Ensuring Color Consistency across Multiple Cameras

Adrian Ilie
Department of Computer Science
University of North Carolina at Chapel Hill
Email: adyilie@cs.unc.edu

April 2004

1 Abstract

Color uniformity across the camera images is important to ensure good reconstruction results in computer vision applications that extract depth from multiple synchronized cameras. A way to correct the camera images is to pre-process them. This report describes a method that uses a color calibration pattern to compute the appropriate parameters for each camera, and an application that performs pre-processing on acquired images so that the results are more similar to each other.

2 Color Matching Between Multiple Camera Images

2.1 The Color Matching Process

Different cameras, even of the same make, may have different color gamuts. One way to make images from different cameras look similar is to calibrate them so that, under a common color model, the same colors correspond to the same numerical values. A good choice for a model is RGB (Red, Green, Blue), one of the models proposed by the International Color Consortium (ICC), an association established in 1993 "for the purpose of creating, promoting and encouraging the standardization and evolution of an open, vendor-neutral, cross-platform color management system architecture and components" [1]. This model has its roots in the Helmholtz-Young trichromatic theory of human vision [2, 3]. It is widely-used by camera manufacturers for encoding images, and cameras have three types of image sensors, sensitive to the same three colors. More information on colors in general can be found in [4].

The classic approach to color matching is closed-loop calibration, which involves calibrating specific pieces of hardware to each other. A common calibration method that is used for calibrating a scanner to a printer is scanning a known target, printing it, and then scanning the printout again. A comparison is performed between the two scanned images, and various methods are used in three different places in the calibration loop to compensate for the difference: adjusting the scanner parameters, pre-processing the image in software before printing, or adjusting the printer parameters. A similar approach can be used for calibrating multiple cameras to the same printer, with the secondary effect that all cameras will be calibrated to each other, the desired result for computer vision applications.

Ideally, the comparison should be performed for all the colors present in the images, to construct a lookup table which contains all the correspondences between the same colors in the two images. This is often impractical. However, the properties of colors can be exploited such that, using a limited number of samples and interpolation in color space, the remaining colors can be approximated with reasonable accuracy. The trade-off between the number of samples and the accuracy of the approximation depends on the application and the devices being calibrated. In practice, given a limited number of samples, an empirical response function is fitted to them and used to compute the color values. The most used response functions are linear, logarithmic and exponential, both because they are relatively easy to fit mathematically to a number of samples and because the devices themselves are usually engineered to exhibit such response functions.

This report presents a method that uses regression to compute the parameters of a linear response function for each camera as a pre-processing step. While this process is able to correct images so that they look more similar, because it is only a conversion, it cannot recover the information that is lost during the

acquisition process. Moreover, the transformation may have adverse effects such as losing even more information due to clamping, and amplifying the noise in the camera images. To minimize these effects, the changes through pre-processing should be minimal. Also, high-quality lenses, coatings and sensors should be used, and all adjustments available on the cameras should be used to get all the images to look perceptually similar before attempting to process them. Additional details about color matching are available in [5].

2.2 Color Matching Process Using a Linear Conversion

A simple but effective calibration method assumes that cameras exhibit a linear response function with two coefficients; gain and bias. The gain is a multiplicative term, and the bias is an additive term.

In order to compute its gain and bias, each camera takes a snapshot of a color calibration pattern. The pattern can consist of a grid of squares of different colors, such as the one shown in Figure 1 (left), or it may be a professional color target [6], shown in Figure 1 (right).

Figure 1: *A color calibration pattern, and a professional color target from Gretag Macbeth™.*

The calibration pattern should contain enough colors to provide an adequate number of samples for the computation process. The pattern shown in Figure 1 (left) contains 63 colors with combinations of RGB values on a linear scale on each color channel, and it is designed to uniformly cover the color spectrum. The color target in Figure 1 (right) is designed for professional photographers, and contains colors usually encountered in nature, such as the colors of skin or foliage.

Figure 2 shows a camera image of the pattern in Figure 1 (left) printed on glossy paper with a laser color printer. The colors are quite different from the original.

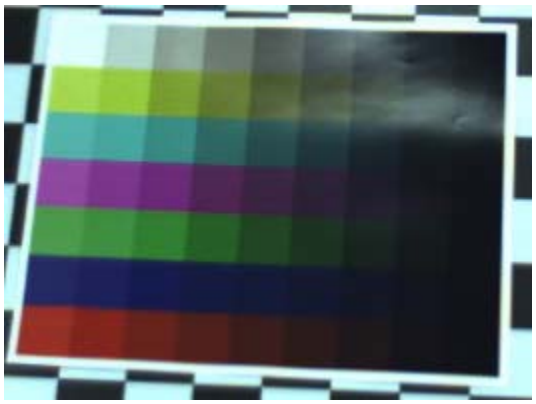

Figure 2: *A sample camera image of the color calibration pattern in Figure 1 (left).*

Factors that affect this image include the lighting conditions, reflectivity of the ink, and glossiness of the paper. For best results, specularities such as the one in the top right corner of Figure 2 should be avoided. Area light sources were used, and the pattern was mounted on a flat cardboard and covered with a material that gives the ink a matte look.

By contrast, professional color targets such as the one in Figure 1 (right) are free of such problems, due to the fact that the color patches are scientifically designed and engineered to reflect light the same way in all parts of the visible spectrum. Figure 3 shows a camera image of the mounted pattern (left), and an image of the professional color target (right).

Figure 3: *A sample camera image of the mounted color calibration pattern, and an image of the GretagMacbeth™ color target.*

The next step of the process is to establish a number of correspondences between squares of the same color in the camera image and the reference image. Given these correspondences, a linear mapping (gain a and bias b) from one to the other can be computed using vertical least-square fitting [7] for each color channel by minimizing the difference in color space between the reference image P and a corrected camera image I', computed follows:

$$\min \sum_{i=1}^{n} \left(I_i' - P_i\right)^2 = \min \sum_{i=1}^{n} \left(\left(a * I_i + b\right) - P_i\right)^2 ,$$

where I_i are samples from the camera image I, a and b are mapping coefficients, P_i are samples from the reference image of the calibration pattern P, and n is the number of samples.

As described in [7], the parameters a and b are computed as follows:

$$a = \frac{\sum_{i=1}^{n} \left(P_i I_i\right) - n\left(\frac{\sum_{i=1}^{n} P_i}{n} \frac{\sum_{i=1}^{n} I_i}{n}\right)}{\sum_{i=1}^{n} \left(P_i\right)^2 - n\left(\frac{\sum_{i=1}^{n} P_i}{n}\right)^2}, \text{ and } b = \frac{\sum_{i=1}^{n} I_i}{n} - a\frac{\sum_{i=1}^{n} P_i}{n}$$

The ideal values for the parameters are close to 1 for *a* and close to 0 for *b*, for which the changes are minimal.

There is currently no standard way to compute the linear conversion parameters from a set of camera images, and apply them to large image datasets before using them as input for computer vision algorithms.

3 The Color Matching Application

This section describes an application that allows manual selection of correspondences and performs the pre-processing step on acquired images. The application was designed to work with images from cameras with a linear response function. It is available upon request from the author. An overview of the application interface and its components is shown in Figure 4.

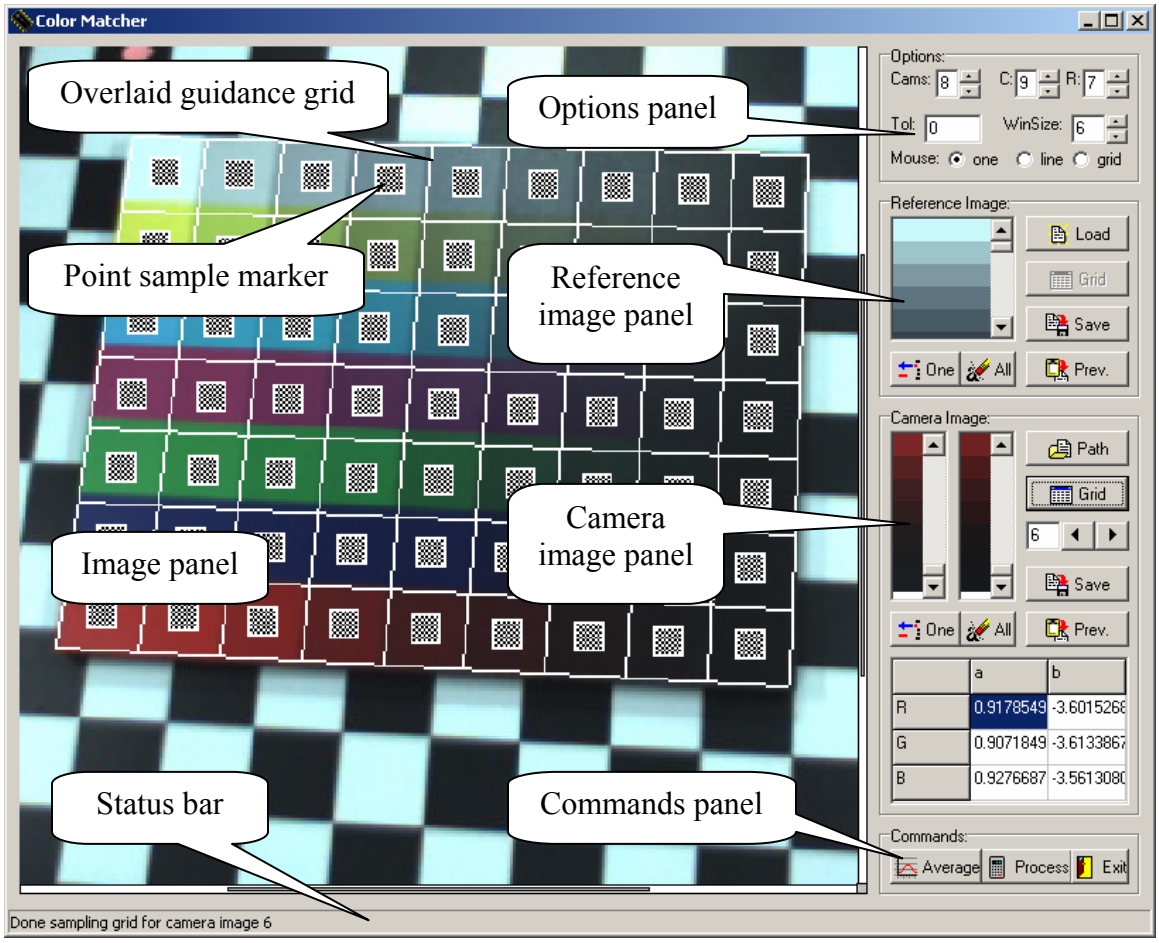

Figure 4: *An overview of the application interface.*

3.1 The Image Panel

The image panel shows the currently loaded image. Clicking the left mouse button is used to mark that point in the image. Samples should be taken in the same order in all images. Clicking the right mouse button and dragging allows zooming into a selected area for greater precision. Clicking the middle mouse button and dragging allows panning. Rotating the mouse wheel allows zooming in and out. The bars on the right and bottom show the zoom level. The gray square in the right-bottom corner, when clicked, restores the view to the entire image.

3.2 The Options Panel

The options panel allows setting some options for the application: the number of cameras in the "*Cam*" text box; the number of rows and columns in the calibration pattern grid in the "*R*" and "*C*" text boxes, respectively; and the mouse mode.
When the selected mouse mode is "*one*", left-clicking on the image panel takes a sample from the image at that point. A square marker will be drawn on the image. The size of the marker reflects the size of the sampling window over which the color of the sample is computed. The "*WinSize*" text box allows setting a different sampling window size. Samples should be taken in the same order e.g., row by row, in all camera images. *R* x *C* samples need to be acquired for each image.

When the selected mouse mode is "*line*", left-clicking two times on the image panel allows setting the two ends of a line segment for automatic sample detection. The program tries to automatically detect the squares on the defined line and place one color sample in each. The color tolerance value in the "*Tol*" text box specifies the minimum distance in color space used by the application to distinguish between two neighboring squares. The "*line*" mouse mode works well for the reference image, but tends to not work for the camera images, due to the variable amount of noise present across them. This is why for camera images, the "*one*" or the "*grid*" mouse modes should be used to collect color samples.

When the selected mouse mode is "*grid*", left-clicking four times on the image panel sets the four corners of a rectangular guidance grid that is overlaid on top of the image to allow easier selection of the samples in the "*one*" mouse mode. If accurate enough, the guidance grid can be decomposed into *R* x *C* rectangular pieces and used to acquire the *R* x *C* samples automatically, at the center of each piece of the grid, by pressing the appropriate "*Grid*" button in the "*Reference Image*" or "*Camera Image*" panels. The application assumes little or no perspective distortion when subdividing the grid into *R* x *C* pieces. The four corners are specified by left-clicking clockwise four times on the image panel, and should be specified in the same order for all images if automatic sampling is used, in order to ensure the same sampling order.

3.3 The Reference Image Panel

The "*Reference Image*" panel shows the parameters of the reference image (containing the colors of the generated calibration pattern itself) and allows sampling its colors. It contains a list of color samples and a few command buttons. The sampling process consists of selecting *R* x *C* samples from the reference image, in a particular order, either manually or automatically.

The "*Load*" button loads the pattern calibration image into the image panel. The image formats accepted by the applications are JPEG (.jpg), bitmap (.bmp) and TIFF (.tif).

The list of color samples shows all the samples that have been acquired so far. The currently selected sample may be deleted by pressing the "*One*" button. Note that the color sample marker in the image panel will not disappear. The entire list is cleared by pressing the "*All*" button, which also reloads the calibration pattern image, clearing all the sample markers.

The "*Grid*" button automatically samples the reference image *R* x *Cs* times on a previously defined grid. It is disabled if a grid has not been defined yet.

The "*Save*" button saves the acquired color samples as a text file, for later use. The "*Prev.*" button loads a list of color samples from a previously saved text file.

Once the color list for the reference image is set (either in the current session, or loaded from a previous session), the correspondences with the camera images can be computed.

3.4 The Camera Image Panel

The "*Camera Image*" panel shows the parameters of the camera images and allows sampling their colors. It contains two lists of color samples, a few command buttons, and a table. The sampling process consists of selecting *R* x *C* samples from each camera image, in the same order in which they were selected from the reference image.
The "*Path*" button sets the pattern for the file names of the camera images. The application displays an image open dialog that allows selecting the first camera image, and then automatically loads all the images in the sequence. The file name pattern that is expected is: "*<prefix>_<CamNumber><suffix>.<ext>*", where *<suffix>* starts with a letter, and *<ext>* is the file extension (jpg, bmp or tif).

The list of color samples on the left contains the colors that are sampled from the current camera image. The list of color samples on the right contains the corrected colors that are computed using the parameters estimated from the samples input so far. The currently selected sample in both lists may be deleted by pressing the "*One*" button. Note that the color sample marker will not disappear. The entire lists are cleared by pressing the "*All*" button, which also reloads the current camera image, clearing all the sample markers. The values of parameters *a* and *b* are estimated each time a new sample is added, but they are only saved when *R* x *C* samples have been acquired.

The left and right arrow buttons allow selecting the current image to work on. Note that selecting a new image clears the guidance grid for the current image, and the current values of the parameters *a* and *b* if they have not been estimated from *R* x *C* samples. The buttons are enabled only after setting the pattern of the file names of the camera images using the "*Path*" button.

The "*Grid*" button automatically samples the current camera image *R* x *C* times on a previously defined grid. It is disabled if a grid has not been defined yet.

The "*Save*" button saves the acquired color samples and the computed parameters as two text files for each camera, for later use. The application displays an input dialog, allowing the user to set the *<prefix>* for the file names. The saved files will be named "*<prefix>_<CamNumber>_original.txt*", "*<prefix>_<CamNumber>_matched.txt*", "*<prefix>_<CamNumber>_variances.txt*" and "*<prefix>_<CamNumber>_params.txt*". The first file contains a list of the original color values, and the second file contains a list of the corrected color values, expressed as 32-bit RBG values (Red + 256 x Blue + 65536 x Green). The third file contains the color variances inside the sampling window for each color in one row, and for each color channel (red, green and blue) separately in three columns. The variance values are only computed during automatic sampling. The last file contains the computed values of the parameters *a* and *b* for the three color channels Red, Green, and Blue, respectively). The button is enabled only after setting the pattern of the file names of the camera images with the "*Path*" button or loading previously-saved data with the "*Prev.*" button.

The "*Prev.*" button loads previously-saved data. The application displays a file open dialog, allowing the user to select any of the files saved with the names specified above, and then loads the parameters for all the cameras. The expected file name pattern is "*<prefix>_<CamNumber>_<suffix>.txt*". This button is enabled only after loading the color list for the reference image.

The "*RGB-ab*" table shows the current computed values for the parameters *a* and *b* for each of the color channels R, G, and B of the current camera image.

Once the parameters for all the camera images are set (either in the current session, or loaded from a previous session), the application is ready to perform the pre-processing step on acquired images.

3.4 The Commands Panel

The "*Commands*" panel allows pre-processing acquired images once the parameters for all the cameras are set. It contains three command buttons.

The "*Average*" button computes the average color for each sample across all images, then replaces the reference image sample values with the average color values computed, and recomputes the parameters *a* and *b* for all the cameras. This processing step is optional, and is aimed at getting all the camera images to look more similar with minimal changes, rather than trying to also get them to look like the original color target. Averaging is used to match each of the samples in all the camera images to their average, thus minimizing the changes in each image, instead of trying to match each color sample to the corresponding color of the original color target.

The "*Process*" button allows performing the processing on a set of images. The application displays an image open dialog, allowing the user to select any of the images. The expected pattern for the file name is "*<prefix>_<CamNumber>f<FrameNumber>.<ext>*". The application then displays an input dialog, allowing the user to set the *<NewPrefix>* to the processed image file names. The images will be saved as "*<NewPrefix>_<OldName>*".

During processing, a down-sampled version of the image being processed is displayed in the camera image panel. The "*Exit*" button also changes to "*Stop*", and by pressing it the processing can be stopped.
The "*Exit*" button closes the application. It is also used to interrupt the pre-processing step, when its label changes to "*Stop*".

3.4 The Status Bar

The status bar is used to display important messages, such as the number of the current guidance grid corner, color sample, or image being processed.

3.5 Other Features

The application offers an easy way to save the values of the color samples, which allows for alternative correction methods to be used, employing more complicated curve-fitting algorithms. It also saves (in automatic sampling mode) the variance inside the sampling window for each sample in each color channel, information that can also be used in more sophisticated algorithms.

4 Results

This section presents the use of the color calibration pattern and the professional color target in Figure 1 to correct the images from several PointGrey DragonFly [8] cameras.

Snapshots of the mounted color calibration pattern in Figure 3 (left) were taken to illustrate how the correction process is applied across the color spectrum. Figure 5 shows the sampled and corrected data (the camera response function) for one of the color channels of one of the cameras plotted against the values taken from the calibration pattern image itself. Notice that the corrected values are closer to the ideal 45 degree line than the original values. The computed parameter values in this case were $a = 1.503834125$ and $b = -25.28260737$, which lead to a greater part of the spectrum than actually necessary being allocated to dark colors, and a lot of bright colors being saturated. One possible reason for this is that the mounting process makes the dark colors look brighter, because the protective coating that was applied reflects light even if the underlying color is dark.

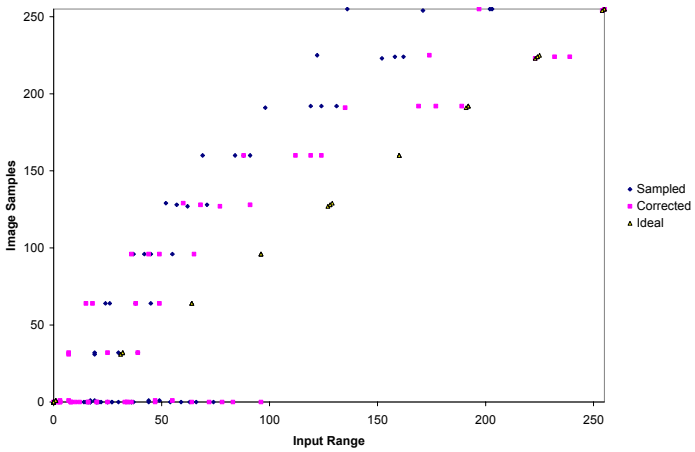

Figure 5: *An example of sampled and corrected color values.*

Figure 6 shows a camera image of the mounted color pattern before and after correction. When using the optional averaging step, the differences are less noticeable (*a* is close to 1 and *b* is close to 0), but the camera images are more similar to each other than before processing and the changes are minimal.

The GretagMacbeth™ professional color target yielded different results for the color calibration. The computed parameter values in this case were much closer to the neutral $a = 1$ and $b = 0$, and the bright colors were no longer saturated.

Figure 6: *A camera image before and after correction.*

Finally, the parameter values computed using this approach were used to process images from eight PointGrey DragonFly cameras, as a step in the reconstruction process presented by Ruigang Yang in his PhD thesis [9]. The GretagMacbeth™ color target was used for the calibration. Figure 7 shows sample results of this reconstruction method, with (left) and without preprocessing (right).

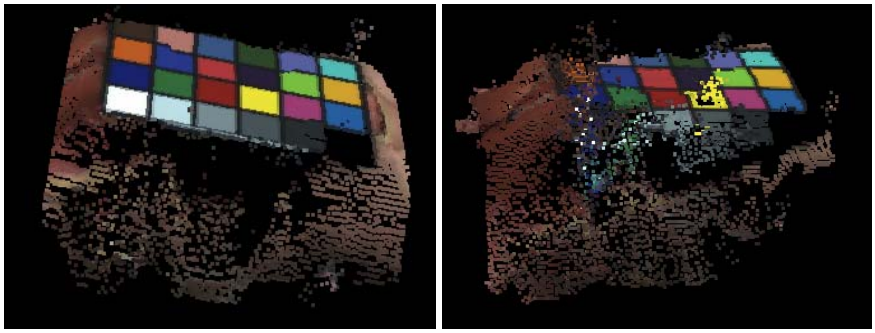

Figure 7: *Reconstruction results with and without preprocessing.*

5 Conclusion and Future Work

This report discussed adjusting images by pre-processing, one of the three different places in closed-loop calibration where differences between multiple camera images can be minimized (the other two being adjusting the parameters of the cameras and the printer). The author is currently working on a method that uses the professional color target in Figure 1 (right) and the hardware capabilities for image adjustment available in some cameras to accomplish the same goal with less clamping errors.

This pre-processing method is recommended for use only in conjunction with prior manual adjustment of the cameras until all images look as similar as possible. It is also recommended that the color samples used in the calibration process are similar to the colors encountered in the scene to be reconstructed: the color pattern in Figure 1 (left), even though it was designed to cover the entire color spectrum uniformly, yielded worse results than the professional color target in Figure 1 (right). When used correctly, the method can bring improvements to computer vision algorithms.

Acknowledgements

The author wishes to thank Greg Welch for his guidance and useful suggestions. This research was supported by NSF ITR Grant no. 0121657: “Electronic Books for the Tele-Immersion Age” and by NLM contract no. NO1-LM-3-3514: "Three-Dimensional Tele-Presence for Medical Consultation: Extending Medical Expertise Throughout, Between, and Beyond Hospitals".

References

[1] *The International Color Consortium*, http://www.color.org .

[2] Thomas Young: “*On the theory of light and colours*” Philosophical transactions of the Royal Society of London, 1802, 92: 12-48 .

[3] Herman von Helmholtz: “*Über die Theorie der zusammengesetzten Farben*”, Archiv für Anatomie, Physiologie und wissenschaftliche Medizin, Berlin, 1850: 461-482 .

[4] Charles Poynton, “*Poynton’s Color FAQ*”, http://www.inforamp.net/~poynton/Poynton-color.html .

[5] David Stone, “*Color Matching*”, http://www.extremetech.com/article2/0,3973,15467,00.asp, 2001 .

[6] *GretagMacbeth, a color sytems and software company*, http://www.gretagmacbeth.com/Source/Gm.asp.

[7] Eric Weisstein, “*Least Squares Fitting*”. In MathWorld—A Wolfram Web Resource, http://mathworld.wolfram.com/LeastSquaresFitting.html .

[8] *PointGrey Research, Inc.*, http://www.ptgrey.com/ .

[9] Ruigang Yang, “*View-Dependent Pixel Coloring - A Physically-Based Approach for 2D View Synthesis*”, University of North Carolina at Chapel Hill, Computer Science Department, http://www/Publications/full_dissertations/yang_dissertation.pdf .