ModelFest Calibration

Andrew B. Watson

Human Factors Research and Technology Division
NASA Ames Research Center
Moffett Field, CA 94035-1000
abwatson@mail.arc.nasa.gov
http://vision.arc.nasa.gov/

Introduction

ModelFest is a collaborative effort among 12 labs to create a public database of psychophysical data that could be used to calibrate and test vision models. Phase 1 of this effort consisted of collecting contrast thresholds for 44 spatial patterns, all contained within a 2x2 degree aperture. Further details regarding this effort are available at http://vision.arc.nasa.gov/modelfest/.

In the Vision Group at NASA Ames Reseaarch Center, we have collected Phase 1 ModelFest data for three observers: abw, brb, and cvr. This report describes some calibration procedures and results obtained for the apparatus used in the ModelFest Phase 1 experiments.

Apparatus

The apparatus for threse experiments consisted of a computer, a graphics display board, an ISR video attenuator , a monochrome monitor, video cabling, a Tektronix J17 photometer, and a Spectra-Physics Photometer/Colorimeter. The details for each device are provided in the following table.

```
Computer        Apple Macintosh PowerPC G3 266 Mhz
Display         Clinton/Richardson M20DCD1RE Monochrome CRT
Attnuator       ISR Video Attenuator
Colorimeter     Photo Research PR-880
Photometer      Tektronix J17 with J1803 Luminance Head
```

The video signal were generated by the built-in video card of the computer. Details for this card are as follows:

```
Card type : display
Card name : ATY,mach64_3DU
Card model : ATY,GT-B
Card revision : 154
Card vendor ID : 1002
```

The ISR Attenuator is a passive device which connects the three R,G,B color outputs of the video card to a single video input on the monitor. The three channels are combined via a resistor network which attenuates the three channels by respective gains of r, g, and b.

Using the Monitors Control Panel on the computer, we set the monitor resolution to 1024 x 768 at 75 Hz. The Clinton Monitor is provided with digital controls for various display settings. Using these controls, we arranged the horizontal and vertical size such that display resolution corresponded to 32 pixels/cm in both dimensions. We set the contrast and brightness to achieve a maximum luminance of approximately 60 cd/m^2. The final digital monitor settings were:

```
brightness    225
contrast    145
horizontal size    150
horizontal center  203
vertical size    76
vertical center    156
rotation    142
pincushion    140
bow      149
trap      139
skew      121
```

The Clinton monitor uses a PC 104 phosphor.

Software

The software used to generate displays and perform calibrations consisted of Apple macintosh System software, including QuickTime media software, *Mathematica* function libraries, and the *MathLink* display function ShowTime. Some version details for this software are as follows:

```
Mac OS overview
 Finder : 8.5
 System : 8.5.1 US
 QuickTime : 3.0.2
 ROM revision : $77D.40F2
 Serial number : XA80701GBG7
 Software bundle : 694-1094

ShowTimeversion1.2
Mathematicaversion3.0.1
```

Calibration

Luminance Measurements

The display system software represents a color as an {R,G,B} triple of values, each of which may range from 0 to 255. The ISR Video Attenuator combines the three color channels, with particular channel gains, into a single signal that may be applied to a monochrome monitor. Thus each triple maps to a prticular luminance. We would like to know the luminance produced by each possible triple, but of course we cannot measure all 255^3 = 16581375 values. Often it is assumed that the three channels contribute independently and additively to the result, but we have not found this to always be true, and have therefor erred on the side of caution. We conduct a scan of the three-dimensional space spanned by the three channels, sampling at each point in a lattice defined by a particular sample spacing (default=32). This is done with the function ISRScan.

```
?ISRScan
```

```
ISRScan[] Collect calibration measurements for the ISR video attenuator.
  Measures the luminance at a three-dimensional lattice of {R,G,B}
  points with a spacing that may be set by the option GrayStep (default
  GrayStep->64). At each point, a square of the corresponding color is
  presented at the center of the screen, which with the ISR attenuator
  in place appears as a monochrome square of a certain luminance. This
  luminance is measured using the Tektronix J17 Photometer. Other
  options that may be specified are: Movie - to set the QuickTime movie
  file that defines the luminance target (this should be a uniform
  region with a graylevel of 1), BackgroundColor - to specify the
  color outside of the luminance target, and Pause - to specify
  the duration of the pause between measurements.
```

Here we illustrate the use of this function.

```
$CurrentISRScan = ISRScan[Pause->2]
```

```
{{{{{0, 0, 0}, 0.013}, {{0, 0, 64}, 0.369},
    {{0, 0, 128}, 5.184}, {{0, 0, 192}, 15.84}, {{0, 0, 255}, 32.08}},
   {{{0, 64, 0}, 0.}, {{0, 64, 64}, 1.121},
    {{0, 64, 128}, 8.323}, {{0, 64, 192}, 21.65}, {{0, 64, 255}, 40.56}},
   {{{0, 128, 0}, 0.013}, {{0, 128, 64}, 1.899},
    {{0, 128, 128}, 10.74}, {{0, 128, 192}, 25.73}, {{0, 128, 255}, 46.32}},
   {{{0, 192, 0}, 0.013}, {{0, 192, 64}, 2.757},
    {{0, 192, 128}, 13.06}, {{0, 192, 192}, 29.63}, {{0, 192, 255}, 51.5}},
   {{{0, 255, 0}, 0.026}, {{0, 255, 64}, 3.733},
    {{0, 255, 128}, 15.35}, {{0, 255, 192}, 33.32}, {{0, 255, 255}, 56.32}}},
  {{{{64, 0, 0}, 0.}, {{64, 0, 64}, 0.488},
    {{64, 0, 128}, 5.83}, {{64, 0, 192}, 17.13}, {{64, 0, 255}, 33.93}},
```

```
 {{{64, 64, 0}, 0.013}, {{64, 64, 64}, 1.385},
  {{64, 64, 128}, 9.207}, {{64, 64, 192}, 23.15}, {{64, 64, 255}, 42.75}},
 {{{64, 128, 0}, 0.}, {{64, 128, 64}, 2.256},
  {{64, 128, 128}, 11.75}, {{64, 128, 192}, 27.5}, {{64, 128, 255}, 48.61}},
 {{{64, 192, 0}, 0.013}, {{64, 192, 64}, 3.232},
  {{64, 192, 128}, 14.11}, {{64, 192, 192}, 31.43}, {{64, 192, 255}, 53.68}},
 {{{64, 255, 0}, 0.013}, {{64, 255, 64}, 4.274}, {{64, 255, 128}, 16.59},
  {{64, 255, 192}, 35.21}, {{64, 255, 255}, 58.95}}},
{{{{128, 0, 0}, 0.013}, {{128, 0, 64}, 0.58},
  {{128, 0, 128}, 6.265}, {{128, 0, 192}, 17.9}, {{128, 0, 255}, 35.1}},
 {{{128, 64, 0}, 0.}, {{128, 64, 64}, 1.57},
  {{128, 64, 128}, 9.748}, {{128, 64, 192}, 24.13}, {{128, 64, 255}, 44.06}},
 {{{128, 128, 0}, 0.013}, {{128, 128, 64}, 2.506}, {{128, 128, 128}, 12.39},
  {{128, 128, 192}, 28.54}, {{128, 128, 255}, 50.08}},
 {{{128, 192, 0}, 0.013}, {{128, 192, 64}, 3.522}, {{128, 192, 128}, 14.84},
  {{128, 192, 192}, 32.58}, {{128, 192, 255}, 55.26}},
 {{{128, 255, 0}, 0.013}, {{128, 255, 64}, 4.643}, {{128, 255, 128}, 17.38},
  {{128, 255, 192}, 36.44}, {{128, 255, 255}, 60.6}}},
{{{{192, 0, 0}, 0.}, {{192, 0, 64}, 0.673},
  {{192, 0, 128}, 6.608}, {{192, 0, 192}, 18.61}, {{192, 0, 255}, 36.13}},
 {{{192, 64, 0}, 0.}, {{192, 64, 64}, 1.715},
  {{192, 64, 128}, 10.25}, {{192, 64, 192}, 24.98}, {{192, 64, 255}, 45.23}},
 {{{192, 128, 0}, 0.013}, {{192, 128, 64}, 2.717}, {{192, 128, 128}, 12.95},
  {{192, 128, 192}, 29.48}, {{192, 128, 255}, 51.3}},
 {{{192, 192, 0}, 0.013}, {{192, 192, 64}, 3.786}, {{192, 192, 128}, 15.49},
  {{192, 192, 192}, 33.5}, {{192, 192, 255}, 56.58}},
 {{{192, 255, 0}, 0.013}, {{192, 255, 64}, 4.96}, {{192, 255, 128}, 18.07},
  {{192, 255, 192}, 37.47}, {{192, 255, 255}, 61.99}}},
{{{{255, 0, 0}, 0.}, {{255, 0, 64}, 0.752},
  {{255, 0, 128}, 6.951}, {{255, 0, 192}, 19.26}, {{255, 0, 255}, 37.09}},
 {{{255, 64, 0}, 0.}, {{255, 64, 64}, 1.886},
  {{255, 64, 128}, 10.7}, {{255, 64, 192}, 25.73}, {{255, 64, 255}, 46.3}},
 {{{255, 128, 0}, 0.026}, {{255, 128, 64}, 2.928}, {{255, 128, 128}, 13.4},
  {{255, 128, 192}, 30.3}, {{255, 128, 255}, 52.42}},
 {{{255, 192, 0}, 0.013}, {{255, 192, 64}, 4.036}, {{255, 192, 128}, 16.05},
  {{255, 192, 192}, 34.39}, {{255, 192, 255}, 57.83}},
 {{{255, 255, 0}, 0.026}, {{255, 255, 64}, 5.25}, {{255, 255, 128}, 18.69},
  {{255, 255, 192}, 38.38}, {{255, 255, 255}, 63.18}}}}
```

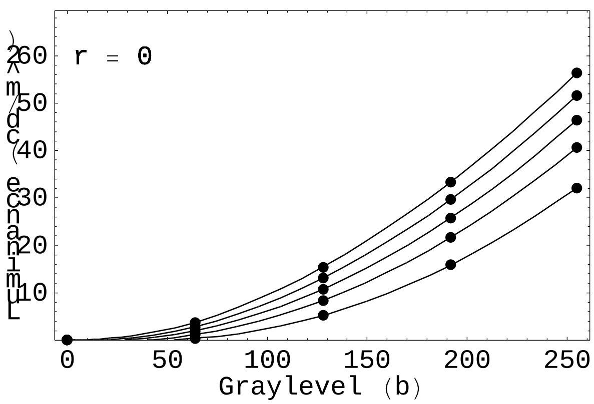
This function returns a data structure consisting of a three-dimensional array, each of whose entries is a list of the color value used and the luminance measured. We model these data using an interpolation function, as they are provided by *Mathematica*. The interpolating function is created by the function ISRInterpolation.

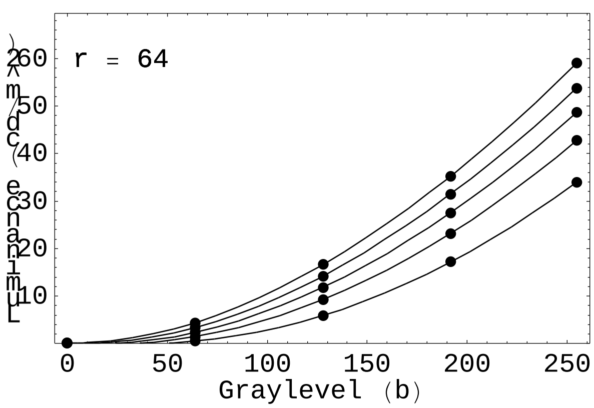
```
?ISRInterpolation
```

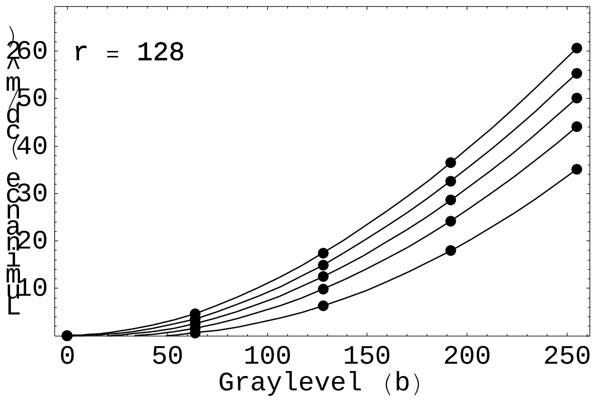
```
ISRInterpolation[data_, order_:3,verbose_:False] - Create an
  InterpolatingFunction for the display calibration data. data is an
  array of dimensions {nR,nG,nB} (typically {5,5,5}), each element of
  which is a list of the form {{r,g,b},luminance} which describes the
  luminance measured from a display employing an ISR attenuator when
  the color {r,g,b} is displayed. The order of the interpolation
  can be specified. If verbose is True, plots of the scan data
  and interpolating function are produced.
```

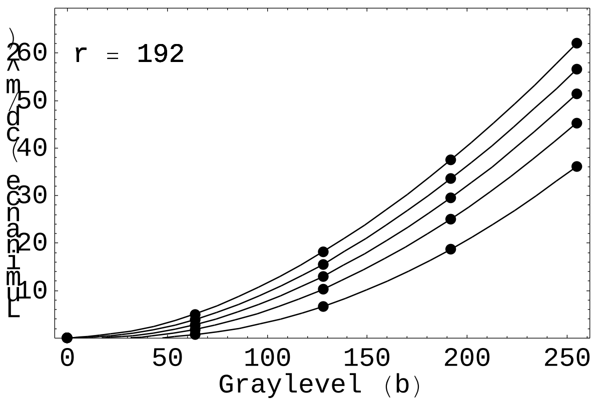
Here we use this function, requesting an order of 2 and plotting results. Note that an interpolating function is returned, which we assign to the symbol $ISRDescription ($ signs indicate global symbols). Examination of the plots, in which each panel is for a distinct value of r, the abscissa is b, and the several curves in each panel are for different values of g, shows that the gains for the three channels are, from lowest to highest, R<G<B.

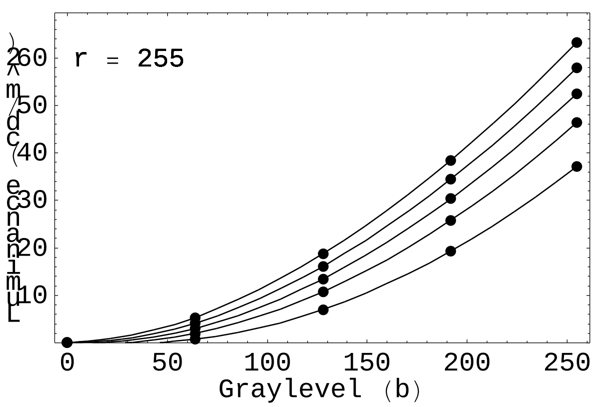
```
$ISRDescription = ISRInterpolation[$CurrentISRScan,2,True]
```












```
InterpolatingFunction[{{0, 255.}, {0, 255.}, {0, 255.}}, <>]
```

The interpolating function takes the arguments r,g,b and returns a predicted luminance.

```
$ISRDescription[1,128,255]
```

```
46.3621
```

We define the desired mean luminance as half the maximum luminance, which we compute from our interpolating function.

```
$MeanLuminance = $ISRDescription[255,255,255]/2.
```

```
31.59
```

Predicted luminances can also be obtained from a slightly more general function which takes a color {r,g,b} and a description.

```
?ISRDisplayModel
```

```
ISRDisplayModel[color_,description_] : Compute the luminance expected
  for a specified color {r,g,b}, on a display connected via an ISR video
  attenuator. Color may also be a list of colors {{r1,g1,b1},{r2,g2,
  b2},...}. The display is characterized by a description, which may be
  in the form of Gamma function parameters, a set of polynomial
  coefficients and exponents, or an interpolating function. These
  may be aqcuired via the functions CalibrateISRAttenuator,
  ISRFitPolynomial, or ISRInterpolation, respectively.
```

Here is an example.

```
ISRDisplayModel[{1,128,255},$ISRDescription]
```

```
46.3621
```

The reason for this generality is that on some displays, simpler models, such as the traditional independent gamma functions model, may be adequate.

Now we can use this interpolating function to create linearized color look-up-tables (CLUTs). We do this with the function ISRCLUT. The algorithm employed by ISRCLUT is described later in this document.

```
?ISRCLUT
```

```
ISRCLUT[contrast_, description_, meanLuminance_, verbose_:False] -
  Compute a linearized color look-up-table (CLUT) with a specified
  contrast, using a specified ISR description and mean luminance. The
  description may be which may be in the form of Gamma function
  parameters, a set of polynomial coefficients and exponents, or
  an interpolating function (see ISRDisplayModel). If verbose
  is TRUE, plots of the CLUT are produced.
```

Here we create a set of 26 CLUTs, for contrasts ranging in 2 dB steps from -50 to 0 dB.

```
$CLUTS = ISRCLUT[#, $ISRDescription, $MeanLuminance]& /@
    dBInverse[Range @@ {-50,0,2}];
```

looking at the dimensions of this data structure, we see that each CLUT is 256 x 3 elements in size.

```
Dimensions[$CLUTS]
```

```
{26, 256, 3}
```

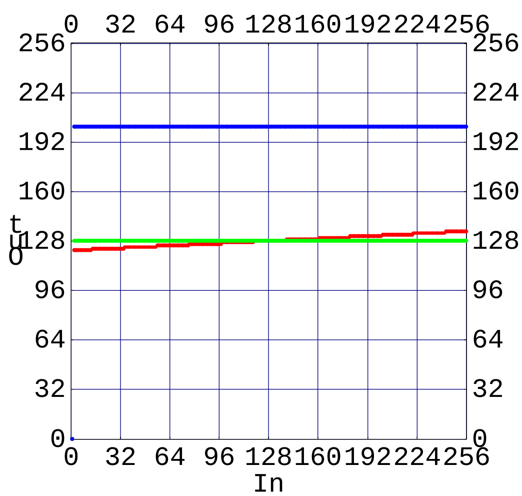
We can examine the CLUTs with the function PlotCLUT.

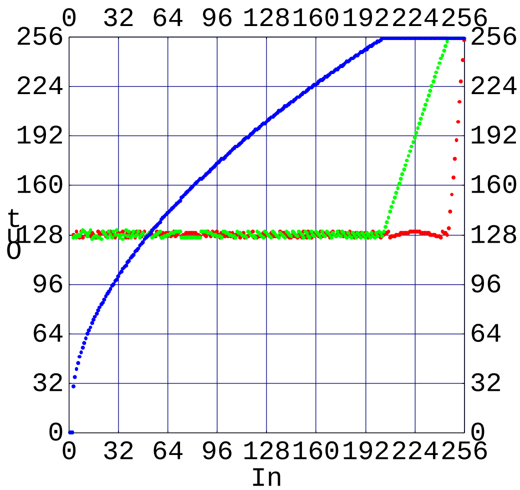
```
?PlotCLUT
```

```
PlotCLUT[clut_, opts___] Plot a color look up-table. The table is
  an array of dimensions {256,3} with each triple representing
  values for red, green and blue. Options for ListPlot can be supplied.
```

Here we look at the CLUTS for the smallest and the largest contrasts. For the smallest contrast, only the red channel is varied. For the highest contrast, all three channels are varied.

```
PlotCLUT /@ $CLUTS[[{1,26}]];
```





These CLUTS may now be used to produce linearized contrast displays using the ShowTime display function.

```
?ShowTime
```

```
ShowTime[filename_,opts___] : Display a QuickTime movie. Available
  options are: HoldLastFrame->0: how long to retain the last frame of
  the movie in seconds; -1 means hold forever, CLUT->Automatic: color
  look-up-table with dimensions {256,3}; if Automatic, three linear
  ramps are generated automatically, Background->{128,128,128}: color of
  background (range = {0,255}, Monitor->1: number of monitor for
  display, Note->0: MIDI tone to play before display, Delay->.05: delay
  in seconds between note end and display start, Duration->.05: duration
  in seconds of note,     Rate->Normal: Display the movie at normal
  speed, or at maximum speed (Rate->Fast)     Sound->NullString
  Name of a QuickTime movie file containing a sound     Scale->
  1: magnification     Clear->False: clear the screen prior
  to display. Note that the mathlink program "showtime" must be
  installed before this function is evaluated.
```

Here is an example

```
showlink=Install["showtime"];
ShowTime["gabor.movie",CLUT->$CLUTS[[12]]];
Uninstall[showlink];
```

The next step is to verify the operation of the linearized CLUTS. This we do with the function CheckISRCLUT.

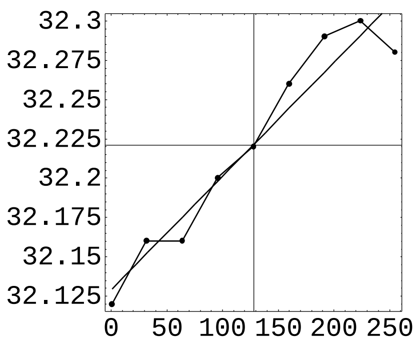
```
?CheckISRCLUT
```

CheckISRCLUT[clut_,opts___Rule] - Check an ISR CLUT by stepping through it and recording the actual luminance produced. The actual mean luminance and contrast are reported (by fitting a linear function to the data). Possible options and defaults are: Simulate->False, to use the ISRDisplayModel, rather than actul measurements; GrayStep->32, to determine the step between graylevels; Description->None, to specify an ISRDescription, where necessary for simulation.

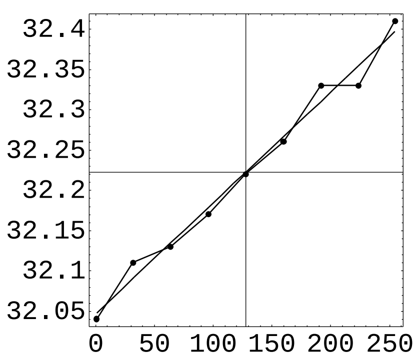
We will apply this function to a subset of the 24 CLUTS.

```
CheckISRCLUT[#,GetLuminance->J17Photometer]& /@ $CLUTS[[{1,4,8,16,20,26}]]
```

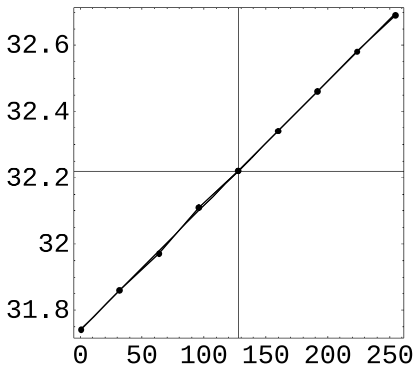
Contrast: = 0.00284633, mean = 32.2211



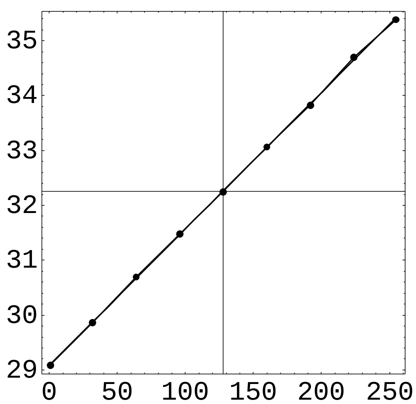
Contrast: = 0.00542012, mean = 32.2222



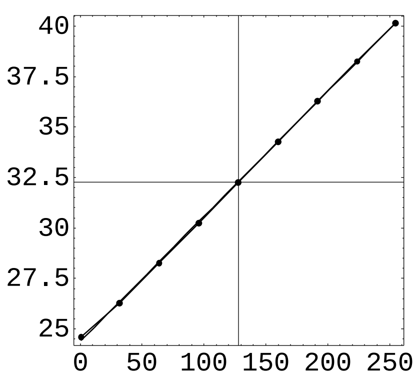
Contrast: = 0.0147819, mean = 32.2189



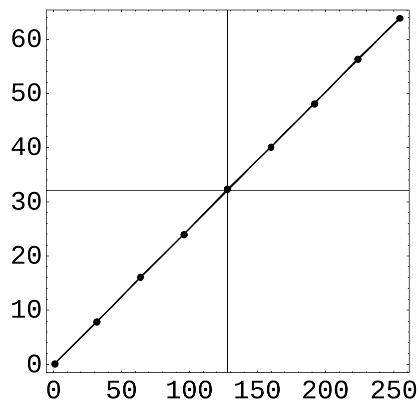
Contrast: = 0.0979016, mean = 32.2544



Contrast: = 0.243355, mean = 32.2689



Contrast: = 0.996908, mean = 32.0007




```
{{{1, 32.12}, {32, 32.16}, {64, 32.16}, {96, 32.2},
   {128, 32.22}, {160, 32.26}, {192, 32.29}, {224, 32.3}, {255, 32.28}},
  {{1, 32.04}, {32, 32.11}, {64, 32.13}, {96, 32.17},
   {128, 32.22}, {160, 32.26}, {192, 32.33}, {224, 32.33}, {255, 32.41}},
  {{1, 31.74}, {32, 31.86}, {64, 31.97}, {96, 32.11},
   {128, 32.22}, {160, 32.34}, {192, 32.46}, {224, 32.58}, {255, 32.69}},
  {{1, 29.08}, {32, 29.86}, {64, 30.69}, {96, 31.47},
   {128, 32.24}, {160, 33.06}, {192, 33.82}, {224, 34.69}, {255, 35.38}},
  {{1, 24.57}, {32, 26.26}, {64, 28.24}, {96, 30.23},
   {128, 32.24}, {160, 34.26}, {192, 36.26}, {224, 38.23}, {255, 40.13}},
  {{1, 0.04}, {32, 7.796}, {64, 16.04}, {96, 23.91}, {128, 32.22},
   {160, 39.97}, {192, 48.01}, {224, 56.25}, {255, 63.77}}}
```

The plots indicate that a linearized grayscale has been obtained. Each plot is accompanied by a printout of the estimated contrast and mean luminance, as obtained from the best fitting straight line. The contrasts obtained can be compared to the contrasts intended.

```
intended = dBInverse[Range @@ {-50,0,2}][[{1,4,8,16,20,26}]]
```

```
{0.00316228, 0.00630957, 0.0158489, 0.1, 0.251189, 1}
```

```
actual = {0.00284633,0.00542012,0.0147819,0.0979016,0.243355,0.996908}
```

```
{0.00284633, 0.00542012, 0.0147819, 0.0979016, 0.243355, 0.996908}
```

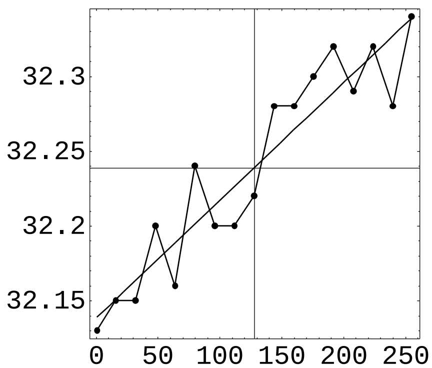
```
actual/intended
```

```
{0.900089, 0.859031, 0.932675, 0.979016, 0.968814, 0.996908}
```

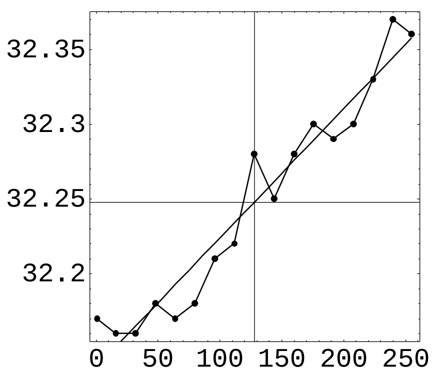
The accuracy of the CLUT is clearly the lowest for the lowest contrast tables. The D/A converters of course impose a lower bound on the residual error. It is of some interest whether the departures from linearity are consistent or due to error in the display and the photometer. To examine this, we make four repeated measurements, at a finer scale, of the lowest contrast table (a nominal contrast of 0.0032).

```
tmp = CheckISRCLUT[#,GetLuminance->J17Photometer,GrayStep->16]& /@
$CLUTS[[{1,1,1,1}]]
```

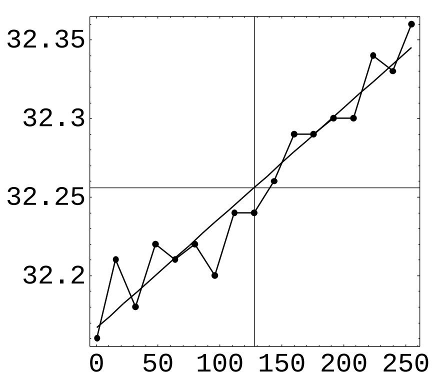
```
Contrast: = 0.00309689,  mean = 32.2388
```



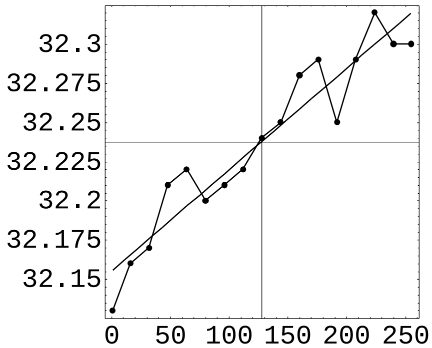

```
Contrast: = 0.00339992,  mean = 32.2476
```




```
Contrast: = 0.00276228,  mean = 32.2559
```

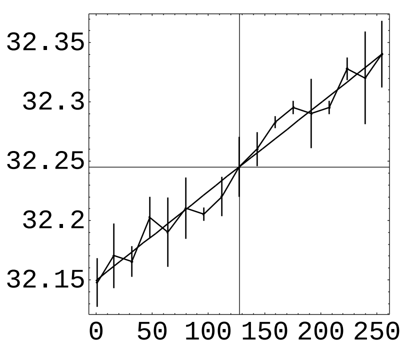


```
Contrast: = 0.00254059,  mean = 32.2376
```




```
{{{1, 32.13}, {16, 32.15}, {32, 32.15}, {48, 32.2},
  {64, 32.16}, {80, 32.24}, {96, 32.2}, {112, 32.2}, {128, 32.22},
  {144, 32.28}, {160, 32.28}, {176, 32.3}, {192, 32.32}, {208, 32.29},
  {224, 32.32}, {240, 32.28}, {255, 32.34}}, {{1, 32.17}, {16, 32.16},
  {32, 32.16}, {48, 32.18}, {64, 32.17}, {80, 32.18}, {96, 32.21},
  {112, 32.22}, {128, 32.28}, {144, 32.25}, {160, 32.28}, {176, 32.3},
  {192, 32.29}, {208, 32.3}, {224, 32.33}, {240, 32.37}, {255, 32.36}},
 {{1, 32.16}, {16, 32.21}, {32, 32.18}, {48, 32.22},
  {64, 32.21}, {80, 32.22}, {96, 32.2}, {112, 32.24}, {128, 32.24},
  {144, 32.26}, {160, 32.29}, {176, 32.29}, {192, 32.3}, {208, 32.3},
  {224, 32.34}, {240, 32.33}, {255, 32.36}}, {{1, 32.13}, {16, 32.16},
  {32, 32.17}, {48, 32.21}, {64, 32.22}, {80, 32.2}, {96, 32.21},
  {112, 32.22}, {128, 32.24}, {144, 32.25}, {160, 32.28}, {176, 32.29},
  {192, 32.25}, {208, 32.29}, {224, 32.32}, {240, 32.3}, {255, 32.3}}}
```

We can then plot means and standard deviations for the luminance estimates, which suggests that most of the larger perturbations are not persistent or systematic, but are rather random fluctuations in display luminance of photometer measurement.



The average standard deviation is 0.0186681 cd/m^2.

Colorimetric Measurements

Colorimetric measurements were made with a Photo Research Pritchard PR-880 Photometer/Colorimeter.The device was used in the "Relative Colorimetric" mode. Other settings were as follows:

```
Aperture : 1 degree
Front Filter  ND - 1
View shutter  closed
Lens    MS - 55
```

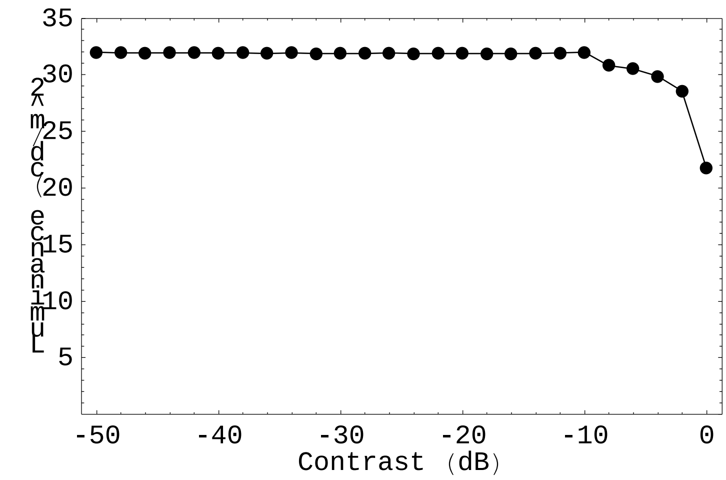
The measurement obtained was as follows:

```
Y = 30.5 cd / m^2
x = 0.270
y = 0.303
CCT = 1026 °K
Frame rate = 74.8 Hz
Date 4 / 29 / 99
Time 10 : 24 : 07
```

Nonlinear Spatial Interactions

It is well known that cathode-ray tube displays are subject to nonlinear spatial interactions between nearby pixels. Essentially, the monitor electronics causes each pixel to smear a bit to its neighbor to the right, which when passed through the non-linear Gamma function, causes nonlinear behaviour of the display in spite of linearization of the display Gamma thorugh look-up tables. An excellent discussion and mode of this phenomenon is available in Mulligan & Stone (1989). Most of the ModelFest stimuli contained modulation primarily in the vertical dimension and are thus immune from this effect. However the binary noise stimulus is the most at risk, since it contains full-excursion (black to white) modulations in the horizontal direction. To guard against such problems the stimulus was pixel-replicated in both horizotal and vertical diemnsions. Here we make some direct measurements of nonlinear spatial interactions to assess their possible impact.

One simple test for these effects is to measure the luminance produced by a single-pixel checkerboard pattern of various contrasts, with the linearizing CLUT in place. If no spatial nonlinearities are present, then the luminance should be equal to the mean luminance, regardless of contrast. If nonlinearities are present, then at some contrast luminance will decline. The plot below shows the measured luminance for a checkerboard for each of the 26 CLUTs computed for these experiments, which range in contrast from -50 to 0 dB in 2dB steps. It is evident that the efects appear at a contrast of about -10 dB. Since our binary noise was pixel-replicated (which reduces the effect) and since threshold for that stimulus were never above -20 dB, it appears that spatial nonlinearities are not a large concern.



Building CLUTS

Given an ISRDescription, a CLUT for a given mean luminance and contrast is constructed as follows. First, with r and g fixed at 128, we find the value b=b0 that yields a luminance L0 closest to the the desired mean. Call this color {r0=128,g0=128,b0}.

For a given contrast, we compute the list of desired luminances for the 254 grayscale values between -127 and 127. Using the ISRDisplayModel, we compute the range of luminances that can be generated by varying only the red (lowest gain) channel, with g=g0, b=b0. If this range includes the range of desired luminances, then by means of the ISRDisplayModel find the values of r for which {r,g0,b0} yields the closest value to each desired luminance. Otherwise, compute the range of luminances that can be generated by varying only the red and green (two lowest gain) channels, with b=b0.If this range includes the range of desired luminances, then by means of the ISRDisplayModel find the values of r and g for which {r,g,b0} yields the closest value to each desired luminance. Otherwise, by means of the ISRDisplayModel find the value of {r,g,b} which yields the closest value to each desired luminance. These rules are similar, but sligtly more general than those described in Pelli & Zhang (1991).

Pelli & Zhang (1991) do not actually describe how to find an {r,g,b} triple that yields a given luminance, given a function like ISRDisplayModel that maps from {r,g,b} to luminance. When only one color channel is allowed to vary (as is the case for very low contrasts), then even direct search may be used since the number of evaluations is limited to 255. But if all three channels are varied this would require up to 255^3=16581375 evaluations. Instead we do the following. We first construct an IntegerBisection operator that uses the method of bisection to find the integer argument of a function that yields a value closest to a given value. To find {r,g,b} in the case where all three are varied, we first fix r and g at 128 and use IntegerBisection to find the closest value of b. Then we vary g, and then we vary r.

References

Berns, R. S. (1996). Methods for characterizing CRT displays. Displays 16(4), 173-182.

Brainard, D. H. (1989). Calibration of a computer controlled color monitor. Colour Research and Applications 14(1), 23-34.

Pelli, D. G. & Zhang, L. (1991). Accurate control of contrast on microcomputer displays. Vision Research 31(7), 1337-1350.

Mulligan, J. B. & Stone, L. S. (1989). Halftoning method for the generation of motion stimuli. JOSA A 6(8), 1217-1227.

Sperling, G. (1971). The Description and luminous calibration of cathode ray oscilloscope visual displays. Behavioral Research Methods and Instrumentation 3,

Watson, A. B., Nielsen, K. R. K., Poirson, A., Fitzhugh, A., Bilson, A., Nguyen, K. & A. J. Ahumada, J. (1986). Use of a raster framebuffer in vision research. Behavioral Research Methods, Instruments, & Computers 18(6), 587-594.

Links

Clinton Monitors	http://www.cec-displays.com/
ISR Video Attenuator	http://rajsky.psych.nyu.edu/tips/Attenuator.html
Richardson Electronics	http://www.rell.com/
Photo Research	http://www.photoresearch.com/
Tektronix J17 Photometer	http://www.tek.com/Measurement/Products/catalog/j17/index.html
ModelFest	http://vision.arc.nasa.gov/modelfest/