# Visualive: Representing Synchronized Visualization Interactions

Rupayan Neogy* Emily Hu† Arvind Satyanarayan‡

MIT CSAIL

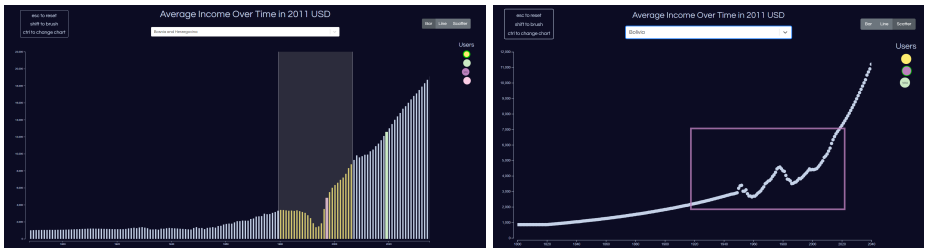


Figure 1: The Visualive synchronized visualization system. Left: several users are interacting simultaneously with the view, with the current user following another user as they brush and zoom into the visualization. Right: the viewing user has hovered over a remote user's bubble, resulting in the appearance of the purple tracking window that represents the remote user's screen area.

## Abstract

Visualization researchers have begun exploring real-time visualization synchronization, but the focus has been on the technology supporting such synchronizations, and very little research looks at the front-end. In this paper, we present Visualive, an exploration of the representation of real-time users in the space of a visualization. Visualive represents online users by embedding dynamic color-coded widgets within a visualization, allowing viewers to easily identify remote actions being performed without leaving the context of a visualization. Visualive is an initial dive into merging collaboration and synchronization with data visualization without compromising on the latter experience.

## 1 Introduction

Interactive data visualizations grow in prominence as an effective method for communicating information and provoking thought on the Internet. However, despite living on the web, few of these experiences take advantage of the interconnectivity afforded by the Internet. Visualive bridges the gap between disconnected visualization experiences by synchronizing interactions across the web.

The primary research goal of Visualive is to explore the representation of real-time users in the space of a visualization. How might viewing others' interactions influence a user's understanding or experience with a graph? Could users work together to transform or filter data to focus on novel collaborative discoveries? To what level should users be able to interact with others on any given visualization? While this paper does not aim to answer any of these questions in detail, it presents some thoughts on how Visualive's choices of user representation assists in cross-network collaboration.

Systems like ManyEyes [3] and Vistrates [1] aim to allow many users to view and work with visualization tools. Vistrates especially aspires to exemplify "ubiquitous analytics" [2], so that it is available to anyone from anywhere. However, in both of these systems, there is a major focus on implementation and functionality, and little discussion of the front-end techniques that enhance real-time user collaboration. Visualive intends to explore this underrepresented area of the design of collaborative visualization environments.

*rneogy@mit.edu
†emilyhu@mit.edu
‡arvindsatya@mit.edu

## 2 The Design of Visualive

In Visualive, several clients display visualizations that mirror changes made on any given machine in real time. In this way, Visualive presents a challenge in distributed state synchronization - each client must be aware of the correct visualization to render based on which of its peers the client should be synchronized with. The system includes a host of core interaction techniques. The user can filter by parameters, highlight data points or lines, zoom or pan around the chart, and brush an area of data. They can even switch between chart types, including bar, line, and scatter plots.

### 2.1 Terminology

*Tracking* - the action of one user viewing the results of another's interactions without synchronizing the two views completely.

*Following* - the action of one user synchronizing their view with another user's view. This results in one user giving up control over their own view until the decide to unfollow the other user.

*Viewing User* - the user that is currently being described. In Figure 1, the viewing user would be the one with the dark-purple color, viewing the interactions of the other users. In particular, the viewing user is *following* the yellow remote user.

*Remote User* - a user who is not the viewing user and is connected through the network. In Figure 1, there are several remote users, all performing a variety of interactions on the visualization.

### 2.2 Synchronizing Interactions

When multiple users access the website, each user is represented by a floating bubble in a side panel. Each user is assigned a different color on connection. Interactions are synchronized between users at three different levels: automatically, on hover, and on click.

*Sync Automatically* - Each time the chart type or data filter changes, the change is reflected across all users viewing the website. Highlighted data can be seen by all users at all times. The color of the highlight corresponds to the color of the highlighting user.

*Sync on Hover* - When the viewing user hovers over the floating bubble that represents a remote user, the viewing user is *tracking* the remote user. A rectangular outline, called the *tracking window*, is drawn on screen in the color of the remote user, which corresponds to the screen area of the remote user. The rectangle resizes and

moves on screen as the remote user zooms and pans. This allows the viewing user to see the remote user's location in the data.

*Sync on Click* - When the viewing user clicks on a floating bubble representing a remote user, they *follow* the remote user. This means that the viewer sees exactly what the remote user sees and does - each zoom, pan, and brush action is reflected from the remote user to the viewing user. The viewing user can desync by clicking the remote user's bubble again or by performing their own interaction.

Visualive provides several interaction types: highlighting, zooming, panning, brushing, and filtering. These interactions were chosen specifically to span a broad space of typical interactions on data visualizations. Some suggestions of additional interactions from informal user tests included sharing mouse cursor position and adding a chatting feature. Though some simple designs were created considering these additions, both were removed from prototypes due to scope and overlap of functionality.

While a permanent cursor might be useful in collaborative text editors such as Google Docs, Visualive aims to add the benefits of collaboration without hampering the main goals of data visualization. In this sense, a permanent in-chart representation of each viewer might prove to be distracting when attempting to parse and analyze a visualization. Due to this reasoning, Visualive renders no permanent representation of users in the visualization itself; instead, the user panel to the side contains gently oscillating bubbles to remind users of the availability of collaboration without interfering with the desired presentation of the data.

### 2.3 Levels of Synchronization

There are multiple levels of synchronization present in Visualive, as described above. Some interactions are always synchronized across clients, such as the data filter and the chart type, while others only synchronize when clients opt in through hovering over or clicking on a user bubble. The motivation to separate these levels of synchronization is two-fold.

First, some interactions can only exist in the context of other synchronized states: for example, a client cannot synchronize a highlighted bar representing the average income in the United States in the year 2000 if another client is viewing a line chart of the income in Paraguay. Of course, one could imagine some form of translating an interaction semantically between one chart type to another, but the difference in data displayed prevents any reasonable synchronization of the interaction. Due to this constraint, Visualive persists the chart data and type across all clients.

Second, and as mentioned in the previous section, Visualive's goal is to enhance visualizations with the benefits of collaboration without hampering their original purpose. By keeping certain synchronized interactions opt-in, users should receive a viewing experience that is desirable for them. Ignore the user panel, and the visualization is relatively isolated and left for the viewer to explore. Track or follow another user, and the visualization synchronizes with their remote interactions. Visualive leaves a certain amount of agency to the user to prevent the frustration that can arrive from overcrowded visuals, uncontrollable interfaces, and lag-filled experiences.

That being said, one could imagine an improved version of Visualive's synchronization level system where users can pick individually which interactions to track for each user. An implementation of such a feature would not require too large of an extension from Visualive's current structure, but it would need design work to incorporate into the interface. The benefit of adding this resolution of control is debatable, but providing the user with the choice to manipulate what they see is essential to the synchronized experience.

### 2.4 The Tracking Window

The tracking window, as seen in Figure 1, exemplifies the mindset of minimal interruption to the visualization experience while capitalizing on the possibilities of live synchronization. This tool allows users to perceive actions of remote users without interrupting their own workflow. The use of a rectangular boundary is the most intuitive representation of another computer screen, and allows a viewer to notice the focus of remote users at a quick glance without obfuscating the underlying data. As mentioned above, the tracking window is not *automatically synced* for all remote users, so that the main visualization is not obscured in the case of multiple tracking windows. The window only renders when a viewing user hovers over a user's bubble on the user panel, keeping the opt-in nature of the synchronization. Once the viewing user begins tracking, the window will continuously update with the remote user's zooms and pans, emphasizing the live collaboration possible with the system. From this point, the viewing user can click, to start *following* the remote user, or move their mouse away, returning to the base visualization with no tracking windows. Though no formal user tests were run to evaluate this feature, initial informal user feedback was positive, indicating that the tracking window effectively conveyed information without cluttering the visualization.

### 2.5 Implementation

Every time a user performs an interaction on a visualization, that client transmits the updated state information to the server through a socket event associated with the particular interaction. The server determines which clients need to receive that information (e.g. clients that are following or tracking), and then broadcasts the updated state to relevant parties. Clients listening for these events update their state upon receiving the information and let the rendering systems (React, D3) handle the rest. This system allows for clients to spend no computational power worrying about who needs to receive their updated state and instead offloads that completely to the server.

## 3 Future Work

Visualive has shown the possibility of a cohesive system that allows users to connect through the web in interacting with a visualization. As a venture into the space of collaborative data visualizations, Visualive has made it simple to share insights and data exploration.

There are several possible avenues of future investigation, many of which have already been mentioned previously. Some collaboration features that may be helpful for group participation are rooms, roles, and permissions, all which could make for more organized and directed user experiences. For example, a teacher might create a visualization and share the experience with their students, enforcing a stricter leader-follower relationship in the room. In contrast, a group of researchers from labs across the world might want to explore a large data set simultaneously. In this scenario, the users would like relative autonomy until they find an area of interest, at which point they could alert their peers to follow their view. Enabling such experiences could be exciting future work for Visualive.

There are also many more interaction techniques that can be experimented with. For example, a more robust data-filtering system could be interesting to track, so that users could apply multiple filters to the data, and followers could copy all or a subset of those selections. Another interesting concept could involve a "mini-map" of the visualization that always indicates the tracking windows without cluttering the main visualization. Finally, additional chart types and datasets can always be incorporated.

## References


[1] S. K. Badam, A. Mathisen, R. Radle, C. N. Klokmose, and N. Elmqvist. Vistrates: A component model for ubiquitous analytics. *Information Visualization*, 25(1):586–596, jan 2019.

[2] N. Elmqvist and P. Irani. Ubiquitous analytics: Interacting with big data anywhere, anytime. 2013.

[3] F. B. Viegas, M. Wattenberg, F. V. 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. doi: 10.1109/TVCG.2007.70577