kind of visualization.

Intermediate states. A transition design that emphasizes intermediate states by, e. g., using the *On Transformation* interpolation together with a *Stagger* animation sequence or a *Staged* interpolation provides valuable additional information that supports users in mentally connecting the representations. In our molecule use case, e. g., the intermediate state, in which the structure formula is arranged the same way as the balls-and-sticks visualization, provides a way to connect the traditional textbook representation. Some spatial datasets lend themselves well to intermediate states, but in general they are difficult to create automatically—even when using our design considerations. Also, providing a *Controlled Animation Progression* (e. g., a scrollbar of a video) can be a powerful tool for data exploration using animated transitions.

Animation authoring. To provide the possibility to create involved animation sequences using our considerations is rather complicated, because of the large variety of spatial data and domain-specific visualization requirements. In addition, only a few hybrid visualization systems exist that are specifically tailored to meet the requirements of a particular domain or application. Existing authoring tools [TLS21] and descriptive animation grammars such as Deimos [LSC*23] already lack the required support to create and render visualizations of spatial data. Features such as those described by our *Environment Transfer* dimension can currently not be found in the literature.

Animation of timesteps. We did not include the animation of time-dependent data, because it does not fit into our "transitions between 2D and 3D" framework. The animation of temporal data value changes is a common theme in visualization animations [HR07, TLLS20], and we believe it is an important use case for visualization transitions in hybrid environments. However, combining the temporal changes in data values with the transition we describe in our design considerations can easily become too complex. That being said, an animation for timesteps could be added before or after an environment adaptive transition, as a separate step, without unnecessarily increasing the complexity of the animation.

Interaction for transitions. Our design considerations focus on integrating spatial information and on the relationship between visualization and the display environment. While the original data contains spatial information potentially sampled from reality, presenting the data within the AR-desktop space may also semantically assign real-world knowledge to the visualization or interaction. The area surrounding the monitor, for instance, can imply different types of transitions based on the direction relative to the monitor [RD19]. These spatial semantics can be stacked or juxtaposed to create complex layouts for more sophisticated designs. In addition, the position in the AR-desktop environment can be represented by three variables. Based on the positioning in the immersive environment, it is thus feasible to assign up to three parallel input variables. Each variable can correspond to a distinct state and can support multiple state changes simultaneously. The monitor's physical appearance is also typically rectangular, which can serve as a reference frame. As we focused on visual appearance in our work, however, we did not include such considerations as dedicated design dimensions.

Multiple datasets. While our case studies only come from three domains (yet with a representative range of different types of data), it is important to consider the broader context of data analytics systems. Each domain typically has unique spatial data types and specific tasks that are not easily generalized across all cases. It is also common for visual analytics systems to display multiple or combined visualizations simultaneously [KMM*18] and has been one of the key considerations of the work of Lee et al. [LCP*22]. A user may have different visualizations of the same dataset; each presenting distinct aspects of the data. We hypothesize that this concept can be applied to lay out multiple visualizations, thus opening the possibility of transitioning between them. In addition, presenting the same information through different visualizations may help us reduce bias and enhance the user's overall understanding of the data.

Generalizability. Describing transitions using our proposed design considerations is highly domain-specific. It may be difficult to generally translate concepts that work on one domain's dataset to another domain. This fact also implies that some of our proposed design dimensions can be extended by considering further domains and datasets, especially for our annotation design dimensions. While we specifically focused on the AR-desktop environment, the transition design within our chosen environment has the potential to extend to other cross-device settings, such as those involving tablets and AR headsets. When portable or multiple devices are involved, the geometric pose design considerations may further consider the spatial relationships among devices, the physical environment, and the visualizations. In addition to cross-device scenarios, transitions across different virtuality levels—such as between VR, AR, and physical-world monitors [FAP*22, AWK*23]—also require considerations when transitioning between different environments.

Hybrid input modalities. For our prototypes, we developed a simple framework that focuses on AR-desktop environments. For a proper generalization of interaction across multiple devices [HMK*19], a more universal framework is needed to integrate different input devices while supporting smooth input transitions and seamless manipulation of virtual objects. Though existing studies [HWF*22, SSP*23, RIK*25] debate whether users like to change input modalities, we believe that the traditional mouse and keyboard input will not be replaced anytime soon for desktop PCs [WBR*20, PNB*21]. For hybrid systems, however, developers should consider input modalities that can be used on all incorporated devices without requiring switching, and which can then be used to control the transitions described in our design considerations.

User feedback. In this work, we deliberately did not conduct an empirical experiment to validate our work, as our focus is on the design considerations rather than evaluating the usability of a specific implemented instance of the interaction design or of given use cases. Our goal is to empower designers to describe, generate, and evaluate designs [BLBM21]. Given the domain-specificity of spatial dataset visualization tasks, a user study would have only offered limited general insight into the design considerations. Instead, we validated our approach through our diverse use case demonstrations in Section 4. Moreover, prior studies have already examined the general effects of AR-desktop hybrid systems and demonstrated user engagement in transferring virtual objects between screen and AR space [CMGS25, LCKP25, RIK*25].

7. Conclusion

With our discussion of animated transitions of spatial data between planar 2D displays and stereoscopic AR environments we extended past work that looked at such transitions [LCP*22]. With our design considerations we demonstrate how to make use of and cater to the unique spatial properties of data that arise in many scientific application domains, such as medicine, astronomy, chemistry, physics, etc., for which we are not at liberty to use the third dimension in AR space to facilitate the transition between the different environments. Instead, in our design considerations we embrace this constraint and demonstrate how we can make use of dedicated animation sequences, annotations, and different spatial mappings to allow viewers to mentally follow the changing representation of the data as it transitions between the spaces. As we noted in our discussion, we do not make the claim that our design considerations would be complete. Domain-specific constraints and practices are likely to give rise to additional possibilities for designing transitions that, in turn, may then be applicable to other use cases. Nonetheless, we have laid the foundation for the discussion of how to design effective AR-desktop environments for the analysis of spatial data.

Acknowledgments

We thank all anonymous reviewers as well as Ambre Assor for their feedback on our manuscript. This work is partially funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2075–390740016. We also acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

Supplemental Material Pointers

We share our additional material (author version of the paper, self-created figures, video) at doi `10.17605/osf.io/ve52m`. The source code for our three examples (Section 4) is available on GitLab at `gitlab.inria.fr/yuclu/hybridvisanidemo`.

Images/figures license/copyright

References

[AGF*23] AUDA J., GRUENEFELD U., FALTAOUS S., MAYER S., SCHNEEGASS S.: A scoping survey on cross-reality systems. *ACM Comput Surv 56*, 4 (2023), 83:1–83:38. doi:10/gtnn22.

[And74] ANDERSON P. S.: An oblique, poly-cylindrical, orthographic azimuthal equidistant cartographic projection: Its purpose, construction and theory. *Cartogr 8*, 4 (1974), 182–186. doi:10/g8rvqx.

[ARES13] ABUKHODAIR F. A., RIECKE B. E., ERHAN H. I., SHAW C. D.: Does interactive animation control improve exploratory data analysis of animated trend visualization? In *Proc. Electronic Imaging* (2013), SPIE, Bellingham, pp. 211–223. doi:10/nvkc.

[AWK*23] AIGNER D., WANG N., KIELMAYER D., STEINER J., HOCHPÖCHLER J., HEINZL C., ROTH D., MAURER F., ANTHES C.: Cardiac visualisation along the RV-continuum — A high-fidelity pilot study. In *Adjunct Proc. ISMAR* (2023), IEEE Comp. Soc., Los Alamitos, pp. 675–680. doi:10/g84w4g.

[BBK*06] BORNIK A., BEICHEL R., KRUIJFF E., REITINGER B., SCHMALSTIEG D.: A hybrid user interface for manipulation of volumetric medical data. In *Proc. 3DUI* (2006), IEEE Comp. Soc., Los Alamitos, pp. 29–36. doi:10/dpgb7z.

[BIRW19] BRUCKNER S., ISENBERG T., ROPINSKI T., WIEBEL A.: A model of spatial directness in interactive visualization. *IEEE Trans Vis Comput Graph 25*, 8 (2019), 2514–2528. doi:10/gjbdnn.

[BLBM21] BEAUDOUIN-LAFON M., BØDKER S., MACKAY W. E.: Generative theories of interaction. *ACM Trans Comput-Hum Interact 28*, 6 (2021), 45:1–45:54. doi:10/pqrb.

[Bre08] BRECHER J.: Graphical representation standards for chemical structure diagrams (IUPAC recommendations 2008). *Pure Appl Chem 80*, 2 (2008), 277–410. doi:10/ccgx95.

[BSB*18] BACH B., SICAT R., BEYER J., CORDEIL M., PFISTER H.: The hologram in my hand: How effective is interactive exploration of 3D visualizations in immersive tangible augmented reality? *IEEE Trans Vis Comput Graph 24*, 1 (2018), 457–467. doi:10/gcp924.

[Cal25] CALIFORNIA INSTITUTE OF TECHNOLOGY: NASA exoplanet archive. Web site: exoplanetarchive.ipac.caltech.edu, 2025. Last visited: Jun. 2025.

[CDF14] CHEVALIER F., DRAGICEVIC P., FRANCONERI S.: The not-so-staggering effect of staggered animated transitions on visual tracking. *IEEE Trans Vis Comput Graph 20*, 12 (2014), 2241–2250. doi:10/228.

[CDH*19] CAVALLO M., DOLAKIA M., HAVLENA M., OCHELTREE K., PODLASECK M.: Immersive insights: A hybrid analytics system forcollaborative exploratory data analysis. In *Proc. VRST* (2019), ACM, New York, pp. 9:1–9:12. doi:10/gm8b84.

[CGS*22] COOLS R., GOTTSACKER M., SIMEONE A., BRUDER G., WELCH G., FEINER S.: Towards a desktop-AR prototyping framework: Prototyping cross-reality between desktops and augmented reality. In *Adjunct Proc. ISMAR* (2022), IEEE Comp. Soc., Los Alamitos, pp. 175–182. doi:10/gsghhv.

[CMGS25] COOLS R., MAEREVOET I., GOTTSACKER M., SIMEONE A. L.: Comparison of cross-reality transition techniques between 3D and 2D display spaces in desktop-AR systems. *IEEE Trans Vis Comput Graph 31*, 5 (2025), 2798–2805. doi:10/p68r.

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, 1999. urn:oclc:record:1036805840.

[CSCM21] CRNOVRSANIN T., SHILPIKA, CHANDRASEGARAN S., MA K.-L.: Staged animation strategies for online dynamic networks. *IEEE Trans Vis Comput Graph 27*, 2 (2021), 539–549. doi:10/gksh3b.

[DBJ*11] DRAGICEVIC P., BEZERIANOS A., JAVED W., ELMQVIST N., FEKETE J.-D.: Temporal distortion for animated transitions. In *Proc. CHI* (2011), ACM, New York, pp. 2009–2018. doi:10/dcrcw5.

[DCZL15] DU F., CAO N., ZHAO J., LIN Y.-R.: Trajectory bundling for animated transitions. In *Proc. CHI* (2015), ACM, New York, p. 289–298. doi:10/gpcs8s.

[DRST14] DÜBEL S., RÖHLIG M., SCHUMANN H., TRAPP M.: 2D and 3D presentation of spatial data: A systematic review. In *Proc. 3DVis* (2014), IEEE Comp. Soc., Los Alamitos, pp. 11–18. doi:10/gftgq4.

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans Vis Comput Graph 14*, 6 (2008), 1539–1148. doi:10/cg49c9.

[FAP*22] FRÖHLER B., ANTHES C., POINTECKER F., FRIEDL J., SCHWAJDA D., RIEGLER A., TRIPATHI S., HOLZMANN C., BRUNNER M., JODLBAUER H., JETTER H.-C., HEINZL C.: A survey on cross-virtuality analytics. *Comput Graph Forum 41*, 1 (2022), 465–494. doi:10/gtnn2w.

[FS91] FEINER S., SHAMASH A.: Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers. In *Proc. UIST* (1991), ACM, New York, pp. 9–17. doi:10/fcmt4b.

[GFM*22] GALL A., FRÖHLER B., MAURER J., KASTNER J., HEINZL C.: Cross-virtuality analysis of rich X-ray computed tomography data for materials science applications. *Nondestr Test Eval 37*, 5 (2022), 566–581. doi:10/pqfc.

[GHFH23] GALL A., HEIM A., FRÖHLER B., HEINZL C.: Uncertainty unveiled: Revealing the uncertainty of distribution visualization through cross reality. In *Adjunct Proc. ISMAR* (2023), IEEE Comp. Soc., Los Alamitos, pp. 20–24. doi:10/pqd9.

[GHW*25] GALL A., HEIM A., WEINBERGER P., FRÖHLER B., KASTNER J., HEINZL C.: Immersive inspection: Intuitive material analysis using X-ray computed tomography data in AR. *J Nondestruct Eval 44*, 3 (2025), 79:1–79:10. doi:10/qjzq.

[HHS*24] HONG J., HNATYSHYN R., SANTOS E. A. D., MACIEJEWSKI R., ISENBERG T.: A survey of designs for combined 2D+3D visual representations. *IEEE Trans Vis Comput Graph 30*, 6 (2024), 2888–2902. doi:10/m7r5.

[HMK*19] HORAK T., MATHISEN A., KLOKMOSE C. N., DACHSELT R., ELMQVIST N.: Vistribute: Distributing interactive visualizations in dynamic multi-device setups. In *Proc. CHI* (2019), ACM, New York, pp. 616:1–616:13. doi:10/ggcn2k.

[HMK*20] HALLADJIAN S., MIAO H., KOUŘIL D., GRÖLLER M. E., VIOLA I., ISENBERG T.: Scale Trotter: Illustrative visual travels across negative scales. *IEEE Trans Vis Comput Graph 26*, 1 (2020), 654–664. doi:10/kt3k.

[HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Trans Vis Comput Graph 13*, 6 (2007), 1240–1247. doi:10/dvqddt.

[HRD*19] HURTER C., RICHE N. H., DRUCKER S. M., CORDEIL M., ALLIGIER R., VUILLEMOT R.: FiberClay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Trans Vis Comput Graph 25*, 1 (2019), 704–714. doi:10/kvhw.

[HWF*22] HUBENSCHMID S., WIELAND J., FINK D. I., BATCH A., ZAGERMANN J., ELMQVIST N., REITERER H.: ReLive: Bridging in-situ and ex-situ visual analytics for analyzing mixed reality user studies. In *Proc. CHI* (2022), ACM, New York, pp. 24:1–24:20. doi:10/g8rvq3.

[Jac05] JACKSON R. E.: Stereographic projection techniques for geologists and civil engineers, second edition (Richard J. Lisle and Peter R. Leyshon). *Environ Eng Geosci 11*, 2 (2005), 184–185. doi:10/dz6fgq.

[Jen12] JENNY B.: Adaptive composite map projections. *IEEE Trans Vis Comput Graph 18*, 12 (2012), 2575–2582. doi:10/f4ftz9.

[KFS*17] KRONE M., FRIESS F., SCHARNOWSKI K., REINA G., FADEMRECHT S., KULSCHEWSKI T., PLEISS J., ERTL T.: Molecular surface maps. *IEEE Trans Vis Comput Graph 23*, 1 (2017), 701–710. doi:10/gfz679.

[KMM*18] KREISER J., MEUSCHKE M., MISTELBAUER G., PREIM B., ROPINSKI T.: A survey of flattening-based medical visualization techniques. *Comput Graph Forum 37*, 3 (2018), 597–624. doi:10/gdw9pq.

[LCKP25] LIAO S., CHAUDHRI S., KARWA M. K., POPESCU V.: SeamlessVR: Bridging the immersive to non-immersive visualization divide. *IEEE Trans Vis Comput Graph 31*, 5 (2025), 2806–2816. doi:10/pqcc.

[LCP*22] LEE B., CORDEIL M., PROUZEAU A., JENNY B., DWYER T.: A design space for data visualisation transformations between 2D and 3D in mixed-reality environments. In *Proc. CHI* (2022), ACM, New York, pp. 25:1–25:14. doi:10/kvj4.

[LGS*25] LI D., GUO X., SHU X., XIAO L., YU L., LIU S.: RouteFlow: Trajectory-aware animated transitions. In *Proc. CHI* (2025), ACM, New York, pp. 1116:1–1116:17. doi:10/qh5f.

[LNP*23] LU F., NANJAPPAN V., PARSONS P., YU L., LIANG H.-N.: Effect of display platforms on spatial knowledge acquisition and engagement: An evaluation with 3D geometry visualizations. *J Vis 26*, 3 (2023), 667–686. doi:10/grd3tz.

[LSC*23] LEE B., SATYANARAYAN A., CORDEIL M., PROUZEAU A., JENNY B., DWYER T.: Deimos: A grammar of dynamic embodied immersive visualisation morphs and transitions. In *Proc. CHI* (2023), ACM, New York, pp. 810:1–810:18. doi:10/g8rvq5.

[MAB*18] MOHAMMED H., AL-AWAMI A. K., BEYER J., CALI C., MAGISTRETTI P., PFISTER H., HADWIGER M.: Abstractocyte: A visual tool for exploring nanoscale astroglial cells. *IEEE Trans Vis Comput Graph 24*, 1 (2018), 853–861. doi:10/gcqk6d.

[MAE*98] MA C., ARIAS E. F., EUBANKS T. M., FEY A. L., GONTIER A.-M., JACOBS C. S., SOVERS O. J., ARCHINAL B. A., CHARLOT P.: The international celestial reference frame as realized by very long baseline interferometry. *Astron J 116*, 1 (1998), 516–546. doi:10/d7xfp3.

[MC01] MULCAHY K. A., CLARKE K. C.: Symbolization of map projection distortion: A review. *Cartogr Geogr Inf Sci 28*, 3 (2001), 167–182. doi:10/fjcg7f.

[MDLI*18] MIAO H., DE LLANO E., ISENBERG T., GRÖLLER M. E., BARIŠIĆ I., VIOLA I.: DimSUM: Dimension and scale unifying map for visual abstraction of DNA origami structures. *Comput Graph Forum 37*, 3 (2018), 403–413. doi:10/gdw4j6.

[MJC25] MCDADE J., JING A., CUNNINGHAM A.: RealityDrop: A framework using multimodal superhuman techniques to manipulate cross-reality virtual content. In *Adjunct Proc. UIST* (2025), ACM, New York, pp. 110:1–110:3. doi:10/qjzp.

[MNT*24] MÖRTH E., NIELSEN C., TURNER M., XIANHAO L., KELLER M., BEYER J., PFISTER H., CHEN Z.-T., GEHLENBORG N.: *A Mixed Reality and 2D Display Hybrid Approach for Visual Analysis of 3D Tissue Maps*. OSF preprint, 2024. doi:10/pqfd.

[MTUK95] MILGRAM P., TAKEMURA H., UTSUMI A., KISHINO F.: Augmented reality: A class of displays on the reality-virtuality continuum. In *Proc. Telemanipulator and Telepresence Technologies* (1995), SPIE, Bellingham, pp. 282–292. doi:10/dh8jnv.

[MVB*17] MEUSCHKE M., VOSS S., BEUING O., PREIM B., LAWONN K.: Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms. *IEEE Trans Vis Comput Graph 23*, 1 (2017), 761–770. doi:10/f92d8h.

[NVV*06] NERI E., VANNOZZI F., VAGLI P., BARDINE A., BARTOLOZZI C.: Time efficiency of CT colonography: 2D vs 3D visualization. *Comput Med Imaging Graph 30*, 3 (2006), 175–180. doi:10/br3jwx.

[PFKJA24] POINTECKER F., FRIEDL-KNIRSCH J., JETTER H.-C., ANTHES C.: From real to virtual: Exploring replica-enhanced environment transitions along the reality-virtuality continuum. In *Proc. CHI* (2024), ACM, New York, pp. 799:1–799:13. doi:10/g8rvq9.

[PNB*21] PAVANATTO L., NORTH C., BOWMAN D. A., BADEA C., STOAKLEY R.: Do we still need physical monitors? An evaluation of the usability of AR virtual monitors for productivity work. In *Proc. VR* (2021), IEEE Comp. Soc., Los Alamitos, pp. 759–767. doi:10/gj3dh5.

[RD19] REIPSCHLÄGER P., DACHSELT R.: DesignAR: Immersive 3D-modeling combining augmented reality with interactive displays. In *Proc. ISS* (2019), ACM, New York, pp. 29–41. doi:10/gtdktf.

[RIK*25] RAU T., ISENBERG T., KOEHN A., SEDLMAIR M., LEE B.: Traversing dual realities: Investigating techniques for transitioning digital objects between desktop and augmented reality environments. In *Proc. CHI* (2025), ACM, New York, pp. 1236:1–1236:16. doi:10/pj4f.

[RKP*24] RAU T., KOCH M., PATHMANATHAN N., ISENBERG T., WEISKOPF D., SEDLMAIR M., KÖHN A., KURZHALS K.: Understanding collaborative learning of molecular structures in AR with eye tracking. *IEEE Comput Graph Appl 44*, 6 (2024), 39–51. doi:10/nt8b.

[RSK24] RAU T., SEDLMAIR M., KÖHN A.: chARpack: The chemistry augmented reality package. *J Chem Inf Model 64*, 12 (2024), 4700–4708. doi:10/nt62.

[SFP*23] SCHWAJDA D., FRIEDL J., POINTECKER F., JETTER H.-C., ANTHES C.: Transforming graph data visualisations from 2D displays into augmented reality 3D space: A quantitative study. *Front Virtual Real 4* (2023), 1155628:1–1155628:19. doi:10/kvj9.

[SM05] SCHROEDER W. J., MARTIN K. M.: Overview of visualization. In *Visualization Handbook*, Hansen C. D., Johnson C. R., (Eds.). Butterworth-Heinemann, Burlington, 2005, pp. 3–35. doi:10/bkdf8t.

[Sny87] SNYDER J. P.: *Map Projections: A Working Manual*. Tech. Rep. 1395, U.S. Government Printing Office, 1987. doi:10/gk8jp7.

[SSP*23] SCHRÖDER J.-H., SCHACHT D., PEPER N., HAMURCULU A. M., JETTER H.-C.: Collaborating across realities: Analytical lenses for understanding dyadic collaboration in transitional interfaces. In *Proc. CHI* (2023), ACM, New York, pp. 97:1–97:16. doi:10/g8rvrc.

[TAA*21] TOMINSKI C., ANDRIENKO G., ANDRIENKO N., BLEISCH S., FABRIKANT S. I., MAYR E., MIKSCH S., POHL M., SKUPIN A.: Toward flexible visual analytics augmented through smooth display transitions. *Visual Inf 5*, 3 (2021), 28–38. doi:10/gm58fs.

[TKAM06] TORY M., KIRKPATRICK A. E., ATKINS M. S., MÖLLER T.: Visualization task performance with 2D, 3D, and combination displays. *IEEE Trans Vis Comput Graph 12*, 1 (2006), 2–13. doi:10/ccjthz.

[TLLS20] THOMPSON J., LIU Z., LI W., STASKO J.: Understanding the design space and authoring paradigms for animated data graphics. *Comput Graph Forum 39*, 3 (2020), 207–218. doi:10/gt24np.

[TLS21] THOMPSON J. R., LIU Z., STASKO J.: Data Animator: Authoring expressive animated data graphics. In *Proc. CHI* (2021), ACM, New York, pp. 15:1–15:18. doi:10/gksmxh.

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: Can it facilitate? *Int J Hum Comput Stud 57*, 4 (2002), 247–262. doi:10/d5frg6.

[WBR*20] WANG X., BESANÇON L., ROUSSEAU D., SERENO M., AMMI M., ISENBERG T.: Towards an understanding of augmented reality extensions for existing 3D data analysis tools. In *Proc. CHI* (2020), ACM, New York, pp. 528:1–528:13. doi:10/gm4jj8.

[WM22] WANG N., MAURER F.: A design space for single-user cross-reality applications. In *Proc. AVI* (2022), ACM, New York, pp. 31:1–31:5. doi:10/g8rvrf.

[YDM*21] YANG Y., DWYER T., MARRIOTT K., JENNY B., GOODWIN S.: Tilt Map: Interactive transitions between choropleth map, prism map and bar chart in immersive environments. *IEEE Trans Vis Comput Graph 27*, 12 (2021), 4507–4519. doi:10/qjcc.

[YJD*18] YANG Y., JENNY B., DWYER T., MARRIOTT K., CHEN H., CORDEIL M.: Maps and globes in virtual reality. *Comput Graph Forum 37*, 3 (2018), 427–438. doi:10/gf78xn.