Online Submission ID: 4276

$\mathbb{C}^3$-palette: Co-saliency based Colorization for Comparing Categorical Visualizations

– Supplementary Material –

Category: n/a

This supplementary material provides additional experimental results for our submitted paper titled "$\mathbb{C}^3$-palette: Co-saliency based Colorization for Comparing Categorical Visualizations".

Scatterplots with Significant Overlap. Since our contrast measures are built on the data-space nearest neighbour graphs, its produced palettes also work well for scatterplots with significant overlap between classes. For example, as shown in the results from *Random Assignment* in Fig. 1, the light orange class is mixed with the light purple class and the teal class has a strong overlap with the grey class, they are hard to distinguish. The *Alpha Blending* result is based on *Optimized Assignment* [2] with Tableau palette, however, due to the changed opacity, classes are hard to split from each other and composite new colors. The results generated from *Palettailor* and *C^3-palette* both have a well separability since they are all built on the data-space nearest neighbour graphs. Specifically, *C^3-palette* highlights the changed classes(classes with green, purple and red) while maintaining discrimination of strong overlapped classes, such as the blue and yellow, red and green classes.

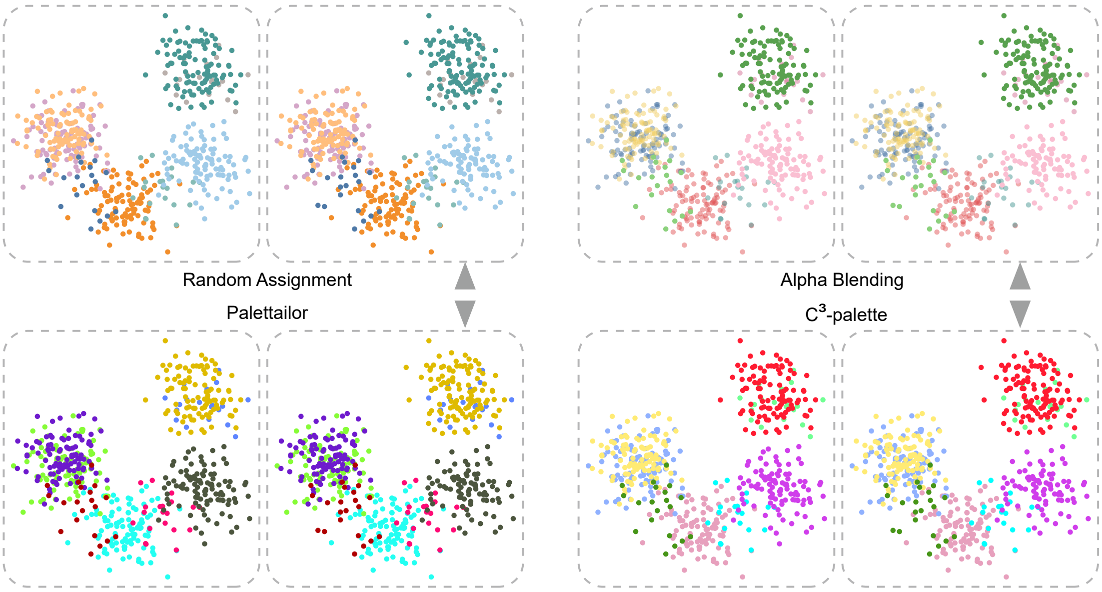

Figure 1: Results for scatterplots with significant overlap: (left) Tableau palette with random assignment versus Palettailor [1]; (right) Alpha Blending versus C^3-palette. Our co-saliency methods (right bottom) can highlight the changed classes while maintaining discrimination of strong overlapped classes.

Bad Case for Alpha Blending on Different Visualizations. In the previous section, we show that overlapped scatterplot points would composite new colors from the alpha blending method. Here we show alpha blending is insufficient even for bar chart and line chart. Fig. 2 shows an example, where the left top two images show our auto-generated results and the left bottom are the alpha blending result based on Tableau 20 palette by maintaining the opacity of the bar with the largest difference being 1.0 while changing others' opacity to 0.4. We still hard to find the largest change immediately. Similarly, this effect exists in line chart as shown in Fig. 2 right. Due to the low contrast against the background, the yellow line is hard to pop out even the opacity of other classes set to 0.2.

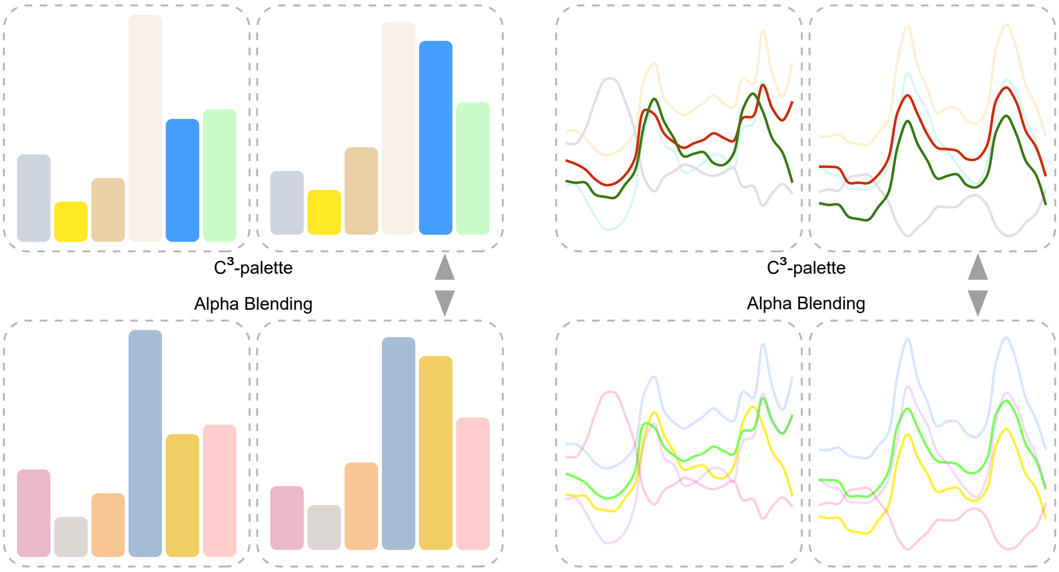

Figure 2: Illustrating the problem of alpha blending. (left) C^3-palette versus Tableau palette with alpha blending for bar charts; (right) C^3-palette versus palette contains low contrast color with alpha blending for line charts.

Pilot Study Details and Statistics. We conducted two pilot studies for the scatterplot experiment, one for *identifying delta task* and the other is for the *counting class task*. The statistics results are shown in Fig. 3. As for *identifying delta task*, we recruited 28 people (see Table. 1) through the Amazon Mechanical Turk. The power analysis is executed between C^3*-palette Generation* and *Random Assignment*. With an effect size Cohen's d of 0.4, alpha level of 0.05 and beta level of 0.8, the power analysis suggested a minimum number of 100 participants for the spot-the-difference task. The procedure of *counting class task* is similar to the previous one. We recruited 29 participants from AMT whose approval rate is larger than 97%. Since we mainly want to compare the class discriminability between C^3*-palette Generation* and *Alpha Blending*, we executed power analysis based on these two conditions. With an effect size Cohen's d of 0.6, the power analysis indicates that we need 50 participants for this task.

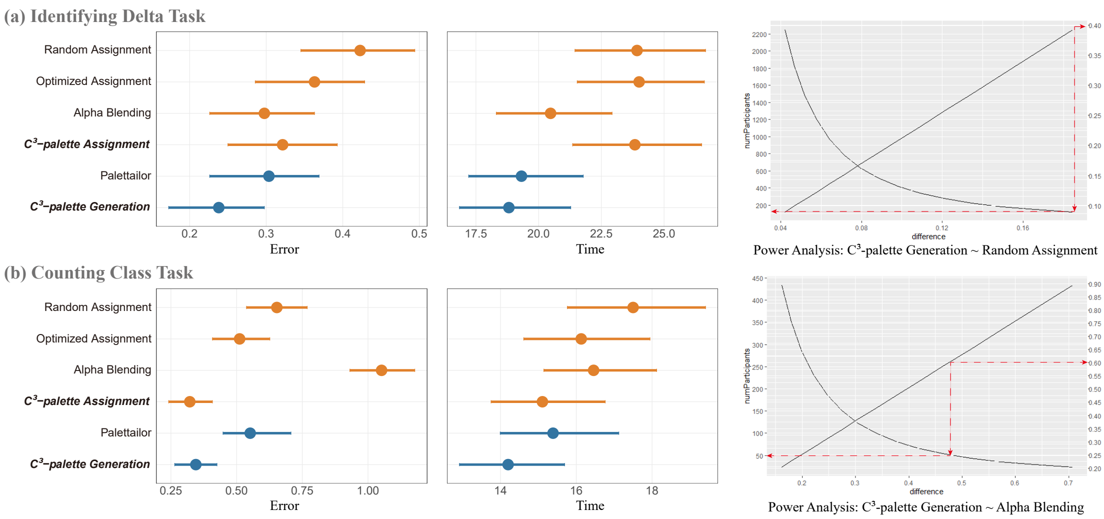

Figure 3: Confidence interval plots and power analysis for the two pilot studies.

Table 1: Participants details for each task of the scatterplot experiment.

Task & Group	identifying delta task		counting class task	
	Pilot(28)	Formal(108)	Pilot(29)	formal(52)
Group 1	5	18	5	9
Group 2	5	17	5	8
Group 3	5	19	4	8
Group 4	3	17	5	9
Group 5	5	19	5	9
Group 6	5	18	5	9

Formal Study Statistics Results. Due to the limited space, we moved the detailed results to supplementary materials. The statistics are shown in Fig. 4.

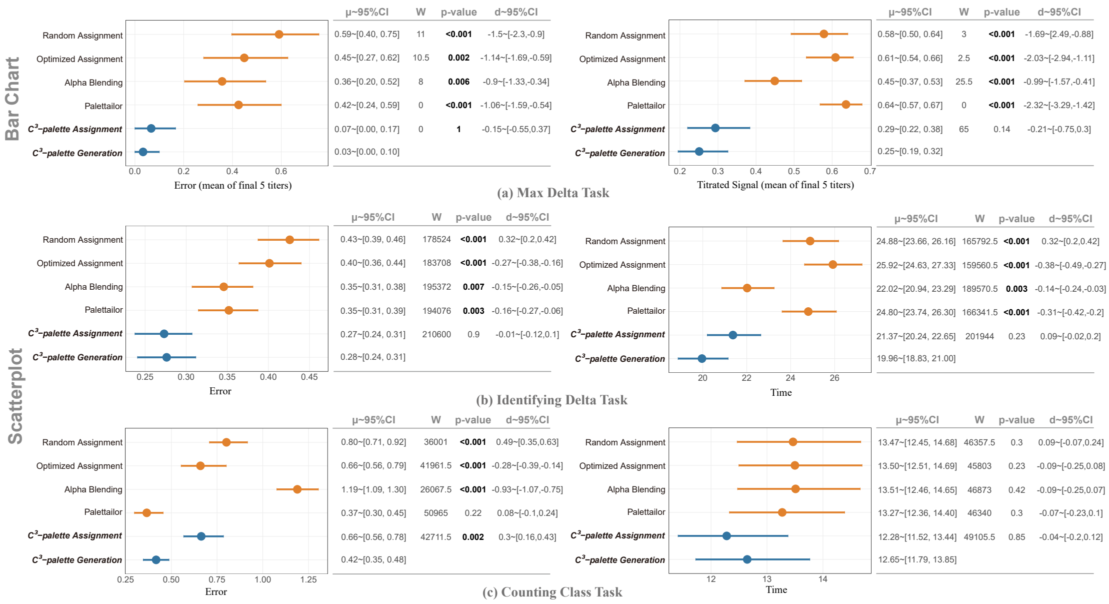

Figure 4: Confidence interval plots and statistical tables for the online controlled experiment. Error bars represent 95% confidence intervals. Each table shows the statistical test results of C^3*-palette Generation* condition compared to the other conditions, including the mean with 95% confidence interval (μ ~95%CI), the W-value and p-value from the Mann-Whitney test, and the effect size (d ~95%CI).

Hue-preserving Palette Generation. We also implemented the hue-preserving palette generation which is achieved by adjusting saturation and lightness while maintaining the hue of each color. However, this method cannot produce satisfactory results. As shown in Fig. 5 (a), for a single hue value, there exist many different colors. For example, with the same hue (50), we can get black, yellow, grey, brown, etc. Applying this method into our optimization process leads to different colors. Fig. 5 (b) shows two examples for different visualization types: the left top scatterplots were generated by default settings, then we maintained the hue of the yellow class in the bottom right and generated new palette, the yellow class changed to grey; this process is similar to line chart, while we maintain the hue of the brown line and finally it changed to grey too. This method is not consistent for user exploration. Hence, we choose color name constraint to maintain consistent color schemes.

Palette Generation for Color Vision Deficiency. To help people with a color vision deficiency, we allow users to generate palettes that can be used for different types of vision problem, such as protanomaly and deuteranomaly which result in poor red-green hue discrimination. This is achieved by adopting a color blindness simulator (the source code can be found at GitHub `https://github.com/MaPePeR/jsColorblindSimulator`) and then used our matrix for palette evaluation. Fig. 6 shows an example, where the left two images show the auto-

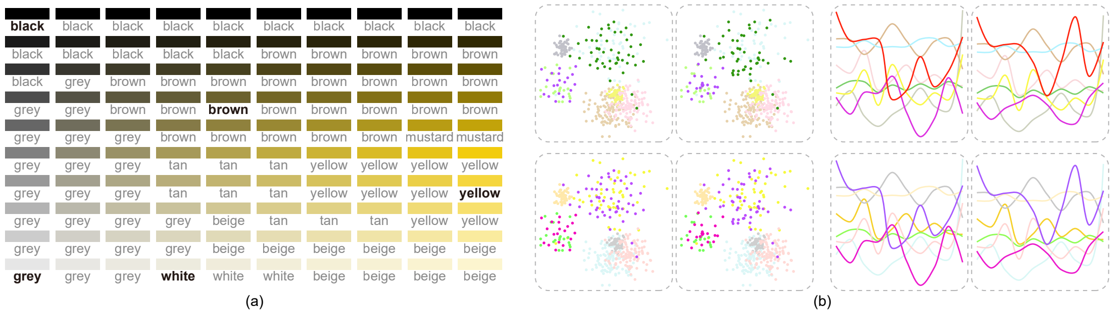

Figure 5: Hue-preserving palette generation. (a) $Hue = 50$, quantizing the HSL color space into 100 discrete colors by sampling every 10 units along saturation and lightness axis starting at the origin, their color name is below each color; (b) Palette generation results for different visualization types (i.e., scatterplot and line chart): (top) result with default settings; (bottom) hue-preserving result for limit the hue of the yellow cluster in the top scatterplot and brown line in the top line chart.

generated results and the right are the simulated results perceived by people with protanomaly. That is, the purple and red classes seen by normal vision will be perceived as dark blue and dark purple by protanomaly. We can see that our results are easy to find changes between scatterplots for people with color vision deficiency and it is still can be distinguished by normal vision. This preliminary result proves that our method can integrate start-of-the-art color blindness simulation algorithm to generate palettes for color vision deficiency.

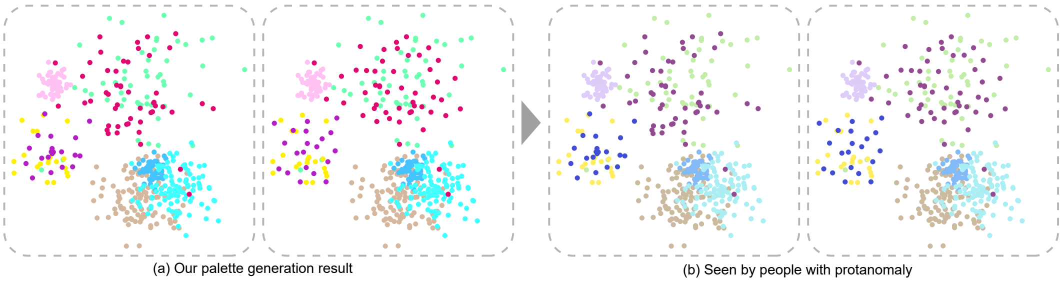

Figure 6: Exploring the ability of our system to generate palettes for both people with normal vision and color blindness. (a) The automatically generated palette makes the two changed classes with large saliency while maintains good separability between other classes. (b) Simulated results which is seen by people with protanomaly. We can see our results maintain a good performance for both people with normal vision and color vision deficiency.

References

[1] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang. Palettailor: discriminable colorization for categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):475–484, 2021. doi: 10.1109/TVCG.2020.3030406

[2] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C.-W. Fu, O. Deussen, and B. Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):820–829, 2019. doi: 10.1109/TVCG.2018.2864912