This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

DashSpace: A Live Collaborative Platform for Immersive and Ubiquitous Analytics

Marcel Borowski, Peter W. S. Butcher, Janus Bager Kristensen, Jonas Oxenbøll Petersen, Panagiotis D. Ritsos, *Member, IEEE*, Clemens N. Klokmose, and Niklas Elmqvist, *Fellow, IEEE*

Abstract—We introduce DashSpace, a live collaborative immersive and ubiquitous analytics (IA/UA) platform designed for handheld and head-mounted Augmented/Extended Reality (AR/XR) implemented using WebXR and open standards. To bridge the gap between existing web-based visualizations and the immersive analytics setting, DashSpace supports visualizing both legacy D3 and Vega-Lite visualizations on 2D planes, and extruding Vega-Lite specifications into 2.5D. It also supports fully 3D visual representations using the Optomancy grammar. To facilitate authoring new visualizations in immersive XR, the platform provides a visual authoring mechanism where the user groups specification snippets to construct visualizations dynamically. The approach is fully persistent and collaborative, allowing multiple participants—whose presence is shown using 3D avatars and webcam feeds—to interact with the shared space synchronously, both co-located and remotely. We present three examples of DashSpace in action: immersive data analysis in 3D space, synchronous collaboration, and immersive data presentations.

Index Terms—Web-based technologies, collaborative visualization, Augmented Reality, eXtended Reality.

1 Introduction

Embracing immersive [1] and ubiquitous analytics [2] (IA/UA) means envisioning scenarios where heterogeneous devices seamlessly integrate into intelligent collaboration environments accessible through diverse modalities such as mobile screens, smartwatches, large displays, and head-mounted displays for XR (VR/AR/MR) [3]. Realizing this vision similarly entails embracing new dynamic, malleable, and shareable [4]–[7] development platforms capable of accommodating a wide array of input, output, computational, and network hardware and allowing for extending and modifying software while it is running. However, this is not the time for entirely clean slates and brand new beginnings: the engineering triumphs of the data visualization field are many—e.g., web-based 2D [8] and 3D [9], [10] toolkits and declarative grammars [11]—and should be preserved. Still, many IA [1] researchers are turning towards native game engines to leverage the latest XR hardware. We maintain that there are compelling reasons to uphold the use of open web technologies for IA/UA software, including rapid development, easy deployment, inherent interoperability, and robust standardization [12].

In this paper, we present our proposed solution to this challenge: DashSpace (Figure 1), a web-based live collaborative and composable platform for IA/UA based on a live reprogrammable software paradigm [5] and building on the Webstrates software stack [7], [13]. The fundamental design principle of DashSpace is to provide a standards-based open web development platform for building IA/UA software for use with both handheld and head-mounted XR displays (HMDs). For this reason, the DashSpace platform supports integrating existing 2D visualizations as well as novel 3D visualizations [14]. Such support can take several shapes: DashSpace supports both importing D3 [8] and Vega-Lite [11] visualizations as textures rendered on flat 2D planes in 3D space as well as extruding Vega-Lite [11] visualizations into 3D space using simple scene graph rewriting rules. Furthermore, the platform also natively supports the Optomancy 3D visualization grammar first introduced in the Wizualization [10] system. Finally, by virtue of being a pure web-based software system, DashSpace also has access to the full web ecosystem of composable software: in our implementation, we demonstrate several native web features, such as live reprogrammability in an integrated web development environment [5], native real-time collaboration and document persistence [7], live audio/video streaming using WebRTC to create remote presence avatars, as well as screen sharing. The platform is accessible using both regular desktop computers and handheld XR, such as on a smartphone or tablet, as well as head-mounted XR devices such as Meta Quest 3 or Apple Vision Pro.

Because authoring and editing visualizations is non-trivial in immersive environments [10], DashSpace presents a prototype visualization authoring mechanism that operates across toolkit boundaries. Each instantiation of a visualization in 3D space is an abstract container to which *components*—a Vega-Lite or Optomancy specification, D3 code, or a dataset—and *snippets*—fragments of a Vega-Lite or Optomancy specification—can be connected to. Depending on the display hardware, the user can either use hand tracking input, motion controllers, or touch actions to "open the hood" of a visualization (exposing the internal connections) and drag and drop data and snippets into a visualization container. Snippets can also be temporarily added to a visualization. The underlying source code is composed into specifications and can be edited live, collaboratively, and in asymmetric setups where some users are immersed while others use a conventional 2D editor.

We claim the following contributions in this paper:

1) The DashSpace platform, a live collaborative and composable platform for ubiquitous and immersive analytics built using

- *Marcel Borowski, Clemens N. Klokmose, and Niklas Elmqvist are with the Department of Computer Science at Aarhus University, Aarhus, Denmark. E-mail: {marcel.borowski, elm, clemens}@cs.au.dk*
- *Janus Bager Kristensen and Jonas Oxenbøll Petersen are with CAVI at Aarhus University, Aarhus, Denmark. E-mail: {jbk, jonas}@cavi.au.dk*
- *Peter W. S. Butcher and Panagiotis D. Ritsos are with the School of Computer Science and Engineering at Bangor University, Bangor, United Kingdom. E-mail: {p.butcher, p.ritsos}@bangor.ac.uk*

Manuscript received XXX XX, 2024; revised XXX XX, 2024.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

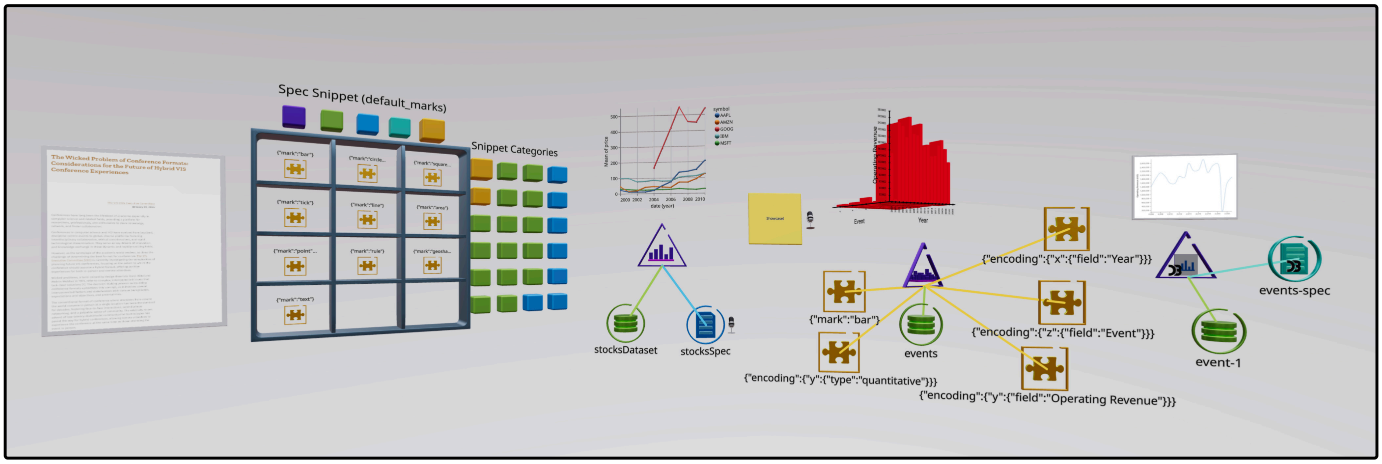


Fig. 1: **DashSpace.** An immersive analytics scene using DashSpace showing three visualizations. Components such as datasets and specifications are connected to visualizations to visualize and explore data—either by re-using existing specifications (here the 2D and 2.5D visualizations) or by building visualizations up using snippets (here the 3D visualization). Multiple platforms (Vega-Lite, Optomancy, D3) can be combined in the same workspace. A component and snippet library in form of a bookshelf (left) enables easy instantiation of objects. Additional media such as sticky notes (center) and images (far left) enable comprehensive analytics workflows.

open and standards-based web technologies;

2) Composition and rendering of cross-toolkit visualizations in D3, Vega-Lite, and Optomancy in full 3D space;
3) A prototype authoring mechanism using direct manipulation for creating visualizations in both 2D and immersive 3D; and
4) Several examples using DashSpace, such as collaborative data analysis, immersive storytelling, and situated analytics.

2 Motivating Scenario and Requirements

To illustrate the kind of ubiquitous and immersive analytics scenarios that we are targeting with DashSpace, we enumerate a three-part motivating scenario. We use this scenario to elucidate our design requirements and show how to realize them in Section 5. Each of the design requirements are derived from UA/IA literature.

2.1 Analyzing Data in Immersive 3D Space

Bo-Yuan is analyzing the financial health for a non-profit organization. The projection for the next fiscal year shows a large deficit, so he is planning to spend the next hour analyzing the past and current data to determine how to best cut the budget for the coming year. He is planning to use DashSpace and the immersive space in his office viewed in AR through his Meta Quest 3 ( ) for this analysis, allowing him to keep a connection with his physical surroundings and his co-workers. This task will require him to correlate multiple datasets and visualizations, and the complexity and scale will benefit from being laid out spatially. These media come from many different sources that must interoperate in DashSpace ( ): data files, specifications, images, and personal notes. Furthermore, while Bo-Yuan anticipates manually coding some of the specifications himself, he plans to directly author other visualizations using controllers and without leaving the immersive space itself ( ).

- **R1 – No-setup Immersive AR:** Handheld or head-mounted display (HMD) augmented reality with no environmental instrumentation or native app installation required [9], [12].
- **R2 – Broad Interoperability:** Integrating multiple sources, open standards, and media types [9], [10], [12].
- **R3 – Live and Flexible Visualization Authoring:** Live visualization authoring at varying tailorability [15]—customization, recombination, extension, and reprogramming—in both 2D editor and immersive 3D [9], [12], [14].

2.2 Collaborating Around Data

Later, Bo-Yuan calls in reinforcements to tackle the budgeting exercise. Naisha joins him in his office with her own headset to collaborate in a co-located ( ) and synchronous ( ) fashion. Since Naisha is most comfortable building charts in D3 [8] and Bo-Yuan is more of a Vega-Lite [11] person, they need a platform that can support multiple visualization toolkits ( ) even in 3D, not to mention toolkits that can utilize 3D to its fullest.

- **R4 – Co-located Collaboration:** Support for working together in the same physical space with the same 3D scene [16].
- **R5 – Synchronous Collaboration:** Support for working together in real-time with consistent updates from others [16].
- **R6 – Multi-toolkit Support:** Support for a range of existing visualization toolkits adapted to the 3D space [12].

2.3 Presenting the Analysis

Much later, Bo-Yuan and Naisha arrive at a plan of action that is apt to be unpopular but necessary for their organization. They must now present it to the executive committee during a remote meeting ( ). The committee members will join the meeting using a multitude of platforms ( ), including HMDs, handheld XR using smartphones, or regular computers or mobile devices. Finally, they will need a persistent 3D space ( ) so that they can prepare it in advance of the meeting and share the results afterwards.

- **R7 – Remote Collaboration:** Support for working together in geographically distributed places and communicating using voice, video, and virtual presence [16].
- **R8 – Multi-platform Support:** Access to the environment from varied hardware, including mobiles, HMDs, and PCs with a single code-base [9], [10], [12].
- **R9 – State Persistence:** Persistent and real-time synchronized environment sharable as a URL [9], [10], [12].

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

3 Related Work

We review the literature at the intersection of platforms for immersive and ubiquitous analytics, visualization grammars, visualization authoring, and collaborative analytics. Table 1 gives an overview of existing toolkits and platforms with a similar purpose.

3.1 Immersive and Ubiquitous Analytics Platforms

Ubiquitous analytics (UA) [2] leverages novel and post-WIMP [22] computing technologies—such as mobile devices, large displays, touch/pen/gesture interaction, and AR/VR/XR technologies—for effective data analytics and visualization; immersive [23] and situated [24] analytics (IA/SA) are subsets of this research area dealing with immersive technologies for data situated in a specific locale. Several platforms have been designed for these analytics flavors over the years [25], [26].

Early UA systems—such as Hugin [27], Munin [28], Shared Substance [29], and Branch-Explore-Merge [30]—were designed for tabletops, display walls, and mobile devices. Starting in 2014, many platforms adopted open web technologies; tools such as PolyChrome [31] and Webstrates [7]. Vistrates [4] is a component library for designing multi-device ubiquitous analytics applications built on top of Webstrates and Codestrates [13], [32] (a computational notebook interface); it uses a live and persistent document model that can be replicated across any number of devices. VisConnect [33] is a web-based collaborative visualization system for synchronous distributed interaction among multiple users that seamlessly integrates with existing web-based SVG visualizations and supports simultaneous interaction through peer-to-peer consensus and low-level event synchronization.

Some platforms have been designed specifically for immersive (or hybrid immersive) environments. ImAxes [19] is such an immersive system enabling modeless interaction with multivariate data through embodied data axes, which users manipulate like physical objects to create diverse visualizations dictated by their proximity and orientation. IBM's Dataspace [34] is a hybrid immersive environment combining high-resolution displays, robotic arms for dynamic reconfiguration, and AR/VR integration for collaborative data analysis. VRIA [9] is a web-based framework for creating immersive analytics experiences in XR, offering rapid development and platform independence using open-standards technologies. RagRug [21], which internally uses IATK, is designed specifically for situated analytics. It focuses on AR and combines visual encoding, physical-virtual models, and IoT dataflow integration in a low-code authoring system. The multi-display graphics environment SAGE [35] developed translational collaboration tools for scientific research over 20 years [20], evolving through multiple iterations to adapt to technological advances and user needs. SAGE3, the current version, integrates AI and a wide array of collaborative tools in a web-based 2D spatial workspace, supporting diverse platforms from high-resolution display walls to personal laptops. Finally, Wizualization [10] system is a web-based ubiquitous analytics platform for AR built using WebXR.

TABLE 1: **Existing IA/UA/SA platforms.** Comparing DashSpace with related platforms using our design requirements.

Platform	R1	R2	R3	R4	R5	R6	R7	R8	R9
DXR [17]	–	✓	✓	–	–	–	–	✓	–
IATK [18]	–	✓	✓	–	–	–	–	✓	–
ImAxes [19]	–	–	✓	–	–	–	–	–	–
SAGE3 [20]	–	✓	–	✓	✓	–	✓	✓	✓
VRIA [9]	✓	✓	✓	–	–	–	–	✓	–
RagRug [21]	–	✓	–	–	–	–	–	✓	–
Wizualization [10]	✓	✓	✓	✓	✓	–	–	–	✓
DashSpace	✓	✓	✓	✓	✓	✓	✓	✓	✓

Comparison. We built DashSpace on the Webstrates software stack [7], [13] using the Varv [5] live reprogrammable programming model. Compared to the above platforms, DashSpace presents several innovations, including integration with existing D3 [8] and Vega [36]/Vega-Lite [11] specifications in 3D, visualization authoring through direct manipulation, and WebXR integration with both handheld XR as well as Meta Quest 3 and Pro, and VR on the Apple Vision Pro. While SAGE3 has experimental XR integration [37], it is currently confined to integrating 2D content from the graphics environment into a VR scene.

3.2 Visualization Grammars

Visualization grammars, as initially conceptualized by Wilkinson in the *Grammar of Graphics* [38], offer a systematic framework for constructing visual representations through declarative syntax. This is a departure from the template-based approach found in tools such as Microsoft Excel as well as the shelf-based one in Tableau.

Several implementations of the concept exists. `ggplot2` [39] is an implementation of the grammar of graphics in the R language. Vega [36] emerged as a low-level specification utilizing JSON, subsequently paving the way for the development of Vega-Lite [11], a higher-level abstraction simplifying interactive visualization creation while shielding the complexities of Vega. Some visualization grammars serve as backend specifications for graphical interfaces. For instance, Tableau (*née* Polaris [40]) is powered by the VisQL grammar, while Lyra [41] and Lyra 2 [42] outputs Vega or Vega-Lite specifications.

Furthermore, specific visualization grammars tailored for immersive analytics, such as IATK [18] and DXR [17], enable users to author 3D visualizations in virtual and augmented reality environments. Both of these toolkits are built using the Unity proprietary game engine; while yielding high performance and a rich ecosystem of components, such native game engines lack interoperability and standardization, makes for complex deployment, and is poorly integrated in current business practice [12], where web-based development is dominant.

Comparison. Our work in this paper builds on the Optomancy declarative grammar for visualization, first introduced in the Wizualization system [10]. Just like the VRIA [9] grammar that preceded it, Optomancy adopts a Vega-like JSON specification and utilizes open web technologies for 2D and 3D immersive analytics.

3.3 Visually Authoring Visualizations

Several interactive visualization authoring tools have been developed to facilitate authoring custom data visualizations without the need for writing code. iVisDesigner [43] provides a web-based platform for creating interactive visualizations using drag-and-drop operations. Lyra [41] presents a similar interactive interface, exporting Vega [36] specifications as JSON code. These tools pioneered interactively binding visual attributes to dataset dimensions using direct manipulation.

Several research projects have since taken these ideas further. Data Illustrator [44] draws inspiration from vector editors and uses lazy data bindings to turn them into visualizations.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

Charticulator [45] transforms interactive chart specifications into mathematical layout constraints, enabling the creation of bespoke and reusable chart layouts without requiring users to choose from predefined options. Lyra 2 [42] introduces the idea of visualization interaction design by demonstration, allowing users to author interactive visualizations through direct manipulation mechanisms directly on the visualization being edited using contextual heuristics and by presenting several design alternatives.

Comparison. Our authoring approach in this paper is designed for immersive settings with hand tracking or motion controllers and no keyboard, relying instead on dragging snippets together to construct a visualization. We are inspired by prior work on chart authoring using gestures and voice [10], but our authoring paradigm uses name-value pairs for JSON-based grammars as building blocks.

3.4 Collaborative Immersive Analytics

Collaboration is key for many real-world information tasks [46], but remains a grand challenge for visualization research [16]. Many collaborative data visualization systems exist for personal computers [47]–[49], mobile devices [4], [30], and large displays [50], [51] exist. However, papers studying collaborative visualization in immersive 3D environments are more rare. Indeed, collaboration for IA is an open area of inquiry [52], [53].

While computer-mediated collaboration is challenging, particularly when it is remote, Yang et al. [54] suggests that collaboration in immersive analytics, facilitated through VR environments, can replicate the sense of physical presence and spatial organization experienced in face-to-face collaboration. In fact, these immersive environments can potentially enhance engagement, improve interaction with data objects, and promote equal participation among team members. Indeed, Cordeil et al. [55] compared the analysis of network connectivity in both HMD and CAVE-style immersive environments, revealing differences that highlight the potential of affordable HMDs for this purpose. Lee et al. [56] investigated how co-located participants arrange authoring tools in a shared VR workspace. Their findings indicate the need for appropriate interface designs that facilitate the collaborative analysis, challenges in view management due to the presence of 3D information, and interesting behaviors between users due to the spatial arrangement of information (e.g., regarding privacy and territoriality). Similarly, Saffo et al. [57] studied collaboration in Virtual Reality, proposing group awareness mechanisms across both VR and desktop platforms by providing varying degrees of view and control of the other party.

Recent studies have also tackled AR, where the physical presence of remote collaborators is necessarily diminished because of the real-world setting. XVCollab [58] leverages collaborative immersive analytics and cross-virtuality analytics (XVA) [59] to enable simultaneous data analysis by two groups using non-immersive desktop and immersive AR interfaces. Benk et al. [60] introduce an AR system for collaborative immersive analytics in machine learning (ML) modeling, evaluating collaboration dynamics among users with varied backgrounds and ML knowledge, and providing recommendations for designing immersive systems to foster sustained collaboration. Butscher et al. [61] studied the use of combination of AR and tabletop-surfaces for the purpose of collaborative analysis of multidimensional, abstract data, providing a set of guidelines for future systems.

Finally, a few toolkits and platforms have been designed specifically to support mixed-presence collaborative data analysis. A user study conducted as followup work to the IBM Dataspace [34] discussed above shows the impact of different levels of data immersion on exploratory data analysis (EDA) processes and compare the performance of their bespoke Immersive Insights system with a state-of-the-art, non-immersive data analysis system. The Wizualization [10] platform discussed above provides a native persistent workspace that supports rudimentary immersive collaboration, but this functionality is not explored.

Comparison. By virtue of being built on the Webstrates [7] software stack, DashSpace is intrinsically replicated, persistent, and collaborative. In this way, our work here is reminiscent of Badem et al.'s [4] past work on Vistrates; however, we adopt XR, visual authoring, and live reprogrammability as core aspects.

4 DashSpace

DashSpace is a platform for composable and collaborative IA/UA. It supports a wide range of devices, visualization types, and collaborative scenarios. Existing Vega-Lite, D3, and Optomancy specifications as well as JSON and CSV datasets can be imported. Furthermore, visualizations can also be authored using either a built-in code editor or an immersive visual authoring mechanism. DashSpace is motivated by the design requirements from Section 2.

4.1 Scene, Objects, and Menu

A DashSpace document centers around a shared 3D *scene* (Figure 2). The scene can be explored within a 2D browser window, in a handheld AR session on smartphones and tablets, or in an immersive AR or VR session on HMDs ( ). Navigation and interaction happens through mouse and keyboard using WASD controls for movement on desktop, touch interaction on mobile, and motion controllers or hand tracking on HMDs.

All *objects* within the scene are persisted and synchronized in real-time between all clients viewing a DashSpace document ( ). Each object has a *handle* that can either be a part of the object, e.g., the icon for a visualization or the whole object for an image. Objects can be moved by clicking and dragging their handle on desktop, tapping and dragging them on mobile, or selecting and dragging them using motion controllers or hands on HMDs.

Clicking, tapping, or selecting an object marks it as selected and colors its handle red. By default, only one object can be dragged at a time, but a multi-select mode can be activated that allows for selecting and moving multiple objects at a time. On desktop, using only the mouse moves objects around on a sphere around the camera. Grabbing and moving via WASD moves the object relative with the camera, e.g., moving the camera in z-direction moves the grabbed object the same distance.

On desktop and mobile, a GUI overlay adds a *menu* with additional functions like adding or removing objects or toggling a multi-select mode. On HMDs the menu is attached to the left controller/hand of the user and can be interacted with using the right controller/hand (Figure 2B).

4.2 Visualizations, Components, and Snippets

DashSpace employs three main categories of objects in the 3D scene: *visualizations*, *components*, and *snippets* (Figure 3). A *visualization* is an empty skeleton of a visualization that can be filled with components and snippets to create an actual visualization. Because DashSpace is designed for interoperability with both 2D web-based visualization toolkits and novel 3D ones ( ), it

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

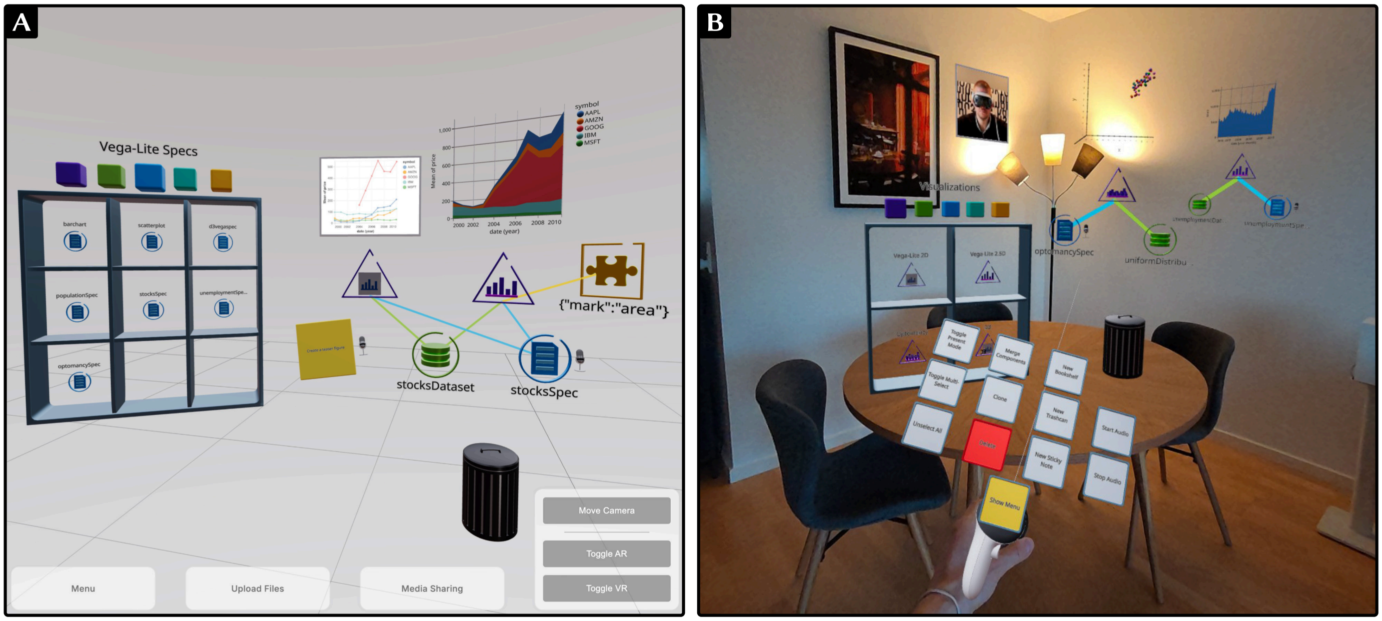


Fig. 2: **DashSpace on Desktop and HMD.** (A) The bookshelf showing available Vega-Lite specs in the library. A 2D and 2.5D visualization connected to the same dataset and spec; the latter is connected to a snippet that changes the mark type. A sticky note is visible in the center. (B) The bookshelf shows available visualization types as well as a 3D Optomancy and a 2.5D Vega-Lite visualization. The controller menu is visible in the foreground; in the background, a virtual image has been placed on the wall besides the physical one.

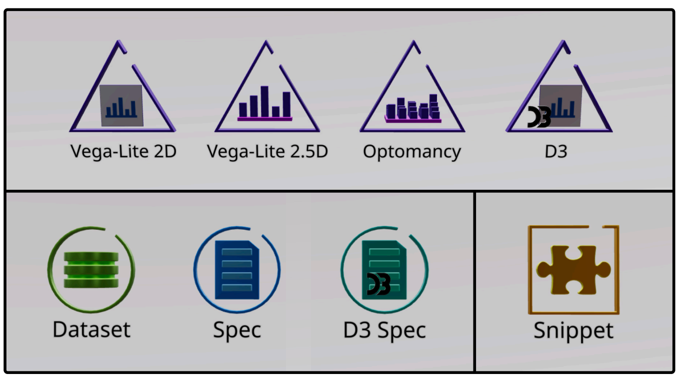


Fig. 3: **DashSpace 3D Icons.** Icons for different objects in a 3D scene. Visualizations (Vega-Lite 2D/2.5D, Optomancy, and D3) are surrounded by a triangle, components (datasets, specs, D3 specs) are surrounded by a circle, and snippets are surrounded by a square.

provides a mechanism for displaying 2D visualizations on a plane or extruding them into 3D ("2.5D"). Therefore, it supports four types of visualizations: Vega-Lite 2D, Vega-Lite 2.5D, Optomancy (3D), and D3 (2D) visualizations ( ). 2D visualizations are rendered on a 2D canvas, "2.5D visualizations are 2D visualizations that are extruded into 3D, and 3D visualizations use all three axes. Currently, rendered visualizations are not interactive, e.g., a user cannot select individual data points or pan a visualization.

Components are referenced code fragments that can be accessed using the Cauldron editor of Codestrates [13] (see below). A code fragment in Codestrates is a container of code, similar to a file, that can be viewed and edited in Cauldron. A component in the 3D scene stores a pointer to a code fragment that is part of the DashSpace document. By storing a pointer instead of the content, it is also possible to create multiple components in the scene that all point to the same code fragment and all react to changes of that code fragment. DashSpace currently supports three types of components: Vega-Lite/Optomancy[1] specifications (JSON), D3 specifications (JavaScript), and datasets (JSON).

A *snippet* is a single part of a Vega-Lite specification. For instance, the mark type of a specification (e.g., `{"mark":"bar"}`) or the encoding type for an axis are possible snippets (e.g., `{"encoding":{"x":{"type":"nominal"}}}`). In contrast to components, snippets are not pointing to a code fragment but self-contain their information in JSON format. Snippets are designed as transient building blocks of a visualization that can be mixed and matched, and discarded when no longer needed ( ). The content of a snippet is immutable and cannot be modified. Once a user is satisfied, they can merge snippets into a new specification that is backed by a code fragment.

4.3 Component and Snippet Libraries

Components and snippets are managed in a library. The available components in the library are directly mapped to the available code fragments in the document. Snippets, on the other hand, are dynamically generated and derived from existing Vega-Lite and datasets (see Section 6.3 for more detail). Components need to be prepared at the desktop by uploading specs or datasets. Snippets are derived from these uploaded components and, additionally, DashSpace provides a small standard library of snippets.[2]

In the 3D scene, components and snippets from the library can be accessed by, first, creating a *bookshelf* (Figure 2 left). The bookshelf enables instantiating visualizations, components, and snippets in the scene and groups these objects in categories. Categories are selected by using the cubes on top of the bookshelf.

1. Henceforth we refer to specifications that are used both by Vega-Lite and Optomancy simply as Vega-Lite specifications in the remainder of this work.

2. Ten mark types and four encoding types for each the x, y, and z-axis.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

If the "Snippets" category is selected, snippet categories can be selected using boxes on the right side of the bookshelf (Figure 1). To instantiate an object, the user can drag a copy out of the shelf using the same interaction that is used to move objects. An additional *trash can* can be added using the menu—objects that are moved close to the trash can will be removed.

To add components to the library, a user needs to add a code fragment to the document for the desired type of component. This can be done by using the upload menu, which allows to drag-and-drop files or to upload files using a file picker. The uploaded files are then added as code fragments into the document. Alternatively, a user can manually create a code fragment using the Cauldron editor (Figure 4). It is also possible to use the editor to manually add or delete code fragments and modify them. Modifying the content of a code fragment in this way live updates visualizations that use components pointing to this code fragment ( ).

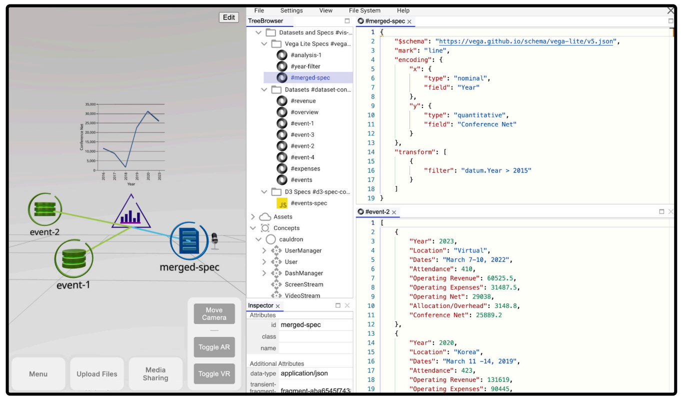


Fig. 4: **The Cauldron IDE.** The Cauldron editor from Codestrates v2 [13] enables editing and managing components from within DashSpace. The editor features a tree browser on the left of the editor and a tiled code editor on the right side.

4.4 Authoring Visualizations in AR

A typical workflow for authoring visualizations in AR with DashSpace, involves setting up a bookshelf and trash can to add and remove objects. A user can then drag objects out of the bookshelf—a simple example could include a visualization, a spec, and a dataset (Figure 1 left). When bringing components or snippets in proximity to visualizations, they will connect to the visualization, indicated using a line between them. All visualization types except for D3 visualizations can connect to datasets, specs, and snippets. D3 visualizations can only connect to datasets and D3 specs.

Once connected, the visualization will retrieve and merge the information of all connected components and snippets. In the case of regular specifications and snippets, the JSON specification will be merged. This works similar to CSS, where rules can be overwritten with new or more specific rules. In DashSpace, the specifications are ordered depending on their height/y-position in the scene and specifications or snippets higher up overwrite those lower in the scene. Datasets or D3 specifications are not merged and, instead, higher up components will simply overwrite lower ones. For instance, if two datasets are connected to the same visualizations, only the data from the one with a higher y-position will be used—allowing for simply reordering of components to switch between them ( ). We describe merging in Section 6.3.

4.5 Collaboration and Awareness

The location and state of all objects in the scene is synchronized in real-time between all clients visiting a DashSpace document. Moving objects on one client is directly visible in other clients, allowing, for instance, to shake objects to refer to them deictically.

Other clients are, furthermore, represented as *avatars* in the scene (Figure 7C). The position of avatars corresponds to the position and view direction of its client. Desktop users are represented as an action camera 3D model, mobile devices as a smartphone 3D model, and Quest headsets as a Quest 3 3D model (see the companion video). For headset users with motion controllers or hand tracking, DashSpace also adds avatars for the controllers as Quest 3 controller 3D models. These avatars—similar to shared cursors and pointers in other collaborative applications—help users gain awareness about what other users are doing ( ).

Lastly, DashSpace supports audio and video communication using WebRTC streaming. This can be activated either on the GUI or controller menu. While video feeds are not supported in most headsets due to the face of a user being covered by the headset, newer headsets like Apple Vision Pro can create synthetic video feeds of the faces of users by using a face scan together with face and eye tracking data from the headset. If a video feed is shared, it is rendered on top of the respective user's avatar in the scene. These features enable remote collaboration from within the platform ( ) without requiring additional video conferencing software.

4.6 Additional Features

Beyond the main visual analytics capabilities, DashSpace also supports additional features and convenience functions:

Calibration. When using XR, a *calibration marker* can be moved within the scene to calibrate the origin of the virtual scene relative to the physical world. This step is required due to a lack of persistent anchor support in current WebXR. Calibration is, on the one hand, useful when setting up a scene in, e.g., an office, and wanting to keep objects in the same location when returning to the scene later. It is also essential for co-located collaboration where the virtual objects are located in the same physical location for all users ( ).

Screen Sharing. DashSpace can share a live screen or window in the scene using WebRTC streaming. Screen casts are displayed as 2D planes in the scene and can be moved like other objects. This can be useful during remote collaboration ( ) or when working cross-device; using a headset for immersive AR but a desktop computer to edit specs with mouse and keyboard (Figure 5B).

Sticky Notes, Images, and Screenshots. DashSpace supports adding sticky notes and images as movable objects to the scene. Images can be uploaded and added to the scene using the menu and sticky notes can be added through the menu and edited using the Cauldron editor ( ). We also added a button in the GUI menu to take screenshots of the 3D scene. These object types are, however, only a first step and proof-of-concept towards richer multi-media support: video files and other media could be added in a similar way as objects to the scene in a future version of DashSpace.

Remote Controlling Cameras. The camera position and direction for each user is shared state in the platform, and by default transmitted to other clients to show avatars. However, this relationship can also be reversed: a headset user can remotely control the camera perspective of desktop users by grabbing their avatar—similar to grabbing other components. After letting go, the camera can be controlled again by the desktop user (see also Section 5.4).

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

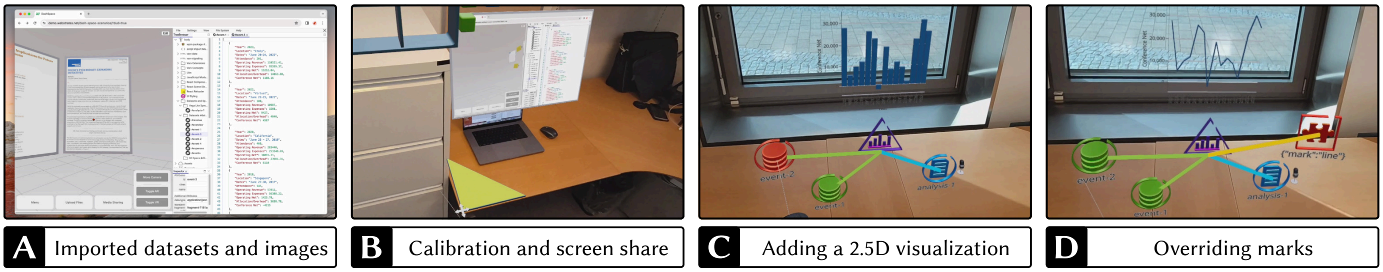


Fig. 5: **Immersive Analytics in the Office.** The scenario involves importing data in DashSpace via a web browser (A), setting-up the AR scene while wearing an HMD (B), situating a visualization in space (C) and manipulating it during exploration (D).

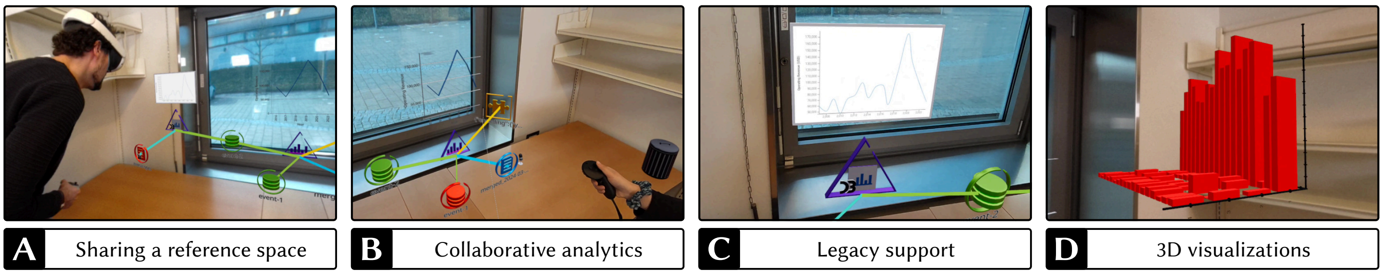


Fig. 6: **Live Collaborative Immersive Analytics.** Bo-Yuan observes the data while watched by Naisha (A). Naisha then makes changes in the scene (B) while being observed by Bo-Yuan. Naisha then adds a D3 visualization by dragging and adding a D3 spec she had from previous analysis (C). Bo-Yuan then adds an Optomancy visualization, using additional snippets to convert it into a 3D bar chart (D).

Currently, one camera can be controlled at a time, but this could be extended to control the cameras of all desktop participants at the same time, e.g., during a presentation.

Voice Input Modality and Specification Assistant. Authoring text in AR using a virtual keyboard is inconvenient. To overcome this, we added a proof-of-concept voice input feature to sticky notes and Vega-Lite specification. It can be activated by clicking on the 3D microphone icon besides notes or components. It records five seconds of audio and transcribes them using the Whisper[3] speech-to-text model. For a sticky note, the text on the note us replaced with the transcribed text (Section 5.4). For a specification, the transcript alongside the current specification in JSON is sent to the GPT-4 Turbo[4] large language model (LLM). The LLM returns an updated specification applying the prompted task.

In our tests, simple changes such as swapping the x and y-axis and changing mark types or colors work most of the time, whereas more complex queries may result in invalid specifications. More exploration and testing is necessary to make this a reliable feature.

5 Realizing the Scenario

We demonstrate how DashSpace enables immersive analytics in the following three-part usage scenario[5] that realizes the motivating scenario (Section 2). First, we highlight its use by a single analyst in an office environment. Second, we demonstrate how building on web technologies enables live co-located collaboration. Third, we showcase how DashSpace can be used to remotely present an analysis through a combination of devices.

3. https://platform.openai.com/docs/guides/speech-to-text
4. https://platform.openai.com/docs/guides/text-generation
5. We demonstrate all three parts of the scenario in the accompanying video.

5.1 Immersive Analytics in the Office

Bo-Yuan is analyzing the financial health data of a non-profit organization. He wants to analyze historical finances to determine how to best cut the budget for next year to compensate for deficits.

Importing Data on the Desktop. To get started, Bo-Yuan creates a new DashSpace document in the web browser on his laptop. Using drag-and-drop, he uploads datasets of the past year's expenses and income and a Vega-Lite specification he used in the past to analyze these datasets. The uploaded files are displayed in the Cauldron editor and added to the component library (Figure 5A, right). Additionally, he adds images of some announcements of the last year from the blog of the non-profit (Figure 5A, left), and some sticky notes with potential directions to explore. He also adds a screen share from his desktop screen to the document (Figure 5B). DashSpace provides interoperability ( ) by enabling him to reuse his existing files.

Setting Up the Scene in AR. Bo-Yuan now puts on his Meta Quest 3 and opens the same document in the web browser of the HMD. He starts the AR session and can see the scene he prepared on his computer, including some floating images and his notes. He quickly calibrates[6] the scene to his office (Figure 5B), and arranges the virtual notes and images on his whiteboard. He did not have to install any application or instrument his office ( ); DashSpace and its dependencies are automatically delivered over the web.

He also adds a bookshelf and a trash can using the menu on his left controller. He places the trash can on a table and the bookshelf opposite the whiteboard.

Flexible Exploration of Visualizations. Bo-Yuan now adds components to the scene. He starts by dragging a Vega-Lite 2.5D visualization, the financial data of two events of the non-profit, and the specification he uploaded earlier from the bookshelf into the

6. Not required once persistent markers are stable in WebXR.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

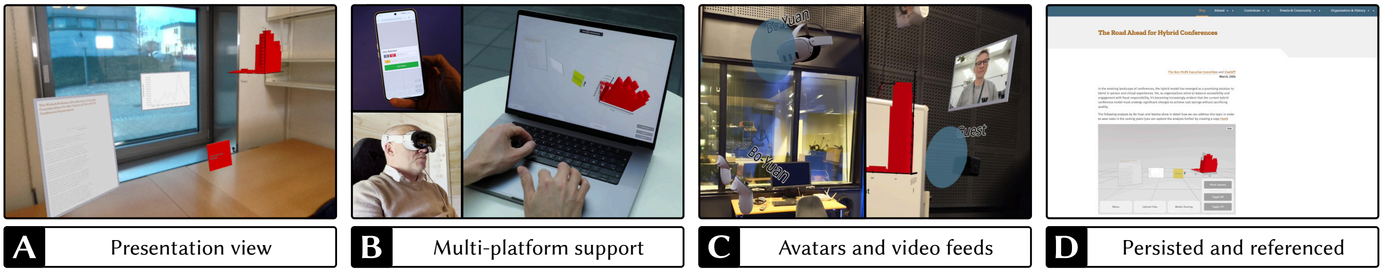


Fig. 7: **Shared Immersive Presentations.** Naisha and Bo-Yuan now switch to presentation mode, hiding unnecessary components (A). They share the URL of the presentation, allowing others to join from their different devices (B). Avatars (with video feeds) are shown in the scene (C). Finally, participants prepare a blog post where the DashSpace document is embedded as read-only (D).

scene. By bringing them close together, a floating visualization appears over the visualization icon (Figure 5C).

His initial specification uses bars; however, he realizes that a line chart would illustrate the data better. He adds a snippet with the mark type "line" to the scene and places it above the specification to override the mark type (Figure 5D). By moving the datasets of different events closer and further away from the visualization he compares the data of different events ( ). This supports seamless and fluid [62] data exploration and using direct manipulation.

He decides to filter for a certain set of years, and moves back to his computer where he uses the mouse and keyboard together with the shared screen to add a new component with the Vega-Lite filter. By bringing it close to the two visualizations, it filters the data in both. He spots a mistake in the visualization and quickly fixes it in the code, which live updates the visualization in the scene ( ).

5.2 Live Collaborative Immersive Analytics

Content with his initial exploration, Bo-Yuan now feels he needs reinforcements and asks his colleague Naisha to help him analyze the data. She comes to his office and brings her Meta Quest Pro.

Shared Reference Space. Naisha starts by putting on her HMD and opening the same document that Bo-Yuan is working on in her web browser. After starting up the AR session, she moves the calibration marker to synchronize the coordinate systems of their 3D scenes ( )—she can now see all objects in the location Bo-Yuan placed them in his office (Figure 6A).

Collaborative Analytics in AR. Bo-Yuan walks Naisha through his virtual notes and then shows her his current visualizations. Naisha moves some of the datasets around to see how the graph looks for different venues (Figure 6B). While moving components in the scene, both of them directly see changes live ( ).

Re-using Legacy Visualizations and Expanding Dimensions. Naisha previously used D3 to analyze an earlier version the event's budget. She adds the D3 spec, which she uploaded from her own laptop before the meeting ( ), together with a corresponding visualization and connects it to the same dataset (Figure 6C).

Directly after Naisha adds this visualization, Bo-Yuan adds ( ) an Optomancy visualization ( ). This lets him visualize the data of all events in a 3D bar chart (Figure 6D).

5.3 Shared Immersive Analytics Presentations

Bo-Yuan and Naisha come up with a plan on how to cut costs. They must now remotely present the plan to the executive committee.

Preparing the Scene. Bo-Yuan and Naisha clean up the scene by deleting no longer needed components. They toggle the presentation mode of the document to hide the components like specs and datasets and only show the final visualizations (Figure 7A).

Presenting the Results to the Committee. Bo-Yuan shares the URL to the document with the three remaining executive committee members of the non-profit ( ). One of the members is in his office and joins from his laptop, being able to explore the scene using mouse and keyboard. The second member is currently traveling and joins from his Android phone while being in a studio, using handheld AR to use their phone as a porthole into the 3D scene. The third member recently bought a Apple Vision Pro and joins using VR (Figure 7B). By virtue of being built on open web standards, the prototype works the same on all platforms ( ).

Once joined, Bo-Yuan shares his audio feed in DashSpace so that the participants can hear him. A floating avatar is shown for each participant in the scene (Figure 7C), allowing Bo-Yuan to ensure that the audience is paying attention during critical parts of the presentation. One of the committee members also activates their video feed, which shows up on top of their virtual avatars ( ). Bo-Yuan presents the results to the committee while the latter follow along in the scene, being able to see the same visualizations as him.

Sharing the Results with the Community. The meeting concludes with a direction for cost-cutting measures for the next year. The committee prepares a blog post to share their decision with the community. In the blog post, they embed a read-only copy of the DashSpace document in presentation mode (Figure 7D) and a link to a full version of the analysis, allowing readers to explore the data that leads to the conclusion themselves ( ).

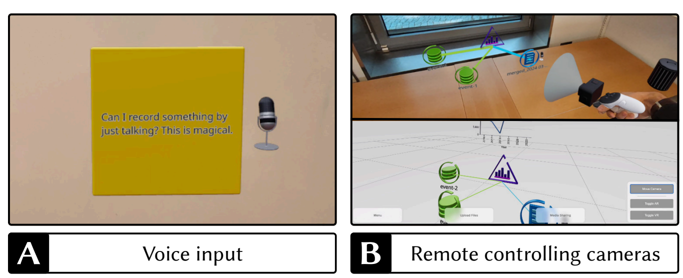


Fig. 8: **Voice Input and Remote Control of Cameras.** (A) The microphone icon allows recording audio and transcribing into the sticky note. (B) The headset user (top) uses their controller to grab and move the camera perspective of a desktop user (bottom).

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

5.4 Additional Examples

To illustrate the extensibility and flexibility of DashSpace, we implemented additional examples that show the IA/UA potential of such a web-based and collaborative platform.

5.4.1 Voice Input and Specification Assistant

While working in AR using his headset, Bo-Yuan comes up with a new idea on what to investigate. He uses the controller menu to add a new empty sticky note and selects the microphone icon to transcribe his idea on the note (Figure 8A).

5.4.2 Remote Controlling Cameras

During a collaborative session, Bo-Yuan is working in AR on his headset while Naisha joins him on her desktop. He wants to explain a certain idea to her by guiding her through his analysis. He is frequently moving between different parts of the analysis. To make sure Naisha follows along, he can grab the avatar of her camera and point it at the parts of the analysis he is referring to (Figure 8B).

5.4.3 Offline Use

Naisha is on an airplane flying to an in-person meeting where she has to present the results she and Bo-Yuan found. She needs to change the analysis to accommodate the audience, so she opens up her laptop and edits the scene accordingly. Because DashSpace works in a local-first manner (Section 6), she does not need an internet connection to do this. As soon as she lands, she uses her phone to push the changes online using a sync server. Any syntactic (low-level) conflicts are handled automatically. More high-level semantic conflicts must be handled by the user; mechanisms to support this are left for future work.

6 Implementation

DashSpace[7] is built on the Webstrates software stack consisting of Webstrates [7], Codestrates [13], and Varv [5]. It uses React Three Fiber for rendering 3D scenes and XR capabilities, and Vega-Lite [11], D3 [8], and Optomancy [10] for visualization.

6.1 Shared Web Platform and State

Webstrates [7] is a backend framework for websites with a persistent and replicated DOM (Document Object Model). This includes JavaScript and CSS, which is stored in the DOM and allows for creating web applications from within the web browser. Codestrates [13] adds a development platform to Webstrates that allows editing and running polyglot code using the built-in Cauldron IDE directly in the browser. Varv [5], lastly, adds a declarative programming model, which enables live and collaborative reprogramming of interactive behavior within the platform.

In DashSpace, we use Webstrates as a synchronization layer for code and content (e.g., storing and synchronizing Vega-Lite specs and locations of components). We also use the WebRTC streaming API from Webstrates to share audio, video, and screen streams between clients. Codestrates is used to edit and run code in the browser. In particular, we make use of the *code fragments*—computation units similar to files that are stored in the DOM—to author specifications and datasets imported to DashSpace from within the system. Varv is mainly used for state and interaction.

Varv originally only provided a DOM-view to render applications using HTML templates. For DashSpace, we added a React

7. DashSpace on GitHub: https://github.com/Webstrates/DashSpace

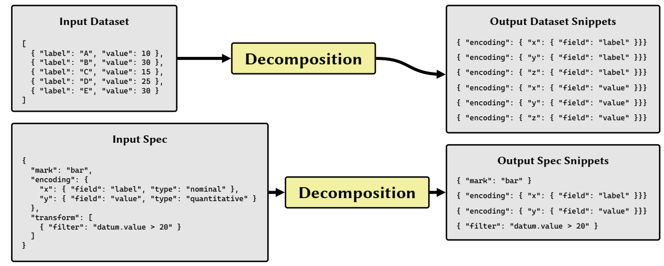


Fig. 9: **Input Decomposition.** Each dataset component (top) yields an encoding snippet for each field and axis, and each specification (bottom) yields snippets for marks, encodings, and filters.

view that improves performance and integration with React Three Fiber, enabling reading and writing Varv properties from React.

Offline editing is possible when running DashSpace on the next generation local-first [63] implementation of Webstrates [64] that relies on a combination of the service worker API and conflict-free replicated data-types (CRDTs) for peer-to-peer synchronization.

6.2 Live Scene Updates with React Three Fiber

The 3D scene is created using React Three Fiber (R3F),[8] a React renderer for Three.js. Persisted state in DashSpace is stored in Varv and passed through the React-view to R3F, which renders the scene and updates live on changes to the Varv state. By using R3F over Three.js, we were able to build on the Varv view architecture to create a declarative specification of the scene that is instantiated with Varv state and to make use of React's optimization features for updates to the scene. Overall, this allows the scene to update live whenever the Varv state or code fragments change.

R3F also includes an XR component[9] that adds hooks to integrate WebXR into an R3F app. We use its functionality to start AR/VR sessions as well as for controller and hand tracking.

6.3 Decomposition and Merging of Specifications

Decomposition is used to generate the snippet library by inferring information from imported specs and datasets. Merging generates visualizations by combining specs and snippets in the scene.

Decomposition. Snippets are generated from datasets and specs. For each dataset, DashSpace generates an encoding field snippet (e.g., `{"encoding":{"y":{"field":"Year"}}}`) for each field and each axis. E.g., a dataset with three fields generates nine snippets. In the case of specs, DashSpace extracts snippets for mark types, encoding fields, and filters (Figure 9). DashSpace also adds all mark types and encodings for all three axes.

Merging. The merging algorithm receives an array of specs or snippets. It starts off with an empty specification and merges snippets and specs into this spec based on their order. New specs are merged into the existing spec by overriding keys in the top level of the object, e.g., overriding all encodings at once. How snippets are merged into the spec depends on their type: mark type snippets override the existing mark, encoding type or field snippets override only the encoding type or field of the given axis, and transform snippets (currently only filters) are appended to the transform array.

8. https://github.com/pmndrs/react-three-fiber
9. https://github.com/pmndrs/xr

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

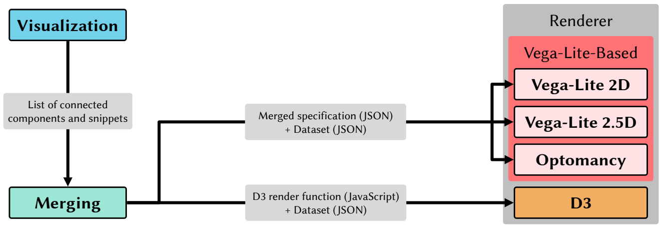


Fig. 10: **Visualization Pipeline.** Conceptual overview of the DashSpace visualization pipeline that merges specs and datasets, and generates one of several supported visualization types.

6.4 Live Visualization Generation

Visualizations in DashSpace are generated and updated live whenever components or snippets are connected/disconnected, and the data within code fragments of components changes. Upon each update, DashSpace collects the content of all connected components and snippets and sends it to the merging component, which returns the final specification in the case of Vega-Lite-based visualizations (Figure 10). In the case of D3 visualizations, DashSpace retrieves a function of the signature `renderVisualization(d3, data, canvas)` (`d3` refers to the D3 library, `data` to the dataset, and `canvas` to the HTML canvas to render to) and the dataset used for a visualization. These are then passed to the respective type of visualization for rendering.

Vega-Lite 2D and the D3 visualizations are rendered into a regular HTML canvas element, which in turn is rendered onto a texture on a 2D plane in the scene. Vega-Lite visualizations are rendered using `vega-embed`[10] and D3 visualizations by using the `renderVisualization` function passed from DashSpace. Optomancy visualizations are rendered using the Optomancy component [10], which is already using R3F for rendering and could be integrated directly into DashSpace. Vega-Lite "2.5D" visualizations, finally, are generated by, first, creating a regular 2D Vega-Lite visualization and then parsing its scene graph[11] to generate respective R3F 3D objects in the scene. For instance, axis lines are converted into thin boxes, and points to spheres. DashSpace's 2.5D visualizations support most mark types of the scene graph, being able to render bar, line, and area charts as well as scatterplots. However, for instance, images or symbols other than circles (for scatterplots) are currently not supported.

7 Discussion

DashSpace represents a robust step towards integrating IA/UA within a collaborative, dynamic environment built entirely on web technologies and open standards. By using the Webstrates software stack as a mature foundation, we think that DashSpace can become a new standard for IA/UA development in the same way that D3 [8] and Vega-Lite [11] are de facto standards for web-based visualization development and visualization grammars, respectively. In fact, beyond paving the way for novel 3D grammars such as Optomancy, DashSpace also provides compatibility for these very toolkits (e.g., D3 and Vega-Lite), enabling both existing and new visualization technologies and toolkits to coexist and complement each other. Here we review our design implications, acknowledge limitations, and outline potential avenues for future work.

10. https://github.com/vega/vega-embed
11. https://github.com/vega/vega/tree/main/packages/vega-scenegraph

7.1 Design Implications

The emergence of platforms such as DashSpace heralds significant implications for the fields of ubiquitous and immersive analytics. While there exist IA toolkits such as IATK [18] and DXR [17] as well as platforms such as RagRug [21] and Wizualization [10], these are all specialized tools that provide reuse at a code level. As a result, the field of IA/UA has long been been characterized by bespoke, one-off software solutions with little synergy from earlier such solutions. Live and collaborative web-based platforms such as DashSpace provide a solid platform for new innovation in IA/UA where tools do not have to be built from scratch time and again, but can leverage an existing ecosystem of IA/UA components.

We believe that DashSpace suggests a future where IA/UA research is increasingly focused on creating adaptable, scalable, and reusable frameworks that can accommodate a wide range of analytical tasks and data types. Rather than relying on the whims of developers of proprietary game engines, these platforms will be built on open, mature, and well-maintained standards [12]. Interoperability between applications will become paramount, enabling seamless integration and exchange of data and visualizations, enriching analytical environments, and fostering collaborative innovation. And by leveraging the already vast potential of the existing web ecosystem, we will make possible ideas and innovations that we can scarcely envision for now.

7.2 Systems-Oriented Evaluation

DashSpace is a user interface systems research contribution; accordingly, we evaluate it through demonstration [65]. We use the set of values for systems-oriented evaluation provided by Olsen [66] to assess DashSpace: it addresses a novel ubiquitous and immersive analytics situation (S), a set of new immersive analytics tasks (T), and potential analyst users (U) who can utilize this tool.

More specifically, DashSpace presents a solution to a *problem not previously solved*. We are not aware of an existing web-based IA/UA platform that allows for live and collaborative authoring across heterogeneous devices and that integrates with existing web-based visualization frameworks. DashSpace *reduces development viscosity* of web-based immersive and ubiquitous analytics by leveraging the Webstrates software stack, where changes to the running software can be applied instantaneously across devices, hence providing flexibility and rapid iteration.

A conventional Unity-based workflow for developing IA/UA requires lengthy compile and deploy procedures, lacks interoperability, and integrates poorly in document-centric workflows [12]. By using the web, DashSpace provides *power in combination* as existing web frameworks can be applied with minimal integration effort. For example, rendering a 2D Vega-Lite visualization in DashSpace is around 70 lines of code, and we integrated Webstrates' WebRTC support for communication with minimal effort. DashSpace enables *inductive combination* by allowing the seamless combination of, e.g., Vega-Lite and Optomancy-based grammars. These are compatible with any future visualization types added. Finally, DashSpace aims to *lower skill barriers* of developing IA/UA by relying on de facto web standards such as React and Three.js, which in turn has the potential to *empower new design participants* as web development skills are widespread [67].

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

7.3 Limitations

There are several limitations in our current DashSpace implementation. Our focus was not primarily on performance, and thus the platform is not fully optimized. This is particularly problematic for HMDs, where resource constraints are pronounced. We believe that integrating the WebGPU API may be a promising avenue for enhancing graphical performance, scalability, and user experience.

The current process of authoring visualizations in DashSpace utilizes a relatively primitive mechanism based on composing snippets of JSON specifications. While we found the approach surprisingly powerful when realizing our motivating examples, this activity also revealed a need for more sophisticated authoring tools that can fully leverage the capabilities of these visualization grammars. Moreover, the current approach supports interoperability even across grammars, but this is mostly due to the declarative nature of JSON grammars and the commonalities between Optomancy and Vega-Lite. In addition, it relies on authors having a solid grasp of these grammars, and is thus not suitable for novices.

Finally, there are limitations with the interactive aspects of the authoring process, primarily due to challenges with working in immersive 3D space. The system's requirement to representing snippets as small 3D icons that must be precisely arranged in space creates interaction challenges and can easily lead to visual clutter. In a seated configuration, the usable space is limited to the table surface, constraining users to only a few simultaneous visualizations. While room-scale setups offer more space, the practical limit of usable objects still depends heavily on room dimensions. Although combining snippets into specifications helps reduce scene complexity, the authoring mechanism needs further development—for example, allowing dataset, specification, and snippet objects to be embedded within visualization objects rather than requiring external connections. The presence of multiple user avatars can contribute to this visual congestion, similar to the confusion caused by numerous concurrent cursors in collaborative tools such as Google Docs. Our bookshelf implementation currently supports 25–30 specification or dataset slots before becoming unwieldy. While a pagination system could potentially address this limitation, such an approach might prove cumbersome when dealing with extensive collections of datasets or specifications.

Embracing principles from traditional visual programming, block-based programming models such as MIT's Scratch [68], or data flow pipelines akin to Vistrates [4] could significantly improve the intuitiveness and effectiveness of creating visualizations in XR.

7.4 Directions for Future Work

There are several avenues of future work suggested by the DashSpace platform. For one thing, since our contributions in this paper are of an engineering nature, we do not report on any empirical evaluation involving human participants using the platform. However, we believe that DashSpace will provide an excellent stepping stone for conducting user studies on novel visualization and interaction techniques made possible by the new platform. In particular, a comparative analysis of 2D versus 3D spatial arrangements for analytics, through controlled lab studies, could illuminate the distinct advantages and optimal application scenarios for each approach, thereby informing future designs. Furthermore, supporting more granular interactions—such as detailed manipulation of data points and integrated interactions across multiple visualizations (e.g., brushing and shared selection)—would substantially broaden its analytical capabilities.

8 Conclusion

We have presented DashSpace, a novel software platform for ubiquitous and immersive analytics that bridges the gap between traditional data visualization, mobile data analytics, and immersive augmented reality experiences. By virtue of being built on the Webstrates software stack as well as WebXR and open standards, DashSpace not only enhances collaborative and dynamic data exploration but also provides a foundation for future innovations in the field. We have demonstrated the utility and flexibility of the framework by presenting its use in three connected scenarios: individual immersive data analytics, co-located collaborative analytics, and distributed asymmetric data presentations.

Acknowledgments

This work was supported partly by Villum Investigator grant VL-54492 by Villum Fonden and the Aarhus University Research Foundation grant AUFF-E-2022-9-33. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, and B. H. Thomas, Eds., *Immersive Analytics*, ser. Lecture Notes in Computer Science. New York, NY, USA: Springer Publishing, 2018, vol. 11190.

[2] N. Elmqvist and P. Irani, "Ubiquitous Analytics: Interacting with Big Data Anywhere, Anytime," *Computer*, vol. 46, no. 4, pp. 86–89, 2013.

[3] N. Elmqvist, "Data analytics anywhere and everywhere," *Communications of the ACM*, vol. 66, no. 12, pp. 52–63, 2023.

[4] S. K. Badam, A. Mathisen, R. Rädle, C. N. Klokmose, and N. Elmqvist, "Vistrates: A component model for ubiquitous analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 586–596, 2019.

[5] M. Borowski, L. Murray, R. Bagge, J. B. Kristensen, A. Satyanarayan, and C. N. Klokmose, "Varv: Reprogrammable interactive software as a declarative data structure," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2022, pp. 492:1–492:20.

[6] A. C. Kay and A. Goldberg, "Personal dynamic media," *Computer*, vol. 10, no. 3, pp. 31–41, 1977.

[7] C. N. Klokmose, J. R. Eagan, S. Baader, W. Mackay, and M. Beaudouin-Lafon, "Webstrates: Shareable dynamic media," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2015, pp. 280–290.

[8] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[9] P. W. S. Butcher, N. W. John, and P. D. Ritsos, "VRIA: A web-based framework for creating immersive analytics experiences," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3213–3225, 2021.

[10] A. Batch, P. W. S. Butcher, P. D. Ritsos, and N. Elmqvist, "Wizualization: A "hard magic" visualization system for immersive and ubiquitous analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, 2024, 501–517.

[11] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2017.

[12] P. W. S. Butcher, A. Batch, D. Saffo, B. MacIntyre, N. Elmqvist, and P. D. Ritsos, "Is native naïve? Comparing native game engines and WebXR as immersive analytics development platforms," *IEEE Computer Graphics and Applications*, vol. 44, no. 3, pp. 91–98, 2024.

[13] M. Borowski, J. B. Kristensen, R. Bagge, and C. N. Klokmose, "Codestrates v2: A development platform for Webstrates," Aarhus University, Tech. Rep., 2021. [Online]. Available: https://pure.au.dk/portal/en/publications/codestrates-v2-a-development-platform-for-webstrates(66e1d4d9-27da-4f6b-85b3-19b0993caf22).html

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

[14] K. Marriott, J. Chen, M. Hlawatsch, T. Itoh, M. A. Nacenta, G. Reina, and W. Stuerzlinger, "Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation," in *Immersive Analytics*, ser. LNCS. Berlin, Germany: Springer, 2018, vol. 11190, pp. 25–55.

[15] J. E. Grønbæk, M. Borowski, E. Hoggan, W. E. Mackay, M. Beaudouin-Lafon, and C. N. Klokmose, "Mirrorverse: Live tailoring of video conferencing interfaces," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2023.

[16] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K. Ma, and H. Hagen, "Collaborative visualization: Definition, challenges, and research agenda," *Information Visualization*, vol. 10, no. 4, pp. 310–326, 2011.

[17] R. Sicat, J. Li, J. Choi, M. Cordeil, W.-K. Jeong, B. Bach, and H. Pfister, "DXR: a toolkit for building immersive data visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 715–725, 2019.

[18] M. Cordeil, A. Cunningham, B. Bach, C. Hurter, B. H. Thomas, K. Marriott, and T. Dwyer, "IATK: An immersive analytics toolkit," in *Proceedings of the IEEE Conference on Virtual Reality*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, pp. 200–209.

[19] M. Cordeil, A. Cunningham, T. Dwyer, B. H. Thomas, and K. Marriott, " ImAxes: Immersive axes as embodied affordances for interactive multivariate data visualisation," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2017, pp. 71–83.

[20] M. Belcaid, J. Leigh, R. Theriot, N. Kirshenbaum, R. S. Tabalba, M. L. Rogers, A. E. Johnson, M. D. Brown, L. Renambot, L. Long, A. Nishimoto, C. North, and J. Harden, "Reflecting on the scalable adaptive graphics environment team's 20-year translational research endeavor in digital collaboration tools," *Computing in Science & Engineering*, vol. 25, no. 2, pp. 50–56, 2023.

[21] P. Fleck, A. Sousa Calepso, S. Hubenschmid, M. Sedlmair, and D. Schmalstieg, "RagRug: A toolkit for situated analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 7, pp. 3281–3297, 2023.

[22] A. van Dam, "Post-WIMP user interfaces," *Communications of the ACM*, vol. 40, no. 2, pp. 63–67, 1997.

[23] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, F. Schreiber, and E. Wilson, "Immersive analytics," in *Proceedings of the International Symposium on Big Data Visual Analytics*. Piscataway, NJ, USA: IEEE, 2015, pp. 1–8.

[24] N. A. M. ElSayed, B. H. Thomas, K. Marriott, J. Piantadosi, and R. T. Smith, "Situated analytics: Demonstrating immersive analytical tools with augmented reality," *Journal of Visual Languages & Computing*, vol. 36, pp. 13–23, 2016.

[25] D. Saffo, S. Di Bartolomeo, T. Crnovrsanin, L. South, J. Raynor, C. Yildirim, and C. Dunne, "Unraveling the design space of immersive analytics: A systematic review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 495–506, 2024.

[26] S. Shin, A. Batch, P. W. S. Butcher, P. D. Ritsos, and N. Elmqvist, "The Reality of the Situation: A Survey of Situated Analytics," *IEEE Transactions on Visualization and Computer Graphics*, 2023, to appear.

[27] K. Kim, W. Javed, C. Williams, N. Elmqvist, and P. Irani, "Hugin: a framework for awareness and coordination in mixed-presence collaborative information visualization," in *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2010, pp. 231–240.

[28] S. K. Badam, E. R. Fisher, and N. Elmqvist, "Munin: A peer-to-peer middleware for ubiquitous analytics and visualization spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 2, pp. 215–228, 2015.

[29] T. Gjerlufsen, C. N. Klokmose, J. Eagan, C. Pillias, and M. Beaudouin-Lafon, "Shared substance: developing flexible multi-surface applications," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 3383–3392.

[30] W. McGrath, B. Bowman, D. C. McCallum, J. D. Hincapié-Ramos, N. Elmqvist, and P. Irani, "Branch-explore-merge: facilitating real-time revision control in collaborative visual exploration," in *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2012, pp. 235–244.

[31] S. K. Badam and N. Elmqvist, "PolyChrome: A cross-device framework for collaborative web visualization," in *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2014, pp. 109–118.

[32] R. Rädle, M. Nouwens, K. Antonsen, J. R. Eagan, and C. N. Klokmose, "Codestrates: Literate computing with Webstrates," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2017, pp. 715–725.

[33] M. Schwab, D. Saffo, Y. Zhang, S. Sinha, C. Nita-Rotaru, J. Tompkin, C. Dunne, and M. A. Borkin, "VisConnect: Distributed event synchronization for collaborative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 347–357, 2021.

[34] M. Cavallo, M. Dholakia, M. Havlena, K. Ocheltree, and M. Podlaseck, "Dataspace: A reconfigurable hybrid reality environment for collaborative information analysis," in *Proceedings of the IEEE Conference on Virtual Reality*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, pp. 145–153.

[35] B. Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. E. Johnson, and J. Leigh, "High-performance dynamic graphics streaming for scalable adaptive graphics environment," in *Proceedings of the ACM/IEEE Conference on High Performance Networking and Computing*. New York, NY, USA: ACM, 2006, p. 108.

[36] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, "Reactive Vega: A streaming dataflow architecture for declarative interactive visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 659–668, 2016.

[37] D. Kobayashi, N. Kirshenbaum, R. S. Tabalba, R. Theriot, and J. Leigh, "Translating the benefits of wide-band display environments into an XR space," in *Proceedings of the ACM Symposium on Spatial User Interaction*. New York, NY, USA: ACM, 2021, pp. 9:1–9:11.

[38] L. Wilkinson, *The Grammar of Graphics*. Berlin, Germany: Springer-Verlag, 1999.

[39] H. Wickham, "A layered grammar of graphics," *Journal of Computational and Graphical Statistics*, vol. 19, no. 1, pp. 3–28, 2010.

[40] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: A system for query, analysis, and visualization of multidimensional relational databases," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 52–65, 2002.

[41] A. Satyanarayan and J. Heer, "Lyra: An interactive visualization design environment," *Computer Graphics Forum*, vol. 33, no. 3, pp. 351–360, 2014.

[42] J. Zong, D. Barnwal, R. Neogy, and A. Satyanarayan, "Lyra 2: Designing interactive visualizations by demonstration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 304–314, 2021.

[43] D. Ren, T. Höllerer, and X. Yuan, "iVisDesigner: Expressive interactive design of information visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2092–2101, 2014.

[44] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. T. Stasko, "Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pp. 123:1–123:13.

[45] D. Ren, B. Lee, and M. Brehmer, "Charticulator: Interactive construction of bespoke chart layouts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 789–799, 2019.

[46] G. W. Hill, "Group versus individual performance: Are n+1 heads better than one?" *Psychological Bulletin*, vol. 91, no. 3, pp. 517–539, 1982.

[47] G. Mark, A. Kobsa, and V. M. González, "Do four eyes see better than two? Collaborative versus individual discovery in data visualization systems," in *Proceedings of the Conference on Information Visualization*. Los Alamitos, CA, USA: IEEE Computer Society, 2002, p. 249.

[48] G. Mark, K. Carpenter, and A. Kobsa, "Are there benefits in seeing double?: A study of collaborative information visualization," in *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, pp. 840–841.

[49] A. D. Balakrishnan, S. R. Fussell, and S. B. Kiesler, "Do visualizations improve synchronous remote collaboration?" in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1227–1236.

[50] S. K. Badam, F. Amini, N. Elmqvist, and P. Irani, "Supporting visual exploration for multiple users in large display environments," in *Proceedings of the IEEE Conference on Visual Analytics Science & Technology*. Los Alamitos, CA, USA: IEEE Computer Society, 2016, pp. 1–10.

[51] M. Klapperstück, T. Czauderna, C. Goncu, J. Glowacki, T. Dwyer, F. Schreiber, and K. Marriott, "ContextuWall: Multi-site collaboration using display walls," *Journal of Visual Languages & Computing*, vol. 46, pp. 35–42, 2018.

[52] M. Billinghurst, M. Cordeil, A. Bezerianos, and T. Margolis, "Collaborative immersive analytics," in *Immersive Analytics*, ser. LNCS. New York, NY, USA: Springer Publishing, 2018, vol. 11190, pp. 221–257.

[53] B. Ens, B. Bach, M. Cordeil, U. Engelke, M. Serrano, W. Willett, A. Prouzeau, C. Anthes, W. Büschel, C. Dunne, T. Dwyer, J. Grubert, J. H. Haga, N. Kirshenbaum, D. Kobayashi, T. Lin, M. Olaosebikan,

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI: 10.1109/TVCG.2025.3537679

F. Pointecker, D. Saffo, N. Saquib, D. Schmalstieg, D. A. Szafir, M. Whitlock, and Y. Yang, "Grand challenges in immersive analytics," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2021, pp. 459:1–459:17.

[54] Y. Yang, T. Dwyer, M. Wybrow, B. Lee, M. Cordeil, M. Billinghurst, and B. H. Thomas, "Towards immersive collaborative sensemaking," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. ISS, pp. 722–746, 2022.

[55] M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriott, and B. H. Thomas, "Immersive collaborative analysis of network connectivity: CAVE-style or head-mounted display?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 441–450, 2017.

[56] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer, "Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1171–1181, 2021.

[57] D. Saffo, A. Batch, C. Dunne, and N. Elmqvist, "Through their eyes and in their shoes: Providing group awareness during collaboration across virtual reality and desktop platforms," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2023, pp. 383:1–383:15.

[58] M. R. Seraji and W. Stuerzlinger, "XVCollab: An immersive analytics tool for asymmetric collaboration across the virtuality spectrum," in *Adjunct Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*. Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 146–154.

[59] B. Fröhler, C. Anthes, F. Pointecker, J. Friedl, D. Schwajda, A. Riegler, S. Tripathi, C. Holzmann, M. Brunner, H. Jodlbauer, H.-C. Jetter, and C. Heinzl, "A survey on cross-virtuality analytics," *Computer Graphics Forum*, vol. 41, no. 1, pp. 465–494, 2022.

[60] M. Benk, R. Weibel, S. Feuerriegel, and A. Ferrario, ""Is It My Turn?": Assessing teamwork and taskwork in collaborative immersive analytics," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–23, 2022.

[61] S. Butscher, S. Hubenschmid, J. Müller, J. Fuchs, and H. Reiterer, "Clusters, trends, and outliers: How immersive technologies can facilitate the collaborative analysis of multidimensional data," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pp. 1–12.

[62] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly, "Fluid interaction for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 327–340, Oct. 2011.

[63] M. Kleppmann, A. Wiggins, P. Van Hardenberg, and M. McGranaghan, "Local-first software: You own your data, in spite of the cloud," in *Proceedings of the ACM Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. New York, NY, USA: ACM, 2019, pp. 154–178.

[64] C. N. Klokmose, J. R. Eagan, and P. van Hardenberg, "MyWebstrates: Webstrates as local-first software," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2024, pp. 42:1–42:12.

[65] D. Ledo, S. Houben, J. Vermeulen, N. Marquardt, L. Oehlberg, and S. Greenberg, "Evaluation strategies for HCI toolkit research," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, p. 36:1–36:17.

[66] D. R. Olsen, "Evaluating user interface systems research," in *Proceedings of the ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2007, pp. 251–258.

[67] Statista, "Most used programming languages among developers worldwide as of 2024," 2024, accessed: October 22, 2024. [Online]. Available: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

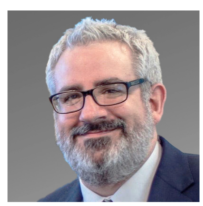
[68] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman *et al.*, "Scratch: Programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.

Marcel Borowski received the Ph.D. degree in computer science in 2022 from Aarhus University in Aarhus, Denmark. He is a Postdoctoral Research Fellow in Department of Computer Science at Aarhus University. His research interests include human-computer interaction, interactive systems, and computational media.

Peter W. S. Butcher received the Ph.D. degree in 2020 from the University of Chester, UK. He is a Lecturer in the School of Computer Science and Engineering, Bangor University, UK. His research interests include human-computer interaction, information visualization, and immersive analytics.

Janus Bager Kristensen is a software system architect at the Center for Advanced Visualization and Interaction (CAVI) in the School of Communication and Culture at Aarhus University.

Jonas Oxenbøll Petersen is a 3D interaction designer at the Center for Advanced Visualization and Interaction (CAVI) in the School of Communication and Culture at Aarhus University.



Panagiotis D. Ritsos received the Ph.D. degree in 2006 from the University of Essex in Colchester, UK. He is a Senior Lecturer in the School of Computer Science and Engineering, Bangor University, UK. His research interests include mixed and virtual reality, information visualization, and visual analytics. He is a member of the IEEE and the IEEE Computer Society.

Clemens N. Klokmose received the Ph.D. degree in computer science in 2009 from Aarhus University in Aarhus, Denmark. He is an Associate Professor in the Department of Computer Science at Aarhus University. His research interests include human-computer interaction, interactive systems, and computational media.

Niklas Elmqvist received the Ph.D. degree in 2006 from Chalmers University of Technology in Göteborg, Sweden. He is a Villum Investigator and professor in the Department of Computer Science at Aarhus University in Aarhus, Denmark. He was previously faculty at University of Maryland, College Park from 2014 to 2023, and at Purdue University from 2008 to 2014. His research interests include visualization, HCI, and human-centered AI. He is a Fellow of the IEEE, the IEEE Computer Society, and the ACM.