

***CIELAB* COLOR MEASUREMENT FROM DIGITAL IMAGES**

SANDRO M. GOÑI – VIVIANA O. SALVADORI

APRIL 2015

Version 1 - EN - Revision 1

CIDCA, CCT-CONICET La Plata - Univ. Nacional de La Plata. 47 y 116, CP: 1900, La Plata, Argentina.

Facultad de Ingeniería, Univ. Nacional de La Plata. 1 y 47, CP: 1900, La Plata, Argentina.

INDEX

1. GENERALITIES.....	2
Before using the software.....	2
2. CONVERSION MODELS.....	3
3. CALIBRATION - TESTING GUI.....	3
3.1. Patches selection	3
3.2. <i>RGB</i> TO <i>CIELAB</i> CONVERSION.....	4
3.2.1 <i>New calibration mode</i>	5
3.2.2 <i>Testing mode</i>	5
4. <i>RGB</i> TO <i>CIELAB</i> CONVERSION <i>GUI</i>.....	5
4.1 READ THE IMAGE	6
4.2 SELECT THE REGION OF INTEREST (<i>ROI</i>)	6
4.3 CONVERT <i>RGB</i> VALUES.....	8
5. <i>L*A*B*</i> TO <i>RGB</i> CONVERSION <i>GUI</i>.....	10
6. REFERENCES.....	10

1. GENERALITIES

A computer vision system (CVS) that performs color measurement is formed by three main components:

1. An image acquisition chamber and a lighting system
2. A digital camera
3. Software that processes the obtained information (i.e. images)

This section details the software that processes the information. New potential users must first address items 1 and 2. The construction of an image acquisition chamber is easy and cheap, and the market provides a large number of digital cameras with different features and costs.

In order to process the images, several conversions between *RGB* and *CIELAB* color spaces were implemented in MATLAB® (The Mathworks Inc., Natick, Mass., USA), version 7.0. The software was developed as a collection of Graphic User Interfaces (*GUIs*), friendly and easy to use.

Before using the software

1. Unzip the *GUIs* files in the desired PC directory (for example, inside \MATLAB\...).
2. Add the unzipped folder to the MATLAB search path: **File/Set path/Add with Subfolders...**; select the unzipped folder and press SAVE and CLOSE. So, the GUI will be visible independently of the MATLAB working directory.

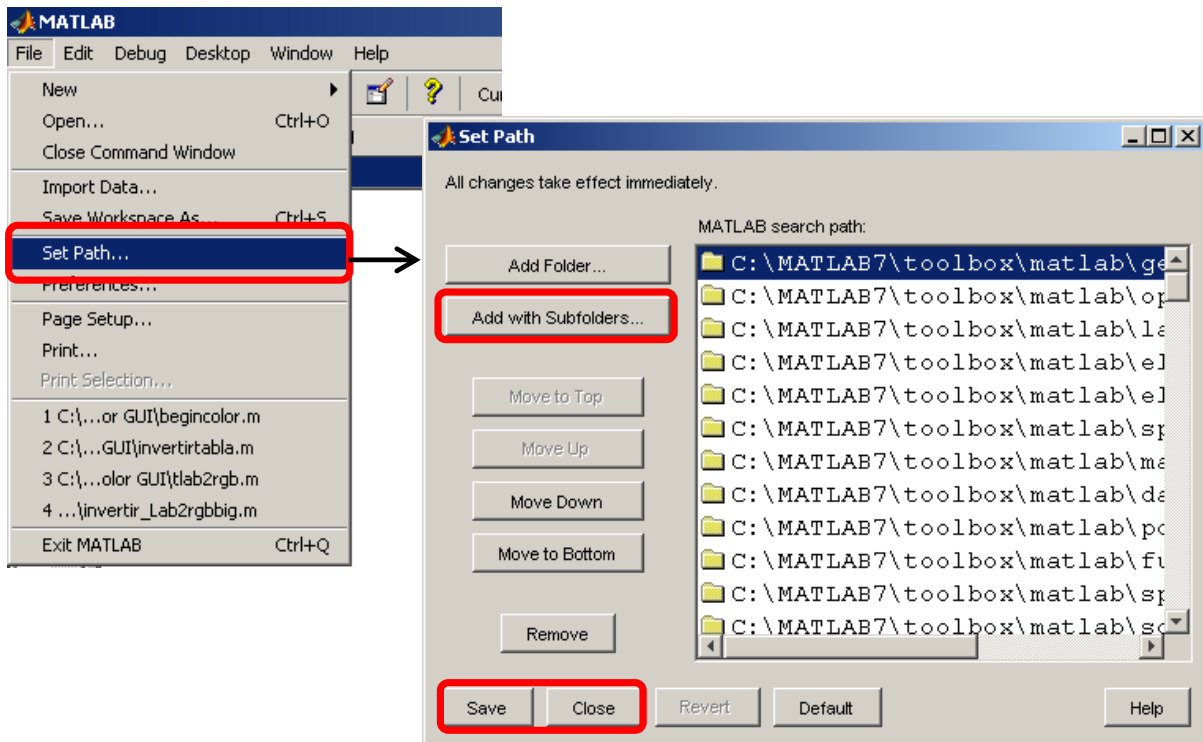


Figure 1. Adding the *GUIs* to MATLAB search path.

3. Finally in the MATLAB command window execute the main start *GUI* (Fig. 2):

```
>> startcolor
```

which will show the primary interface, with three elements for interacting with it:

- Calibration-testing *GUI*,
- *RGB* to *L*a*b** conversion *GUI*,
- *L*a*b** to *RGB* conversion *GUI*.

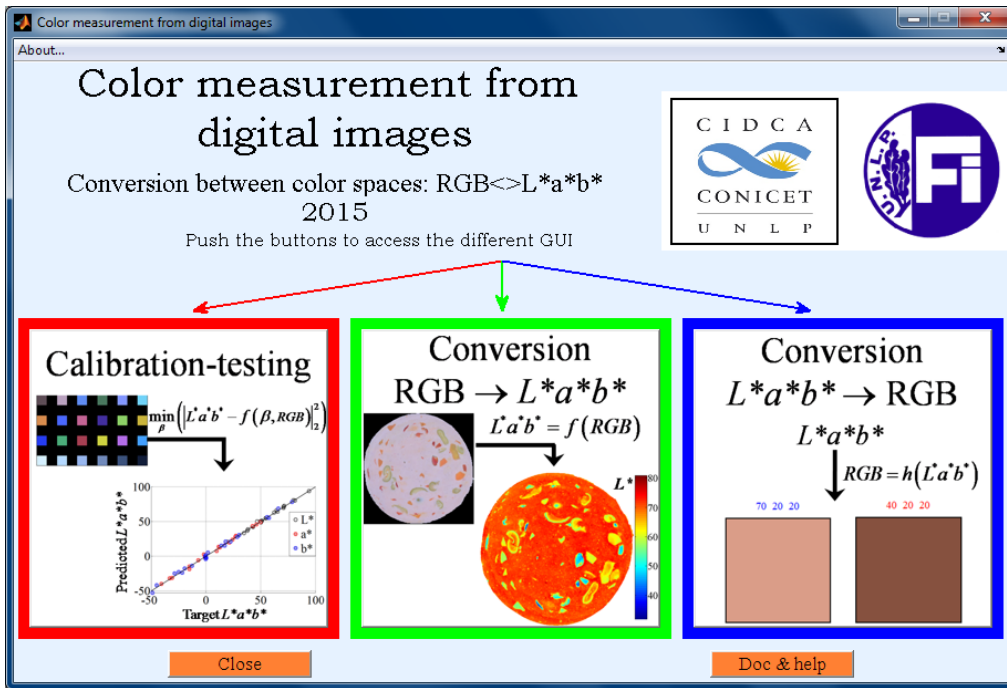


Figure 2. Main GUI.

- It is recommended not to push any action button until the task of the previous button has ended.
- The GUIs requires MATLAB® Image Processing Toolbox™ (The Mathworks Inc., Natick, Mass., USA).
- This version of the software is provided like pre-parsed files (*.p files).

2. CONVERSION MODELS

The software implements the conversion from RGB to L*a*b* space color using one of five different mathematical models, which is selected from a popup menu:

- Direct. CIE 1976 L*a*b* color space, which is a theoretical conversion model (IEC 61966-2-1, 1999; Gonzalez and Woods, 2002).
- Direct to fit. This is an empirical adaptation of the direct model.
- Linear. Empirical polynomial model.
- Quadratic. Empirical polynomial model.
- Quadratic plus interactions. Empirical polynomial model.

For further details about the implemented conversion models the reader is referred to Goñi and Salvadori (2016).

3. CALIBRATION - TESTING GUI

Calibration is performed using the classic rendition chart ColorChecker (X-Rite, USA), which has 24 patches of different colors. By default, D65 illuminant is used.

3.1. Patches selection

The GUI for calibration - testing of the system (Fig. 3) allows reading an image of the color chart, selecting the patches and converting their color values from RGB to CIELAB.

The GUI was designed to work with a properly aligned chart (i.e. aligned to Cartesian coordinates).

1. If the chart is not aligned, it must be rotated using the slider bar, and the new position must be fixed using the “Done? Set rotation” button.
2. Then, push the “Define patch centers” button, click with the left mouse button on the centers of the left-top patch (brown) and the right-bottom patch (black). The other patches will be automatically selected and calculated.

3. A dialog box appears to modify the mask size. The mask shape can be a square or a circle, and the size refers to the side length or diameter, respectively; the mask edges are shown in red over the original image (Fig. 3).

The mask size must be representative in order to have many pixels for each patch, but not too big to avoid causing mask overlapping. The axis scale can be useful in order to select the mask size. The average and standard deviation of *RGB* values in each patch are informed in the MATLAB command window.

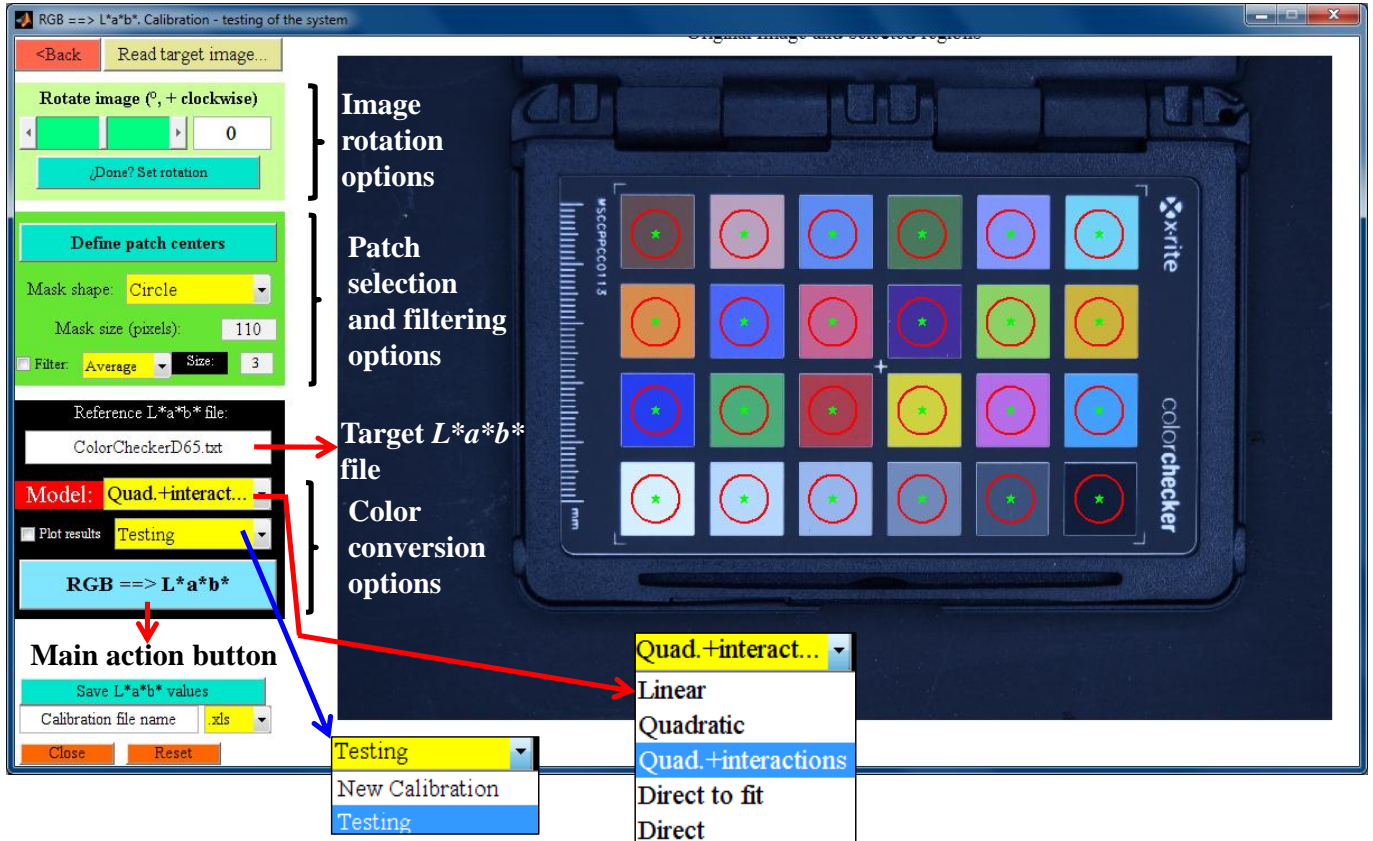


Figure 3. Calibration - testing of the system GUI. Patches are selected with a circle mask, mask edges are shown in red.

3.2. RGB TO CIELAB CONVERSION

The "*RGB => L*a*b**" button applies the selected conversion model to the average *RGB* values of each patch (Fig. 3). From a popup menu, the user can select two operation modes: "New calibration" or "Testing".

The target *L*a*b** values of each patch (Pascale, 2006) are saved in the "ColorCheckerD65.txt" file (for D65 illuminant). This is the default calibration file and its name appears on an editable field of the GUI (Fig. 3). If the user wishes to employ his own target data, a new *.txt file must be defined; this file must be written from top to bottom and from left to right, starting from the brown patch; 1° column is *L**, 2° is *a**, and 3° is *b** (Fig. 4).

When using the direct model, only D65 illuminant is allowed. When using the empirical models, other illuminant can be used, provided the corresponding reference file is available, in this case the calibration process will fit the parameters of the empirical model.

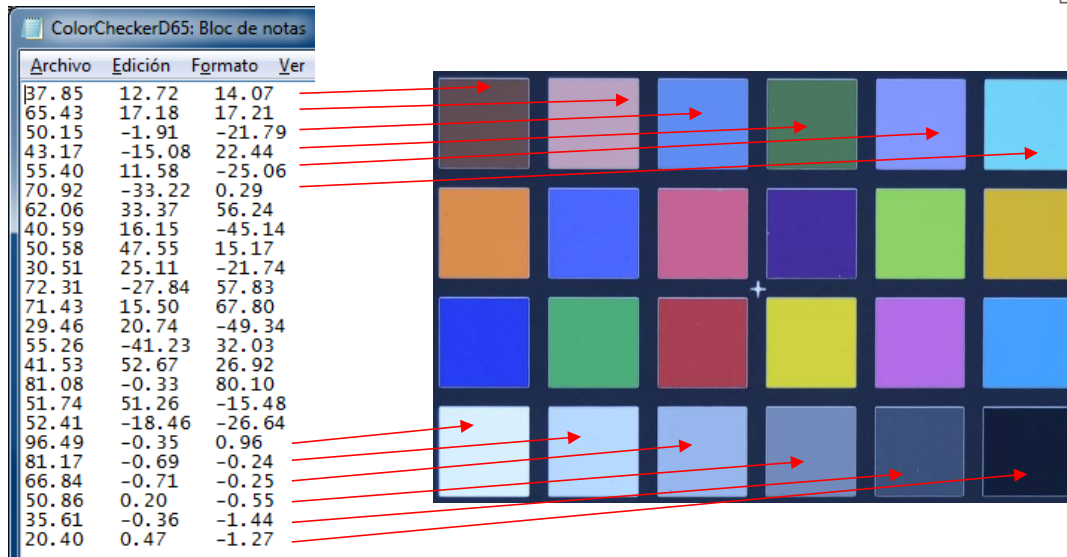


Figure 4. Details of the calibration reference file.

3.2.1 New calibration mode

In this mode, the GUI reads the average RGB value of each patch, the selected conversion model and the target $L^*a^*b^*$ values; then the software fits the conversion model parameters, which are saved in a *.mat file in the current directory.

3.2.2 Testing mode

In this mode, the GUI reads the average RGB value of each patch, the selected conversion model and its parameters; then the software predicts the $L^*a^*b^*$ values of each patch.

The model parameters are stored in a *.mat file. First, this file is searched in the current directory; if it doesn't exist, the default *.mat file stored in the GUI directory is used.

In both operation modes (New calibration and Testing) the conversion models can be selected from a popup menu, (Fig. 3). The direct model does not have any parameters, so when the "New Calibration" mode is selected with this (direct) model, the GUI automatically changes to "Testing".

In the MATLAB command window the following details are displayed:

- the equation of the selected model (only if an empirical one is used),
- the predicted $L^*a^*b^*$ values for each patch, as well as the average total color difference ΔE calculated respect to target values,
- average absolute residuals of $L^*a^*b^*$ and ΔE values.

Besides, a graphic correlation between predicted and target $L^*a^*b^*$ values is shown.

4. RGB TO CIELAB CONVERSION GUI

This is the main GUI that the user employs to measure the color of his own samples. It transforms RGB values to $L^*a^*b^*$ color space, using some of the previously fitted models.

It is worth mentioning that the GUI includes pre-fitted parameters for all models, obtained in a particular image acquisition chamber with a particular digital camera. New users must calibrate the software using their own image acquisition system.

The steps for performing color space conversion are: (i) read the image; (ii) select the region of interest (ROI); (iii) convert RGB values, main button "RGB => $L^*a^*b^*$ ".

Fig. 5 shows the main control panel of this GUI.

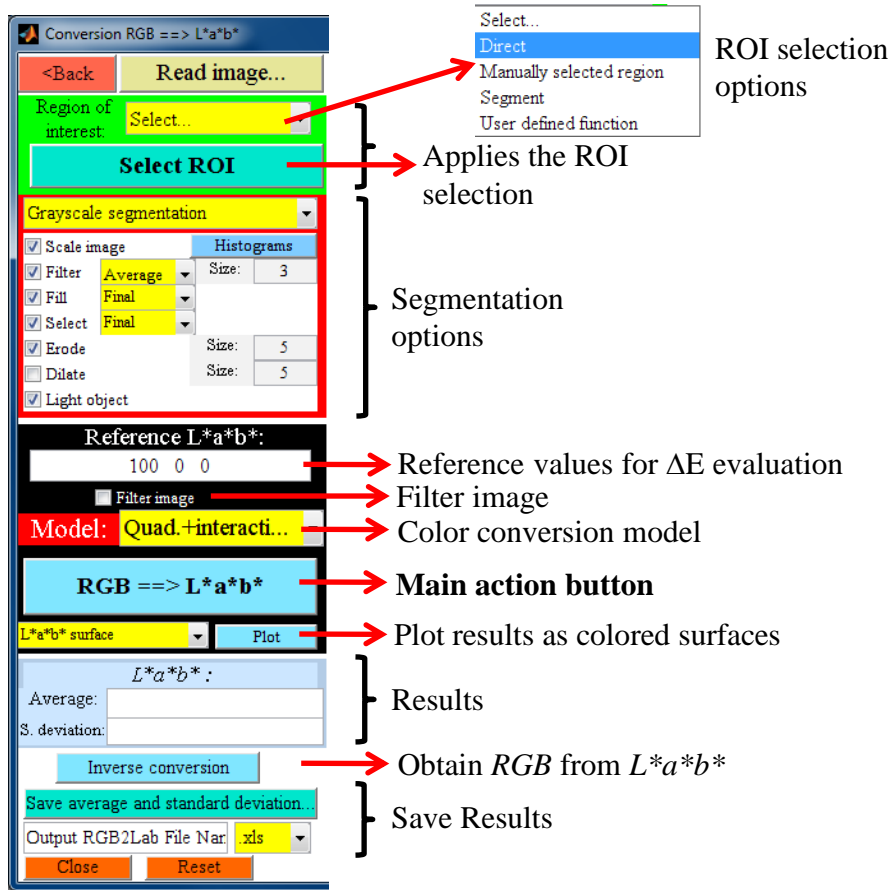


Figure 5. Control panel of the main GUI.

4.1 READ THE IMAGE

Any image format compatible with the MATLAB "imread" function is allowed.

4.2 SELECT THE REGION OF INTEREST (ROI)

The ROI is the specific area where the user intends to obtain color. To perform the ROI selection, there are one direct mode and three user-defined modes, which can be selected from the popup menu (Fig. 5) and executed pushing the "Select ROI" button:

1. **Direct:** employs the entire image to perform color conversion.
2. **Manual:** uses a selected area on the image to perform color conversion. Select points over the image with the left mouse button, forming a closed region. Click twice with the right mouse button, to end the point selection, and to create the mask. Fig. 6 shows an example using an image of a sausage slice. This is the simplest option to define a spatial ROI different from the whole image.

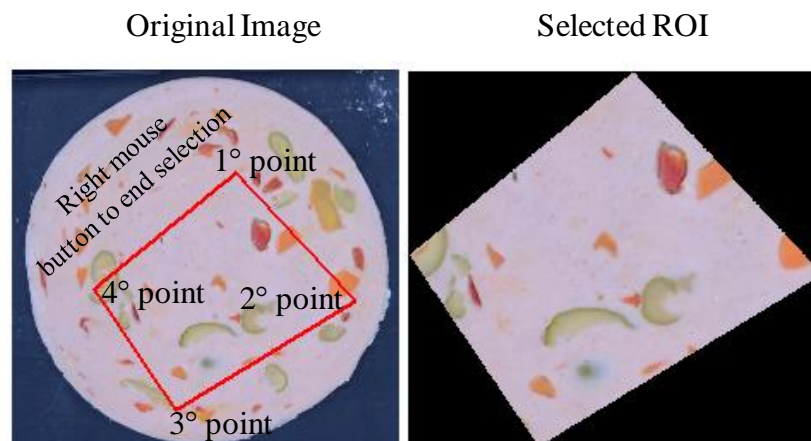
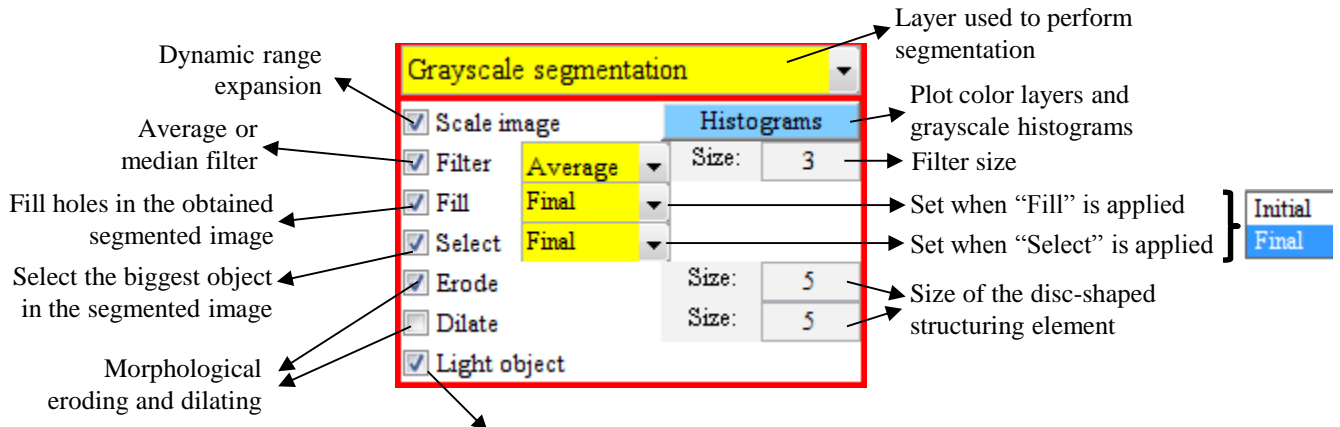


Figure 6. Manually selected ROI.

Segmentation: applies a global thresholding segmentation using one of the image color layers (red, green or blue), or a grayscale image ("rgb2gray" MATLAB function). A new window is opened and successive steps are shown. Segmentation options are detailed in Fig. 7. After a few steps, the histogram of the selected layer is shown and a dialog box will ask the user to enter the threshold; by default the background is assumed darker than the foreground (i.e. the sample), and the GUI take the lighter region as the sample. Two threshold values can be used; in such case the intermediate values define the ROI. Fig. 8 shows the segmentation of a sweet cookie image, using the red layer and thresholds of 75 y 180. Both "Fill" and "Select" options can be applied over the segmented image in two instances, initial (immediately after segmentation) o final (the last step of ROI selection). By default morphological eroding is active, which can help to delete noise or small objects in the image. The morphological dilating can help to restore the size of the main segmented objects.

3. **User defined segmentation:** allows the user to write his own segmentation function. An example file is provided ("segment_example.m").



By default, the GUI assumes that the sample is clearer than the background. If that is not the case, unclick this option, so that the mask is inverted during the process.

Figure 7. Details of ROI selection segmentation options.

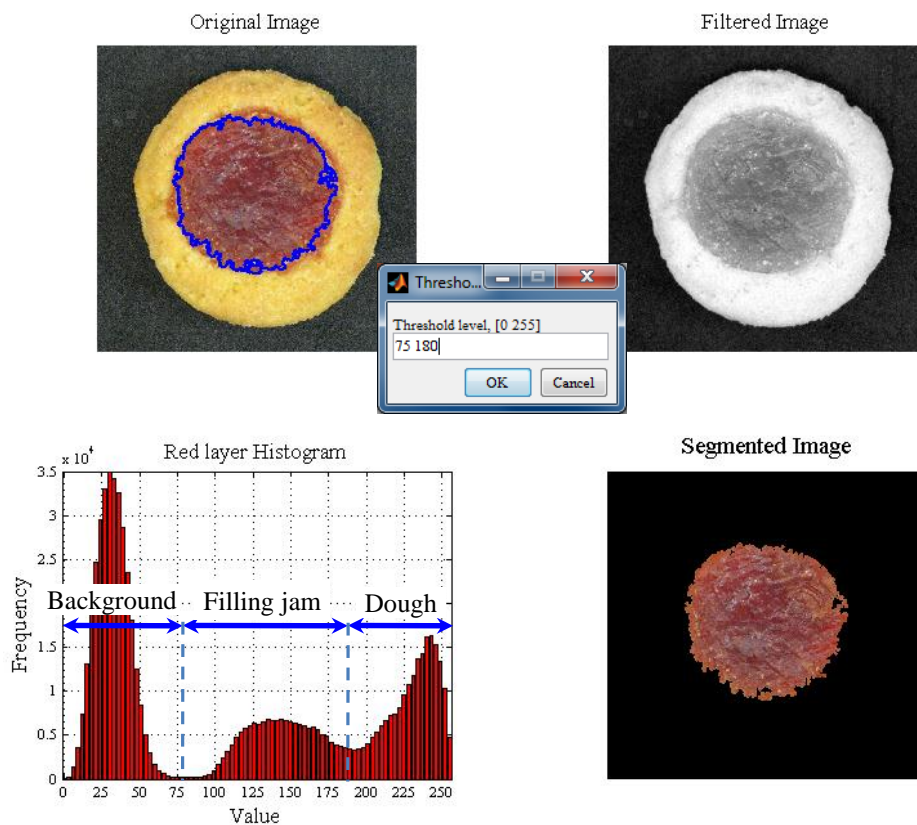


Figure 8. Image segmentation of a sweet cookie sample, using two thresholds.

4.3 CONVERT RGB VALUES

The main button "**RGB => L*a*b***" performs the color conversion pixel by pixel. It is done on the entire image, but only the results extracted from the mask are shown in the *GUI* panel and in the MATLAB command window, as is detailed as follows:

Elapsed time is 1.934000 seconds.

```
Average L*a*b*: 51.069      32.9464      30.8406
S. deviation L*a*b*: 5.8506  5.8063      4.3098
```

```
Average DE*: 66.903
S. deviation DE*: 6.4314
```

```
Minimum L*a*b*: 32.4137      11.665      13.5971
Maximum L*a*b*: 94.7529      55.8438      49.5011
19.2525% pixels converted
```

The “Filter image” option is not selected by default. When it is selected, it uses the filter that was defined in the segmentation control panel.

For further additional processing, the average and standard deviation values are stored in the workspace variables named “average” and “deviation”, respectively. $L^*a^*b^*$ values for each pixel are stored in the variable named “Lab” ($m \times n \times 3$ size; 1° layer is L^* , 2° is a^* , and 3° is b^*), and ΔE (matrix, $m \times n$ size) is stored in the variable named “DE”. The surface distribution of the color parameters can be plotted ($L^*a^*b^*$ or $L^*C^*h^*$) (Fig. 9), as well as the histograms of color parameters (Fig. 10). The value of each color parameter in a specific pixel is automatically indicated in the left bottom corner of the figure.

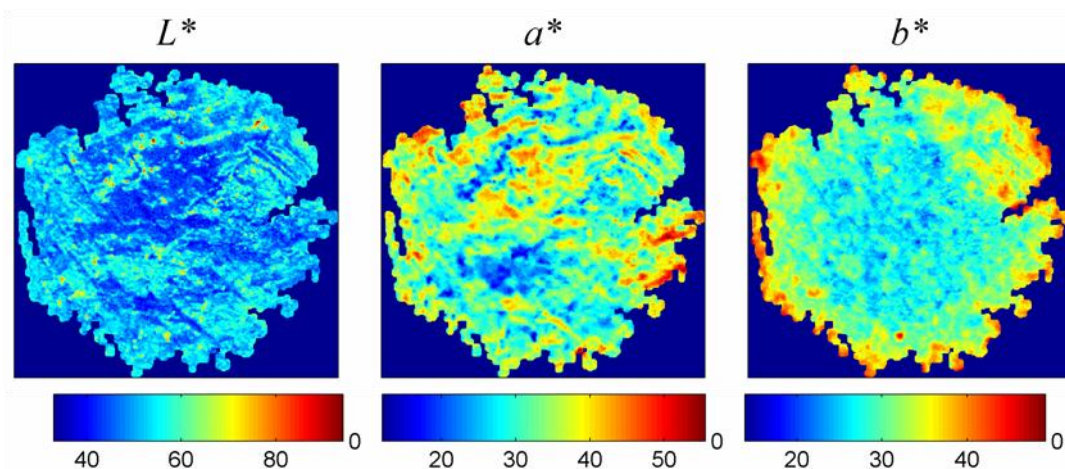


Figure 9. Surface distribution of $L^*a^*b^*$ values.

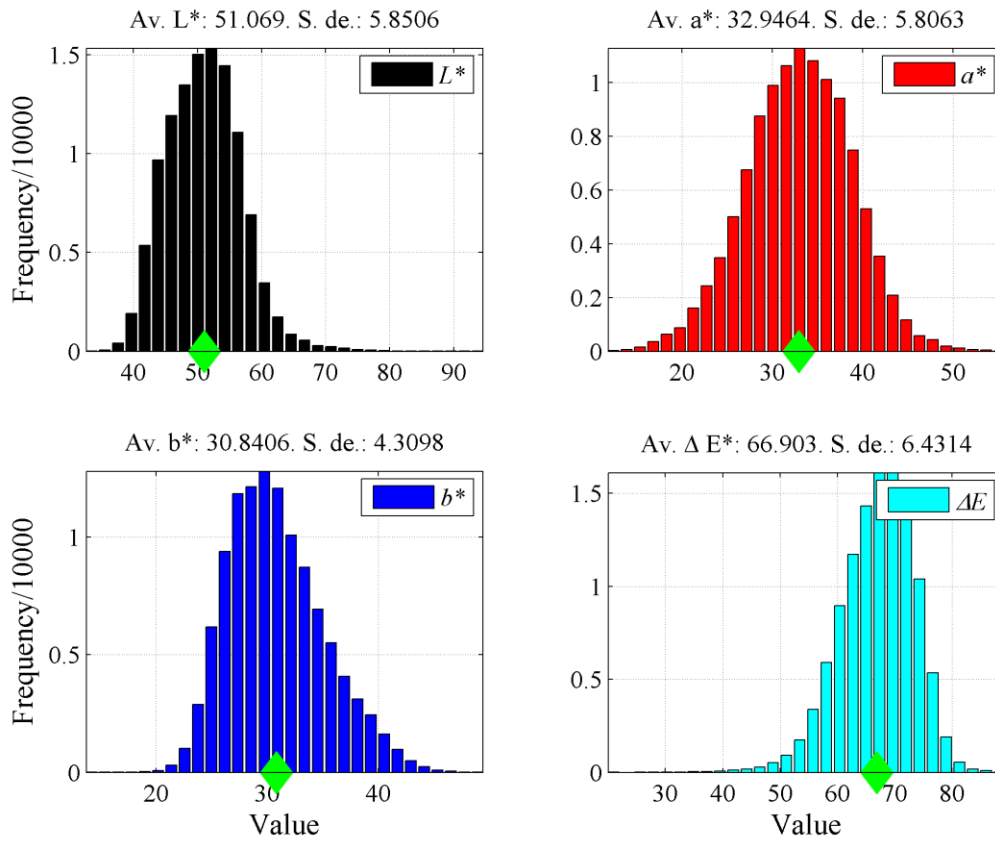


Figure 10. Histograms of $L^*a^*b^*$ values.

The button “**Inverse conversion**” performs the inversion of $L^*a^*b^*$ values back to RGB and displays the obtained image together with the original image. Fig. 12 shows the results of the inversion of the quadratic plus interaction model for the sweet cookie sample shown in Fig. 8. If the direct model is used, only D65 illuminant is allowed. For the linear empirical model the RGB values can be obtained easily. For the empirical models (quadratic, and quadratic plus interactions) the RGB values must be numerically obtained since the relationship is nonlinear. For each pixel in the mask, a system of 3 non-linear equations is solved using the Newton-Raphson method. In consequence, if the image contains a large number of pixels, the inversion process last several minutes.

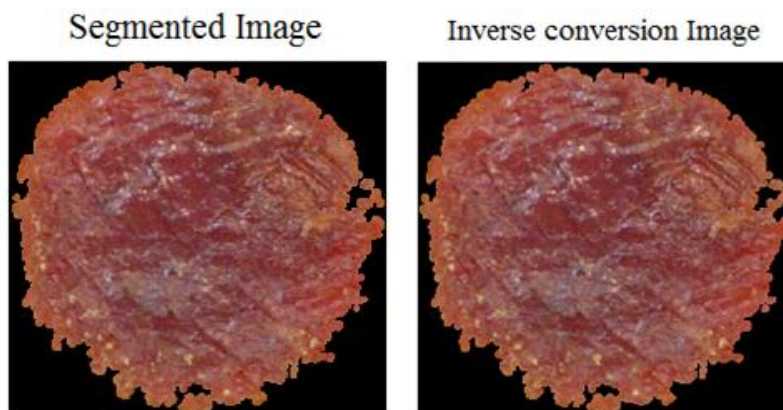


Figure 12. Inversion of the quadratic plus interactions model for the sweet cookie sample.

A summary of the calculations is displayed in the MATLAB command window (134020 pixels are converted, from a total of 696118):

```
19.2525% of pixels to convert
134020*3 non-linear equations
Elapsed time is 33.596831 seconds.
```

Finally, the user can save the calculated values (average and standard deviation ones) in a *.xls file, in the current directory.

5. $L^*A^*B^*$ TO RGB CONVERSION GUI

This interface allows the user to compare different color measurements, obtained in the $L^*a^*b^*$ space color (i.e. by means of a colorimeter or a CVS). Necessarily, to display these values in a monitor, they must be converted from $L^*a^*b^*$ to RGB .

Two options are provided for the conversion: the direct model (D65 illuminant), and a second model (selectable from a popup menu). Two images (patches with a single uniform color) are shown; also an image of the sample can be read and shown in the GUI. Results are qualitative since they depend on the monitor configuration.

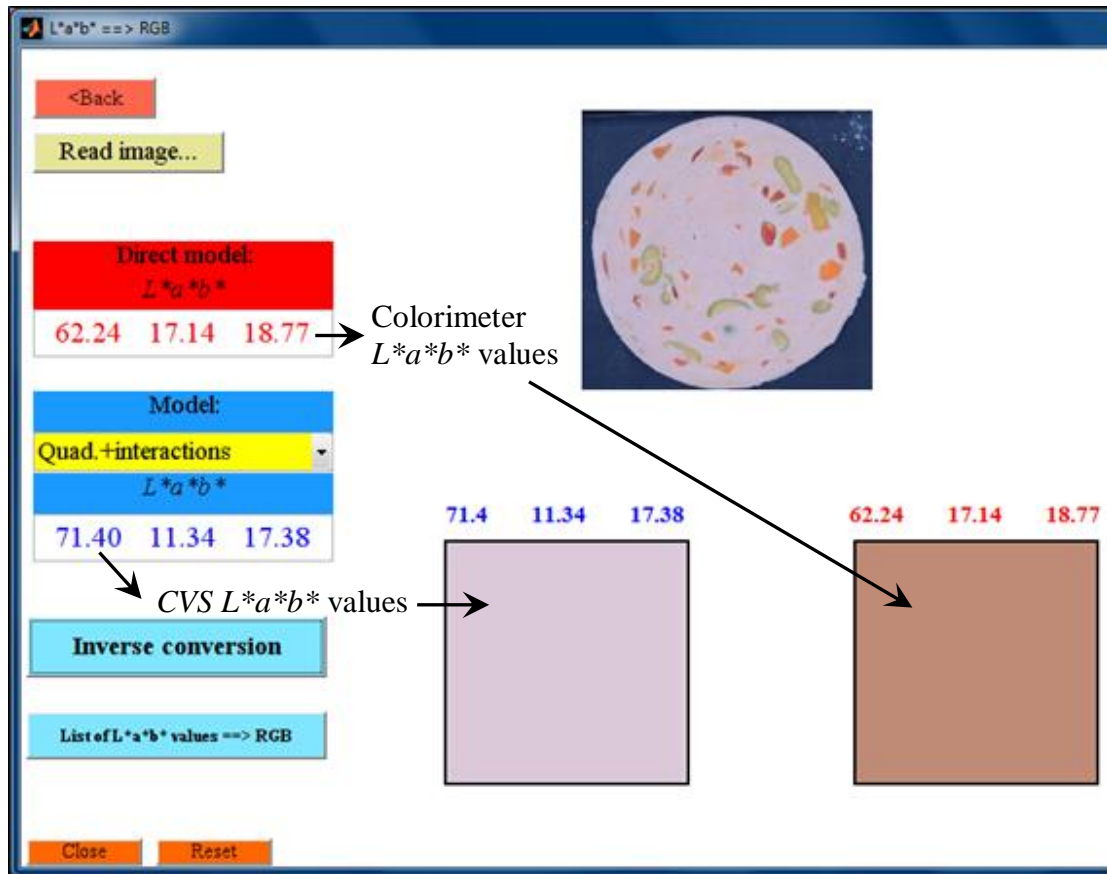


Figure 11. GUI used to perform $L^*a^*b^*$ to RGB conversion. Example using a pork sausage sample.

The button “**List of $L^*a^*b^*$ values => RGB** ” allows the user to read an Excel file with several $L^*a^*b^*$ values and convert it to RGB , which are listed in the command window. Besides, if there are less of 25 points in the data set, a new window is opened which shows RGB patches obtained from such values. An example file is provided, “Lab list example.xlsx”, which have the ColorChecker D65 $L^*a^*b^*$ values.

6. REFERENCES

Gonzalez, R. C., Woods, R. E. (2002). *Digital Image Processing* (2nd Ed.). New Jersey, Prentice Hall.

Goñi, S. M., Salvadori, V. O. (2016). Color measurement: comparison of colorimeter vs. computer vision system. *Journal of Food Measurement and Characterization*. DOI: 10.1007/s11694-016-9421-1.

IEC 61966-2-1. (1999). *Colour Measurement and Management in Multimedia Systems and Equipment - Part 2-1: Default RGB Colour Space - sRGB*. 1^o Ed.

Pascale, D. (2006). *RGB Coordinates of the Macbeth ColorChecker*. [http://www.babelcolor.com/download/RGB Coordinates of the Macbeth ColorChecker.pdf](http://www.babelcolor.com/download/RGB%20Coordinates%20of%20the%20Macbeth%20ColorChecker.pdf) (accessed 28/11/2014).