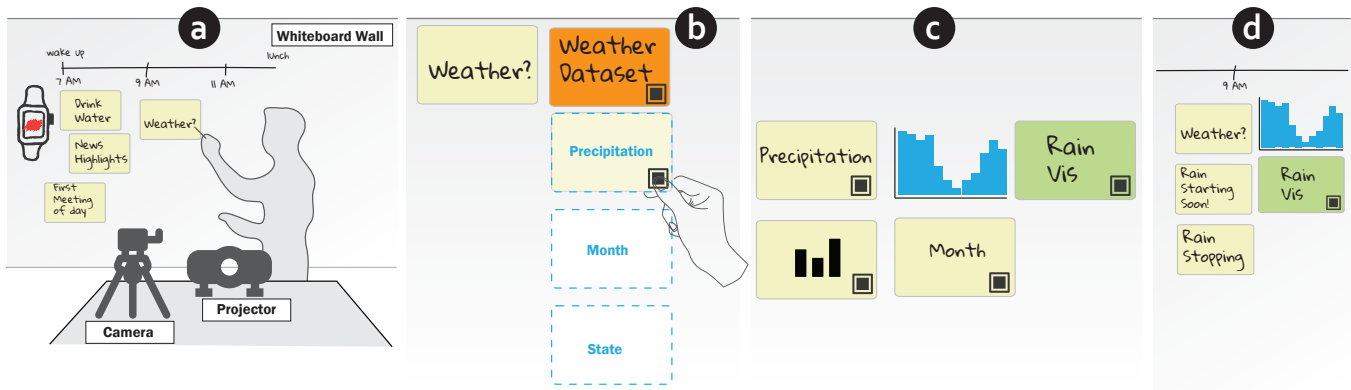


# Composites: A Tangible Interaction Paradigm for Visual Data Analysis in Design Practice

Hariharan Subramonyam  
Stanford University  
Stanford, CA, USA  
harihars@stanford.edu

Eytan Adar  
University of Michigan  
Ann Arbor, MI, USA  
eadar@umich.edu

Steven M. Drucker  
Microsoft Research  
Redmond, WA, USA  
sdrucker@microsoft.com



**Figure 1: A designer conducting data-driven brainstorming using Composites: (a) the designer creates sticky notes for potential information to show on a smart watch display; (b) To explore how to present weather data, the designer adds the weather dataset (orange) sticky note to Composites's work surface. In response, Composites projects available attributes in the dataset; (c) The designer constructs a bar chart with month on the  $x$ -axis and precipitation along the  $y$ -axis using Composites's spatial grammar, and the resulting visualization is projected onto the wall. (d) the designer captures the visualization in a new (green) sticky note and moves it to the cluster of brainstorming notes.**

## ABSTRACT

Conventional tools for visual analytics emphasize a linear production workflow and lack organic “work surfaces.” A better surface would simultaneously support collaborative visualization construction, data and design exploration, and reasoning. To facilitate data-driven design within existing design tools such as card sorting, we introduce *Composites*, a tangible, augmented reality interface for constructing visualizations on large surfaces. In response to the placement of physical sticky-notes, *Composites* projects visualizations and data onto large surfaces. Our *spatial grammar* allows the designer to flexibly construct visualizations through the use of the notes. Similar to affinity-diagramming, the designer can “connect” the physical notes to data, operations, and visualizations which can then be re-arranged based on creative needs. We develop mechanisms (*sticky interactions*, *visual hinting*, etc.) to provide guiding feedback to the end-user. By leveraging low-cost technology,

*Composites* extends a working surface to support a broad range of workflows without limiting creative design thinking.

## CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools; Interaction paradigms; Mixed / augmented reality.**

## KEYWORDS

Visual Data Analysis, Data-Driven-Design, Augmented Reality

### ACM Reference Format:

Hariharan Subramonyam, Eytan Adar, and Steven M. Drucker. 2022. *Composites: A Tangible Interaction Paradigm for Visual Data Analysis in Design Practice*. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces (AVI 2022)*, June 6–10, 2022, Frascati, Rome, Italy. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3531073.3531091>

## 1 INTRODUCTION

Data insights are helpful in design thinking and creative problem-solving tasks (e.g., brainstorming design features for an ambient display, deliberating COVID-19 policy changes, or solving a cybercrime). However, generating and incorporating data insights in creative practices can be a source of friction. For instance, imagine designing an ambient smartwatch display (e.g., the Siri Watch face [2]). In a brainstorming session, the designer (or a software team) create “sticky notes” on a wall about various contextual information that the display can show during the course of a day

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

AVI 2022, June 6–10, 2022, Frascati, Rome, Italy

© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/10.1145/3531073.3531091>

(see Figure 1 a). To define rules about whether, when, and how to present the information (e.g., such as alerting the possibility of rain), the designer may want to know the frequency of rain in different geographic regions. Unfortunately, conventional visual data analysis tools (e.g., Tableau [47] or Jupyter Notebook [23]) don't smoothly integrate into a design workflow or work surface.

The graphical user interface (GUI) format of visual analysis tools contradicts the tangible and organic “work surfaces” typical in design practice (e.g., whiteboard, tangible sticky notes, etc.). A data-driven design approach will require pausing the design activity to switch work surfaces (e.g., to a desktop interface) and later transfer insights back to the design work surface. Conducting data analysis before brainstorming can be limiting as new possibilities, questions, and design choices may emerge during the *generative* design process. Furthermore, many existing visual analysis tools implement a linear workflow: one visualization leading to an insight leading to another visualization. While effective for rapid construction of individual visualizations, such tools make finding, modifying, combining, and grouping previous analytical units (e.g., visualizations) difficult. This is essential to higher-level sensemaking and reasoning about design. Large wall-size displays could potentially address these issues but are costly and inaccessible to a large number of users. Augmented reality (AR) environments such as projection-based or head-mounted displays are *relatively* inexpensive, but do not support the same range of interactions as their touch-screen counterparts, and gestural interactions may be problematic (e.g., *gorilla-arm effect* [18]). Ideally, designers should maintain their existing work surfaces while seamlessly executing data-related tasks.

In this work, we explore a tangible interaction paradigm, *Composites*, to support data work in the context of existing design practice (e.g., affinity diagramming, card sorting, whiteboard-based concept maps etc.). Though the system can be adapted to AR or touch-screen environments, we focus on a ‘low-cost’ version. *Composites* combines a projector, a camera, and low-cost sticky notes to implement a tangible user interface (TUI). To support analysis, *Composites* offers a ‘grammar’ for producing visualizations by placing sticky notes. As shown in Figure 1b, the designer can access the weather dataset within the design work surface by placing the orange dataset sticky note on the wall. By reading the ArUco marker [32] on the dataset note, *Composites* projects corresponding data attributes (columns) on the work surface. The designer can ‘bind’ tangible sticky notes to individual data attributes and construct visualizations using our spatial grammar. In Figure 1c, the designer creates a bar chart of precipitation ( $y$  – axis) across different months of the year ( $x$  – axis) and ‘binding’ the resulting visualization to a new green sticky note. Where the note is moved, the visualization follows (Figure 1d).

In building *Composites*, we specifically address the challenge of maintaining a large work surface that supports creative design tasks *and* the construction and modification of visualizations (Figure 1). The designer can create, duplicate, modify, name, move, and group visualizations by simply writing on, and moving, physical notes (as opposed to a pointer, touch, or mid-air interactions). Critically, *Composites* allows higher-order sensemaking and reasoning by taking advantage of spatial cognition and spatial memory. In data-driven design using *Composites*, the designer can seamlessly generate and test their hypotheses about the data, fold data insights into their design context, and deliberate about design choices using

data. Our approach is scalable, supports collaboration and sharing as visualizations persist throughout the design activity.

Our key contributions include (1) a novel approach to large-screen visual analytics using near-range, direct manipulation interactions on non-touchscreen displays, (2) a spatial visualization grammar for TUIs, and (3) a set of mixed-reality interactions to connect physical tokens with on-screen information.

## 2 RELATED WORK

In *Composites* we build on prior work in the domains of visual analytics, visual programming languages, and tangible interaction techniques. We are specifically interested in how those approaches support large-surface *views*, flexible *arrangement* of visualizations, and the integrated ability to *modify/create* visualizations.

### 2.1 Visualization Tools

Many commercial visualization and visual analytics tools (e.g., Tableau, Excel, etc.) are optimized for mouse and keyboard interactions but do not lend well to direct manipulation interactions on large displays (fixed tab-based views and layouts, window management problem [37] etc). Research systems [8, 20, 40, 51] have provided ways of extending work surfaces in various ways. For example, ExPlates [20] continuously creates new visualizations on a large virtual display. The analyst can pan and zoom to see the provenance of visualizations and to get a sense of the analytics pipeline. While the placement of visualizations is largely constrained in ExPlates to a grid, Sandbox [51] supports arbitrary placement of information (i.e., “put-that-there” spatial cognition), and uses process model templates for analytical tasks. However, because it supports clipping from numerous applications, functions for visual analytics are not built into the tool.

While pan-and-zoom interactions provide access to large virtual surfaces, large physical displays can better support multiple views (data, navigation, and knowledge views), support greater data dimensionality, show multiple simultaneous visualizations with more data-details, and also provide a larger workspace for collaboration and sensemaking [1]. Dataspace, for instance, couples 15 high resolution displays with AR and VR for visual data analysis [5]. Unfortunately, it is not possible to directly translate existing design and visualization tools into this mode of operation due to their dependency on mouse and keyboard. Furthermore, when interacting on a large screen display—beyond a certain “tipping-point”—is it desirable to trade virtual navigation for physical affordances. This requires new design considerations that leverage the extent of human abilities (whole-body interaction, spatial cognition etc.) rather than solve for display medium’s technological limitations [1].

We are specifically interested in large displays to support the comparison of multiple visualizations. Alternatives include small multiples, and snap-together visualizations (e.g., [34]) which provide simultaneous views for comparison purposes. Uplift is a system for data exploration between diverse stakeholders which combines 3D models with tangible interactions, AR, and mid-air data visualization to support collaborative tasks [9]. MADE-Axis [40] provides a wireless ‘composable’ controller with actuated sliders for visual analytics in mixed-reality interfaces. Interactivity (e.g., linked brushing), extend the ability to coordinate and compare across views.

Gleicher et al. propose a general taxonomy of visual designs for comparison that groups designs into three basic categories: juxtaposition (e.g., two time series side-by-side), superposition (e.g. multiple time series in the same plot), and explicit encodings (e.g., a time series showing the difference between the original two) [13]. Different language constructs in Composites can produce these views either by inferring likely intent or through explicit construction. Thus not only can we support arrangement of individual visualizations so they can be positioned near each other, but also intelligently combine grouped views to enhance comparisons.

## 2.2 Visual Programming Languages

In Composites we take a visual querying approach similar to past work using interactive, visual querying of relational data [31, 33, 35, 42]. However, traditional approaches in this space require mouse and keyboard interactions to construct the “query.” In contrast, in visual programming languages (VPLs) queries are constructed through alternative graphical metaphors. While such languages have not traditionally focused on visualization, we can adapt them. VPLs are inherently spatial, and the relative positioning of primitives corresponds to relationships between those primitives or groups of primitives [14]. In Composites we use a spatial syntax to author and embed visualizations. This reduction in the gap (spatial, temporal, and conceptual) between visualization construction UI and the actual visualization is beneficial as it provides tight integration between action and feedback [15]. With Composites we utilize a consistent language, centered around the movable sticky notes, for creation, modification, viewing, and arranging.

## 2.3 Interaction Techniques

Two lines of research in interaction techniques relate to Composites: how visualizations can be viewed, navigated, or manipulated by the end-user, and how the system supports complex tasks.

Interaction with large surfaces has often focused on navigation. Solutions have ranged from body-centric gestures for control of map visualizations [39] to the use of tangible props ([6, 21]) such as a cube with three perpendicular rods that can be pushed or pulled along the three axes in order to explore 3D visualizations [10]. Visual analytics, however, is about more than navigation which is why direct-manipulation techniques are more natural for interacting with visual information, in that they minimize the distance between intent and execution of the intent [7, 38, 43]. In TouchVis[7], Drucker et al. conducted a comparison study between traditional desktop interfaces and FLUID (touch) interfaces, and report that touch interactions were better suited for analytic tasks as they lay emphasis on the task than on the interface. Visfer implements a camera based solution for cross device visualization transfer [4]. Marvis combines head-mounted AR displays with mobile devices to support visual analytic tasks [26].

Direct manipulation techniques for interacting with visualizations outside of a standard desktop environment have included pen and touch [27], as well as custom [11, 36, 40], and non-custom tangible tokens [12, 19, 22, 24]. The former two (pen/touch and custom tokens) are often highly expensive. Custom tokens, in particular, call for fabrication and engineering effort to add functionality *prior* to use in analysis. This can be problematic for visual

data analysis as one cannot anticipate query and analysis needs beforehand. Non-custom tokens are often less powerful than pen and tokens with integrated electronics, but can nonetheless support sophisticated interactions. The DigiPost system uses physical post-it notes to issue commands for editing documents (e.g. rename, save, etc.) [22]. Post-Post-it offers a spatial-ideation system using novel VR interactions modeled after physical sticky notes [29]. Other work-surfaces such as Dynamic Land [48] and Affinity Lens [44] use paper based tangible interfaces coupled with an augmented reality display. While Affinity Lens is intended for data augmented affinity diagramming (a design task), it does not support visualization construction. Composites builds on these ideas to form a more complete visual analytics framework.

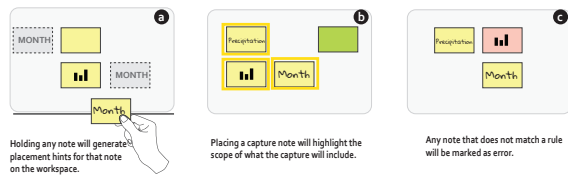
Because the visual language of Composites is broad, we overlay digital content on top of the sticky-notes to facilitate the analysis process. To support seamless interaction between the sticky-notes and on-screen digital information, we seek inspiration from work on tangible augmented reality (AR) interfaces [3, 17, 28, 41, 49]. For example, we employ a strategy similar to White et al. [49] for presenting hints for interaction in the form of textual, diagrammatic, and ghosted hints that are projected onto the work surface.

## 3 USER EXPERIENCE

To understand the tangible interactions and spatial visualization grammar in Composites, we revisit the earlier example of designing features for a smart ambient display. The designer loads Composites on a desktop or laptop computer connected to a camera and projector, both pointing to a whiteboard wall (Figure 1 a). The designer also prints out sticky-notes with ArUco [32] markers (similar workflow to AffinityLens [44]): some markers are ‘unlinked’ and can be bound to data during analysis; others are configured to datasets and data services that the designer’s software team has curated; and a final set is configured with standard chart types such as bar chart, line graph, and scatter plot.

In our earlier example, we saw that while brainstorming what data to show to end-users on the ambient display, the designer was interested in displaying weather information. Specifically, the designer questions whether to alert end-users when the weather changes during the day, for instance alerting about the chance of rain. To further explore this direction, the designer places a physical sticky note with the dataset name “weather dataset” anywhere on the whiteboard wall (Figure 1b). This action loads the dataset, and also displays *virtual notes* for all the columns in that dataset (projections of sticky-notes). Next, the designer employs what we term *sticky interaction*: he creates a physical version of the projected *precipitation* and *month* sticky-notes and places them on top of their projected images. This *binds* the physical post-it notes to their database columns. We call the operation of associating abstract information with a physical note *capturing*. The designer can then move these two notes into a specific spatial arrangement (the *Composites spatial grammar*) which will generate a bar chart showing amount of rain across different months (see Figure 1c).

The designer hypothesizes that based on geography, some users may receive frequent alerts throughout the day. He creates a new sticky note for ‘Washington’ (a state known to receive frequent rainfall) and the year ‘2021’ using *capture* interactions. He then



**Figure 2: Various hints supported by Composites: (a) hint for note placement, (b) capture scope, (c) error hints.**

applies the two values as *filters* to the previously generated bar chart. Next, taking an unlinked note, the designer writes “precipitation in Seattle” and places it next to the projected bar chart. Composites interprets this as a request to *capture* the visualization and “bind” it to the new note. The designer can move the new note anywhere on the wall and the configured visualization will be projected next to the note. Next, the designer replaces the month column for the ‘day’ column to visualize precipitation for each day filtered for the year 2020. This automatically reconfigures the bar chart and the designer captures it with a new note “Seattle Daily Precipitation” He can create visualizations for other states, facet the visualization by month of year, and compare precipitation across different states by moving and placing the corresponding notes next to each other. Through this analysis he determines that rain alerts may be too frequent for some users.

In the above vignette, the designer iteratively creates, and captures visualizations using physical notes augmented by virtual information projected in close proximity to the physical note. Here, sticky-notes, and their spatial-arrangement are the primary mode of interaction for creating visualizations. To support such interactions, Composites offers hints for note placement and data description, and sticky interactions to instantiate and associate physical notes with virtual objects (dataset, columns, visualizations, etc.).

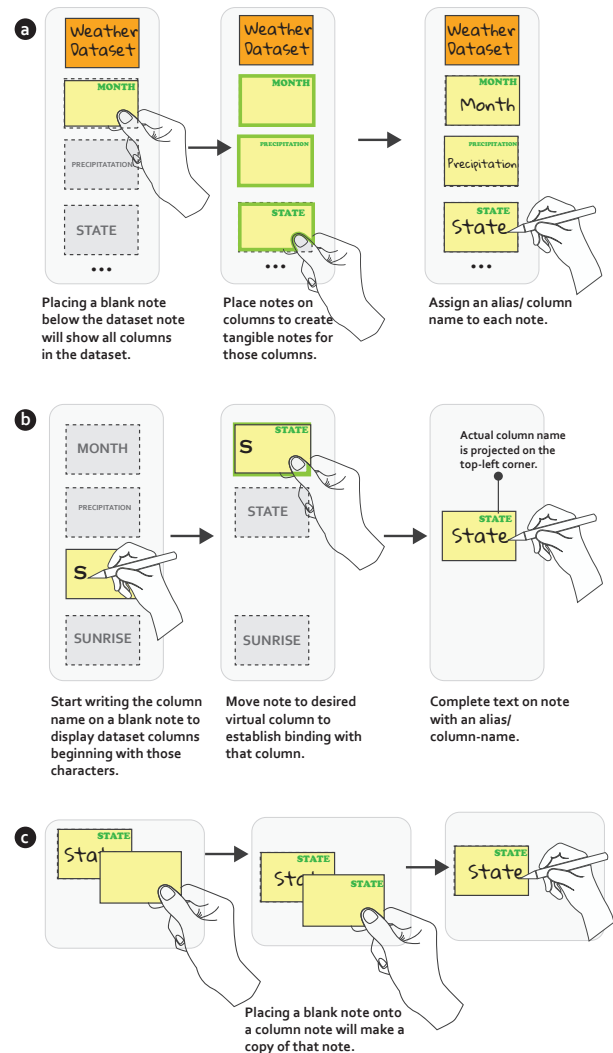
### 3.1 Hints

While our spatial grammar is intended to be simple, Composites gives immediate feedback (hints) to help facilitate interactions. Hints are mostly ephemeral, and show up primarily when the system recognizes a sticky-note in the user’s hand that is yet to be placed, or briefly after a new note has been placed.

The first type of hints will highlight regions where the note can be placed to modify the visualizations (Figure 2a). This is how, in the above scenario, the user learned where to place a note to color or facet the visualization. A second type of hint helps show the scope of what the *capture* interaction will include (e.g. a particular facet of a visualization, the filter, or the entire visualization (Figure 2b). When capturing a visualization, the system highlights all the notes that are part of its specification. The last type of hint (the only one that is present all the time) helps with error conditions. Notes that violates the spatial grammar are highlighted for revision (Figure 2c).

### 3.2 Sticky Interactions

When the designer writes on a note, the note acts as an alias for some underlying construct (e.g., data, operation, visualization). If a visualization is already on the screen, it is obvious that a new note



**Figure 3: Sticky Interactions for binding sticky notes with virtual columns.**

can be bound to that visualization as an alias (e.g., “precipitation across different months”). However, this is harder when the designer is starting from scratch. If the designer knows the name of the column (e.g., ‘precipitation’) they can simply write it. However, in many situations they may be unaware of what is available or find names that are unwieldy to write (e.g., *student\_full\_name*). *Sticky Interactions* speed up this process by helping connect new notes to data (e.g., columns), operations, or both.

When placing a blank note next to the *dataset note*, Composites will display virtual (i.e., projected) notes for all columns in that dataset. The designer can simply place a blank physical note on top of the virtual one (e.g., *student\_full\_name*) and write freeform text on it (e.g., *name*). This physical note becomes bound to the data (see Figure 3a) and when moved around the workspace, Composites will project the full column name on the top-right corner of the

note. A second type of sticky interaction allows the analyst to auto-complete notes (Figure 3b). Here the analyst places a blank note on the surface and write one (or few) characters of the desired column name. This displays virtual notes for all columns that begin with those characters. As before, moving the note onto one of the virtual notes will bind it. The last sticky interaction is for making a copy of a note. Placing a blank note over an already initialized note, assigns that same value to the note, and can be used in composition by writing an alias over the blank note (Figure 3c).

## 4 COMPOSITES

In developing the spatial grammar for Composites, we reference the “Grammar of Graphics” (GoG) syntax [50]. In GoG, visualizations are specified by components that include data and aesthetic mappings, geometric objects, statistical transformations, and facets. For example, in the R implementation (`ggplot2`) we specify a scatterplot of diamond price versus carat as:

```
ggplot() + layer(data = diamonds,
mapping = aes(x = carat, y = price)
, geom = "point", stat = "identity", position = "identity")
+ scale_y_continuous() + scale_x_continuous() +
coord_cartesian() + facet_wrap(color)
```

Composites offers a *spatialized* version of the grammar in which these components are represented by tangible notes and their role in the specification is inferred by relative spatial position or through identifiable markings on the note. Optionally, notes of different colors can be used to help users better manage the roles of the notes (e.g., ones bound to data versus operations).

Figure 4 represents the general specification for creating a visualization. The specification can be divided into different ‘parts’—*data transformations*, *visualizations*, and *view transformations*. Data transformation consists of one or more filters (implicitly combined using the AND operator) in which each filter operation is specified using a *filter note* (which describes the filter condition, e.g., “> 800”) which is placed below using a *column name note* (e.g., “price”). Visualization specification consists of a *mark* or *schema type* note. These may be placed next to a pair of column notes configured in an ‘L’ shape that indicate the *x* and *y* axes (e.g., a line chart of price versus carat). Optionally, the visualization can be faceted by placing a *facet-by note* in the fourth corner.

To the right of the visualization specification are specifications for view transformation and aesthetics such as sorting, and encoding (by color). For example, a bar chart can be sorted in ascending order by placing an *arrange-by note* on the right of the visualization specification. Unlike filter specifications which are conjunctive, visual transformation tokens overwrite other similar token types to the left of that token (e.g. the right most color-by token will overwrite other color-by tokens to its left, etc.). Finally, a specification can be ‘captured’ as a single note by placing a *capture note* to the right of the specification. A specification need not be a complete visualization. For example, if the specification only contains filters, the captured note can be used as a composite filter in combination with other visualization specifications. Other specifications with visualizations can be modified to generate new visualizations, or used for comparison operations such as juxtaposition, superposition, or explicit encoding [13] (Figure 5). The benefit is that partial

code ‘snippets’ (expressed as specifications) can be retained and re-used.

### 4.1 Parser

The Composites parser uses a grid-based clustering approach to parse the spatial arrangement of notes into visualization specifications. The process consists of five steps: (1) pre-processing, (2) gridifying, (3) clustering, (4) parsing, and (5) visualizing (see Figure 6). Note positions are extracted either directly (in the case of the online sandbox) or through a camera feed. In the case of the camera, the image is parsed using computer vision to extract note positions as contours, and also recognize each note. The notes are then assigned to cells in a grid structure based on their position in space. Cells are approximately the size of a standard 3 inch square stick-note. Starting at the top-left note as cell (0,0), we calculate cell positions for all other notes relative to that note. Notes are assigned to different clusters such that all adjacent notes (based on cell position) are assigned to the same cluster, and different clusters are separated by surrounding empty space (the cell size or larger).

Composites parses each cluster independently of other clusters, based on a set of syntactic rules. The only exception is the dataset cluster, which provides the data context for all other clusters. An advantage of this assumption is that any data pre-processing (filters) can be done at this level, and will reflect across all clusters. A rule in Composites consists of a collection of placeholder notes, their types and relative position, along with corresponding specification variables that they map to. For example, the syntax for creating a visualization with two axes consists of two placeholder notes for variables *C1*, and *C2* at cells (0,0) and (1,1) respectively, and also a visualization note, *S1*, at cell (1,0). The corresponding specification rule is made of three variables—*chart-x*, *chart-y*, and *chart-type* and is assigned to notes *C1*, *C2*, and *S1*. Parsing happens through template matching, in which each cluster is parsed left to right, and matching placeholder notes are assigned with instance values (e.g. *C1* = *StudentID*, *C2*= *Score*, *S1*= *Bar*). This design allows for creating new specification, and makes the system more scalable. For example, one could define a new specification for parallel-coordinate plot by writing a new template with a visualization note followed by a set of columns (one for each axis).

If the note is part of the captured specification (e.g. it was previously bound to a complex filter), the note is unpacked by retrieving its stored specifications. After parsing, any new captures are stored in the same list. Constraints mentioned in the previous section about order to data, encoding, and view specifications are followed, and any note violating this constraint is marked for error handling. While parsing, any partial templates and corresponding notes are stored as hints. The output of the parser is a specification for each cluster, along with hints and errors. The specification is then rendered by the visualization module, by calculating placement position from note coordinates within each cluster.

### 4.2 Interface

Using the Composites parser, we implemented two instantiations of the interface—a browser-based ‘sandbox’, and a TUI version. The set-up for the TUI version (Figure 8) consists of a projector which renders the Composites workspace onto a large wall, along

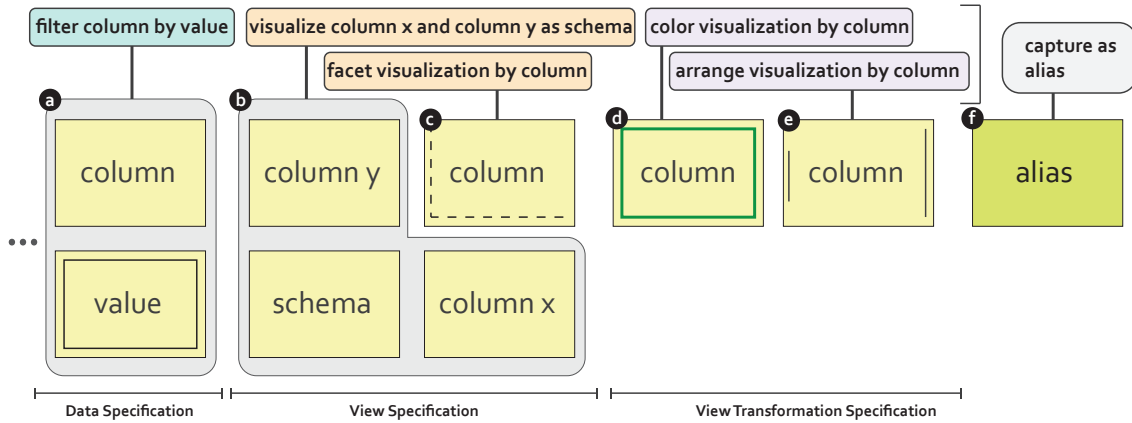


Figure 4: Specification of Composites spatial grammar : (a) specification for filter, (b) visualization, (c) facet, (d) color-by, (e) arrange, and (f) capture.

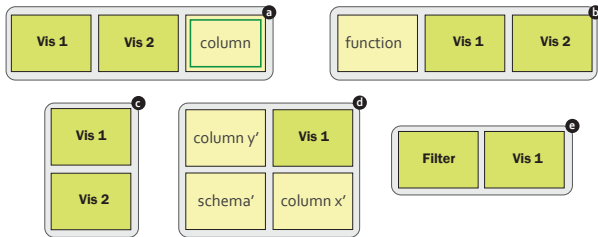


Figure 5: Operations on captured notes: (a), &(b) Juxtaposition for comparison, (c) Superposition, (d) editing, (e) capture as filter.

with a camera mounted on a tripod that serves as the input feed to the parser. In earlier prototypes we implemented handwriting recognition using off-the-shelf algorithms, but we observed poor accuracy due to variations in handwriting. We believe this can be addressed with a more sophisticated recognition system. but to simulate a better interaction experience, we modified the computer vision module to read ArUco markers [32] attached to each sticky-note. This implementation was primarily used in designing and developing the augmented TUI experience.

The sandbox version (Figure 7) can be operated using a mouse and keyboard (though we have also used it on a touch screen display). The layout is comprised of a toolbox on the left, and large gridded workspace spanning the rest of the window. The toolbox includes virtual sticky notes for datasets, and other visualization functions. Similar to the TUI experience, the sandbox provides note placement hints on mouse hover on any note, and notes can be placed and rearranged anywhere on the workspace using drag-drop interactions. The workspace itself is scrollable, and supports pan-zoom interactions to interact with multiple visualizations across the workspace. The visible grid in both the projected and sandbox versions is not strictly necessary but we found that (a) it encourages unambiguous placement of notes, and (b) can be used to calibrate the projector/camera.

## 5 PRELIMINARY EVALUATION

To validate Composites’s spatial grammar we conducted a small user study. There are numerous dimensions on which a visual language can be evaluated [16]. In building the Composites grammar, we contrasted our grammar to the Grammar of Graphics to ensure sufficient *expressiveness* in the language (i.e., that it was possible to generate a wide array of visualizations). With the user study, our specific evaluation goals were on learnability and usability. We conducted this study with 15 participants (8 women) recruited from a major university. Ten out of fifteen participants took a course on information visualization or statistics in the past two years, and all participants reported proficiency with one or more visual analytics tools (e.g., Tableau, Excel, R, SPSS, Power BI).

### 5.1 Tasks

The study consisted of 28 tasks, broadly divided into three types: (1) tasks that required participants to provide the visualization, given a specification (4 tasks), (2) tasks that required them to provide the specification for a given visualization (4 tasks), and (3) tasks that required them to create one or more visualizations to answer visual analytics questions (20 tasks). For the initial eight tasks, we used a *Store Sales* dataset, and the analytics tasks were based on three different datasets – *Coffee Sales* (10 tasks), *Titanic* (5 tasks), and *Baby Names* (5 tasks). Example tasks include simple tasks like “Which region has the highest coffee sales?” and more complex tasks such as “For the product with the lowest sales, which state sold the most of that product?”

### 5.2 Procedure

The study was conducted using the sandbox version of Composites, which was deployed on a standard Windows 10 machine connected to a 23-inch full HD display. We elected to use the sandbox to both scale our testing and to eliminate confounds due to the implementation of the computer vision infrastructure (our goal was to test the language and not to confound it with errors in the off-the-shelf recognition software). Each session lasted between 70-90 minutes, and participants were compensated with \$15 in cash for their time.

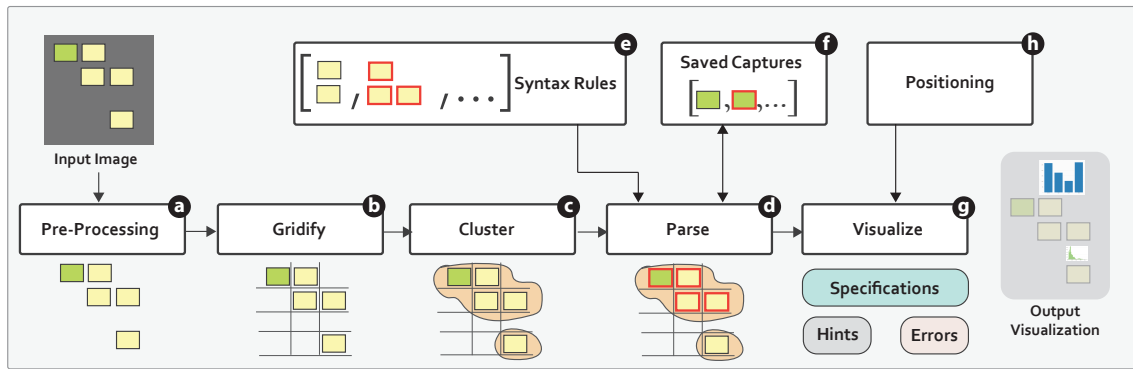


Figure 6: Composites parser depicting how input image is parsed to generate visualizations.

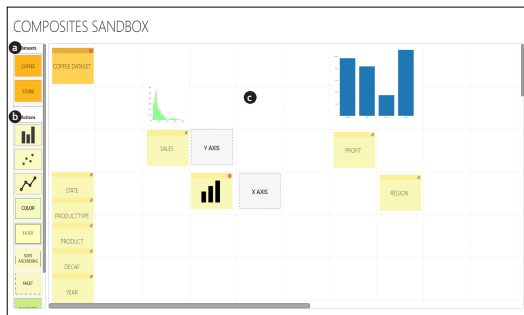


Figure 7: Composites Sandbox: (a) Dataset notes, (b) notes for visualization specification, and (c) zoomable workspace.

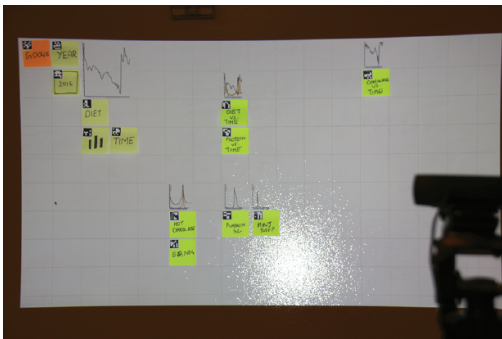


Figure 8: TUI implementation of system using a projector and camera (shown in foreground).

After completing a brief pre-test questionnaire, all participants were provided with an overview of Composites, and given a guided tutorial along with a demo that explained the syntax, and other interaction features of Composites. Once participants indicated familiarity with the syntax they proceeded to answer the first eight tasks which primarily tested their understanding of the syntax. Task instructions were provided in a PowerPoint presentation format,

and participants were instructed to paste screenshots of the sandbox along with notes and visualizations as part of their response. After successfully completing the initial eight tasks, participants worked on the remaining analytics tasks. Finally, all participants filled out the *USE Questionnaire* [30] to report their experience and any additional feedback.

### 5.3 Results

Overall, participants correctly responded to most of the analytics tasks (mean = 82.9%, sd=20.92%), with five participants (33%) who got all of them correct. We discovered that two participants did not appear to understand our instructions and only attempted 9, and 10 tasks respectively. When excluded from analysis, the mean grade improves to 89.68% (sd=11.17%). In analyzing the responses, we observed two common reasons for error: (1) confusion between note ‘strokes’ for different functionality in which some participants incorrectly used filter instead of color, etc., and (2) a few participants did not fully grasp the idea of capture and reuse. The former issue can be easily remedied by using ‘strokes’ that are more visually distinct from each other. In the latter, the participants did not leverage the surface and instead tried to produce a single visualization to answer the question. There are a number of possible reasons for this including that they believed that we wanted only one visualization as an answer but may also be due to their past experiences with visual analytics where there is less emphasis on combining visualizations in different ways.

In the post-test questionnaire, participants rated Composites on a 7-point Likert Scale along the following attributes: usefulness ( $\mu=5.87$ ,  $sd=0.91$ ), ease-of-use ( $\mu=5.26$ ,  $sd=1.09$ ), learnability ( $\mu=5.33$ ,  $sd=1.33$ ), easy-to-remember ( $\mu=5.73$ ,  $sd=1.03$ ), and satisfaction ( $\mu=5.6$ ,  $sd=1.04$ ). In addition, participants reported that they found the drag-and-drop interface was easy to use, and “placement-indicators” made it easy to learn the syntax. One of the participants who got all of the tasks correct reported that “...I like to be able to think in patterns...” while another participant mentioned that they liked the consistency of the interface with fewer clicks to get the results. Participants also expressed the desire for more complex interactions such as sliders, complex filters, and expanding and editing captures. In our future work, we will incorporate this

feedback and evaluate the TUI set-up of Composites for design and creative problem solving tasks.

## 6 DISCUSSION AND FUTURE WORK

Composites was designed to address two problems simultaneously. First, our goal was to find a way of using tangible objects in an ‘immersive’ space to support tasks akin to those of a standard visual analytics platform. Ideally, this could be done cheaply and in a large work surface. Second, we wanted to find a way to extend what many designer’s view as their ‘habitat’—large surfaces and sticky notes—to support quantitative tasks. Composites allows us to address both challenges simultaneously. Sticky notes retain their usual roles (e.g., for card sorting, affinity diagramming) but also become the tangible language and interface for the analytics platform.

The use of data in design tasks is an important and challenging problem. New classes of design—for example, creating AI-powered applications—involve connecting personas with data [45]. Data-assisted affinity diagramming [44] is one solution. Here, the designer can view a summary visualization of underlying notes through the lens of a camera phone. Composites offers a powerful extension that enables the designer to build their own analyses through a tangible language. The resulting projected visualizations can sit in the same space as sticky notes from interviews or surveys. Having built Composites, our continuing goal is to understand how it is used in specific design scenarios.

### 6.1 Assumptions and Limitations

Our current instantiation of Composites demonstrates a novel approach for conducting visual data analysis within existing design practices and flexible and low cost work surfaces. While sticky notes are readily available, and provide great flexibility and control, the notes themselves are temporary, and unlike a Jupyter “document,” require some effort to save, share, and reload. To save Composites specifications would mean making everything virtual, and rebinding sticky notes to virtual information upon loading. The tight coupling between sticky notes and Composites makes the results/insights less accessible outside the AR environment. A partial solution is to capture the entire workspace and make it available in the digital sandbox.

Due to our use of off-the-shelf computer vision and handwriting recognition, Composites struggles with certain conditions (e.g., messy handwriting, or poor lighting conditions). Additionally, while our current implementation, which only requires a projector and standard webcam, is cheap and accessible, it also suffers occlusion and resolution issues. This limits the range of end-user’s mobility in space, and would require special adaptation to overcome occlusion. As part of ongoing work, we are experimenting with projecting picture-in-picture style ‘proxy’ views for occluded region where the display is not blocked. Head-mounted AR displays such as the HoloLens, in which both display and camera are positioned between the user and the wall, mitigates occlusion. However, HoloLens-style systems have a limited field of view prevents good awareness outside of center of interest. Another alternative is the use of dual projectors which can eliminate occlusion ‘shadows’ (e.g., [46]).

### 6.2 Future Work

While the current study examines whether people can understand and use the spatial grammar component of the Composites to construct visualizations, there are several directions which we would like to explore further. Three particularly exciting avenues for future work are visual analytics education, higher level semantic interaction, and collaborative decision making scenarios. Systems like Tableau are effective for exploring tabular data, but face a steep learning curve for novice users who are unfamiliar with the kind of table-based analytics that they support. Composites makes it easy to capture an existing visualization into a template that can be reused and edited. These captured visualizations can represent starting points for novices where direct manipulation can be used for subsequent changes, and hints for modification can be overlaid directly on the visualization itself. We would like to investigate whether the spatial grammar (along with contextual hints) can be learned more easily than other, existing systems. Another avenue to explore more deeply is the utility of higher level interactions with full visualizations. Composites lends itself naturally to incorporate (and extend) recent work on semantic snapping [25] so that alternate representations can be suggested when multiple visualizations are brought together in close proximity. The ability to create and capture new visualizations by combining existing visualizations could allow the user to answer deeper questions without worrying about the low-level interactions required to create them. Finally, exploring whether the spatial affordances that the system was designed for truly facilitates collaborative decision making is an important future step.

## 7 CONCLUSION

We describe the design and implementation of Composites, a new tangible interaction paradigm for data-driven design practices. Composites allows designers to use their existing design work surfaces to conduct visual data analysis. Construction and modification of visualization is made possible directly on the surface through the use of tangible sticky-notes and a spatial grammar. The tangible tokens allow for quick rearrangement and encapsulation of insights, and our expressive *spatial grammar* is scalable, and can support a wide range of analytic needs. Using Composites, designers can easily create, modify, group, and combine multiple visualizations, all in the same view. The fluidity in movement between tangible and virtual tokens preserves the best affordances of sticky notes, while providing computational scaffolding to assist with complex visualization tasks and may be applicable to other AR-based analytical systems.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers and our study participants for their time and helpful feedback. We also thank Curtis Wong, Dave Brown, and Roland Fernandez for their inputs on the design of Composites.

## REFERENCES

- [1] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. 2011. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization* 10, 4 (2011), 341–355.



- [2] Apple. 2021. Apple Watch faces and their features. <https://support.apple.com/guide/watch/faces-and-features-apde9218b440/watchos>
- [3] Shunsuke Asai and Masahito Hirakawa. 2009. Dual face interaction in hand-held augmented reality environments. In *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on*. IEEE, 187–193.
- [4] Sriram Karthik Badam and Niklas Elmqvist. 2019. Visfer: Camera-based visual data transfer for cross-device visualization. *Information Visualization* 18, 1 (2019), 68–93.
- [5] Marco Cavallo, Mishal Dholakia, Matous Havlena, Kenneth Ocheltree, and Mark Podlaseck. 2019. Dataspace: A reconfigurable hybrid reality environment for collaborative information analysis. In *2019 IEEE Conf. on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 145–153.
- [6] Kelvin Cheng and Kevin Pulo. 2003. Direct interaction with large-scale display systems using infrared laser tracking devices. In *Proc. of the Asia-Pacific symposium on Information visualisation-Volume 24*. Australian Computer Society, Inc., 67–74.
- [7] Steven M Drucker, Danyl Fisher, Ramik Sadana, Jessica Herron, et al. 2013. TouchViz: a case study comparing two interfaces for data analytics on tablets. In *Proc. of SIGCHI*. ACM, 2301–2310.
- [8] Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2879–2888.
- [9] Barrett Ens, Sarah Goodwin, Arnaud Prouzeau, Fraser Anderson, Florence Y Wang, Samuel Gratzl, Zac Lucarelli, Brendan Myle, Jim Smiley, and Tim Dwyer. 2020. Uplift: A tangible and immersive tabletop system for casual collaborative visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1193–1203.
- [10] Bernd Fröhlich and John Plate. 2000. The cubic mouse: a new device for three-dimensional input. In *Proc. of SIGCHI*. ACM, 526–531.
- [11] Johannes Fuchs, Roman Rädle, Dominik Sacha, Fabian Fischer, and Andreas Stoffel. 2013. Collaborative data analysis with smart tangible devices. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 90170C–90170C.
- [12] Florian Geyer, Ulrike Pfeil, Anita Höchtl, Jochen Budzinski, and Harald Reiterer. 2011. Designing reality-based interfaces for creative group work. In *Proc. of the 8th ACM Conf. on Creativity and cognition*. ACM, 165–174.
- [13] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309.
- [14] Eric J Golin. 1991. Parsing visual languages with picture layout grammars. *Journal of Visual Languages & Computing* 2, 4 (1991), 371–393.
- [15] Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. 2013. A survey of visualization construction user interfaces. *EuroVis-Short Papers* (2013), 19–23.
- [16] T.R.G. Green and M. Petre. 1996. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages & Computing* 7, 2 (1996), 131 – 174. <https://doi.org/10.1006/jvlc.1996.0009>
- [17] Taejin Ha, Steven Feiner, and Woontack Woo. 2014. WeARHand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*. IEEE, 219–228.
- [18] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proc. of the 32nd annual ACM Conf. on Human factors in computing systems*. ACM, 1063–1072.
- [19] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proc. of the ACM SIGCHI Conf. on Human factors in computing systems*. ACM, 234–241.
- [20] Waqas Javed and Niklas Elmqvist. 2013. ExPlates: spatializing interactive analysis to scaffold visual exploration. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 441–450.
- [21] Seokhee Jeon, Jane Hwang, Gerard J Kim, and Mark Billinghurst. 2006. Interaction techniques in large display environments using hand-held devices. In *Proc. of the ACM symposium on Virtual reality software and technology*. ACM, 100–103.
- [22] Hao Jiang, Ron B Yeh, Terry Winograd, and Yuanchun Shi. 2007. DigiPost: writing on post-its with digital pen to support collaborative editing tasks on tabletop displays. *UIST Posters* (2007).
- [23] Jupyter. 2021. Jupyter Notebook. <https://jupyter.org/>
- [24] Scott Klemmer, Mark W Newman, and Raecine Sapien. 2000. The designer's outpost: a task-centered tangible interface for web site information design. In *CHI'00 extended abstracts on Human factors in computing systems*. ACM, 333–334.
- [25] Yngve S Kristiansen, Laura Garrison, and Stefan Bruckner. 2021. Semantic Snapping for Guided Multi-View Visualization Design. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 43–53.
- [26] Ricardo Langner, Marc Satkowski, Wolfgang Büschel, and Raimund Dachselt. 2021. Marvis: Combining mobile devices and augmented reality for visual data analysis. In *Proc. of the 2021 CHI Conf. on Human Factors in Computing Systems*. 1–17.
- [27] Bongshin Lee, Greg Smith, Nathalie Henry Riche, Amy Karlson, and Sheelagh Carpendale. 2015. SketchInsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*. IEEE, 199–206.
- [28] Gun A Lee, Mark Billinghurst, and Gerard Jounghyun Kim. 2004. Occlusion based interaction methods for tangible augmented reality environments. In *Proc. of the 2004 ACM SIGGRAPH international Conf. on Virtual Reality continuum and its applications in industry*. ACM, 419–426.
- [29] Joon Hyub Lee, Donghyeok Ma, Haena Cho, and Seok-Hyung Bae. 2021. Post-Post-it: A Spatial Ideation System in VR for Overcoming Limitations of Physical Post-it Notes. In *Extended Abstracts of the 2021 CHI Conf. on Human Factors in Computing Systems*. 1–7.
- [30] Arnold M Lund. 2001. Measuring Usability with the USE Questionnaire12. *Usability interface* 8, 2 (2001), 3–6.
- [31] Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)* 5, 2 (1986), 110–141.
- [32] Juan Mellado. 2018. ArUco JavaScript. <https://github.com/jcmellado/js-aruco>
- [33] Chris North, Nathan Conklin, and Varun Saini. 2002. Visualization schemas for flexible information visualization. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*. IEEE, 15–22.
- [34] Chris North and Ben Shneiderman. 2000. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proc. of the working Conf. on Advanced visual interfaces*. ACM, 128–135.
- [35] C Olsten, Michael Stonebraker, Alexander Aiken, and Joseph M Hellerstein. 1998. VIQING: Visual interactive querying. In *Visual Languages, 1998. Proceedings. 1998 IEEE Symposium on*. IEEE, 162–169.
- [36] Jun Rekimoto, Brygg Ullmer, and Haruo Oba. 2001. DataTiles: a modular platform for mixed physical and graphical interactions. In *Proc. of SIGCHI*. ACM, 269–276.
- [37] George Robertson, Mary Czerwinski, Patrick Baudisch, Brian Meyers, Daniel Robbins, Greg Smith, and Desney Tan. 2005. The large-display user experience. *IEEE computer graphics and applications* 25, 4 (2005), 44–51.
- [38] Ben Shneiderman. 1997. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proc. of the 2nd international Conf. on Intelligent user interfaces*. ACM, 33–39.
- [39] Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, and Kellogg S Booth. 2010. Body-centric interaction techniques for very large wall displays. In *Proc. of the 6th Nordic Conf. on Human-Computer Interaction: Extending Boundaries*. ACM, 463–472.
- [40] Jim Smiley, Benjamin Lee, Siddhant Tandon, Maxime Cordeil, Lonni Besançon, Jarrod Knibbe, Bernhard Jenny, and Tim Dwyer. 2021. The MADE-Axis: A Modular Actuated Device to Embody the Axis of a Data Dimension. *Proc. of the ACM on Human-Computer Interaction* 5, ISS (2021), 1–23.
- [41] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: projected visualizations for hand movement guidance. In *Proc. of SIGCHI*. ACM, 179–188.
- [42] Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on* 8, 1 (2002), 52–65.
- [43] Hariharan Subramonyam and Eytan Adar. 2018. SmartCues: a multitouch query approach for details-on-demand through dynamically computed overlays. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 597–607.
- [44] Hariharan Subramonyam, Steven M Drucker, and Eytan Adar. 2019. Affinity Lens: Data-Assisted Affinity Diagramming with Augmented Reality. In *Proc. of the 2019 CHI Conf. on Human Factors in Computing Systems*. 1–13.
- [45] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. 2021. Towards A Process Model for Co-Creating AI Experiences. (2021), 1529–1543. <https://doi.org/10.1145/3461778.3462012>
- [46] R. Sukthankar, Tat-Jen Cham, and G. Sukthankar. 2001. Dynamic shadow elimination for multi-projector displays. In *Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 2. II–II. <https://doi.org/10.1109/CVPR.2001.990943>
- [47] Tableau. 2021. Vizable by Tableau. <https://vizable.tableau.com/>. <https://vizable.tableau.com/>
- [48] Bret Victor. 2021. Dynamic Land. <https://dynamicland.org/>
- [49] Sean White, Levi Lister, and Steven Feiner. 2007. Visual hints for tangible gestures in augmented reality. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 47–50.
- [50] Leland Wilkinson. 2006. *The grammar of graphics*. Springer Science & Business Media.
- [51] William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. 2006. The Sandbox for analysis: concepts and methods. In *Proc. of SIGCHI*. ACM, 801–810.