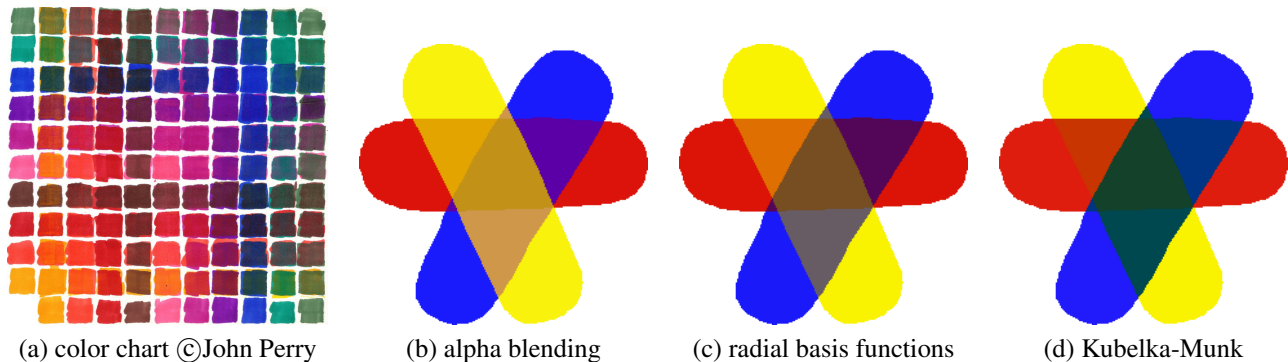


# RealPigment: Paint Compositing by Example

Jingwan Lu<sup>1,2</sup> Stephen DiVerdi<sup>3</sup> Willa A. Chen<sup>1</sup> Connelly Barnes<sup>4</sup> Adam Finkelstein<sup>1</sup>  
<sup>1</sup>Princeton University <sup>2</sup>Adobe Research <sup>3</sup>Google <sup>4</sup>University of Virginia



**Figure 1:** Color compositing: (a) a color chart crafted in marker in order to observe how different pigments appear when layered; (b) alpha blending, though standard in computer graphics, fails to capture known effects like yellow over blue gives green; (c) radial basis functions interpolate and extrapolate from the data in the chart to produce more realistic effects; (d) fitting parameters of a Kubelka-Munk model to the data in the chart improves on radial basis functions for paint-like behaviors.

## Abstract

The color of composited pigments in digital painting is generally computed one of two ways: either alpha blending in RGB, or the Kubelka-Munk equation (KM). The former fails to reproduce paint like appearances, while the latter is difficult to use. We present a data-driven pigment model that reproduces arbitrary compositing behavior by interpolating sparse samples in a high dimensional space. The input is an of a color chart, which provides the composition samples. We propose two different prediction algorithms, one doing simple interpolation using radial basis functions (RBF), and another that trains a parametric model based on the KM equation to compute novel values. We show that RBF is able to reproduce arbitrary compositing behaviors, even non-paint-like such as additive blending, while KM compositing is more robust to acquisition noise and can generalize results over a broader range of values.

**CR Categories:** I.3.4 [Computer Graphics]: Graphics Utilities

**Keywords:** pigment, paint, kubelka munk, color, compositing

## 1 Introduction

Current digital drawing and painting tools mimic many aspects of traditional media with increasing fidelity. Much of the research has addressed the shape and texture of individual strokes, as well as secondary effects like bleeding and smearing. However,

the interaction of multiple pigments has received relatively less attention, yet remains crucial for plausibly simulating the effects of many media including watercolor and oil paint. The major strategies for layering digital pigment have been either to rely on simple ad-hoc methods like alpha-blending using various blend modes, or to resort to complex physical models such as that of Kubelka and Munk (KM) [1931]. Simple alpha blending maps trivially onto the traditional graphics pipeline, but spectacularly fails to capture straightforward paint mixing principles such as “yellow and blue makes green” (Figure 1b). On the other hand, physically accurate models like KM (Figure 1d) require many parameters to capture each individual pigment, and even more parameters when they are mixed. Of course it is possible to measure such KM parameters for a set of paints and use them in a digital palette, but this requires sophisticated hardware and restricts the artist to working within the fixed set of paints, and when drawing still requires specifying more parameters such as density and thickness – this extra complexity leads to cumbersome interfaces in a digital painting tool.

Instead we propose a data-driven approach toward layering color in digital painting tools. In our process, the way that colors blend is specified by leveraging a physical *color chart* like the one shown in Figure 1a. Artists use such instruments to understand how the range of colors expressible in a particular palette of paints will combine, before applying paint to canvas. The artist first applies a blob of a particular color across each cell in a row, and moves onto the next row with a different color, and so on. Next, the artist applies a blob of color across each cell in a column, with a different upper color for each column. The resulting grid depicts visually how all pairs of color will appear when layered, and informs their application in a subsequent painting.

In our proposed digital painting interface, the user simply selects a color chart (the *exemplar*) in order to specify how colors will be composited. The color chart may be selected from a set of pre-existing charts, or may be created anew in order to specify a novel mixing behavior. Next, the user simply selects (RGB) colors using a standard color picker and paints strokes, and the system plausibly composites colors in a way that mimics the exemplar chart.

This paper makes a series of contributions to enable this interface. The first contribution is a novel formulation of the pigment compositing problem as interpolation among a sparse set of high dimensional samples. Our second contribution is, based on this novel formulation, detailing how to acquire such a set of samples via a color chart, and interpolate them using radial basis functions (RBF) [Powell 1987] (Figure 1c). The third contribution is to improve upon RBF by using a priori knowledge about the shape of the function by using the KM equation, effectively interpolating in KM space (Figure 1d). Finally, we explore the performance of these methods compared against a baseline of optimized alpha blending. We show that for pigment models that can be reproduced by the KM equation, the optimized KM interpolation is able to produce better results that smooth over acquisition errors and can be extrapolated to a broader range of compositions. Conversely, RBF interpolation better approximates the exact behaviors in the color chart, whether or not those behaviors are realizable by pigments (e.g. additive blending).

## 2 Related Work

There is extensive research on modeling physical pigments to accurately predict the color when they are combined to improve printing processes and make realistic natural media painting systems.

### Kubelka-Munk and Other Physically-Accurate Models

The seminal work by Kubelka and Munk [1931] (KM) defines a per-wavelength equation for the reflectance of a homogeneous, isotropic pigment layer atop a substrate in a continuous setting (see Section 3). Alternate formulations have been developed in the same domain [Giovanelli 1995; Hecht 1983; Vargas and Niklasson 1997], but KM remains the most commonly used.

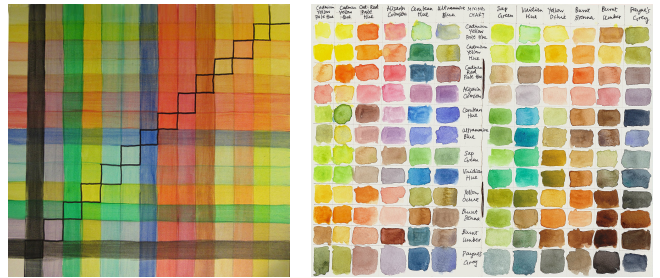
Much work has been done to improve KM. Saunderson [1942] provided a correction based on surface reflection of the pigment layer, and Berns and De la Rie [2003] extended this result by considering reflective varnishes for oil paint. Granberg and Edström [2003] showed the KM model mis-estimates strongly absorbing pigments, and Yang and Kruse [2004] proposed a revised KM model which corrects for path-length errors in highly scattering media. However, Edström [2007a] later showed this revision had other errors.

Extensions have been proposed to relax the assumptions made by KM. Shakespeare and Shakespeare [2003] proposed a method to incorporate fluorescence, while Yang et al. [2004] added inhomogeneous materials. Edström extended these to also include anisotropic reflections [2007b; 2008]. Finally, Sandoval and Kim [2014] re-derived the KM equation from the radiant transport equation, thereby generalizing it to handle inhomogeneous materials with boundaries, and with greater accuracy.

### Approximations to Kubelka-Munk

Many natural media painting systems have used the KM model in various forms to improve the quality of paint rendering. The Stokes Paint program [Baxter et al. 2004a] uses linear combinations of four or eight acquired real pigments with KM coefficients to generate a wide gamut of paints. Impasto [Baxter et al. 2004b] improves on this with a numerical approximation of the same pigments for reduced memory and computation.

Curtis et al. [1997] uses a reduced KM model with coefficients for only three wavelengths, which are computed from user-specified RGB colors. A further approximation is presented by Wang and Wang [2007], which uses a single coefficient variant of KM, for each of three wavelengths. Perhaps the simplest approximation is



(a) thin acrylic ©Chris Gentes (b) watercolor ©Denise Chan  
**Figure 2:** Example physical color charts available online.

found in the wax crayon simulation of Rudolf et al. [2005], which uses one absorption and scattering coefficient pair plus an RGB color, thereby entirely obviating multispectral rendering.

Gossett and Chen [2004] create a paint-inspired color model they label RYB (for Red Yellow Blue), which defines three paint pigment primaries and linearly interpolates among them to define a paint-like gamut. This is most similar to our approach, but we generalize the concept to support arbitrary, non-orthogonal, sparse sets of paint samples with data-defined compositing behavior.

Finally, instead of explicitly defining an equation to model the pigment composition effect, Xu et al. [2007] and Westland et al. [2002] both propose using a neural network to learn how pigments combine to produce the final reflectance spectra. These algorithms must be trained on acquired pigment composition samples with KM coefficients, and do not provide clear mechanisms for exerting artistic control over the resulting model.

### Pigment Acquisition

Acquiring the parameters of individual pigments to use in these various models is an arduous task, involving multispectral measurements of controlled samples in controlled environments [Okumura 2005]. Mohammadi and Berns [2006] show that it is crucial to capture the full KM coefficients for a pigment in order to accurately reproduce the range of generated colors. Kokla et al. [2006] propose a way to use a set of captured pigments to determine from a simple image of a painting or manuscript the mixture of pigments used for each local color, via an optimization.

## 3 Pigment Model

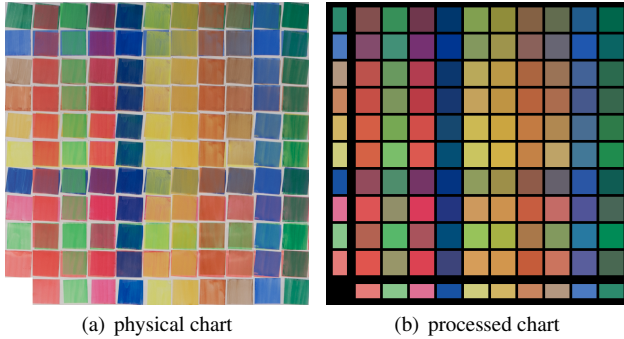
The commonly accepted method for predicting the reflectance of a layer of pigment is to use the Kubelka-Munk (KM) equation [Kubelka and Munk 1931; Glassner 1994]:

$$R = \frac{1 - \xi(a - b \coth bSh)}{a - \xi + b \coth bSh} \quad (1)$$

$$a = 1 + \frac{A}{S}, \quad b = \sqrt{a^2 - 1}$$

where  $h$  is the thickness of the pigment layer,  $S$  and  $A$  are the pigment's scattering and absorption coefficients,  $\xi$  is the substrate reflectance, and  $R$  is the resulting reflectance of the pigment layer. All of these quantities are per-wavelength, so a pigment  $p$  is modeled as a set of scattering and absorption coefficients over a set of wavelengths  $L$  as  $p = \{S_l, A_l | l \in L\}$ .

Realizing the RGB to display in an image corresponding to a stroke of pigment on a canvas is a standard multispectral rendering problem. Let us denote this transform as  $c = \{r, g, b\} = f(p)$ , for



**Figure 3:** We capture a physical gouache chart using uniform glazes on transparent plastic.

a pigment  $p$ .  $f(p)$  is the perceptual response of the pigment, which could be regarded as the space that artists think in when selecting pigments for painting. That is, rather than understanding pigments in terms of a set of coefficients or a reflectance spectrum, an artist may associate pigment  $p$  with a particular color  $c$ , for some nominal conditions (illuminant, thickness, substrate).

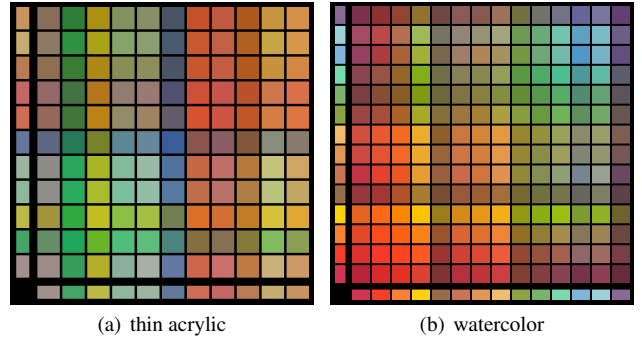
Now consider two pigments,  $p_1$  and  $p_2$ , painted on a white canvas such that  $p_2$  is on top of  $p_1$ . The perceived color of this composition is  $f(p_2 \circ p_1)$ , which can be computed by the recursive application of the KM equation. However, from the artist’s perspective, the behavior is better modeled as  $g(f(p_2), f(p_1))$ , where  $g$  is a composition function that operates on RGB values. Indeed, a digital artist selects an RGB to paint with, and needs a predictable and plausible outcome when painting over the RGBs on the canvas.

The function  $g : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is an approximate model of the pigment composition process in which foreground and background RGB colors combine to form a new RGB color. Our data driven color model sparsely represents  $g$  by a set of a-priori known composition results that are interpolated to generate the result for arbitrary foreground and background colors. These sparse samples come from the color chart provided by the user; a different color chart will result in a different function  $g$  and therefore a different color model. For example, an oil paint chart would produce a different  $g$  than a watercolor chart.

The nature of  $g$ ’s approximation is that it is not possible in general to find an analytical form of  $g$  such that  $g(f(p_2), f(p_1)) = f(p_2 \circ p_1)$  for all  $p_1$  and  $p_2$ . One reason is that  $g$  cannot express the behavior of *metamers* – different pigments  $p_a \neq p_b$  that share the same perceived color:  $f(p_a) = f(p_b)$ . In general, these metamers will yield different results when painted over some other color  $p_c$ :  $f(p_a \circ p_c) \neq f(p_b \circ p_c)$ . This phenomenon cannot be expressed by  $g$ , which treats pigments only in terms of perceived color, i.e.  $f(p_a) = f(p_b) \Leftrightarrow p_a = p_b$ .

While  $g$  has these inherent ambiguities regarding metamers, we regard this limitation as a benefit of our approach. While metamerism is important for simulation of physical reality, it is a highly non-intuitive effect. Understanding the expected behavior of each combination of real pigments is a difficult part of learning to paint, which is why even expert painters often create color charts to help them predict these behaviors. Thus, our pigment model can improve the predictability of paint compositing by eliminating this difficult phenomenon.

While this discussion has focused on paint composition, in which one layer of pigment is applied atop another, it can be trivially extended to paint mixing, in which two pigments are blended into a single layer. In the KM model, pigment mixing is accomplished by



**Figure 4:** Processed charts used in our experiments in two different media. The acrylic data (left) comes from Figure 2a, whereas the original watercolor chart not shown is similar to Figure 2b.

interpolating between the two pigments’ scattering and absorption coefficients, with resulting color  $f(p_2 \otimes p_1)$ . This can be modeled in the same manner, by a function  $m(f(p_2), f(p_1)) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , which is defined by a sparse set of color mixing samples. The primary difference between  $m$  and  $g$  is that  $m$  is symmetric,  $m(b, a) = m(a, b)$ , while  $g$  is not,  $g(b, a) \neq g(a, b)$ .

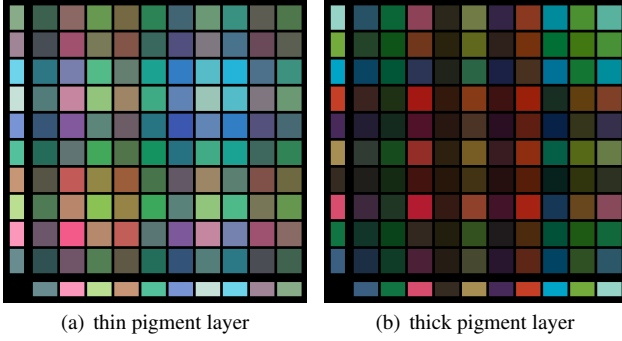
## 4 Data Acquisition

Artists create color charts to familiarize themselves with properties of different pigments. They experiment with glazing a layer of pigment on top of a dried layer to study the effect of compositing. In order to choose the right pigments to paint with, they observe the color chart and speculate on behaviors of pigments outside the chart. This process can be automated by prediction models, which learn from the known compositing examples in the chart and predict the result given an arbitrary pair of new colors. To train our prediction models (see Section 5), we collect a set of physical and synthetic color charts.

**Physical charts.** As a common practice, a compositing chart is created as follows: Given  $n$  base pigments, divide the empty canvas into a grid of  $n^2$  cells. As the background layer, paint each row  $i$  with base pigment  $i$ , keeping uniform distribution of the pigment for all cells of the row. After the background layer is completely dried, as the foreground layer, paint each column  $j$  with base pigment  $j$ . Artists frequently paint charts using this procedure and post images of their results online. We collect these images from artists’ web pages for several different natural media, including marker pen (Figure 1a), acrylic (Figure 4a), and watercolor (Figure 4b). In addition to the online charts, we also captured a gouache chart ourselves in a controlled setting. To control paint thickness, we made uniform glazings of gouache on transparent plastic, cut the plastic pieces to obtain background and foreground squares, and then captured every layered combination of these squares (Figure 3a).

To process these physical charts, we undo the gamma correction and extract a single color from each grid cell by averaging pixels around the cell center. The cell borders are avoided to prevent the boundary effects, such as watercolor edge-darkening, to adversely influence the measured color. After processing, each grid cell contains a *composite color* – the result of glazing a particular foreground pigment over a background pigment or a *base color* – the result of applying a base pigment on the empty canvas (the leftmost column and the bottommost row in Figures 2a and 3). Figures 3b and 4 show the processed charts. Though subtle, the compositing charts are not symmetric. The result of applying





**Figure 5:** Synthetic color compositing charts using different thickness of pigment layer, created using acquired Kubelka-Munk pigment coefficients and multispectral rendering.

pigment  $A$  on top of  $B$  is slightly different from the result of applying  $B$  over  $A$ . For example, in Figure 3b, column 4 within the composite matrix shows clear asymmetry from row 4, because the dark blue paint is very absorbent of light whereas other colors are more reflective.

To simplify our inference problem and reduce the training parameters, we make some **assumptions** about the physical color charts. We assume that the thickness of the pigment layer is more or less constant for all rows and columns and that the charts are captured under roughly uniform lighting. The assumptions are reasonable, since they are the guiding principles for creating the charts. But they are hard to enforce precisely and therefore small variations in brightness and layer thickness can be observed. For example, Figure 2(b) shows luminance variation within many cells due to the difficulty of depositing watercolor pigment uniformly.

**Synthetic charts.** Uniform lighting and constant layer thickness is desirable, but not guaranteed, in the hand-painted charts. To understand how violations of these assumptions influence the prediction and to provide a baseline evaluation of our models, we algorithmically construct some synthetic charts. Specifically, we apply the KM model (see Section 3) with the physically measured coefficients [Okumura 2005] of five acrylic pigments, including Phthalo Blue, Pyrrole Red, Hansa Yellow, Carbon Black and Titanium White. Each of the  $n$  base pigments is generated as a random mixture of the five primaries. The grid cells of the chart are calculated by two evaluations of the KM model. One evaluation of KM applies the background pigment onto the substrate and another applies the foreground pigment onto the rendered background. For KM, we use constant layer thickness for all pigment layers, standard illuminant D65 [Ohta and Robertson 2006] and ideal white substrate (completely reflective at all wavelengths). For some natural media, the compositing behaviors might vary with different layer thickness. For example, for oil or acrylic, a thin layer of foreground pigment allows the background pigment to show through (Figure 5a), while a thick layer heavily obscures the background (fifth column from the right in Figure 5b) with the exception of using light-colored pigments as the foreground (first column from the right). Additionally, high light absorbance can cause certain colors to result in dark composites if used in either background or foreground (seventh column and row from the top and the right).

## 5 Prediction Models

Suppose a color chart is painted with  $N$  base pigments, then we have  $N$  base color samples, represented as  $B_i$ . And we have  $N^2$  composite color samples, represented as  $C_{ij}$ , produced by layering

base pigment  $j$  (foreground) on top of base pigment  $i$  (background). In the empty region of the chart, we can sample the substrate color  $S$ . We convert all color values into the CIELAB color space for its perceptual linearity. Each color chart can be seen as a 6D point cloud of  $N^2$  points,  $\{(B_i, B_j) \mid i, j \in \{1 \dots N\}\}$ . We use the standard quick hull algorithm [Barber et al. 1996] to calculate the convex hull,  $\mathcal{H}$ , of the point cloud, which is utilized in the following prediction models.

When a user selects a color chart to specify compositing behavior and then paints a given foreground color  $Q_f$  over a different background color  $Q_b$ , our goal is to predict the composite color  $Q_c$ . To address this goal, we propose three prediction models, *optimized alpha compositing*, *radial basis interpolation* and *optimized KM compositing*, with different pros and cons. Optimized alpha compositing (Section 5.1) is included as a baseline, which demonstrates the best possible performance of alpha blending to approximate the real paint compositing in the chart. Radial basis interpolation (Section 5.2) is a straightforward implementation of our model from Section 3 that interpolates and extrapolates the matrix  $g$ . Alternately Section 5.3 presents an optimized KM compositing scheme that refines the model for color charts that contain paint-like pigments. Each model contains a training phase conducted per color chart, and a prediction phase performed per query color. We solve all proposed optimizations using ALGLIB [Shearer and Wolfe 1985].

### 5.1 Optimized Alpha Compositing

Alpha compositing is often used in digital painting for its simplicity and efficiency. During painting, an alpha value is often specified by the user together with a target color. Choosing the right alpha to closely reproduce the appearance of a target medium is challenging. For example, there is no alpha value that can perfectly simulate the darkening effect of watercolor. However, given a color chart, we can solve a constrained least squares problem to optimize the alpha value that best explains the chart. Then the optimized alpha can be used to predict the compositing result given new query colors. We include this method to provide a reference point for evaluating the performance of our other methods.

**Training.** Given an intrinsic color  $\beta_i = (L_i, a_i, b_i)$  and a chosen alpha value  $\alpha$ , alpha compositing on empty canvas gives  $b_i = \alpha\beta_i + (1-\alpha)S$ . Using the same alpha, given another intrinsic color  $\beta_j$ , applying alpha compositing recursively on top of the previous result gives  $c_{ij} = \alpha\beta_j + (1-\alpha)b_i = \alpha\beta_j + (1-\alpha)\alpha\beta_i + (1-\alpha)^2S$ . Given a color chart, we want to minimize the difference between  $b_i$  and the measured base color  $B_i$  as well as the difference between  $c_{ij}$  and the measured  $C_{ij}$ , solving for the  $\alpha$  and the  $\beta = \{\beta_i, i \in \{1 \dots N\}\}$  parameters:

$$\alpha^*, \beta^* = \underset{\alpha, \beta}{\operatorname{argmin}} E_a \quad (2)$$

$$\text{such that } \alpha \in [0, 1], \text{ and } L_i \in [0, 100], \forall i \in \{1 \dots N\}$$

$$E_a = \sum_i^N \|B_i - b_i\|^2 + w_c \frac{1}{N} \sum_i^N \sum_j^N \|C_{ij} - c_{ij}\|^2 \quad (3)$$

We use  $w_c = 1$  to give equal importance to fitting both the base and the composite colors. We initialize  $\alpha$  to be 0.5 and  $\beta$  to be random. The optimized alphas  $\alpha^*$ , in the order of Table 1, are 0.58, 0.42, 0.49, 0.63, 0.31, 0.45, plausibly reflecting the thinner or thicker paint behaviors in the charts.

**Prediction.** After training, we obtain the optimized alpha value  $\alpha^*$  that best describes the chosen color chart. Suppose the unknown intrinsic color of the foreground is  $q_f = (L_f, a_f, b_f)$ , then the



methods	Average prediction error (CIELAB) for queries, outside gamut, near gamut, and combined																	
	marker			acrylic			watercolor			gouache			thin synthetic			thick synthetic		
alpha	30.75	29.20	29.99	10.92	9.02	9.88	21.23	18.37	19.80	10.89	8.19	9.54	3.02	2.40	2.71	11.90	8.74	10.32
rbf	16.78	9.01	12.90	9.90	4.85	7.14	9.19	7.05	8.12	8.45	4.80	6.63	1.30	0.87	1.09	5.53	4.49	5.01
rbf-nopoly	19.02	9.58	14.30	14.39	5.19	9.38	9.69	7.14	8.41	9.60	5.08	7.34	6.55	2.89	4.72	7.54	5.33	6.43
km	12.32	10.72	11.52	8.04	8.26	8.16	8.83	8.98	8.90	9.17	6.68	7.93	0.41	0.34	0.38	2.51	2.04	2.28

**Table 1:** Hold-one-out evaluation results, broken up to show error for outside gamut samples, near gamut samples, and both sets combined.

following holds,  $Q_f = \alpha^* q_f + (1 - \alpha^*)\mathcal{S}$ . The composite color  $Q_c = (L_c, a_c, b_c)$  can be estimated as:

$$Q_c = Q_b(1 - \alpha^*) + \alpha^* q_f = Q_b(1 - \alpha^*) + Q_f - (1 - \alpha^*)\mathcal{S} \quad (4)$$

with a caveat that the resulting luminance might be outside the valid range,  $L_c \notin [0, 100]$ , when the 6D query point is outside the convex hull of the color chart,  $(Q_b, Q_f) \notin \mathcal{H}$ .

To avoid this situation, we perform a fast runtime optimization to make small adjustment to the alpha estimation. In particular, we use  $\alpha^*$  and  $\beta^*$  obtained from the training to initialize the following optimization, solving for  $\hat{\alpha}$ ,  $\hat{\beta}$  and  $\hat{q}_f$ :

$$\hat{\alpha}, \hat{\beta}, \hat{q}_f = \underset{\alpha, \beta, q_f}{\operatorname{argmin}} E_a + w_a \|Q_f - \alpha q_f - (1 - \alpha)\mathcal{S}\|^2 \quad (5)$$

such that  $\alpha \in [0, 1]$ , and  $L_f \in [0, 100]$

We set  $w_a = 3$  to encourage the query color  $Q_f$  to be well fitted. The composite color can then be predicted by substituting  $\hat{\alpha}$  and  $\hat{q}_f$  into the equation (4).

## 5.2 Radial Basis Function Interpolation

The Radial Basis Functions (RBF) is one of the primary tools for interpolating multidimensional scattered data. RBF is suitable for our problem for the following reasons: it does not require the data points to lie on a regular grid, and gives good interpolation accuracy even when the data is sparse. RBF is defined as:

$$g(\hat{x}) = \sum_t^n \lambda_t \phi(\|\hat{x} - \hat{x}_t\|), \hat{x} \in \mathcal{R}^D \quad (6)$$

**Training.** A color chart can be interpreted as a 6D point cloud, with each data point corresponding to a 3D value. In our case,  $\{\hat{x}_i\}$  in equation (6) corresponds to  $\{(B_i, B_j) \mid i, j \in \{1 \dots N\}\}$ . We have  $n = N^2$  and  $D = 6$ . Given a sparse point cloud of several hundreds points, an arbitrary query is often near or outside its convex hull,  $\mathcal{H}$ . In these situations, the basic RBF interpolation is inaccurate. We use augmented RBF, which adds a polynomial term to the equation (6):

$$f(\hat{x}) = g(\hat{x}) + \sum_m^M \gamma_m p_m(\hat{x}) \quad (7)$$

We use the parameter-free linear kernel,  $\phi(r) = r$  and the following polynomial basis,  $\{p_m(\hat{x}) \mid m \in \{1 \dots M\}\} = \{1, \hat{x}, \hat{x}^2, \dots\}$ ,  $M = 2$  (that is,  $\{1, \hat{x}\}$ ). The linear polynomial term can improve the interpolation accuracy, especially at and outside the boundary (second and third rows in Table 1), but little benefits can be obtained going beyond linear [Wright 2003]. The polynomial term also allows simple extrapolation beyond the convex hull of the chart. Note that, our data values  $\{C_{ij}\} = \{C_t \mid t \in \{1 \dots N^2\}\}$  are 3 dimensional, therefore we train three separate RBFs for each color dimension  $\hat{f}(\hat{x}) = (f_l(\hat{x}), f_a(\hat{x}), f_b(\hat{x}))$ .

We write the following constraints in a matrix form and solve for the expansion parameters,  $\lambda$  and  $\gamma$ .

$$\hat{f}(\hat{x}_t) = C_t, \quad \forall t \in \{1 \dots N^2\} \quad (8)$$

$$\sum_t^{N^2} \lambda_t p_m(\hat{x}_t) = 0, \quad \forall m \in \{1 \dots M\} \quad (9)$$

**Prediction.** With the trained  $\lambda$  and  $\gamma$ , we predict the composite color by substituting the query point  $(Q_b, Q_f)$  into equation (7).

## 5.3 Optimized KM Compositing

Similar to Section 5.1, we attempt to train a parametric model to explain the color samples from the compositing chart. The difference is that this section assumes the underlying model to be Kubelka-Munk (KM), which is a powerful multi-spectrum model that more accurately simulates the appearance of physical pigments.

**Training.** We use  $L = 8$  uniformly spaced wavelength samples to approximate the full visible light spectrum. We find that using more wavelengths does not further improve the prediction accuracy. On the other hand, it introduces more unknowns to the optimization, which requires a larger color chart to avoid overfitting. We assume each base pigment of the chart is created by mixing different proportions of  $K = 3$  distinct primary pigments, which corresponds to the common art practice that a color on the palette is usually created by mixing several different paints. This prediction model has the following parameters:

- $\sigma = \{s_l^k, a_l^k \mid l \in \{1 \dots L\}, k \in \{1 \dots K\}\}$ : the scattering and absorption coefficients of each primary pigment at each sample wavelength, in total  $2LK$  parameters.
- $m = \{m_i \mid i \in \{1 \dots N\}\}$ , where  $m_i = \{m_i^k \mid k \in \{1 \dots K\}\}$ : the proportions for combining the  $K$  primaries, for each of the  $N$  base pigments, in total  $NK$  parameters.
- $\xi = \{\xi_l \mid l \in \{1 \dots L\}\}$ : the reflectance coefficients of the substrate at each sample wavelength, in total  $L$  parameters.

Given a mixing proportion  $m_i^k$ , the absorption and scattering coefficients of the mixed pigment  $i$  at each wavelength  $l$  is given by  $A = \sum_k^K m_i^k a_l^k$  and  $S = \sum_k^K m_i^k s_l^k$ . We concisely represent the pigment mixing and the KM equation (Section 3), using the following notation  $\mathbf{km}(\sigma, m_i, \xi)$ , where the input are  $\sigma$  and  $\xi$ , the output is the reflectance spectrum, and we use constant layer thickness,  $h = 1$ . We represent the standard transform function that converts a reflectance spectrum to tristimulus CIELAB values as  $\mathbf{lab}(\xi)$ , where we use the standard illuminant D65. Then, each base color  $b_i$  and composite color  $c_{ij}$  can be evaluated by KM:

$$r_i = \mathbf{km}(\sigma, m_i, \xi) \quad (10)$$

$$b_i = \mathbf{lab}(r_i) \quad (11)$$

$$c_{ij} = \mathbf{lab}(\mathbf{km}(\sigma, m_j, r_i)) \quad (12)$$

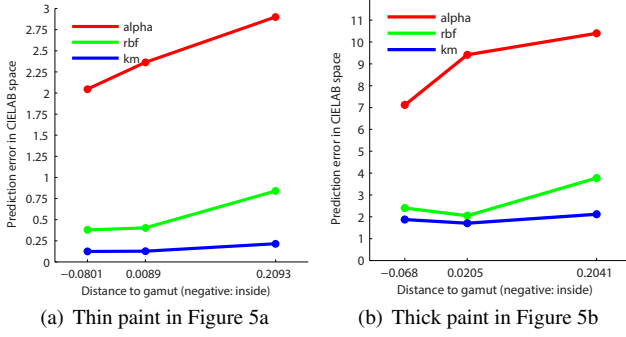


Figure 6: Gamut influence on prediction accuracy.

We train the model parameters as follows:

$$\sigma^*, m^*, \xi^* = \operatorname{argmin}_{\sigma, m, \xi} E_k + w_c E_c + w_p E_p + w_s E_s \quad (13)$$

such that  $s_i^k \in [0, \infty)$ ,  $a_i^k \in [0, \infty)$ ,  $m_i^k \in [0, 1]$ ,

$$E_k = E_a + w_\xi \|\mathcal{S} - \mathbf{lab}(\xi)\|^2 \quad (14)$$

$$E_c = \frac{1}{K} \sum_i \left( 1 - \sum_k m_i^k \right)^2 \quad (15)$$

$$E_p = \frac{N}{(L-1)K} \sum_k \left( \sum_l (s_l^k - s_{l+1}^k)^2 + (a_l^k - a_{l+1}^k)^2 \right) \quad (16)$$

$$E_s = \frac{N}{L-1} \sum_l (\xi_l - \xi_{l+1})^2 \quad (17)$$

where  $E_k$  replaces the  $b_i$  and  $c_{ij}$  in equation (3) with the new definitions in equation (11) and (12), and adds another term to fit the measured substrate color  $\mathcal{S}$ .  $E_c$ ,  $E_p$  and  $E_s$  serve as the regularization.  $E_c$  constrains that for each base pigment, the mixing proportions of the primaries sum up to 1.  $E_p$  encourages the scattering and absorption coefficient curves to be smooth for each primary.  $E_s$  encourages the reflectance curve of the substrate to be smooth. We initialize  $\sigma$ ,  $m$  and  $\xi$  randomly within their respective valid range. We set  $w_c = 1.5$ ,  $w_p = w_s = 60$ ,  $w_\xi = 1$ . Rendering the optimized primaries on white substrate with KM equations results in three distinct colors that well represent the base colors in the marker chart, for example (inset figure).



**Prediction.** Given the query foreground color  $Q_f$  and the optimized parameters  $\sigma^*$ ,  $\xi^*$ , we solve a least squares problem to predict the proportion,  $m_q = \{m_q^k \mid k \in \{1 \dots K\}\}$ , by which the primary pigments are mixed to generate the query foreground color. We could use the same least squares problem to solve for the mixture of primary pigments for the background color  $Q_b$ , but this would take twice as long and throw away previously computed data needlessly. Instead we cache and re-use the reflectance spectrum of the canvas due to previous strokes,  $\mathcal{R} = \{r_l \mid l \in \{1 \dots L\}\}$ . Then the composite color  $Q_c$  can be predicted as follows:

$$m_q^* = \operatorname{argmin}_{m_q} \|\mathbf{lab}(\mathbf{km}(\sigma^*, m_q, \xi^*)) - Q_f\|^2 \quad (18)$$

such that  $m_q^k \in [0, 1], \forall k \in \{1 \dots K\}$ ,

$$Q_c = \mathbf{lab}(\mathbf{km}(\sigma^*, m_q^*, \mathcal{R})) \quad (19)$$

Prediction in this way is efficient, but can be problematic at times. The trained primary pigments  $\sigma^*$  define a color gamut (the range

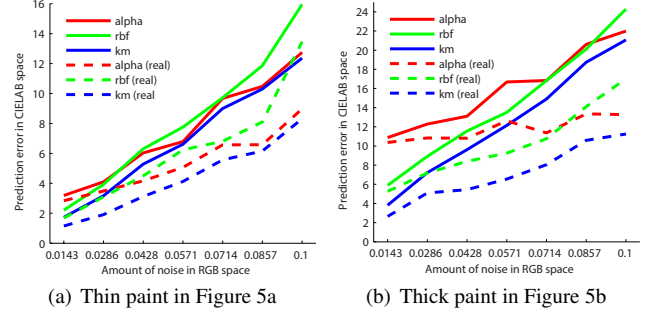


Figure 7: Noise influence on prediction accuracy.

of colors that can be represented by mixing the primaries). When the query color  $Q_f$  is outside the gamut, the least squares solution might be far from the true answer. We use the convex hull of the color chart,  $\mathcal{H}$ , to approximate the color gamut.

In this case, we propose the following remedy. Similar to the idea in Section 5.1, we perform a runtime optimization taking into account the query color  $Q_f$  using the trained parameters,  $\sigma^*$  and  $m^*$ , as initialization and having the substrate spectrum fixed at  $\xi^*$ . We solve the following least squares problem, slightly modified from equations (13) and (14):

$$\hat{\sigma}, \hat{m}, \hat{m}_q = \operatorname{argmin}_{\sigma, m, m_q} E_q + w_c E_c + w_p E_p + w_s E_s \quad (20)$$

such that  $s_i^k \in [0, \infty)$ ,  $a_i^k \in [0, \infty)$ ,  $m_i^k \in [0, 1]$ ,  $m_q^k \in [0, 1]$ ,

$$E_q = E_k + w_k \|\mathbf{lab}(\mathbf{km}(\sigma, m_q, \xi^*)) - Q_f\|^2 \quad (21)$$

The term  $E_q$  adjusts the primary coefficients and the query mixing proportion to constrain the query color to lie inside the gamut. After the optimization, we predict the composite color by substituting  $\hat{\sigma}$  and  $\hat{m}_q$  into equation (19). We set  $w_k = 3$  and initialize  $m_q$  using the mixing proportion of the base pigment that has the most similar color to the query.

## 6 Results and Discussion

We use a number of quantitative and qualitative means to evaluate the quality of our pigment model, and discuss the impact of various design considerations on our results.

### 6.1 Quantitative Results

We report errors in terms of L2 difference in CIELAB colorspace, which represents the perceptual error in the predicted color. The just noticeable difference (JND) in LAB is between 1.0 and

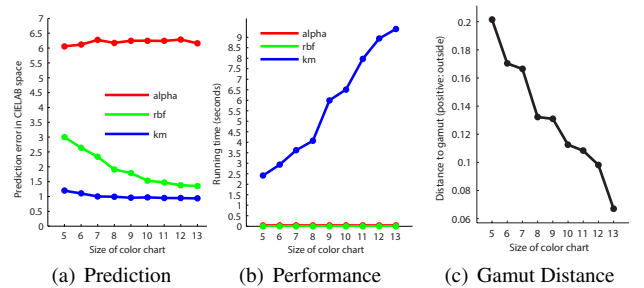
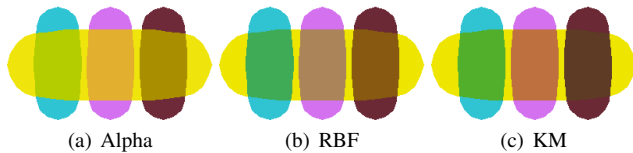
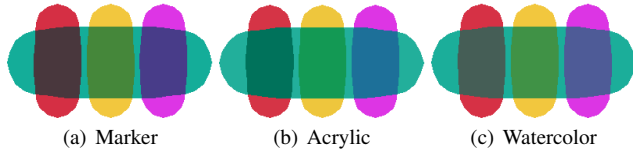


Figure 8: Experiments with synthetic charts.



**Figure 9:** Prediction methods comparison. All strokes are synthesized using the marker chart (Figure 1a). Foreground: horizontal.



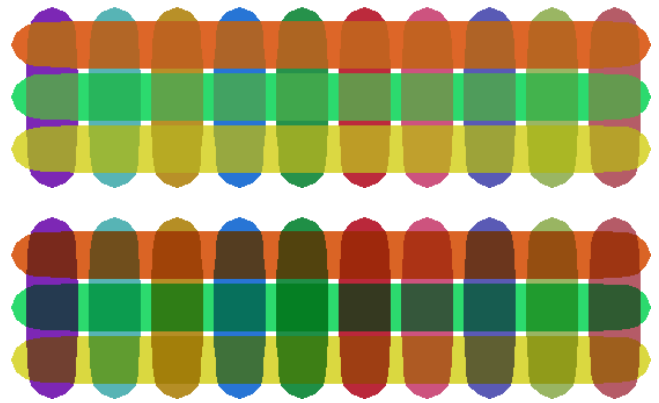
**Figure 10:** KM prediction results using different color charts from Figures 1a, 4a and 4b, respectively. Foreground: horizontal.

2.3 [Robertson 1990] – differences less than this value are not generally perceivable to casual viewers.

**Hold-one-out Evaluation.** To evaluate the quality of our prediction, we would ideally compare the predicted color of a composition with the ground truth color for a given medium. However, ground truth data is difficult to come by, particularly for color charts from the Internet. To work around this we use a hold-one-out methodology. If we have  $N$  base colors in a mixing chart, for color  $i$  we create a new pigment model on the  $N - 1$  other pigments (removing column and row  $i$  from the chart), and then use this reduced model to predict the composition results of color  $i$  with the other  $N - 1$  colors, which we have ground truth for (from the original chart of  $N$  base colors). Data from this evaluation is presented in Table 1. We find that for media of lower scattering properties (i.e. less opacity), namely marker and watercolor, RBF and KM both perform dramatically better than alpha blending. More highly scattering pigments such as acrylic and gouache are closer in performance, though RBF and KM still show an improvement over alpha blending. Table 1 also shows that RBF with the polynomial term outperforms the basic RBF in all test cases, especially for the outside gamut samples.

**Influence of Gamut.** Because the pigment model is a 6D function and there are relatively few samples, in general all the samples are on the convex hull which defines the gamut of the model, or the volume within which samples can be interpolated. This means that when performing the hold-one-out evaluation, the held out sample  $i$  is usually outside the gamut of the remaining  $N - 1$  samples. However, depending on the shape of the gamut, the sample  $i$  may be near the gamut (e.g. if  $i$  is an orange color and both red and yellow remain) or far outside the gamut (e.g. if  $i$  is blue and the remaining  $N - 1$  samples are all reds, oranges, and yellows). The prediction of near gamut samples is generally better than outside gamut as the prediction is closer to interpolation than extrapolation. For that reason, Table 1 splits the aggregate errors into near and outside gamut values, to see the difference. The important detail is that RBF tends to have a bigger discrepancy between near and outside gamut samples than KM, because RBF relies on extrapolation while KM fits a more restricted model that generalizes better.

To evaluate the error on inside gamut samples requires many more samples than we can reasonably acquire so we rely on synthetic charts made from measured KM coefficients of real paint pigments [Okumura 2005] and multispectral rendering. Figure 6 shows the relationship between distance from the gamut boundary for the thin and thick synthetic charts from Figure 5. We approximate the query distance from the gamut boundary (x axis value in Figure 6, negative means inside) by the Euclidean distance to the



**Figure 11:** KM Optimization results. Top matrix: synthesized with the gouache chart in Figure 3a. Bottom matrix: synthesized with the thick synthetic chart in Figure 5b. Foreground: horizontal.

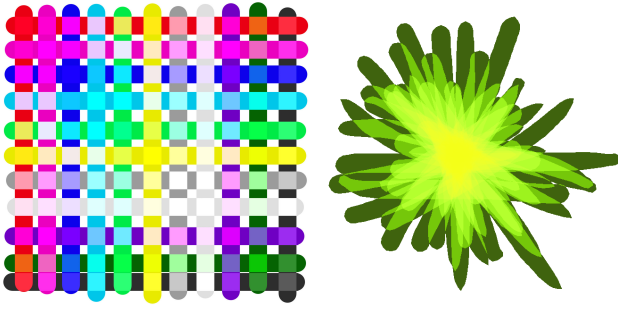
training convex hull normalized by the average distance of pairs of training samples. The performance of KM remains roughly the same, while RBF and alpha blending are both correlated with the signed distance from the gamut boundary.

**Influence of Noise.** The acquisition process for a physical chart involves a fair amount of noise. Most significantly, the swatches may not all be painted with the same pigment thickness and lighting may not be constant across the chart. We again rely on synthetic color charts to gain some understanding of the influence of noise on our results. Figure 7 shows prediction error for our thin and thick synthetic paint charts with increasing amounts of Gaussian noise (maximum  $[-0.1, 0.1]$ ) added to each RGB channel. The graphs show error in two ways: the solid lines represent the same hold-one-out error calculation as explained before. However, in this experiment, the ground truth values have noise added to them, which may increase the error over the “real” ground truth values that are the actual composition colors without noise. This “real” error is shown with dashed lines, and demonstrates that our hold-one-out methodology may overestimate the error between our models and actual pigments.

KM and RBF handle noise differently. RBF will simply try to reproduce whatever noise may be present in the color chart – systematic errors such as luminance changes across the chart can be reproduced well. KM on the other hand fits a single pigment model to all the data, effectively finding a best fit that minimizes the errors due to noise. While this may result in a higher apparent error for physical pigments due to our hold-one-out methodology, Figure 7 suggests that the actual error between the predictions and the real paints may be overall less.

**Influence of Chart Size.** It is interesting to examine the impact of chart size on performance, to determine how many pigments should be acquired (color charts online tend to have between 6 and 10 base colors). Figure 8 shows these results, where for each  $N = \{5 \dots 13\}$ , we created 5 random synthetic charts with base colors sufficiently different from each other and evaluate the error for a persistent set of 10 random samples. What we find is that KM is quickly able to generalize a model and does not change significantly as the chart size increases, whereas RBF is initially worse than KM and slowly approaches its performance as the chart size improves. On the other hand, performance for alpha blending is roughly constant and very fast regardless of chart size, while KM’s time to compute a prediction increases linearly with the number of pigments, as the size of the optimization problem becomes harder. We measure the running time using a desktop computer with Intel





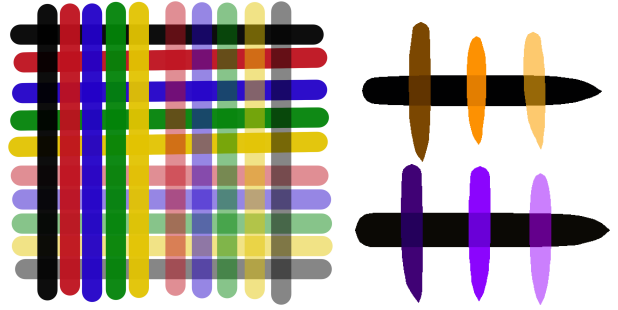
**Figure 12:** A non-paint-like color chart that exhibits additive blending, created in Adobe Photoshop. RBF is used to create the strokes on the right that reproduce the additive effect.

Core i7-3770, 3.40GHz. Finally, the average sample distance from the model gamut decreases also roughly linearly as the chart size increases. With a practical chart size, random queries are more likely to be outside the gamut, which underscores the importance of handling extrapolation.

For KM it is also important to note that as the chart size grows, the amount of data increases faster than the number of unknowns. For  $N$  base colors,  $L = 8$  wavelengths and  $K = 3$  primary pigments (disregarding substrate optimization), we have  $2LK + NK$  unknowns and  $N^2 + 2N + 4K$  equations. Therefore, we want  $48 + 3N < N^2 + 2N + 12 \Rightarrow N^2 - N - 36 > 0$ , which means that when  $N \geq 7$ , we have more equations than unknowns. For  $N < 7$ , the system is underconstrained.

**Influence of Parameters.** The alpha and KM prediction methods both have a number of parameters to tailor their optimizations. However, we found that the results were not particularly sensitive to the specific values of the parameters. Similarly, optimizing the illuminant spectrum and the substrate reflectance spectrum, versus using standard values, does not make a large difference on the results, with the exception of the acrylic chart where the substrate is far from white. The impact of the regularization error terms is to reduce the number of optimization steps, which for KM is between 200 and 1000 iterations, and for alpha is between 20 to 60. A more significant factor in the quality of the prediction and speed of convergence is the amount of acquisition noise in the color charts – noisier charts require more optimization steps. Note that KM uses non-linear optimizations. To avoid getting trapped in a local minimum, off-line we perform the training several times with random initialization and pick the parameters giving the lowest training error. Then, initializing the runtime per-query optimization with the trained parameters improves the prediction performance and reduces the influence of local minima. Though the global minimum is hard to reach, we find that the prediction results of the synthetic charts are very close to the groundtruth suggesting that adequate local minimums are usually obtained. We solve the non-linear least squares problem using the standard Levenberg-Marquardt algorithm provided by ALGLIB [Shearer and Wolfe 1985]. To reduce the complexity of the runtime optimization, we can fix the mixing proportions  $m^*$  for the  $N$  base colors and only optimize the pigment coefficients,  $\sigma$ , for the  $K$  primaries and the mixing proportion,  $m_q$ , for the query.

Finally, we have noticed that charts with more bold or vivid colors (e.g. marker, thick synthetic) have higher prediction errors than lighter, more pastel charts (e.g. watercolor, thin synthetic). We believe this is due to our hold-one-out methodology and the larger gamuts of these charts. When holding out pigment  $i$ , the absolute distance to the gamut of the  $N - 1$  pigments is going to be larger for charts that have a wider gamut. Reporting relative



**Figure 13:** A color chart (left) with both opaque and translucent compositions, and some strokes (right) painted with the resulting pigment model (foreground: vertical). Note that lighter colored strokes are more transparent, while darker colored strokes are more opaque. Also all colors do not appear in the chart.

error, or normalizing the errors with respect to the gamut size, would therefore be necessary to compare the absolute estimation accuracy among different charts / media, but we have not found a reason to perform such comparisons. Many of the numerical values corresponding to the visual results shown in this paper will be available on our project page [Lu 2014].

## 6.2 Qualitative Results

We present a qualitative comparison of our model’s prediction quality in Figures 1 and 9. We choose saturated colors, the behaviors of which are familiar to casual viewers, to demonstrate our results using different prediction models. Figure 1 shows that for queries that are outside the training gamut (similar to, but different from the training base colors), the “best effort” alpha blending fails to reproduce the paint-like appearance. RBF improves the prediction of blue and yellow over red, but fails to produce green when applying yellow over blue, due to the fact that bright yellow is not represented in the chart, Figure 1a. KM, on the other hand, provides more robust extrapolation producing dark green as the result. Figure 9 shows more results of using the marker chart. Note that when layering bright yellow over cyan and pink, KM produces light green and orange respectively, which are closer to the behaviors in the chart. When using a darker brownish red as the background color, layering bright yellow results in dark brown making the yellow appear more transparent. We also compare our KM optimization results on the same set of queries, based on different charts. Figure 10 shows that different color charts produce perceptually distinct results. For example, cyan over red produces the light brownish color when using the watercolor chart (similar to  $C_{2,8}$  from the top and right in Figure 4b) and darker brownish red when using the marker chart (similar to  $C_{6,16}$  in Figure 1a). Note that though the saturated magenta is outside the gamut of the acrylic chart, paint-like composition is still obtained. Figure 11 demonstrates more results of KM based on two different training charts. We experiment with less saturated colors that are different from the training base colors and show that the prediction results faithfully simulate the high opacity and darkening effects in Figures 3a and 5b respectively.

We also show a non-paint-like model in Figure 12. We use Adobe Photoshop to create a color chart with RGB colors and the additive blend mode, which simulates the behavior of light (e.g. red and green makes yellow). This chart is not well modeled by alpha blending or KM, as neither has the ability to represent this type of behavior, but RBF handles it well, as can be seen in the simple painting in Figure 12, where colors not in the chart are repeatedly laid atop one another until the color is completely saturated. The hold-one-out evaluation on this chart produces

combined error values of 50.97, 19.92, and 51.74 for Alpha, RBF, and KM respectively.

### 6.3 Controlling Opacity

Our pigment model does not include an explicit parameter for “alpha” in the regular digital painting sense, in which a user can select an RGB and set the alpha value to determine how much of the background shows through. This information is however implicitly encoded in the color chart, as any set of paint composition behaviors can be represented. For example, one color chart could use thinned paint while another could use thick paint, and the different models would then include the corresponding transparent or opaque composition behavior. Furthermore, a single color chart could include samples from a dark red pigment applied thickly, appearing dark red, and thinly, appearing light pink – the model would then produce opaque results for dark red colors, and translucent results for light pink colors. See Figure 13 for an example.

The main limitation of this approach is that, because the pigment model cannot represent metamers, some flexibility is lost. For example, it is not possible to have both a thin, translucent stroke of dark red pigment that appears light pink, as well as a thick opaque stroke of light pink pigment, as these strokes would have the same RGB value in the color chart and so would conflict with one another.

In the implementation of our pigment model in a painting application, it may still be desirable to support some type of alpha control that operates in a way that users are accustomed. It would be straightforward to provide such a control which would interpolate between the background color and the composited result. That is, at  $\alpha = 1$ , the resulting color is the pigment model’s prediction, at  $\alpha = 0$ , the result is the background color, and at  $\alpha = 0.5$ , it is the average of the two, etc. Another approach would be to use  $\alpha$  to interpolate between two charts, one with a thin application of the foreground pigment and the other with thicker paint applied.

## 7 Conclusions

This paper presents a data-driven color model that can be adopted by existing painting systems for more realistic color compositing effects. We propose and compare a class of compositing methods and show their respective advantages. Using our framework, novice users can take a picture of a color chart and then paint strokes with it. The query strokes allow arbitrary color and emulate the compositing behaviors of the chart. We conduct quantitative analysis to evaluate our color model and develop a simple painting program that demonstrates the applications.

This project raises a number of possible areas for future work.

**Modeling noise.** We do not directly model noise in the acquisition process due to variation in the paints, thickness, opacity, lighting etc. A more sophisticated fitting scheme might be able to model these components and therefore respond to them more robustly.

**Support for paint mixing charts.** Section 3 discusses briefly how the chart-based approach we use for compositing could easily be extended to paint mixing. Nevertheless, any digital painting application that allows mixing needs controls for *both* operations. The cross product of all mixed and composited colors is obviously a substantially larger space; it may necessitate acquisition of considerably more data and/or require more care from the UI perspective.

**Simpler model outside convex hull.** We have found that the compositing behavior is high-quality when pigment combinations are selected that are within the convex hull of the data in the chart. As colors move further away the ability for our methods

to extrapolate well breaks down. Of course it is always possible to add more exemplars in the region one wishes to paint, but we also speculate that it might be possible to smoothly transition to some simpler, plausible model far away from the exemplar data.

## Acknowledgements

We are grateful to artists Denise Chan, Chris Gentes, and John Perry for sharing color charts on the internet and allowing us to show them here. We also thank the reviewers for helpful feedback on our paper. This research was sponsored in part by generous gifts from Adobe and Google.

## References

- BARBER, C. B., DOBKIN, D. P., AND HUHDANPAA, H. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*
- BAXTER, W., LIU, Y., AND LIN, M. C. 2004. A viscous paint model for interactive applications. *Computer Animation and Virtual Worlds 15*, 3-4, 433–441.
- BAXTER, W., WENDT, J., AND LIN, M. C. 2004. IMPaSTo: a realistic, interactive model for paint. In *Non-photorealistic animation and rendering*, 45–148.
- BERNS, R. S., AND DE LA RIE, E. R. 2003. The effect of the refractive index of a varnish on the appearance of oil paintings. *Studies in Conservation 48*, 4, 251–262.
- CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-generated watercolor. In *SIGGRAPH*.
- EDSTRÖM, P. 2007. Examination of the revised Kubelka-Munk theory: considerations of modeling strategies. *Journal of the Optical Society of America A 24*, 2, 548–556.
- EDSTRÖM, P. 2007. *Mathematical Modeling and Numerical Tools for Simulation and Design of Light Scattering in Paper and Print*. PhD thesis, Mid Sweden University.
- EDSTRÖM, P. 2008. Next generation simulation tools for optical properties in paper and print. In *Modeling and Simulation in the Pulp and Paper Industry*, 156–169.
- GIOVANELLI, R. G. 1995. Reflection by semi-infinite diffusers. *Optica Acta: International Journal of Optics 2*, 4.
- GLASSNER, A. S. 1994. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- GOSSETT, N., AND CHEN, B. 2004. Paint inspired color mixing and compositing for visualization. In *Information Visualization*, 113–118.
- GRANBERG, H., AND EDSTRÖM, P. 2003. Quantification of the intrinsic error of the Kubelka-Munk model caused by strong light absorption. *Journal of Pulp and Paper Science 29*, 11, 386–390.
- HECHT, H. G. 1983. A comparison of the Kubelka-Munk, Rozenberg, and Pitts-Giovanelli methods of analysis of diffuse reflectance for several model systems. *Applied Spectroscopy 37*, 4, 315–403.
- KOKLA, V., PSARROU, A., AND KONSTANTINOU, V. 2006. Ink recognition based on statistical classification methods. In *Document Image Analysis for Libraries*, 254–264.

- KUBELKA, P., AND MUNK, F. 1931. An article on optics of paint layers. *Z. Tech. Phys* 12, 593–601.
- LU, J., 2014. RealPigment project page. [http://gfx.cs.princeton.edu/pubs/Lu\\_2014\\_RPC/](http://gfx.cs.princeton.edu/pubs/Lu_2014_RPC/). [Online; accessed 4-June-2014].
- MOHAMMADI, M., AND BERNS, R. S. 2006. Testing instrumental-based color matching for artist acrylic paints. Tech. rep., Rochester Institute of Technology, College of Science, Munsell Color Science Laboratory.
- OHTA, N., AND ROBERTSON, A. R. 2006. Chapter 3. CIE standard colorimetric system. In *Colorimetry: Fundamentals and Applications*. John Wiley & Sons, Ltd, Chichester, UK.
- OKUMURA, Y. 2005. *Developing a spectral and colorimetric database of artist paint materials*. PhD thesis, Rochester Institute of Technology.
- POWELL, M. J. 1987. Radial basis functions for multivariable interpolation: a review. In *Algorithms for approximation*, 143–167.
- ROBERTSON, A. R. 1990. Historical development of CIE recommended color difference equations. *Color Research & Application* 15, 3, 167–170.
- RUDOLF, D., MOULD, D., AND NEUFELD, E. 2005. A bidirectional deposition model of wax crayons. *Computer Graphics Forum* 24.
- SANDOVAL, C., AND KIM, A. D. 2014. Deriving Kubelka-Munk theory from radiative transport. *Journal of the Optical Society of America A* 31, 628–636.
- SAUNDERSON, J. L. 1942. Calculation of the color of pigmented plastics. *Journal of the Optical Society of America* 32, 727–736.
- SHAKESPEARE, T., AND SHAKESPEARE, J. 2003. A fluorescent extension to the KubelkaMunk model. *Color Research & Application* 28, 1, 4–14.
- SHEARER, J. M., AND WOLFE, M. A. 1985. ALGLIB, a simple symbol-manipulation package. *Commun. ACM*.
- VARGAS, W. E., AND NIKLASSON, G. A. 1997. Applicability conditions of the Kubelka-Munk theory. *Applied Optics* 36, 22, 5580–5586.
- WANG, C.-M., AND WANG, R.-J. 2007. Image-based color ink diffusion rendering. *Visualization and Computer Graphics, IEEE Transactions on* 13, 2, 235–246.
- WESTLAND, S., IOVINE, L., AND BISHOP, J. M. 2002. Kubelka-Munk or neural networks for computer colorant formulation? In *Congress of the International Colour Association*, 745–748.
- WRIGHT, G. 2003. *Radial basis function interpolation: numerical and analytical developments*. PhD thesis, University of Colorado.
- XU, S., TAN, H., JIAO, X., LAU, F. C., AND PAN, Y. 2007. A generic pigment model for digital painting. *Computer Graphics Forum* 26, 3.
- YANG, L., AND KRUSE, B. 2004. Revised Kubelka-Munk theory. I. theory and application. *Journal of the Optical Society of America A* 21, 10, 1933–1941.
- YANG, L., KRUSE, B., AND MIKLAVCIC, S. J. 2004. Revised Kubelka-Munk theory. II. unified framework for homogeneous and inhomogeneous optical media. *Journal of the Optical Society of America A* 21, 10, 1942–1952.