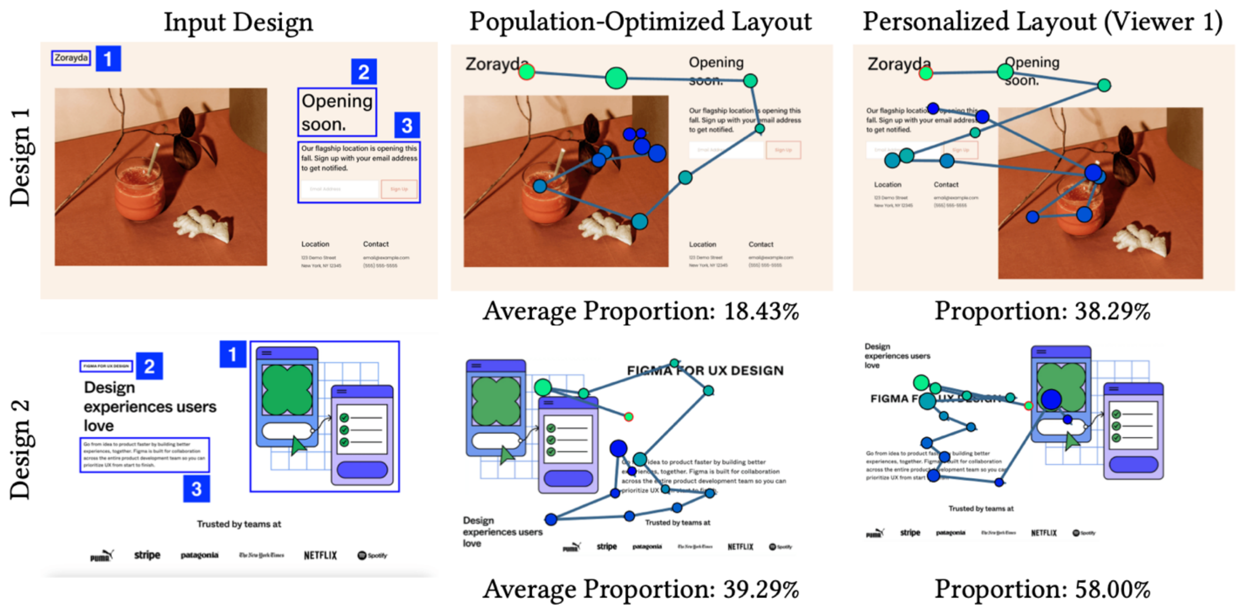
ELEC-E7852 Computational Interaction and Design 2024

Antti Oulasvirta and Yue Jiang
Aalto University


Input Design
Population-Optimized Layout
Personalized Layout (Viewer 1)
Design 1
Design 2
Zorayda
Opening soon.
1
2
3
Design experiences users love
FIGMA FOR UX DESIGN
Trusted by teams at
Average Proportion: 18.43%
Proportion: 38.29%
Average Proportion: 39.29%
Proportion: 58.00%

Assignment 3a: Saliency models (mandatory, 5p)

Topic. Our topic is the optimization of an information display's *visual flow*. Visual flow refers to the order in which it attracts users' attention when they first see it; example: (1: header), (2: OK button), (3: face).

Task: Given a display and three elements marked as important, in decreasing order or importance, the goal is to *algorithmically* change the display such that the desired flow is achieved. This requires that your algorithm can 1) generate candidate designs and 2) use EyeFormer to assess if the desired visual flow has been achieved.

Case. Next, define a *case* for your topic. The case can be related to user interfaces, posters, infographics or the like. Find a few instances of the case, or create them yourself. In your report, motivate your case. Tell why someone might want to do what you do, including their objectives.

Tip: The way you represent display for optimization is one challenge in task. Will you be editing images, HTML, or something else? You need to solve how you can edit the display and then use EyeFormer on it to assess it. We give some tips below.

Your report must:

- Explain the case with screenshots or other illustration
- Explain how you represent the UI for optimization (screenshot of key part of code)
- Explain how you formulate an objective function such that you can use EyeFormer
- Show results for 3-5 instances of your case; pinpoint success and failure cases
- Conclude: discuss what works and what does not

Grading:

- Each bullet point above is one point

Appendix A: Programming tips if you want to create an image of a webpage

Let's assume we have a web page (HTML code) for example showing a sign-up form and we want to make the submit button of the form more salient. In the following, we will use JavaScript (Nodejs) to modify programmatically the style of the button.

We have to write:

1. A program (or function) that takes a snapshot from a webpage URL. You can use PhantomJS within a Nodejs program.
2. A program (or function) that queries the DOM and modifies the style of the form button. You can use the cheerio package within Nodejs.

Then we need a main.js program to orchestrate everything:

- Call (1) and save the screenshot of the initial web page.
- Call (2) and change the style of the button.

How to modify the style is something you have to learn on your own. There are several things you can change to improve saliency; e.g. color, size, border, padding, etc. Remember to save the modified HTML file after changing the style of the button.

- Call (1) and save the screenshot of the modified page.

If you don't know JavaScript, you can apply the same strategy to the design of an Android or iOS app. Alternatively, in Python, you can create a mockup UI with GTK.

Appendix B: Tips if you want to compute saliency for an image

You can also take a UI image as input and use common image manipulation techniques. For example, in Python:

- Load the UI image, e.g. with Pillow
- Select the bounding box of the element you want to modify and apply some changes to the element pixels to improve saliency.