Slide Gestalt: Automatic Structure Extraction in Slide Decks for Non-Visual Access

Yi-Hao Peng*
Carnegie Mellon University
Pittsburgh, PA, USA
yihaop@cs.cmu.edu

Peggy Chi
Google Research
Mountain View, CA, USA
peggychi@google.com

Anjuli Kannan
Google
New York, NY, USA
anjuli@google.com

Meredith Ringel Morris
Google Research
Seattle, WA, USA
merrie@google.com

Irfan Essa
Google Research, Georgia Tech
Atlanta, GA, USA
irfanessa@google.com

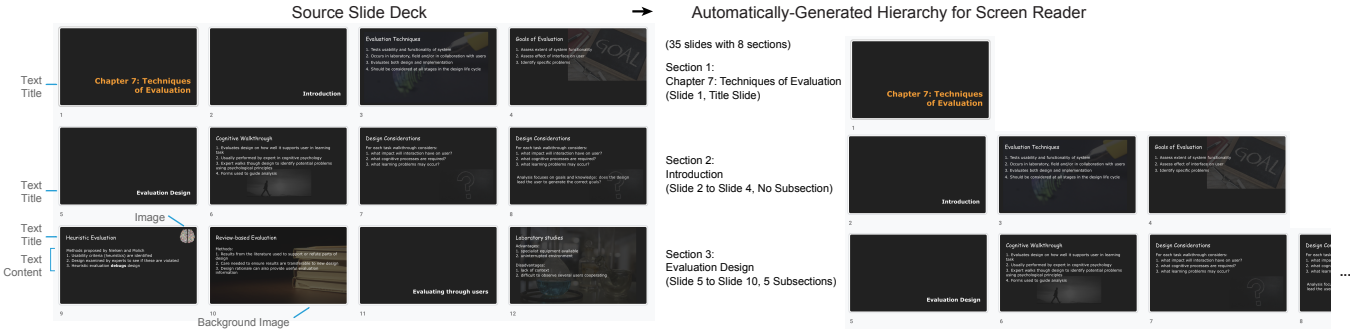

Figure 1: Slide Gestalt automatically generates hierarchical groupings based on the visual and textual correspondences between slides in a slide deck. It helps readers to navigate from the higher-level sections to the lower-level descriptions via our accessible interface.

ABSTRACT

Presentation slides commonly use visual patterns for structural navigation, such as titles, dividers, and build slides. However, screen readers do not capture such intention, making it time-consuming and less accessible for blind and visually impaired (BVI) users to linearly consume slides with repeated content. We present Slide Gestalt, an automatic approach that identifies the hierarchical structure in a slide deck. Slide Gestalt computes the visual and textual correspondences between slides to generate hierarchical groupings. Readers can navigate the slide deck from the higher-level section overview to the lower-level description of a slide group or individual elements interactively with our UI. We derived side consumption and authoring practices from interviews with BVI readers and sighted creators and an analysis of 100 decks. We performed our pipeline with 50 real-world slide decks and a large dataset. Feedback from eight BVI participants showed that Slide Gestalt helped navigate a slide deck by anchoring content more efficiently, compared to using accessible slides.

CCS CONCEPTS

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

KEYWORDS

Accessibility; Slide deck; Presentation; Screen reader; Multimodal correspondence and alignment

ACM Reference Format:
Yi-Hao Peng, Peggy Chi, Anjuli Kannan, Meredith Ringel Morris, and Irfan Essa. 2023. Slide Gestalt: Automatic Structure Extraction in Slide Decks for Non-Visual Access. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3544548.3580921

1 INTRODUCTION

Presentation slides (e.g., created via Apple Keynote, Google Slides, and Microsoft PowerPoint) are a popular format for people to convey ideas through a series of slides with graphics and text, widely used in lectures, talks, and meetings. In addition to their use for giving presentations, slide decks' simple, linear format also makes it easy to share and follow the content asynchronously. However, studies show that slide decks are often inaccessible to blind and

*This work was done while the author was a Student Researcher at Google.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580921

visually impaired (BVI) users who consume via a screen reader. Issues include a lack of metadata (such as alternative text for figures and tables [47]) and incorrect reading orders of elements within each slide, which can cause misinterpretation [27, 53]. Furthermore, existing authoring practices of visual presentations potentially introduce challenges for screen reader users to follow. Slide authors commonly build content hierarchies and consistent visual patterns for a compelling presentation via similar titles, layouts, and build slides [13, 62]. Such a design intention is never revealed by a screen reader. This leads to repeated content when a sequence of slides contains slightly modified materials, e.g., an added sentence to a list, or a bounding box to highlight an element. As a result, screen reader users must consume the full readout while mentally tracking and filtering redundant information.

Prior art makes presentation slides more accessible by automating captions or encouraging authors to annotate metadata [45, 47, 49]. There are computational approaches to create a high-level outline of a slide deck for sighted users to quickly access to a specific slide [3, 13, 24]. While these methods are effective to describe slide elements, we argue that it is critical to reveal a deck's structure and styles contextually (e.g., overview) while reading individual component (e.g., detail). Challenges remain in how to capture patterns between slides from an in-depth understanding of the text and visual elements and their intentions.

In this paper, we present Slide Gestalt, an automatic approach that identifies the hierarchical structure in a slide deck to help BVI slide readers navigate efficiently. Slide Gestalt provides BVI readers access to up to two levels of the structure (via sections and subsections) in a slide deck. It identifies visual and textual correspondences between slides, and generates the hierarchical slide grouping based on alignment scores. Slide Gestalt's interface allows users to flexibly navigate between a high-level overview of the slide structure and detailed descriptions of the slide groupings and elements. This enables readers to quickly gain slide information with different levels of granularity based on their goals, such as to skim, search, or scrutinize.

We conducted an analysis with 100 slide decks and interviewed seven presentation authors and four BVI slide users to derive common authoring and reading practices that inspired our design. To demonstrate the feasibility of our automatic approach, we performed Slide Gestalt's end-to-end pipeline with 50 real-world slide decks we collected and report the performance (F1-Score=0.81). We evaluated our results with eight BVI readers by reviewing the content and structures of two lecture slide decks with similar topics, lengths, and hierarchies. Using Slide Gestalt, participants skimmed through more slides than using the baseline interface (29.3 vs. 19.4 number of slides). They answered informational questions about the slide decks by anchoring the relevant information more quickly with our UI (45.8 vs. 96.9 seconds) and navigating less redundant elements (6.5 vs. 20.6 number of elements) than when using the baseline interface. All participants preferred Slide Gestalt to the baseline interface and to their prior experiences with slide software.

To summarize, we make the following contributions:

- Formative studies of consuming and authoring practices in common slide decks.
- An automatic approach that identifies and generates a hierarchical structure for a slide deck.
- Evaluations of the quality and utility of our automatically generated hierarchies and descriptions of slide decks, including a technical analysis and a study with eight BVI participants.

2 RELATED WORK

Our research builds on prior work of tools for slide creation and navigation support, and techniques to make slides and other applications accessible.

2.1 Slide Authoring and Structuring

Software applications (such as Apple Keynote, Google Slides, and Microsoft PowerPoint) have made slide creation available to a wide audience [28]. Advanced techniques further support effective creation of slide content from a document [56], data [63], or sketches [37], and to animate elements [65]. Most software restricts a slide deck to be linear, i.e., slides are organized sequentially. To structure a slide deck, some tools allow users to manually group consecutive slides and optionally annotate section titles [2, 38]. Prior work proposed computational methods to create an outline of a slide deck given a user input [3] or by slide components [24]. Users could manage the flow and time via a directed graph [12, 58], or to flexibly author between an outline and a canvas of slide content [13]. Finally, by measuring slide similarities, users could inspect content progression between multiple presentations [10] or reuse materials [57].

We are inspired by these prior efforts that automatically generate an outline or connect multiple slides or decks to enhance authoring experiences. However, while other research has shown that hierarchical information can be useful to navigate multimedia like movies [43], tutorial videos [8, 59], live streams [17], and images [34], it requires research to understand how revealing a slide deck structure could further help readers consume the content, especially for BVI users with a screen reader.

2.2 Presentation Consumption Support

People create slides widely for sharing. A common use case for slide creation is to perform a live presentation, where users control the slide playback using presentation software while talking through the content. To assist presenters in better control, researchers have developed interactive tools to visually focus on materials [20] or follow a visual hint for better pacing [9, 58].

Another use case is for sharing slides asynchronously without a talk-through. For efficient slide deck consumption, He et al. proposed a summarization method dedicated for slide decks [25], which can be especially useful for readers who are not familiar with the content. Outlines and graphs support not only authoring, but navigation [3, 12]. However, to consume individual slides in a deck, it requires time to process information linearly for both general and BVI users. We focus on supporting the offline reading experience of a slide deck, specifically to identify the underlying structure that can be read explicitly by a screen reader. In turn, we aim to improve user's understanding of a deck structure and avoid redundantly consuming sequences of similar content.

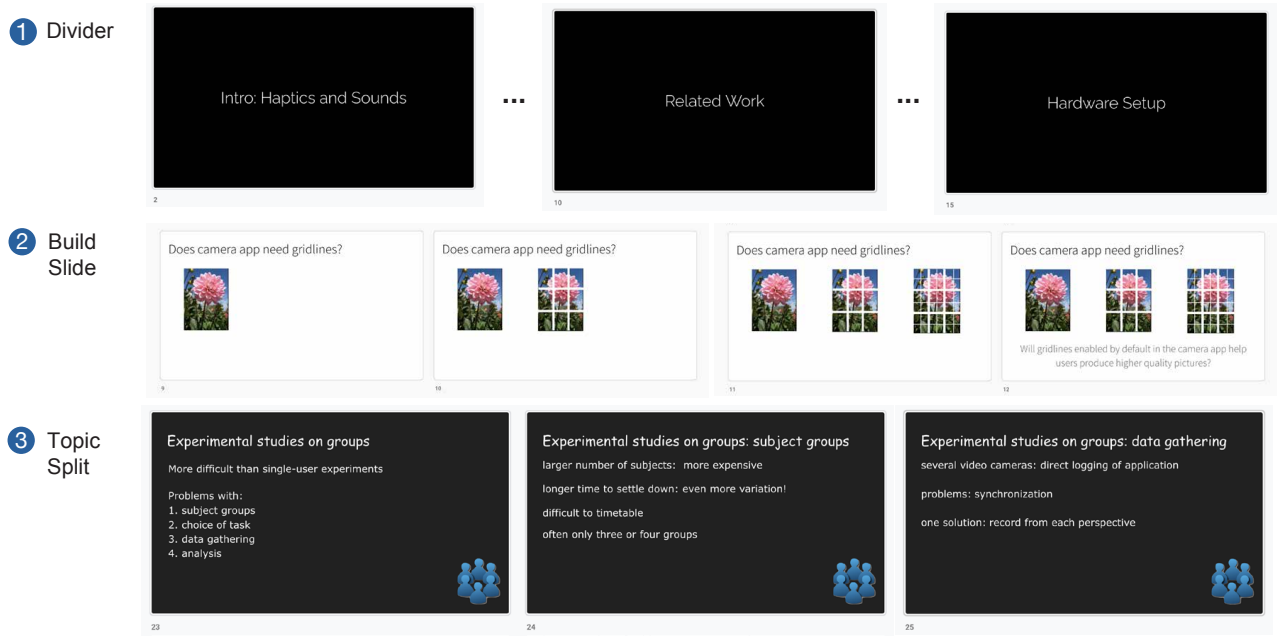

Figure 2: Structure patterns and examples that we observed in our slide deck analysis, including (1) a divider pattern for transitioning to different sections, (2) a build-slide pattern that shows elements one by one, and (3) a topic-split pattern that breaks similar topics into a sequence of slides.

2.3 Improving Slide Content Accessibility

Presentation tools make it easy to create visual supports for conveying a concept via graphics and animations. However, without a live presentation, slides are often inaccessible to BVI audiences due to the lack of alt text describing visual elements [47, 49] or the incorrect read order of slide elements [39]. Prior work has introduced techniques to enable efficient slide annotations from authors. Sato et al. [54] and Ishihara et al. [27] built systems to annotate slide diagrams and convert a slide deck into the HTML format for screen readers. Peng et al. [45] demonstrated automatic techniques to extract slides in a presentation video and convert them to an accessible slide deck on demand, which allows BVI users to read elements that were not explicitly described by the authors. While these approaches improved the reading experiences for BVI users, they are primarily designed for making slides accessible at an element level. Our work instead focuses on making slides accessible at a deck level. By generating a high-level overview and low-level groupings, we investigate how hierarchical information could enable screen reader users to navigate a slide deck more efficiently.

2.4 Accessible Technologies for Multimedia Formats

Our work is built upon prior research that enables accessible content of structured formats, including web pages [31], PDF documents [4], mobile applications [16, 48, 64], and graphical user interfaces [34, 35, 61]. For other multimedia formats such as an edited video, recent research improves the accessibility based on visual signals and text transcripts [26, 32, 41, 44, 46]. These efforts provide us insights to design an automatic approach to enhance existing slide decks. We suggest that similar to a web document, a slide deck contains a hierarchy of text (including titles and supporting text) and visual (images, figures, and graphics) elements. However, it is different from prior work given its unique slide-based format, where recurring patterns could exist in or between a series of slides that this paper focuses on.

3 UNDERSTANDING SLIDE DECK ACCESSIBILITY

To better understand the accessibility issues of a slide deck from both audiences' and authors' perspectives, we conducted (1) informal interviews with four BVI slide readers to learn their current reading experiences, and (2) formal interviews with seven sighted slide authors to obtain their authoring practices. We further reviewed the content structures of 100 slide decks to verify the generalizability of our findings in the interviews. While studies have examined the accessibility of individual slides or canvases [55], our work is the first to investigate the accessibility of slides both at the individual slide level and the deck level, which considers a series of slides and their patterns.

3.1 Informal Interviews with BVI Slide Readers

We conducted an informal study of 30-minute online interviews with four BVI participants (two female and two male, age=32-41) to understand their current practices for reading slide decks. All participants were active slide readers who consumed multiple slide decks using screen readers on a weekly basis for work and education

needs. We asked participants to share their experiences following slide decks offline and any challenges they encountered.

All participants reported that they read slide decks using either Microsoft PowerPoint or Google Slides. They mainly read the elements in each slide sequentially through the presenter mode to avoid accidentally editing the slides. They noted that they often focused on the text content in a slide deck given that visual elements (such as images or charts) rarely had alternative text.

Participants shared that they would identify incorrect element read order in a slide (e.g., the title of the slide was read out after other elements). Furthermore, they found the "recurring" patterns that appeared across the slide deck confusing. They occasionally got lost between slides when the screen reader repeated the same or similar elements. They would spot these repetitions from identical elements when switching between slides, or between similar titles across a sequence of slides. This made it difficult for participants to navigate through the slide deck.

Participants found the inconsistent length of content between slides confusing at times, when some slides contained a short word or sentence, while others included a lot of text and visuals. Such inconsistency made it unclear whether a slide contained new information. To address these accessibility issues, participants suggested adding element annotations (such as alt text and read order) and explaining the purpose of the recurring structures or patterns (such as sections and animations) of a slide deck.

In response to the issues (e.g., missing alt text, incorrect read order, confusing recurring patterns, and inconsistent length of content) identified in our informal study with BVI readers, we conducted a follow-up formal study with sighted slide authors to better understand their current design strategies and how they impact the accessibility of the resulting content for BVI individuals.

3.2 Interviews with Slide Authors

We recruited seven sighted participants (four female, three male, age=26-35) from our organization and conducted 30-minute semi-structured online interviews to gain insight into their slide creation practices. All of our participants regularly created slides for work or educational purposes. During the interview, we asked participants to share the tools they commonly used to create slides, their creation process, and to provide 1-2 examples of slide decks they had recently created. We asked about their experiences organizing and presenting materials, and how they would describe the hierarchies or structures in the slide decks. We inquired if they were aware of any annotations or design patterns, such as alt text, read order, and repetitive elements, that could impact the accessibility and readability of a slide deck for BVI audiences.

3.2.1 Current practices. All of our participants had experience creating slide decks with PowerPoint, Google Slides, and Keynote (three regularly used PowerPoint, three used Google Slides, and one used Keynote). Six of them shared that they created their decks by adding slides one by one in a linear order. One participant preferred to draft the main points they planned to include in a document before creating the deck. Participants mentioned that they sometimes created an "outline" slide to describe the overall topic, especially when the slide deck became longer. The outline slide would not cover every high-level subject or low-level detail in the slide deck. Overall, we found that sighted authors often organized and presented slide information in a visually cohesive way by using elements with similar visuals or semantic attributes across slides.

3.2.2 Authoring Patterns. Through analyzing the example decks and feedback from participants, we identified three common structures (divider, build, and topic split slides) that authors frequently used in their slide decks:

Divider slides, also known as section divider slides, visually divide a slide deck into segments, such as sections or subsections (see Figure 2-1). This pattern helps create a consistent and organized visual presentation. Divider slides often include a short section title with a distinguished background color or graphics, or recurring bullet points highlighting the progression of an agenda.

Build slides are used when a series of visually similar slides simulate a graphical animation (see Figure 2-2). This pattern presents content gradually, making it easier for the audience to focus on one element at a time. Though this design has a similar effect to adding animation to individual slide elements, all participants noted that they preferred using build slides over element-based animations in one slide. They explained that build slides offered benefits for presenters to split their speaker note to multiple slides, navigate more efficiently, and avoid repetitive animation configuration of elements. They could also utilize transitions to create a more engaging and dynamic visual presentation.

Topic split slides are a design technique where a single topic is divided into multiple slides with similar or identical titles (see Figure 2-3). This can effectively reduce visual clutters and make a presentation organized and connected. For example, the second and third slides in Figure 2-3 expand upon the same title with more context. Another common example is adding "*(continued)*" or "*(cont'd)*" to the original title on follow-up slides. This pattern helps indicate that the topic is being continued from the previous slide.

3.2.3 Accessibility Considerations. These structure patterns are primarily designed for a visually appealing and engaging presentation. We further discuss the accessibility implications of these authoring techniques with sighted authors. All participants acknowledged that they were not fully aware of the challenges that BVI slide readers faced in accessing the content. After learning about the potential accessibility issues, they found it challenging to address these concerns in practice. On the slide level, participants found it time consuming to add alternative text or correct the read order of elements. At the deck level, they would not explicitly describe the structures due to the labor-intensive process of manual annotations. When asked about how they would describe the hierarchy of the slide decks if more flexible annotation tools existed, all participants said that they would mostly use the first title of each grouping to describe the grouping or structures. If there is no title in the initial slides, they would look at the visuals on the slides and describe them as titles, or use the title on the next-available slides.

Through our studies with BVI slide readers and sighted slide authors, we observed that common design decisions made by sighted authors are visual driven without considering the non-visual representation for screen readers. This leads to accessibility issues for BVI users. We suggest that by automating the annotation and

description process of the slide structure, it could improve the accessibility of a slide deck for blind readers, and possibly enable sighted authors to improve annotations more efficiently.

3.3 Accessibility Analysis of Content and Structures in Slide Decks

To further confirm the design consideration for an automatic tool, we conducted an analysis of 100 existing slide decks. Our goal is to obtain a comprehensive overview of the design strategies that authors frequently employ, as well as the accompanying accessibility issues of the slide deck structure from authoring practices.

3.3.1 Data. We collected a diverse range of slide decks from public sources and within our organization, covering a wide range of topics, from technology, design, marketing, to science. We randomly selected a set of slide decks for each topic and filtered decks that were too short (less than 10 slides) in order to better observe the structures. In total, we obtained 100 slide decks with 25 decks for each of the four selected topics. The average number of slides per deck in each topic was between 32 and 36. All slides were served by Google Slides [22] as the platform allows for consistent sharing and reading of slides across different operating systems, making it ideal for our analysis.

3.3.2 Annotations. For each slide deck, we annotated each slide using the common structures (divider, build and topic split, see Figure 2 for examples[1]) and groupings (sections, subsections) that we learned in the interviews. Two annotators (authors of this paper, sighted) decided the definition of structures (i.e., the repetitive structure shown throughout the slide deck) by following prior research [19, 47] to first reach the agreement on the small set of slides and then annotate the rest afterwards.

3.3.3 Results: The 100 slide decks we inspected contained 33.75 slides on average (σ = 20.85, [min, max] = [15, 120]) and 7.50 elements (e.g., texts, images, shapes) per slide (σ = 3.45). 94% of the slide decks included a title slide, all presented as the first slide in a deck. Only 21% of the decks included outline slides, and these decks contained 67.18 number of slides on average (σ = 15.56). This is aligned with our interview finding with sighted authors, where outline slides were usually added for a long deck.

We observed 44% of the slide decks included at least two divider slides. Dividers were only used in decks of more than 21 slides in our collection. 58% of the decks included at least one group of build slides (with 3.04 slides in each group on average). Meanwhile, we found in-slide animation in 33% of the decks, with an average of 2.25 elements animated on a slide. Finally, 73% of the slide decks included at least one group of topic split slides (with 3.25 slide in each group on average). The average number of levels in the slide decks was 1.74 (σ = 0.69; each level deck counts: 0-level=4, 1-level=26, 2-level=64, 3-level=4, and 4-level=2).

Regarding the accessibility of the collected slides, we found only 1.1% of image elements contained alt text and 5.5% of slides had a correct read order. In total, only 2% of slide decks were fully accessible with complete alt text of all slide elements and the correct read order. Through the analysis of the collected slide decks, we confirmed that the slide content and patterns raise accessibility concerns, which aligns with our interview findings.

[1]The examples used in this paper are intentionally selected to include only lectures or coursework, presenting insensitive topics.

3.4 Reflection

Our discussions with BVI slide readers suggest that there are barriers to follow content and patterns in a slide deck using screen readers. One of the issues is that the information is primarily designed and presented using visuals, which rarely support non-visual representations (e.g., alt text, section or animation descriptions). This creates a significant challenge for BVI users who intend to access and understand the information in a slide deck. Our interviews with sighted slide authors and the analysis of existing slide decks revealed that there are multiple reasons for the common inaccessibility stemmed from slide creators, including (1) the intention to create visually appealing narratives, (2) the design strategies [7, 11, 51] that may not be interpretable by screen readers (e.g., to create segments with dividers, present a topic gradually with build slides, and to break a complex topic into pieces with topic split slides), and (3) a lack of awareness or willingness to improve the readability for BVI audiences.

To make slide decks accessible to BVI individuals, guidelines suggest incorporating non-visual representations of the information, such as alt text [1, 15, 33]. In addition to adding annotations per element, guidelines also recommend creating correct sequences [1, 14, 42, 60]. The principle applies to both the slide and deck levels in order to explain the context (e.g., element orders and section titles), intention (e.g., to show the structure of building animation or splitting content), and progress (e.g., to indicate the start and end of a pattern) in a slide deck.

In this work, we build upon existing efforts on understanding and making slides accessible at the element level (alt text and read order) [5, 6, 27, 55] and further address accessibility at the deck level by identifying implicit slide hierarchies (section and subsections) and explaining the context and design intention of visual structures (dividers, build and topic split slides). Our goal is to provide an automatic approach to making slide decks more accessible and interpretable to BVI slide readers. We aim to improve accessibility for the visual design of a slide deck for BVI users to access and follow the slides in a structured way.

4 SLIDE GESTALT

We present Slide Gestalt, an automatic approach that extracts the structure embedded in a slide deck, which in turn enables BVI users to efficiently navigate the slide content via two key components in an independent reader view (see Figure 3): (1) a *structure description header*, which supports user navigation from high-level sections to low-level slide grouping descriptions of a slide deck, and (2) a *slide element description list*, which lays out elements in each slide as an ordered list of descriptions (including text content, image alt text, and shape types) for the detailed content. Below we illustrate the design of each component:

The **structure description header** displays the extracted structure of a slide deck in a two-level hierarchy (see Figure 3a, b, and c that outlines the deck overview). The first level, a section description header, shows the overview of a section, each has the

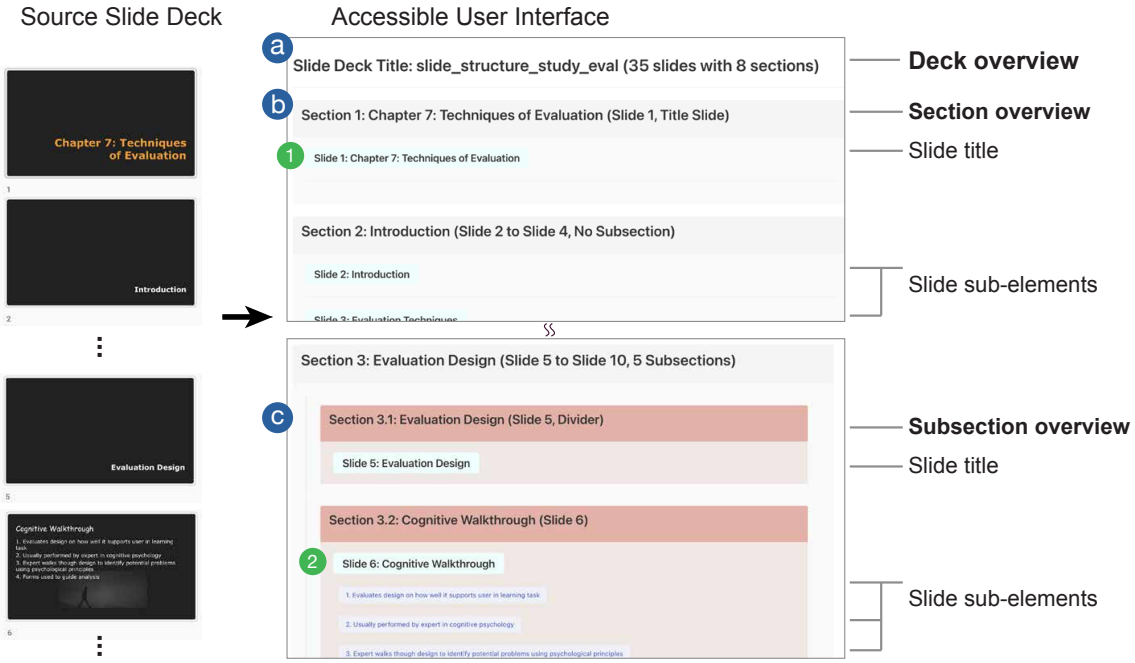

Figure 3: Slide Gestalt's accessible user interface reveals the hierarchy of the input source slide deck, including (a) the deck overview that describes the overall structure, (b-c) the section and subsection overviews that show the structure type, header, and indices, and (1-2) the slide element description list.

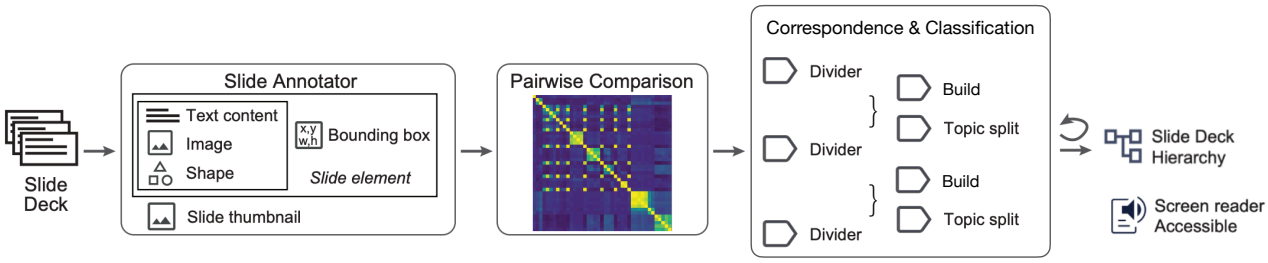

Figure 4: Slide Gestalt's automatic pipeline for identifying a slide deck's two-level hierarchy. By comparing the visual and text correspondence, Slide Gestalt classifies the slide groups and generates a multi-level hierarchical tree.

description, the start and end slide number, and the number of subsections in the section (see Figure 3b, tagged as <h2> in our UI). The second level, a subsection description header, contains the overview description of a subsection and the structure type identified by Slide Gestalt, i.e., divider, build, or topic split slides (see Figure 3c, tagged as <h3> in our UI). For subsections of either the build or topic split pattern, the description contains the slide numbers of the start and end slides to indicate the length of a slide group. The build slide pattern shows additional instruction in the header to allow users to jump to the slides with the most complete information.

The **slide element description list** shows all the elements in a slide (see Figure 3.1 and 3.2 for examples). Each element list starts with a header element that contains the slide number and title (extracted from metadata) and allows built-in navigation using screen readers.

Finally, we provide an **Editor Mode** for slide authors to modify the visual descriptions (including image alt text and shape type) and the structure information of the slide deck.

5 ALGORITHMIC METHOD

Slide Gestalt automatically annotates an input slide deck to retrieve its text and visual elements and locations. It identifies and describes the structure of the slide deck for BVI readers by (1) computing correspondence between slides, (2) detecting dividers and groupings in the slide deck with visual and semantic distances, and (3) generating hierarchical groupings and descriptions. Figure 4 shows our end-to-end pipeline.

5.1 Extracting Slide Deck Data

We built an automatic pipeline that renders and processes an input slide deck on the cloud using the Google Slides API [21]. Our Slide Annotator retrieves individual visual elements on each slide, including text, images, tables, and shapes (such as a box or a line), sorted by their depth ordering. It annotates each element's bounding box, which is the absolute region relative to the slide canvas. It captures the thumbnail of each slide based on the final state after animation if any, stored as image byte.

5.2 Computing Slide Correspondence

With the slide annotations, Slide Gestalt computes the correspondence between slides based on their visual and textual properties. We intentionally remove the first slide in the input deck with an assumption to infer it as a title slide, learned from our analysis. For the remaining slides, we featurize both the slide visuals (thumbnails, element layouts) and the content (titles, elements) using neural embedding models [29, 36, 40] to construct an embedding for each slide. Next, for each slide pair (define source $slide_i$ and target $slide_j$) in the deck of N slides beside the title slide, we construct a distance matrix $D \in \mathbb{R}^{NxN}$ where $D_{i,j}$ refers to the cosine distances between $slide_i$ and $slide_j$ embedding vectors. The resulting distance matrix is then fed into an unsupervised sequence alignment model [52] (that was used in machine translation tasks to perform many-to-many entity mapping) to cluster the slides into an initial set of slide groupings via the iterative matching process and the optimization algorithm to set the threshold for entity alignment. As a result, each slide aligns to a range of corresponding slides (from 0 to $N-1$) throughout the deck. Slide Gestalt performs this iterative process to generate a set of groups with corresponding slides for classification.

5.3 Detecting Dividers and Slide Deck Groupings

With the initial set of slide groupings, we next detect the underlying structure to improve accessibility for a screen reader that we identified in Section 3. Specifically, we focus on constructing a two-level hierarchy, as it is one of the most common multi-level structures. First, to detect dividers that create a high-level structure, we use a rule-based approach by calculating the "deck distance" between slides. We define the deck distance as the difference of slide indices. We first filter groups that contain only one slide. Groups that contain non-consecutive slides (i.e., with a sufficient gap between the slide indices within a group) are then classified as dividers (see the example shown in Figure 2.1). When multiple groupings are considered as dividers, we find the one that covers the largest range of slides as the first-level section dividers. Though our approach supports structure detection for a slide deck with more than two levels in a hierarchy (e.g., it is possible to iteratively generate slide dividers within each section by the initial dividers), we chose to present two-level information in our interface. This avoids a lengthy section title and complex structure, such as "section 2.1.1.1" that can be difficult to trace by a user.

Next, for slides in each section (segmented by the first set of dividers), we apply the same neural model and matching algorithm to generate the groupings of sequential slides based on slide title encoding. If the subgroups are found in the initial set of slide groupings, we label them as build slides (Figure 2.2). Otherwise, we label them as topic split slides (Figure 2.3). We consider the rest of the slides with no structure label as independent slides, where the screen readers can read them individually.

5.4 Generating Structure Descriptions

Based on the detection results, Slide Gestalt generates the description for each structural pattern. To describe the structures with simple and concise terms, we use the first slide title in a sequence as the description of a section or a subsection. If there is no text title in a slide (e.g., with only image or graphical elements), we use the author-provided alt text or the automatically-generated image caption (using Google Vision API [23]) of the largest image presence (which indicates its importance [50]) as the description. Besides the section and subsection titles, we embed additional details for each structure, including the range of slides in the section or subsection, the number of subsections in a section, and the classified structure (divider, build, and topic split slide) and its detailed elements, as described in Section 4. Finally, Slide Gestalt adjusts the element read order in each slide based on the distance between the position of elements and the upper-left corner (closer placed first).

5.5 Limitation

Our current approach does not consider in-slide animation detection, despite it being a common technique that sighted authors apply. Animations might lead to content occlusions or overlapped elements in a thumbnail image, which can potentially affect the aligning performance of our correlation matrix. In addition, our method requires a direct access to slide elements, which may not always be available for certain formats, such as PDF slides. We suggest processing metadata of animated elements and applying computer vision techniques to improve the accessibility, such as object detection and optical character recognition, in future work.

We acknowledge that our assumption of sections and groupings in a slide deck, learned from our formative analysis, may not always hold true. Slides can be structured in a more recursive manner, or may not have an explicit structure at all. In such cases, future opportunities include providing an authoring tool to encourage authors to describe the structure explicitly for screen readers.

Finally, our current approach incorporates multiple models for content analysis in a slide deck. With the recent advancements in zero-shot or few-shot learning and reasoning using large language models, we suggest future work to investigate translations and effective reasoning between different modalities.

6 TECHNICAL EVALUATION

To our knowledge, there is no public slide deck dataset that contains annotations of multi-level structures or fine-grained element labels. To evaluate the effectiveness of Slide Gestalt, we created a dataset of 50 slide decks with quality content annotations. We further inspected how our approach can be generalized to a large-scale dataset with loose labels and sampled the results. We describe our data, evaluation methods, and results below.

	Precision					Recall					Macro-F1				
Source	**Overall**	Divider	Build	Topic Split	None	**Overall**	Divider	Build	Topic Split	None	**Overall**	Divider	Build	Topic Split	None
Visual-only	**0.69**	*0.78*	*0.86*	*0.88*	*0.52*	**0.84**	*0.83*	*0.91*	*0.76*	*0.87*	**0.76**	*0.80*	*0.88*	*0.66*	*0.65*
Content-only	**0.81**	*0.75*	*0.79*	*0.90*	*0.80*	**0.68**	*0.62*	*0.69*	*0.88*	*0.51*	**0.74**	*0.69*	*0.72*	*0.89*	*0.62*
Hybrid	**0.79**	*0.78*	*0.82*	*0.77*	*0.78*	**0.83**	*0.76*	*0.87*	*0.85*	*0.82*	**0.81**	*0.77*	*0.84*	*0.81*	*0.80*

Table 1: The performance of our generated correspondence and groupings, including the Precision, Recall, and F1 score. Overall, our pipeline achieved a better performance using a hybrid approach with both visual and semantic data than using either visual-only or content-only information.

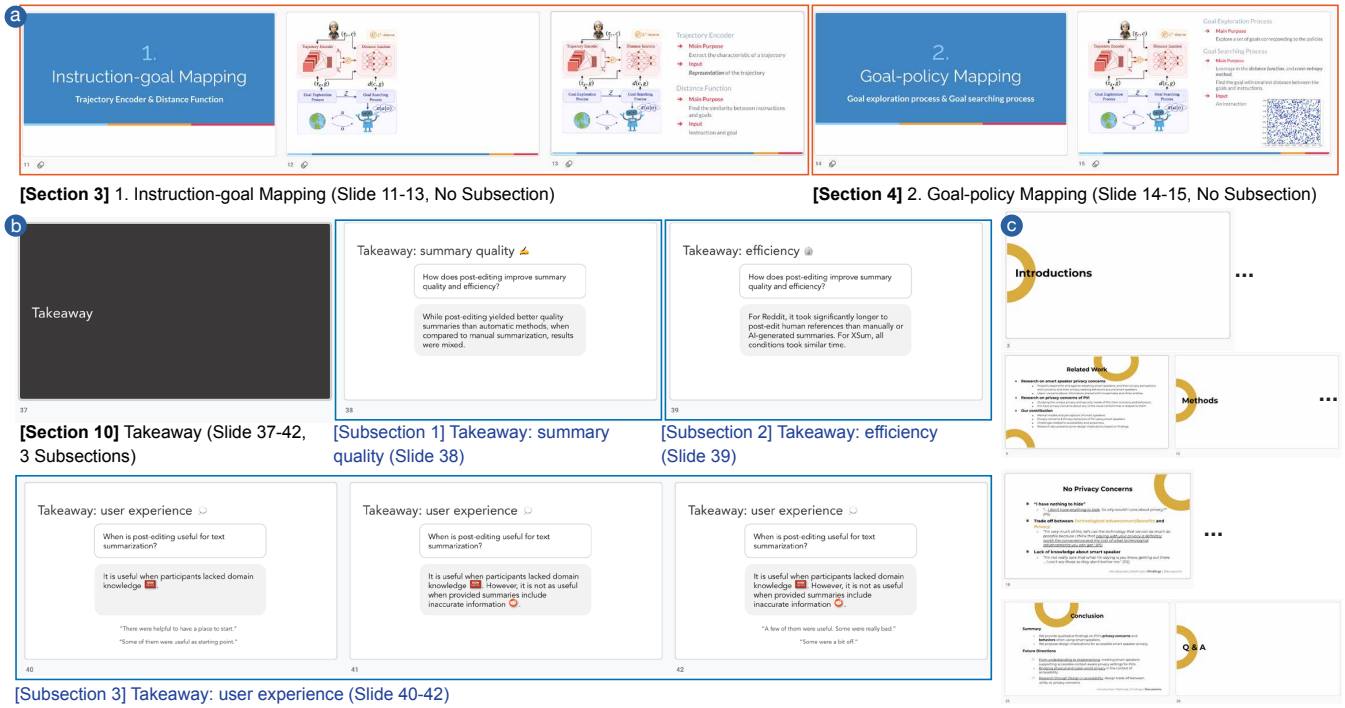

Figure 5: Examples of automatically-generated results by Slide Gestalt: (a) For similar slides separated by divider slides, our pipeline is able to create correct hierarchical sections. (b) Slide Gestalt successfully identifies subsections that fulfill the topic split pattern, although the slides have high visual similarity. (c) For a slide deck where the layouts from a template have similar graphical designs, Slide Gestalt correctly avoids under segmentation via our multimodal approach.

6.1 Test Dataset

We collected 50 slide decks that were created by three of the authors of this paper prior to the start of this project, between year 2018 and 2022. We did not modify the slide content or ordering. The length of these decks ranged between 18 and 118 slides. The topics varied from science, education, engineering, and design for lectures or conference talks. We ran the Slide Annotator of our pipeline to extract individual elements of each slide and retrieve fine-grained labels via the Google Slides API [21]. The average number of slides in a deck is 37.44 (σ = 19.64, [min, max] = [18, 118]) with 7.8 elements per slide (σ = 3.20), which is similar to the the data we observed in Section 3. Then, we manually annotated the ground-truth hierarchy and groupings based on the findings we derived in Section 3, to label each slide as a title, divider, build, topic split, or no pattern up to two levels.

6.2 Method

We evaluated the performance of different slide properties as input sources for identifying slide correspondences within each deck. Specifically, we examine three types of slide data sources (1) *visual-only*: how well the pipeline performed with visual data, such as thumbnails and element layouts, as features, and (2) *content-only*: how well the pipeline performed with content-based data, such as titles and element descriptions, as features, and (3) *hybrid*: how well the pipeline performed with visual and content properties altogether as features.

For featurization, we encoded the selected data using a combination of pre-trained vision [29, 36] and document [40] models. Given that our generated hierarchies were dependent on the quality of slide correspondence derived from the featurized embeddings, we evaluated our results based on the performance of the computed correspondence [52]. We calculated the precision, recall and F1-score of the classification results against our ground-truth labels and report the overall macro F-1 scores.

6.3 Results

Table 1 shows the F1-score of the input sources that we specified to denote Slide Gestalt's general and class-specific performance. We observed that the divider pattern and the build slide pattern could be better captured by visual characteristics (the visual-only approach shows high F1-scores.) On the other hand, the topic split pattern tends to be correlated with each other semantically (the content-only approach shows high F1-score.) We found that the hybrid approach achieves the best overall generation quality as compared to visual-only or content-only approach alone. The higher recall rate implies that the hybrid method tends to avoid over-grouping and segmentation, which is derived from the characteristics of content embedding comparison where our pipeline was conservative to generate structures (demonstrated by the high recall rate of using content-only source data from slides). The visual-based approach shows slightly better performance compared to the content-only approach since some structures were instantiated by visual characteristics (e.g., dividers), yet the high sensitivity of the vision-only approach leads to a low precision rate where the pipeline generated more incorrect mappings and segments. Based on the results, we chose a hybrid approach that considered both visual and content labels to generate the structures and descriptions that we used in our user study (Section 7).

6.4 Large Dataset

Visual styles and formats of a slide deck vary significantly according to authoring choices, target audiences, and presentation goals. We aim to further validate the generalizability and stress test of our pipeline. We created a dataset of over 121,000 slide decks served on Google Slides that were shared openly by the slide authors among our organization. The topics in this large-scale dataset covered tech talks, academic research, software or device tutorials, marketing, operations, and other subjects.

To handle data labeling at this scale, we built a fully automatic pipeline that annotated the corpus and retrieved the elements with less fine-grained labels without human inspection. For examples, the thumbnail image might partially render a slide if the source URL of an image element became invalid; text of different fonts and sizes in the same box was not separately marked; the bounding box of a text element covered the entire text box instead of the words.

While we were unable to record the creation dates of these decks, we observed the data spanned from Year 2011 to 2022, based on the dates listed on the first slide from a random sample of 50 decks. The length of these decks ranged from 1 and over 300 slides. We carefully processed this internally-public data in our organization for research purposes under data policy.

ID	Gender	Age	Level of Vision	# Years	Reading Frequency
P1	F	28	Light perception	Since birth	Couple of times a week
P2	F	58	Totally blind	Since age 22	Couple of times a week
P3	M	55	Totally blind	Since age 36	Once a week
P4	F	22	Light perception	Since birth	Couple of times a week
P5	M	44	Light perception	Since age 31	Once a week
P6	M	56	Totally blind	Since birth	Couple of times a week
P7	F	52	Totally blind	Since birth	Couple of times a week
P8	M	64	Light perception	Since birth	Once a week

Table 2: Participants' demographic information in our user evaluation, which includes their gender, age, level of vision and years at the designated level of vision, and how frequently they read a slide deck.

We selected slide decks that were in English and filtered decks of lengths between 25 and 50 slides for processing purposes. We randomly sampled and performed our pipeline on 500 decks, processing each deck individually. Our model did not learn or improve performance from the process. We inspected 10% of the structure results in terms of the grouping, types, and structure descriptions via a script. We observed that Slide Gestalt was able to create a reasonable structure, especially for slide decks that seemed to be created from a template (which defined layouts of title, section header, and title and body). Since the layout types were not named and labeled consistently per slide, we suggest that data-driven approach is critical to support the scale. We observed that thumbnail quality impacts the results, similar to the finding in Table 1.

To conduct a more profound analysis, future work should consider several aspects. First, it is critical to understand the variety of the slide decks in terms of topics, authoring styles, and presentation goals. This could help identify any patterns or trends in the data and provide insights into the different ways slide decks are used and created. Defining a taxonomy of visual design styles of presentations would be helpful to understand the semantics and the aesthetics variations across the slide decks. Second, it would be valuable to investigate the correlation between the length of a slide deck and its structure at a larger scale, e.g., whether longer decks have more complex or hierarchical structures. Finally, as the visual styles and formats of slide decks vary significantly, it is crucial to assess whether the model could handle these variations and produce accurate structure descriptions. Testing the model on slide decks from different sources or in different languages would also be an important step for future work.

7 USER EVALUATION

To evaluate the usability of Slide Gestalt, we conducted a user study with eight BVI participants to compare our results with a baseline of accessible decks, where we provided a control interface that surfaced on Slide Gestalt's UI without hierarchical information.

7.1 Method

Our one-hour online study with each BVI participant consisted of two parts: (1) a semi-structured interview with the participant to understand their prior experiences reading slides, and (2) a usability study in which the participant read two slide decks we provided with similar themes and structures using two interfaces.

Participants: We recruited eight BVI participants outside our organization (four female, four male, age=22-64) who had prior experiences reading presentation slide decks with a screen reader (five primarily used NVDA, two used JAWS, and one used VoiceOver; see Table 2 for participants' demographic information.) Participants had varying reasons for reading slides: four of them primarily used them for work (with two focused on accessibility-related topics, one on audio production, and one on software engineering), two used for school and educational purposes, and two used for information inquiry, such as for community events and government announcements. Their years of experience reading slides varied from 5 to 11 years. We compensated each participant with a $75 voucher for completing a one-hour study session.

Interview Process: Each study session started with an interview, where we asked participants to share their general strategy of reading a slide deck, the challenges they encountered during the reading and navigation process, and their solutions to overcome any issues. We did not restrict any context or goal of their prior reading experiences to share with us. We asked about their experiences regarding structures in a slide deck and their interpretation of the hierarchies we observed (i.e., a divider, build and topic split slide). We took detailed notes during the study and analyzed the interview by grouping the interview notes into themes, and synthesizing the feedback by themes along with specific quotes.

Usability Study Procedure: After the interview, we conducted a usability testing to evaluate our automatically-generated results. Specifically, we compared two interfaces that presented slide deck information differently: (1) a *control interface* modified from Slide Gestalt's UI, where all the element descriptions (e.g., image alt text) were accessible and presented on a web page with the title header indicating the beginning of each slide (as a stronger baseline than Google Slides, which did not support header navigation), and (2) Slide Gestalt's *hierarchical interface* where participants could access both the structure information of a slide deck and the accessible element descriptions.

We provided a brief tutorial of the two interfaces. In the tutorial, we chose a slide deck of the topic about the relationship between disability and COVID-19. We asked participants to read through the slide deck twice, one with Slide Gestalt's hierarchical interface and one with the control interface. We briefly described the arrangement of the pages and how to navigate through the slides using the arrow key and heading level (supported by all screen readers natively). The deck we chose consisted of multiple sections, subsections, and structures (dividers, build slides, and topic split slides) to allow users to familiarize themselves with the navigation.

Next, we conducted the formal study using the same set of interfaces. We prepared two slide decks: One deck was titled "*Experimental Design*", which included 35 slides with eight sections and described the methodology of a good experimental design by using guidelines of the camera app as an example. The other deck was titled "*The Technique of Evaluation*", which also included 35 slides with eight sections, and depicted a similar topic on various techniques to do an effective evaluation.

Participants reviewed each slide deck using a unique interface – either the control interface or Slide Gestalt's hierarchical interface in a counterbalanced order to reduce the learning effect. For each slide deck, we asked participants to perform a set of tasks. We first asked them to skim through the content given a fixed period of time (three minutes). Then, we asked participants to summarize what they had read and answer two questions with respect to the content of the slides. We allowed participants to seek the relevant information in their familiar way. After each set of tasks, we asked a series of questions in a 7-point Likert scale. The questions included: how helpful the interface supported them understanding the structure and the content of the slide deck, and how helpful the interface supported them navigating and skimming through the slide deck. We also asked about the usefulness of each type of structure description. We concluded the study by asking open-ended questions about their preferences of the two interfaces and the differences, compared to their existing reading tools.

7.2 Feedback on Prior Slide Consumption Experiences

All participants reported that most of the slide deck they read were not completely accessible, at either the element level or the deck level. We summarize the detailed responses:

How did BVI people read a slide deck? All participants reported that they used PowerPoint as the major tool to read slides as they felt it was one of the most screen-reader accessible presentation tools on the market. Participants primarily relied on four techniques to read slide decks, including (ordered by popularity): (1) read the slides in the presenter mode as if playing the "*slideshow*" (which could present content that was not revealed in the editor view, such as animation) (P1, P4, P5), (2) export the slides to the PDF format and use another PDF reader software (Adobe Acrobat) to read the slide decks (P2, P7), (3) export the slides to a text-only format or outline (e.g., Microsoft Word) and only read the text content of the slide deck (P3, P8), and (4) read the slides directly in the editor mode (P6). All participants mentioned that they read slides linearly regardless of the platforms they used.

Slide-level accessibility challenges: Participants described accessibility issues that were also reported in prior work [27], including the lack of alt text and the incorrect element read order. These issues could be due to the lack of author annotations or the missing metadata from format conversion. In terms of the reading mode, P6 expressed that the editor view could be challenging to read, given that they could unintentionally modify the file [55]: "*To be honest, reading slides in editor view is like walking on the street — you'll never know what you encounter, and whether you bump into something that you have no idea what it is. I remember I accidentally deleted elements on slides countless times while reading the deck (I ended up just never saving the file after I read it). The only reason I am still using it is just that I do not remember how to switch to the presenter mode, so gradually I got used to the (editor) interface*" (P6). Despite the challenges, most of the participants expressed that they did not find a viable solution that allowed them to independently overcome the issues. They would ask the presenters to share any relevant text documents if available (e.g., a source course document that covered the information in a slide deck). Alternatively, they would read an accessible section and ignore non-annotated parts in a slide deck.

Deck-level accessibility challenges: Besides the inaccessible slide elements and reading interface, participants mentioned confusing moments that they encountered when reading slides:

"*I can remember there are a couple of times when I read the slides and all of a sudden I feel like I am falling into an infinite loop (with build slide), where I just kept reading the same content again and again. After the fourth time, I just decided to skip the following slide whenever I read the same initial element again*" (P1).

"*In many of the slides (even for the slides I made), people tend to split the contents under the same topic into different slides to avoid clustering all the information scattered in one slide. I understand it makes sense visually, but sometimes I lose the track of some specific topics. I wish I can know where is the end point of the topic and the start point of another topic if this split strategy was used by the authors, especially since not all the titles in each topic split will be exactly the same.*" (P3)

"*I noticed that I have read slides where each of those slides is just a short (title) text. I am not quite sure about the purpose of those slides even though they showed up periodically*" (P4).

Based on participants' feedback, we discovered that while they were able to recognize the repetitive patterns (such as dividers, build and topic split slides) used throughout the slide deck, they had a limited understanding of these difficult-to-navigate structures. These findings were consistent with our earlier observations in Section 3.1. After our explanation on each visual pattern, all participants realized the purposes of those information structures. They unanimously agreed that it would be helpful to be aware of the structures while reading through the slides for them to skim, read, or search within a slide deck in a systematic way.

7.3 Usability Study Results

We used One-way Repeated ANOVA to perform parametric tests and Wilcoxon Signed-Rank test for non-parametric tests to validate the significant level of each comparison (where each subject received both treatments). We report our study findings:

Skimming and searching information: In the skimming stage of our study, participants read significantly more slides using Slide Gestalt's hierarchical interface compared to the control interface ($\mu = 29.3$, $\sigma = 3.1$ vs. $\mu = 19.4$, $\sigma = 2.7$) ($F(1,7) = 127.4; p < 0.0001$). When searching for information, although all participants obtained the correct answers, they read through fewer redundant or repetitive slide elements using the hierarchical interface compared to the control interface ($\mu = 6.5$, $\sigma = 3.3$ vs. $\mu = 20.6$, $\sigma = 7.2$) ($F(1,7) = 66.2; p < 0.0001$). As a result, participants spent significantly less time obtaining the correct answers using the hierarchical interface compared to the control interface ($\mu = 45.8$, $\sigma = 7.1$ vs. $\mu = 96.9$, $\sigma = 8.2$) ($F(1,7) = 721.48; p < 0.0001$).

For the experiences of reading and navigating through the slide information, all participants preferred the hierarchical interface to the control interface, where participants rated the hierarchical interface significantly higher than the control interface for information retrieval and navigation ($\mu = 6.4$, $\sigma = 0.5$ vs. $\mu = 4.4$, $\sigma = 0.9$) ($Z = 2.52$; $p < 0.001$) as well as skimming ($\mu = 6.1$, $\sigma = 0.9$ vs. $\mu = 3.4$, $\sigma = 1.2$) ($Z = 2.48$; $p < 0.001$). When reading the slide deck with the control interface, participants navigated the slides and elements linearly and rarely jumped between the slides. All participants expressed that this linear style was similar to their current reading practice. On the other hand, all participants made use of the section and subsection headers when reading the slides with the hierarchical interface. Five participants (P1, P2, P4, P5, P8) first skimmed through all the sections and then went back to the first section to dive into subsections and other details within each section. Two participants (P3, P7) navigated the first couple of sections (2-3) and then went back to the top to read the rest of the content. P5 read the slides in the most unique way, where he navigated back and forth between different sections and subsections and selectively dove into the details in each section or subsection. P5 explained:

"*This [hierarchical] interface really allows me to go wherever and whenever I would like to. I can decide to skim through the high-level ideas and go deep into the details for specific topics, or I can search for specific terms first and then anchor the specific contents and the relevant groups of information. I think the system provides readers a novel perspective to consume the slide deck such that it makes some of the independent slides connect to each other coherently, just like a book or document, and creates a platform for the audience to read the slide decks based on their goal — either it's just for skimming or for finding and consuming the information*" (P5).

In response to the effort for navigation, participants rated it easier to navigate with the hierarchical interface than the control interface ($\mu = 5.9$, $\sigma = 0.6$ vs. $\mu = 3.9$, $\sigma = 1.4$ in a 7-point Likert scale, where 7 means the reading experience took the least effort) ($Z = 2.31$; $p < 0.05$). They explained that the hierarchical interface organized repetitive information into a more concise and interpretable format. Still, two participants expressed that it might take some time for users to learn and get familiar with the hierarchical

	Objective			Subjective (out of 7, higher numbers showing more positive experiences on the task)				
Interface	Number of Slides Read	Number of Redundant Elements Read	Time to Get Answers	Structure Understanding	General Navigation	Skimming	Content Understanding	Consumption Easiness
Slide Gestalt	29.3	6.5	45.8 seconds	6.5	6.4	6.1	5.8	5.9
Control	19.4	20.6	96.9 seconds	3.8	4.4	3.4	3.9	3.9

Table 3: In the usability study, we compared the performance of two interfaces: Slide Gestalt and Control Interface. We recorded objective metrics, such as the number of slides read in a set amount of time, the number of redundant elements read during navigation, and the time it took to get correct answers. We also asked participants to rate their experience in a 7-point Likert scale, including their understanding of the structure and content, ability to navigate, skim and consume the slide content.

control. All participants agreed that both interfaces allowed better navigation than their current tools as they could use all the commands supported by screen readers natively to read through the slides with heading levels (which is not yet supported by any presentation tool). Overall, all participants found the hierarchical interface supported them in reading the slide decks at various timescales (short-term skimming or long-term searching or reading) and manners (linear or non-linear) flexibly.

Understanding the slide deck and its structure: Participants expressed that the hierarchical interface not only provided them more supports to understand the embedded structure in a slide deck than using the control interface ($\mu = 6.5$, $\sigma = 0.5$ vs. $\mu = 3.0$, $\sigma = 0.5$) ($Z = 2.53$; $p < 0.001$), but also helped them understand the overall content more thoroughly ($\mu = 5.8$, $\sigma = 1.2$ vs. $\mu = 3.9$, $\sigma = 1.3$) ($Z = 2.36$; $p < 0.05$) in a short period of time. P7 commented: "*I think showing the information of main sections and sub-topics helped me understand things beyond structures. To some degree, it allowed me to learn and follow how the authors build the visual-language narratives gradually*" (P7).

Among the three structure descriptions we provided, all participants found the descriptions of the build slides ($\mu = 6.75$, $\sigma = 0.43$) and topic split slides ($\mu = 6.25$, $\sigma = 0.97$) highly useful, since it indicated the clear groupings, the start and end range of a series of relevant slides, and the purpose of those groupings. This helped them comprehend if the slides either conveyed different content under the same topic or were snapshots of a group. In terms of the descriptions of a divider, all participants considered it helpful to show the start index of a section and indicate the purpose of those specific slides, yet three participants (P5, P7, P8) thought the system could filter or hide the description given that it conveyed less information non-visually than visually. Overall, participants agreed that our hierarchical interface enabled them to understand a slide deck at both the high level (structures) and low level (content).

Although participants did not encounter errors in our structure descriptions throughout the study, we asked feedback on computational assistance and potential errors. All participants expressed that they appreciated an automatic approach to improve their reading experiences, and therefore could tolerate system issues. While this work is a first step to surface the hierarchical information of a slide deck, we encourage future research to design interactive tools for slide authors to provide quality annotations of both the structures and detailed content of a slide deck.

8 DISCUSSION AND OPPORTUNITIES

We demonstrated that by identifying the structure in a slide deck, the hierarchical information can support BVI users consuming slides more efficiently and effectively with a screen reader friendly interface. We discuss the limitations and opportunities Slide Gestalt introduces for future research.

Towards quality structure descriptions: Considering different design practices that slide authors could have, to improve the quality of the structure descriptions, future work could expand our current highlight detection methods by extracting visual [18, 50] or semantic [30] salience of the content. A system could also consider outline slides based on a language model or heuristics (e.g., a slide title of "*outline*", "*overview*", or "*agenda*"). We also noted that another authoring practice is to present abstractive content, such as an image slideshow with a few text overlays. To support various design styles, we suggest collecting define-grained structure annotations at a large scale for further investigation.

Supporting mix-initiative authoring for accessible structures: Slide Gestalt improves the non-visual accessibility and usability of a slide deck by generating the hierarchy for BVI readers after a deck is created. Going forward, we suggest designing real-time techniques for slide authors to embed the structure metadata during the authoring phase. Similar to prior research on tools for improving image alt text, one opportunity is to integrate our pipeline with existing slide authoring tools by generating initial structure annotations for authors to edit, which we leave for future work.

Improving reading experiences for sighted readers: While Slide Gestalt is designed for BVI users, we assume that the generated information could also support sighted users to consume a slide deck more efficiently. We had prototyped multiple interfaces to visualize the automatically-extracted hierarchy of a slide deck by augmenting the filmstrip or the grid view of an existing slide authoring tool. We encourage future research to make the slide decks accessible and useful for all audiences across devices beyond the scope of this paper.

Generalization with ethical considerations: Our work structures a slide deck by its text and visual materials. We focus on trusted content to make responsibly-created slide decks more accessible. We acknowledge that our findings are limited to the sample size and the recruitment strategy from a close network. We suggest future studies should consider long-term and diverse user feedback.

9 CONCLUSION

In this paper, we present Slide Gestalt, an automatic approach that identifies the hierarchical structure in a slide deck. This approach uses visual and textual information to group slides into a two-level hierarchy, which allows users to navigate the content via a screen reader. The design of Slide Gestalt is based on interviews with BVI audiences and sighted creators, as well as an analysis of 100 slide decks and accessibility guidelines for digital documents. We tested Slide Gestalt on 50 real-world slide decks and found that it effectively identified hierarchical structures. Feedback from BVI participants indicated that Slide Gestalt helped them navigate slide decks more efficiently and effectively compared to traditional accessible slides.

ACKNOWLEDGMENTS

We thank all the participants in our user studies for their feedback. This work has been possible thanks to the support of people including, but not limited to the following (in alphabetical order of last name): Bea Alessio, Halit Erdogan, Rebecca Hsieh, Lu Jiang, David Salesin, and Lijun Yu.

REFERENCES

[1] Shadi Abou-Zahra and EOWG Participants. 2020. How to Make Your Presentations Accessible to All. https://www.w3.org/WAI/teach-advocate/accessible-presentations/#preparing-slides-and-projected-material-speakers.

[2] Apple. 2022. *Group or ungroup slides in Keynote on Mac - Apple Support.* Apple. Retrieved September, 2022 from https://support.apple.com/guide/keynote/group-or-ungroup-slides-tan98841ef85/mac

[3] Lawrence Bergman, Jie Lu, Ravi Konuru, Julie MacNaught, and Danny Yeh. 2010. Outline Wizard: Presentation Composition and Search. In *Proceedings of the 15th International Conference on Intelligent User Interfaces* (Hong Kong, China) *(IUI '10)*. Association for Computing Machinery, New York, NY, USA, 209–218. https://doi.org/10.1145/1719970.1719999

[4] Jeffrey P. Bigham, Erin L. Brady, Cole Gleason, Anhong Guo, and David A. Shamma. 2016. An Uninteresting Tour Through Why Our Research Papers Aren't Accessible. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 621–631. https://doi.org/10.1145/2851581.2892588

[5] Jeffrey P Bigham, Ryan S Kaminsky, Richard E Ladner, Oscar M Danielsson, and Gordon L Hempton. 2006. WebInSight: making web images accessible. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. 181–188.

[6] Jeffrey P Bigham and Kyle Murray. 2010. WebTrax: visualizing non-visual web interactions. In *International Conference on Computers for Handicapped Persons*. Springer, Springer, New York, NY, 346–353.

[7] Chilli Breeze. 2022. Common Design of PowerPoint. https://www.chillibreeze.com/presentation-design/6-must-have-layout-slides-for-your-business-powerpoint-templates/.

[8] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: Automatic Generation of Step-by-step Mixed Media Tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. ACM, New York, NY, USA, 93–102. https://doi.org/10.1145/2380116.2380130

[9] Pei-Yu Chi, Bongshin Lee, and Steven M. Drucker. 2014. DemoWiz: Re-Performing Software Demonstrations for a Live Presentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1581–1590. https://doi.org/10.1145/2556288.2557254

[10] Steven M Drucker, Georg Petschnigg, and Maneesh Agrawala. 2006. Comparing and managing multiple versions of slide presentations. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 47–56.

[11] Nancy Duarte. 2008. *Slide: ology: The art and science of creating great presentations*. Vol. 1. O'Reilly Media Sebastapol.

[12] Darren Edge, Joan Savage, and Koji Yatani. 2013. HyperSlides: Dynamic Presentation Prototyping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 671–680. https://doi.org/10.1145/2470654.2470749

[13] Darren Edge, Xi Yang, Yasmine Kotturi, Shuoping Wang, Dan Feng, Bongshin Lee, and Steven Drucker. 2016. SlideSpace: Heuristic Design of a Hybrid Presentation Medium. *ACM Trans. Comput.-Hum. Interact.* 23, 3, Article 16 (jun 2016), 30 pages. https://doi.org/10.1145/2898970

[14] Mirette et al. Elias. 2018. SlideWiki: towards a collaborative and accessible platform for slide presentations.

[15] Vocal Eyes. 2018. Making your conference presentation more accessible to blind and partially sighted people. https://vocaleyes.co.uk/services/resources/guidelines-for-making-your-conference-presentation-more-accessible-to-blind-and-partially-sighted-people/.

[16] Raymond Fok, Mingyuan Zhong, Anne Spencer Ross, James Fogarty, and Jacob O. Wobbrock. 2022. A Large-Scale Longitudinal Analysis of Missing Label Accessibility Failures in Android Apps. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 461, 16 pages. https://doi.org/10.1145/3491102.3502143

[17] C. Ailie Fraser, Joy O. Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. 2020. Temporal Segmentation of Creative Live Streams. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376437

[18] Ankit Gandhi, Arijit Biswas, and Om Deshmukh. 2015. Topic Transition in Educational Videos Using Visually Salient Words. *International Educational Data Mining Society* (2015).

[19] Cole Gleason, Amy Pavel, Emma McCamey, Christina Low, Patrick Carrington, Kris M Kitani, and Jeffrey P Bigham. 2020. Twitter A11y: A Browser Extension to Make Twitter Images Accessible. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12.

[20] Lance Good and Benjamin B. Bederson. 2002. Zoomable User Interfaces as a Medium for Slide Show Presentations. *Information Visualization* 1, 1 (mar 2002), 35–49. https://doi.org/10.1057/palgrave/ivs/9500004

[21] Google. 2022. *Google Slides API | Google Developers*. Google. Retrieved September, 2022 from https://developers.google.com/slides/api

[22] Google. 2022. *Google Slides: Online Slideshow Maker | Google Workspace*. Google. Retrieved September, 2022 from https://www.google.com/slides/about/

[23] Google. 2022. *Vision AI | Derive Image Insights via ML | Cloud Vision API | Google Cloud*. Google. Retrieved September, 2022 from https://cloud.google.com/vision

[24] Tessai Hayama, Hidetsugu Nanba, and Susumu Kunifuji. 2008. Structure Extraction from Presentation Slide Information. In *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence* (Hanoi, Vietnam) *(PRICAI '08)*. Springer-Verlag, Berlin, Heidelberg, 678–687. https://doi.org/10.1007/978-3-540-89197-0_62

[25] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. 2000. Comparing Presentation Summaries: Slides vs. Reading vs. Listening. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 177–184. https://doi.org/10.1145/332040.332427

[26] Mina Huh, Saelyn Yang, Yi-Hao Peng, Xiang Chen, Young-Ho Kim, and Amy Pavel. 2023. AVscript: Accessible Video Editing with Audio-Visual Scripts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.

[27] Tatsuya Ishihara, Hironobu Takagi, Takashi Itoh, and Chieko Asakawa. 2006. Analyzing visual layout for a non-visual presentation-document interface. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. 165–172.

[28] Jeff A. Johnson and Bonnie A. Nardi. 1996. Creating Presentation Slides: A Study of User Preferences for Task-Specific versus Generic Application Software. *ACM Trans. Comput.-Hum. Interact.* 3, 1 (mar 1996), 38–65. https://doi.org/10.1145/226159.226161

[29] Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, Yi-Ting Chen, Yaxi Gao, Tom Duerig, Andrew Tomkins, and Sujith Ravi. 2019. Graph-rise: Graph-regularized image semantic embedding. *arXiv preprint arXiv:1902.10814* (2019).

[30] Manish Kanadje, Zachary Miller, Anurag Agarwal, Roger Gaborski, Richard Zanibbi, and Stephanie Ludi. 2016. Assisted keyword indexing for lecture videos using unsupervised keyword spotting. *Pattern Recognition Letters* 71 (2016), 8–15.

[31] Shinya Kawanaka, Yevgen Borodin, Jeffrey P. Bigham, Darren Lunn, Hironobu Takagi, and Chieko Asakawa. 2008. Accessibility Commons: A Metadata Infrastructure for Web Accessibility. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility* (Halifax, Nova Scotia, Canada) *(Assets '08)*. Association for Computing Machinery, New York, NY, USA, 153–160. https://doi.org/10.1145/1414471.1414500

[32] Pin-Sung Ku, Yu-Chih Lin, Yi-Hao Peng, and Mike Y Chen. 2019. PeriText: Utilizing peripheral vision for reading text on augmented reality smart glasses. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 630–635.

[33] Richard E Ladner and Kyle Rector. 2017. Making your presentation accessible. *interactions* 24, 4 (2017), 56–59.

[34] Jaewook Lee, Jaylin Herskovitz, Yi-Hao Peng, and Anhong Guo. 2022. ImageExplorer: Multi-Layered Touch Exploration to Encourage Skepticism Towards Imperfect AI-Generated Image Captions. In *CHI Conference on Human Factors in Computing Systems*. 1–15.

[35] Jaewook Lee, Yi-Hao Peng, Jaylin Herskovitz, and Anhong Guo. 2021. Image Explorer: Multi-Layered Touch Exploration to Make Images Accessible. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–4.

[36] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022. Dit: Self-supervised pre-training for document image transformer. *arXiv preprint arXiv:2203.02378* (2022).

[37] Yang Li, James A. Landay, Zhiwei Guan, Xiangshi Ren, and Guozhong Dai. 2003. Sketching Informal Presentations. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (Vancouver, British Columbia, Canada) *(ICMI '03)*. Association for Computing Machinery, New York, NY, USA, 234–241. https://doi.org/10.1145/958432.958476

[38] Microsoft. 2022. *Organize slides into sections*. Microsoft. Retrieved September, 2022 from https://support.microsoft.com/en-us/office/organize-your-powerpoint-slides-into-sections-de4bf162-e9cc-4f58-b64a-7ab09443b9f8

[39] Microsoft. 2022. PowerPoint Accessibility. https://support.microsoft.com/en-us/office/make-your-powerpoint-presentations-accessible-to-people-with-disabilities-6f7772b2-2f33-4bd2-8ca7-dae3b2b3ef25.

[40] Sheshera Mysore, Arman Cohan, and Tom Hope. 2021. Multi-Vector Models with Textual Guidance for Fine-Grained Scientific Document Similarity. *arXiv preprint arXiv:2111.08366* (2021).

[41] Rosiana Natalie, Jolene Loh, Huei Suen Tan, Joshua Tseng, Ian Luke Yi-ren Chan, Ebrima H Jarjue, Hernisa Kacorri, and Kotaro Hara. 2021. The efficacy of collaborative authoring of video scene descriptions. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–15.

[42] Bureau of Internet Accessibility. 2018. Tips for Making Your Presentations Accessible. https://www.boia.org/blog/tips-for-making-your-presentations-accessible.

[43] Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. In *Proc. UIST'15*. ACM, 181–190.

[44] Amy Pavel, Gabriel Reyes, and Jeffrey P. Bigham. 2020. Rescribe: Authoring and Automatically Editing Audio Descriptions. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 747–759.

https://doi.org/10.1145/3379337.3415864

[45] Yi-Hao Peng, Jeffrey P Bigham, and Amy Pavel. 2021. Slidecho: Flexible Non-Visual Exploration of Presentation Videos. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–12.

[46] Yi-Hao Peng, Ming-Wei Hsi, Paul Taele, Ting-Yu Lin, Po-En Lai, Leon Hsu, Tzuchuan Chen, Te-Yen Wu, Yu-An Chen, Hsien-Hui Tang, et al. 2018. Speechbubbles: Enhancing captioning experiences for deaf and hard-of-hearing people in group conversations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–10.

[47] Yi-Hao Peng, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. 2021. Say It All: Feedback for Improving Non-Visual Presentation Accessibility. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[48] Yi-Hao Peng, Muh-Tarng Lin, Yi Chen, TzuChuan Chen, Pin Sung Ku, Paul Taele, Chin Guan Lim, and Mike Y Chen. 2019. PersonalTouch: Improving touchscreen usability by personalizing accessibility settings based on individual user's touchscreen interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[49] Yi-Hao Peng, Jason Wu, Jeffrey P. Bigham, and Amy Pavel. 2022. Diffscriber: Describing Visual Design Changes to Support Mixed-ability Collaborative Presentation Authoring. In *Proceedings of the 35rd Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[50] M. R. Rahman, S. Shah, and J. Subhlok. 2020. Visual Summarization of Lecture Video Segments for Enhanced Navigation. In *2020 IEEE International Symposium on Multimedia (ISM)*. 154–157. https://doi.org/10.1109/ISM.2020.00033

[51] Garr Reynolds. 2011. *Presentation Zen: Simple ideas on presentation design and delivery*. New Riders, Indianapolis, IN, USA.

[52] Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728* (2020).

[53] Daisuke Sato, Masatomo Kobayashi, Hironobu Takagi, and Chieko Asakawa. 2009. What's next? a visual editor for correcting reading order. In *IFIP Conference on Human-Computer Interaction*. Springer, Springer, New York, NY, USA, 364–377.

[54] Daisuke Sato, Hironobu Takagi, and Chieko Asakawa. 2006. Accessibility evaluation based on machine learning technique. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. 253–254.

[55] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O Wobbrock. 2021. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–19.

[56] Athar Sefid, Prasenjit Mitra, and Lee Giles. 2021. SlideGen: An Abstractive Section-Based Slide Generator for Scholarly Documents. In *Proceedings of the 21st ACM Symposium on Document Engineering* (Limerick, Ireland) *(DocEng '21)*. Association for Computing Machinery, New York, NY, USA, Article 11, 4 pages. https://doi.org/10.1145/3469096.3474939

[57] Moushumi Sharmin, Lawrence Bergman, Jie Lu, and Ravi Konuru. 2012. On Slide-Based Contextual Cues for Presentation Reuse. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces* (Lisbon, Portugal) *(IUI '12)*. Association for Computing Machinery, New York, NY, USA, 129–138. https://doi.org/10.1145/2166966.2166992

[58] Ryan Spicer, Yu-Ru Lin, Aisling Kelliher, and Hari Sundaram. 2012. NextSlidePlease: Authoring and Delivering Agile Multimedia Presentations. *ACM Trans. Multimedia Comput. Commun. Appl.* 8, 4, Article 53 (nov 2012), 20 pages. https://doi.org/10.1145/2379790.2379795

[59] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. 2021. Automatic Generation of Two-Level Hierarchical Tutorials from Instructional Makeup Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 108, 16 pages. https://doi.org/10.1145/3411764.3445721

[60] Utku Uckun, Ali Selman Aydin, Vikas Ashok, and IV Ramakrishnan. 2020. Breaking the accessibility barrier in non-visual interaction with pdf forms. *Proceedings of the ACM on Human-computer Interaction* 4, EICS (2020), 1–16.

[61] Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.

[62] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. 2016. Automatic Generation of Visual-Textual Presentation Layout. *ACM Trans. Multimedia Comput. Commun. Appl.* 12, 2, Article 33 (feb 2016), 22 pages. https://doi.org/10.1145/2818709

[63] Chengbo Zheng, Dakuo Wang, April Yi Wang, and Xiaojuan Ma. 2022. Telling Stories from Computational Notebooks: AI-Assisted Presentation Slides Creation for Presenting Data Science Work. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 53, 20 pages. https://doi.org/10.1145/3491102.3517615

[64] Yu Zhong, T. V. Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P. Bigham. 2014. JustSpeak: Enabling Universal Voice Control on Android. In *Proceedings of the 11th Web for All Conference* (Seoul, Korea) *(W4A '14)*. Association for Computing Machinery, New York, NY, USA, Article 36, 4 pages. https://doi.org/10.1145/2596695.2596720

[65] Douglas E. Zongker and David H. Salesin. 2003. On Creating Animated Presentations. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 298–308.