Non-device-specific display characterization software, Mcalibrator2 manual

Ban, H., and Yamamoto, H. (2013). A non-device-specific approach to display characterization based on linear, non-linear, and hybrid search algorithms. Journal of Vision.

Hiroshi Ban

April 19 2012

Introduction – what is Mcalibrator2 ?

Display characterization is an essential part of the experimental procedures in vision science because almost all current experiments are conducted in computer-aided environments: visual stimuli are manipulated via a programming language and displayed on a computer display, and observer's responses are acquired by pressing keys connected to a computer. To ensure that visual stimuli – their luminance, color, timing, etc. are presented precisely in such computer-based experiments, researchers need to characterize display devices accurately in advance of actual experiments.

CRT (Cathode Ray-Tube) displays are the most widely-used devices for current vision experiments, and the calibration procedures to characterize their luminance and chromaticities are well-established in the two-stage procedures: gamma-correction, followed by a linear color transformation (**Berns et al., 1996; Brainard et al., 2002; Ban et al., 2006**). The calibration results obtained through this standard two-step procedure has been tested (**Brainard et al., 2002; Ban et al., 2006**) and the quality of luminance and chromatic stimuli on CRT displays satisfy the researchers' criterion.

However, non-CRT devices such as a LCD (Liquid Crystal Display) and a DLP (Digital Light Processing) have come into the main stream recently and researchers are required to use non-CRT over CRT devices. There is no evidence that the standard display characterization method cannot be applied to these new types of devices since the current widely-used standard display calibration method is established based on the internal model of CRT devices.

We have therefore developed a non-device-specific approach to display characterization. Specifically, our new methods use model-free gamma-correction procedure combined with a linear/non-linear (Nelder-Mead Simplex, **Nelder & Mead, 1965; Dennis & Woods, 1987**) hybrid or line search (Powell's method with Coggins constrain, **Powell, 1964; Brent, 1973; Press et al., 2007; Farhi, 2011; Farhi et al, 2012**) algorithm to get the optimal RGB video input values to produce the required luminance and chromaticities. The methods only assume 1) a monotonic increment of luminance against the increment of video input values, and 2) a piece wise linearity of the system in the initial estimation step. These new methods have a much broader range of applicability because they do not presume the internal model of the display device and can handle non-linearity of the device.

The whole procedures are integrated into a GUI-based display characterization software written in MATLAB (The Mathworks, Inc, USA) language and termed

“Mcalibrator2”. The applicability and efficiency of our software to a wide range of display devices, including LCD and DLP type of devices, were confirmed by comparing the calibration accuracies of our procedures with that of the standard two-stage method. We found that all of these new approaches improved calibration accuracies for non-CRT devices compared with the standard display characterization procedure.

Mcalibrator2 has provided some additional components to communicate with any photometers that researchers are using in their daily measurements. The software has also provided simple frameworks to add alternative chromaticity optimization procedures researchers may want to use. Further, the software can automatically generate gamma-correction tables compatible with Psychtoolbox, one of the most widely used vision science tools (**Brainard, 1997; Pelli, 1997**). These functions will assist researchers in characterizing their display devices.

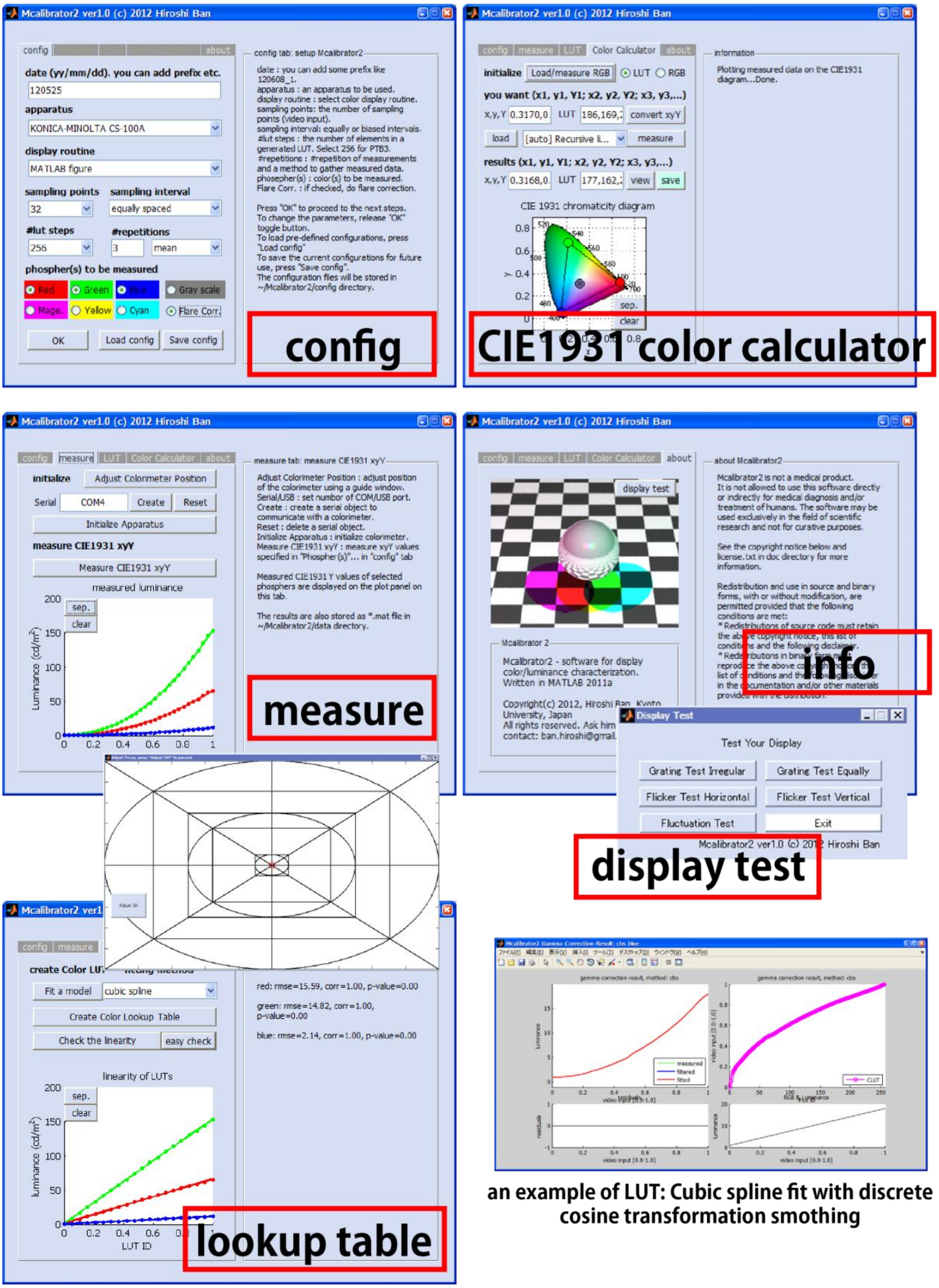
The authors provide this software suite in the hope that it will benefit researchers performing calibration of their display devices efficiently and improve accuracies of stimulus displays regardless of the display types.

References

1. Ban, H., Yamamoto, H., Ejima, Y. (2006). Mcalibrator: MATLAB integrated GUI software for display calibration — a proposal of a new color calibration procedure applicable to a wide range of display devices and evaluation of its efficiency —. The Japanese Journal of Psychonomic Science. 24(2). 149-161.
2. Berns, R.S. (1996). Methods for Characterizing CRT Displays. Displays. 16, 173-182.
3. Brainard, D.H. (1997). The psychophysics toolbox. Spatial Vision. 10, 433-436.
4. Brainard, D.H., Pelli, D.G., Robson, T. (2002). Display Characterization. Encyclopedia of Imaging Science and Technology. Wiley, 172-188
5. Brent, R.P. (1973). Section 7.3: Powell's algorithm. In *Algorithms for minimization without derivatives*. Englewood Cliffs, N.J.: Prentice-Hall.
6. Dennis, J.E.Jr., Woods, D.J. (1987). New Computing Environments: Microcomputers in Large-Scale Computing. Edited by A. Wouk, SIAM, 116-122.
7. Farhi, E. (2011). The iFit data analysis library, ILL, Computing for Science. Online reference: http://ifit.mccode.org
8. Farhi, E., Debab, Y., Willendrup, P. (in press). Journal of Neutron Research.
9. Nelder, J.A., Mead, R. (1965). A Simplex Method for Function Minimization. Computer Journal, 7, 308-313.
10. Pelli, D.G. (1997). The video toolbox software for visual psychophysics: transforming numbers into movies. Spatial Vision. 10, 437-442.
11. Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer Journal 7 (2): 155-162.

12. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (2007). Section 9.3. Van Wijngaarden–Dekker–Brent Method. *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press.

Mcalibrator2 GUI



an example of LUT: Cubic spline fit with discrete cosine transformation smothing

Installation

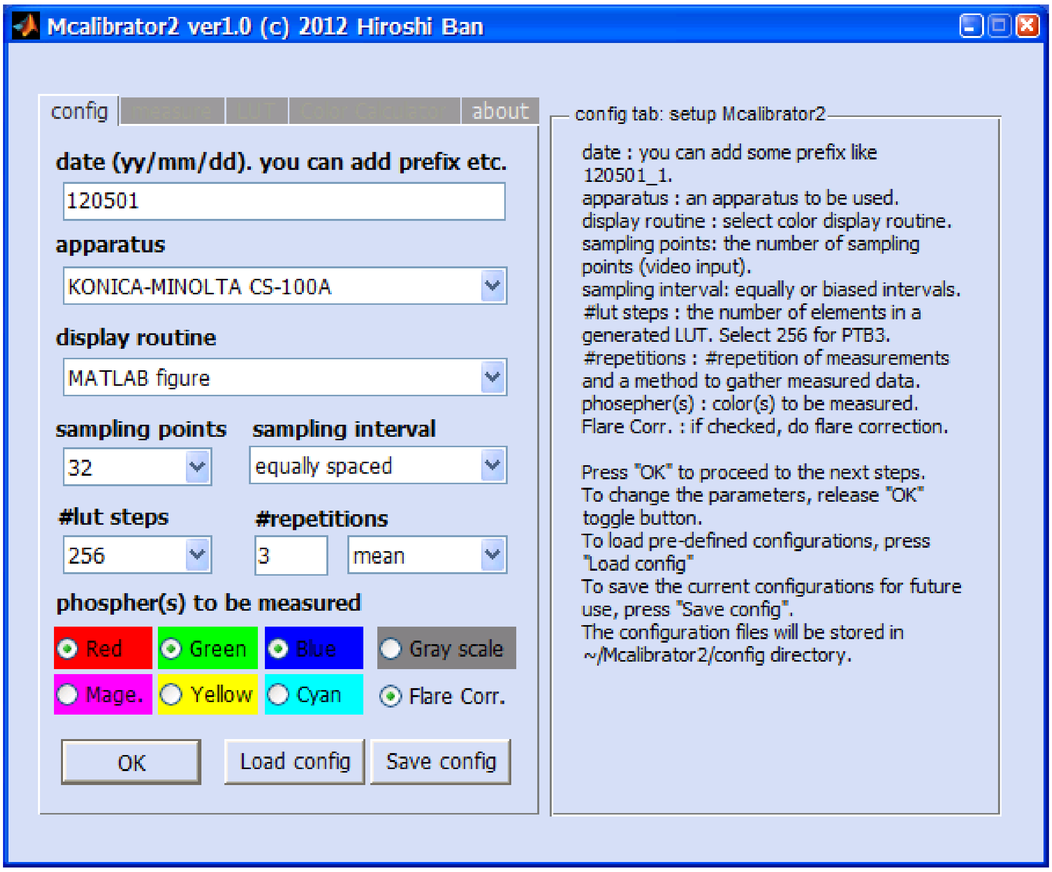
1. You need to install MATLAB (Mathworks, USA) to use Mcalibrator2. You additionally need to install "Optimization Toolbox" (Mathworks official add-on toolbox) to use all the functions of Mcalibrator2.
2. Please download Mcalibrator2.tar.gz from the link below,
   http://www.cv.jinkan.kyoto-u.ac.jp/site/mcalibrator/
   copy it on your computer, extract it, and set MATLAB-path to the directory from MATLAB. If you use MATLAB after 2011a, you do not need to add subdirectories. Mcalibrator2 can handle related subroutines to access all the sub-functions correctly once Mcalibrator2.m is set on the path. If you use MATLAB before 2011a, please set MATLAB-path to the Mcalibrator2 including all the subdirectories.

Launching Mcalibrator2 from MATLAB and setting-up initial parameters

To launch Mcalibrator2, run the command below on MATLAB command line,

>> *Mcalibrator2 [ENTER]*

Then, you will see the main window of Mcalibrator2 as below.

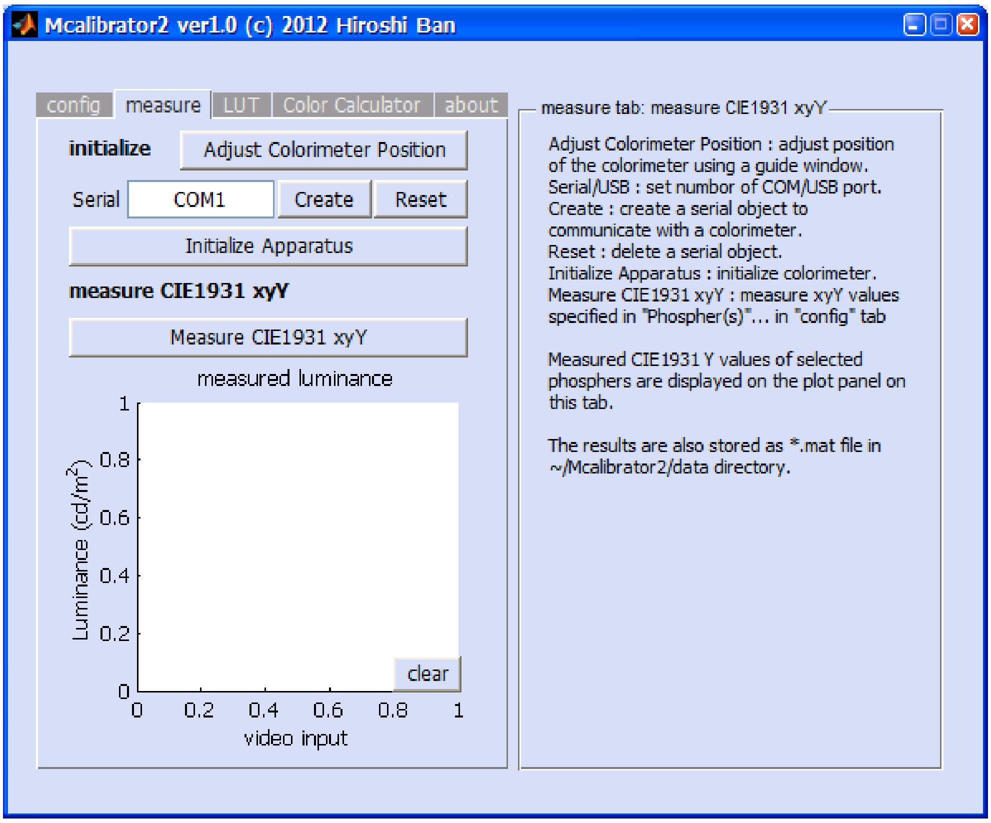


Mcalibrator2 config window (for setting measurement/calibration parameters)

This "config" tab is for setting measurement parameters.

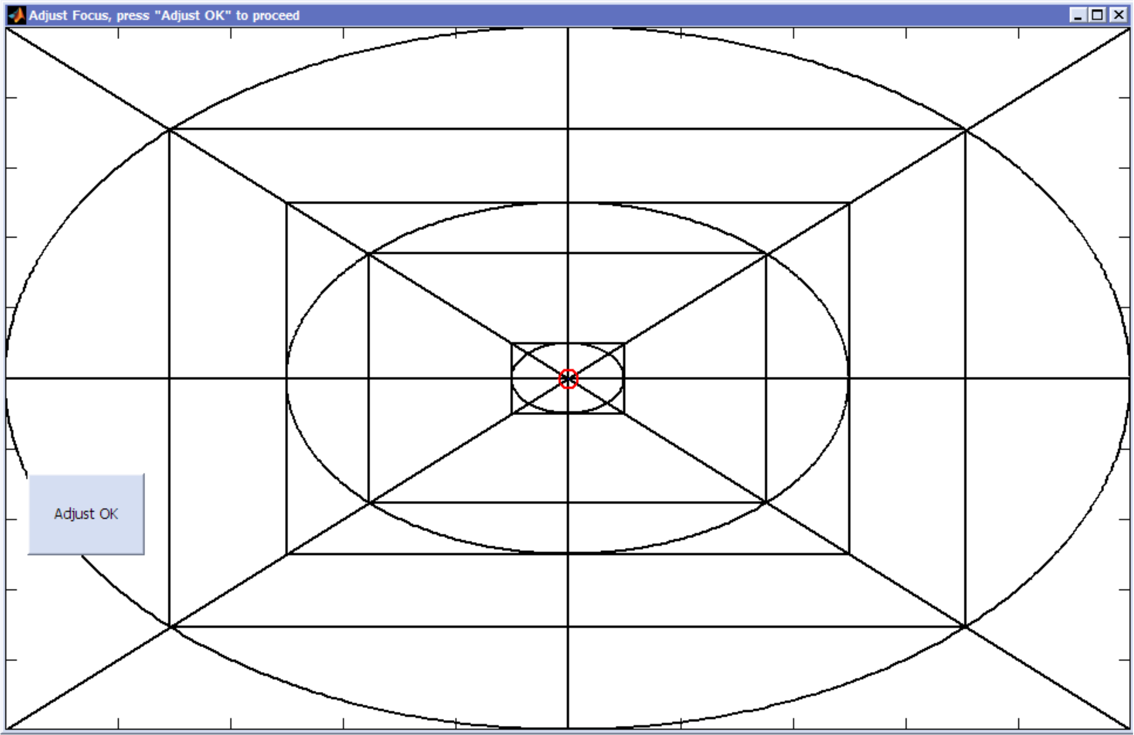
1. You can set all the measurement/calibration parameters in this tab. The details of configuration parameters are described on the "Info" window. After setting all the parameters, please press "OK" button. Then, all the parameters will be saved and "measure", "LUT", "Color Calculator" tabs will be enabled.
2. If you press "Save config" button, the current parameters will be saved under ~/Mcalibrator2/config directory. By pressing "Load config" button, you can load parameters you saved.
3. On the [date] editing window, you can set the data name as you like such as yymmdd_1, yymmdd_2, or yymmdd_display1, as well as yymmdd format. When you need to calibrate a display device twice or more in a day, please change [date] names so that they are not overlapped.
4. When you set a past date on the [date] window, and if there is a data directory already measured previously in ~/Mcalibrator2/data, you can generate LUT by loading the previous data.

Next, you can proceed to measure RGB phosphor luminance values. Please go to "measure" tab.



Measure tab (you can measure CIE1931 xyY here)

You can set up a photometer and measure luminance values on this "measure" tab. First, please press "Adjust Colorimeter Position" button. The grid window as below will be displayed on your screen. You can adjust a position of colorimeter using this window.



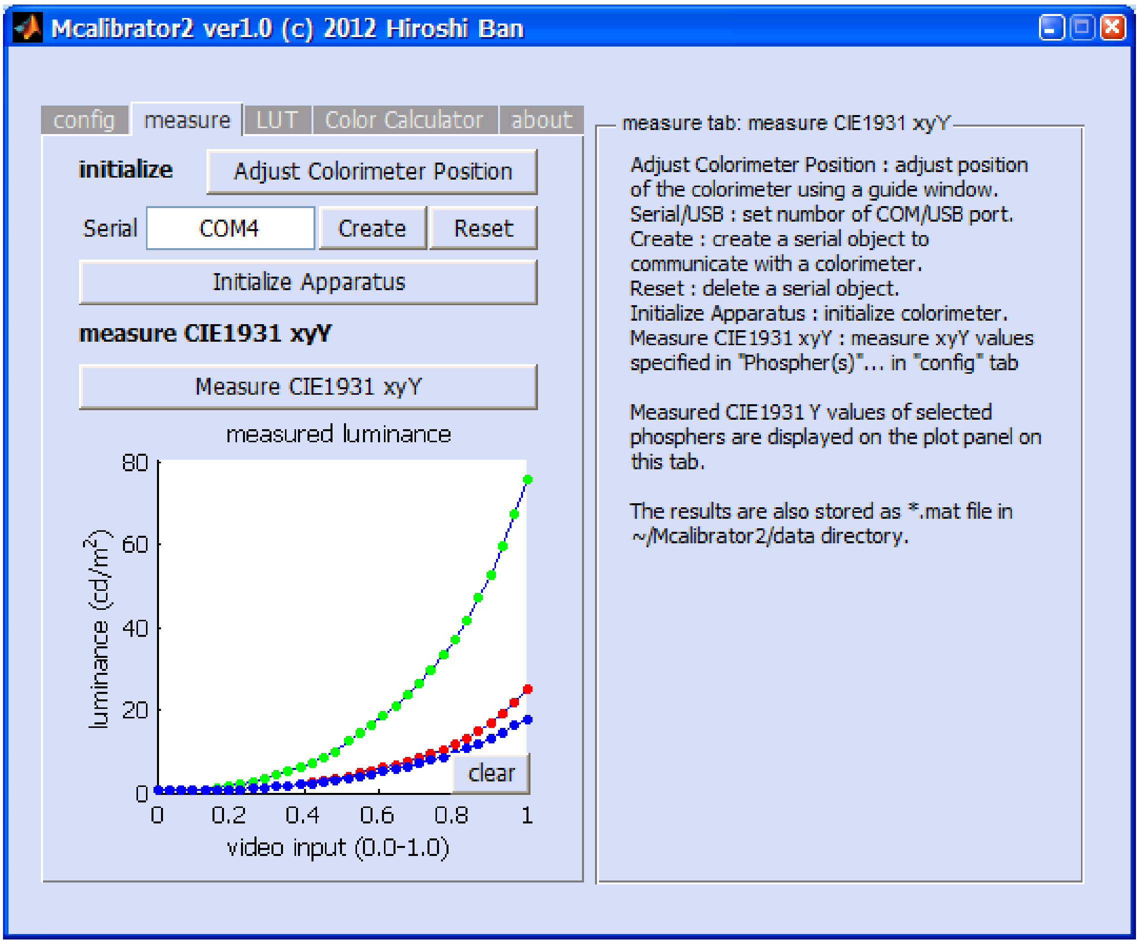
Window to adjust a position of a colorimeter

You can adjust and check a position of a colorimeter anytime when you press "Adjust Colorimeter Position" button on "measure" tab.

Next, specify an ID of a serial (or USB) port to which your colorimeter is connected. On Windows, the ID will be like COM1, COM2, …., on Linux, it will be like /dev/ttyS0, and on Mac, it will be like /dev/tty.KeySerial1 (Please note that authors have mainly tested all the functions on Windows PC. We have not tested this software on Mac or Linux machines). Then, by pressing "Create" button, you can generate a serial (or USB) object to communicate with a colorimeter. Finally, by pressing "Initialize Apparatus" button, the colorimeter will be initialized through the port.

The initial setups finished now. If you press "Measure CIE1931 xyY" button, actual measurement of luminance values against video inputs for each of the RGB phosphors will start.

After finishing the measurements, the relationship between video input and luminance output values will be displayed on the graph as below.

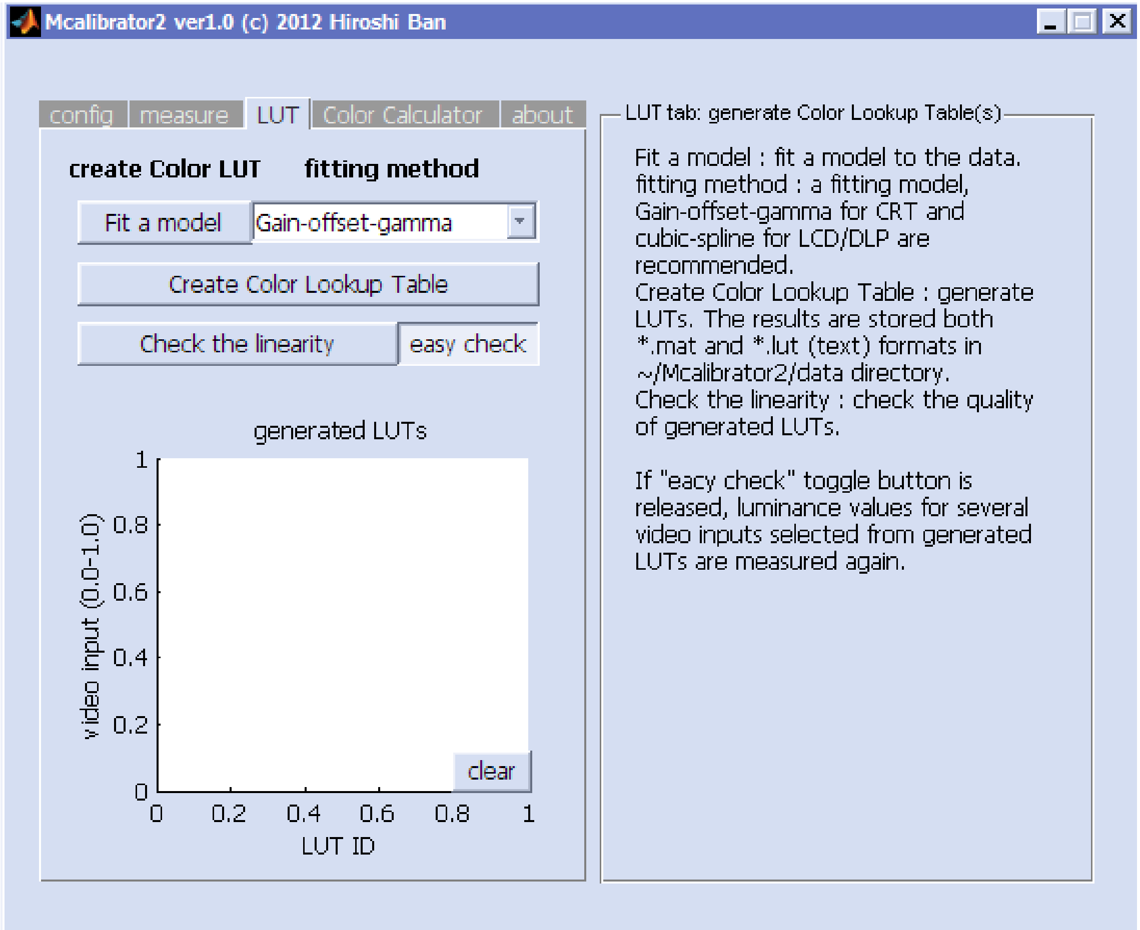


Measure tab after luminance measurement

The whole procedure on "measure" tab is completed now.

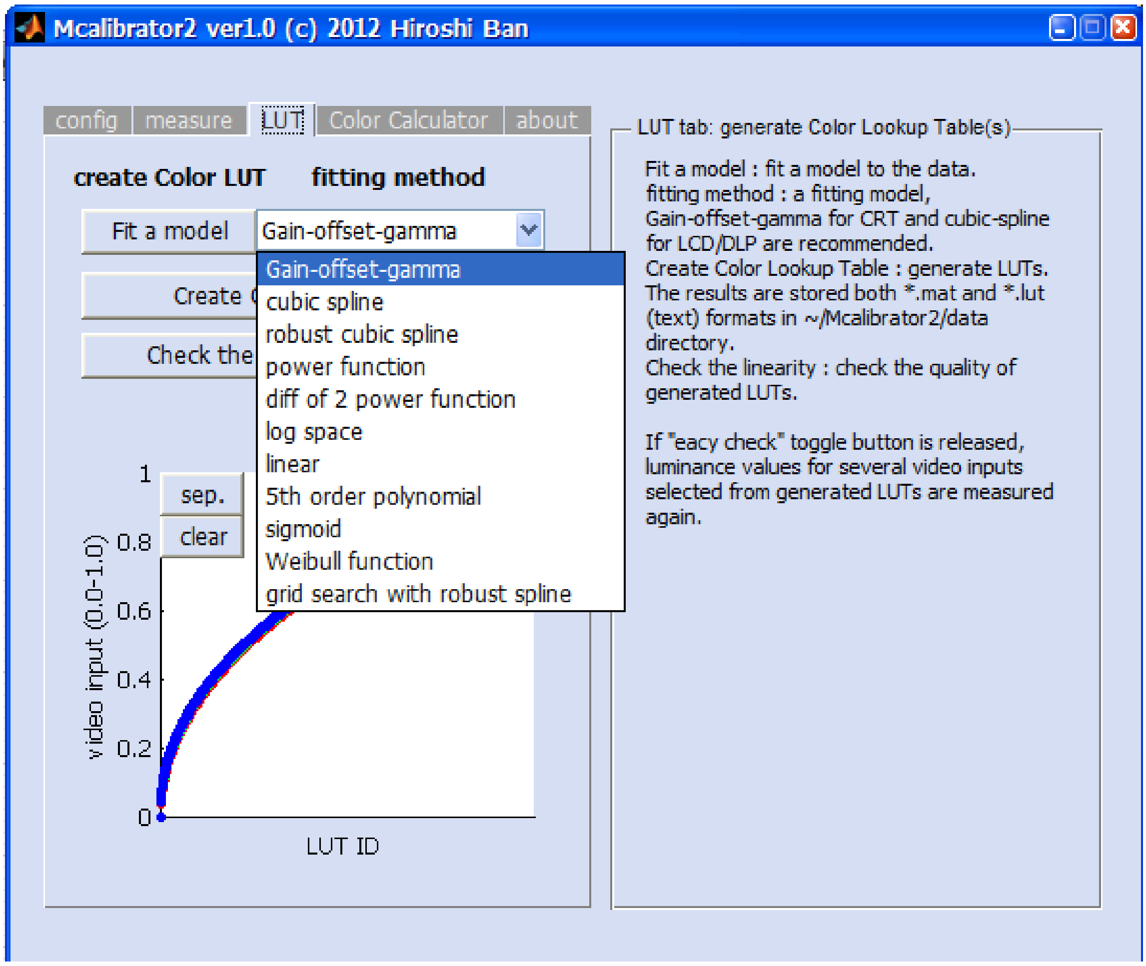
Next, Mcalibrator2 is going to generate Color Lookup Tables (CLUTs) by linearizing the input/output relationship. Please go to "LUT" tab.

Linearization of a relationship between video inputs and luminance outputs (Gamma-correction)



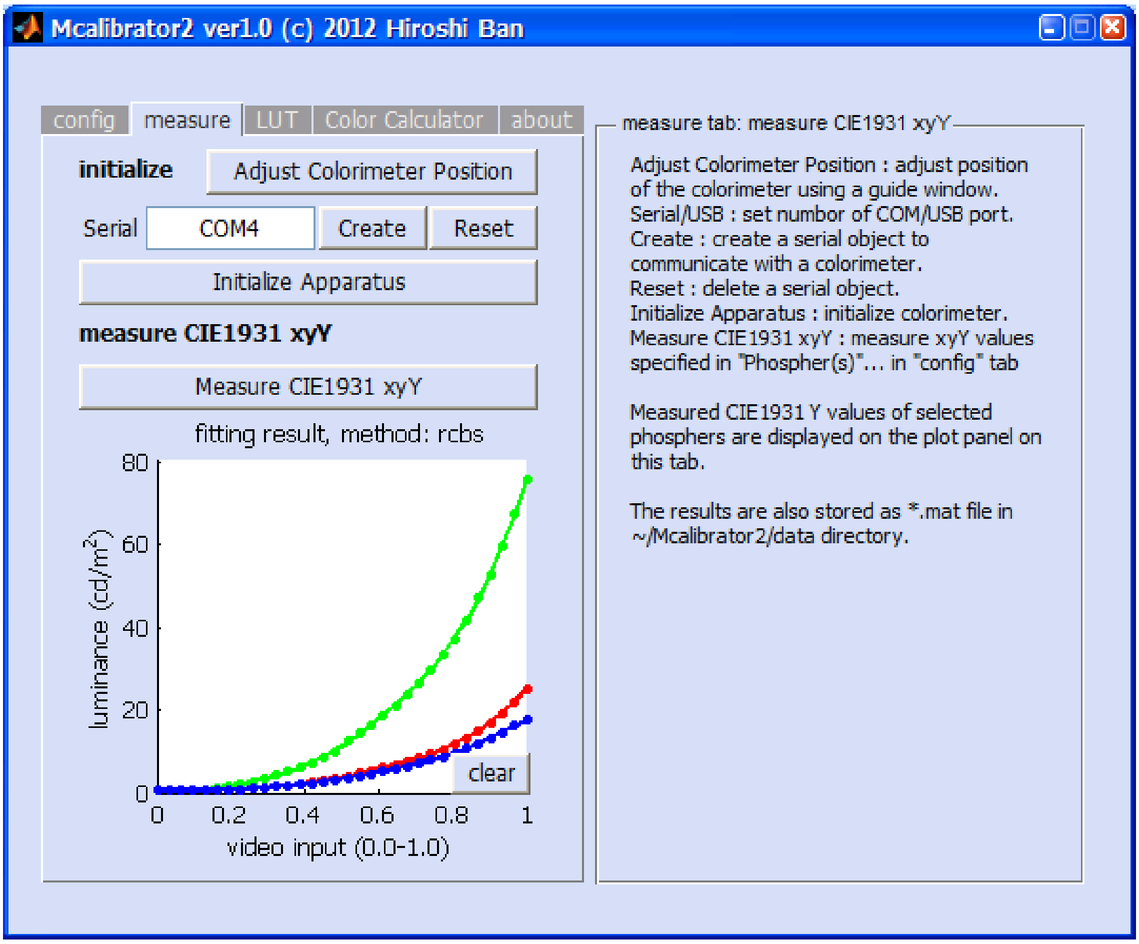
LUT tab (performing gamma-correction and generating CLUTs)

1. Please select a fitting model to describe the input/output relationship. When you are using CRTs, "Gain-Offset-Gamma" (a model used in a standard gamma-correction) is recommended. When you are using a LCD or DLP, "cubic spline" or "robust cubic spline" is recommended. If you select "grid search···", LUT will be generated by directly searching the best values, but it will only work when you measure all 256 points (when you set #LUT step to 256 on a 8-bit phosphor).
2. Please press "Fit a model" button, a function will be fitted to the data and the results will be displayed on "measure" tab.

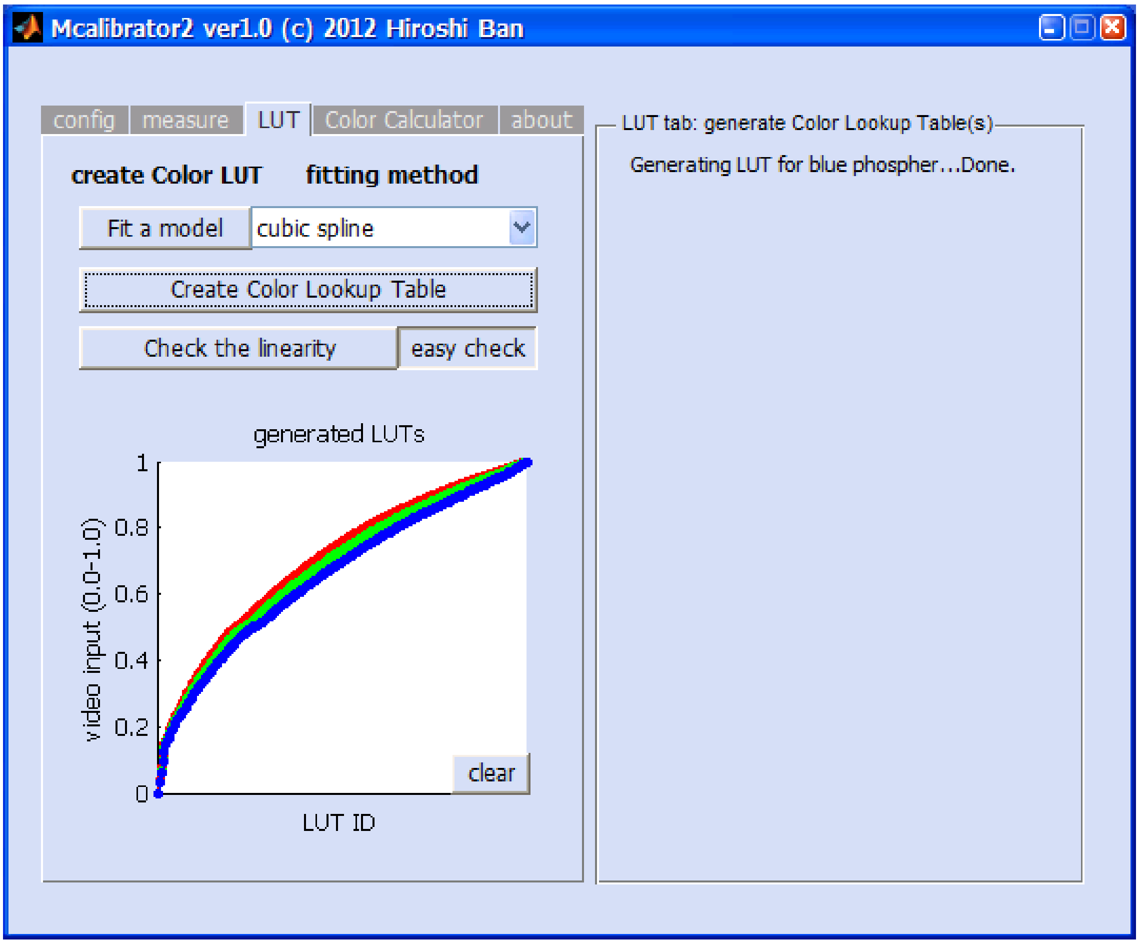


Selection of a fitting method

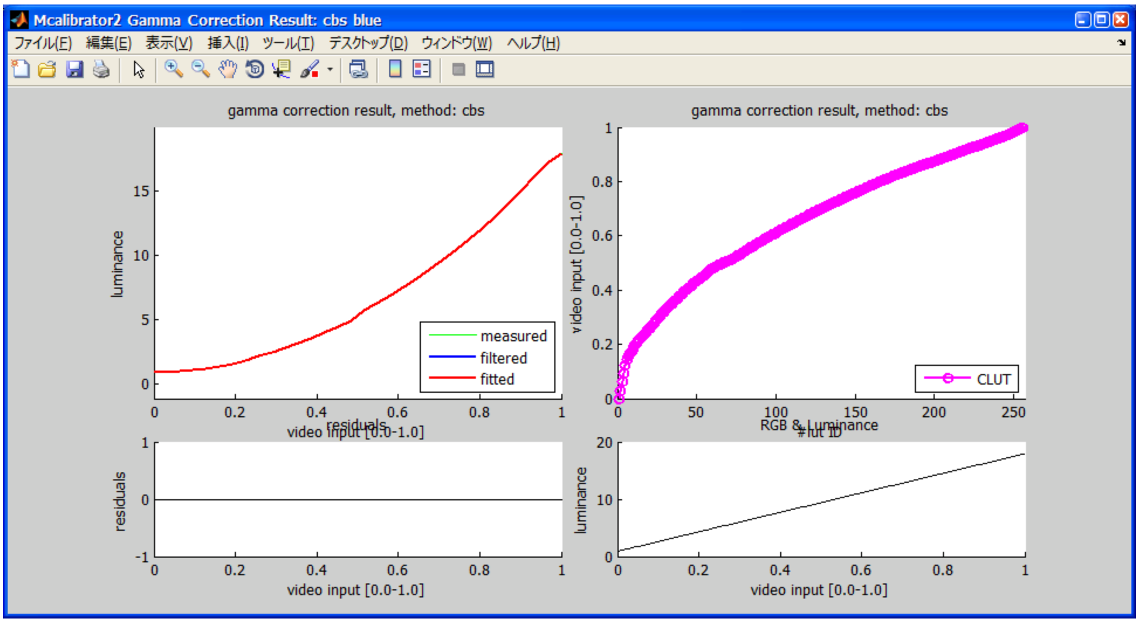
3. If the fitting results are good, please press "Create Color Lookup Table" button. CLUTs will be generated and saved to ~/Mcalibrator2/data/(date etc.).
4. The input/output relationship before/after gamma-correction, LUTs, residuals of the fits etc. will be plotted on the external window. If you are using Windows and Microsoft PowerPoint is installed on your computer, these plots will be saved to a PowerPoint slide file and stored to ~/Mcalibrator2/data/(date etc.) automatically.
5. You can repeat LUT-generation procedure by changing a fitting model until you can get the best results.



Measure tab after modeling the input/output relationship by a cubic spline procedure

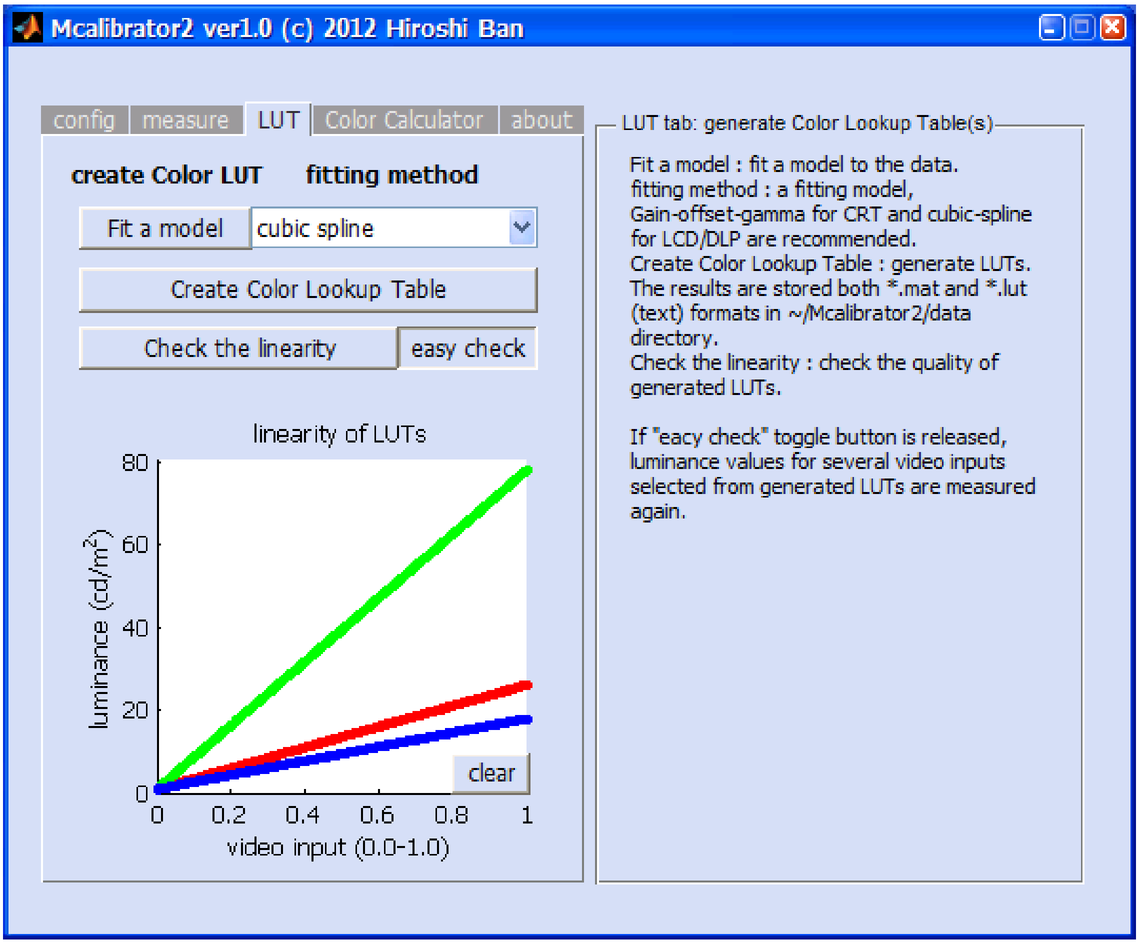


LUT tab after generating CLUTs

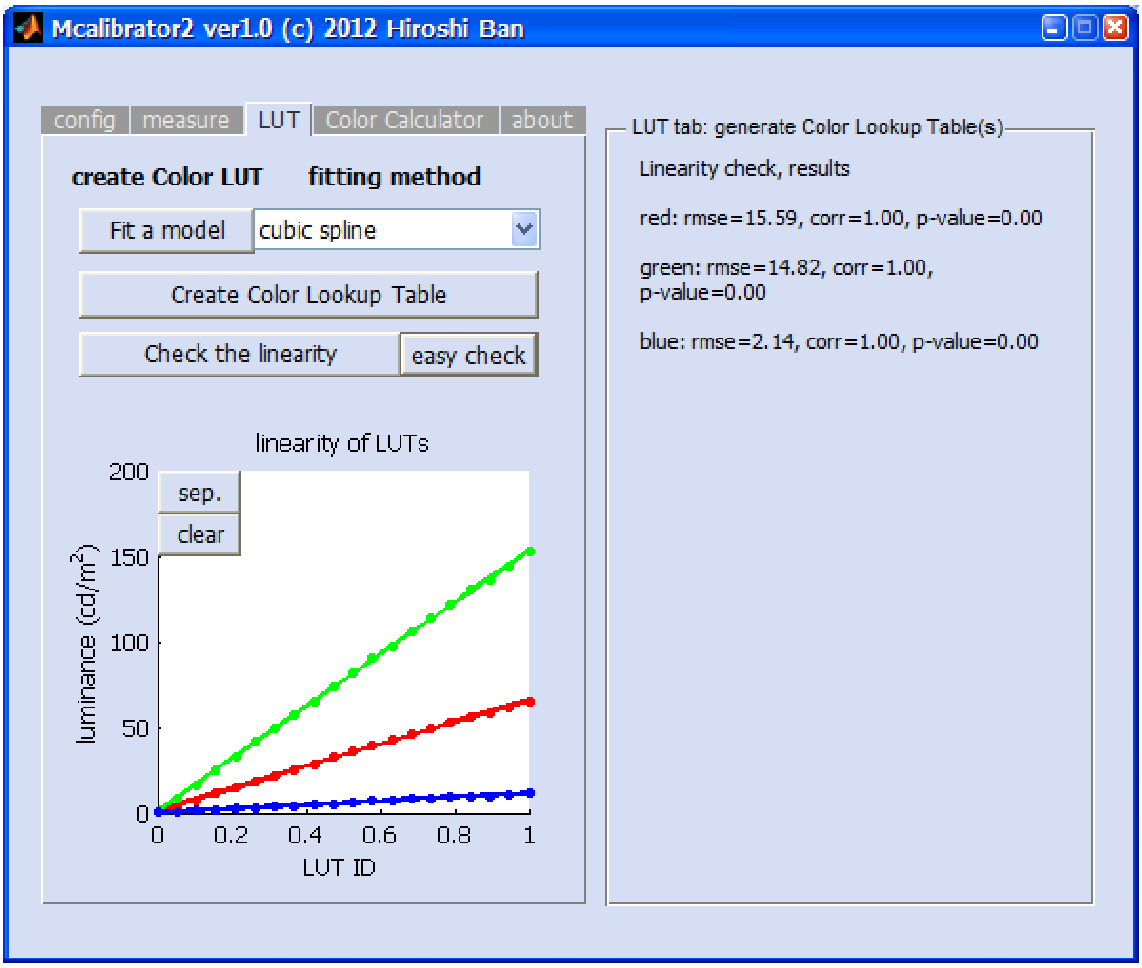


Plots of a raw/corrected input/output relationship, LUTs, and residuals

6. To check how correctly the linearization is performed, please press "Check the linearity" button. When "easy check" toggle button is ON (default), the generated CLUTs will be simply displayed on LUT tab. If it is OFF, luminance values for 20 video inputs (equally sampled from 0.0 to 1.0) from generated CLUTs will be re-measured, and the linearization verification test will be conducted. If the results are not good, please re-generate CLUTs by changing a fitting model.



Verification of linearity of a relationship between video inputs and luminance outputs (Easy Check Mode)

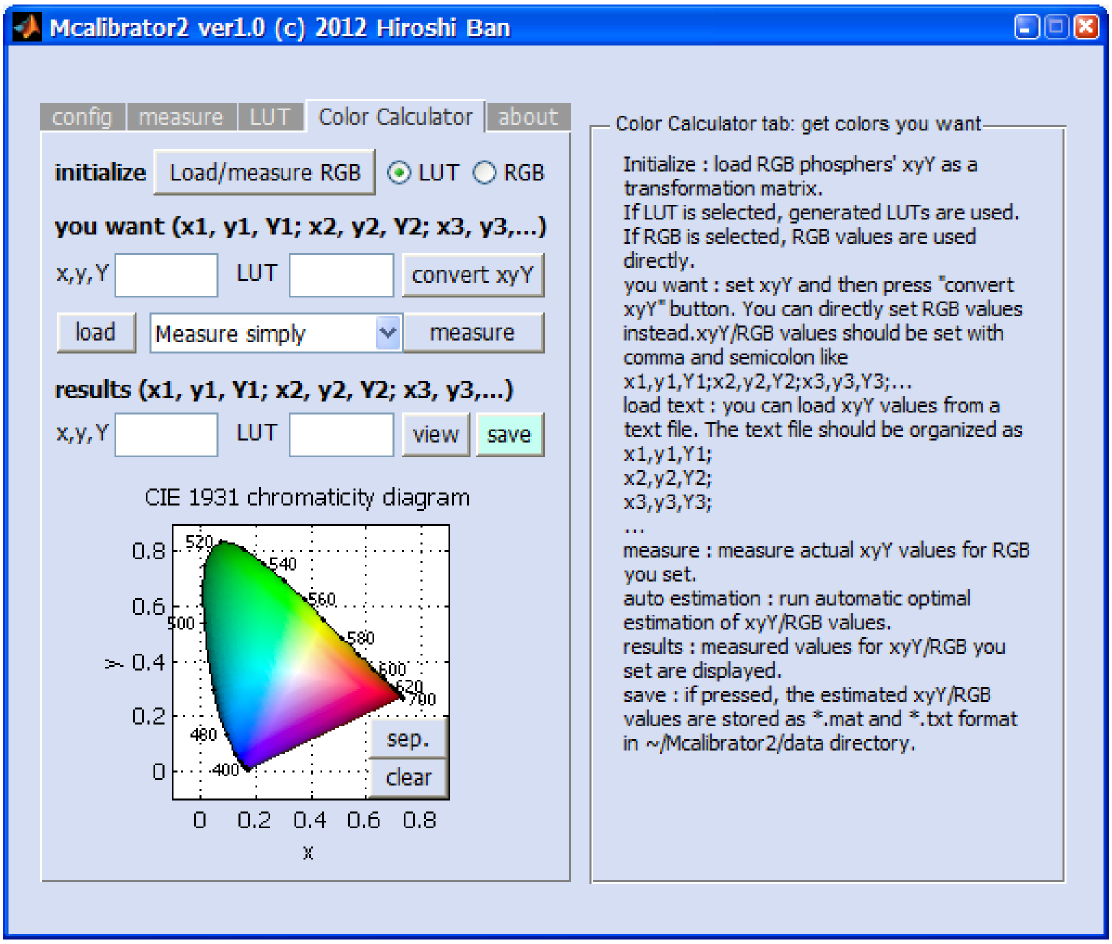


Verification of linearity of a relationship between video inputs and luminance outputs (Re-measuring Mode)

The whole procedures of gamma-correction and CLUT generation are completed. If you do not need to get optimal RBG video input values for required chromaticities (CIE1931 xyY, not only Y (luminance)), the display characterization finishes now.

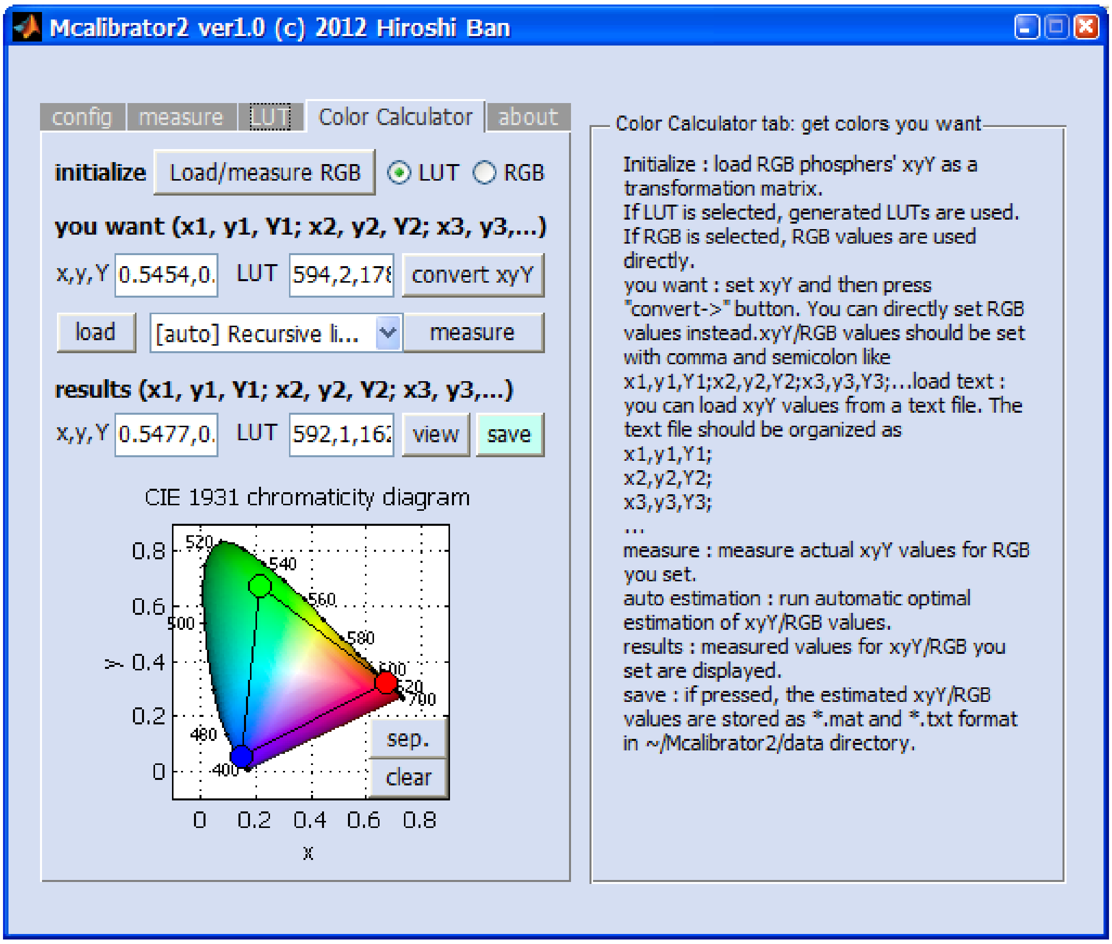
If you need to perform chromaticity calibration, please go to “Color Calculator” tab.

Chromaticity calibration procedure



Color Calculator tab (perform chromaticity calibration)

1. First, to estimate chromaticity by a simple linear transformation, please load CIE1931 xyY values for each of the RGB phosphors at their maximum levels. To do this, please press "Load/measure RGB" button. The results will be displayed on CIE1931 chromaticity diagram as 3 circles and a triangle defined by these circles as its vertices. The range of chromaticity values you can display is a region enclosed by this triangle.



Color Calculator tab after loading CIE1931 xy values of the RGB phosphors at their maximum levels

2. Then, if you want to use the CLUTs generated in the previous procedures, please set LUT radio-button ON (default). If you want to use the raw RGB values without gamma-correction, please set RGB radio-button ON. If the RGB radio-button is ON, you can directly set RGB values you want to measure. Please use this option when you want to measure RGB values you used in your stimuli etc.
3. Next, set CIE1931 xyY values you want to display in your experiment on "x,y,Y" editing window just below the "you want (x1, y1, Y1; x2, y2, Y2; x3, y3,..)" message. Please describe 3 (x, y, Y) values with a comma as a delimiter. If you want to measure multiple values, please use a semi-colon to separate each CIE 1931 xyY value. The format is thus as below.

   x1,y1,Y1; x2,y2,Y2; x3,y3,Y3;…
4. You can also load these CIE1931 xyY values from a text file. To do this, please press

"load" button. CIE1931 xyY values in the text file should be described with commas and semi-colons as delimiters as below.

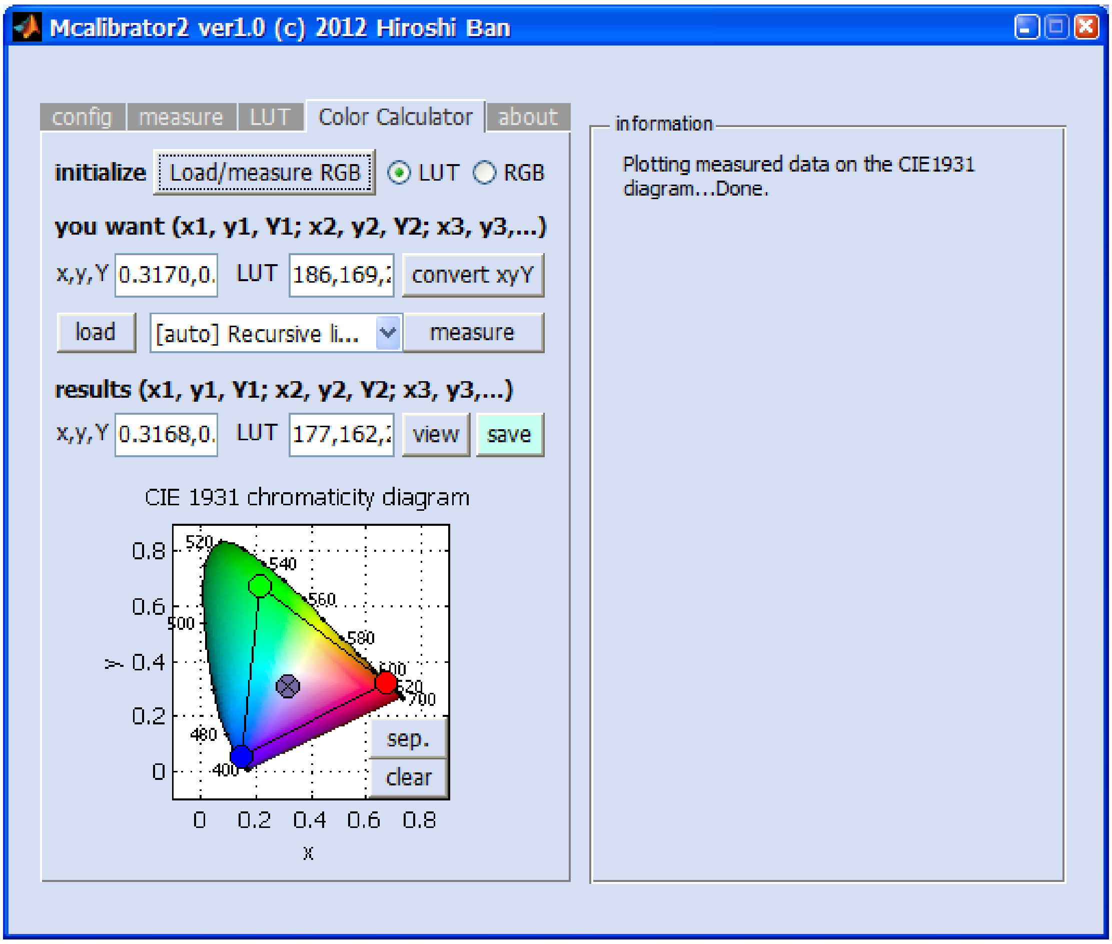
x1, y1, Y1;

x2, y2, Y2;

x3, y3, Y3;

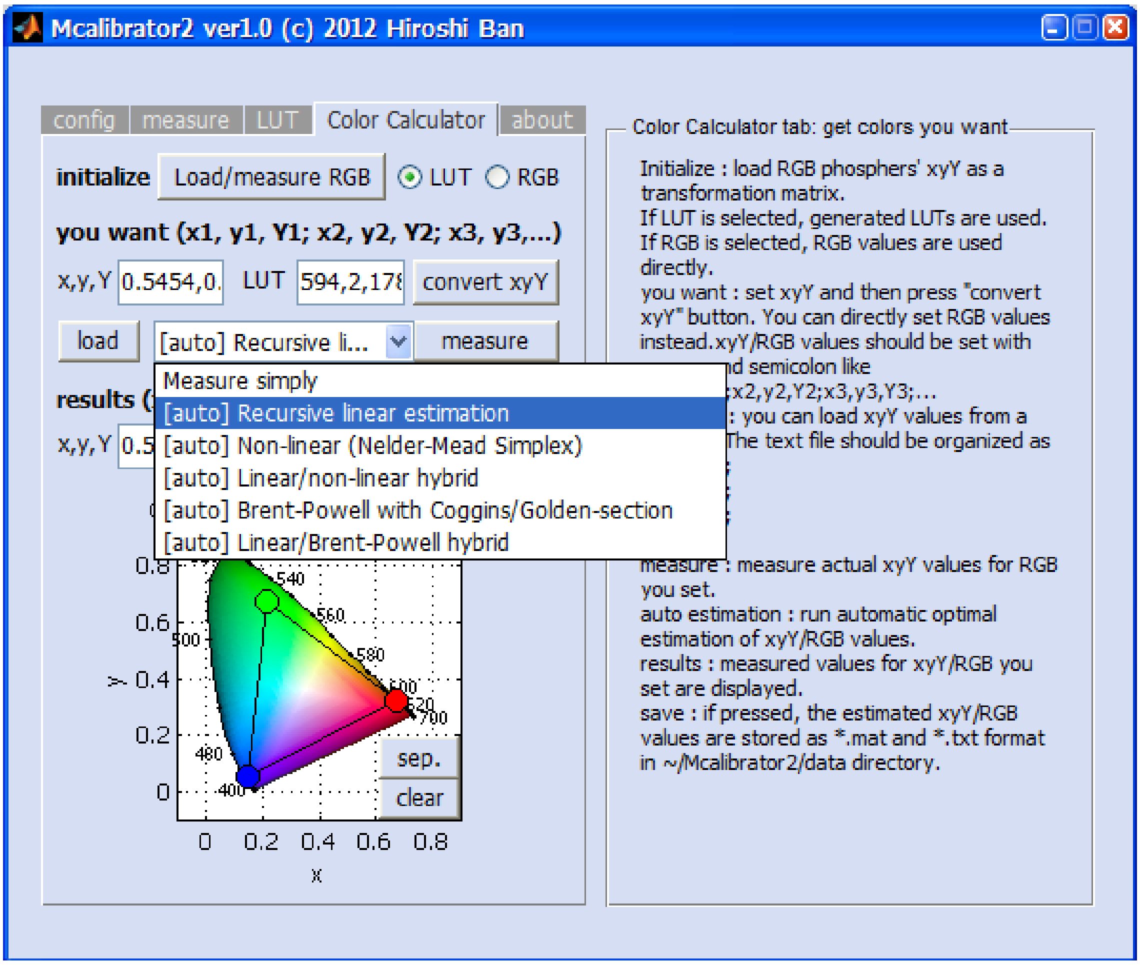
…

5. After setting xyY values you want, please press "convert xyY" button. If the LUT radio-button is ON, the xyY values are converted to CLUT IDs. If the RGB radio-button is ON, the raw RGB values required are calculated by a simple linear transformation. The results are presented on "LUT (or RGB)" editing window on the right side.
6. To measure actual CIE1931 xyY values for these initial estimations, please press "measure" button. The chromaticities for the estimated RGB values are measured. The results are presented on "x,y,Y" and "LUT(or RGB)" editing windows just below the "results (x1, y1, Y1; x2, y2, Y2; x3, y3,..)" message. In addition, the CIE1931 xy values you want and the actual measured values for the estimations are plotted on "CIE1931 chromaticity diagram" by a circle and a cross. If these two values are close enough, the chromaticity calibration finishes successfully.

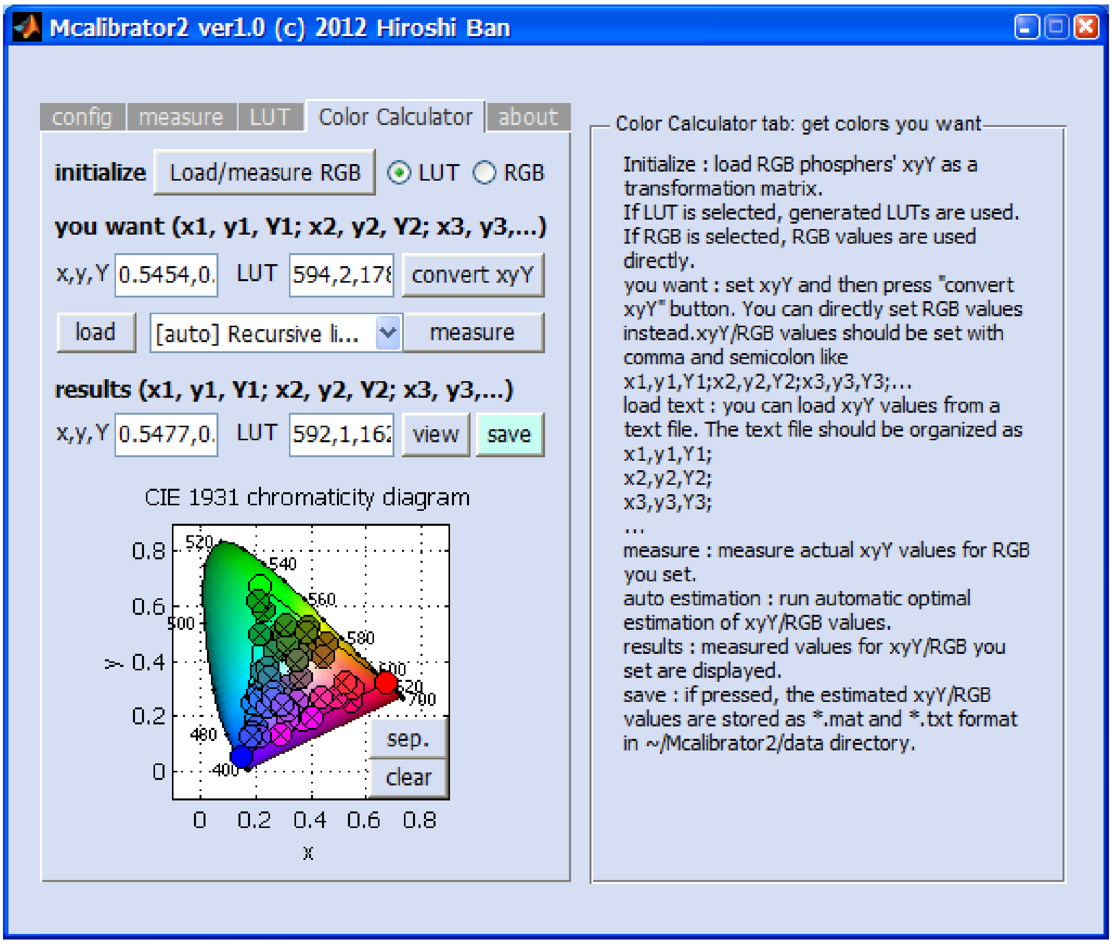


Plots of a chromaticity you want (O) and actual measurement from estimations (X)

7. If the estimations are not good, you need to correct LUT IDs or RGB values by an automatic chromaticity estimation/correction method. Mcalibrator2 currently has 5 automatic chromaticity estimation methods; recursive linear estimation based on a least-squares method, a non-linear direct search algorithm, a line search algorithm and their hybrid methods. For more details of the estimation algorithms, please see comments on the source codes of Mcalibrator2 and see the paper on this software (Ban and Yamamoto, 2013, Journal of Vision). You can select one of them by using a dropdown-box next to "measure" button. The estimation for a chromaticity will take 1-5 min. After finishing the estimations, the results are presented on "x,y,Y" and "LUT(or RGB)" editing windows just below the "results (x1, y1, Y1; x2, y2, Y2; x3, y3,..)" message. In addition, the CIE1931 xy values you want and the actual measured values for the estimations are plotted on "CIE1931 chromaticity diagram" by a circle and a cross.



Dropdown-list for selecting a chromaticity estimation method

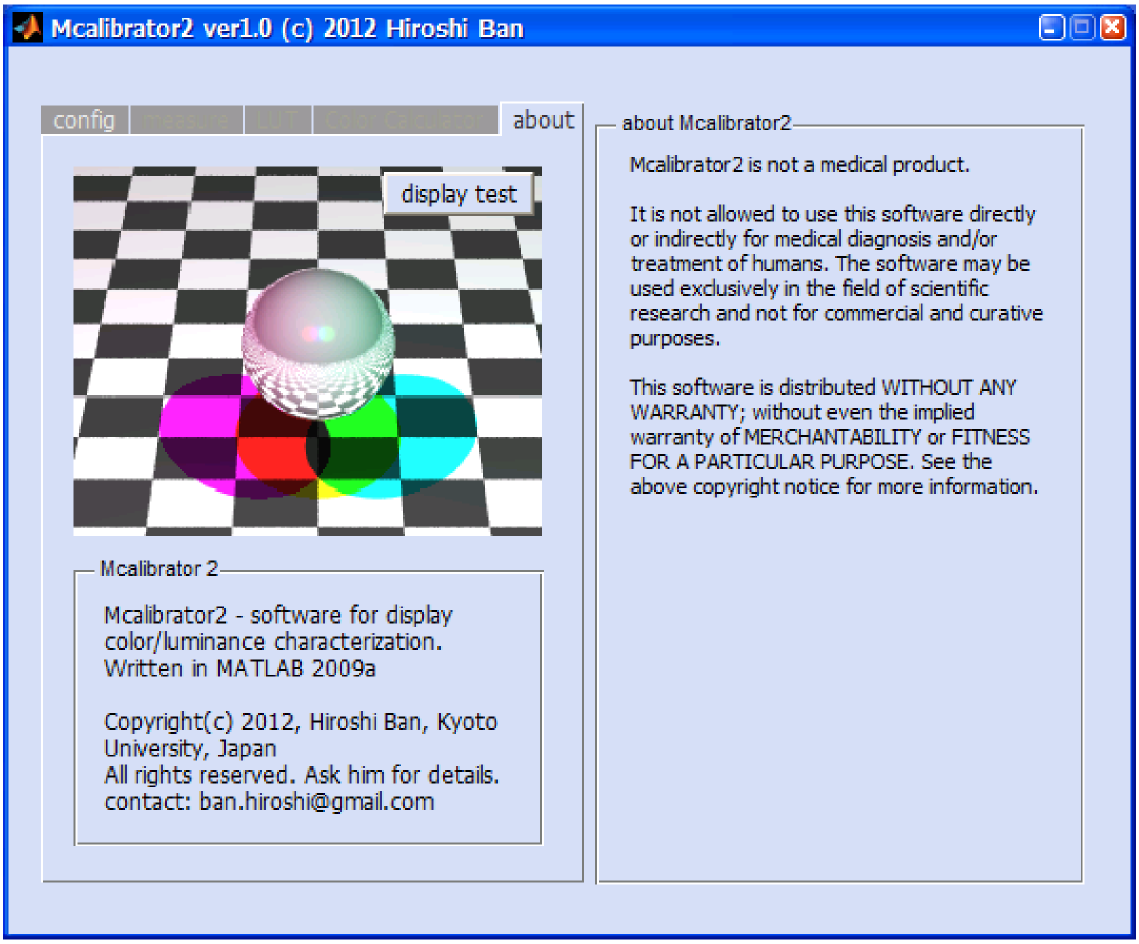


Results of automatic chromaticity calibration (50 chromaticities)

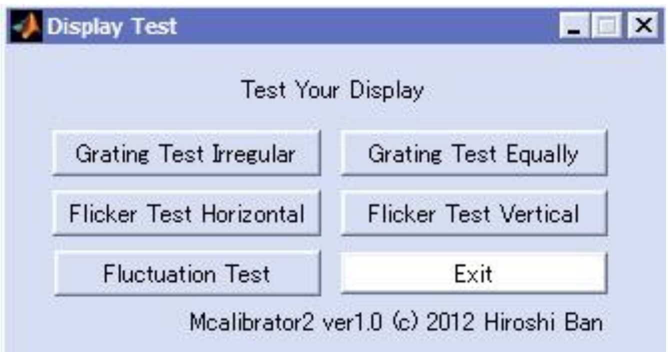
8. To view the chromaticity estimation results on MATLAB window, please press “view” button. The estimation results, residuals etc. are displayed on MATLAB command window.
9. To save the estimation results, please press “save” button. The results are saved as a *.mat file to ~/Mcalibrator2/data/(date etc.), with a file name like, “estimate_files_001.txt”, “estimate_files_002.txt” ,… (a prefix like _001 is added automatically). Please be careful for that no result will be saved automatically without pressing “save” button.

If you can reproduce your chromaticities accurately, all the calibration procedures are completed. If the results are not good, please retry the estimation using different methods.

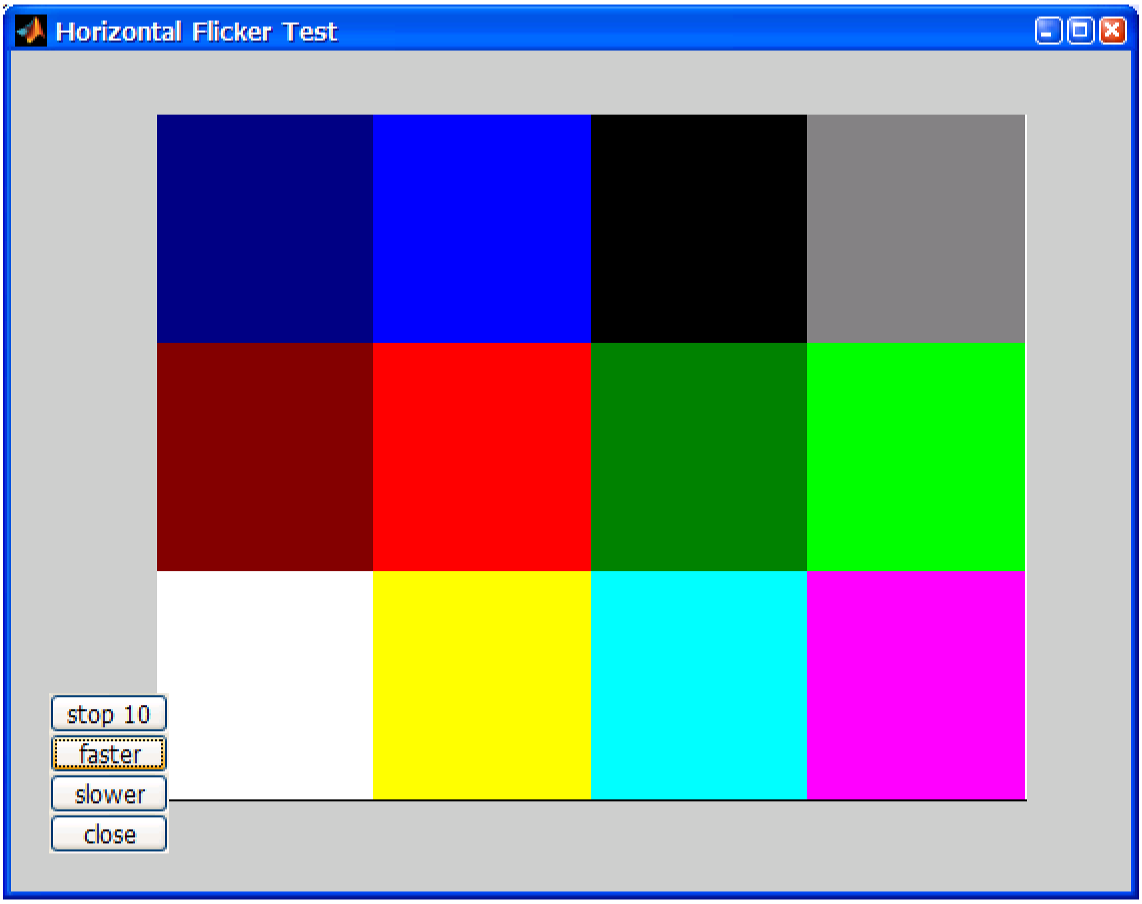
On “About” tab, information on Mcalibrator2 software is displayed. If you press “display test” button on this tab, you can perform several simple display tests.



About tab (information on Mcalibrator2 and performing display tests)



Window for selecting simple display tests



Color-flickering test window

How to communicate with your own colorimeter through Mcalibrator2

Currently, Konica-Minolta CS-100A (tested), Admesy Brontes-LL (tested), Photo-research PR-650 (tested), and Cambridge Research Systems ColorCal2 (not tested) can be used in Mcalibrator2. You can also use your own colorimeter(s) with Mcalibrator2 if you create some simple communication codes written in MATLAB. To do this, you need to

1. Register your own colorimeter(s) to a apparatus-list file
2. Create a file to define a MATLAB object to communicate with your own colorimeter.

To register your colorimeter, please edit
~/Mcalibrator2/subfunctions/colorimeter/colorimeter_list.m
By default, 5 photometers are registered as below in this file.

```
colorimeters{1}={'Photo Research PR-650','pr650',0};
```

colorimeters{2}={'KONICA-MINOLTA CS-100A','cs100a',0};
colorimeters{3}={'Admesy Brontes-LL','brontesLL',1};
colorimeters{4}={'Cambridge Research Systems ColorCal','colorcal',1};
colorimeters{5}={'Cambridge Research Systems ColorCal2','colorcal2',1};

So, please add your colorimeter(s) as colorimeters{6}, colorimeters{7},… with a MATLAB cell format. The first variable is a name of your apparatus displayed on Mcalibrator2, the second variable is a name of a function you create to communicate with this apparatus through MATLAB (= a class file name to generate MATLAB object), and the third variable is a value (0/1) to specify how to communicate with this apparatus (0=serial connection, 1=USB connection).

Then, please generate a MATLAB class file (*.m) to communicate with your apparatus. Specifically, please define a MATLAB object with a "classdef" keyword. The object should have 2 properties and 4 methods described below.

・Property (Hidden)
  portname、rscom
・Property (public)
  init_flg
・Method
  gen_port、reset_port、initialize、measure

Please use the same input/output variables for these methods with the other object. As examples, please see an object to communicate with Konica-Minolta CS-100A
~/Mcalibrator2/subfunctions/colorimeter/cs100a.m (serial communication)
~/Mcalibrator2/subfunctions/colorimeter/brontesLL.m (USB communication)

After finishing these setups, please launch Mcalibrator2 and your own colorimeter will be registered: your apparatus will be listed in a dropdown-list on "config" tab and you can use the apparatus with Mcalibrator2.

How to add your own chromaticity estimation methods to Mcalibrator2

You can easily add your own chromaticity estimation procedure(s) as you like. To do this, you need to

1. Register your own estimation methods to a estimation-method file.
2. Create MATLAB functions which describe your own chromaticity estimation procedure

To register your method(s) to Mcalibrator2, please edit
~/Mcalibrator2/subfunctions/colorcalculator/ measurement_method_list.m
By default, the 8 chromaticity estimation methods are registered as below.

```
meas_methods{1}={'Measure simply','calculator_measure_simply'};
meas_methods{2}={'[auto] Recursive linear estimation with unifrnd()','calculator_auto_estimation_linear'};
meas_methods{3}={'[auto] Recursive linear estimation with Grid','calculator_auto_estimation_linear_grid'};
meas_methods{4}={'[auto] Adjust residuals by linear coefficients','calculator_auto_estimation_lincoeff'};
meas_methods{5}={'[auto] Non-linear (Nelder-Mead Simplex)','calculator_auto_estimation_nonlinear'};
meas_methods{6}={'[auto] Linear/non-linear hybrid','calculator_auto_estimation_hybrid'};
meas_methods{7}={'[auto] Brent-Powell with Coggins/Golden-section','calculator_auto_estimation_powell'};
meas_methods{8}={'[auto] Linear/Brent-Powell hybrid','calculator_auto_estimation_powell_hybrid'};
```

So, please add your method(s) as meas_methods{9}, meas_methods{10},… with a MATLAB cell format. The first variable is a name of your method displayed on Mcalibrator2, the second variable is a name of a function in which your chromaticity estimation method is described.

Then, please create a MATLAB function which defines your own chromaticity estimation procedure. The function should have 3 input variables below and have no output variable.

hObject : A handle to Mcalibrator2 Objects. Generally this variable is not used, but it is required to follow MATLAB GUI handling conventions.

eventdata : A variable booked for a future version of MATLAB. This variable is also not used currently, but it is required to adjust a format with the other default functions of Mcalibrator2.

handles : A structure which includes handles to child objects of Mcalibrator2 and user-defined data

In addition, please declare the global variables below in your function with a "global" keyword.

```
global config;                    % a structure which stores configuration of Mcalibrator2
global colorimeterhandler; % an object to handle a colorimeter
global displayhandler;       % an object to present a chromaticity on display
global phosphers;              % CIE1931 xyY values of RGB phosphors
                                          at their maximum levels
global flares;                    % Flares (zero-level luminance) of RGB phosphers
```

Using these global variables and an input variable, "handles", you can manipulate Mcalibrator2 through your own chromaticity estimation function.

As an example, please see
~/Mcalibrator2/subfunctions/colorcalculator/calculator_auto_estimation_linear.m
~/Mcalibrator2/subfunctions/colorcalculator/calculator_auto_estimation_hybrid.m

(End of this file)