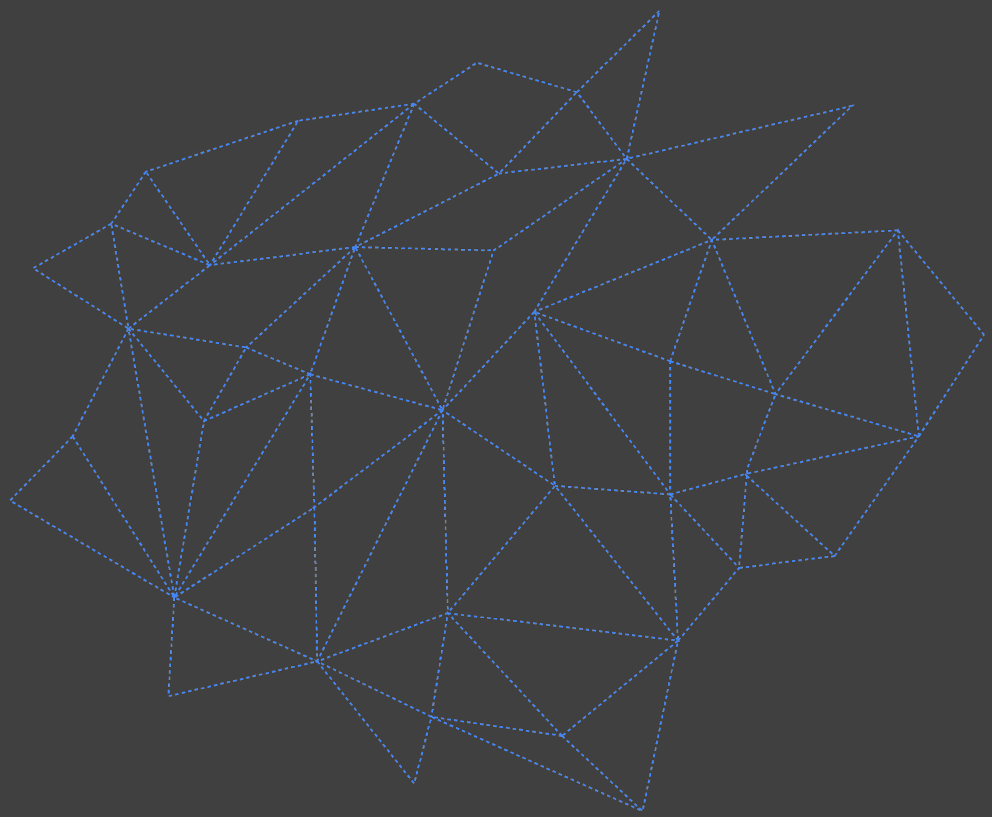
Effective Visualization (Color) Design



Connor Gramazio
@ccgramazio

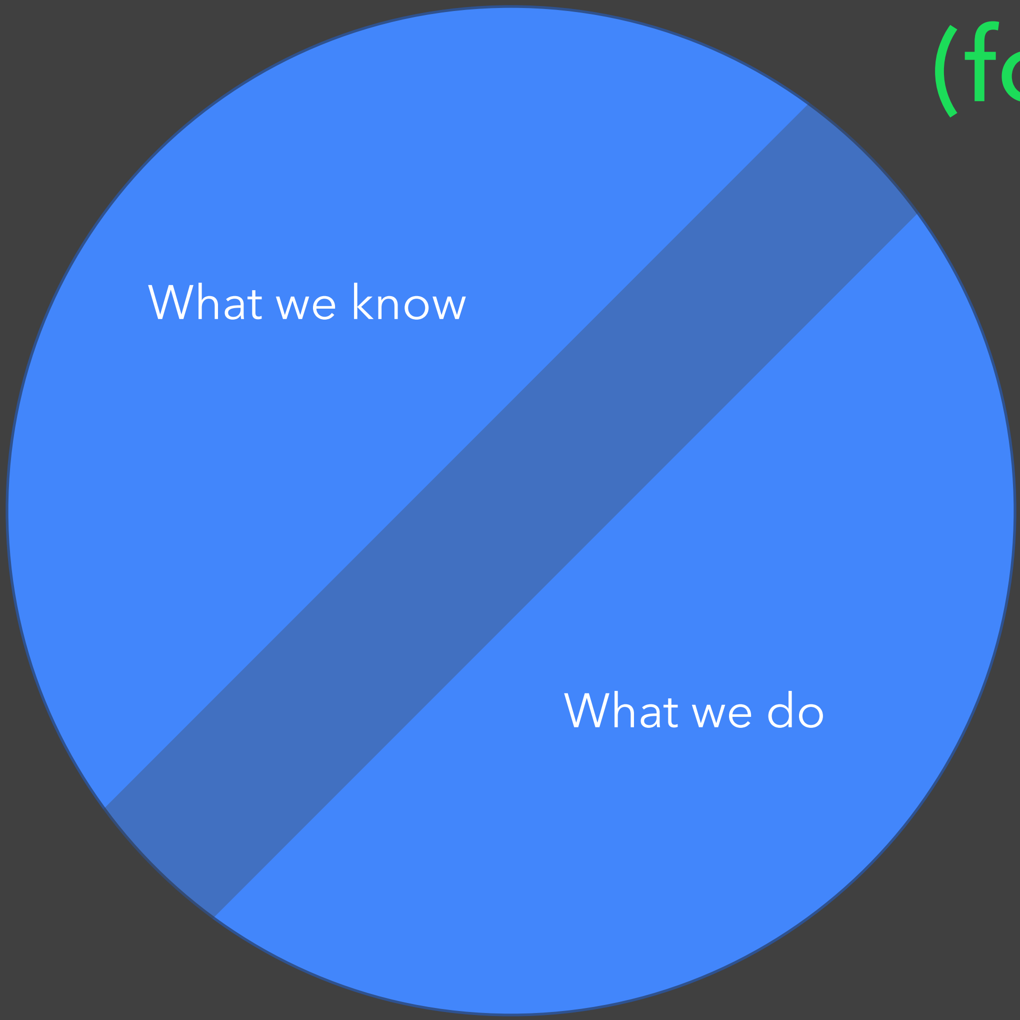
OVC 2017

https://youtu.be/DjJr8D4Bxjw

Aim: increase accessibility of theory → practice

Aim: increase accessibility of theory → practice *via application*

(for color)



d3-jnd

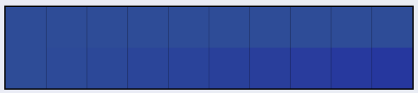
gramaz.io/d3-jnd

colorgorical

gramaz.io/colorgorical

d3-cam02

gramaz.io/d3-cam02



d3-jnd

gramaz.io/d3-jnd

colorgorical

gramaz.io/colorgorical

d3-cam02

gramaz.io/d3-cam02

What does it mean for

visualization design

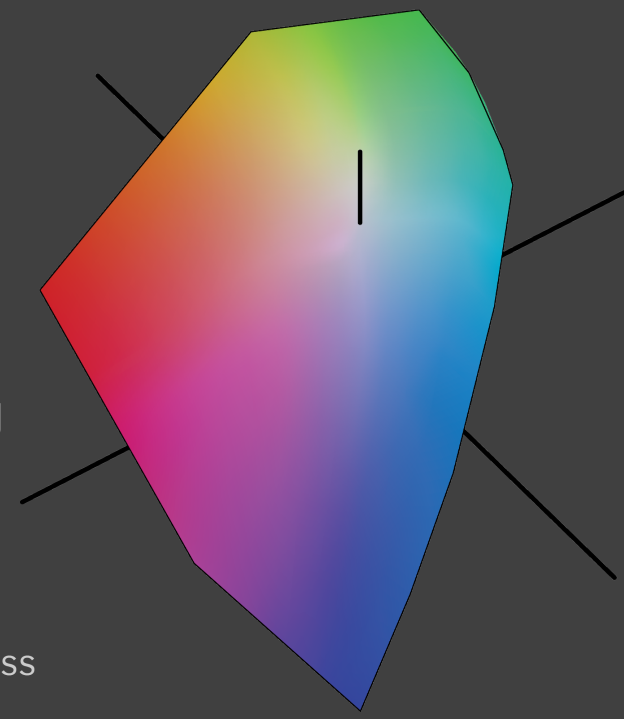
to be “effective”?

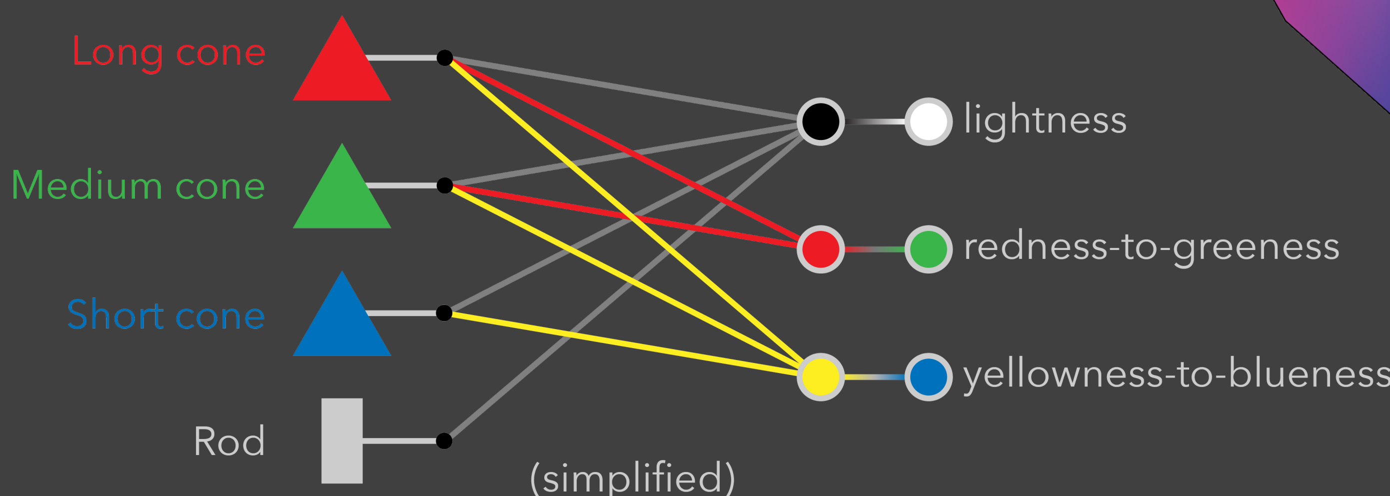
Color Discriminability

CIELAB perceptual color space

L*: lightness
a*: redness-to-greenness
b*: yellowness-to-blueness

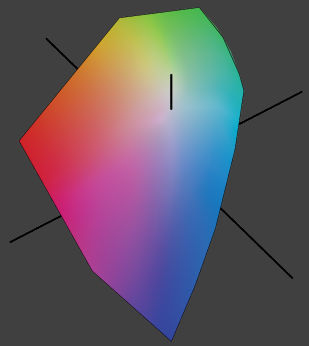
Approximates opponent color processing



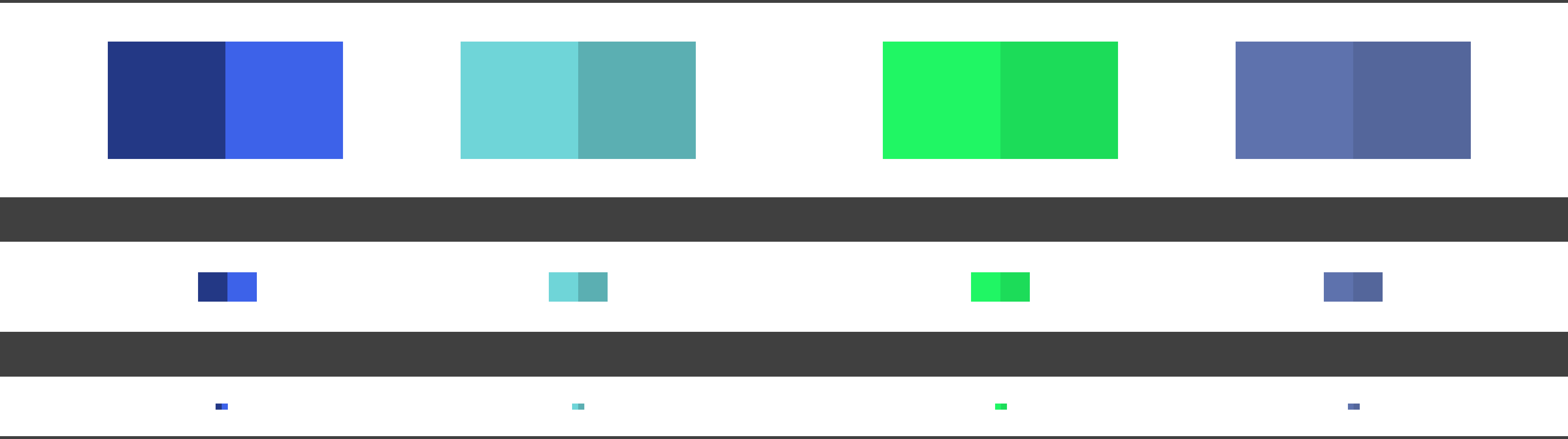


JND: Just Noticeable Difference

Minimum differentiable color distance
→ Many ways (+ papers) to define color JNDs



Color Discriminability: Distance in CIELAB



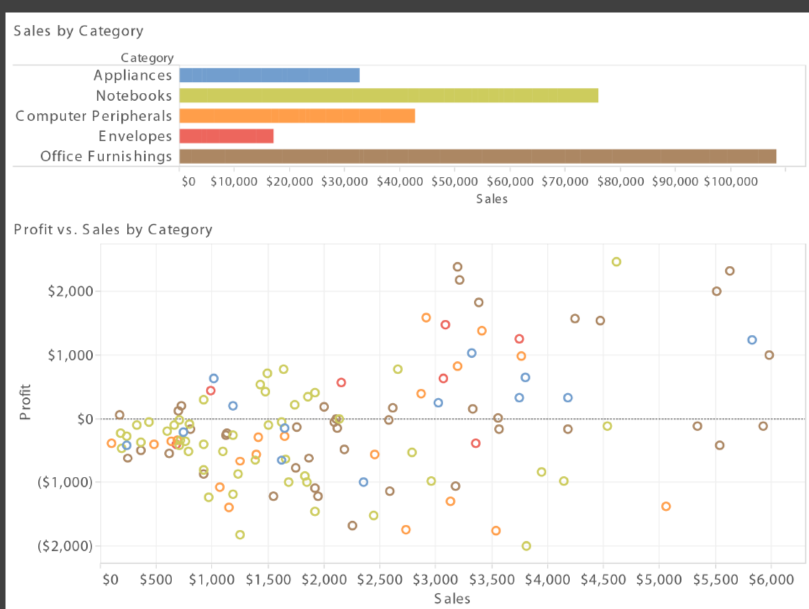
Color Discriminability: Distance in CIELAB + size

JNDs differ with size

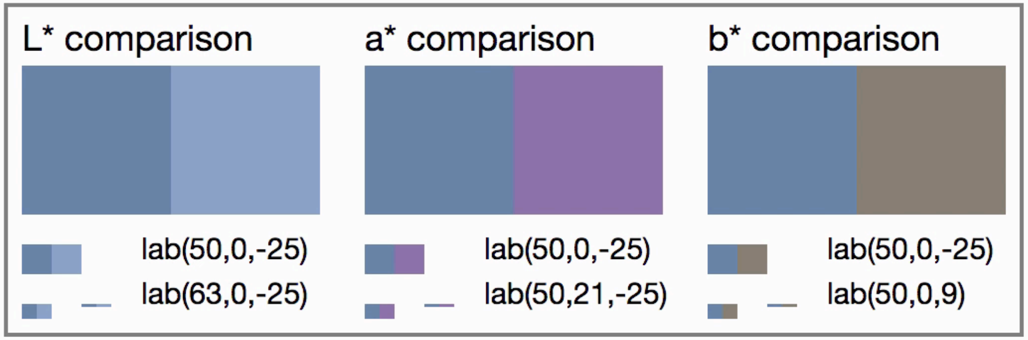
Stone, Albers Szafir, Setlur formalize relation in CIELAB

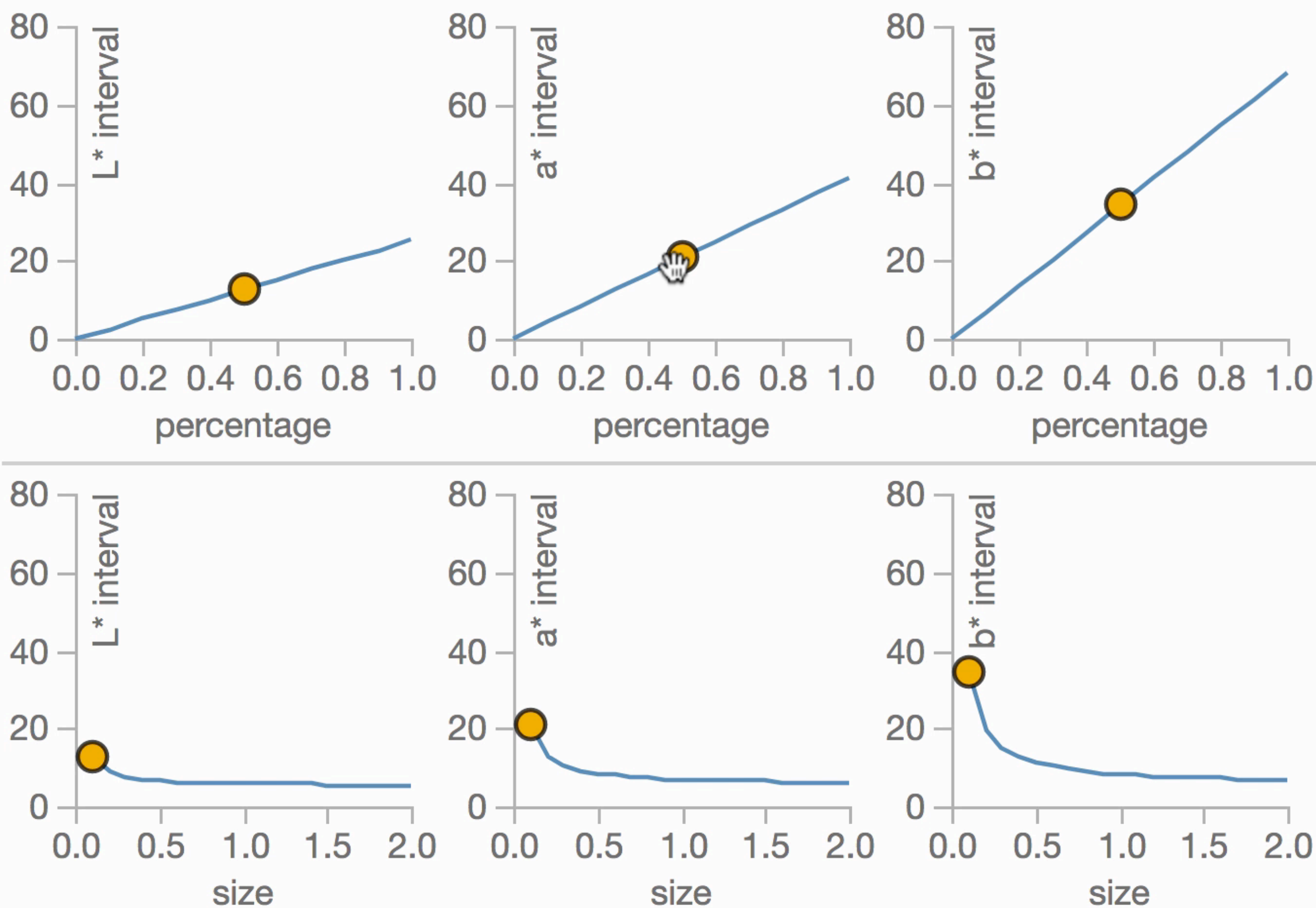
"An Engineering Model for Color Difference as a Function of Size"
Stone, Albers Szafir, Setlur. 2014. Link @ https://gramaz.io/d3-jnd

$$ND(\%,size) = \{\ \Delta L^*, \Delta a^*, \Delta b^*\}$$



Arg: Percent	Arg: Size	ΔL*	Δa*	Δb*
0.5	0.1	12.58	20.74	34.05





Apply JNDs+size in design?

1. Consider color-encoded area size in the graphic
2. How cautious do you need to be?
   (infographic vs. emergency response management)

```
d3.noticeablyDifferent(c1, c2[, percent, size]);

d3.jndInterval(percent, size); // { l, a, b }
```

d3-jnd
gramaz.io/d3-jnd

colorgorical
gramaz.io/colorgorical

d3-cam02
gramaz.io/d3-cam02

Joint work with
David Laidlaw
Karen Schloss

IEEE VIS 2016

discriminability

What do you do when you need to make your own categorical palette?

aesthetic preference

discriminability

Colorgorical applies color science to reduce design boundaries

aesthetic preference

Colorgorical Roadmap

Background

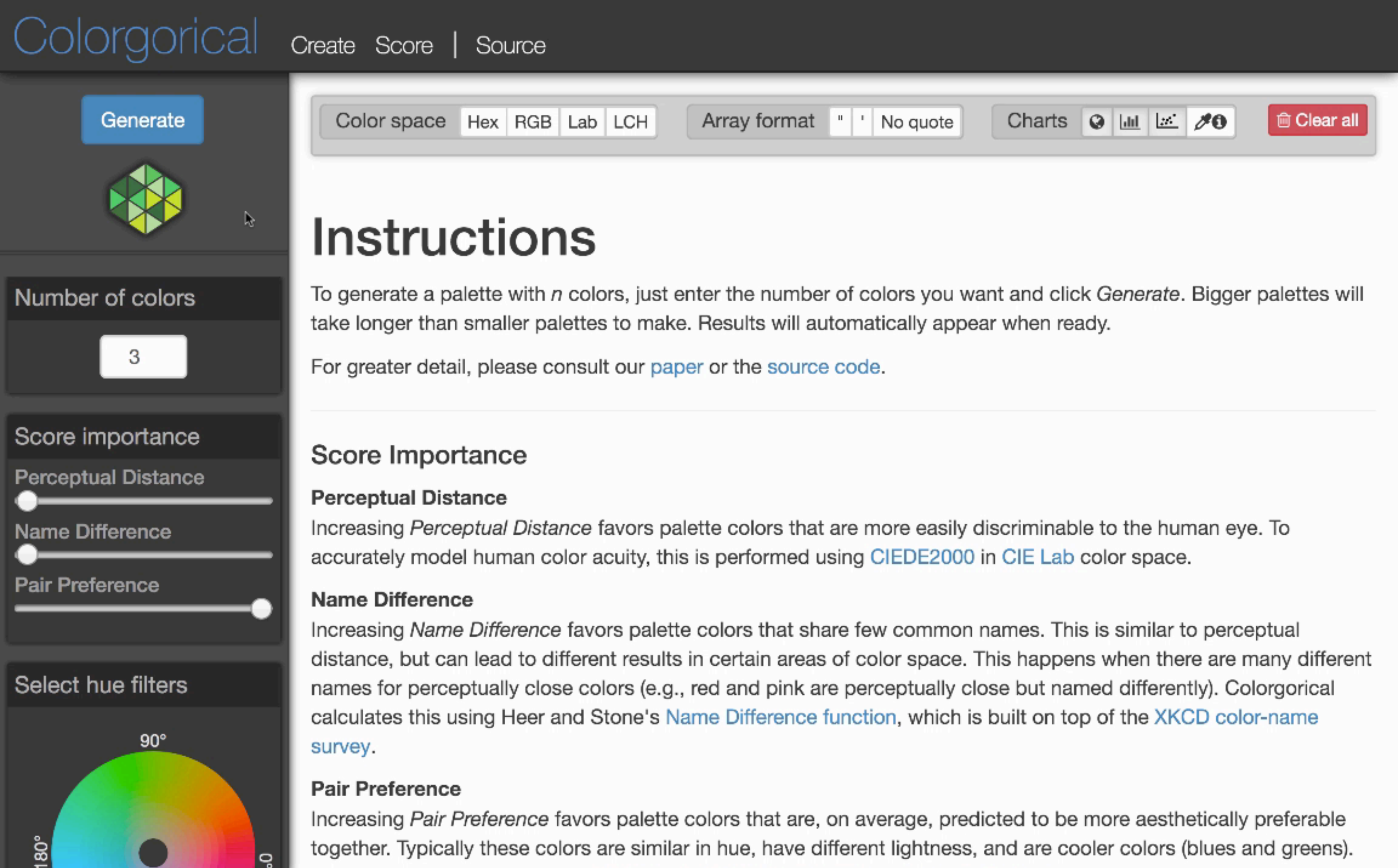
Colorgorical technique

Experiment 1: Does Colorgorical work?

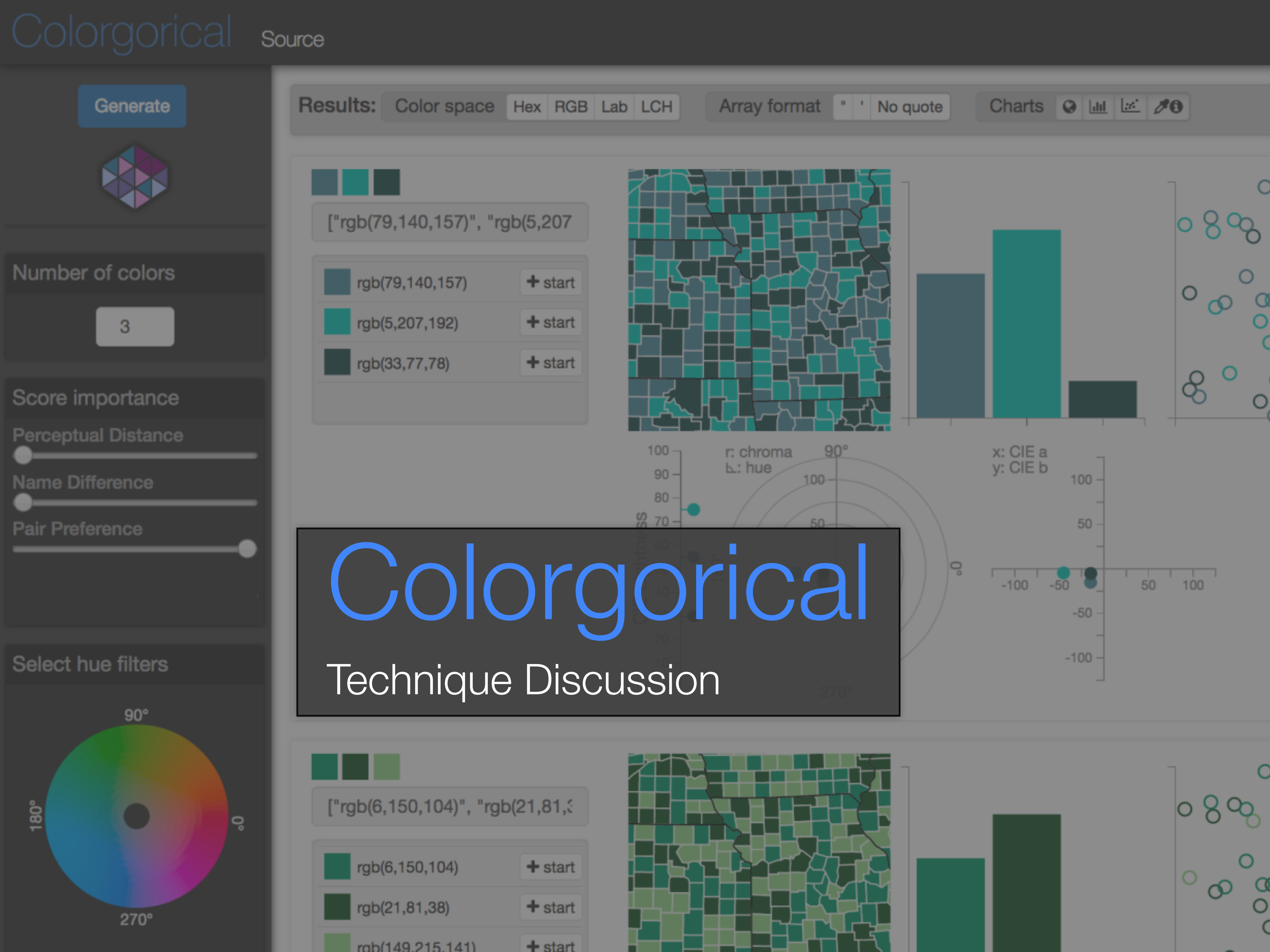
Experiment 2: Colorgorical vs. standards

Colorgorical

Example



Paper, Demo @ https://gramaz.io/colorgorical
Source @ https://github.com/connorgr/colorgorical


Colorgorical
Technique Discussion
Colorgorical
Source
Generate
Number of colors
3
Score importance
Perceptual Distance
Name Difference
Pair Preference
Select hue filters
90°
180°
0°
270°
Results:
Color space
Hex
RGB
Lab
LCH
Array format
No quote
Charts
["rgb(79,140,157)", "rgb(5,207
rgb(79,140,157)
rgb(5,207,192)
rgb(33,77,78)
+ start
r: chroma
b.: hue
x: CIE a
y: CIE b
["rgb(6,150,104)", "rgb(21,81,3
rgb(6,150,104)
rgb(21,81,38)
rgb(149,215,141)

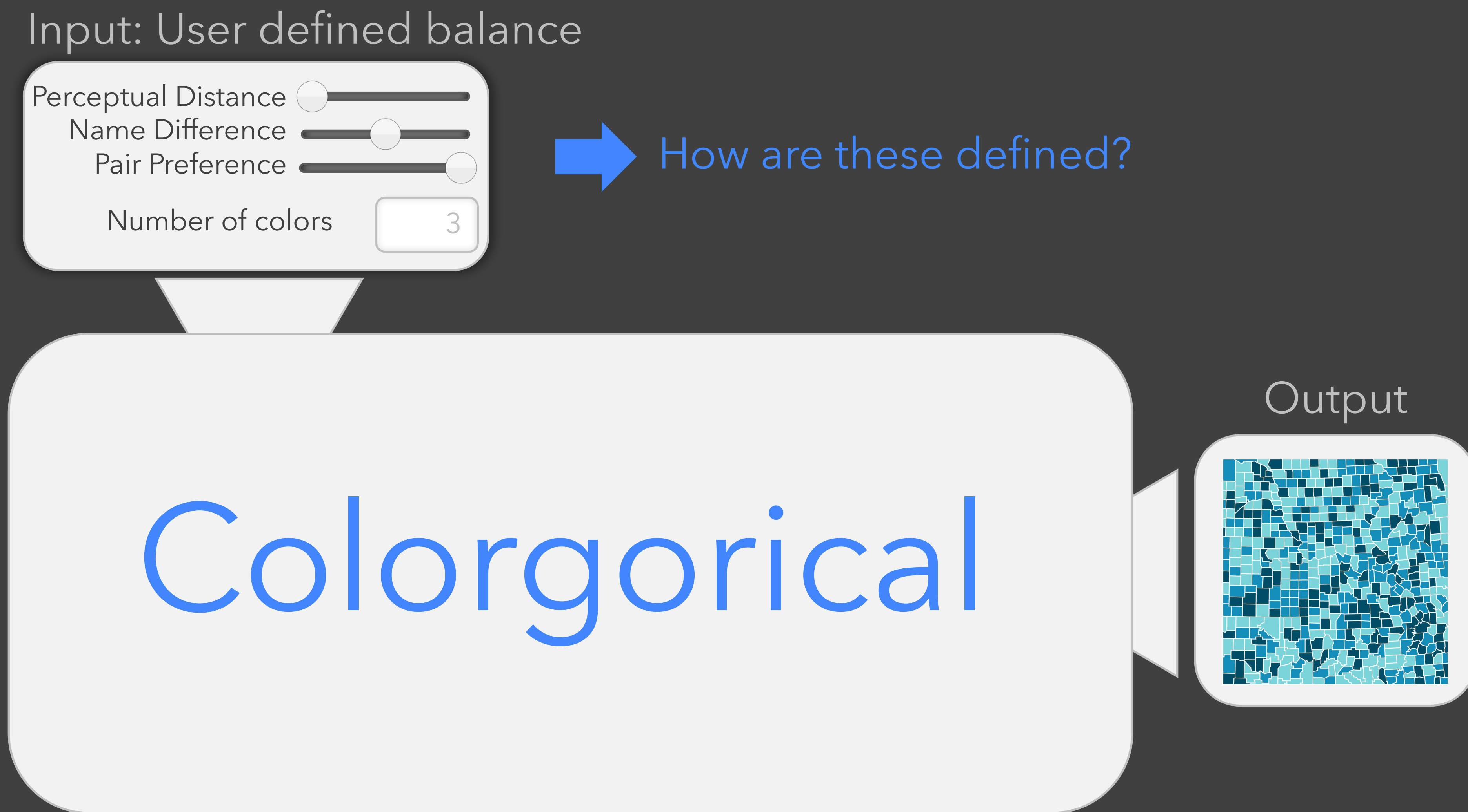
JS

python™

Tornado

NumPy + C functions

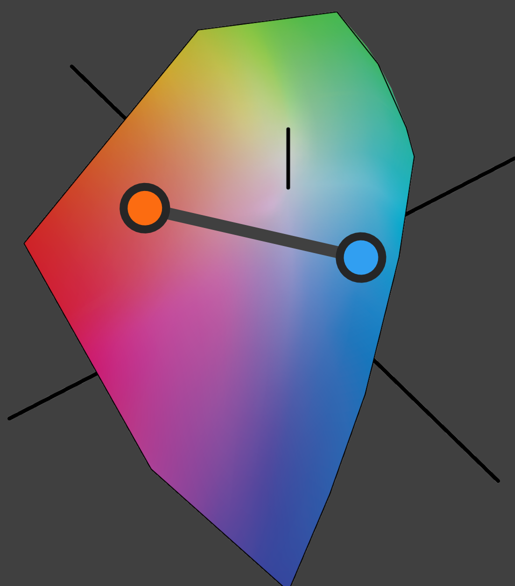
Paper, Demo @ https://gramaz.io/colorgorical
Source @ https://github.com/connorgr/colorgorical

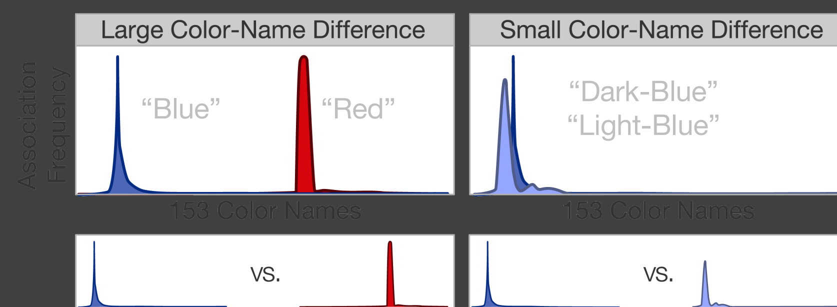

Input: User defined balance
Perceptual Distance
Name Difference
Pair Preference
Number of colors
3
How are these defined?
Colorgorical
Output

2 Measures of discriminability

Perceptual Distance

Name Difference

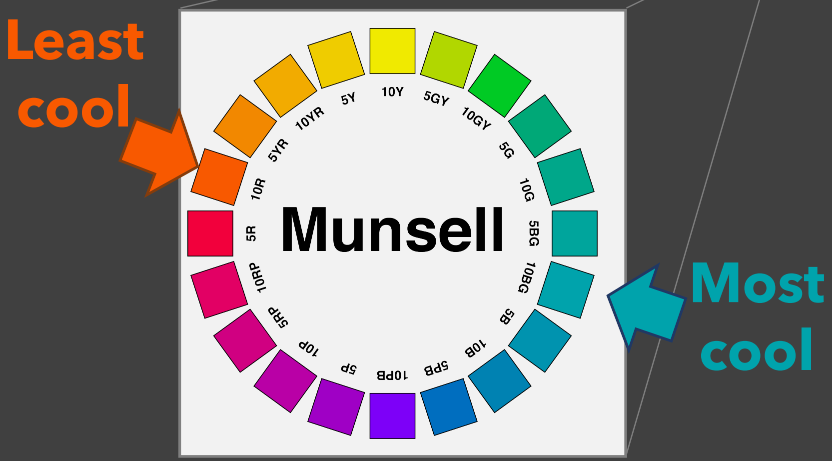


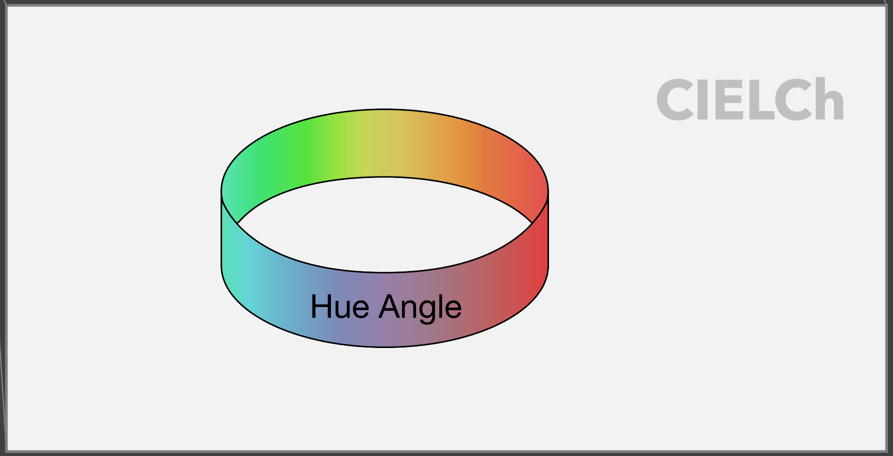


Sharma et al. 2006 (CIEDE2000); Heer and Stone 2012 (Hellinger distance)

Measure of aesthetic preference

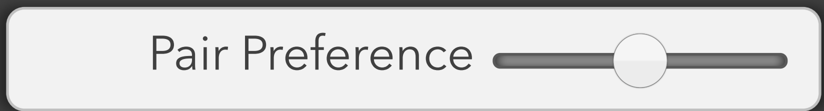
Preferable = “cool” colors with similar hues



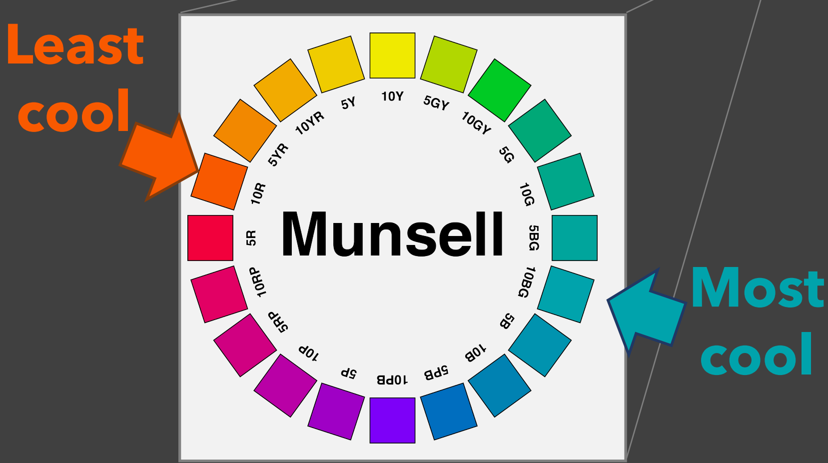


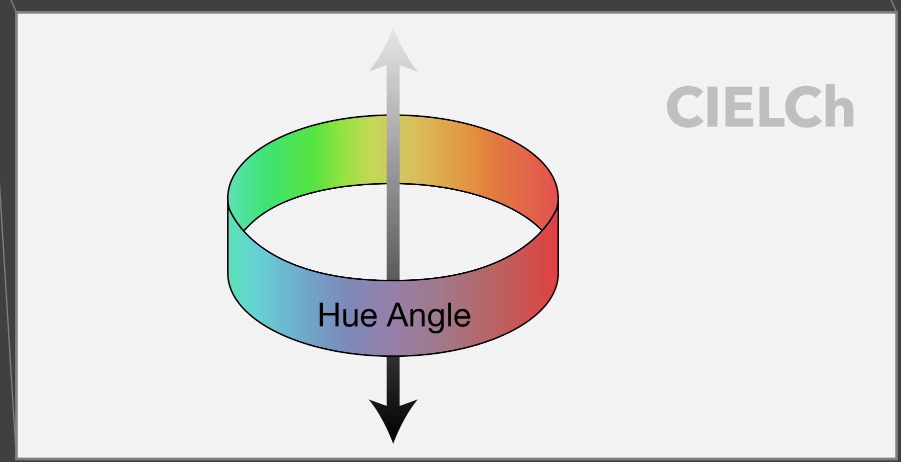
Schloss and Palmer 2011

Measure of aesthetic preference



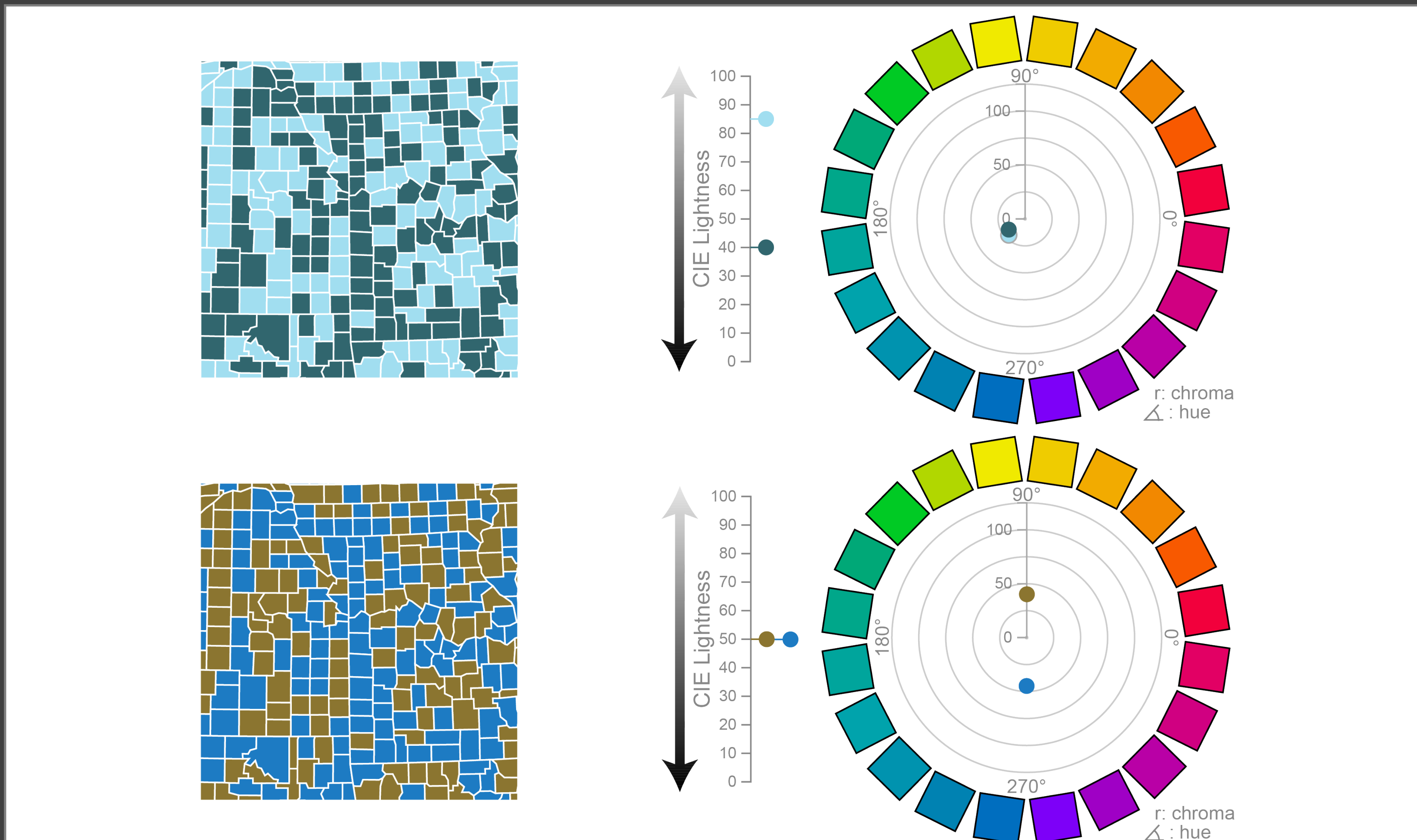
Preferable = “cool” colors with similar hues and different lightness





Schloss and Palmer 2011

Preferable = “cool” colors with similar hues and different lightness

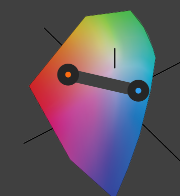


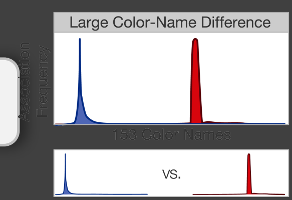
Scoring colors with many slider changes: slider-weighted sum

User defined balance

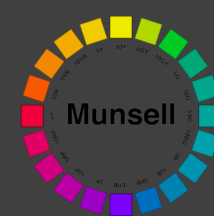
Perceptual Distance
Name Difference
Pair Preference

Slider: [0%, 100%]

Perceptual Distance  (c_1, c_2)

Name Difference 

(c_1, c_2)

Pair Preference 

(c_1, c_2)

+

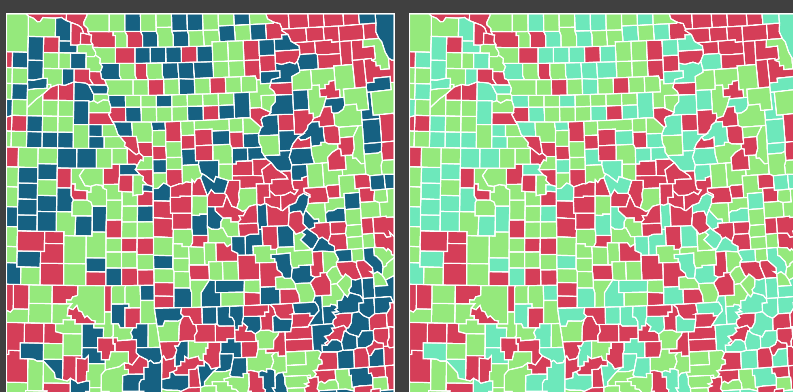
$$\text{colorPairScore}(c_1, c_2)$$

A palette is only as good as its worst pair.

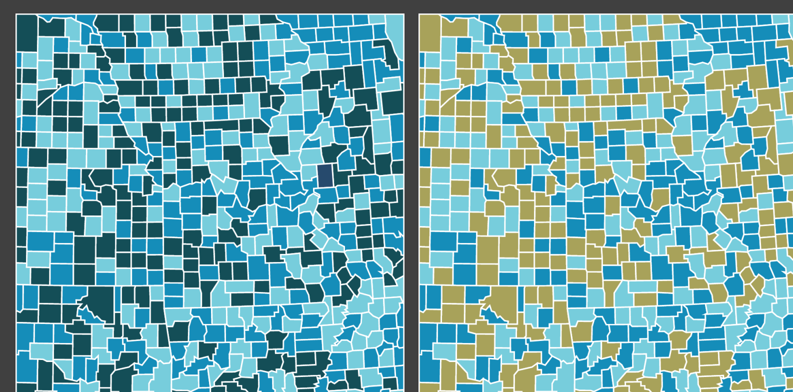
Palette score: how would a new color c change palette *P*'s score?

$$\text{paletteScore}(c, P) \\ = \min(\text{colorPairScore}(c, p_i)\ \forall\ p_i \in P)$$

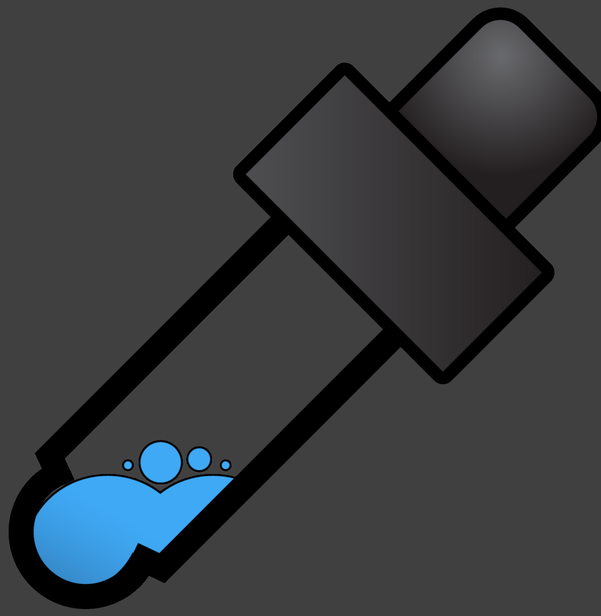
Discriminability

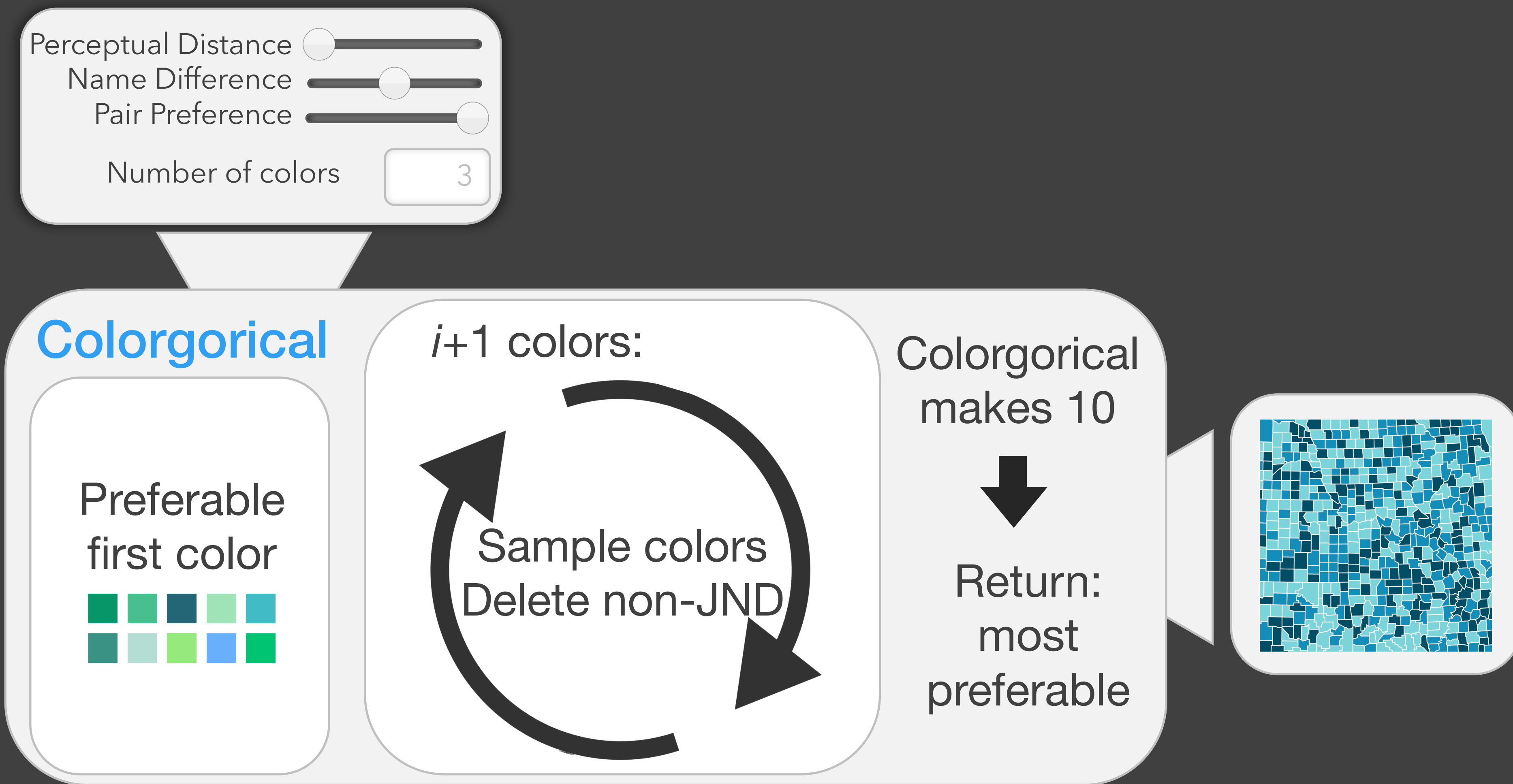


Aesthetic Preference



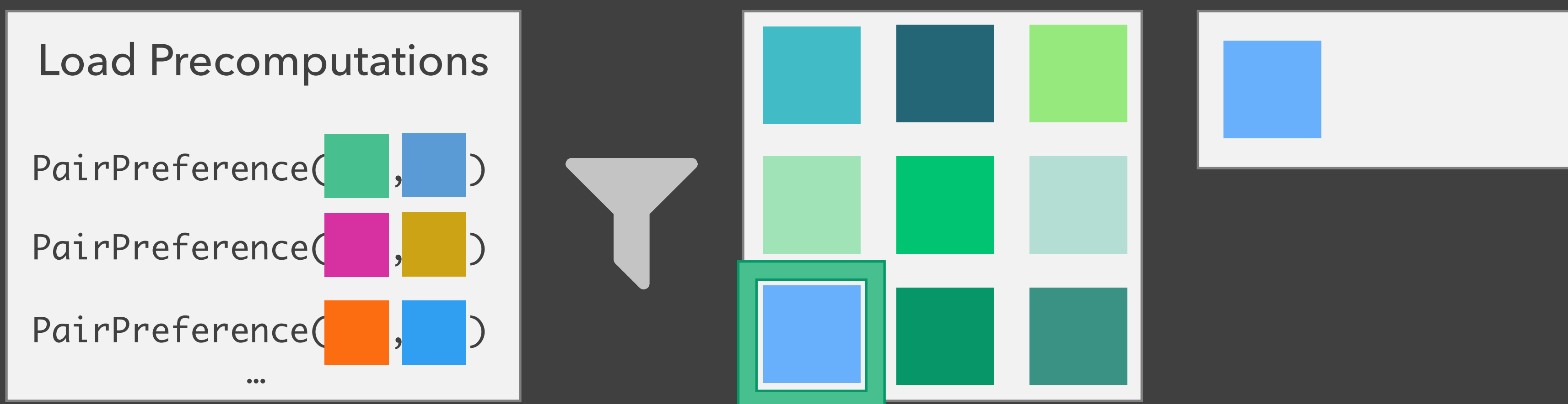
Colorgorical
sampling procedure



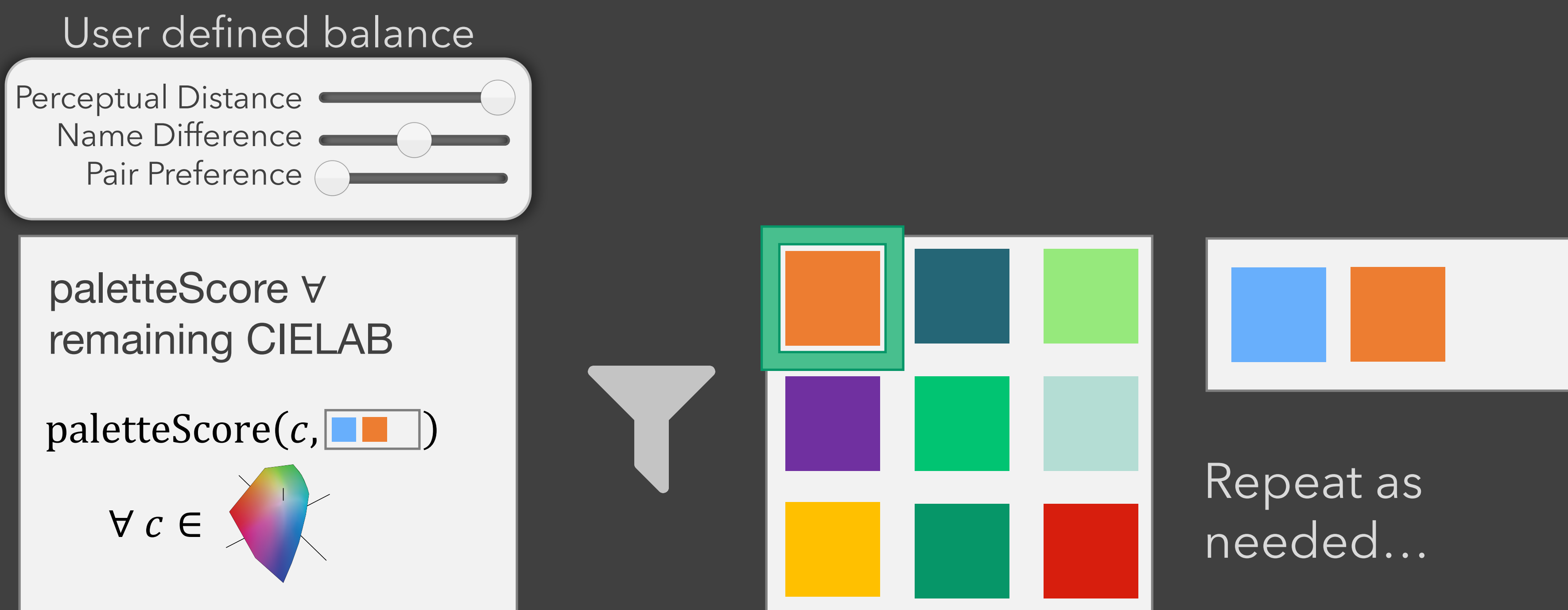

Perceptual Distance
Name Difference
Pair Preference
Number of colors
3
Colorgorical
Preferable first color
i+1 colors:
Sample colors
Delete non-JND
Colorgorical makes 10
Return: most preferable

Step 1:
Colorgorical chooses 1st color

Goal: choose a color that exists in highly preferable pairs



Step 2:
Colorgorical chooses i+1 colors

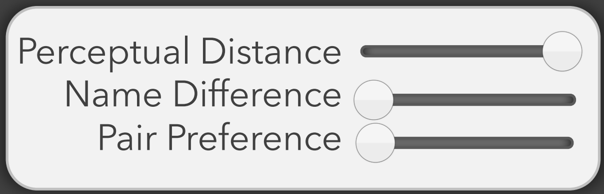


Experiment 1

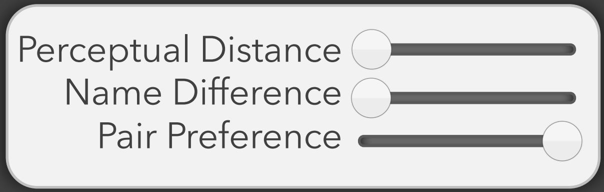
How do Colorgorical settings map onto human discriminability & preference?

Exp. 1:
Settings → Human Judgement?

User defined balance



?



?

User Behavior

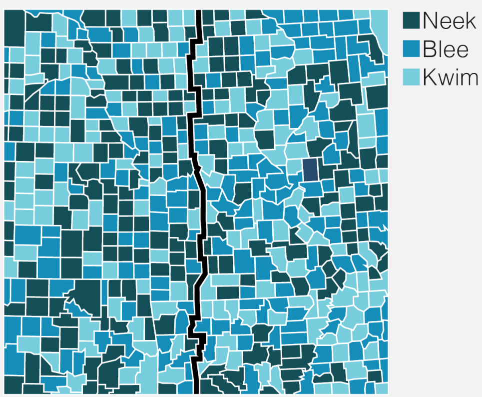
Low error

High preference ratings

Task: crowdsourced on MTurk

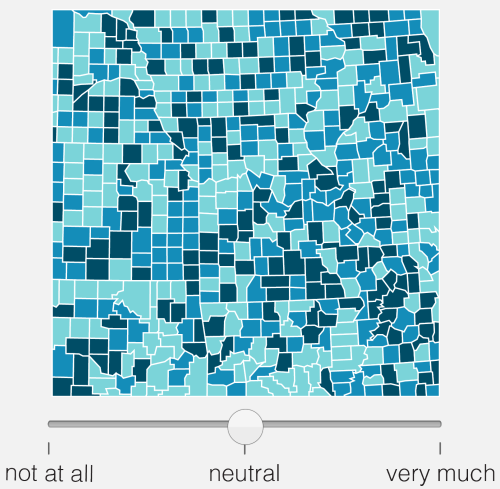
Discriminability
More "Neek": left or right?

Response: Error



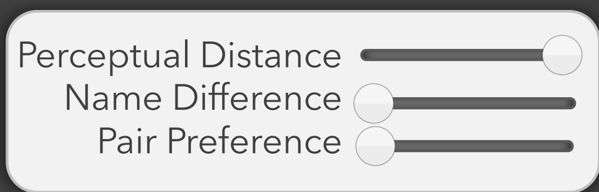
Preference Rating
Is color combination preferable?

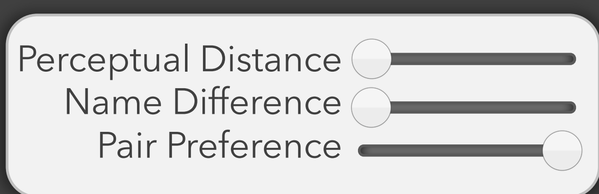
Response: Rating [-100,100]



Exp. 1:
Settings → Human Judgement?

User defined balance





User Behavior

Low error

3-color: range(r) = [0.697, 0.887]
5-color: range(r) = [0.898, 0.945]
8-color: range(r) = [0.731, 0.838]

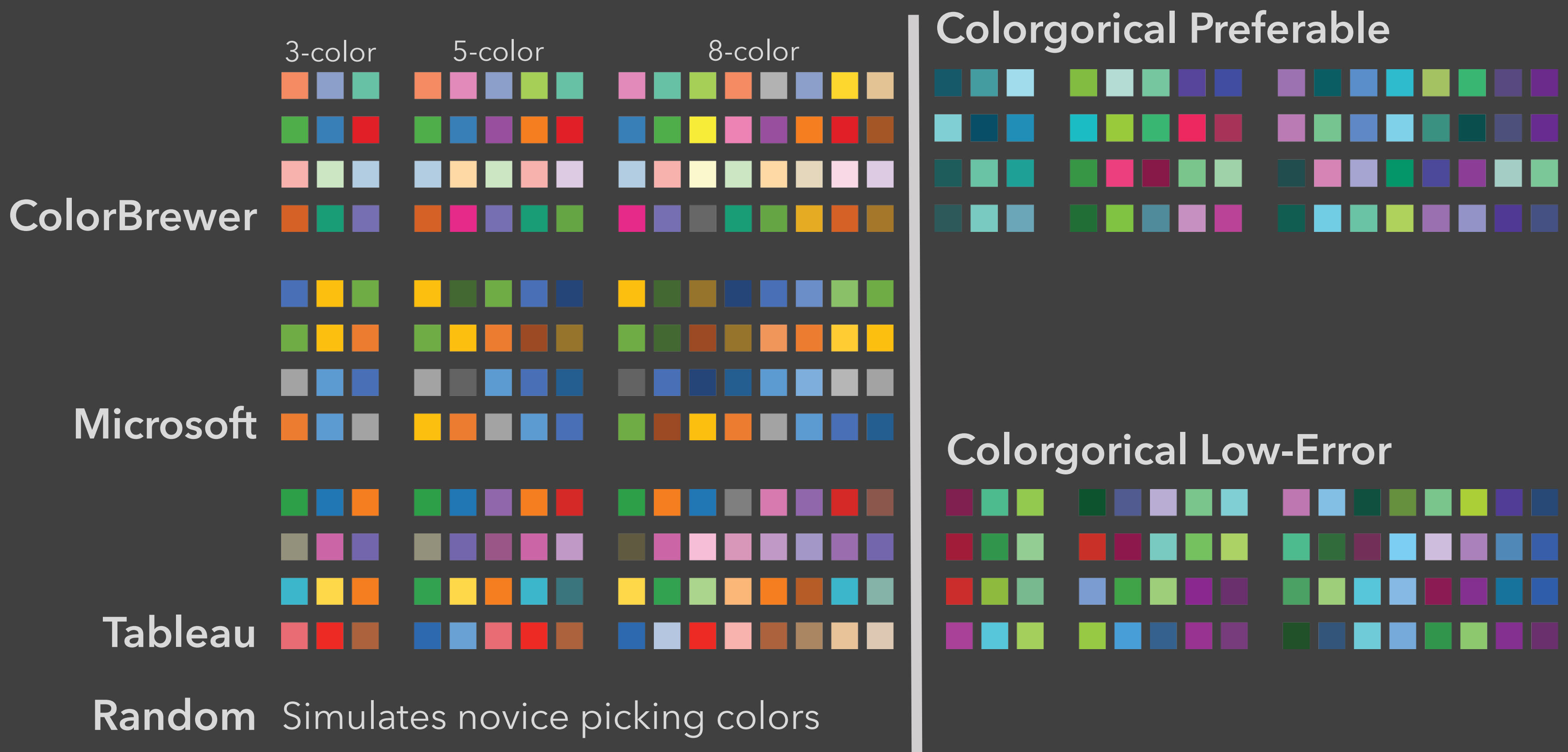
High preference ratings

3-color: range(r) = [0.897, 0.971]
5-color: range(r) = [0.412, 0.570]
8-color: range(r) = [0.751, 0.891]

Experiment 2

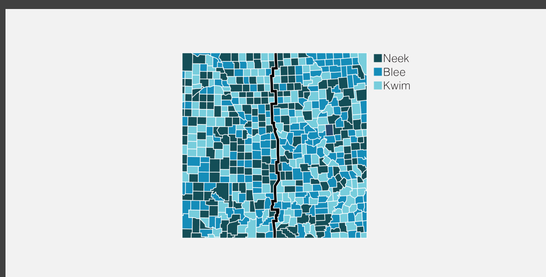
How does Colorgorical compare to industry standards?

Industry Standard Comparisons

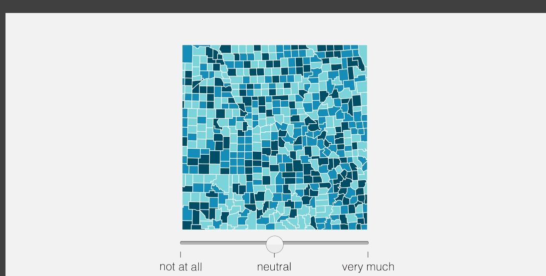


Exp. 2 Design:
Same task as Exp. 1

Discriminability
More "Neek": left or right?



Preference Rating
Is color combination preferable?



3 palette sizes

N=20/size

3 5 8

6 palette sets

4 versions/set

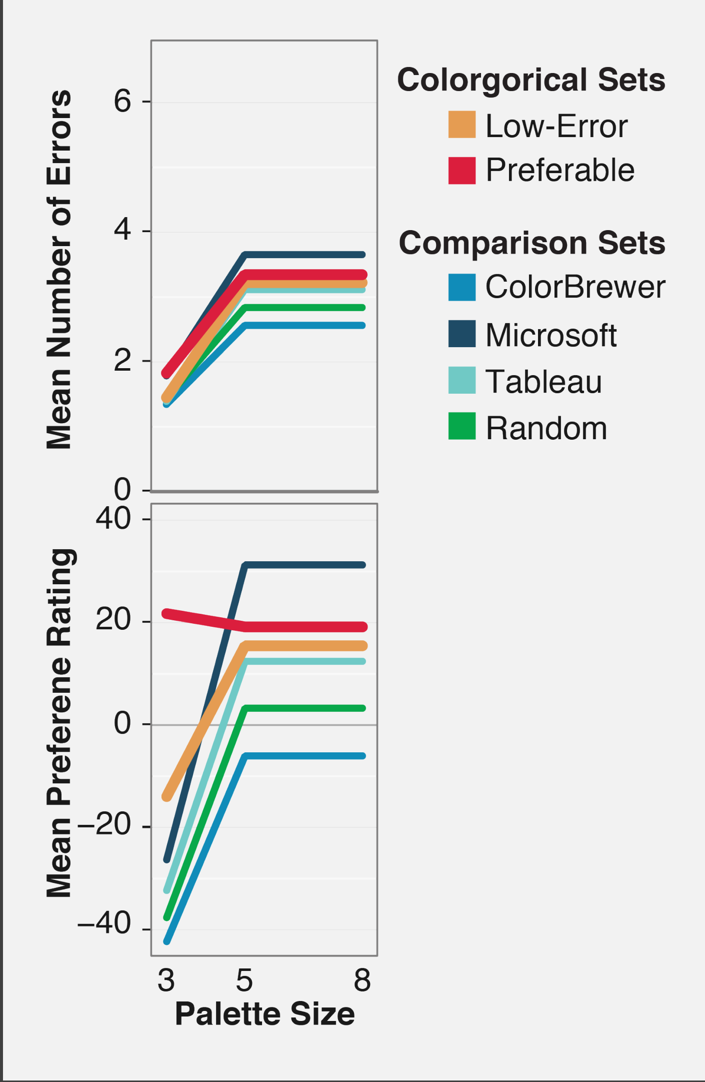
4 repetitions (discriminability)

= 24 preference trials
= 96 discriminability trials

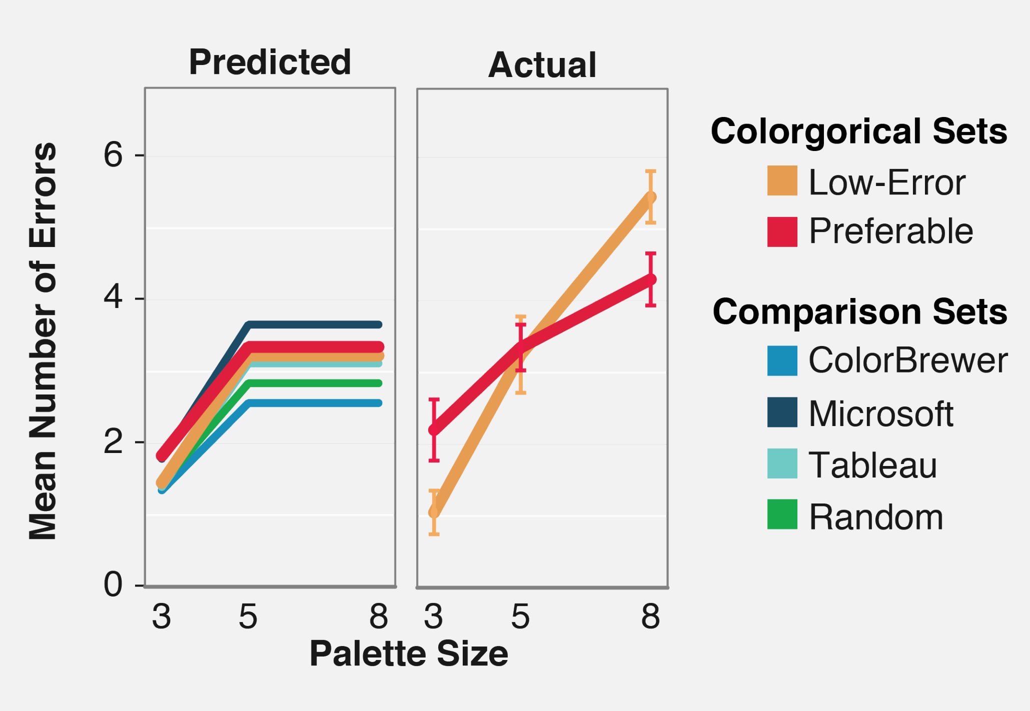
Experiment 2 Prediction

Based on linear regressions trained with Exp. 1 data

Experiment 2 Colorgorical palettes chosen anew



Discriminability error-rate: Colorgorical

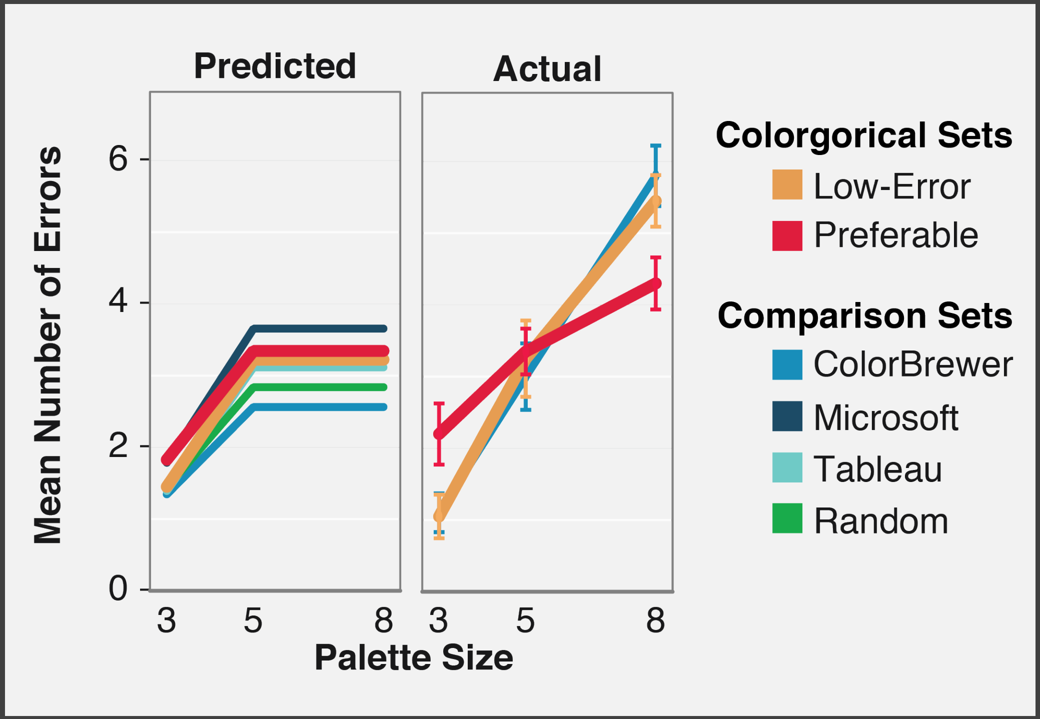


Prediction:
Colorgorical is comparably discriminable

Error: # incorrect out of 16

Bars: standard error

Discriminability error-rate

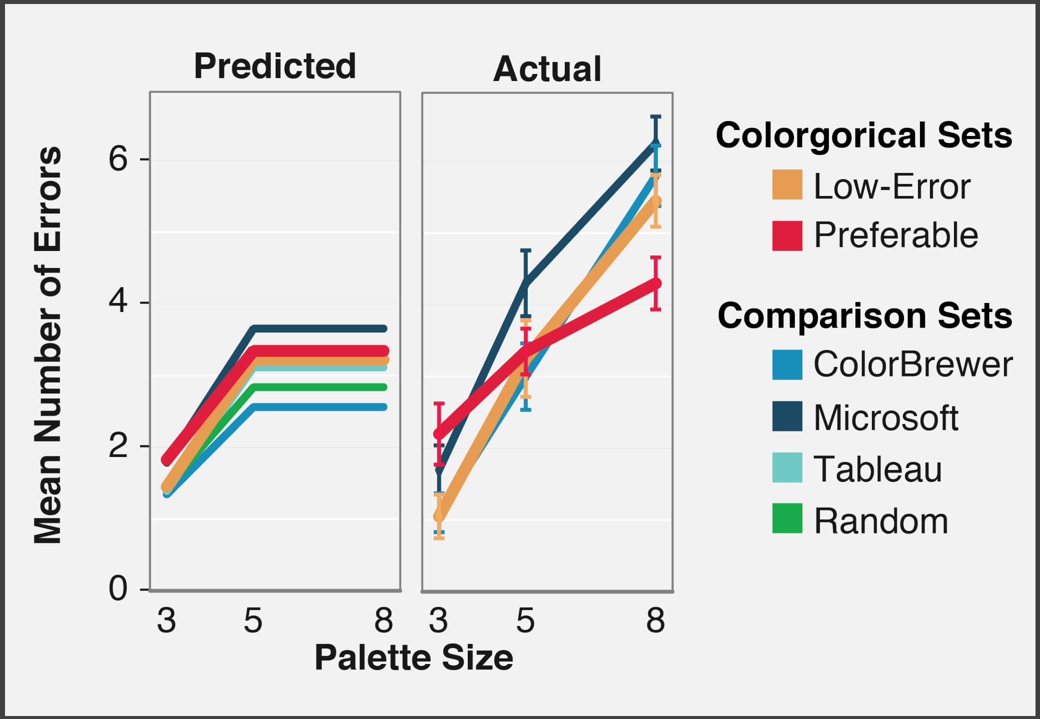


Prediction:
Colorgorical is comparably discriminable

Error: # incorrect out of 16

Bars: standard error

Discriminability error-rate

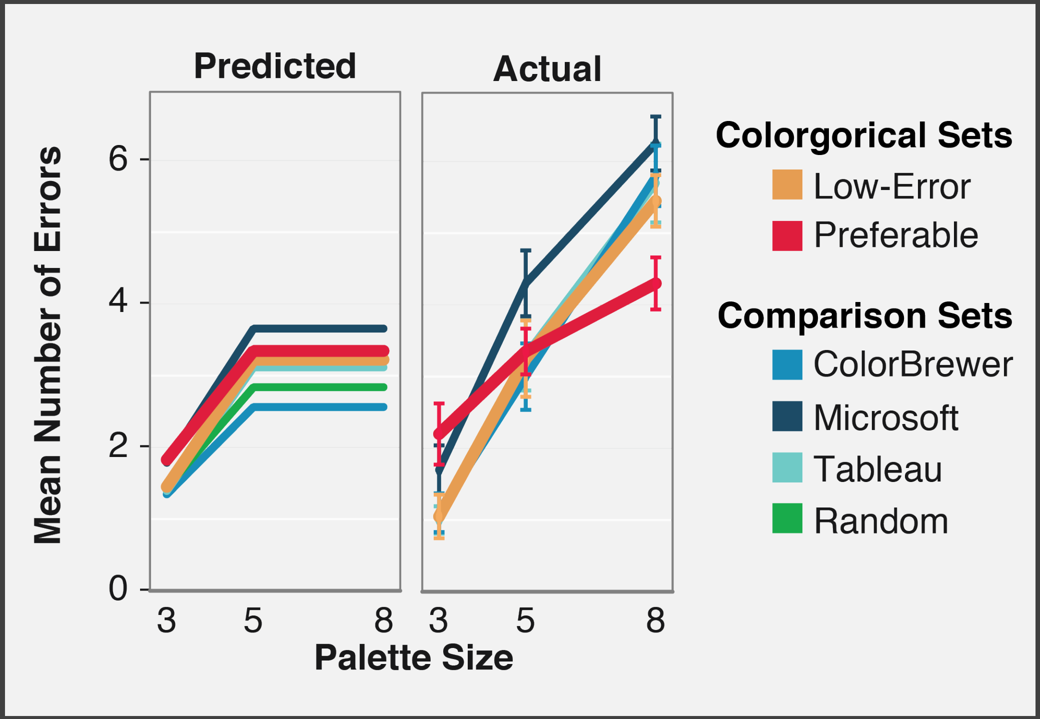


Prediction:
Colorgorical is comparably discriminable

Error: # incorrect out of 16

Bars: standard error

Discriminability error-rate

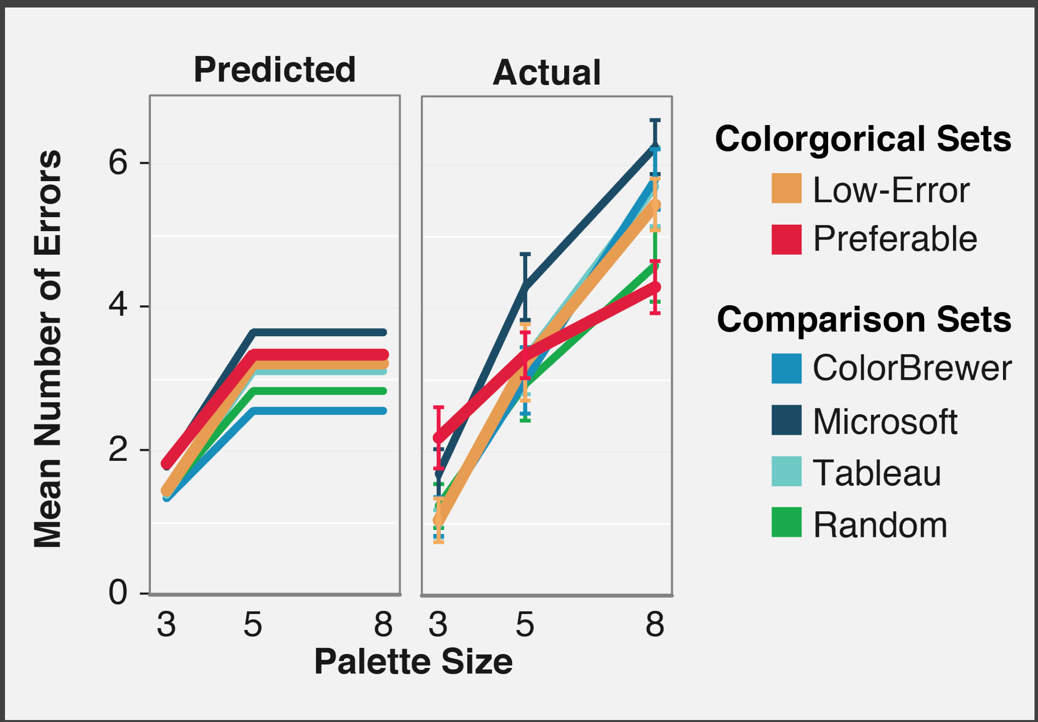


Prediction:
Colorgorical is comparably discriminable

Error: # incorrect out of 16

Bars: standard error

Largely no significant difference in error rates

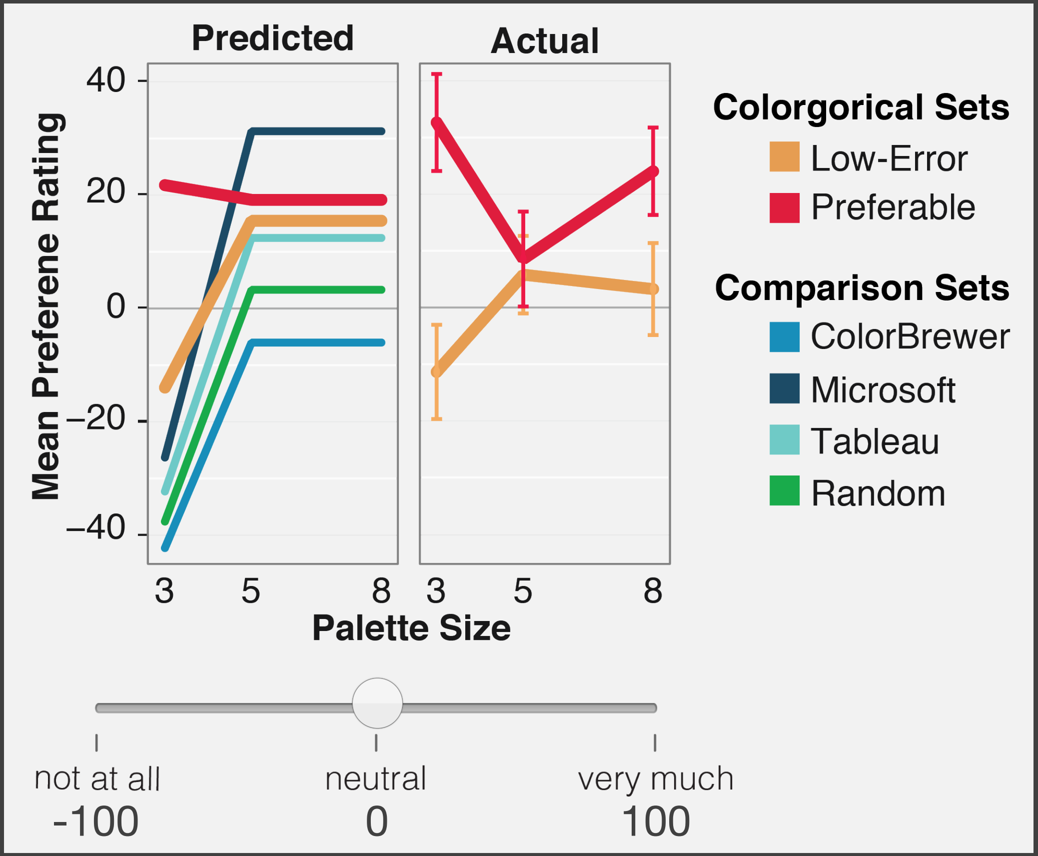


Prediction:
Colorgorical is comparably discriminable

Error: # incorrect out of 16

Bars: standard error

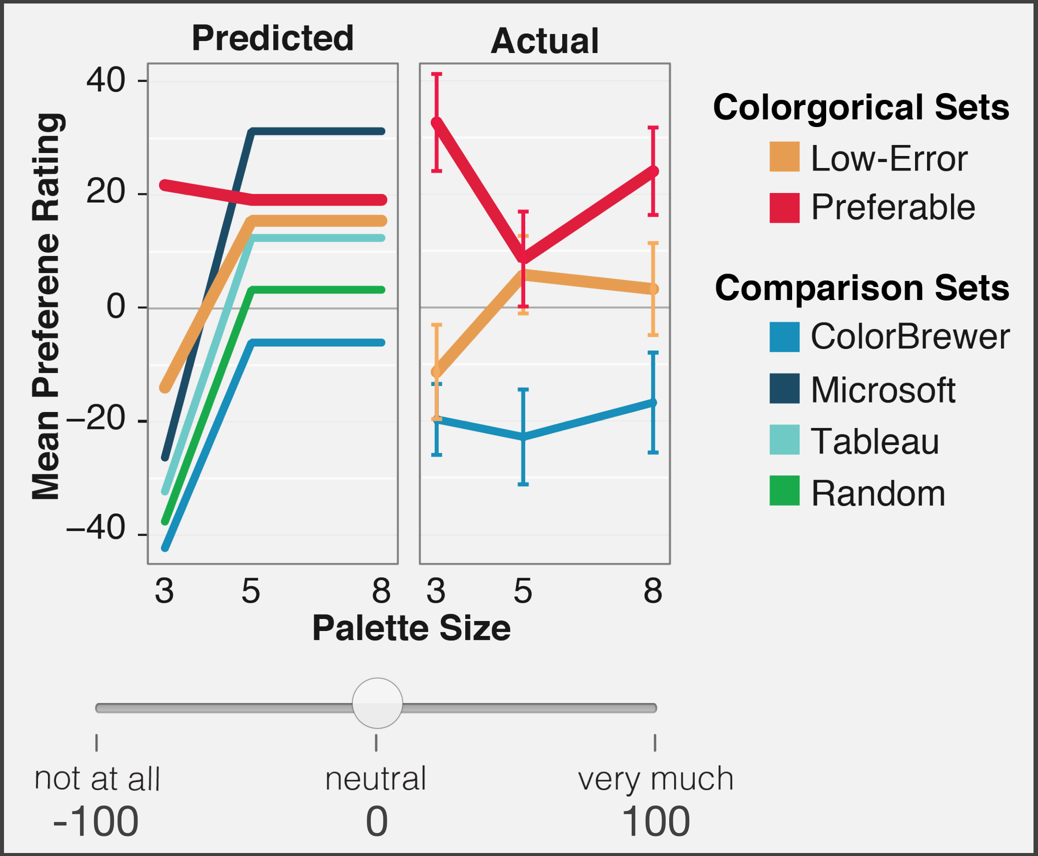
Preference ratings



Prediction
Colorgorical is typically more preferable

Bars: standard error

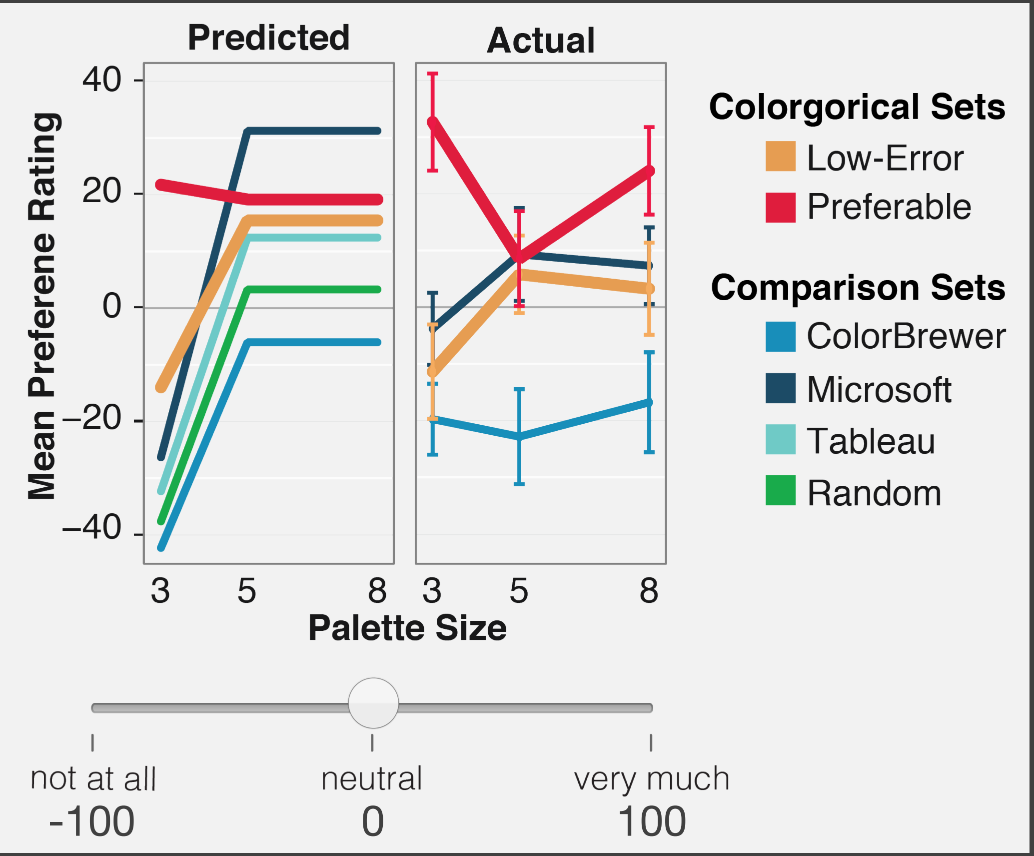
Preference ratings



Prediction
Colorgorical is typically more preferable

Bars: standard error

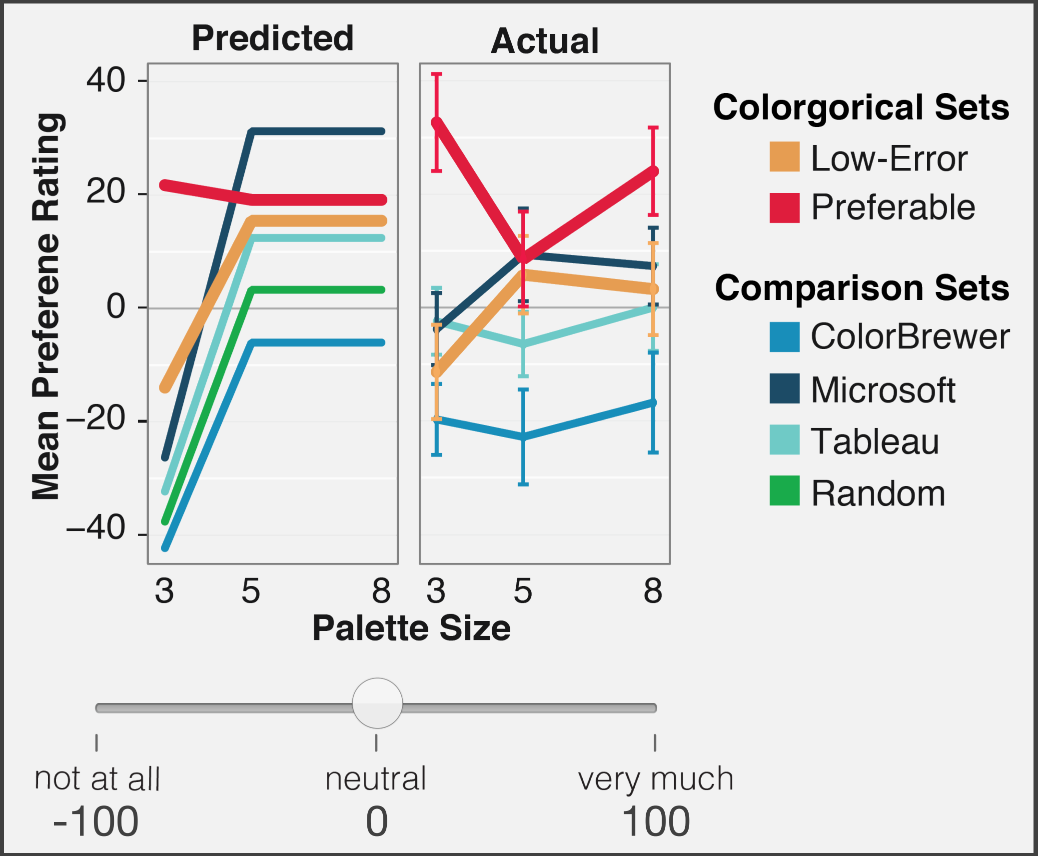
Preference ratings



Prediction
Colorgorical is typically more preferable

Bars: standard error

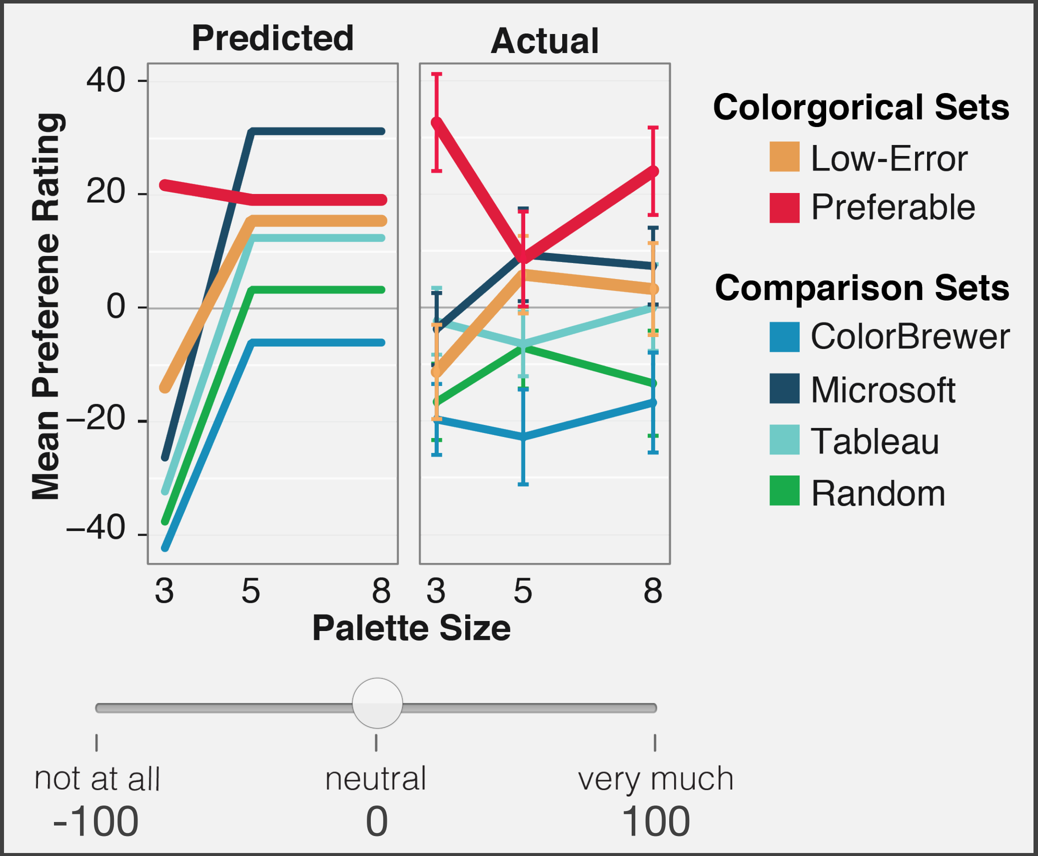
Preference ratings



Prediction
Colorgorical is typically more preferable

Bars: standard error

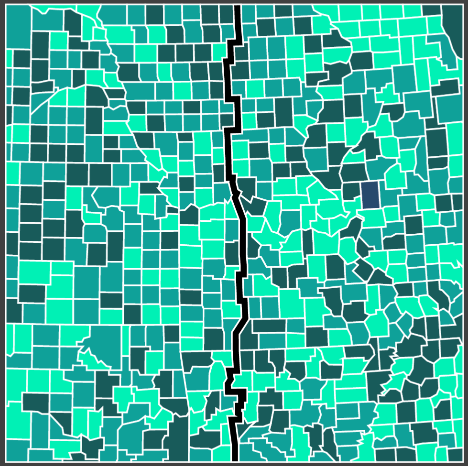
Cologorical-Preferable more, -Low-Error sometimes more



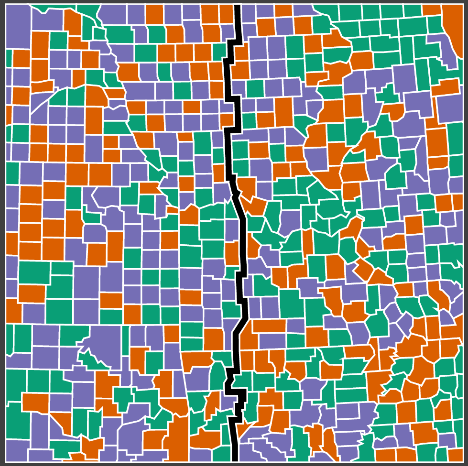
Prediction
Colorgorical is typically more preferable

Bars: standard error

Colorgorical is comparably effective.



Colorgorical



ColorBrewer

Similar levels of discriminability

Typically more preferable

→ See paper for more findings

Colorgorical

Summary

Colorgorical
Contributions

1. User-defined discriminability vs. preference

2. Δ sliders -> Δ discriminability and preference

3. Comparable to industry standards

→ Reduces design expertise requirements to make palettes

d3-jnd

gramaz.io/d3-jnd

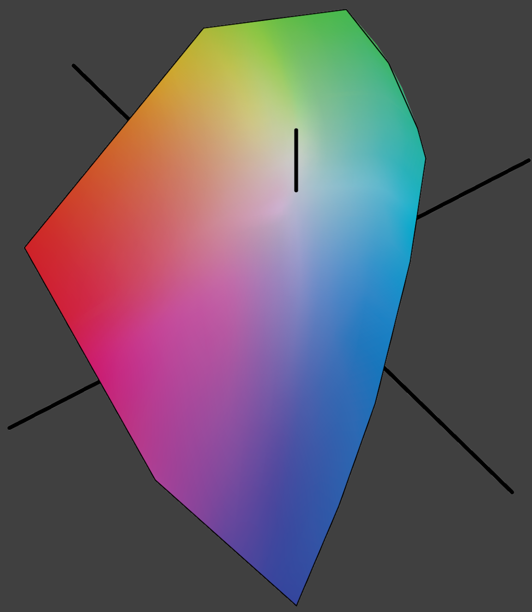
colorgorical

gramaz.io/colorgorical

d3-cam02

gramaz.io/d3-cam02

Perceptual color goal: approximation



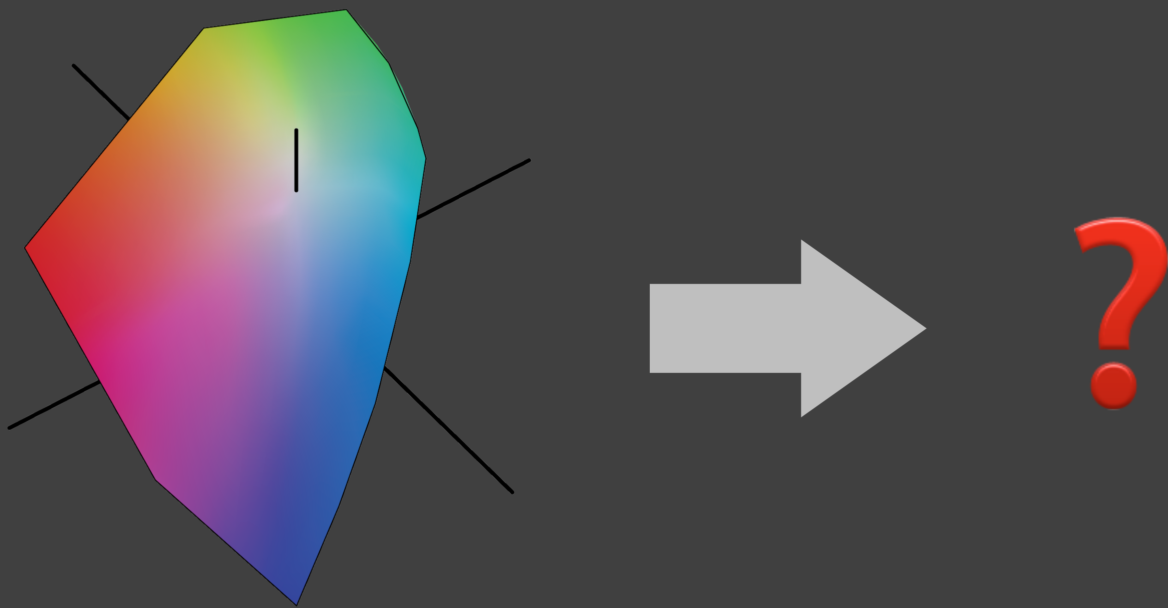
Perceptual uniformity:

$$\Delta E = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2} = $$ 👁️

In reality, CIELAB ΔE is

$$\sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C}\frac{\Delta H'}{k_H S_H}}$$

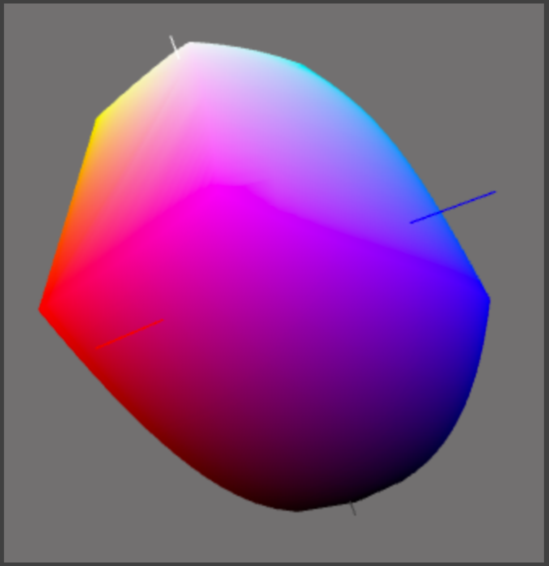
How can we change color spatiality to better model human color perception?



Better approximate: CIECAM02-UCS

CAM02: Color Appearance Model 2002
UCS: Uniform Color Space

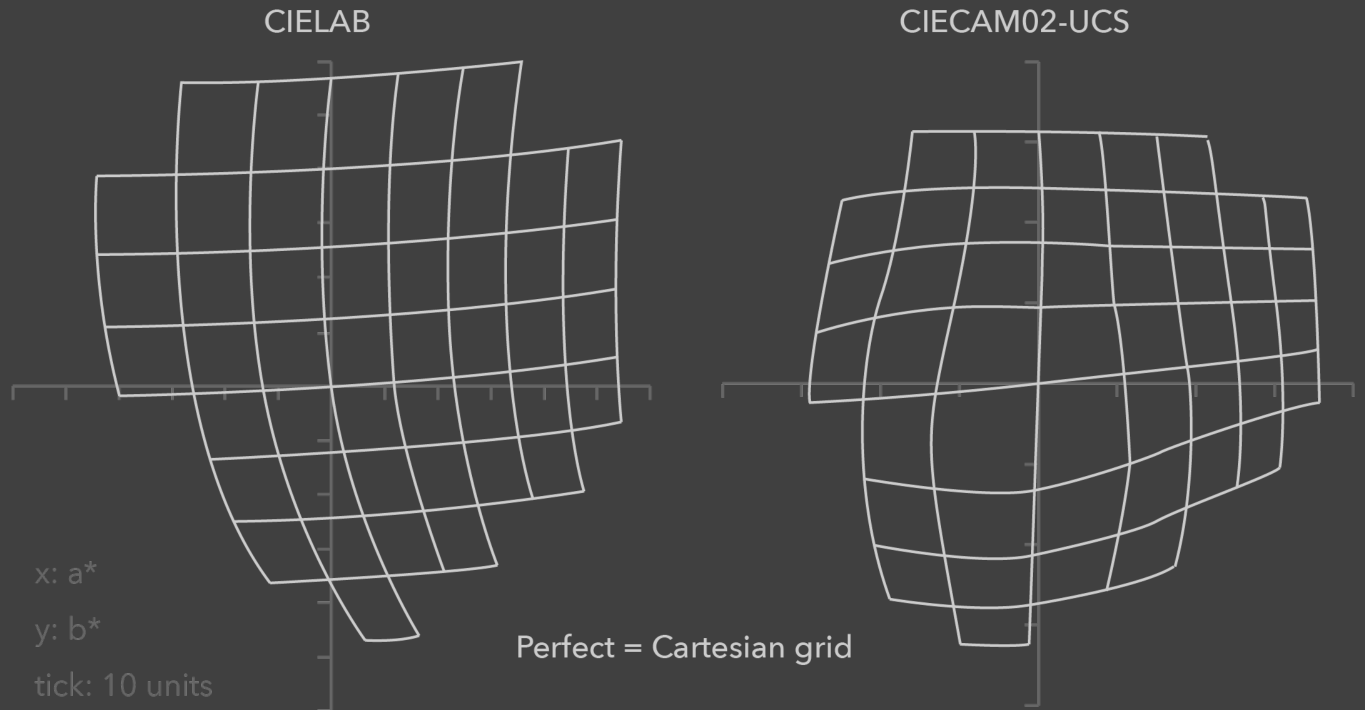
J*: Lightness
a*: redness-to-greenness
b*: blueness-to-yellowness



(CAM02 assumptions listed online)

Source: Li, Cui, and Luo 2006

Better approximate: CIECAM02-UCS



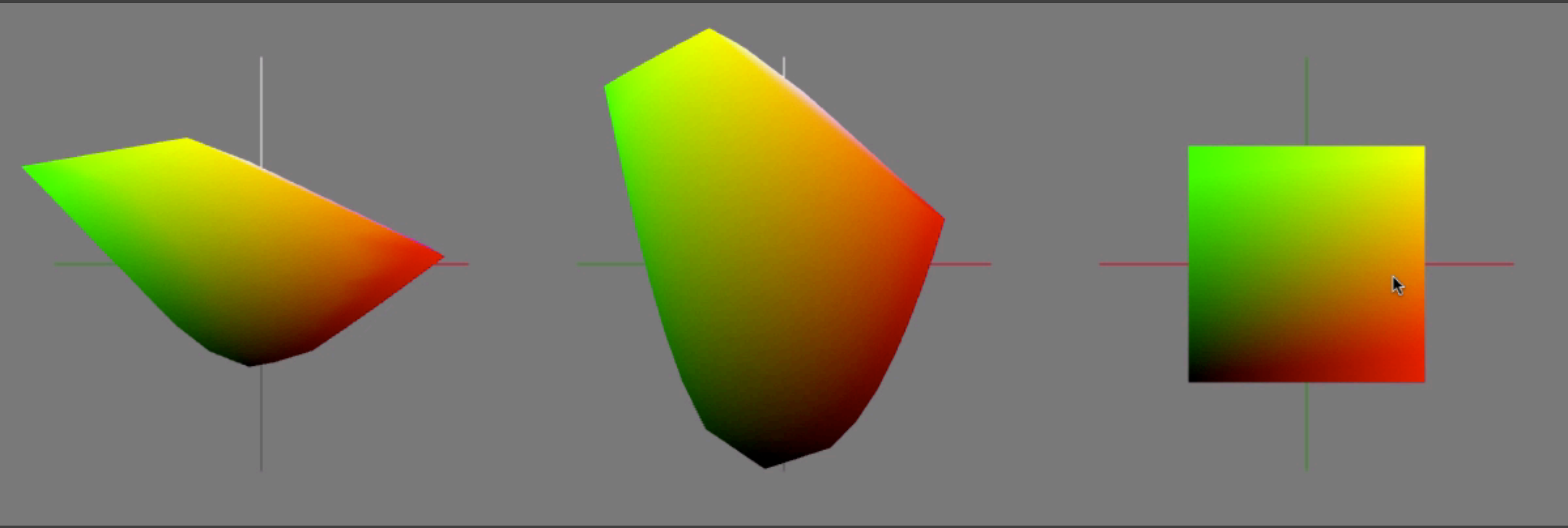
Source: Li, Cui, and Luo 2006

How do these color spaces compare?

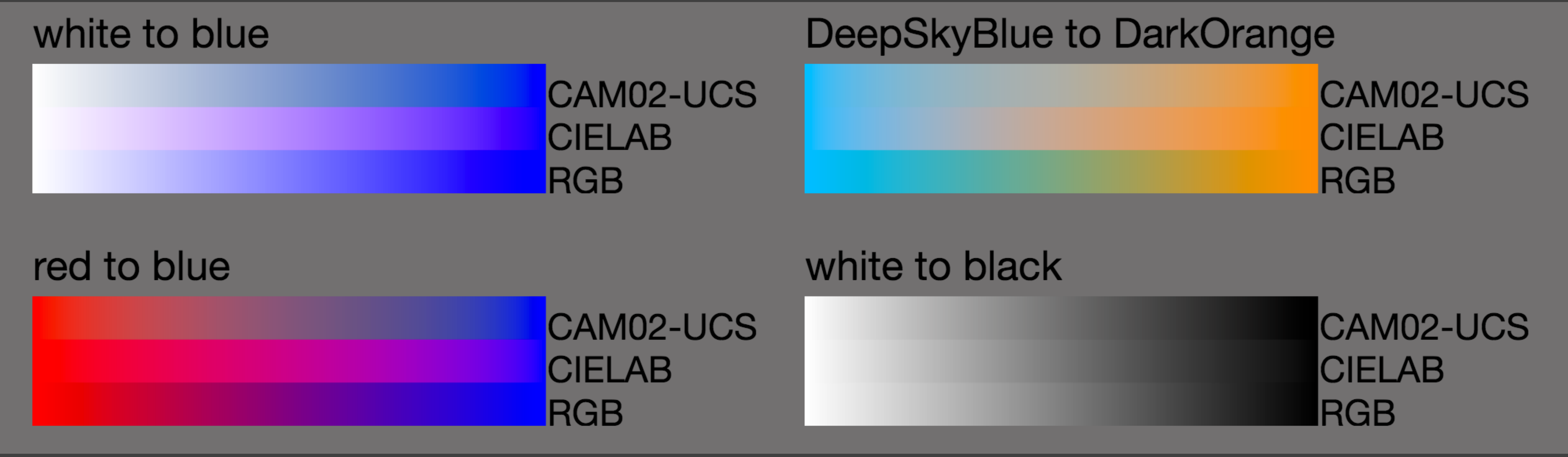
CIELAB

CAM02-UCS

RGB Cube



How do these color spaces compare?



d3-cam02 functions

```
d3.jab(J, a, b[, opacity])
d3.jab(specifier)
d3.jab(color)

// lightness, chroma, hue
d3.jch(J, C, h[, opacity])
d3.jch(specifier)
d3.jch(color)
```

Caveat: evaluation needed

CAM02-UCS validated in color science

Lack of formal evaluation for visualization

But: matplotlib uses CAM02-UCS! So, precedent!

Wrap Up

Just Noticeable Differences

Look at distance for discriminability, but also size

Colorgorical

Effective automation of color palette design with customizable appearance

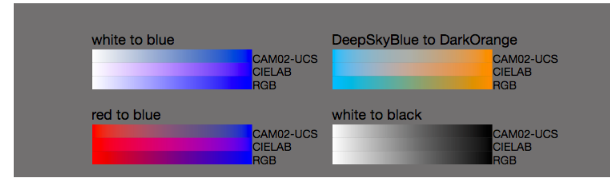
CIECAM02-UCS

Use perceptual spaces knowledgeably;
Consider other options for versatility

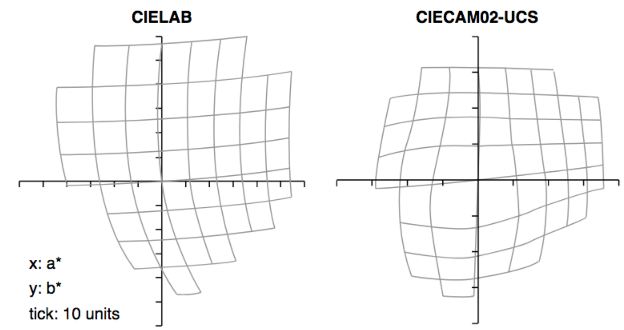
Each project has lots of online documentation + resources

CIECAM02-UCS vs. CIELAB

Perhaps the most common perceptually uniform color space that is currently used by designers is CIELAB color space characterized with CIE Standard Illuminant D65. If you've ever used d3.lab, you've used this color space. So how does CIECAM02-UCS differ from CIELAB? Maybe the best way is to see color interpolation differences:



As evident, even though CAM02-UCS and CIELAB are both perceptually uniform approximations, they define color in different ways. Also note how each compares to RGB, which is not perceptually grounded. You can see the assymetry between CIELAB and CIECAM02-UCS in Li, Cuo, and Luo's plot comparing uniformity results using colors from the Optical Society of America, where more grid-like meshes reflect greater perceptual uniformity:



CIELAB and CIECAM02-UCS comparison reconstructed from Li, Cui, and Luo 2006 [6]. Non-skewed grids reflect perceptual uniformity. Data based on fitting perceptually uniform colors as determined by the Optical Society of America.

So, is it worth using CIECAM02-UCS instead of CIELAB? As with most design decisions, there isn't a definite answer. Although CIECAM02 gives a more uniform approximation, it is unclear what the actual magnitude of difference would be on average for online audiences given the diaspora of displays that an audience could use. But, greater precision never hurts either. Ultimately, by even considering perceptual uniform spaces to begin with you are taking a step in the right direction, regardless of which you select.

Examples

Interactive CIECAM02 and CIECAM02-UCS color picker

Further reading

Citations

1. Wikipedia entry on CIECAM02 color
2. Stone, Szafir, Setlur. "An Engineering Model for Color Difference as a Function of Size," *22nd IS&T Color and Imaging Conference*. 2014.
3. Wikipedia entry on LMS color space
4. Luo and Li. "CIECAM02 and its recent developments," *Advanced Color Image Processing and Analysis*. 2013.
5. Wikipedia entry on color difference (ΔE or DE)
6. Luo, Cui, and Li. "Uniform Colour Spaces Based on CIECAM02 Colour Appearance Model," *Color Research & Application*. 2006.

Other useful links

Open source is necessary to bridge the divide, but think about accessibility, too.

Thanks!

@ccgramazio
https://gramaz.io

d3-jnd: https://gramaz.io/d3-jnd

Colorgorical: https://gramaz.io/colorgorical

d3-cam02: https://gramaz.io/d3-cam02