Tactile Vega-Lite: Rapidly Prototyping Tactile Charts with Smart Defaults

Mengzhu (Katie) Chen
EECS
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
mzc219@mit.edu

Isabella Pedraza Pineros
M.I.T.
Cambridge, Massachusetts, USA
ipedraza@mit.edu

Arvind Satyanarayan
CSAIL
MIT
Cambridge, Massachusetts, USA
arvindsatya@mit.edu

Jonathan Zong
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
jzong@mit.edu

Abstract

Tactile charts are essential for conveying data to blind and low vision (BLV) readers but are difficult for designers to construct. Non-expert designers face barriers to entry due to complex guidelines, while experts struggle with fragmented and time-consuming workflows that involve extensive customization. Inspired by formative interviews with expert tactile graphics designers, we created Tactile Vega-Lite (TVL): an extension of Vega-Lite that offers tactile-specific abstractions and synthesizes existing guidelines into a series of smart defaults. Predefined stylistic choices enable non-experts to produce guideline-compliant tactile charts quickly. Expert users can override defaults to tailor customizations for their intended audience. In a user study with 12 tactile graphics creators, we show that Tactile Vega-Lite enhances flexibility and consistency by automating tasks like adjusting spacing and translating braille while accelerating iterations through pre-defined textures and line styles. Through expert critique, we also learn more about tactile chart design best practices and design decisions.

CCS Concepts

• **Human-centered computing → Visualization systems and tools**; **Accessibility systems and tools**.

Keywords

Tactile Graphics, Accessible Data Representation

ACM Reference Format:
Mengzhu (Katie) Chen, Isabella Pedraza Pineros, Arvind Satyanarayan, and Jonathan Zong. 2025. Tactile Vega-Lite: Rapidly Prototyping Tactile Charts with Smart Defaults. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3706598.3714132

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3714132

1 Introduction

Tactile charts are an essential tool for blind and low vision (BLV) people to independently explore data and participate equally in discussions involving statistical analysis [22, 55]. As a subset of tactile graphics, tactile charts use braille, raised lines, textured areas, and distinct marks to represent statistical data through touch [21]. Research has found that tactile charts facilitate independent data exploration, communicate spatial relationships, provide non-sequential access to information, and support content comprehension and memorization [22]. They surpass printed or electronic tables in helping blind users understand the correlations between variables [55], helping students learn statistical concepts, and developing tactile graphic literacy [2, 21].

However, when we conducted formative interviews with tactile graphics designers (section 3), we found that the existing workflows for designing and prototyping tactile charts are, unfortunately, tedious and time-consuming, requiring a high level of skill and experience. We found that because best practices for tactile design differ significantly from practices in visual design, non-expert designers face significant barriers to entry. For example, in visual design, it is common to optimize for space efficiency or aesthetics by orienting text sideways or scaling it while still maintaining legibility. However, this practice does not translate well to tactile design. Braille, the standard form of text in tactile charts, cannot be scaled, and orienting it in different directions can make it impossible to read. Prioritizing space efficiency can lead to overcrowding, which hinders the clarity and usability of tactile charts, especially for non-expert designers. Existing guidelines for tactile graphic design have a steep learning curve and can be overwhelming for those new to the field. Even for expert tactile designers, difficulties arise due to a lack of dedicated software tools for creating tactile charts. Designers typically need to master multiple tools to create a single chart, including general-purpose vector drawing tools and braille translation software. Each has its own learning curve, and designers must repeat tedious operations, such as transferring designs between tools, to make manual adjustments, even for minor modifications. The labor-intensive, specialized work involved in this process hinders the adoption of tactile charts in practice.

To address these challenges, we present Tactile Vega-Lite (TVL): extensions to Vega-Lite [46] for rapidly prototyping tactile charts.

In contrast to existing workflows that rely on fragmented functionality across various general-purpose tools, TVL introduces a set of domain-specific abstractions for tactile chart design (section 4). These include support for tactile encoding channels, braille, navigational aids, and layout configuration. These extensions to Vega-Lite enable designers to benefit from the advantages of existing visualization grammars—such as the ability to systematically enumerate a design space—while still expressing tactile-specific affordances. They also enable authors of Vega-Lite visualizations to make small edits to existing visual specifications to create tactile charts without requiring deep tactile design expertise.

Just as Vega-Lite provides *smart defaults* that enable authors to easily specify common visual chart forms, TVL's defaults generate tactile charts that align with established guidelines. For example, TVL renders grid lines as the least tactually distinct lines on a graph, as recommended by the Braille Authority of North America (BANA) guidelines [10]. Also, TVL automatically places the x-axis title below the labels and aligns it to the leftmost x-axis label, where readers conventionally expect to find it. Expert designers can override these defaults to make context-specific adjustments for their output medium (e.g. braille embosser, swell paper) or intended audience. Similarly, predefined palettes of lines and area textures allow non-experts to choose from a set of reliable options while enabling experts to rapidly prototype with reusable texture configurations.

We implement a prototype TVL editor that takes a declarative TVL specification and outputs a tactile chart in SVG format. These SVGs can be printed using various production methods; popular options include an embosser or a swell form machine. Through an example gallery (section 5), we demonstrate that TVL expresses a variety of chart forms and layouts. We also conducted a user study (section 6) with 12 tactile graphics designers to understand how TVL compares to their existing design workflows. We found that participants valued predefined textures and line styles for consistency and efficiency, aligning with existing workflows. Customizability was crucial for adapting tactile charts, especially for younger readers who benefit from grid lines and clear spacing. Differences in expectations between professional designers and Teachers of Students with Visual Impairments (TVIs)[1] revealed a trade-off between advanced design capabilities and practical, easy-to-use solutions, with professionals seeking more complex tools and TVIs favoring simplicity and efficiency. Through an expert critique of the system's defaults, we found that designers appreciated the tactile-specific design assets, such as predefined textures and line styles, which encouraged reasoning and prototyping to determine the optimal design for different audiences. This feedback validates our dual approach of providing guideline-aligned defaults while allowing for customization to accommodate context-specific needs, balancing efficiency, functionality, and reader experience.

[1]The term "Teacher of Students with Visual Impairments" (TVI) is widely used by organizations such as the American Printing House for the Blind (APH) and the Braille Authority of North America (BANA) to refer to educators who work with blind and low vision students. We acknowledge that the term "visually impaired" can be considered harmful language [6], or otherwise not preferred [28]. In this paper, we avoid using this term to describe people. However, we use TVI for clarity when referencing the specific job title, to maintain consistency with its use by educational institutions like APH [23], by educators [12], and by the state in licensing teachers [5].

2 Related Works

2.1 Tactile Chart Design

The effectiveness and legibility of tactile charts are heavily influenced by their design, particularly how well the chart accommodates the tactile reader's ability to discern shapes, lines, and textures through touch. To inform design, researchers have studied the relative effectiveness of different types of tactile charts, such as diagrams [25], heatmaps [19], schematics [43], and network graphs [57]. Another body of work studies the effectiveness of specific tactile design elements, such as textures [42, 51, 53, 54], or grid lines [3, 9, 36]. These studies highlight the need for deliberate design choices that prioritize tactile discernibility.

The reader's experience with tactile reading also affects the effectiveness of a particular design. Experienced tactile readers, for example, can more easily interpret dense information and advanced tactile symbols, while novice readers may require simpler charts with fewer details. Tactile reading strategies can vary, with some individuals using the palm of their hand to explore larger areas and their fingertips for more detailed inspection [29, 31]. As a result, the design of the tactile chart must be tailored to its intended audience.

Finally, the methods and materials with which a tactile chart is produced can affect a reader's experience. The most commonly used production methods for creating tactile charts include microcapsule paper, embossers, or vacuum-forming techniques. Tactile charts can also be viewed as a form of data physicalization [33], where data is transformed into physical artifacts to leverage human tactile perception. For instance, the use of collages and 3D models has been less common due to their limited replicability but is popular among teachers of blind and low-vision students [41]. Refreshable braille displays [30] provide digital tactile output, enabling users to access dynamic content; however, they are limited in size and struggle to represent detailed graphics. The choice of production method matters because each method offers a different tactile experience and has implications for the chart's longevity, texture fidelity, and ease of reading.

In our work, we create an authoring system for tactile charts intended for use with embossers or swell form paper. We incorporate best practices for tactile chart design into our system's default specifications, making it easy for an author to rapidly create an effective chart. An author can make small edits to a declarative specification to make adjustments for their intended audience.

2.2 Authoring Tactile Charts

The most common way to author tactile charts is through manual authoring using vector drawing tools, such as Adobe Illustrator and Corel Draw, or physical materials like braille graph paper and raised-line drawing kits. These methods offer high control over the final tactile output but are labor-intensive and require specialized knowledge of tactile design principles. A designer must meticulously adjust line thickness, spacing, and texture to ensure a chart is legible and informative for tactile users. This often involves trial and error, with multiple revisions needed to achieve the desired outcome.

Automated and semi-automated systems for authoring tactile charts provide an alternative to manual creation, offering more efficient workflows. These systems are generally either image-driven systems or data-driven systems. Image-driven systems, such as those using image processing, automatically convert existing visual charts into tactile formats [15, 27, 30, 34, 35, 39, 40, 56]. While these methods are effective for general image conversion, they are less ideal for visualizations. The main issues include a loss of data granularity and a compromise in fidelity, which can obscure critical information. Data-driven systems, by contrast, directly convert data into tactile charts without relying on a visual reference. One such example is SVGPlott [18], a tool that transforms structured data (such as CSV files) into tactile visualizations. However, existing systems exhibit several shortcomings in expressiveness and customizability [18, 24, 52, 55]. These include restrictive output formats (PNG) that limit post-creation editing [52], a lack of support for custom visual encodings [18], and inadequate compliance with established accessibility guidelines [24, 55]. Additionally, these systems do not support advanced data transformations or iterative design processes, which are essential for refining and optimizing tactile charts. The limited customization options, especially regarding textural enhancements and detailed layout adjustments, further constrain their effectiveness and usability.

In our work, we create an automated, data-driven authoring system. Our work aims to strike a balance between rapid authoring through guideline-compliant defaults and customizability through tactile abstractions for encoding, layout, navigational aids, and braille.

3 Motivation

To better understand the practical challenges involved with making tactile charts, we conducted formative interviews with tactile graphics experts to motivate our system design. From these interviews, we synthesized a set of design challenges that inform the design of tactile chart authoring systems. In this section, we first describe our process for conducting formative interviews. Then, we summarize a tactile chart designer's workflow: first by walking through an example of a professional designer's workflow and then discussing how an educator's workflow might differ. Finally, we discuss how these workflows translate into design challenges that tactile chart authoring systems can address.

3.1 Methods

We conducted seven formative interviews with experts in the field, including three Teachers of Students with Visual Impairments (TVIs), one assistive technology specialist who creates tactile graphics, two developers of assistive technologies, and one professional tactile graphics designer. Experts were sourced through snowball sampling, leveraging connections through acquaintances, and reaching out to relevant institutes and organizations, such as the Andrew Heiskell Braille and Talking Book Library in New York City. The interviews, lasting between 30 minutes to 1 hour, were conducted both in person and online. We asked participants about the current state of tactile graphics, their perception of existing tools, best practices in tactile graphics design and ways to learn about it, how they create tactile graphics, and their production methods.

Additionally, we conducted an hour-long observational session with a professional tactile graphics designer (one of our seven interview participants) to better understand their design and authoring workflow. During the session, we observed the designer creating new charts and updating and adjusting existing charts in Adobe Illustrator. The designer explained their process and reasoning aloud as they worked, giving us valuable insights into their decision-making approach.

3.2 Expert Tactile Chart Designer Workflows

Our interviews revealed rich insights on how expert tactile designers conduct their work. Here, we present an example design workflow synthesized from our interviews and our observation session with the professional tactile graphics designer, using a walkthrough of how a designer would create an example multi-series line chart (Figure 1) depicting the average fertility rate of China and Australia from 1955 to 2005. We synthesized this example workflow by re-watching video recordings of the observational study and systematically analyzing screenshots and intermediate design files produced during the authoring process. This allowed us to identify key decision-making steps, iterative refinements, and recurring patterns, which informed the construction of a generalized workflow representative of tactile chart creation practices.

3.2.1 Step 1: Familiarizing with complex guidelines. Tactile chart designers rely on established guidelines to create accessible and effective graphics. The most authoritative resource in this field is the 2022 Guidelines and Standards for Tactile Graphics published by the Braille Authority of North America (BANA) [10]. Spanning 426 pages, this comprehensive document provides best practices for everything from basic tactile design to advanced techniques for representing complex information.

Designers usually keep the guidelines open during their work or rely on experienced colleagues for quick consultations. For instance, to design a line graph like the one in our example (Figure 1), a designer would look up the relevant section of the guidelines under *Unit 6: Diagrams for Technical Material*, specifically in *6.6 Graphs* [10]. The designer might begin by reviewing and taking notes on the specific instructions regarding grid spacing, line thickness, and label placement. For example, grid lines should be the least distinct lines in a graph (6.6.4.4); the x-axis (horizontal) and y-axis (vertical) lines must be tactually distinct and stronger than the grid lines, and plotted lines should be the strongest and most tactually distinct lines on the graph (6.6.2.2) [10]. The designer would take note of these details during the planning stage.

New designers often find guidelines challenging to navigate due to their highly specific and segmented structure. For instance, the guidelines are divided into distinct sections for each type of graph—such as line graphs (6.6.4), bar graphs (6.6.6). Information is often repeated across multiple sub-sections, but there are often subtle yet critical differences between graphs that are important to the accessibility of the result. This lack of generalizable principles means designers frequently jump between sections, ensuring they are referencing the correct one for the specific graph type they are working on.

Furthermore, the guidelines are filled with exceptions and precise details that can be difficult to generalize. For example, bars in tactile bar charts should be between 3/8 inch (1 centimeter) and 1 inch (2.5 centimeters) wide—except for histograms, which should not have

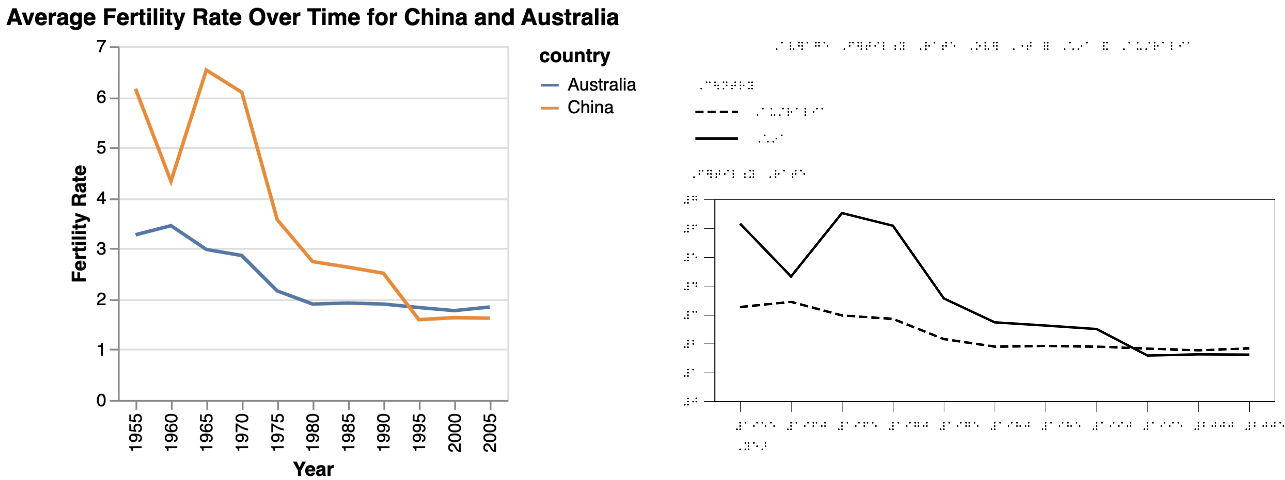


Figure 1: Comparison of visual and tactile charts representing fertility rate trends for China and Australia from 1955 to 2005. This comparison shows design considerations necessary when transforming visual data into tactile formats, such as converting text to braille, adjusting scaling and spacing of chart elements, re-arranging the legend, and substituting visual encodings with tactile encodings.

spacing between adjacent bars. This specificity makes it harder for designers to develop a broad, transferable understanding of tactile design principles and often requires constant cross-referencing, adding to the already complex process.

3.2.2 Step 2: Constructing the graphic. Professional designers often favor vector graphics tools like Adobe Illustrator, valuing their precision and robust capabilities despite a steep learning curve. Typically, a designer begins with a visual reference: either using an image, PDF, or PowerPoint slide, or generating a new chart from the data in a tool like Excel. They then import this visual into vector graphics software to trace key elements, including lines, axes, grid lines, and tick marks (Figure 2.1).

For our example line chart, the designer differentiates between the two data series by applying distinct line styles for China and Australia (Figure 2.2). Since creating textures and line styles from scratch can be time consuming, many professionals utilize Illustrator's symbols feature to generate reusable graphic assets. Maintaining a customized library of these symbols helps streamline their workflow, though building and managing such libraries can also be labor intensive.

The designer manually selects the grid lines to adjust their properties. They set them to be the least distinct lines relative to the x- and y-axis, which are more tactually distinct. Axes are emphasized to provide a clear reference frame for the reader. The designer then creates a legend that mirrors the exact line styles and textures used in the chart (Figure 2.3). According to guidelines, the legend should be placed before the graph and on the same page when possible.

The designer also adjusts the layout to optimize for the perceptual limitations of touch. Since tactile readers typically use their palms for an overview and fingertips for detail, designers usually scale lines while maintaining the relative spacing ratio between the lines and tactual distinctiveness. In this case, they manually select each grid line in Illustrator and increase the spacing between them. They also resize line styles in the legend to make them more discernible by fingertips. Further adjustments might involve individually selecting and adjusting the spacing between chart elements, such as the chart title and y-axis label, to maintain clarity (Figure 2.4).

Designers typically use Duxbury Braille Translation software to convert the chart title, x- and y-axis title and labels, and other text elements into braille. The designer first copies the text and selects the braille code appropriate for the chart audience. For instance, younger children often use Grade 1 uncontracted braille, while adults typically use Grade 2 contracted braille. For example, the word `Australia` would be `,AUSTRALIA` Grade 1 ASCII braille, but would be `,AU/RALIA` in Grade 2 ASCII braille (the visual difference between these strings rendered in Unicode braille is shown in Figure 3). In Grade 2 braille, the letters `st` have been contracted as a single cell. The designer then copies the translated braille back into their graphic design software. Braille introduces spatial constraints because it occupies more space than print text and cannot be resized. The designer then has to re-adjust the layout to fit the braille, including scaling the axes to avoid overlapping braille labels (Figure 2.5).

3.2.3 Step 3: Iterating for context and audience. At this design stage, the designer might make further adjustments based on the chart's intended purpose and the reader's familiarity with tactile graphics. For example, the designer might want to simplify the chart by eliminating borders or grid lines that do not contribute to the chart's core message. One educator we interviewed stated that "less is more," and their goal was "cutting down to the essentials." However, the degree of simplification depends on the chart's purpose and audience—designers may retain grid lines if readers are expected to interpret specific data points or add additional vertical grid lines for young readers who need help tracking values across the chart.

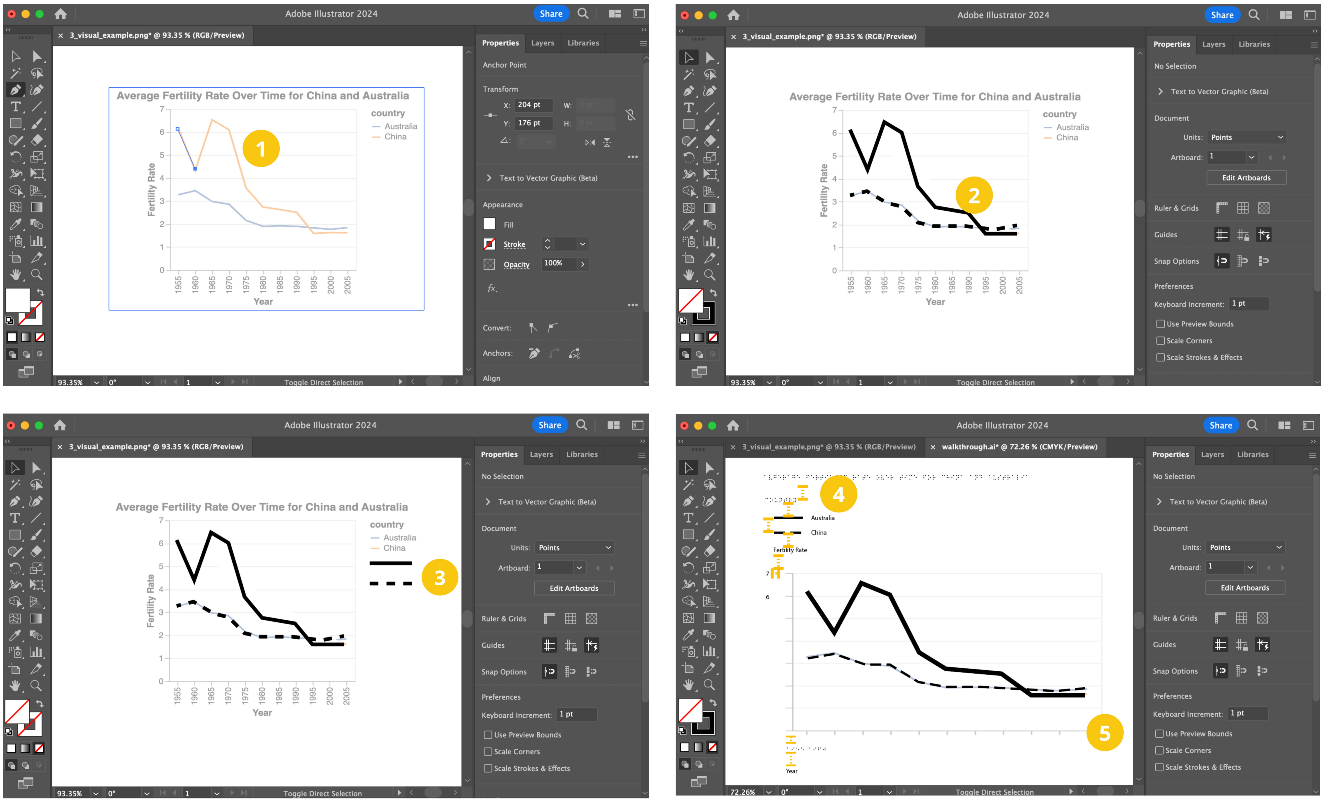


Figure 2: Example walkthrough of an expert designer's tactile chart creation process. 1) Tracing a visual reference. 2) Creating tactile encodings. 3) Creating a legend. 4) Adjusting spacing. 5) Scaling the axes to avoid overlaps.

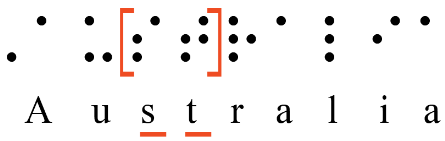
Similarly, a designer might adjust line thickness for different readers. Thinner lines are conducive to precise reading of data values; however, one educator mentioned that precision is less critical for younger students, so they prefer thicker lines because they are easier to locate and trace. Balancing varying reader needs requires designers to adapt their designs thoughtfully, ensuring each chart is tailored to the user's proficiency and tactile experience while maintaining accessibility and effectiveness.

Once the initial design is complete, the designer needs to iterate on it using specialized preview software to ensure it will effectively translate when physically produced. For example, designers use Tiger Designer Suite to preview how a design would render on an embosser. Designers and TVIs often refer to this step as seeing "how it is going to tiger out". To preview the design, the designer first saves their file from Illustrator and then re-opens it in Tiger Designer Suite to preview. While in preview mode, the designer checks the layout to identify potential issues, such as lines being too close or elements overlapping. Depending on the outcome of the preview, the designer then returns to the Illustrator file to make additional edits. They repeat the process above until they are happy with the preview results. Although essential for creating a high-quality tactile chart, previewing and adjusting can be time-consuming and frustrating, which requires a designer to constantly jump back and forth across different software.

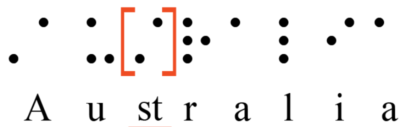
3.2.4 Step 4: Producing a physical artifact. Our interviews primarily discussed two common methods for converting digital chart designs into physical prints.

Embossing involves creating raised lines and textures directly on a sheet of paper, making the final product durable and suitable for long-term use. Embossers like the ViewPlus Delta or Columbia are commonly used for this process. However, embossed lines, especially curved ones, can sometimes appear jagged or segmented, as the embosser cannot always produce perfectly smooth curves. Different graphic embossers can produce varying outcomes due to differences in resolution, dot density, and embossing techniques. Variations in these factors can affect the raised elements' clarity, texture, and precision, leading to discrepancies in how the same image is rendered. Consequently, the tactile quality and readability of the embossed chart can differ depending on the specific embosser used. This is why previewing and adjustments specific to different machines are essential.

Swell forming uses a special type of paper coated with microcapsules. When black lines are printed on the paper and the sheet is run through a machine that applies heat, the lines swell up, creating raised textures that can be felt. This method allows for smoother curves and more varied textures than embossing, but the resulting tactile graphic may wear down faster with use. Different swell form machines and microcapsule papers can affect the final



(a) Grade 1 uncontracted braille representation of the word "Australia". Each letter is an individual cell.



(b) Grade 2 contracted braille representation of the Word "Australia". "st"has been contracted to one cell.

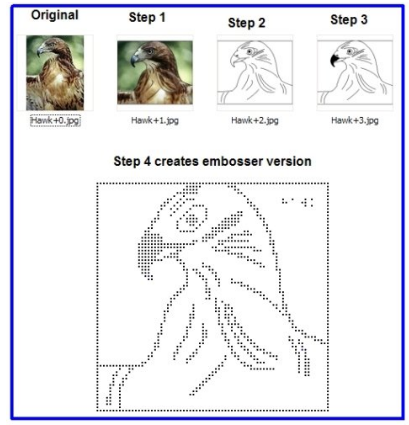
Figure 3: Braille representations of the word 'Australia' in Grade 1 (left) and Grade 2 (right).

tactile output because of variations in their sensitivity and embossing capabilities. Swell form machines may differ in temperature and pressure settings, affecting the depth and clarity of the raised features. Similarly, variations in microcapsule paper quality and formulation can influence how well the paper responds to the embossing process, resulting in differences in texture and detail in the final tactile representation.

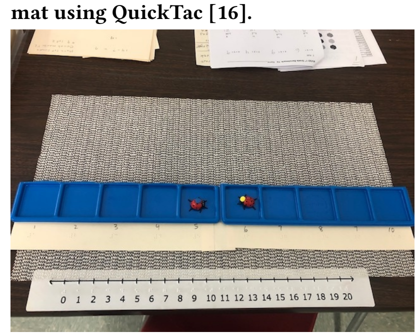
3.3 Educator Workflows

In subsection 3.2, we focused mainly on the workflows of our interview group's professional tactile graphic designers and assistive technology specialists. However, we found that our interviewers, who were educators, prioritized different design goals. Where professional designers prioritize precise control and customization, educators instead prioritize affordability, availability, and ease of use. These priorities result in a different approach to tactile chart creation. Though we found that professional designers and educators have largely overlapping thought processes, educators typically use more manual approaches to constructing physical charts (Step 2). This section briefly discusses how an educator's chart construction step differs from a professional designer's.

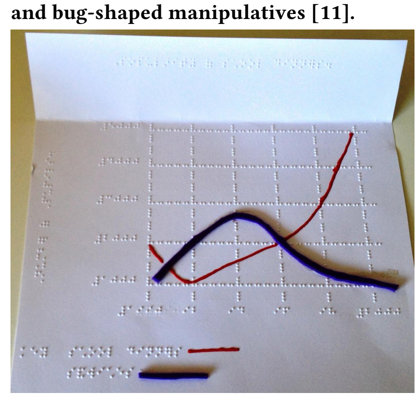
In contrast to professional designers, educators tend to prioritize ease of use and speed. One educator with 14 years of teaching experience said that "time and graphics determine what production is used." As a result, they often rely on more straightforward tools like QuickTac (Figure 4a) and physical objects such as pre-printed braille number lines and braille labels (Figure 4b). QuickTac [16] is free software widely used by educators to trace images and convert them into tactile graphics with basic formatting. It allows users to trace images and import them into the Duxbury Braille Translator (DBT) for braille translation. Although it is tailored for braille graphics, it offers limited customization compared to professional design tools.



(a) Step-by-step process of converting a photo of a hawk into a tactile embosser format using QuickTac [16].



(b) Hands-on math activity setup featuring blue compartment trays, a number line, and bug-shaped manipulatives [11].



(c) An example of how a TVI might make tactile charts for the classroom [38].

Figure 4: Educators often use a combination of specialized software, embossing techniques, and hands-on materials to create tactile graphics that convey visual information in accessible formats for blind and low-vision students.

Educators we interviewed also shared that they often begin with existing classroom materials, such as worksheets or textbook graphics, as their primary reference. An educator who works at the Marin County Office of Education shared that they "make tactile charts as often as it comes up in the curriculum." Using QuickTac, they trace or draw the graphic and then enhance it with tactile elements created from textured materials, such as braille labels or pre-printed number lines. These elements are manually glued together to form the final tactile chart (Figure 4c). While this approach is fast and practical for classroom use, it is less precise than workflows employed by professional designers, often resulting in lower accuracy and less consistency across charts.

Educators turn to this pragmatic approach because they lack easy-to-use tools that offer both customization and precision and lack time to develop skills in more complex software. While there might be some pedagogical advantages to using tangible materials like pipe cleaners for line charts, it is likely that many TVIs would like the option to use more automated tools if they were more available and user-friendly. The reliance on manual assembly and adaptation of existing materials often reflects a need to bridge the gap between the ideal of professionally designed tactile graphics and the practical constraints of time, resources, and available technology in educational settings.

3.4 Design Challenges

Based on our formative interviews with experts, we identified several challenges in designing tactile chart authoring systems. Here, we describe these design challenges that guided our decision-making as we iterated on the design of Tactile Vega-Lite (TVL). In section 4, we elaborate on how our design rationale for TVL responds to these challenges.

DC1: Guidelines are difficult to learn, apply, and generalize. Existing tactile graphics guidelines are highly prescriptive in that they often consist of lengthy and detailed lists of criteria that should be met in a tactile chart. While this level of detail makes it clear how designers should implement guidance and makes it easy to tell when charts are not complying with guidelines, there are also drawbacks to this approach. For instance, designers can sometimes find these guidelines challenging to learn independently, particularly when limited examples or experienced mentors are available. Even when designers are familiar with guidelines, remembering and applying them consistently can be difficult. Additionally, correctly implementing the guidelines in a design can be extremely tedious and error-prone, involving numerous manual adjustments to account for fine details.

Further, the guidelines are non-exhaustive; they only cover a small subset of data visualizations. For example, guidelines provide detailed instructions on creating bar charts, line charts, pie charts, and scatter plots but do not cover strip plots, bubble plots, grouped bar charts, and area charts, all of which are commonly used in practice by journalists and other visualization designers. Generalizing guidelines to other chart forms is difficult because they focus on prescriptive guidance rather than higher-level design principles. When the guidelines do not cover a certain chart type or situation, designers must make judgment calls that typically require years of experience or extensive practice.

DC2: Existing tools require high skill even to create basic charts. Designers face a trade-off between complexity and control. Tools like Adobe Illustrator that provide extensive customization options are often challenging to master, especially for those without specialized training. While there are ways to streamline workflows—such as using shortcuts or creating reusable symbols and textures—newcomers are often unaware of these tips and tricks. Mastering these techniques requires time and experience, which many novice designers or those new to tactile graphics lack. On the other hand, tools such as SVGPlott [18] that automate many aspects of the process offer limited customization, restricting designers' ability to create tailored tactile graphics that meet specific user needs. The lack of intermediate tools—those that balance ease of use with meaningful customization—forces designers to choose between two extremes: highly complex, professional-grade software or overly simplified, automated solutions. This trade-off can be particularly frustrating when trying to meet the needs of diverse audiences, from beginners learning to read tactile graphics to advanced users requiring highly detailed and accurate representations.

DC3: Tools lack affordances specific to tactile chart design. Most professional designers use general-purpose vector graphics software, like Adobe Illustrator, to create tactile charts. Because these tools are not designed specifically for tactile chart design, workflows can become unwieldy as designers translate tactile chart concepts into low-level graphical primitives.

For example, when a designer wants to make a change to a data-driven tactile mark, they will typically want to apply this change to every mark. However, they are forced to manually select and adjust each element, as most tools lack the ability to propagate changes across identical elements easily. This manual effort adds considerable time to the workflow.

Moreover, designers often transfer data between different platforms; for instance, they create a chart in Excel and then move it to a vector design tool like Illustrator to trace it. Designers must also preview the tactile chart in separate software to ensure it renders correctly for the intended production method (e.g., embossing or swell-forming). If the preview reveals issues, they must return to Illustrator, make edits, and repeat the process, leading to a frustrating cycle of back-and-forth adjustments. This cross-platform transfer introduces the risk of losing accuracy in the tactile representation and is only necessary because of the lack of domain-specific tool support.

These inefficiencies underscore the need for tools designed for tactile chart creation, with built-in affordances for managing tactile-specific assets, propagating changes across elements, and streamlining the iterative process.

4 System Design

Tactile Vega-Lite (TVL) is a set of extensions to the Vega-Lite grammar that enables rapid prototyping of tactile charts. We chose to extend Vega-Lite because we wanted to express a wide variety of tactile charts using a concise set of primitives, which is made possible by Vega-Lite's Grammar of Graphics (GoG)-based approach. We also hoped to enable visualization authors to easily specify default charts that are informed by best practices by making small edits to existing visual specifications without requiring tactile expertise. Vega-Lite's design and use of defaults lends itself to this approach.

To design TVL, we alternated between prototyping the desired syntax of TVL as fragments of JSON and prototyping a TVL compiler that could produce SVG tactile charts from our syntax. Our prototype TVL compiler is open source, and its code can be found at https://github.com/mitvis/tactile-vega-lite.

We also conducted informal testing of TVL charts by producing them on swell paper and testing with blind colleagues—one proficient in tactile graphics and one less familiar—to ensure that they could successfully interpret chart contents and answer data-related questions. They also provided feedback on design choices and defaults, which then in turn informed our language design.

In this section, we first introduce TVL's tactile abstractions and default specifications. Then, we discuss our design rationale and how it addresses the design challenges we identified.

4.1 Tactile Vega-Lite Abstractions

TVL extends Vega-Lite with new abstractions that address design considerations specific to tactile charts. These abstractions enable the specification of tactile encodings, braille usage, navigational aids, and layout configuration. For each specification, TVL provides default values (Appendix A) motivated by best practices in tactile design. In this section, we review each language extension, addressing how they build on Vega-Lite's existing grammar of graphics and explaining our choice of defaults.

4.1.1 Tactile Encodings. In visualization, encodings are mappings of data fields to perceptual properties such as color, size, and position. Tactile Vega-Lite shares many encodings with base Vega-Lite, such as the `x` and `y` positional encodings, and encodings like `size` and `shape`. However, it does not include encodings for visual-only properties like `color` or `opacity`. Instead, TVL introduces a `texture` encoding that maps a data field to a set of patterns that are perceptually distinct by touch. Drawing on empirical research and guidelines on pattern sets [10, 17, 42, 53, 54], we designed a default texture palette for TVL with 10 fill textures (Figure 6).

By default, the system automatically assigns discrete texture encodings to a subset of textures from the built-in palette. If only one data value is present in the encoding, we assign `solidGrayFill` as the default texture because empty bars can complicate the tactile differentiation between bars and white space around them [13, 20]. When more than five textures are used in the chart, the system alerts the author and recommends that they consider alternate encodings. This is because readers may have difficulty learning and recalling the mapping between textures and data values when more than five textures are used.

A designer can override the default texture by specifying an alternate set of named textures from our set of ten (Figure 7a). Though Vega-Lite can technically support arbitrary SVG textures, they are difficult to specify concisely — for instance, a variety of colors can be expressed concisely through RGB notation, but there is no analogous notation for texture. Thus, we leave specialized language support in TVL for arbitrary textures for future work.

Where textures define a tactile pattern for an area, TVL uses the existing `strokeDash` encoding to define tactile patterns for lines and `strokeWidth` encoding to control the thickness of the line. TVL provides a set of pre-defined `lineStyles`, which include `dashed`, `solid`, `dotted`, `longDashed` (Figure 6). These line styles, combined with variations in `strokeWidth`, enable designers to create a wider range of tactile line patterns. Like with texture, the system prompts the author to consider creating multiple charts when more than four different line styles are present.

By default, when there is only a single plotted line, the system always chooses the solid line style because it is the most easy-to-recognize tactile pattern, minimizing confusion for readers unfamiliar with other line styles. In a multi-series line chart, the system will always choose the solid line as one of the line styles and pair other styles with it because the contrast between a solid line and other patterns, such as dashed or dotted lines, is higher. Designers can customize arbitrary line styles by specifying the `strokeDash` encoding using SVG's standard dash array notation.

4.1.2 Braille Usage. Braille uses six raised dots arranged in a systematic pattern of two columns with three dots each, forming what is known as a Braille cell [32]. Unified English Braille (UEB) is the standardized system of English Braille [32], which includes uncontracted braille (Grade 1) and contracted Braille (Grade 2). Uncontracted braille is typically easier for beginners to learn but requires more space, limiting its use primarily to early elementary education. Contracted braille, the more advanced and widely used form, incorporates abbreviations, contractions, and shorthand symbols to represent common words, parts of words, or groups of letters. The numeric indicator in braille is a specialized symbol that signals the subsequent braille cells should be interpreted as numbers rather than letters. This indicator is used only once in sequences with multiple digits before the first digit. This distinction is crucial in contexts like tactile charts, where accurate representation of numbers is frequently required.

TVL implements braille translation using the open-source library LibLouis [37], which is well-regarded within the braille transcriber community. The default settings in TVL use versatile and high-quality `Swell Braille` font at `24pt` a font size for readable Swell Braille in tactile graphics, applying Grade 2 braille with the `en-ueb-g2.ctb` translation table from LibLouis. Given that braille fonts can vary depending on production methods, designers can choose from additional braille fonts, such as `California Braille`, which refers to a specific braille dot and cell spacing standard unique to California, designed for compliance with state regulations for public signage, or `Braille29`, which is used to create tactile graphics embossed using the Tiger embosser, available in many languages.

4.1.3 Navigational Aids. Vega-Lite provides navigational aids such as grid lines, axes, and axis ticks, which help users interpret data by offering reference points for scale and structure. These visual aids assist with understanding relationships between data points and orienting users within a chart. Tactile Vega-Lite builds upon these elements by introducing a clear information hierarchy among navigational aids, which assist users in orienting and navigating the chart's layout. This hierarchy organizes the relative prominence of grid lines, axes, tick marks, and data lines, ensuring that critical information is easy to detect while supporting elements remain unobtrusive.

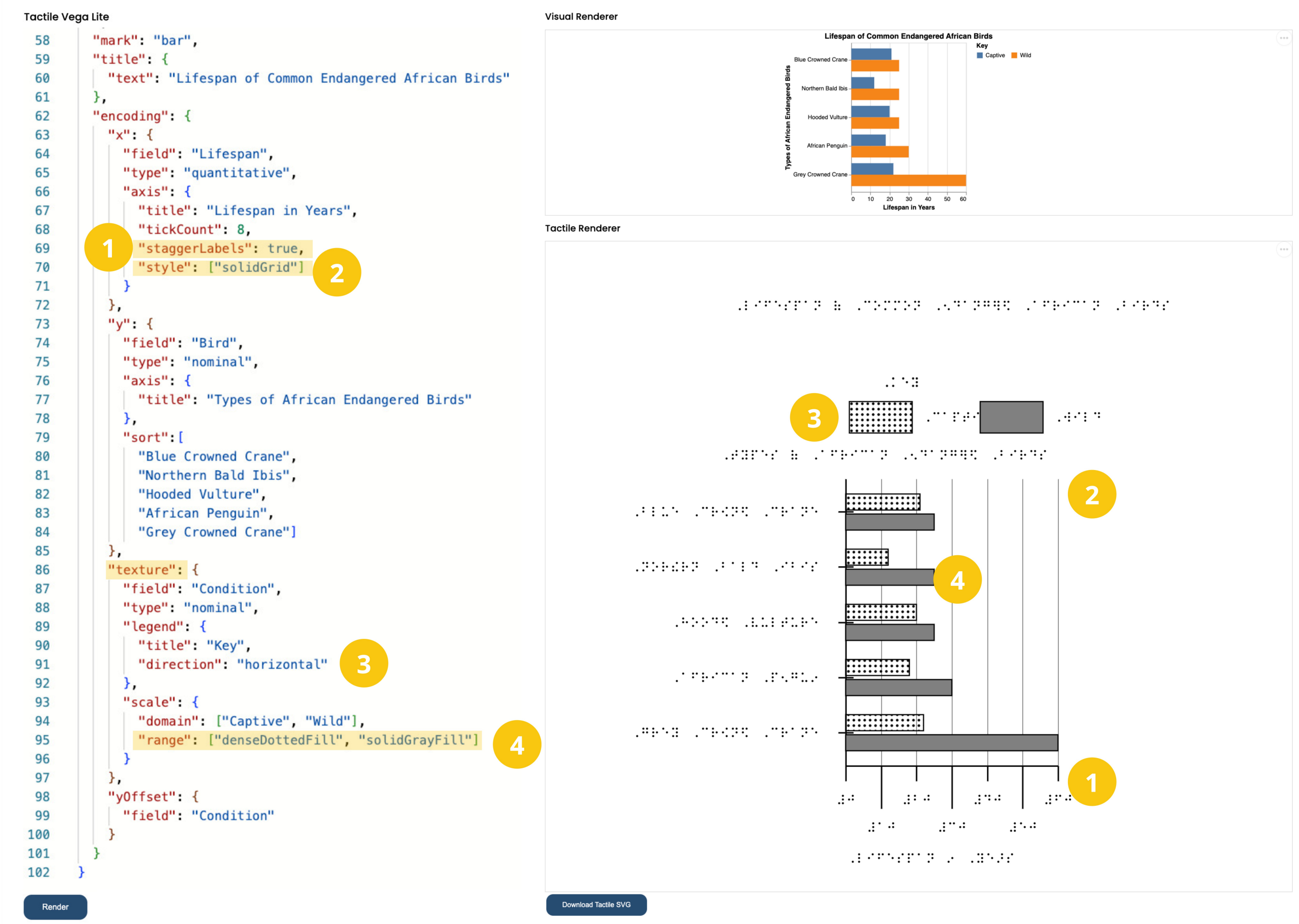


Figure 5: Tactile chart creators can use the TVL code editor to customize key properties such as axis labels, tick marks, sorting, and texture encodings. The rendered outputs on the right showcase the parallel visual and tactile representations. Highlighted code snippets in yellow show properties that were added in TVL.

For tactile charts, clear navigational aids are critical due to differences in visual and tactile wayfinding and exploration. According to Lucia Hasty, co-author of the BANA Guidelines and Standards for Tactile Graphics, unlike sighted individuals who typically grasp visual graphics in a "whole-to-part" manner, blind readers usually explore tactile graphics in a "part-to-whole" sequence [31]. They begin by examining individual elements of the graphic and gradually piece them together to understand the overall structure and context. Tactile readers rely heavily on anchors—specific points on a graphic that they can consistently refer back to while navigating the rest of the image. These elements facilitate the reading and comparison of values and serve as navigational breadcrumbs, allowing readers to trace their path back to previous points. It is crucial to balance these enhancements with minimizing clutter and maintaining clarity [21].

Information hierarchy for navigational aids is expressed through variations in line styles, thickness, and placement [10]. According to guidelines, grid lines should be the least distinct lines in a chart. The x-axis and y-axis should be tactually distinct and stronger than the grid lines. Tick marks may have the same line strength as the axes or have a line strength stronger than grid lines and weaker than axes.

However, research has not reached definitive conclusions on how to judge a line as more or less tactually distinct. For example, it is unclear whether line style or thickness plays a more significant role in tactile distinctiveness (e.g., whether a thinner solid line is more or less distinct than a thicker dotted line). TVL offers various customization options for designers to adjust on a case-by-case basis.

TVL uses default configurations to express the navigational aid hierarchy by varying only the line thickness. Conventionally, solid lines represent primary paths, while dashed or dotted lines indicate secondary or auxiliary lines. Line thickness can also convey hierarchy, with thicker lines representing more prominent features or paths and thinner lines for less significant ones. Additionally,

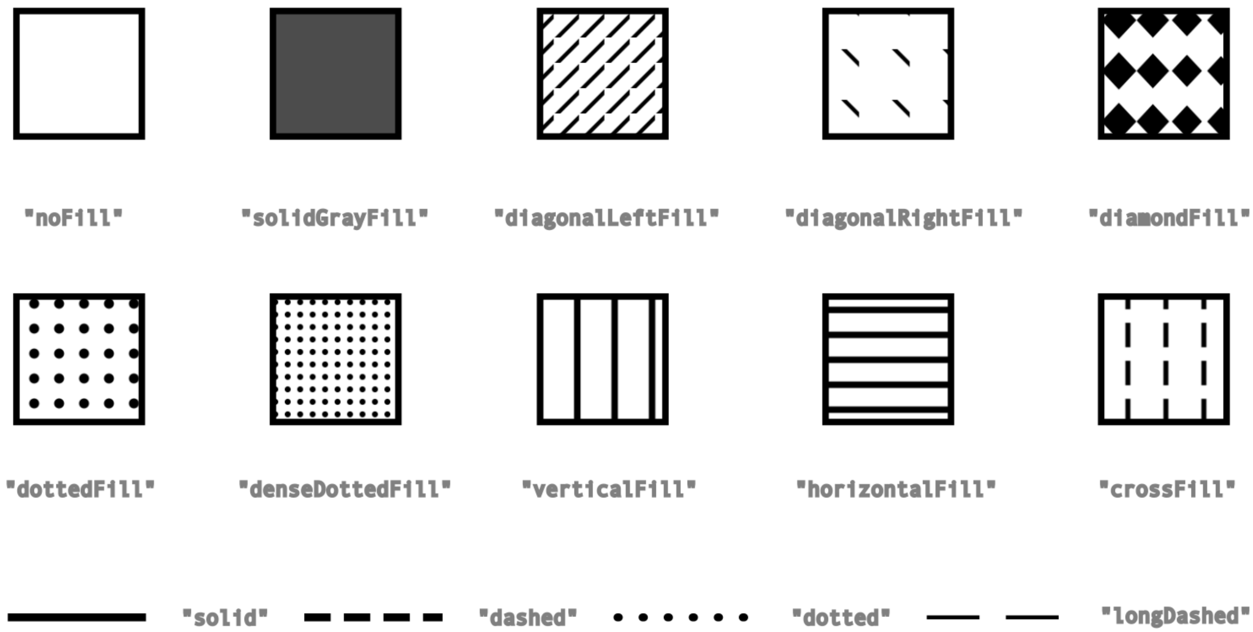


Figure 6: A collection of textures and line styles designed for accessibility in charts and diagrams. The textures include various fill patterns such as solid gray, diagonal lines, dots, and grid patterns, while the line styles include solid, dashed, dotted, and long-dashed lines.

the placement of these elements in relation to other tactile encodings, such as texture or braille labels, helps users orient themselves within the information structure. TVL provides customizable ways to express this hierarchy, allowing designers to adjust line thickness, style, and placement to meet specific tactile needs.

Grid lines. Grid lines provide a reference framework that helps readers track the positions of data points relative to the axes. Grid lines are useful when readers need to track values accurately, such as in bar or line charts. Research shows grid lines can improve readability, reading speed, and accuracy, though they can increase reading time [3, 8, 24, 26, 36]. TVL applies grid lines by default to quantitative axes based on the chart's encoding. A designer can customize grid line styles and change the background/foreground stacking order (Figure 7b).

Axes and axis ticks. The x- and y-axis provide a frame of reference for tactile wayfinding by defining the bounds of the chart. Guidelines suggest that they should be tactually distinct and stronger than the grid lines but less than plotted lines [10]. For tactile readers, axis ticks serve a dual function: on quantitative scales, they facilitate the reading of specific values and maintain orientation, while on categorical axes, they enable the identification of mark positions and quantity through tactile scanning [20]. They also aid readers in differentiating chart elements, discerning units, and accurately associating labels with the corresponding axes.

TVL's `staggerLabels` parameter staggers the x-axis labels, placing alternating values one or two lines below the x-axis — as suggested by the BANA guidelines [10]. When `staggerLabels` is set to `true`, a lead line extends the tick marks to the staggered labels on the lower level (Figure 7c). *Lead lines* are connecting lines that link data points to axes or other reference points and labels, making it easier for tactile readers to follow relationships between elements. By default, `staggerLabels` is set to `"auto"`; in this case, the system staggers axis labels when the length of the labels exceeds the threshold.

4.1.4 Layout Configuration. Layout refers to chart elements' relative positioning, spacing, alignment, and orientation. Tactile readers often employ a systematic scanning technique, using two hands to gather information from the top to the bottom of a page. This scanning method helps identify key features such as titles, labels, and other critical elements before diving into the finer details. A good layout uses alignment and negative space to enhance readability and comprehension while respecting readers' reading habits. Here, we discuss TVL's use of positioning, spacing, and alignment to manage layout considerations.

Positioning. By default, the chart title is centered at the top of the page. This placement allows readers to quickly identify the chart's subject and orientation, as the title is typically the first element sought by blind users. If there is a legend, then the legend is positioned directly below the title, with each legend entry positioned vertically on the same page to maintain continuity. This helps readers familiarize themselves with the different textures and line styles before exploring the data points. Legend entries, by default, are stacked vertically to provide a straightforward, predictable reading path (Figure 7d). Although placing the legend above the chart takes up space, it enables users to anticipate the chart content and recognize textures as they encounter them [20, 24].

Spacing. The general rule of thumb for spacing is that 1/8 inch between any two elements is required to perceive individual pieces of information [10]. TVL spaces axis labels 1/8 inch from the tick marks or axis lines on the x-axis and y-axis and adds additional padding between the axes and the marks in the chart. The system also adds spacing equivalent to 1 or 2 braille cell heights after the chart title and the x and y-axis titles. This spacing ensures that the tactile elements are not too densely packed, allowing users to differentiate between components. TVL sets the width of the chart based on these spacing specifications.

Alignment. By default, the system left-aligns the chart to the left of the plotting area to make it easier for readers to locate information quickly by systematically scanning from left to right. The x-axis title is left-aligned with the left-most x-axis label. We center the labels within the width of the bar or set of bars as recommended by the BANA guidelines [10]. Similarly, the y-axis title is left-aligned with the y-axis labels, while the labels are center-aligned with the corresponding tick marks.

4.2 Design Rationale

Tactile Vega-Lite leverages *smart defaults* to handle tedious formatting tasks, maintain consistency, and accelerate chart creation. By pre-configuring layout settings—including alignment, spacing, and positioning rules—TVL ensures that charts adhere to tactile design guidelines (DC1). These defaults eliminate the need for designers to manually adjust each element, speeding up the design process and reducing the cognitive load of applying complex guidelines. TVL smart defaults configure the navigational aids—such as grid lines, axis line width, and axis ticks—according to best practices for information hierarchy, ensuring consistency across multiple charts.

In addition to being guideline-compliant, TVL's default templates are reusable, allowing designers to easily apply consistent designs across different projects by loading the same TVL specification with new data. These reusable templates minimize manual adjustment while allowing designers to override default settings, ensuring that both efficiency and customization are balanced (DC1, DC2).

While smart defaults streamline the process, TVL recognizes that the needs of tactile readers are diverse, and designers often have the best knowledge of how to meet those needs. The system allows for complete customization of chart elements, enabling designers to adjust sizes, line styles, textures, and layout to suit specific contexts and users (DC2). This flexibility empowers designers to create tailored solutions while still benefiting from the efficiency of default settings. By combining smart defaults with customization, TVL strikes a balance between simplifying the design process and providing the control necessary to accommodate unique user requirements, addressing the trade-off between complexity and control (DC2).

TVL offers affordances for tactile design-specific needs with predefined styles for grid lines, textures, and line styles, allowing designers to quickly prototype charts without needing to create tactile assets manually. The system integrates braille translation directly, eliminating the need for designers to switch between multiple platforms like Duxbury and Illustrator. This built-in functionality starts with data and reduces the inefficiencies of moving across platforms, ensuring that tactile charts are produced efficiently and with fewer errors. By minimizing platform-switching and automating key steps, TVL enhances productivity and ensures designers can focus on higher-level decisions, solving key workflow issues specific to tactile chart design (DC3).

4.3 Limitations

Our implementation of TVL has several limitations. First, TVL currently only supports a subset of chart types supported by Vega-Lite. As we discuss in section 5, we have prioritized the most common tactile chart types. However, it is also unclear for many other chart types what best practices are for translating them. For example, area charts are not mentioned in the BANA guidelines, so we have not prioritized them in our prototype TVL compiler. Second, TVL assumes a screen-based rendering model, limiting its ability to precisely model the size of the physically produced end result. Tactile charts rely heavily on precise measurements for usability, but precise pixel-to-physical size calculations are dependent on hardware parameters. For similar reasons, TVL does not currently provide machine-specific previews (e.g. for specific embossers). Consequently, it might be difficult for users to anticipate how charts will look when produced on their devices. This lack of feedback can impact the quality of the final tactile outputs.

(a) Comparison of default and customized textures in tactile bar charts.

(b) Example of gridlines passing through the chart.

(c) Example of dynamically staggering labels based on their length.

(d) Comparison of default vertically stacked and customized horizontally aligned legend entries in tactile bar charts.

Figure 7: Examples of tactile charts generated using TVL, showcasing both default and customized configurations for textures, gridlines, staggered labels, and legends.

5 Evaluation: Example Gallery

The Tactile Vega-Lite (TVL) example gallery showcases a diverse range of charts and graphs, bar charts, line charts, pie charts, and stacked and grouped bar charts, highlighting the versatility and adaptability of the tool for presenting complex data in tactile-friendly formats. To understand what chart types were most important to express, we collected 49 tactile chart examples from various sources, including guidelines, research papers, institutional libraries, and industry standards [1, 4, 8, 10, 17, 44]. Our categorization revealed a dominant preference for bar charts (both grouped and stacked), line charts, and pie charts. As a result, we prioritized these simpler, more commonly used charts in the TVL example gallery. However, TVL cannot easily express charts not within base Vega-Lite's expressive gamut. For example, network visualizations are outside of TVL's scope.

The simple bar chart (Figure 8.1) illustrates how x-axis labels are staggered for better tactile readability, as well as the `dottedGrid` configuration. The grouped and stacked bar charts (Figure 8.2) and (Figure 8.6), along with the pie chart (Figure 8.7), showcase tactually distinct textures and with different legend positions, allowing readers to locate and familiarize themselves with the legend before exploring the chart itself. The two-series line chart (Figure 8.3) demonstrates the hierarchical relationship between encoded lines and navigational aids, with the encoded lines representing data being the most prominent. The multi-series line chart (Figure 8.4) highlights the variety of line styles available. Lastly, the scatterplot (Figure 8.5) demonstrates different tactile shape encodings available.

To show that TVL is expressive enough to support real-world use cases, we replicated an example chart created by an expert tactile designer (Figure 9). In this case, the smart defaults for a grouped bar chart in TVL were able to mostly approximate the original design. We made a series of customizations so that our chart more closely matches the real-world example. These included manually choosing the two textures from the TVL palette that most closely matched the original, manually adding line breaks to the chart title, and adjusting the width of the chart to create similar spacing. This exercise demonstrates that we could rapidly author a chart in TVL that closely approximates the layout and texture choices made by a professional designer.

6 Evaluation: User Study

6.1 Methods

Participants. We recruited 12 participants, including four tactile graphics designers, three teachers of students with visual impairment (TVIs), and five braille transcribers. All participants indicated they have considerable or expert-level experience with creating tactile graphics. 66.7% (n=8) participants self-rated as extremely confident in their ability to create tactile graphics. 41.7% (n=5) participants had more than 10 years of experience, 41.7% (n=5) participants had 5-10 years of experience. 41.7% (n=5) participants created tactile graphics every day, 25% (n=3) participants created tactile graphics once or twice a week. Adobe Illustrator and physical objects were the most frequently used tools for creating tactile graphics, followed by CorelDraw and Tiger Designer Suite. 91.7% (n=11) participants used embossers to produce tactile graphics, and 75% (n=9) participants used swell paper. Participants' other production methods included 3D printing, hand-drawn methods, UV printing, vacuum forming, and collages. Among the responses, participants identified the time-consuming nature of creating tactile graphics as the most significant challenge, followed by technical difficulties and lack of training and knowledge. In the examples shared by participants, most participants have made at least one of the following chart types: bar, pie, line chart.

Study setup. We interviewed each of our 12 participants for 60 minutes. We began the session with open-ended interview questions centered around the participant's current tactile graphics creation process. We then presented participants with two example designs in the TVL editor. Participants interacted with each design for about 20 minutes. The designs utilized TVL's smart defaults to implement guidelines, recommendations, and known best practices. Participants first critiqued and evaluated the default charts created by the system. Participants then made their desired adjustments to the default charts to meet their standards or requirements. When participants had little or no critique, we prompted them to explore the chart with a list of tasks designed to evaluate the predefined styles. As participants modified the chart, we asked them questions to understand their thought process. We then conducted a closing interview and asked participants to complete a Likert scale survey to evaluate the system and its defaults.

Example charts. Both examples used the `gapminder.json` dataset from the Vega dataset repository [45]. The dataset includes the fertility rate and life expectancy of countries around the world from 1955 to 2005, at a 5-year interval. The first example was a bar chart that displays the average life expectancy for the USA, China, and Australia (Figure 8.1). The second example was a multi-series line chart showing China and Australia's average fertility rate from 1955 to 2005 (Figure 8.3).

Analysis. We analyzed user testing results by reviewing video recordings, transcripts, and notes taken during the 12 interview sessions. We used an open coding approach to annotate transcripts, following a grounded theory method. We then did another pass to categorize these codes into broader themes, such as feedback on the tool's functionality, the default specification, customization capabilities, chart-specific concerns, and suggestions for improvement.

Motivation. Our study prioritized an exploratory focus to understand the diverse practices and challenges faced by tactile graphics creators. Given the variability in workflows, tools, and priorities among professionals like educators, designers, and braille transcribers, we sought to observe how experts interacted with TVL's features to critique and customize default charts, revealing their design rationale and iterative processes. This approach allowed us to gather rich, context-specific insights while working within the practical constraints of a 60-minute session.

6.2 Survey Results

We designed a Likert survey to evaluate the system's default choices and understand the usefulness of predefined styles. Results are shown in (Table 1).

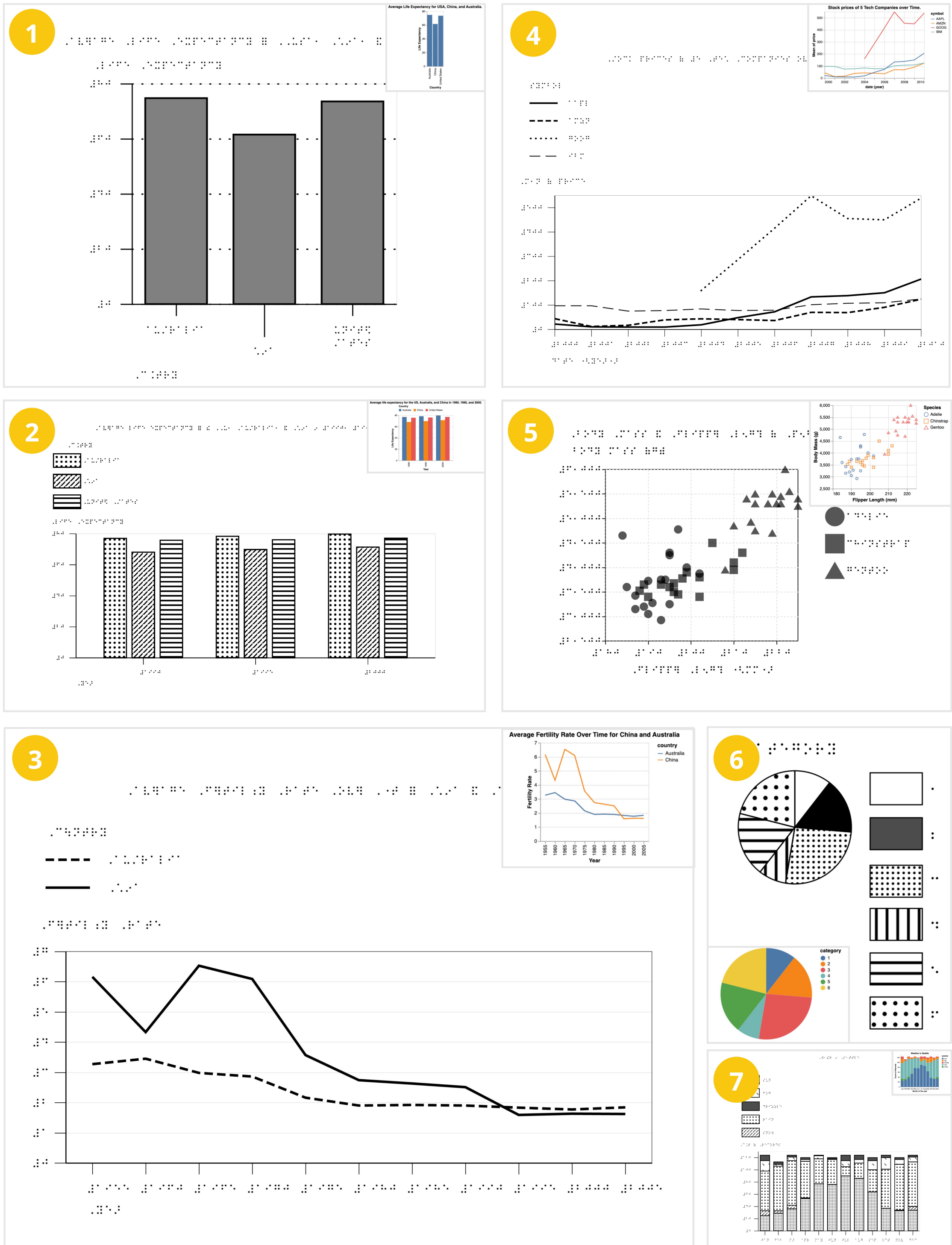


Figure 8: Gallery of tactile chart examples demonstrating various chart types, including bar charts, line charts, scatter plots, pie charts, and stacked bar charts. Each tactile chart is accompanied by its visual counterpart, showcasing how texture-based encodings, braille annotations, and tactile legends make data tactually accessible.

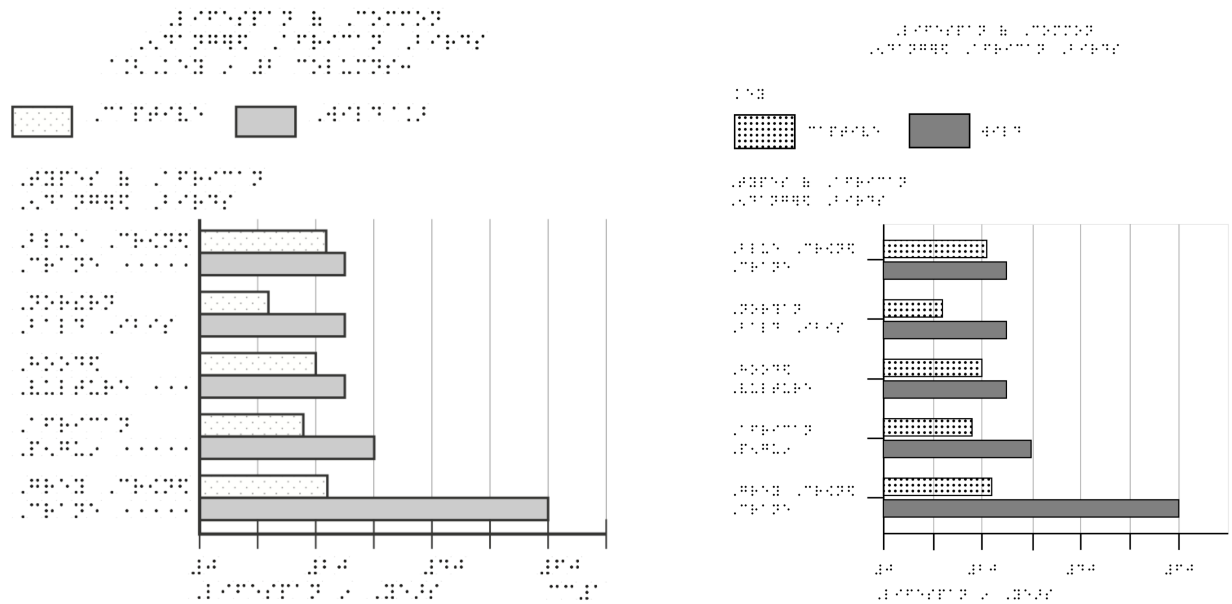


Figure 9: Comparison of original and replicated tactile bar charts representing the lifespan of common endangered African birds. Left graphic courtesy of the Media and Accessible Design Lab at LightHouse, San Francisco, and An Intervention to Provide Youth with Visual Impairments with Strategies to Access Graphical Information in Math Word Problems project funded by the Institute of Education Sciences (R324A160154).

Table 1: Rating scores for different aspects of the default tactile charts and customization experience on a five-point Likert scale where 1 = Not useful or satisfied at all and 5 = Extremely useful or satisfied. Median scores are shown in bold, averages in brackets [], and standard deviations in parentheses ().

Survey Questions	Score
Q1. How would you describe the predefined line styles in the prototype?	**4** [3.9] (0.67)
Q2. How satisfied are you with the predefined textures in the prototype?	**4** [4] (0.74)
Q3. How useful was the braille provided by the prototype?	**3.5** [3.4] (0.90)
Q4. How useful were the navigational aids (grid lines, grid line styles, etc) in the prototype?	**4** [3.9] (0.67)
Q5. How satisfied are you with the layout of the prototype?	**4** [4] (1.1)
Q6. How well do you think the output of the prototype adheres to guidelines?	**3** [3.1] (1.08)
Q7. How satisfied are you with the default chart produced by the prototype?	**3.5** [3.3] (1.14)
Q8. Are you able to create the tactile graphics that you wanted to create using our prototype?	**3** [3.5] (0.90)
Q9. How do the prototype's capabilities compare to the tools that you are using?	**3** [2.8] (1.07)
Q10. How would you rate your overall experience with the prototype?	**4** [3.7] (1.07)

Responses suggested that participants highly appreciated predefined line styles (Q1) and textures (Q2), as they offer consistency and reduce design time. Our interviews offered further context to these responses. For instance, P1 highlighted that pre-built textures are very similar to their existing workflow in that they have a texture sampler that they frequently refer to. Participants generally liked the default texture and line style choices and shared preferences for directional patterns (like `DiagonalLeft`, `DiagonalRight`) (P1, P7, P10). Participants also particularly liked the density of the dotted textures: not too dense such that it feels like a smooth surface, but also not too loose (P2, P3). P1 liked being able to "set differentiable tactile hierarchy for different types of lines."

Similarly, participants thought that using grid lines (Q4) was beneficial for helping students track information and orienting readers (P12), particularly for younger kids (P10). Even though participants

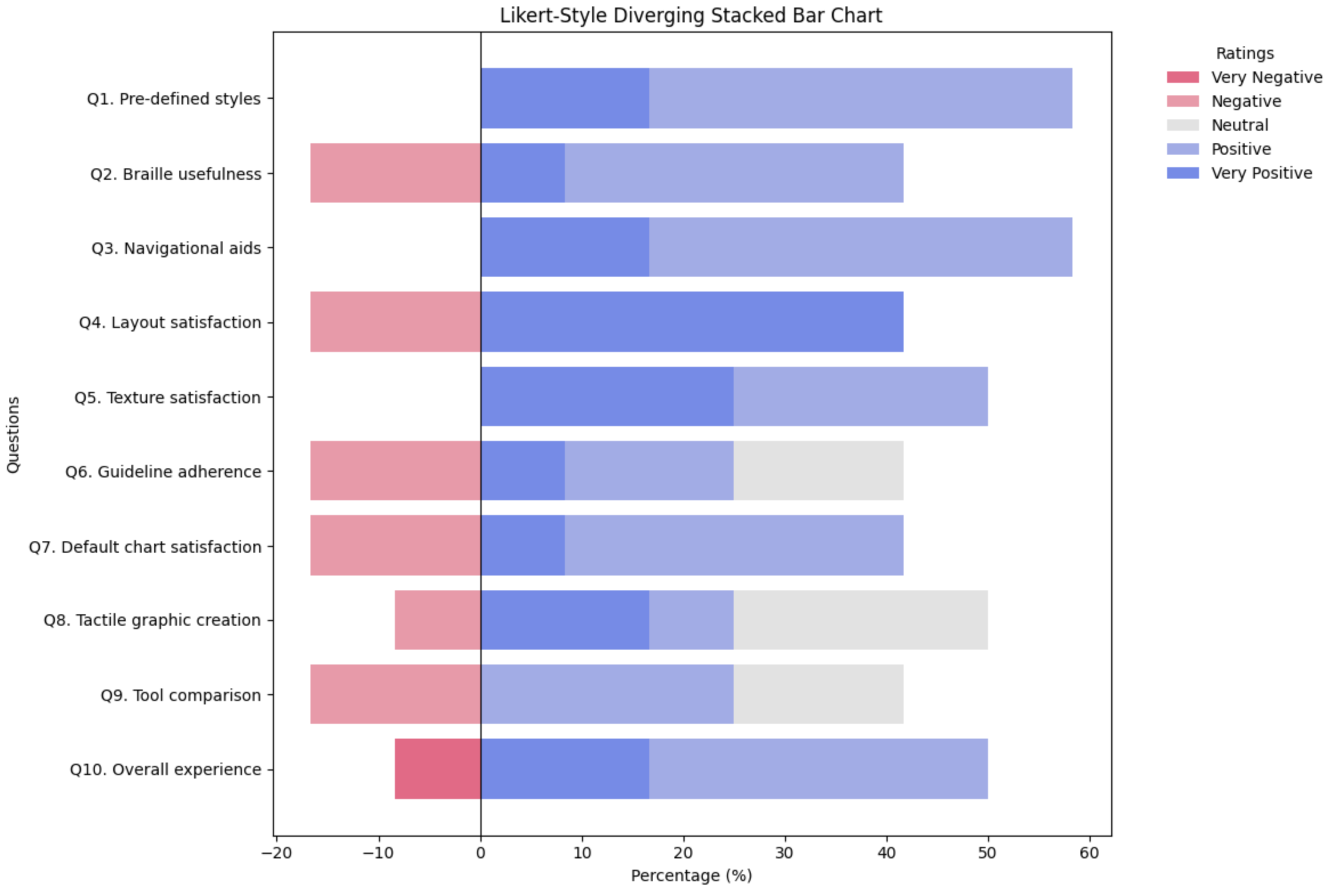


Figure 10: Study results show positive responses for styles(Q1), texture designs(Q5), navigational design(Q3), and mixed feedback for other categories like guideline adherence (Q6) and tool comparability(Q9).

noted that grid lines could make the design more cluttered and tactually complex (P9), they would still use them for younger kids who often have "a hard time bringing their hands horizontally across the page, needing some sort of horizontal grid lines" (P8). This further illustrates that the reader's needs and abilities play a key role in these design decisions and, thus, the importance of customizability. Regarding layout (Q5), there was general agreement that spacing and alignment are critical to ensuring tactile legibility, with participants preferring well-aligned axis labels and sufficient spacing around key elements like braille and marks (P9, P10).

The relatively modest scores for braille usefulness (Q3) reflect the limitations of our initial implementation. Specifically, the text was converted to lowercase, leading to discrepancies with the visual representation in capitalization. Certain braille rules were not fully implemented, such as when to include or omit numeric indicators, were not fully implemented. These challenges highlight the complexities of braille translation. While the current implementation provides a functional starting point, we aim to incorporate this feedback into future iterations of TVL.

Interestingly, the relatively middling score for adherence to guidelines (Q6, Q7) revealed differing expectations among participants. In our design process, we did our best to implement TVL according to the guidelines. However, the implementation we used in the user study was imperfect. TVIs, who may prioritize practical usability over strict adherence to guidelines, were generally less concerned with whether every guideline was followed precisely. In contrast, guideline writers, who possess a strong commitment to established standards and a high familiarity with those guidelines, were more likely to scrutinize the output charts rigorously for adherence and identify more issues. Specific issues noted by participants included tick marks not straddling the axis and the lack of white space around the intersection of the line charts.

Q8's score reflects the challenges participants faced with the prototype's specification-based interface. Several participants (P5, P6, P7, P8, P12) commented that introducing a user-friendly graphical interface would improve ease of use, particularly for users less familiar with programmatic design. We plan to incorporate this feedback in future versions of TVL.

Participants' varied perceptions of how TVL compares to existing tools (Q9) can be attributed to two primary factors. First, many participants, particularly educators, expressed reluctance to adopt new tools due to time constraints and a lack of resources. In certain areas, access to technology is limited, and introducing new tools requires state approval and additional training for educators, making the process difficult and slow. Second, professional tactile graphics designers who have developed a high level of proficiency with powerful design software like CorelDraw and Illustrator, often find these established tools more suitable for their advanced needs, which may have influenced their lower scores for the prototype.

6.3 User Study Results

6.3.1 Designers break rules to meet individual reader needs. Adhering to the guidelines was important for our participants, especially when designing for broader audiences or educational purposes. However, we found broad agreement that breaking from the guidelines is justified when it enhances functionality. Participants explained that they break format rules when working individually with students (P5) or when important information needs to be included, and the only way to do so effectively goes against the guidelines (P12). P5 emphasized that "guidelines are not rules" and highlighted the flexibility needed to accommodate unique scenarios, stating, "There's nothing specifically set in stone that you can't adjust." This flexibility is essential given the diverse experience levels of tactile graphics readers, with P2 noting that the effectiveness of a graphic can vary significantly depending on the reader's familiarity and skill.

6.3.2 Educators need charts efficiently, even at the expense of guidelines. When making charts to address immediate needs, efficiency often takes priority over perfection. P8, a teacher, noted, "I do my best to include the essential things… but would my spacing be perfect? No." This prioritization reflects the reality of balancing guideline adherence with the demands of lesson planning and grading. P10 confirmed that guidelines are often deprioritized in favor of focusing on student academics, stating, "We don't use guidelines as much as we should."

However, our educator participants also reflected on the broader implications of not consistently following guidelines, particularly in high-stakes contexts like standardized testing. P10 underscored the importance of guidelines for ensuring consistency in state testing, pointing out the potential challenges when students are suddenly exposed to strictly guideline-compliant materials after using less standardized resources. This highlights the importance of striking a balance between rapid authoring, flexibility, and maintaining standardization for critical applications like assessments.

For participants, these observations underscore the importance of tools like TVL to their work. What often gets overlooked or deprioritized during a time crunch are the more manual and tedious aspects of tactile graphic design, such as ensuring proper spacing and consistent formatting. Participants appreciated that TVL automated these tedious details. P2, who has tried creating scripts for Adobe Illustrator to "automate processes for standardization", commented that found "it's great to have [TVL] do all the formatting because that's a huge time saver to make sure things space out." P2 felt that they could "generate [charts] quickly" without having to worry about implementing every low-level guideline.

6.3.3 Expert design critique reinforced the importance of customizability. When we asked expert designers to give feedback on example default designs, there was room for reasonable disagreement between experts. Prompted by the predefined line styles in TVL, participants debated ways of expressing hierarchy through line weight and styling, as well as the inclusion of grid lines. To express the least tactually significant grid lines, P5 and P10 used a solid line with hairline thickness for grid lines, expressing concern that readers might confuse `dottedGrid` with "just random dots being embossed." However, P7 advocated for dotted grid lines, stating, "I think they are better as the lightest thing." P11 suggested that grid lines could have different textures to make them fainter, further highlighting the diverse approaches that designers take to meet different readers' needs.

The use of grid lines in a multi-series line chart was another point of contention. P7 would add grid lines for readers, just to make their lives a little easier, even though grid lines are not needed in this situation according to guidelines. P8 and P12 disagreed on whether to provide horizontal grid lines for value readout and vertical grid lines for tracing the year axis—P8 leaned towards only vertical grid lines, while P12 created grid lines for both axes.

These varying opinions emphasize that, as P5 pointed out, "If you give designers the same data, you will get 20 different things." In other words, there is no single best chart design. This comment highlights the inherent variability in how readers perceive tactile charts and underscores the need for tools to support customization and flexibility.

6.3.4 Instant feedback accelerates tactile design workflows. Participants appreciated the immediate feedback provided by the TVL editor and tactile renderer. P7 remarked, "[TVL] creates instantly, there's nothing else that does that," while P10 noted that they like it when they can "alter [TVL spec] and see it in real-time. There's no lag and having to wait and see how it's gonna look." Additionally, P10 emphasized the advantage of simultaneous editing and visualization, noting that they could make edits as needed while having the original graph readily available for reference. Participants identified the ability to preview how designs would be embossed on different embossers as essential, although this presents challenges due to the proprietary nature of many embossing machines. Incorporating a "send to printer" feature, as suggested by P1, could significantly enhance TVL's utility. This functionality would allow direct connections to output devices like embossers, facilitating seamless transitions from design to production and integrating the entire workflow into a streamlined system.

6.3.5 Balancing tactile-first design with information equity is difficult and important. Participants revealed a push-and-pull between prioritizing tactile-specific design and ensuring that tactile charts convey the same information as their visual counterparts, particularly in maintaining the integrity of data interpretation. P7 highlighted the importance of replicating visual impressions, such as the dramatic drop in data, to ensure blind students receive the same contextual information as sighted students. This concern reflects the need for tactile charts to be as visually similar as possible in certain contexts while still accommodating the tactile medium's unique requirements.

7 Discussion and Future Work

In this paper, we presented Tactile Vega-Lite (TVL), a system for rapidly prototyping tactile charts using extensions to the Vega-Lite grammar of graphics. We motivated the design of TVL through formative interviews with seven expert tactile designers, surfacing rich insights about the challenges of existing tactile design workflows. For instance, non-experts often struggle with the complexity of tactile guidelines, while experts are hindered by fragmented, time-consuming workflows that require extensive manual formatting. We address these challenges through TVL's tactile-specific abstractions, which provide smart defaults that allow non-experts to quickly generate guideline-aligned tactile charts and experts to prototype and customize based on reader's needs. A user study with 12 participants found that TVL enhances both flexibility and consistency by automating tedious tasks such as adjusting spacing and translating braille, and accelerates design iterations with pre-defined textures and line styles.

7.1 Tactile charts and multimodality

Recent work in accessible data representations has underscored the importance of using multiple data representations together in a complementary fashion [48–50, 58]. We found that many moments in our user studies support this idea. For example, a recurring request from designers (P1, P10) was the ability to overlay visual representations on tactile charts. Participants referenced the importance of hybrid visual and tactile representations to low-vision users (P9). In particular, one participant mentioned that she started introducing one of her low-vision students to tactile charts because they often get fatigued when only reading visually. Another participant (P1) noted that it is hard to read braille visually, so low-vision users might also benefit from the presence of both text and braille. These insights suggest that moving back and forth between multiple modalities can benefit users, so researchers should be attentive to ways of incorporating tactile charts into multimodal systems, as many have begun to do [7, 14, 47].

7.2 Perceptual research on tactile information hierarchy

In our design process and user studies, we noticed a relative lack of consensus in perceptual research on tactile graphics when it comes to information hierarchy. A key question is how different techniques for creating tactual distinctiveness—such as line styles, thickness, and placement—affects users' ability to understand information hierarchy. Comparing the relative perceptual effectiveness of these tactile features can provide insights into which designs are more easily distinguishable and effective in conveying complex data. Users' strategies for tactile reading and navigation could also changes which approaches to tactile information hierarchy are most effective. Understanding these distinctions can guide the development of best practices for creating tactile data representations that ensure BLV readers are able to quickly discover and navigate chart elements.

7.3 Designing tactile-first data representations

Tactile charts serve an important purpose in making existing visualizations accessible. This is important for helping blind and low-vision readers establish common ground with sighted readers of visualizations, especially in educational settings where students are learning how to use charts. However, as a result, the goal of tactile chart designers has largely been to faithfully reproduce visual chart forms in the tactile modality.

In our design process and exploratory tests with blind readers, we sometimes felt that existing visual forms were not well suited to tactile representation. For example, bar charts have a lot of empty interior space. While they are helpful for making length comparisons visually, they provide less tactile signal. Often there were other possible chart forms (like a dot plot with a size encoding) that seemed potentially more suited to the same data from a tactile perspective. For blind designers creating tactile graphics directly (instead of trying to translate existing visualizations), future work has an opportunity to advance tactile data representation by designing and evaluating the effectiveness of tactile-first data representations.

References

[1] American Printing House. 2019. The Tactile Graphics Image Library: Helping Students Succeed. https://www.aph.org/the-tactile-graphics-image-library-helping-students-succeed/

[2] Frances Aldrich, Linda Sheppard, and Yvonne Hindle. 2003. First Steps Towards a Model of Tactile Graphicacy. *Cartographic Journal* 40, 3 (Dec. 2003), 283–287. https://doi.org/10.1179/000870403225013014 Publisher: Taylor & Francis Ltd.

[3] Frances K. Aldrich and Alan J. Parkin. 1987. Tangible Line Graphs: An Experimental Investigation of Three Formats Using Capsule Paper. *Human Factors* 29, 3 (June 1987), 301–309. https://doi.org/10.1177/001872088702900304 Publisher: SAGE Publications Inc.

[4] American Printing House for the Blind, Inc. 1998. The Good Tactile Graphic. https://sites.aph.org/files/manuals/7-30006-00.pdf

[5] Tanni Anthony and Barbara Adams. 2005. The Definition and the Role of a TVI.

[6] Ather Sharif. 2024. I am disabled but not â€˜impairedâ€™. https://interactions.acm.org/blog/view/i-am-disabled-but-not-impaired

[7] Catherine M. Baker, Lauren R. Milne, Ryan Drapeau, Jeffrey Scofield, Cynthia L. Bennett, and Richard E. Ladner. 2016. Tactile Graphics with a Voice. *ACM Transactions on Accessible Computing* 8, 1 (Jan. 2016), 3:1–3:22. https://doi.org/10.1145/2854005

[8] John L. Barth. 1982. *Tactile Graphics Guidebook*. American Printing House for the Blind, Louisville, KY.

[9] John L. Barth. 1984. Incised grids: Enhancing the readability of tangible graphs for the blind. *Human Factors* 26, 1 (1984), 61–70. https://doi.org/10.1177/001872088402600106 tex.eprint: https://doi.org/10.1177/001872088402600106.

[10] Braille Authority of North America. 2022. Guidelines and Standards for Tactile Graphics | Braille Authority of North America. https://www.brailleauthority.org/guidelines-and-standards-tactile-graphics

[11] Diane Brauner. 2024. Teaching number line math skills: Part 2. https://www.perkins.org/resource/teaching-number-line-math-skills-part-2/

[12] Carmen Willings. 2020. Teacher of Students with Visual Impairments. https://www.teachingvisuallyimpaired.com/teacher-of-students-with-visual-impairments.html

[13] Ben P. Challis and Alistair D.N. Edwards. 2001. Design principles for tactile interaction. In *Haptic Human-Computer Interaction (Lecture Notes in Computer Science)*, Stephen Brewster and Roderick Murray-Smith (Eds.). Springer, Berlin, Heidelberg, 17–24. https://doi.org/10.1007/3-540-44589-7_2

[14] Pramod Chundury, Yasmin Reyazuddin, J. Bern Jordan, Jonathan Lazar, and Niklas Elmqvist. 2023. TactualPlot: Spatializing Data as Sound using Sensory Substitution for Touchscreen Accessibility. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 1–11. https://doi.org/10.1109/TVCG.2023.3326937

[15] David Crombie, Roger Lenoir, Neil McKenzie, and George Ioannidis. 2004. The Bigger Picture: Automated Production Tools for Tactile Graphics. In *Computers Helping People with Special Needs*, Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Dominique Burger (Eds.). Springer, Berlin, Heidelberg, 713–720. https://doi.org/10.1007/978-3-540-27817-7_106

[16] Duxbury Systems. 2024. QuickTac free software. https://www.duxburysystems.com/quicktac.asp

[17] Polly K Edman. 1992. *Tactile Graphics*. Vol. 15. American Foundation for the Blind.

[18] Christin Engel, Emma Franziska MÃ¼ller, and Gerhard Weber. 2019. SVGPlott: an accessible tool to generate highly adaptable, accessible audio-tactile charts for and from blind and visually impaired people. In *Proceedings of the 12th ACM*

International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '19). Association for Computing Machinery, New York, NY, USA, 186–195. https://doi.org/10.1145/3316782.3316793

[19] Christin Engel, Emma Franziska MÃ¼ller, and Gerhard Weber. 2021. Tactile Heatmaps: A Novel Visualisation Technique for Data Analysis with Tactile Charts. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021)*. Association for Computing Machinery, New York, NY, USA, 16–25. https://doi.org/10.1145/3453892.3458045

[20] Christin Engel and Gerhard Weber. 2017. Analysis of Tactile Chart Design. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '17)*. Association for Computing Machinery, New York, NY, USA, 197–200. https://doi.org/10.1145/3056540.3064955

[21] Christin Engel and Gerhard Weber. 2017. Improve the Accessibility of Tactile Charts. In *Human-Computer Interaction - INTERACT 2017 (Lecture Notes in Computer Science)*, Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler (Eds.). Springer International Publishing, Cham, 187–195. https://doi.org/10.1007/978-3-319-67744-6_12

[22] Christin Engel and Gerhard Weber. 2019. User Study: A Detailed View on the Effectiveness and Design of Tactile Charts | SpringerLink. https://link-springer-com.libproxy.mit.edu/chapter/10.1007/978-3-030-29381-9_5

[23] Diana Garcia-Mejia. 2024. The Central Role of the Teacher of Students with Visual Impairments. https://aphconnectcenter.org/familyconnect/education/iep-individualized-education-program-3-years-to-22-years-old/central-role-of-the-tvi/

[24] Cagatay Goncu and Kim Marriott. 2008. Tactile chart generation tool. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (Assets '08)*. Association for Computing Machinery, New York, NY, USA, 255–256. https://doi.org/10.1145/1414471.1414525

[25] Cagatay Goncu, Kim Marriott, and Frances Aldrich. 2010. Tactile Diagrams: Worth Ten Thousand Words?. In *Diagrammatic Representation and Inference (Lecture Notes in Computer Science)*, Ashok K. Goel, Mateja Jamnik, and N. Hari Narayanan (Eds.). Springer, Berlin, Heidelberg, 257–263. https://doi.org/10.1007/978-3-642-14600-8_25

[26] Cagatay Goncu, Kim Marriott, and John Hurst. 2010. Usability of Accessible Bar Charts. In *Diagrammatic Representation and Inference*, Ashok K. Goel, Mateja Jamnik, and N. Hari Narayanan (Eds.). Vol. 6170. Springer Berlin Heidelberg, Berlin, Heidelberg, 167–181. https://doi.org/10.1007/978-3-642-14600-8_17 Series Title: Lecture Notes in Computer Science.

[27] Ricardo Gonzalez, Carlos Gonzalez, and John A. Guerra-Gomez. 2019. Tactiled: Towards More and Better Tactile Graphics Using Machine Learning. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 530–532. https://doi.org/10.1145/3308561.3354613

[28] Vicki L. Hanson, Anna Cavender, and Shari Trewin. 2015. Writing about accessibility. *Interactions* 22, 6 (Oct. 2015), 62–65. https://doi.org/10.1145/2828432

[29] Lucia Hasty. 2024. Teaching Tactile Graphics. https://www.perkins.org/resource/teaching-tactile-graphics/

[30] Leona Holloway, Peter Cracknell, Kate Stephens, Melissa Fanshawe, Samuel Reinders, Kim Marriott, and Matthew Butler. 2024. Refreshable Tactile Displays for Accessible Data Visualisation. https://doi.org/10.48550/arXiv.2401.15836 arXiv:2401.15836 [cs].

[31] Laura HospitÃ¡l. 2024. Tips for Reading Tactile Graphics in Science with a Focus on State Assessment. https://www.perkins.org/resource/tips-reading-tactile-graphics-science-focus-state-assessment/

[32] International Council on English Braille (ICEB). 2024. Unified English Braille (UEB). https://www.iceb.org/ueb.html

[33] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper HornbÃ¦k. 2015. Opportunities and Challenges for Data Physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3227–3236. https://doi.org/10.1145/2702123.2702180

[34] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. 2007. Automated tactile graphics translation: in the field. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (Assets '07)*. Association for Computing Machinery, New York, NY, USA, 75–82. https://doi.org/10.1145/1296843.1296858

[35] Stephen E. Krufka, Kenneth E. Barner, and Tuncer Can Aysal. 2007. Visual to Tactile Conversion of Vector Graphics. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15, 2 (June 2007), 310–321. https://doi.org/10.1109/TNSRE.2007.897029 Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[36] Susan J. Lederman and Jamie I. Campbell. 1982. Tangible Graphs for the Blind. *Human Factors* 24, 1 (Feb. 1982), 85–100. https://doi.org/10.1177/001872088202400109 Publisher: SAGE Publications Inc.

[37] Liblouis. 2024. Liblouis - An open-source braille translator and back-translator. https://liblouis.io/

[38] Paths to Literacy. 2014. Creating Large Print and Tactile Graphs. https://www.pathstoliteracy.org/creating-large-print-and-tactile-graphs/

[39] Mrinal Mech, Kunal Kwatra, Supriya Das, Piyush Chanana, Rohan Paul, and M. Balakrishnan. 2014. Edutactile - A Tool for Rapid Generation of Accurate Guideline-Compliant Tactile Graphics for Science and Mathematics. In *Computers Helping People with Special Needs*, Klaus Miesenberger, Deborah Fels, Dominique Archambault, Petr PeÅ^Ã¡z, and Wolfgang Zagler (Eds.). Springer International Publishing, Cham, 34–41. https://doi.org/10.1007/978-3-319-08599-9_6

[40] Omar Moured, Morris Baumgarten-Egemole, Karin MÃ¼ller, Alina Roitberg, Thorsten Schwarz, and Rainer Stiefelhagen. 2024. Chart4Blind: An intelligent interface for chart accessibility conversion. In *Proceedings of the 29th international conference on intelligent user interfaces (Iui '24)*. Association for Computing Machinery, New York, NY, USA, 504–514. https://doi.org/10.1145/3640543.3645175 Number of pages: 11 Place: Greenville, SC, USA.

[41] Denise Prescher, Jens Bornschein, and Gerhard Weber. 2014. Production of Accessible Tactile Graphics. In *Computers Helping People with Special Needs*, Klaus Miesenberger, Deborah Fels, Dominique Archambault, Petr PeÅ^Ã¡z, and Wolfgang Zagler (Eds.). Vol. 8548. Springer International Publishing, Cham, 26–33. https://doi.org/10.1007/978-3-319-08599-9_5 Series Title: Lecture Notes in Computer Science.

[42] Denise Prescher, Jens Bornschein, and Gerhard Weber. 2017. Consistency of a Tactile Pattern Set. *ACM Transactions on Accessible Computing* 10, 2 (April 2017), 7:1–7:29. https://doi.org/10.1145/3053723

[43] Lauren Race, Chancey Fleet, Joshua A. Miele, Tom Igoe, and Amy Hurst. 2019. Designing Tactile Schematics: Improving Electronic Circuit Accessibility. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 581–583. https://doi.org/10.1145/3308561.3354610

[44] Naomi Rosenberg. 2024. Touching the News. https://lighthouse-sf.org/ttn/

[45] Hans Rosling. 2006. The best stats you've ever seen. https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen.

[46] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 341–350. https://doi.org/10.1109/TVCG.2016.2599030

[47] JooYoung Seo, Yilin Xia, Bongshin Lee, Sean Mccurry, and Yu Jun Yam. 2024. MAIDR: Making statistical visualizations accessible with multimodal data representation. In *Proceedings of the 2024 CHI conference on human factors in computing systems (Chi '24)*. Association for Computing Machinery, New York, NY, USA, 1–22. https://doi.org/10.1145/3613904.3642730 Number of pages: 22 Place: Honolulu, HI, USA tex.articleno: 211.

[48] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Usersâ€™ Experiences with Online Data Visualizations. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '21)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3441852.3471202

[49] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. 2022. VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3491102.3517431

[50] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. 2023. Chart Reader: Accessible Visualization Experiences Designed with Screen Reader Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. https://doi.org/10.1145/3544548.3581186

[51] Purdue University. 2002. Tactile Diagram Manual (2002, Print Version). https://www.yumpu.com/en/document/view/25596604/tactile-diagram-manual-2002-print-version-purdue-university

[52] Tetsuya Watanabe, Kosuke Araki, Toshimitsu Yamaguchi, and Kazunori Minatani. 2016. Development of Tactile Graph Generation Web Application Using R Statistics Software Environment. *IEICE Transactions on Information and Systems* E99.D, 8 (2016), 2151–2160. https://doi.org/10.1587/transinf.2015EDP7405

[53] Tetsuya Watanabe and Naoki Inaba. 2018. Textures Suitable for Tactile Bar Charts on Capsule Paper. *Transactions of the Virtual Reality Society of Japan* 23 (2018), 13–20.

[54] Tetsuya Watanabe and Hikaru Mizukami. 2018. Effectiveness of Tactile Scatter Plots: Comparison of Non-visual Data Representations. In *Computers Helping People with Special Needs (Lecture Notes in Computer Science)*, Klaus Miesenberger and Georgios Kouroupetroglou (Eds.). Springer International Publishing, Cham, 628–635. https://doi.org/10.1007/978-3-319-94277-3_97

[55] Tetsuya Watanabe, Toshimitsu Yamaguchi, and Masaki Nakagawa. 2012. Development of Software for Automatic Creation of Embossed Graphs. In *Computers Helping People with Special Needs*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos,

Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Klaus Miesenberger, Arthur Karshmer, Petr Penaz, and Wolfgang Zagler (Eds.). Vol. 7382. Springer Berlin Heidelberg, Berlin, Heidelberg, 174–181. https://doi.org/10.1007/978-3-642-31522-0_25 Series Title: Lecture Notes in Computer Science.

[56] T.P. Way and K.E. Barner. 1997. Automatic visual to tactile translation. I. Human factors, access methods and image manipulation. *IEEE Transactions on Rehabilitation Engineering* 5, 1 (March 1997), 81–94. https://doi.org/10.1109/86.559353 Conference Name: IEEE Transactions on Rehabilitation Engineering.

[57] Yalong Yang, Kim Marriott, Matthew Butler, Cagatay Goncu, and Leona Holloway. 2020. Tactile Presentation of Network Data: Text, Matrix or Diagram?. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376367

[58] Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. 2024. Umwelt: Accessible Structured Editing of Multi-Modal Data Representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–20. https://doi.org/10.1145/3613904.3641996

A Appendix: Tactile Vega-Lite defaults

Table 2: Default configuration values for tactile charts in Tactile Vega-Lite, categorized by key design elements such as Braille, alignment, and tactual hierarchy. The rationale for the default values is primarily derived from established tactile design guidelines and insights gathered from our formative interviews about existing industry practices.

Property	Default Value	Description	Guidelines
Braille			
config.font	"Swell Braille"	Font used for text in the chart.	Swell Braille produces high-quality braille on both embossers and Swell Form machines.
titleFontSize, labelFontSize, subtitleFontSize (title, axis, legend)	24	Font size for default braille font Swell Braille	Each braille font has a fixed size for standard readability, as altering the size makes it unreadable.
axis.titleFontWeight, title.fontWeight, axis.labelFontWeight	"normal"	Braille fonts do not have different font weights.	Braille text must maintain uniform weight to preserve tactile clarity.
brailleTranslation	"en-ueb-g2.ctb"	Braille translation table used, including braille grade, language and braille code.	This translation table converts English text into Grade 2 contracted braille, adhering to the UEB standard, which is widely adopted by braille readers in English-speaking countries.
Tactual Hierarchy			
axis.gridWidth	1	Width of the gridlines.	Guideline 6.6.2.2: Lines Grid lines should be the least distinct lines on the graph.
axis.domainWidth	2.5	Width of the axis lines.	Guideline 6.6.2.2 Lines The x-axis (horizontal) and y-axis (vertical) lines must be tactually distinct and stronger than the grid lines.
axis.tickSize	26.5(px)	Default tick mark length.	Guideline minimum sizing rule.
axis.tickWidth	2.5(px)	Line thickness of the axis ticks.	Same line width as the axis lines to ensure the same level of tactual hierarchy.
axis.gridColor, axis.domainColor, axis.tickColor	"black"	Color of the axis ticks.	The color black generally prints well on the embosser and swell form machine.
axis.staggerLabels	"auto"	Automatically stagger x axis label when label length exceeds threshold.	

Table 3: Default Values for Tactile Charts in Tactile Vega-Lite Cont

Property	Default Value	Description	Guidelines
Positioning			
legend.direction	"vertical"	Stacks legend entries vertically.	Vertical stacking supports easier tactile navigation for certain readers.
legend.orient	"top-left"	Determines the position of the legend in the chart.	Placing the legend in a predictable location improves usability for tactile users by minimizing search time.
axis.titleAngle, axis.labelAngle	0	Rotation angle for axis titles (0 means no rotation).	Based on tactile reading habits, users usually do not anticipate rotated angles and might be confused or take longer to read, as tactile reading relies on uniformity and consistency for efficient interpretation.
Alignment			
title.align	"center"	Alignment of chart title and other elements.	Guideline 5.3.1: The most commonly used heading in a graphic is the centered heading. It is used for the title of a graphic.
axis.titleAlign	"left"	Left align title text.	Guideline 6.6.4.5: The heading label for the horizontal values should be placed below the values and should be left-justified with the first cell of the first horizontal value.

Table 4: Default Values for Tactile Charts in Tactile Vega-Lite Cont

Property	Default Value	Description	Guidelines
Spacing and Size			
title.offset	50	Add spacing between the title and chart area.	Guideline 5.3.1: Blank lines should be left before and after centered headings.
axis.titlePadding	20	Padding, in pixels, between the axis title and the axis.	Guideline 6.6.4.5: The heading label (axis title) for the horizontal values should be placed below the values.
axis.labelPadding	20(px)	Padding, in pixels, between labels and axis ticks.	Guideline 6.6.4.5: On the horizontal axis, value should be spaced 1/8 inch from the tick mark or axis line.
axis.titleY	-10(px)	Y-coordinate offset for the axis title relative to the axis group.	Guideline minimum spacing rule of 1/8 inch.
legend.titlePadding	20(px)	Padding, in pixels, between the legend title and the legend.	Guideline minimum spacing rule of 1/8 inch.
legend.offset	20(px)	Padding between the bottom legend and the top of the chart.	Guideline minimum spacing rule of 1/8 inch.
legend.columnPadding	20(px)	Padding between legend columns.	Guideline minimum spacing rule of 1/8 inch.
legend.rowPadding	20(px)	Padding between legend rows.	Guideline minimum spacing rule of 1/8 inch.
config.padding	{"top": 100, "bottom": 100, "left": 200, "right": 200}	Padding around the chart to ensure elements are not cut off.	Guideline minimum spacing rule of 1/8 inch.
legend.symbolSize	3000	Size of legend symbols.	Guideline minimum sizing rule.