# Website Color Palette Optimization Utilizing Graph-Based Visual Salience

**Lari Haapaniemi**
Aalto University
Helsinki, Finland
Lari.Haapaniemi@aalto.fi

## ABSTRACT

Choosing a color palette that highlights specific elements of the website without giving up the harmonization takes a practiced eye and manual work from developers and designers. To help developers with this task, salience models can be used to output salience maps, which show well a specific element stands out from its surroundings. These salience maps can be examined manually, but also by measuring the brightness of certain elements on a map trough computational means. Salience maps however do not consider whether the website design itself is too glaring. The goal of our research is to combine Graph-Based Visual Salience (GBVS) model with harmonious color palettes and calculate which combination of those colors creates the most efficient salience for defined elements. This process can be done to specifically tailored websites which define colors of elements as variables in CSS and allow local hosting of the website implementation. The process starts with defining the required parameters locally and executing a script which calculates the best design considering the given harmonious color palette and preferred elements. The calculation is done by creating a salience map of the website and extracting the salience values for a specific elements area which we want to optimize. This tool will help developers and designers to save time when deciding the scheme and assignment of colors and allow extracting exact numerical values for how well a certain element draws the gaze. The tool could be further developed to work as a plugin to evaluate websites more easily.

## INTRODUCTION

The need for good software practices and automation have been highlighted by the current agile development methods. DevOps software model [1] encourages continuous development and user centred focus, which is supported by the tool presented in this paper. Manual color palette optimization is highly subjective, time consuming and takes resources out of other design steps. Designers and developers currently rely on estimating their design by their own intuition, user and peer feedback or in rare cases, by utilizing computational methods such as salience.

**Figure 1. Results of the tool when optimizing for buttons and side content (going to be replaced with a more modern example)**

In figure 1 we can see the output of our tool on a variety of color palettes. These palettes were created using websites that utilize harmonious color picker methods [2]. The total outlook of the website is highly dependent on the color palettes chosen, and what elements the user wants the salience models to optimize for. In our context, salience optimization means finding the best possible design for drawing attention to given element. This tool utilizes Graph-Based Visual Salience [3] (GBVS) implementation created for Python [4], which extracts and normalizes a salience map from an image, resulting in an easy way to find the most prominent areas.

Companies that design web-services want to save costs and have concrete data to show how good their design is compared to competition or previous iterations. Since the market size for web design only in US was estimated to be around 4.2 billion in 2022 [5], good development practices are a must to stay ahead of the competition. While the tool presented requires some structural decisions such as defining colors as variables which are not tied to specific elements, this is a good practice for supporting dark mode, A/B testing and other common practices in Agile web-development.

Currently estimating a harmonious color palette for an entire page which highlights specific elements takes several steps. If we were to assume that the designer utilizes existing computational methods for choosing the palette and estimating salience, they would have to utilize multiple software's and a fair amount of manual work. In this task analysis we will not

be using any specific tools, but the inspiration is drawn from references [2][4].

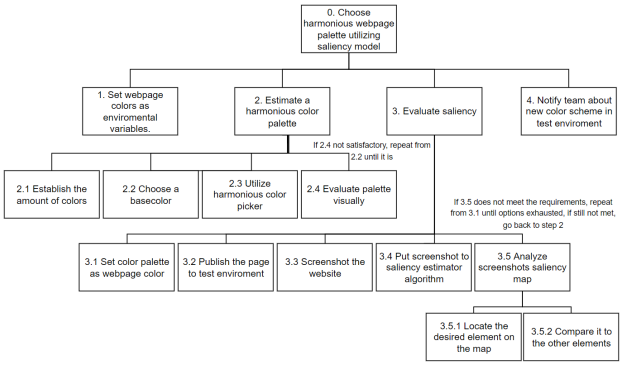


**Figure 2. Task analysis of how computational methods are currently used in estimating salience and harmonious color palette.**

As we can see in figure 2, in the current method can require multiple iterations to get right and relies quite a lot on one's intuition and previous understanding of good design. With the tool proposed in the paper, some of the second and the third part will be fully automated, so it is no longer a concern for the designer or developer.

This papers contribution is to introduce a tool that can find the most efficient design for given harmonious palettes, when the goal is to highlight a specific element. This is done by utilizing existing research and implementation on Graph-Based Visual Salience models and open-source website templates. A variety of harmonious color palette pickers were used during development and utilize the same foundations as Hu et al. [2].

## RELATED WORK

The paper by Hu et al. [2] provides an easier way for designers to find harmonious color schemes. This is done by utilizing two color harmony principles. The research created an interactive visualization tool, that allows users to create color palettes with as few steps as possible, resulting in efficient workflow according to a pilot study. Previous work had already created interactive tools based on color theory, but the goal of this paper was to create a simpler method for creating palettes.

The objective of the paper by Baveye et al. [6] was to allow recoloring of images to more harmonious palettes. The research problem was implementing an algorithm that considered three areas: salience, efficient color segmentation and new color mapping. Earlier work had already started work on defining harmonious palettes, and applying them to content, but Baveye et al. focused on restraining their limitations

Jonah et al. [3] aimed to create a new visual salience model to accurately predict where the human eye fixates on an image. Graph-Based Visual Salience (GBVS) model shown in the paper is a two-step model, where first an activation map of the image is formed on certain feature maps, and then normalized to highlight the most prominent areas. This research and model was done since the previous salience models were not accurate enough.

Similar to our goals, Volkova et al. [7] strived to create an automatic web page re-colorization for web design. The new color palette would be picked from the picture given by the user. To recolor a webpage utilizing the algorithm provided, it is required to edit .CSS, .html and .svg files. Previous work had allowed extraction of color themes from an image, but no tool existed for recoloring an entire site. While our tool has similar elements, we are trying to create a low-fidelity design, which can be slotted into different web development practices, and requires minimal user input. In Volkova et al. design, the colors extracted from the image are not modifiable, and you in general have less control over the outcome.

Lastly to highlight what the future of estimating user attention, it is worth considering the Unified Model of Salience and Importance (UMSI) [8]. UMSI is a deep learning model which predicts visual importance, so compared to GBVS, it handles novel images better. This was also the main motivation for the tool, and the evaluation in the paper [8] show that UMSI reaches closest to the human perception of an image when compared with salience and importance models. We will not be using this tool in our paper because of time constraints, but it is worth noting that for a more accurate prediction, this tool should be considered.

## SALIENCE IN WEB DEVELOPMENT

The main purpose of salience models is to predict where the users gaze will be drawn to first. While there are a variety of different salience models available, such as Itti and Koch, Graph Based Visual Saliency (GBVS) [3] model was chosen for this tool for its ease of execution, superiority compared to most models and existing implementations. While there have been new models that have surpassed GBVS, such as UMSI [8], GBVS is still accurate enough to be considered a good fit for our purposes.

Since salience models are not something that many software developers are familiar with when creating a prototype or a color scheme for a website, valuable information about how the user might behave with the current design is lost. Harmonious color schemes are something you see more and more in modern web design, so it would be natural to utilize a tool that assesses whether you are using this scheme correctly compared to the design layout. Even if these developers and designers were convinced to utilize salience models, the current method of utilizing such models is very clumsy, as portrayed in figure 2.

DevOps is a software development philosophy, where the company aims to automate and as much of the products lifecycle as possible [1]. The development practice has gained a lot of traction and is used by Companies such as Adobe and Netflix [10]. The tool presented in this paper is tailored towards DevOps practices and can be used to automate color palette estimation inside a source code repository (such as GitLab) with minimal input. The tool can be used to estimate the most efficient prototype color assignment for highlighting a specific

element or finding the most efficient color palette and its best utilization from a variety of options.

## DESIGN OF THE TOOL

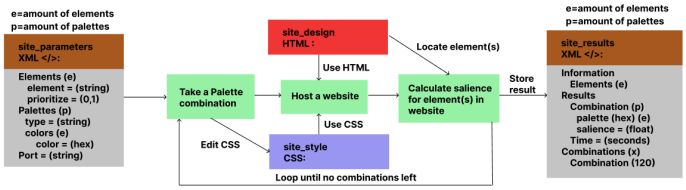


**Figure 3. The simplified design of the tool**

In figure 3 we can see the simplified design of the tool. The tool requires the user to give it the HTML and CSS files of the site formatted in the following way: CSS should have the editable colors as variables in the beginning, HTML should have the id: s of controlled elements defined in a similar fashion as in the site_parameters XML file. The XML file site_parameters includes the element names and whether specific elements should be prioritized in the design, harmonious website palettes defined by the user with a unique type, which is used to identify that palette, and the available port the design can be hosted to during the run-time.


**Figure 4. Figure of how we extract the salience from the website. Might have to edit this figure again.**

Figure 4 shows us how we extract salience of a specific element using GBVS. The GBVS implementation [4] created by Harel et al. allows us to turn the screenshot (on the left) into a salience map (on the right). As we can see, the white areas show the most prominent features of the design, in this case the side content and header of the site. The green rectangle around an element symbolises the area that is pinpointed from the HTML and parameterized for calculations.

The tool allows for a maximum of 5 different unique elements defined in the HTML, which are prioritized based on the given parameters. In figure 4 we are prioritizing for the sidebar and the buttons, so we extract their values from the salience map, and use the equation 1 to calculate a specific elements salience. The readability of the elements is not considered due to time constraints, since they can be easily changed by the user to be more readable from the newly colored elements. However, this was a problem with some palettes used, and should be on the short list of things to address if this tool was to be further developed. This can be done with existing Python libraries, such as GIVE AN EXAMPLE.

$$S_e = \text{Saliency of element},\ S_c = \text{Brightness of given coordinate}$$

$$S_e = \frac{\sum_{x=y1}^{y2} \sum_{y=x1}^{x2} S_c(y,x)}{(x2-x1)\cdot(y2-y1)} \tag{1}$$

In equation 1 we can see how each elements salience is calculated. We add the brightness of a every coordinate in a given area and divide them by the number of coordinates considered. This will result the average salience of an element. If there are more than a single element to consider, the salience of those elements is added together, and then divided based on the number of elements.

Its not meaningful to try every combination of every palette because just from their HEX codes we know that these values will have a poor salience when next to each other, or poor aesthetic value in the eyes of the user. Therefore, some optional rules have been set when choosing which combinations to try. Each site should have defined a background. A background element will either be the closest value to black or the closest value to white. This is because in the early iterations of the tool, it seemed to prefer putting the "shock" colors as the background, and the more reserved as the elements. While this also draws attention to the required elements, it is not seen as aesthetically pleasing.

These are optional rules, because they can be broken by not naming an element as the background. In this case, the tool will look trough every combination it can. This can help the designer or developer find an alternative design that works better, but requires more calculation time, and can result in an invert of the option that the base rule design would have chosen. Hence its only recommended as an experimental feature.

After execution of the tool, the output is the site_results XML file. This file contains the initial information given, each combination from each palette, and the best possible designs and their respective salience values. The values between each palette are directly comparable, and hence can be used to estimate how well a specific palette did compared to others. The tool also stores pictures of the website and their GBVS salience maps using the best possible combinations for each palette. These pictures are shown in both figure 1 and figure 5.

## EVALUATION

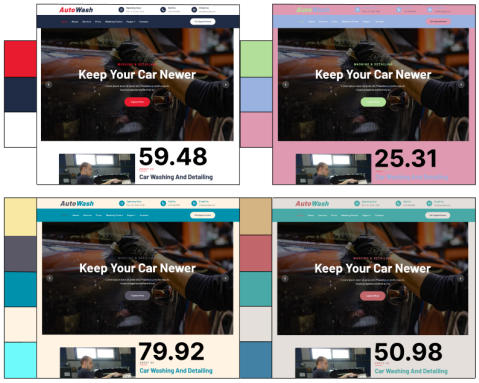


**Figure 5. Best combinations for four palettes with their relative salience values displayed when prioritizing buttons and navigation header**

Figure 5 shows us the outputs of four different harmonious palettes. These palettes were prioritized based on the navigation header and the button in the center of the screen, in a similar manner as figure 4 highlights specific elements. The pictures display the used color palettes on the left of the image (upper ones had three colors, the ones below had 5 colors to choose from). The sites original design used three colors and is displayed on the upper left corner of figure 5. Therefore, when presented with five colors, the tool chose the three most prominent one's trough iterating different permutations.

The original designs color palette was given to the tool, and the output was the same color placement as in the original one. On the top right corner, we can see another design with only three colors, but this was a lot less successful compared to the original design. This is displayed with the number at the bottom of each design. These numbers represent the average salience score of the site when prioritizing the central button and the navigation header. According to this value, the original design was more than two times more effective than its alternative harmonious design.

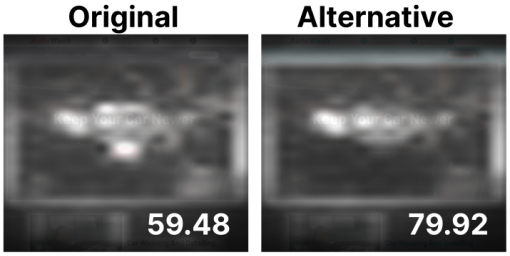


**Figure 6. The original design and the bottom left design from figure 5**

The bottom two options on figure 5 represent palettes generated using harmonious color pickers similarly to those mentioned in the related work [2]. The bottom left design was more effective at getting the users attention to the prioritized area when comparing to all the other options, even the original design. The original and the bottom left designs salience maps are displayed in figure 6. Here we can see that the source for the alternative designs improved score over the original was the more noticeable navigation header.

The time the tool needs to execute varies depending on the number of elements, palettes and the size of the website being considered. On average the tool takes around 5 seconds per element. After running a variety of palettes and sites, I would give an estimate of around 5 minutes per palette when there are two elements being prioritized, and roughly 2 minutes per palette when only one.

Setting up the tool on an existing site takes roughly an hour, depending on how familiar the implementer is with the tools design. It requires minimal tweaks to work well and can easily be implemented as a part of the build process for a website, even thought it should be only run when there are noticeable changes in the site design or color palettes given.

## DISCUSSION

As we can see from the figure 1 and 5, the tool is successful at utilizing aesthetically pleasing palettes at different sites. However, in order to truly get an accurate estimation about the tools effectiveness, I would suggest consulting designers about the results. This could be done by asking designers to assign the same color palettes as the tool uses to the same websites to highlight specific elements. After that, they could be asked to rate the tools designs for other color palettes. Lastly, we would compare the designers color assignations to the tools, and if those designs were different, gather feedback whether the designers thought the tool was successful than they were at the task.

I would estimate that the designers would implement similar designs as the tool, since as shown in figure 5, our tools design was the same as the original designers who created the website template, when given the same color palette.

This tool is relatively easy to use but transporting it across different designs could always be more efficient. This could be improved with more sophisticated analysis of CSS files and HTML. We could also expand this to work as a plugin, so it could be utilized in any website, and the output would be a CSS file with the optimized color palette. Related to the design choices, an area of improvement could be to allow more elements to be considered, or allow creating a group of elements, such as all the buttons on the site. Currently when you give a button to be prioritized, unless they are defined under the same container, only one of them will be considered in the calculation.

To improve the tools accuracy, we could change the estimation model from GBVS to UMSI [8]. Since the tool uses GBVS to only analyse salience maps, these could be replaced by the heat maps of UMSI. I would estimate the effort required for this to be well worth the payoff.

## CONCLUSION

In this paper we presented a tool that can be used to generate a variety of designs for a given site based on harmonious color palettes. The tool allows the designer or developer to choose what elements they want to prioritize using the GBVS

model. Since the tool requires some manual definitions in HTML and CSS level, it leaves the developers with the power to utilize the tool how they see fit, since the same color can be applied in multiple different areas, while only prioritizing one, leading to a more natural design, as seen in figure 5. Our tool is tailored to work in DevOps manner, focusing on automating an otherwise manual task. It can be incorporated to the developmental process of a website prototype or a testing environment, where it can be tailored to suit the needs of the product.

Further areas of improvement include expanding the evaluation phase to suit designers feedback. Designers could be used to evaluate designs made by the tool, and compare their own color assignments to what our tool created. Design could be expanded to allow for a wider variety of websites, for example implementing the tool as a plug-in for a browser.

### FEEDBACK TO CONSIDER

### REFERENCES


[1] Atlassian, "What is DevOps? | Atlassian," Atlassian, 2016. https://www.atlassian.com/devops.

[2] G. Hu, Z. Pan, M. Zhang, D. Chen, W. Yang, and J. Chen, "An interactive method for generating harmonious color schemes," Color Research  Application, vol. 39, no. 1, pp. 70–78, Aug. 2012, doi: 10.1002/col.21762.

[3] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Salience," 2007. Accessed: Jan. 28, 2022. [Online].

[4] J. Harel, C. Koch, and P. Perona, "Implementation of Graph Based Visual Saliency Algorithm," GitHub, 2019. https://github.com/shreelock/gbvs (accessed 2022).

[5] "IBISWorld - Industry Market Research, Reports, and Statistics," www.ibisworld.com, 2022. https://www.ibisworld.com/industry-statistics/market-size/web-design-services-united-states/.

[6] Y. Baveye, F. Urban, C. Chamaret, V. Demoulin, and P. Hellier, "Salience-Guided Consistent Color Harmonization," Springer, Berlin, Heidelberg, 2013. Accessed: Jan. 28, 2022. [Online].

[7] P. Volkova, S. Abrishami, and P. Kumar, "Automatic Web Page Coloring," Advances in Visual Computing, 2016, doi: 10.1007/978-3-319-50835-1.

[8] F. Camilo, C. Vincent, B. Amish, O. Peter, H. Aaron, and B. Zoya, "Predicting Visual Importance Across Graphic Design Types," pp. 249–260, 2020, doi: 10.1145/3379337.341582.

[9] Adobe, "Create and discover unique color themes | Adobe Color," www.adobe.com. https://www.adobe.com/products/color.html (accessed Feb. 21, 2022).

[10] sYstango, "Blog," Systango, 2021. https://www.systango.com/blog/5-companies-that-are-nailing-it-with-devops/ (accessed Apr. 18, 2022).