FlexDoc: Flexible Document Adaptation through Optimizing both Content and Layout

Yue Jiang*, Christof Lutteroth†, Rajiv Jain‡, Christopher Tensmeyer‡,
Varun Manjunatha‡, Wolfgang Stuerzlinger§, Vlad I. Morariu‡
**Aalto University, Espoo, Finland, yue.jiang@aalto.fi*
†*University of Bath, Bath, United Kingdom, cl2073@bath.ac.uk*
‡*Adobe Research, College Park, United States, {rajijain, tensmeye, vmanjuna, morariu}@adobe.com*
§*Simon Fraser University, Vancouver, Canada, w.s@sfu.ca*

Abstract—**Designing adaptive documents that are visually appealing across various devices and for diverse viewers is a challenging task. This is due to the wide variety of devices and different viewer requirements and preferences. Alterations to a document's content, style, or layout often necessitate numerous adjustments, potentially leading to a complete layout redesign. We introduce FLEXDOC, a framework for creating and consuming documents that seamlessly adapt to different devices, author, and viewer preferences and interactions. It eliminates the need to manually create multiple document layouts, as FLEXDOC enables authors to define desired document properties using templates and employs both discrete and continuous optimization in a novel comprehensive optimization process, which leverages automatic text summarization and image carving techniques to adapt both layout and content during consumption dynamically. Further, we demonstrate FLEXDOC in real-world scenarios.**

I. INTRODUCTION

Document layout is important for effective information consumption in applications ranging from print media to digital forms such as web pages and interactive news readers. As device variety increases, a single document needs to adapt to different screen sizes, orientations, and aspect ratios. This variety also increases the effort for document authors and personalization for individual viewers.

Existing commercial document creation tools have limitations in generating adaptive documents (Table I). These arise from system designs being focused on layout capabilities, an author's capabilities, such as being able to code, and a viewer's options during consumption. From a system perspective, existing tools necessitate manually defined breakpoints or coding to create multiple versions. For instance, Adobe Acrobat Liquid Mode reflows documents to a single-column format but does not generate multi-column formats, and images on smaller screens shrink instead of reflowing. Similarly, CSS struggles with adaptive multi-column layouts and requires specific configurations, such as a fluid grid, to facilitate image flows. Authors using these tools typically need to code and/or manually set breakpoints for diverse layout sizes. Tools like Webflow eliminate coding but still require authors to set breakpoints and lack support for image flows.

Prior work proposed optimization for generating responsive documents without coding or breakpoints. Table I lists the most related work. Some work [1]–[6] used pre-defined alternatives for optimization requiring authors to craft multiple versions of the same content. Such manual specification makes it hard to deal with a whole collection of documents and lacks content personalization; once the document design is finalized, the content remains fixed. Laine et al. [7] enabled some personalization by allowing viewers to select important elements displayed in a larger format, but their approach could not alter the layout structure, only adjusting element sizes.

We propose FLEXDOC, a novel approach for dynamically adapting documents to different screen sizes, and author and viewer preferences. Combining discrete and continuous optimization, FLEXDOC creates documents with flexible layouts and adaptive content. Our system applies image and natural language processing techniques to automatically generate content variations, such as images of varying sizes and aspect ratios and text summaries of varying lengths. For authors, FLEXDOC offers flexible templates editable interactively without coding or specifying breakpoints. For viewers, FLEXDOC enables interactive adaptation of document layout and content for optimal consumption. We evaluate FLEXDOC with the following research questions: *RQ1) Do viewers benefit from interactive adapted content at viewing time according to their preferences?* and *RQ2) Do authors benefit from document adaptations with layout and content alternatives?* Our findings show that document authors can use FLEXDOC to edit layout templates while maintaining readability, and viewers benefit from dynamic documents adapting to their preferences. We present the following main contributions:

1) A novel method for generating and optimizing dynamic content and layout to interactively adapt a document to various devices, author preferences, and viewer preferences. Both authors and viewers can influence the layout and content shown in a document.
2) An optimization approach that combines both discrete and continuous optimization of global properties, such as layout and aesthetics, and local properties, such as information loss, content preferences, and interactive adjustments in the level of detail.
3) A demonstration within application scenarios, showing that FLEXDOC supports an immersive and interactive

*This work was done in part while the first author was an intern at Adobe.

	Functionality	FLEXDOC	Acrobat Liquid	CSS	Webflow
System	Adaptive content	✓	✗	✗	✗
	Image flow	✓	✗	✶	✗
	Adaptive multi-column	✓	✗	✗	✓
Author	No breakpoints required	✓	✓	✗	✗
	No Need to Code	✓	✓	✗	✓
Viewer	Viewer preferences	✓	✗	✗	✗

TABLE I: Comparative analysis of FLEXDOC and existing commercial document tools regarding document adaptation capabilities from system, author, and viewer perspectives. '✓' and '✗' indicate the presence and absence of functionality. '✶' denotes functionality achievable through coding.

	Functionality	FLEXDOC	O'Donovan et al.	Borning et al.	Laine et al.
Author	No need to create all alternatives manually	✓	✗	✗	✓
	No need to modify low-level constraints	✓	✓	✗	✓
	No image distortion	✓	✓	✓	✗
Viewer	Viewer preferences	✓	✗	✓	✓
	No need to modify low-level constraints	✓	✓	✗	✓
	Layout modification	✓	✗	✗	✗

TABLE II: Comparative analysis of FLEXDOC and existing document tools from prior research regarding their ability to adapt documents to various authors' and viewers' preferences. '✓' and '✗' indicate the presence and absence of functionality.

approach for reading documents.

II. Document Optimization

Designing an adaptive document involves optimizing content and layout to fit screen properties, author preferences, and viewer preferences. This requires numerous element-specific and layout-related decisions. To realize such functionality, we formulate the document optimization problem as a joint discrete and continuous optimization process. Implementation details are available in the supplementary materials.

A. Problem Formulation

We define the document problem as an optimization problem, where we decide on the positions and sizes of document elements, denoted as $e_i = (x_i, y_i, w_i, h_i)$. Here, x_i and y_i represent the coordinates of the top-left corner of the i-th element, and w_i and h_i represent its width and height, respectively. We focus on rectangular or rectangular bounding box elements without considering hierarchies.

We now define a continuous loss term $\mathcal{L}_{\text{cont}}$ and the discrete loss term $\mathcal{L}_{\text{disc}}$. The continuous loss term focuses on screen and element properties, such as element sizes and overall aesthetics; the discrete loss term focuses on author and viewer preferences. The overall objective function is as follows:

$$\begin{aligned} &\mathcal{L}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}, \mathbf{W}_{\text{disc}}) \\ &= \mathcal{L}_{\text{cont}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}) \\ &+ \mathcal{L}_{\text{disc}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N; \mathbf{W}_{\text{disc}}), \end{aligned} \tag{1}$$

where N denotes the total number of elements, $\hat{e}_i = (\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ represent the predicted position and size of each GUI element, and $e_{pi} = (x_{pi}, y_{pi}, w_{pi}, h_{pi})$ denote the preferred element positions and sizes. $\mathbf{W}_{\text{cont}}$ is a set of weights assigned to specific continuous properties, and $\mathbf{W}_{\text{disc}}$ consists of weights associated with discrete properties. We then minimize the objective function to optimize the positions and sizes of document elements:

$$\begin{aligned} \{\hat{e}_1, \hat{e}_2, ..., \hat{e}_N\}^* = \text{argmin}_{\{\hat{e}_1, \hat{e}_2, ..., \hat{e}_N\}} \\ \mathcal{L}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N; \mathbf{W}_{\text{cont}}, \mathbf{W}_{\text{disc}}). \end{aligned} \tag{2}$$

B. Continuous Loss Term

The continuous loss term $\mathcal{L}_{\text{cont}}$ specifies the relationship between elements and their properties. Here, we include image loss $\mathcal{L}_{\text{img}}$, text loss $\mathcal{L}_{\text{text}}$, and alignment loss $\mathcal{L}_{\text{align}}$:

$$\begin{aligned} &\mathcal{L}_{\text{cont}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}) \\ &= \mathbf{w}_{\text{img}}\mathcal{L}_{\text{img}} + \mathbf{w}_{\text{text}}\mathcal{L}_{\text{text}} + \mathbf{w}_{\text{align}}\mathcal{L}_{\text{align}}, \end{aligned} \tag{3}$$

where $\mathbf{W}_{\text{cont}} = \{\mathbf{w}_{\text{img}}, \mathbf{w}_{\text{text}}, \mathbf{w}_{\text{align}}\}$ are the weights for images, texts, and alignments, currently set to 1.

1) Image Loss: To optimize an image, we penalize deviations from its preferred size (size loss) and aspect ratio (aspect ratio loss). Directly using the difference in image *area* is not advisable since it can significantly distort the image.

2) Text Loss: We penalize text that is too small to read by considering its size deficit, i.e., by how much its font size $\hat{f}_i$ is smaller than the viewer's preferred font size f_{pi}. If the text size exceeds the preferred font size, the size deficit is 0. Our system can dynamically generate shortened versions of text to better fit the document. In such cases, we further penalize text changes if the shortened version is used.

3) Alignment Loss: We use a measure of the overall aesthetic of a layout based on established visual principles [8]. This overall aesthetics loss term could be easily extended to consider additional aesthetic principles.

C. Discrete Loss Term

The discrete loss term $\mathcal{L}_{\text{disc}}$ involves the selection of templates and individual content alternatives. For each element e_i, if the viewer has no specific preference, the discrete loss for this element is determined by the author preference loss, $\mathcal{L}_{\text{author},i}$. When the viewer indicates their preferences without interacting directly, the discrete loss shifts to the viewer preference loss, $\mathcal{L}_{\text{viewer},i}$. However, if the viewer actively interacts with the content, the discrete loss is governed by the viewer interaction loss, $\mathcal{L}_{\text{int},i}$, ensuring that the content dynamically adjusts to their direct input.

$$\begin{aligned} &\mathcal{L}_{\text{disc}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N; \mathbf{W}_{\text{disc}}) \\ &= \sum_i \mathbf{w}_{\text{author},i}\mathcal{L}_{\text{author},i} + \mathbf{w}_{\text{viewer},i}\mathcal{L}_{\text{viewer},i} + \mathbf{w}_{\text{int},i}\mathcal{L}_{\text{int},i}, \end{aligned} \tag{4}$$

where one of $\{\mathbf{w}_{\text{author},i}, \mathbf{w}_{\text{viewer},i}, \mathbf{w}_{\text{int},i}\}$ is 1 and the other two are 0, depending on whether the viewer sets their preferences or interacts with the content.

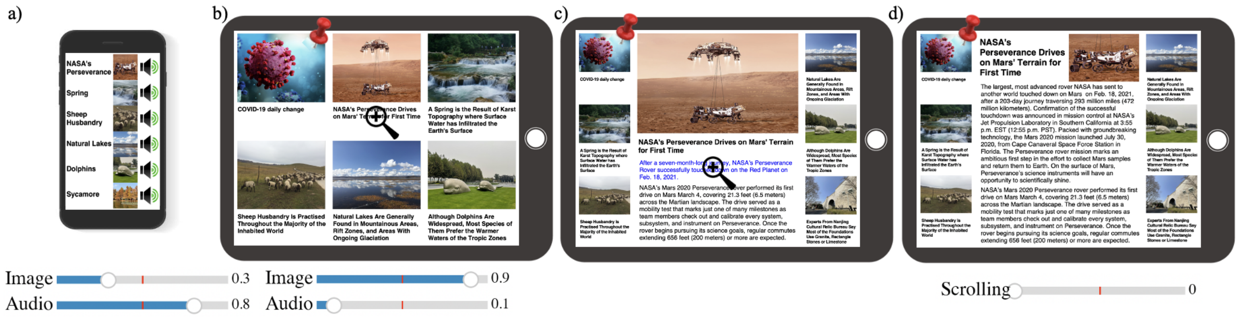


Fig. 1: a) FLEXDOC adapts a news page on a mobile phone to provide a compact overview with quick access to audio content for each article, based on viewer preferences (sliders below images), which prioritize audio content over images and text here. b) The same news page adapted for a tablet device with a user preference for image content. c) As the viewer 'pins' the COVID article and 'zooms in' on the Mars article, FLEXDOC rearranges the layout accordingly, keeping the pinned article in place. d) As the viewer 'zooms in' on the blue text paragraph in the previous image with a preference for avoiding scrolling, FLEXDOC extends the paragraph to provide more details and crops the top image, avoiding the need for scrolling[1].

1) Author Preference Loss: Document authors can define alternatives for both layout templates and content, each assigned preference ranks. Higher loss values are assigned to lower-ranked templates. Specifically, the m-th ranked template is assigned a loss value of $-1000 \cdot (M + 1 - m)$, prioritizing more preferred templates, where M is the number of template alternatives. This approach creates a gradient of loss values across ranks, allowing for optimization within a template before transitioning to another.

2) Viewer Preferences: Viewer preferences have higher priorities than those specified by authors, as the end goal of our approach is to enhance the viewing experience. As shown in Figure 1, viewers can adjust their preferences with the sliders. For example, if the viewer increases their preference for "images" using the corresponding slider, the loss value of all other alternatives will be decreased so that images are more likely to be chosen.

3) Viewer Interactions: FLEXDOC can dynamically change the screen's content in response to viewer interactions. Viewer interactions are given the highest priority since they represent direct requests from the user. Thus, if the viewer chooses a specific template or content alternative through an interaction, that alternative must be selected unless no solution exists. Other contents are then optimized accordingly.

D. Dynamic Content Generation

Given a screen/window size, we optimize the positions and sizes of elements and alternative selection. However, fitting content into a layout is often challenging, especially for smaller screen sizes. To accommodate the diversity of screen sizes and document author and viewer preferences, FLEXDOC dynamically selects or generates alternative content. It applies seam carving for image adaptation and BERT-based text summarization for variable-sized texts. It then optimizes across the potential alternatives with the given screen size and viewer preferences. Further details are in the supplementary materials.

[1]Image credits: Production Perig/stock.adobe.com and NASA/JPL-Caltech

III. DOCUMENT AUTHORING AND VIEWING

FLEXDOC optimizes the document based on the screen properties, author preferences, and viewer preferences. Authors can define their content preferences using FLEXDOCEDITOR, a graphical document template editor. This editor allows *authors* to guide the optimization process by providing different layout templates, content alternatives, and preference rankings. Subsequently, the document can then be optimized based on the screen size and author preferences. On the other hand, *viewers* can also adapt a document by selecting different layout templates or adjusting their preferences via simple operations. Once the screen size changes and/or the viewer changes their preferences, the document can adapt accordingly. More details are in the supplementary materials.

IV. APPLICATIONS

We demonstrate FLEXDOC in multiple real-world application scenarios. Here, we show an example of news reading. Other examples are in the supplementary materials.

Viewers have different preferences for news consumption, based on personal interests and desired levels of detail. Modern news websites use location and browsing history to recommend and preview news items on the front page. These previews typically highlight critical information within a concise format, without comprehensive details and background context. Further, individual news items often mention only the latest events without reference to previous news messages in the series or background information. This requires viewers who want more detail or background information to search for related documents or follow links provided in the document they are reading. Instead of redirecting to other documents, viewers could be better served by extending the document (using content from related articles, not AI-generated ones) they are reading based on their needs, generating more detailed information within the document.

1) Front Page Optimization: News front pages constantly update with the latest news, which can replace older items and

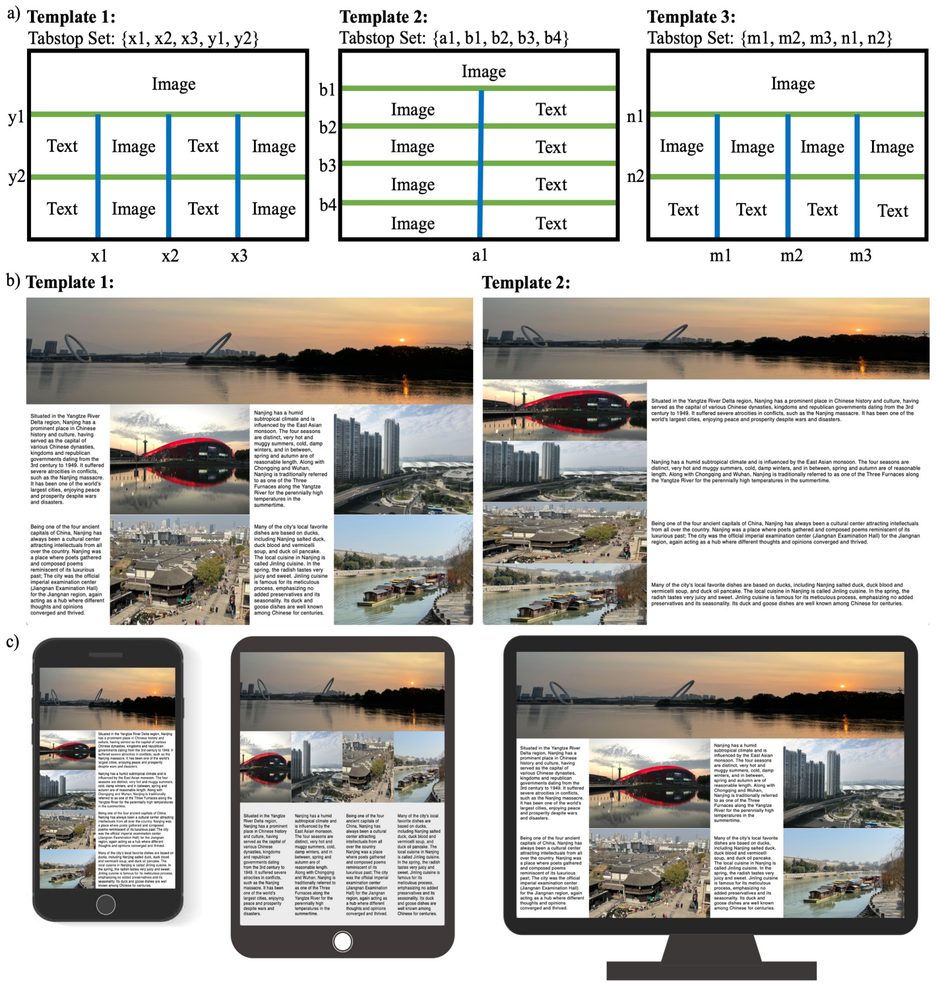


Fig. 2: Document optimization results: a) The author defines three templates. FLEXDOC optimizes tabstop positions based on the objective function. b) Document results when the viewer prefers the first or second template, respectively. c) Results on different devices, which balance both layout structure and the amount of content.

give viewers access to previous articles. Allowing viewers to "pin" their interests, like stock market or COVID-19 news, with FLEXDOC enables continuous access to related updates.

FLEXDOC enables viewers to adapt a news front page to their individual preferences (Figure 1a and b). Viewers can choose preferred modalities and detail levels, and 'pin' items of interest in place. This approach gives freedom to both authors and viewers, influencing the final content and layout. Authors benefit from FLEXDOC automating much of the document adaptation work. Authors predefine layout and content alternatives, allowing viewers to finalize choices. Viewers can adjust reading preferences interactively, like 'zooming in' on specific news without losing the overall context of the news front page (Figure 1c).

2) Dynamic Documents: FLEXDOC aims to generate different aesthetically pleasing document alternatives automatically and dynamically based on a viewer's personal preferences. Unlike news recommendation websites like Google News, which essentially only reorder news items based on viewer interests, FLEXDOC enhances the reading experience by dynamically extending and shortening document content on demand. For instance, a viewer can 'zoom in' on a part of an article to get more detail, and FLEXDOC then automatically re-optimizes the layout to accommodate this extra detail (Figure 1d).

V. EVALUATION

To understand the benefits and challenges of FLEXDOC, we examine how different users might use FLEXDOC from both author and viewer perspectives.

1) Participants: We interviewed 13 interface designers (6M, 7F) including 10 *Professional Designers* with over 2 years of professional UI/UX design experience in industry or research labs; and 3 *Non-Professional Designers* who are HCI graduate students with some interface prototyping experience. Five participants were interviewed in person, and the others were interviewed remotely through video conferencing.

2) Materials: Participants used laptops for authoring documents with FLEXDOCEDITOR and for viewing them.

3) Experiment Design: The study used a within-subject design, requiring participants to compare the use of FLEXDOC and the existing document tools they normally use.

4) Procedure: After explaining the basic ideas of FLEXDOC, participants could experience FLEXDOC from both the author's and the viewer's perspectives through three tasks: a) Participants used the FLEXDOCEDITOR to create a news website resembling Figure 1. They then interacted with the created news website and compared it with their experience using other news websites like Google News. b) Participants used FLEXDOCEDITOR to create their own document templates and assigned content, observing how FLEXDOC adapts these documents to different devices and viewer-preferred templates. They then compared FLEXDOC with their usual design tools for document creation. c) Focusing on the viewer's perspective, participants experienced how FLEXDOC adapts a draft of the FLEXDOC paper to different devices and formats. They compared this with their previous experience of reading papers or similar documents on different devices.

5) Findings: We perform qualitative analysis from both the author's and viewer's perspectives.

RQ1: Do viewers benefit from interactive adapted content at viewing time according to their preferences?

Viewers benefit from the dynamic content generation. Participants were most excited by the dynamic level of detail provided, which facilitated easier access to desired content. (*"It happened a lot that right after I opened the news article, I realized that I am not interested []. I really like the idea of showing [something like an] abstract before opening the article."* (P1), *"I can get the suitable amount of information I need. It will save me a lot of time."* (P6), *"see the summary and being able to hop around"* (P13)).

Flexibility to adapt to different devices and viewer preferences. The second-most mentioned advantage of FLEXDOC was its ability to adjust the document layout to different devices (*"Everything becomes almost unreadable on my phone. This image reflow function solves the problem."* (P6), *"Google News is not optimized very well for mobile devices"* (P3)). In summary, all participants identified tangible benefits due to their flexible reading experience with FLEXDOC.

RQ2: Do authors benefit from document adaptations with layout and content alternatives?

Reduced design effort to create responsive documents. Most participants emphasized that designing content for a variety of formats and sizes is a common requirement today (*"In the future of digital publishing authors cannot account for all the screen sizes they need to account for, and designers are pressured to produce content for more and more different surfaces"* (P11)). Some participants noted that FLEXDOCEDITOR offers greater capabilities compared to their current design tools, particularly in providing flexible layouts for different devices (*"XD does this in a simplistic way, which is more about devices... [*FLEXDOC*] is much more powerful"* (P11)). Despite the added authoring complexity, participants found FLEXDOCEDITOR fairly easy to use (*"For people who are not a software developer, the template creation looks intuitive and easy to do."* (P5)). They appreciated how it allowed them to work at a higher level of abstraction in designing document layouts(*"Designers often cannot think about the overall layout. Designers mostly focus on the component perspective and often ignore the overall layout. This system fills [] this gap."* (P4), *"don't need to get to be too specific about something like alignment. I think alignment is generally time-consuming to deal with ... spent so much time on those small issues. So I think this system can help them avoid those issues. It is good for overall responsive, adaptive layout creation."* (P6)). Furthermore, many participants recognized that templates could save them time: *"I think this kind of templates help me solve the alignment issue. I think it can significantly reduce the design effort."* (P4), *"I hope to have some predefined templates so that I don't need to think about how to design the template myself."* (P1), and *"The general idea of using templates is cool and very convenient."* (P2). This overall positive reception indicates that participants recognized the benefits that FLEXDOC can provide for document authors and that they valued the ease of creating adaptive content.

Lack of functionality to preview resize behaviors. Some participants noted that while it was fairly easy to create flexible documents, it was less straightforward to understand their layout resize behavior across the many possible sizes (*"from the designer's point of view you almost need a simulator to show me what my design is going to look like...you can't show me everything...the challenges is do the designers have the ability to preview the results"* (P11)).

Template editing can be challenging for less experienced designers. Some participants mentioned that editing FLEXDOC templates might require technical expertise that not every designer had (*"would probably be intimidating for an everyday commonplace user, so being able to make it less technical looking might help"* (P12), *"There are many different layout problems that can come up with this, so just the interactions with those decisions might require more testing"* (P11)).

VI. DISCUSSION AND FUTURE WORK

Our FLEXDOC approach dynamically optimizes document structure and content to adapt to various devices and user preferences. Applicable to a wide range of document-centric applications, FLEXDOC enhances both reading and authoring experiences. Designers can use FLEXDOCEDITOR to create flexible layouts that ensure readability across different use cases. FLEXDOC generates suitable versions of images and text to fit these layouts and can adapt UIs interactively in under half a second on an Intel i5 laptop. It can be applied to any PDF or webpage by detecting element types and bounding boxes using document object detection methods [9].

FLEXDOC does not currently consider semantic relationships or hierarchy among document elements, nor handles elements with irregular boundaries. Future work could extend to elements with irregular boundaries and optimize based on document semantics. Additionally, the lack of standard metrics for evaluating adaptive UIs makes quantitative comparison difficult. Future work can establish such metrics.

REFERENCES

[1] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Learning layouts for single-pagegraphic designs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, pp. 1200–1213, 2014.

[2] P. O'Donovan, A. Agarwala, and A. Hertzmann, "Designscape: Design with interactive layout suggestions," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), p. 1221–1224, Association for Computing Machinery, 2015.

[3] C. Domshlak, R. I. Brafman, and S. E. Shimony, "Preference-based configuration of web page content," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, (San Francisco, CA, USA), p. 1451–1456, Morgan Kaufmann Publishers Inc., 2001.

[4] E. Marcotte, *Responsive Web Design*. A book apart, 2011.

[5] M. Nebeling and M. C. Norrie, "Responsive design and development: Methods, technologies and current issues," in *Proceedings of the 13th International Conference on Web Engineering*, ICWE'13, (Berlin, Heidelberg), p. 510–513, Springer-Verlag, 2013.

[6] C. Jacobs, W. Li, and D. H. Salesin, "Adaptive document layout via manifold content," in *Proceedings of Workshop on Web Document Analysis*, pp. 1–4, 2003.

[7] M. Laine, Y. Zhang, S. Santala, J. P. P. Jokinen, and A. Oulasvirta, "Responsive and personalized web layouts with integer programming," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, May 2021.

[8] D. Schölgens, S. Müller, C. Bauer, R. Tilly, and D. Schoder, "Aesthetic measures for document layouts: Operationalization and analysis in the context of marketing brochures," in *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, (New York, NY, USA), p. 21–30, Association for Computing Machinery, 2016.

[9] K. Li, C. Wigington, C. Tensmeyer, H. Zhao, N. Barmpalios, V. I. Morariu, V. Manjunatha, T. Sun, and Y. Fu, "Cross-domain document object detection: Benchmark suite and method," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12915–12924, 2020.

FlexDoc: Flexible Document Adaptation through Optimizing both Content and Layout (Supplementary Materials)

Yue Jiang*, Christof Lutteroth†, Rajiv Jain‡, Christopher Tensmeyer‡,
Varun Manjunatha‡, Wolfgang Stuerzlinger§, Vlad I. Morariu‡
*Aalto University, Espoo, Finland, yue.jiang@aalto.fi
†University of Bath, Bath, United Kingdom, cl2073@bath.ac.uk
‡Adobe Research, College Park, United States, {rajijain, tensmeye, vmanjuna, morariu}@adobe.com
§Simon Fraser University, Vancouver, Canada, w.s@sfu.ca

***Abstract*—In these supplementary materials, we review related work, mention implementation details of the document optimization process, dynamic content generation, and the document authoring and viewing interfaces. We also present applications, use cases, and qualitative comparisons to previous methods. Additionally, we discuss the limitations and future work in more detail.**

I. Related Work

A. Layout Alternatives and Customized Layout Generation

Previous work has proposed optimization-based approaches for customized layout generation to improve the viewer experience across different screen sizes and viewer requirements [1]–[8]. SUPPLE [9], [10] automatically optimized user interfaces by applying alternative widgets or groupings to accommodate screen-size constraints and customize user interfaces for people with disabilities [11]. Personalization was further improved by specifying a cost function to meet users' preferences and target devices [12] and maintaining consistency [13]. Arnauld [14] generated optimized parameters for layouts based on a cost function, resulting in more optimal document layouts. Recent work has explored layout design using program synthesis techniques. Scout [15] enabled designers to explore alternatives and receive design feedback by generating potential layouts based on user-provided high-level constraints. Yet, Scout only provided fixed-size layout suggestions without dynamic resize behaviors nor guaranteed diverse results. InferUI [16] inferred constraints to describe a layout from layout examples, but only with linear constraints expressing relative mutual alignments of widgets and a single topological arrangement. It could not handle dynamic topological layouts such as flows and alternative positions. In contrast, FlexDoc can handle layouts containing textflows with proper resize behaviors and dynamic topology and can adapt the document to screen size and viewer preferences.

B. Adaptive Documents

Early document layout builders focused on document architecture and formatting to arrange text into lines, paragraphs, and other high-level structures [17]–[19]. LaTeX uses a dynamic programming approach to solve the problem of breaking a paragraph into lines [20]. Later work generated adaptive web and document layouts by varying document representations. Chen et al. [21] proposed adapting web pages for small screen devices by dividing the original page into smaller blocks. Xie et al. [22] presented a novel tree representation for displaying documents on various screens. FrameKit [23] generates resizable UIs via keyframe interpolation. Domshlak et al. [24] proposed a system for personalized presentation and a preference-based configuration process for web pages. Their approaches also optimized the selection of document content alternatives based on the author's and the viewer's preferences. However, all the alternatives were discrete and predefined. In contrast, in addition to predefined content alternatives, FlexDoc can automatically generate new ones and further optimize the selected alternatives to fit better into the space available for the document in the current context.

Recent document layout generation research used deep learning approaches to avoid manually defining constraints and templates. LayoutGAN [25] proposed a generative model to place graphics elements into a document layout. LayoutGAN++ [26] improved the generative layout model with transformer blocks and latent space exploration. Zheng et al. [27] added a content-awareness factor to generate document layouts based on the document topics. Neural Design Network [28] generated document layouts via deep learning networks to produce layouts that follow given constraints. However, deep learning approaches can only produce the document styles represented in their training data, which typically biases their outputs. Furthermore, they give authors less control over the generated documents and cannot generate documents that adapt dynamically, *i.e.,* to the current window/screen size. In contrast, FlexDoc gives control over the documents to both authors and viewers and, at the same time, automatically generates proper text and images to fit the screen better.

*This work was done in part while the first author was an intern at Adobe.

C. Constraint-based Resizable Layout

The need for resizable layouts is driven by the vast diversity of screen sizes and aspect ratios of current devices and the ability of desktop graphical user interfaces to resize windows interactively. While early layout models such as group, grid, table, and grid-bag layouts [29]–[31] provided basic functionality, more recent constraint-based layout models [32], [33] offer advanced options for generating resizable and responsive layouts [34]–[37]. Recent work on a more expressive layout model for graphical user interfaces (GUIs), called ORC Layout [38], unifies flow layouts and conventional constraint-based layouts through OR-constraints (ORC). OR-constraints are a combination of hard and soft constraints, where the entire OR-constraint is a hard constraint, and each disjunctive part is a soft constraint. This allows for the definition of alternatives for layout components, enabling a single layout specification to create adaptive layouts for a wide range of screen sizes, orientations, and aspect ratios. ORC layout specifications for GUIs can be efficiently solved using ORCSolver [39] and reverse-engineered from interfaces through ReverseORC [40]. FLEXDOC applies OR-constraints to optimize the layout of adaptive documents, combined with methods for generating content alternatives, to jointly optimize layout and content.

II. DOCUMENT OPTIMIZATION

Designing an adaptive document involves optimizing both content and layout to fit the screen's properties, author preferences, and viewer preferences. Our objective is to provide a comprehensive method that considers all these aspects. This involves making a large number of both element-specific and layout-related decisions. To achieve this, we formulate the document optimization problem as a joint discrete and continuous optimization process.

A. Problem Formulation

We define the document problem as an optimization problem. With this formulation, we decide on the positions and sizes of document elements (denoted as $e_i = (x_i, y_i, w_i, h_i)$, where the coordinates (x_i, y_i) represent the top-left corner of the i-th element and (w_i, h_i) represents its width and height), along with the layout of the document. Here, we focus on a setting where all the elements are rectangular or in rectangular bounding boxes, and we do not consider hierarchies.

We define two objective terms: the continuous loss term $\mathcal{L}_{\text{cont}}$ and the discrete loss term $\mathcal{L}_{\text{disc}}$. The continuous loss term focuses on the screen and element properties, such as element sizes and overall aesthetics; the discrete loss term focuses more on the author preferences and viewer preferences. The overall objective function is defined as follows:

$$\begin{aligned}
&\mathcal{L}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}, \mathbf{W}_{\text{disc}}) \\
&= \mathcal{L}_{\text{cont}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}) \\
&+ \mathcal{L}_{\text{disc}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N; \mathbf{W}_{\text{disc}}),
\end{aligned} \tag{1}$$

where the total number of elements is N, the predicted position and size of each GUI element as $\hat{e}_i = (\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$, and the preferred element positions and sizes are represented by $e_{pi} = (x_{pi}, y_{pi}, w_{pi}, h_{pi})$. $\mathbf{W}_{\text{cont}}$ is a set of weights assigned to specific continuous properties, and $\mathbf{W}_{\text{disc}}$ consists of weights associated with individual discrete properties.

We then minimize the objective function to optimize the positions and sizes of document elements. The optimization process can be represented as

$$\begin{aligned}
\{\hat{e}_1, \hat{e}_2, ..., \hat{e}_N\}^* = \text{argmin}_{\{\hat{e}_1, \hat{e}_2, ..., \hat{e}_N\}} \\
\mathcal{L}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N; \mathbf{W}_{\text{cont}}, \mathbf{W}_{\text{disc}}).
\end{aligned} \tag{2}$$

B. Continuous Loss Term

The continuous loss term $\mathcal{L}_{\text{cont}}$ addresses the relationship between elements and their properties. Here, we include image loss $\mathcal{L}_{\text{img}}$, text loss $\mathcal{L}_{\text{text}}$, and alignment loss $\mathcal{L}_{\text{align}}$:

$$\begin{aligned}
&\mathcal{L}_{\text{cont}}(\hat{e}_1, \hat{e}_2, ..., \hat{e}_N, e_{p1}, e_{p2}, ..., e_{pN}; \mathbf{W}_{\text{cont}}) \\
&= \mathbf{w}_{\text{img}}\mathcal{L}_{\text{img}} + \mathbf{w}_{\text{text}}\mathcal{L}_{\text{text}} + \mathbf{w}_{\text{align}}\mathcal{L}_{\text{align}},
\end{aligned} \tag{3}$$

where $\mathbf{W}_{\text{cont}} = \{\mathbf{w}_{\text{img}}, \mathbf{w}_{\text{text}}, \mathbf{w}_{\text{align}}\}$ are the weights for images, texts, and alignments. We currently set all these to 1.

1) Image Loss: To optimize an image, we penalize deviations from an image's preferred size (size loss) and aspect ratio (aspect ratio loss). We cannot use the difference in image *area* directly since this can significantly distort the image by changing its aspect ratio. Thus, we measure image size only based on the difference in size in each dimension. Hence, we define image size loss $\mathcal{L}_{\text{s}}$ as

$$\mathcal{L}_{\text{s}} = \frac{1}{N_{\text{img}}} \sum_{i=1}^{N_{\text{img}}} |\hat{w}_i - w_{pi}|^2 + \frac{1}{N_{\text{img}}} \sum_{i=1}^{N_{\text{img}}} |\hat{h}_i - h_{pi}|^2, \tag{4}$$

where N_{img} is the number of images.

Aspect ratio is another important measure for images, defined as w/h. The ideal situation is to maintain the same aspect ratio as for the preferred size, *i.e.,* $\hat{w}_i/\hat{h}_i = w_{pi}/h_{pi}$, which is equivalent to $\hat{w}_i \cdot h_{pi} = \hat{h}_i \cdot w_{pi}$. Hence, we define the aspect ratio loss $\mathcal{L}_{\text{ar}}$ as

$$\mathcal{L}_{\text{ar}} = \frac{1}{N_{\text{img}}} \sum_{i=1}^{N_{\text{img}}} |\hat{w}_i \cdot h_{pi} - \hat{h}_i \cdot w_{pi}|^2. \tag{5}$$

The final image loss is the sum of the size and aspect ratio loss terms. $\mathcal{L}_{\text{img}} = \mathcal{L}_{\text{s}} + \mathcal{L}_{\text{ar}}$.

2) Text Loss: We penalize text that is too small to read by considering its size deficit, i.e., by how much its font size $\hat{f}_i$ is smaller than the viewer's preferred font size f_{pi}. If the text size is larger than the preferred font size, then the size deficit is 0. Our system can dynamically generate shortened-version texts to fit the document better. In this case, we further penalize text changes if the shortened version is used. We use the summarization evaluation metric BERTScore [41] to measure the similarity between the shortened version $\hat{t}_i$ and the original version t_{orig} to improve the overall readability. A higher BERTScore indicates greater similarity. Thus, the text loss in FLEXDOC is defined as

$$\mathcal{L}_{\text{text}} = \frac{1}{N_{\text{text}}}\sum_{i=1}^{N_{\text{text}}}\max(f_{pi}-\hat{f}_i,0)-\frac{1}{N_{\text{text}}}\sum_{i=1}^{N_{\text{text}}}\text{BERTScore}(\hat{t}_i,t_{\text{orig}}) \tag{6}$$

where N_{text} is the number of text items.

3) Alignment Loss: Previous work explored how to measure the overall aesthetics of a layout based on established visual principles [42]. This overall aesthetics loss term could be easily extended to consider different aesthetic principles. FLEXDOC currently uses alignment loss. For example, we penalize when the images e_i and e_j in the same row are not aligned along the horizontal midline:

$$\mathcal{L}_{\text{align}_{(i,j)}} = |(\hat{y}_i + \frac{1}{2}\hat{h}_i) - (\hat{y}_j + \frac{1}{2}\hat{h}_j)|^2 \tag{7}$$

C. Discrete Loss Term

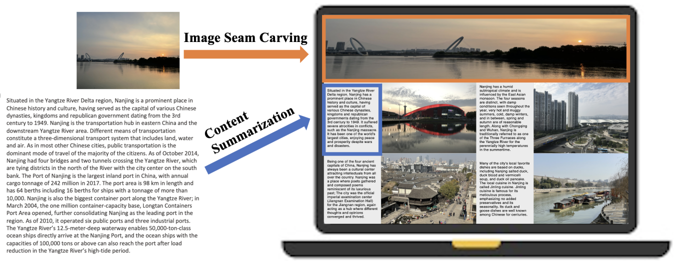
The discrete loss term $\mathcal{L}_{\text{disc}}$ involves the selection of templates and individual content alternatives. For each element e_i, if the viewer has no specific preference, the discrete loss for this element is determined by the author preference loss, $\mathcal{L}_{\text{author},i}$. When the viewer indicates their preferences without interacting directly, the discrete loss shifts to the viewer preference loss, $\mathcal{L}_{\text{viewer},i}$. However, if the viewer actively interacts with the content, the discrete loss is governed by the viewer interaction loss, $\mathcal{L}_{\text{int},i}$, ensuring that the content dynamically adjusts to their direct input.

$$\begin{aligned}&\mathcal{L}_{\text{disc}}(\hat{e}_1,\hat{e}_2,...,\hat{e}_N;\mathbf{W}_{\text{disc}})\\ &=\sum_i \mathbf{w}_{\text{author},i}\mathcal{L}_{\text{author},i}+\mathbf{w}_{\text{viewer},i}\mathcal{L}_{\text{viewer},i}+\mathbf{w}_{\text{int},i}\mathcal{L}_{\text{int},i},\end{aligned} \tag{8}$$

where one of $\{\mathbf{w}_{\text{author},i},\ \mathbf{w}_{\text{viewer},i},\ \mathbf{w}_{\text{int},i}\}$ is 1 and the other two are 0, depending on whether the viewer sets their preferences or interacts with the content.

1) Author Preference Loss: Document authors have the flexibility to define alternatives for both layout templates and content, each associated with preference ranks. Larger loss values are assigned to lower-ranked templates. In practical terms, the m-th ranked template is assigned a loss value of $-1000 \cdot (M+1-m)$, prioritizing more preferred templates, where M is the number of template alternatives. This approach creates a gradient of loss values between ranks, allowing for optimization within a template before transitioning to another. Similarly, for the i-th content, the k-th ranked template is assigned a loss value of $-50 \cdot (K_i + 1 - k_i)$ to prioritize more preferred alternatives, where K is the number of alternatives for that specific content. The final loss for the author's preference is then calculated as follows:

$$\mathcal{L}_{\text{author}} = -1000\cdot(M+1-m) - \sum_{i=1}^{N} 50\cdot(K_i+1-k_i). \tag{9}$$



Situated in the Yangtze River Delta region, Nanjing is a prominent place in Chinese history and culture, having served as the capital of various Chinese dynasties, kingdoms and republican government dating from the 3rd century to 1949. Nanjing is the transportation hub in eastern China and the downstream Yangtze River area. Different means of transportation constitute a three-dimensional transport system that includes land, water and air. As in most other Chinese cities, public transportation is the dominant mode of travel of the majority of the citizens. As of October 2014, Nanjing had four bridges and two tunnels crossing the Yangtze River, which are tying districts in the north of the River with the city center on the south bank. The Port of Nanjing is the largest inland port in China, with annual cargo tonnage of 242 million in 2017. The port area is 98 km in length and has 64 berths including 16 berths for ships with a tonnage of more than 10,000. Nanjing is also the biggest container port along the Yangtze River; in March 2004, the one million container-capacity base, Longtan Containers Port Area opened, further consolidating Nanjing as the leading port in the region. As of 2010, it operated six public ports and three industrial ports. The Yangtze River's 12.5-meter-deep waterway enables 50,000-ton-class ocean ships directly arrive at the Nanjing Port, and the ocean ships with the capacities of 100,000 tons or above can also reach the port after load reduction in the Yangtze River's high-tide period.

Situated in the Yangtze River Delta region, Nanjing has a prominent place in Chinese history and culture, having served as the capital of various Chinese dynasties, kingdoms and republican governments dating from the 3rd century to 1949. It suffered severe atrocities in conflicts, such as the Nanjing massacre. It has been one of the world's largest cities, enjoying peace and prosperity despite wars and disasters.

Nanjing has a humid subtropical climate and is influenced by the East Asian monsoon. The four seasons are distinct, with damp conditions seen throughout the year, very hot and muggy summers, cold, damp winters, and in between, spring and autumn are of reasonable length. Along with Chongqing and Wuhan, Nanjing is traditionally referred to as one of the Three Furnaces along the Yangtze River for the perennially high temperatures in the summertime.

Being one of the four ancient capitals of China, Nanjing has always been a cultural center attracting intellectuals from all over the country. Nanjing was a place where poets gathered and composed poems reminiscent of its luxurious past; The city was the official imperial examination center (Jiangnan Examination Hall) for the Jiangnan region, again acting as a hub where different thoughts and opinions converged and thrived.

Many of the city's local favorite dishes are based on ducks, including Nanjing salted duck, duck blood and vermicelli soup, and duck oil pancake. The local cuisine in Nanjing is called Jinling cuisine. Jinling cuisine is famous for its meticulous process, emphasizing no added preservatives and its seasonality. Its duck and goose dishes are well known among Chinese for centuries.

Fig. 1: Examples of image seam carving and content summarization.

2) Viewer Preferences: Viewer preferences have higher priorities than those specified by authors, as the end goal of our approach is to enhance the viewing experience. As shown in Figure 2, viewers can adjust their preferences with the sliders. For example, if the viewer increases their preference for "images" using the corresponding slider, the loss value of all image alternatives will be decreased so that images are more likely to be chosen. The range of sliders is within [0, 1], where 0.5 indicates no change in preferences. We denote the slider value to be s_k. Other loss values remain the same as defined by author preferences if not set by the viewer. Thus, the viewer preference loss is defined as

$$\mathcal{L}_{\text{viewer}} = \sum_{i=1}^{N}(0.5 - s_k)\cdot 50\cdot(K_i+1-k_i)). \tag{10}$$

3) Viewer Interactions: FLEXDOC can dynamically change the screen's content in response to viewer interactions. Viewer interactions are given the highest priority since they represent direct requests from the user. Thus, if the viewer chooses a specific template or content alternative through an interaction, that alternative must be selected unless no solution exists. Other contents are then optimized accordingly.

III. DYNAMIC CONTENT GENERATION

Given a screen/window size, we optimize the positions and sizes of elements and the selection of alternatives through the optimization process. However, fitting all content into a layout is often challenging, especially for smaller screen sizes. One solution is to scale images and text. However, this would reduce content readability if images or fonts become too small. Alternatively, we could maximize readability by cutting content, but then we would lose (too) much information.

To accommodate the vast diversity of screen sizes and document author and viewer preferences, FLEXDOC selects or dynamically generates alternative content. It applies image processing techniques such as seam carving to adapt images and BERT-based text summarization to generate alternative texts in variable sizes. It then optimizes across all the potential alternatives that conform to the given screen size and viewer preferences. For example, FLEXDOC will generate a smaller version of an image or a summarized version of a paragraph when the document is read on a small-screen device like a mobile phone. FLEXDOC also supports "content replacement plugins", meaning that the two implemented methods can be easily replaced with different image resizing and text summarization algorithms.

A. Image Seam Carving

Free-form scaling is the most common way to generate alternative images that fit a different screen size. However, standard image scaling often leads to a change in aspect ratio and thus distorts images [43]. An effective way to resize images based on geometric constraints considers the image content. Seam carving is a content-aware image resizing approach that supports both image reduction and expansion [44], which can re-target an image to fit the expected size and aspect ratio while maintaining important content in the image and reducing unexpected distortions (Figure 1 a). FLEXDOC uses this seam carving method to generate image alternatives automatically.

B. Content Summarization

Due to the diversity in screen sizes, a layout may not be able to include the complete text, even if vertical scrolling is enabled. Previous work [45] directly scaled fonts, which can significantly affect readability. Instead, it is often preferable to shorten the content so that the overall result maintains readability without losing important information. Additionally, different viewers likely have different background knowledge/interests in the same content. Some people might need more detailed information, while too detailed information might be redundant for others who are only interested in the gist of the text. It is thus helpful to have access to text alternatives with different lengths to meet these varying needs. To generate reasonable shortened versions of a piece of text, we utilize the BERT model [46] to automatically generate summarized versions [47] (Figure 1 b). Alternatively, we could use large language models (LLMs) to generate various versions of the text, but LLMs may result in higher latency.

C. Alternative Modalities

Depending on screen sizes, author and viewer preferences, and requirements such as accessibility, the same information may have to be presented through alternate modalities. For example, an image may require alternative text. We currently expect document authors to provide the content for different alternative modalities. However, FLEXDOC's content replacement plugin architecture makes it easy to automate the generation of alternative modalities if suitable algorithms are available, e.g., alternative text or audio generated by machine learning models, like in Figure 2a. FLEXDOC then optimizes the selection of alternative content modalities. For example, to support people who prefer more visual information, FLEXDOC can use images to replace text to meet their needs.

IV. DOCUMENT AUTHORING AND VIEWING

FLEXDOC optimizes the document based on the screen properties, author preferences, and viewer preferences. Authors can define their content preferences using FLEXDOCEDITOR, a graphical document template editor (Figure 4). This editor allows *authors* to guide the optimization process by providing different layout templates, content alternatives, and preference rankings. Subsequently, the document can then be optimized based on the screen size and author preferences. On the other hand, *viewers* can also adapt a document by selecting different layout templates or adjusting their preferences via simple operations. Once the screen size changes and/or the viewer indicates their preferences, the document can adapt accordingly. These changes are saved and applied to other documents, reducing the need for repeated adjustments and ensuring a consistent viewing experience.

A. Layout Templates

FLEXDOC uses *tabstop*-based layout templates to create adaptive documents due to their flexibility [33]. A tabstop is a symbolic object that represents the alignments of widgets in a layout [37], [40], [48]–[50]. A *tabstop* is essentially a variable, with an x-tabstop defining a vertical grid line through a position on the x-axis and a y-tabstop defining a corresponding horizontal grid line on the y-axis.

A *template* is defined as a set of tabstops used to align the elements that make up a document's content. Each document element is assigned to an area defined by two horizontal tabstops and two vertical tabstops. If multiple items are assigned to the same area, they flow in that area one after the other in a specified direction, similar to text. A document element may have different alternatives, with preferred alternatives having higher priorities during optimization. Templates can be predefined, reused, and shared, which reduces the burden of document authoring.

B. Authoring Documents (Author Interface)

1) Template Creation: Authors can also use the FLEXDOCEDITOR to create templates. Authors can edit tabstops, element areas defined by tabstops, and preference ranks for alternative content, as illustrated in Figure 4. Tabstops are defined by clicking on the editing canvas to set relative positions. Left clicks create horizontal tabstops (green lines), and right clicks create vertical tabstops (blue lines). Authors can then place document elements by selecting two horizontal and two vertical tabstops or layout boundaries; the selected tabstops and respective areas are highlighted in yellow.

2) Content Alternatives: Authors can define alternatives for each document element. FLEXDOCEDITOR supports alternatives with mixed modalities for the same element, *e.g.,* a document element can have both image and text alternatives. The preference ranks of alternatives can be modified via a list widget. Preferred alternatives rank higher in the list and thus get higher priority during optimization (Figure 4 b). The resulting tabstops and element alternatives with their preference ranks are then exported as a JSON file, directly used in the optimization process when viewing a document. FLEXDOC can also use multiple templates to optimize a single document, allowing authors to rank the templates available for a document by preference. Authors can use FLEXDOCEDITOR to visually create templates for web documents without

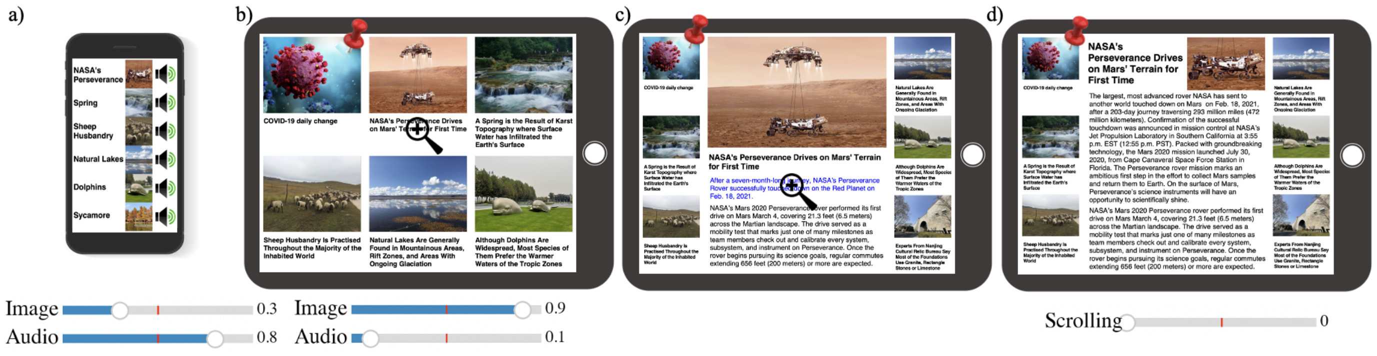

a)
b)
c)
d)
NASA's Perseverance
Spring
Sheep Husbandry
Natural Lakes
Dolphins
Sycamore
COVID-19 daily change
NASA's Perseverance Drives on Mars' Terrain for First Time
A Spring is the Result of Karst Topography where Surface Water has Infiltrated the Earth's Surface
Sheep Husbandry Is Practised Throughout the Majority of the Inhabited World
Natural Lakes Are Generally Found in Mountainous Areas, Rift Zones, and Areas With Ongoing Glaciation
Although Dolphins Are Widespread, Most Species of Them Prefer the Warmer Waters of the Tropic Zones
Image 0.3
Audio 0.8
Image 0.9
Audio 0.1
Scrolling 0

Fig. 2: a) FLEXDOC adapts a news page on a mobile phone to provide a compact overview with quick access to audio content for each article, based on (previously expressed) viewer preference (sliders below images), which prioritizes audio content over images and text here. b) The same news page adapted for a tablet device with a user preference for image content. c) As the viewer 'pins' the COVID article and 'zooms in' on the Mars article, FLEXDOC rearranges the layout accordingly, keeping the pinned article in place. d) As the viewer 'zooms in' on the blue text paragraph in the previous image with a preference for avoiding scrolling, FLEXDOC extends the paragraph to provide more details and crops the top image, avoiding the need for scrolling. Image credits: Production Perig/stock.adobe.com and NASA/JPL-Caltech.

manually defining properties in HTML files, easing the burden of adaptive website creation.

C. Viewing Documents (Viewer Interface)

FLEXDOC provides simple operations that allow viewers to express other preferences and interactively adapt documents while reading:

a) Sliders: FLEXDOC allows viewers to adjust their high-level preferences using sliders (Figure 2 ab). Viewers can use these sliders to express preferences or dislikes for specific content modalities (e.g., images). FLEXDOC then automatically generates the corresponding objective terms, optimizing the document to align as closely as possible with the viewer's preferences. These preferences can be applied across documents to ensure a consistent reading experience.

b) Zoom In: Viewers can indicate their interest in specific content by clicking on or touching it (Figure 2 bcd). FLEXDOC then increases the detail of the content where viewers have shown interest. In the backend, FLEXDOC re-ranks alternatives for the indicated content, prioritizing those with more detail (larger images, longer text).

c) Zoom Out: This operation is the inverse of 'Zoom In', causing FLEXDOC to re-rank alternatives for the content so that those with less detail are prioritized.

d) Pin: Viewers can pin content to its current location by double-clicking on the content element (Figure 2 bcd). FLEXDOC fixes this content while optimizing other content.

e) Switch Template: If authors have provided multiple alternative layout templates for a document, viewers can express a preference for a specific template by selecting it from a list (Figure 3 b). If a valid solution exists, FLEXDOC will optimize the document using the selected layout template.

f) Switch Element: Viewers can switch to an alternative representation of a document element (e.g., text vs. image, longer vs. shorter text) by right-clicking on the element and selecting a preferred alternative from the available options (Figure 6 b). Then, FLEXDOC optimizes the document based on the selected option, provided a valid solution exists.

V. APPLICATIONS

We demonstrate FLEXDOC in multiple real-world application scenarios. In addition to the new website shown in the main paper, we show three additional examples.

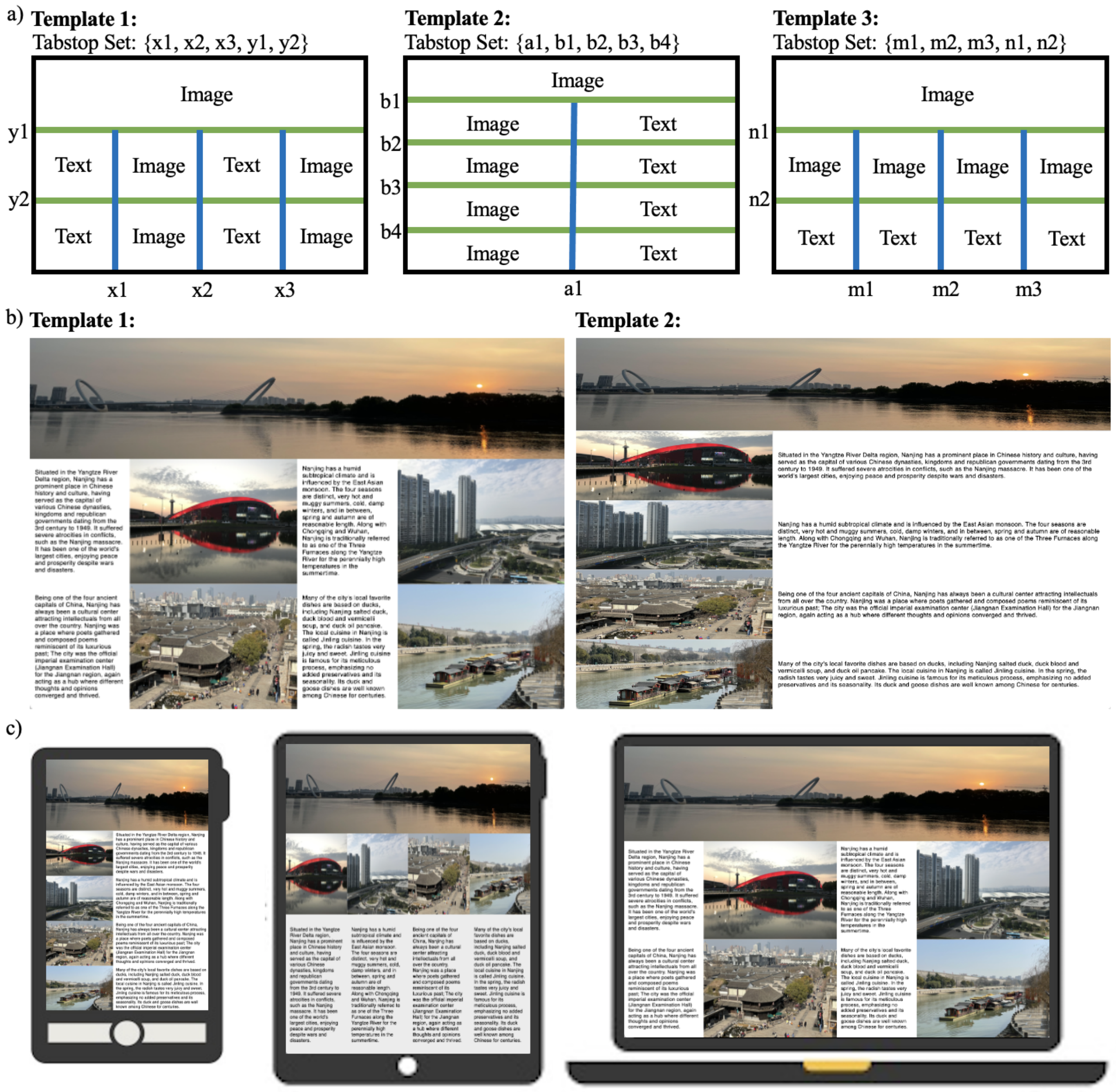
A. Scientific Paper

Reading scientific papers on devices with small screens, especially when they contain wide figures, can be challenging. Additionally, when reading papers, people often have to scan the entire document to find relevant sections they want to read first. Some individuals prefer to start by reading the abstract and the body text, while others might prefer to begin by scanning the figures and their captions to get a general idea of the paper. Furthermore, the ability to automatically convert a paper into different formats can significantly save time for scientific authors and help viewers better understand the content.

FLEXDOC addresses these use cases by adapting a paper according to device properties and viewer preferences (see Figure 5). For instance, if the viewer prefers to see only sections to gain an overview, FLEXDOC can generate a version with section titles only (Figure 5a). If the viewer prefers a figure-only two-column format of the paper that fits the screen to gain a visual overview, FLEXDOC can generate a version with figures only (Figure 5b). On a desktop screen, a wide figure can be displayed. The same figure is shown on a small screen with automatic reflowing, allowing for scrolling down.

B. Academic Website

FLEXDOC can generate adaptive websites with lists of publications for different devices by selecting the most suitable content alternatives based on device properties and viewer preferences. Corresponding alternative content is used in the



Situated in the Yangtze River Delta region, Nanjing has a prominent place in Chinese history and culture, having served as the capital of various Chinese dynasties, kingdoms and republican governments dating from the 3rd century to 1949. It suffered severe atrocities in conflicts, such as the Nanjing massacre. It has been one of the world's largest cities, enjoying peace and prosperity despite wars and disasters.

Nanjing has a humid subtropical climate and is influenced by the East Asian monsoon. The four seasons are distinct, very hot and muggy summers, cold, damp winters, and in between, spring and autumn are of reasonable length. Along with Chongqing and Wuhan, Nanjing is traditionally referred to as one of the Three Furnaces along the Yangtze River for the perennially high temperatures in the summertime.

Being one of the four ancient capitals of China, Nanjing has always been a cultural center attracting intellectuals from all over the country. Nanjing was a place where poets gathered and composed poems reminiscent of its luxurious past; The city was the official imperial examination center (Jiangnan Examination Hall) for the Jiangnan region, again acting as a hub where different thoughts and opinions converged and thrived.

Many of the city's local favorite dishes are based on ducks, including Nanjing salted duck, duck blood and vermicelli soup, and duck oil pancake. The local cuisine in Nanjing is called Jinling cuisine. In the spring, the radish tastes very juicy and sweet. Jinling cuisine is famous for its meticulous process, emphasizing no added preservatives and its seasonality. Its duck and goose dishes are well known among Chinese for centuries.

Fig. 3: Document optimization results: a) The author defines three templates. FLEXDOC optimizes tabstop positions based on the objective function. b) Document results when the viewer prefers the first or second template, respectively. c) Document results on different devices. The results balance both layout structure and the amount of content.

different layouts, highlighted in blue boxes. For example, the conference logo is displayed on desktops, the full name of the conference is shown on tablets, and the conference abbreviation is presented on mobile devices (see Figure 6a). If the viewer prefers to see the full names of authors and conferences, FLEXDOC can generate a personalized mobile version of the website by allowing the viewer to switch elements. This can be done by right-clicking on the element and selecting a preferred alternative from the available options.

C. News Website with Advertisements

We present an additional news website example containing a "VL/HCC2024" advertisement to demonstrate how FLEXDOC can adapt documents to different devices (Figure 7). On the desktop, all the rows are aligned, while on a tablet, columns are aligned. Different alternatives to the "VL/HCC2024" advertisement are selected based on the device properties and

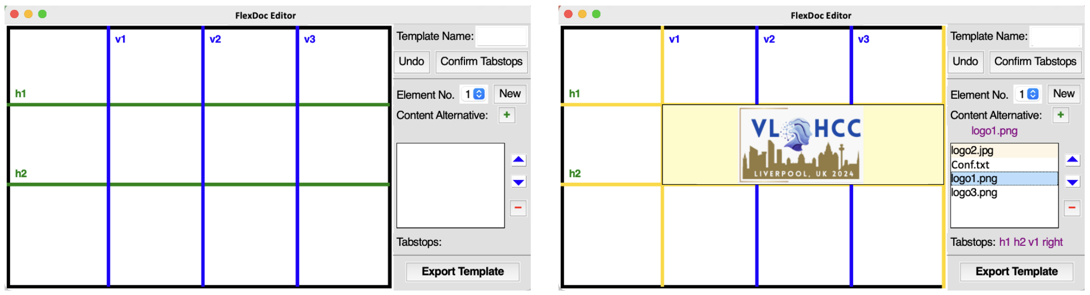


Fig. 4: FLEXDOCEDITOR is used for authoring adaptive documents. It allows authors to create templates by specifying tabstops and document elements: a) Horizontal (green) and vertical tabstops (blue) are created by clicking on the canvas. b) Document elements are placed by selecting the surrounding tabstops or layout boundaries (yellow). Authors can then use the +/- buttons to add/delete alternatives for a document element and the up/down arrow buttons to specify their preference ranks.

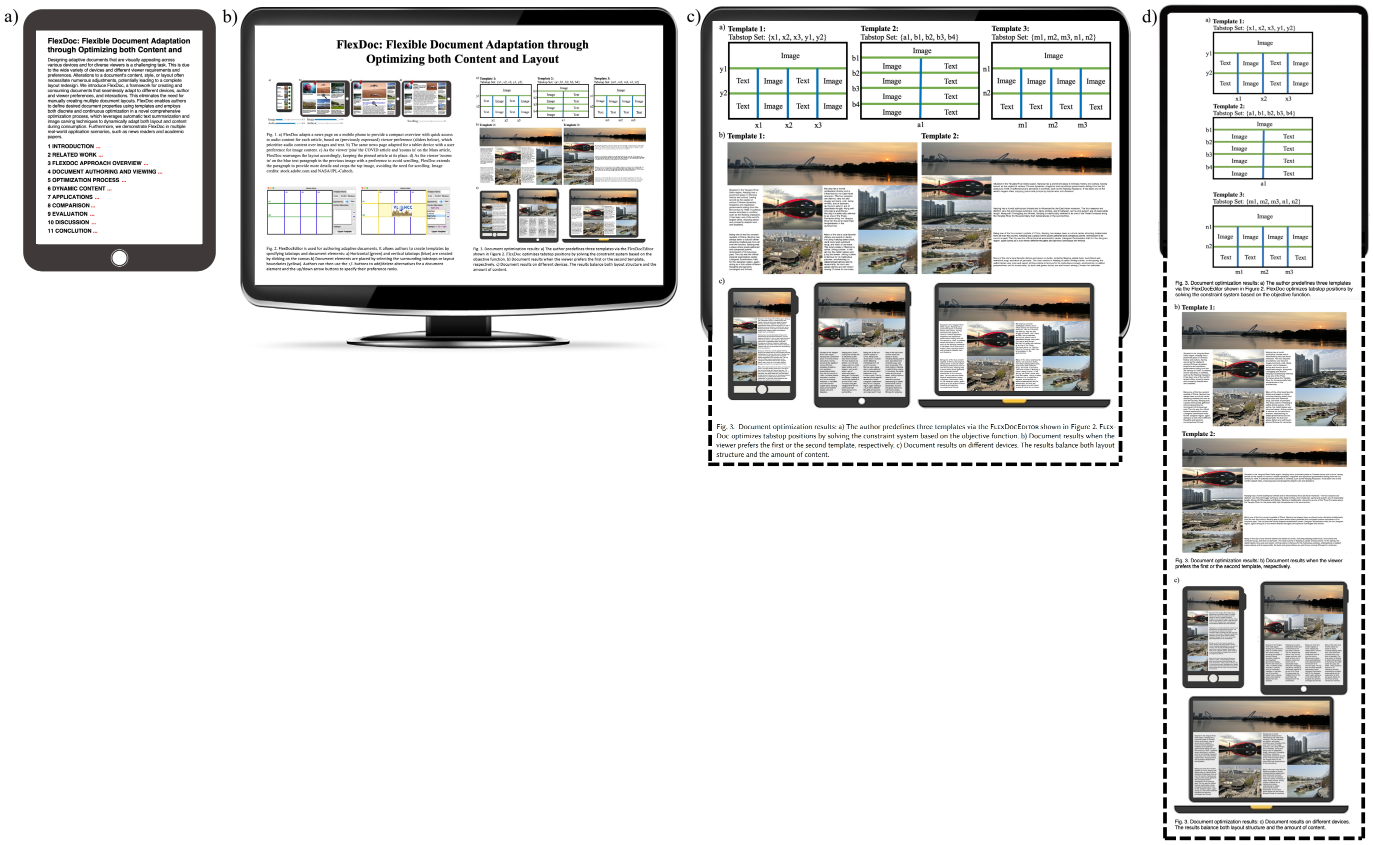
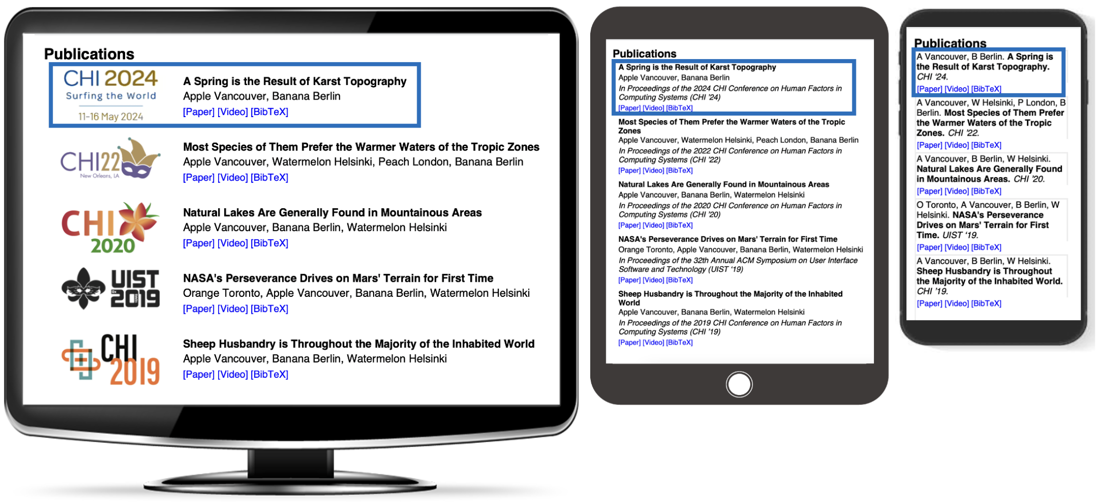

a)
b)
c)
d)
FlexDoc: Flexible Document Adaptation through Optimizing both Content and Layout
1 INTRODUCTION
2 RELATED WORK
3 FLEXDOC APPROACH OVERVIEW
4 DOCUMENT AUTHORING AND VIEWING
5 OPTIMIZATION PROCESS
6 DYNAMIC CONTENT
7 APPLICATIONS
8 COMPARISON
9 EVALUATION
10 DISCUSSION
11 CONCLUTION
Template 1:
Tabstop Set: {x1, x2, x3, y1, y2}
Template 2:
Tabstop Set: {a1, b1, b2, b3, b4}
Template 3:
Tabstop Set: {m1, m2, m3, n1, n2}
Image
Text

Fig. 5: An application of FLEXDOC for adapting the layout of scientific papers based on the screen size and author and viewer preferences. The figure illustrates the results of applying FLEXDOC to our FLEXDOC paper in various scenarios: a) The viewer prefers to see only sections, for instance, to gain an overview. b) The viewer prefers a figure-only two-column format of the paper that fits the screen, for instance, to gain a visual overview. c) A wide figure is shown on a desktop screen, including the option to scroll down, for instance, to inspect it in detail. d) The same figure is shown on a small screen with automatic reflowing, including the option to scroll down.

the layout structure to avoid the advertisement occupying too much space while still drawing attention.

VI. COMPARISON

We illustrate the comparison between our algorithm and other existing tools in adapting a desktop document to a smaller screen in Table I. Our FLEXDOC algorithm is capable of generating adaptive content, enabling image flow, and producing adaptive documents with multiple columns, all without the need for breakpoints (screen sizes at which the document layout changes) or coding. In contrast, Adobe a)



Publications

A Spring is the Result of Karst Topography
Apple Vancouver, Banana Berlin
[Paper] [Video] [BibTeX]

Most Species of Them Prefer the Warmer Waters of the Tropic Zones
Apple Vancouver, Watermelon Helsinki, Peach London, Banana Berlin
[Paper] [Video] [BibTeX]

Natural Lakes Are Generally Found in Mountainous Areas
Apple Vancouver, Banana Berlin, Watermelon Helsinki
[Paper] [Video] [BibTeX]

NASA's Perseverance Drives on Mars' Terrain for First Time
Orange Toronto, Apple Vancouver, Banana Berlin, Watermelon Helsinki
[Paper] [Video] [BibTeX]

Sheep Husbandry is Throughout the Majority of the Inhabited World
Apple Vancouver, Banana Berlin, Watermelon Helsinki
[Paper] [Video] [BibTeX]

Publications

A Spring is the Result of Karst Topography
Apple Vancouver, Banana Berlin
In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)
[Paper] [Video] [BibTeX]

Most Species of Them Prefer the Warmer Waters of the Tropic Zones
Apple Vancouver, Watermelon Helsinki, Peach London, Banana Berlin
In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)
[Paper] [Video] [BibTeX]

Natural Lakes Are Generally Found in Mountainous Areas
Apple Vancouver, Banana Berlin, Watermelon Helsinki
In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)
[Paper] [Video] [BibTeX]

NASA's Perseverance Drives on Mars' Terrain for First Time
Orange Toronto, Apple Vancouver, Banana Berlin, Watermelon Helsinki
In Proceedings of the 32th Annual ACM Symposium on User Interface Software and Technology (UIST '19)
[Paper] [Video] [BibTeX]

Sheep Husbandry is Throughout the Majority of the Inhabited World
Apple Vancouver, Banana Berlin, Watermelon Helsinki
In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)
[Paper] [Video] [BibTeX]

Publications

A Vancouver, B Berlin. **A Spring is the Result of Karst Topography.** *CHI '24.*
[Paper] [Video] [BibTeX]

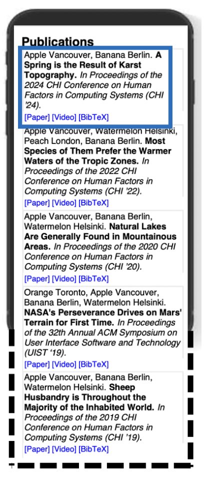
A Vancouver, W Helsinki, P London, B Berlin. **Most Species of Them Prefer the Warmer Waters of the Tropic Zones.** *CHI '22.*
[Paper] [Video] [BibTeX]

A Vancouver, B Berlin, W Helsinki. **Natural Lakes Are Generally Found in Mountainous Areas.** *CHI '20.*
[Paper] [Video] [BibTeX]

O Toronto, A Vancouver, B Berlin, W Helsinki. **NASA's Perseverance Drives on Mars' Terrain for First Time.** *UIST '19.*
[Paper] [Video] [BibTeX]

A Vancouver, B Berlin, W Helsinki. **Sheep Husbandry is Throughout the Majority of the Inhabited World.** *CHI '19.*
[Paper] [Video] [BibTeX]

b)



Publications

Apple Vancouver, Banana Berlin. **A Spring is the Result of Karst Topography.** *In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24).*
[Paper] [Video] [BibTeX]

Apple Vancouver, Watermelon Helsinki, Peach London, Banana Berlin. **Most Species of Them Prefer the Warmer Waters of the Tropic Zones.** *In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22).*
[Paper] [Video] [BibTeX]

Apple Vancouver, Banana Berlin, Watermelon Helsinki. **Natural Lakes Are Generally Found in Mountainous Areas.** *In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20).*
[Paper] [Video] [BibTeX]

Orange Toronto, Apple Vancouver, Banana Berlin, Watermelon Helsinki. **NASA's Perseverance Drives on Mars' Terrain for First Time.** *In Proceedings of the 32th Annual ACM Symposium on User Interface Software and Technology (UIST '19).*
[Paper] [Video] [BibTeX]

Apple Vancouver, Banana Berlin, Watermelon Helsinki. **Sheep Husbandry is Throughout the Majority of the Inhabited World.** *In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19).*
[Paper] [Video] [BibTeX]

Fig. 6: a) FLEXDOC can generate adaptive websites with lists of publications for different devices. Corresponding alternative content is used in the different layouts (in blue boxes). For example, the conference logo (on the desktop), the full name of the conference (on a tablet), and the conference abbreviation (on a mobile) are alternatives. b) Personalized mobile version of the website if the viewer prefers to see the full names of authors and conferences.

Acrobat Liquid Mode is restricted to a single-column format for mobile document viewing, lacks automatic adaptation to a multi-column format, and does not support image flows. Responsive Web Design (RWD) utilizes HTML and CSS to automatically resize, hide, shrink, or enlarge a website to fit different devices. However, it requires coding and places the onus on authors to manually code and set breakpoints for different layout sizes. While Webflow eliminates the need for coding, it still requires authors to establish breakpoints for varying screen sizes and does not support image flows without designing different layouts for different devices. Furthermore, all these approaches, except our FLEXDOC, fail to generate adaptive content. This means that once the design is finalized, the content becomes fixed, thereby limiting viewers from modifying the content according to their preferences. FLEXDOC, on the other hand, can adapt to different viewer preferences, a feature currently unavailable in other tools.

In addition, we compare FLEXDOC to previous adaptive document layout methods. In contrast to Zooming User Interfaces (ZUIs), e.g., [51], which make some components invisible while zooming, FLEXDOC enables adaptive content generation and optimizes layouts to emphasize the object of interest while also keeping other components visible/readable and the layout reasonably close to the original one. Compared to content management systems (CMSs), FLEXDOC provides more expressiveness for authors and a more personalized experience for viewers. FLEXDOC could be integrated into a CMS to address the limitations of such systems. Responsive web design (RWD) uses HTML and CSS to automatically resize, hide, shrink, or enlarge a website to fit different devices. Compared to RWD, FLEXDOC can adapt to different viewer preferences, which is not currently possible for RWD.

VII. USE CASES

FLEXDOC's ability to adapt to different devices and content enables adaptive documents in various use cases:

1) Adaption to Different Devices: Most current documents are static. Once a document has been created, viewers have very limited options to change how they view the document. For example, many documents cannot be easily shown on small screens such as mobile phones. Generating and adapting documents to different devices with various sizes, aspect ratios, and orientations is a challenging problem. In contrast, documents made with FLEXDOC can adapt more easily to different devices.

2) Accessibility: FLEXDOC benefits people with different document viewing requirements. It generates documents with more images when the user prefers visual content over text. FLEXDOC can also generate documents with different font sizes or even audio options if the user has vision impairments. Furthermore, alternative versions of a text with different languages can be used to support internationalization.

3) Task-Specific Customization: FLEXDOC enables task-specific customization depending on viewers' needs. For example, viewers often prefer first an overview of the latest news and then consume some of the news items in more detail. FLEXDOC supports the display of content at different levels of detail, showing more detailed versions whenever the viewer indicates their interests via clicking or touching the screen.

VIII. LIMITATION AND FUTURE WORK

FLEXDOC employs automated content generation methods to produce suitable images and texts to fit the document better. However, these methods need to be evaluated for quality and efficiency. Sometimes, a text summarization model may generate low-quality text [52], or an intelligent seam carving

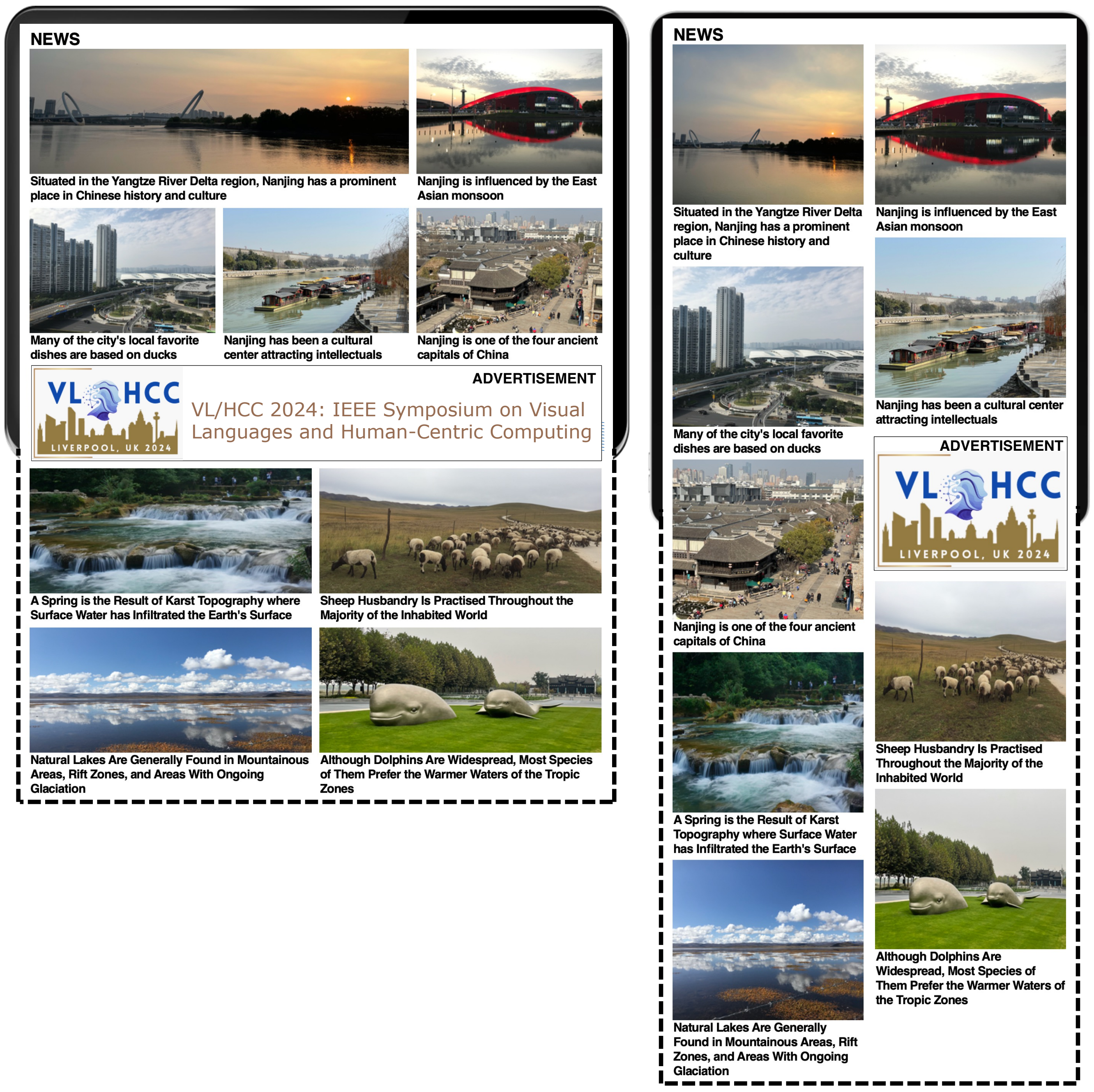

NEWS
Situated in the Yangtze River Delta region, Nanjing has a prominent place in Chinese history and culture
Nanjing is influenced by the East Asian monsoon
Many of the city's local favorite dishes are based on ducks
Nanjing has been a cultural center attracting intellectuals
Nanjing is one of the four ancient capitals of China
ADVERTISEMENT
VL HCC
LIVERPOOL, UK 2024
VL/HCC 2024: IEEE Symposium on Visual Languages and Human-Centric Computing
A Spring is the Result of Karst Topography where Surface Water has Infiltrated the Earth's Surface
Sheep Husbandry Is Practised Throughout the Majority of the Inhabited World
Natural Lakes Are Generally Found in Mountainous Areas, Rift Zones, and Areas With Ongoing Glaciation
Although Dolphins Are Widespread, Most Species of Them Prefer the Warmer Waters of the Tropic Zones

Fig. 7: A news website example on different devices with a "VL/HCC2024" advertisement, including the option to scroll down. On the desktop, all the rows are aligned while on a tablet, columns are aligned. Different alternatives to the "VL/HCC2024" advertisement are selected based on the device properties and the layout structure to avoid the advertisement occupying too much space while still drawing attention.

approach may distort images or create visual artifacts. Nevertheless, any content generation method can be used within FLEXDOC. We provide an approach for "content replacement plugins" within FLEXDOC, and the two methods implemented in our prototype can be easily replaced with any desired image resizing and text summarization algorithms. Poor content generation can be identified by the author and manually addressed by replacing content accordingly. Alternatively, automated feedback, such as an ML critic model, could be used to evaluate the quality of content alternatives.

While FLEXDOC reduces the authoring effort for templates, we are currently limited to ones that can be defined via tab-stops. Future work could enable the adaptation to infographics or images with irregular boundaries to different screens and

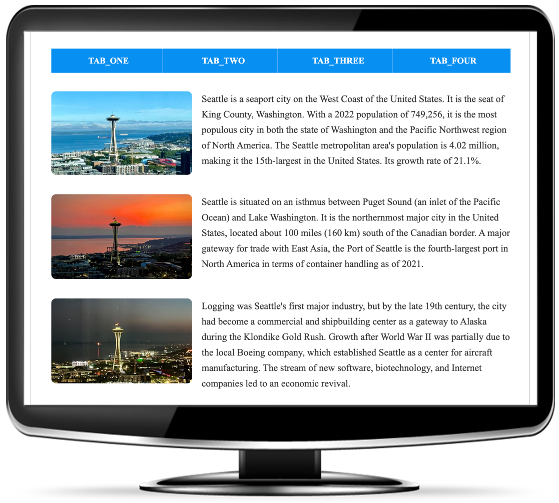
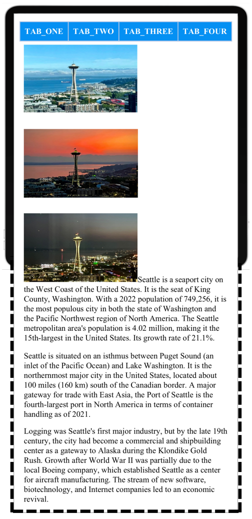
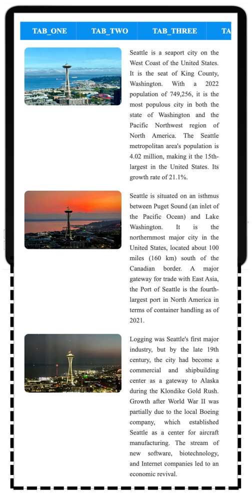
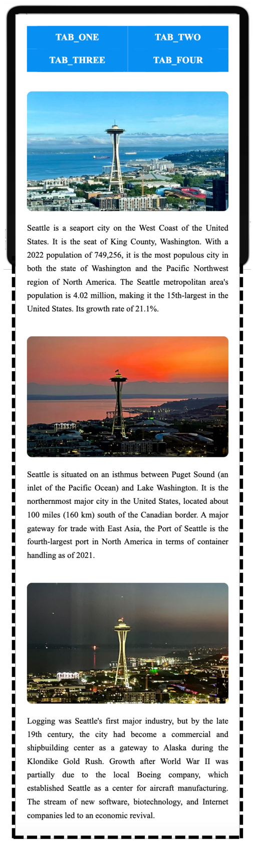
Input Document	Adobe Liquid Mode	Webflow	Ours
 Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. With a 2022 population of 749,256, it is the most populous city in both the state of Washington and the Pacific Northwest region of North America. The Seattle metropolitan area's population is 4.02 million, making it the 15th-largest in the United States. Its growth rate of 21.1%. Seattle is situated on an isthmus between Puget Sound (an inlet of the Pacific Ocean) and Lake Washington. It is the northernmost major city in the United States, located about 100 miles (160 km) south of the Canadian border. A major gateway for trade with East Asia, the Port of Seattle is the fourth-largest port in North America in terms of container handling as of 2021. Logging was Seattle's first major industry, but by the late 19th century, the city had become a commercial and shipbuilding center as a gateway to Alaska during the Klondike Gold Rush. Growth after World War II was partially due to the local Boeing company, which established Seattle as a center for aircraft manufacturing. The stream of new software, biotechnology, and Internet companies led to an economic revival.	 Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. With a 2022 population of 749,256, it is the most populous city in both the state of Washington and the Pacific Northwest region of North America. The Seattle metropolitan area's population is 4.02 million, making it the 15th-largest in the United States. Its growth rate of 21.1%. Seattle is situated on an isthmus between Puget Sound (an inlet of the Pacific Ocean) and Lake Washington. It is the northernmost major city in the United States, located about 100 miles (160 km) south of the Canadian border. A major gateway for trade with East Asia, the Port of Seattle is the fourth-largest port in North America in terms of container handling as of 2021. Logging was Seattle's first major industry, but by the late 19th century, the city had become a commercial and shipbuilding center as a gateway to Alaska during the Klondike Gold Rush. Growth after World War II was partially due to the local Boeing company, which established Seattle as a center for aircraft manufacturing. The stream of new software, biotechnology, and Internet companies led to an economic revival.	 Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. With a 2022 population of 749,256, it is the most populous city in both the state of Washington and the Pacific Northwest region of North America. The Seattle metropolitan area's population is 4.02 million, making it the 15th-largest in the United States. Its growth rate of 21.1%. Seattle is situated on an isthmus between Puget Sound (an inlet of the Pacific Ocean) and Lake Washington. It is the northernmost major city in the United States, located about 100 miles (160 km) south of the Canadian border. A major gateway for trade with East Asia, the Port of Seattle is the fourth-largest port in North America in terms of container handling as of 2021. Logging was Seattle's first major industry, but by the late 19th century, the city had become a commercial and shipbuilding center as a gateway to Alaska during the Klondike Gold Rush. Growth after World War II was partially due to the local Boeing company, which established Seattle as a center for aircraft manufacturing. The stream of new software, biotechnology, and Internet companies led to an economic revival.	 Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. With a 2022 population of 749,256, it is the most populous city in both the state of Washington and the Pacific Northwest region of North America. The Seattle metropolitan area's population is 4.02 million, making it the 15th-largest in the United States. Its growth rate of 21.1%. Seattle is situated on an isthmus between Puget Sound (an inlet of the Pacific Ocean) and Lake Washington. It is the northernmost major city in the United States, located about 100 miles (160 km) south of the Canadian border. A major gateway for trade with East Asia, the Port of Seattle is the fourth-largest port in North America in terms of container handling as of 2021. Logging was Seattle's first major industry, but by the late 19th century, the city had become a commercial and shipbuilding center as a gateway to Alaska during the Klondike Gold Rush. Growth after World War II was partially due to the local Boeing company, which established Seattle as a center for aircraft manufacturing. The stream of new software, biotechnology, and Internet companies led to an economic revival.

TABLE I: Comparison between our algorithm and other existing tools, in a scenario where when we adapt a document on the desktop to a smaller screen.

user preferences. Also, our current work does not consider semantic categories or the hierarchical structure of documents. Future work could explore more extensive optimization based on the semantics of document elements and the relationships among those elements, particularly hierarchical structures.

IX. Conclusion

We introduced FlexDoc, an innovative adaptive document approach that facilitates dynamic optimization of both content and layout structure. This optimization is not solely dependent on screen or window sizes, but also takes the preferences of both authors and viewers into account. FlexDoc optimizes the layout structure using an objective function, condenses content to accommodate the layout, and further allows the outcome to adapt based on user interactions. We anticipate that our method could have broad applications across diverse screen sizes and document types. The versatility of FlexDoc paves the way for potential future document-centric applications. Additionally, future work can move beyond the current model where authors predefine extended content or extract background information from previous articles. Instead, we could develop more sophisticated approaches to dynamically extend documents, and generate more relevant, high-quality details. For example, in future work content generation approaches such as Large Language Models (LLMs) could be used as part of the creation of intelligent, flexible documents to provide suitable content for layouts that are in real time optimized to fit the available space and context.

References

[1] D. S. Weld, C. Anderson, P. Domingos, O. Etzioni, K. Gajos, T. Lau, and S. Wolfman, "Automatically personalizing user interfaces," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 1613–1619, Morgan Kaufmann Publishers Inc., 2003.

[2] J. Fogarty and S. E. Hudson, "Gadget: A toolkit for optimization-based approaches to interface and display generation," *ACM Trans. Graph.*, vol. 23, p. 730, aug 2004.

[3] Y. Jiang, Y. Lu, J. Nichols, W. Stuerzlinger, C. Yu, C. Lutteroth, Y. Li, R. Kumar, and T. J.-J. Li, "Computational approaches for understanding, generating, and adapting user interfaces," in *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[4] Y. Jiang, Y. Lu, C. Lutteroth, T. J.-J. Li, J. Nichols, and W. Stuerzlinger, "The future of computational approaches for understanding and adapting user interfaces," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[5] Y. Jiang, Y. Lu, C. Kliman-Silver, C. Lutteroth, T. J.-J. Li, J. Nichols, and W. Stuerzlinger, "Computational methodologies for understanding, automating, and evaluating user interfaces," in *Extended Abstracts of the*

2024 CHI Conference on Human Factors in Computing Systems, CHI EA '24, 2024.

[6] Y. Jiang, "Computational representations for graphical user interfaces," in *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI EA '24, 2024.

[7] Y. Jiang, C. Zhou, V. Garg, and A. Oulasvirta, "Graph4gui: Graph neural networks for representing graphical user interfaces," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–18, 2024.

[8] L. Hegemann, Y. Jiang, J. G. Shin, Y.-C. Liao, M. Laine, and A. Oulasvirta, "Computational assistance for user interface design: Smarter generation and evaluation of design ideas," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–5, 2023.

[9] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Decision-theoretic user interface generation," in *AAAI'08*, pp. 1532–1536, AAAI Press, 2008.

[10] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Automatically Generating Personalized User Interfaces With Supple," *Artif. Intell*, vol. 174, no. 12-13, pp. 910–950, 2010.

[11] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Improving the Performance of Motor-Impaired Users With Automatically-Generated, Ability-Based Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 1257–1266, ACM, 2008.

[12] K. Gajos, R. Hoffmann, and D. Weld, "Improving user interface personalization," in *Supplementary Proceedings of UIST'04*, ACM, 2004.

[13] K. Gajos, A. Wu, and D. S. Weld, "Cross-device consistency in automatically generated user interfaces," in *Proceedings of the 2nd Workshop on Multi-User and Ubiquitous User Interfaces*, pp. 7–8, ACM, 2005.

[14] K. Gajos and D. Weld, "Preference elicitation for interface optimization," in *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology*, pp. 173–182, ACM, 01 2005.

[15] A. Swearngin, C. Wang, A. Oleson, J. Fogarty, and A. J. Ko, *Scout: Rapid Exploration of Interface Layout Alternatives through High-Level Design Constraints*, p. 1–13. New York, NY, USA: Association for Computing Machinery, 2020.

[16] P. Bielik, M. Fischer, and M. Vechev, "Robust relational layout synthesis from examples for android," *Proc. ACM Program. Lang.*, vol. 2, Oct. 2018.

[17] A. J. H. Peels, N. J. M. Janssen, and W. Nawijn, "Document architecture and text formatting," *ACM Trans. Inf. Syst.*, vol. 3, p. 347–369, Oct. 1985.

[18] D. E. Knuth and M. F. Plass, "Breaking paragraphs into lines," *Software: Practice and Experience*, vol. 11, no. 11, pp. 1119–1184, 1981.

[19] R. Furuta, J. Scofield, and A. Shaw, "Document formatting systems: Survey, concepts, and issues," *ACM Comput. Surv.*, vol. 14, p. 417–472, Sept. 1982.

[20] F. Mittelbach, "Formatting documents with floats a new algorithm for latex 2ε," in *Volume 21, Number 3/September 2000 2000 Annual Meeting Proceedings*, p. 278, 2000.

[21] Y. Chen, X. Xie, W.-Y. Ma, and H.-J. Zhang, "Adapting web pages for small-screen devices," *IEEE internet computing*, vol. 9, no. 1, pp. 50–56, 2005.

[22] X. Xie, C. Wang, L.-Q. Chen, and W.-Y. Ma, "An adaptive web page layout structure for small devices," *Multimedia Systems*, vol. 11, no. 1, pp. 34–44, 2005.

[23] J. Wu, K. Todi, J. Chan, B. A. Myers, and B. Lafreniere, "Framekit: A tool for authoring adaptive uis using keyframes," in *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, (New York, NY, USA), p. 660–674, Association for Computing Machinery, 2024.

[24] C. Domshlak, R. I. Brafman, and S. E. Shimony, "Preference-based configuration of web page content," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, (San Francisco, CA, USA), p. 1451–1456, Morgan Kaufmann Publishers Inc., 2001.

[25] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Generating graphic layouts with wireframe discriminators." arXiv preprint arXiv:1901.06767, 2019.

[26] K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, "Constrained graphic layout generation via latent optimization," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 88–96, ACM, 2021.

[27] X. Zheng, X. Qiao, Y. Cao, and R. W. H. Lau, "Content-aware generative modeling of graphic design layouts," *ACM Trans. Graph.*, vol. 38, July 2019.

[28] H.-Y. Lee, L. Jiang, I. Essa, P. B. Le, H. Gong, M.-H. Yang, and W. Yang, "Neural design network: Graphic layout generation with constraints." arXiv e-prints: 1912.09421, 2019.

[29] B. Myers, S. E. Hudson, and R. Pausch, "Past, present, and future of user interface software tools," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, p. 3–28, Mar. 2000.

[30] B. A. Myers, "User interface software tools," *ACM Trans. Comput.-Hum. Interact.*, vol. 2, p. 64–103, Mar. 1995.

[31] B. A. Myers, R. G. Mcdaniel, R. C. Miller, A. S. Ferrency, A. Faulring, B. D. Kyle, I. C. Society, I. C. Society, A. Mickish, A. Klimovitski, and P. Doane, "The amulet environment: New models for effective user interface software development," *IEEE Transactions on Software Engineering*, vol. 23, pp. 347–365, 1997.

[32] C. Zeidler, G. Weber, A. Gavryushkin, and C. Lutteroth, "Tiling algebra for constraint-based layout editing," *Journal of Logical and Algebraic Methods in Programming*, vol. 89, pp. 67–94, 2017.

[33] C. Lutteroth, R. Strandh, and G. Weber, "Domain specific high-level constraints for user interface layout," *Constraints*, vol. 13, no. 3, pp. 307–342, 2008.

[34] S. Karsenty, J. A. Landay, and C. Weikart, "Inferring Graphical Constraints With Rockit," in *Proceedings of the Conference on People and Computers VII*, HCI'92, pp. 137–153, Cambridge University Press, 1993.

[35] A. Scoditti and W. Stuerzlinger, "A New Layout Method for Graphical User Interfaces," in *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pp. 642–647, IEEE, 2009.

[36] G. Weber, "A Reduction of Grid-Bag Layout to Auckland Layout," in *Proceedings of the 2010 21st Australian Software Engineering Conference*, ASWEC '10, pp. 67–74, IEEE Computer Society, 2010.

[37] C. Zeidler, C. Lutteroth, G. Weber, and W. Stürzlinger, "The auckland layout editor: An improved gui layout specification process," in *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*, CHINZ '12, (New York, NY, USA), p. 103, Association for Computing Machinery, 2012.

[38] Y. Jiang, R. Du, C. Lutteroth, and W. Stuerzlinger, "Orc layout: Adaptive gui layout with or-constraints," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[39] Y. Jiang, W. Stuerzlinger, M. Zwicker, and C. Lutteroth, "Orcsolver: An efficient solver for adaptive gui layout with or-constraints," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, (New York, NY, USA), p. 1–14, Association for Computing Machinery, 2020.

[40] Y. Jiang, W. Stuerzlinger, and C. Lutteroth, "Reverseorc: Reverse engineering of resizable user interface layouts with or-constraints," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[41] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert." arXiv preprint arXiv:1904.09675, 2019.

[42] D. Schölgens, S. Müller, C. Bauer, R. Tilly, and D. Schoder, "Aesthetic measures for document layouts: Operationalization and analysis in the context of marketing brochures," in *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, (New York, NY, USA), p. 21–30, Association for Computing Machinery, 2016.

[43] M. Laine, Y. Zhang, S. Santala, J. P. P. Jokinen, and A. Oulasvirta, "Responsive and personalized web layouts with integer programming," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, May 2021.

[44] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, (New York, NY, USA), p. 10–es, Association for Computing Machinery, 2007.

[45] A. Borning, R. K.-H. Lin, and K. Marriott, "Constraint-based document layout for the web," *Multimedia systems*, vol. 8, no. 3, pp. 177–189, 2000.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805, 2018.

[47] D. Miller, "Leveraging bert for extractive text summarization on lectures." arXiv preprint arXiv:1906.04165, 2019.

[48] O. Hashimoto and B. A. Myers, "Graphical styles for building interfaces by demonstration," in *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology*, UIST '92, (New York, NY, USA), p. 117–124, Association for Computing Machinery, 1992.

[49] S. E. Hudson and S. P. Mohamed, "Interactive specification of flexible user interface displays," *ACM Trans. Inf. Syst.*, vol. 8, p. 269–288, July 1990.

[50] C. Lutteroth and G. Weber, "User interface layout with ordinal and linear constraints," in *Proceedings of the 7th Australasian User Interface Conference - Volume 50*, AUIC '06, (AUS), p. 53–60, Australian Computer Society, Inc., 2006.

[51] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, "Pad++: A zoomable graphical sketchpad for exploring alternate interface physics," *Journal of Visual Languages & Computing*, vol. 7, no. 1, pp. 3–32, 1996.

[52] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 540–551, Association for Computational Linguistics, Nov. 2019.