



**HAL**  
open science

# Exploring the Design of Animated Transitions between Abstract and Concrete Visualizations in Immersive Environments

Ambre Assor, Michael J McGuffin, Arnaud Prouzeau, Pierre Dragicevic,  
Martin Hachet

## ► To cite this version:

Ambre Assor, Michael J McGuffin, Arnaud Prouzeau, Pierre Dragicevic, Martin Hachet. Exploring the Design of Animated Transitions between Abstract and Concrete Visualizations in Immersive Environments. 2024. hal-04552817

**HAL Id: hal-04552817**

**<https://hal.science/hal-04552817v1>**

Preprint submitted on 19 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Exploring the Design of Animated Transitions between Abstract and Concrete Visualizations in Immersive Environments

AMBRE ASSOR, Inria, CNRS, Université de Bordeaux, France

MICHAEL J. MCGUFFIN, ÉTS, Canada

ARNAUD PROUZEAU, PIERRE DRAGICEVIC, and MARTIN HACHET, Inria, CNRS, Université de Bordeaux, France

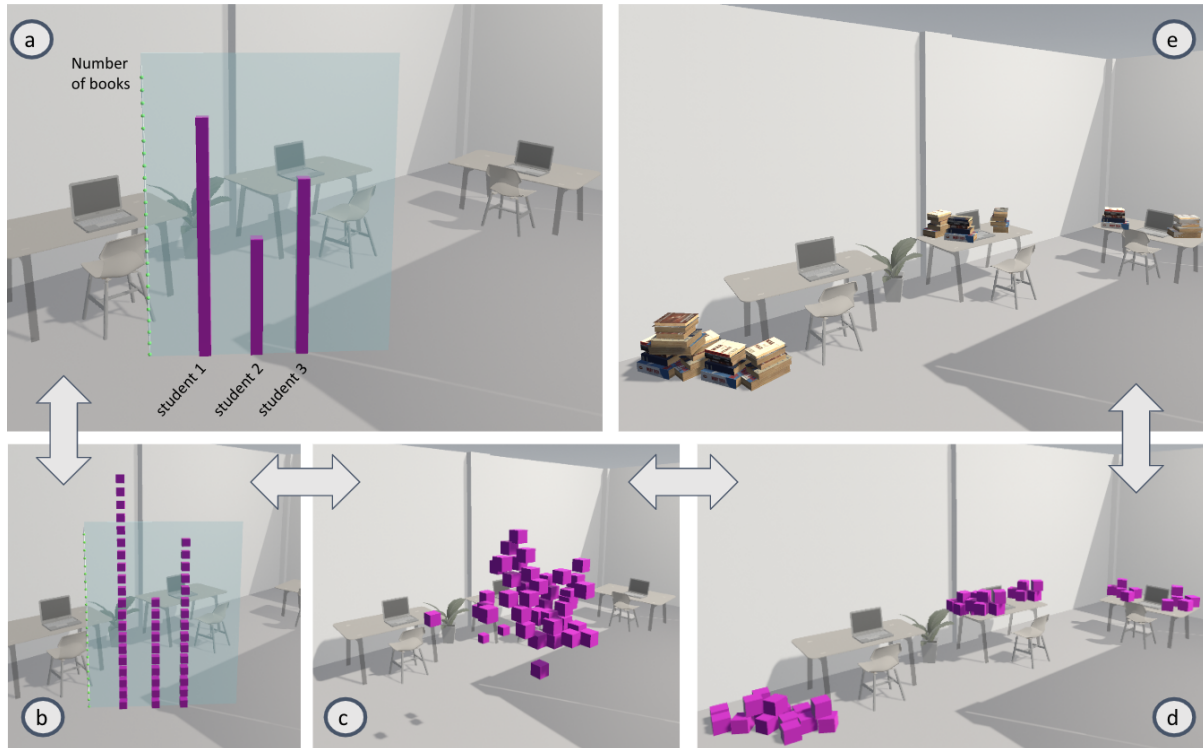


Fig. 1. Stages of an animated transition between an abstract (a) and concrete (e) visualization. The scene shows three students' desks, and how many books were borrowed by each student over the past month, either with bars (a) or with piles of books (e). To smoothly animate a transition from (a) to (e), the bars are decomposed into cubes (b) that are scaled and translated toward the appropriate desk (c,d) and then morphed (e) into books.

While data visualizations are typically abstract, there is a growing body of work around concrete visualizations, which use familiar objects to convey data. Concrete visualizations can complement abstract ones, especially in immersive analytics, but it is unclear how to design smoothly animated transitions between these two kinds of representations. We investigate a design space of abstract and concrete visualizations, where animated transitions are pathways through the design space. The design space is defined with four axes, each corresponding to a different transformation. We consider different ways to design animated transitions by staging and ordering the transformations along these axes. In a controlled experiment conducted in

Authors' addresses: [Ambre Assor](mailto:ambre.assor@inria.fr), ambre.assor@inria.fr, Inria, CNRS, Université de Bordeaux, Bordeaux, France; [Michael J. McGuffin](mailto:michael.mcguiffin@etsmtl.ca), michael.mcguiffin@etsmtl.ca, ÉTS, Montreal, Canada; [Arnaud Prouzeau](mailto:arnaud.prouzeau@inria.fr), arnaud.prouzeau@inria.fr; [Pierre Dragicevic](mailto:pierre.dragicevic@inria.fr), pierre.dragicevic@inria.fr; [Martin Hachet](mailto:martin.hachet@inria.fr), martin.hachet@inria.fr, Inria, CNRS, Université de Bordeaux, Bordeaux, France.

virtual reality with 16 participants, we compared four types of animated transitions and found quantitative and qualitative evidence of the superiority of a specific staging approach over the simultaneous application of all transformations. Our study pre-registration is available at [https://osf.io/8mu73?view\\_only=f5ed74fd9e2346228214f5dec973b208](https://osf.io/8mu73?view_only=f5ed74fd9e2346228214f5dec973b208)

CCS Concepts: • **Human-centered computing** → **Visualization techniques**; *Information visualization*; *Empirical studies in visualization*.

Additional Key Words and Phrases: information visualization, AR/VR, animation

## 1 INTRODUCTION

Virtual and augmented reality headsets enable the representation of data in a 3D human-scaled space. Such visualizations can be made more *concrete* [10] by employing marks that resemble physical objects and/or that are positioned with respect to physical objects, as in Figure 1e where numbers of books are shown by realistic piles of books on the desks of the respective students who borrowed them. Other examples of concrete visualizations involve human figures [27, 28], distances shown as physical lengths [28], calories shown as equivalent food items, waste shown as a pile of trash bags [2], or layouts based on the physical world (e.g., the desks in Figure 1).

These concrete visualizations can make it easier for a user to understand data in relation to everyday activities, and can sometimes evoke a stronger emotional response [28]. More traditional *abstract* visualizations, like bar charts, have complementary advantages, such as being untethered to any physical space, and enabling more precise quantitative comparisons. For example, Figure 1a makes it clear that student 1 borrowed the most books. One strategy to combine the advantages of concrete and abstract visualizations is to enable *animated transitions* [23] between them, so the user may freely change the way the data is shown according to their needs. Figure 1 shows stages of an immersive animated transition between a bar chart and virtual books. The animation disaggregates the bars into cubic units (b) allowing a more straightforward representation of book quantities, enabling them to travel in space toward their corresponding locations in the physical world (c), and semantically connecting the abstract representation to these locations (d). The use of realistic book shapes alongside familiar objects (e) can provide a more tangible sense of volume of the data, and enhance our understanding of their scale.

Within existing literature, the closest work we know to this topic is Ivanov et al.'s [27], who animate between different concrete visualizations in VR. We know of no previous work focusing on the design or evaluation of animated transitions between abstract and concrete immersive visualizations. Our work helps to fill this gap by investigating several variations of transitions between concrete and abstract visualizations in immersive environments. Specifically, we: (1) present a design space with 4 axes (Marks, Embedding, View, and Scale), to frame our approach for designing these transitions; (2) present a controlled experiment in VR, yielding quantitative and qualitative evidence that our [ME-VS] condition (which displays changes in Marks and Embedding first, followed by View and Scale transformations) outperforms the [MEVS] condition (a transition showing all transformations simultaneously over the same total duration).

## 2 BACKGROUND

We review previous work related to concrete visualizations and animated transitions. In this article we adopt a broad view of concrete visualizations, according to which data visualizations lie on a continuum from abstract to concrete. By abstract we mean a conventional data visualization, where data values are mapped to elementary visual variables such as position, size, or color [37]. We start this section by discussing one way of making bar charts – a common type of abstract visualization – slightly more concrete: turning it into a unit chart.

### 2.1 Unit Charts

A *unit chart* is a chart that contains repetitions of visual symbols, whose number is proportional to the quantity represented [22, 39]. For example, a unit chart may represent 20,000 cars with 20 car icons, each one standing for

1,000 cars. Unit charts include *block charts*, using abstract symbols like circles or squares, slightly more concrete than bar charts as quantities are more explicit (not aggregated and mapped into bar lengths) and *pictorial unit charts*, even more concrete because they use pictograms that reveal the semantics of the data being represented. IsoTypes are a prominent family of pictorial unit charts [21]. Even in visual unit charts, symbols vary in realism, and the use of realistic symbols can make the chart more concrete. For instance, in unit charts about people, individuals can be depicted more or less realistically, ranging from geometric symbols to icons to photographs to actual individuals [36].

A further step towards more concrete visualizations can be taken by rendering realistic unit charts in 3D and embedding them in familiar 3D environments. We review this strategy next.

## 2.2 Concrete Visualizations

*Concrete visualization* has been defined by Nieman [38] as “*the process of transforming entities that are usually represented abstractly as numbers in a table or lines on a chart into objects in space (or a representation of real 3D space) available to our senses*”. Although this definition is vague on purpose, many examples from Nieman’s own work are pictorial unit charts rendered in 3D and embedded in familiar 3D environments. For example, in his short movie “Wood for Good - The Story of the Tree” (<http://www.carbonvisuals.com/projects/wood-for-good>), volumes of carbon emissions are conveyed using heaps of spheres rendered inside 3D environments. Nieman’s concept of concrete visualization was refined by Chevalier et al. [10] who defined *concrete scales* as the “*process of visually relating complex measures with familiar objects from the real world*”. The concrete scale framework focuses on strategies to convey complex quantities and units using familiar references. One such strategy is *unitization*, consisting of breaking down abstract quantities into more relatable chunks. For example, to represent the amount of sugar in a soft drink, one may add a pile of sugar cubes right next to the drink [10].

In all such examples, although the concrete visualizations occupy a 3D space, they are shown on flat displays. In this article, we are interested in concrete visualizations experienced in immersive environments, which we review next.

## 2.3 Immersive Visualizations

There has been a surge of interest lately in using immersive displays – e.g., virtual reality (VR) and augmented reality (AR) displays – to convey data, known as *immersive analytics* as “*the science of analytical reasoning facilitated by immersive human-computer interfaces*” [17, 43]. While many immersive analytic prototypes proposed so far use abstract visualizations (e.g., 3D node-link diagrams, 3D scatterplots), a small number of them use data representations that are concrete in the sense of Nieman [38]. We review them now.

Lee et al. [28] introduced *data visceralization* as “*a data-driven experience which evokes visceral feelings within a user to facilitate intuitive understanding of physical measurements and quantities*”. Many of their examples are concrete visualizations implemented in VR, such as a scene that conveys US debt through huge piles of dollar bills standing next to known landmarks, or realistic reconstructions of protests that convey protest attendance numbers. Similarly, Ivanov et al. [27] presented an immersive and interactive concrete visualization of mass-shooting data, where each victim is represented with a human silhouette in a VR environment. They speculate that immersing users into concrete visualizations can enhance their emotional connection with the data. Meanwhile, two studies explored the use of concrete visualizations in AR to show food waste [24] and domestic waste [2] in the user’s physical surroundings. The first is a usability study highlighting the limitations of current AR displays as a major impediment for the user’s experience, while the second one is a controlled study suggesting that concrete visualizations in AR (heaps of trash bags standing for waste amounts) evoke more emotions than numerals or concrete visualizations conveyed on a regular screen. Finally, Chen et al. [7] presented a tool for designing and creating 3D unit charts and concrete visualizations in AR.

## 2.4 Hybrid Abstract/Concrete Systems

Researchers have argued that concrete visualizations – whether shown on regular computer displays or with immersive displays – can make it easier for audiences to engage with the data and to understand magnitudes and quantities on a visceral level [2, 10, 38]. However in most situations, abstract visualizations are more suitable: when studying a dataset in depth, they are more likely to be effective at conveying accurate values and revealing patterns in complex datasets (like multidimensional datasets and time-series).

Understanding how visualization systems can support both abstract and concrete visualizations is crucial. These hybrid systems not only cater to diverse audiences but also facilitate various tasks, enabling users to answer a wider range of questions and gain diverse perspectives on the same data. HANNAH is an example of hybrid system that integrate 3D models of real objects and environments with abstract visualizations, blending 3D models of a waste water treatment plant with abstract visualizations of the plant’s process data. However, such systems do not support abstract and concrete representations of the *same* data. The immersive visualization of mass shooting data from Ivanov et al. [27], which we already discussed, comes close to being a hybrid abstract/concrete system, because it supports transitions from views where the user is immersed in a crowd to distant views analogous to unit charts.

Although the space of hybrid concrete-abstract visualizations has started to be explored, to our knowledge, there is no formal study that investigates how to smoothly transition between concrete and abstract representations.

## 2.5 Situated and Embedded Visualizations

Many AR visualizations [2, 24] are *situated*, in the sense that they are shown next to the physical object, physical person or physical space to which the data refers [47]. Situated visualizations that are deeply integrated into the physical environment have been called *embedded* [44, 47]. Willett et al. [47] argued that we should “*consider the feasibility of hybrid solutions that would let people transition between non-situated, situated, and embedded modes*”. Because embedded visualizations tend to be more concrete than non-embedded visualizations, in order to design such hybrid systems, it is important we know how to design effective animated transitions between abstract and concrete visualizations.

## 2.6 Animated transitions

Animated transitions are a type of user interface animation, where abrupt visual changes are turned into rapid and smooth animations [9, 16]. Animated transitions have long been used in user interfaces [9], and have more recently been employed in data visualization applications, often to help users remain oriented while switching between different visual representations of the same data [3, 23, 39]. Animated transitions have also been used in visualization to convey dynamic data [4, 25] and to explain data analysis pipelines [20]. Recently, there has been interest in using animated transitions in immersive visualizations systems, with researchers proposing conceptual frameworks and prototypes to illustrate what is possible in terms of transitions between 2D and 3D immersive visualizations [29, 30, 42, 48]. However, this work almost exclusively focuses on animated transitions between abstract (or mostly abstract) visualizations. On the other hand, the immersive visualization of mass shooting data by Ivanov et al. [27] (discussed in [subsection 2.4](#)) demonstrates animated transitions between concrete (or mostly concrete) visualizations. We are not aware of systems supporting animated transitions between fully abstract and fully concrete visualizations, as we propose.

Some studies have been conducted in immersive environments, such as Yang et. al’s who compared animated transitions between geographic visualizations with a side-by-side condition [48], in VR, through an elementary cartographic task or Schwajda et al. [42] who studied animated transitions between on-screen 2D node-link diagrams and AR 3D node-link diagrams, asking participants to track clusters between representations.

Doerr et al.[13] investigated how to connect a 2D visualization with corresponding physical referents in immersive environments. They specifically compared static approaches (visual links) to dynamic ones (pointing arrows) that move in space, semantically linking abstract localized visualizations with real-world objects. The latter requires users to track abstract virtual objects from a situated panel to the positions of objects in the real world. However, they did not explore the implications related to user perception and strategies related to the visual tracking of these objects in space, and they limited themselves to a single direction of animation (from abstract to concrete). Our work contributes to a better understanding of design parameters related to this type of animations.

### 3 DESIGNING ANIMATED TRANSITIONS BETWEEN CONCRETE AND ABSTRACT VISUALIZATIONS

Throughout approximately 10 group meetings, we (the co-authors) discussed examples of abstract and concrete visualizations, and realized that these could be mixed together (for example, upper right image of Figure 2 shows abstract bars that are concretely positioned on a geographic map), and possibly placed on a continuum. We also discussed the transformations involved in designing animated transitions between these variations. These discussions, which were informed by an analysis of existing literature, culminated in a design space involving 4 axes, illustrated in Figure 3 and Figure 4. In those figures, to remove the details of any particular visualization, we use cubes as proxies for concrete visualization objects and bars for abstract visualization marks. This decision stems from our choice to prioritize the study of spatial transformations associated with these animated transitions and to avoid delving into the intricacies of designing complicated morphing or dis-aggregation processes which would be necessary with more complicated objects.

#### 3.1 Design Space of Concrete and Abstract Visualizations

To design animated transitions between abstract and concrete visualizations, we define a space that enables comparisons along a common set of dimensions. Then, we will be able to define a set of transformations that must be performed on the visualizations to create seamless animated transitions.

**Scale:** Abstract visualizations such as Scatterplots or Barcharts are usually displayed at **reduced** scale (AAR3). However, Danyluk et al. [12] evaluated what they call *room-scaled* visualizations (ACP1, AAP1) that *allow viewers to explore data by physically walking through, under, and around it*. Unfortunately, they found that these performed worse than with smaller scales. Still, such larger scale visualizations, which we call **physical** scale, are relevant for concrete visualizations (UAP1, UCP1) as discussed in subsection 2.2. As a result, we consider both scales in our design space.

**View:** when a user visualizes data, the visualization is usually located in front of them, through a **third person view** (i.e., on a screen as AAR3). In cases similar to the one described just above for *room-scaled* visualizations, the user can be surrounded by graphical elements (UCP1, Figure 1). In this case, we refer as **first-person view**.

To avoid impractical combinations of Scale and View, and to make our design space easier to understand, we show the Scale and View axes varying together rather than independently in Figure 3 and Figure 4. We nevertheless distinguish between Scale and View because certain animated transitions may transform Scale and View during different stages, and we need a way to precisely describe such transitions.

**Marks:** Munzner defines marks as “*basic geometric elements that depict items or links*” (e.g., point, line, area) [37]. In the case of a concrete visualization, each visual **mark** refers to an object (UCR3, UAR3, UAP1 and UCP1), whereas in the case of abstract visualizations, usually depicting aggregations, a visual mark shows a quantitative value for an attribute (ACR3, AAR3, AAP1 and ACP1). Aggregates and units marks may come in different colors and shapes (e.g., the representation of a physical object, and a scatter plot data point respectively) and go beyond

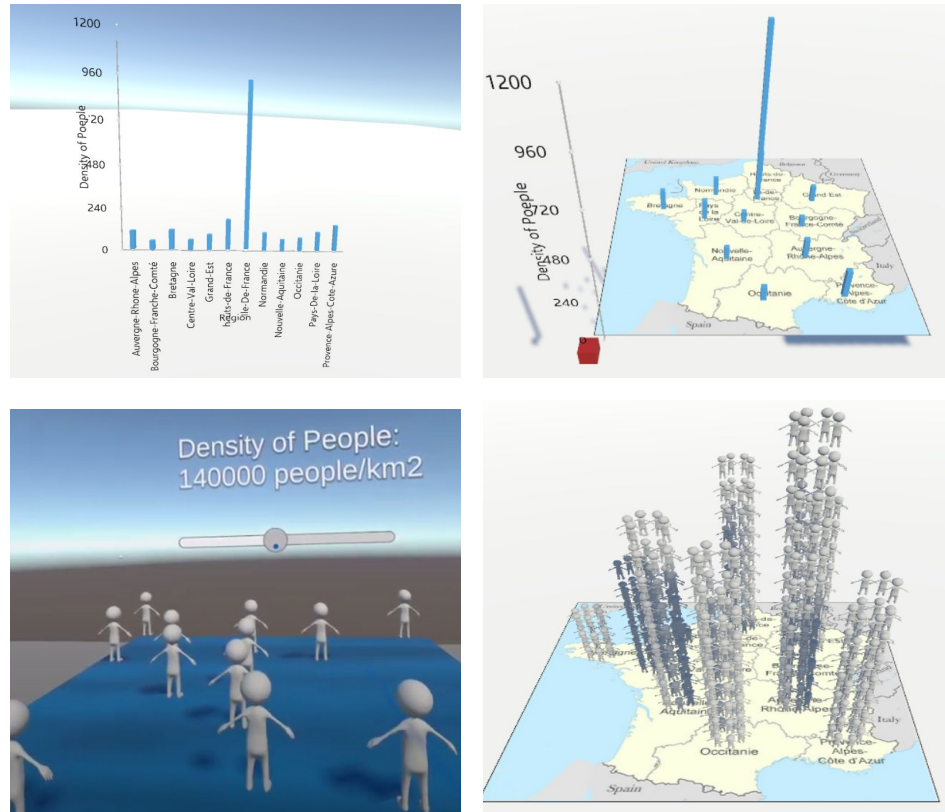


Fig. 2. Four visualizations of population densities in VR. Clockwise from upper left, the visualization is increasingly concrete. Top left: a barchart shows population densities of regions of France. Top right: the same bars are positioned (embedded) on a map of France. Bottom right: the bars replaced by stacked stickmen units (1 unit =  $10^5$  people). Bottom left: people matching the density selected with the slider are displayed on a ground plane (these human figures could be shown at a reduced scale, to provide an overview of a unit of geographic area, or scaled up to full size, to more closely simulate a viewer's real-world experience).

what we depict in [Figure 4](#). For simplicity, we kept variables such as color, texture, and shape constant, and do not discuss transformations such as morphing or color changes.

**Embedding:** If the visualization's mark positions derive from the physical world characteristics, the embedding is **concrete**, if not it is **abstract**. Indeed, an abstract visualization is relatively confined in a restricted area of space (as (a) on [Figure 1](#) or AAR3), whereas a concrete one tends to have greater disposition to exist at multiple specific locations (see [subsection 2.5](#), [Figure 1\(b\)](#) or UCP1).

We clarify that what is referred to as “*the spatial substrate in which marks are embedded*” in [23] also changes regarding our axes. In both [23] and [32], they mention that a scaling of the visualization may require a different scale for the substrate, which is often a graph axis. In our case, the substrate differs when the marks are abstractly or concretely **embedded** as the latter demands that the marks exist at a specific spatial location. For instance, in [Figure 1](#), marks exist in a 2D space on (a) and in a 3D space of a specific shape on (d), which results in different substrate dimensions and shape (respectively a graph axis and the virtual ground).


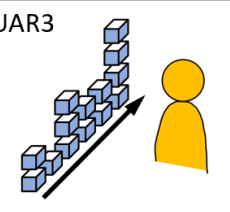
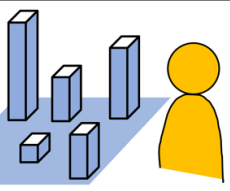
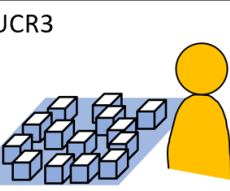
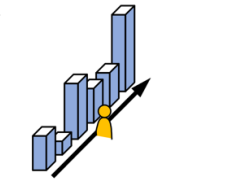
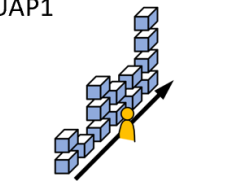
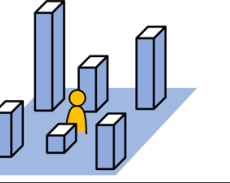
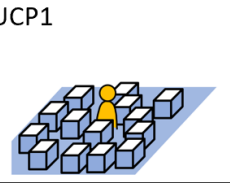
		Marks	
		Aggregated	Unit
Scale is Reduced, View is 3rd person	Embedding is Abstract	AAR3 	UAR3 
	Embedding is Concrete	ACR3 	UCR3 
Scale is Physical, View is 1st person	Embedding is Abstract	AAP1 	UAP1 
	Embedding is Concrete	ACP1 	UCP1 

Fig. 3. Our design space. Unit Marks show individual items (cubes) whereas Aggregated Marks show collections (bars); these are positioned according to some physical or geographic space (Concrete Embedding) or by some algorithm (Abstract Embedding as a barchart axis); View may be 3rd person (in front of the user) or 1st person (surrounding the user); Scale may be Reduced (small enough for comfortable viewing) or Physical (e.g., life size). Each 4-letter acronym identifies a variant within Marks × Embedding × Scale × View = {U,A}×{C,A}×{R,P}×{1,3}. To avoid impractical combinations of Scale and View, and to avoid making this table excessively complicated, we show the Scale and View axes varying together rather than independently.

### 3.2 Design Considerations

We use the term *transition* to refer to a pathway through the design space of the previous section (Figure 4). Each transition involves one or more *transformations*, which may be changes to the Marks, Embedding, View or Space (denoted with the first letter of these words). More generally, a transformation may be a translation, rotation, morphing, or fading in/out of elements. The transformations in a transition may be simultaneous or may be separated into sequential stages, causing the transition to pass through intermediate vertices in the design space.



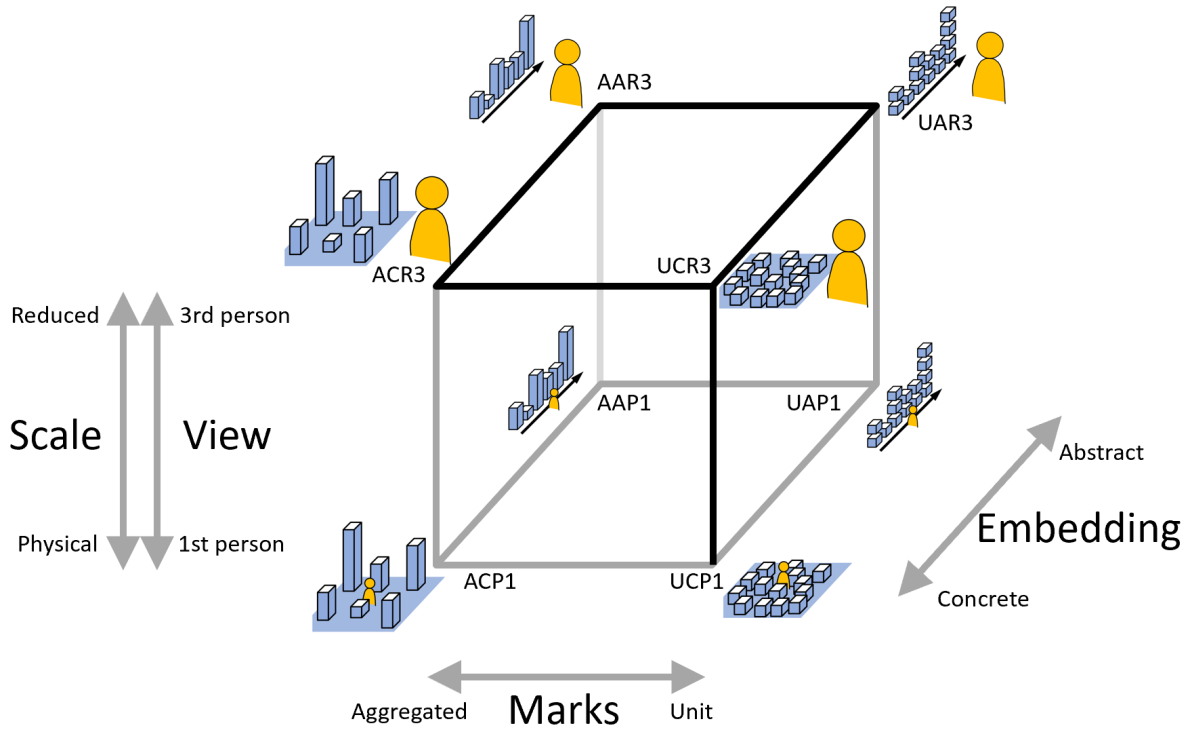


Fig. 4. A diagram of our design space. Animated transitions are pathways through this design space. To avoid impractical combinations of Scale and View, and to avoid making the diagram overly complicated, we show the Scale and View axes varying together.

Since there are many possible transitions (i.e., pathways) through the design space to get from one vertex to another, it would be helpful to have some rules for choosing the best transitions. We formulated several Design Considerations (**DC**) for animated transitions, some of which are supported by previous work. Some of these considerations conflict with each other and cannot all be complied with. As we will see later, different transitions constitute different tradeoffs with these considerations.

▷ Design Consideration **DC1** *Use stages*: divide transitions into stages to show different kinds of transformations at different times.

SpaceTrees [40] are the earliest example we know using animated transitions broken into stages in an information visualization. Stages can help the user understand a transition by showing a simpler set of transformations at each moment.

▷ **DC2** *Move slower*: avoid high-speed translations over long distances. (This conflicts with DC1, assuming a constant total duration.)

Very fast translations cannot be followed by the user, making it difficult to understand the relationship between items before and after a transition. If transitions have a fixed total time (as is the case in our experiment in section 4), there is a tradeoff between DC1 and DC2: having more stages leaves less time for each stage, which means that translations will be at a higher speed. We compared transitions comprising one, two, and three stages in our experiment, enabling an indirect exploration of this tradeoff.

▷ **DC7** *Move fewer*: reduce the number of items that are visible during translations. For example, items to be removed by the end of a transition can be removed before translating other items, and items to be added by the end of a transition can be added after others are translated. This idea is used in [34, 40].

▷ **DC4** *Move each subset as one*: If many items must undergo individual translations, and these items form a meaningful subset, it is preferable to first translate the entire subset as a single rigid body before applying individual translations to the items in the subset.

An example of DC4 is seen in the “hybrid” transitions of [19] when a parent node must be translated and its children must also be permuted to new locations within the parent: the parent and its children are first translated as a single object, and then the children undergo smaller translations to their new locations. This results in one coarse-grain motion first, followed by finer-grain motion of individual items, allowing the user to progressively narrow their attention, as desired.

▷ **DC5** *Move smaller*: if items must be translated and must also change size, it is preferable to order stages so items are smaller during translations (e.g., shrinking before translating, or translating before growing).

▷ **DC6** *Move aggregations*: if items must be translated and must also be aggregated into a whole (or disaggregated into parts), it is preferable to translate aggregations rather than parts. For example, aggregate many items into a whole before translating, or translate before disaggregating into parts. This consideration can be seen as a special case of DC7, but may sometimes conflict with DC5 if the aggregations are visually larger than their parts.

DC7-DC6 are all ways to simplify the appearance of motions and/or reduce inter-item occlusion, by reducing the size or number of moving items.

▷ **DC3** *Use exocentrism*: when transformations involve all items, it is preferable to perform them within the user’s field of view.

DC3 recommends using an exocentric view (a 3rd-person view, in front of the user) rather than egocentric (a 1st person view, where the user is surrounded by items). This reduces the amount of head and eye rotation necessary to perceive the items. Exocentricity can lead to higher accuracy in judgment of target location [46] and is often the best frame in terms of visualization task accuracy in 3D virtual environments [33, 45].

▷ **DC8** *Slow down complexity*: slow down translations when items are closely spaced, and/or when items will be moving in different, unpredictable directions.

DC8 is motivated by reasoning similar to that behind the ‘adaptive’ animation technique of [16], where the animation is slower when the scene is more complex.

The next two considerations were formulated after seeing the results of our pilot study (subsection 4.7). They were not initially obvious to us, as they directly contradict some of the previous considerations, but they came out of wondering why certain transitions seemed to perform better than others.

▷ **DC9** *Transform all as one*: if items must be translated and scaled to extend beyond the user’s field of view, items can first be translated to their new relative positions while still within the user’s field of view during a first stage, and then in a 2nd stage, the entire collection of items can be scaled up as a single object, maintaining the relative positions of items during this 2nd stage. In the reverse direction, the entire collection can first be scaled down as a single object to fit in the user’s field of view (1st stage), and then items can be individually translated to new positions (2nd stage). This conflicts with DC5.

DC9 contradicts DC5 in deliberately varying the size of items as they are translated, rather than only translating items at their smallest size. The potential advantage of DC9 is that the user may be able to remember an item’s placement with respect to nearby items before a transition, and hence even if the user loses track temporarily of an item during the translation, the user might more easily find the item again by looking for its remembered location with respect to nearby items, e.g., by recognizing the ‘shape’ of a subset of items.

▷ **DC10** *Don’t move aggregations*: if items must be translated and must also be aggregated into a whole (or disaggregated into parts), it is preferable to translate parts rather than aggregations. For example, disaggregate

into parts before translating, or translate items before aggregating them into a whole. This consideration may support DC5 if aggregates are visually larger than their parts, but conflicts with DC7 and especially DC6.

The potential advantage of DC10 is that, by breaking up large aggregates before translating individual items, this might make it easier for the user to identify an individual item during motion, and/or reduce occlusion because the items are smaller than their aggregations.

As we will see in the next section, not all of these design considerations are directly applicable to the choice of transitions that we made, and different transitions constitute different tradeoffs with these considerations. We nevertheless enumerated the set of ten above to make it clearer how some considerations support or conflict with others. Also, the degree to which each consideration is valid likely depends on the details of the visualization in question, hence conflicts between them are not necessarily a sign of any consideration being “wrong”. The rationale behind each consideration may nevertheless be useful to designers for considering choices.

## 4 STUDY

### 4.1 Choice of transitions for study

Of the eight vertices in the design space, there are three that are of less interest: (ACP1), (UAP1), and (AAP1). These are at the back and bottom of [Figure 4](#), and correspond to large-scale visualizations with either aggregated marks or an abstract embedding, which seem to offer little theoretical advantage. [12] evaluated a condition similar to (ACP1) and found that it performed poorly compared to smaller scale visualizations.

We therefore focus on transitions involving the 5 remaining vertices. The most challenging transitions involve all 4 transformations (M, E, V, and S), i.e., transitions between diagonally opposing vertices. The only transitions like this within the 5 vertices of interest are transitions between (AAR3) and (UCP1). [Figure 5](#) shows the seven transitions that we chose for evaluation.

The first transition, [MEVS], shows all transformations simultaneously, in a single stage. The others involve two or three stages. In choosing transitions to evaluate, we eliminated transitions involving four stages (DC2), and also eliminated transitions passing through any of the three vertices ((ACP1), (UAP1), and (AAP1)) on the bottom that seem less useful. This results in transitions that never perform the V or S transformations before M or E, which makes sense if we want to avoid changes happening outside the user’s field of view (DC3). The [ME-V-S] transition is the only one that separates the V and S transformations (DC5), all other transitions we chose perform V and S within the same stage.

[Figure 6](#) shows the extent to which each of the seven transitions implements the design considerations in [subsection 3.2](#).

### 4.2 Choice of task

The specific placement of the concrete visualization in space is key for the user’s semantic understanding of the data initially or finally conveyed abstractly. This necessarily involves a trajectory in the immersive 3D space required by the transitioning from an abstract-to-concrete or concrete-to-abstract embedding for the visualization. We choose to evaluate strategies that might support the user in their cognitive ability to follow this trajectory, which must happen within a reasonable time for an animated transition. We therefore chose a visual tracking task for our study. This task has been used in many previous evaluations of animated transitions [8, 16, 23, 35] as object tracking is often a low-level component of many higher-level tasks: if a user cannot even follow individual objects during animated transitions, it is unlikely that they will be able to answer questions about the animated visualization’s dataset. Choosing a low-level task also enables us to achieve more experimental control (since we can fine-tune the characteristics of the trials) and more statistical power (the short duration of the task allows us to give participants more trials, and thus record more observations).

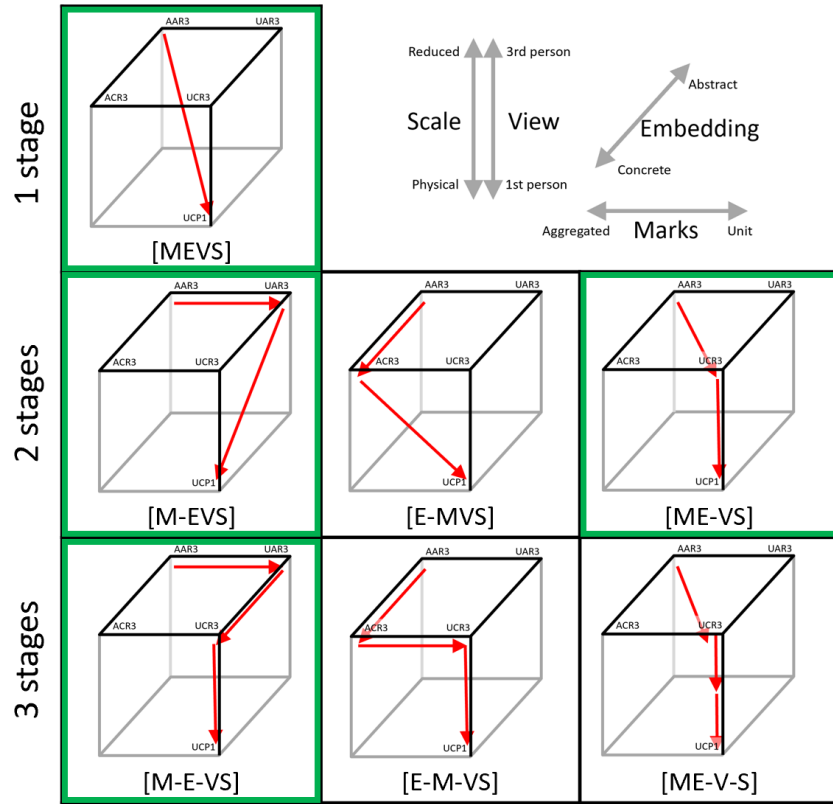


Fig. 5. The seven animated transitions evaluated in the pilot. Of these, the four highlighted in green ([MEVS], [M-EVS], [ME-VS], [M-E-VS]) were retained for the full experiment.

In our task, a single item (i.e., a cube, possibly within an aggregated bar) is designated with a visual highlight. The highlight is then removed, an animated transition plays out (Figure 7), and the user must indicate where the item is now located by selecting it. Early testing with simultaneously tracking of 2 items revealed this to be too difficult (especially when two targets moved to opposite sides of the user simultaneously), thus we limited our experimental task to single-target tracking.

### 4.3 Transfer function and duration

We define the *transfer function* as the function that maps a fraction of time of a stage to the fraction of progress of a transformation. Both fractions are numbers in the interval [0,1].

During early testing, our transfer function was linear, and we found it subjectively difficult to track items when they were small and densely packed, and also when they were moving at high speed over a long distance. For instance, in a Forward-direction transition (Figure 7), after items are disaggregated, the items are small and closely-spaced, before traveling at relatively high speed to reach their destinations which could be to the side of, or even behind, the user, requiring them to rotate their head. We reasoned that slowing down the initial motion would facilitate tracking (DC8), even at the cost of making the subsequent motion faster. Indeed, the user must be able to initially distinguish the target from other small items nearby. Then, once the user has seen the general

Relevant Design Guideline		1 stage		2 stages		3 stages		
		[MEVS]	[M-EVS]	[E-MVS]	[ME-VS]	[M-E-VS]	[E-M-VS]	[ME-V-S]
DG1	Has stages, with different transformations in each stage, possibly making each stage easier to understand		✓	✓	✓	✓	✓	✓
DG2	Has fewer stages, which reduces the speed of long-distance translations	✓	✓	✓	✓			
DG5	The <b>View</b> is changed before increasing the <b>Scale</b> from <b>Reduced</b> to <b>Physical</b> (and in the backward direction, the <b>Scale</b> is decreased from <b>Physical</b> to <b>Reduced</b> before changing the <b>View</b> ), avoiding occlusion during the change in <b>V</b>							✓
DG6	Changes the <b>Embedding</b> of bars before disaggregating them into cubes ( <b>M</b> ) (and in the backward direction, aggregates cubes into bars ( <b>M</b> ) before changing their <b>Embedding</b> ), so that fewer items are moving during the change in <b>E</b>			✓			✓	
DG7	Changes <b>Marks</b> and/or their <b>Embedding</b> at <b>Reduced Scale</b> and in <b>3rd</b> person, so that changes to <b>M</b> and <b>E</b> may be easier to see		✓	✓	✓	✓	✓	✓
DG9	Cubes have the same relative positions and same relative sizes before and after the change in <b>Scale</b>				✓	✓	✓	
DG10	Disaggregates bars into cubes ( <b>M</b> ) before changing their <b>Embedding</b> (and in the backward direction, changes the <b>Embedding</b> of cubes before aggregating them into bars ( <b>M</b> )), to reduce occlusion and make units more distinct		✓			✓		
	Retained after pilot study with 4 participants	✓	✓		✓	✓		
	Best in the final experiment with 16 participants				✓			

Fig. 6. Each transition (column) that complies with a design consideration (row) is indicated with a checkmark. Partial compliance is shown with a smaller checkmark.

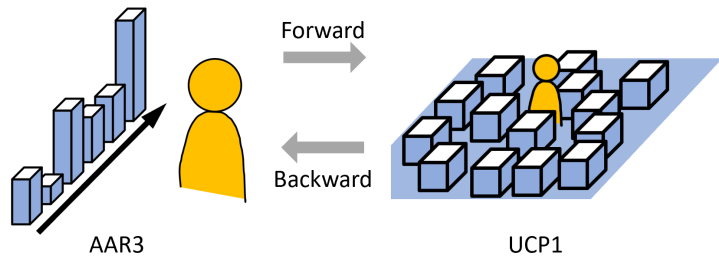


Fig. 7. In the pilot and full experiment, all transitions were between these two visualizations, with half of the trials in each direction. In the 2- and 3-stage transitions, there were also intermediate states not shown here.

direction of motion of an item, they could estimate where it might end up with less difficulty since the cubes are more spaced out later in the transition.

To determine how to modify the transfer function, our software was instrumented to allow the user to interactively adjust two control points that define a piecewise linear transfer function. Two of us (co-authors) each, independently, viewed several repetitions of a transition and adjusted the control points to our satisfaction, for a stage showing translations and lasting a total of 1 second. The resulting transfer functions are shown in

**Figure 8.** We finally chose the S-shaped function (Figure 8, right) for our experiment, because it is more similar to commonly-used ‘slow in, slow out’ animations [16], is symmetrical and therefore easier to implement in software without regard to the direction of a transition, and it results in an aesthetically-pleasing deceleration at the end of the transition.

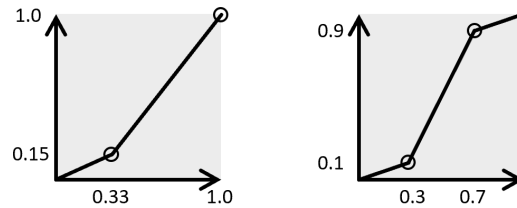


Fig. 8. Two transfer functions that were independently tuned by two of the co-authors. The one on the right was chosen for the experiment.

To determine the duration of our transitions, we looked to previous work where transitions lasted between 1.25 seconds [23] to 10-15 seconds [42]. We also analyzed screen-captured videos to measure the time it took for tooltips to appear in a variety of software applications on Microsoft Windows, and found that the time varied from 450ms to 850ms. Informal testing with our software revealed that if a stage lasted less than 1 second, it was often too difficult to follow. We therefore fixed the total duration of our transitions to 3 seconds. Hence, transitions having one, two, or three stages allocate 3.0, 1.5, or 1.0 seconds to each stage, respectively.

#### 4.4 Hardware

Users wore a Meta Quest Pro headset, untethered, and in VR mode (i.e., without video pass-through). The headset was adjusted to their interpupillary distance. Users could see some of the physical world through peripheral vision, as the headset does not completely block the physical world from view. Users also had two handheld controllers to point at objects. We measured a frame-rate of 71 fps when using our experimental stimuli.

During the experiment, participants were free to walk around a physical space measuring 10×10 feet, although there was a starting position marked on the floor that users were asked to return to at the start of each trial.

#### 4.5 Details of the task

Each Forward-direction trial involved showing the user a transition from a barchart in front of the user (AAR3) to a set of large cubes surrounding the user (UCP1), as shown in Figure 7 and Figure 9. In Backward-direction trials, the transition happened in the opposite direction (Figure 10).

Each barchart was composed of 6 bars, made of a total of 35 unit cubes, each corresponding to one of the 35 large cubes that would surround the user in the (UCP1) state. Each bar of the barchart was between 1 and 10 unit cubes tall, with the unit cubes distributed randomly between bars, and each unit cube measuring 6cm tall. The barchart was positioned 1.2m in front of the user’s starting position, and 1.3m above the ground plane.

Each large cube in the (UCP1) state was randomly positioned in a 10×10m square on the ground plane, and each measured 60cm tall, a size which is of the same order of magnitude as the marks in Danyluk et al.’s table-scale visualization [12] and in other immersive concrete visualizations [2, 24, 27]. Our choice of sizes (6cm and 60cm) implies a change in size during the transition of 10×, making the design of the transition more challenging.

At the start of each trial, there was a pre-animation phase that lasted 5 seconds. During this phase, one of the unit cubes in a bar or one of the large cubes was designated with a highlight to indicate that it would be

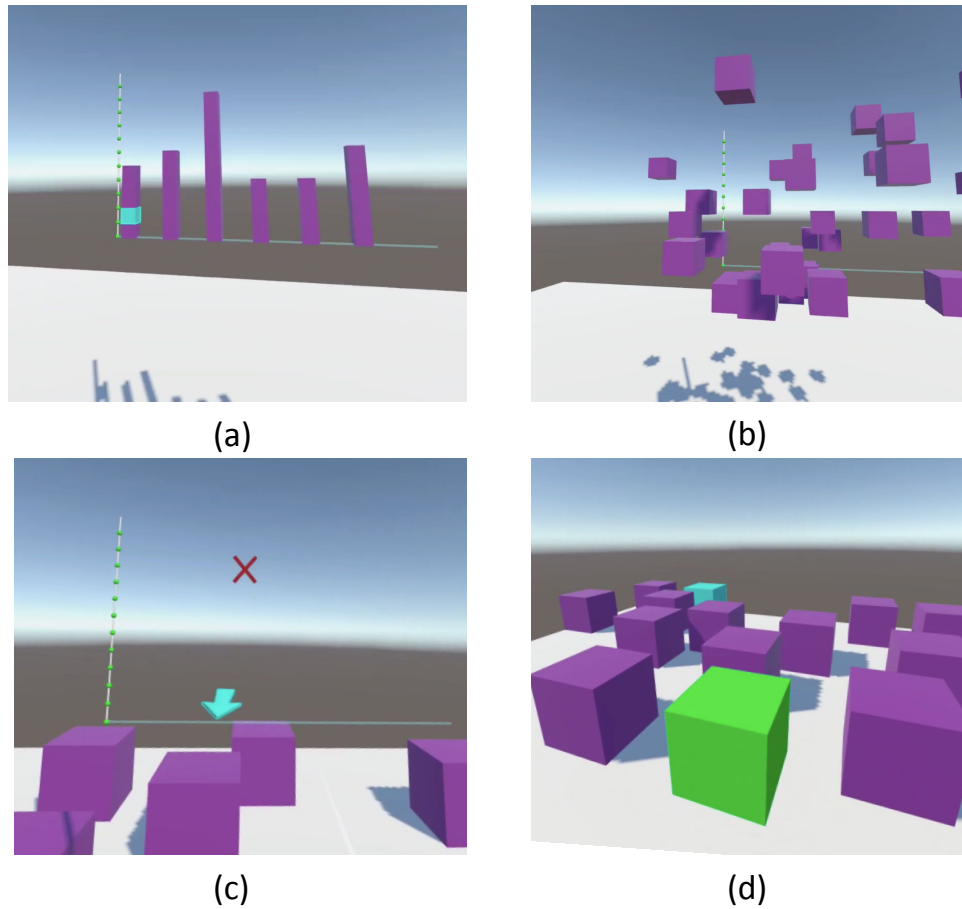


Fig. 9. Example of a Forward direction trial. (a) Pre-animation phase: the target cube to track is highlighted on the barchart. (b) The animated transition from abstract barchart to concrete visualization occurs. (Not shown) After the transition, the user selected the wrong cube and confirmed their choice. (c) Post-animation phase: a red cross indicates that the answer is wrong, and a blue arrow indicates that the correct cube is out of the participant’s field of view. (d) The user turns back a see their selected cube in blue and the correct answer in green.

the target. In the case of (UCP1), sometimes the designated target cube could be behind the user, so an arrow positioned in front of the user pointed to where the target cube was, helping the user find it.

At the end of the pre-animation phase, the highlight was removed from the target, and the transition was played, lasting 3 seconds.

Next, the user had to select the large cube, or (in the case of a Backward trial) the small unit cube, which they believed was the target. They performed this selection by pointing a raycast from a handheld controller at the cube and pressing a button. In case of a mistake, they could optionally select again to redo the selection. They confirmed their choice by selecting a “Validate” button in the 3D scene.

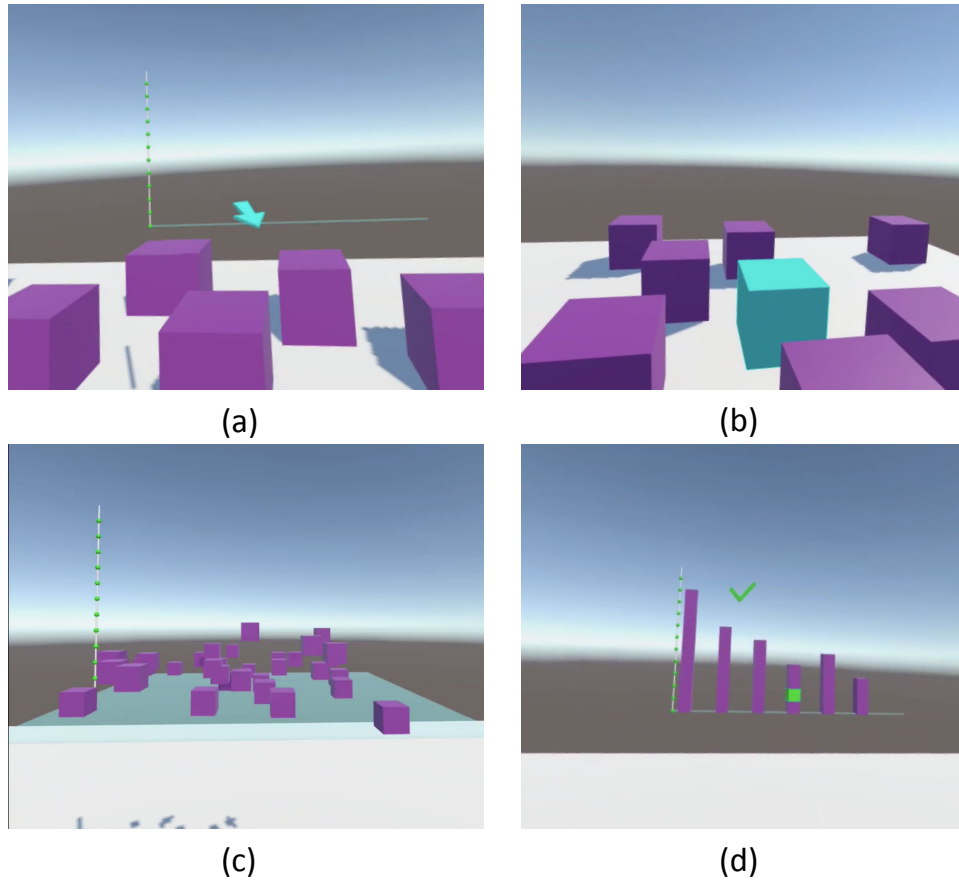


Fig. 10. Example of a Backward direction trial. (a) Pre-animation phase: the blue arrow points toward the cube to track, situated out of the user's field of view. (b) The participant turns around to see the cube to track, highlighted in blue. (c) The transition to barchart occurs. (Not shown) After the transition, the user selected the correct unit cube in the barchart. (d) Post-animation phase: a green check mark above the barchart indicates that the answer is correct.

Next, there was a post-animation phase lasting 5 seconds, during which time visual feedback indicated to the user if their selection was correct or not, and if not, a highlight showed the correct target cube. This completed the trial.

#### 4.6 Variables Measured

The time for each trial was measured from the end of the animated transition to the time of the last selection performed by the user before selecting the “validate” button.

For each trial, we recorded whether the user selected the correct target or not (as a boolean flag), to compute an error rate for each user in each condition. We also recorded the magnitude of the error in each trial in a few different ways. For our pilot study, we used the distance between the correct target and the cube selected by the user, and recorded this as both a euclidean error distance (in meters) and an angular error (where the angle



is calculated with respect to the user). In addition, the euclidean error distance was divided by the minimum distance between two cubes to obtain a normalized error distance.

After the pilot study, we realized that all of these metrics of error size are sensitive to local changes in the spacing between items, even the normalized error distance (since it is normalized with respect to a global minimum distance). We borrowed an idea from [18] and used a different way to quantify the magnitude of an error: for each trial in the full experiment, we recorded the number of cubes closer to the correct target (including the correct target itself) than the cube selected by the user, and we called this quantity **ItemsCloser**. If the user selects the correct target T, then ItemsCloser is zero; if the user selects a cube R that is close to T, with no other cube closer to T, then ItemsCloser is one. In other words, ItemsCloser is the number of responses that would have been better than the user’s response. We propose this as a better way of quantifying error when users are asked to select from a set of discrete possibilities, since it is not sensitive to changes in the local spacing between items.

#### 4.7 Pilot Study

Our pilot compared all 7 conditions in Figure 5, with 4 users.

The normalized error distances indicated that three of the multi-stage transitions might be better than the [MEVS] transition, while the other three multi-stage transitions appeared worse than [MEVS]. Since the [MEVS] transition would be a useful baseline for the final experiment, we kept it along with the three most promising multi-stage transitions, namely [M-EVS], [ME-VS], [M-E-VS], for the final experiment.

We wondered why these three particular multi-stage transitions might perform better than the others, and realized that each pair of them had some theoretical advantage. The first two transitions, [M-EVS] and [ME-VS], have the advantage of only having 2 stages, making translations slower than in 3-stage transitions (DC2). The last two transitions, [ME-VS] and [M-E-VS], transform View and Scale in the same stage, meaning that even if that stage involves some rapid translations and the user loses the target temporarily, the user might be able to remember a target’s position relative to other elements in the previous stage, and find the target again later based on the relative positions of elements. This follows DC9 and is illustrated in Figure 11, top. Finally, the first and third transitions, [M-EVS] and [M-E-VS], follow DC10: disaggregating bars before translating units may reduce occlusion and make an individual unit easier to follow. The other design considerations (Figure 6), some of which conflict with these, may be less important in this context.

#### 4.8 Full Experiment

Our final, full experiment was preregistered ([https://osf.io/8mu73?view\\_only=f5ed74fd9e2346228214f5dec973b208](https://osf.io/8mu73?view_only=f5ed74fd9e2346228214f5dec973b208)), declaring our sample size, hypothesis, and analysis.

A sample size of 16 users was chosen in the preregistration, not including the 4 pilot participants. While this choice was not based on a power analysis, after a comment from a reviewer, we note that this sample size yields a power of 0.75 to detect a “medium” standardized effect size of  $d=0.5$  in a within-subject experiment like ours.<sup>1</sup> Of the 16 users, 10 were women, 6 men; 14 right handed, 2 left handed, but all with a habit of using the mouse with their right hand; age 20 to 32 years (average 25.6); inter-pupillary distance (IPD) 58 to 66mm (average 61.8).

Within each condition, the user experienced an equal number of trials covering 2 directions  $\times$  8 repetitions in random order, for a total of 4 conditions ([MEVS], [M-EVS], [ME-VS], [M-E-VS]) in random order  $\times$  2 directions (Forward, Backward)  $\times$  8 repetitions  $\times$  16 users = 1024 trials, not counting warm-up trials.

We **hypothesized** that the [MEVS] condition will result in greater error (in terms of ItemsCloser, defined in subsection 4.6) than each of the three other animated transition conditions.

<sup>1</sup>Calculation made with G\*Power, for a one-sample t-test with  $\alpha=0.5$ , which is roughly equivalent to a 95% CI for a within-subject difference in means that does not overlap zero – see bottom of Figure 12.

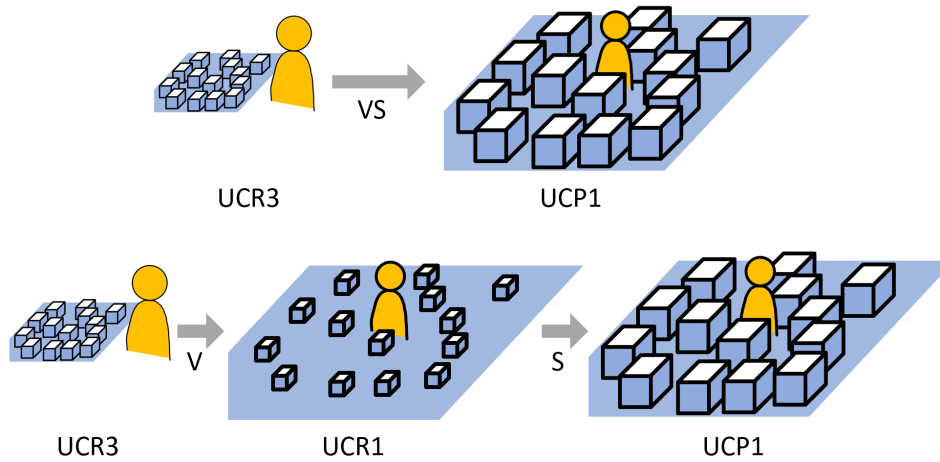


Fig. 11. In the [ME-VS], [M-E-VS], and [E-M-VS] conditions, one stage of the transition transformed View and Scale together (top), maintaining the relative positions and distances of elements, as if the entire scene was a single object being scaled up. In the [ME-V-S], by contrast, View and Scale were transformed in sequence (bottom), making elements appear more sparse in the middle of these transformations.

## 4.9 Results

**4.9.1 Planned Analysis.** For these results, advice was adapted from Dragicevic [14]. We present effect sizes visually with confidence intervals (CIs), following Tips 15 and 16 in [14]. Our CIs are computed using one (averaged) value for each (user, condition) pair (Tip 9). The use of null hypothesis significance testing (NHST) can cause misleading, dichotomous thinking [1, 11, 15, 31] and we decided to forego any NHST in our analysis (Tip 25 in [14]). In accordance with our pre-registration, we do not adjust for multiplicity: doing so is less necessary when reporting only a small number of planned comparisons (in our case, we have only three, all pre-registered), and when using an estimation approach to statistical inference (Tip 31 in [14]).

Figure 12 shows the result of our preregistered analysis. Of the three multi-stage transitions that were compared, we find evidence that [ME-VS] is better than [MEVS] in terms of ItemsCloser.

**4.9.2 Subjective Feedback and Exploratory Data Analysis.** Users were asked which condition they liked the most and the least, and were asked to answer four questions on a 7-point scale: “What mental effort was required?”, “What physical effort was required?”, “How much frustration did you feel?” (1 for “little”, 7 for “a lot”) and “How did you find the speed of the transition?” (1 for “too slow”, 7 for “too fast”). Figure 13 shows the results. In the pilot, the 7 conditions were grouped into sets of 1-, 2-, and 3-stage transitions, which users were asked to compare. In the final experiment, [ME-VS] was preferred by most users, and received the most favorable scores for mental effort and frustration.

7 out of the 16 users said that trials were more difficult when the target moved behind them, because they had to turn around quickly. 5/16 users said they easily lost track of the target when items crossed trajectories, creating occlusion.

However, 8/16 users remarked that when items re-arranged themselves in front of the user (i.e., with a 3rd person view), this helped them to track the correct target. This echoes DC3 and was more the case with [ME-VS] and [M-E-VS]. Of these two conditions, [ME-VS] only has 2 stages, and thus displays each stage more slowly than [M-E-VS]. This helps explain the success of [ME-VS].

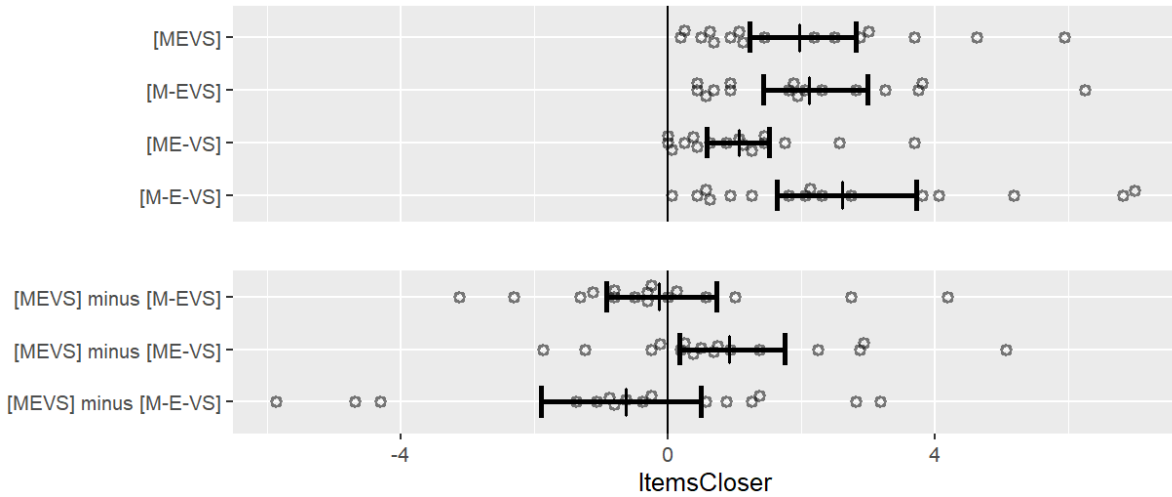


Fig. 12. ItemsCloser quantifies the size of errors (subsection 4.6). The upper part of the chart shows the four conditions, where each dot shows the average for one user, and each bar is a 95% CI computed with bootstrapping. The lower part of the chart shows paired differences, revealing evidence that [ME-VS] yields smaller errors than [MEVS].

	Conditions	Number of users		Likert score (1 to 7)							
		Most favorite	Least favorite	Mental effort	Physical effort	Speed	Frustration				
Pilot (4 users)	[MEVS]	0	1	4.0	3.8	1.5	2.3	3.0	3.0	2.0	2.5
	[M-EVS], [E-MVS], [ME-VS]	4	0	3.5	3.5	2.0	2.0	4.0	4.0	2.0	2.5
	[M-E-VS], [E-M-VS], [ME-V-S]	0	3	5.5	5.5	2.5	3.3	5.5	5.5	4.5	4.0
Final (16 users)	[MEVS]	3	3	3.5	3.6	1.0	1.8	4.0	4.4	2.5	3.0
	[M-EVS]	0	6	5.0	4.7	1.5	2.4	5.0	4.9	4.5	3.8
	[ME-VS]	10	0	2.0	3.0	1.0	1.9	4.0	3.9	1.0	1.4
	[M-E-VS]	3	7	5.0	4.8	2.0	2.3	5.0	5.1	3.5	3.8

Fig. 13. Subjective ratings by users. Blue shows counts, green shows averages, black shows medians, and grey shows interquartile ranges.

In Figures 14 and Figure 15 each dot shows the average for one user, and each bar is a 95% CI computed with bootstrapping.

Figure 12 suggests that [ME-VS] may be the overall best transition; however, Figure 14 hints that other transitions may be better if we consider each direction separately. (This suggests that future work could investigate asymmetrical transitions, where the kind of transition shown to the user depends on the direction of the transition.)

Figure 15 shows that, unsurprisingly, error rates generally increased when the target was moving at a higher speed, except for an apparent drop in error rate at the highest speeds achieved, which occurred with the only 3-stage transition, [M-E-VS].

8/16 users stated that trials were more difficult when the target traveled very close to them (these were trials where the target traveled through a nearly 180 degree angle relative to the user, and achieved the highest angular speed). Interestingly, one user stated that it was easier to track the target when it went through their body, despite its high speed, because this gave them a precise indication of the target’s trajectory and how to find it after

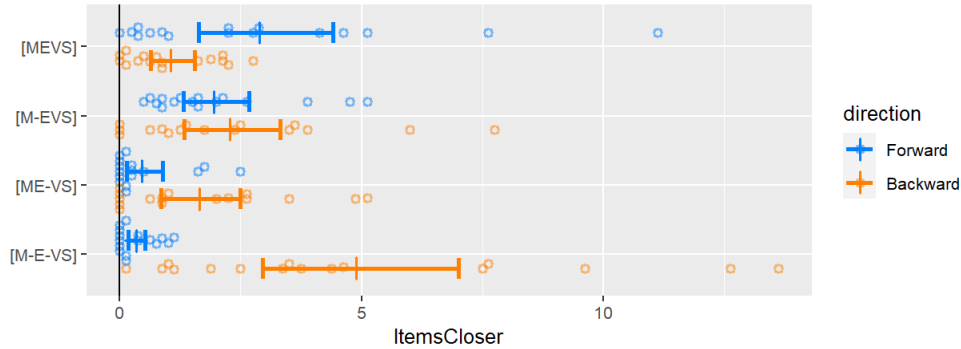


Fig. 14. ItemsCloser, broken down by direction.

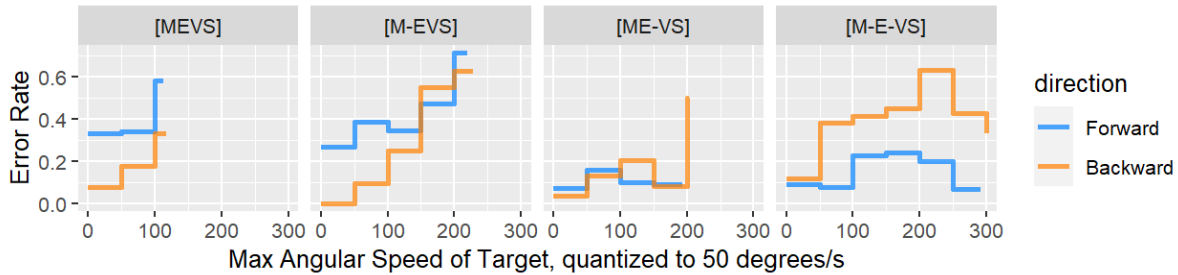


Fig. 15. Trials were binned by max angular speed of the target (a bin size of 50°/s was chosen using the Freedman-Diaconis rule). For each bin, the average error rate is shown on the vertical axis. The right extremity of each curve indicates the highest max angular speed achieved.

turning around. This may explain the drop in error rate observed at the highest angular speed in the [M-E-VS] condition in Figure 15.

4.9.3 Discussion. The apparent success of the [ME-VS] transition may be due to the importance of DC2 (i.e., not having 3 stages), DC3 (i.e., M and E transformations occurring in 3rd person), and/or DC9 (“Transform all as one”) for the experimental conditions that were tested, despite the conflict between DC9 and DC5 (“move smaller”) and not supporting DC10 (“Don’t move aggregations”).

Rather than eliminate or claim to have confirmed any particular design considerations, we propose that our design considerations must still be viewed as tentative and would require additional evaluations of various kinds to better understand. The rationale behind each consideration may nevertheless be useful for generating ideas to design future transitions.

We also propose some additional considerations that could be investigated in future work. The first came from reflecting on conflicts between our earlier considerations, and the last two emerged from imagining how to make a tracking task easier by using visual effects that were beyond the scope of our present study.

▷ **DC11 Make units distinct:** make units inside aggregations visually distinct. (This could be a way to combine the intended advantages of DC6 and DC10, by making individual units easier to track while still keeping the number of moving objects small.)

▷ **DC12 Use transparency:** make items and aggregations semi-transparent.

▷ **DC13** *Show tracks*: display trails or line segments showing where elements will move and/or where they came from after a translation. (This is comparable to ideas in 2D in [5, 6], and comparable to the visual feedback in [26].)

We observe several ways to group our tentative considerations according to the underlying motivation behind them: they simplify the complexity of transformations to make them easier to interpret (DC1), reduce the angular speed of translations and/or the need for large rotations of the head or eyes (DC2, DC3, DC8), reduce the density of items (DC7, DC6) simplify motions so they are easier to interpret (DC4, DC6, DC9), reduce inter-item occlusion (DC5, DC10, DC12), or otherwise facilitate tracking of individual items (DC10, DC11, DC13).

## 5 CONCLUSION AND FUTURE DIRECTIONS

We have presented a design space with four axes that characterize concrete and abstract visualizations. This enables us to identify a set of transformations necessary to design animated transitions between these two types of visualizations. Within the pool of possible transitions, we compared four of them in a controlled study in VR and found quantitative and qualitative evidence in favor of one particular transition.

Animated transitions are only one way to combine the use of abstract and concrete visualizations. Another strategy would be to display both kinds of visualizations side-by-side, possibly with visual links connecting them [41]. Many other design options were not tested. For instance, our scope puts aside questions associated with abstract marks morphing into more complex objects. Future research could focus on analyzing such transformations and how and when they could happen within an animated transition. This also pertains to how those parameters (e.g., shape, color, texture) could complement our design space.

Furthermore, our transitions could involve supplementary design elements. The display of marks trajectories' predictions [26] or traces in space may help track their back-and-forth movements. Additionally, playing with the transparency could prevent occlusion between marks during transitions. Potentially, people can miss portions of the animations when an object goes outside their field of view; future work can explore techniques similar to Phosphor [5] or Mnemonic Rendering [6] to show or replay what has been missed.

Furthermore, our work would benefit from further evaluation with higher level tasks. Our study involved a very clean virtual reality scene with a 1D or 2D visualization substrate, whereas more realistic embedded visualizations impose much more complex 3D substrates such as in Figure 1. To go further, it would be interesting to conduct a similar study in AR, utilizing the real-world environment. We have only touched the surface of a rich research area, defining concrete and abstract visualizations based on limited existing designs. The continuum between concrete and abstract visualizations could benefit from an extended definition, unlocking a broader range of creative possibilities, especially for concrete visualizations.

## ACKNOWLEDGMENTS

This research was supported by NSERC, and by the Agence Nationale de la Recherche (ANR) (grant number ANR-19-CE33-0012), and by the LaBRI mobility program.

## REFERENCES

- [1] Valentin Amrhein, Sander Greenland, and Blake McShane. 2019. Scientists rise up against statistical significance. *Nature* 567 (2019), 305–307.
- [2] Ambre Assor, Arnaud Prouzeau, Pierre Dragicevic, and Martin Hachet. 2023. Augmented Reality Waste Accumulation Visualizations. <https://hal.science/hal-03907474>.
- [3] Benjamin Bach, Nathalie Henry-Riche, Tim Dwyer, Tara Madhyastha, J-D Fekete, and Thomas Grabowski. 2015. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. *IEEE CG&A* 34, 3 (2015), 31–40.
- [4] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2013. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE TVCG* 20, 5 (2013), 740–754.
- [5] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: explaining transitions in the user interface using afterglow effects. In *ACM UIST*. 169–178.
- [6] Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. 2006. Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays. In *ACM UIST*. 159–168.
- [7] Zhutian Chen, Yijia Su, Yifang Wang, Qianwen Wang, Huamin Qu, and Yingcai Wu. 2019. MARVisT: Authoring glyph-based visualization in mobile augmented reality. *IEEE TVCG* 26, 8 (2019), 2645–2658. <https://doi.org/10.1109/TVCG.2019.2892415>
- [8] Fanny Chevalier, Pierre Dragicevic, and Steven Franconeri. 2014. The not-so-staggering effect of staggered animated transitions on visual tracking. *IEEE TVCG* 20, 12 (2014), 2241–2250.
- [9] Fanny Chevalier, Nathalie Henry Riche, Catherine Plaisant, Amira Chalbi, and Christophe Hurter. 2016. Animations 25 Years Later: New Roles and Opportunities. In *AVI*. 280–287. <https://doi.org/10.1145/2909132.2909255>
- [10] Fanny Chevalier, Romain Vuillemot, and Guia Gali. 2013. Using Concrete Scales: A Practical Framework for Effective Visual Depiction of Complex Measures. *IEEE TVCG* 19, 12 (2013), 2426–2435. <https://doi.org/10.1109/TVCG.2013.210>
- [11] Geoff Cumming. 2014. The new statistics: Why and how. *Psychological science* 25, 1 (2014), 7–29.
- [12] Kurtis Danyluk, Teoman Ulusoy, Wei Wei, and Wesley Willett. 2020. Touch and beyond: Comparing physical and virtual reality visualizations. *IEEE TVCG* 28, 4 (2020), 1930–1940. <https://doi.org/10.1109/TVCG.2020.3023336>
- [13] Nina Doerr, Benjamin Lee, Katarina Baricova, Dieter Schmalstieg, and Michael Sedlmair. 2024. Visual Highlighting for Situated Brushing and Linking. *arXiv preprint arXiv:2403.15321* (2024).
- [14] Pierre Dragicevic. 2016. Fair statistical communication in HCI. In *Modern Statistical Methods for HCI*. Springer, 291–330.
- [15] Pierre Dragicevic. 2017. Statistical Dances: Why No Statistical Analysis is Reliable and What To Do About It. <https://youtu.be/UKX9iN0p5fA>.
- [16] Pierre Dragicevic, Anastasia Bezerianos, Waqas Javed, Niklas Elmqvist, and Jean-Daniel Fekete. 2011. Temporal distortion for animated transitions. In *ACM CHI*. 2009–2018. <https://doi.org/10.1145/1978942.1979233>
- [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. In *ACM CHI*. <https://doi.org/10.1145/3411764.3446866>
- [18] Thibault Friedrich, Arnaud Prouzeau, and Michael J. McGuffin. 2021. The effect of increased body motion in virtual reality on a placement-retrieval task. In *ACM VRST*. 1–5. <https://doi.org/10.1145/3489849.3489888>
- [19] David Guilmaine, Christophe Viau, and Michael J McGuffin. 2012. Hierarchically animated transitions in visualizations of tree structures. In *AVI*. 514–521. <https://doi.org/10.1145/2254556.2254653>
- [20] Yi Guo, Nan Cao, Ligan Cai, Yanqiu Wu, Daniel Weiskopf, Danqing Shi, and Qing Chen. 2023. Datamator: An Intelligent Authoring Tool for Creating Datamations via Data Query Decomposition. <https://doi.org/10.48550/arXiv.2304.03126>
- [21] Steve Haroz, Robert Kosara, and Steven L Franconeri. 2015. ISOTYPE visualization: Working memory, performance, and engagement with pictographs. In *ACM CHI*. 1191–1200. <https://doi.org/10.1145/2702123.2702275>
- [22] Robert L Harris. 1999. *Information graphics: A comprehensive illustrated reference*. Oxford University Press, USA.
- [23] Jeffrey Heer and George Robertson. 2007. Animated transitions in statistical data graphics. *IEEE TVCG* 13, 6 (2007), 1240–1247. <https://doi.org/10.1109/TVCG.2007.70539>
- [24] Dolf Honee, William Hurst, and Antonius Johannus Luttikhoud. 2022. Harnessing Augmented Reality for Increasing the Awareness of Food Waste Amongst Dutch Consumers. *Augmented Human Research* 7, 1 (2022), 2. <https://doi.org/10.1007/s41133-022-00057-7>
- [25] Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. 2013. Visual Sedimentation. *IEEE TVCG* (2013). <https://doi.org/10.1109/TVCG.2013.227>
- [26] Yuta Itoh, Jason Orlosky, Kiyoshi Kiyokawa, and Gudrun Klinker. 2016. Laplacian vision: Augmenting motion prediction via optical see-through head-mounted displays. In *Proc. Augmented Human International Conference*. 1–8.
- [27] Alexander Ivanov, Kurtis Danyluk, Christian Jacob, and Wesley Willett. 2019. A Walk Among the Data. *IEEE CG&A* 39, 3 (2019), 19–28. <https://doi.org/10.1109/MCG.2019.2898941>

- [28] Benjamin Lee, Dave Brown, Bongshin Lee, Christophe Hurter, Steven Drucker, and Tim Dwyer. 2021. Data Visceralization: Enabling Deeper Understanding of Data Using Virtual Reality. *IEEE TVCG* 27, 2 (2021), 1095–1105. <https://doi.org/10.1109/TVCG.2020.3030435>
- [29] Benjamin Lee, Maxime Cordeil, Arnaud Prouzeau, Bernhard Jenny, and Tim Dwyer. 2022. A Design Space For Data Visualisation Transformations Between 2D And 3D In Mixed-Reality Environments. In *ACM CHI*. <https://doi.org/10.1145/3491102.3501859>
- [30] Benjamin Lee, Arvind Satyanarayan, Maxime Cordeil, Arnaud Prouzeau, Bernhard Jenny, and Tim Dwyer. 2023. Deimos: A Grammar of Dynamic Embodied Immersive Visualisation Morphs and Transitions. In *ACM CHI*. 1–18. <https://doi.org/10.1145/3544548.3580754>
- [31] Dong Kyu Lee. 2016. Alternatives to P value: confidence interval and effect size. *Korean Journal of Anesthesiology* 69, 6 (2016), 555–562.
- [32] Wenchao Li, Yun Wang, He Huang, Weiwei Cui, Haidong Zhang, Huamin Qu, and Dongmei Zhang. 2021. AniVis: Generating Animated Transitions Between Statistical Charts with a Tree Model. <https://doi.org/10.48550/arXiv.2106.14313>
- [33] Edward P McCormick, Christopher D Wickens, Rachel Banks, and Michelle Yeh. 1998. Frame of reference effects on scientific visualization subtasks. *Human Factors* 40, 3 (1998), 443–451.
- [34] Michael J McGuffin, Gord Davison, and Ravin Balakrishnan. 2004. Expand-ahead: a space-filling strategy for browsing trees. In *IEEE Symp. Information Visualization (InfoVis)*. 119–126. <https://doi.org/10.1109/INFVIS.2004.21>
- [35] Hauke S. Meyerhoff, Frank Papenmeier, and Markus Huff. 2017. Studying visual attention using the multiple object tracking paradigm: A tutorial review. *Attention, Perception, & Psychophysics* 79 (2017), 1255–1274. <https://doi.org/10.3758/s13414-017-1338-1>
- [36] Luiz Morais, Yvonne Jansen, Nazareno Andrade, and Pierre Dragicevic. 2020. Showing data about people: A design space of anthropographics. *IEEE TVCG* 28, 3 (2020), 1661–1679. <https://doi.org/10.1109/TVCG.2020.3023013>
- [37] Tamara Munzner. 2014. *Visualization analysis and design*. CRC Press.
- [38] Adam Nieman. 2012. Concrete vs abstract visualisation: the real world as a canvas for data visualisation. In *Proc. ADS-VIS2011: Making visible the invisible: Art, design and science in data visualisation*. 49–56. <http://eprints.hud.ac.uk/id/eprint/12775/>.
- [39] Deokgun Park, Steven M. Drucker, Roland Fernandez, and Niklas Elmqvist. 2018. Atom: A Grammar for Unit Visualizations. *IEEE TVCG* 24, 12 (2018), 3032–3043. <https://doi.org/10.1109/TVCG.2017.2785807>
- [40] Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson. 2002. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In *IEEE Symp. Information Visualization (InfoVis)*. 57–64. <https://doi.org/10.1109/INFVIS.2002.1173148>
- [41] Arnaud Prouzeau, Antoine Lhuillier, Barrett Ens, Daniel Weiskopf, and Tim Dwyer. 2019. Visual link routing in immersive visualisations. In *Proceedings of the 2019 ACM international conference on interactive surfaces and spaces*. 241–253.
- [42] Daniel Schwajda, Judith Friedl, Fabian Pointecker, Hans-Christian Jetter, and Christoph Anthes. 2023. Transforming graph data visualisations from 2D displays into augmented reality 3D space: A quantitative study. *Frontiers in Virtual Reality* 4 (2023). <https://doi.org/10.3389/frvir.2023.1155628>
- [43] Richard Skarbez, Nicholas F Polys, J Todd Ogle, Chris North, and Doug A Bowman. 2019. Immersive analytics: Theory and research agenda. *Frontiers in Robotics and AI* 6 (2019). <https://doi.org/10.3389/frobt.2019.00082>
- [44] Bruce H Thomas, Gregory F Welch, Pierre Dragicevic, Niklas Elmqvist, Pourang Irani, Yvonne Jansen, Dieter Schmalstieg, Aurélien Tabard, Neven AM ElSayed, Ross T Smith, et al. 2018. Situated Analytics. *Immersive analytics* 11190 (2018), 185–220.
- [45] Jorge Wagner, Wolfgang Stuerzlinger, and Luciana Nedel. 2022. The Effect of Exploration Mode and Frame of Reference in Immersive Analytics. *IEEE TVCG* 28, 9 (2022), 3252–3264. <https://doi.org/10.1109/TVCG.2021.3060666>
- [46] Christopher D Wickens and Tyler T Prevett. 1995. Exploring the dimensions of egocentricity in aircraft navigation displays. *Journal of Experimental Psychology: Applied* 1, 2 (1995), 110.
- [47] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. 2017. Embedded Data Representations. *IEEE TVCG* 23, 1 (2017), 461–470. <https://doi.org/10.1109/TVCG.2016.2598608>
- [48] Yalong Yang, Tim Dwyer, Kim Marriott, Bernhard Jenny, and Sarah Goodwin. 2020. Tilt map: Interactive transitions between choropleth map, prism map and bar chart in immersive environments. *IEEE TVCG* 27, 12 (2020), 4507–4519. <https://doi.org/10.1109/TVCG.2020.3004137>