Plotting Chromaticity Loci of Optimal Colors

Reproducing plots of MacAdam Limits

Glenn Davis <gdavis@gluonics.com>

November 18, 2018

Introduction

The goal of this vignette is to reproduce 2 figures in [Gun82], and to make similar ones. The key function from **colorSpec** used in this vignette is `probeOptimalColors()`. But it requires some help from the functions `computeOptimals()` and `plotOptimals()` in the file `optimal-help.R`.

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

Illuminant A

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

```
A.eye = product( A.1nm, "VARMATERIAL", xyz1931.1nm, wavelength='auto' )
```

Compute a data frame with 3600 rows. There are 10 gray levels and 360 angles for each level.

```
A.data = computeOptimals( A.eye, .angles=360 )
```

Now, make the plot:

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
plotOptimals( A.eye, A.data )
```

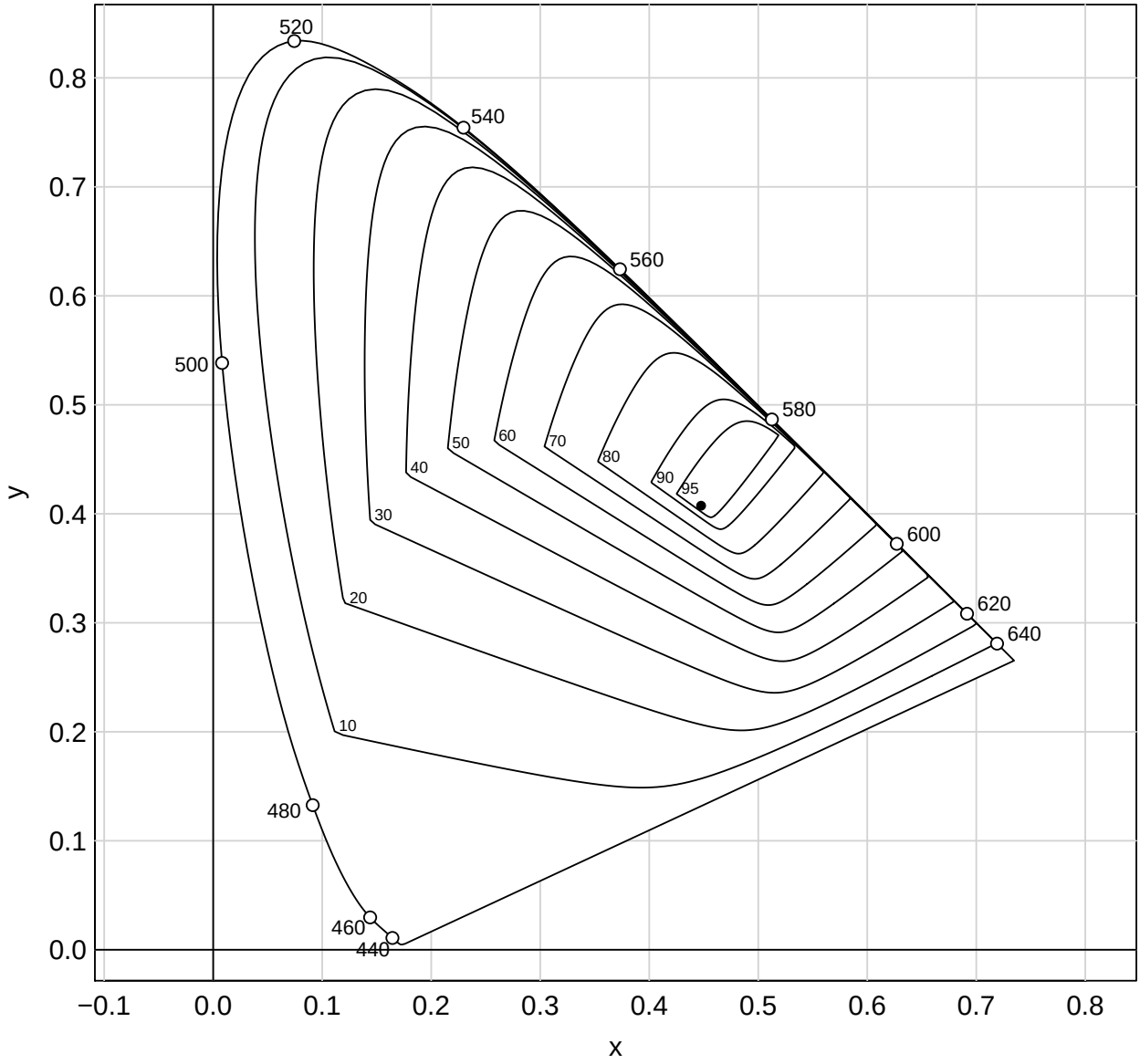

Figure 1: MacAdam Limits for Illuminant A

Compare this with Figure 3(3.7) from [Gun82].

Illuminant D65

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

```
D65.eye = product( D65.1nm, "VARMATERIAL", xyz1931.1nm, wavelength='auto' )
```

Compute a data frame with 3600 rows. There are 10 gray levels and 360 angles for each level.

```
D65.data = computeOptimals( D65.eye, .angles=360 )
```

Now, make the plot:

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
plotOptimals( D65.eye, D65.data )
```

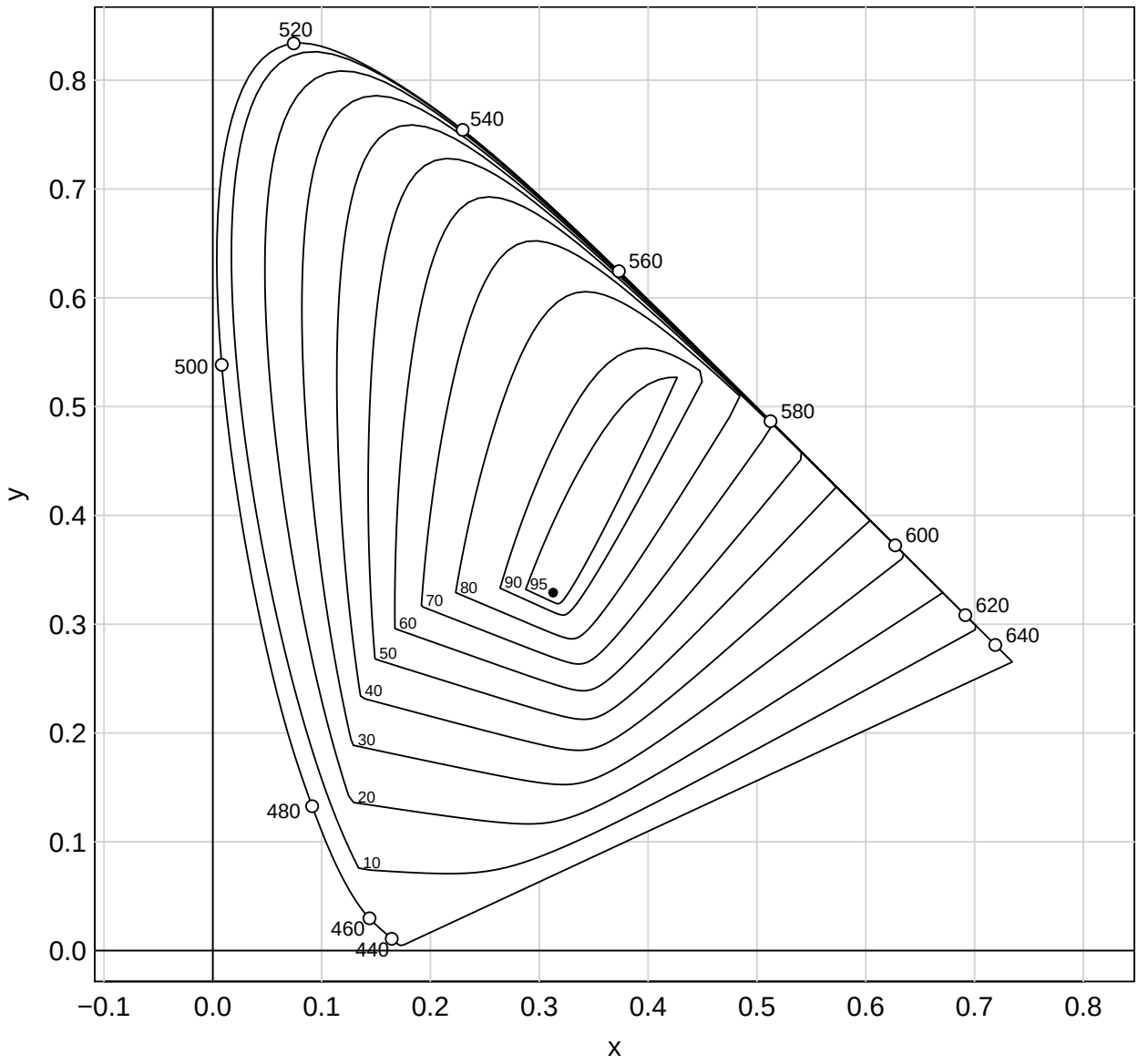

Figure 2: MacAdam Limits for Illuminant D65

Compare this with Figure 4(3.7) from [Gun82].

Illuminant C

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

```
C.eye = product( C.5nm, "VARMATERIAL", xyz1931.1nm, wavelength='auto' )
```

Compute a data frame with 3600 rows. There are 10 gray levels and 360 angles for each level.

```
C.data = computeOptimals( C.eye, .angles=360 )
```

Now, make the plot:

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
plotOptimals( C.eye, C.data )
```

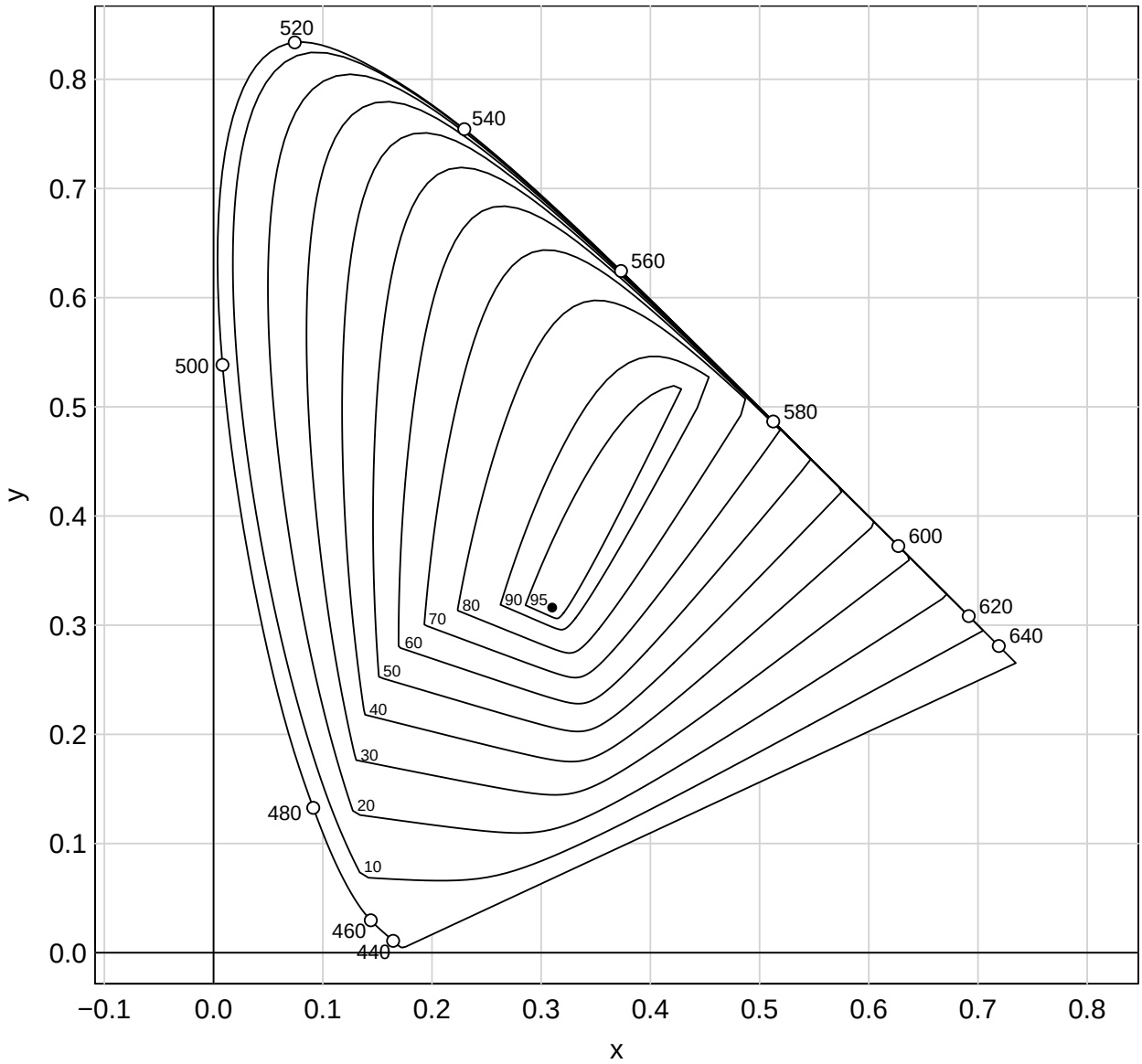

Figure 3: MacAdam Limits for Illuminant C

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:

```
Flea2.scanner = product( A.1nm, "VARMATERIAL", Flea2.RGB, wavelength=420:680 )
```

Compute a data frame with 3600 rows. There are 10 gray levels and 360 angles for each level.

```
Flea2.data = computeOptimals( Flea2.scanner, .angles=360 )
```

Now, make the plot:

```
par( omi=rep(0,4), mai=c(0.5,0.6,0,0) )
plotOptimals( Flea2.scanner, Flea2.data )
```

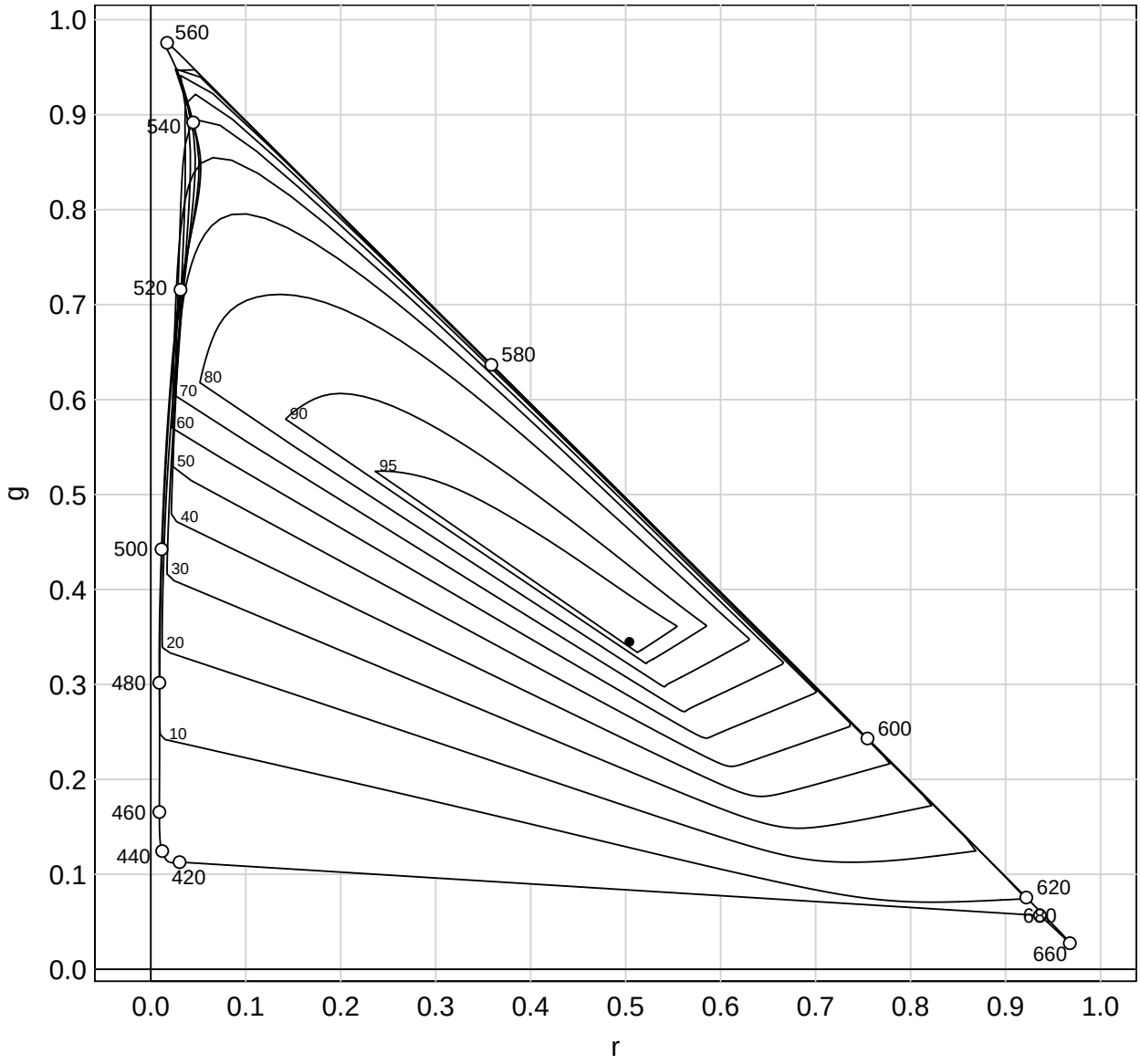

Figure 4: Approximate Output Limits for an RGB scanner

The wavelengths have been trimmed at each end to avoid weak responsivities that wander around too much. Note that the chromaticity diagram is not convex. The system does not satisfy the 2-transition assumption, and the so-called "optimal colors" here are not necessarily optimal, see [Wes83]. The limits here can be regarded as the approximate output limits of the scanner.

References

[Gun82] Gunther Wyszecki and W.S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulae*. Wiley-Interscience, second edition, 1982.

[Wes83] 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 November 18, 2018 with the following configuration:

- R version 3.5.1 (2018-07-02), `i386-w64-mingw32`
- Running under: `Windows 7 (build 7601) Service Pack 1`
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: colorSpec 0.7-5, knitr 1.20
- Loaded via a namespace (and not attached): MASS 7.3-50, Rcpp 0.12.19, backports 1.1.2, compiler 3.5.1, digest 0.6.17, evaluate 0.11, highr 0.7, htmltools 0.3.6, magrittr 1.5, microbenchmark 1.4-4, minpack.lm 1.2-1, rmarkdown 1.10, rootSolve 1.7, rprojroot 1.3-2, stringi 1.1.7, stringr 1.3.1, tools 3.5.1, yaml 2.2.0