

# How to Print A Munsell Book

Paul Centore

© December 30, 2012

## 1 Introduction

The Munsell colour system's perceptual properties (hue, value, and chroma) suit artists well, for both education and colour selection. For maximum effectiveness, painters need a physical exemplification, such as a printed Munsell book. As of December 2012, only two such products are available: a book by Jim Long,<sup>1</sup> and a book produced by X-Rite.<sup>2</sup> Long's book is reasonably priced at about \$65 US, but only contains about 250 colours. X-Rite's book presents a wide colour gamut, but also costs well over \$500 US. Recent spectrophotometer and printer technology enables controlled colour printing at home, so a Munsell book printing project was initiated.

The goal of this project is to print an affordable, accurate, wide-gamut Munsell book, at home. This article details the steps taken to achieve this goal. An algorithm was developed and implemented, to identify *RGB* triples that produce desired Munsell colours, for a fixed printing system. Accuracy was considered sufficient if the printed colours agreed with the Munsell aimpoints to within two CIE DE2000 units. Readers are invited to try these steps themselves, and to modify and improve the open-source computer code as needed.

## 2 The Munsell Colour System

Colours can be roughly classified as either light sources, or surface colours. A surface colour, such as a patch of paint on a canvas, is perceived when it is illuminated by a light source, which produces a visual stimulus by reflecting off the surface colour, and into a viewer's eye. Visible light varies in wavelength from about 400 to 700 nm, and a coloured surface reflects different percentages of the impinging light, varying

with wavelength. The reflectance percentage, as a function of wavelength, is called a *reflectance spectrum*.

The human visual system filters a stimulus (which takes on different values for different wavelengths) into three coordinates. In 1931, the Commission Internationale de l'Éclairage (CIE) standardized these coordinates, and denoted them  $X$ ,  $Y$ , and  $Z$ . Under simple viewing conditions, two visual stimuli have the same colour if and only if their  $X$ ,  $Y$ , and  $Z$  coordinates are identical. The CIE has also defined some equivalent coordinate systems, such as  $xyY$  and  $Lab$ . The choice of coordinate system is a matter of convenience, as it is always possible to convert amongst them. The  $Lab$  system will be useful later, because the perceptual difference between two colours,  $L_1a_1b_1$  and  $L_2a_2b_2$ , is roughly their Euclidean distance:

$$\sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2}. \quad (1)$$

In 2000, the CIE defined a more refined colour difference (abbreviated DE), denoted CIE DE2000 (Ref. 3). DE2000 can be calculated from any of the standard CIE coordinate systems. As a general rule, if the DE between two colours is 1 or less, then the two colours are indistinguishable. If the DE is 2 or less, the colours match well enough for most practical purposes.

The Munsell system classifies surface colours, such as patches of paint on a canvas. The system is defined both perceptually and quantitatively. The perceptual definition identifies colour attributes that are appropriate for painters or designers. The quantitative definition provides measurement standards that are appropriate for technical and scientific use.

Perceptually, a colour is specified by its hue, value, and chroma. Hue is notated by a number between 0 and 10, which prefixes one of ten hue names: red (R), yellow-red (YR), yellow (Y), green-yellow (GY), green (G), blue-green (BG), blue (B), purple-blue (PB), purple (P), and red-purple (RP). There are a total of 100 hues with integer prefixes. Value, indicating how light a colour is, is a number between 0 (signifying black) and 10 (signifying white). For any given Munsell hue and value, chroma starts at 0 (grey) and extends to a variable positive number, which represents the most saturated colour of that hue and value. A colour with chroma 0 is called *achromatic*, while other colours are called *chromatic*. The 100 hues with integer prefixes are evenly spaced perceptually, as are values and chromas. A Munsell specification is of the form  $H V/C$ . For example, 6GY 8.4/2.6 is a pale, dull, yellowish green. Neutral greys are expressed in the form  $NV$ , where  $V$  is the grey's value.

Munsell originally developed his system, using the perceptual definition, in the early 20th century. In 1943, the Munsell notation<sup>4</sup> defined the Munsell system

quantitatively, and the renotation remains the standard today. The renotation is a look-up table that expresses Munsell samples in CIE coordinates. The table contains about 2750 Munsell samples, all with integer Munsell values, even Munsell chromas, and hue prefixes that are integer multiples of 2.5.

This project aims to print colours that agree with the Munsell system. The renotation provides an objective test of this agreement. Say we have printed a colour that is supposed to match 5PB 6/6. The reflectance spectrum of the printed sample can be measured with a spectrophotometer. A computer program calculates the CIE coordinates that result when Illuminant C (a standard illuminant, used in the renotation) shines on the sample. The renotation tables provide the CIE coordinates of 5PB 6/6. Finally, the DE2000 value for the two sets of coordinates is calculated. The smaller the DE is, the better the match is.

### 3 Equipment

The equipment to produce a Munsell book at home is readily available, and not too expensive:

1. A higher-end printer. For this project, an Epson Stylus Photo R2880 was used, which cost \$600 US,
2. Higher-quality photographic paper. Kirkland Signature Pro Glossy Inkjet Photo Paper (Item #503626) was used, at a cost of about 16 cents per sheet,
3. A spectrophotometer. An X-Rite ColorMunki was used, which cost \$450 US, and
4. A computer, equipped with Octave or MATLAB. MATLAB is expensive, but Octave is an open-source MATLAB clone that can be downloaded for free.

The following sections discuss the equipment in greater detail.

#### 3.1 Printer

A variety of printers are available for the home market. The cheapest are less than \$100 US, but their limited colour gamuts, and inability to handle photographic paper, make them inadequate for printing a Munsell book.

The limited colour gamut is particularly obvious in the dark colours. Typically, a quantity called  $D_{\max}$  measures the darkness obtainable with a given printer-ink-paper combination.<sup>5</sup> The  $D$ -value for a printed colour is defined by

$$10^{-D} = \frac{Y}{100}, \quad (2)$$

where the CIE coordinate  $Y$  is the *reflectance factor* of the printed colour. A printed colour's reflectance factor is the percentage of light that that colour reflects, weighted with respect to human sensitivity to different wavelengths.  $D$ -values are usually applied to blacks, whites, and greys, whose reflectance spectrum does not vary much with wavelength. A middle grey, for example, might reflect 20% of every wavelength. In this case,  $Y$  is the common reflectance percentage, 20%.  $D_{\min}$  and  $D_{\max}$  are the minimum and maximum  $D$ -values that a printer-ink-paper combination can reach. The larger  $D_{\max}$  is, the darker the black that can be printed.

$D$ -values refer not just to a printer, but to a combination of printer, printing inks, and paper. The same printer can have a different  $D_{\max}$  (or  $D_{\min}$ ) if a different paper is used. Some printers can use multiple sets of inks, and the choice of ink can also affect  $D_{\min}$  and  $D_{\max}$ . Because of this dependency, the printer, inks, and paper should be chosen jointly, rather than separately.

Equation 2 converts between  $Y$  and  $D$ . In addition, there is a fifth-degree polynomial<sup>6</sup> that converts between  $Y$  and Munsell value. Although Munsell value steps are perceptually equal, changes in  $Y$  are not. For example, the increase of  $Y$  from 0.6% to 2.0% is perceptually equal to the increase from 58% to 77%, because they are both one Munsell value step. Table 1 lists some conversions between  $Y$ ,  $D$ , and Munsell value.

Reflectance Factor ( $Y$ )	$D$	Munsell Value
0.1	3.0	0.1
0.6	2.2	0.5
1.0	2.0	0.8
1.2	1.9	1.0
2.0	1.7	1.5
3.0	1.5	2.0
57.6	0.2	8.0
76.7	0.1	9.0
100.0	0.0	10.0

Table 1: Comparison of Reflectance Factor,  $D$ , and Munsell Value

The  $D_{\max}$  of many cheap printers is not much more than 1.5, which corresponds to a Munsell value of 2. Furthermore, this  $D_{\max}$  is probably only achievable with pure black ink, so no chromatic colours of value 2 or less can be printed. This limitation can severely restrict the gamut of the printed Munsell book. To achieve a Munsell value of 1, a printer with a  $D_{\max}$  of about 2 is necessary. In order to print chromatic colours of Munsell value 1,  $D_{\max}$  should exceed 2. In general, then, a

higher-end printer will be needed to achieve a good set of dark Munsell colours.

Higher-end printers tend to use specially coated photographic papers, rather than office paper. The coatings hold a thick layer of ink, and provide opacity. The thick layers of ink allow darker and more saturated colours, which increases the gamut. The opacity provides consistent viewing conditions. Since most office paper is slightly translucent, the colour seen through the paper affects the colour printed on the paper. Opacity generally requires a thicker paper, which does not bend readily. Many cheaper printers bend their paper 180° around a narrow roller, which would crease photographic paper, so higher-end printers minimize paper bending.

An additional consideration is the printer's hue gamut. Basic inksets contain four inks: cyan, magenta, yellow, and black (CMYK). More sophisticated printers contain additional colours, such as orange, or contain multiple versions of magenta or cyan. The printer selected, an Epson Stylus Photo R2880, uses eight inks. There are two versions of magenta, two versions of cyan, and two different greys, as well as yellow. In addition, the user can choose between a matte black, and a glossier photo black. For this project, the photo black was used, because it produces a higher  $D_{\max}$ .

In summary, then, to produce a Munsell book with consistent colours and a wide enough gamut, a higher-end printer is necessary, of the kind that a photographer would use. The Epson R2880 was selected for this project, though many other printers would also have been adequate.

Some printers, including the Epson R2880, require the user to set various parameters. For this project, colour management (a printer setting) was turned off. This setting was used consistently throughout the project.

## 3.2 Paper

Even with the same printer and inks, colour can vary significantly with the paper. To control the production, then, one kind of paper should be selected at the start, and used throughout. For this project, Kirkland Signature Pro Glossy Photo Paper was chosen, after evaluating several brands on four criteria:  $D_{\max}$ ,  $D_{\min}$ , weight, and price.

$D_{\max}$  and  $D_{\min}$  are important because they determine the darkest and lightest colours that can be produced, and thus bound the gamut.  $D_{\max}$  was evaluated by printing a patch with  $RGB$  coordinates (0,0,0), which is the darkest colour that can be specified. The reflectance spectrum of the printed patch was measured with a spectrophotometer. Then  $Y$  was calculated from the reflectance spectrum, and  $D_{\max}$  and the Munsell value were calculated from  $Y$ .

$D_{\min}$  was evaluated by printing a patch with  $RGB$  coordinates (1, 1, 1), which is

the lightest colour that can be specified. Most printers, including the Epson R2880, “print” this colour by leaving the paper blank. (A few very sophisticated printers use a white ink.) Therefore,  $D_{\min}$  was determined by measuring the reflectance factor of the paper itself with a spectrophotometer. Many paper manufacturers list a “brightness” value. This brightness is actually the CIE coordinate  $L$ , calculated with respect to Illuminant D50 or D65. To about two significant digits, Munsell value is just  $L/10$ , so  $D_{\min}$  can be estimated directly from the brightness value. This estimate agreed well with the  $D_{\min}$  calculated from the spectrophotometer.

A paper with brightness less than 90 has a Munsell value less than 9. Unless white ink were used, it would be impossible to produce any colours of Munsell value 9 or lighter on such a paper, which would severely restrict the gamut. In practice, if any chromatic colours of Munsell value 9 are to be obtained, the paper’s brightness should be near 95.

Since a Munsell book is a working tool, it is desirable to print it on a thicker, heavier paper, that can withstand some use.

Table 2 shows the results of the paper testing. Both matte black ink and the glossier photo black ink were tested. Overall, Kirkland Signature Pro Glossy Inkjet Photo Paper (Item #503626) achieved both dark blacks and bright enough whites for a wide gamut. Given that its price was significantly less than any of the competitors, and its weight was on par, it was decided to use Kirkland Signature Pro, along with photo black ink, throughout the project.

### 3.3 Spectrophotometer

A spectrophotometer measures (i.e. finds the reflectance spectrum of) a printed colour patch. CIE coordinates can be calculated from a reflectance spectrum, and compared with the Munsell renotation standard, to evaluate how well a printed colour matches a Munsell aimpoint. For this project, an X-Rite ColorMunki spectrophotometer was used. This section describes how the ColorMunki was used, and evaluates its accuracy.

The ColorMunki is shaped like a large tape measure. It was intended for calibrating printers and monitors, but can be adapted for producing a Munsell book. To measure a printed colour patch, position the ColorMunki over that patch, and press the large button on the side. A USB cable connects the ColorMunki to a computer, where software produces a ColorMunki window. Colours appear in succession in that window, as they are measured.

Paper	Black Ink	$D_{\max}$ (Munsell Value)	$D_{\min}$ (Munsell Value)	Price (¢/sheet)	Weight (g/m <sup>2</sup> )
Epson Ultra Premium Presentation Matte	Matte	1.67 (1.58)	0.05 (9.56)	35	192
Epson Photo Paper Glossy	Matte	1.67 (1.58)	0.05 (9.58)	27	225
Soho Matte	Matte	1.50 (2.05)	0.08 (9.33)	25	230
Soho Glossy	Matte	2.11 (0.67)	0.08 (9.32)	22	180
Strathmore Matte Photo	Matte	1.57 (1.85)	0.05 (9.59)	45	207
Moab Lasal Photo Matte	Matte	1.60 (1.77)	0.04 (9.65)	41	235
Epson Ultra Premium Presentation Glossy	Matte	1.65 (1.64)	0.03 (9.70)	60	297
Epson Photo Paper Glossy	Photo	2.21 (0.55)	0.05 (9.57)	27	225
Epson Ultra Premium Presentation Matte	Photo	1.34 (2.51)	0.04 (9.62)	35	192
Soho Glossy	Photo	1.59 (1.78)	0.07 (9.37)	22	180
Office Max High Gloss Photo	Photo	1.69 (1.53)	0.06 (9.45)	35	N/A
Staples Photo Supreme High Gloss	Photo	2.27 (0.47)	0.07 (9.35)	60	300
Kirkland Signature Pro Glossy Inkjet Photo, #126755	Photo	2.25 (0.49)	0.05 (9.52)	22	255
Epson Ultra Premium Presentation Glossy	Photo	2.22 (0.54)	0.05 (9.56)	60	297
Kirkland Signature Pro Glossy Inkjet Photo, #503626	Photo	2.24 (0.51)	0.07 (9.41)	16	255

Table 2: Comparison Of Different Papers

Reflectance spectrum data is not displayed, but can be obtained as follows:

1. Click File, then Export... A window will appear, labeled “Save as” at the top.
2. In that window there is a dropdown menu for “Format.” Select “Comma Separated.”
3. Choose a file name and location, and select “Save.”

The resulting file is a text file with the suffix “.csv.” The file’s top row is a header that lists wavelengths from 380 to 730 nm, in steps of 10 nm. Each subsequent row gives one colour’s reflectance spectrum, consisting of the reflection percentages for each wavelength in the header. Some of the open-source programming routines have been written to read in files in the ColorMunki .csv format, so that reflectance spectra can be used in calculations.

Tests were run to evaluate the consistency of ColorMunki measurements, and the methods of using the ColorMunki. The tests involved measuring a printed colour patch twice, and finding the DE2000 difference between the two measurements. Since the patch itself is presumed not to change between the two measurements, a perfectly consistent measuring device and protocol would have a DE of zero.

ColorMunki consistency is probably more a function of how measurements are made, than of the ColorMunki itself. The Color Munki emits light at a known intensity. This light reflects off a printed colour, and the intensity of the return is measured. For each wavelength, the intensity of the return, divided by the intensity of the original emission, is the reflectance percentage. These measurements will be unreliable if any of the emitted light escapes to the outside world or, conversely, if any stray light from the outside world is measured as part of the return. In order to prevent light from entering or leaving, the ColorMunki has a flat bottom, with a small circular aperture. The aperture is placed against the colour sample to be measured. Ideally, the flat bottom will be flush with the sample, and will prevent any light from escaping or entering.

In practice, of course, light does escape and enter. Measurements varied the least when the ColorMunki was pressed onto the sample during measuring, and the ColorMunki was held in place for a second or two after pressing the measurement button. Figure 1 gives an example of the variability obtained under these conditions. In this example, 1000 colour samples were printed with the Epson R2880. The samples’ *RGB* values were chosen to span the maximum gamut, in hue, value, and chroma, that can be obtained from the printer. Reflectance spectra of the 1000 samples were measured with the ColorMunki. Then the same printout was measured again, with the same device, to produce a second set of reflectance spectra. 1000 DE values were calculated from the 1000 duplicate measurements. Figure 1 shows a histogram



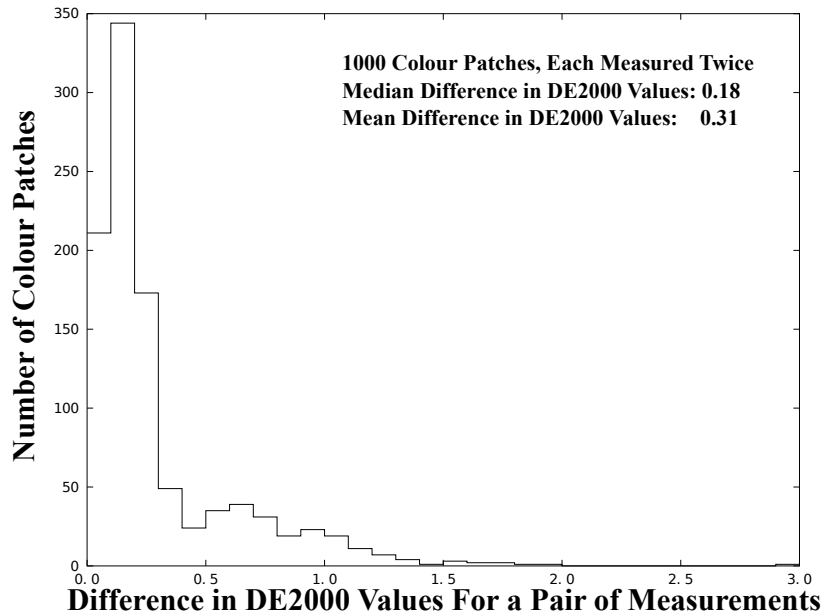


Figure 1: Histogram of DE2000 Value Differences

of the DEs. DEs on the order of 1 or 2 are sufficient for this project. Since the data in Figure 1 shows that half the measurement differences are less than 0.2, this amount of variability is probably acceptable. The low variability seen in Figure 1 disappeared when measurements were made casually, without pressing down firmly, or without letting the ColorMunki linger on the sample. It was not uncommon for the median difference to be considerably greater than 1 when care was not taken. Figure 2 shows an example, when 216 colour patches were measured casually. The mean and median DE are both near 2.

Another factor that was not investigated here, but which could be relevant, is inter-instrument agreement. Errors could occur if two different researchers used two different ColorMunkis. A 2010 informational brochure<sup>7</sup> from X-Rite gives an average DE of 0.6, and a maximum DE of 1.5, for inter-instrument differences. This variability is much greater than what was seen in Figure 1. The current project used the same ColorMunki throughout, so this source of variability was avoided.

Figure 3 shows that greater measurement variation occurs for darker colours. The figure is a scatterplot, with one point for each printed colour patch. The horizontal coordinate is the Munsell value of the patch, while the vertical coordinate is the DE between the two measurements. The plot shows that larger measurement differences tend to occur for lower Munsell values, when the colour is darker. Eight of the

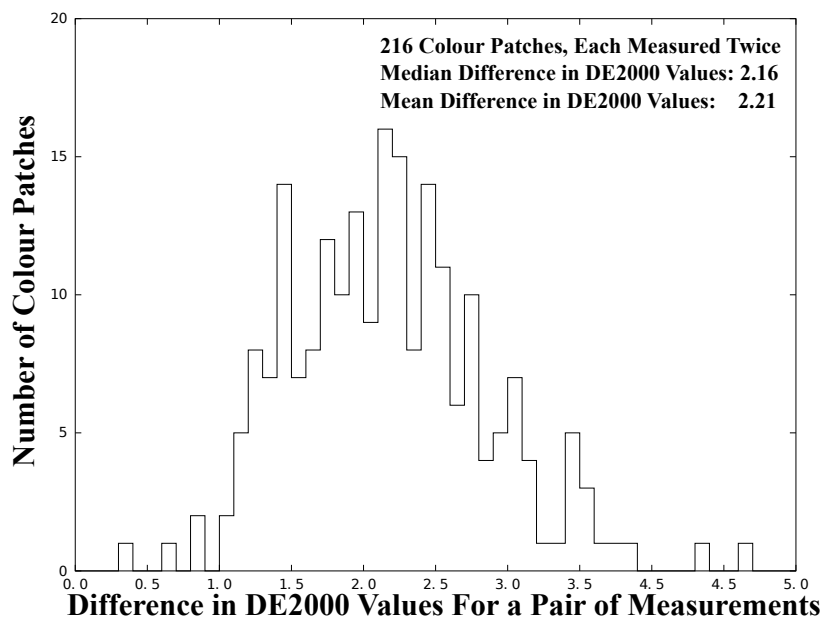


Figure 2: DE2000 Value Differences, Without Careful Measurements

ten differences above 1.5, for example, occur when the Munsell value is less than 2.5. When the Munsell value is 7 or higher, on the other hand, the difference is usually considerably less than 0.5. Darker colours therefore show more measurement variation.

Stray light can explain this correspondence. The absolute quantity of light that enters or escapes through the aperture, which depends on how tightly the Color-Munki presses the sample, does not depend on the colour being measured. The relative quantity of stray light, as a percentage of the reflected light, however, varies consistently with Munsell value. Suppose, for the sake of discussion, that the intensity of the stray light varied uniformly between 2 and 5 percent of the intensity of the emitted light. Then the measured reflectance factor would be in error by somewhere between 2 and 5 percent. This error, however, is physical rather than perceptual. Table 1 shows that Munsell values are “compressed” with respect to the reflectance factor. A Munsell value of 1 corresponds to a reflectance factor of 1.2%, while Munsell value 2 corresponds to a reflectance factor of 3.0%. The physical difference between these factors is 1.8%. When the Munsell values are 8 and 9, the corresponding reflectance factors are 57.6% and 76.7%, a physical difference of nearly 20%. Though the second physical difference is 10 times the first physical difference, the perceptual difference is identical: one Munsell value step. The DE2000 value

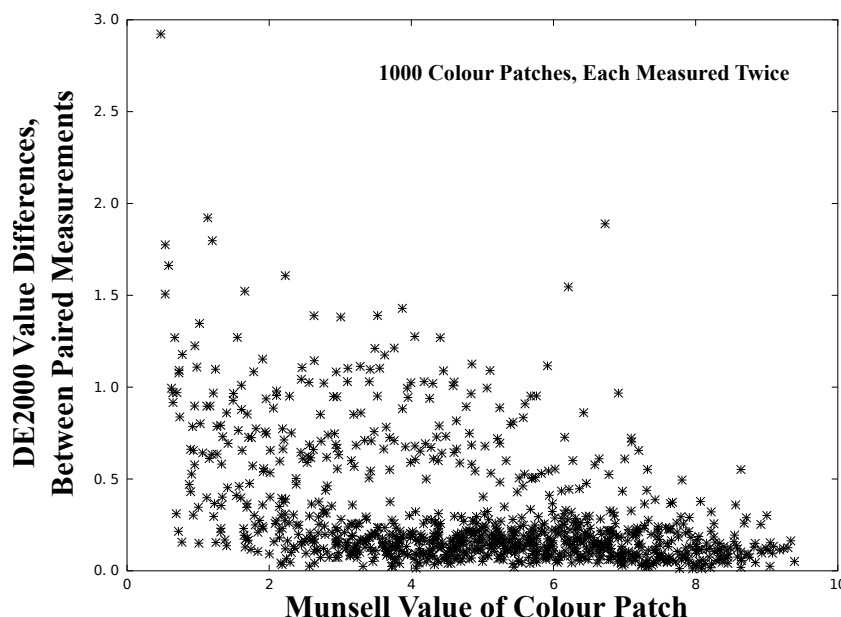


Figure 3: Munsell Value vs. DE2000 Variability

applies to perceptual difference, so a stray light error of 2% causes a much bigger perceptual difference for a sample of Munsell value 1, than a sample of Munsell value 8. In general, then, stray light affects dark colours more than light colours. The reduction of differences when stray light is reduced, by pressing firmly on the ColorMunki, also implies that stray light is responsible for most of the variability seen here.

The ColorMunki was only evaluated for consistency, not for accuracy. Even if repeated measurements give an identical result, it is possible that that result is not very accurate. In the absence of any information to the contrary, however, it was assumed for this project that the ColorMunki was sufficiently accurate.

### 3.4 Computer Program

An iterative algorithm was coded, in order to determine printed *RGB* samples that matched Munsell aimpoints closely enough. The algorithm was coded in Octave, which is a freely downloadable, open-source clone of MATLAB. MATLAB itself was decided against, because of its expense.

For those who are interested, the computer code written for this project is open source, and is available online at [www.99main.com/~centore](http://www.99main.com/~centore). Criticisms, comments,

and bug reports are welcome. Other researchers are also free, and in fact are encouraged, to use, modify, and extend the code in their own work, with the understanding that they will make their own code freely available.

## 4 Algorithm

To print a Munsell book, an algorithm was developed, and implemented in Octave, that would determine, for a given Munsell aimpoint, an *RGB* specification. When printed on a fixed printer under fixed conditions, the *RGB* matches the aimpoint, to within a given DE threshold.

The algorithm treats the printer as a “black box,” meaning that the printed output for a particular *RGB* input can be found, but that the printer’s internal workings are unknown. A developer can print any *RGB* colour and measure its reflectance spectrum. These measurements are the only data the algorithm will use. Of course, the printer uses a colour management module (CMM) and device-specific instructions, to determine what densities of ink, in what order, are applied when given a certain *RGB* input. The developer, though, does not know these details, and does not need to know them.

The following sections will describe the algorithm mathematically. A function (also known as a *mapping*)  $f$  will be constructed, that associates an *RGB* input with the measured *Lab* coordinates of that *RGB* input, when printed. The set of *RGB* inputs is called the *domain*, and the set of *Lab* outputs is called the *codomain*. The process of determining an *Lab* output for a given *RGB* input is called *evaluating* the function  $f$ . The set of *Lab* coordinates for an *RGB* input is called the *image* of that *RGB* input under  $f$ , and, vice versa, the *RGB* input is the *pre-image* of the *Lab* output. The printing problem is how to invert  $f$ , that is, given a desired *Lab* output, what *RGB* will produce it?

The function  $f$  has properties that allow us to use some heuristics, or approximate rules of thumb, in the inversion. Using these heuristics, an adaptive, iterative algorithm will be developed. The algorithm is adaptive in that the function  $f$  will be evaluated more finely around the Munsell aimpoints than elsewhere. It is iterative in that successive estimates will be made for the pre-image of a particular aimpoint, with each estimate closer than the one before it. The general inversion strategy is to generate a set of *Lab* outputs, whose *RGB* pre-images are known, that surround an aimpoint. If the surrounding *Lab* outputs are close enough to one another, then linear interpolation can be used to estimate the pre-image of the aimpoint.

The values of  $f$  that the algorithm uses will be determined by printing and measuring specific *RGB* triples, rather than being known *a priori*. Since measurements

can be time-consuming, the algorithm will ideally minimize the number of *RGB* triples that it evaluates. Of course, the set of evaluated points will be stored in a shade bank, for further use.

## 4.1 Mathematical Formulation

When formulated mathematically, the problem of printing a Munsell book becomes the problem of inverting the function  $f$ . The problem starts with a Munsell colour, or aimpoint, which it is desired to print. The Munsell notation provides CIE  $xyY$  coordinates for that aimpoint. The  $xyY$  coordinates are converted into the more convenient *Lab* coordinates, which  $f$  uses. Inverting  $f$  means finding the *RGB* triple which, when printed, has the desired *Lab* coordinates, and thus the desired Munsell colour.

To develop an inversion algorithm, *RGB* space, the domain of  $f$ , will be parameterized as a cube. The  $R$ ,  $G$ , and  $B$  coordinates each represent a primary that can be turned off entirely (denoted by value 0), or can be continuously increased to some maximum (denoted by value 1). The three coordinates are independent, so *RGB* space is equivalent to  $[0, 1] \times [0, 1] \times [0, 1]$ . Geometrically, this set is the unit cube in  $\mathbf{R}^3$ .

*Lab* space, the codomain of  $f$ , can be similarly parameterized. The lightness  $L$  goes from a minimum of 0, which only occurs when the stimulus is perfectly black, to a maximum of 100, which only occurs when the stimulus is perfectly white.  $a$  and  $b$  can be positive or negative, but their magnitudes are still bounded for printed colours. The bounds are complicated functions, called the MacAdam limits, which depend on  $L$ .

The image of all *RGB* space under  $f$  is a subset of *Lab* space, and is called the printer's gamut. A printer's gamut, even for sophisticated printers, rarely occupies more than about half of *Lab* space. For cheaper printers and papers, the gamut can occupy considerably less than half. The Munsell colours fill *Lab* space completely, so a printer cannot produce many of the Munsell colours. In practice, lower-chroma colours, near the neutral axis, can be produced while higher-chroma colours cannot. Because of this restriction,  $f$  cannot be inverted for many Munsell colours. The algorithm will therefore check that a colour is within the gamut before attempting an inversion.

Both *RGB* and *Lab* spaces have three parameters, and can be viewed as subsets of  $\mathbf{R}^3$ , so  $f$  is a function from (a subset of)  $\mathbf{R}^3$  to (a subset of)  $\mathbf{R}^3$ . Inverting a function from  $\mathbf{R}^3$  to  $\mathbf{R}^3$  is an intractable problem in the general case. In practice, it can only be solved when there are some restrictions on the function. In the current case, such

restrictions, called heuristics, are available. The following sections will identify these heuristics, and show how they can be used to make tetrahedral tessellations. Then tetrahedral constructions will be used in an iterative inversion algorithm.

#### 4.1.1 Heuristics

An *heuristic* is a simplifying assumption that is probably only approximate, but that is close enough for the particular problem at hand. The inversion algorithm will use the following heuristics:

1.  $f$  is one-to-one, or *injective*. Formally, if two  $RGB$  triples are different, then the printed colours they produce will be different,
2.  $f$  is approximately perceptually uniform. In other words, the difference between two printed colours is roughly proportional to the Euclidean distance between the two  $RGB$  triples that produced them. Conversely, if two  $RGB$  triples are very similar, then the printed colours they produce are also similar,
3.  $f$  can be well-approximated by piecewise linear functions. In other words, the partial derivatives of  $f$  are not very high, so  $f$  does not change suddenly,
4. The finer the pieces in a piecewise linear approximation to  $f$ , the better that approximation is, and
5.  $f$  can be evaluated without error. This heuristic will break down when greater accuracy is required than the spectrophotometer can provide. This point marks the limits of the inversion's accuracy.

The first three heuristics, as well as the fifth, derive from engineering. The printer was presumably designed to be useful to humans. Humans would find it confusing if two different  $RGB$ s produced the same colour, so engineers avoid this problem by making  $f$  one-to-one. Similarly, if the scale of reds going from  $RGB = (0, 0, 0)$  to  $RGB = (1, 0, 0)$  were printed out, it should be perceptually uniform. In fact, designers use gamma correction just to make such scales more perceptually uniform. Such uniformity requires that there are no sudden changes in output when there is a small change in  $R$ ,  $G$ , or  $B$ .

A human designer would likely make  $f$  differentiable, or at least piecewise differentiable, and slow changes are insured by keeping derivatives small. Injectivity insures that a function cannot fold back on itself, so it cannot contain any peaks or valleys. When injectivity is combined with slowly changing derivatives, it follows that piecewise linear functions can approximate  $f$  well, as Heuristic 3 states. In the algorithm, the linear pieces will be functions from tetrahedra in  $RGB$  space, to tetrahedra in  $Lab$  space. Linear functions are easily inverted, so the inverse of  $f$  can

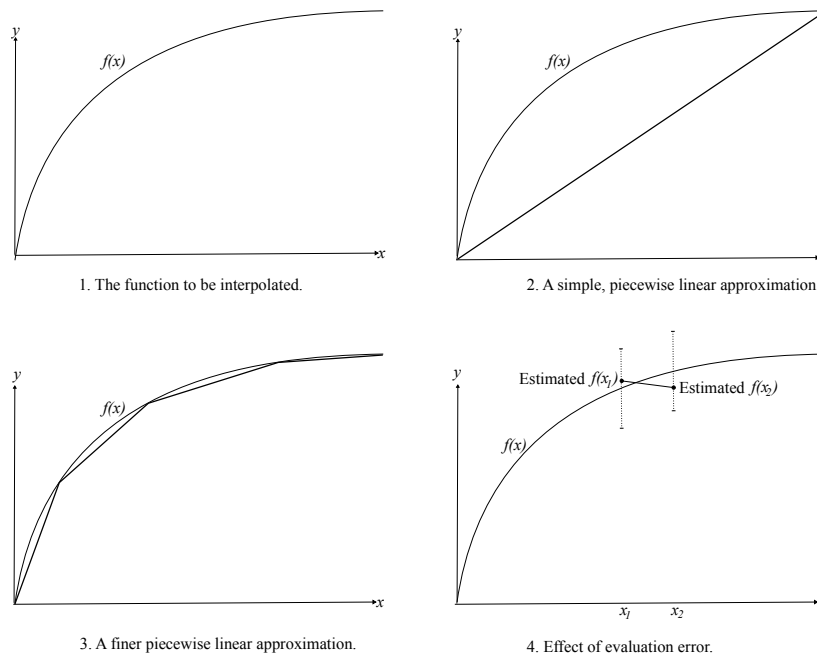


Figure 4: Heuristics and Their Limitations

be easily approximated. The fifth heuristic, of course, just says that the spectrophotometer was engineered to be accurate.

Heuristic 4 is a mathematical statement, rather than an engineering result. It makes the intuitively plausible assertion that a closer look at a function provides more accurate information.

Though these heuristics are useful, they eventually break down, particularly Heuristic 5. Figure 4 shows how the breakdown makes linear interpolation unreliable. For clarity, the function in the figure goes from  $\mathbf{R}$  to  $\mathbf{R}$ , rather than from  $\mathbf{R}^3$  to  $\mathbf{R}^3$ , but the concepts work in either dimension.

Item 1 of Figure 4 shows a sample function  $f(x)$ , from  $\mathbf{R}$  to  $\mathbf{R}$ . By the first heuristic,  $f$  is one-to-one, so it must be monotonic, as drawn. Item 2 shows a one-piece linear approximation. Though crude, this secant line can be used in place of  $f(x)$ . For example, one could interpolate  $f$  at a point  $x$  by evaluating the secant line at  $x$ . One could also approximate the derivative of  $f$  with the slope of the secant line. This approximation would apply to every point of  $f$  shown, so it would be very rough.

The fourth heuristic, illustrated in Item 3, says that the piecewise linear approximation can be improved by using more, and smaller, pieces. The single secant in

Item 2 has been refined, to four secants in Item 3. The new piecewise linear function will provide more accurate interpolations, and better approximations to derivatives. The secant on the far right is such a good fit for  $f$  that further refinement, which requires more computation, might not improve accuracy.

Item 4 illustrates the fifth heuristic, and its limitations. Despite the heuristic,  $f(x_1)$  is only known to within some interval, illustrated as a vertical line over  $x_1$ . In this project,  $x_1$  is an *RGB* triple, and evaluating  $f$  at  $x_1$  means measuring the printed output of that triple. The spectrophotometer’s limited accuracy causes the evaluation error seen in Item 4. In addition to evaluation errors, slopes can be estimated incorrectly. Suppose, for example, that  $f$  has been evaluated at  $x_1$  and  $x_2$ . Because of measurement error, the evaluated  $f(x_1)$  and  $f(x_2)$  are as shown in Item 4. To estimate the derivative of  $f$  between  $x_1$  and  $x_2$ , we would use the slope of the line joining  $f(x_1)$  to  $f(x_2)$ . This slope is negative, however, while the derivative of  $f$  is positive, so  $f$  appears to be decreasing, when in fact it is increasing. The closer together  $x_1$  and  $x_2$  are, the more likely it is that derivatives cannot be estimated well. This fact works against the fourth heuristic, which says that  $f$  itself can be interpolated better, when  $x_1$  and  $x_2$  are closer together. Measurement error, then, limits the algorithm’s accuracy.

#### 4.1.2 Linear Functions on Tetrahedra

The third heuristic allows the algorithm to approximate  $f$  with piecewise linear functions. The “pieces” of the approximating function will be tetrahedra. The approximating function has the linear form

$$L = M_{11}R + M_{12}G + M_{13}B + \delta_L, \tag{3}$$

$$a = M_{21}R + M_{22}G + M_{23}B + \delta_a, \tag{4}$$

$$b = M_{31}R + M_{32}G + M_{33}B + \delta_b, \tag{5}$$

on tetrahedral subsets of the *RGB* domain. In more compact, matrix notation:

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \mathbf{M} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} \delta_L \\ \delta_a \\ \delta_b \end{bmatrix}. \tag{6}$$

Geometrically, each linear piece maps a tetrahedron in the domain to a tetrahedron in the codomain. Each vertex of the domain tetrahedron is mapped to a vertex of the codomain tetrahedron. Similarly, edges are mapped to edges, and faces to faces. In fact, the entire domain interior is mapped linearly to the codomain interior.



Barycentric coordinates<sup>8</sup> simplify computations involving linear functions on tetrahedra. Suppose  $\mathbf{v}_1$  through  $\mathbf{v}_4$ , considered as vectors, are the four vertices of a tetrahedron in  $\mathbf{R}^3$ . Then every point  $\mathbf{u}$  in  $\mathbf{R}^3$  can be written with unique coefficients  $\alpha_1$  through  $\alpha_4$ , as

$$\mathbf{u} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3 + \alpha_4\mathbf{v}_4, \quad (7)$$

provided that we require

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1. \quad (8)$$

The coefficients  $\alpha_i$  are called the *barycentric coordinates* of  $\mathbf{u}$ . They are all non-negative when, and only when,  $\mathbf{u}$  is inside or on the boundary of the tetrahedron.

Now consider a function  $g$  from  $\mathbf{R}^3$  to  $\mathbf{R}^3$ . The images  $g(\mathbf{v}_1)$  through  $g(\mathbf{v}_4)$  are the vertices of a new tetrahedron in the codomain. If  $g$  is linear on the original tetrahedron, and  $\mathbf{u}$  is the point in Equation 7, then

$$g(\mathbf{u}) = \alpha_1g(\mathbf{v}_1) + \alpha_2g(\mathbf{v}_2) + \alpha_3g(\mathbf{v}_3) + \alpha_4g(\mathbf{v}_4). \quad (9)$$

Equation 9 says that  $\mathbf{u}$  and its image  $g(\mathbf{u})$  have the same barycentric coordinates. Equation 9 simplifies computations, by allowing  $g$  to be calculated anywhere on a tetrahedron, knowing only the values of  $g$  on the vertices.

Now that linear functions have been constructed on tetrahedra, a piecewise linear approximation to  $f$  can be constructed by tessellating its domain, the *RGB* cube. A tessellation is a set of tetrahedra, with no overlapping interiors, that exactly cover the *RGB* cube. Although the interiors of any two tetrahedra are disjoint, the tetrahedra can have faces, edges, and vertices in common. Since faces, edges, and vertices have volume 0, almost every point in the cube is in the interior of a unique tetrahedron. If  $V$  is any finite set of points in  $\mathbf{R}^3$ , then it is possible to tessellate their convex hull, such that the set of vertices of all the tessellation's tetrahedra is identical with  $V$ . In the next section, we will choose a lattice of vertices on the *RGB* cube as our set  $V$ , and construct such a tessellation.

## 4.2 Iterative Algorithm

Heuristics and tetrahedra will be used to develop an iterative Munsell printing algorithm. The iterative steps occur in pairs. In the first iteration of a pair, a set of *RGBs* is generated, printed, and measured. In other words, the function  $f$  is evaluated at that set of *RGBs*. The second iteration of a pair uses these evaluations to estimate the *RGB* combination that, when printed, will match a given Munsell

aimpoint. This estimate will be printed and measured. If it matches the aimpoint to within a desired DE threshold, the algorithm terminates. If not, a further pair of iterative steps is used. Data from the most recent iteration will be used to generate an additional set of *RGB*s. After printing and measuring these, another estimate is calculated and printed, and so on. Rather than using just a single aimpoint at a time, the implemented algorithm runs on many aimpoints simultaneously. This section describes the iterative steps in detail.

**Iteration 1.** The first iteration generates a set of *RGB* triples that fills out the gamut in some detail. If available, a shade bank can be used instead. Otherwise, 10 equally spaced points are chosen on the *R* axis, 10 on the *G* axis, and 10 on the *B* axis. Together these form a lattice of 1000 *RGB* triples. The function  $f$  is evaluated at these 1000 *RGB* triples by printing and measuring them, and calculating their *Lab* images. Lattices of more than 1000 points can give more information about  $f$ , but with a correspondingly higher measurement load.

A built-in Octave routine, `deleunayn.m`, tessellates the lattice of 1000 points into a set of tetrahedra. The vertices of each tetrahedron in the tessellation are a subset of the 1000 original points, and the images under  $f$  of all the vertices are known. The function  $f$  transfers not only the *RGB* triples to *Lab* space, but also the tessellation. The image of every tetrahedron in the *RGB* tessellation is another tetrahedron, in a corresponding *Lab* tessellation. This transfer is possible only because  $f$  is one-to-one, which guarantees that the images of two disjoint *RGB* tetrahedra will not intersect in *Lab* space.

The tessellation transfer seems simple, but is in fact rather subtle, and depends on the initial *RGB* lattice. A tricky, yet critical, point is that the tessellation must be performed in the *RGB* domain, rather than in the *Lab* codomain. It cannot be performed in the *Lab* codomain, and then transferred to the *RGB* domain. Figure 5 shows why, using a two-dimensional square for clarity, rather than the three-dimensional *RGB* cube. The left side of the figure shows a tessellation of lattice points in *RGB* space. The right side shows the images of those points under  $f$  in the *Lab* codomain. The two sides are nearly identical, except that  $f(\text{RGB}_4)$  is indented slightly. The solid lines on the right show the transferred tessellation.

Now suppose that a tessellation of the nine *Lab* points had been transferred to the *RGB* domain. The tessellation would include all the solid lines, and in addition the dotted line joining  $f(\text{RGB}_1)$  and  $f(\text{RGB}_7)$ . When this tessellation was transferred back to the domain, the triangle bounded by  $f(\text{RGB}_1)$ ,  $f(\text{RGB}_4)$ , and  $f(\text{RGB}_7)$ , would collapse onto the straight line containing  $\text{RGB}_1$ ,  $\text{RGB}_4$ , and  $\text{RGB}_7$ . This problem can occur whenever  $f$  maps three collinear point to three non-collinear

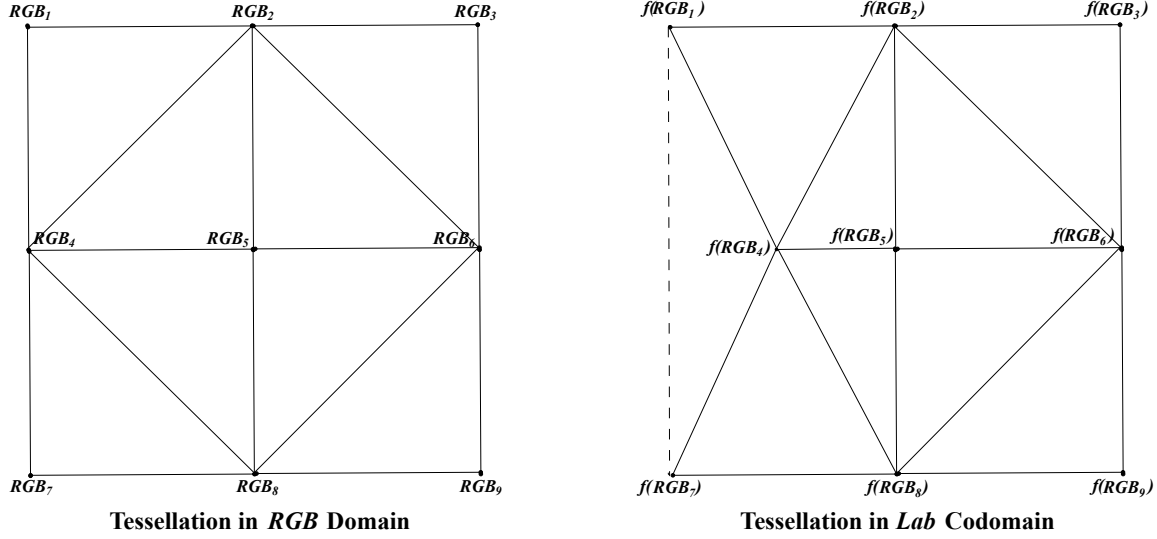


Figure 5: Tessellation Transfer

points: the non-collinear points form a triangle in the tessellation, but the pre-image of the triangle is not a triangle, so there can be no correspondence between the tessellations.

**Iteration 2.** The first iteration tessellated both the *RGB* cube and the *Lab* printer gamut. Furthermore, each tetrahedron in either tessellation maps uniquely to another tetrahedron in the other tessellation. Consider a sample tetrahedron in *RGB* space with vertices  $\mathbf{v}_1$  through  $\mathbf{v}_4$ . It is mapped to the tetrahedron in *Lab* space with vertices  $f(\mathbf{v}_1)$  through  $f(\mathbf{v}_4)$ . For later reference, write  $\mathbf{w}_i = f(\mathbf{v}_i)$ .

Now define a linear function from each *RGB* tetrahedron to the corresponding *Lab* tetrahedron. Let  $\mathbf{u}$  be a point in the *RGB* tetrahedron, whose barycentric coordinates are given by Equation 7. Similarly to Equation 9, define

$$p(\mathbf{u}) = \alpha_1 f(\mathbf{v}_1) + \alpha_2 f(\mathbf{v}_2) + \alpha_3 f(\mathbf{v}_3) + \alpha_4 f(\mathbf{v}_4) \quad (10)$$

$$= \alpha_1 \mathbf{w}_1 + \alpha_2 \mathbf{w}_2 + \alpha_3 \mathbf{w}_3 + \alpha_4 \mathbf{w}_4. \quad (11)$$

The combination of those linear functions, over all tetrahedra, makes  $p$  a piecewise linear approximation for  $f$ , defined on the entire *RGB* cube. The pieces of the approximation are the tetrahedra in the tessellation. The function  $p$  agrees with the function  $f$  on all the vertices.

Let  $\mathbf{m}_j$  be the  $j$ th Munsell aimpoint, in *Lab* coordinates. If  $\mathbf{m}_j$  is outside the *Lab* tessellation, then it is likely outside the gamut, so it cannot be printed. If

the aimpoint is within the gamut, then a unique tetrahedron in the tessellation contains the aimpoint. The Octave routine `tsearchn.m` finds this tetrahedron. Write the vertices of the containing tetrahedron as  $\mathbf{w}_1$  through  $\mathbf{w}_4$ , and use barycentric coordinates to write

$$\mathbf{m}_j = \beta_1 \mathbf{w}_1 + \beta_2 \mathbf{w}_2 + \beta_3 \mathbf{w}_3 + \beta_4 \mathbf{w}_4. \quad (12)$$

We seek an *RGB* triple that will produce  $\mathbf{m}_j$ . The piecewise linear approximation  $p$  associates a unique tetrahedron in *RGB* space to the containing tetrahedron in *Lab* space. A natural candidate is the pre-image,  $p^{-1}(\mathbf{m}_j)$ . Since  $\mathbf{w}_i = f(\mathbf{v}_i)$ , where the  $\mathbf{v}_i$ 's are the vertices of the *RGB* tetrahedron, this pre-image is given by

$$\mathbf{e}_{2j} = p^{-1}(\mathbf{m}_j) = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2 + \beta_3 \mathbf{v}_3 + \beta_4 \mathbf{v}_4. \quad (13)$$

The set of *RGB* triples  $\mathbf{e}_{2j}$ , over all  $j$ , is the outcome of the second iteration. These *RGB* triples are printed and measured. If the colour difference from  $\mathbf{m}_j$  is small enough, then the algorithm terminates for that aimpoint. Otherwise, further iterations are used.

**Iteration 3.** The second iteration has produced an initial *RGB* estimate,  $\mathbf{e}_{2j}$ , of the Munsell aimpoint  $\mathbf{m}_j$ , but the printed colour is often not sufficiently close to the aimpoint. In addition, the first two iterations have associated two tetrahedra, one in *RGB* space and one in *Lab* space, with  $\mathbf{e}_{2j}$ . The third iteration selects further *RGB* triples, near the *RGB* tetrahedron, to allow a more refined estimate,  $\mathbf{e}_{4j}$ , to be made in the fourth iteration.

Figure 6 illustrates the progress that has been made so far, towards  $\mathbf{m}_j$ . The left half of the figure is located in *Lab* space. A triangle is used in place of a tetrahedron for ease of presentation. On the left is the aimpoint  $\mathbf{m}_j$ , contained inside the vertices  $f(\mathbf{v}_i)$  in *Lab* space. Also shown is the image,  $f(\mathbf{e}_{2j})$ , of  $\mathbf{e}_{2j}$ . In the right half of the figure, in *RGB* space,  $f$  and  $p$  both map the vertices on the right to the vertices on the left. The point  $\mathbf{e}_{2j}$  has the same barycentric coordinates as the point  $\mathbf{m}_j$  on the left.

$f(\mathbf{e}_{2j})$  should be in the vicinity of  $\mathbf{m}_j$ , though it is not within the DE threshold. The distance between  $f(\mathbf{e}_{2j})$  and  $\mathbf{m}_j$  in Figure 6 occurs because the linear function  $p$  was used to approximate the non-linear function  $f$ . Though  $p$  and  $f$  agree on the vertices, the linear approximation introduces distortions in the interior. Since the position of  $f(\mathbf{e}_{2j})$  in *Lab* space is to the left of the point  $\mathbf{m}_j$  that we were aiming at, we should correct by aiming at the small triangular region shown in the left of Figure 7.

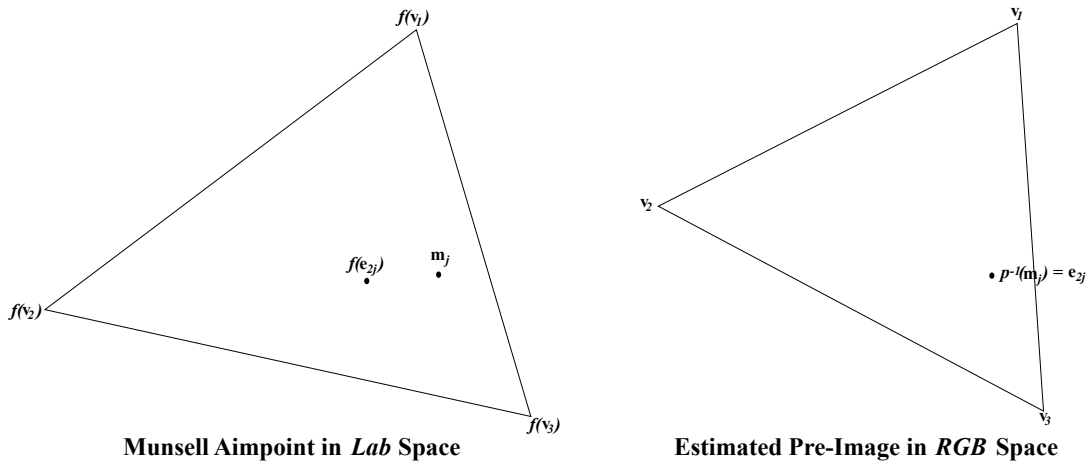


Figure 6: Containing Tetrahedra, in *Lab* and *RGB* Space

We will choose some points which, together with  $f(\mathbf{e}_{2j})$ , create a tetrahedron that bounds the triangular region. Linear interpolation can then zero in on  $\mathbf{m}_j$ . This approach exemplifies Heuristic 4, that smaller tetrahedral pieces should lead to better approximations. Though two new points are used in the figure, the tetrahedral construction actually uses three new points. The right half of Figure 7 shows the construction, pulled back to *RGB* space. The barycentric coordinates of the new points are identical on both the left and the right. The new points, in *RGB* space, are the output of the third iteration. They will be printed and measured, to define the function  $f$  around the aimpoint more closely.

The details of the choice of new points were decided by trial and error. The driving factor is the effect of evaluation error, as seen in Item 4 of Figure 4. The new tetrahedron should contain the Munsell aimpoint  $\mathbf{m}_j$ . If a more accurate spectrophotometer were used, the new tetrahedra could probably be made considerably smaller, which would make the algorithm converge faster. The choice of *Lab* coordinates, over other CIE coordinate systems such as XYZ, is also important here. While not as accurate as DE2000, the Euclidean metric defined by Expression 1 is a helpful approximation. The angles and distances used in determining the new points for Iteration 3 were calculated with this metric. As a result, no pair of vertices of the new tetrahedron should be much closer than any other pair. This factor leads to stability, especially in the presence of measurement error.

**Iteration 4.** The third iteration constructed, for each Munsell aimpoint  $\mathbf{m}_j$ , a tetrahedron of *RGB* vertices whose printed images should be near  $\mathbf{m}_j$ . The fourth

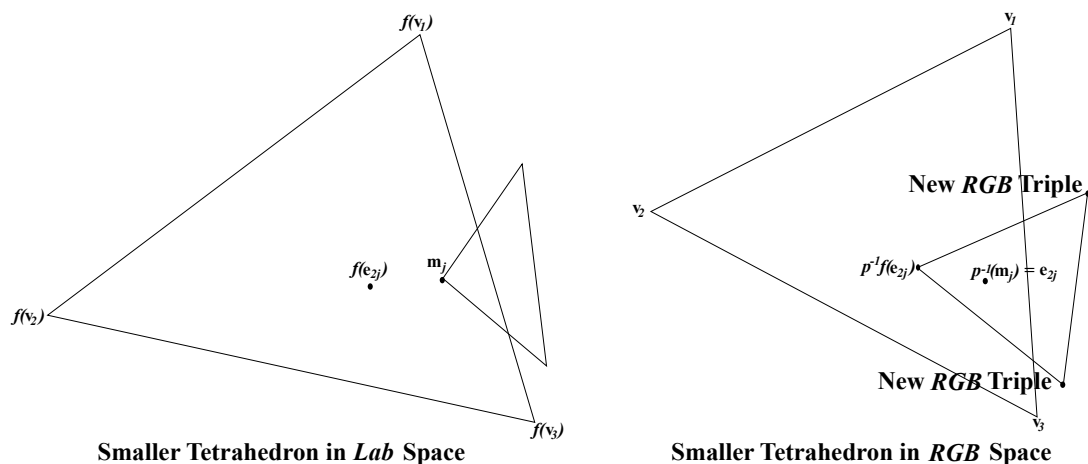


Figure 7: New Points for Iteration 3

iteration is much like the second iteration. Express the *Lab* values for  $\mathbf{m}_j$  in barycentric coordinates, relative to the *Lab* values for the printed images, as was done in Equation 12. Then use Equation 13 to produce a new *RGB* estimate,  $\mathbf{e}_{4j}$ . Print and measure  $\mathbf{e}_{4j}$ . If it is sufficiently near  $\mathbf{m}_j$ , then terminate the algorithm for  $\mathbf{m}_j$ . If not, repeat Iterations 3 and 4 as needed, to refine the estimate sufficiently.

**Final Step.** As the algorithm is run, a shade bank is assembled. The shade bank consists of each *RGB* that was printed, the resulting *Lab*, and the measured reflectance spectrum. Some aimpoints will be outside the gamut, so the iterative steps will not be run on them. When the algorithm is completed, however, the shade bank will be checked to see which *RGB* is closest to a given out-of-gamut aimpoint. It is possible that an *RGB* on the boundary is within the DE threshold of an aimpoint, even though the aimpoint is out of gamut.

### 4.3 Example

Figures 8 through 11 show an example of the algorithm being run. In this example, the DE threshold was set at 2.0. Rather than finding *RGBs* for the entire Munsell book, the only Munsell aimpoints for which *RGBs* were found were the nine neutral greys, and colours of hue 5R. Figure 8 shows Iteration 1, which used a lattice of 1000 evenly spaced *RGBs* as a starting set. The starting colours are printed out on five different sheets, and measured with the ColorMunki. The *RGB* lattice is tessellated tetrahedrally, and the tessellation is transferred to *Lab* space.

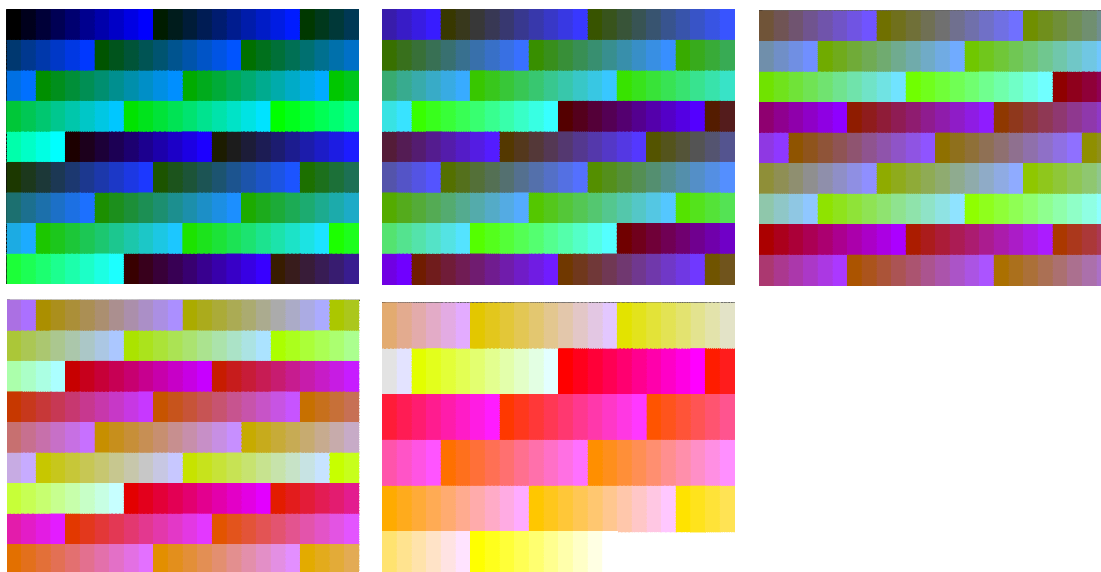


Figure 8: Example of Iteration 1, Using 1000 Colours

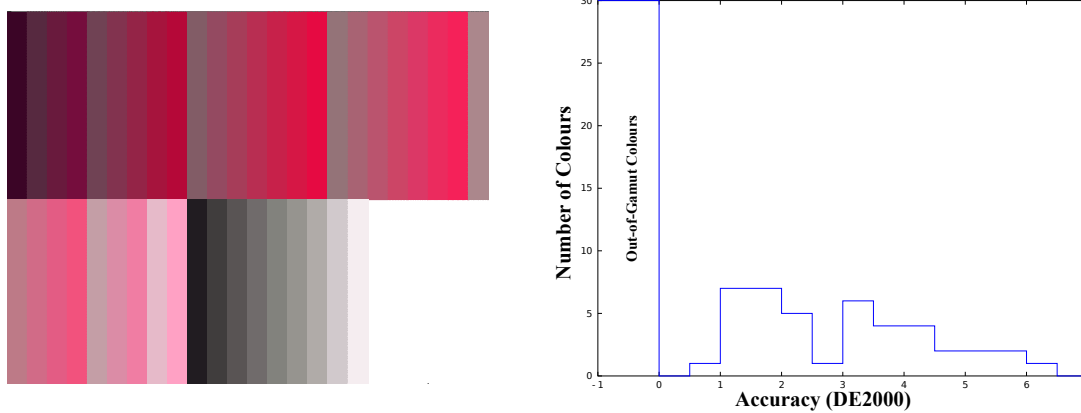


Figure 9: Example of Iteration 2

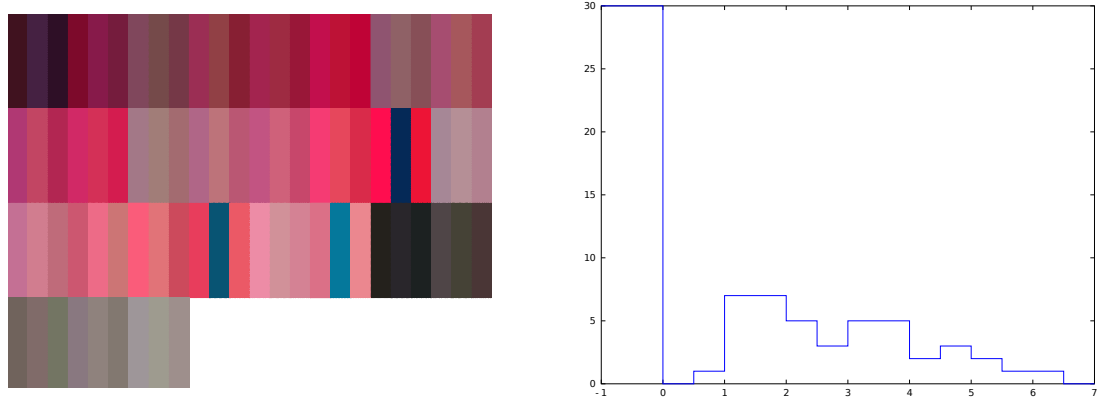


Figure 10: Example of Iteration 3

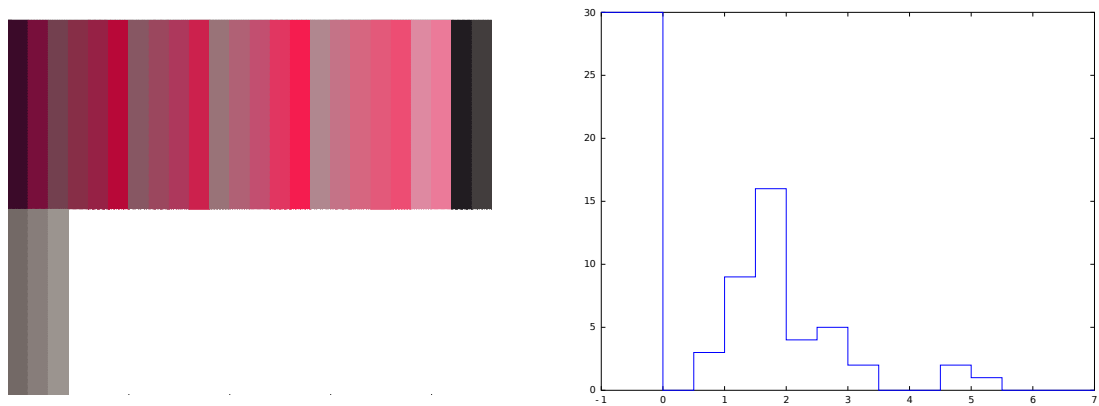


Figure 11: Example of Iteration 4



Iteration 2, shown in Figure 9, interpolates linearly over the transferred tessellation. There are 63 standard Munsell colours of hue 5R. When added to the nine neutral greys, there is a total of 72 possible aimpoints. 30 of those aimpoints are outside the printer gamut. Potential *RGB* values can be calculated for the remaining 42. These *RGB* values are printed out on the left side of the figure. The reds are printed first, from darkest to lightest, followed by the greys, also from darkest to lightest. After measuring the printed *RGBs* on the left, the DEs between the *RGBs* and the aimpoints are calculated, and plotted as a histogram on the right. The horizontal axis of the histogram is DE value. For ease of display, out-of-gamut aimpoints are assigned a negative DE. As a result, the tall bar on the histogram's left represents the 30 out-of-gamut colours. The histogram shows that 15 of the 42 in-gamut colours are already matched to within the DE threshold of 2.0. The algorithm must therefore find *RGBs* for the remaining 27 aimpoints.

In Iteration 3, shown in Figure 10, a new containing tetrahedron is constructed for each of the 27 remaining aimpoints. One vertex of the new tetrahedron is the estimate from Iteration 2, which has already been printed and measured. Each remaining aimpoint requires three new vertices to construct a tetrahedron, so there are 81 (i.e. 27 times 3) new *RGBs*, which are printed out on the left, and measured. Some of those new vertices are nearer the aimpoint than the initial estimate in Iteration 2. Thus, the histogram on the right is slightly different from the histogram in Figure 9. While most of the printed values in Figure 10 have a reddish hue, a few are distinctly blue. If the size of the new tetrahedra were increased, there would be more definitely non-red hues. If the size were decreased, there would be fewer non-red hues, but the tetrahedra would be less likely to contain an aimpoint. The size should be tailored to the accuracy of the spectrophotometer, so that most new tetrahedra contain an aimpoint. In this example, 20 of the 27 new tetrahedra contained the aimpoint.

Iteration 4 interpolates (or extrapolates) linearly over the tetrahedra of Iteration 3, to produce new estimates for the remaining 27 aimpoints. The new estimates are printed, and appear on the left of Figure 11. After measuring them, the new DEs are calculated, and plotted in the histogram on the right. 13 of the 27 aimpoints are now within the DE threshold, while 14 are not. In total, then, accurate *RGBs* have been found for 28 aimpoints, 30 aimpoints are out of gamut, and 14 aimpoints need further investigation. Further iterations will find accurate *RGBs* for these 14 aimpoints.

## 5 Results

### 5.1 Algorithm Output

A Munsell book was produced by running the algorithm. Iteration 1 used a starting set of 8000 *RGB* samples (20 *R* values times 20 *G* values times 20 *B* values). Though the threshold of interest was 2.0 DE, the algorithm was run with a threshold of 0.5 DE, to leave a margin for measurement error. Iteration 2 measured the initial *RGB* estimates, made by linear interpolation on the data in Iteration 1. Of the 2745 standard Munsell colours, about 1400 were in the printer gamut. 1120 of these 1400 showed DEs of 2 or less. Since the algorithm was run with a DE threshold of 0.5, however, most of these 1120 were refined in later iterations, to reduce the DEs further. In all, eight iterations were run, by which time the rate of improvement had slowed to nearly zero. After the iterations, the final step looked for Munsell colours that, while outside the printer gamut, were within 2 DEs of some point on the printer gamut's boundary. These colours were included in the final list, which contained *RGBs* for just over 1500 Munsell colours.

The output of the algorithm is a lookup table of *RGB* specifications for the standard Munsell colours. When printed on Kirkland paper on an Epson R2880 printer, the *RGB* triples should be accurate to within a DE of 2. Figure 12 shows the printed output for the Munsell hue 5R. The empty boxes indicate out-of-gamut colours. Viewed on a computer monitor, even one that is colour-calibrated, the colours are likely inaccurate, because a monitor's colour gamut differs from a printer's gamut. If Figure 12 were printed with an Epson R2880, on Kirkland Glossy Professional Photo Paper, then the colours would be accurate. With a different printer or paper, the colours probably would not be correct. With a cheap printer and office paper, they would be very inaccurate.

Table 3 presents the *RGB* lookup table for the row of Figure 12 with Munsell value 5. The DEs are also given. The DE threshold of 2 is met for lower chroma colours, but not for higher chroma colours. In Table 3, the red of chroma 16 just exceeds the threshold, at a DE of 2.5, while reds of higher chroma are well above it. These higher chroma colours are outside the printer gamut. The *R* value is at its maximum of 1, when the chroma is 18 or 20. At this point, the *RGB* triple is on the boundary of the *RGB* cube, and therefore on the boundary of the printer gamut. The last two *RGBs* are identical. Their common value is the point on the printer gamut's boundary that is nearest to both 5R 5/18 and 5R 5/20.

Table 4 lists six Munsell aimpoints, for which the algorithm failed to achieve a DE of 2. To avoid gaps in the printed pages, the closest *RGB* was used, even though

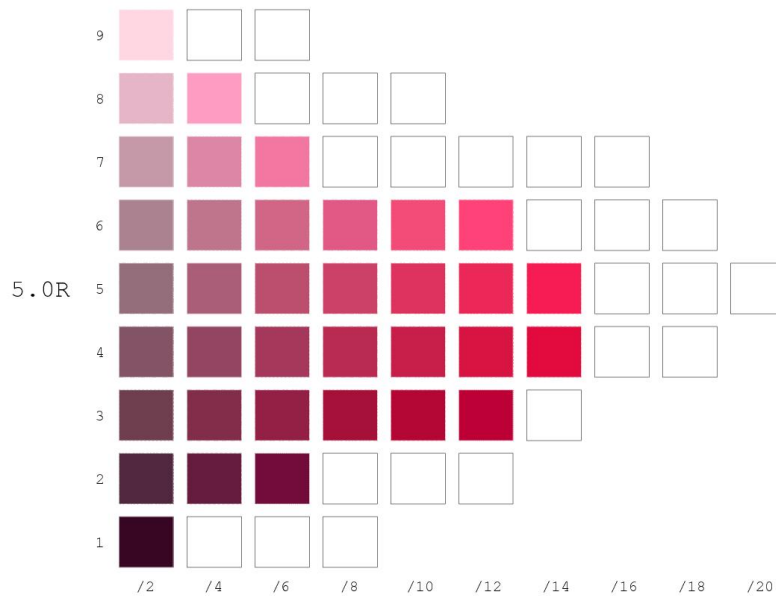


Figure 12: Sample Munsell Colours, for Hue 5R

it exceeded the threshold. The table shows that the DEs are not much greater than 2, and at worst about 3.

Figure 13 shows a histogram of the DEs that the printed book attained. In all, 1530 standard Munsell colours were reproduced. The average and median DE were both about 0.7. These seemingly excellent accuracies should be taken with a grain of salt. Several *RGB* estimates were made for each Munsell aimpoint. When repeated measurements are made, and the measuring device shows some variability, a printed sample's DE will occasionally be very low, just because of random variation. Nevertheless, even if there is a measurement error of 1 DE, for every measurement in Figure 13, the DEs would still be almost all less than 2. The next section shows that this accuracy compares favorably with other Munsell products.

Analysis of the algorithm as it ran showed that the limiting factor was measurement error. Eventually, measurement error causes the algorithm estimates to stop improving. For example, Iterations 2, 4, 6, 8, etc., provide successive *RGB* estimates for a given aimpoint. The DE for Iteration 4 should be less than the DE for Iteration 2, Iteration 6 should be less than Iteration 4, and so on. The DE decreased steadily between Iteration 2 and Iteration 4, but less steadily after that, and at times the DE actually increased. This stall is likely to occur when the measurement error is on par with the quantity being measured. When the measurement error is 1 DE, for

Munsell Aimpoint	$R$	$G$	$B$	DE
5R 5/2	0.578	0.431	0.480	0.3
5R 5/4	0.671	0.368	0.474	1.6
5R 5/6	0.736	0.309	0.433	0.7
5R 5/8	0.806	0.265	0.404	0.6
5R 5/10	0.868	0.205	0.372	0.9
5R 5/12	0.921	0.158	0.342	0.4
5R 5/14	0.969	0.105	0.328	0.4
5R 5/16	0.974	0.105	0.316	2.5
5R 5/18	1.000	0.053	0.263	3.8
5R 5/20	1.000	0.053	0.263	5.3

Table 3: Examples of  $RGB$ s for Munsell Aimpoints of Hue 5R and Value 5

Munsell Aimpoint	DE
2.5Y 6/2	2.1
2.5R 3/4	2.2
5.0BG 3/8	2.2
5.0YR 8/2	2.3
2.5PB 4/6	2.8
7.5BG 4/6	3.1

Table 4: Munsell Aimpoints With DE  $> 2$

example, a difference of less than 1 DE cannot be readily measured.

A similar sign of algorithm breakdown occurs in the odd-numbered iterations. As shown in Figure 7, the vertices of a new  $RGB$  tetrahedron are calculated, printed, and measured. The result is a tetrahedron in  $Lab$  space, which should contain the Munsell aimpoint. In Iterations 3 and 5, the new tetrahedra did contain the aimpoints. In later iterations, however, the aimpoints were often outside the tetrahedra. The likely cause of this breakdown is measurement errors, which lead to three-dimensional versions of the effect seen in Item 4 of Figure 4. If one vertex of the tetrahedron is displaced by 1 DE, when every vertex of the tetrahedron is within 1 or 2 DEs of the aimpoint, then it is easy to see how non-containment occurs, and why the algorithm no longer improves its estimates.

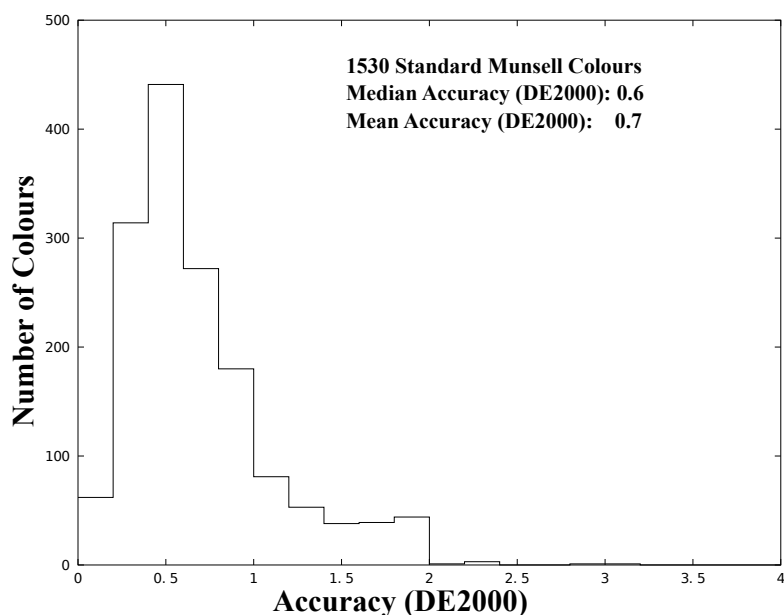


Figure 13: Histogram of Accuracies of Algorithm Output

## 5.2 Accuracy of Other Munsell Books

The accuracies attained in this project were found to be at least on par with two other Munsell books.<sup>1,2</sup> The first Munsell book is a student colour set, and the second is the 2007 edition of the “Big Book of Color.” The two books were measured with a ColorMunki. A third possible comparison source<sup>9,10,11</sup> consists of publicly available reflectance spectra, measured at Finland’s Joensuu University, of the Munsell Company’s 1976 book. An analysis showed that reliable accuracies could not be calculated from these measurements, so the results are not included here.

The books contain both standard and non-standard Munsell colours. A Munsell colour is considered standard if it appears in the Munsell renotation data. Standard colours have integer Munsell values, even Munsell chromas, and hues prefixed with 0.0, 2.5, 5.0, 7.5, or 10.0. Non-standard colours are interpolations between the standard colours. For example, the second book contains many colour samples with chroma 1; since 1 is not an even number, these samples were discarded as non-standard.

A weak point in this analysis is that measurements were made under non-laboratory conditions. The measurement protocol, rather than the sample itself, could be causing a large part of the colour difference between the sample and its renotation aim-

point. Since ColorMunki measurements can easily vary by 0.5 DE or more, the errors in the previous Munsell books are likely to have been overestimated. Similarly, random effects of the printing algorithm likely lead to underestimates of the errors of the current project. Therefore, even though (as will be seen) the average errors in this project were about 1 DE less than the average error in previous books, much of this difference could be illusory. Nevertheless, an average difference of 1 DE is enough to conclude that this project's accuracies are on par with previous Munsell products.

Laboratory measurements of Munsell books would increase users' confidence in them, and provide standards for producers. Researchers who have made controlled measurements of Munsell products are invited to send their reflectance spectra to the author. A detailed analysis, similar to those in the following sections, can be made and shared.

**The New Munsell Student Color Set, 2<sup>nd</sup> ed.** This Munsell set consists of swatches that a student sorts himself, and glues onto Munsell sheets. This book is affordable, but limited in its gamut. Only hues prefixed with 5.0 are presented, and the chromas stay near the neutral axis. Figure 14 is a histogram of the differences (calculated with respect to DE2000) between the printed samples and their renotation aimpoints. Out of 235 printed colours, the median DE was 1.6, and the mean DE was 1.8. A very large DE of 10.2, occurs for the colour N9, which is printed far too light. Removing this point makes the mean error slightly less.

Besides the overall colour difference, the individual Munsell components (hue, value, and chroma) were also analyzed, to identify possible biases. To investigate the individual components, the CIE coordinates of each sample were converted to the Munsell system.<sup>12</sup> Then the aimpoint's hue, value, or chroma, was subtracted from the sample's hue, value, or chroma, to measure the shift. A positive value shift means that the sample is lighter than the aimpoint, while a negative value shift means the sample is darker. Similarly, a chroma shift is positive when the sample is more saturated than the aimpoint, and negative when it is less saturated. By convention, hue shift is positive when red shifts toward yellow, yellow toward green, and so on around the hue circle; it is negative when green shifts toward yellow, yellow, toward red, and so on. When the aimpoint was a neutral colour, no hue shift was calculated.

The median value shift was 0.01 Munsell value steps, indicating that the samples were not consistently lighter or darker than their aimpoints. Even though the signed shift is small, the absolute shift could be large, which would indicate high inaccuracy. In fact, the median absolute value shift was only 0.05, which is well below human perceptibility. The median chroma shift was -0.02, and the median absolute chroma

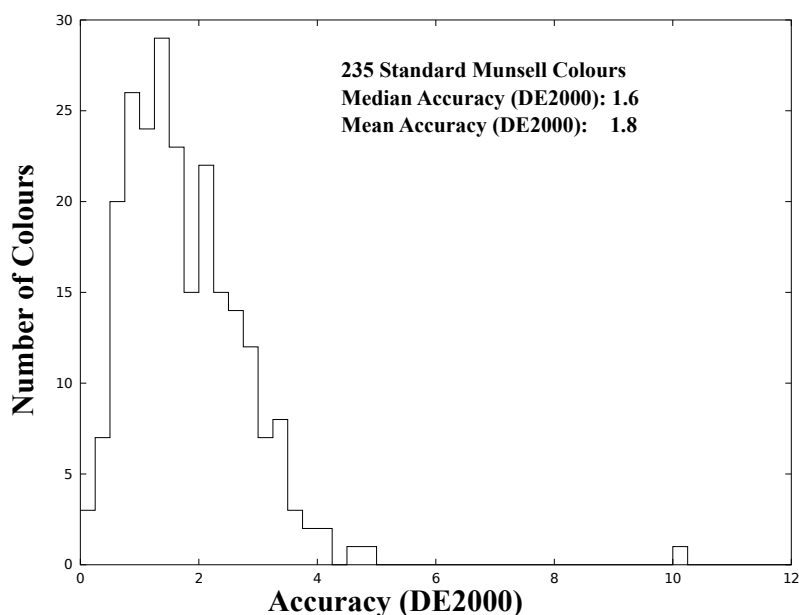


Figure 14: Histogram of Accuracies for *The New Munsell Student Color Set, 2<sup>nd</sup> ed.*

shift was 0.11, indicating likewise that chroma shifts are unbiased, and usually imperceptible. The median hue shift was -0.24, and the median absolute hue shift was 0.38. The distance between two adjacent standard hues is 2.5 Munsell hue units, so the hue shifts are also unbiased and minor. As a general conclusion, then, the errors in *The New Munsell Student Color Set* are unbiased in all three Munsell components, and no component is particularly inaccurate.

Other possible correlations were also evaluated. Figure 15 shows one correlation that occurred consistently across all the books examined: hue shifts tend to be larger for duller colours than for higher chroma colours. As can be seen in the figure, colours with chromas of 2 or 4 were associated with hue shifts of up to 4. Colours with chromas above 8, on the other hand, were rarely associated with a hue shift greater than 1. It is not clear why this result occurs. It was seen, again in all the books examined, that there was no tendency for colours of certain hues or values to produce higher DEs than other hues or values.

**X-Rite’s “Big Book of Color,” 2007.** For many years, X-Rite has produced its “Big Book of Color,” a Munsell book that is an industry standard. An extensive set of inks is used, to attain a wide gamut. More recent versions have over 1600 removable printed samples. In the 2007 book, 1301 were standard Munsell colours.

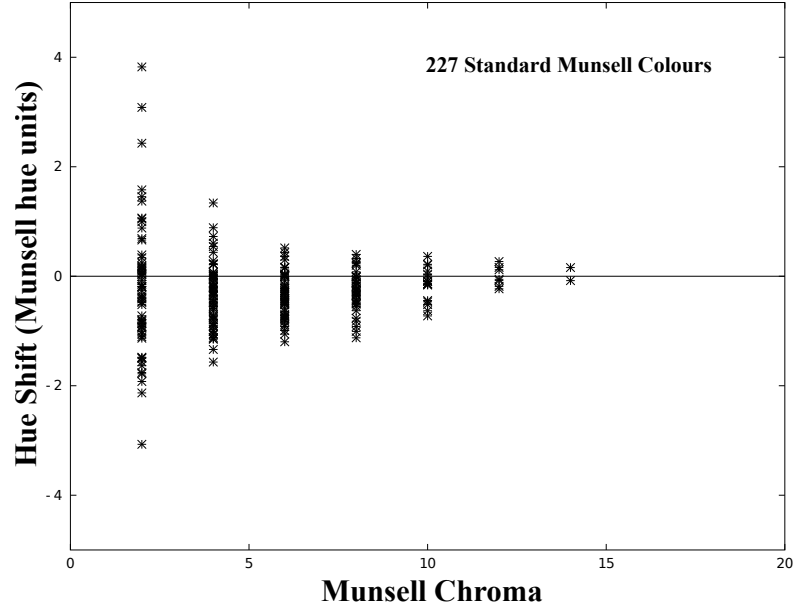


Figure 15: Chroma vs. Hue Shift in *The New Munsell Student Color Set*, 2<sup>nd</sup> ed.

Their median DE, when compared to renotation aimpoints, was 1.9, and the mean DE was 2.1. Overall, then, this Munsell book achieved a DE accuracy of about 2.0, not much different from *The New Munsell Student Color Set*.

To test for bias, statistics were calculated for the individual Munsell components. The median value shift was 0.10 Munsell value units, and the median absolute shift was 0.11 units. The median chroma shift was 0.18 units, and the median absolute shift was 0.19 units. The median hue shift was -0.12 units, and the median absolute hue shift was 0.34 units. Like *The New Munsell Student Color Set*, all these differences are near enough to 0 to conclude that there is no consistent bias in any individual component, and no component is particularly inaccurate.

### 5.3 Limitations

The Munsell book produced by this project has met the goals of accuracy, affordability, a wide gamut, and home production, but it also has some limitations that other Munsell books might not:

**Limited gamut.** The Munsell colours produced are limited by the printer and the inks used. It is generally difficult to produce colours of high chroma, or colours



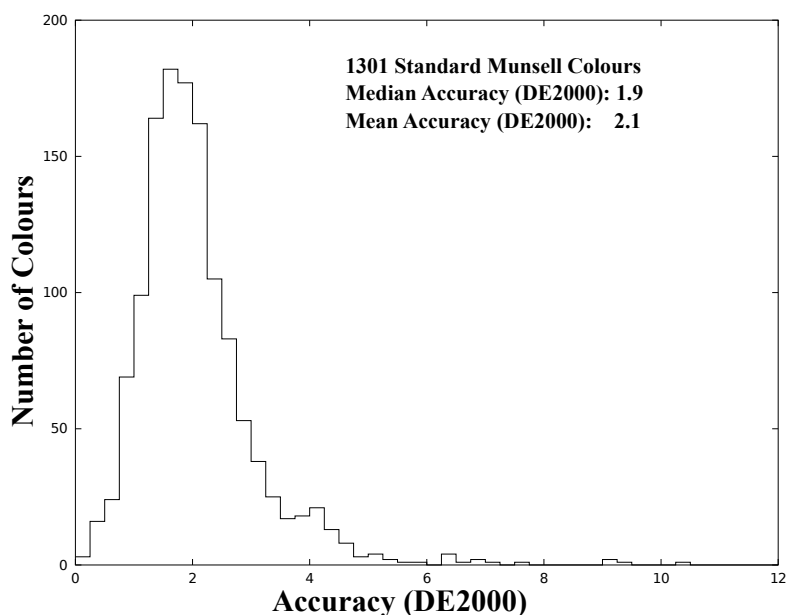


Figure 16: Histogram of Accuracies for the 2007 “Big Book of Color”

of values 1 or 9. This shortcoming is particularly evident in the light oranges (hue 5YR), where chromas stay very near the neutral axis. More sophisticated printing systems, such as Hexachrome, use an orange ink to overcome this limitation.

**Variability.** Consistently printing the Munsell book requires that the printer, inks, and paper, do not change over time. In fact, of course, manufacturing variations, or changes in the manufactured items over time, can cause the output Munsell colours to vary. Even without changes in the physical objects, production conditions, such as humidity when printing, can cause variability. These factors are particularly difficult to control in a home environment.

**Non-removable samples.** Some Munsell books feature removable swatches that can be held directly against a colour of interest, for visual comparison. To keep costs down, the Munsell samples in this book are not removable.

**Samples cannot be painted on.** A painter can visually evaluate the difference between a Munsell aimpoint, and a colour he has mixed, simply by dabbing some paint directly on the aimpoint. If the sample is coated appropriately, then the paint can be wiped off, without harming the sample. The solvents in artist’s paints will harm most papers, including the paper used here. As a result, this project does not

allow this method of comparison. A suggested alternative is to paint the mixture of interest on the edge of a piece of canvas paper, and hold that edge directly against the sample.

**Unknown permanence.** Any printed colours, including the Munsell colours printed for this project, have an uncertain permanence, especially when used as a working tool. It is likely that the colours will fade sooner or later, or otherwise alter. Testing every few years would be needed to see how long the colours maintain the desired DE.

## References

1. Jim Long & Joy Turner Luke, *The New Munsell Student Color Set*, 2<sup>nd</sup> ed., Fairchild Publications, New York, 2001.
2. X-Rite, Munsell Book of Colors, Glossy Collection, 2007.
3. Gaurav Sharma, Wencheng Wu, & Edul N. Dalal, "The CIEDE2000 Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations," *COLOR Research and Application*, Vol. 30, Number 1, February 2005, pp. 21-30.
4. Sidney Newhall, Dorothy Nickerson, & Deane B. Judd. "Final Report of the O. S. A. Subcommittee on the Spacing of the Munsell Colors," *JOSA*, Vol. 33, Issue 7, 1943, pp. 385-418.
5. Bruce Fraser, Chris Murphy, and Fred Bunting, *Real World Color Management: Industrial-Strength Production Techniques*, 2<sup>nd</sup> ed., Peachpit Press, Berkely, CA, 2005, pp. 39-40.
6. ASTM Standard D 1535-08. "Standard Practice for Specifying Color by the Munsell System," ASTM International, West Conshohocken, PA, 2008.
7. X-Rite. "X-Rite ColorMunki Measured Material," informational brochure, 2010.
8. Steven R. Lay. *Convex Sets and Their Applications*, Dover Publications, 2007, Sect. 1.2.
9. [http://cs.joensuu.fi/spectral/databases/download/munsell\\_aotf.htm](http://cs.joensuu.fi/spectral/databases/download/munsell_aotf.htm), accessed October 30, 2012.
10. Oili Kohonen, Jussi Parkkinen, & Timo Jääskeläinen, "Databases for Spectral Color Science," *COLOR Research and Application*, Vol. 31, Number 5, October 2006, pp. 381-390.
11. J. P. S. Parkkinen, J. Hallikainen, & T. Jääskeläinen, "Characteristic spectra of Munsell colors," *JOSA A*, Vol. 6, Issue 2, 1989, pp. 318-322.

12. Paul Centore, “An Open-Source Inversion Algorithm for the Munsell Renotation,”  
COLOR Research and Application, Vol. 37, No. 6, December 2012, pp. 455-464.