Check for updates

# HOW TO EVALUATE AND COMPARE COLOR MAPS

*MATTEO NICCOLI, MyCarta*

## Introduction

This is the first of a two-part tutorial on color maps. My goal is to share the methods I developed (Niccoli, 2012; Niccoli and Lynch, 2012; Niccoli, 2013) to evaluate default color maps and choose more perceptual alternatives (Part 1) and to make your own map based on perceptual principles (Part 2). For both parts, there will be an accompanying IPython Notebook with extended examples and analyses and open data. You can find the notebooks and code at github.com/seg/tutorials.

## The rainbow is dead

Rainbow palettes are not perceptual; they do not match how the brain intuitively interprets color. A wealth of literature (e.g., Montag, 1999; Rogowitz et al., 1999; Rogowitz and Kalvin, 2001; Light and Bartlein, 2004; Borland and Taylor, 2007; Borkin et al., 2011) shows that nonperceptual palettes are poor choices for maps and data visualization. To represent interval data (e.g., elevation or time structure) or ratio data (e.g., seismic amplitude), the equal steps in the magnitude of the data require equal perceptual distances between points in the color scale. Therefore, good palettes are those that have a strictly increasing lightness (or intensity) profile.

You can convince yourself of how bad some palettes are with the simple experiment in Figure 1, in which I deconstruct *spectral*, one of the default palettes from the matplotlib plotting library commonly used in scientific computing in Python. The full code to generate this figure and all the figures in this article are in the IPython Notebook.

The key commands are:

```
data = numpy.arange(16)
imshow(data, interpolation='none', cmap='spectral')
intensity = 0.2989 * rgb[:,0] \
          + 0.5870 * rgb[:,1] \
          + 0.1140 * rgb[:,2]
imshow(intensity, interpolation='none', cmap='gray')
sorted_intensity = numpy.sort(intensity.astype(int))
```

The conceptual premise for the experiment comes from Rogowitz et al. (1999), who examined the hue-saturation-lightness components of several color maps to see how each component encoded the magnitude information in the data. At first sight, the *spectral* color bar in Figure 1a might seem like a pleasant and ordered sequence of hues. However, if we extract the colors as RGB triplets and then convert to integers representing intensity, the sequence does not look right anymore (Figure 1b) because the intensity goes up and down seemingly randomly.

If we sort the intensity values by rearranging them from lowest to highest (as in Figure 1c) and then look at the rearranged hues, we see an unappealing result (Figure 1d). In a sense, we are back at square one because we have fixed the

intensity, but now the hues are confusing. As a footnote, if you happen to have deficient color vision (more commonly referred to as color blindness), then even the original *spectral* color bar would have looked ugly and confusing (e.g., Light and Bartlein, 2004).

This is not just about making pretty (or ugly) maps. There can be a significant and quantifiable impact from the misuse of color maps. Medical practitioners recognized long ago that color visualization is not trivial and that rainbowlike color maps in particular should be avoided. For example, Borkin et al. (2011) argue that using rainbow colors in artery visualization has a negative impact on task performance and might cause more misdiagnoses of heart disease.

On the geophysical side, Froner et al. (2013) show in a carefully controlled experiment that the area of a shallow-marine sand body mapped by hand on a time slice colored using a spectrum was 235% larger than the area resulting from mapping the sand on the same time slice colored in gray scale.

## Perceptual test

Figure 2 uses the *spectral* color map again, this time to color a model of the Great Pyramid of Giza (see data in the IPython Notebook). The code used to generate the lightness profile of *spectral* in Figure 2b is:

```
import skimage.color as skc
lab = skc.rgb2lab(rgb)
colorline(numpy.arange(256),
          lab[0,:,0],
          linewidth=2,
          cmap='spectral')
```
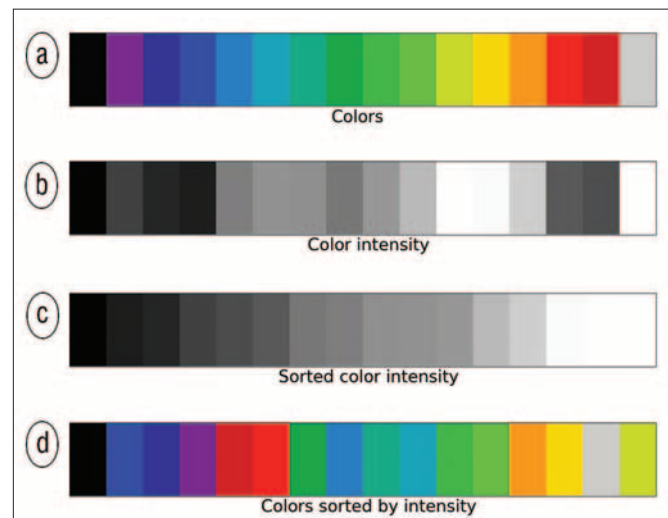


**Figure 1.** (a) Color bar made using the *spectral* color map. (b) Intensity of the color bar, calculated with the algorithm shown in the text. (c) The same map, with elements sorted by intensity. (d) Full-color version of the intensity-sorted map.
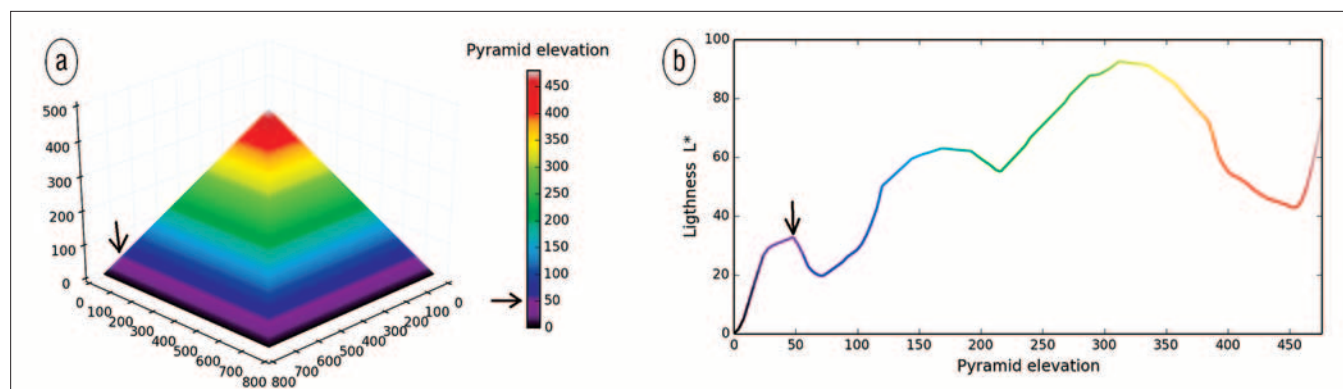
**Figure 2.** (a) The Great Pyramid, with monotonically increasing elevation, using the *spectral* palette, and (b) the lightness of the color map. Clearly, we have introduced artificial discontinuities that are not present in the data, which by construction form a perfect polyhedron.
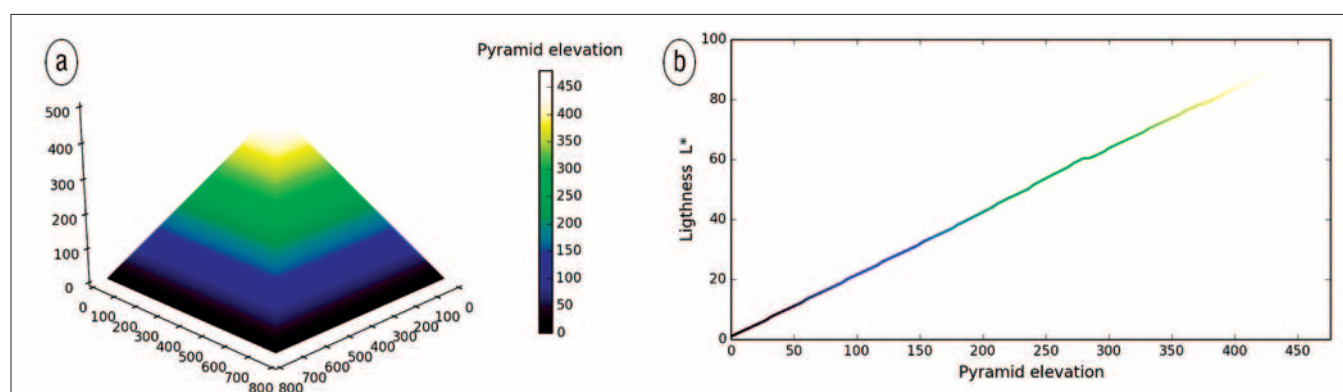


**Figure 3.** (a) The Great Pyramid with the Linear*L* palette from my blog. (b) It has a linear lightness profile.

In line 1, we import the SciKits image-processing library. In line 2, we use one of its functions to convert the *spectral* palette from RGB to CIE Lab color space (I omit here an intermediate step to convert a 256-×-3 RGB array to the 256-×-256-×-3 array that SciKits requires). In line 3, we plot the 256-×-1 lightness (L) array shown in Figure 2b.

To me, the combination of pyramid and lightness graph is a very effective perceptual test. You can see the changes in gradient magnitude and direction (sign) of lightness and can link them directly with the artifact you observe on the surface of the pyramid.

The erratic lightness profile in Figure 2 highlights the many issues with *spectral*. The curve gradient changes several times, indicating a nonuniform perceptual distance between samples. Worse, inversions occur in the gradient (one example is at the arrow). If this palette is used to map elevation, it will interfere with the correct perception of relief, particularly if shading is added, and it will hinder our interpretation when we do not have a priori knowledge of the structures in the data. Conversely, when using the Linear*L* palette as in Figure 3, the pyramid surface is smoothly colored, without perceptual artifacts, as expected.

### Horizon comparison

I used code from Matt Hall's first tutorial (Hall, 2014) to import the same Penobscot 3D horizon. At first glance, the map in Figure 4a might seem better because of the apparent higher contrast (higher magnitude gradients), but at what cost? With insight from the perceptual tests, we recognize that the cost is that of introducing artificial structures not present

in the data combined with the obfuscation of subtle structures that are present conversely in the data (one example is at the black arrow, more clearly seen in the map in Figure 4b). So are we interpreting the color maps or the data or both?

### More perceptual palettes

Next time, I will show readers how to make perceptual palettes from scratch. In the meantime, I can recommend using any of the three perceptual rainbows downloadable from my blog at wp.me/P1IlJY-Z6 or using *cubehelix*, which is available by default in matplotlib and was used in Hall (2014) for the Penobscot horizon. *TLE*

### References

Borkin, M. A., K. Z. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. J. Rybicki, C. L. Feldman, and H. Pfister, 2011, Evaluation of artery visualizations for heart disease diagnosis: IEEE Transactions on Visualization and Computer Graphics, **17**, no. 12, 2479–2488, http://dx.doi.org/10.1109/TVCG.2011.192.

Borland, D., and M. R. Taylor II, 2007, Rainbow color map (still) considered harmful: IEEE Computer Graphics and Applications, **27**, no. 2, 14–17, http://dx.doi.org/10.1109/MCG.2007.323435.

Froner, B., S. J. Purves, J. Lowell, and J. Henderson, 2013, Perception of visual information: The role of colour in seismic interpretation: First Break, **31**, no. 4, 29–34, http://dx.doi.org/10.3997/1365-2397.2013010.

Hall, M., 2014, Smoothing surfaces and attributes: The Leading Edge, **33**, no. 2, 128–129, http://dx.doi.org/10.1190/tle33020128.1.

Light, A., and P. Bartlein, 2004, The end of the rainbow? Color schemes for improved data graphics: Eos, Transactions, American Geophysical Union, **85**, no. 40, 385–391, http://dx.doi.org/10.1029/2004EO400002.
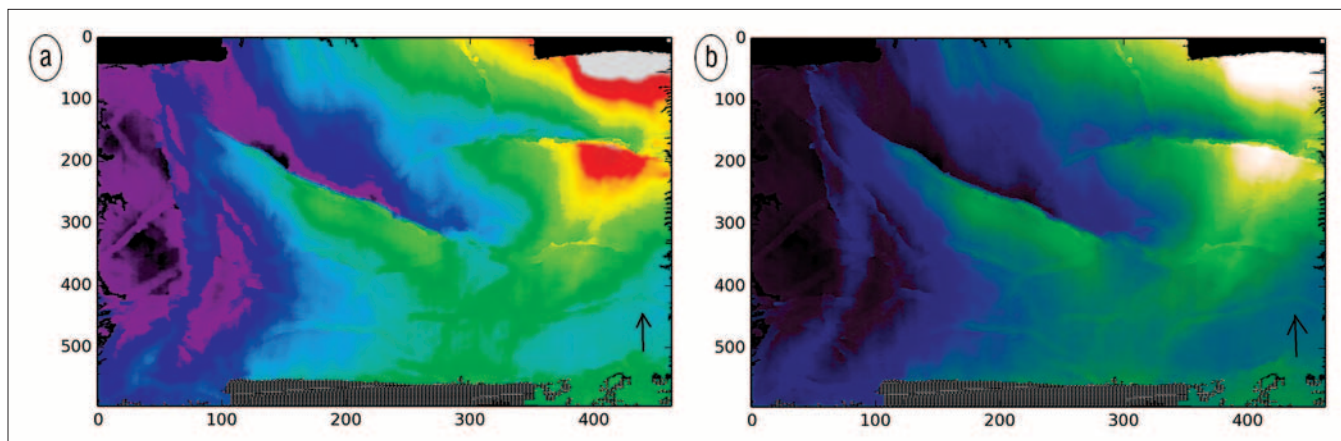
**Figure 4.** A horizon from the Penobscot 3D seismic volume, colored with (a) *spectral* and (b) Linear*L* color maps. Color bars are omitted intentionally.

Montag, E., 1999, The use of color in multidimensional graphical information display: IS&T Seventh Color Imaging Conference: Color Science, Systems, and Applications.

Niccoli, M., 2012, How to assess a colourmap, *in* M. Hall and E. Bianco, eds., 52 things you should know about geophysics: Agile Libre, 36–37.

Niccoli, M., 2013, Several articles on the My Carta blog, mycarta. wordpress.com; see wp.me/p1IlJY-js.

Niccoli, M., and S. Lynch, 2012, A more perceptual color palette for structure maps: Canada GeoConvention, abstract.

Rogowitz, B., and A. Kalvin, 2001, The "Which Blair project": A quick visual method for evaluating perceptual color maps: IEEE Proceedings, Visualization 2001.

Rogowitz, B., A. Kalvin, A. Cohen, and T. Watson, 1999, Which trajectories through which perceptually uniform color spaces produce appropriate color scales for interval data? IS&T Seventh Color Imaging Conference: Color Science, Systems, and Applications.

*Corresponding author: matteo@mycarta.ca*