Plotting Chromaticity Loci of Optimal and Schrödinger Colors

Reproducing plots of MacAdam Limits

Glenn Davis <gdavis@gluonics.com>

February 10, 2025

Introduction

The goal of this vignette is to reproduce 2 figures in [2], and to make similar ones. These figures plot sections of the surface of 2-transition colors. These colors are also called the *Schrödinger colors*, following [6]. The sections are often called the *MacAdam limits*, after [4].

To these plots we add sections of the the *optimal colors*, which are the colors on the boundary of the object color solid, which is also called the *Rösch Farbkörper*. The plots show that the difference is not significant; which confirms the statement in [1] that the difference: "... has no impact on practical colorimetric computations."

In all plots, the Schrödinger colors are plotted in black, and the optimal colors in red.

The featured functions from **colorSpec** used in this vignette are `sectionSchrodingerColors()` and `sectionOptimalColors()`. But they requires some help from the function `plotSections()` and others in the file `optimal-help.R`.

```
library( colorSpec )
source( "optimal-help.R" )

# make vector of levels to be used for the sections in all the plots
Ylevel=c( seq( 0.10, 0.90, by=0.1 ), 0.95 )
```

Illuminant A

First, build the "material responder" from Illuminant A and standard CMFs:

```
wave  = seq(380,800,by=2)
A.eye = product( A.1nm, "material", xyz1931.1nm, wavelength=wave )
white = product( neutralMaterial(1,wave=wave), A.eye )
```

Make the plot.

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
seclist = sectionOptimalColors( A.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, col='red' )
seclist = sectionSchrodingerColors( A.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, add=TRUE )
```

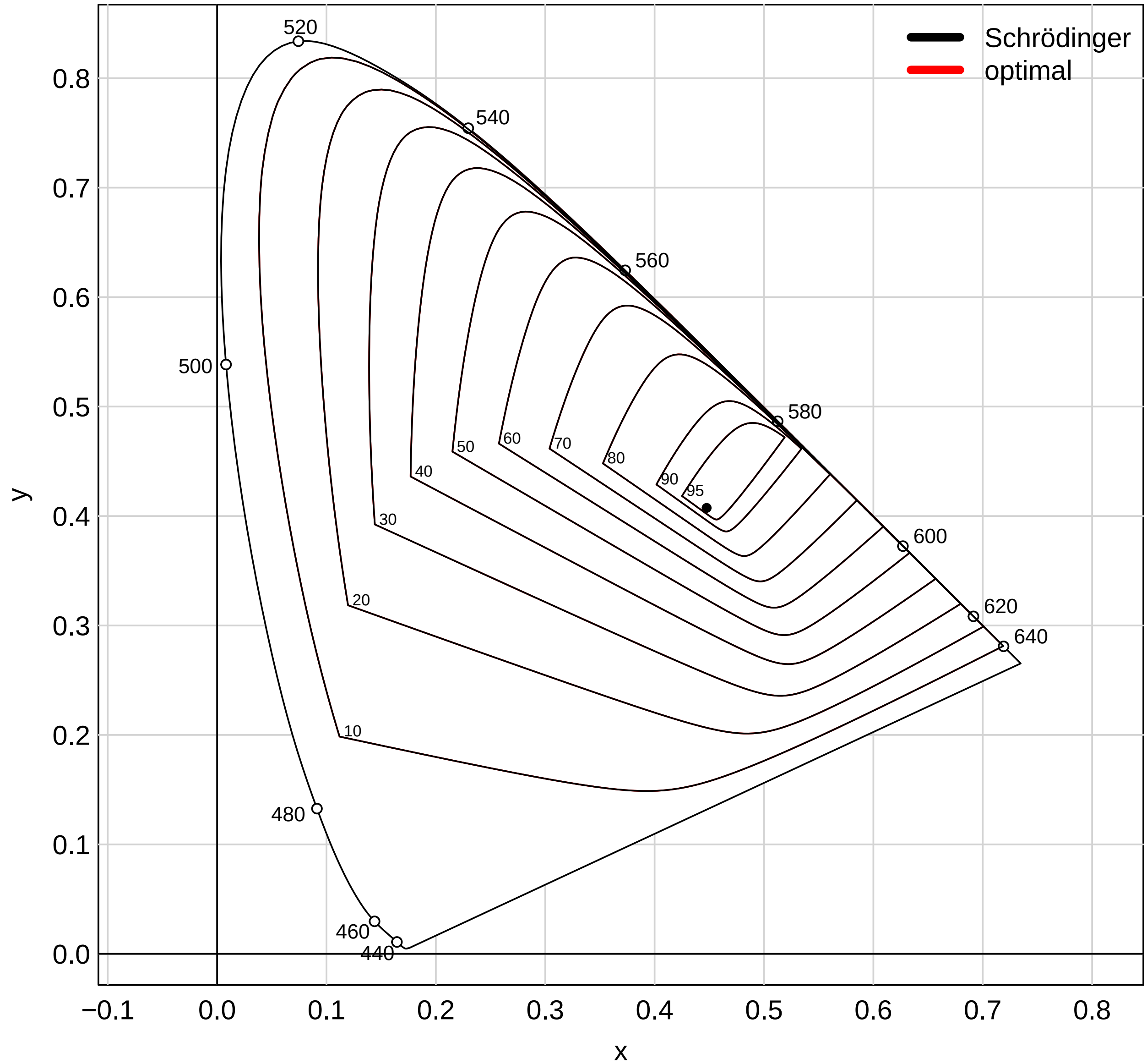


Figure 1: MacAdam Limits for Illuminant A

Compare this with Figure 2 in [4], and Figure 3(3.7) in [2]. The optimal colors are plotted first in red, and then the Schrödinger colors in black. As you can see, there is no significant difference between them. This is because the outer chromaticity diagram is almost convex.

Illuminant D65

First, build the "material responder" from Illuminant D65 and standard CMFs:

```
wave  = seq(380,800,by=2)
D65.eye = product( D65.1nm, "material", xyz1931.1nm, wavelength=wave )
white = product( neutralMaterial(1,wave=wave), D65.eye )
```

Make the plot:

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
seclist = sectionOptimalColors( D65.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, col='red' )
seclist = sectionSchrodingerColors( D65.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, add=TRUE )
```

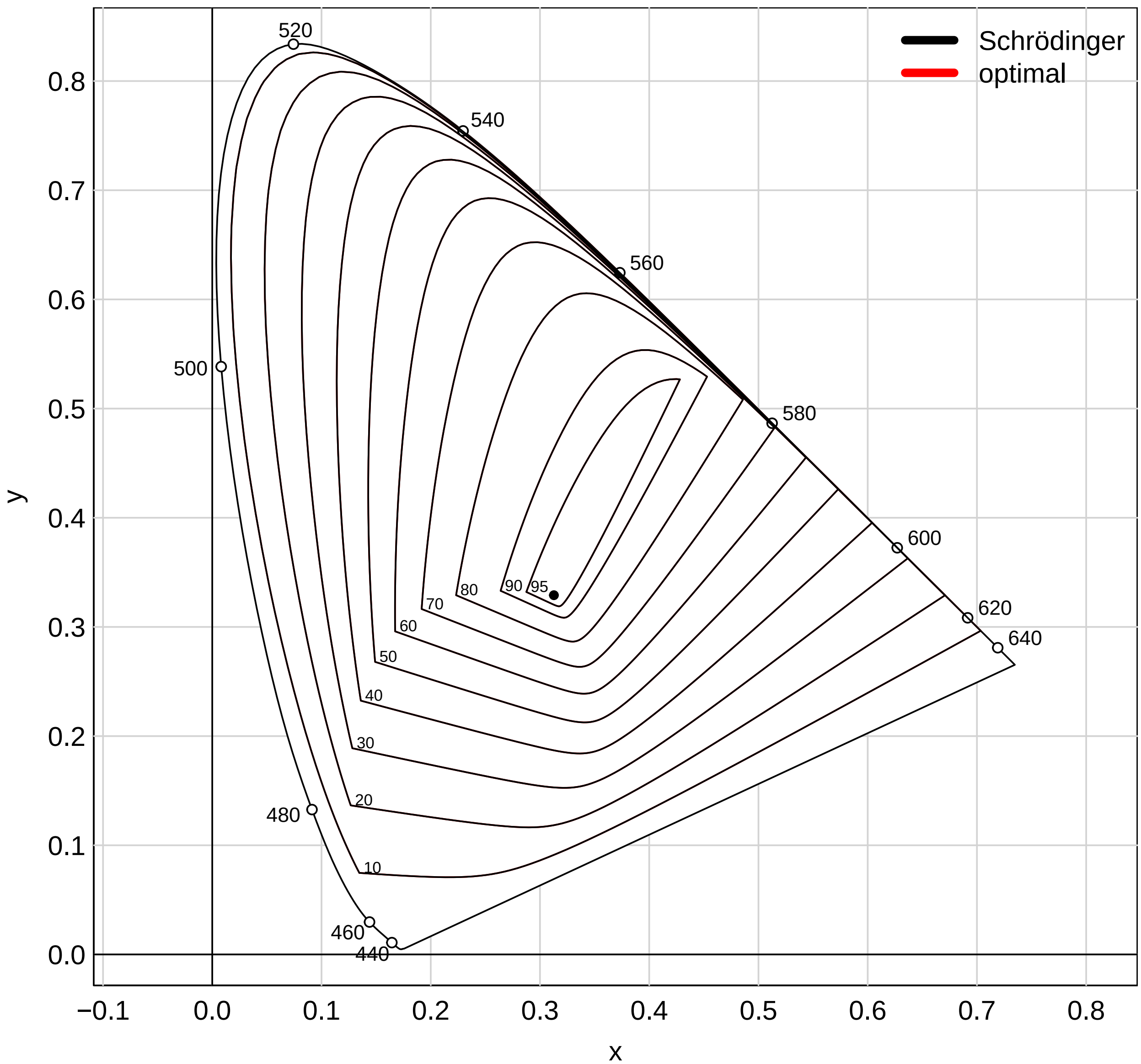


Figure 2: MacAdam Limits for Illuminant D65

Compare this with Figure 4(3.7) in [2].

Illuminant D65, with the Cone Fundamentals of Stockman and Sharpe

First, build the "material responder" from Illuminant D65, but this time with the updated cone fundamentals from [5].

```
wave  = seq(380,800,by=5)
D65.eye = product( D65.1nm, "material", lms2000.1nm, wavelength=wave )
white = product( neutralMaterial(1,wave=wave), D65.eye )
```

Make the plot:

```
par( omi=rep(0,4), mai=c(0.5,.6,0,0) )
normal = c(1,1,1)/3  ;  beta = sum(white*normal) * Ylevel
seclist = sectionOptimalColors( D65.eye, normal=normal, beta=beta )
plotSections( seclist, Ylevel, lms2000.1nm , white, col='red' )
seclist = sectionSchrodingerColors( D65.eye, normal=normal, beta=beta )
plotSections( seclist, Ylevel, lms2000.1nm , white, add=TRUE )
```

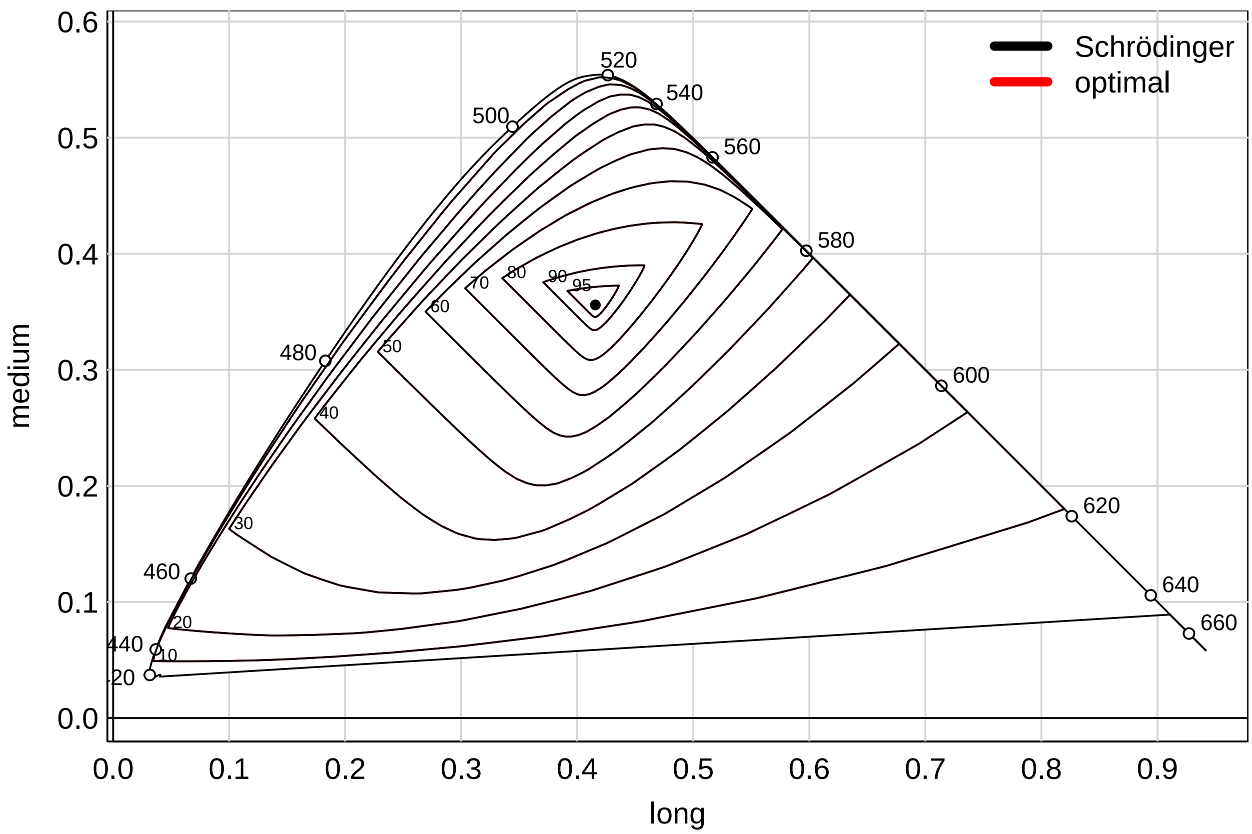


Figure 3: MacAdam Limits for Illuminant D65, with updated cone fundamentals

Compare with Figure 7 in [3]. Although the chromaticity polygon reverses itself on the right side, and is not convex. None of the levels are dark enough to reveal any significant difference between the red and black sections.

Illuminant C

First, build the "material responder" from Illuminant C and standard CMFs:

```
wave  = seq(380,780,by=2)
C.eye = product( C.5nm, "material", xyz1931.1nm, wavelength=wave )
white = product( neutralMaterial(1,wave=wave), C.eye )
```

Make the plot.

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
seclist = sectionOptimalColors( C.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, col='red' )
seclist = sectionSchrodingerColors( C.eye, normal=c(0,1,0), beta=white[2]*Ylevel )
plotSections( seclist, Ylevel, xyz1931.1nm, white, add=TRUE )
```

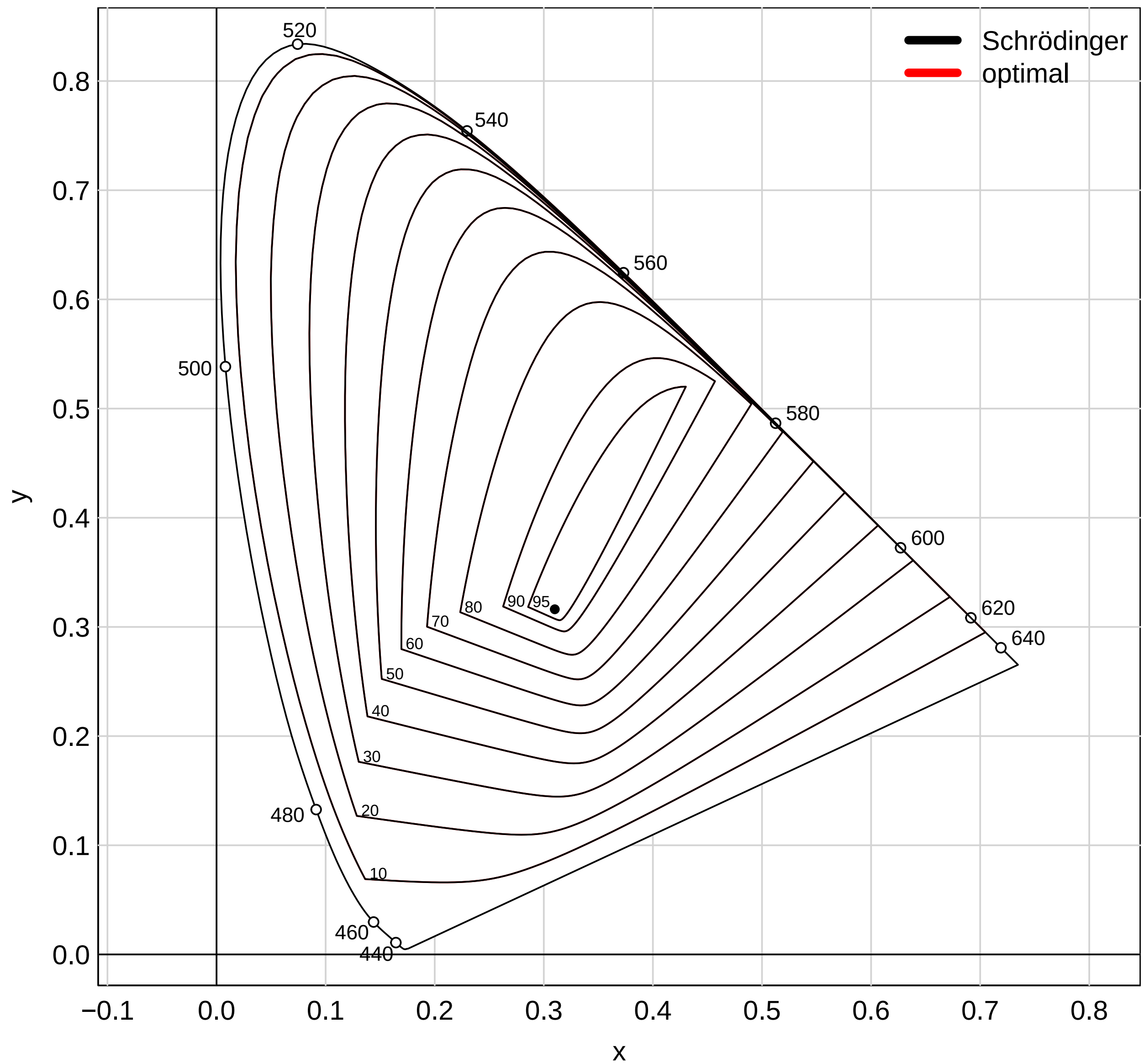


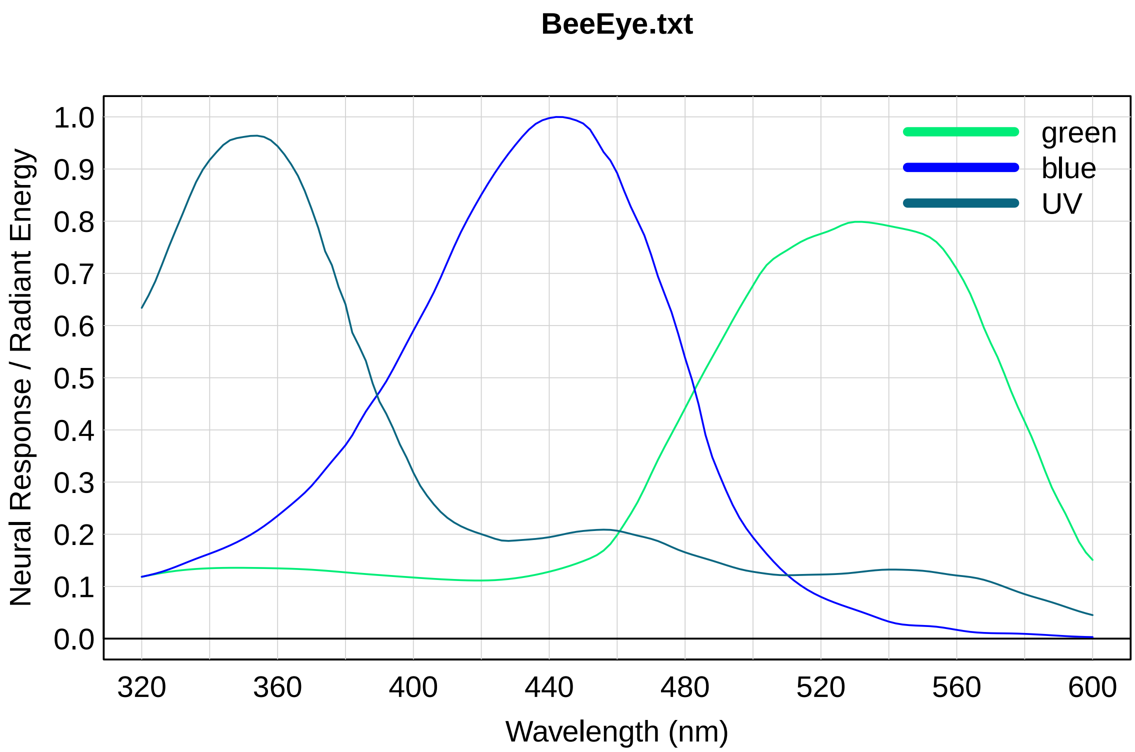
Figure 4: MacAdam Limits for Illuminant C

Compare this with Figure 3 in [4].

The Eye of a Bee

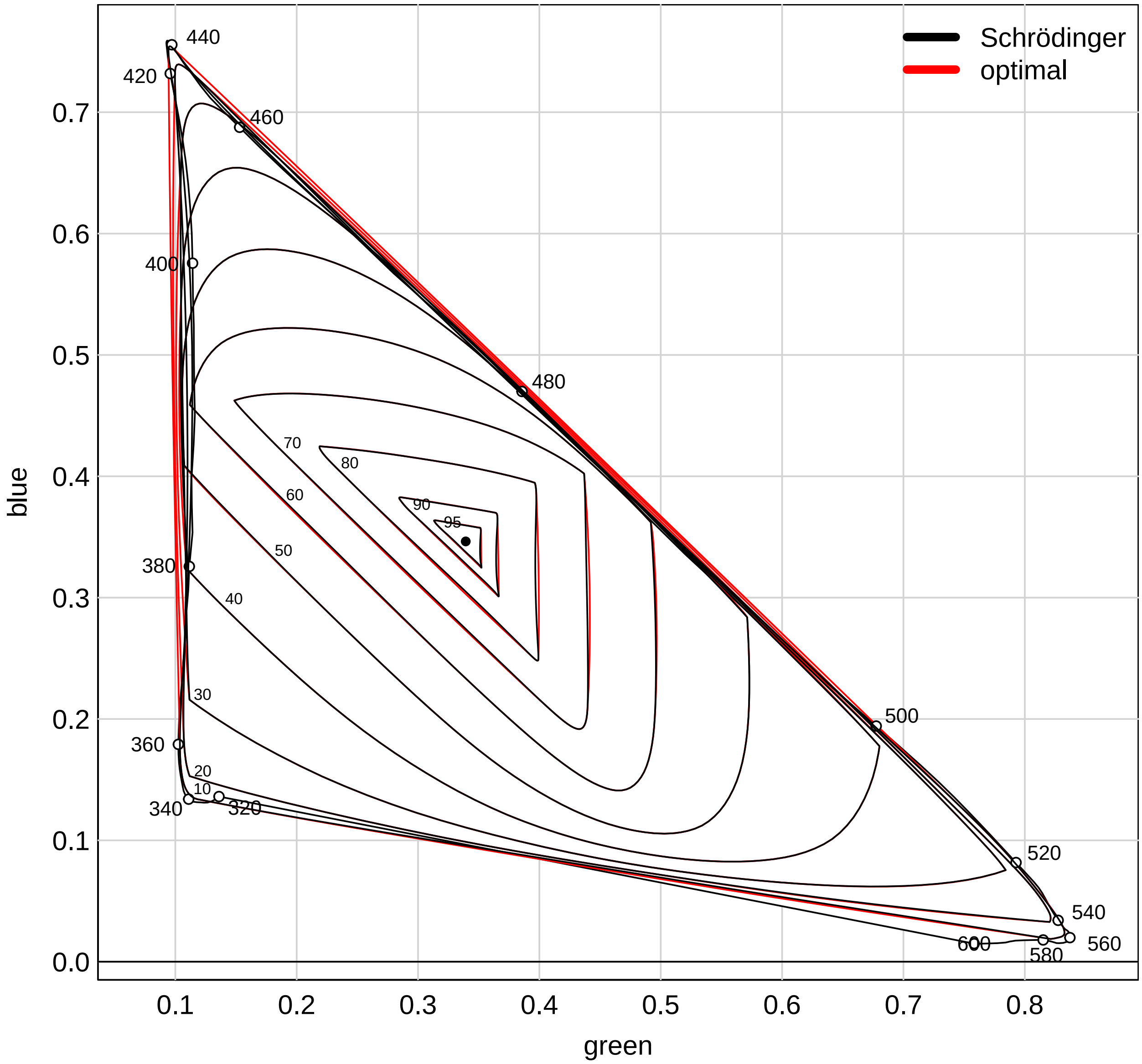
One of the first known examples of an eye with a non-convex chromaticity polygon is that of the eye of a bee. We use the plots from [6].

```
wave  = seq(320,600,by=2)
path  = system.file( 'extdata/eyes/BeeEye.txt', package='colorSpec' )
bee   = readSpectra( path, wavelength=wave )
plot( bee )
```



Compare with Figure 3(a) in [6].

```
E.eye = product( illuminantE(1,wavelength=wave), "material", bee )
white = product( neutralMaterial(1,wave=wave), E.eye )
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
normal = c(1,1,1)/3  ;  beta = sum(white*normal) * Ylevel
seclist = sectionOptimalColors( E.eye, normal=normal, beta=beta )
plotSections( seclist, Ylevel, bee, white, col='red' )
seclist = sectionSchrodingerColors( E.eye, normal=normal, beta=beta )
plotSections( seclist, Ylevel, bee, white, add=TRUE )
```



Compare with Figure 3(b) in [6]. For the first time in this vignette, we see a significant difference between the Schrödinger and optimal colors. Note that the optimal colors (red) are always *outside* the Schrödinger colors (black).

An RGB Scanner

This also works with object color from an electrical RGB scanner. The chromaticities in this case are:

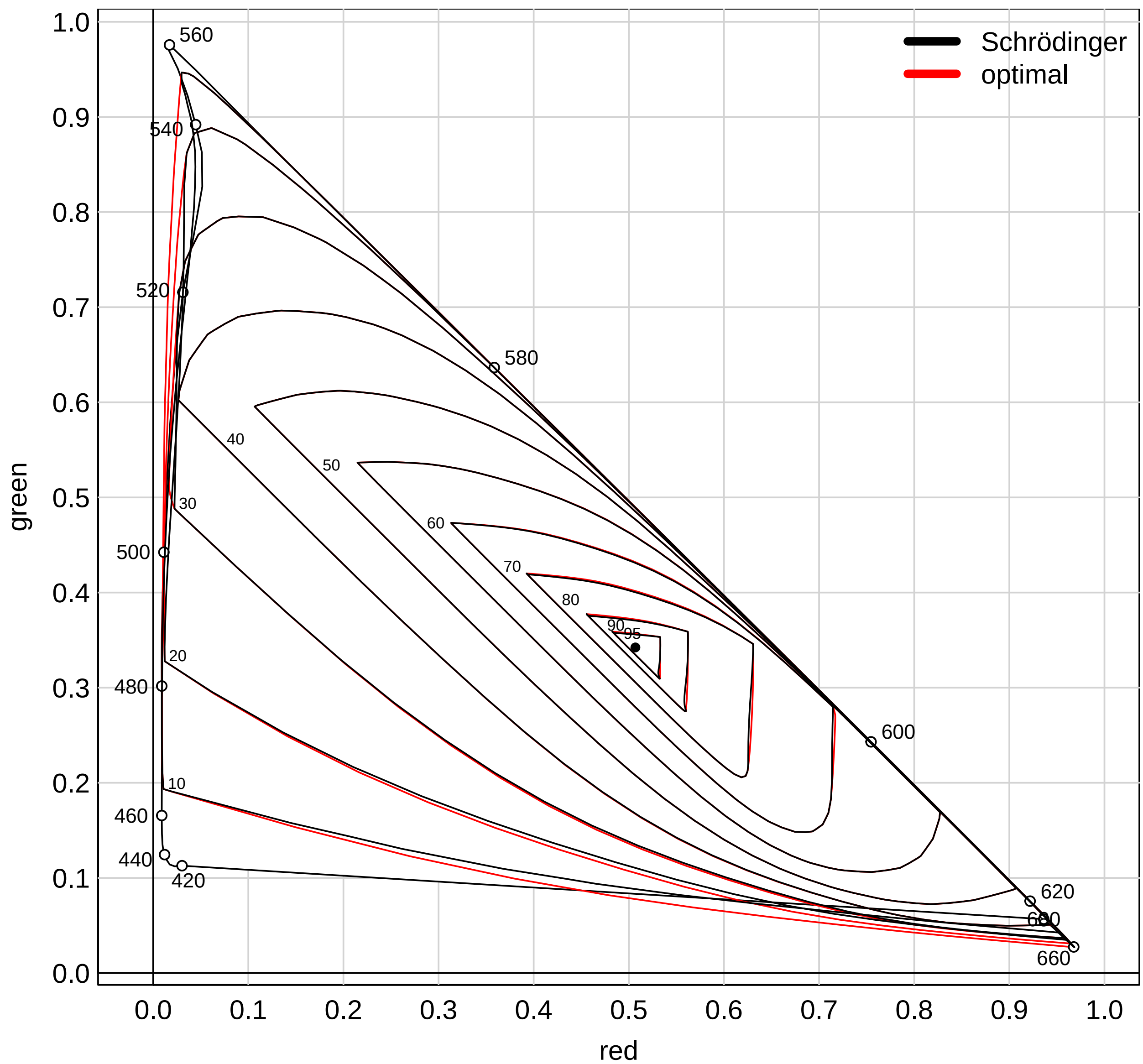
$$r = R/(R+G+B) \qquad g = G/(R+G+B)$$

Make a scanner from a tungsten source and a Flea2 camera:

```
wave = seq(420,680,by=5)
Flea2.scanner = product( A.1nm, "material", Flea2.RGB, wavelength=wave )
white = product( neutralMaterial(1,wave=wave), Flea2.scanner )
```

Make the plot.

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
normal = c(1,1,1)/3  ;  beta = sum(white*normal) * Ylevel
seclist = sectionOptimalColors( Flea2.scanner, normal=normal, beta=beta )
plotSections( seclist, Ylevel, Flea2.scanner, white, col='red' )
seclist = sectionSchrodingerColors( Flea2.scanner, normal=normal, beta=beta )
plotSections( seclist, Ylevel, Flea2.scanner, white, add=TRUE )
```



The wavelengths have been trimmed at each end to avoid weak responsivities that wander around too much. Even after trimming, the spectrum locus is not convex in the inverval from about 500 to 560 nm. This shows that the color solid does *not* satisfy the 2-transition property. The optimal color chromaticity locii are convex, as they must be; and some optimal colors are outside the spectrum locus. See Figure 3 and more discussion in [6].

References

[1] Scott A. Burns. The location of optimal object colors with more than two transitions. *Color Research & Application*, 46(6):1180–1193, 2021.

[2] Gunther Wyszecki and W.S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulae.* Wiley-Interscience, second edition, 1982.

[3] Alexander D. Logvinenko. An object-color space. *Journal of Vision*, 9(11):5, 2009. `https://jov.arvojournals.org/article.aspx?articleid=2203976`.

[4] David L. MacAdam. Maximum visual efficiency of colored materials. *Journal of the Optical Society of America*, 25:249–252, 1935.

[5] Andrew Stockman and Lindsay T. Sharpe. The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype. *Vision Research*, 40(13):1711–1737, 2000.

[6] 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

This document was prepared February 10, 2025 with the following configuration:

- R version 4.4.2 (2024-10-31 ucrt), `x86_64-w64-mingw32`
- Running under: `Windows 11 x64 (build 26100)`
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: colorSpec 1.7-0, knitr 1.49, spacesRGB 1.7-0, spacesXYZ 1.5-1
- Loaded via a namespace (and not attached): MASS 7.3-61, R6 2.5.1, bslib 0.8.0, cachem 1.1.0, cli 3.6.3, compiler 4.4.2, digest 0.6.37, evaluate 1.0.1, fastmap 1.2.0, glue 1.8.0, highr 0.11, htmltools 0.5.8.1, jquerylib 0.1.4, jsonlite 1.8.9, lifecycle 1.0.4, logger 0.4.0, microbenchmark 1.5.0, rlang 1.1.4, rmarkdown 2.29, rootSolve 1.8.2.4, sass 0.4.9, tools 4.4.2, xfun 0.49, yaml 2.3.10, zonohedra 0.4-0