AN OPTIMIZATION TECHNIQUE FOR PAINT MIXTURES

Zach Buckner
University of Virginia
Department of Systems and Information Engineering
P.O. Box 400747
151 Engineer's Way
Charlottesville, VA 22904
buckner@virginia.edu

ABSTRACT

A technique is presented to optimize the formulation of paint mixtures. A model approximates the complex mixture process and a novel optimization algorithm is described which, unlike existing high-dimensional mixture optimization algorithms, incrementally and greedily adds new mixture components. The technique is effective and applicable to other mixture optimization problems. The unique algorithm allows the user to limit the total number of mixture components.

1 INTRODUCTION

The complex process of mixing paints has long eluded scientists and artists. Difficulties arise because paint pigments mix together in complex and unpredictable ways. Classical tools like the color wheel are useful in approximating color mixture, but offer little precision. While commercial tools for paint formulation exist, they are undocumented and known to require large, exhaustive databases of empirical paint mixture results. This paper presents a solution to this complex problem requiring minimal empirical mixture data.

The problem is a *mixture* or *composition* optimization problem. Because the number of component palette colors is typically high (up to 100 colors), the search space is high-dimensional. Also, it is desirable to limit the number of component palette colors for a given mixture to reducing the time required to physically mix the colors. These two characteristics warranted the development of a new optimization algorithm, as existing algorithms perform poorly with very high dimensions and are unable to effectively constrain the number of mixture components.

The new algorithm incrementally and greedily adds mixture components. The optimization algorithm also takes advantage of certain characteristics of the solution space.

2 SEARCH SPACE

The search space is *n* dimensional, each dimension representing a mixture component / palette color. The value along each dimension represents the fraction of the given palette color that will be incorporated into the solution. Thus,

$$\forall_{1 \le i \le n}, 0 \le x_i \le 1 \quad (1)$$

, where x_i is the value of the solution along dimension i. The space of solutions is constrained such that sum of all component fractions equals one:

$$\sum_{i=1}^{n} x_n = 1 \quad (2)$$

Thus, each solution $\overline{x}$ represents a recipe for mixing palette colors.

3 COLOR SPACE

The search space is abstract and far removed from the appreciable aspects of *color*. Thus, a different space is used to describe the palette colors, the desired color, and the resultant mixture colors.

The Munsell color system was chosen for this space. This system places colors in a three dimensional space defined by cylindrical coordinates *hue*, *chroma*, and *value*. Hue represents the color (red, orange, etc), value represents the lightness, and chroma represents the brilliance of the color. The following diagram shows the orientation of this space. Hue is the angular axis, chroma is the distance from the cylinder's main axis, and value is the z-axis coordinate:

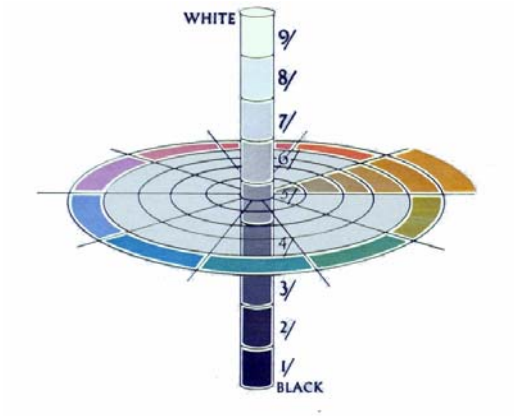

Figure 1: Munsell Color Space

Several other choices for the color space exist, including the more rigorously defined CIELAB color system, but the Munsell system was chosen due to the availability of the useful data and experimental results from paint manufacturers.

4 MAPPING SEARCH SPACE TO COLOR SPACE

A point within the search space represents a mixture recipe; a point within the color space represents a color. Accurately translating a point from the search space to the color space requires the user to physically mix paints and use colorimetric tools to measure the resultant color's Munsell properties.

However, since formulating and characterizing an actual paint mixture is time consuming, an empirical model of paint mixture is used during the optimization process. This model approximates the resultant color from a mixture solution. It is also used to generate the score for each optimization iteration.

4.1 Notation

The following notation is used: *paint1* and *paint2* refer to the 2 component colors within a mixture. These colors are ordered triples with

paint1 = (hue1, value1, chroma1)
paint2 = (hue2, value2, chroma2) (3)

r refers to the fraction of *paint1*'s mass within the mixture. Accordingly, (1-*r*) represents the fractional mass of *paint2* within the mixture. Finally, the function *mix(paint1, paint2, r)* refers to the resulting color from model's approximation of the mixing *paint1* and *paint2* with the desired fraction *r*. This notation can be used to describe the mixture of more than 2 colors, by nesting *mix*() functions.

4.2 Mixture Approximation

When two paint colors are physically mixed, the resulting color varies along a continuous path between the two starting colors' positions within the color space. The following diagram shows the actual color paths for several combinations of common artists paints:

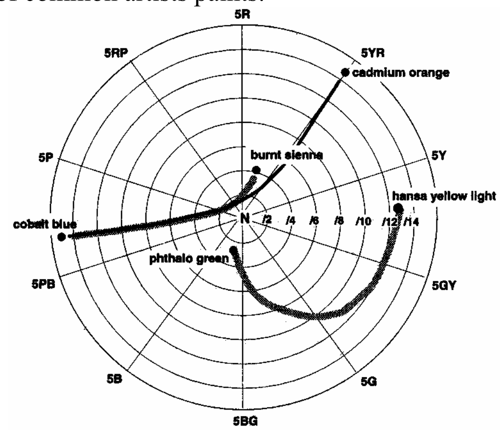

Figure 2: Empirical Color Paths (Long 80)

The figure shows that the paths bend toward the z axis, toward values with lower chroma. As a first order approximation of these paths, a linear interpolation is used. Note that this operation is performed in Cartesian (x, y, z) coordinates

p1 = munsellToCartesian(paint1)
p2 = munsellToCartesian(paint2)
out = p1 + (p2 - p1)*(1 - r1); (4)

To better approximate the curved character of the mixture color paths, the following operations bend the path toward the origin:

o = -1*(p1 + (p2 - p1)/2)
p = (-4*(r1-.5)^2 + 1)
d = cdiff(c1,c2)
out = out + o*p*d*adjust
out = cartesianToMunsell(outc) (5)

The vector o points from the center of the linear interpolation to the origin. This vector defines the direction to bend. The scalar p is the bending coefficient. The bending coefficient defines the degree to which the linear interpolation should be bent. It is a quadratic function of r:

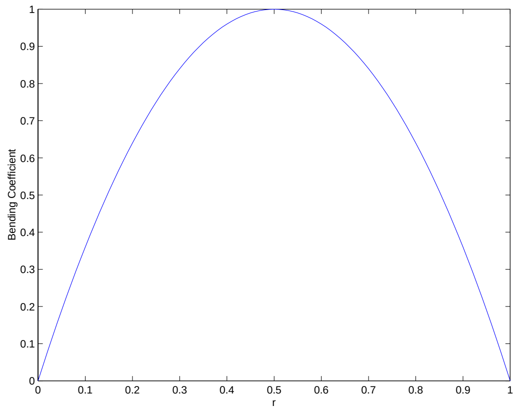

Figure 3: Bending Coefficient Function

The scalar *d* represents the L2 distance from p1 to p2. This factor makes the overall degree of bending a function of distance between the two component palette colors. The scalar constant *adjust* is an user-defined adjustment parameter. It is arbitrarily set to match the empirical mixture samples. An *adjust* factor of .0333 was found to optimal.

The following diagram shows the approximate color path from mixing the same paints that were demonstrated in Figure 2:

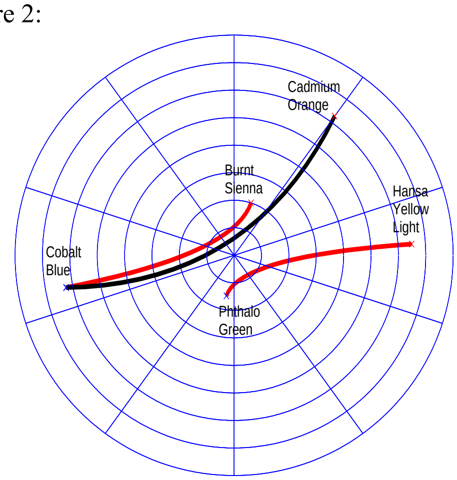

Figure 4: Approximate Mixture Results

As can be seen from comparison with Figure 2, the approximate mixture model is reasonably accurate. It correctly approximates both the Blue/Orange and Blue/Sienna mixtures. However, the model misjudges in the approximate the Green / Yellow mixture. This is because Hansa Yellow is a *strong* color. Future improvements to the model could be made by factoring color strength into the approximation.

5 PALETTE COLORS

Palette colors are the base components to a mixture. The number and coordinates of the palette colors vary by application. As a starting point for analysis, 91 palette colors were chosen. These colors cover the entire product line of acrylic paint colors sold by Golden Paint Company. The Munsell color coordinates for each paint is available on the company's website.

The following diagram shows a projection of each paint color onto the Munsell hue/chroma plane. Each concentric ring represents a step of 2 chroma units.

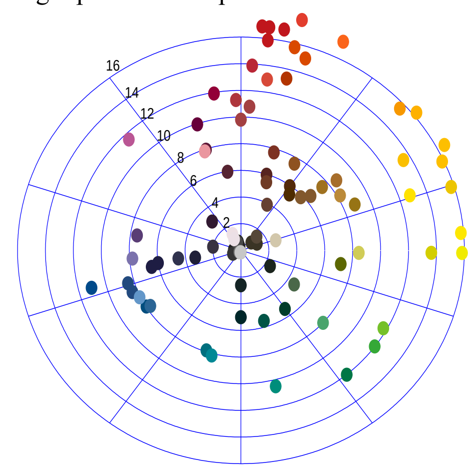

Figure 5: Palette Colors

The set *PALETTE* is used hereafter to refer to the set of ordered Munsell triples representing the palette colors.

6 SCORE FUNCTION

The score function for candidate solutions is based on the L2 distance in the Munsell color space between the desired color D and resultant mixture color. This requires that each both the candidate and desired color be converted to Cartesian coordinates:

$$[x1,y1,z1] = \text{munsellToCartesian}(D); \quad (6)$$

$$[x2,y2,x2] = \text{munsellToCartesian}(C); \quad (7)$$

Thus, the score function is:

$$\text{score} = \sqrt{(x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2} \quad (8)$$

7 OPTIMIZATION ALGORITHM

The problem is a *mixture* or *composition* optimization problem. This class of problems is well studied and several optimization algorithms exist. However, a new optimization algorithm was developed that offers two key advantages to traditional mixture optimization algorithms:

- It can effectively constrain the total number of mixture components. This can reduce the time required to formulate the mixture.
- It takes advantages of certain characteristics of the paint mixture response surface. This greatly simplifies and speeds up the optimization process.

The algorithm incrementally and greedily adds mixture components. The steps of the algorithm are as follows:

7.1 Initial Component Selection

First, the best paints from the *PALETTE* set are selected. This set, called *INIT* and of cardinality *k,* contains the palette colors that have the lowest score function. For the simple score function defined earlier, this selection is simply the closest k component colors in color space.

$$\begin{aligned} &\text{INIT} \subseteq \text{PALETTE such that} \\ &\forall \text{in} \in \text{INIT}, \\ &\forall \text{out} \notin \text{INIT}, \\ &\text{score(in)} < \text{score(out)} \end{aligned} \quad (9)$$

7.2 Optimal Partner Selection

The goal of this stage is to find the best partner palette color for each color $C \in \text{INIT}$. The partner is chosen such that the distance from the desired color D to the line segment that joins C and partner(C) is minimal.

The rationale for this choice is the based on the assumption that that the resultant path of colors that arise from mixtures of two colors can be approximated by a line segment that joins the colors. This assumption is reasonable for two reasons:

1. The color path must, by definition, start and stop at the two palette colors (100% blue must yield blue).
2. Although the entire path of colors that exist between two palette color is unlikely to be linear, it is a reasonable approximation in the vicinity of a palette color. This requirement is met by only considering pairs that contain a color in the INIT set.

For example, consider a simple palette consisting of 4 colors. These palette colors are represented in the following diagram as circled '*p*'s. The desired color is represented as the letter D:

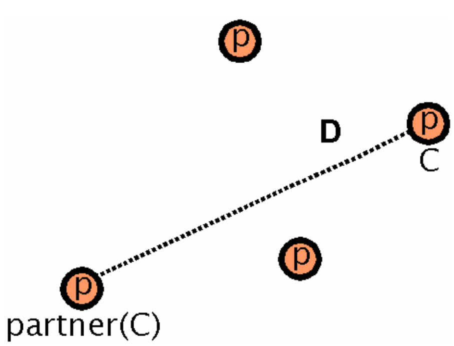

Figure 6: Best choice of *partner(C)* for a given color *C*

In this example, the palette colored labeled *C* will certainly be in the *INIT* set, as it is the closest palette color to the desired color. For this color C, the partner is labeled *partner(C)*. No other palette color except partner(C) creates a joining line segments that is closer to the desired color D.

This selection process can be defined formally as follows. Let the function dist(X,Y) be defined as the distance from the desired color D to the segment that joins colors X and Y. Thus:

$$\begin{aligned} &C \in \text{INIT}, \\ &\text{partner}(C) \in \text{PALETTE s.t.} \\ &\forall X \in \text{PALETTE}, \\ &\text{dist}(C, \text{partner}(C)) \le \text{dist}(C, X) \end{aligned} \quad (10)$$

7.3 Pair-Segment Optimization

Let PAIRS be the set of all pairs (C,partner(C)) where C is a color in the INIT set and partner(C) is the partner as defined in the previous stage.

$$\text{PAIRS} = \{(C, \text{partner}(C)) \mid C \in \text{INIT}\} \quad (11)$$

Each pair represents a set of two colors that can be mixed to approximate the desire color. To find the optimal mixture ratio for each pair, the Golden Section search algorithm is used with parabolic interpolation. This algorithm returns only a *local* minimum but this behavior is sufficient.

The algorithm must optimize the variable r subject to the constraint that $0 \le r \le 1$. The optimization seeks to minimize the score function:

$$\text{score}(D, \text{mix}(C1, C2, r)) \quad (12)$$

Let $r' = \text{opt}(P)$ be the optimal value of r for the given color pair P, and let P1 and P2 be the first and second colors within the pair P. Formally:

$$\begin{aligned} &r' = \text{opt}(P) \text{ iff} \\ &\forall 0 \le x \le 1, \\ &\text{score}(D, \text{mix}(P_1, P_2, r')) \le \text{score}(D, \text{mix}(P_1, P_2, x)) \end{aligned} \quad (13)$$

Finally, let *BestPair* be the best performing pair from the previous optimization operations and *BestR* be the corresponding mix ratio. Thus:

$$\begin{aligned} &\text{BestPair, BestR such that} \\ &\text{BestR} = \text{opt(BestPair)}, \text{ and such that} \\ &\forall P \in \text{PAIRS}, \\ &\text{BestScore} \le \text{PScore}, \text{ where} \\ &\text{BestScore} = \text{score}(D, \text{mix}(\text{BestPair}_1, \text{BestPair}_2, \text{BestR})), \\ &\text{PScore} = \text{score}(D, \text{mix}(P_1, P_2, \text{opt}(P))) \end{aligned}$$

Thus, the solution for the first pass of the optimization is to mix mass fraction *BestR* of *BestPair*$_1$ paint with mass fraction *(1-BestR)* of paint *BestPair*$_2$. The resulting color and its score are :

$$\text{resultColor} = \text{mix}(\text{BestPair}_1, \text{BestPair}_2, \text{BestR}) \quad (15)$$

$$\text{resultScore} = \text{score}(\text{mix}(D, \text{mix}(C1, C2, r))) \quad (16)$$

7.4 Subsequent Component Additions

The preceding steps will find an optimal two-color mixture for a desired color D. To further reduce the score of the resultant mixture, additional colors may be greedily incorporated. This is performed by performing additional iterations of the preceding stages, but with the INIT set redefined to contain only the *resultColor* after each iteration of the Pair-Segment Optimization stage.

8 RESULTS AND CONCLUSION

The following image shows a painting containing over 200 colors. This represents a collection of desired colors:

Figure 7: Painting with 200 Desired Colors

The following figure shows the result of substituting each color in the original image with an optimized mixture of two palette colors.

Figure 8: Painting with 2-Color Mixture Substitutes

The colors in this image greatly resemble the originals. By allowing the algorithm to use additional paint colors, color accuracy increase. Future improvements and explorations in this areas will include:

- Comparison of the novel optimization algorithm to more traditional, simplex-based algorithms
- Performing image color substitution tests with more than two component palette colors.
- Investigation of different score functions, including those that reduce the dependence on hue and chroma when value is exceptionally high or low. This will allow the algorithm to better match very dark and very light colors.

REFERENCES

Cornell, J. A. *Experiments with Mixtures.* John Wiley & Sons, New York, 1990.

Long, Jim. *The New Munsell Student Color Set.* Fairchild Books, 2001.

Munsell Notation Listings GOLDEN Heavy Body Acrylics. Available online via <www.goldenpaints.com/munsell.htm> [accessed December 7, 2003].