

Estimating a Spectrum from its Response

Inverse Colorimetry

Glenn Davis <gdavis@gluonics.com>

September 20, 2024

Introduction

Given a “material responder” (e.g. a scanner), two materials that produce the same response from the responder are called *metamers* for that responder. A given response typically has infinitely many metamers; the goal of this **colorSpec** vignette is to demonstrate the calculation of a “good” metamer from a response. Koenderink [5] calls this topic *inverse colorimetry*. In the case that the estimated spectrum is a reflectance spectrum, the topic is often called *reflectance estimation* or *reflectance recovery*, see Bianco [1].

The *centroid method*, which is the default and the featured method in this package, computes the *centroid* of the set of all the metamers (if any). This centroid is computed in an infinite-dimensional function space and its definition is expounded further in Davis [3]. Also illustrated in this vignette is the *TLSS method*, developed by Burns [2].

The featured function from **colorSpec** used in this vignette is `invert()`. Some lengthy plotting functions are imported from the file `invert-help.R`.

```
library( colorSpec )
source( "invert-help.R" ) # for plotOriginalPlusEstimates() and plotReparam3()
```

Make a “material responder” from Illuminant E and the CIE 1931 standard observer.

```
wave = 400:700
E.eye = product( illuminantE(1,wave), "material", xyz1931.1nm, wavelength=wave )
```

Compare the original and equalized responsivities after reparameterization, see Figure 1.

```
specnames( E.eye ) = c('xE', 'yE', 'zE')
plotReparam3( E.eye )
```

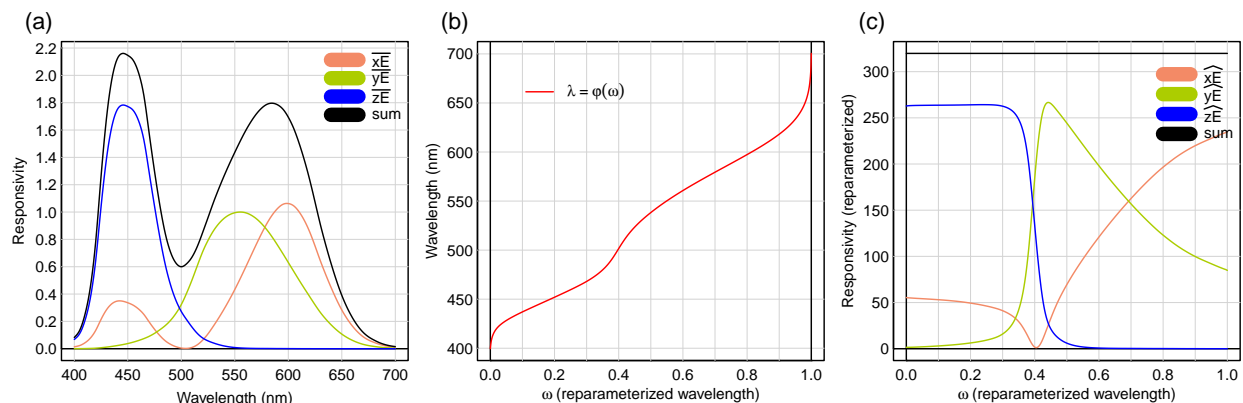


Figure 1: (a) the original responsivities of ‘E.eye’, and their sum. (b) the reparameterization to interval [0,1]. (c) the equalized responsivities, and their sum.

1 Comparison of Unequalized and Equalized Responsivities - Centroid Method

For the centroid method both unequalized and equalized versions are available. For the equalized version, a perfectly neutral reflectance spectrum is estimated exactly (up to numerical precision).

Read reflectances from the standard NCSU collection, Vrhel [10], and select a subset of 6.

```
NCSU = readSpectra( system.file( 'extdata/objects/NCSU.txt', package='colorSpec' ) )
NCSU6 = resample( subset( NCSU, c(18,32,54,62,142,170) ), wave=wave )
```

Compute responses (XYZ) for these 6 and then estimate spectra from the responses.

```
XYZ = product( NCSU6, E.eye, wavelength=wave )
est.eq = invert( E.eye, XYZ, method='centroid', alpha=1 )
est.uneq = invert( E.eye, XYZ, method='centroid', alpha=NULL )
```

Compare the original spectra, and the spectra estimated from the responses, see Figure 2.

```
plotOriginalPlusEstimates( list(NCSU6,est.eq,est.uneq), ymax=0.9 )
```

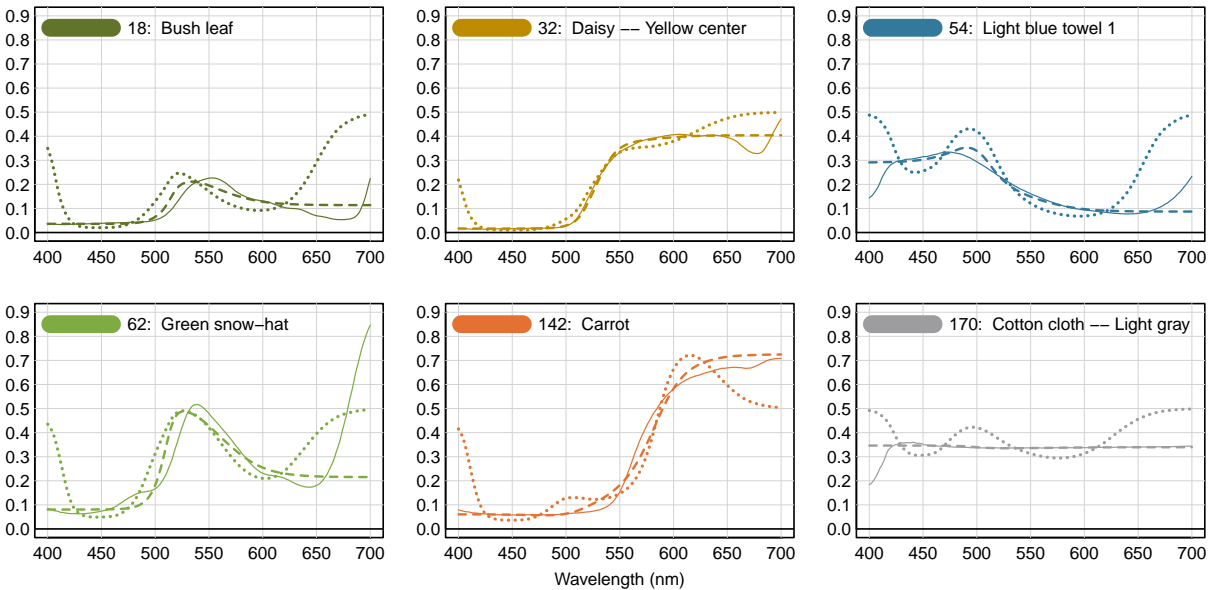


Figure 2: The original reflectance spectrum is solid linestyle; the equalized estimate is dashed; and the unequalized estimate is dotted. Both estimates use the centroid method.

Note that the unequalized (dotted) estimates have 2 artifacts: undulations and tending to 0.5 at both ends. The undulations are especially noticeable for 170 Cotton cloth – Light gray. The equalized estimates (dashed) suppress the undulations and now both ends tend to flatten.

2 Comparison of Centroid and Hawkyard Methods

The Hawkyard method is explained in Hawkyard [4] and Bianco [1]. As in the centroid method, both unequalized and equalized versions are available. Compare the equalized verions of both methods, see Figure 3.

```
est.c = invert( E.eye, XYZ, method='centroid', alpha=1 )
est.h = invert( E.eye, XYZ, method='Hawkyard', alpha=1 )
plotOriginalPlusEstimates( list(NCSU6,est.c,est.h) )
```

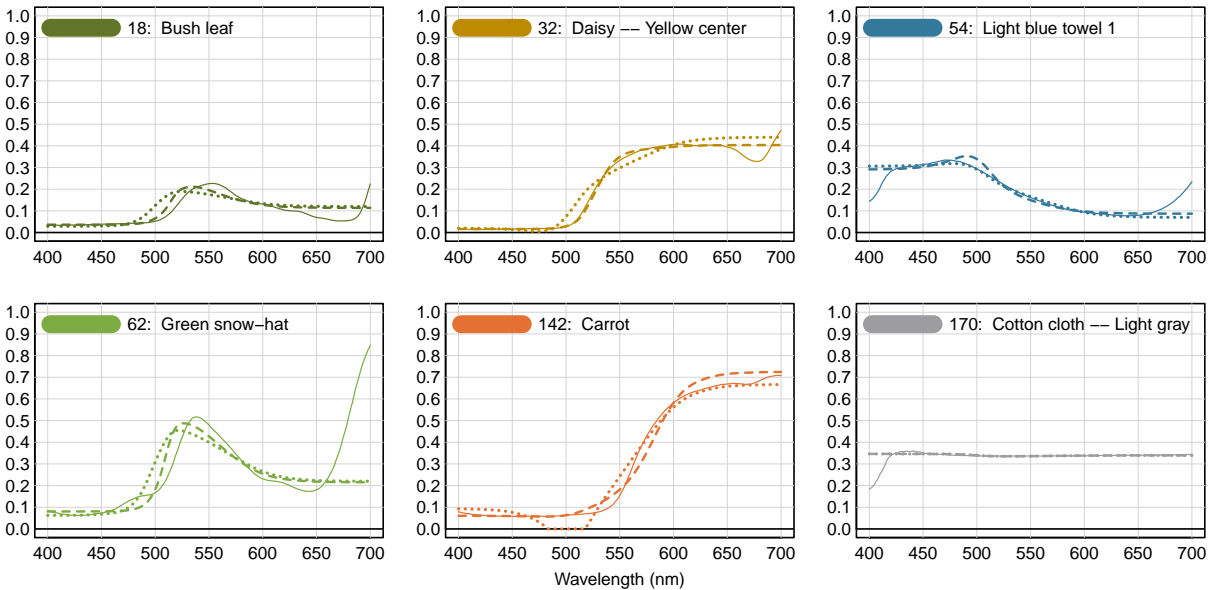


Figure 3: The original reflectance spectrum is solid linestyle; the centroid estimate is dashed; and the Hawkyard estimate is dotted. Both estimates are equalized, and the illuminant is E.

The estimates are similar, but note that for 142 Carrot the Hawkyard estimate (dotted) was negative near 500nm and had to be clamped. For this material, the Hawkyard estimate is not metameric to the original spectrum.

3 Near Optimal Colors - Centroid Method

An *optimal color* (*Optimalfarbe*) is generated by a reflectance spectrum that only takes 2 values - 0 and 1. The centroid method cannot invert an optimal color; the numerical iteration fails to converge. But it *can* invert *near-optimal* colors. In this section all spectra take the 2 reflectance values - 0.01 and 0.99.

The first group is the *near-edge colors* (*Kantenfarben*) that have exactly one transition. See Figure 4. The estimates are good, except when the reflectance is concentrated near the ends, where the responsivity of the eye is low.

```
lambda = c(-Inf,450, -Inf,500, -Inf,550, 650,Inf, 600,Inf, 550,Inf )
nearedge = rectangularMaterial( lambda, 0.98, wave ) # chromatic amplitude = 0.98
XYZ = product( nearedge, E.eye, wavelength=wave )
est.c = invert( E.eye, XYZ, method='centroid' )
# range( extradata(est.c)$estim.precis ) # est.c$estim.precis also works
plotOriginalPlusEstimates( list(nearedge,est.c), ymax=1.1 )
```

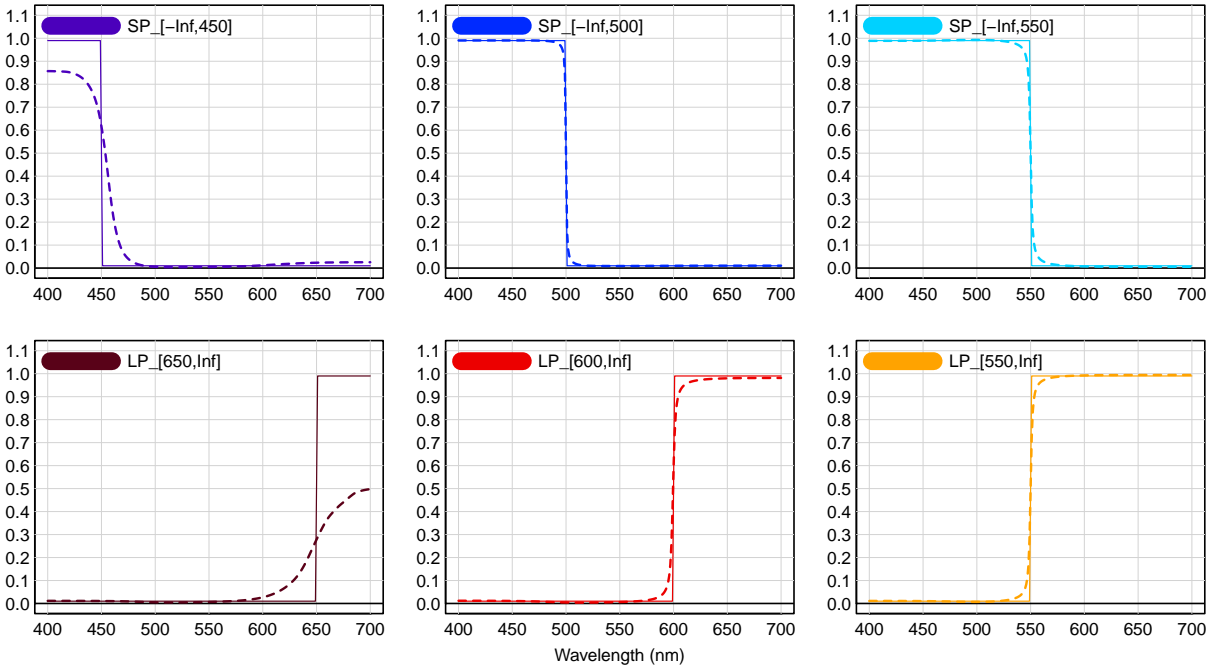


Figure 4: The original reflectance spectrum is solid linestyle, and the centroid estimate is dashed. The light source is Illuminant E.

The next group is the near-optimal with exactly two transitions; in [11] these are called the *Schrödinger colors* because they were studied by Schrödinger [9].

```
lambda = c(600,450, 650,500, 450,500, 500,550, 550,600, 600,650 )
lambda = c( lambda, 500,525, 525,550, 550,575 )
nearschro = rectangularMaterial( lambda, 0.98, wave ) # chromatic amplitude = 0.98
XYZ = product( nearschro, E.eye, wavelength=wave )
est.c = invert( E.eye, XYZ, method='centroid' )
# range( extradata(est.c)$estim.precis ) # est.c$estim.precis also works
plotOriginalPlusEstimates( list(nearschro,est.c), ymax=1.1, mfrow=c(3,3) )
```

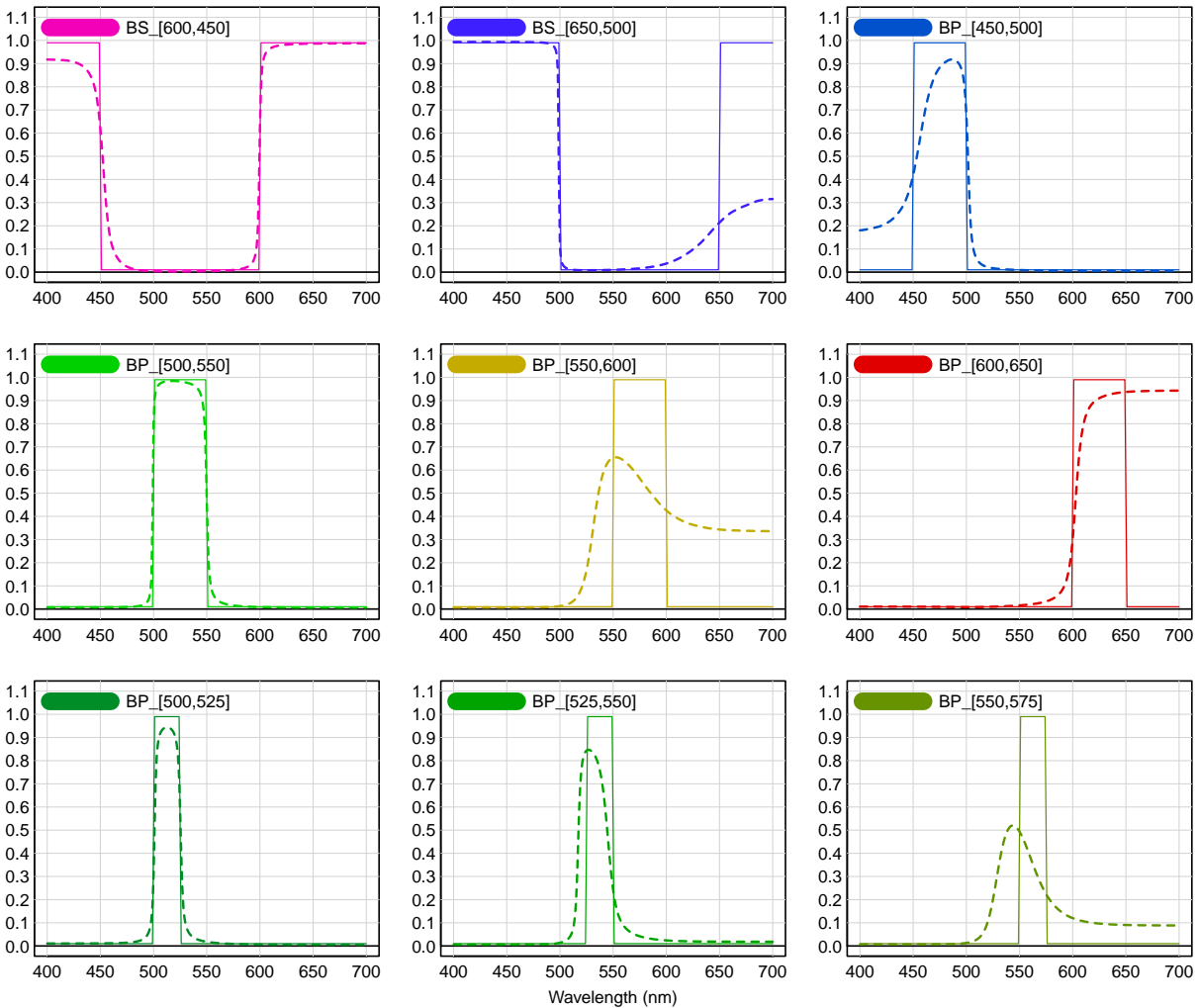


Figure 5: The original reflectance spectrum is solid linestyle; the centroid estimate is dashed. The light source is Illuminant E.

See Figure 5. The method is now clearly having trouble with these extreme spectra. It does best in the region around 500nm where the discrimination of the responsivities is the best, see Figure 1.

4 Fluorescent Light Source

Now we turn from perfectly constant Illuminant E to a fluorescent illuminant, which is as spiky as you can get. Make a “material responder” from Illuminant F10 and the CIE 1931 standard observer. Compare the original and equalized responsivities

```
F10.eye = product( subset(Fs.5nm,'F10'), "material", xyz1931.1nm, wavelength=wave )
specnames( F10.eye ) = c('xF10','yF10','zF10')
plotReparam3( F10.eye )
```

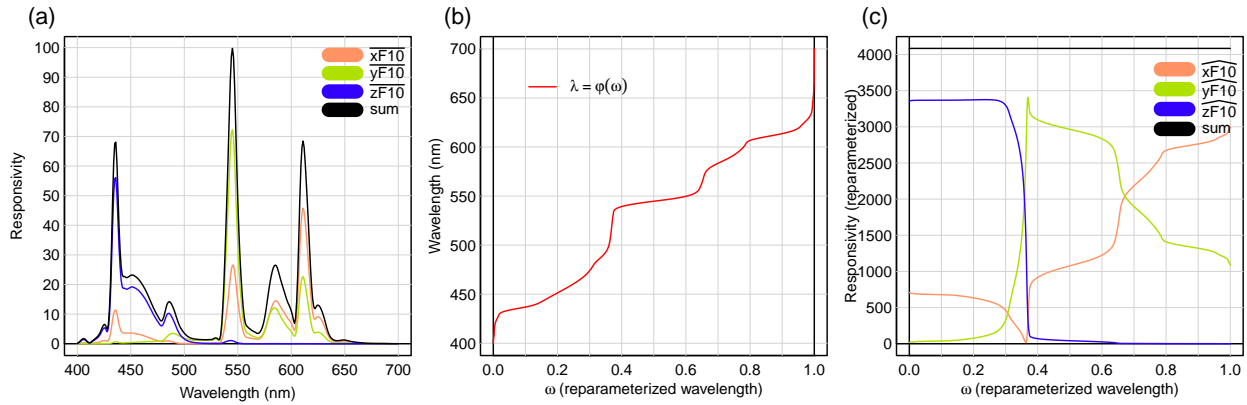


Figure 6: (a) the original responsivities of ‘F10.eye’, and their sum. (b) the reparameterization to interval [0,1]. (c) the equalized responsivities, and their sum.

Because the spectrum of F10 is so spiky, the equalized responsivities are much more irregular than they were for Illuminant E, compare with Figure 2. Now compute the XYZ responses, and compare the original spectra, and the spectra estimated from the responses.

```
XYZ = product( NCSU6, F10.eye, wavelength=wave )
est.eq = invert( F10.eye, XYZ, method='centroid', alpha=1 )
# range( extradata(est.eq)festim.precis ) # est.eqfestim.precis also works
plotOriginalPlusEstimates( list(NCSU6,est.eq), ymax=0.9 )
```

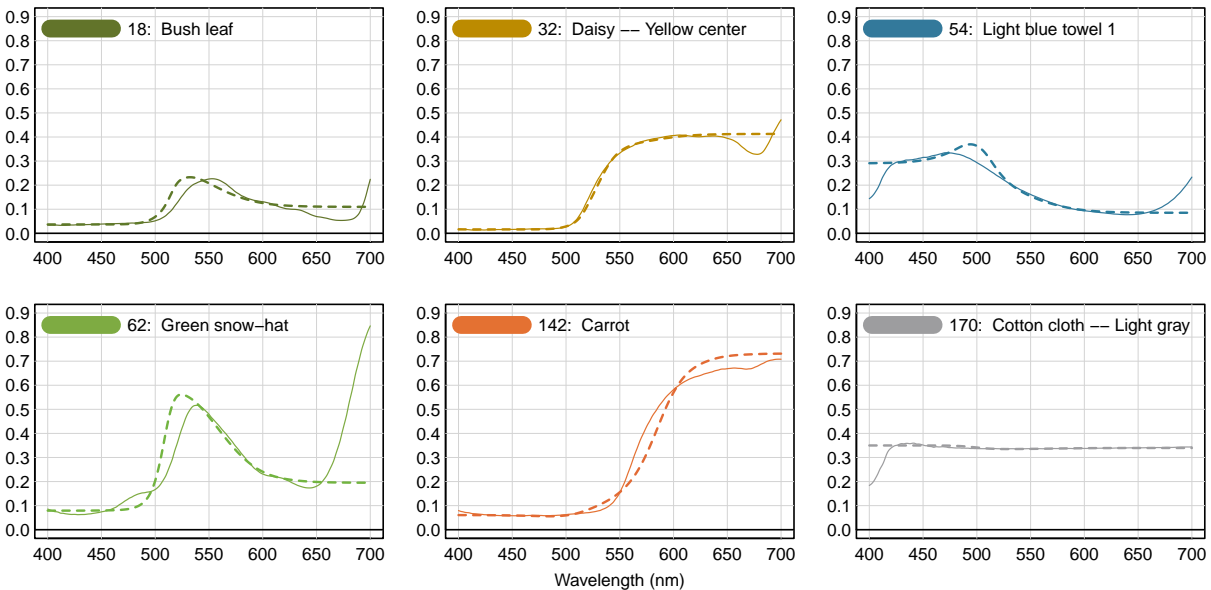


Figure 7: The original reflectance spectrum is solid linestyle, and the equalized estimate is dashed. The light source is Illuminant F10.

Despite the irregularities in the equalized responsivities, there is enough cancellation in the linear combinations that the estimated spectra are not that irregular; compare with Figure 3.

5 An Electronic Camera

Now we turn from the 1931 Standard Observer to an electronic camera. Make a “material responder” from Illuminant E and the Flea2.RGB camera. Compare the original and equalized responsivities

```
E.Flea2 = product( illuminantE(1,wave), "material", Flea2.RGB, wavelength=wave )
specnames( E.Flea2 ) = c('rE','gE','bE')
plotReparam3( E.Flea2 )
```

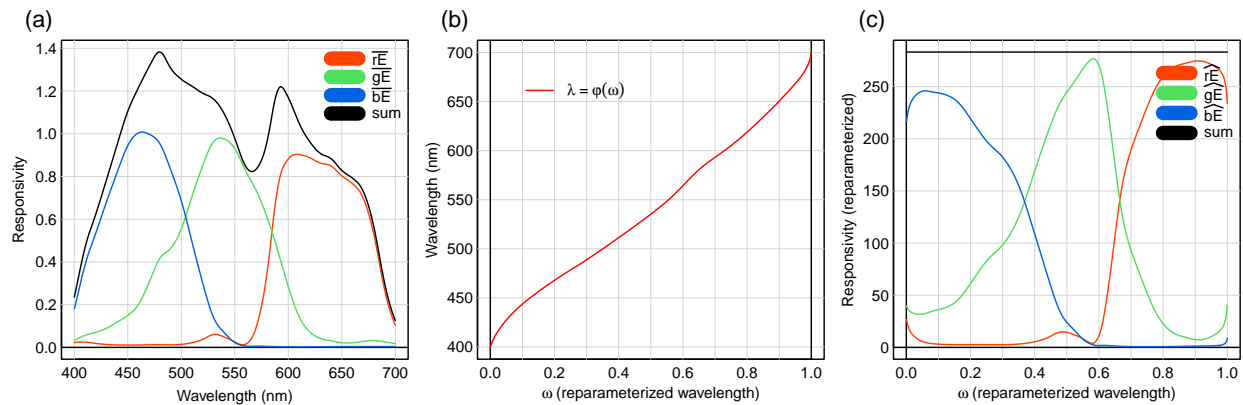


Figure 8: (a) the original responsivities of E.Flea2, and their sum. (b) the reparameterization to interval [0,1]. (c) the equalized responsivities, and their sum.

Compute the RGB responses, and compare the original spectra and the spectra estimated from the responses, see Figure 9. The results are comparable to those with a human eye, see Figure 3.

```
RGB      = product( NCSU6, E.Flea2, wavelength=wave )
est.eq   = invert( E.Flea2, RGB, method='centroid', alpha=1 )
# range( extradata(est.eq)$estim.precis )      # est.eq$estim.precis also works
plotOriginalPlusEstimates( list(NCSU6,est.eq), ymax=0.9 )
```

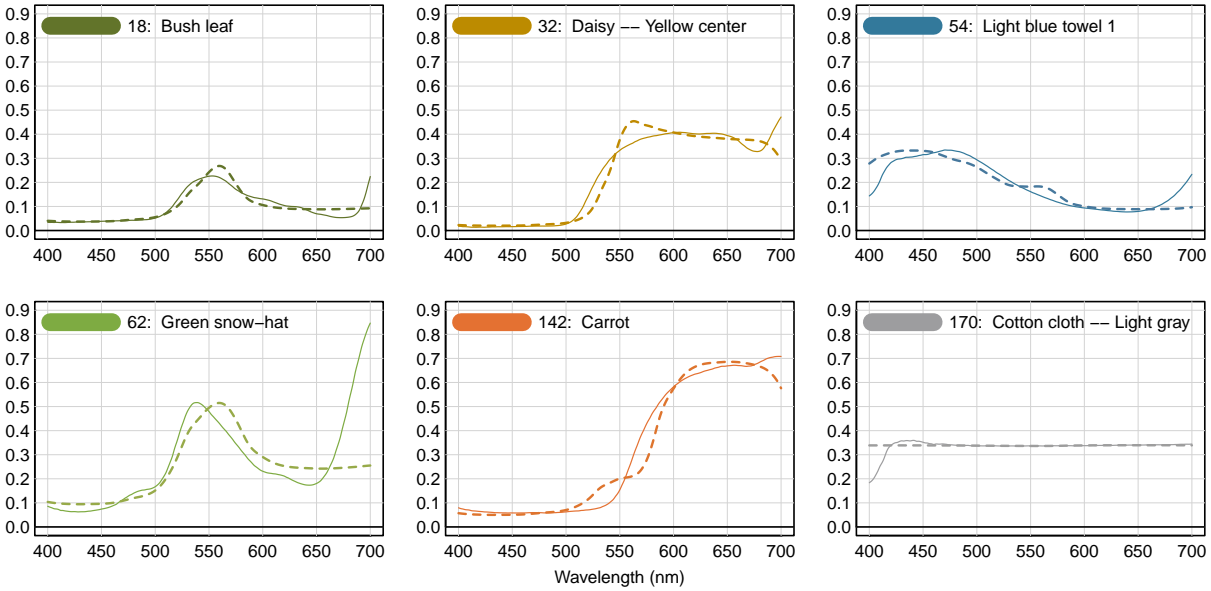



Figure 9: The original reflectance spectrum is solid linestyle; the equalized estimate is dashed. The light source is Illuminant E, and the camera is a Flea2.

Now we turn to techniques for improving the accuracy of the estimate. One classical technique is to keep the same camera, and add more light sources. Here, we will add Illuminant A, which has lots of energy at long wavelengths, and also add a black body radiator with a temperature of 9000 K, which has lots of energy at short wavelengths. There are now 9 responsivities, see Figure 10.

```
A.Flea2 = product( A.1nm, "material", Flea2.RGB, wavelength=wave )
specnames( A.Flea2 ) = c('rA','gA','bA')
P.Flea2 = product( planckSpectra(9000), "material", Flea2.RGB, wavelength=wave )
specnames( P.Flea2 ) = c('rP','gP','bP')
PEA.Flea2 = bind( P.Flea2, E.Flea2, A.Flea2 )
plotReparam3( PEA.Flea2 )
```

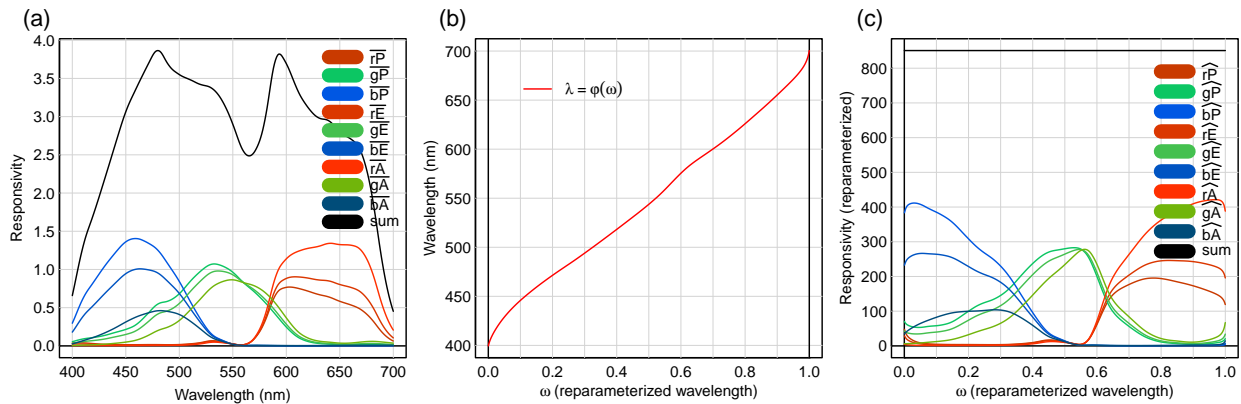


Figure 10: (a) the original responsivities of 'E.flea', and their sum. (b) the reparameterization to interval $[0,1]$. (c) the equalized responsivities using 3 different light sources, and their sum.

Compute the responses, and compare the original spectra and the spectra estimated from the responses. Note that each response is now a 9-vector instead of a 3-vector. See Figure 11.

```
response = product( NCSU6, PEA.Flea2, wavelength=wave )
est.eq    = invert( PEA.Flea2, response, method='centroid', alpha=1 )
# range( extradata(est.eq)$estim.precis ) # est.eq$estim.precis also works
plotOriginalPlusEstimates( list(NCSU6,est.eq), ymax=0.9 )
```

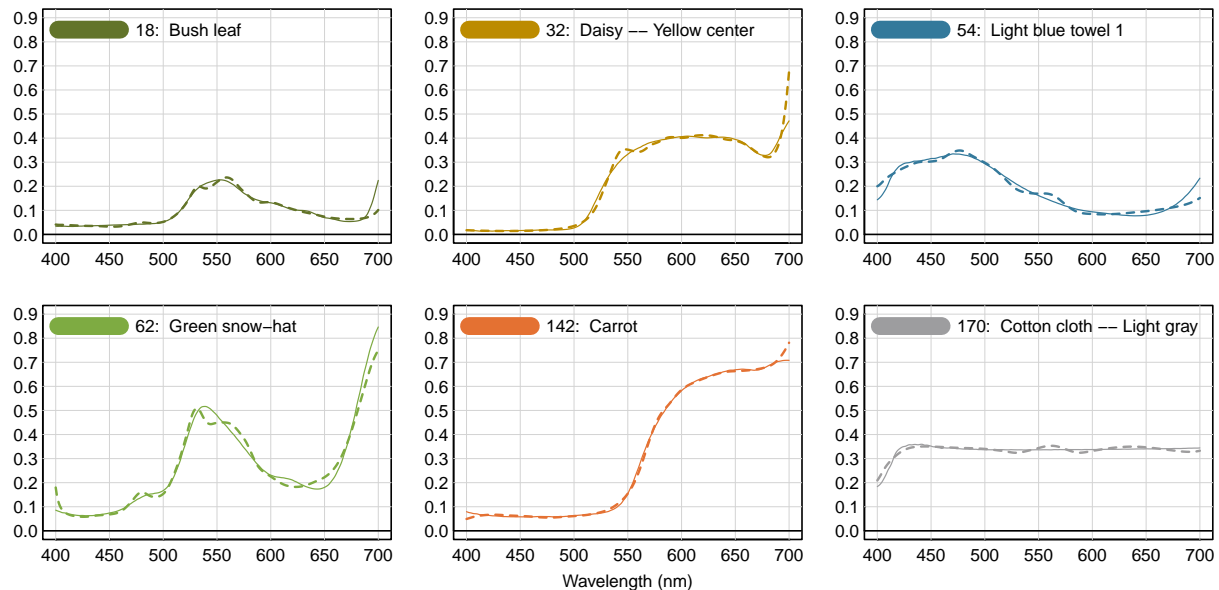


Figure 11: The original reflectance spectrum is solid linestyle; the equalized estimate is dashed. This estimate uses 3 different light sources.

The accuracy of all materials is now improved over Figure 9, except for 170 Cotton cloth – Light gray, which now has some undulations. It appears that the extra accuracy in matching the dip below 425nm has introduced these undulations. There’s a trade-off - oh well.

6 Estimating Light Sources

The centroid and TLSS methods also work for estimating light sources, but the mathematics is now different because there is no upper limit constraint on the energy spectrum of light. Create an arbitrary sequence of light spectra with increasing correlated color temperature (CCT), and then estimate the spectra from the XYZ response.

```
eye    = resample( xyz1931.1nm, wave )
spec   = planckSpectra( c(3000,4000), wave )
spec   = bind( spec, daylightSpectra( c(5000,6504,9000), wave ) )
spec   = bind( spec, planckSpectra( 11000, wave ) )
XYZ    = product( spec, eye )
est.c  = invert( eye, XYZ, method='centroid' ); est.t = invert( eye, XYZ, method='tlss' )
plotOriginalPlusEstimates( list(spec,est.c,est.t), mfrow=c(2,3), ymax=NA )
```

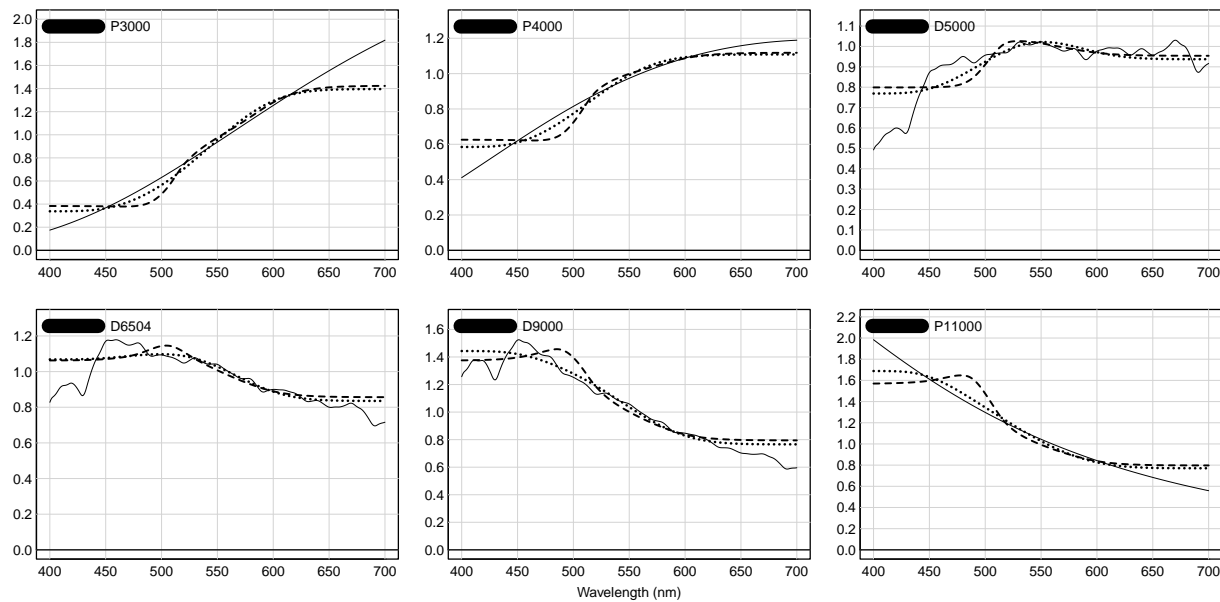


Figure 12: The original light source spectrum is solid linestyle; the centroid-based estimate is dashed, and the TLSS-based estimate is dotted.

These estimates are poor, but the TLSS-based ones are somewhat better. They tend to flatten out at the ends, while the originals do not. Moreover, there are some XYZ vectors for which the centroid method does not work at all; for more on this see Appendix B and [3] Section 13.

References

- [1] Simone Bianco. Reflectance spectra recovery from tristimulus values by adaptive estimation with metameric shape correction. *Journal of the Optical Society of America A*, 27(8):1868–1877, 2010. <https://www.osapublishing.org/josaa/abstract.cfm?uri=josaa-27-8-1868>.
- [2] Scott Burns. Generating Reflectance Curves from sRGB Triplets. (2019-06-04). URL:<http://scottburns.us/reflectance-curves-from-srgb/>.
- [3] Glenn Davis. A Centroid for Sections of a Cube in a Function Space, with application to Colorimetry. *ArXiv e-prints*, November 2018. <https://arxiv.org/abs/1811.00990>.
- [4] C J Hawkyard. Synthetic reflectance curves by additive mixing. *Journal of the Society of Dyers and Colourists*, 109(10):323–329, 1993. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1478-4408.1993.tb01507.x>.
- [5] J.J. Koenderink. *Color for the Sciences*. MIT Press, 2010.
- [6] Alexander Logvinenko. The colour cone. *Journal of the Optical Society of America A*, 32(2):314–322, 2015. <https://www.osapublishing.org/josaa/abstract.cfm?uri=josaa-32-2-314>.
- [7] Alexander D. Logvinenko. An object-color space. *Journal of Vision*, 9(11):5, 2009. <https://jov.arvojournals.org/article.aspx?articleid=2203976>.

- [8] Johannes Meng, Florian Simon, Johannes Hanika, and Carsten Dachsbacher. Physically meaningful rendering using tristimulus colours. *Computer Graphics Forum*, 34(4):31–40, 2015. <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12676>.
- [9] Erwin Schrödinger. Theorie der pigmente von größter leuchtkraft. *Annalen der Physik*, 367(15):603–622, 1920.
- [10] Michael J. Vrhel, Ron Gershon, and Lawrence S. Iwan. Measurement and analysis of object reflectance spectra. *Color Research & Application*, 19(1):4–9, 1994.
- [11] West, G. and Brill, M. H. Conditions under which Schrödinger object colors are optimal. *Journal of the Optical Society of America*, 73:1223–1225, 1983.

Appendix A - The Macbeth ColorChecker™

In Figure 7 of the supplemental information of Meng et. al. [8] is a full page of spectral estimates for the Macbeth ColorChecker. Figure 13 is an imitation of their Figure 7 (although the 24 original reflectances seem to be a little different).

```
wave = seq( 400, 700, by=5 )
E.eye = product( illuminantE(1,wave), "material", xyz1931.1nm, wavelength=wave )
path = system.file( 'extdata/targets/CC_Avg30_spectrum_CGATS.txt', package='colorSpec' )
MacbethCC = readSpectra( path, wavelength=wave )
MacbethCC = subset( MacbethCC, c( 21:24, 17:20, 13:16, 9:12, 5:8, 1:4 ) )
XYZ = product( MacbethCC, E.eye, wavelength=wave )
est.ce = invert( E.eye, XYZ, method='centroid', alpha=1 )
est.tl = invert( E.eye, XYZ, method='TLSS' )
plotOriginalPlusEstimates( list(MacbethCC,est.ce,est.tl), ymax=NA, mfrow=c(6,4) )
```

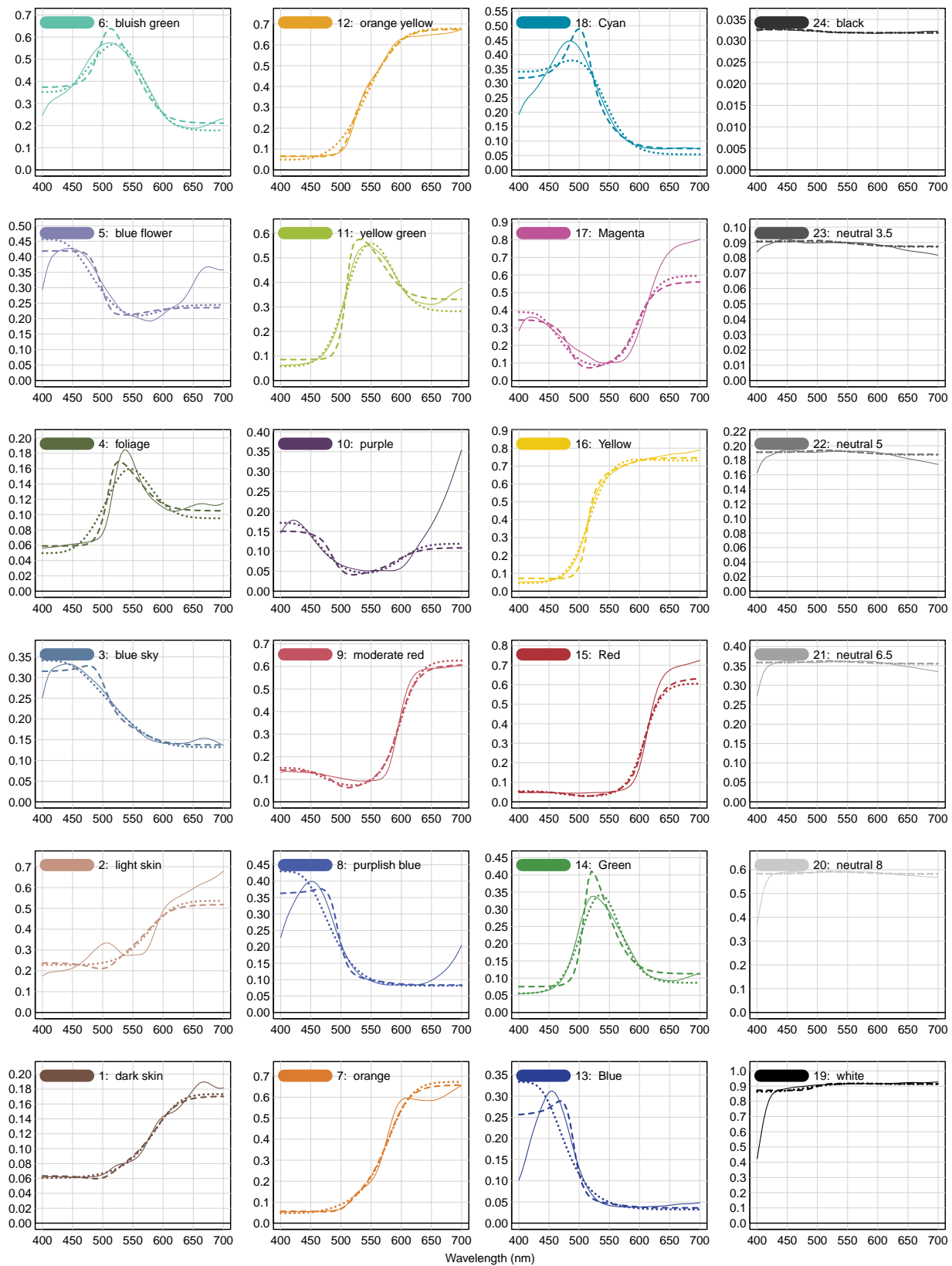


Figure 13: The original reflectance spectrum is solid linestyle, the centroid-based estimate is dashed, and the TLSS-based estimate is dotted. In each plot the 3 spectra are metameric for Illuminant E.

Appendix B - Inverse Colorimetry

This appendix is a formal mathematical view of *inverse colorimetry*. Let Q^∞ be the space of all (measurable) functions from the interval $[\lambda_1, \lambda_2]$ to $[0, 1]$. Each function in Q^∞ corresponds to the reflectance function of a material. A “material responder” is a linear operator $\Lambda : Q^\infty \rightarrow \mathbb{R}^3$, where points in \mathbb{R}^3 are tristimulus responses. Define $Z := \Lambda(Q^\infty)$; Z is a compact convex body - the *object color solid* (*Rösch Farbkörper*). Points in the boundary ∂Z correspond to *optimal colors* (*Optimalfarben*). Non-optimal colors correspond to points in the interior $\text{int}(Z)$. The centroid method for spectral reflectance estimation is a continuous function $\gamma : \text{int}(Z) \rightarrow \text{int}(Q^\infty)$ so that

the composition $\text{int}(Z) \xrightarrow{\gamma} \text{int}(Q^\infty) \xrightarrow{\Lambda} \text{int}(Z)$ is the identity on $\text{int}(Z)$.

Whenever $\Lambda \circ \gamma = \text{id}_{\text{int}(Z)}$, γ is said to be a *right-inverse for Λ* (or a *section of Λ*). Note that γ is not defined for optimal colors in ∂Z . For more details on γ , see Davis [3]. There are many other possible right-inverses. Assuming that Λ has the property that every optimal color comes from a spectrum with ≤ 2 transitions (jumps), Alexander Logvinenko has constructed a right-inverse that is defined on all of Z (including ∂Z) and whose image is a 3D space of step functions (the *rectangle color atlas*) that is the same for every such Λ , see [7].

There is a similar situation for a “light responder” except that the domain Q^∞ is replaced by the *non-negative orthant* L_+^1 , which is all integrable functions from the interval $[\lambda_1, \lambda_2]$ to $[0, \infty)$. Each function in L_+^1 represents the spectrum of a source of light. The image $C := \Lambda(L_+^1)$ is now an unbounded convex cone, see Logvinenko [6]. But in this unbounded case the centroid method right-inverse cannot be defined on all of C , but only on a proper subset of C . For more about this situation see Davis [3] Section 13.

Session Information

This document was prepared September 20, 2024 with the following configuration:

- R version 4.4.1 (2024-06-14), x86_64-pc-linux-gnu
- Running under: Ubuntu 24.04.1 LTS
- Matrix products: default
- BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
- LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-r0.3.26.so ; LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: colorSpec 1.5-0, knitr 1.48, rmarkdown 2.28, spacesRGB 1.5-0, spacesXYZ 1.3-0
- Loaded via a namespace (and not attached): MASS 7.3-61, R6 2.5.1, bslib 0.8.0, buildtools 1.0.0, cachem 1.1.0, cli 3.6.3, compiler 4.4.1, digest 0.6.37, evaluate 1.0.0, fastmap 1.2.0, highr 0.11, htmltools 0.5.8.1, jquerylib 0.1.4, jsonlite 1.8.8, lifecycle 1.0.4, maketools 1.3.0, microbenchmark 1.5.0, rlang 1.1.4, rootSolve 1.8.2.4, sass 0.4.9, sys 3.4.2, tools 4.4.1, xfun 0.47, yaml 2.3.10