# 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.

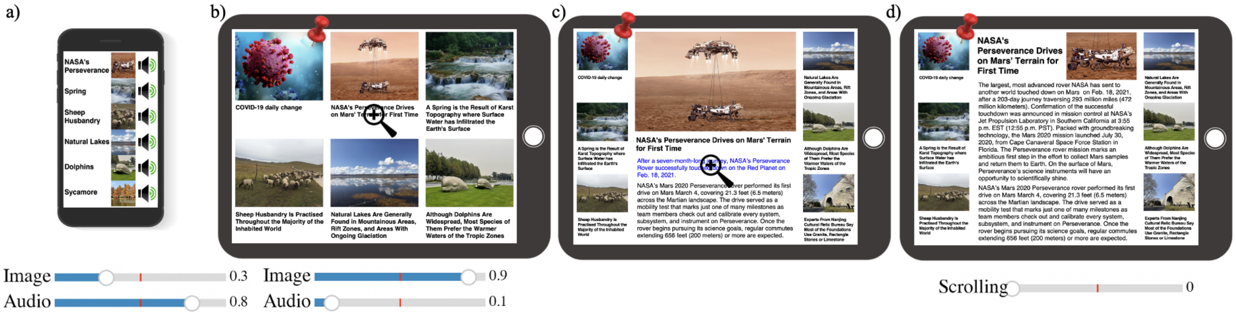


Fig. 1: a) FlexDoc adapts a news page on a mobile phone to provide a compact overview with quick access to audio content for each article, based on viewer preferences (sliders below images), which prioritize audio content over images and text here. b) The same news page adapted for a tablet device with 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

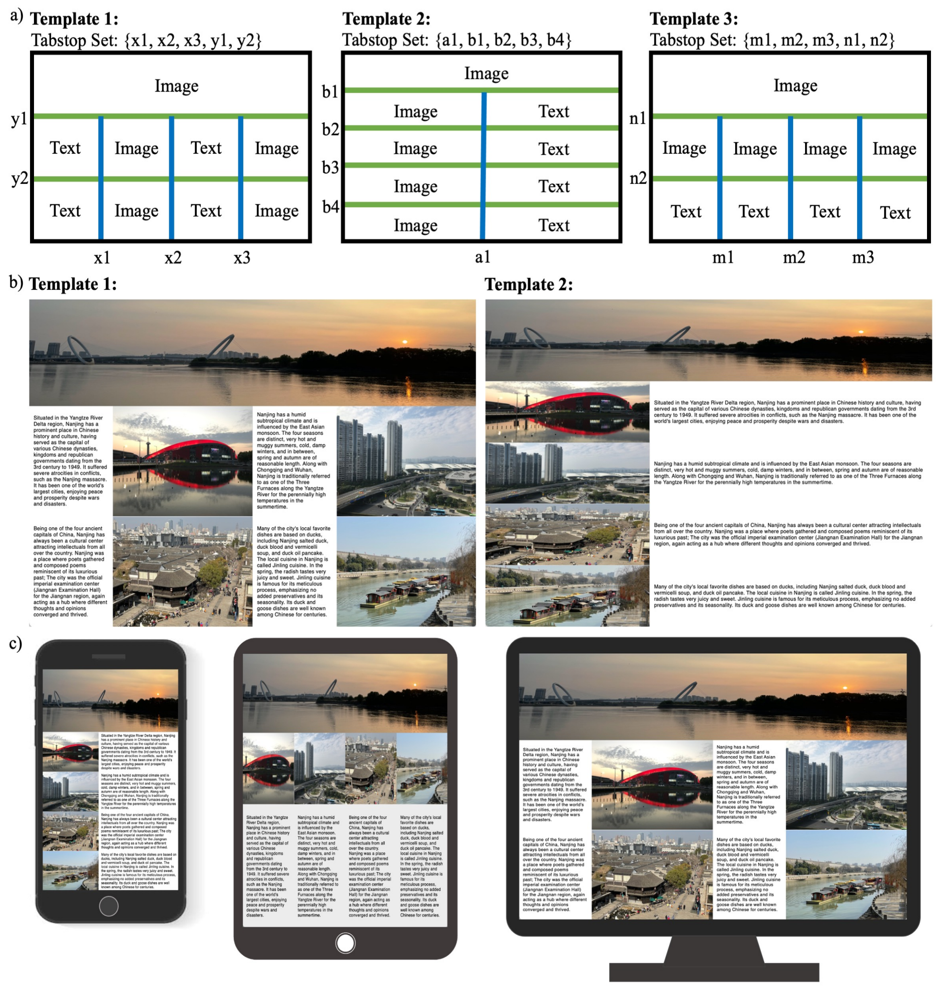


Fig. 2: Document optimization results: a) The author defines three templates. FLEXDOC optimizes tabstop positions based on the objective function. b) Document results when the viewer prefers the first or second template, respectively. c) Results on different devices, which balance both layout structure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