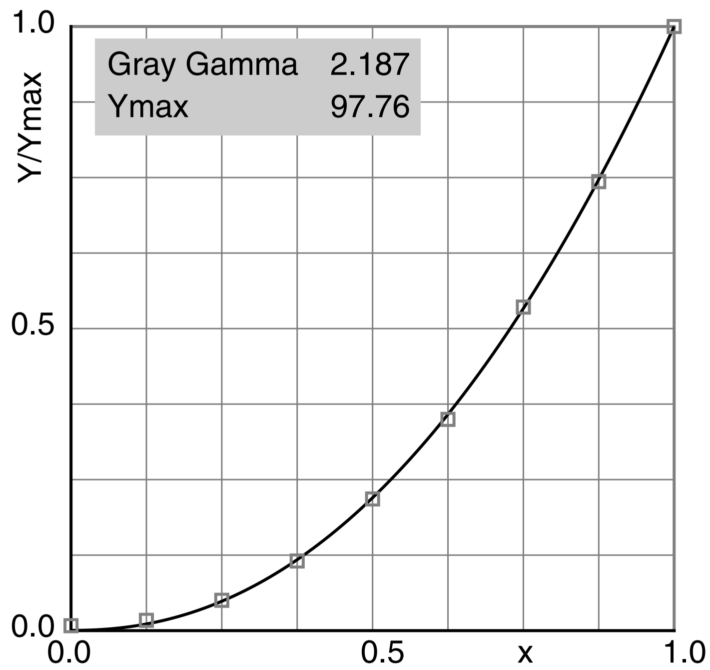
Gernot Hoffmann

Measuring Tone Reproduction Curves for CRT Monitors Examples for TFT Monitors

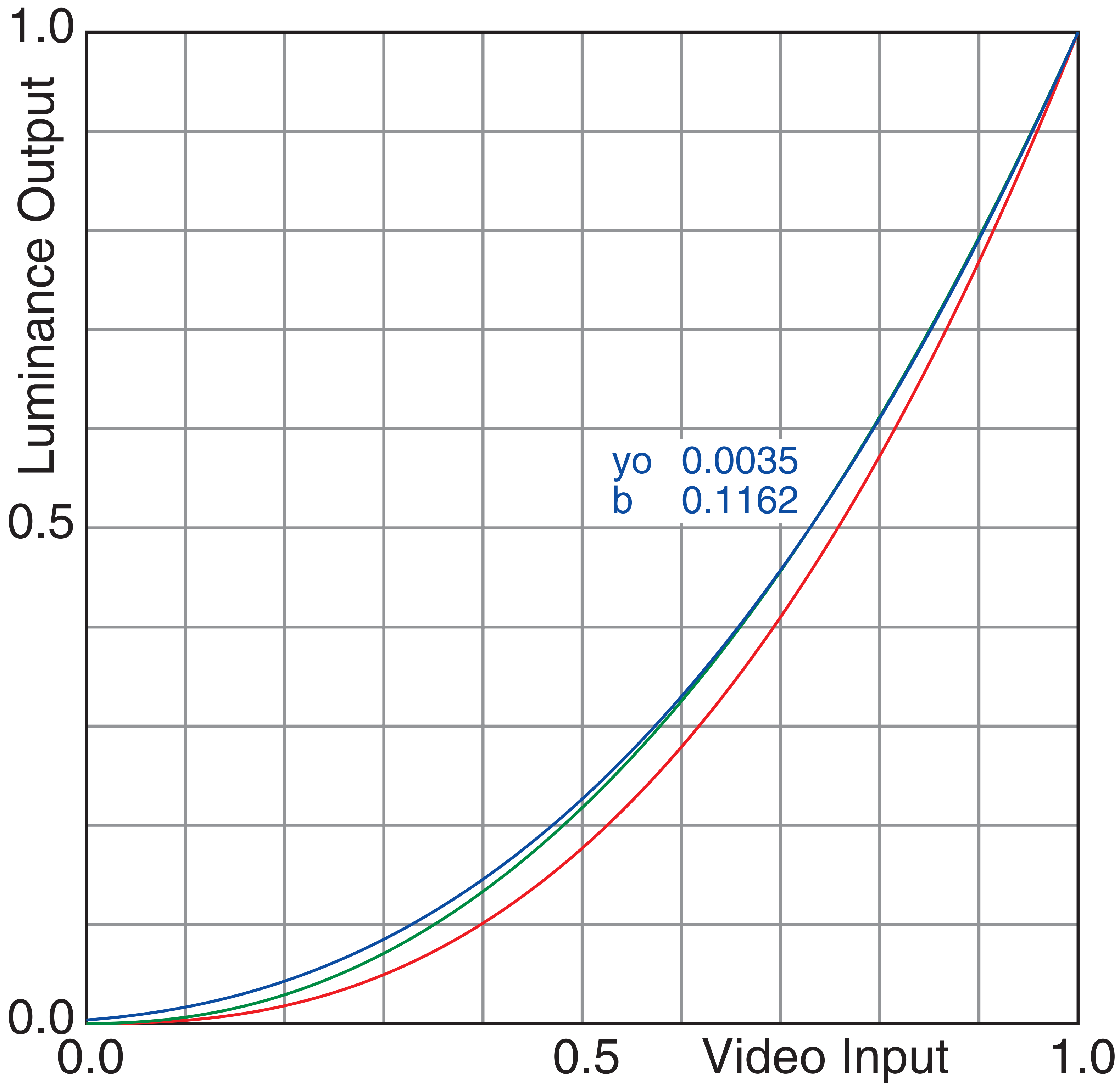


Contents

1. Introduction 2
2. Practical Application 4
3. Test Patterns and Diagrams 5
4. Mathematics 8
5. PostScript Code 11
6. Examples LaCie 19, Eizo CG19 17
7. Measuring the Generic TRC 21
8. Comparison Gamma=1.8 versus Gamma=2.2 22
9. References 24

1.1 Introduction

The diagram shows tone reproduction curves for one channel of a CRT monitor. The video input x is normalized for 0...1 for any channel R,G,B=0...255. The luminance output y(x) is normalized for 0...1. We assign 0 to black and 1 to the brightest available value.
A typical CRT monitor delivers the not-normalized luminance $L(x)=K(b+cx)^G$, according to [2], but somewhat simplified. The input offset b represents the brightness. c means contrast.
The function can be written normalized by $y(x)=(b+cx)^G/(b+c)^G$.
For generic uncalibrated monitors we have about G=2.5. This is the red curve for b=0.
The green curve shows the desired tone reproduction curve $y(x)=x^{2.2}$.
The maximal luminance is typically L_m=100 cd/m². The minimal luminance L_o is not zero but about 0.5 cd/m², here 0.35 cd/m².
b is calculated by $y(0)=y_o=L_o/L_m=b^G/(b+c)^G$, which delivers $b=c/(y_o^{-1/G}-1)$.
The blue curve is valid for y_o=0.0035 and c=1. The effective Gamma is near to 2.2.
Common test patterns with black/white stripes and a gray square, valid for L=0.5, would indicate indeed Gamma=2.2. Test patterns with several levels as in [4] would indicate the increased luminance at the dark end.
Obviously, the *effective* tone reproduction curve is much affected by the black level settings. Once the monitor is adjusted like here, then a further calibration by graphics card look-up tables (LUTs) does not require strong corrections.



1.2 Introduction

Now it is assumed that the monitor is calibrated:

1. by adjusting the black level as explained on the previous page.
2. by adjusting look-up tables on the graphics card, based on measurements by an instrument.

Which mathematical model should be used for the identification of the real tone reproduction curve (TRC)?

This is a simple Gamma function, in fact a power function:

Type A: $y = x^G$

$y = Y/Ymax$ is the normalized CIE XYZ luminance.
$x = RGB/255$ is the normalized input signal for either R,G,B or Gray R=G=B.
The exponent should be $G = 2.2$ for PC applications.
Test patterns provide RGB values for black (0) and eight steps (1...8) .
The test patterns are measured by an instrument like X-Rite DTP92 (Monitor Calibrator).
It turned out that the above model is not appropriate - the brightness offset is missing.

Therefore we use an improved model which contains additionally an offset y_o:

Type B: $y = y_o + (1-y_o)x^G$

The parameters y_o and G are optimized by a numerical function minimization, using the Steepest Descent algorithm (gradient methods).

$F = \text{Sum}\,(i=1..8)\,[y_o + (1-y_o)x_i^G - y_i]^2 = \text{Min}(y_o,G)$

The value for black is mostly uncertain and therefore *not* used.

Finally the model by *Kang* [1], which is equivalent to *Berns* [2], was tested:

Type C: $y = [y_o + (1-y_o)x]^G$

The numerical match works if negative bases are excluded, but the visual match is bad.
The numerical problems for negative bases can be overcome by a new formulation for the components s_i of the error sum:

Type C: $b_i = y_o + (1-y_o)x_i$
If $b_i \geq 0$ then $s_i = [b_i^G - y_i]^2$ else $s_i = 0$

The curve does not have a horizontal slope at x=0. Though this model is based on cathode ray tube physics it does not deliver convincing results.
The model is not appropriate, because the *calibrated* monitor should have a TRC which consists of a power function and a brightness offset (Type B).

Type A can be solved by a Gauss Least Squares approach by $u=\ln(x)$, $v=\ln(y)$ and $v=Gu$:
$G = \text{Sum}(i=1..8)[u_i v_i] / \text{Sum}(i=1..8)[u_i u_i]$. The results are very bad, because the minimization happens in the logarithmical space.

Improvement June 17-18 / 2008:
The test patterns are now defined accurately as integer numbers 0, 32, 64,..., 224, 255.
The inputs in the tables are now float numbers 0/255, 32/255, 64/255,...,224/255, 255/255.
The tables are now one-dimensional arrays: one for x, four for y (R,G,B,Gray).
Chapters 3.2, 5, 6.3 and Reference [5] were updated.

2. Practical Application

1. Show test patterns (next page) without any color management
   Professional Acrobat: Edit / Preferences / General / Color / CMS Off

2. Set instrument for Monitor Emissive / D65

3. Measure White Reference

4. Measure nine values Y for each channel R,G,B and for Gray

5. Write raw values Y into tables Taby in the PostScript EPS file

6. Open new RGB page A3 297mm x 420mm in Photoshop and place EPS file

7. The updated diagrams should be shown within one second

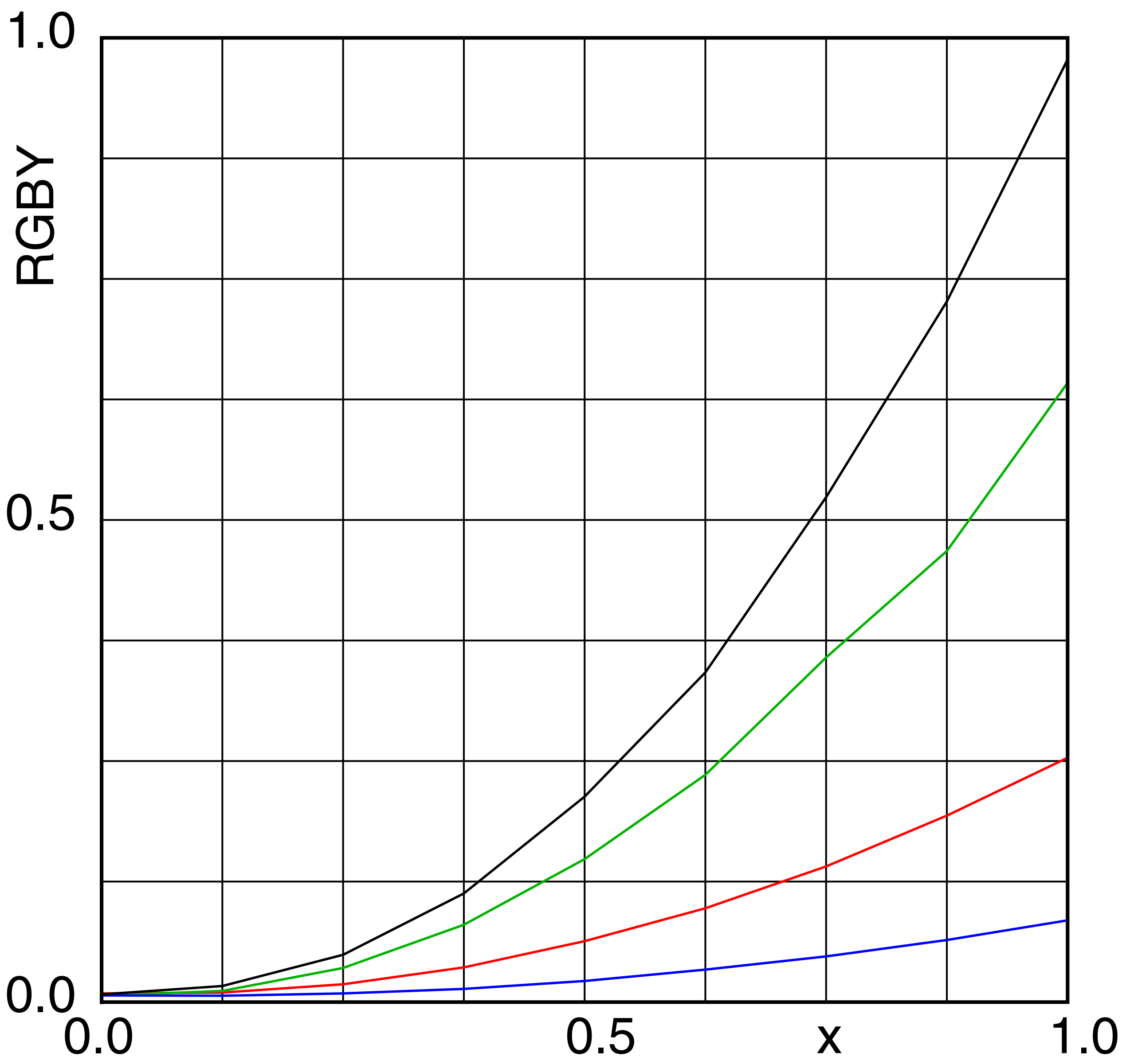
   Errors are mostly due to typos in the EPS code.

   Inaccuracies can be expected for Blue because of low luminance.

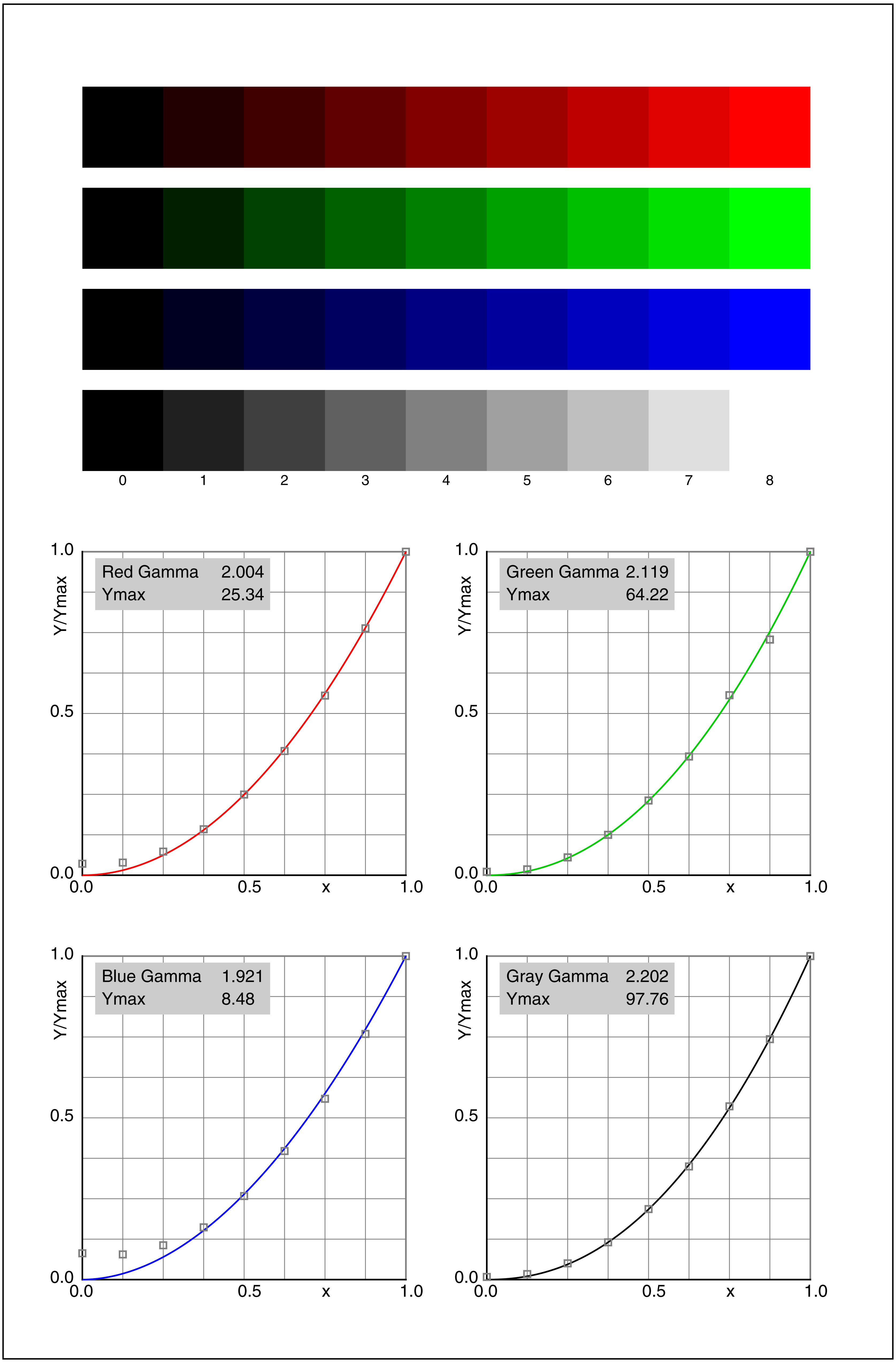
   The luminance for a patch is not the same for different positions on the screen.

   For EyeOne Pro and GMB ProfileMeasure Tool use the mode Spot and indicate CIE XYZ for measuring Y.

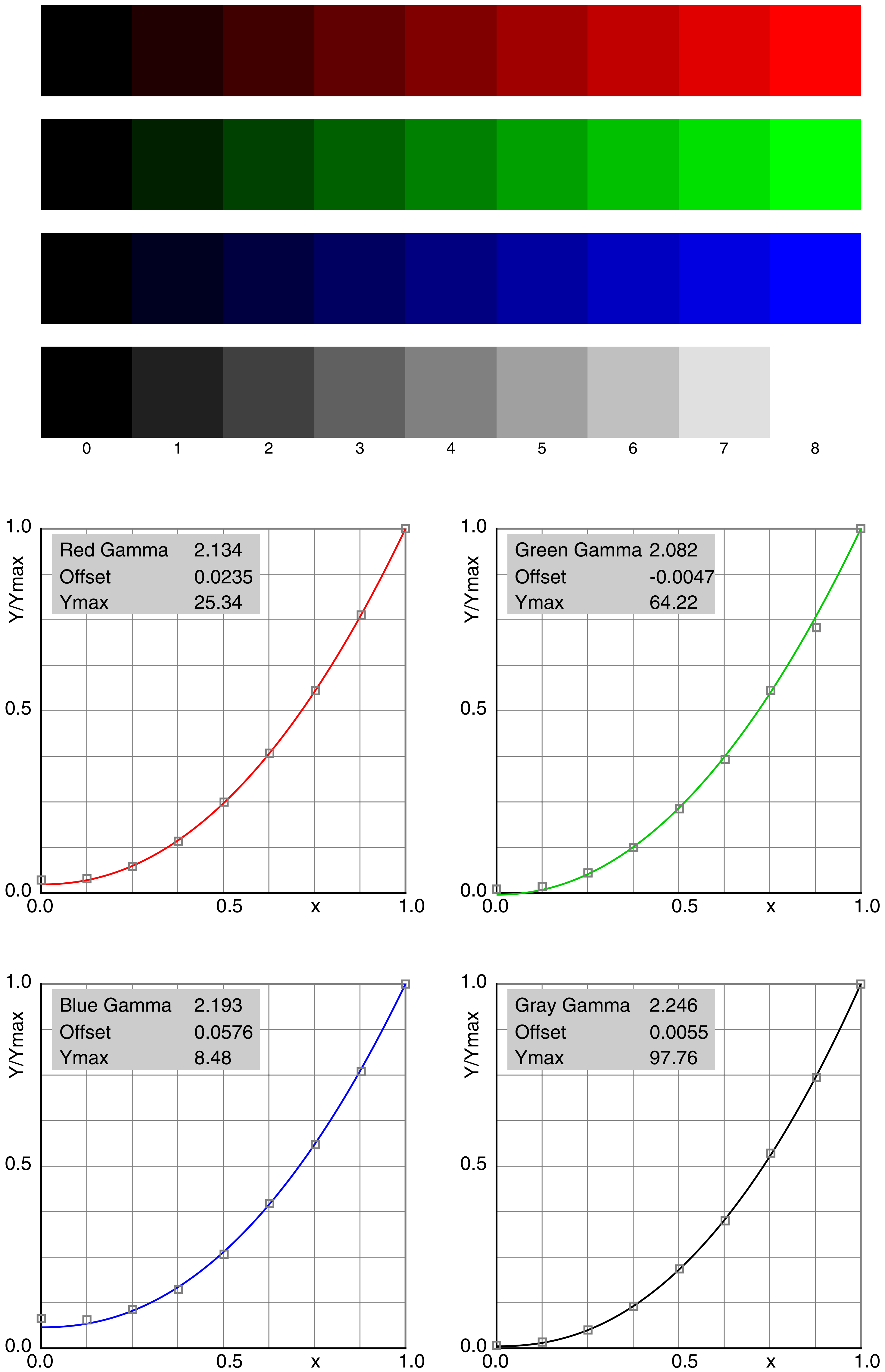
   This diagram shows typical raw luminance data.
   The maximum 1.0 represents for instance Y=100cd/m².



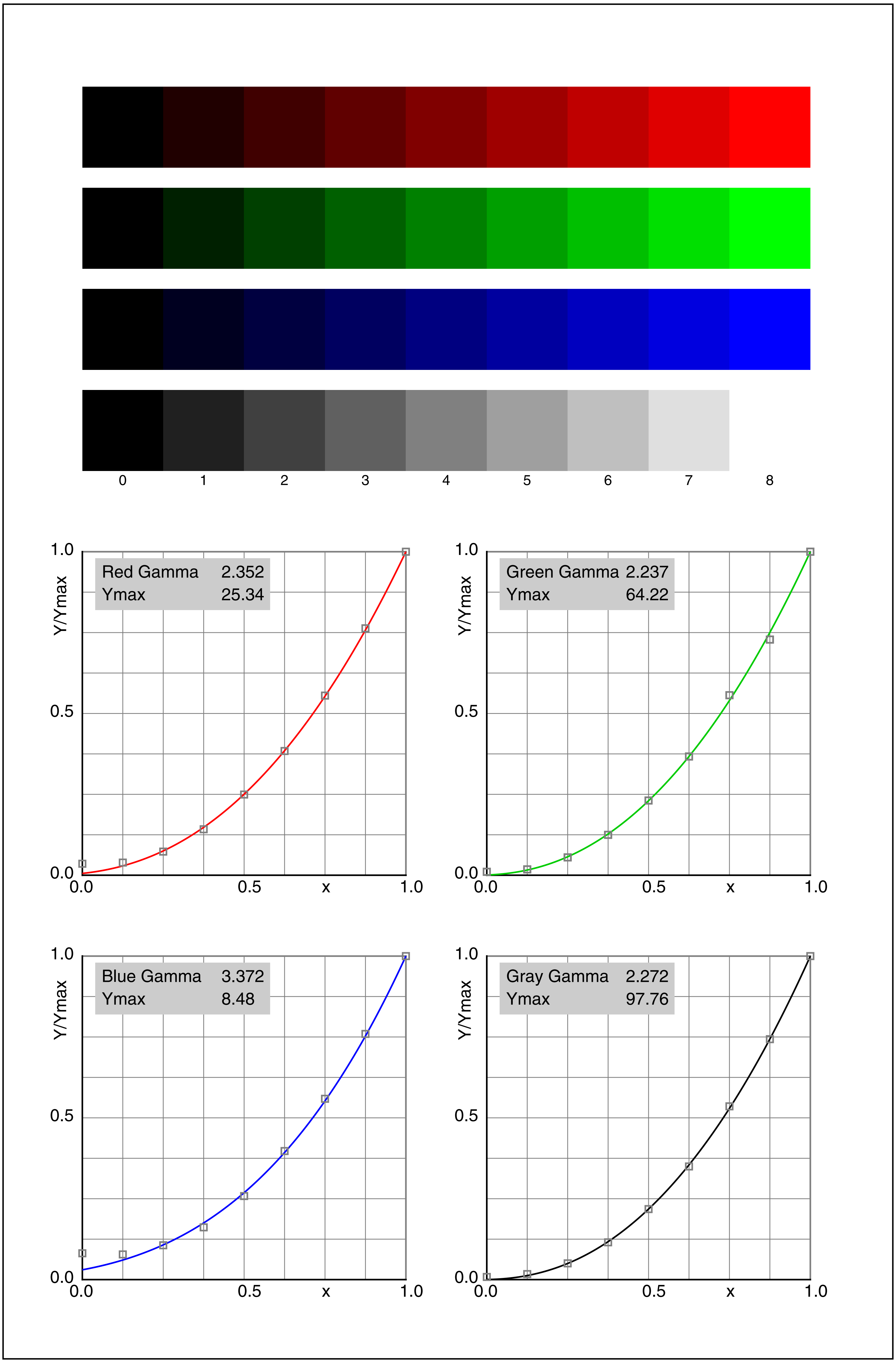
3.1 Test Patterns and Diagrams Type A



3.2 Test Patterns and Diagrams Type B (Standard)



3.3 Test Patterns and Diagrams Type C



4.1 Mathematics

Without one-dimensional search (simple, not used):

Steepest Descent Function Minimization
Without one-dimensional search

Function

$$F(p_1,p_2) = \sum_{i=1}^{8} [p_1 + (1-p_1)x_i^{p_2} - y_i]^2$$

Iteration, using numerical gradients

$p_1 \quad = \quad 0.0$
$p_2 \quad = \quad 1.0$
$\sigma \quad = \quad 0.7$
$\varepsilon \quad = \quad 10^{-8}$
$dp \quad = \quad 10^{-4}$
$k \quad = \quad 1$
$Fo \quad = \quad 999$
Begin loop
$Fu \quad = \quad F(p_1,p_2)$
If $abs(Fo - Fu) < \varepsilon$ Then exit
$Fo \quad = \quad Fu$
$g_1 \quad = \quad [F(p_1+dp,p_2) - Fo]/dp$
$g_2 \quad = \quad [F(p_1, p_2+dp) - Fo]/dp$
$dn \quad = \quad g_1^2 + g_2^2$
If $dn < \varepsilon$ Then exit
$fc \quad = \quad -\sigma Fo/dn$
$p_1 \quad = \quad p_1 + g_1 fc$
$p_2 \quad = \quad p_2 + g_2 fc$
$k \quad = \quad k+1$
If $k>1000$ Then exit
End loop

The step size is made slightly smaller than the theoretical value by $\sigma<1$.

4.2 Mathematics

One-dimensional search by comparison of ten samples (used in PostScript code):

Steepest Descent Function Minimization
Using one-dimensional search

Function

$$F(p_1,p_2) = \sum_{i=1}^{8} [p_1 + (1-p_1)x_i^{p_2} - y_i]^2$$

Iteration, using numerical gradients

$p_1 = 0$

$p_2 = 1.0$

$\sigma = 0.07$

$\varepsilon = 10^{-8}$

$dp = 10^{-4}$

$k = 1$

$Fo = 999$

Begin loop

$Fu = F(p_1,p_2)$

If $abs(Fu - Fo) < \varepsilon$ Then exit

$Fo = Fu$

$g_1 = [F(p_1+dp,p_2) - Fo]/dp$

$g_2 = [F(p_1, p_2+dp) - Fo]/dp$

$dn = g_1^2 + g_2^2$

If $dn < \varepsilon$ Then exit

$fc = -\sigma Fo/dn$

$r_1 = fc\, g_1$

$r_2 = fc\, g_2$

One-dimensional search

```
    q1 = p1
    q2 = p2
    Fm = Fo
    For j = 1 to 10 Do
    Begin
    p1 = p1 + r1
    p2 = p2 + r2
    If F(p1,p2) < Fm Then
       Begin
       Fm = F
       q1 = p1
       q2 = p2
       End
    End
```

$p_1 = q_1$

$p_2 = q_2$

$k = k+1$

If k>1000 Then exit

End loop

4.3 Mathematics

One-dimensional search by parabolic interpolation (not recommended):

Steepest Descent Function Minimization
Using parabolic one-dimensional search

Function

$$F(p_1,p_2)=\sum_{i=1}^{8}[p_1+(1-p_1)x_i^{p_2}-y_i]^2$$

Iteration, using numerical gradients

$p_1 = 0$
$p_2 = 1.0$
$\sigma = 0.07$
$h = 1.0$
$\varepsilon = 10^{-8}$
$dp = 10^{-4}$
$k = 1$
$Fo = 999$
Begin loop
$Fu = F(p_1,p_2)$
If $abs(Fu - Fo) < \varepsilon$ Then exit
$Fo = Fu$
$g_1 = [F(p_1+dp,p_2)-Fo]/dp$
$g_2 = [F(p_1,p_2+dp)-Fo]/dp$
$dn = g_1^2+g_2^2$
If $dn < \varepsilon$ Then exit
$fc = -h\,Fo/dn$
$r_1 = fc\,g_1$
$r_2 = fc\,g_2$
One-dimensional parabolic search $F(s)=a_0+a_1 sh+a_2(sh)^2$

- $q_1 = p_1$
- $q_2 = p_2$
- $F1 = F(p_1+0.5h\,r_1, p_2+0.5h\,r_2)$
- $F2 = F(p_1+h\,r_1, p_2+h\,r_2)$
- $a_2 = 0.5(Fo-2F1+F2)$
- $a_1 = 0.5(F2-Fo)$
- If $a_2 > 0$ Then
- Begin
- $s_{min} = -\dfrac{a_1}{2a_2}$
- If $s_{min} > +1$ Then $s_{min} = +1$
- If $s_{min} < -1$ Then $s_{min} = -1$
- $t_{min} = 0.5h(s_{min}+1)$
- $q_1 = q_1 + t_{min} r_1$
- $q_2 = q_2 + t_{min} r_2$
- End Else

(cont.)

- End Else
- Begin
- Search minimum of ten samples as usual
- End

$p_1 = q_1$
$p_2 = q_2$
$k = k+1$
If k>1000 Then exit
End loop

5.1 PostScript Code

```
%!PS-Adobe-3.0  EPSF-3.0
%%BoundingBox:  0 0 626 952
%%Creator:      Gernot Hoffmann
%%Title:        MeasureGam A or B
%%CreationDate: June 21 2008

/mm {2.834646 mul} def

/n  100 def
/dx   1 n  div def

/sx  80 mm def
/sy  80 mm def

/x0  20 mm def
/y0  20 mm def

/lw  0.4 mm def
 0 setgray
 newpath lw lw moveto 220 mm 0 rlineto 0 335 mm rlineto -220 mm 0 rlineto closepath stroke

/Patterns
{
/x1 0 def
/y1 0 def
/dx1 20 mm def
/dy1 25 mm def
0 1 8
 { /i exch def
   Tabx i get dup dup setrgbcolor
   newpath x1 y1 moveto dx1 0 rlineto 0 dx1 rlineto dx1 neg 0 rlineto closepath fill
   /x1 x1 dx1 add def
   } for
/x1 0 def
/y1 y1 dy1 add def
/i 0 def
0 1 8
 { /i exch def
   0 0 Tabx i get setrgbcolor
   newpath x1 y1 moveto dx1 0 rlineto 0 dx1 rlineto dx1 neg 0 rlineto closepath fill
   /x1 x1 dx1 add def
   } for
/i 0 def
/x1 0 def
/y1 y1 dy1 add def
0 1 8
 { /i exch def
   0 Tabx i get 0 setrgbcolor
   newpath x1 y1 moveto dx1 0 rlineto 0 dx1 rlineto dx1 neg 0 rlineto closepath fill
   /x1 x1 dx1 add def
   } for
/i 0 def
/x1 0 def
/y1 y1 dy1 add def
0 1 8
 { /i exch def
   Tabx i get 0 0 setrgbcolor
   newpath x1 y1 moveto dx1 0 rlineto 0 dx1 rlineto dx1 neg 0 rlineto closepath fill
   /x1 x1 dx1 add def
   } for
/fh 10 def
/Helvetica findfont fh scalefont setfont
/buf 20 string def
 0 setgray
/y1 -3.5 mm def
/x1  9 mm def
0 1 8
{/i exch def
  x1 y1 moveto i buf cvs show
 /x1 x1 dx1 add def
} for
} def
```

5.2 PostScript Code

```
/Grid
{ 0.2 mm sx div setlinewidth
  newpath
  0.5 setgray
 /mg 8 def
 /dg 1 mg div def
 /y dg def
 newpath
 1 1 mg 1 sub
 { 0 y moveto 1 y lineto /y y dg add def
 } for
 stroke
 /x dg def
 newpath
 1 1 mg 1 sub
 { x 0 moveto x 1 lineto /x x dg add def
 } for
 stroke
} def

/Axes
{ 0.4 mm sx div setlinewidth
  2 setlinecap
  newpath 0 1 moveto 1 1 lineto 1 0 moveto 1 1 lineto 0.5 setgray stroke
  newpath 0 0 moveto 1 0 lineto 0 0 moveto 0 1 lineto 0.0 setgray stroke
 } def

/Dots
{/d1 0.02 def
 /d2 d1 -0.5 mul def
 /i 0 def
0.5 setgray
0 1 8
{/i exch def
  /x Tabx i get def /y Taby i get def
   newpath x y moveto
   d2 d2 rmoveto d1 0 rlineto 0 d1 rlineto d1 neg 0 rlineto closepath stroke
} for
} def

/Text
{0 setgray
/fh 12 sx div def
/Helvetica findfont fh scalefont setfont
/buf 20 string def
/x -0.04 def
/y 0 fh sub def
 x y moveto (0.0) show
/x 0.48 def
 x y moveto (0.5) show
/x 0.98 def
 x y moveto (1.0) show
/x 0.74 def
 x y moveto (x) show
/x -0.10 def
/y -0.01 def
 x y moveto (0.0) show
/y 0.49 def
 x y moveto (0.5) show
/y 0.99 def
 x y moveto (1.0) show
/y 0.74 def
/x -0.05 def
 x y moveto +90 rotate (Y/Ymax) show -90 rotate
0.8 setgray
/x 0.03 def
/y 0.765 def
/d1 0.57 def
/d2 0.22 def
newpath x y moveto d1 0 rlineto 0 d2 rlineto d1 neg 0 rlineto closepath fill
0 setgray
/x 0.05 def
/y 0.924 def
 x y moveto txt show  x 0.37 add y moveto Gam 1000 mul round 1000 div buf cvs show
/y 0.85 def
 x y moveto (Offset) show  x 0.37 add y moveto Ofs 10000 mul round 10000 div buf cvs show
/y 0.78 def
 x y moveto (Ymax) show  x ymax 100 lt{0.37 add}{0.33 add} ifelse y moveto ymax buf cvs show
```

5.3 PostScript Code

```
% Loop control message / use true or false
  false
  {/y 0.66 def
   emod 1 eq {/txe (Exit Func) def} if
   emod 2 eq {/txe (Exit Grad) def} if
   emod 3 eq {/txe (Exit Loop) def} if
   x y moveto txe show
  /y y fh sub def
   x y moveto (loops ) show k buf cvs show
  /y y fh sub def
   x y moveto (Error per point ) show Fo 8 div sqrt buf cvs show} if
 } def

/GamCalc
{ % Steepest descent with one-dim search
% yi = Yi/Ymax
% Minimize F=Sum(1..8)[p1+(1-p1)*xi^p2-yi]^2
% Fo=F(p1,p2)
% g1=(F(p1+dp1,p2)-Fo)/dp
% g2=(F(p1,p2+dp2)-Fo)/dp
% dn=g1*g1+g2*g2
% p1=p1-sig*Fo*g1/dn
% p2=p2-sig*Fo*g2/dn
% Function
/Fun
 {% table entry No.0 ignored
  /Fu 0 def
   1 1 8
  {/i exch def
   /xi Tabx i get def
   /yi Taby i get def
 % Use either Type A or Type B
 % Type A
 % /Fu Fu xi p2 exp yi sub dup mul add def
 % Type B, here active
   /Fu Fu p1 1 p1 sub xi p2 exp mul add yi sub dup mul add def
  } for
 } def
% Normalize
/ymax Taby 8 get def
 0 1 8
 {/i exch def
   Taby i Taby i get ymax div put
 } for
% Steepest descent
 /p1 0.0 def /p2 1.0 def /q1 p1 def /q2 p2 def
 /Fo  999  def
 /dp  1e-4 def
 /eps 1e-8 def
 /sig 0.07 def
 /k 1 def
 { Fun
   Fu Fo sub abs eps lt {/emod 1 def exit} if
  /Fo Fu def
  /p1 p1 dp add def Fun /g1 Fu Fo sub dp div def /p1 q1 def
  /p2 p2 dp add def Fun /g2 Fu Fo sub dp div def /p2 q2 def
  /dn g1 dup mul g2 dup mul add def
   dn eps lt {/emod 2 def exit} if
  /fc Fo sig mul dn div neg def
  /r1 fc g1 mul def
  /r2 fc g2 mul def
  /Fm Fo def % one-dim search
  /q1 p1 def
  /q2 p2 def
   1 1 10
   { pop
    /p1 p1 r1 add def /p2 p2 r2 add def
     Fun
     Fu Fm lt {/Fm Fu def /q1 p1 def /q2 p2 def} if
   } for
  /p1 q1 def
  /p2 q2 def
  /k k 1 add def
   k 1000 eq {/emod 3 def exit} if
   } loop
  /Ofs p1 def
  /Gam p2 def
} def
```

5.4 PostScript Code

```
/TestTab
{% Fill Tab for test
 % 2.2=arbitrary gamma
 % 130=arbitrary gain
 % 0.5=arbitrary offset
  0 1 8
  {/i exch def
    Taby i Tabx i get 2.2 exp 130 mul 0.5 add put
  }for
} def

/TabRGB [   0
           32
           64
           96
          128
          160
          192
          224
          255 ] def

/Tabx [ 0.000000
        0.125490
        0.250980
        0.376471
        0.501961
        0.627451
        0.752941
        0.878431
        1.000000 ] def

/GamRed
{/Taby [ 0.90
         0.99
         1.85
         3.60
         6.32
         9.73
        14.07
        19.34
        25.34 ] def
 %TestTab
  GamCalc
 /GamR Gam def
 /OfsR Ofs def
} def

/GamGrn
{/Taby [ 0.68
         1.16
         3.54
         8.02
        14.83
        23.57
        35.74
        46.78
        64.22 ] def
 %TestTab
  GamCalc
 /GamG Gam def
 /OfsG Ofs def
} def

/GamBlu
{/Taby [ 0.69
         0.66
         0.90
         1.37
         2.19
         3.37
         4.74
         6.44
         8.48 ] def
 %TestTab
  GamCalc
 /GamB Gam def
 /OfsB Ofs def
} def
```

5.5 PostScript Code

```
/GamWht
{/Taby [ 0.80
          1.67
          4.91
         11.27
         21.31
         34.19
         52.36
         72.66
         97.76 ] def
 %TestTab
  GamCalc
 /GamW Gam def
 /OfsW Ofs def
} def

/FuncS
{/y x 2.2 exp def
} def

/FuncR
{/y x GamR exp 1 OfsR sub mul OfsR add def
} def

/FuncG
{/y x GamG exp 1 OfsG sub mul OfsG add def
} def

/FuncB
{/y x GamB exp 1 OfsB sub mul OfsB add def
} def

/FuncW
{/y x GamW exp 1 OfsW sub mul OfsW add def
} def

false setstrokeadjust

gsave
x0 y0 200 mm add translate
Patterns
grestore

gsave
x0 y0 100 mm add translate
sx sx scale
Grid
Axes
lw sx div setlinewidth
0 setlinecap

/txt (Red Gamma) def
GamRed
1 0 0 setrgbcolor
/x 0 def
newpath
FuncR x y moveto
1 1 n { pop
       /x x dx add def
        FuncR x y lineto
      } for
stroke
Dots
Text
grestore

gsave
x0 100 mm add y0 100 mm add translate
sx sx scale
Grid
Axes
lw sx div setlinewidth
0 setlinecap
```

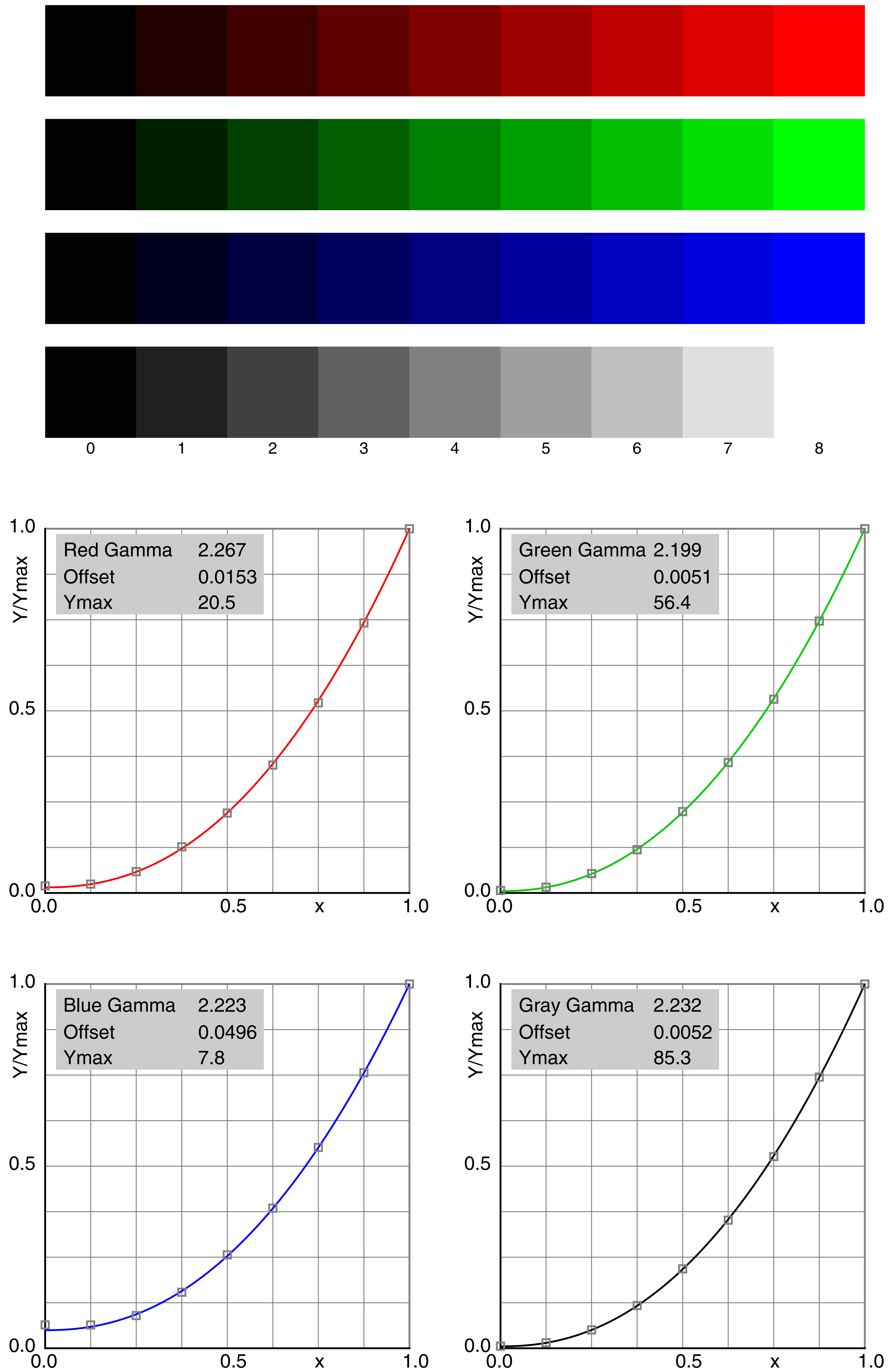
5.6 PostScript Code

```
/txt (Green Gamma) def
GamGrn
Dots
0 0.8 0 setrgbcolor
/x 0 def
newpath
FuncG x y moveto
1 1 n { pop
       /x x dx add def
        FuncG x y lineto
      } for
stroke
Dots
Text
grestore
gsave
x0 y0 translate
sx sx scale
Grid
Axes
lw sx div setlinewidth
0 setlinecap

/txt (Blue Gamma) def
GamBlu
0 0 1 setrgbcolor
/x 0 def
newpath
FuncB x y moveto
1 1 n { pop
       /x x dx add def
        FuncB x y lineto
      } for
stroke
Dots
Text
grestore
gsave
x0 100 mm add y0 translate
sx sx scale
Grid
Axes
lw sx div setlinewidth
0 setlinecap

/txt (Gray Gamma) def
GamWht
0 setgray
/x 0 def
newpath
FuncW x y moveto
1 1 n { pop
       /x x dx add def
        FuncW x y lineto
      } for
stroke
Dots
Text

showpage
```

6.1 Examples / LaCie 19 / GMB Eye-One Pro / Type B (Standard)

Profile data: RedG=2.238, GreenG=2.176, BlueG=2.176. No offsets available.

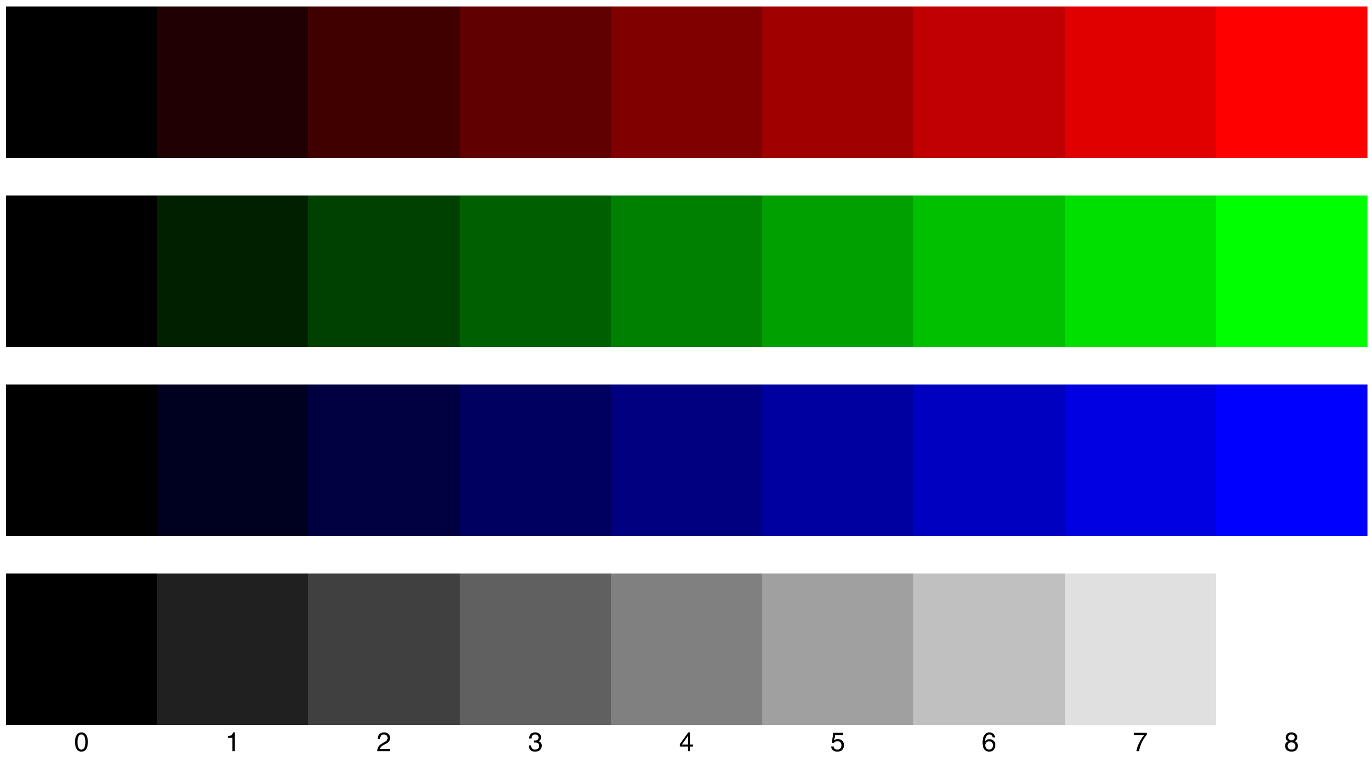
Reference patches below
Measured on patches 8cm x 8cm

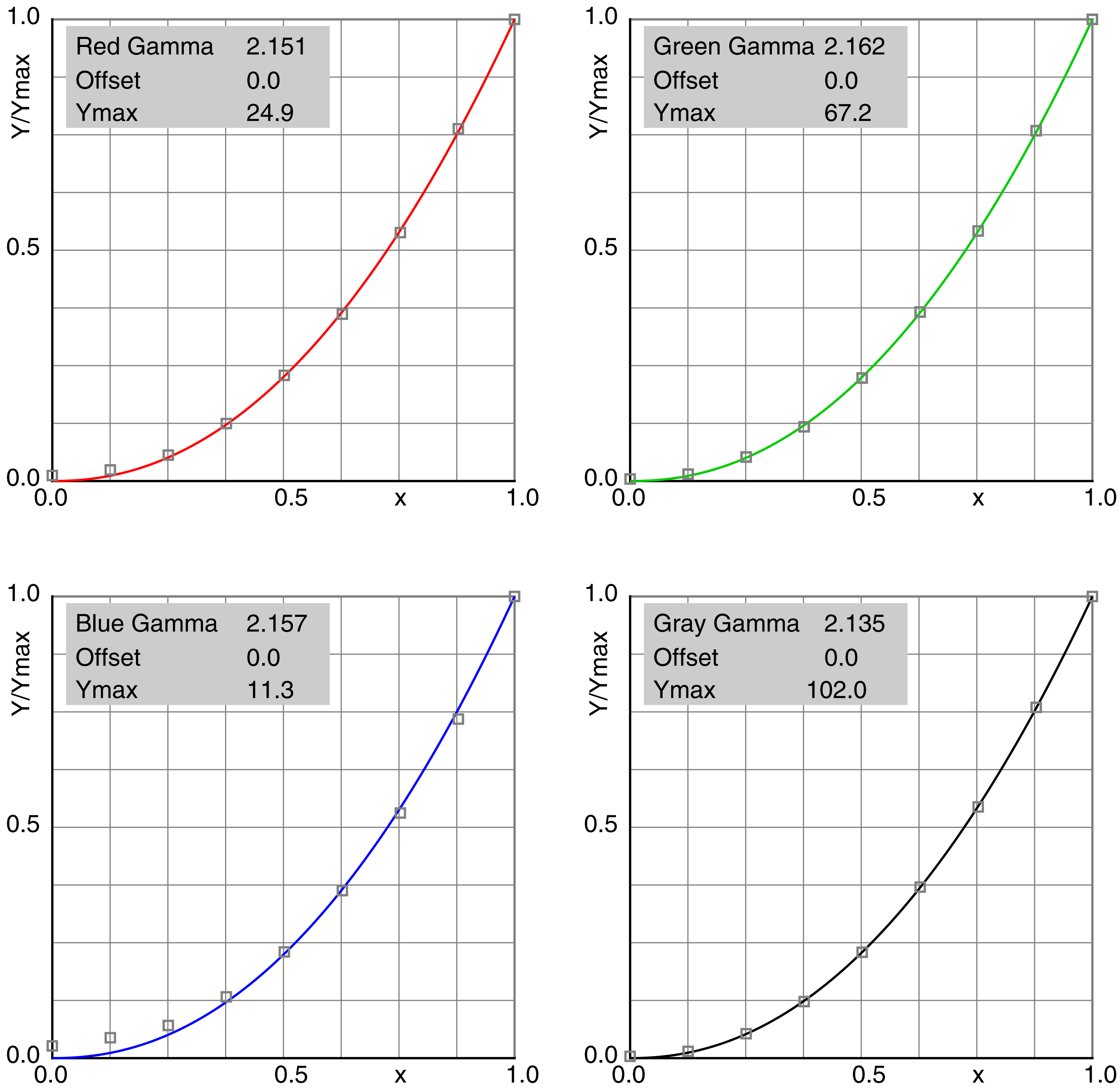


6.2 Examples / Eizo CG19 / GMB Eye-One Pro / Type A (Power)

Profile data: RedG=2.191, GreenG=2.191, BlueG=2.273. No offsets available. TFT-Monitor. The pure power function Type A might be useful for LCD / TFT monitors. Code was changed for both options Type A and Type B.

Reference patches below
Measured on patches 8cm x 8cm in the screen center

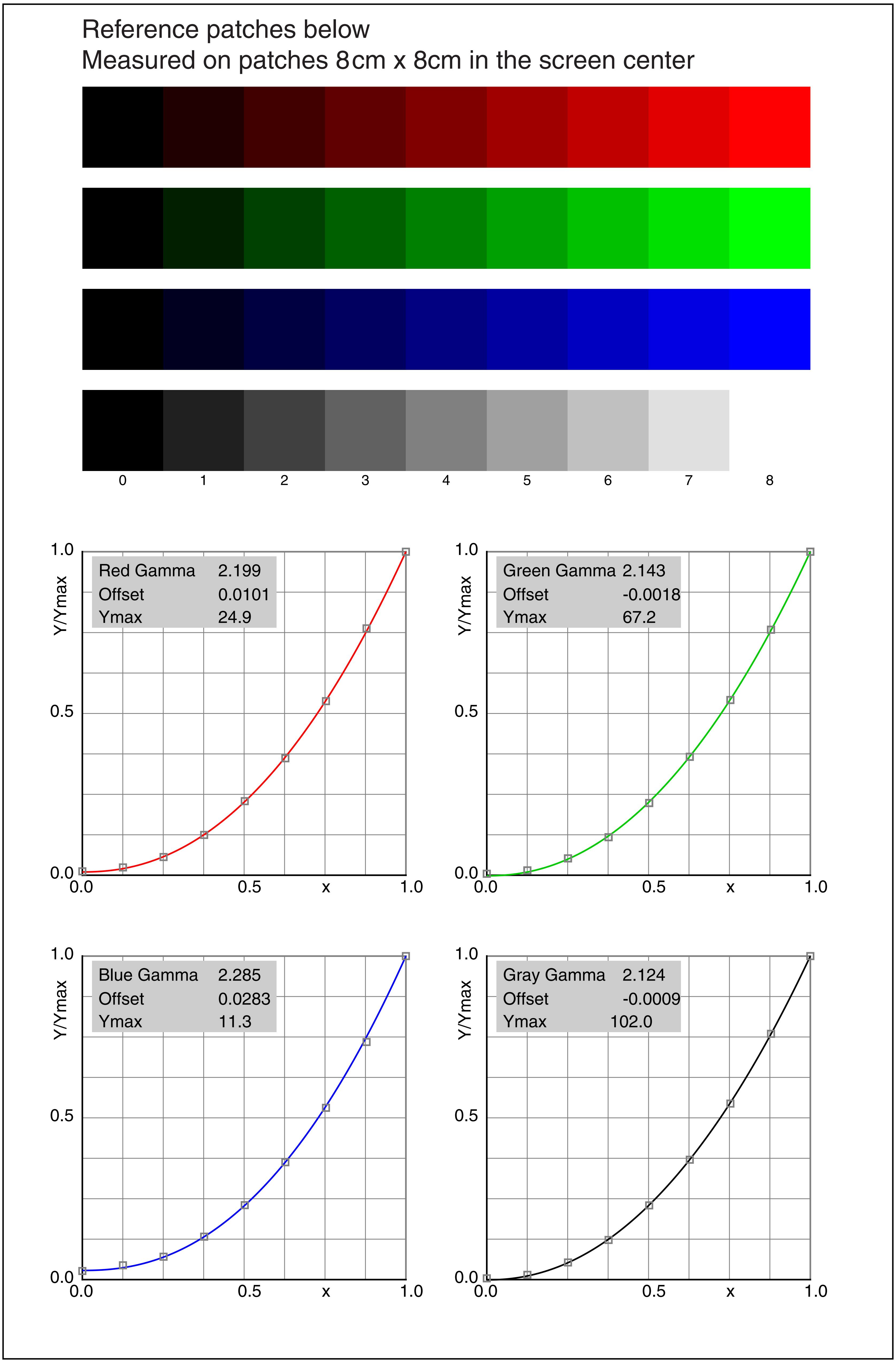





```
/GamRed
{/Taby [ 0.3
         0.6
         1.4
         3.1
         5.7
         9.0
        13.4
        19.0
        24.9 ]
  def
  GamCalc
 /GamR Gam def
 /OfsR Ofs def
} def

/GamGrn
{/Taby [ 0.3
         1.0
         3.5
         7.9
        15.0
        24.6
        36.4
        51.0
        67.2 ]
  def
  GamCalc
 /GamG Gam def
 /OfsG Ofs def
} def

/GamBlu
{/Taby [ 0.3
         0.5
         0.8
         1.5
         2.6
         4.1
         6.0
         8.3
        11.3 ]
  def
  GamCalc
 /GamB Gam def
 /OfsB Ofs def
} def

/GamWht
{/Taby [ 0.4
         1.5
         5.4
        12.5
        23.4
        37.8
        55.5
        77.5
       102.0 ]
  def
  GamCalc
 /GamW Gam def
 /OfsW Ofs def
} def
```

6.3 Examples / Eizo CG19 / GMB Eye-One Pro / Type B (Standard)

Profile data: RedG=2.191, GreenG=2.191, BlueG=2.273. No offsets available.TFT-Monitor. The pure power function Type A might be useful for LCD/TFT monitors. Code was changed for both options Type A and Type B.




```
/GamRed
{/Taby [ 0.3
         0.6
         1.4
         3.1
         5.7
         9.0
        13.4
        19.0
        24.9 ]
  def
  GamCalc
 /GamR Gam def
 /OfsR Ofs def
} def

/GamGrn
{/Taby [ 0.3
         1.0
         3.5
         7.9
        15.0
        24.6
        36.4
        51.0
        67.2 ]
  def
  GamCalc
 /GamG Gam def
 /OfsG Ofs def
} def

/GamBlu
{/Taby [ 0.3
         0.5
         0.8
         1.5
         2.6
         4.1
         6.0
         8.3
        11.3 ]
  def
  GamCalc
 /GamB Gam def
 /OfsB Ofs def
} def

/GamWht
{/Taby [ 0.4
         1.5
         5.4
        12.5
        23.4
        37.8
        55.5
        77.5
       102.0 ]
  def
  GamCalc
 /GamW Gam def
 /OfsW Ofs def
} def
```

6.4 Examples / GMB Service Programs

This screenshot shows the content of the LUTs and three helpful programs by GMB ProfileMaker for the example monitor LaCie19.

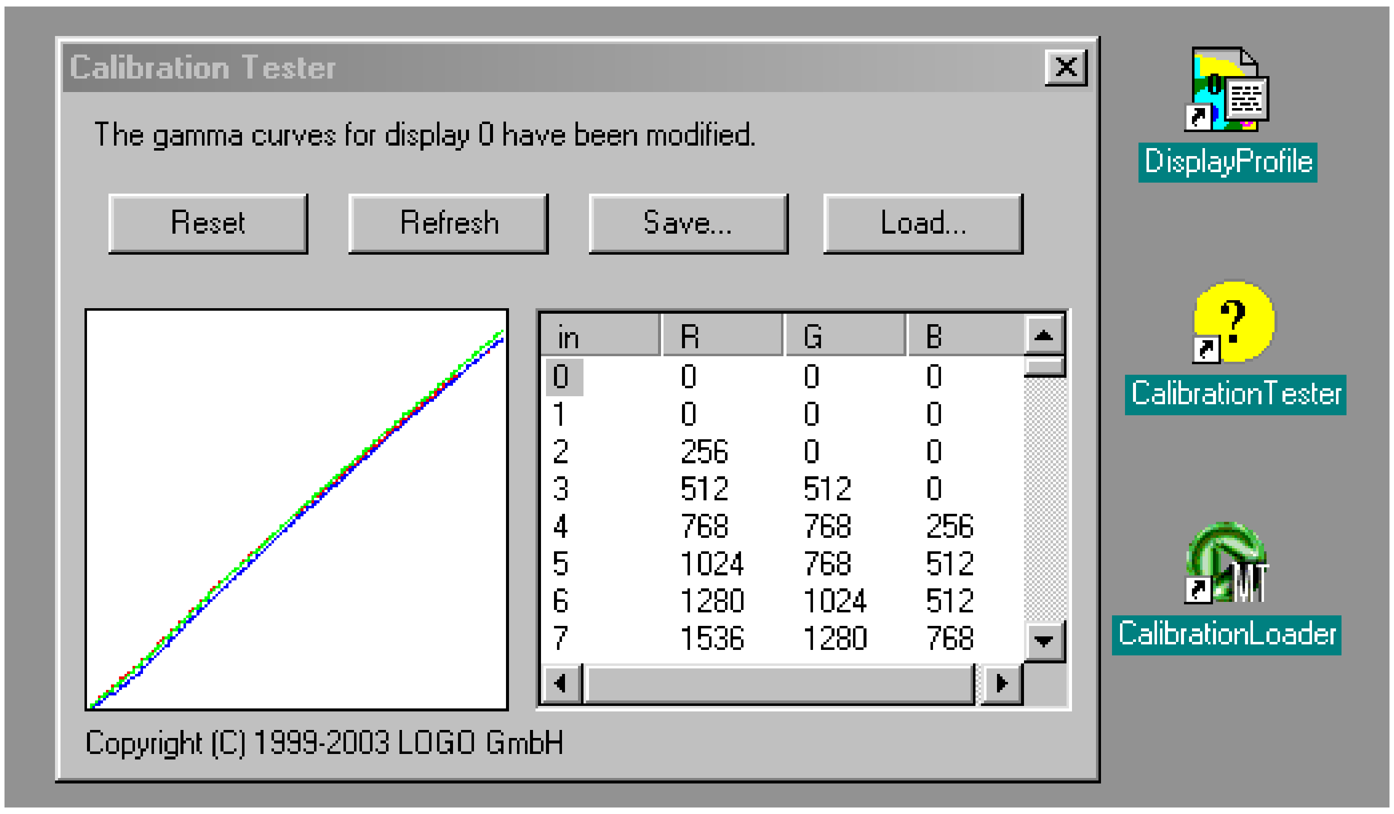
DisplayProfile and CalibrationTester from the CD, not automatically copied to the harddisk.

CalibrationLoader
Loads the LUTs and assigns the profile as Windows system profile.

CalibrationTester
Shows the content of the LUTs.
Input range 0..255, output range 0..65535.

DisplayProfile
Here we can choose a profile.
It is indicated whether the profile contains data for LUTs.

The intended luminance for this example was less than 100%,
(dimmed room), therefore the LUTs apply a global attenuation.



7. Measuring the Generic TRC

The report [6] describes a very accurate method how to derive the generic Gamma from measured values for an uncalibrated monitor with input and output offsets.

This should not be confused with the identification of the *calibrated* monitor and with the interpretation of *effective* gamma.

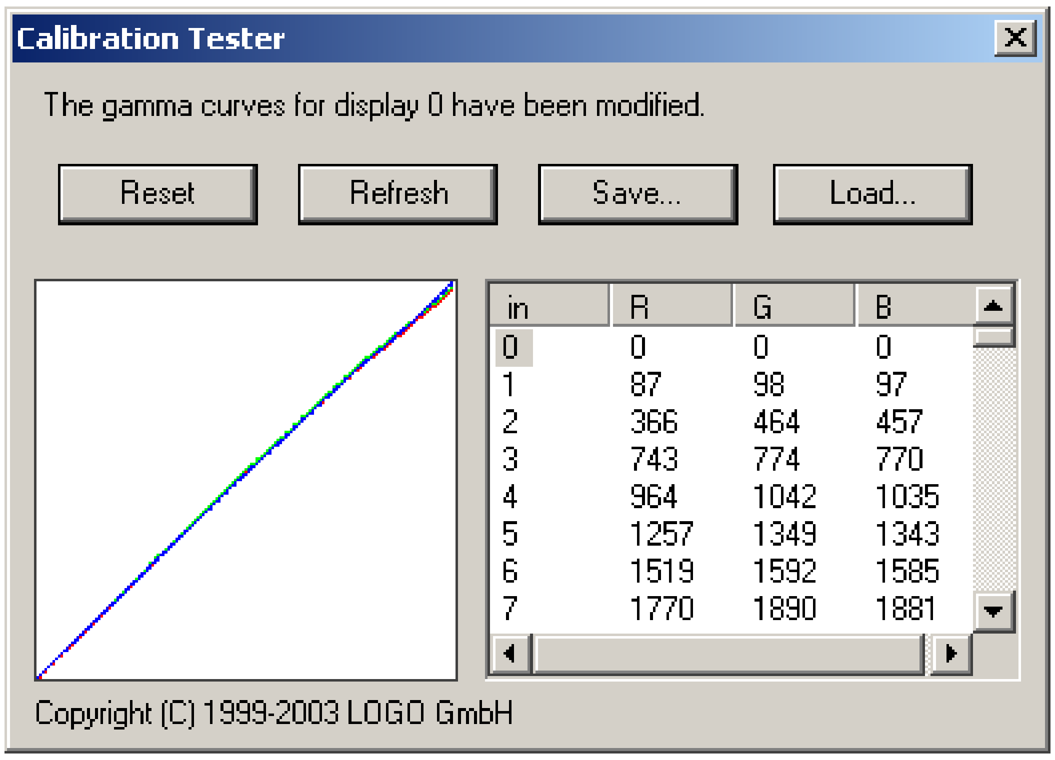
8.1 Comparison Gamma=1.8 versus Gamma=2.2

In this chapter it will be proved that Photoshop takes the monitor profile into account. The same image, working space sRGB, will look on a Gamma=1.8 monitor and a Gamma=2.2 monitor almost alike. The menue colors are not color managed, therefore the Gamma=1.8 monitor looks a little brighter.

The monitor Eizo CG19 was calibrated twice:

a) for 6500K and Gamma=2.2
b) for 6500K and Gamma=1.8

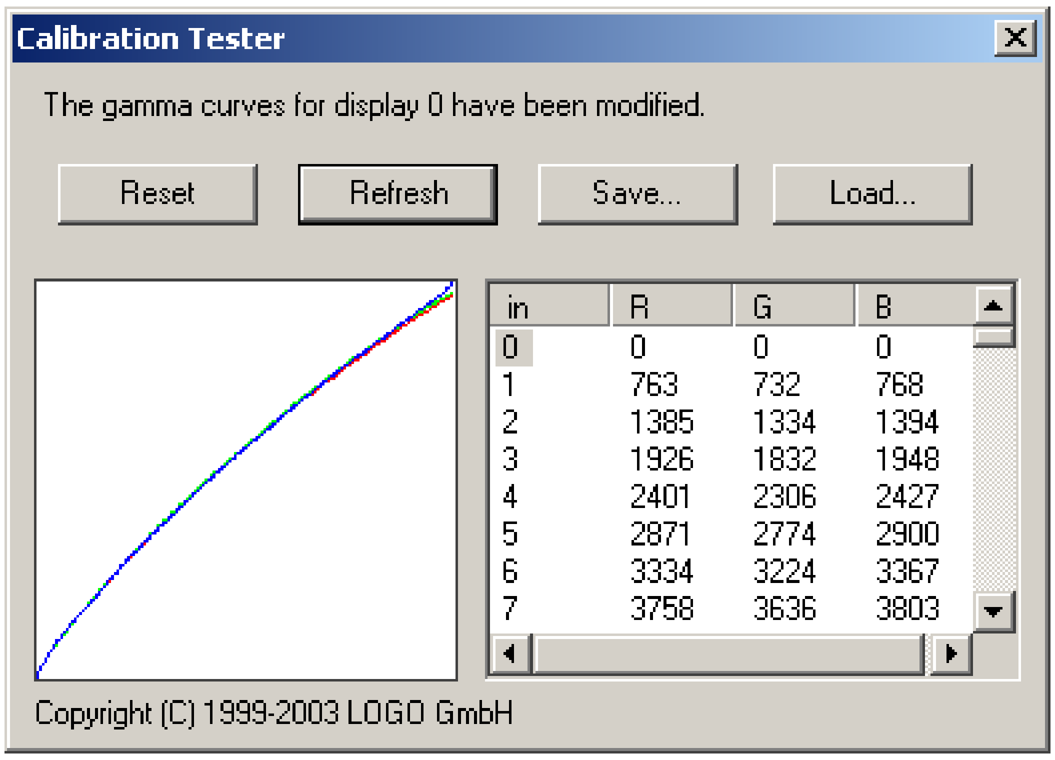
For the first case the monitor was adjusted by monitor menues so, that only small corrections by graphics card LUTs were necessary:



in	R	G	B
0	0	0	0
1	87	98	97
2	366	464	457
3	743	774	770
4	964	1042	1035
5	1257	1349	1343
6	1519	1592	1585
7	1770	1890	1881

LUT content for Gamma=2.2

For the second case the monitor settings were left unchanged. The correction happens by graphics card LUTs:

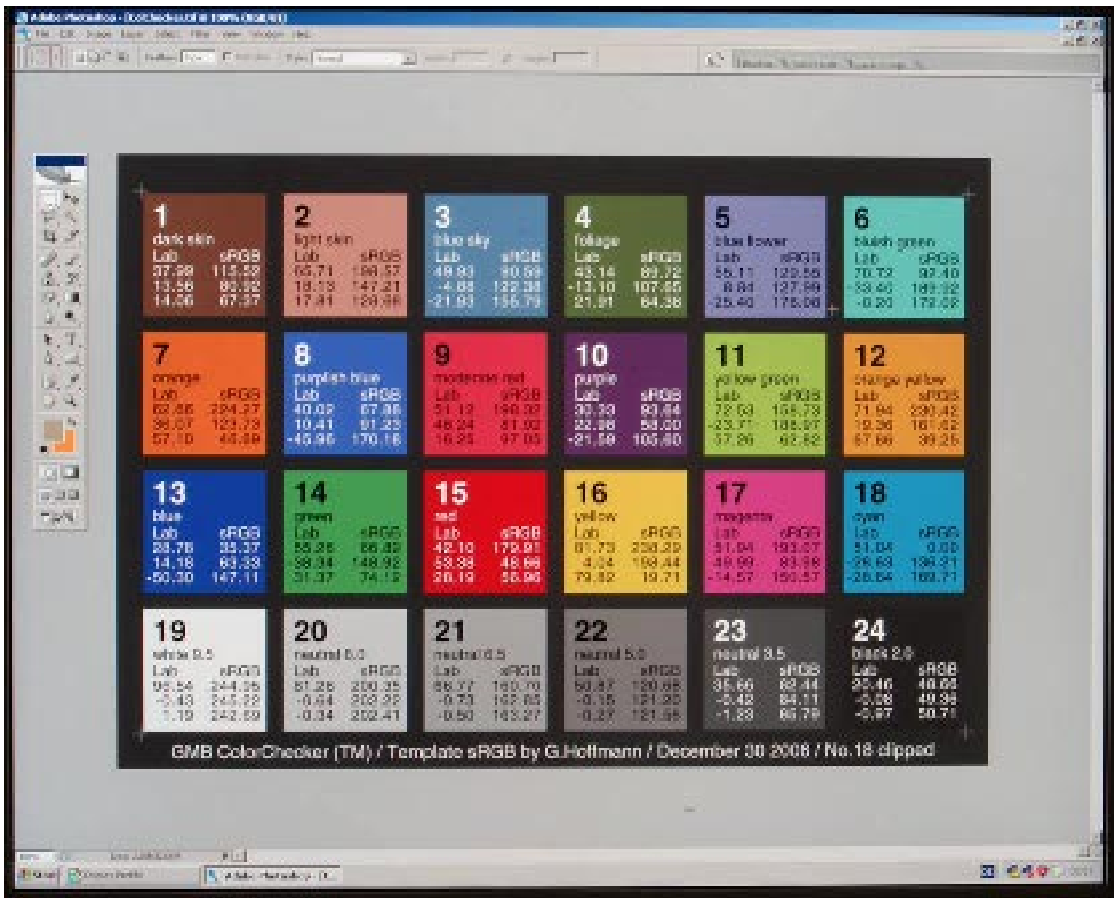


in	R	G	B
0	0	0	0
1	763	732	768
2	1385	1334	1394
3	1926	1832	1948
4	2401	2306	2427
5	2871	2774	2900
6	3334	3224	3367
7	3758	3636	3803

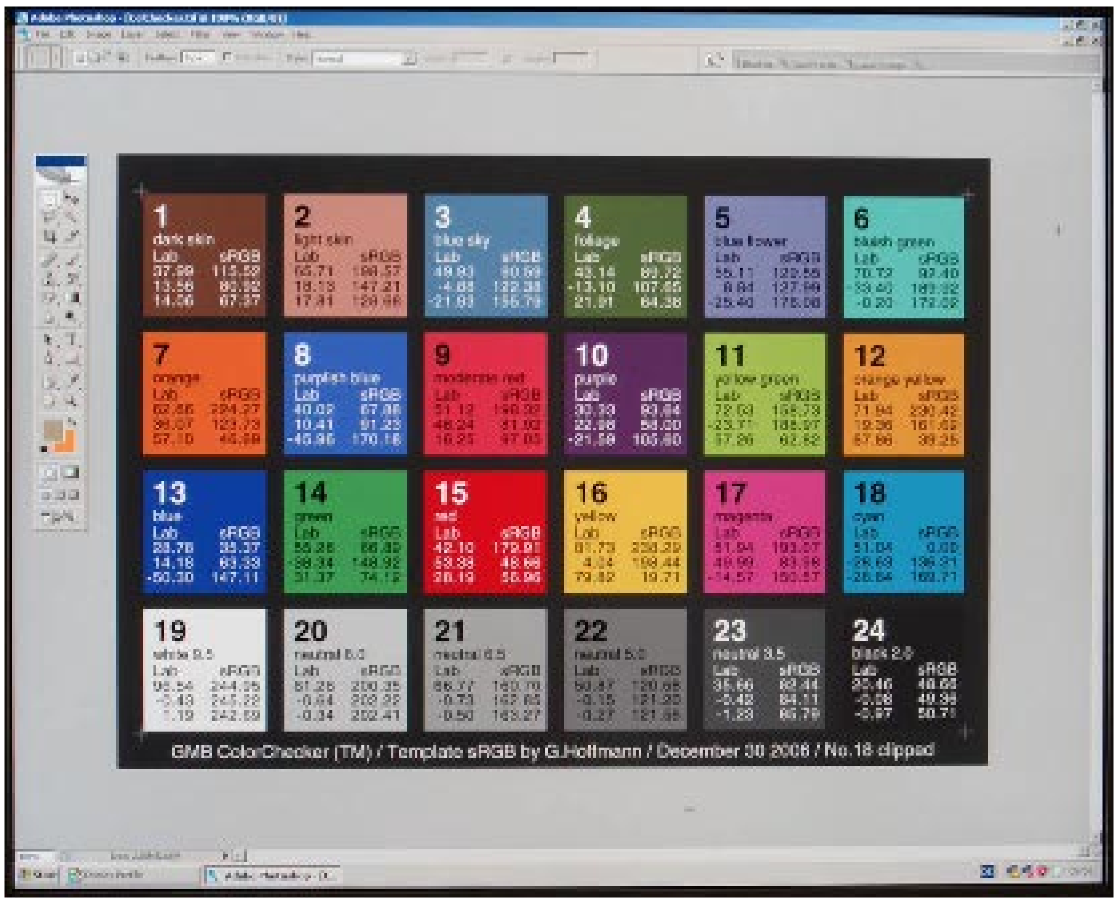
LUT content for Gamma=1.8

8.2 Comparison Gamma=1.8 versus Gamma=2.2

A digital version of the GMB ColorChecker was loaded in Photoshop in both modes.
Changing the monitor profile requires a new start of the program.
Images were taken by an uncalibrated digital camera. One has to avoid aliasing, which can cause low frequency artifacts by interfering monitor pixels and camera pixels.
Both images were saved in sRGB and a little sharpened and cropped for this doc.



Photo, monitor profile Gamma=2.2



Photo, monitor profile Gamma=1.8

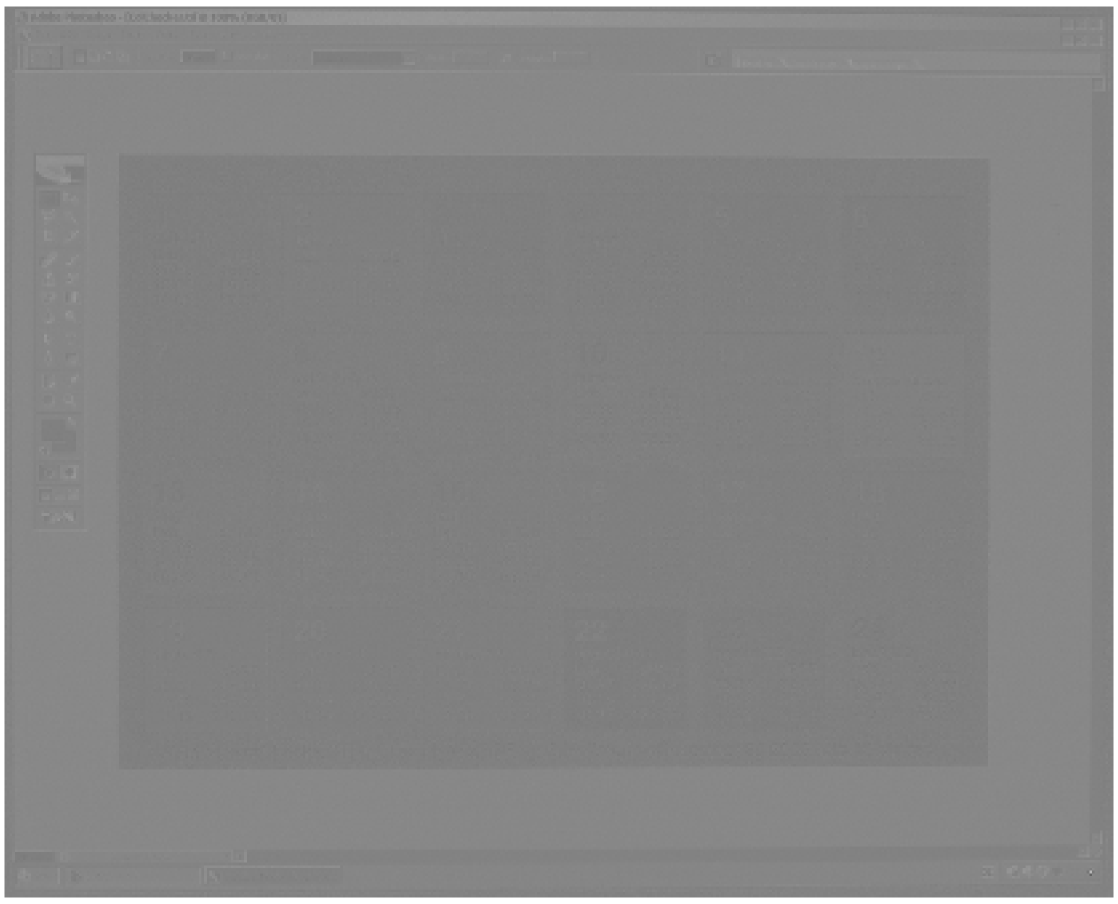


Image calculations: difference of grays for image 2.2 (source 1) and image 1.8 (source 2)
Offset=128
Scalefactor=1

9. References

[1] Henry R.Kang
Color Technology for Electronic Imaging Devices
SPIE Optical Engineering Press
Bellingham, Washington USA, 1997

[2] Roy S.Berns + Naoya Katoh
Methods for characterizing displays
in Phil Green + Lindsay MacDonald (Editors)
Color Engineering
John Wiley & Sons, 2003

[3] Gernot Hoffmann
The Gamma Question
http://docs-hoffmann.de/gamquest18102001.pdf

[4] Gernot Hoffmann
Hardware Monitor Calibration
http://docs-hoffmann.de/caltutor270900.pdf

[5] Gernot Hoffmann
PostScript Code
http://docs-hoffmann.de/measgamma.txt
Rename as *.eps

[6] Alan Roberts
Measurement of display transfer characteristic (gamma, γ)
EBU Technical Review Autumn 1993
http://www.ebu.ch/en/technical/trev/trev_257-tutorial.pdf

This doc:
http://docs-hoffmann.de/measgamma10022004.pdf

Gernot Hoffmann
October 10 / 2004 — February 09 / 2013
Website
Load browser / Click here