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

ANONYMOUS AUTHOR(S)

ACM Reference Format:

ANONYMOUS AUTHOR(S). 2018. $\mathbb{C}^3$-palette: Co-saliency based Colorization for Comparing Categorical Visualizations . In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

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 data-space nearest neighbour graphs, their produced palettes also work well for scatterplots with significant overlap between classes. For example, as shown in the results from the *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, all of them are hard to distinguish. The *Alpha Blending* result is based on *Optimized Assignment* [2] with the Tableau palette; however, due to the changed opacity, classes are hard to split from each other and new colors appear. The results generated from *Palettailor* and *C^3-palette* have a good separability since both are built on data-space nearest neighbour graphs. Specifically, *C^3-palette* highlights 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.

Bad Cases for Alpha Blending on Different Visualizations. In the previous section, we show that overlapping scatterplot points would composite new colors in the case of alpha blending. Here we show that alpha blending is insufficient even for bar and line charts. Fig. 2 shows an example, where the two images on the top left show our auto-generated results, on the bottom left are alpha blending results based on the Tableau 20 palette. We maintained an opacity of 1.0 for the bar with the largest difference, while changing the opacity of the others to 0.4. It is still hard to find the largest change immediately. This effect also exists in the line chart shown in Fig. 2 right. Due to the low contrast against the background, the yellow line does not pop out visually even when the opacity of the other classes set to 0.2.

Pilot Study Details and Statistics. We conducted two pilot studies for the scatterplot experiment, one *identifying delta task* and a *counting class task*. The statistical results are shown in Fig. 3. For the *identifying delta task*, we recruited 28 people (see Table. 1) through Amazon Mechanical Turk. The power analysis was computed between *C^3-palette Generation* and *Random Assignment*. With an effect size Cohen's *d* of 0.4, an alpha level of 0.05 and a beta level of 0.8, the power analysis suggested a minimum number of 100 participants for this task. The procedure of *counting class task*

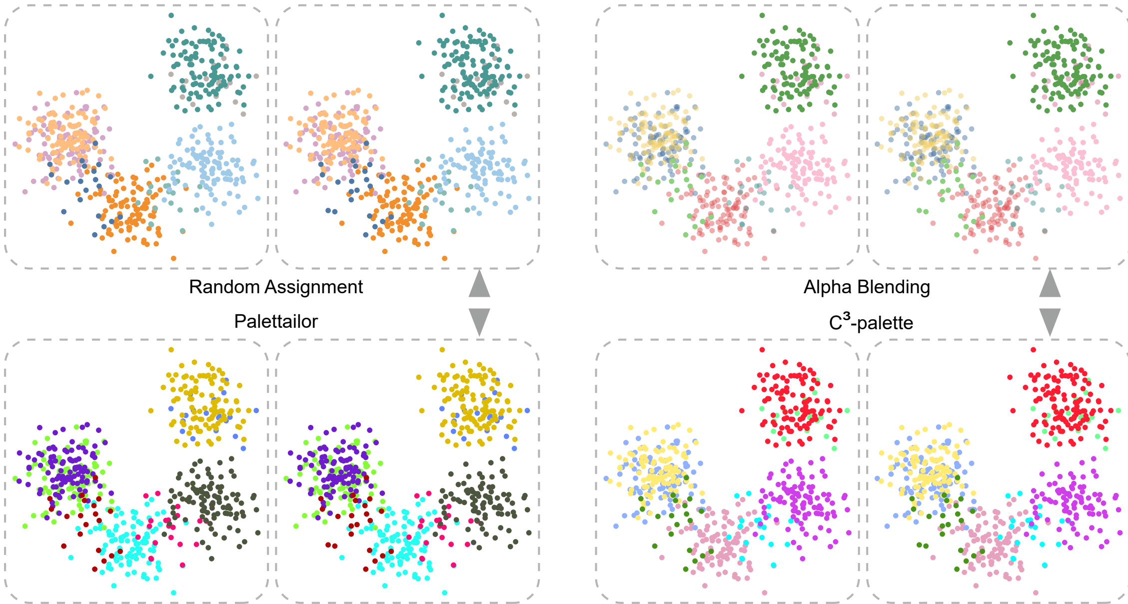

Fig. 1. Results for scatterplots with significant overlap: (left) Tableau palette with random assignment versus Palettailor [1]; (right) Alpha Blending versus C^3-palette. Our co-saliency methods (right bottom) are able to highlight changed classes while maintaining discrimination of strongly overlapping classes.

was similar to the previous one. We recruited 29 participants from AMT whose approval rate was larger than 97%. Since we mainly wanted to compare the class discriminability between C^3*-palette Generation* and *Alpha Blending*, we executed the power analysis based on these two conditions. With an effect size Cohen's d of 0.6, the analysis indicated that we needed 50 participants for this task.

Table 1. Participants details for each task within 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. Here we show the detailed results. The statistics are shown in Fig. 4.

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

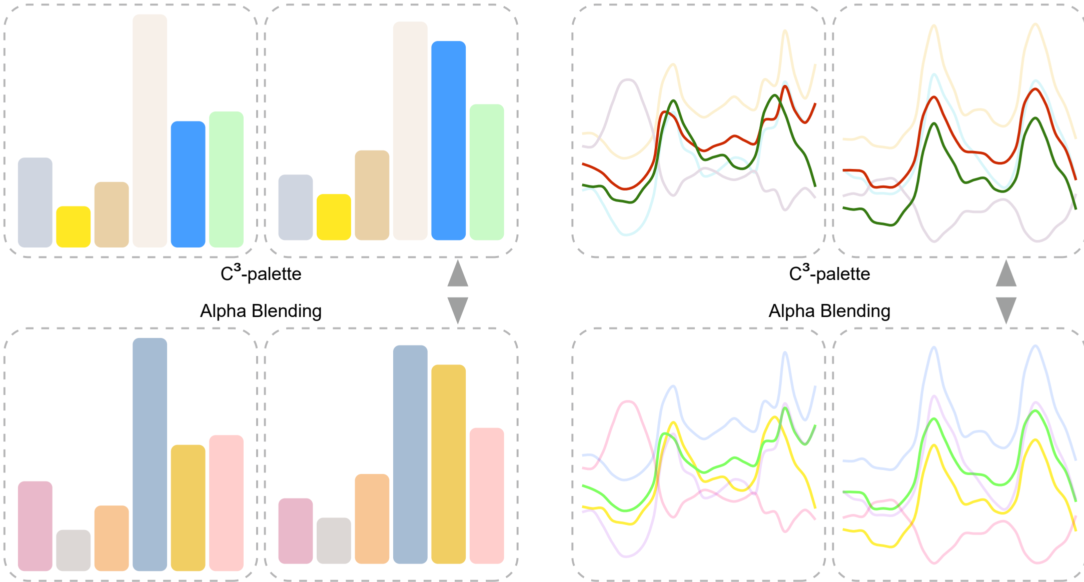

Fig. 2. The problem of alpha blending: (left) C^3-palette versus Tableau palette with alpha blending for bar charts; (right) C^3-palette versus a palette with low contrast colors using alpha blending for line charts.

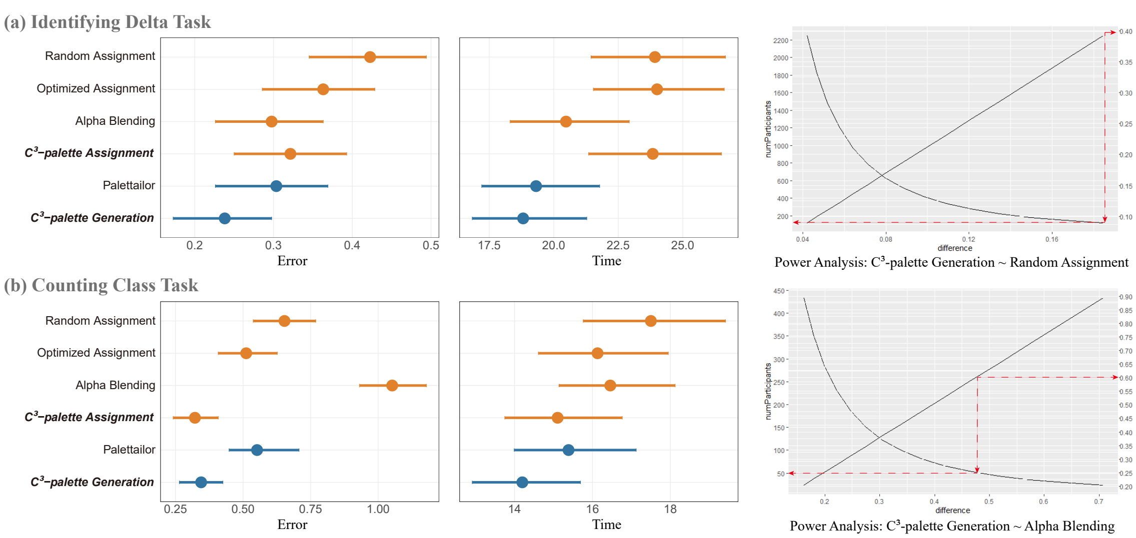

Fig. 3. Confidence interval plots and power analysis for the two pilot studies.

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

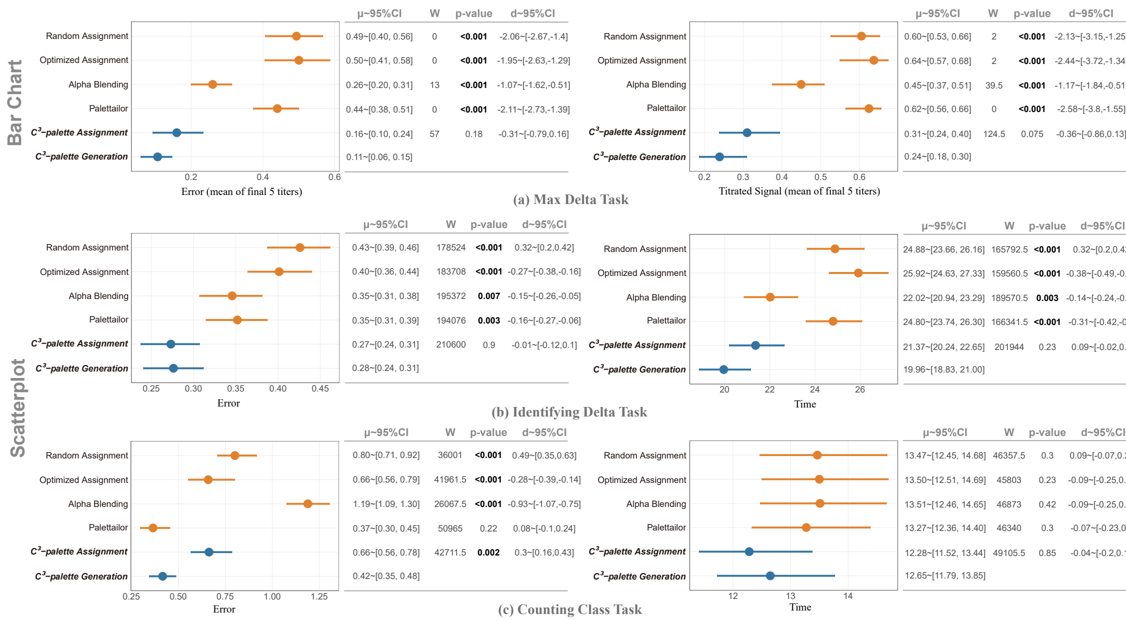

Fig. 4. Confidence intervals and statistical tables for the online controlled experiment. The error bars represent 95% confidence intervals. Each table shows the test results for the C^3*-palette Generation* condition compared to the other conditions, including the mean with 95% confidence interval (μ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

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.

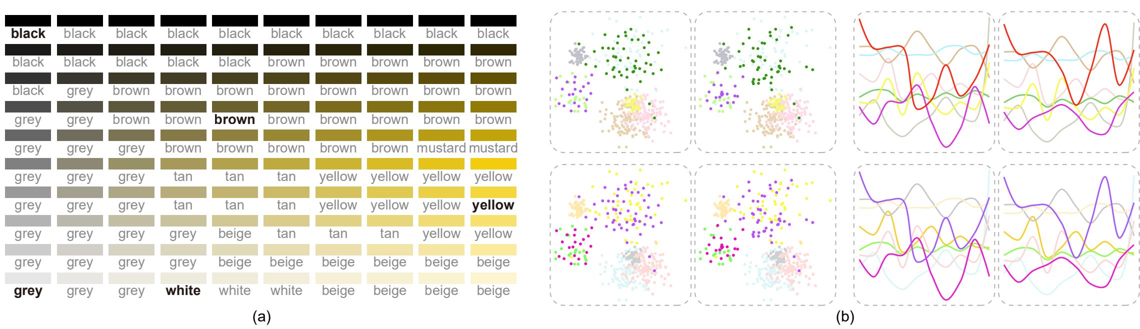

Fig. 5. Hue-preserving palette generation. (a) $Hue = 50$, quantizing the HSL color space into 100 discrete colors by sampling every 10 units along the saturation and lightness axis starting at the origin, their color name is given below each color; (b) Palette generation results for different visualization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

REFERENCES

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

[2] Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen, and Baoquan Chen. 2019. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 820–829. https://doi.org/10.1109/TVCG.2018.2864912