



A LON publication



Colorimetry v1.24 User Manual

Written by J.Romaya W.D.C.N.

23rd October 2002

Terms and conditions

Cogent Graphics is provided to you free of charge under the following conditions:-

1) Acknowledgement

Acknowledgement from users helps us justify the time we are spending further developing and maintaining this free software. Therefore we request that, when you use Cogent Graphics for your experiments, you include the following statement in your publication:

"This experiment was realised using Cogent Graphics developed by John Romaya at the LON at the Wellcome Department of Imaging Neuroscience."

2) Copying

This package may be distributed freely as long as it is distributed in its original form. It may not be sold without permission.

3) Liability

The package is distributed as it is, without any warranty implied. No liability is accepted for any damage or loss resulting from the use of these routines.

Changes v1.17 to v1.24

There are no changes between v1.17 and v1.24.

Changes v1.16 to v1.17

The DspCalib function has been modified so that it works with the new v1.17 time functions which return a time in seconds as well as the pre-v1.17 functions which returned a time in microseconds.

Changes v1.08 to v1.16

There are no changes between v1.08 and v1.16. The version number has simply been incremented to keep in step with the other graphics libraries.

Changes v1.07 to v1.08

1/ Global variables have been tidied up and are cleared when the function exits.

Table of Contents

Introduction	A.1
CIEPlot	A.2
DspAlyze	A.5
DspCalib	A.9
ReadDCF	A.12
Spc2XYZ	A.12
SpcPlot	A.14
xyY2XYZ	A.16
XYZ2xyY	A.16

Introduction

This manual describes a number of colorimetric tools provided to supplement the Cogent Graphics package. These may be downloaded from the Cogent Graphics website. The tools provided are:-

DspCalib	Calibrate a display for colorimetry and save a display calibration file (dcf).
DspAlyze	Analyze a display calibration file (dcf).
ReadDCF	Read a display calibration file into a structure in the Matlab workspace.
SpcPlot	Plot a radiant spectrum.
CIEPlot	Plot a CIE diagram.
Spc2XYZ	Convert a radiant spectrum into CIE (1931) XYZ values.
XYZ2xyY	Convert CIE (1931) XYZ values into CIE (1931) xyY values.
xyY2XYZ	Convert CIE (1931) xyY values into CIE (1931) XYZ values.

A knowledge of colorimetry is assumed in this manual and the reader is referred for further information to:-

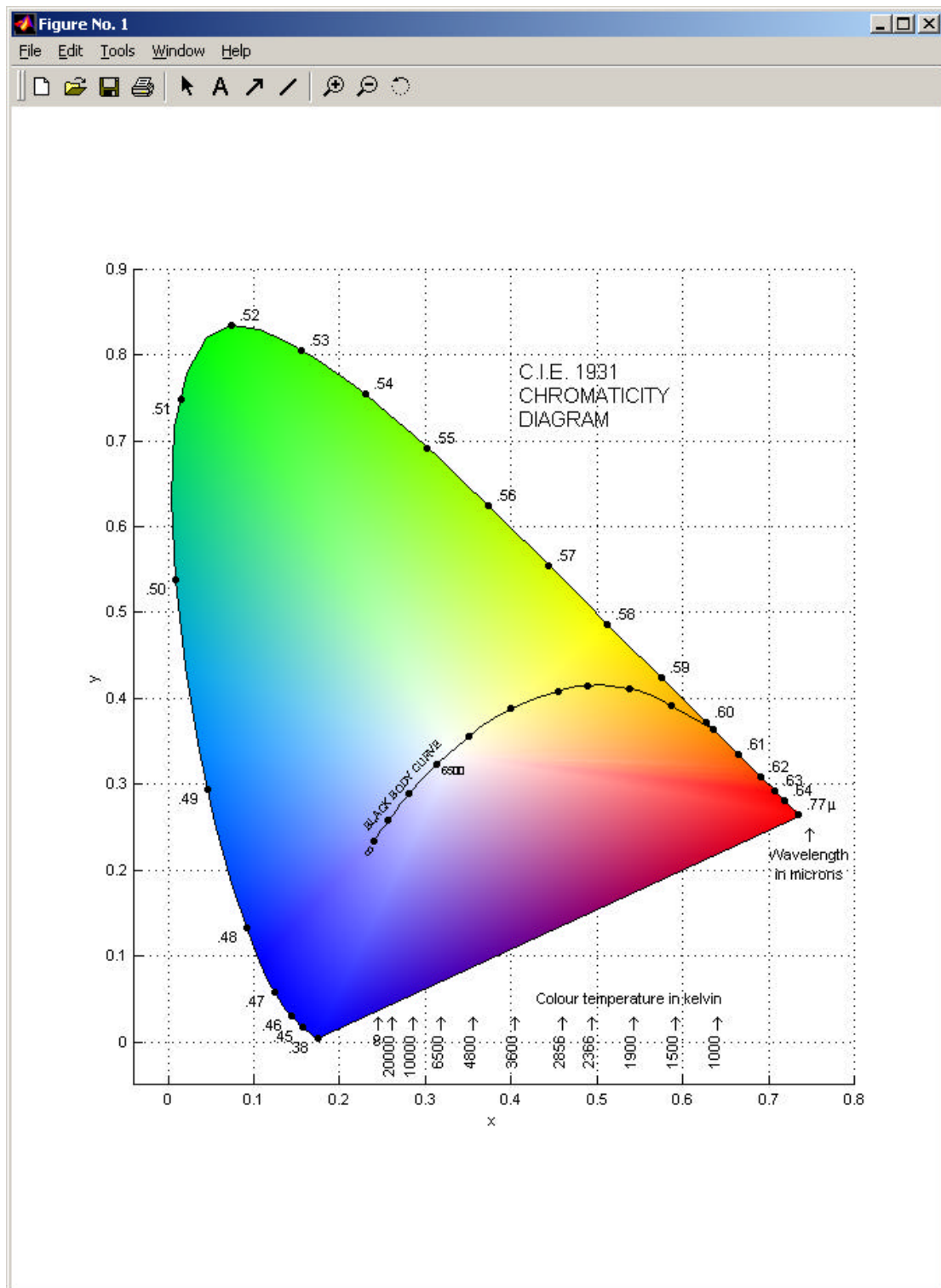
Color Science: Concepts and Methods, Quantitative Data and Formulae Second Edition
Günter Wyszecki & W.S.Stiles 1982 John Wiley & Sons

The display calibration file referred to in this manual is an ascii file and can be read as text but it should be edited with great caution as the precise format of the file is very strictly defined.

Two sample data files; 'spc.mat' and 'test.dcf.txt' can be downloaded from the Cogent Graphics website for use in this manual.

CIEPlot

When you type the command CIEPlot you get the figure below; the CIE 1931 Chromaticity diagram:-

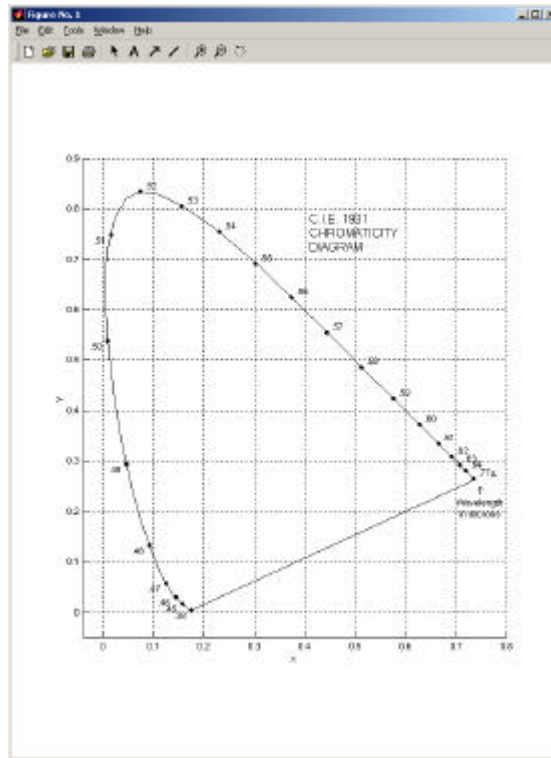


Initially the plot will appear rather small and in that case you should enlarge the figure window to whatever size you require and then type CIEPlot again.

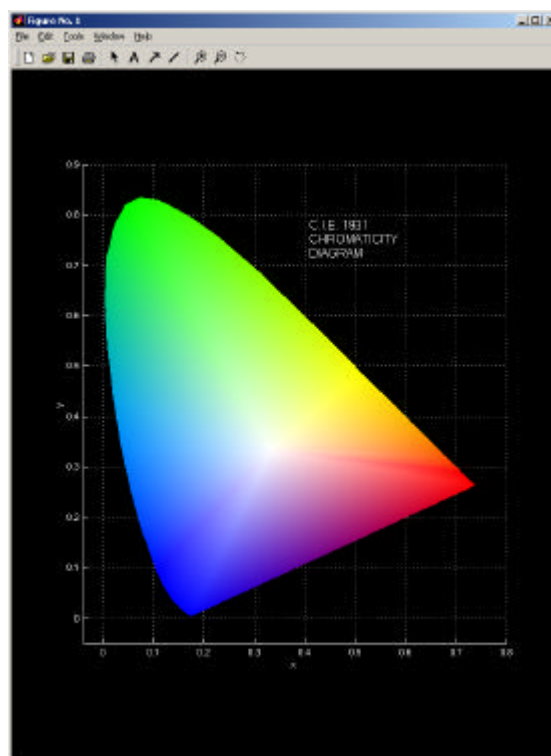
You can control what elements appear on the plot by passing a 'Flags' argument to the function. The Flags value is the arithmetic sum of the following components:-

- 1 Plot the figure as white on a black background
- 2 Plot the wavelength outline on the figure
- 4 Plot the central colour guide
- 8 Plot the blackbody radiation curve

For example, if you just want to plot the wavelength outline alone, use CIEPlot(2):-

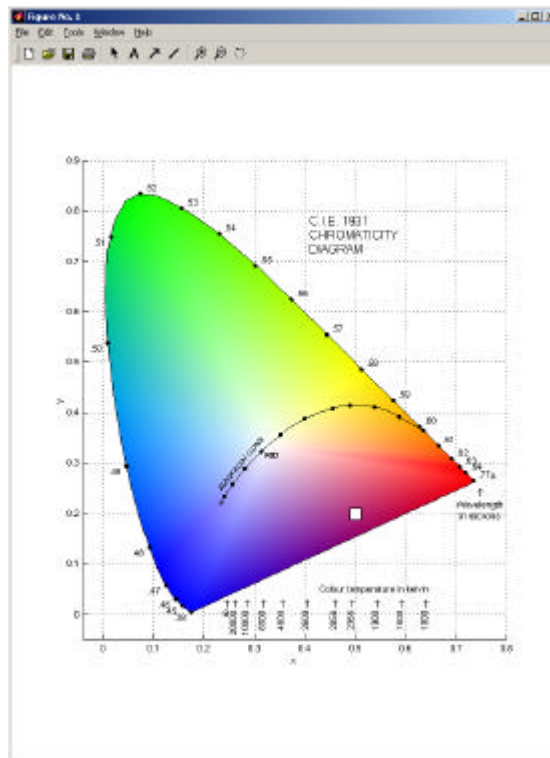


Or, to plot the central colour guide on a black background use (1 + 4); CIEPlot(5):-



Once you have your plot you can add points to it in the following way:-

```
EDU» CIEPlot
EDU» hold on
EDU» plot(.5,.2,'s','MarkerEdgeColor','k','MarkerFaceColor','w','MarkerSize',12)
EDU» hold off
```



The matlab **plot** command has been used to plot a point on the graph.

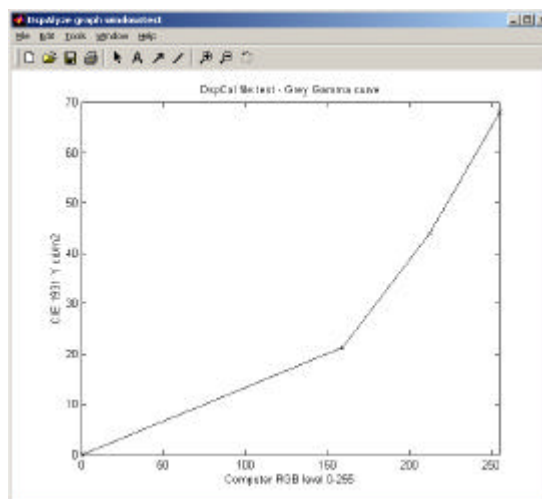
DspAlyze

This function is used to analyze the data in a display calibration file. A sample file can be downloaded from the website, named 'test.dcf.txt'.

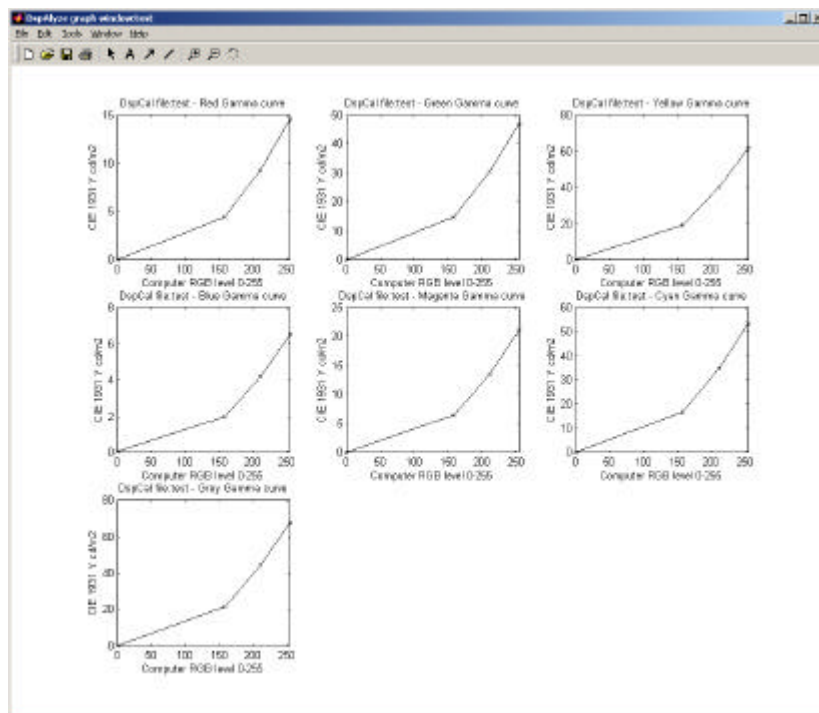
A number of analyses can be made including:-

Gamma curve	A plot of the monitor gamma curve for the colours red, green, yellow, blue, magenta, cyan and grey. Strongly curved plots are undesirable.
Phosphor independence	A plot of the independence of the phosphors for different colours. The independence curve for grey for example shows the ratio of the intensity of a given grey level to the sum of the individual red, green and blue intensities at the same level. Curves can be plotted for yellow, magenta, cyan and grey. A flat line close to unity is desirable.
CIE (1931) x & y	A plot of how particular colours change with varying intensity. A horizontal line is desirable.
CIE plot	A plot of the monitor red, green and blue phosphors on the CIE (1931) chromaticity diagram. A large triangle is desirable.
Notes	Ancillary information made during the calibration.

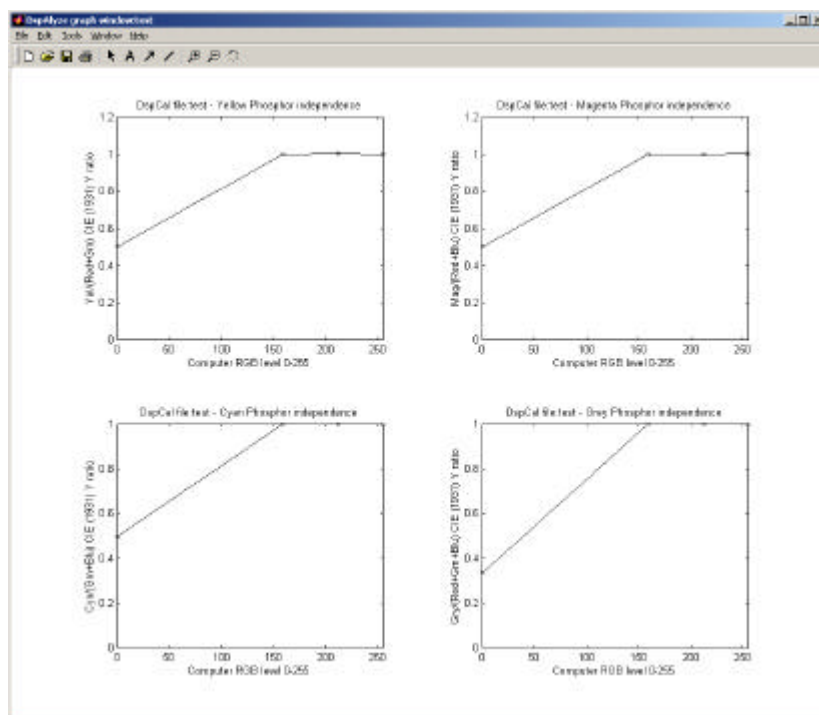
The initial plot shows the gamma curve for grey as well as a menu which allows you to select other plots:-



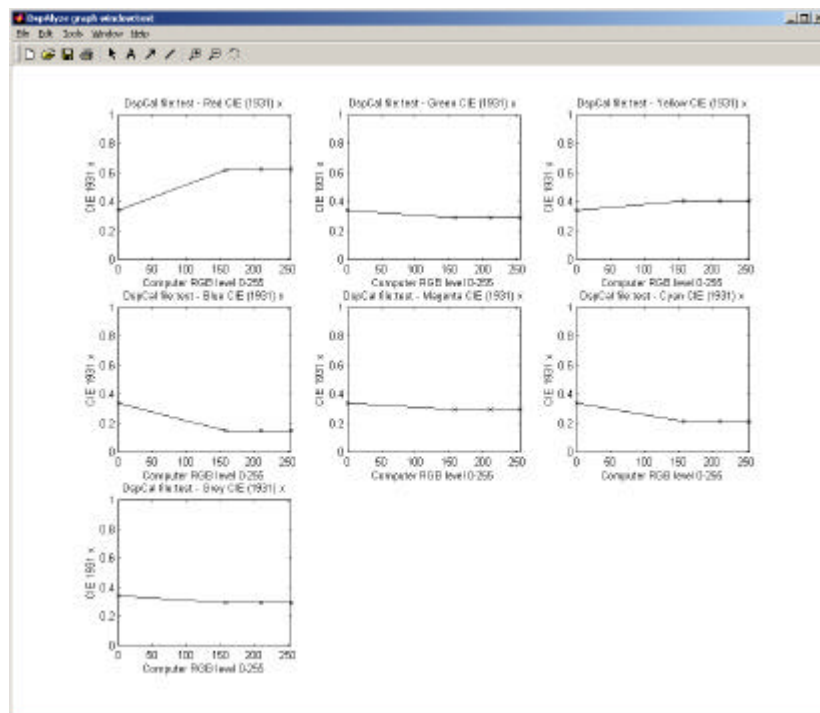
If you select 'All' a multiple plot of the gamma curves of all the colours is shown:-



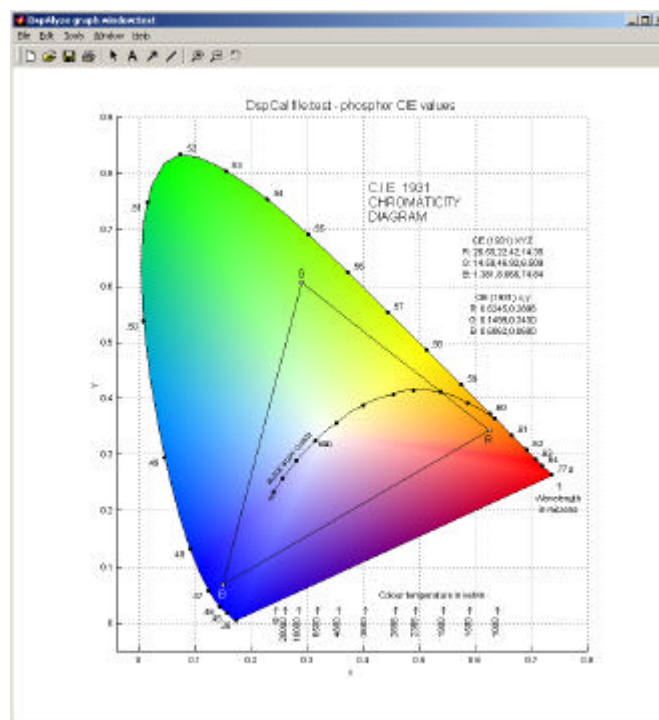
If you now select 'Phosphor Independence' you get a plot of the phosphor independence for all colours:-



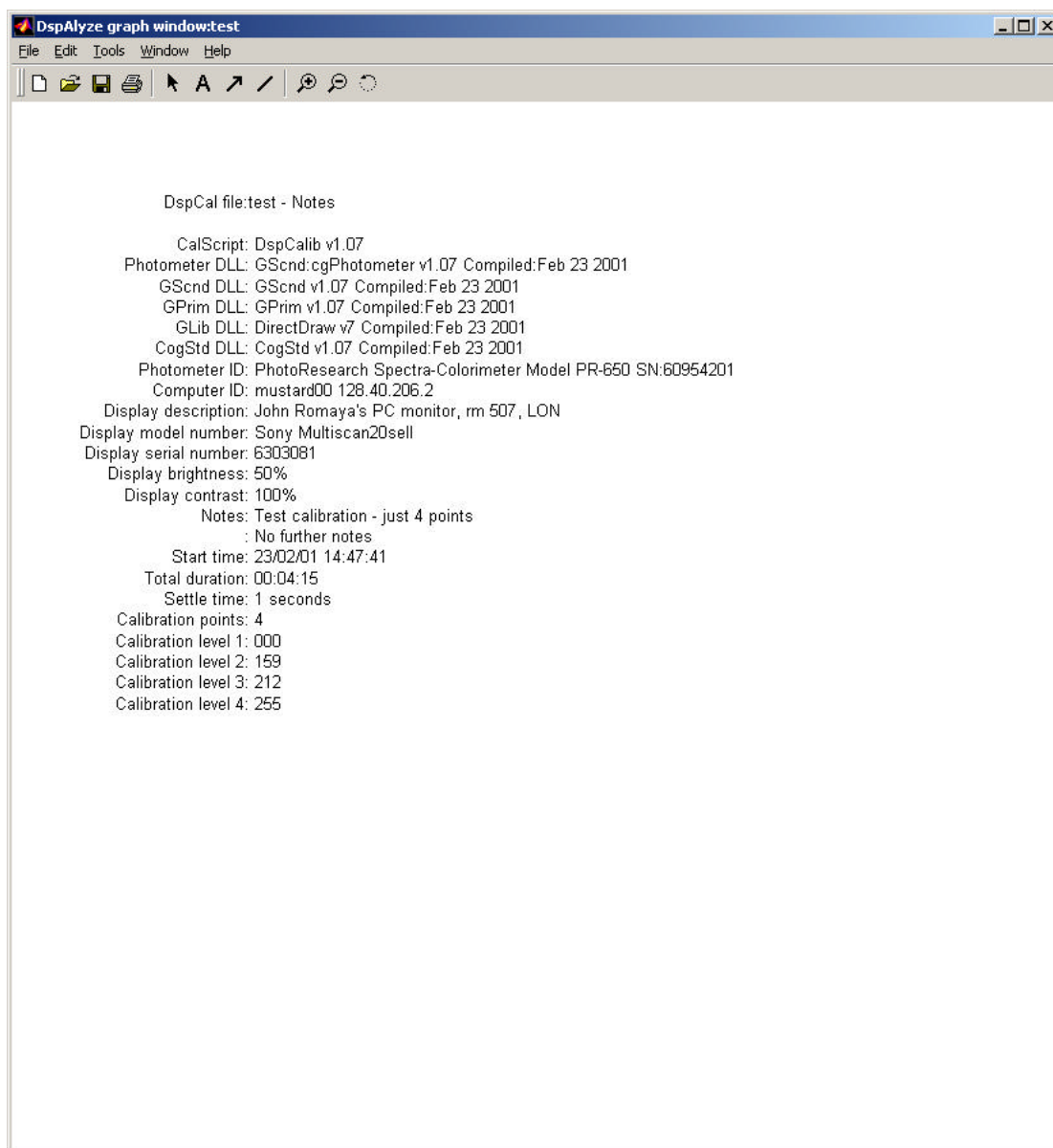
You can obtain a plot of CIE (1931) x and y co-ordinates by hitting the appropriate button...



The CIE plot button shows the monitor phosphors on the CIE chromaticity diagram:-



Finally, the Notes page shows all the other ancilliary information that is stored in the file:-



DspCalib

This function is used to calibrate a display and create a display calibration file. Various colours are displayed on the screen and the CIE XYZ values are measured with the photometer.

This function has been modified so that it works with the v1.17 timing functions which return a time in seconds as well as the pre-v1.17 timing functions which returned a time in microseconds.

Usage: DspCalib(Port,CalPoints,SettleTime,LeaveTime,Filename)

Port	serial or COM port (1-8)
CalPoints	Number of calibration points (2-256)
SettleTime	seconds to allow display to settle (0-60)
LeaveTime	seconds to leave the room (0-60)
Filename	name for display calibration data file

The port number should identify the serial (COM) port to which you connect the photometer. Normally COM1 should be used (Port = 1).

You may select anything from 2 to 256 calibration points. As a general guide I would recommend either 16 or 32 for this value. This gives a good compromise between accuracy and calibration time.

The settle time allows the display to settle before each colour is measured. A value of 5 seconds should be ample.

The leave time gives you some time to leave the room if you want to get on with something during the calibration. It is best to calibrate in a blacked out room so this gives you time to open the door and get out without affecting the calibration.

When you start the calibration you are asked a number of questions as shown below:-

```
EDU» DspCalib(1,16,5,60,'test')
```

```
Enter Display Description: Computer room main monitor
Enter Display Model Number: Sony Multiscan20seII
Enter Display Serial Number: 6303081
Enter Display Brightness: 50%
Enter Display Contrast: 100%
Enter a line of notes: sample calibration
```

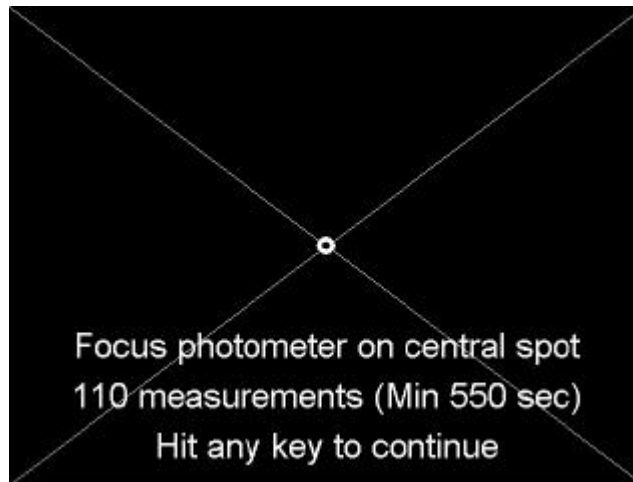
```
Connect PR650 directly to serial port COM1
Switch in CTRL position
Hit a key when ready
```

You should enter all the values as accurately as possible, especially the display brightness and contrast settings. If you are in doubt about some of these values, please ask one of the technicians what you should enter in response to the questions.

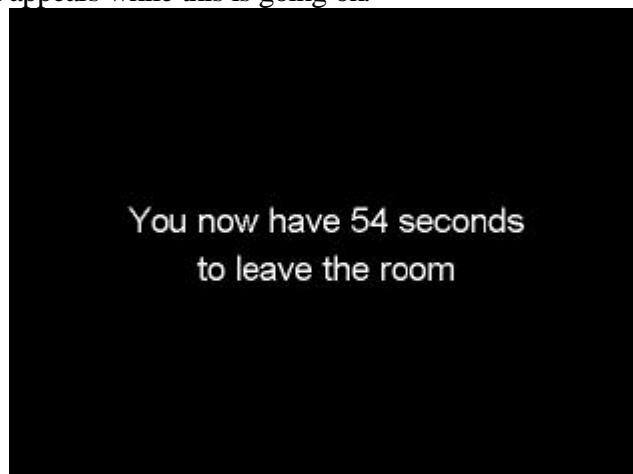
At this point you should connect the photometer to the computer and switch the photometer on by pressing the red '0/1' button. The power light on the photometer should glow.

Then hit a key on the keyboard to go on to the next stage...

The photometer display should illuminate to indicate that communications have been established and a target should appear on the display you are going to calibrate:-

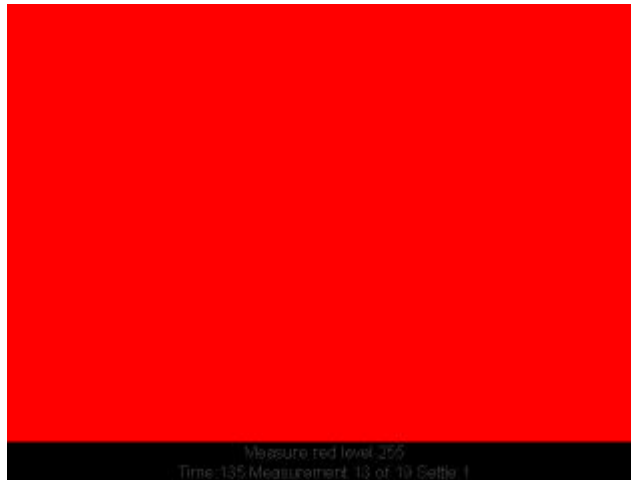


Focus and aim the photometer target spot at the central circle and then make the room you are in as dark as possible. Then press any key to start the calibration. If you wish to leave the room you should have set the 'LeaveTime' appropriately and you should then have time to leave the room and lock the door behind you. A countdown screen appears while this is going on:-



The calibration now begins. First of all four grey screens of varying brightness are displayed to get a rough idea of how the monitor brightness varies over the full range. These screens are measured and used to estimate a linear distribution of calibration points across the monitor output range. Then, for each of the calibration levels the screen XYZ is measured for red, green, yellow, blue, magenta, cyan and grey in turn.

During this phase the display typically appears a uniform colour or as shown below during the settle time:-



The thin dark band at the bottom contains text which gives some indication of how the calibration is progressing; typically:-

Measure Red Level 255
Time:456 Measurement 145 of 149 Settle:5

Finally, when the calibration has finished the screen returns to Windows and you should see a message on the matlab window similar to:-

```
*****  
* Calibration completed successfully      *  
* Calibration file "test" saved          *  
* Now disconnect and pack the photometer *  
*****
```

ReadDCF

This function can be used to read a display calibration file into a structure in the matlab workspace:-

```
EDU» dcf=readdcf('test')

dcf =

    CalScript: 'DspCalib v1.07'
    PhotometerDLL: 'GScnd:cgPhotometer v1.07 Compiled:Feb 23 2001'
    GScndDLL: 'GScnd v1.07 Compiled:Feb 23 2001'
    GPrimDLL: 'GPrim v1.07 Compiled:Feb 23 2001'
    GLibDLL: 'DirectDraw v7 Compiled:Feb 23 2001'
    CogStdDLL: 'CogStd v1.07 Compiled:Feb 23 2001'
    PhotometerID: [1x58 char]
    ComputerID: 'mustard00 128.40.206.2'
    DspDsc: 'John Romaya's PC monitor, rm 507, LON'
    DspModNo: 'Sony Multiscan20seII'
    DspSerNo: '6303081'
    DspBrnt: '50%'
    DspCnt: '100%'
    DspCnf: [1x1 struct]
    Notes: {2x1 cell}
    StartTime: [1x1 struct]
    TotalDuration: [1x1 struct]
    SettleTime: 1
    CalibPoints: 4
    CalibLevel: [0 159 212 255]
    XYZ: [7x4x3 double]
```

The display configuration structure has the following members:-

```
EDU» dcf.DspCnf

ans =

    Width: 640
    Height: 480
    Bits: 32
    Hz: 59.6700
```

The start time structure has the following members:-

```
EDU» dcf.StartTime

ans =

    yr: 1
    mon: 2
    day: 23
    hr: 14
    min: 47
    sec: 41
```


The total duration structure has the following members:-

```
EDU» dcf.TotalDuration
```

```
ans =
```

```
hr: 0  
min: 4  
sec: 15
```

The CalibPoints array gives the RGB level points for each of the requested calibration points and the XYZ array gives the XYZ values for each of the 7 colours at each calibration point.

Spc2XYZ

This function converts a spectrum into CIE (1931) XYZ values. A sample spectrum file 'spc.mat' can be downloaded from the website.

```
EDU» load spc
```

```
EDU» XYZ=spc2xyz(spc)
```

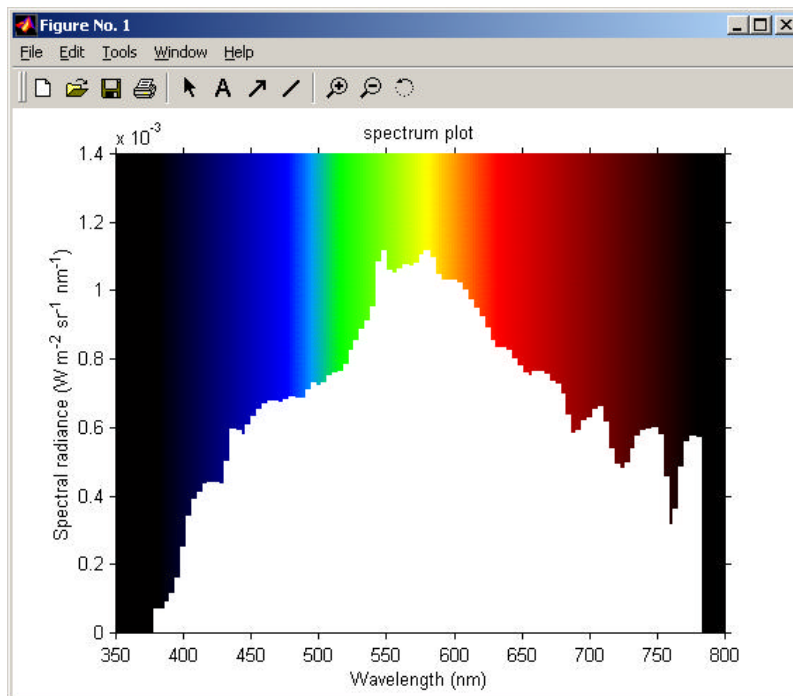
```
XYZ =
```

```
66.9183    69.6944    44.4374
```

SpcPlot

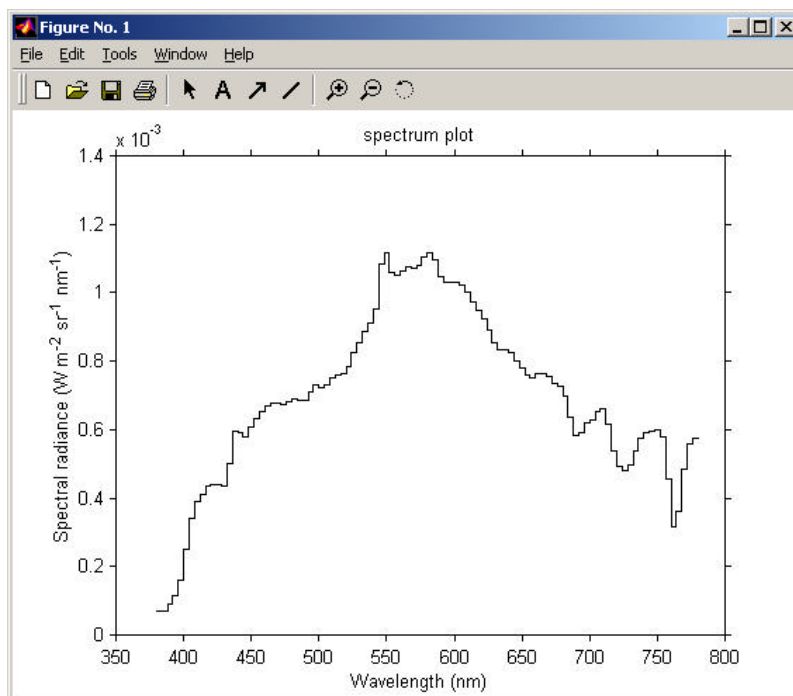
This function plots a spectrum in graphical form. You may download a sample spectrum data file 'spc.mat' from the website:-

```
EDU> load spc
EDU> SpcPlot(spc)
```

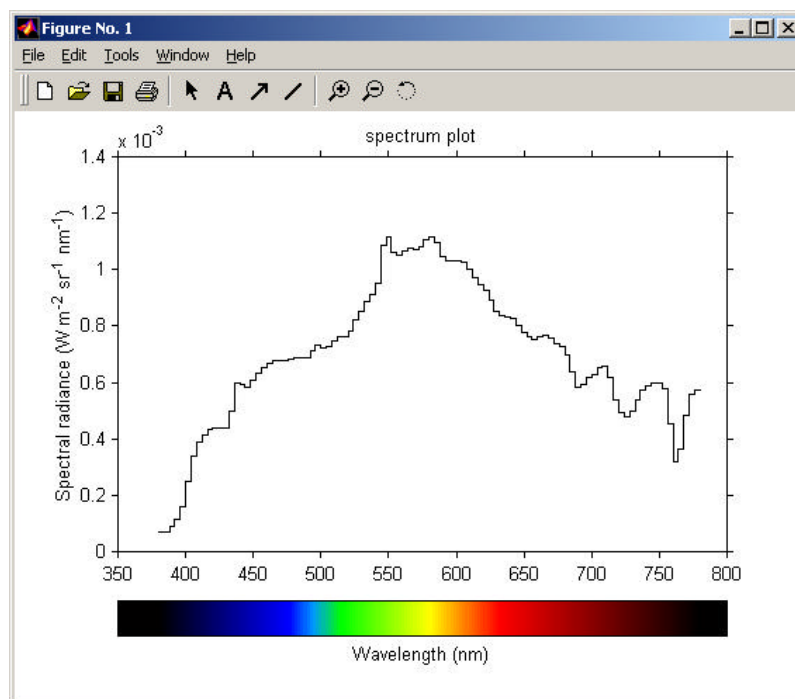


The spectrum is plotted out as shown above. You can also use the command SpcPlot(spc,3) to obtain the figure shown above.

You may plot a simple black and white plot with the command SpcPlot(spc,1):-



Or you can plot a simple plot with a horizontal colour bar using SpcPlot(spc,2):-



XYZ2xyY

This function converts an array of CIE (1931) XYZ values to xyY values:-

```
EDU» XYZ = xyY2XYZ([.2 .5 50;.3 .4 40])
```

```
XYZ =
```

```
20.0000    50.0000    30.0000
30.0000    40.0000    30.0000
```

xyY2XYZ

This function converts an array of CIE (1931) xyY values to XYZ values:-

```
EDU» xyY = XYZ2xyY([20 50 30;30 40 30])
```

```
xyY =
```

```
0.2000    0.5000    50.0000
0.3000    0.4000    40.0000
```