Representing Real-Time Multi-User Collaboration in Visualizations

Rupayan Neogy* Jonathan Zong† Arvind Satyanarayan‡

MIT CSAIL MIT CSAIL MIT CSAIL

ABSTRACT

Establishing common ground and maintaining shared awareness amongst participants is a key challenge in collaborative visualization. For real-time collaboration, existing work has primarily focused on synchronizing constituent visualizations — an approach that makes it difficult for users to work independently, or selectively attend to their collaborators' activity. To address this gap, we introduce a design space for representing synchronous multi-user collaboration in visualizations defined by two orthogonal axes: *situatedness*, or whether collaborators' interactions are overlaid on or shown outside of a user's view, and *specificity*, or whether collaborators are depicted through abstract, generic representations or through specific means customized for the given visualization. We populate this design space with a variety of examples including generic and custom synchronized *cursors*, and *user legends* that collect these cursors together or reproduce collaborators' views as thumbnails. To build common ground, users can interact with these representations by *peeking* to take a quick look at a collaborator's view, *tracking* to follow along with a collaborator in real-time, and *forking* to independently explore the visualization based on a collaborator's work. We present a reference implementation of a wrapper library that converts interactive Vega-Lite charts into collaborative visualizations. We find that our approach affords synchronous collaboration across an expressive range of visual designs and interaction techniques.

Index Terms: Human-centered computing—Visualization—Visualization systems and tools;

1 INTRODUCTION

As interactive visualization is increasingly integrated into data journalism, science communication, and public health response, there is a growing need for research on how interactivity might facilitate collaborative sensemaking. To meet these changing needs, a recent thread of research in *collaborative visualization* has investigated methods for multiple users to interactively explore and analyze the same visualization across multiple locations and devices. Systems such as Polychrome [2] and Vistrates [3] provide techniques for synchronizing a visualization, such that interactions performed on one device immediately appear on all others. Though these systems recognize the value of maintaining shared awareness of collaborators' interactive exploration, they do not yet support ways of doing so while allowing users to conduct their own independent, divergent exploration. sense.us [12] and Many Eyes [21] support collaborative analysis through graphical annotations and textual comments, and information scent techniques have also been explored to depict the expectations of other readers [16]. However, such approaches target *asynchronous* collaboration; for synchronous collaboration, prior work has considered only specific interaction techniques such as brushing & linking [8] or query widgets [13].

To generalize from these approaches, we introduce a design space for augmenting visualizations with representations of remote users

---

*e-mail: rneogy@mit.edu

†e-mail: jzong@mit.edu

‡e-mail: arvindsatya@mit.edu

that enable shared awareness without sacrificing independent exploration. This design space is described by two orthogonal axes: *situatedness*, or whether collaborators are represented within the visualization or externally, and *specificity*, or whether collaborators' representations are generic or tailored to the visual idioms of the visualization. We present designs for *cursors*, high-level representations of remote users that encode information about their interactions. Cursors do not actively modify the visualization state, allowing users to maintain independent viewing experiences, while providing *information scent* [19] of remote interaction. By default, cursors appear as small color-coded dots on the visualization, situated in the location of a remote user's interaction. However, their appearance can be customized to increase their specificity to a particular visualization. We also introduce designs for *user legends*, which are collections of user representations situated alongside the visualization. Legends can similarly be more or less specific to a visualization. For instance, legends may collect cursors into one place to prevent crowding within the visualization, or fully represent remote user views as thumbnails.

In addition to these visual approaches, we contribute a set of interactive methods to help users build common ground. These techniques include *peeking*, a temporary look at a remote user's view; *tracking*, a continuous synchronized view of a remote user's visualization; and *forking*, a divergent exploration starting from a remote user's state. By interacting with cursors, users can examine their collaborators' interactive exploration in an ambient fashion. For example, a user can hover over a cursor to peek at the remote user's view before returning to their previous interaction state, or they can click on the cursor to actively update their visualization state with the remote one. In this way, each user has their own visualization to explore, and remote interactions do not affect their experience until the user chooses to interact with a cursor. Augmenting each user's separate instance ensures that users gain collaborative benefits while retaining agency over their own exploration.

To validate our designs, we implement a wrapper for the Vega-Lite visualization grammar [20] that converts Vega-Lite specifications into synchronous, collaborative visualizations. We explore a variety of user representation designs, demonstrating that our contribution affords collaboration across an expressive range of visualization designs and interaction techniques.

2 RELATED WORK

Our work builds on prior methods for collaborative visualization and computer-supported cooperative work (CSCW), and draws inspiration from techniques for representing multiple users collaborating around textual documents.

2.1 Collaborative Visualizations and Documents

Early systems such as Many Eyes [21], the Baby Name Voyager [22], and sense.us [12] helped identify that visualizations can indeed serve as social artifacts. Through graphical annotations and textual comments, users of these systems eagerly engaged in collaborative data analysis. However, this activity could only occur asynchronously and, in a more recent thread of work, researchers have explored how visualizations can also facilitate synchronous collaboration. For example, systems like Polychrome [2] and Vistrates [3] support both asynchronous and synchronous collaboration by allowing users to keep their visualizations synchronized with one another.

Reflecting on this work, researchers have defined the terms *collaborative visualization* [14] and *ubiquitous analytics* [6] and have identified their associated challenges [23] and design considerations [9]. In particular, a chief concern is how to establish common ground among collaborators, and allow them to maintain a shared awareness of each other's work. sense.us addressed this concern through both traditional social navigation mechanisms (e.g., user profiles and comment listings) and by introducing "doubly-linked" discussions where textual comments deep-link to specific interactive visualization states, and corresponding textual comments automatically populate the discussion panel as a user interacts with the visualization [12]. However, for synchronous collaborative visualization, existing work instead primarily focuses on the backend frameworks and architectures necessary to support this type of behavior [2, 3].

Here, collaborative word processing packages such as Google Docs, Etherpad, and Overleaf suggest a way forward. Like their visualization counterparts, these tools enable synchronous collaboration by maintaining a shared state and keeping each user's view synchronized. As a result, common ground is established in a straightforward fashion as every user sees the same visual output. And, to maintain a shared awareness, these systems augment the view by assigning each user their own color-encoded cursor. Researchers have adopted these ideas for visual analytics systems but in the context of specific visual idioms or interaction techniques — for instance, using color-coded representations for brushing & linking [8], query widgets [13], and externalizations of network data [17]. Our work goes a step further by generalizing these point examples into a rich design space that is agnostic to or customized for mark types and is applicable to a broad range of selection-based interaction techniques. Moreover, our work also considers how to instrument these shared awareness representations with techniques for re-establishing common ground.

2.2 Computer-Supported Cooperative Work (CSCW)

The CSCW literature organizes collaborative behavior using the *time-space matrix*: the time dimension indicates whether participants interact synchronously or asynchronously, and the space dimension describes whether collaborators are co-located or geographically distant [1, 15]. Wood et al. instantiated these ideas in the space of visualization, coining the term computer supported collaborative visualization (CSCV) [24]. In a literature survey [4], they situate a variety of visualizations systems within this matrix. However, this discussion is primarily concerned with visualization models and architectures (i.e., the components and abstractions available for constructing distributed or collaborative visualizations) and does not dwell on aspects of user experience. Our work addresses the synchronous (same time) column of the time-space matrix, and is agnostic to whether users are co-located or distant.

Our work is moreover guided by Grudin's principle of *unobtrusive accessibility*: collaborative features should not disrupt an individual's activity [7]. This principle helps us identify a shortcoming with simply synchronizing views across participants. While only mildly disruptive for collaborative document editing, keeping views synchronized can prevent independent exploration of visualizations for a large swath of interactions performed by remote users — for example, filtering data out or panning and zooming to explore a different subset of the data. In contrast, under our approach, each user maintains their own independent instance of a visualization and interactive state is synchronized and depicted through in situ or external graphical representations (Fig. 1). To mitigate any loss of common ground, we additionally contribute interactive behaviors that a user can use to *peek* into or *track* their collaborators' activity.

3 A DESIGN SPACE FOR REPRESENTING REMOTE USERS

To formalize the design dimensions for representing remote users on visualizations, we used an iterative design process seeded by prior approaches discussed in the previous section. In particular, we
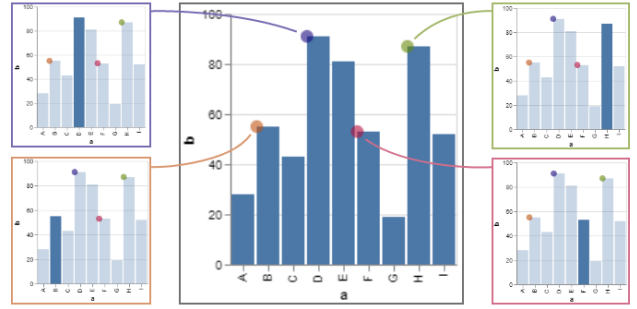

Figure 1: Selections from various remote users are translated into interactive cursors on each view.

sought to distill a minimal set of dimensions capable of covering a diverse range of visualization types and interaction techniques. We prototyped candidate designs using Vega-Lite [20], as its expressive grammar allowed us to systematically enumerate mark types and interaction techniques. To refine our designs, we conducted informal feasibility tests with members of our research group and other students at our research institution. These tests revealed the tradeoffs at stake: while participants preferred direct annotations on the visualization, this could also cause visual crowding or conflicts with interactions that alter the main view. We ultimately conceptualize the design space of representations of remote users along the following two orthogonal dimensions.

Situatedness refers to the location of user representations in relation to the visualization: marks representing remote user interactions can be placed either in situ (overlaying the visualization) or externally. This axis describes the tradeoff between depicting remote users in context and the degree to which doing so can disrupt reading a visualization. For instance, in situ representations can leverage the visualization's encodings, scales, axes, and legends to aid a viewer's awareness of their collaborator's activity. On the other hand, such representations may also crowd the visualization and obscure legibility, especially when there are large numbers of remote users. In such cases, external representations might be preferred.

Specificity refers to whether user representations are designed generically, without respect to the content of the visualization, or if they are customized to the visual idioms and interaction techniques of a particular visualization. For example, representing users with minimal circular marks will generically apply to any visualization without much knowledge of what interactions it supports. However, on a scatterplot with a brushing interaction, we can represent remote users as color-coded, synchronized rectangles — a representation tailored to this specific interaction technique. Specificity determines how information-dense representations of remote users are: a more specific representation communicates more information about a remote user, but requires added effort to design.

We use *cursor* to refer to the basic unit of user representation in a collaborative visualization. Each cursor corresponds to a single remote user and can vary in appearance based on trade-offs between *situatedness* and *specificity*. Fig. 2 provides a variety of example designs that occupy different positions within these two dimensions.

3.1 Generic Cursors

Generic cursors, shown in Fig. 2A, are small, translucent, color-coded marks overlaid on the visualization in the position of each remote user's interaction. They are designed to provide awareness of remote collaborators in context while minimally interfering with a user's visualization experience — for instance, users can selectively attend to them, and they are sufficiently unobtrusive to be ignored altogether. Moreover, their generic design allows them to operate across a variety of visualizations (e.g., histograms, scatter plots,
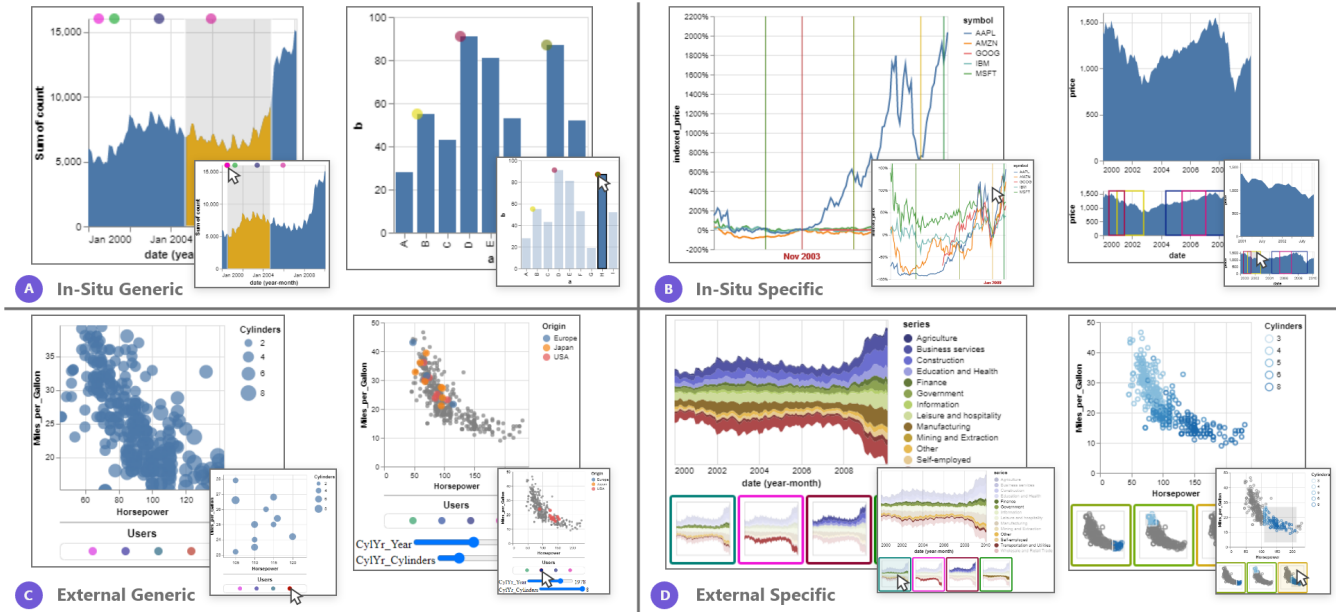
Figure 2: Example gallery of user representations on collaborative visualizations. Inset views show the result of the *peeking* hover interaction for each example. A) Generic cursors overlay each visualization, indicating remote interaction. B) Specific cursors provide more visualization-specific information about remote user interaction. C) Cursor legends provide an external, minimal representation of remote users. D) Thumbnail legends visually communicate a specific overview of the remote interaction state.

line charts, etc.) and interaction techniques (e.g., reflecting a user's mouse position, the top-left coordinate of a brushed region, etc.).

We explored various generic abstractions of remote user interactions. Given the rich space of interaction techniques for data visualizations [11, 25], we found position to be the property of remote user behavior that is most broadly applicable and meaningful—it either depicts attributes of interaction (e.g., brush extents) or falls back to reflecting the remote user's cursor location.

3.2 Specific Cursors

In situ cursors can be customized heavily in a variety of ways to present more specific representations of remote user interactions. Fig. 2B illustrates how color-coded vertical rule cursors indicate the index points remote users are exploring when interactively re-normalizing stock price time-series data. The same figure also presents a synchronized version of an overview + detail plot, where brushes on the lower chart change the zoom and pan of the upper view. Custom color-coded rectangles outline the brushed regions from remote users. Note, while our examples use color to distinguish remote users, any categorical encoding channel can be used to do so (e.g., encoding each remote user using a unique shape).

Specific cursors replicate recent approaches for representing users in collaborative visualization (e.g., synchronized brushes and selections à la Vistrates [3], or zoom windows, as presented in Visualive [18]). They afford more information-dense displays of remote user interactions but risk trading off readability. For example, having multiple brush outlines on the overview can interfere with a user's ability to define a new brush. Similarly, the vertical lines on the stock-index chart can potentially confuse the viewer about what point in time is rescaling the data. These examples demonstrate that specific cursors must be designed carefully and with purpose to serve a specific collaborative goal, whether that may be presentation or independent exploration of the data.

3.3 Cursor Legends

While in situ cursor designs work for a variety of visualizations types and interaction techniques, there are many instances where an external representation may be preferred. For example, perhaps the visualization is already very data-dense, such as a scatter plot with a large number of points. Overlaying translucent circular cursors could obfuscate the data or confuse the viewer. For cases where overlaying cursors is not appropriate or effective, we present the *cursor legend*, an off-visualization container for cursors as shown in Fig. 2C. The first example uses a cursor legend to depict remote panning & zooming behaviors. This interaction technique presents an interesting challenge: a remote user may have panned or zoomed entirely out of the current user's view. In such a situation, in situ cursors would either disappear entirely or could be made to clamp to the corners of the view rectangle—in either case, there is the potential for confusion from their typical behavior. With a cursor legend, all remote users are presented in a consistent manner regardless of where they have panned or zoomed to. The second example illustrates dynamic query widgets. Here too, the cursor legend is arguably a more natural representation both because the interactive widgets are external to the visualization and because a remote user's interactive state is the joint combination of all sliders—an in situ alternative might overlay each slider with a cursor but would require additional effort from a viewer to piece back together.

In summary, cursor legends preserve the original visualization experience, with no modifications or overlays, but trade off the ambient information-scent affordance of their in situ counterparts. Moreover, cursor legends can also be used in conjunction with in situ cursors as a secondary representation of remote users, or to provide more information for each user (e.g., usernames or avatars as often seen in collaborative word processing tools).

3.4 Thumbnail Legends

Thumbnail legends, shown in Fig. 2D, recall bookmark trails from sense.us [12] as well as graphical histories [10]. They provide high-level previews of remote visualizations and their interactive state. Just as specific cursors provide an alternative to generic cursors by representing remote interactions using designs tailored to the visualization and interaction technique, thumbnail legends provide a specific alternative to cursor legends. Relative to other representations,

thumbnail legends allow more direct visibility into collaborators' activity. However, they are less compact than cursor legends.

Thumbnails are designed as minimal representations of remote visualizations. Rather than simply scaling down a view, thumbnails strip out elements such as legends, titles, axes, and labels—information that is unnecessary for a high-level understanding of the remote interaction. Future work might explore ways to semantically simplify visualizations, for example, by selecting representative points in a scatterplot that communicate the overall shape of the data. Such visual reductions may require additional data transformation.

4 INTERACTIVELY ESTABLISHING COMMON GROUND

Our design space describes how to design representations of collaborators that overlay or augment a visualization such that a user can conduct their own individual, independent exploration of the data. However, when independent explorations cause views to diverge, such as when two users zoom into different parts of the visualization, collaborators risk losing common ground [5]. To mitigate this risk, we present an additional layer of interactivity afforded by these cursors to allow for novel collaborative experiences.

4.1 Peeking

We define *peeking* as the action of temporarily setting the viewer's visualization state to that of a remote user. Semantically, peeking refers to a user's intention to glance at a remote user's view without necessarily committing their view to that state. We deem it important that users can easily peek at a remote user's visualization and then return to their original visualization state. In other words, peeking is a non-committal interaction between users, and the peeker does not relinquish control over their visualization at any given moment. By default, the viewer can peek at a remote interaction state by hovering over a cursor. The ephemeral quality of the mouse-over interaction matches well with the temporary nature of the peek interaction. Of course, peeking can be triggered by any low-level event the visualization designer sees fit, but the technique is important for providing users the ability to explore remote user states easily.

4.2 Tracking

We define *tracking* as the action of actively synchronizing the viewer's visualization state to that of a remote user. In other words, when one user tracks another, their visualization mirrors the view of the remote visualization. This form of synchronization follows the work of most recent research collaborative visualization systems [2, 3] in allowing users to view remote interactions in situ and in real-time. Tracking is a persistent version of peeking, and implementing one should effectively enable the other. In our implementation, the viewer can track other users by clicking on a cursor, which follows from hovering to peek at a remote interaction state.

4.3 Forking

We define *forking* as the action of taking a remote user's interaction state as the starting point for new, divergent exploration. After a user has been tracking another user, they might identify something interesting in the remote view and choose to interact with the visualization themselves to modify the view. Forking, by definition, ends the act of tracking another user, as new interactions modify local state separately from remote state. Forking is especially useful for visualizations that have many interaction techniques, such as several query filters. One might find a particular selection of filters interesting and choose to fork from that state by narrowing down on a subset of the data. Or, one might choose to flip one of the filters around to see how the same set of selections applies along a different dimension of the data. Under our approach, as every user is giving their own instance of a collaborative visualization, forking occurs automatically when a user begins interacting with a visualization that was tracking another user.
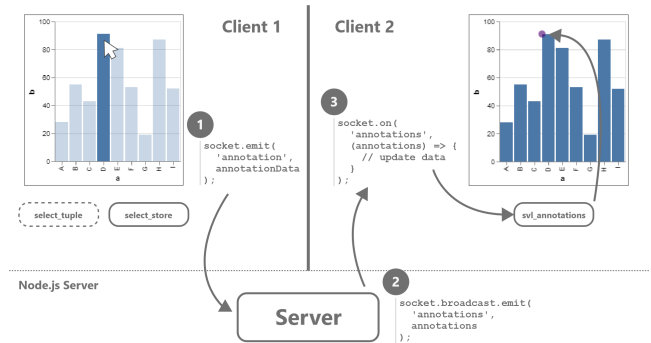

Figure 3: An overview of the how interactions on one client translate into cursors on another. (1) Client 1 reacts to a click event by emitting the corresponding data tuples to the server; (2) the server transmits this data to all connected clients; (3) Client 2 receives new data and updates its state, rendering new cursor marks on the visualization.

5 IMPLEMENTATION AND EXAMPLE GALLERY

We demonstrate the expressiveness of our designs by implementing a wrapper around the Vega-Lite visualization grammar [20]. Our wrapper converts any Vega-Lite specification into a synchronous, multi-user collaborative visualization. Our interaction model consequently assumes interactions that are expressible as Vega-Lite selections, though future work may explore compatibility with custom interactions. Our implementation consists of a client and a server package (Fig. 3). The client package annotates a given Vega-Lite specification with extra marks, signals, and data to render the cursors; it translates user interactions on the visualization into messages that are sent to collaborators; and, it parses incoming messages to update visualization state. The server package relays information between clients and maintains user state (e.g., color assignments).

We evaluate expressiveness by creating an example gallery (Fig. 2). Examples are implemented with our prototype, demonstrating that our cursor designs and interaction techniques cover an expressive range of visualization types (histograms, scatter plots, line charts) and interaction types (brushing, query widgets, hover interactions, and single/multiple point selections).

6 CONCLUSION

We present a design space for representing multiple users collaborating around a visualization without impeding an individual user's ability to read and analyze the visualization. To re-establish common ground between collaborators, we additionally contribute interactive techniques that allow users to peek or track their collaborators activity, or fork their own path forward. We instantiate these ideas in a wrapper for Vega-Lite, and demonstrate coverage across a diverse range of visualization types and interaction techniques. Our prototype implementation is available as open-source software at `https://github.com/mitvis/synchronized-vega-lite`.

Our design space and interaction techniques enable further exploration of complex collaborative user interactions. For example, Akia might be watching Deniz highlight features of their data. Akia might notice an interesting avenue of exploration and ask to take over. Once Deniz relinquishes control, Akia filters the data as Deniz watches. Using our definitions, Akia started off tracking Deniz, then forked Deniz's state, as Deniz switched to tracking Akia. Perhaps this interaction warrants its own name, or raises the question of when explorations should diverge or come together. Our core set of interaction techniques facilitates further discussion about user representation and collaboration on visualizations.

REFERENCES

[1] L. M. Applegate. Technology support for cooperative work: A framework for studying introduction and assimilation in organizations. *Journal of Organizational Computing and Electronic Commerce*, 1(1):11–39, 1991.

[2] S. K. Badam and N. Elmqvist. Polychrome: A cross-device framework for collaborative web visualization. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, p. 109–118. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2669485.2669518

[3] 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*, 25(1):586–596, 2019.

[4] K. W. Brodlie, D. A. Duce, J. R. Gallop, J. P. Walton, and J. D. Wood. Distributed and collaborative visualization. In *Computer graphics forum*, vol. 23, pp. 223–251. Wiley Online Library, 2004.

[5] H. H. Clark and S. E. Brennan. Grounding in communication. 1991.

[6] N. Elmqvist and P. Irani. Ubiquitous analytics: Interacting with big data anywhere, anytime. *Computer*, 46(04):86–89, apr 2013. doi: 10.1109/MC.2013.147

[7] J. Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):92–105, 1994.

[8] A. H. Hajizadeh, M. Tory, and R. Leung. Supporting awareness through collaborative brushing and linking of tabular data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2189–2197, 2013.

[9] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008. doi: 10.1057/palgrave.ivs.9500167

[10] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 14(6):1189–1196, 2008.

[11] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Queue*, 10(2):30–55, 2012.

[12] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. In *ACM Human Factors in Computing Systems (CHI)*, pp. 1029–1038, 2007.

[13] S. R. Hong, M. M. Suh, N. Henry Riche, J. Lee, J. Kim, and M. Zachry. Collaborative dynamic queries: Supporting distributed small group decision-making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173640

[14] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K.-L. Ma, and H. Hagen. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization*, 10(4):310–326, 2011. doi: 10.1177/1473871611412817

[15] R. Johansen. *Groupware: Computer support for business teams*. The Free Press, 1988.

[16] Y.-S. Kim, K. Reinecke, and J. Hullman. Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2018.

[17] N. Mahyar and M. Tory. Supporting communication and coordination in collaborative sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1633–1642, 2014. doi: 10.1109/tvcg.2014.2346573

[18] R. Neogy, E. Hu, and A. Satyanarayan. Visualive: Representing Synchronized Visualization Interactions. In *IEEE InfoVis Posters*, 2019.

[19] P. Pirolli and S. Card. Information foraging. *Psychological Review*, 106(4):643–675, 1999. doi: 10.1037/0033-295x.106.4.643

[20] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016.

[21] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[22] M. Wattenberg. Baby names, visualization, and social data analysis. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 1–7. IEEE, 2005.

[23] J. Wood, H. Wright, and K. Brodie. Collaborative visualization. In *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, pp. 253–259, 1997.

[24] J. Wood, H. Wright, and K. Brodlie. Cscv-computer supported collaborative visualization. In *Proceedings of BCS Displays Group International Conference on Visualization and Modelling*, pp. 13–25. Citeseer, 1995.

[25] J. S. Yi, Y. ah Kang, J. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231, 2007.