This is an excerpt from Frank Elavsky's dissertation on *Tool-making as an Intervention on the Accessibility of Interactive Data Experiences*, which can be accessed in full at this archival link:

http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-26-103.pdf

This document contains the following sections:

- Abstract
- Table of contents
- Chapter 1: Introduction
- Chapter 2: Background & Related Work
- Chapter 3: Overview of Contributions
- Chapter 6: *Skeleton*: Visual Authoring of Non-visual Data Experiences
- Chapter 7: *Cross-perception*: Rethinking Input Design Towards Blind Analytical Interaction
- Chapter 9: Discussion & Future Work
- References

This document does not contain the following chapters:

- Chapter 4: *Chartability*: Heuristics as a Tool and Resource
- Chapter 5: *Data Navigator*: Low-level Tooling for Creating Navigable Data Structures
- Chapter 8: *Softerware*: Enabling Personalization of Interactive Data Representations for Users with Disabilities
- Chapter 10: Biographical Sketch

Contents

List of Figures

Abstract

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosytems.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part I

Introduction

Chapter 1
Introduction

This thesis is a body of research situated within the existing research area focused on making interactive data representations more accessible for people with disabilities. Much of the work in this existing area is situated within the context of making interactive data *visualizations* accessible, particularly (but not exclusively) for people who are blind. My work, contributed here in this thesis, is focused on using *tools* as a specific intervention and sub-area of study for making interactive data *representations* accessible for people with disabilities, broadly speaking. ("Representations" here is an intentionally broader term than "visualizations," which are exclusively visual representations of data.)

Before we begin, two things must be understood up front, or else the rest of this thesis could be interpreted with disruptive assumptions: we must interrogate the phrase "making visualizations accessible" and unpack why *tools* are a meaningful area of study.

1.1 Is "accessible visualization" really an oxymoron?

The first assumption that must be disrupted is perhaps the motivating cornerstone of this research, which is that the phrase "making visualizations accessible," while a noble goal, is not the semantically correct phrasing nor precisely what describes my work. This can be misleading. I do use the phrase "accessible visualization" but will admit that this seems to confuse certain people with very particular opinions about things. We will clear this up.

Villains of our field's past have written incendiary and ableist perspectives on why "no forms of data visualization, not just dashboards jam-packed with graphics, can be made fully accessible to someone who is blind," and that "[a blind man] will never be able to analyze data as I do visually, because many aspects of vision cannot be duplicated by his other senses" [44]. However, this position misunderstands what the goal of accessibility is, and arguably even what the goal of visualization itself is.

Making visualizations accessible *isn't* about the visualization, it's about making the outcomes of the visualization accessible.

Visualizations are ubiquitous and paramount for decision-making. However, the *artifact* that is a visualization is not even the goal of the act of visualizing: developing understanding, insight, confidence, and communication among and between human beings are the goals of visualization. Visualization is about making data easier to use for all kinds of things. Yes, our visual system enables us more than any other form of sensory cognition that we have [17, 52, 154]. But we aren't trying to make sight itself accessible. We are trying to make it possible for people to make meaningful decisions, gain valuable information, build conjectures, and effectively communicate with others.

Many, many people who I've spoken to over the course of my career, even before embarking on this thesis journey, misunderstand this simple fact: making a "visualization" accessible *isn't* about the visualization itself but rather making what the visualization is meant to *accomplish* accessible. It's about equal outcomes, not equal interactions with an artifact.

People with disabilities are no small portion of the world's population. In the United States, 27% of people self-report living with at least one disability that affects their daily lives [120] and all of us will eventually age into disability (if we are lucky to live a long life).

People with disabilities (again, that will be all of us *eventually*) deserve to participate fully in life. They deserve financial independence. They also deserve loving care and interdependence. People with disabilities have a right to make informed decisions, to know about the status of a global pandemic, and to have an understanding of local and national politics [42]. While we use visualizations to navigate all of these domains, the goal is not to make the charts and graphs themselves somehow equally useful to all people. That would be a false measurement of success.

Our goal then, is measured by the success of lives led by people with disabilities [160]. Many other measurements are just metrics along the journey towards that goal. We then ask: Can people with disabilities also use data to live full lives? Can they make *fast* decisions based on data? Meaningful, careful, *slow* decisions? Communicate complex ideas? Crunch and clean data, develop models, find errors, and build hypotheses? Can they have memorable, immersive, beautiful, aesthetic experiences with data too [77]? Making "visualizations" accessible really is a misnomer. We are ultimately trying to make everything about what interactive data experiences *accomplish* for people equitable and accessible.

Again, if the goal of accessible visualization were about visualizations themselves, then the correct course of action would be one framed by the medical model [108]: that there is a normative state of behavior and capability (in this case, it would be "normal" to be able to read a visualization and make a decision) and any deviation from that norm must be corrected. This framing first assumes that the visualization should not be altered or improved. And then this framing puts the burden on the bodies of people with disabilities: that they must be "fixed" and given sight or brought to some equivalent state as someone who is "healthy," normal, and sighted. Plenty of scholars have already discussed why this framing is a problem, not only because it places undue burden on people with disabilities, produces pathologies and hierarchies of disability, but also because it is fundamentally not economically or ethically feasible.

So we then turn to other models of disability, such as the social model. The social model is heavily discussed by disability scholars and is not the end-game or last and total way of thinking about disability [108, 126, 128, 152, 184]. But the core motivation is that society, not medicine, is also a path towards solving problems that people with disabilities face. A few important concepts and concretely actionable things come from the social model that can help motivate the work of this thesis.

First, we look to the historical birth of the social model of disability: in the 504 sit-ins that took place in the United States in 1977. Cities had curbs and curbs are a barrier for people who use wheelchairs. So protests happened because decisions were being made without people with disabilities at the table. In this instance, people acknowledged that political power was an exclusive club and fought to ensure their cry "nothing about us, without us!" materialized.

And this leads us to the first and most-foundational philosophical framing for this thesis: that our *artifacts*, these things we've created from curbs to data visualizations, can become *barriers* for people with disabilities. And it is then the artifact, not the body of the person with a disability, where disability is produced in this model. Rather than a comparison to a normative state as a way to frame disability (the medical model), we instead must observe and evaluate material outcomes based on human-made problems.

So, the social model is framed around society "solving" inequities: we get involved and make political and legal change tangible. But a second model also emerges from within the social model: one where we can now frame *who is first responsible* for repair: the curb designers and implementers.

And knowing who is first responsible for access leads us into the moral and ethical imperative that motivates this thesis: the builders and makers of visualizations are ultimately the ones who provide exclusive value for only a subset of people: those *without* disabilities. **We must first change how builders and makers do their work.**

So the phrase "accessible visualization" is really about recognizing that visualizations produce barriers for people. That means that it is our ethical imperative, as builders and makers, to fix them. And that act of fixing barriers leads us away from mere visual representations of data into a wide variety of other senses and interaction modalities. There are many paths forward towards fuller and more-equitable lives led by people with disabilities.

1.2 On *tools*, *tool-making*, and *human-tool* interaction

Then the act of making becomes immensely important: we, the builders and makers of our world, need to get things right; there is a risk involved when making things that we will exclude people with disabilities. We need to make sure that we build a better world than the one we have now. We must care for new things we create and tend to the repair and maintenance of what we've already made. And this ethical imperative leads us to the topic of *tools* and *tool-making*.

So the second thing that must be understood before we embark on this thesis is that *tools* are not the same as *solutions* or *applications*. Sometimes tools can be used to *solve* things and are certainly, in ideal circumstances, *applied* in various contexts. But understanding the role of the "tool" in human-tool interaction is paramount for engaging in the work of making anything accessible for people with disabilities.

We use tools to shape our world, break old things, and make new things. But a tool, like the hammer (as an example), does not inherently *solve* something like homelessness. But a hammer can be used to build homes if there are social policies in place and proper resources allocated. This means that for the success of tooling, there is often a larger material, social, legal, and policy reality that supports and necessitates those tools. This thesis will not be focusing on changing the upstream dependencies, but optimistically operating as if they were true (or will be true in time).

However, in some cases, tools can *destroy*. The hammer has a claw and can easily pry apart boards and tear down homes. So tools carry potential to do all kinds of things, both good and bad, and how a tool is used is often open-ended, variable, and heavily dependent on socio-technical realities. Tools participate in personal and political agendas [178] and are sometimes, for this reason, regulated or made proprietary and controlled by powerful entities [56, 171].

So tools are not without any sort of ethics. We cannot just blame tool-users for outcomes when much of a tool depends on these larger systems and structures. Technologies (tools included) encode the assumptions and biases of their *creators* as much as, if not more than, their users. Tools that build things for others to use can be loaded with assumptions about what people are *able* to do [179] and also rules and guardrails about what anyone downstream from that tool's design *should* do [56, 170]. These assumptions, biases, and rules *limit*, *enforce*, *magnify*, *exclude*, and *enable* what a tool-user is capable of.

Tools for visualizing data are a perfect case study in this problem: virtually every major data visualization library, application, or software ever made was made entirely with the assumption that data should be transformed into visual representations. This is a reasonable assumption, since virtually all of the tool-makers are sighted and visualization is incredibly helpful to our cognition of and communication with data [47].

So data visualization, as a field, has focused its tool-making efforts on reducing the difficulty involved in visualizing data. Some visualization tools are concise [133], others are lower level but much more expressive [11]. Tool-making in visualization has focused on making it easier to scaffold a wide variety of interactions both with the visualizations as well as with their underlying data models [67].

However, as time has moved on, people began to speak out about color-vision deficiency in data visualization. Some people, primarily those with X/Y chromosomes (largely men) who are of European ancestry, have a deficiency in their ability to perceive certain colors. Then a plethora of research arose that began to look into the barriers that folks who are colorblind face in data visualization. As a result, our practices and tools improved. We began to educate practitioners, develop new color palettes, researched new methods for testing our designs, and built new systems for handling automatic color encoding. Our tools evolved.

But now data visualizations have arguably become ubiquitous in daily life. By comparison, we have far more tools now for making visualizations quickly and easily than we do for representing data in non-visual ways. We also have far more research, relatively speaking, into how sighted end users interact with visualizations.

So this thesis engages gaps that arise in this space: Practitioners face immense challenges when crafting accessible data experiences. We first need to educate practitioners on what accessibility barriers actually are in interactive visualizations. Then, we must help them engage the hardest barriers in this work and create building blocks that help them to construct navigable data experiences, build design frameworks that can inform entirely new kinds of data interaction, and develop software systems for end-user personalization and agency. Our research seeks to advance the state of the art in tools that assist in accessible data interaction while also using tool-making as an intervention that helps us to better understand and characterize *why* and *how* data practitioners face barriers themselves in this work.

Chapter 2
Background & Related Work

2.1 Practitioners and Tools

2.1.1 Understanding Builders, Makers, Designers, and Developers

Research investigating the practices and experiences of individuals who create with computers employs a range of high-level methods. Ethnographic studies, case studies, and design ethnographies are common approaches, allowing researchers to immerse themselves in communities such as the DIY/maker and assistive technology spaces [76, 79]. These methods capture the nuanced challenges practitioners face when engaging in new and unfamiliar work, including the transition from traditional to digital fabrication, coding, and tool creation [78]. By observing and interviewing practitioners in naturalistic settings, researchers uncover the social, cultural, and technical factors that shape how makers adapt and evolve their work practices.

Participatory design and co-creation are also central to this field [153]. These approaches encourage collaboration between researchers and practitioners or end-users, enabling a deeper understanding of the cognitive and creative processes behind design and development [60]. Such collaborative sessions reveal how designers shift their thinking when encountering novel challenges, embracing iterative processes that blend experimentation with reflection. Similarly, developers often modify their applications, tools, and even programming languages through feedback loops and community-driven innovation, highlighting a dynamic interplay between individual creativity and collective knowledge.

Additionally, design-based and case-study research methods explore how new practices can augment the existing work of practitioners [22, 85]. This involves not merely filling gaps or solving isolated problems but reimagining the possibilities for creative and technical expression. Researchers in this space envision systems that support continuous learning, adaptation, and innovation [58]. The focus is on enabling practitioners to extend their capabilities—providing scaffolds for experimentation, fostering environments where unconventional approaches are encouraged, and integrating new technologies in ways that amplify creativity and intelligence rather than simply addressing deficits [89, 162, 179].

Overall, the research methods used in this area are multidisciplinary, combining qualitative insights with iterative design practices to offer a holistic picture of the challenges and opportunities of builders, makers, designers, developers.

2.1.2 Approaches to Tool-making in Human-Computer Interaction

In human-computer interaction, tool-making research spans both the creation of entirely new capabilities and the enhancement of existing systems. One prominent approach involves piggybacking on current systems—leveraging their established functionalities to introduce improvements that streamline workflow or unlock new interactions [57]. This method often focuses on integrating with widely used platforms to amplify their usability, enabling users to perform tasks in more intuitive or efficient ways. By building on existing infrastructures, researchers can demonstrate how small, targeted modifications have the potential to transform user experiences.

Another significant approach centers on the notion of appropriation [30, 31, 131, 159]. Here, research examines how users adapt tools for uses beyond their original intent. Studies in this vein explore the creative processes behind such re-purposing, uncovering the latent functionalities and opportunities that emerge when practitioners modify systems to suit their unique needs. This perspective often leads to the development of modular, extensible tools that encourage experimentation and user customization, fostering a more personalized interaction with technology. In some cases, theory has been developed from the study of emergent and generative tool-use [4, 7], broadly informing future tooling projects as well as general theories of creative human interaction with technology.

Beyond these, tool-making in HCI also includes the development of systems designed to empower users by providing entirely new capabilities, sometimes explicitly named "toolkits" and other times generally just referred to for their ability to enable novel interaction and outcomes [96, 122, 132, 145, 145]. These projects may range from novel software environments that facilitate rapid prototyping and live programming to innovative hardware devices that bridge the gap between digital and physical interactions [70, 72, 122]. The emphasis is not solely on problem-solving but on enabling creative exploration, new possibilties, and even hacking the potential of technologies towards new ends [71]. Such projects often present their contributions through demonstrative prototypes and case studies that reveal potential applications, even if they are accompanied by minimal formal evaluations [36].

This body of work reflects a balance between novelty and practicality. While some projects aim to introduce groundbreaking new ways to interact with data and systems, others refine existing practices to improve efficiency and accessibility. Together, these approaches underscore a commitment to enhancing human capabilities, allowing users to not only solve problems more effectively but also to unlock new avenues for creativity and innovation.

2.2 Data, Accessibility, and Data *and* Accessibility

2.2.1 Advancements in Interactive Data Visualization and Data Science

Recent years have witnessed significant advancements in interactive data science and visualization, driven by innovations that enhance both the performance and usability of data tools. Cross-filtering, as a subtype of cross-linked interaction, has emerged as a powerful technique, enabling users to interact with multiple data dimensions simultaneously [5, 66, 99, 167]. By linking various filters, analysts can quickly build hypotheses and isolate patterns, trends, and anomalies in complex datasets, leading to more informed decision-making. Stress has been placed in recent years on developing fast systems that are optimized showing more and more data at once while reducing latency in user interaction as much as possible [67, 99, 181].

Automated data processing and cleaning have revolutionized workflows by reducing the time spent on manual data wrangling [39]. Sophisticated algorithms now automatically detect inconsistencies, fill missing values, and transform raw data into usable formats. These improvements enable researchers and practitioners to focus more on analysis rather than preparation.

Faster tooling has further accelerated data exploration. Enhanced computational frameworks and optimized libraries allow for real-time data manipulation, making interactive visualization more responsive [67]. Coupled with easier-to-use grammars and scripting languages, these tools lower the barrier to entry, empowering users with limited visualization, geometry, trigonometry, data, and graphics coding experience to generate complex, interactive, visual representations of data [133]. New visualization types and techniques—ranging from dynamic dashboards, faceting, to immersive 3D visualizations—offer novel ways to explore and interpret data [181].

Despite significant breakthroughs, current advancements have largely neglected the needs of people with disabilities. Innovations in data science and visualization have focused on sighted user populations, prioritizing visual clarity and interaction speed using direct pointer techniques (such as with touch or a mouse) [109]. This focus often overlooks accessibility requirements for individuals who are blind, have low vision, experience cognitive or vestibular challenges, or possess motor disabilities that limit traditional pointer use [176].

2.2.2 Accessibility and Assistive Technology in Research versus Practice

2.2.2.1 Research: Focus on Blindness and Computer Output

Research and standards are both somewhat limited by a strong bias towards visual disabilities. In *Chartability*, 36 of the 50 criteria related to accessible visualization considerations involve visual disabilities [35, 42]. Marriott et al. also found that visual disability considerations are the primary focus of data visualization literature [109], leaving the barriers that many other demographics face unstudied. Accessibility research broadly has traditionally concentrated on the experiences of individuals who are blind, investigating how they perceive and interpret computational output [103]. Studies in this area explore alternative modalities for conveying data, such as auditory representations (through synthesized speech), tactile interfaces, and sonification techniques. Researchers focus on identifying effective methods for transforming visual data into formats that blind users can easily comprehend. This body of work not only examines the perceptual challenges but also delves into cognitive processing differences, aiming to optimize the accessibility of complex information and interactive systems for users with visual impairments.

While research has made strides in converting visual outputs into auditory or tactile forms for blind users, interactive input methods remain underdeveloped. Most efforts have concentrated on optimizing screen reader navigation and information retrieval, leaving text entry and command execution cumbersome. Screen readers, as they currently exist, offer limited support for efficient input, making it challenging for users to perform complex interactions. Although tactile interfaces hold promise for providing more intuitive input methods, they are still in the experimental stage and have not been fully integrated into mainstream accessible computing solutions, perpetuating a critical gap in effective user interaction.

2.2.2.2 Practice: Focus on Standards and Specialization

In contrast, practical accessibility efforts are often centered on the implementation and adherence to established standards and guidelines, such as WCAG [163]. There has been a growing interest in developing guidelines for practitioners [33, 35] and even applying guidelines as a method of validation alongside human studies evaluations and co-design [42, 101, 102, 188]. Existing accessibility standards bodies like the Web Content Accessibility Guidelines do stress the importance of accurate, functional semantics in order for screen reader users to know how to interact with elements [165]. For interactive visualizations this means that button-like or link-like behavior should expressly be made using elements that are semantically buttons and links.

Accessibility professionals, who typically possess specialized expertise, act as intermediaries between the design and development of digital products and the strict requirements of accessibility standards. Their role involves translating abstract guidelines into concrete design solutions, ensuring that websites, applications, and services meet regulatory benchmarks. By focusing on a standards-based approach, practitioners help organizations navigate the complexities of legal and technical requirements, thus ensuring that accessible design principles are integrated into mainstream technology development. This dual focus on rigorous standards and specialized expertise ensures that accessibility is both technically sound and legally compliant across diverse digital environments.

However, accessibility standards are inherently reactive, often lagging behind rapid technological advancements by five, ten, or even twenty years (or more). This delay occurs because developing, vetting, and formalizing standards requires consensus among diverse stakeholders and extensive testing to ensure compatibility and compliance. In contrast, cutting-edge interfaces and computational capabilities evolve swiftly, driven by dynamic market forces and user innovations. Consequently, accessibility guidelines tend to reflect outdated technologies, creating a persistent gap between modern interactive systems and current best practices in accessibility.

2.2.3 Data and Accessibility

In parallel to Mack et al.'s "What do we mean by Accessibility Research?" [103] nearly all topics of study at the intersection of accessibility and data are focused on visualization and vision-related disabilities [176]. Largely, access issues other than vision that affect data visualization (such as cognitive/neurological, vestibular, and motor concerns) are almost entirely unserved in this research space. Kim et al. found that 56 papers have been published between 1999 and 2020 that focus on vision-related accessibility (not including color vision deficiency), with only 3 being published at a visualization venue (and only recently since 2018) [94]. Marriott et al. found that there is no research at all that engages motor accessibility [109]. We have found 2 papers that engage cognitive/neurological disability in visualization and 1 student poster from IEEE Vis, which are all recent (specifically intellectual developmental disabilities [183] and seizure risk [150, 151]). We found no papers that engage vestibular accessibility, such as motion and animation-related accessibility. We also found that there is no research specific to low vision disabilities (not blindness or color vision deficiency) unless conflated with screen reader usage in data visualization. Blind and low vision people are often researched together, but in practice may use different assistive technologies (such as magnifiers and contrast enhancers) and have different interaction practices (such as a combination of sight, magnification, and screen reader use) [158].

Since the 1990s, the most prominent and active accessibility topic in data has been color vision deficiency in data visualization [20, 97, 111, 119, 121]. Research projects that explore tactile sensory substitutions to charts have been a topic in computational sciences dating back to the 1983 [136], with tactile sensory substitutions being used for maps and charts as far back as the 1830s [59]. Sonification used both in comparison to and alongside visualization and tactile methods for accessibility dates as far back as 1985 [14, 26, 46, 107, 112, 186]. Some more recent work has explored robust screen reader data interaction techniques [53, 149], screen reader user experiences with digital, 2-D spatial representations, including data visualizations [134, 141], dug deeper into the semantic layers of effective chart descriptions [101], and investigated how to better understand the role of sensory substitution [21]. Jung et al. offer guidance that expands beyond commonly cited literature that chart descriptions are preferably between 2 and 8 sentences long, written in plain language, and with consideration for the order of information and navigation [88].

A wide array of emerging research projects investigate screen reader users needs, barriers, and preferences, and offer guidelines, models, and considerations for creating accessible data visualizations [21, 42, 101, 141]. Jung et al. offer guidance to consider the order of information in textual descriptions and during navigation [88]. Kim et al. collected screen reader users' questions when interacting with data visualizations, which could open the door for more natural language data interaction [93].

Data visualization accessibility has come far in recent years. But little work has been done to explore what disability scholars call "access friction" - a tension that arises when access must be negotiated [62, 77]. This friction is often a result of static barriers in shared spaces: one artifact or approach designed to include some people may end up excluding others.

Yet despite these resources, making data visualizations more accessible remains a difficult task for practitioners [87, 143]. Some accessibility guidelines even conflict, for example on the topic of patterns and textures used in charts. One side stresses that patterns are harmful to cognitive and visual accessibility [135] while another stresses that redundant encoding strategies are necessary [35].

These difficulties point to a deeper problem: the tools practitioners use to build data experiences were not designed with accessibility in mind, and the resulting gaps are not evenly distributed. Even in an ideal state where guidelines agree, the hardest remaining challenges cluster in three domains: navigation, interaction, and personalization. In these, practitioners lack the structural, technical, and infrastructural means to reason about accessible design and act on those considerations.

Chapter 3
Overview of Contributions

In this dissertation, we contribute practical advancements in tool-making as an intervention on the accessibility of interactive data experiences. The thesis of this dissertation is as follows: *This dissertation argues that the tools practitioners use to build interactive data experiences are themselves sites where accessibility barriers are produced, prevented, or alleviated for both end users and authors. This work contributes five tools—Chartability, Data Navigator, Softerware, Cross-perception, and Skeleton—that collectively center accessibility work on the empowerment of disabled and non-disabled practitioners across the full arc of evaluation, data navigation, analytical interaction, and personalization.*

Rather than framing accessibility research solely around ideal experiences for end users with disabilities, this thesis investigates why accessibility work is so difficult for the practitioners who build interactive data experiences and what tool-making can reveal about those difficulties. We organize this investigation around four domains where practitioners face the most persistent challenges: evaluation, navigation, interaction, and personalization. *Chartability*, a heuristic framework contributed first and maps the full landscape of accessibility barriers and identified these three latter domains (navigation, interaction, and personalization) as the areas where the most severe gaps remain. *Data Navigator* and *Skeleton* then investigate navigation, finding that practitioners struggle because navigation structure has no visible, manipulable representation in their workflows. *Cross-perception* engages interaction, demonstrating that blind data analysis has been constrained by existing tools and that a new interaction design framework can reshape what analytical work is possible. *Softerware* addresses personalization, revealing that access needs genuinely conflict across users and that meaningful personalization requires system-level infrastructure that does not yet exist in practitioner tooling ecosytems.

We engage each domain below with the questions: "Why does this work matter?", "Why is it hard?", and "What has tool-making within this domain showed us?"

The 4 domains of work that I engage in this thesis start first with **evaluation**. In work contexts where someone is designing and developing interactive data experiences, the practitioner must have the knowledge, tools, and resources available to systematically identify how their interfaces produce barriers for people with disabilities. A significant portion of professional accessibility work (arguably most, if not all) is founded on auditing and evaluating barriers to access. This work is pre-dominantly done through a standards-based approach [163], although in more robust evaluation work, people with disabilities are actively involved in the process [123].

Evaluation is difficult work because much of it is contextually defined by the author themselves, and most tasks at this intersection require careful, non-automated processes and methods [29, 123]. To make matters more difficult, no comprehensive guidelines, tests, and tools exist in any singular location. Practitioners often must gather these resources themselves, which tend to be situated towards accessibility in general or are high-level and provide minimal usefulness in practice. Additionally, practitioners themselves often have little knowledge about the veracity or quality of any given bit of information they gather [87, 143], and often do the work themselves to synthesize this disparate space of information into a usable format they can apply to their own evaluation work.

To engage this, our first main chapter focuses on *Chartability*, a heuristic framework that enables designers, developers, and auditors to systematically evaluate data visualizations and interfaces for a wide range of accessibility barriers, considering people with visual, motor, vestibular, neurological, and cognitive disabilities. In this project, we did the hard work for other practitioners and contributed our collection of synthesized accessibility resources in a single workbook. We had practitioners try out our resource in real environments, in-situ, in order to learn more about the challenges and barriers they themselves faced in evaluation work. With *Chartability*, practitioners, especially those with limited accessibility expertise, gained more confidence and clarity in assessing and improving their work. Additionally, *Chartability* has since become widely applied as a framework that isn't just used for evaluation but also as design guidance in many contexts, internationally, including policy organizations, governmental groups, and more than 100 companies and businesses.

Chartability then opened up a significant landscape of new projects and research directions. From a combination of my existing expertise as a visualization designer and engineer, in addition to continued application of *Chartability* in the wild, we began to identify the trickiest and most-difficult domains of work for practitioners. Chartability has 50 total heuristics, or tests, each organized under one of 7 principles. But 3 larger domains began to emerge as the areas where the most severe and dramatic accessibility barriers remained unaddressed: on data *navigation*, analytical *interaction*, and interface *personalization*.

So the next section of this thesis engages the first of these three: **navigation**. Navigation is a fundamental type of interaction that is leveraged by modern software-based assistive technologies. Screen readers, the primary tool used by people who are blind to interact with computers, navigate content. Additionally, many other assistive technologies, such as a sip and puff device (like the "POSSUM" from as far back as '63 [105]) also navigate. Navigational technologies are leveraged by people with a significant array of disabilities, yet tend to be entirely ignored by existing data visualization tools, which are pre-dominantly built to support direct input (using a computer mouse).

Empirical work has already demonstrated that structural navigation is actually good [188], even when regular alternative text (image descriptions) exist. This is both because people who are blind can gain both a high level understanding (from the description) as well as lower-level sense of the data's structure and arrangement, in addition to the fact that discrete, structural navigation exposes interactivity that may exist on any visualization elements (such as they can be hovered or clicked with a mouse in order to perform some action). So, if good empirical work exists: *why haven't practitioners put this research into action? What makes this work hard to do?*

We first built *Data Navigator* to provide the building blocks we needed in order to address the technical and conceptual gaps that were required to make any visualization or visualization tool provide a navigable, interactive structure. *Data Navigator* is a low-level toolkit which can be used to construct accessible navigation structures such as lists, trees, and diagrams from an underlying graph structure. We leveraged graph theory for an applied HCI problem: nodes and edges represent any relationships within the data as a structure, which then supports rich expressiveness of data navigation experiences. Users can navigate discrete marks in a visualization, clusters, groupings, and more. In addition to its structural scaffolding, *Data Navigator* also supports a wide array of input modalities leveraged by people with disabilities (screen readers, keyboards, speech, and gestures).

Data Navigator provided a substrate, but this contribution alone wasn't enough to engage the question *why is navigation so hard, in practice?*. So the chapter following *Data Navigator* introduces *Skeleton*, a data navigation authoring tool built on top of *Data Navigator*. Our novel approach in *Skeleton* involves visualizing and making manipulable the nodes, edges, and textual data that comprise non-visual end user experiences. *Skeleton* visualizes the building blocks that comprise *Data Navigator*. Additionally, *Skeleton* provides expressive, rapid scaffolding capabilities that leverage data visualization rendering engines. This scaffolding engine helps practitioners quickly create common configurations for non-visual data navigation structures that retain visual congruence to the underlying structure.

But most importantly, *Skeleton* serves as a framework that shapes designerly consideration. Our conjecture was that because sighted practitioners cannot *see* navigation building blocks, they will not treat those elements as iterable design materials. We conjectured: Navigation is hard in practice because sighted designers face barriers to iteration and understanding. We conducted an empirical study with sighted practitioners and found that making non-visual elements visual helped practitioners shift from treating accessibility as a compliance task to treating it as a design problem, re-iterating on the visual aspects of their design, and engaging in the complex and nuanced components that comprise data navigation experiences.

Now, **interaction** becomes the next area we wanted to engage. Existing accessible data interaction for people who are blind, including our previous work on navigation (which is a form of interaction), predominantly seeks to expose information. This is what we call *access-oriented interaction*. In terms of low-level analytical tasks, most are then made feasible through navigation, sonification, or summarization-based and question-answering approaches: retrieving values, filtering, computing derived values, sorting, determining ranges, clustering, and finding outliers. What remains are analytical tasks that, despite being "low level" (understood as *unable to be reduced into more fundamental tasks*), are cognitively highly complex: finding correlations and characterizing distributions [3]. These tasks require complex hypothesization and exploration, rather than a system that simply encourages surfacing what is known or what is already present in the data: it requires combining, remixing, restructuring, and dividing data.

But blind *analytical interaction* isn't just important to engage because it is understudied, it is important to engage because many of interactive information visualization's most impactful tools for data science enable it [46, 67, 117, 167]. In visualization, *cross-filtering* is one example of an interaction that enables a user to filter one visual space while seeing a coordinated change in another visual space simultaneously and near-instantaneously. The speed of input interaction and perception of output also matters: even a small bit of latency changes the quality of a user's data exploration activities [99]. We conjectured that a screen reader, the most-used tool leveraged by blind people when interacting with computers, may be insufficient for engaging this task.

To engage this, this thesis introduces *Cross-perception*, an approach for building analytical interactions that support perception in one space of input interaction with simultaneous, non-competing perception of output in another space of data representation. We first formalized a design framework for producing *cross-perception* experiences and then built a novel prototype device, the *cross-feelter*, that enables blind *cross-perception* of a cross-filtering data exploration interface. In an empirical study with blind users (with and without existing data expertise), we found *cross-perception* speeds up analytical exploration by 90% and helps blind users consider vastly more questions of their dataset (+188% computational queries, +54% spoken aloud) compared to a screen reader-driven interaction.

Beyond performance, we found that our input modality itself shaped the character of analytical engagement: participants didn't just work faster, they considered more dimensions of their data and asked qualitatively different questions. The *cross-feelter* also reduced anxiety and substantially increased enjoyment, particularly for participants without prior data expertise, suggesting that the barriers blind practitioners face in data work are not only functional but affective. Additionally, we had our blind practitioners imagine new interaction possibilities that *cross-perception* could enable including and beyond our *cross-feelter* device.

Our final domain of work is the most difficult for visualization practitioners to engage: **personalization**. While *navigation* demanded better software tooling and visual support and *interaction* required new hardware, *personalization* completely re-orients how software authoring takes place. Personalization matters because of *access friction*, which is a design challenge where one design or interface configuration that might be accessible for one person or group of people turns out to create barriers for someone else [62, 77]. In existing work on personalization and accessibility, studies have demonstrated that end-user control is great to have and can alleviate friction [86, 182], but little work has been done to explore what personalization looks like for an existing data visualization library and how practitioners should build and maintain their existing systems to support it.

Our final chapter introduces *Softerware* to address the tension between standardized accessible design and the diverse needs of real users with disabilities. In the wild, *access friction* exists in every design that reaches a public audience; it is inevitable. This tension ultimately means that some users have a worse experience, and may even face exclusion, with any particular design configuration. So for this work, we conducted our research in-situ with visualization software engineers and designers and worked to build a scalable, flexible software system dubbed "softerware" that enables end users to manipulate the appearance and functionality of the charts and graphs they encounter according to their own preferences. We conducted empirical research to inform our collaborators, as well as other visualization system authors, with guidelines and considerations for building *softerware* systems. In our study, no two participants chose the same preference configuration and participants with the same diagnosed condition sometimes needed opposite design treatments. Practitioners, meanwhile, immediately raised ethical concerns about whether personalization would let designers off the hook for poor defaults. These findings revealed that the real barriers to personalization are not at the level of any individual chart but at the level of system infrastructure: without persistence, cross-system interoperability, and shared standards, the effort required of end users exceeds the value they receive.

Combined, these contributions provide empirical insights and practical advancements in the state of the art for tooling that bridges gaps in current accessibility practices in visualization and data science. Our work ultimately enables people with and without disabilities to better evaluate barriers in, analyze with, design for, develop, and personalize interactive data experiences. We demonstrate that tool-making is a productive intervention that both engages accessibility barriers and elucidates why those gaps exist in practitioner work.

Part III

Navigation: Making Data Structures Traversable

Chapter 6
Skeleton: Visual Authoring of Non-visual Data Experiences

This chapter was adapted from my paper, currently under review with IEEE VIS:

> F. Elavsky, C. Nnadozie, L. Nadolskis, P. Carrington, and D. Moritz, '*Skeleton*: Visual Authoring of Non-visual Data Experiences', *IEEE Transactions on Visualization and Computer Graphics*, 2026.

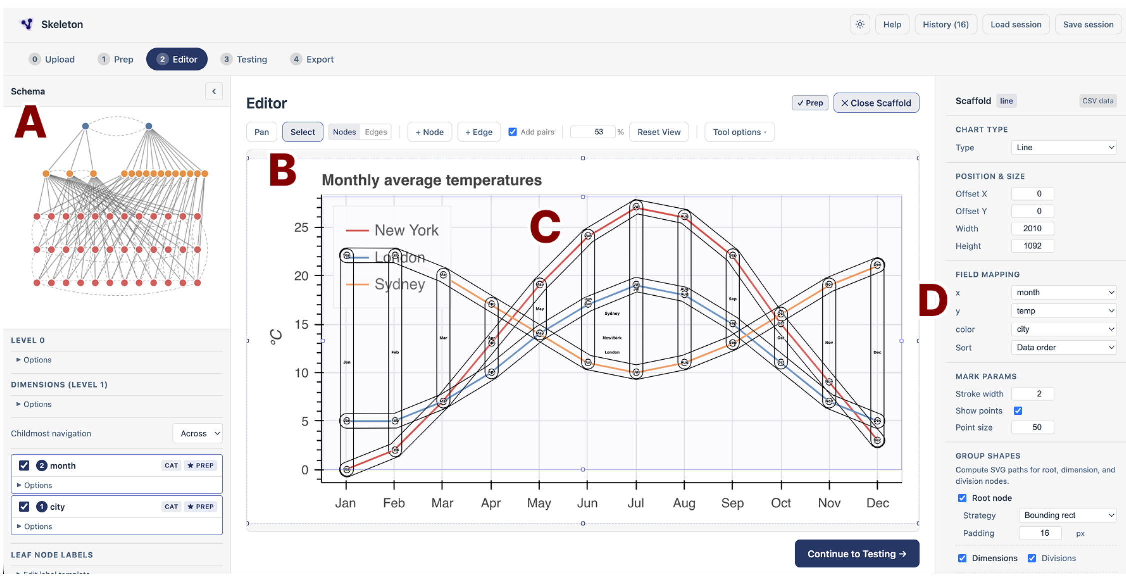

Figure 6.1: Low-fidelity design draft of *Skeleton*'s main user interface components and interactions. A. *Skeleton*, our graphical user interface for creating and debugging screen reader navigation experiences of data visualizations. B. Users can add nodes wherever they want over the chart, manually or automatically with algorithmic assistance. C. Users can then "draw" edges between nodes, which signify navigation paths through the visualization.

6.1 Abstract

When sighted practitioners author accessible data visualizations, they build navigation structures (the nodes, edges, and input bindings that govern how assistive technologies traverse an interface) entirely in code, with no visual representation. This invisibility makes navigation structures difficult to inspect, debug, and iterate on. To sighted practitioners, every other aspect of a visualization is iterated on because it is visible; navigation structure ships as a first draft, if at all,

because it is not. Without a representation to react to, practitioners cannot develop judgment about what makes navigation good or bad, and the quality ceiling of non-visual experiences is set by the absence of a feedback loop. We address this problem through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, and make three contributions. First, we introduce technical advancements for making the properties of accessible navigation structure visible and directly manipulable during authoring, grounded in two foundational pieces of infrastructure produced by our co-design work: an *Inspector* that renders navigation graphs as interactive node-link diagrams, and a *Dimensions API* that expresses navigation in terms of data dimensions rather than explicit graph construction. Second, building on these, we present *Skeleton*, a direct-manipulation authoring environment in which the properties of an accessible navigation structure are translated into visual representations authors can observe and manipulate. Key techniques include a dual-view editor that simultaneously shows the system's navigation model and the end user's spatial experience, a scaffolding engine that automates spatial node placement by repurposing a visualization rendering pipeline, a live label-template editor with real-time screen-reader-output preview, and a testing mode that makes traversal sequence visually trackable. Third, we evaluate *Skeleton* through an in-situ study with 8 practitioners across visualization design, engineering, and research. Making navigation structure visible changed how practitioners engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem.

6.2 Overview

We start this work with a provocation: How might the discipline of visualization help the discipline of accessibility?

Visualization has spent decades developing techniques for a specific class of problem: representing and interacting with information visually. Grammars of visual encoding [113, 133, 169, 172, 189], direct-interaction interfaces [37, 80, 146], and iterative feedback loops between representation and understanding [45, 66, 99] are all methods that enable abstraction and manipulation of the information that underlies the visual representation. We argue these methods have a direct application within the discipline's own accessibility challenges, one that has not yet been explored.

Sighted practitioners who build accessible data visualizations face an unusual authoring problem. The non-visual navigation structures they construct (the nodes, edges, focus states, input bindings, and semantics that govern how assistive technologies traverse a chart) exist only as code. A practitioner can write a navigation hierarchy, but cannot see it, click on a node to inspect what will be announced, observe the spatial relationship between a navigation path and the chart it overlays, and then manipulate its properties through direct interaction. Every other aspect of a visualization has a visible, inspectable representation during authoring: the visual encodings are visible, the layout is visible, the interaction states are visible. And yet, navigation structure is not.

This invisibility has practical consequences. Without a way to see what they are building, sighted practitioners cannot easily catch structural errors, compare design alternatives, and iterate. Accessibility becomes downstream of every other design choice, not because practitioners choose to deprioritize it, but because the authoring conditions do not support anything else [87, 143]. The floor and ceiling of non-visual data experiences are constrained by what sighted authors can perceive of their own work.

We engage this space with the following research questions:

R1 (Qualitative, Exploratory): What challenges do sighted practitioners face when designing and engineering navigation structures for accessible visualizations?
R2 (Qualitative, Exploratory): How do sighted authors reason about the non-visual experiences that accompany their visualizations?
R3 (System, Design): How can we make the properties of accessible navigation structure visible and directly manipulable during authoring?
R4 (Qualitative): In what ways does a directly manipulable visual representation of navigation structure change how practitioners find errors and improve upon their designs?

We address these questions through longitudinal co-design with practitioners across cartography, design systems, and open-source visualization, following an action research orientation [65] in which the research team was embedded in each community's active development work. This paper makes three contributions:

First, **we introduce technical advancements** for making the properties of accessible navigation structure visible and directly manipulable during authoring. These techniques are grounded in our co-design collaborations, which produced two foundational pieces of infrastructure: an *Inspector* that renders any navigation graph as an interactive node-link diagram, and a *Dimensions API* that formalizes a declarative grammar for expressing navigation in terms of data dimensions rather than explicit graph construction (**R1**, **R2**, **R3**).

Second, building on this infrastructure, **we present *Skeleton*, a direct-manipulation authoring environment** in which the topology, spatial mapping, semantics, and input logic of an accessible navigation structure are made visible and directly manipulable. *Skeleton* is built on Data Navigator [36], a code-based library for constructing interactive data navigation structures (**R3**). Our intention with *Skeleton* is to continue to develop it towards a usable, practical system beyond the scope of this research.

Third, **we contribute findings from an in-situ interview study** with 8 practitioners across visualization design, engineering, and research. We evaluate Skeleton as a design probe [61, 81] rather than a deployable system, focusing on qualitative shifts in practitioner engagement rather than task performance. We find that making navigation structure visible shifted how participants engaged with accessible design: they reconsidered the architecture of their own visualizations, attended to a broader range of input modalities, and shifted from treating accessibility as a compliance task to treating it as a design problem (**R1**, **R2**, **R4**).

6.3 Related Work

6.3.1 Non-visual Data Experiences

Blind people who rely on assistive technologies interact with data in fundamentally different ways than sighted users who use a direct pointer, like a mouse [94, 109, 141]. A substantial body of research has documented what these experiences look like across modalities, and what it takes to make them meaningful. In the context of "data experiences," this paper focuses specifically on interactive navigation structures, but we briefly survey adjacent modalities to situate our contribution.

Alternative text and natural language. A dominant strand of this work concerns the generation and evaluation of textual descriptions of visualizations [88, 91, 95, 101]. More recent LLM-driven systems and Q/A approaches can caption charts with varying degrees of semantic depth, some at a risk of producing bias [34, 43, 93]. While alt text makes a visualization's message available without sight, it is by nature static: a description conveys what a visualization says, but not how a user might explore or interact with it.

Sonification, haptics, and tactile rendering. Non-visual data experiences extend well beyond text. Sonification encodes data as sound [14, 69, 107], with declarative grammars emerging for authoring these experiences [92]. Haptic and tactile representations offer another channel through refreshable displays, 3D-printed graphics, and multimodal touchscreen interactions [18, 74, 110, 127]. Recent systems integrate multiple non-visual modalities around a single data representation [21, 75, 138].

Interactive navigation structures. The primary focus of our work centers on the state of research related to structured navigation: the traversal of data points, groupings, and interface elements through assistive technologies and keyboard input [36, 149, 166, 188]. Existing systems and interfaces have demonstrated that going beyond static descriptions to support hierarchical, traversable data structures meaningfully improves how blind users can explore and reason about charts [161, 188]. Giving users control over the textual tokens surfaced during navigation improves comprehension and agency [86]. And more recent work has found that perceptually congruent navigation structures for charts and diagrams can improve goal-driven exploration [114].

6.3.2 Authoring Non-visual Data Experiences

Accessible visualization has historically centered the experiences of disabled users, but a parallel and increasingly urgent body of work examines the experiences of the people who build these experiences: visualization designers, engineers, and researchers.

Practitioner challenges. Research consistently finds that sighted visualization practitioners struggle with accessibility [42, 87, 143]. Most visualizations in the wild are inaccessible and designers themselves report lacking guidance, especially for complex and interactive graphics [87, 143]. And screen reader users experience the downstream effects of these gaps: inconsistent structure, poor keyboard support, and information that is present visually but absent in the accessibility tree [42, 141]. The pattern across this work is consistent: the practitioners who build visualizations lack the tools and feedback mechanisms to make non-visual experiences effective, useful, and good.

Across practitioner-centered literature, a recurring finding is that non-visual experiences are treated as downstream of visual design choices, added after the visual representation is finished rather than designed in parallel [87, 102, 143, 190]. This sequencing has consequences: what is navigable and how it is structured is constrained by whatever visual decisions came first.

Authoring-oriented systems. A distinct line of work has focused on building authoring tools and libraries that give practitioners more tractable paths to accessible output. Few visualization tools support the kinds of interactive navigation structures that assistive technology users most benefit from [95]. Of those that do, most rely on code-based approaches [10, 36, 140, 142]. *Umwelt* [190] takes a different and notable approach: it is a structured editing environment where authors specify representations *across* modalities (sonification and visualization) in an integrated interface, where navigation is made available over the visualization using *Olli* [10]. The latest work in this space is *Arboretum*, a tool that provides automatic conversion of diagrams to a tabular structure, navigable structure, and tactile representation [177]. In *Arboretum*, the input visual is treated as the ground truth for the navigation structure, which is treated as downstream output.

Communicating visually, authoring invisibly. There is a revealing irony across this body of related work: research about navigation structures almost invariably communicates those structures visually. Papers such as "rich screen reader experiences" [188], *ChartReader* [161], *Benthic* [114], and *Data Navigator* [36] each use visual node-link diagrams and hierarchical schematics to explain navigation paths to their readers. The same pattern holds in adjacent domains that involve structuring data for navigation, such as PDF remediation [118].

And in authoring-oriented systems, none provide a visual interface through which practitioners can interactively inspect and manipulate navigation structures as a first-class design material. Structure output is either downstream of code or static visuals. Structure is always *derived* or *specified*; indirectly manipulated. Additionally, verification of the structure across all of these systems requires developers to manually navigate using a screen reader after the structure has been authored and rendered, before returning to the upstream visual design space or code.

Navigation structure is understood and communicated visually by sighted researchers and designers, yet built entirely without visual feedback by developers. We seek to address this gap.

6.4 Co-design Foundations

Our research follows an *action research* orientation [65], in which knowledge is generated by engaging with a community to solve a real problem in-situ, alongside them rather than studying it from the outside. These collaborations started from a shared motivation: practitioners needed to make their existing systems accessible to navigational assistive technologies, using Data Navigator [36] as a foundation. Across three projects, we worked with 12 individuals outside our research team. Three blind co-designers shaped the work throughout: CD1 and CD2 (anonymous) and a co-author on this paper, [REDACTED]. CD1's contributions are noted in Section 6.4.1, CD2's are noted in Section 6.5.3. [REDACTED] contributed to early ideation, problem formation, and framing for the project as a whole, helping define the authoring challenges that *Skeleton* addresses, in addition to feedback on study design.

The co-design literature on accessibility has centered people with disabilities as primary design partners [26, 27, 124, 161, 187, 188], and rightly so. Our co-design takes sighted practitioners as primary partners because the authoring challenges we address happen on the sighted side of the process: it is sighted authors who cannot see what they are building.

Two themes converged across all three engagements, and we present them here before describing the individual projects that surfaced them. First, **practitioners communicated and reasoned about accessible navigation using visual representations**: while designing for language, sound, and structure, our collaborators drew on paper, built wireframes of nodes and edges, and reflected on the design space using visual artifacts. Even when collaborating with blind co-designers, a visual medium was the first language of sighted authors. Second, **development that followed visual design work faced severe iteration barriers**: verifying a navigation experience required building a working code prototype and manually navigating it with a screen reader. The gap between visual design and code-based scaffolding with manual testing produced repeated mistakes, misinterpretations, and abandoned prototypes. Each project below contributed distinct evidence for these themes and surfaced specific requirements that shaped our infrastructure and tooling.

6.4.1 Geologic Map

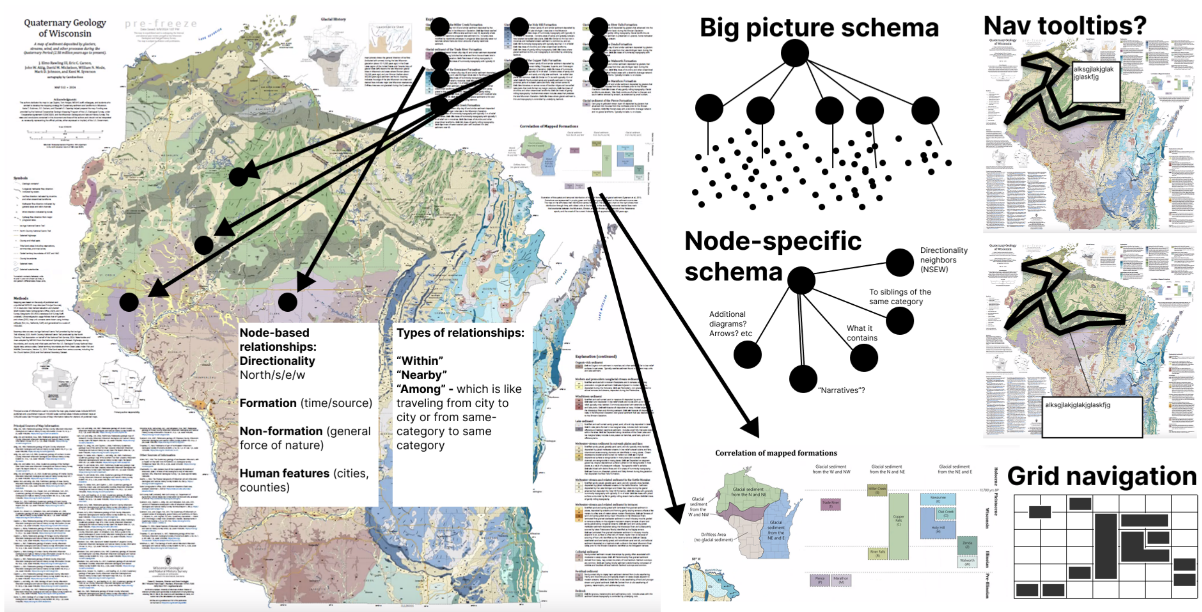

Figure 6.2: Our visual design work in Figma over a static geologic infographic map of Wisconsin. We use visual forms and illustrations over and beside the map to communicate flows, structure, navigation styles, and interaction patterns.

Our longest engagement spanned just over two years, with a cartographer, accessibility consultant, and blind co-designer (CD1) building an interactive version of a quaternary geologic map of Wisconsin [125]. The map combined dozens of irregular geographic regions organized categorically by a legend.

Early design sessions took place in Figma (Figure 6.2), where we laid out node-link diagrams of how a screen reader user might traverse the map, legend, and peripheral information. We annotated each node with text a screen reader would announce and connected them to relevant regions. Drawing the structure made it possible to discuss design choices, catch dead ends, and debate traversal strategies. We were informally doing what *Skeleton* later formalized.

The friction began when we moved from design to implementation. We had no way to verify that our designed structure would be good or ideal, and scaffolding the project into Data Navigator was arduous: the translation from even simple navigational designs to functional code was too complex to hand off or meaningfully iterate on. The collaboration also surfaced a design-space boundary: for within-map spatial navigation, an egocentric audio-game approach [9] fit better than a node-link graph, a distinction CD1 helped surface. Reaching this boundary was only possible because we could see the structure we were trying to build.

6.4.2 Design System Library

Our second engagement, spanning 7 months, was with 6 engineers and designers on Adobe's React Spectrum Charts library, an open-source chart component system. We worked in their codebase and in Miro on navigation design for bar charts, clustered and stacked bars, line charts, and related types. Because we needed generalized, reusable patterns, our design problems differed from the geologic map: we needed to account for use, re-use, and edge cases across chart types.

Our Miro sessions produced two kinds of artifacts: diagrams of navigation structure for specific chart *instances* and *schema* diagrams capturing generalized patterns in dimensional terms. The concept of a "dimension" emerged naturally: common transformations on Data Navigator's graph structure corresponded to properties within a dataset, such as a "categorical" dimension or a "numerical" dimension, where traversal took place within collections of grouped siblings. This dimensional thinking directly foreshadowed the Dimensions API (Section 6.4.4).

The dominant friction was iteration speed. We could sketch and converge on a schema in Miro in an hour, but verifying the design in a functional example required embedding Data Navigator into a large codebase, implementing changes, rebuilding, and manually testing with a screen reader. The collaboration also surfaced a limitation: mobile screen reader navigation uses swipe gestures rather than keyboard input, and our keyboard interaction model did not account for it. Together, these gaps made two requirements clear: a higher-level navigation abstraction and a visual tool for inspecting structures without a screen reader or a fully integrated build.

6.4.3 Open Source Visualization Library

Our third collaboration, spanning nearly two years, was with Quansight Labs and contributors to Bokeh, a Python-based open-source visualization library. We performed an accessibility audit [40] based on Chartability [35] and identified that Bokeh visualizations with interactive chart elements needed to be navigable by assistive technologies.

Unlike Adobe, Bokeh has no native chart types. Its API operates at the level of glyphs, renderers, and data sources, which users assemble freely. There was no standard unit around which to anchor a navigation pattern, and any grammar or tooling would need to accommodate an open-ended range of encoding combinations. The iteration gap from Adobe was present in a more severe form: making library-wide contributions to a fully open-source project required incremental tooling for the most common cases. For Bokeh, we needed to test functional, data-driven navigation abstractions without fully embedding Data Navigator into the library.

6.4.4 Infrastructure from Practice

The three collaborations converged on two concrete requirements: a way to visually render and inspect navigation structures, and a higher-level abstraction for specifying navigation without hand-wiring every node and edge. We built two pieces of infrastructure to address these requirements. Both were used in subsequent co-design work and became the foundation on which *Skeleton* was built.

6.4.4.1 An Inspector Gadget

We built an *Inspector* (`@data-navigator/inspector`) to render any Data Navigator structure as an interactive node-link graph using D3, with an accompanying console for debugging. Hierarchical structures are colored by level; edges are drawn as directed links; the entry point is visually marked. The *Inspector*'s graph can itself be navigated using Data Navigator, with visual focus tracking during navigation, allowing practitioners to manually verify structure and reachability.

This made structural verification immediate: a practitioner could generate a structure and check at a glance whether the hierarchy had the right levels, whether circular extents produced expected wrap-around edges, or whether a particular path was reachable. The interactive console logs API information and underlying data when nodes are activated, and hovering or focusing logged information highlights the corresponding node in the graph.

The *Inspector* remains a developer tool, however. It requires code familiarity to attach and renders structure as an abstract graph with no connection to the spatial layout of the underlying visualization. It shows topology but not instantiated geometry: a practitioner can see that two nodes are connected but not where their focus indicators will appear on-screen. This gap between navigable structure and its spatial instantiation over a rendered chart motivated *Skeleton*.

6.4.4.2 Alternative Dimensions

The original Data Navigator library requires explicit graph construction: practitioners specify nodes, edges, and navigation rules by hand. This is general but scales poorly.

The *Dimensions API* introduces a declarative abstraction one level above this. Rather than specifying the graph directly, a practitioner describes the *dimensions* of their data, the meaningful axes along which a user might want to navigate, and the API constructs the full node-link structure automatically. A dimension has a type (`categorical` or `numerical`), a data key, and behavioral properties governing traversal. Two properties are central: `extents` determines boundary behavior (`terminal` stops at edges; `circular` wraps around), and `childmostNavigation` determines whether leaf-level nodes are reachable laterally across a dimension's divisions without first returning to a parent.

The generated structure is a multi-level hierarchy: each dimension produces a root node, below which division nodes group the data, below which leaf nodes represent individual data points. Multiple dimensions over the same dataset share leaf nodes, so users navigating via different dimensions reach the same data through different paths. With the Dimensions API, bar chart navigation that would otherwise require constructing every node and edge by hand is expressed as a single dimension declaration:

```
dimensions: {
  values: [
    {
      dimensionKey: 'month',
      type: 'categorical',
      behavior: { extents: 'circular' }
    }
  ]
}
```

The abstraction is chart-type-agnostic: bar charts, scatter plots, line charts, and layered charts all use the same vocabulary, with different combinations producing different navigation topologies. This directly addressed Bokeh's problem: the API mirrors data fields and encoding choices as a set of dimensions.

6.5 *Skeleton*: System Design

With a visual structure renderer (the Inspector) and a declarative abstraction for producing navigation structures (the Dimensions API), the remaining problem was to bring these capabilities into an integrated, direct-manipulation authoring environment where practitioners could not only see structure but manipulate it interactively, test it with real input, and iterate on it with immediate feedback. *Skeleton* is that environment. It integrates the Inspector and Dimensions API into a unified workflow and extends both with a guided preparation phase, a spatial rendering canvas, and a live testing mode. Each of Skeleton's authoring techniques makes visible a specific property of non-visual interaction that sighted practitioners otherwise cannot see during authoring: the topology of what is navigable, the spatial mapping of where navigation lives over the chart, the semantics of what is announced, and the temporal sequence in which a user encounters nodes.

Where the Inspector requires a practitioner to write code first and visualize after, *Skeleton* reverses the direction: practitioners design a navigation structure visually and inspect the code representation as a consequence of that design.

6.5.1 Staging: Input and Preparation for Authoring

The authoring workflow proceeds through four stages: upload, prepare, edit, and test. Making a visualization accessible involves decisions about what is navigable, how navigation is triggered, and where focus indicators appear in space. These decisions are related but distinct, and collapsing them into a single undifferentiated interface, as code-only workflows effectively do, makes each one harder to reason about. The stage architecture surfaces them as separate concerns. It also preserves a direct correspondence between what practitioners see in the interface and the structure of the Data Navigator API [36]: each stage maps onto a distinct layer of the API, lowering the barrier to moving from visual authoring into code when production deployment requires it.

Upload. The upload phase is deliberately permissive. Practitioners can bring a dataset, a chart image, both, or neither. When a dataset is present, *Skeleton* parses its fields and infers dimension types automatically, producing a default, starting configuration for the *Prepare* step. When only an image is present, practitioners proceed directly to editing and construct nodes and edges manually over the image. This was motivated by our geologic map co-design work: many bespoke visualizations do not have a single underlying dataset, and any tool that requires structured data as a precondition for excludes the cases that need it most. *Skeleton* can be applied to any 2D image surface, not only to visualizations in the conventional sense.

Prepare. The prepare stage addresses a hard authoring bottleneck: not placing nodes, but deciding what structure to build at all. A practitioner who has never designed a navigation structure faces an open configuration space with no obvious entry point. The prep stage presents a four-chapter Q&A wizard that moves through authoring decisions sequentially: (1) whether the chart should have a root node and what it should announce, (2) which data fields should be navigable dimensions and how those dimensions should behave at their boundaries, (3) which keyboard interactions each dimension should be assigned to, and (4) what text labels each level of the hierarchy should produce. Each chapter is accompanied by an illustrative schematic diagram of which part of the heirarchy is being edited as well as a diagram showing examples of what these decisions look like when showing on a chart. The wizard's output populates a configuration in the editor that practitioners can then inspect, refine, and revise.

6.5.2 Edit: Interacting with Topology, Layout, and Semantics

Seeing the system, seeing the experience. The editor is *Skeleton*'s primary authoring environment (Figure 7.1) and presents two interlinked representations of the same navigation structure. A schema panel shows the structure as an abstract hierarchical tree layout that makes levels and parent-child relationships immediately readable. A graph canvas shows the same structure rendered as geometric elements positioned over the uploaded chart image, representing what an end user would encounter spatially. These two representations are bidirectionally linked: selecting a node in either view propagates the selection to the other, so practitioners can simultaneously hold in mind both the abstract topology of what is navigable and the spatial instantiation of where that navigation will live. The dual-view design makes visible a real conceptual divide between the system's model of navigation and a user's experience of it, one that practitioners recognize once they can see it, even without prior vocabulary for it.

Leveraging visualization as a scaffolding engine. Manually positioning leaf nodes over each data mark is the most mechanical step in the authoring workflow, and it scales poorly with dataset size. In our early pilot sessions, actually placing nodes in the canvas space was the slowest and most tedious part of the process. To address this, *Skeleton* includes a scaffold tool (Figure 6.3) that automates spatial placement by repurposing Vega [133] as a layout engine.

The scaffold renders a Vega chart specification to a hidden, off-screen container and extracts

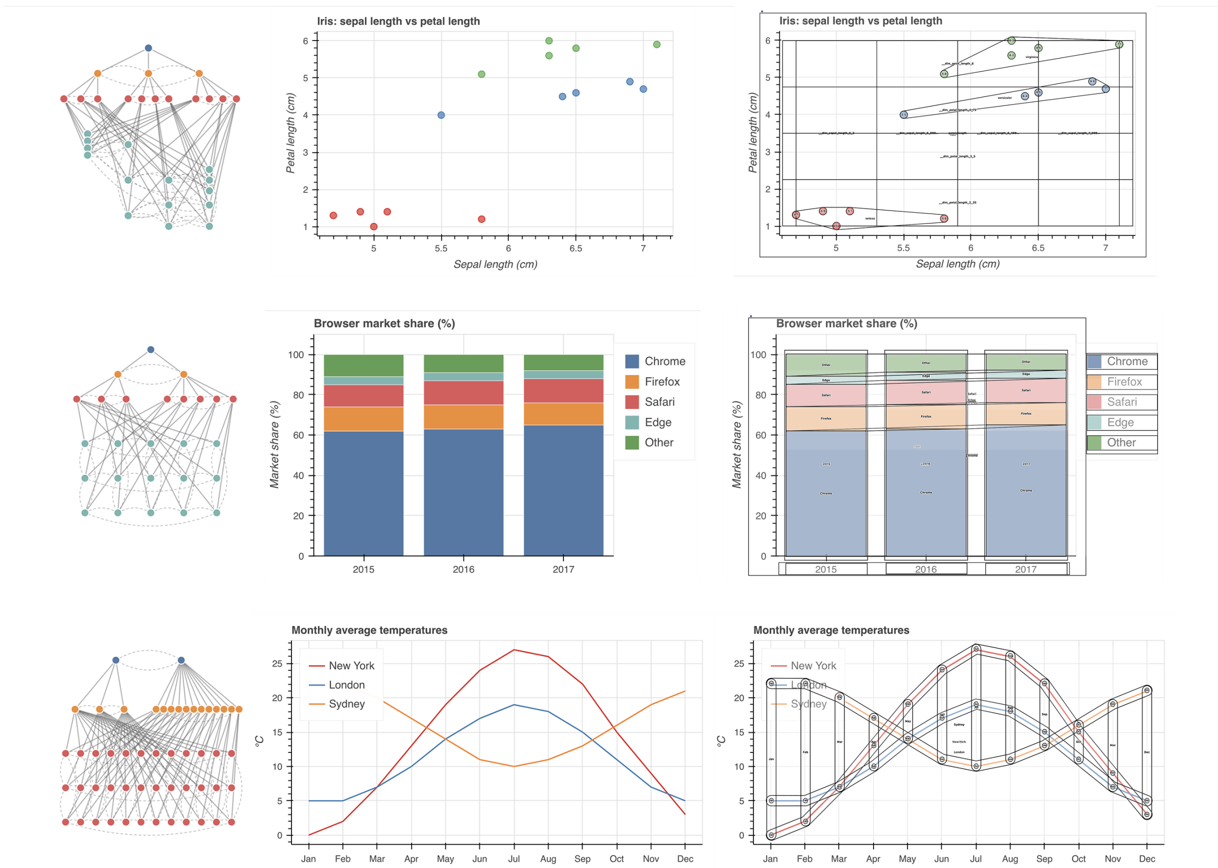

Figure 6.3: Input data transformed into a navigable structure using the *Dimensions API* and visualized with our *Inspector* gadget (left). The input chart (middle). The navigable structure is transformed and drawn over the chart using the *Scaffolding Engine* (right).

node positions via the library's internal scale and view APIs, using the rendering engine purely as a coordinate computation step: no actual chart is ever shown to the practitioner. Coincidentally, none of our co-designers were crafting visualizations using Vega or Vega-Lite (or derivatives), yet the Vega rendering engine could faithfully reconstruct every necessary mark position over the underlying, inaccessible data visualization provided to *Skeleton*.

Additionally, positions and outline strategies for category-level group nodes are computed from the leaf positions using geometric algorithms: a union of child node paths, a convex hull, a grid over numerical bounds, or a bounding rectangle. These designs were consistently produced as ideal treatments during co-design, so we chose them for our starting set of outline strategies.

The scaffold is optional and works by generating synthetic placeholder positions that practitioners can adjust manually. It dramatically improved authoring speed: in light pilot tests, a research team member completed a three-dimension structure without scaffolding in 8 minutes 22 seconds and with scaffolding in 56 seconds. A co-designer completed the task incorrectly (failing to account for one dimension's division nodes) in 13 minutes 7 seconds without scaffolding, and correctly in 2 minutes 44 seconds with it.

Specifying token patterns, editing instances. Selecting any node populates a properties panel with spatial properties (position, size, shape) and semantic properties (ARIA role, description, and a label template editor; Figure 6.4). The label template allows practitioners to assemble the text a screen reader will announce at that node from tokens drawn from data fields, including aggregate statistics at group-level nodes and precise data values at leaf nodes [86]. Editing labels can apply to all nodes of the same type or to a single instance.

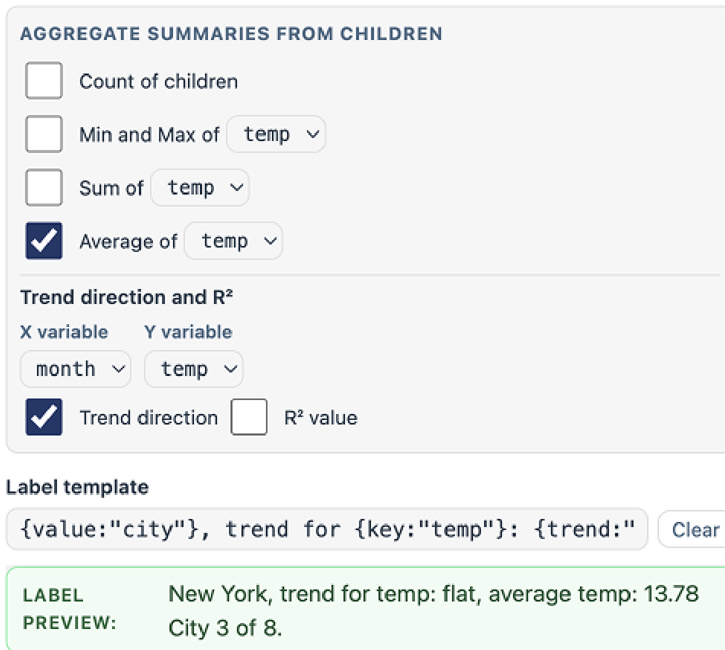

Figure 6.4: Group label pattern builder, including an array of aggregate summary options, template formatter field, and preview.

At the bottom of the semantic section, a live preview displays the full assembled announcement string in the exact form a screen reader would produce: role, semantics, group membership, and label combined into a single rendered output that updates in real time. Prior to this preview, understanding what would be announced at a given node required running a screen reader and navigating to it sequentially. In deeply nested structures, this meant spending several seconds listening to labels and drilling in. This was consistently the main point where bugs were produced and missed during our co-design work.

The preview makes text announcements inspectable as visible, editable objects. This surfaces a class of highly specific, low-level problems that code-only workflows leave invisible: redundancies in announced text, missing contextual framing, and label ordering and punctuation that affects comprehension and reading speed. Of all the authoring decisions *Skeleton* exposes, label templates involve the most degrees of freedom and have the most direct bearing on the quality of the non-visual experience. Getting the structure right ensures navigability; getting the labels right determines whether navigation communicates anything meaningful.

6.5.3 Test: Debugging Interaction Interactively

The testing stage allows practitioners to navigate the structure they have built using the same keyboard input and navigation rules that assistive technologies would use, without leaving the tool. When a practitioner enters testing, the three Data Navigator modules are instantiated in sequence: the structure module rebuilds the navigation graph from the current configuration, the rendering module creates an HTML layer positioned over the chart image at each node's spatial coordinates, and the input module registers keyboard listeners for all navigation rules. The result is a live, keyboard-navigable structure. An event log records navigation events in order, letting practitioners verify that all nodes are reachable and that label sequences make sense when encountered serially rather than read simultaneously.

The abstract graph continues to display during testing. As the practitioner navigates, the focused node is highlighted simultaneously in the canvas (showing its spatial position over the chart image) and in the abstract graph (showing its structural position in the hierarchy). This parallel tracking makes the temporal traversal sequence visible, showing the order and path through which a user encounters nodes, so practitioners can verify at a glance both where focus is and what role it occupies. A text-chat mode is also available, in which practitioners navigate by typing natural language commands, motivated as a design by the Adobe collaboration (Section 6.4.2), to explore interaction alternatives for mobile screen reader users.

The testing stage also served, during development, as the primary debugging interface for *Skeleton*'s own data pipeline. Errors in position computation, label resolution, or structure generation that would propagate silently through code became visible the moment a node highlight appeared in the wrong location or a label read as undefined. A tool for making non-visual structure visible turned out to benefit from the same property during its own construction.

Practice-based validation. CD2, a subject matter expert who professionally evaluates interfaces for screen reader access and is familiar with other visualization navigation systems, used *Skeleton*'s testing stage to evaluate navigation output across several chart types and dimension configurations: line charts (3 configurations), bar charts (2), stacked bar charts (3), and scatterplots (4). This evaluation followed a manual, systematic approach combining standards-based criteria with expert screen reader testing. Scatterplots required the most iteration, surfacing bugs in Data Navigator's core library that were then fixed. CD2 also recommended that we rely on list-based navigation while in the editor (before the testing stage) in case users build themselves into a keyboard trap.

6.6 User Study

Our co-design work followed an action research orientation: the research team was embedded in practitioner communities, and our system work was motivated by the problems those communities faced. This process generated the techniques and infrastructure that comprise *Skeleton*, but we still needed to understand the impact our system had on visualization practitioners more broadly. Our co-designers were deeply familiar with the problem space, having spent months or years working on accessible navigation. We needed to understand what happens when practitioners who are *not* embedded in this process design, author, and debug navigation structure visually: whether the representations we built are legible to them, whether the techniques change how they reason about accessible design, and what new questions or problems emerge when navigation structure becomes visible. These are empirical questions that required a study.

To evaluate how *Skeleton* influences the way practitioners engage with accessible design, we conducted an in-situ interview study with 8 participants across visualization design, engineering, and research. The study was conducted remotely over video call, took approximately 45–60 minutes per session, and was approved by our IRB. Participants provided verbal consent at the start of each session. Video and audio were not recorded, however some participants consented to share the data/image they brought to the study as well as screenshots of their workflow; data collection was note-based throughout.

6.6.1 Participants

Participants were recruited through snowball sampling within the visualization and accessibility community, and through referrals from co-designers involved in our earlier collaborations. We asked each participant to self-report their primary work role (engineering, research, design, or student) and their existing level of accessibility expertise on a 1–5 Likert scale. We also asked whether the visualizations they build are ever bespoke, that is, custom rather than instances of a recognizable, standard chart type. This distinction mattered because bespoke visualizations represent an especially underserved case in accessible design tooling: no library pattern applies, and every navigation structure must be designed from scratch. Participants were not compensated.

6.6.2 Procedure

Each 45-minute session proceeded in four phases. Before the session, all participants were asked to prepare a chart image they were currently working on or had recently built, for use in the third phase.

Phase 1: Introduction and demographics (5 minutes). After obtaining verbal consent and recording a pseudonym, we collected self-reported role, accessibility expertise level, and whether the participant regularly builds bespoke visualizations.

Phase 2: Generic chart think-aloud [2] (10 minutes). Participants used *Skeleton* on a provided bar chart and dataset of fruit counts (Apples, Pears, Nectarines, Plums, Grapes), asked to design an accessible navigation experience for a screen reader user with no instructions on how the tool worked. The tool loaded with a default structure having both a categorical dimension (`fruit`) and a numerical dimension (`count`) active. This default was intentionally problematic for two reasons: numerical navigation sorts by count value and groups data into subdivided ranges, producing a traversal order different than the visual layout an additional, largely unhelpful level in the hierarchy for such a simple chart. We observed how participants reasoned about what they saw and whether and how they noticed this extra dimension.

Phase 3: Own-chart think-aloud (15 minutes). Participants loaded their own chart image and attempted the same task, except they were also asked to explain their graphic to the research team (purpose, role, data, and domain). This phase was open-ended: charts ranged from standard types to bespoke visualizations, and the goal was to observe how participants reasoned about navigation structure when the context was their own work.

Phase 4: Reflective interview (15 minutes). We conducted a semi-structured interview in which participants reflected on their decisions in Phase 2 versus Phase 3, their experience with the generic versus their own chart, and their assessment of the tool's capabilities and limitations. We asked what felt possible or impossible, what they wanted to do that they could not, and what they found themselves thinking about that they had not considered in Phase 2.

6.6.3 Analysis

Notes from each session were compiled into a shared document. Participant quotes reported in the results are reconstructed from these researcher notes, not verbatim transcripts. We analyzed the data using a combination of thematic analysis [12] and affinity diagramming [63], iterating across both methods to surface recurring patterns while preserving the specificity of individual participant experiences. Analysis attended particularly to differences in how participants engaged with accessible design before and after using the tool, the range of input modalities and user scenarios they considered, and moments when participants reconsidered or wished to redesign their own visualizations.

6.7 Results

We organize our findings into five themes that emerged from thematic analysis and affinity diagramming across all eight sessions. Each theme captures a qualitative pattern in how practitioners engaged with accessible navigation design when its structure was made visible and manipulable. We report these findings descriptively and ground them in specific participant moments; interpretation follows in the Discussion.

6.7.1 Seeing Navigation Made Structural Problems Legible as Design Problems

The generic bar chart in Phase 2 loaded with an intentionally problematic default: both a categorical dimension (`fruit`) and a numerical dimension (`count`) were active, producing overlapping navigation structures with different traversal orders over the same data. This configuration is a poor design choice, arguably a design failure, but one that would be difficult to detect in code alone (**R1**, **R4**).

Participants varied widely in how quickly they recognized the problem. P1 turned off the numerical dimension within seconds of seeing the editor, without commenting on it. Most participants, however, initially struggled to understand what they were seeing, remarking on the unfamiliar structure: "what is this? what are these?" when encountering the numerical divisions for the first time. P8 spent time trying to guess what the extra divisions represented but did not remove them during Phase 2, only realizing during Phase 3 that the additional dimension was "probably bad." P2 and P5 expressed suspicion early: P5 asked, "Is this too much data? This seems like way too much to just navigate through," and P2 noted, "I feel like a lot of data points would be bad, yeah? Like, too many at once is bad?" Both of these remarks were prompted by the visual density of the structure, not by navigating it.

The testing stage (Section 6.5.3) proved critical for resolution. P4, P5, and P7 each removed the extra dimension only after navigating the structure with keyboard input in testing mode, where the traversal sequence made the redundancy experientially apparent. In total, five of eight participants resolved the problem during the session: P1 and P2 during editing, and P4, P5, and P7 after testing.

Beyond the intentional default, participants identified other problems through visual inspection. P8 reacted to a generated node name: "Okay dim_fruit node. . . that is horrible, what is that?" During Phase 3, P7 looked at the edges of their multi-line chart and asked, "are all these bad? Is it bad that I don't even really know what the takeaway of this [structure] is?" P3, seeing a full hierarchy for their own simple six-item bar chart (during Phase 3), concluded "I should just skip the root and grouping and go straight to the data. This seems like too many steps." In each case, the visual representation of their navigation structure motivated judgment about potential negative design qualities.

6.7.2 Practitioners Developed a Designerly Interest in What Constitutes Good Navigation

The most pervasive pattern across sessions was that participants began asking design questions about navigation quality, unprompted by any instruction or guidance from the research team (**R2**). These questions went beyond identifying errors: participants wanted to know what *good* navigation would be for their charts.

P2, working through the bar chart in Phase 2, deliberated over boundary behavior: "Loop back or stop? I don't think there is a right way. I will just pick *fruit* for now and *loop* and see what this does." P6, who brought a bespoke flower visualization, wondered how to translate the affective quality of their chart: "I think my visualization should be more about the vibes, but I don't know how to make the alt text have good vibes. What is *fun*?" P4 asked fundamental questions about the interaction model itself: "Why do screen readers and keyboards have to work this way? Do people like that?. . . why do we navigate?" And later after testing, P4 concluded, "I bet we should make this *faster*" before cutting out the additional numerical dimension in Phase 2.

Several participants engaged with the concept of narrative and flow. P8 articulated this as a question about the goal of the visualization: "sometimes I want a big picture, not precision. I may want to drill down a little. . . " and observed that "we think too much in terms of components. . . sometimes accuracy isn't the actual goal, it's getting a general sense of something." P4, upon discovering that *Skeleton* supports text-based input, asked, "How do you make that good, though? Like chatGPT, or do people want to, like, interact with the chart [elements]?'

During the interview, several participants explicitly requested guidance. P1, P2, P3, and P6 wanted to see examples of well-designed navigation experiences. P1, P3, P4, P5, P6, and P7 wanted embedded guidelines within the tool. P2 and P4 actually used web search to look for "chart navigation for accessibility guidelines" (P2) and "accessible viz screen reader design" (P4). P3 was interested in automation and heuristics that could suggest reasonable defaults. These requests are consistent with the pattern (**R2**): practitioners who could see the design space wanted orientation within it.

6.7.3 Iteration Was Substantive, Self-directed, and Concentrated on Semantics

Every participant iterated on their navigation designs, and this iteration was neither perfunctory nor prompted by the research team (**R4**). Participants revisited decisions, revised configurations, and in some cases restructured their entire approach after encountering their design in a new stage of the tool.

The most sustained iteration concerned labels and text announcements. Every participant spent the largest share of their authoring time in the label template editor (Figure 6.4), editing the text that a screen reader would announce at each node. This editing was granular: participants wrote full sentences, rearranged the order of data tokens, debated whether to include field keys alongside values or values alone, experimented with how to name groups and individual elements, and considered the length and density of the resulting announcements. At the division and dimension levels, some participants added multiple aggregate statistics (count, sum, average, range, trend) and then returned to trim them. P2, for instance, edited data point labels, left them, and returned to revise them two additional times: "This label is way too complicated, I think." These label iterations were small, fast, and frequent, and they reflected an intuitive grasp of the importance of the textual tokens that constitute a screen reader user's primary interface to data.

A second, distinct pattern of iteration emerged around testing. The editor displays all navigation nodes simultaneously, showing the full structure at a glance. The testing stage (Section 6.5.3), by contrast, shows only the currently focused node, highlighting it in the canvas one at a time as the practitioner navigates. This difference consistently prompted participants to revise. Several returned to the editor after testing to adjust group node outlines, because outline strategies that looked distinct when displayed simultaneously (such as bounding rectangles vs. convex hulls) became harder to differentiate when encountered one at a time. Others adjusted their dimension configurations: P3 and P5 restructured their dimensions after testing, and P4, P6, and P8 experimented with different key bindings and navigation rules. P2 used the testing stage specifically to identify labels that needed revision at the dimension and division levels. In these cases, testing was not only treated as a bug-finding activity but also as a way to encounter, reason about, and then improve a design that had looked adequate in the editor but felt inadequate in sequential traversal.

The most extensive iteration came from P1, who returned to the preparation wizard after reaching the testing stage and re-took the entire wizard from scratch. Seeing the full structure in testing motivated them to re-evaluate their earliest decisions. They reduced their navigation from three dimensions to one, producing a navigation experience they described as deliberately minimal. In the reflective interview, P1 explained that they had been thinking about trimming excess, joking that they were "working on a data-to-word ratio."

The scaffolding tool shaped the pace and character of these iterations. Participants who used the scaffold (Figure 6.3) generally required only minor manual adjustment of node positions, spending one to two minutes refining placements in-aggregate before moving on. Iteration on spatial layout was primarily about the appearance of group-level outlines rather than individual node positions: because the scaffold computed leaf positions from the chart's encoding, the main remaining spatial decision was how division and dimension nodes should visually indicate their grouping. When participants returned to scaffolding, it was most often because they wanted to change the chart type used for coordinate computation or to try a different group outline strategy.

6.7.4 Seeing Navigation Prompted Practitioners to Reconsider the Architecture of Their Own Charts

An unexpected finding was that working with *Skeleton* prompted several participants to question the design of the visualization they had brought, not just the navigation structure they were building over it (**R4**). Five participants (P4, P5, P6, P7, P8) expressed a desire to simplify their own charts after seeing what the corresponding navigation structure required. Three (P1, P5, P6) considered whether a different chart type might serve the same communicative goal with a simpler navigational architecture. And in 2 cases (P1, P6), participants questioned whether a chart was the right medium at all.

P5, who brought a scatter plot with over two thousand data points, initially tried to build a navigation structure over the full dataset, recognized that the result was unmanageable, and then asked: “do I even need visualization?” They went on to wonder whether the information could be communicated as “a few sentences or like, some data [users] can prompt” (referring to using a large-language model). P6, who brought a bespoke flower visualization in which each petal encoded a different variable, initially tried to express the chart’s full complexity in navigation (grouping by flower and then navigating by petal) but then reversed course: “okay, what if we actually just treat this like a bar chart?” Using the scaffold tool, they produced a simple list-style navigation that worked well for their data, and reflected: “my visualization has a lot, but actually, this could be pretty easy to navigate, I think.” P6 also wondered whether there was a non-chart way to “tell their story,” and in further discussion described something closer to a scrollytelling article or guided walkthrough than a single interactive chart.

P8’s case was the most striking. They brought a voronoi pie chart (Figure 6.5) used to communicate to students that 26 assignments make up 45% of their grade, the largest slice of the pie. Working through *Skeleton*, P8 first considered making all 26 voronoi cells navigable, then considered making only the three main slices of the pie navigable, and then questioned whether navigation was needed at all. The “point of this,” as they described it, was to communicate a single ratio; students did not need to traverse individual cells to understand it. P8 concluded that well-written alternative text was sufficient for the screen reader experience and chose not to build a navigation structure. They also reflected on the design of the visualization itself, concluding that the voronoi treatment served a visual purpose (it is visually striking) that was separable from the informational purpose (communicating a grade breakdown). They kept the visual design and simplified the non-visual experience accordingly.

6.7.5 Experiencing Keyboard Navigation Surfaced a Broader Range of Users and Input Technologies

The testing stage (Section 6.5.3) provided what was, for most participants, their first experience of keyboard navigation through a data visualization. Only three participants (P1, P2, P8) reported prior experience using a screen reader, and only two (P1, P8) had previously navigated a visualization using keyboard input alone. Yet during the study, every participant navigated using

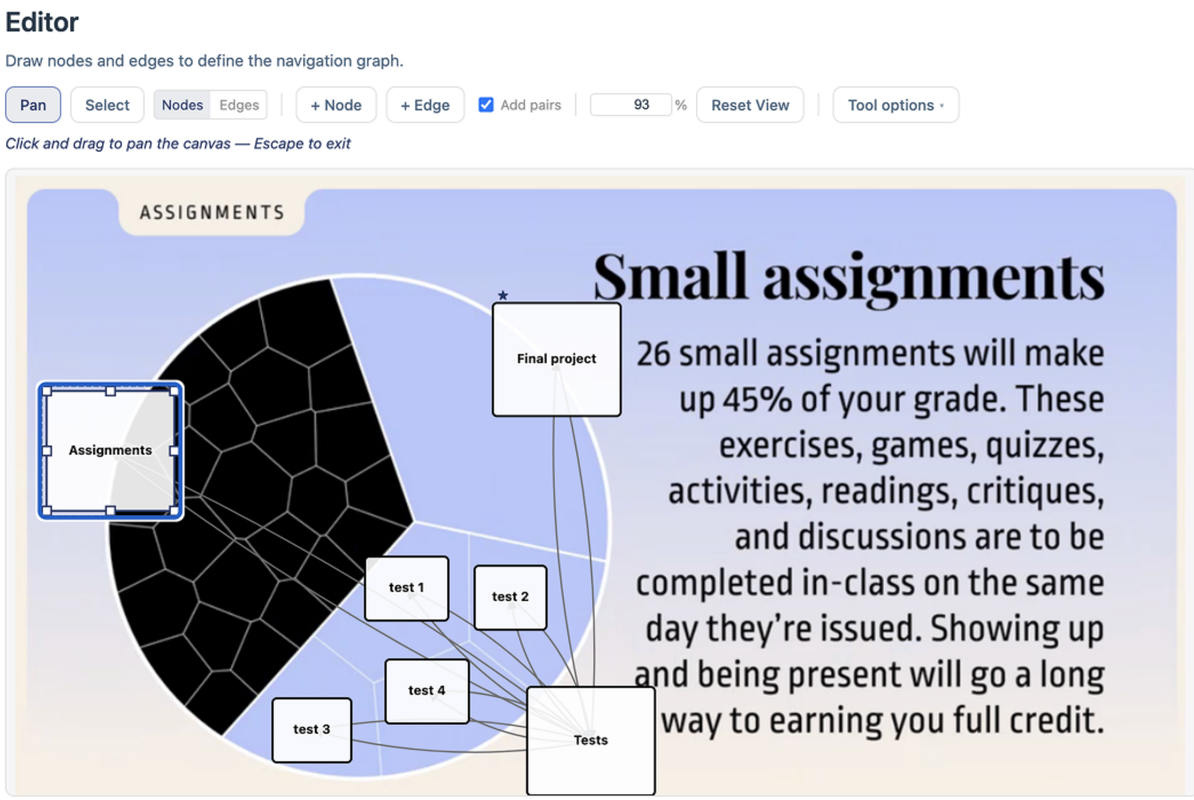

Figure 6.5: Re-creation of P8's moment of realization, placing nodes manually: not every element in their voronoi pie chart *should* be navigable.

a keyboard.

Several participants responded to this experience with genuine engagement. P4 reacted with enthusiasm: "Woah, this is incredible. Wait, what other visualizations can do this?" and later, "Maybe you could make a visualization game, where you can move around the data?" P7 said, "I love this. I want to make charts with this." P5, who had not previously encountered keyboard-navigable charts, said, "I didn't know you could do this." These reactions were not about the tool; they were about the interaction modality itself, about the experience of traversing data spatially using directional input.

The experience also expanded participants' sense of who these structures serve (**R2**, **R4**). P8 adjusted a node's spatial dimensions and said, "Let's make the hitbox as big as possible here. . . for my neighbor with Parkinson's to click these," treating the navigation structure as relevant to motor accessibility, not only screen reader access. P3, observing the visual focus indicators that appeared during scaffold-based placement, remarked that "this is for more people than [someone] blind," recognizing that sighted keyboard users and users with partial vision also rely on visible focus states. During the interview, P4 asked sustained questions about what kinds of technologies different people with disabilities use, and P4 and P7 both asked why focus indication was designed to be visible rather than invisible.

P4's curiosity extended to alternative input modalities. Upon learning that *Skeleton* supports text-based navigation and (due to leveraging Data Navigator) also supports a wide array of other input modalities, they asked, "Can you talk at it, too? Is that what some people do?" and followed up with, "How cool is that? How do you make that good, though?" P8 raised a concern about discoverability in the current interaction model: "I've never used j or w to drill out, always shift arrow or option arrow..." and worried that a user might not know how to exit a nested level. These observations reflect an broad mental model of the user, from an abstract "screen reader user" to a person with multiple capabilities, preferences, and interaction patterns (**R2**).

6.8 Discussion

6.8.1 Visibility as a Precondition for Iteration

The central finding of this work is not that *Skeleton* taught sighted practitioners how to design accessible navigation, but that it gave them something to react to. When navigation structure was invisible, our co-designers would only seek to verify whether navigation followed our design documentation. Once visible, questions shifted to whether it was good, why, how it could be improved, and what else it could do. Visibility encouraged iteration, and iteration is enabled continuous design improvements.

This mechanism is consistent with Schön's account of reflective practice [137]: designers iterate by externalizing a representation, perceiving its properties, and responding to what they see. The representation talks back, and the designer adjusts. But this loop requires a representation. In our co-design work, we assumed that the gap was hand-off between design and development, and the slow process of manual verification. Instead, the gap is that development was not participating in, and encouraging, further design reflection. In a developer's code-only workflow, navigation structure has no externalization that supports this kind of perceptual engagement. A practitioner can read the code that specifies a navigation graph, but they cannot see the graph, cannot perceive its topology at a glance, cannot notice that a label is redundant or that a hierarchy is too deep by looking at it. The reflective loop is disrupted and occluded at the first step.

Skeleton encourages this loop by rendering navigation structure in a form that supports the same kind of perceptual judgment that visualization practitioners already apply to every other aspect of their work. The results show what happened when this loop was available: every participant iterated, substantively and self-directedly (Section 6.7.3). They revised labels repeatedly, restructured dimensions after testing, adjusted group outlines when sequential traversal revealed problems that simultaneous display had hidden. P1 restarted the entire preparation wizard after seeing their structure in testing mode. These are not the behaviors of practitioners following a specification; they are the behaviors of practitioners negotiating with a design material.

The co-design work reported in Section 6.4 provides complementary evidence: in each collaboration, sighted practitioners naturally reached for visual representations when reasoning about navigation, through node-link diagrams in Figma, schema sketches in Miro, and annotated wireframes on paper. They were already thinking visually about non-visual structure; the development tooling simply had not caught up. This has a practical implication for the broader field: if sighted authors depend on visibility, then any authoring workflow that keeps navigation structure invisible limits design iteration. Making structure visible does not guarantee good design, but it is a precondition for the kind of sustained, judgment-driven refinement that good design requires.

6.8.2 From Compliance to Design

A consistent pattern across the accessibility literature is that practitioners frame accessibility as a compliance problem: a set of requirements to satisfy, a checklist to complete, a legal or institutional obligation to meet [35, 87, 143]. The framing matters because compliance and design orient practitioners toward fundamentally different activities. Compliance asks: "does this pass?" Design asks: "is this good?" Our results suggest that *Skeleton* produced a partial shift from the first orientation to the second. Participants' initial instincts clustered around compliance-oriented responses: provide alternative text, follow guidelines, ask an expert. But alongside that desire for guidance, they began doing something compliance framing does not typically produce: they encountered complexity and then *iterated*. They revised labels repeatedly, restructured dimensions after testing, debated boundary behavior and hierarchical depth. This sustained, self-directed refinement is the behavioral signature of design, not compliance. As P4 put it: "I'd love to have someone blind actually just with me while I make this, but I also understand that I should learn what makes a good experience too."

The shift extended beyond the non-visual experience itself. As reported in Section 6.7.4, five participants reconsidered the design of the visualization they had brought, not just the navigation structure overlaid on it. Making the accessibility consequences of visual design choices visible prompted practitioners to question whether a different chart type, a simpler encoding, or a non-chart medium might better serve their communicative goals. This interrelation between non-visual and visual design suggests that the widespread treatment of accessibility as a compliance activity may be partly a consequence of tooling that offers no legible, manipulable design surface. Auditing frameworks are valuable, but they are *evaluative* tools, not *authoring* tools. The field needs both.

6.8.3 Bespoke Visualizations as an Unaddressed Accessibility Research Problem

Diagrams, infographics, and data-driven illustrations are often one-off, custom designs with bespoke symbols, layouts, and visual languages. These representations are increasingly common in journalism, scientific communication, personal projects, art, and public-facing data work, and they represent the cases where accessibility-focused tools are needed most and available least. *Skeleton*'s image-based workflow (upload any 2D image, place nodes manually) provides a starting point, but the study made clear that bespoke visualizations need more than node placement. They need support for reasoning about what navigational structure (if any) is appropriate when no template applies, a research problem that remains largely unexplored.

6.8.4 What Visualization Owes Accessibility

Our approach, to make visual non-visual experiences, should not be limited to data visualization's own accessibility challenges. Navigation structure is a foundational component of accessible experience across domains: PDF and document reading order, web page structures, and software application layouts. In each of these areas, sighted practitioners author non-visual experiences without visual feedback, and in each, the same gap between design intent and verifiable outcome constrains quality. Testing is slow, error prone, and requires expertise in assistive technology use. Visual tooling for authoring, inspecting, and debugging non-visual structure (**R3**) is a tractable and high-value problem across application domains.

There is also a deeper question about what visibility can accomplish in principle. Work on multi-modal authoring environments has argued for de-centering visual representation and treating modalities as equal partners in the design process [190], an important ethical commitment. *Skeleton* does not do this: it re-centers visual representation as the medium through which sighted practitioners engage with non-visual structure. We believe this is justified pragmatically, because sighted authors need to articulate navigation design in their own perceptual language before they can reason about it at all, and this paper provides evidence that they do. But articulating a design in one's own language is not the same as understanding how it will be experienced in someone else's. The structures participants built during our study were never evaluated by blind users, and visibility alone cannot substitute for that evaluation. The risk we want to name is that making non-visual structure visible to sighted practitioners could be mistaken for making it *understood*, when in fact it makes it *designable*, a real but bounded gain. The fuller design process requires collaboration with disabled users, not as an occasional supplement but as a regular practice. *Skeleton* can make that collaboration more productive by giving both parties a shared representation or space of translation between representations, but it cannot replace it.

What visualization owes accessibility, then, is not simply better output but authoring tools that better stimulate reasoning, both individually and collaborative, about design.

6.9 Limitations and Future Work

Skeleton makes navigation structure visible to sighted authors, but it cannot reassure those authors whether the structure they have built is good for the people who will use it. The tool surfaces design questions; it does not answer them. Several participants asked what constitutes good navigation, and *Skeleton* had nothing authoritative to offer. Our study with sighted practitioners evaluated whether making navigation structure visible stimulated design consideration, not whether the designs sighted practitioners produced were actually good. These are related but distinct questions, and the second remains open. CD2's expert screen reader evaluation of *Skeleton*'s navigation output (Section 6.5.3) provided practice-based validation for several common chart types and surfaced concrete bugs, but this evaluation was neither comprehensive nor controlled: many configurations remain untested, and expert review is not a substitute for evaluation with a broader population of end users.

Additionally, we treated *Skeleton* as a design probe [61, 81] rather than comparing it to a controlled condition: the goal was to elicit qualitative insight about how the tool elicits engagement, not to measure performance differences.

Our approach using *Skeleton* as a design probe only with sighted participants is both a limitation and, we believe, the right sequencing: *Skeleton* improves the intentionality and iterability of what sighted practitioners produce, which is a necessary precondition for a subsequent evaluation or future collaborative design work between sighted and blind authors. Further research and evaluation should close this loop, ideally to engage how mixed-ability teams co-design multi-modal data experiences.

Skeleton is also a prototype with substantial work remaining. Not within the scope of the paper, but our participants provided ample feedback on the functionality of the prototype itself. The most urgent gap is export functionality: practitioners can design and inspect navigation structures in the tool but cannot yet produce deployable output.

6.10 Conclusion

Accessible navigation structure has long occupied an awkward position in visualization practice: known to matter, difficult to design, and invisible to the people responsible for building it. The invisibility was not incidental. Without a way to see what they were making, sighted practitioners could not catch errors, could not iterate, and could not develop the kind of considered judgment that good design requires. Accessibility remained downstream of every other decision not because of any single failure, but because the authoring conditions did not support anything else.

Skeleton demonstrates that those conditions can be changed. Making navigation structure visible and manipulable, as an interactive graph rendered over the spatial layout of a real visualization with live label previews and testable traversal, shifted how practitioners engaged with and reasoned about accessible design. They began asking qualitatively different questions than someone seeking compliance: whether the features and design of their structure was good, despite not having readily available answers.

If any conclusive take-aways can be gleaned from this project: for researchers interested in engaging accessibility, this would mean future projects might explore translational spaces between visuals and non-visuals that help sighted partners engage blind designers. For practitioners who build, design, or audit this would mean that more visualization is needed in current tooling, to enhance and make multi-modal existing non-visual methods of authoring and evaluation.

What the paper leaves open is more than what it closes. We used *Skeleton* as a design probe with sighted practitioners; we did not evaluate the navigation structures they produced with the end users those structures are meant to serve. And the broader disciplinary conversation, about what visualization research owes accessibility and what methods might transfer between them, has more questions than answers. We offer *Skeleton* not as a solution to these problems but as evidence that engaging them directly, with the full weight of visualization's methodological tradition, is both possible and fruitful.

Part IV

Interaction: Exploring New Possibilities for Blind Data Science

Chapter 7
Cross-perception: Rethinking Input Design Towards Blind Analytical Interaction

This chapter was adapted from my paper, currently under review with IEEE VIS:

F. Elavsky, Y. Li, J. Jang, L. Nadolskis, P. Carrington, and D. Moritz, '*Cross-Perception*: Rethinking Input Design Towards Blind Analytical Interaction', *IEEE Transactions on Visualization and Computer Graphics*, 2026.

7.1 Abstract

Interactive visualization research has produced a rich repertoire of interaction techniques (brushing, linking, and cross-filtering chief among them) that accelerate exploratory data analysis for sighted users. And accessible visualization research has made substantial progress on making data perceivable and navigable for blind users. Yet what remains largely unaddressed is the *analytical* tier of blind interaction: the capacity to actively filter, conjecture, and test hypotheses through direct data manipulation. In existing systems that accomplish interactive, analytical tasks, users predominantly query text or sequentially navigate across discrete units of information, paradigms which focus the user on one piece of information at a time. These experiences limit a user's ability to rapidly explore and interrogate multiple dimensions within a dataset simultaneously. To target this analytical gap, we introduce *cross-perception*, an interaction technique that adapts cross-filtering for non-visual, tactile interaction. *Cross-perception* enables coordinated tactile brushing over one data space to produce perceptible computational output in a separate, linked space, without requiring serial navigation between them. We instantiate *cross-perception* in the *cross-feelter*, a prototype hardware device with dual motorized faders, and evaluate it against screen reader methods in a within-subjects study with 15 blind participants, 7 of whom have professional data expertise. Cross-feelter substantially improves exploration speed (+90%), query generation (+188% computational, +54% spoken), enjoyment, and reduces anxiety (particularly for participants without prior data expertise). We discuss implications for interaction design in accessible visualization and propose *cross-perception* as a framework for a broader class of non-visual, analytically-oriented data interaction techniques.

7.2 Overview

The visualization community has spent decades formalizing interaction techniques for sighted users: brushing scatterplots [5], linking coordinated views [16], and cross-filtering aggregated data to rapidly surface relationships across dimensions [1, 167]. These techniques are not only conveniences. Empirical work has shown that fast, direct manipulation of data fundamentally

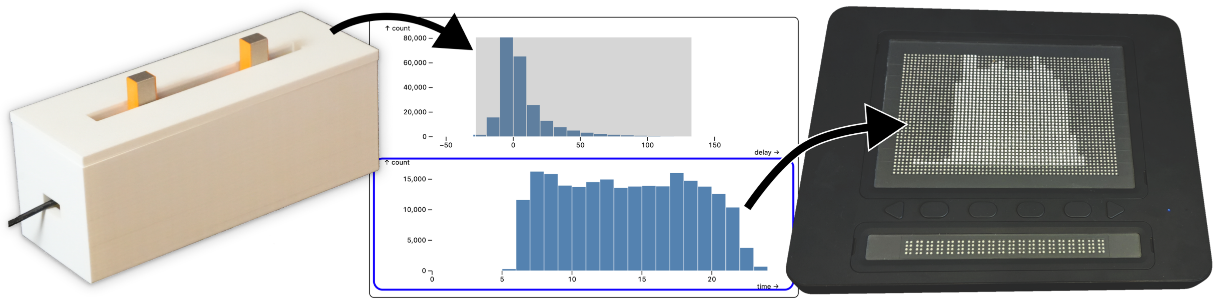

Figure 7.1: Interaction and perception in one space while being able to perceive output in a separate space is the cornerstone of cross-filtering. Our prototype *cross-feelter* (left) can manipulate a visual cross-filter on one visualization (middle) in order to produce output in a separate visualization, as a tactile graphic (right).

changes the speed and character of hypothesis generation, and that even small increases in interaction latency measurably reduce the quality of exploratory analysis [45, 99].

Accessible visualization research has made substantial and genuine progress in recent years. Blind users can access data through rich text descriptions [88, 101], sonification [75, 92, 190], and tactile representations [18, 74, 109, 110, 148, 157]. Increasingly, this work extends beyond rendering to *interaction*: screen readers support structured traversal of chart elements and their relationships at varying levels of granularity [36, 114, 188], and multimodal systems pair conversational agents with tactile displays to support rich data exploration [127].

What existing work addresses is, broadly, the question: *what is here?* A blind user can navigate a data representation to understand its structure, locate specific values, extract summary statistics, ask descriptive questions, and build a mental model of what the data contains. This is *access-oriented* interaction, and current tools support it with increasing sophistication.

What remains largely absent from accessible visualization is a different kind of interaction entirely: the *analysis-oriented* kind, in which a user not only reads a dataset but actively operates on it: filtering subsets, conjecturing relationships, testing hypotheses, and generating new knowledge through iterative manipulation. These are the fundamental activities of interactive data science, and they have a different character from access-oriented interaction. Questions shift from "*what is here?*" to "*what happens to Y when I constrain X?*" According to taxonomies of analytical tasks, existing work has largely not addressed "characterizing distributions" and "correlating" tasks, which are fundamental to hypothesis-construction [3].

We propose *cross-perception*, an interaction technique that adapts cross-filtering for non-visual contexts while preserving the simultaneity that makes it analytically useful. Cross-perception is defined by two properties. First, *coordinated tactile brushing*: the user manipulates a persistent tactile input in one interaction space, selecting a range across a data dimension. Second, *linked perceptual output*: the computational result of that manipulation is immediately perceivable in a separate space, without requiring serial navigation between them. Together, these properties support not only access to data but active analytical engagement with it: the capacity to ask iterative questions through direct manipulation.

To evaluate cross-perception and explore its design space, we built the *cross-feelter*: a prototype device with two motorized linear faders that maintain persistent, absolute tactile position for brush handles, paired with a refreshable braille display for tactile output. We conducted a within-subjects study with 15 blind participants, comparing the cross-feelter against a screen reader baseline on structured and open-ended data exploration tasks.

In our work, we address four research questions:

- **R1 (Exploratory):** How might we formalize a tactile interaction approach that enables coordinated, non-visual interaction and perception across multiple spaces?
- **R2 (Quantitative: Objective):** In what measurable ways can we observe improvements to blind data interaction using our approach, in terms of speed, accuracy, precision, and quantity of data queries?
- **R3 (Quantitative: Subjective):** What are the self-reported effects on stress, cognitive load, anxiety, and enjoyment when using our approach?
- **R4 (Qualitative):** In what ways do blind data practitioners imagine extending our approach and its instantiating hardware to new contexts?

This paper makes three contributions. First, we formalize *cross-perception* as an interaction technique for non-visual *analytical* data interaction; one that equips blind users to actively filter, conjecture, and test hypotheses through direct tactile manipulation, not only to access and navigate data (Section 7.4). Second, we provide an empirical evaluation demonstrating that cross-perception yields substantial improvements in exploration speed, query generation, and user experience relative to screen reader interaction, while maintaining comparable accuracy and precision (Section 7.5–7.7). Third, we argue that accessible visualization has made significant progress at the access-oriented tier of interaction, and that our field's frontier has shifted: the work now should be to give blind users *analytical agency*: the ability to interrogate data, not only to read it (Section 7.8).

7.3 Related Work

This paper sits at the intersection of two bodies of work: research on cross-filtering and analytical interaction in data visualization, and research on making data accessible to blind users. We review each in turn, then characterize the specific gap that cross-perception addresses.

7.3.1 Cross-Filtering and Interactive Visual Analysis

A note on terminology. *Brushing* is the direct manipulation gesture of selecting a contiguous visual range across a data dimension [5]. *Linking* propagates that selection across coordinated views [16]. *Cross-filtering* applies the linked operation as a filter on shared underlying data, constraining what is visible across views rather than merely annotating selections [1, 167].

What makes cross-filtering analytically valuable is not that users can manipulate the interface, but that the gesture and its computational consequence are perceived simultaneously. Even 500ms of additional latency measurably reduces the rate at which users generate observations and formulate hypotheses [99], and providing partial results incrementally reduces but does not eliminate this cost [45]. Temporal coupling between input and output is the mechanism by which direct manipulation supports hypothesis generation, not an implementation detail [66]. Recent work on scalable cross-filtering has continued to optimize this coupling through improved architectures [67], including pre-computed aggregates [117]. This distinction, between analytical interaction that constructs new information through manipulation and navigational interaction that locates information already present, motivates the core design challenge of cross-perception.

Previous work establishes that cross-filtering supports *analytical* interaction, in which the user constructs and tests hypotheses through rapid, iterative, direct manipulation, rather than navigating toward information that is already present. The value of this analytical interaction is that the user, not a system, gains new insights, questions, and understanding [146].

7.3.2 Non-Visual Data Representations

Accessible visualization has produced many systems for making data perceivable to blind users across text, audio, and tactile modalities [38, 54, 88, 92, 101, 124, 157]. We position cross-perception as complementing this work: many rendering and representation problems have made significant advancements in recent years; but analytical interaction problems, comparatively, have not.

Sonification. Sonification, mapping data variables to auditory properties such as pitch or timbre, is a distinct and sophisticated field with its own tooling, theory, and evaluation methodologies [69]. Recent work has formalized sonification as a first-class rendering approach with grammar-level expressiveness, enabling mappings comparable in richness to visual encodings [92]. Some work has explored data-centric approaches that treat sonification and visualization as parallel co-equal representations [190].

A note on auditory perception is warranted: the human auditory system is capable of parsing concurrent streams and segregating them into distinct perceptual objects [13]. However, the relevant constraint for blind and low vision users working with data is more specific: in task-driven contexts, synthesized speech from a screen reader competes with data-representing audio in the same channel, increasing cognitive load of the user and providing complex challenges for authors when designing speech audio and sonification interfaces [50, 92, 161]. Dual-task and multi-task operations become difficult or impossible, which is a design challenge for cross-filtering specifically, where the filter state and its output must be perceived simultaneously.

Tactile and haptic output. Research on tactile data representation spans several decades, from foundational work on tactual perception [136] to recent studies formalizing tactile alternatives to common chart types [38, 124]. More recent work has explored simultaneous tactile-and-audio output [18], slide-tone and tilt-tone haptic encodings for shape characteristics of graphs [41], comparative studies on tactile vs audio output [23], and design guidelines for multimodal touchscreen-based graphics that inform both resolution and tactile landmark placement [54].

7.3.3 Access-Oriented Data Interaction

In parallel with rendering advances, accessible visualization has developed increasingly sophisticated interaction techniques for blind users, such as serial and multi-directional navigation across data structures using a screen reader [36, 188]. However, screen reader traversal is sequential and discrete, which impacts a blind user's ability to extract information from visualizations, impacting accuracy by 61% and costing 210% more time doing so [141]. Empirical work has engaged "questions" that everyday blind folks have of data representations [93], and most recently, conversational agents have been paired with refreshable tactile displays to support question-based exploration combining voice input with tactile output [127].

These systems share a common character: they support *access-oriented interaction*, attempting to bridge the gaps that exist between sighted and blind access to information. However, the activities that remains unsupported requires constructing information through manipulation rather than locating it through navigation; the work of professional data science. In analysis-oriented interaction, the question is not merely lookup or query tasks such as *what is the value at this position* but along the lines of *what is the joint distribution within this subrange*, an answer that does not exist until a filter is applied (and a question that may not even exist unless within a system that supports direct interaction and iteration with the data).

7.3.4 Tactile and Haptic *Input*

Tactile and haptic input for data interaction has been explored in immersive contexts (not related to accessibility or blind interaction). For example, Cordeil and colleagues' *embodied axes* allow manipulation of data axes as physical objects in augmented reality [25], and the *MADE-Axis* extends this to motorized dual-axis faders for AR/VR data environments, demonstrating a cross-filtering use case [147]. Both leverage tactile input and motorized output, though both presuppose sighted users in visual contexts.

Within accessibility-specific work, a touchpad-based brush interface showed 1D filter selection feasibility but provides no queryable positional state when the user's hand is lifted [144]. And a non-motorized slider within an accessible smartphone interface shows the principle of persistent tactile state [185] but cannot be programmatically positioned, which cross-filtering requires for two-way synchronization.

The gap that remains is specific: persistent physical state and simultaneity of input and output perception are both necessary for analytical interaction, and no existing accessible system combines them.

Cross-perception is our attempt to address this gap. Our design is informed by the principles reviewed above: the temporal coupling of cross-filtering, the persistent-state affordances of motorized tactile input, and the insights from prior work on non-visual data interaction regarding the importance of simultaneity, spatial memory, and multi-channel perception. The following section formalizes these into a set of design goals for the interaction technique.

7.4 Formalizing Cross-Perception

Cross-perception is a tactile interaction technique defined by two properties. First, *coordinated tactile brushing*: the user manipulates a persistent tactile input in one interaction space, selecting a range across a data dimension. Second, *linked perceptual output*: the computational result of that manipulation is immediately perceivable in a separate space, without requiring serial navigation between them. These properties directly instantiate the simultaneity requirement established in Section 7.3: brush state and its consequence are perceived together, in the same moment, supporting the rapid inference cycles of analysis-oriented interaction.

In this section, we formalize cross-perception by deriving six design goals from the principles reviewed in the related work. We begin with a design vignette with our co-designer and co-author [REDACTED], that illustrates how the same cognitive mechanisms that underpin our goals appear in naturalistic blind interaction with physical documents.

7.4.1 A Design Vignette: Fuzzy Search and Dual-Task Comparison in Naturalistic Blind Reading

The following vignette, shared with permission by our blind co-designer and co-author [REDACTED], illustrates two tactile interaction strategies that correspond directly to our design requirements formalized below. We present this vignette as both grounded in the cognitive science and visualization literature reviewed in Section 7.3 as well as evidence that the principles we derive from that literature describe real, expert-level behavior in the wild.

[REDACTED], a blind neuroscience engineer, was reading an embossed academic paper and explaining a particular equation to the primary author as the equation recurred and evolved across multiple pages. The embossed format provides high-resolution tactile rendering at up to 100 dots per inch, and [REDACTED] interacted with it using two strategies we have named *fuzzy tactile search* and *dual-task comparison*.

Fuzzy tactile search is a spatial indexing strategy. [REDACTED] picked up the stack of pages and thumbed to an approximate location based on tactile position in the stack, touched near the top of the page, immediately recognized from the local texture pattern that he was one page off, flipped to the next page, and navigated directly to the spatial region where the equation began. He then repeated this process to locate the equation's recurrence later in the paper. The strategy relies on two cognitive functions reviewed in Section 7.3: spatial memory, encoding the approximate location of content within a document's physical structure, [15, 24, 28] and *object-location memory*, the ability to retain and retrieve the position of objects without continuous visual or tactile contact [48, 98, 100]. Both functions require that objects *have* a stable spatial position to encode: they cannot operate on a system where position is reassigned or meaningless.

Dual-task comparison is a simultaneous spatial reasoning strategy. Having located the same equation in two places, [REDACTED] placed one hand on each and skimmed both simultaneously, stopping when his hands detected the point of divergence between them. He explained the evolution of the equation by reference to that divergence. This strategy demonstrates exactly the interaction character that cross-filtering supports for sighted users: two data regions perceived in parallel, with relational structure emerging from simultaneous contact rather than sequential reading. It also demonstrates the additional requirement of *simultaneous multi-location tactile interaction*: a user resting a pinky on a known reference point while exploring relative to it with another finger, a pattern documented in prior empirical work on tactile interactive systems [54].

Together, these strategies illustrate a key observation: skilled blind readers of physical documents are not limited to sequential access. The constraint is architectural: the medium must maintain a persistent, meaningful spatial state so that object-location memory can operate, and it must allow simultaneous multi-point contact so that comparison can occur. Our design goals operationalize these requirements for computational, analytical interaction with tactile interfaces.

7.4.2 Design Goals

We derive six design goals from the principles established in Section 7.3 and illustrated by the vignette above.

G1: Maintain and reflect system state through tactile location. Object-location memory requires objects with stable, queryable positions [48, 100]. Tactile input elements must therefore have absolute, persistent positional state: the knob must remain where the user or system last placed it and be readable by touch without performing an action. This also enables motorized two-way synchronization, where the system sets knob position as output and the user sets it as input.

G2: Allow for multiple simultaneous channels of perception. Synthesized speech competes with data-representing audio under cognitive load [50]. Cross-perception routes data manipulation to a tactile input channel, eliminating this competition. Audio output via ARIA-live announcements remains available for precise positional feedback but is kept serial and non-competing.

G3: Correspond tactile-to-computational functional semantics. Interface elements should have names and behaviors that describe what they do [35, 164]. The visual cross-filtering metaphor involves dragging handles along a spatial dimension; cross-perception preserves this by mapping a physical rail sliding to setting a filter boundary, reducing translation between user action and analytical action.

G4: Provide linked interaction across multiple data dimensions. Cross-filtering requires that a filter on one dimension immediately propagates to constrain coordinated views of others that share the same underlying dataset [1, 16]. The filter state set by tactile input must immediately determine the output displayed in linked views; without this, the technique only accomplishes single-view selection.

G5: Support fast, dynamic, and reversible interaction. The analytical value of cross-filtering depends on rapid hypothesis iteration [45, 99]. Filter adjustments must be continuous and incremental, system response must be low-latency, and any state must be immediately reversible. These motivate motorized faders with 1024 units of precision and a pre-computed data cube backend [67].

G6: Facilitate user-driven exploration. Cross-filtering is a direct manipulation paradigm in which the user, not an algorithm, controls the analytical flow [1, 146]. The user's physical manipulation of the filter handles is the direct and sole determinant of what the tactile output displays, with no intermediary statistical, language model, or non-deterministic interpretation.

7.5 Prototype: The Cross-Feelter

We instantiate cross-perception in a hardware prototype we call the *cross-feelter*: a device with two motorized linear faders that provide persistent, absolute tactile input state for brush handles, paired with a refreshable braille display for coordinated tactile output. All hardware schematics, firmware, and interface code are open-sourced at [REDACTED]; 3D-printable STL files

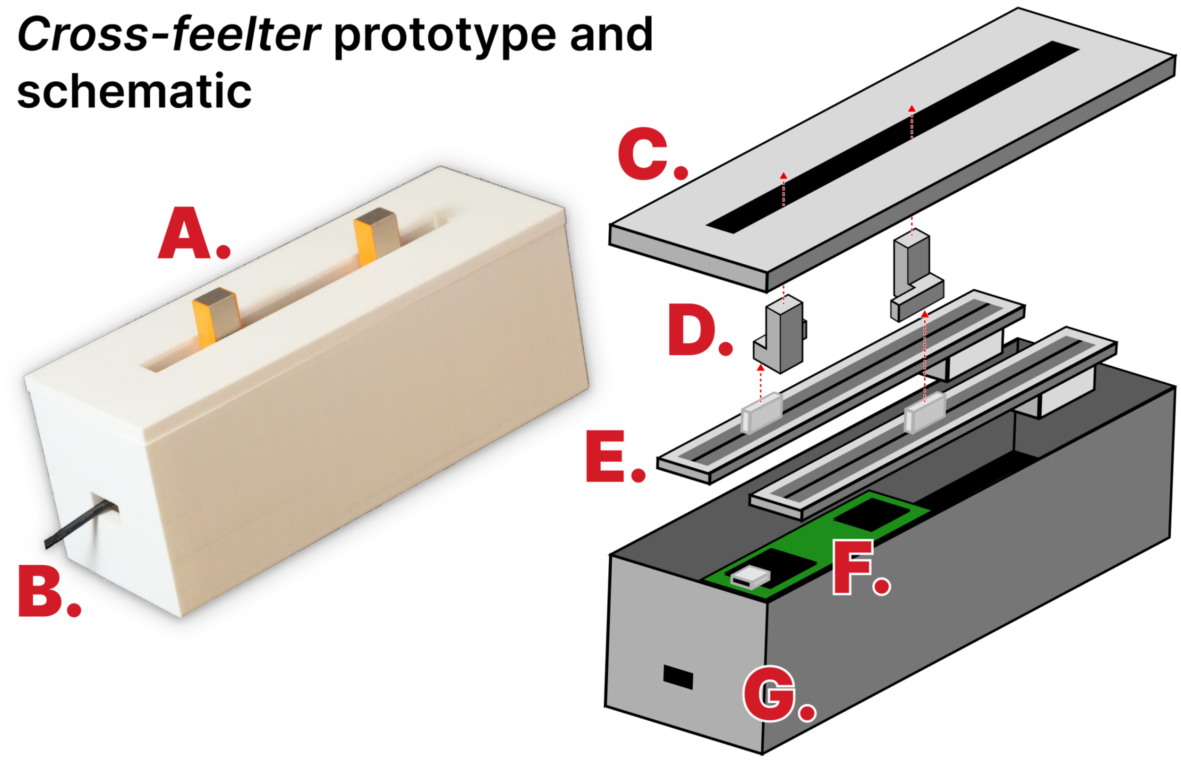

Figure 7.2: A. Our prototype *cross-feelter*. B. USB output (to computer). C. 3D printed rail cover. D. 3D printed knobs that consolidate rails into one. E. 2 motorized linear potentiometers. F. Arduino board. G. 3D printed casing.

for the housing components are available in our supplemental materials and on Thingiverse at `[REDACTED]`. The total cost of hardware components is $30–50 USD, which we designed intentionally to enable replication by the DIY-AT and maker communities [79, 148].

7.5.1 Interaction Flow

We begin with an idealized walkthrough of what a blind analyst actually does when using the cross-feelter to perform a cross-filtering task. This is the ground truth from which the hardware design choices derive; we describe the hardware in Section 7.5.2 with reference to the steps below.

Consider an analyst exploring a dataset of local rent prices and transit access across city neighborhoods. Three linked histograms are rendered in sequence on the refreshable braille display. The analyst wants to ask: *do neighborhoods with the highest rents also have better transit access?* At startup, the left knob sits at 0% and the right at 100%, representing an unfiltered view of the full distribution.

Step 1: Feel and filter. The analyst feels the starting state of the tactile data representations on the refreshable braille display, navigating between each chart and hearing supplemental screen reader descriptions alongside the touchable shapes formed from the pins. The analyst moves their hand or hands to the feelter device and slides the left knob rightward, narrowing the filter from below. As they move, an ARIA-live announcement reports the current percentage and knob name via the screen reader, providing precise serial feedback for positioning (G2). The analyst moves the right knob slightly leftward, hearing another announcement. The two knobs now define a range, say 75% to 90% of the rent distribution.

Step 2: Perceive the consequence. During filtering, the analytical environment updates linked histograms immediately on each knob movement. The analyst navigates the braille display to the transit histogram without touching the knobs. Because the knobs are motorized and maintain position, the filter state is preserved and readable by touch at any time (G1). They feel the updated distribution and compare it to the shape they perceived in the unfiltered state.

Step 3: Iterate. The analyst slides the left knob further right, narrowing to the top 10% of rents. The display updates. They feel the new shape. The pattern sharpens or changes, and they iterate again, building toward a hypothesis in the same cycle that sighted analysts perform visually. Each knob adjustment is a computational query: a gesture that constructs a question and immediately exposes its answer.

7.5.2 Hardware Design

The cross-feelter uses two motorized linear potentiometers (commonly called motorized faders in sound design hardware) mounted side by side in a 3D-printed housing (see Figure 7.2). Each fader provides a physical knob on a bounded rail.

Motorized, absolute-position faders (G1, G5). We chose rail-based faders over alternatives (touchpad, rotary encoder, joystick) for two reasons grounded in the G1 requirement. First, the fader is *absolute*: the knob's physical position on the rail directly corresponds to the filter boundary value, with no drift or accumulation. Second, the fader is *motorized*: the system can programmatically set knob position as well as read it. This bidirectionality is essential for cross-filtering: the system must be able to reset filter state on condition changes, counterbalance, and initialization, not only receive input. Prior work on brush-based accessible interfaces [144] used a touchpad, which provides no queryable positional state when the hand is lifted; prior work using non-motorized sliders [185] cannot be programmatically positioned. The motorized fader solves both problems.

Each rail terminates in a tactile boundary, a strong edge inset 4mm from the housing wall per tactile design guidelines [54], so the analyst can feel both the knob position *and* its position relative to each end of the scale (G1). The API reads and writes fader position every 5ms, providing up to 1024 units of precision, which maps well to histogram bin resolutions in practical cross-filtering datasets (G6).

Merged-rail housing (G3). We designed two housing configurations: one exposing the rails independently, and one using angled knobs to place both rails along a single track (Figure 7.2, part D). The merged-rail design, used for all data tasks in the study, ensures the left knob remains to the left of the right knob at all times (they cannot cross over). This preserves the functional semantics of a spatial filter range: left boundary is always left, right boundary is always right, matching the computational filter and the visual histogram (G3). The independent-rail configuration was used for the video interaction training task (Section 7.6.4), where the two rails served different functions (volume and position).

Our dual-fader design is similar in form to motorized fader-based systems proposed for axis interaction in AR/VR data environments [25, 147]; however, those systems presuppose sighted users operating in 3D visual contexts. Our design differs in three ways: (a) the housing constrains the two faders to a single interaction dimension to support filter semantics; (b) the motorized positioning enables two-way synchronization with the data interface; and (c) the entire system is designed for eyes-free operation as the primary use mode.

ARIA-live auditory feedback (G2). When a knob is moved, the interface emits an ARIA-live announcement to the screen reader reporting the new filter percentage. ARIA-live is set to `polite` mode, meaning announcements queue behind any active screen reader speech rather than interrupting it. This provides precise, serial auditory feedback for filter positioning without competing with other auditory streams in the environment (G2). Tactile feedback (the knob's physical position) and auditory feedback (the ARIA announcement) thus serve complementary roles: tactile provides continuous, queryable approximate state; auditory provides precise on-demand readout triggered by movement.

7.5.3 Analytical Environment

The cross-feelter operates within a web-based analytical environment purpose-built for this study (Figure 7.3). The environment presents three linked histograms derived from different variables in a shared dataset. Each histogram is rendered both visually (for debugging and sighted collaborators) and as a 60×40 pin array sent to the Dot Pad refreshable braille display.

Cross-filtering backend (G4, G5). We use Mosaic [67] to manage the cross-filtering logic. Mosaic pre-aggregates filter results across binned input values, enabling fast, low-latency responses to filter changes even over larger datasets (G5). It also provides a straightforward API for specifying a brush-filter interaction layer over a histogram dimension, including programmatic control of filter state and undo actions (G5). When the user moves a fader knob, the new filter boundaries are sent to Mosaic, which immediately updates the linked histograms, which are then re-rendered on the braille display. The render cycle from fader movement to updated tactile output is 500–1200ms on the Dot Pad hardware, fast enough to support iterative exploration, though slower than visual cross-filtering systems and a recognized limitation (Section 7.10).

Braille display output (G2). The Dot Pad provides a 60×40 binary pin array (pins raise or lower on command). We render each histogram as a bitmap at this resolution, trading fidelity for speed relative to a high-resolution braille embosser (which takes 3–4 minutes per page). At any time, the braille display shows one chart; the analyst navigates between charts using the screen reader. The visual representation of the target pin array is visible to sighted observers and collaborators in the environment (Figure 7.3, part A), illustrating the resolution gap that future advances in refreshable display technology could close.

Screen reader compatibility. The analytical environment is a standard web interface, accessible via any JAWS, NVDA, or VoiceOver-compatible screen reader. Cross-filter input controls are exposed as labeled range inputs so they can be operated by screen reader alone (providing the baseline condition for our study, Section 7.6). The ARIA-live region for fader announcements is separate from these controls and does not interfere with standard screen reader navigation of the interface.

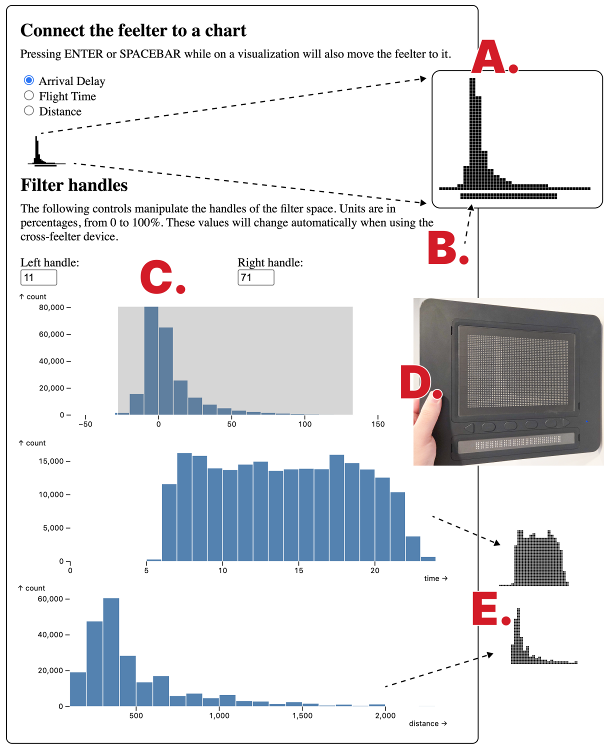

Figure 7.3: Our analytical environment. A. a 1 to 1 translation between visual and tactile that represents a 60x40 pixel-to-pin array. B. A tactile version of the filter location (if on the chart being focused). C. Cross-filtering controls, including text inputs for screen-reader only manipulation. D. Refreshable tactile display output. E. Example renderings of the other 2 charts.

7.6 Data Exploration Study

We conducted a within-subjects study comparing cross-feelter-based cross-filtering (CF condition) against screen-reader-based cross-filtering (SR condition) across structured and open-ended data exploration tasks. Sessions were 90 minutes, conducted in person, and approved by our IRB.

7.6.1 Study Design

Overview. Our study employed a 2×2 Latin square design, fully counterbalanced across two factors: (1) condition order (CF-first vs. SR-first) and (2) dataset order (flight data vs. neighborhood data), producing four participant groups (Table 7.1).

Within each condition, participants performed two tasks on the same dataset: a 3-minute structured data-relationship task followed by a 10-minute open-ended data exploration. Counterbalancing both orders controlled for dataset familiarity, learning effects, and dataset-specific task difficulty.

Conditions. In the **CF condition**, participants used the cross-feelter (merged-rail housing;

Table 7.1: 2×2 Latin square study design. CF = cross-feelter condition; SR = screen reader condition.

Group	Condition order	Dataset order
A (n=4)	CF → SR	Flight → Neighborhood
B (n=4)	SR → CF	Flight → Neighborhood
C (n=4)	CF → SR	Neighborhood → Flight
D (n=3)	SR → CF	Neighborhood → Flight

Section 7.5.2) to manipulate cross-filter bounds, with the Dot Pad displaying linked histograms. In the **SR condition**, participants used their screen reader to operate labeled text input fields exposing the same filter controls, with the same Dot Pad display. The analytical environment was identical across conditions; only the input method differed.

Session procedure. Each session followed this order regardless of group:

1. Tech setup, baseline interview, introduction to tactile histograms and cross-filtering, baseline anxiety Likert question *(20 min)*
2. Device familiarization: video interaction task and brief semi-structured interview *(15 min)*
3. First data condition: structured task, open-ended exploration, Likert-scale interview *(20 min)*
4. Second data condition: structured task, open-ended exploration, Likert-scale interview *(20 min)*
5. Closing semi-structured interview *(15 min)*

7.6.2 Participants

We conducted our study with 15 blind participants: 8 without prior professional data expertise and 7 with it (data analysts, scientists, or engineers currently or recently employed in data-related roles). 7 participants identified as male, 8 as female.

Participants without data expertise, plus 2 with it, were recruited locally through community outreach. The remaining 5 participants with data expertise were recruited through professional networks and coordinated at mutual academic conferences, due to difficulty recruiting blind data scientists locally. All participants were screened for eligibility (blind, screen reader proficient, 18 or older, US citizen or permanent resident), informed of risks and voluntary participation, and compensated with a $50 gift card.

7.6.3 Datasets

Study tasks used two datasets selected through a community data-preference workshop. Pilot sessions revealed that participants were not engaged by our initial generic dataset, so we held a workshop at a local braille and talking book library (n=21 community members). Participants voted on datasets from topics including local politics, housing, and community life. The two highest-voted were: (1) rent prices across city neighborhoods, and (2) fish fry locations across city neighborhoods (a dataset reflecting a regional Catholic Lent food tradition observed broadly across denominations). Both are publicly available. A generic flight dataset was included as the second counterbalanced dataset.

Each dataset was configured with three histogram variables from a shared source, with cross-filter tasks designed to target a 75%–100% filter range producing a clear, unambiguous relationship between variables.

7.6.4 Device Familiarization: Video Interaction Task

Rationale. Before data tasks, participants completed a device familiarization activity using the cross-feelter to control a video feed rather than a data chart.

The goal was to build fluency with the motorized fader mechanism (absolute-position input, motorized output, the relationship between physical knob position and system state) without pre-exposing participants to the cross-filtering metaphor, data tasks, or the study datasets. A practice task using data histograms would have constituted condition-specific training, introducing learning effects that could confound the within-subjects comparison. Video scrubbing and volume control are familiar and intuitive interaction goals for all of our users, which served as a way for all of them to understand how to operate the device.

Task. Participants were asked to find the name of the person in the video recording, announced early in the clip. This simple information-retrieval goal required genuine device use while keeping cognitive demand low before the main study.

7.6.5 Introduction to Tactile Histograms and Cross-Filtering

At session start, after consent and the baseline interview, we provided a scripted introduction to tactile histograms and cross-filtering. We explained histogram structure, introduced the concept of linked histograms sharing a source dataset, and demonstrated cross-filtering with three histograms from a fruit example dataset, using two sheets of paper held over one histogram to physically simulate a spatial filter.

We framed filtering explicitly as an analytical activity: "filtering is a way to ask questions more than to find answers, to look for possible relationships that might exist in the data." Two follow-up questions gauged comprehension and interest before proceeding.

7.6.6 Data Tasks

7.6.6.1 Structured data-relationship task

Participants cross-filtered the dataset and answered a three-option multiple-choice question within three minutes. The question and options were available via screen reader in the interface; the facilitator restated them only if asked and did not otherwise intervene.

For the flight dataset: *"What is the relationship between filtering for longer flight distances and the arrival time of flights?"* (Correct: longer flights tend to arrive earlier compared to all flights.)

For the neighborhood dataset: *"What is the relationship between filtering for a higher number of fish fry locations and the rent price of a neighborhood?"* (Correct: neighborhoods with more fish fry locations tend to have higher rent prices.)

We measured task completion rate (binary), time to completion (seconds), filter placement precision (distance of handles from target positions of 75% and 100%, scored in $\pm 4\%$ increments awarding 0.5/0.3/0.1 points per handle tier), and answer accuracy (binary).

7.6.6.2 Open-ended data exploration

Participants then performed a 10-minute open-ended exploration of the same dataset using a prompted, concurrent think-aloud method [2]. Before beginning, we framed the activity: "the purpose is not to *answer* questions but to *come up with* questions. This is how many data scientists explore unfamiliar datasets, by looking for possible relationships."

We measured: (1) *computational queries*: any filter manipulation followed by braille display inspection of one or more linked charts (filter changes without output inspection did not count); and (2) *spoken queries*: any data-related question voiced aloud, regardless of formality or answerability via cross-filtering.

7.6.7 Measures and Analysis

Quantitative analysis. Objective measures (completion rate, time, precision, accuracy, query counts) and subjective measures (5-point Likert scales for stress, cognitive load, future-task anxiety, and enjoyment) were collected per condition. A pre-study baseline anxiety item provided a pre-condition reference.

Analysis was conducted in Python (code and de-identified data in supplemental materials). Objective continuous measures were compared using paired t-tests. Likert-scale differences were evaluated using Wilcoxon Signed-Rank tests given the ordinal nature of the data. We report means, standard deviations, test statistics, and p-values; group-level results (data experts vs. non-experts) are reported separately where they diverge.

Qualitative analysis. Qualitative data comprised session audio/video recordings, facilitator notes, and closing semi-structured interview transcripts. The interview covered three topics: overall experience and condition comparison, interface and environment feedback, and future application ideas.

Analysis proceeded in two stages. In the first stage, two members of the research team independently reviewed all recordings, notes, and transcripts, generating initial codes at the utterance level. Codes captured themes across four domains: device usability (physical interaction, feedback quality, learning curve); condition comparison (speed, cognitive load, perceived control); analytical engagement (question generation, hypothesis articulation, confidence with data); and future use speculation. The two coders then resolved all disagreements through discussion, reaching consensus before proceeding.

In the second stage, codes were organized using affinity diagramming [63], conducted collaboratively by members of the research team working from the reconciled code set. Resulting themes inform the qualitative findings in Section 7.7 as well as our future use cases in Section 7.9.

7.7 Results

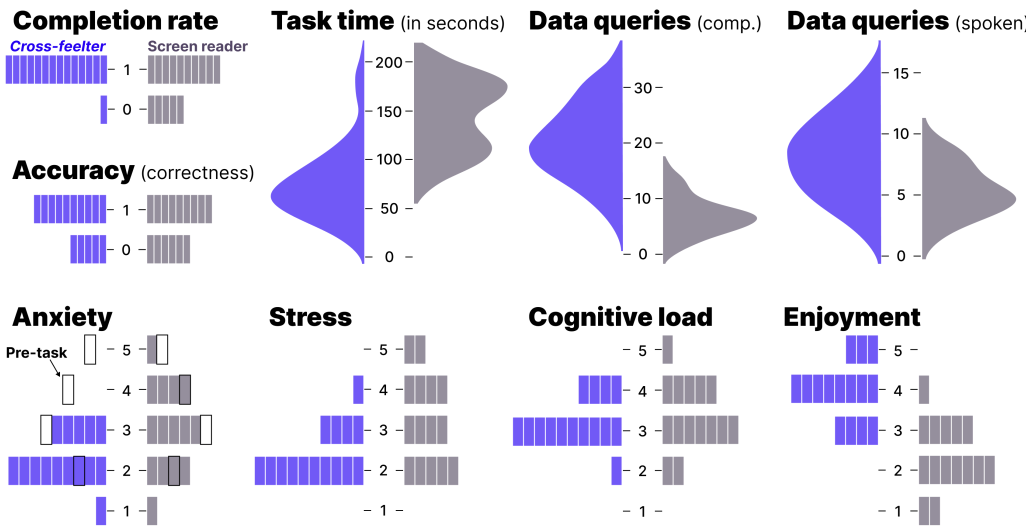

Figure 7.4: Results. Completion rate and accuracy are counts. Task time, data queries (computational), and data queries (spoken) are objective, observational measures. Anxiety, stress, cognitive load, and enjoyment are Likert-scale results from 1 ("very little") to 5 ("quite a lot").

Our analysis compared 15 blind participants performing cross-filtering tasks under two conditions: using the cross-feelter (**CF**) versus using a screen reader alone (**SR**). We report objective and subjective measures in turn. Full analysis code and de-identified data are available in our supplemental materials.

7.7.1 Objective Results

7.7.1.1 Substantially improved task completion rate and speed

For the structured data-relationship task, participants achieved a significantly higher completion rate with CF (mean = 0.93, SD = 0.26) compared to SR (mean = 0.60, SD = 0.51; paired t-test: $t = 2.646$, $p = 0.019$). Task completion time was significantly lower in the CF condition (mean = 75.53s, SD = 36.14) than in the SR condition (mean = 143.93s, SD = 34.65; $t = -6.876$, $p < 0.001$), a 90% improvement in speed.

7.7.1.2 Comparable accuracy and precision

Accuracy (correct multiple-choice answer) was higher in the CF condition (mean = 0.67, SD = 0.49) than SR (mean = 0.53, SD = 0.52), but this difference was not statistically significant ($t = 1.468$, $p = 0.164$). Notably, when filtering for only completed tasks, SR users showed marginally higher accuracy (SR: mean = 0.78; CF: mean = 0.71), though again non-significant ($t = 0.555$, $p = 0.594$). We interpret this in Section 7.8.2.

Filter placement precision (scored by proximity to target handle positions of 75% and 100%) was marginally higher in CF (mean = 0.66, SD = 0.37) than SR (mean = 0.59, SD = 0.34), with no significant difference ($t = 0.617$, $p = 0.547$).

7.7.1.3 Substantially increased quantity of data queries

During the open-ended exploration task, CF elicited substantially more computational queries (mean = 20.13, SD = 6.45) than SR (mean = 7.00, SD = 3.18; $t = 9.567$, $p < 0.001$), a 188% increase. Spoken queries were also significantly higher in CF (mean = 8.20, SD = 3.14) than SR (mean = 5.33, SD = 1.99; $t = 3.765$, $p = 0.002$), a 54% increase.

7.7.2 Subjective Results

Subjective measures varied between participants with and without professional data expertise; we report group-level differences where they diverge meaningfully.

7.7.2.1 Reduced stress

Stress ratings were significantly lower in the CF condition (mean = 2.40, median = 2.0) than SR (mean = 3.20, median = 3.0; Wilcoxon $W = 4.000$, $p = 0.013$). The effect was driven primarily by participants without data expertise (CF: mean = 2.62, SR: mean = 3.75; $W = 2.500$, $p = 0.047$). Data experts reported similarly low stress across both conditions (CF: mean = 2.14, SR: mean = 2.57; $W = 0.000$, $p = 0.083$).

7.7.2.2 Comparable cognitive load

Cognitive load showed an observed difference in means (CF: mean = 3.13, SR: mean = 3.47) but was just outside statistical significance ($W = 4.000$, $p = 0.059$). We interpret both conditions as producing comparable cognitive load during task performance.

7.7.2.3 Lower future-task anxiety

Participants in the CF condition reported significantly lower anxiety about performing a future timed data task using the same tools (CF: mean = 2.27, SR: mean = 3.00; $W = 10.000$, $p = 0.017$). CF anxiety was also significantly lower than participants' baseline anxiety measured before any tasks ($W = 5.000$, $p = 0.003$), while SR anxiety was not ($W = 8.000$, $p = 0.132$), suggesting the cross-feelter reduced anticipatory anxiety relative to participants' pre-study state, while the screen reader did not.

7.7.2.4 Substantially improved enjoyment

Enjoyment was substantially higher in CF (mean = 3.93, median = 4.0) than SR (mean = 2.33, median = 2.0; $W = 0.000$, $p = 0.001$). This effect was strongest for participants without data expertise (CF: mean = 3.875, SR: mean = 2.000; $W = 0.000$, $p = 0.015$). One participant's unprompted remark captures the qualitative texture of this finding: *"Sighted people get to have this much fun when they work with data?"*

7.8 Discussion

Our findings suggest that cross-perception shows meaningful promise as an approach to accessible analytical data interaction. We discuss our results in relation to our four research questions, then develop the broader implications for the field.

7.8.1 Research Question Responses

R1 (Exploratory): How might we envision and formalize a tactile interaction approach that enables coordinated, non-visual interaction across multiple spaces? Cross-perception, as formalized in Section 7.4, provides one answer: an interaction technique grounded in principles from visual cross-filtering, spatial cognition, and blind information interaction, instantiated in a device with persistent-state motorized input and a linked tactile display.

The six design goals (G1–G6) constitute the formalization, and the design vignette demonstrates that the underlying cognitive mechanisms are ecologically grounded in expert blind reading practice. We intend this formalization to be extensible and generative; we demonstrate this extensibility in our future use cases in Section 7.9.

R2 (Quantitative: Objective): In what measurable ways can we observe improvements to blind data interaction using our tactile interaction approach? Cross-perception produced significant improvements in task completion rate (+55% absolute), task completion speed (90% faster), and quantity of data queries during open exploration (+188% computational, +54% spoken). Accuracy and precision were statistically comparable between conditions, though with a nuance discussed in Section 7.8.2 below.

R3 (Quantitative: Subjective): What are the self-reported benefits, opportunities, and challenges that blind people experience with our tactile interaction approach? Participants using the cross-feelter reported significantly lower stress, lower future-task anxiety, and substantially higher enjoyment than in the screen reader condition. These improvements were particularly pronounced for participants without data expertise, suggesting cross- perception may lower affective barriers to data work for blind users who are new to or learning data analysis.

R4 (Qualitative): In what ways do blind data scientists imagine extending our approach? Participants, especially those with professional data expertise, proposed substantive extensions including: a pre-rendered "interaction cube" for environments without a refreshable display; feelter-only navigation of data arrays and sequences; sonification paired with rail brushing; and multi-dimensional panning, zooming, and scatterplot brushing using multiple feelter units. These proposals are detailed in Section 7.9.

7.8.2 Accuracy and Precision Considerations

The comparable accuracy and precision results merit careful interpretation. Participants maintained similar correctness across conditions while performing substantially faster in CF. However, when filtering for completed tasks only, SR users showed marginally higher accuracy. One interpretation is that the additional time the serial screen reader interface imposed allowed for more deliberate verification before committing to an answer, while the cross-feelter's speed encouraged more rapid, and occasionally less careful, decision-making.

This is not necessarily a weakness of cross-perception. In genuine exploratory analysis, speed of iteration is often more valuable than precision on any individual query, since the point is to generate and test many hypotheses rather than verify a single one carefully. Some precision can be a little "fuzzy" if the gained effect has analytical benefit overall [84]. This difference in interaction patterns warrants further investigation and suggests the two approaches may support different analytical strategies rather than being simply faster and slower versions of the same strategy.

Future iterations should explore mechanisms that preserve the speed benefits of direct tactile manipulation while supporting careful verification when the task demands it (or example, a confirmation gesture or a "hold" mode that locks the filter and prompts output inspection before advancing).

7.8.3 The Case for Analytical Interaction as a Primary Research Site

The result most significant for the field is not the speed improvement but the query count. 20 computational queries in 10 minutes compared to 7 is not merely a faster version of the same activity. This increase in speed also reflects a qualitative shift in what participants were doing. In the CF condition, participants iterated hypotheses through direct manipulation; in the SR condition, they navigated toward output values that were structurally separate from their space of interaction. This maps directly to the distinction between analysis-oriented and access-oriented interaction: the screen reader, however capable at navigation, kept participants in a mode that structurally separated interaction and information; a descriptive mode. The cross-feelter enabled a *generative* mode where interaction and information are coupled.

The enjoyment gap is legible through the same lens. One participant's unsolicited remark, *"Sighted people get to have this much fun when they work with data?"* describes the experience of analytical agency. This is what visualization researchers have understood for decades to be the core value of interactive exploratory analysis: an experience of power over and with data interaction. This agency has not been consistently available to blind data practitioners, and the affective signature of its absence is measurable in our stress and anxiety results.

The implication is that analytical interaction accessibility constitutes a distinct research problem, not an extension of the accessible rendering/description problem. The accessible visualization community has made substantial progress on helping blind users read and navigate data. What our results suggest is that reading data and analyzing data are sufficiently different activities that the techniques developed for one cannot simply be adapted for the other: they require different interaction architectures, different evaluation approaches, different design primitives, and sometimes even different hardware. We encourage researchers working at this intersection to treat analytical interaction as a primary site of inquiry: studying the existing practices of blind data analysts during open-ended exploration, identifying what analytical operations (filtering, outlier detection, hypothesis iteration, assumption testing) current tools support or foreclose, whether existing tools support direct interaction with data representations in these tasks, and designing interaction techniques grounded in these blind-centered findings, rather than in adaptations of what sighted-user interfaces already provide.

7.8.4 Toward Hardware and Physicalization in Accessible Data Interaction

Building on that previous point, a recurring practical constraint in accessible data interaction research is the emphasis on solutions that work within existing consumer device ecosystems: screen readers, keyboards, web browsers, and computationally synthesized audio. This is reasonable: it maximizes immediate reach and avoids asking blind users to acquire specialized equipment if the capabilities are already realized on existing computer and mobile hardware. But it also implicitly constrains the design space to what those platforms can support, and some of the interaction properties that matter most for analytical work (such as persistent physical state, absolute spatial position, simultaneous bilateral manipulation) are not well-approximated by software alone on current consumer hardware. Existing work (mentioned in Section 7.3) hasn't ignored this problem, but this existing work is a small minority. Priorities in research have predominantly focused on leveraging *existing* software, hardware, and computational models towards problems and barriers blind people face, rather than innovation towards *novel* hardware and physicalization systems.

The cross-feelter's component cost ($30–50 USD) and the availability of all schematics under an open license suggest that the hardware barrier is lower than it might appear. The more substantive barrier is that the visualization and accessibility communities have not yet developed strong shared practices around hardware prototyping and physical interaction design. Work from the tangible and physicalization communities is directly relevant here, including the insight that physical materials afford interaction properties that screens do not, among them permanence, resistance, and the kind of spatial memory that our design vignette illustrates. Bringing these ideas into contact with accessible data interaction research is, we believe, a productive direction.

The future use cases our participants proposed, discussed in detail in Section 7.9, illustrate the range of what becomes thinkable when the design space is extended beyond the screen: pre-rendered interaction outputs for environments without powered displays, motor-driven data array navigation, multi-axis spatial exploration using multiple feelter units, and sonification navigated through physical rail position. These proposals emerged from a structured speculative futuring exercise [32] with participants who are themselves blind data practitioners. They are worth treating as design starting points, not only as participant feedback.

The field has successfully applied careful empirical and design methods to the challenge of making non-visual data representations perceivable. Extending that rigor to the challenge of making data manipulable through the hands, with all the hardware and prototyping work that entails, is the direction we are advocating.

7.9 Future Use Cases

The following use cases were developed through the speculative futuring exercise [32] conducted as part of our closing interview, with contributions from participants and the research team. We

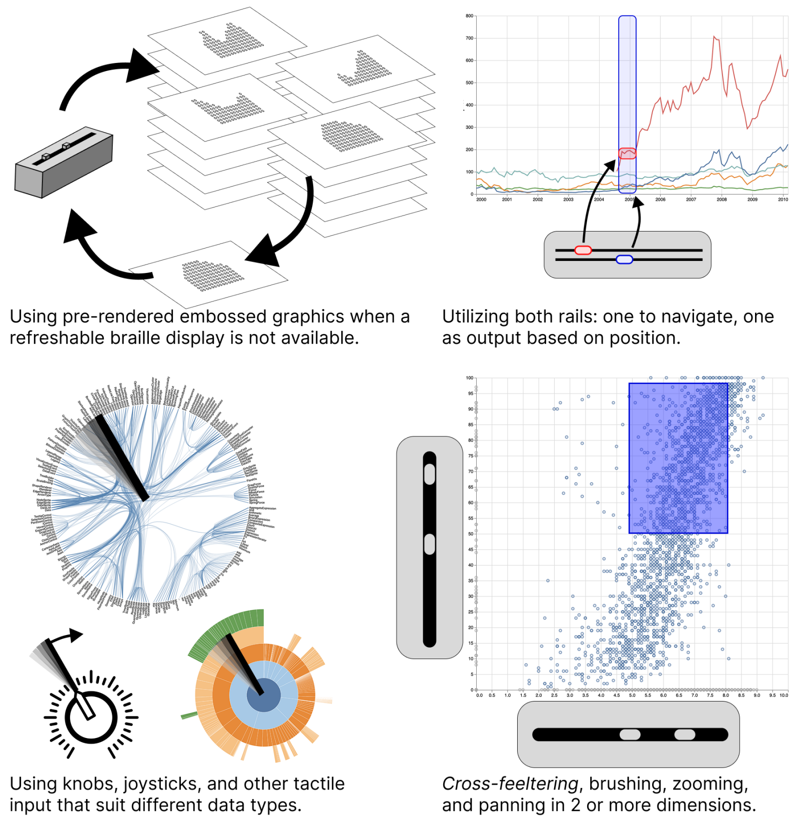

Figure 7.5: Speculative future use cases for our *cross-perception* approach, contributed by our participants.

present them as design starting points grounded in the cross-perception framework, each addressing a limitation of the current prototype or extending the approach to new contexts. Figure 7.5 illustrates all four cases.

7.9.1 Cross-Perception Without a Powered Display: The Pre-Rendered Interaction Cube

Refreshable braille displays remain expensive ($8k–$20k USD) and are not universally available. The current prototype depends on the Dot Pad to render cross-filter output in real time, which means the full cross-perception loop is unavailable to users without access to one.

One solution is to pre-render the interaction outputs for a given dataset before a session begins. The cross-feelter's filter operates over a bounded, discretizable range; it is feasible to pre-compute a set of interaction results at binned filter positions and emboss them onto paper as an "interaction cube" of tactile states, analogous to the data cube that Mosaic pre-computes for fast cross-filtering over large datasets [67]. After each filter manipulation, the user turns to the corresponding page in a booklet of embossed tactile charts, reducing the interaction loop from 3–4 minutes (a full embosser render) to roughly 10 seconds. This approach trades the continuous, real-time update of the powered display for a discrete but physically permanent output, which has its own advantages for users who prefer to annotate or revisit prior states or even compare multiple outputs simultaneously, like in the dual-task comparison vignette in Section 7.4.1.

7.9.2 Sequential Data Navigation with Continuous Positional Feedback: Feelter-Only Chart Interaction

Serial navigation of data arrays using a screen reader is slow because the user must step through observations one at a time without any persistent sense of their position within the full range [36].

A single cross-feelter could replace this with continuous physical scrubbing. Most charts and datasets have finite, bounded observations in any given dimension; these bounds map naturally onto the full range of a fader rail. One rail would scrub through the ordered observations as the user moves the knob, with the screen reader announcing the current value. A second rail, motorized, would move its knob to a position encoding the corresponding value in a second data dimension, like a rising and falling bar in a bar chart.

More broadly, different input devices could be matched to different data topologies. A rotary dial would suit circular or periodic data (seasonal patterns, directional data); a joystick would suit data with an infinite or continuous range such as a parametric function or model output.

7.9.3 Sonification Navigation via Physical Rail Position

Navigating within a sonification (seeking to a specific region, comparing two positions, returning to a reference point) is currently difficult even on touch-enabled devices, because scrubbing through audio provides no queryable positional state when the hand is not actively in contact with the surface [21, 22, 23]. This is the same object-permanence problem that motivated the cross-feelter's design for cross-filtering.

A cross-feelter rail could serve as the navigation input for a sonification: physical position on the rail maps to a position within the audio timeline or data domain, and the motorized mechanism maintains that position when the hand is lifted. The user can navigate to a region of interest, remove their hand, let the sonification play, return to the rail, and feel exactly where they left off.

7.9.4 Multi-Dimensional Spatial Interaction: Rails as Input and Output

Two cross-feelter units, one arranged as an x-axis controller and one as a y-axis controller, could support spatial panning, zooming, and brushing over tactile maps or scatterplots, which are interactions that currently lack a satisfactory non-visual implementation because pinch-and-zoom gestures provide no queryable positional state for the zoom extent or pan position. Each feelter unit would maintain the current viewport bounds in its two rails, readable by touch at any time.

More substantially: the motorized rails could serve as *system output* for spatial queries. When a user asks "where is this object?" of a map or image, the system could drive the rails to the corresponding x and y positions and bounds. This would let the user feel the location directly rather than receiving a verbal coordinate and build a spatial mental model of a spatial layout.

7.10 Limitations

Our study's controlled task design (three-minute timed tasks with multiple-choice answers) prioritized speed and query quantity, which are the right measures for a proof of concept but do not capture the full texture of blind data analysts' real exploratory workflows. Activities common in open-ended analysis, such as wrangling messy data, identifying outliers, and questioning assumptions, were not represented. An ethnographic or longitudinal study would be better suited to those questions, and we encourage that work.

The Dot Pad's 500–1200ms render latency falls outside the max 500ms threshold at which Liu and Heer identify measurable degradation in exploratory analysis quality [99]. Our query count results are therefore a conservative lower bound on what cross-perception could support with faster display hardware.

Novelty effects may have inflated subjective measures. The enjoyment and anxiety improvements may attenuate with extended exposure, though the effect may persist in contexts such as some classroom, work, or onboarding settings, where engagement is itself valuable.

Finally, we compared against a screen reader baseline because screen readers are the dominant interaction tool for blind users working with data and all of our participants were proficient in screen reader use. More granular comparative work against other emerging tactile and voice-based input strategies would further characterize where cross-perception sits in the broader interaction space.

7.11 Conclusion

We introduced cross-perception, a tactile interaction technique that adapts cross-filtering for blind data analysts by preserving the simultaneity of input and output perception that makes the technique analytically valuable. We formalized it through six design goals grounded in principles from visual analytics, spatial cognition, and blind information interaction; instantiated it in the *cross-feelter*, a low-cost motorized fader device; and evaluated it against a screen reader baseline with 15 blind participants.

In our study, Cross-perception substantially improves task completion speed, data query generation, and subjective experience, with particularly strong effects for participants without prior data expertise. The result that matters most is not the speed improvement but the query count: participants using the cross-feelter generated nearly three times as many analytical queries during open exploration, reflecting both a quantitative improvement and a qualitative shift away from navigating data toward interrogating it.

We argue that analytical interaction accessibility—the capacity for blind users to filter, conjecture, and test hypotheses through direct manipulation—is an open research problem that deserves sustained attention as a primary site of inquiry, distinct from accessible rendering. The future use cases our participants proposed suggest that extending the design space to include hardware and physicalization opens directions that software-only approaches cannot reach.

Part VI

Conclusion

Chapter 9
Discussion & Future Work

9.1 What is a "tool?" A reflection on the social and material identity of tools

In the introduction of this dissertation, I use the example of a hammer: a hammer can destroy and it can construct. So is the *use* of a technology what constitutes it? Do we understand the hammer as the *thing we swing, to destroy and to build*? Should we?

This thesis engages domains of tools and tool-making for accessibility: evaluation, navigation, interaction, and personalization. But these categories for work do not fully characterize the upstream conditions that our software systems and data interfaces inherit.

In my work specifically on accessibility, a larger social reality becomes apparent that shapes the question, "what is a tool?" far more than how an individual might use one, or the domains of work that our tool-making inhabits. My research journey has navigated multiple social and political thresholds, from changes to law in the European Union, to the enactment of Title II as part of the update to the Americans with Disabilities Act. These laws have motivated a significant interest in accessibility research, solutions, guidelines, and technologies. In the midst of this, we have seen the rise of overlays and generative AI solutionism [64] and subsequent lawsuits and grass-roots resistance.

For my work, this is mostly good news. Legal change produces motivation, and even with predatory technology attempting to address real problems, pushback is widespread and active. But this paints a picture of the reality that my work inherits: many tools cannot even be used, or cease to be used, if there is not a social, political, and material set of conditions in place motivating those tools, providing resources for their construction, regulating their use, and examining the outcomes of what they accomplish. Tools and technologies are often a response to social, cultural, political, and legal realities that we first negotiate.

I recently spoke on this at a keynote in Australia, on how a hammer isn't *just* a tool and that the idea that "the only thing that matters is how a tool is used" limits how we really understand tools. Instead, I spoke about how a standard, household hammer requires iron and wood. That alone leads to a whole universe of different questions. Western Australia's conservation efforts were disrupted when a significant amount of natural iron was discovered in a wildlife preserve. So laws were passed and now iron is mined there. That iron is largely exported. And Australia then, whether with Australian iron or not, mostly imports their small tools. Iron is sent out, and through a complex network of trade (likely indirectly related to the iron), hammers are brought in. A "hammer," to even exist at all, relies on multiple levels of human governance, international relations, and complex infrastructures of trade.

And while my metaphor is largely motivated to encourage younger practitioners to consider the "iron mines" in the technologies they use, such as modern generative AI, it is also an area that is not adequately explored and addressed in terms of accessibility research.

Research on accessibility is dependent on funding, which is often dependent on political priorities and action. Depending on the current social and political state of the world at large, accessibility research itself may never gain the opportunities required in order to innovate and produce new tools at all. And as the US's 2025 federal cuts to research demonstrated, millions of dollars devoted to accessibility research can be lost to political agendas. It is for this reason that engagement with policy recommendation and guidance is essential. Personal political activity and involvement is also essential. Researchers who genuinely believe in accessibility as a human right or as a dignity that all people deserve should work with policymakers to ensure that there are material and structural resources in place for this work to continue. We cannot naively believe that technology, divorced from the realm of social and political forces, is capable of solving accessibility barriers [152]. Without enforcement and threat of litigation, very little accessibility work has been accomplished in the past by technology companies alone.

Not featured in these chapters (as they were merely stapled in research papers from previous publications) is the policy and outreach work involved in seeing that work like *Chartability* and *Data Navigator* are used in real contexts, including by organizations that govern and influence the lives of many people. Immediate incentives to produce novelty may not be enough to sustain the larger socio-cultural and political ecosystems that our work participates in and is downstream from. We must also get involved.

9.2 "Applied" accessibility work and *low research*

I came into an academic research environment already as an engineer. I had questions to big problems, especially about how tools might shape human behavior and how we can use that towards ethically good outcomes. But *Chartability* was a project I made first for myself, to engage the problems I, personally, was facing.

And *Data Navigator* was motivated by my existing experience making visualizations navigable. I knew we had to build better tools, because continually trying to make a hierarchical substrate like HTML work for complex, non-hierarchical relationships simply limited what was possible and easy to do.

These two projects then led to many opportunities to test them in applied contexts. Our CZI grant was motivated towards the application of both, Adobe was interested in *Data Navigator*, and the University of Wisconsin's GIS folks wanted to figure out how to make a complex map navigable, *Data Navigator* or not. Yet, as a researcher I wasn't incentivized to pursue these projects. In fact, they all proved to have no immediately apparent novelty. It was a risk to spend my time in these spaces.

Applied work is structurally and socially devalued in academic research environments, seen often as a lower form of research and design compared to more "pure" forms of knowledge production. (I conjecture that in part this is due to the scope of knowledge that practitioners produce: they work to take broad knowledge and apply it to specific, small problem spaces. Foundational research tends to assume that broad knowledge does not then bubble up from this work, that it is a one-way flow.)

In my first year as a PhD student I was given the advice by a senior researcher not to hold my experience as a practitioner too highly because, "practitioners often don't know what they are doing." But I did, of course, to some degree. Yet, this attitude is pervasive in some spaces and is a problem because it limits what kinds of knowledge we care to attend to.

Initially, I felt that with *Chartability* I had lucked out. But after 4 more collaborations which resulted in funding and 2 research publications, I can now argue that this method has been reliable. In *Softerware* and *Skeleton*, pursing practical problems in applied environments *did* serve as a fruitful source of generalizable knowledge and advancements in the state of the art. But these projects, in addition to *Chartability*, were all motivated by immediate, non-novel problems: people knew their charts were inaccessible and wanted some design guidance or engineering solutions.

So perhaps "low research" works.

Yet some research communities, especially *ASSETS*, have yet to reckon with "applied" work like this. If you trace the lineage of *action research* from Hayes [65] (which is how we frame our applied work in *Skeleton*), it leads into fantastic areas of participatory and community-based work, often with or by marginalized people and people with disabilities [6, 77, 102, 106, 152, 155, 156, 162, 175]. I would even argue that this lineage of work, which prioritizes and centers the lived experience of a person, is also why we have seen so many successful auto-biographical and auto-innovation styles of research as well recently [9, 49, 51, 73, 138, 173, 174].

And yet, action research largely does not touch on how practitioners (particularly those without disabilities) do applied work in accessibility [87, 143]. I believe that significant gaps exist in the space between traditional "high" knowledge production and applied, "lowly" spaces. Researchers tend to pursue novelty and the lifecycle of research production is highly risk adverse. And yet, relatively little work has been done within the accessibility research community to intellectually address the actual social and cultural risks involved producing decades of useless, lost, and forgotten prototypes and design guidelines [78, 82]. I would conjecture that research-producing academics are overly concerned with academic readership relative to the myriad of ways that their knowledge production actually is used by the rest of the world [115]. I would challenge *ASSETS* and the AT/accessibility communities of *CHI* in particular to take seriously the lack of applied (industry, government, and non-academic) work it produces as a community.

9.3 Who is responsible for repair?

Lastly, I want to revisit one of my opening points, where I argue that the *tool-makers* are first responsible for repair. This is true. However, the most pressing issue I have faced in recent years is mostly unmentioned across these research projects: tool-makers might be responsible, but this is because they are the only ones who have the *power* to make things accessible. Does this always need to be the case? Can we imagine an artifact's authority over the user's ability to access being designed towards self-subversion [55] or de-centralized agency [19, 90, 116], instead? What might that look like, concretely?

In *Softerware*, we begin to engage this larger problem in terms of an idealized state where a user can repair or re-design their own experiences. But to me, this self-repair is like laying down train tracks for yourself as you move a locomotive, but then lifting up your own tracks behind you as you go. You're the only one helping yourself. This is not ideal, for you or others.

What we need are broad, lasting, infrastructural changes. On the web, this problem becomes quite difficult to solve. A personal computer or device? Again, someone can auto-design their interfaces into a better state. But back when I started *Chartability*, the WebAim Million's report showed more than 95% of the top one million website home pages contain at least one critical accessibility error. And now, more than 6 years later, that proportion is unchanged [168].

Some had imagined that generative AI would solve the massive infrastructural repair problems we face. But unfortunately, the latest WebAim Million report shows that since 2020, ARIA usage has increased and correlates to more errors, while use of tabindex on a page has increased nearly 300% and also correlates to more errors on a page. If anything, during the age of generative AI, we have seen existing bad patterns worsen in prevalence and complexity.

I firmly believe that a tools-based approach is not enough on its own. Tool-making cannot be the *only* intervention on inaccessibility. Tools and tool-making, as our thesis argues, have a powerful role to play. But we simply can't tool our way out of failed infrastructure and inadequate policy when someone else *owns* the tools and tool-making. Visiting a website is like going into someone else's home: arranged according to their effort, tastes, and so on. If you can't access their home, you essentially need to request that they let you in personally. Website repair always falls to the owner and maintainer of a website, and they largely don't take any meaningful action.

Sidewalks outside of homes are a good parallel to this problem. Sidewalk accessibility is a massive infrastructural problem [130], and yet localities treat sidewalk maintenance in different ways: some, like where I presently live in the south hills of Pittsburgh, put the onus on the homeowner whose house and property the sidewalk touches. In other places, sidewalks are considered a public path, similar to a roadway, and are maintained through public tax and resource management. To no surprise, privately-maintained, public-access sidewalks are worse for people in pretty much every way than publicly-maintained ones [180]. This is because private homeowners don't care about sidewalk maintenance unless the city manages to fine them or they get sued.

And the web is a collection of private spaces that you visit privately. There is no truly shared, universally democratic, public space on the web. Centralization is partly to blame: sharing space while scaling leads to consolidation.

So my future work will continue to wrestle with the same tensions of scale, repair, and anti-consolidation of power, motivated by the same WebAim Million report. But now I look to questions of *democratic* and *radical* access to accessibility repair. The barriers I hope to tackle in the future are political and infrastructural. Perhaps tool-making will participate in this work, but it seems clear now from my work that the upstream technical problems and socio-political conditions that tools inherit, will likely not be addressed by tools alone.

What does "democratic" and "radical" infrastructure work look like? It will probably be an extension of *Softerware*, to some degree. I imagine future research that explores public-first spaces, ones where access is socially negotiated and repair belongs to all of us. Is this an autonomous space, like an autonomous zone [8] separate from the web? Above it, looking down into it, like shared annotation tools but capable of sharing the manipulation of websites [129]? A space with ambient co-repair, modeled after projects that bring people together [139] or that allow community "fixing" of misinformation [83]? Perhaps feminist thought on the ethics of care can help us [68, 104]? Or maybe it will be something else entirely; I'm not yet sure. But what made the web fantastic years ago is long gone; most of it has been hedged into corporate spaces that are controlled, maintained, and repaired by corporate power. And these entities are notoriously bad at repair. What I imagine in the future involves reclaiming a sense that the web is *ours*, belongs to *us*, and that ultimately *we* are responsible for making it accessible.

References

[1] Christopher Ahlberg and Ben Shneiderman. Visual information seeking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 313–317, New York, NY, April 1994. ACM. doi: 10.1145/191666.191775. URL https://doi.org/10.1145/191666.191775. 7.2, 7.3.1, **G4:**, **G6:**

[2] Obead Alhadreti and Pam Mayhew. Rethinking thinking aloud. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–12. ACM, April 2018. doi: 10.1145/3173574.3173618. URL https://doi.org/10.1145/3173574.3173618. 6.6.2, 7.6.6.2

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 111–117. IEEE, November 2005. doi: 10.1109/infvis.2005.1532136. URL https://doi.org/10.1109/infvis.2005.1532136. 3, 7.2

[4] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E. Mackay. Generative theories of interaction. *ACM Transactions on Computer-Human Interaction*, 28(6):1–54, December 2021. ISSN 1557-7325. doi: 10.1145/3468505. URL https://doi.org/10.1145/3468505. 2.1.2

[5] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29 (2):127–142, May 1987. ISSN 1537-2723. doi: 10.1080/00401706.1987.10488204. URL https://doi.org/10.1080/00401706.1987.10488204. 2.2.1, 7.2, 7.3.1

[6] Cynthia L. Bennett, Erin Brady, and Stacy M. Branham. Interdependence as a Frame for Assistive Technology Research and Design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, pages 161–173, New York, NY, USA, 10 2018. Association for Computing Machinery. doi: 10.1145/3234695.3236348. Accessed: 2021-09-03. 9.2

[7] Dan Bennett, Alan Dix, Parisa Eslambolchilar, Feng Feng, Tom Froese, Vassilis Kostakos, Sebastien Lerique, and Niels van Berkel. Emergent interaction: Complexity, dynamics, and enaction in hci. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–7. ACM, May 2021. doi: 10.1145/3411763.3441321. URL https://doi.org/10.1145/3411763.3441321. 2.1.2

[8] Hakim Bey. *TAZ: The temporary autonomous zone, ontological anarchy, poetic terrorism*. Autonomedia, 2003. 9.3

[9] Brandon Biggs, Christopher Toth, Tony Stockman, James M. Coughlan, and Bruce N.

Walker. Evaluation of a non-visual auditory choropleth and travel map viewer. In *Proceedings of the 27th International Conference on Auditory Display (ICAD 2022)*, pages 82–90. International Community for Auditory Display, June 2022. doi: 10.21785/icad2022.027. URL https://doi.org/10.21785/icad2022.027. PMCID: PMC10010675. 6.4.1, 9.2

[10] Matt Blanco, Jonathan Zong, and Arvind Satyanarayan. Olli: An Extensible Visualization Library for Screen Reader Accessibility. In *IEEE VIS Posters*, 2022. URL https://vis.csail.mit.edu/pubs/olli. Accessed: 2026-03-30, https://vis.csail.mit.edu/pubs/olli. 6.3.2

[11] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011. ISSN 1077-2626. doi: 10.1109/tvcg.2011.185. URL https://doi.org/10.1109/tvcg.2011.185. 1.2

[12] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. ISSN 1478-0895. doi: 10.1191/1478088706qp063oa. URL https://doi.org/10.1191/1478088706qp063oa. 6.6.3

[13] Albert S. Bregman. *Auditory Scene Analysis*. The MIT Press, Cambridge, MA, May 1990. ISBN 9780262269209. doi: 10.7551/mitpress/1486.001.0001. URL https://doi.org/10.7551/mitpress/1486.001.0001. 7.3.2

[14] S. Brewster. Visualization tools for blind people using multiple modalities. *Disability and Rehabilitation*, 24(11-12):613–621, January 2002. ISSN 1464-5165. doi: 10.1080/09638280110111388. URL https://doi.org/10.1080/09638280110111388. 2.2.3, 6.3.1

[15] Nicola J. Broadbent, Larry R. Squire, and Robert E. Clark. Spatial memory, recognition memory, and the hippocampus. *Proceedings of the National Academy of Sciences*, 101 (40):14515–14520, October 2004. ISSN 1091-6490. doi: 10.1073/pnas.0406344101. URL https://doi.org/10.1073/pnas.0406344101. 7.4.1

[16] A. Buja, J.A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proceeding Visualization '91*, pages 156–163,, Los Alamitos, CA, December 2002. IEEE Comput. Soc. Press. doi: 10.1109/visual.1991.175794. URL https://doi.org/10.1109/visual.1991.175794. 7.2, 7.3.1, **G4:**

[17] Jennifer L. Burke, Matthew S. Prewett, Ashley A. Gray, Liuquin Yang, Frederick R. B. Stilson, Michael D. Coovert, Linda R. Elliot, and Elizabeth Redden. Comparing the effects of visual-auditory and visual-tactile feedback on user performance. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI06, pages 108–117. ACM, November 2006. doi: 10.1145/1180995.1181017. URL https://doi.org/10.1145/1180995.1181017. 1.1

[18] Matthew Butler, Leona M Holloway, Samuel Reinders, Cagatay Goncu, and Kim Marriott. Technology developments in touch-based accessible graphics: A systematic review of research 2010-2020. In *Proceedings of the 2021 CHI Conference on Human Factors in Com-*

puting Systems, CHI '21, pages 1–15. ACM, May 2021. doi: 10.1145/3411764.3445207. URL https://doi.org/10.1145/3411764.3445207. 6.3.1, 7.2, 7.3.2

[19] Katherine Casey. Radical decentralization: Does community-driven development work? *Annual Review of Economics*, 10(1):139–163, August 2018. ISSN 1941-1391. doi: 10.1146/annurev-economics-080217-053339. URL http://dx.doi.org/10.1146/annurev-economics-080217-053339. 9.3

[20] Alex Chaparro and Maria Chaparro. Applications of Color in Design for Color-Deficient Users. *Ergonomics in Design*, 25(1):23–30, 1 2017. doi: 10.1177/1064804616635382. Accessed: 2021-09-06. 2.2.3

[21] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, January 2022. doi: 10.1109/tvcg.2021.3114829. URL https://doi.org/10.1109/tvcg.2021.3114829. 2.2.3, 6.3.1, 7.9.3

[22] Pramod Chundury, Urja Thakkar, Yasmin Reyazuddin, J. Bern Jordan, Niklas Elmqvist, and Jonathan Lazar. Understanding the visualization and analytics needs of blind and low-vision professionals. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–5. ACM, October 2024. doi: 10.1145/3663548.3688496. URL https://doi.org/10.1145/3663548.3688496. 2.1.1, 7.9.3

[23] Pramod Chundury, J. Bern Jordan, Yasmin Reyazuddin, Niklas Elmqvist, and Jonathan Lazar. Sound, touch, or the full monty? a comparative study of accessible data exploration systems for blind users. *ACM Transactions on Accessible Computing*, 19(1):1–41, March 2026. ISSN 1936-7236. doi: 10.1145/3798100. URL https://doi.org/10.1145/3798100. 7.3.2, 7.9.3

[24] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI02, pages 203–210. ACM, April 2002. doi: 10.1145/503376.503413. URL https://doi.org/10.1145/503376.503413. 7.4.1

[25] Maxime Cordeil, Benjamin Bach, Andrew Cunningham, Bastian Montoya, Ross T. Smith, Bruce H. Thomas, and Tim Dwyer. Embodied axes: Tangible, actuated interaction for 3d augmented reality data spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12. ACM, April 2020. doi: 10.1145/3313831.3376613. URL https://doi.org/10.1145/3313831.3376613. 7.3.4, 7.5.2

[26] Clare Cullen and Oussama Metatla. Co-designing inclusive multisensory story mapping with children with mixed visual abilities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, pages 361–373. ACM, June 2019. doi: 10.1145/3311927.3323146. URL https://doi.org/10.1145/3311927.3323146. 2.2.3, 6.4

[27] Lilian de Greef, Dominik Moritz, and Cynthia Bennett. Interdependent variables: Remotely designing tactile graphics for an accessible workflow. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–6. ACM, October 2021. doi: 10.1145/3441852.3476468. URL https://doi.org/10.1145/3441852.3476468. 6.4

[28] Sergio Della Sala, Colin Gray, Alan Baddeley, Nadia Allamano, and Lindsey Wilson. Pattern span: a tool for unwelding visuo–spatial memory. *Neuropsychologia*, 37(10): 1189–1199, September 1999. ISSN 0028-3932. doi: 10.1016/s0028-3932(98)00159-6. URL https://doi.org/10.1016/s0028-3932(98)00159-6. 7.4.1

[29] Deque. Automated Testing Identifies 57 Percent of Digital Accessibility Issues. Technical report, Deque, 3 2021. Accessed: 2021-09-06. 3

[30] Alan Dix. Designing for appropriation. In *Electronic Workshops in Computing*. BCS Learning & Development, September 2007. doi: 10.14236/ewic/hci2007.53. URL https://doi.org/10.14236/ewic/hci2007.53. 2.1.2

[31] Sebastian Draxler, Gunnar Stevens, Martin Stein, Alexander Boden, and David Randall. Supporting the social context of technology appropriation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2835–2844. ACM, May 2012. doi: 10.1145/2207676.2208687. URL https://doi.org/10.1145/2207676.2208687. 2.1.2

[32] Anthony Dunne and Fiona Raby. *Speculative everything*. The MIT Press. MIT Press, London, England, December 2013. 7.8.4, 7.9

[33] Eloi Durant, Mathieu Rouard, Eric W. Ganko, Cedric Muller, Alan M. Cleary, Andrew D. Farmer, Matthieu Conte, and Francois Sabot. Ten simple rules for developing visualization tools in genomics. *PLOS Computational Biology*, 18(11):e1010622, November 2022. doi: 10.1371/journal.pcbi.1010622. URL https://doi.org/10.1371/journal.pcbi.1010622. 2.2.2.2

[34] Frank Elavsky and Cindy Xiong Bearfield. Playing telephone with generative models: "verification disability," "compelled reliance," and accessibility in data visualization. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 14–24. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00008. URL https://doi.org/10.1109/accessviz68666.2025.00008. 6.3.1

[35] Frank Elavsky, Cynthia Bennett, and Dominik Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, June 2022. doi: 10.1111/cgf.14522. URL https://doi.org/10.1111/cgf.14522. 2.2.2.1, 2.2.2.2, 2.2.3, 6.4.3, 6.8.2, **G3:**

[36] Frank Elavsky, Lucas Nadolskis, and Dominik Moritz. Data navigator: An accessibility-centered data navigation toolkit. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327393. URL https://doi.org/10.1109/tvcg.2023.3327393. 2.1.2, 6.2, 6.3.1, 6.3.2, 6.4, 6.5.1, 7.2, 7.3.3, 7.9.2

[37] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual

analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, July 2013. ISSN 0272-1716. doi: 10.1109/mcg.2013.53. URL https://doi.org/10.1109/mcg.2013.53. 6.2

[38] Christin Engel and Gerhard Weber. Analysis of tactile chart design. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '17, pages 197–200. ACM, June 2017. doi: 10.1145/3056540.3064955. URL https://doi.org/10.1145/3056540.3064955. 7.3.2

[39] Will Epperson, Vaishnavi Gorantla, Dominik Moritz, and Adam Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 2160-9306. doi: 10.1109/tvcg.2023.3327367. URL https://doi.org/10.1109/tvcg.2023.3327367. 2.2.1

[40] Pavithra Eswaramoorthy, Frank Elavsky, Tania Allard, and gabalafou. Quansight-Labs/bokeh-a11y-audit: v1.0.0, February 2025. URL https://bokeh-a11y-audit.readthedocs.io/. 6.4.3

[41] Danyang Fan, Alexa Fay Siu, Wing-Sum Adrienne Law, Raymond Ruihong Zhen, Sile O'Modhrain, and Sean Follmer. Slide-tone and tilt-tone: 1-dof haptic techniques for conveying shape characteristics of graphs to blind users. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517790. URL https://doi.org/10.1145/3491102.3517790. 7.3.2

[42] Danyang Fan, Alexa Fay Siu, Hrishikesh Rao, Gene Sung-Ho Kim, Xavier Vazquez, Lucy Greco, Sile O'Modhrain, and Sean Follmer. The accessibility of data visualizations on the web for screen reader users: Practices and experiences during covid-19. *ACM Transactions on Accessible Computing*, 16(1):1–29, March 2023. doi: 10.1145/3557899. URL https://doi.org/10.1145/3557899. 1.1, 2.2.2.1, 2.2.2.2, 2.2.3, 6.3.2

[43] Ali Mazraeh Farahani, Peyman Adibi, Mohammad Saeed Ehsani, Hans-Peter Hutter, and Alireza Darvishy. Automatic chart understanding: A review. *IEEE Access*, 11:76202–76221, 2023. ISSN 2169-3536. doi: 10.1109/access.2023.3298050. URL https://doi.org/10.1109/access.2023.3298050. 6.3.1

[44] Stephen Few. Visual Business Intelligence - Data Visualization and the Blind. https://www.perceptualedge.com/blog/?p=1756, 2013. [Accessed 05-04-2025]. 1.1

[45] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust me, i'm partially right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1673–1682. ACM, May 2012. doi: 10.1145/2207676.2208294. URL https://doi.org/10.1145/2207676.2208294. 6.2, 7.2, 7.3.1, **G5:**

[46] John H. Flowers, Dion C. Buhman, and Kimberly D. Turnage. Cross-Modal Equivalence of Visual and Auditory Scatterplots for Exploring Bivariate Data Samples. *Human Factors*, 39(3):341–351, 9 1997. doi: 10.1518/001872097778827151. Accessed: 2021-09-03. 2.2.3, 3

[47] Steven L. Franconeri, Lace M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica

Hullman. The science of visual data communication: What works. *Psychological Science in the Public Interest*, 22(3):110–161, December 2021. ISSN 1539-6053. doi: 10.1177/15291006211051956. URL https://doi.org/10.1177/15291006211051956. 1.2

[48] Sara Garcia-Navarra, Lucia Solares, and Marta Mendez. Virtual reality and mixed reality in the assessment of spatial memory. *Exploration of Digital Health Technologies*, 3, June 2025. ISSN 2996-9409. doi: 10.37349/edht.2025.101151. URL https://doi.org/10.37349/edht.2025.101151. 7.4.1, **G1:**

[49] Paula Gardner and Jess Rauchberg. *Feminist, Postcolonial, and Crip Approaches to Human-machine Communication Methodology*. Sage Publishing, 2023. 9.2

[50] Elena Gherri and Martin Eimer. Active listening impairs visual perception and selectivity: An erp study of auditory dual-task costs on visual attention. *Journal of Cognitive Neuroscience*, 23(4):832–844, April 2011. ISSN 1530-8898. doi: 10.1162/jocn.2010.21468. URL https://doi.org/10.1162/jocn.2010.21468. 7.3.2, **G2:**

[51] Kate S Glazko, Momona Yamagami, Aashaka Desai, Kelly Avery Mack, Venkatesh Potluri, Xuhai Xu, and Jennifer Mankoff. An autoethnographic case study of generative artificial intelligence's utility for accessibility. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–8. ACM, October 2023. doi: 10.1145/3597638.3614548. URL http://dx.doi.org/10.1145/3597638.3614548. 9.2

[52] Michele E. Gloede and Melissa K. Gregg. The fidelity of visual and auditory memory. *Psychonomic Bulletin & Review*, 26(4):1325–1332, August 2019. ISSN 1531-5320. doi: 10.3758/s13423-019-01597-7. URL https://doi.org/10.3758/s13423-019-01597-7. 1.1

[53] A. Jonathan R. Godfrey, Paul Murrell, and Volker Sorge. An accessible interaction model for data visualisation in statistics. In *Lecture Notes in Computer Science*, pages 590–597. Springer International Publishing, 2018. ISBN 9783319942766. doi: 10.1007/978-3-319-94277-3_92. URL https://doi.org/10.1007/978-3-319-94277-3_92. 2.2.3

[54] Jenna L. Gorlewicz, Jennifer L. Tennison, P. Merlin Uesbeck, Margaret E. Richard, Hari P. Palani, Andreas Stefik, Derrick W. Smith, and Nicholas A. Giudice. Design guidelines and recommendations for multimodal, touchscreen-based graphics. *ACM Transactions on Accessible Computing*, 13(3):1–30, September 2020. ISSN 1936-7236. doi: 10.1145/3403933. URL https://doi.org/10.1145/3403933. 7.3.2, 7.4.1, 7.5.2

[55] David Graeber and Charlie Rose. Never cower to authority, 2006. URL https://www.youtube.com/watch?v=qt5op6wft-8. Accessed [2026]. 9.3

[56] David Gray Widder, Sarah West, and Meredith Whittaker. Open (for business): Big tech, concentrated power, and the political economy of open ai. *SSRN Electronic Journal*, 2023. ISSN 1556-5068. doi: 10.2139/ssrn.4543807. URL https://doi.org/10.2139/ssrn.4543807. 1.2

[57] Catherine Grevet and Eric Gilbert. Piggyback prototyping. In *Proceedings of the 33rd*

Annual ACM Conference on Human Factors in Computing Systems, CHI '15, pages 4047–4056. ACM, April 2015. doi: 10.1145/2702123.2702395. URL https://doi.org/10.1145/2702123.2702395. 2.1.2

[58] Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. The knowledge accelerator. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI'16, pages 2258–2270. ACM, May 2016. doi: 10.1145/2858036.2858364. URL https://doi.org/10.1145/2858036.2858364. 2.1.1

[59] Jen Hale. Extensive digitization of tactile map collection. https://www.perkins.org/extensive-digitization-of-tactile-map-collection/, 2016. Perkins Archives Blog, Perkins School for the Blind. July 2016. Accessed: 2022-02-23. 2.2.3

[60] Foad Hamidi, Patrick Mbullo, Deurence Onyango, Michaela Hynie, Susan McGrath, and Melanie Baljko. Participatory design of diy digital assistive technology in western kenya. In *Proceedings of the Second African Conference for Human Computer Interaction: Thriving Communities*, AfriCHI '18, pages 1–11. ACM, December 2018. doi: 10.1145/3283458.3283478. URL https://doi.org/10.1145/3283458.3283478. 2.1.1

[61] Noor Hammad, Frank Elavsky, Sanika Moharana, Jessie Chen, Seyoung Lee, Patrick Carrington, Dominik Moritz, Jessica Hammer, and Erik Harpstead. Exploring the affordances of game-aware streaming to support blind and low vision viewers: A design probe study. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–13. ACM, October 2024. doi: 10.1145/3663548.3675665. URL https://doi.org/10.1145/3663548.3675665. 6.2, 6.9

[62] A. Hamraie. Making access critical: Disability, race, and gender in environmental design. https://belonging.berkeley.edu/aimi-hamraie-making-access-critical-disability-race-and-gender-enviror 2019. [Online]. 2.2.3, 3

[63] Gunnar Harboe and Elaine M. Huang. Real-world affinity diagramming practices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 95–104. ACM, April 2015. doi: 10.1145/2702123.2702561. URL https://doi.org/10.1145/2702123.2702561. 6.6.3, 7.6.7

[64] Parker Hartman and Tim Gorichanaz. Evaluating ai-powered website accessibility overlays. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, page 1–4. ACM, October 2025. doi: 10.1145/3663547.3759761. URL http://dx.doi.org/10.1145/3663547.3759761. 9.1

[65] Gillian R. Hayes. The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction*, 18(3):1–20, July 2011. ISSN 1073-0516. doi: 10.1145/1993060.1993065. URL https://doi.org/10.1145/1993060.1993065. 6.2, 6.4, 9.2

[66] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages

171–178. IEEE, October 2007. doi: 10.1109/vast.2007.4389011. URL https://doi.org/10.1109/vast.2007.4389011. 2.2.1, 6.2, 7.3.1

[67] Jeffrey Heer and Dominik Moritz. Mosaic: An architecture for scalable & interoperable data views. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2023. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/tvcg.2023.3327189. URL https://doi.org/10.1109/tvcg.2023.3327189. 1.2, 2.2.1, 3, 7.3.1, **G5:**, 7.5.3, 7.9.1

[68] Ana O Henriques, Anna R. L. Carter, Beatriz Severes, Reem Talhouk, Angelika Strohmayer, Ana Cristina Pires, Colin M. Gray, Kyle Montague, and Hugo Nicolau. A feminist care ethics toolkit for community-based design: Bridging theory and practice. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, page 1–26. ACM, April 2025. doi: 10.1145/3706598.3713950. URL http://dx.doi.org/10.1145/3706598.3713950. 9.3

[69] Thomas Hermann, Andy Hunt, and John G Neuhoff, editors. *The Sonification Handbook*. Logos Verlag Berlin, Berlin, Germany, December 2011. ISBN: 978-3-8325-2819-5. 6.3.1, 7.3.2

[70] Jaylin Herskovitz, Yi Fei Cheng, Anhong Guo, Alanson P. Sample, and Michael Nebeling. Xspace: An augmented reality toolkit for enabling spatially-aware distributed collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS):277–302, November 2022. ISSN 2573-0142. doi: 10.1145/3567721. URL https://doi.org/10.1145/3567721. 2.1.2

[71] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Hacking, switching, combining: Understanding and supporting diy assistive technology design by blind people. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pages 1–17. ACM, April 2023. doi: 10.1145/3544548.3581249. URL https://doi.org/10.1145/3544548.3581249. 2.1.2

[72] Jaylin Herskovitz, Andi Xu, Rahaf Alharbi, and Anhong Guo. Programally: Creating custom visual access programs via multi-modal end-user programming. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, pages 1–15. ACM, October 2024. doi: 10.1145/3654777.3676391. URL https://doi.org/10.1145/3654777.3676391. 2.1.2

[73] Megan Hofmann, Devva Kasnitz, Jennifer Mankoff, and Cynthia L Bennett. Living disability theory: Reflections on access, research, and design. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '20, page 1–13. ACM, October 2020. doi: 10.1145/3373625.3416996. URL http://dx.doi.org/10.1145/3373625.3416996. 9.2

[74] Leona Holloway, Peter Cracknell, Kate Stephens, Melissa Fanshawe, Samuel Reinders, Kim Marriott, and Matthew Butler. Refreshable tactile displays for accessible data visualisation, 2024. URL https://arxiv.org/abs/2401.15836. 6.3.1, 7.2

[75] Leona M Holloway, Cagatay Goncu, Alon Ilsar, Matthew Butler, and Kim Marriott. Infosonics: Accessible infographics for people who are blind using sonification and voice. In

CHI Conference on Human Factors in Computing Systems, CHI '22, pages 1–13. ACM, April 2022. doi: 10.1145/3491102.3517465. URL https://doi.org/10.1145/3491102.3517465. 6.3.1, 7.2

[76] Jonathan Hook, Sanne Verbaan, Abigail Durrant, Patrick Olivier, and Peter Wright. A study of the challenges related to diy assistive technology in the context of children with disabilities. In *Proceedings of the 2014 conference on Designing interactive systems*, DIS '14, pages 597–606. ACM, June 2014. doi: 10.1145/2598510.2598530. URL https://doi.org/10.1145/2598510.2598530. 2.1.1

[77] Stacy Hsueh, Beatrice Vincenzi, Akshata Murdeshwar, and Marianela Ciolfi Felice. Cripping data visualizations: Crip technoscience as a critical lens for designing digital access. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, page 1–16. ACM, October 2023. doi: 10.1145/3597638.3608427. URL http://dx.doi.org/10.1145/3597638.3608427. 1.1, 2.2.3, 3, 9.2

[78] Amy Hurst and Shaun Kane. Making "making" accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 635–638, New York, NY, USA, 6 2013. Association for Computing Machinery. doi: 10.1145/2485760.2485883. Accessed: 2021-09-03. 2.1.1, 9.2

[79] Amy Hurst and Jasmine Tobias. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 11–18. ACM, October 2011. doi: 10.1145/2049536.2049541. URL https://doi.org/10.1145/2049536.2049541. 2.1.1, 7.5

[80] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Human–Computer Interaction*, 1(4):311–338, December 1985. ISSN 0737-0024. doi: 10.1207/s15327051hci0104_2. URL https://doi.org/10.1207/s15327051hci0104_2. 6.2

[81] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. Technology probes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24. ACM, April 2003. doi: 10.1145/642611.642616. URL https://doi.org/10.1145/642611.642616. 6.2, 6.9

[82] Liz Jackson, Alex Haagaard, and Rua Williams. Disability Dongle. https://blog.castac.org/2022/04/disability-dongle/, 2022. 9.2

[83] Farnaz Jahanbakhsh, Amy X. Zhang, Karrie Karahalios, and David R. Karger. Our browser extension lets readers change the headlines on news articles, and you won't believe what they did! *Proceedings of the ACM on Human-Computer Interaction*, 6 (CSCW2):1–33, November 2022. ISSN 2573-0142. doi: 10.1145/3555643. URL http://dx.doi.org/10.1145/3555643. 9.3

[84] Bin Jiang, Ferjan Ormeling, and Wolfgang Kainz. Visualization support for fuzzy spatial analysis. In *1995 ACSM/ASPRS Annual Convention & Exposition: Technical Papers*, vol-

ume 4: Auto Carto 12, pages 291–310, United States, 1994. American Congress on Surveying and Mapping. ISBN 1-57083-019-3. 12th International Symposium on Computer-Assisted Cartography, Auto-Carto 1995, Auto-Carto XII ; Conference date: 27-02-1995 Through 01-03-1995. 7.8.2

[85] Chutian Jiang, Wentao Lei, Emily Kuang, Teng Han, and Mingming Fan. Understanding strategies and challenges of conducting daily data analysis (dda) among blind and low-vision people. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, pages 1–15. ACM, October 2023. doi: 10.1145/3597638.3608423. URL https://doi.org/10.1145/3597638.3608423. 2.1.1

[86] Shuli Jones, Isabella Pedraza Pineros, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. "customization is key": Reconfigurable textual tokens for accessible data visualizations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–14. ACM, May 2024. doi: 10.1145/3613904.3641970. URL https://doi.org/10.1145/3613904.3641970. 3, 6.3.1, 6.5.2

[87] Shakila Cherise S Joyner, Amalia Riegelhuth, Kathleen Garrity, Yea-Seul Kim, and Nam Wook Kim. Visualization accessibility in the wild: Challenges faced by visualization designers. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517630. URL https://doi.org/10.1145/3491102.3517630. 2.2.3, 3, 6.2, 6.3.2, 6.8.2, 9.2

[88] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating visualizations without visuals: Investigation of visualization alternative text for people with visual impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1095–1105, January 2022. doi: 10.1109/tvcg.2021.3114846. URL https://doi.org/10.1109/tvcg.2021.3114846. 2.2.3, 6.3.1, 7.2, 7.3.2

[89] Hyeonsu B. Kang, Xin Qian, Tom Hope, Dafna Shahaf, Joel Chan, and Aniket Kittur. Augmenting scientific creativity with an analogical search engine. *ACM Transactions on Computer-Human Interaction*, 29(6):1–36, December 2022. ISSN 1557-7325. doi: 10.1145/3530013. URL https://doi.org/10.1145/3530013. 2.1.1

[90] Os Keyes, Josephine Hoy, and Margaret Drouhard. Human-computer insurrection: Notes on an anarchist hci. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13. ACM, May 2019. doi: 10.1145/3290605.3300569. URL http://dx.doi.org/10.1145/3290605.3300569. 9.3

[91] Gyeongri Kim, Jiho Kim, and Yea-Seul Kim. "explain what a treemap is": Exploratory investigation of strategies for explaining unfamiliar chart to blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pages 1–13. ACM, April 2023. doi: 10.1145/3544548.3581139. URL https://doi.org/10.1145/3544548.3581139. 6.3.1

[92] Hyeok Kim, Yea-Seul Kim, and Jessica Hullman. Erie: A declarative grammar for data sonification. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–19. ACM, May 2024. doi: 10.1145/3613904.3642442. URL https://doi.org/10.1145/3613904.3642442. 6.3.1, 7.2, 7.3.2

[93] Jiho Kim, Arjun Srinivasan, Nam Wook Kim, and Yea-Seul Kim. Exploring chart question answering for blind and low vision users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15. ACM, April 2023. doi: 10.1145/3544548.3581532. URL https://doi.org/10.1145/3544548.3581532. 2.2.3, 6.3.1, 7.3.3

[94] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40(3):173–188, June 2021. doi: 10.1111/cgf.14298. URL https://doi.org/10.1111/cgf.14298. 2.2.3, 6.3.1

[95] N. W. Kim, G. Ataguba, S. C. Joyner, Chuangdian Zhao, and Hyejin Im. Beyond alternative text and tables: Comparative analysis of visualization tools and accessibility methods. *Computer Graphics Forum*, 42(3):323–335, June 2023. ISSN 1467-8659. doi: 10.1111/cgf.14833. URL https://doi.org/10.1111/cgf.14833. 6.3.1, 6.3.2

[96] Kitware, Inc. *The Visualization Toolkit user's guide*, January 2003. URL http://www.kitware.com/publications/item/view/1269. 2.1.2

[97] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, Kim Marriott, and John Stasko. Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications*, 40(2):82–90, 2020. doi: 10.1109/MCG.2020.2968244. 2.2.3

[98] Xiaoou Li, Laura A. Carlson, Weimin Mou, Mark R. Williams, and Jared E. Miller. Describing spatial locations from perception and memory: The influence of intrinsic axes on reference object selection. *Journal of Memory and Language*, 65(2):222–236, August 2011. ISSN 0749-596X. doi: 10.1016/j.jml.2011.04.001. URL https://doi.org/10.1016/j.jml.2011.04.001. 7.4.1

[99] Zhicheng Liu and Jeffrey Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, December 2014. ISSN 1077-2626. doi: 10.1109/tvcg.2014.2346452. URL https://doi.org/10.1109/tvcg.2014.2346452. 2.2.1, 3, 6.2, 7.2, 7.3.1, **G5:**, 7.10

[100] Tania Llana, Marta Mendez, M.-Carmen Juan, and Magdalena Mendez-Lopez. Navigational object-location memory assessment in real and virtual environments: A systematic review. *Behavioural Brain Research*, 480:115388, March 2025. ISSN 0166-4328. doi: 10.1016/j.bbr.2024.115388. URL https://doi.org/10.1016/j.bbr.2024.115388. 7.4.1, **G1:**

[101] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1073–1083, January 2022. doi: 10.1109/tvcg.2021.3114770. URL https://doi.org/10.1109/tvcg.2021.3114770. 2.2.2.2, 2.2.3, 6.3.1, 7.2, 7.3.2

[102] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *2019 IEEE Visualization Conference (VIS)*, pages 16–20. IEEE, October 2019. doi: 10.1109/visual.2019.8933762. URL https://doi.org/10.1109/visual.2019.8933762. 2.2.2.2, 6.3.2, 9.2

[103] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. What do we mean by "accessibility research"? In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–18, New York, NY, USA, May 2021. ACM. ISBN 978-1-4503-8096-6. doi: 10.1145/3411764.3445412. URL https://doi.org/10.1145/3411764.3445412. Accessed: 2022-02-22. 2.2.2.1, 2.2.3

[104] Els Maeckelberghe. Feminist ethic of care: A third alternative approach. *Health Care Analysis*, 12(4):317–327, December 2004. ISSN 1573-3394. doi: 10.1007/s10728-004-6639-6. URL http://dx.doi.org/10.1007/s10728-004-6639-6. 9.3

[105] R G Maling and D C Clarkson. Electronic controls for the tetraplegic (possum) (patient operated selector mechanisms—p.o. s.m.). *Spinal Cord*, 1(3):161–174, November 1963. ISSN 1476-5624. doi: 10.1038/sc.1963.15. URL http://dx.doi.org/10.1038/sc.1963.15. 3

[106] Jennifer Mankoff, Gillian R. Hayes, and Devva Kasnitz. Disability studies as a source of critical inquiry for the field of assistive technology. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 3–10, New York, NY, USA, 10 2010. Association for Computing Machinery. doi: 10.1145/1878803.1878807. Accessed: 2021-09-06. 9.2

[107] Douglass L. Mansur, Merra M. Blattner, and Kenneth I. Joy. Sound graphs: A numerical data analysis method for the blind. *Journal of Medical Systems*, 9(3):163–174, June 1985. ISSN 1573-689X. doi: 10.1007/bf00996201. URL https://doi.org/10.1007/bf00996201. 2.2.3, 6.3.1

[108] Deborah Marks. Models of disability. *Disability and Rehabilitation*, 19(3):85–91, January 1997. ISSN 1464-5165. doi: 10.3109/09638289709166831. URL https://doi.org/10.3109/09638289709166831. 1.1

[109] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti Hearst, Kathleen McCoy, and Danielle Albers Szafir. Inclusive data visualization for people with disabilities. *Interactions*, 28(3):47–51, May 2021. doi: 10.1145/3457875. URL https://doi.org/10.1145/3457875. 2.2.1, 2.2.2.1, 2.2.3, 6.3.1, 7.2

[110] Kim Marriott, Matthew Butler, Leona Holloway, William Jolley, Bongshin Lee, Bruce Maguire, and Danielle Albers Szafir. From vision to touch: Bridging visual and tactile principles for accessible data representation. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):659–669, January 2026. ISSN 2160-9306. doi: 10.1109/tvcg.2025.3634254. URL https://doi.org/10.1109/tvcg.2025.3634254. 6.3.1, 7.2

[111] Rubén Alcaraz Martínez, Mireia Ribera Turró, and Toni Granollers Saltiveri. Methodology for heuristic evaluation of the accessibility of statistical charts for people with low vision and color vision deficiency, February 2021. URL https://doi.org/10.21203/rs.3.rs-156959/v1. 2.2.3

[112] David K. McGookin and Stephen A. Brewster. Soundbar: Exploiting multiple views in

multimodal graph browsing. In *Proceedings of the 4th Nordic conference on Human-computer interaction: Changing roles*, NordiCHI '06, pages 145–154, New York, NY, USA, 10 2006. Association for Computing Machinery. doi: 10.1145/1182475.1182491. Accessed: 2021-09-03. 2.2.3

[113] Andrew M. McNutt. No grammar to rule them all: A survey of json-style dsls for visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. ISSN 2160-9306. doi: 10.1109/tvcg.2022.3209460. URL https://doi.org/10.1109/tvcg.2022.3209460. 6.2

[114] Catherine Mei∗, Josh Pollock∗, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. Benthic: Perceptually congruent structures for accessible charts and diagrams. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '25, pages 1–17. ACM, October 2025. doi: 10.1145/3663547.3746342. URL https://doi.org/10.1145/3663547.3746342. 6.3.1, 6.3.2, 7.2

[115] Ehsan Mohammadi, Mike Thelwall, Stefanie Haustein, and Vincent Larivière. Who reads research articles? an altmetrics analysis of ¡scp¿m¡/scp¿endeley user categories. *Journal of the Association for Information Science and Technology*, 66(9):1832–1846, April 2015. ISSN 2330-1643. doi: 10.1002/asi.23286. URL http://dx.doi.org/10.1002/asi.23286. 9.2

[116] Giles Mohan and Kristian Stokke. Participatory development and empowerment: The dangers of localism. *Third World Quarterly*, 21(2):247–268, April 2000. ISSN 1360-2241. doi: 10.1080/01436590050004346. URL http://dx.doi.org/10.1080/01436590050004346. 9.3

[117] Dominik Moritz, Bill Howe, and Jeffrey Heer. Falcon. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–11. ACM, May 2019. doi: 10.1145/3290605.3300924. URL https://doi.org/10.1145/3290605.3300924. 3, 7.3.1

[118] Peya Mowar, Aaron Steinfeld, and Jeffrey P. Bigham. iTagPDF: Towards finally automating PDF accessibility. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26. ACM, 2026. doi: 10.1145/3772318.3790289. URL https://peyajm29.github.io/itagpdf.html. Best Paper Award. 6.3.2

[119] Jamie R. Nuñez, Christopher R. Anderton, and Ryan S. Renslow. Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PLOS ONE*, 13(7):e0199239, August 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0199239. URL https://doi.org/10.1371/journal.pone.0199239. 2.2.3

[120] CA Okoro, ND Hollis, AC Cyrus, and S. Griffin-Blake. Prevalence of disabilities and health care access by disability status and type among adults — united states, 2016. *Centers for Disease Control and Prevention MMWR*, 67:882–887, 2018. doi: 10.15585/mmwr.mm6732a3. 1.1

[121] Manuel M. Oliveira. Towards More Accessible Visualizations for Color-Vision-Deficient Individuals. *Computing in Science Engineering*, 15(5):80–87, 9 2013. doi: 10.1109/

MCSE.2013.113. Conference Name: Computing in Science Engineering. 2.2.3

[122] Fabio Pittarello and Manuel Semenzato. Towards a data physicalization toolkit for non-sighted users. In *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, January 2024. doi: 10.1109/ccnc51664.2024.10454861. URL `https://doi.org/10.1109/ccnc51664.2024.10454861`. 2.1.2

[123] Christopher Power, André Freire, Helen Petrie, and David Swallow. Guidelines are only half of the story: Accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, CHI '12, pages 433–442, New York, NY, USA, 2012. Association for Computing Machinery. doi: 10.1145/2207676.2207736. 3

[124] Lauren Race, Chancey Fleet, Danielle Montour, Lindsay Yazzolino, Marco Salsiccia, Claire Ferrari, Sheri Wells-Jensen, and Amy Hurst. Designing while blind: Nonvisual tools and inclusive workflows for tactile graphic creation. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, pages 1–8. ACM, October 2023. doi: 10.1145/3597638.3614546. URL `https://doi.org/10.1145/3597638.3614546`. 6.4, 7.3.2

[125] J.E. Rawling, E.C. Carson, J.W. Attig, D.M. Mickelson, W.N. Mode, M.D. Johnson, and K.M. Syverson. Quaternary geology of wisconsin. Technical report, Wisconsin Geological and Natural History Survey, 2025. URL `https://doi.org/10.54915/xqpw9883`. 6.4.1

[126] Blake Ellis Reid. The curb-cut effect and the perils of accessibility without disability. *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4262991. URL `https://doi.org/10.2139/ssrn.4262991`. 1.1

[127] Samuel Reinders, Matthew Butler, Ingrid Zukerman, Bongshin Lee, Lizhen Qu, and Kim Marriott. When refreshable tactile displays meet conversational agents: Investigating accessible data presentation and analysis with touch and speech. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):864–874, January 2025. ISSN 2160-9306. doi: 10.1109/tvcg.2024.3456358. URL `https://doi.org/10.1109/tvcg.2024.3456358`. 6.3.1, 7.2, 7.3.3

[128] Rod D Roscoe, Erin K Chiou, and Abigail R Wooldridge, editors. *Advancing diversity, inclusion, and social justice through human systems engineering*. CRC Press, London, England, December 2019. 1.1

[129] Heather Ruland Staines. Digital open annotation with hypothesis: Supplying the missing capability of the web. *Journal of Scholarly Publishing*, 49(3):345–365, April 2018. ISSN 1710-1166. doi: 10.3138/jsp.49.3.04. URL `http://dx.doi.org/10.3138/jsp.49.3.04`. 9.3

[130] Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. Project sidewalk: A web-based crowdsourcing tool for collecting sidewalk accessibility data at scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14. ACM, May 2019. doi: 10.1145/3290605.3300292. URL

http://dx.doi.org/10.1145/3290605.3300292. 9.3

[131] Antti Salovaara. Inventing new uses for tools: A cognitive foundation for studies on appropriation. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 4(2):209–228, November 2008. ISSN 1795-6889. doi: 10.17011/ht/urn.200811065856. URL https://doi.org/10.17011/ht/urn.200811065856. 2.1.2

[132] Elizabeth B.-N. Sanders and Pieter Jan Stappers. Probes, toolkits and prototypes: three approaches to making in codesigning. *CoDesign*, 10(1):5–14, January 2014. doi: 10.1080/15710882.2014.888183. URL https://doi.org/10.1080/15710882.2014.888183. 2.1.2

[133] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, January 2017. doi: 10.1109/tvcg.2016.2599030. URL https://doi.org/10.1109/tvcg.2016.2599030. 1.2, 2.2.1, 6.2, 6.5.2

[134] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O. Wobbrock. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 5 2021. Association for Computing Machinery. Accessed: 2021-09-06. 2.2.3

[135] D. Schepers and F. Elavsky. "accessible data viz". *The Investigative Reporters and Editors Journal*, 4:8–9, December 2024. 2.2.3

[136] William Schiff and Emerson Foulke, editors. *Tactual Perception*. Cambridge University Press, Cambridge, England, March 1982. 2.2.3, 7.3.2

[137] Donald Schön and John Bennett. Reflective conversation with materials, April 1996. URL https://doi.org/10.1145/229868.230044. 6.8.1

[138] JooYoung Seo, Yilin Xia, Bongshin Lee, Sean Mccurry, and Yu Jun Yam. Maidr: Making statistical visualizations accessible with multimodal data representation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–22. ACM, May 2024. doi: 10.1145/3613904.3642730. URL https://doi.org/10.1145/3613904.3642730. 6.3.1, 9.2

[139] Hanieh Shakeri, Ye Yuan, Benett Axtell, Denise Y. Geiskkovitch, and Carman Neustaedter. Designing smart home technology for passive co-presence over distance. In *Designing Interactive Systems Conference*, DIS '24, page 3389–3406. ACM, July 2024. doi: 10.1145/3643834.3661508. URL http://dx.doi.org/10.1145/3643834.3661508. 9.3

[140] Ather Sharif and Babak Forouraghi. evographs — a jquery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE, January 2018. doi: 10.1109/ccnc.2018.8319239. URL https://doi.org/10.1109/ccnc.2018.8319239. 6.3.2

[141] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users' experiences with online data visualizations. In *Pro-*

ceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '21, pages 1–16. ACM, October 2021. doi: 10.1145/3441852.3471202. URL https://doi.org/10.1145/3441852.3471202. 2.2.3, 6.3.1, 6.3.2, 7.3.3

[142] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. Voxlens: Making online data visualizations accessible with an interactive javascript plug-in. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19. ACM, April 2022. doi: 10.1145/3491102.3517431. URL https://doi.org/10.1145/3491102.3517431. 6.3.2

[143] Ather Sharif, Joo Gyeong Kim, Jessie Zijia Xu, and Jacob O. Wobbrock. Understanding and reducing the challenges faced by creators of accessible online data visualizations. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–20. ACM, October 2024. doi: 10.1145/3663548.3675625. URL https://doi.org/10.1145/3663548.3675625. 2.2.3, 3, 6.2, 6.3.2, 6.8.2, 9.2

[144] Ather Sharif, Venkatesh Potluri, Jazz Rui Xia Ang, Jacob O. Wobbrock, and Jennifer Mankoff. Touchpad mapper: Examining information consumption from 2d digital content using touchpads by screen-reader users. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, pages 1–4. ACM, October 2024. doi: 10.1145/3663548.3688505. URL https://doi.org/10.1145/3663548.3688505. 7.3.4, 7.5.2

[145] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. Tickers and talker. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4896–4907, New York, NY, USA, May 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858507. URL https://doi.org/10.1145/2858036.2858507. Accessed: 2021-09-03. 2.1.2

[146] Ben Shneiderman, Christopher Williamson, and Christopher Ahlberg. Dynamic queries. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, CHI '92, pages 669–670. ACM Press, 1992. doi: 10.1145/142750.143082. URL https://doi.org/10.1145/142750.143082. 6.2, 7.3.1, **G6:**

[147] Jim Smiley, Benjamin Lee, Siddhant Tandon, Maxime Cordeil, Lonni Besançon, Jarrod Knibbe, Bernhard Jenny, and Tim Dwyer. The made-axis. *Proceedings of the ACM on Human-Computer Interaction*, 5(ISS):1–23, November 2021. ISSN 2573-0142. doi: 10.1145/3488546. URL https://doi.org/10.1145/3488546. 7.3.4, 7.5.2

[148] Jim Smiley, Swamy Ananthanarayan, Leona M Holloway, Matthew Butler, and Tim Dwyer. Magnepins: A modular, affordable, and diy refreshable braille and tactile display. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, pages 1–12. ACM, September 2025. doi: 10.1145/3746059.3747692. URL https://doi.org/10.1145/3746059.3747692. 7.2, 7.5

[149] Volker Sorge. Polyfilling accessible chemistry diagrams. In *Lecture Notes in Computer Science*, pages 43–50. Springer International Publishing, 2016. ISBN 9783319412634. doi: 10.1007/978-3-319-41264-1_6. URL https://doi.org/10.

1007/978-3-319-41264-1_6. 2.2.3, 6.3.1

[150] Laura South and Michelle Borkin. Generating Seizure-Inducing Sequences with Interactive Visualizations. In *Proc. Of the IEEE VIS 2020 posters*, 10 2020. doi: 10.31219/osf.io/85gwy. 2.2.3

[151] Laura South, David Saffo, and Michelle Borkin. Detecting and Defending Against Seizure-Inducing GIFs in Social Media. Technical report, OSF Preprints, 1 2021. Accessed: 2021-08-20. 2.2.3

[152] Katta Spiel, Kathrin Gerling, Cynthia L. Bennett, Emeline Brulé, Rua M. Williams, Jennifer Rode, and Jennifer Mankoff. Nothing about us without us. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–8. ACM, April 2020. doi: 10.1145/3334480.3375150. URL https://doi.org/10.1145/3334480.3375150. 1.1, 9.1, 9.2

[153] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52 (2):163–174, 2005. 2.1.1

[154] Sebastian Paul Suggate. Beyond self-report: Measuring visual, auditory, and tactile mental imagery using a mental comparison task. *Behavior Research Methods*, 56(8):8658–8676, September 2024. ISSN 1554-3528. doi: 10.3758/s13428-024-02496-z. URL https://doi.org/10.3758/s13428-024-02496-z. 1.1

[155] Cella M Sum, Rahaf Alharbi, Franchesca Spektor, Cynthia L Bennett, Christina N Harrington, Katta Spiel, and Rua Mae Williams. Dreaming disability justice in hci. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, CHI '22, page 1–5. ACM, April 2022. doi: 10.1145/3491101.3503731. URL http://dx.doi.org/10.1145/3491101.3503731. 9.2

[156] Cella M. Sum, Franchesca Spektor, Rahaf Alharbi, Leya Breanna Baltaxe-Admony, Erika Devine, Hazel Anneke Dixon, Jared Duval, Tessa Eagle, Frank Elavsky, Kim Fernandes, Leandro S. Guedes, Serena Hillman, Vaishnav Kameswaran, Lynn Kirabo, Tamanna Motahar, Kathryn E. Ringland, Anastasia Schaadhardt, Laura Scheepmaker, and Alicia Williamson. Challenging ableism: A critical turn toward disability justice in hci. *XRDS: Crossroads, The ACM Magazine for Students*, 30(4):50–55, June 2024. ISSN 1528-4980. doi: 10.1145/3665602. URL http://dx.doi.org/10.1145/3665602. 9.2

[157] Ruoting Sun, Rong Luo, Xiwen Yao, Xinran She, Kotaro Hara, and Yang Jiao. Tactile data comics: A step-by-step multimodal presentation method on a refreshable tactile display for blind and visually impaired individuals. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8, New York, NY, April 2025. ACM. doi: 10.1145/3706599.3720192. URL https://doi.org/10.1145/3706599.3720192. 7.2, 7.3.2

[158] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. How people with low vision access computing devices. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, pages 171–180, New York, NY, USA, October 2016. ACM. ISBN 9781450341240. doi: 10.1145/2982142.2982168. URL https://doi.org/10.1145/2982142.

2982168. 2.2.3

[159] Pierre Tchounikine. Designing for appropriation: A theoretical account. *Human–Computer Interaction*, 32(4):155–195, July 2017. ISSN 1532-7051. doi: 10.1080/07370024.2016.1203263. URL https://doi.org/10.1080/07370024.2016.1203263. 2.1.2

[160] Yuanyang (YY) Teng and Darren Gergle. Access is not enough: Toward developmental flourishing. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI EA '26, page 1–6. ACM, April 2026. doi: 10.1145/3772363.3798552. URL http://dx.doi.org/10.1145/3772363.3798552. 1.1

[161] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. Chart reader: Accessible visualization experiences designed with screen reader users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18. ACM, April 2023. doi: 10.1145/3544548.3581186. URL https://doi.org/10.1145/3544548.3581186. 6.3.1, 6.3.2, 6.4, 7.3.2

[162] Alexandra To, Angela D. R. Smith, Dilruba Showkat, Adinawa Adjagbodjou, and Christina Harrington. Flourishing in the everyday: Moving beyond damage-centered design in hci for bipoc communities. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pages 917–933. ACM, July 2023. doi: 10.1145/3563657.3596057. URL https://doi.org/10.1145/3563657.3596057. 2.1.1, 9.2

[163] W3C. Web content accessibility guidelines (wcag) 2.1, 2018. 2.2.2.2, 3

[164] WAI. Understanding Success Criterion 1.3.1 — Understanding WCAG 2.0. https://www.w3.org/TR/UNDERSTANDING-WCAG20/content-structure-separation-programmatic.html, 2008. [Accessed 01-04-2025]. **G3:**

[165] WAI. Understanding success criterion 4.1.2: name, role, value. WCAG standard, W3C, 2016. Accessed: 2022-03-04. 2.2.2.2

[166] WAI. Understanding success criterion 2.1.1: keyboard. WCAG standard, W3C, 2017. URL https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html. Accessed: 2026-03-20, https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html. 6.3.1

[167] Chris Weaver. Multidimensional visual analysis using cross-filtered views. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 163–170. IEEE, October 2008. doi: 10.1109/vast.2008.4677370. URL https://doi.org/10.1109/vast.2008.4677370. 2.2.1, 3, 7.2, 7.3.1

[168] WebAIM. Webaim: The WebAIM Million - An annual accessibility analysis of the top 1,000,000 home pages. https://webaim.org/projects/million/#wcag, 2021. Accessed: 2021-09-03. 9.3

[169] Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, January 2010. ISSN 1537-2715. doi: 10.1198/jcgs.2009.07098.

URL https://doi.org/10.1198/jcgs.2009.07098. 6.2

[170] David Gray Widder and Richmond Wong. Thinking upstream: Ethics and policy opportunities in ai supply chains, 2023. URL https://arxiv.org/abs/2303.07529. 1.2

[171] David Gray Widder, Meredith Whittaker, and Sarah Myers West. Why 'open' ai systems are actually closed, and why this matters. *Nature*, 635(8040):827–833, November 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08141-1. URL https://doi.org/10.1038/s41586-024-08141-1. 1.2

[172] Leland Wilkinson. The grammar of graphics. In *Handbook of Computational Statistics*, pages 375–414. Springer Berlin Heidelberg, December 2012. ISBN 9783642215506. doi: 10.1007/978-3-642-21551-3_13. URL https://doi.org/10.1007/978-3-642-21551-3_13. 6.2

[173] Rua M. Williams and LouAnne E. Boyd. Prefigurative politics and passionate witnessing. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, page 262–266. ACM, October 2019. doi: 10.1145/3308561.3355617. URL http://dx.doi.org/10.1145/3308561.3355617. 9.2

[174] Rua M. Williams, Kathryn Ringland, Amelia Gibson, Mahender Mandala, Arne Maibaum, and Tiago Guerreiro. Articulations toward a crip hci. *Interactions*, 28(3):28–37, April 2021. ISSN 1558-3449. doi: 10.1145/3458453. URL http://dx.doi.org/10.1145/3458453. 9.2

[175] Rua Mae Williams, Louanne Boyd, and Juan E. Gilbert. Counterventions: a reparative reflection on interventionist hci. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, page 1–11. ACM, April 2023. doi: 10.1145/3544548.3581480. URL http://dx.doi.org/10.1145/3544548.3581480. 9.2

[176] B. L. Wimer, L. South, K. Wu, D. A. Szafir, M. A. Borkin, and R. A. Metoyer. Beyond vision impairments: Redefining the scope of accessible data representations. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7619–7636, December 2024. 2.2.1, 2.2.3

[177] Brianna L. Wimer, Ritesh Kanchi, Kaija Frierson, Venkatesh Potluri, Ronald Metoyer, Jennifer Mankoff, Miya Natsuhara, and Matt X. Wang. Nonvisual support for understanding and reasoning about data structures, 2026. URL https://arxiv.org/abs/2601.19168. 6.3.2

[178] Langdon Winner. Do artifacts have politics? In *Computer Ethics*, pages 177–192. Routledge, May 2017. ISBN 9781315259697. doi: 10.4324/9781315259697-21. URL https://doi.org/10.4324/9781315259697-21. 1.2

[179] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3(3):1–27, April 2011. doi: 10.1145/1952383.1952384. 1.2, 2.1.1

[180] Mintesnot Woldeamanuel and Andrew Kent. Measuring walk access to transit in terms of

sidewalk availability, quality, and connectivity. *Journal of Urban Planning and Development*, 142(2), June 2016. ISSN 1943-5444. doi: 10.1061/(asce)up.1943-5444.0000296. URL `http://dx.doi.org/10.1061/(ASCE)UP.1943-5444.0000296`. 9.3

[181] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1): 649–658, January 2016. ISSN 1077-2626. doi: 10.1109/tvcg.2015.2467191. URL `https://doi.org/10.1109/tvcg.2015.2467191`. 2.2.1

[182] R. Wood et al. “creatures of habit”: influential factors to the adoption of computer personalization and accessibility settings. *Universal Access in the Information Society*, 23(2): 927–953, March 2023. 3

[183] Keke Wu, Emma Petersen, Tahmina Ahmad, David Burlinson, Shea Tanis, and Danielle Albers Szafir. Understanding Data Accessibility for People with Intellectual and Developmental Disabilities. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, pages 1–16, New York, NY, USA, 5 2021. Association for Computing Machinery. doi: 10.1145/3411764.3445743. Accessed: 2021-09-03. 2.2.3

[184] Rebecca Yeo and Karen Moore. Including disabled people in poverty reduction work: “nothing about us, without us”. *World Development*, 31(3):571–590, March 2003. ISSN 0305-750X. doi: 10.1016/s0305-750x(02)00218-8. URL `https://doi.org/10.1016/s0305-750x(02)00218-8`. 1.1

[185] Xiaoyi Zhang, Tracy Tran, Yuqian Sun, Ian Culhane, Shobhit Jain, James Fogarty, and Jennifer Mankoff. Interactiles. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’18, pages 131–142. ACM, October 2018. doi: 10.1145/3234695.3236349. URL `https://doi.org/10.1145/3234695.3236349`. 7.3.4, 7.5.2

[186] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. Data sonification for users with visual impairment. *ACM Transactions on Computer-Human Interaction*, 15 (1):1–28, May 2008. ISSN 1073-0516. doi: 10.1145/1352782.1352786. URL `https://doi.org/10.1145/1352782.1352786`. 2.2.3

[187] Jonathan Zong. Using real names of disabled participant-contributors to practice citational justice in accessibility. In *2025 IEEE Workshop on Accessible Data Visualization (AccessViz)*, pages 30–33. IEEE, November 2025. doi: 10.1109/accessviz68666.2025.00011. URL `https://doi.org/10.1109/accessviz68666.2025.00011`. 6.4

[188] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. Rich screen reader experiences for accessible data visualization. *Computer Graphics Forum*, 41(3):15–27, June 2022. doi: 10.1111/cgf.14519. URL `https://doi.org/10.1111/cgf.14519`. 2.2.2.2, 3, 6.3.1, 6.3.2, 6.4, 7.2, 7.3.3

[189] Jonathan Zong, Josh Pollock, Dylan Wootton, and Arvind Satyanarayan. Animated vega-lite: Unifying animation with a grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):149–159, January 2023. ISSN 2160-

9306. doi: 10.1109/tvcg.2022.3209369. URL https://doi.org/10.1109/tvcg.2022.3209369. 6.2

[190] Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. Umwelt: Accessible structured editing of multi-modal data representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–20. ACM, May 2024. doi: 10.1145/3613904.3641996. URL https://doi.org/10.1145/3613904.3641996. 6.3.2, 6.8.4, 7.2, 7.3.2