Estimating Colorfulness of an Image

Eric Chu, Erin Hsu, Sandy Yu

Psych 221/EE 362 Final Project Paper

March 20, 2007

Introduction

The colorfulness of a rendered black and white image is an indication of the distortion introduced from image sampling and processing. Image metrics such as CIELAB and spatial-CIELAB (sCIELAB) provide a basis for identifying the relative colorfulness of an image, but there is no metric to quantify the distortion. Furthermore, it is difficult to simulate whether an image will appear colorful or not. Past work includes computing colorfulness based on statistical parameters of the pixel cloud along red-green and yellow-blue axes (Palus); a spatial extension to the CIELAB color metric (Wandell and Zhang); and a colorfulness metric meant to be used real time on video streams (Hasler and Süsstrunk). Our metric is designed to evaluate the colorfulness of an image based on behavioral data integrated with a variety of classifiers to gain insight into factors that impact a viewer's perception of color distortion.

Method

The general approach to tackle this problem is four-fold. First, to obtain useable results, we collected as much behavioral data as possible. The second step was to come up with a way to measure the color distortion in an image by creating a reference image. The third step was to collect this data into a useful metric. The fourth step was to model—using standard machine learning algorithms and classifiers—the behavioral and perceptual results using the data we collected and test the model against a general scene.

Behavioral data was obtained over the course of two weeks; color distortion was measured with CIELAB, spatial CIELAB, and the Hasler colorfulness metric[1]. The classifiers used were logistic regression, support vector machines, and a Naïve Bayes classifier. All the classifiers were binary classifiers except the Naïve Bayes classifier, which allowed multivariate classification.

i. Collecting Behavioral Data

In order to create test images for behavioral testing, we used ISET to generate 100 different 'harmonic' scenes, varying the orientation from 0-90° in 10° increments and the frequency from 1-10 cycles, as seen in Figure 1.

[1] It has been pointed out to us that CIELAB and spatial CIELAB also measure deviation in the luminance axis and therefore measure color *difference* and not only *colorfulness*. Fortunately, the luminance difference between the test and reference images is constant, so while there will be numerical differences, the conclusions still hold (as the "error" is constant through all calculations). See follow-up section for more discussion.

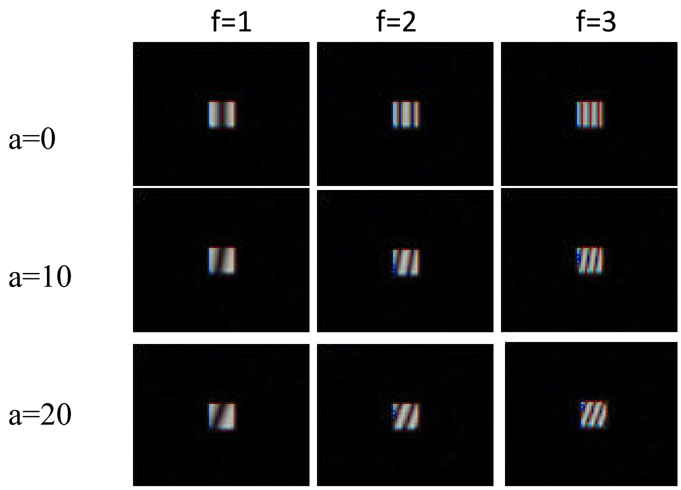


Figure 1: Examples of test images for behavioral experiments

Other than the frequency and orientation of the harmonic images, we also wanted to experiment with the viewer distance. Thus, we tested distances of 6, 12, 24, and 36 inches from the screen with 20 subjects. The subjects would view individual images and record a '1' if it appeared colored and a '0' if it appeared black and white. While some of the images looked obviously colored or not colored, many images were ambiguous and resulted in a wide array of responses.

To average out ambiguous responses, data from the subjects was tallied. If a majority of subjects said that a certain harmonic appeared colored, then we assigned it a value of '1'. Similarly, if a majority thought that a certain harmonic did not appear colored, we assigned it a value of '0.' Contested harmonics (those without a clear majority either way) were discarded for binary classification, since they would be a source of error and labeled as 'unknown' for the multivariable classifier. We hoped that this averaging would produce a general observer, such that our data would classify a general scene in such a way that all observes could agree that certain harmonics are colored and others are not. We test this assumption later by applying our classifiers to a general 'frequencyOrientation' scene.

ii. Creating Reference Images

There were two types of reference images we needed to create: one for training the classifiers and one for evaluating how accurate the classifiers were.

a. Training Reference

In order to train the classifiers, we had to create reference images that would be comparable to test images generated for the behavioral experiments. Since a variety of parameters were available for creating harmonic patterns, we were able to create the reference images with specific characteristics such as number of rows and cols.

b. Evaluating Reference

Creating the reference for evaluating the training results with a general image was less trivial. We decided to use the ‘frequencyOrientation’ scene to evaluate our system. In order to match the test images, we interpolated the reference image so that the two would have the same size, and then divided each into 64 12-by-12 blocks, as seen in Figure 2.

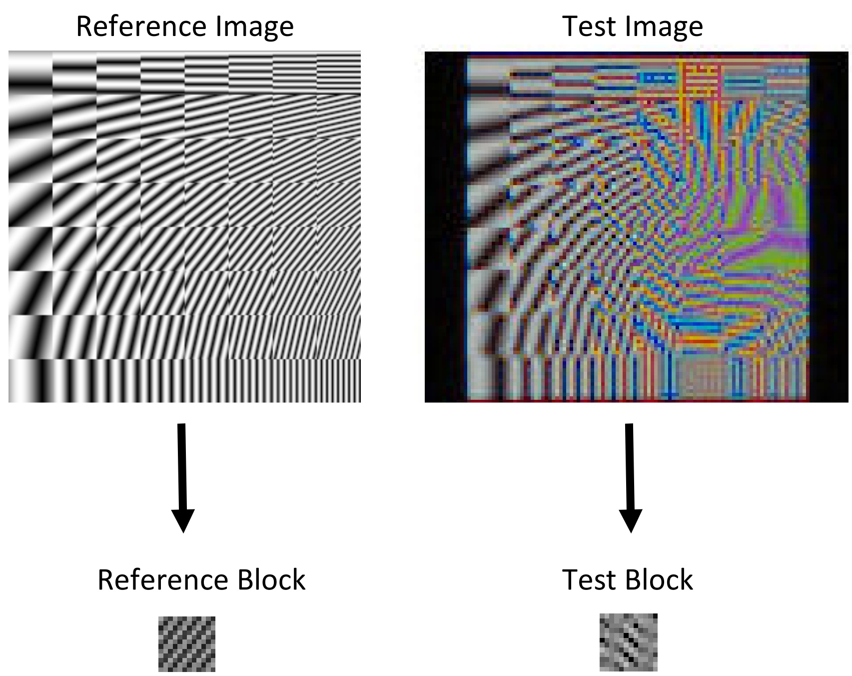


Figure 2: Creating reference image for evaluating classifiers

iii. Separating Color data

While the Hasler metric provides a single error number, both CIELAB and spatial CIELAB produce an error image. To aggregate the entire error image into a single value, we could take the average, the maximum, or other such functions of the error image. Since the appearance of color depends not only on a single pixel, but rather on a group of pixels, we decided to Gaussian blur the error image to build a sense of “locality” into the metric. This operation is redundant with spatial CIELAB, since spatial CIELAB already takes the center-surround function of ganglion cells into account. However, this blurring improves CIELAB slightly.

After blurring our error image, we then took the mean of the error image and added 3 standard deviations to it. The general idea is that if an error image has a large mean, it is more likely colored than if it had a small mean. Furthermore, if the image has a large spread of error, it is also more likely colored than an image that has a fairly uniform error across the image.

iv. Classifiers

Three classifiers were applied to the data in order to compare their effectiveness in classifying colorfulness. Two of them were binary classifiers – logistic regression and support vector machine – and the other was a multivariate classifier, Naïve Bayes.

a. Binary Classifiers: Logistic Regression

Logistic regression is a binary classifier, meaning it attempts to divide the data into two regions—a "colored" and a "not colored" region. To achieve this division, logistic regression maximizes the probability that a harmonic is colored, given the viewing distance, frequency, orientation, and error value. Mathematically, this can be expressed as $P(y = 1 \mid x; \theta)$, where y can take on the values of 0 (not colored) and 1 (colored), x is the vector containing the parameters, and θ is a vector of coefficients that describe how to calculate this probability.

Logistic regression models $P(y \mid x; \theta)$ as the logistic function, 1/(1+exp(-z))), where the argument is described by the inner product of θ and x. Hence, for a single data point x, with a given distance, frequency, orientation, and error value, the probability that it is colored is given as:

$$P(y = 1 \mid x; \theta) = 1/(1 + \exp(- \theta^T x))$$

And the probability that it is not colored is given as:

$$P(y = 0 \mid x; \theta) = 1 - 1/(1 + \exp(- \theta^T x))$$

For $\theta^T x > 0$, the probability approaches 1, and for $\theta^T x < 0$, the probability approaches 0. Hence, $\theta^T x = 0$ gives precisely the separating hyper-plane. Using some advanced calculus, the exact values of θ can be obtained from known values of x and y—that is, if we know which harmonics appear "colored" or "not colored", we can use that data to find the coefficients θ. Afterwards, we can use θ to predict whether any (general) harmonic is "colored" or "not colored" by calculating $\theta^T x$ and seeing if it is a negative or positive number.

b. Binary Classifiers: Support Vector Machine

The support vector machine is also a binary classifier that works by finding the separating hyper-plane which maximizes the size of the margin (distance) between the "colored" and "not colored" samples. It involves a complicated optimization algorithm and cannot be efficiently programmed in Matlab; fortunately, there is a free version of the support vector machine called SVM_light (Joachims), which can be downloaded off the internet. This program was used to run our classifications.

As with logistic regression, the support vector machine allows us to separate the data with a hyper-plane and store the coefficients for the hyper-plane. Hence, to compare whether a new sample is "colored" or "not colored," we plug it into the equation for the hyper-plane and see if it is positive or negative. The advantage of the support vector machine, however, is that it can model non-linear functions. Instead of using a separating hyper-plane, we could potentially use a separating polynomial or a circle.

c. Multivariable Classifier: Naïve Bayes

The Naïve Bayes classifier, also known as Idiot's Bayes, was another classifier applied to the data, chosen because it is a multivariable classifier. This allowed the option of categorizing the data into more than the two bins used before – colored and uncolored. The points could now be sorted into three bins – colored, unknown, and uncolored. Naïve Bayes is a simple, probabilistic classifier that applies Bayes Theorem with strong independence assumptions. The probability model is a conditional model over a dependent variable C, the number of bins, conditional on several feature variables, which are distance, frequency, orientation, and error. However, unlike the other classifiers, the result is not a dividing line or hyper-plane but simply a classification of data points into different bins. One advantage of using Naïve Bayes is that it requires only a small amount of training data to estimate the parameters needed for classification. Once the classifier has been trained, it runs a simple computation on any set of data and decides on the most likely bin using the following formula:

$$\mathrm{classify}(f_1,\dots,f_n) = \mathrm{argmax}_c\ P(C=c)\prod_{i=1}^{n} p(F_i=f_i|C=c)$$

Initial Results

Each of the three classifiers – logistic regression, support vector machine, and Naïve Bayes – was run on values from the three different metrics, resulting in nine different outputs. Initially, the classifiers were tested by using the training data as the input to give a visual idea of how well they worked.

i. Classifier Results

a. Logistic Regression

Figures 3-5 show the result of logistic regression and the hyper-plane that separates the data. Every point is labeled as not colored (blue 'o') or colored (red 'x') according to the results of the behavioral experiments. Keep in mind that the data is in four dimensions, while the figures only show two-dimensions. One can get a sense of how effective this classifier is by seeing how well it separates the data into a "colored" and "not colored" half.

For the most part, the algorithm appears to work well on our data. Below is the equation for the best separating hyper-plane, defined as the one that produced the least error.

$$\text{Set } dist^2 + freq^2 + orien^2 + error^2 = 1, then:$$

$$-3.7197 \times dist + 19.7184 \times freq + 0.1108 \times orien + 2.399 \times error + 0.0944 = 0$$

From the above equation, coefficients that are larger probably play a more important role in classification, and smaller coefficients have less of an effect. One would expect orientation to contain little or no information about the colorfulness of a harmonic. Furthermore, if one rearranges the equation and ignores the frequency and orientation term (set them to 0), one gets the following:

$$error = 1.551 \times dist - 0.0394$$

This equation suggests that the error threshold is directly proportional to distance, which is as expected—the closer one is, the lower the error threshold needed to perceive color distortion.

Of note is the separating hyper-plane for spatial CIELAB. As spatial CIELAB already contains distance information, one would expect the hyper-plane to be fairly constant with respect to distance. In figure 4, the separating hyper-plane is fairly close to a vertical line, with only a dependence on the error value and very little dependence on distance. Hence, we can safely conclude that spatial CIELAB is working as intended.

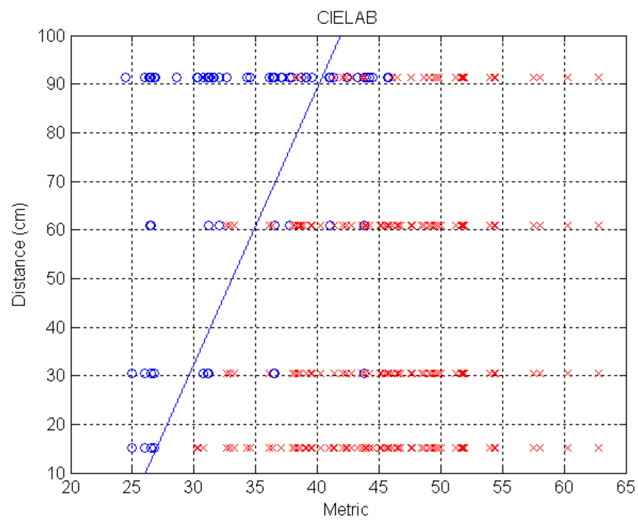


Figure 3. Logistic Regression hyper-plane using CIELAB values

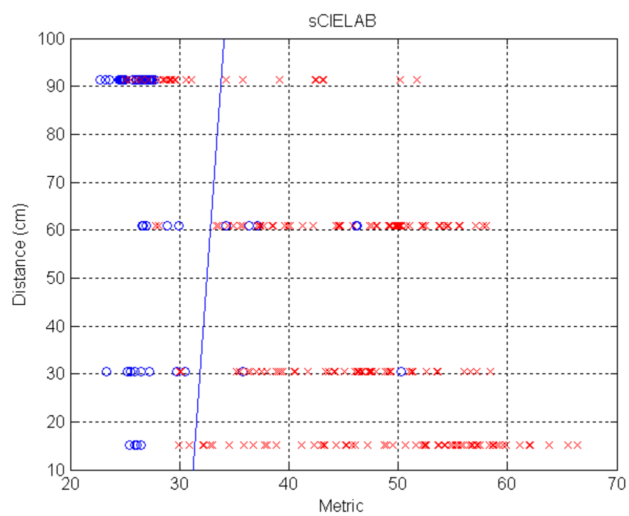


Figure 4. Logistic Regression hyper-plane using sCIELAB values

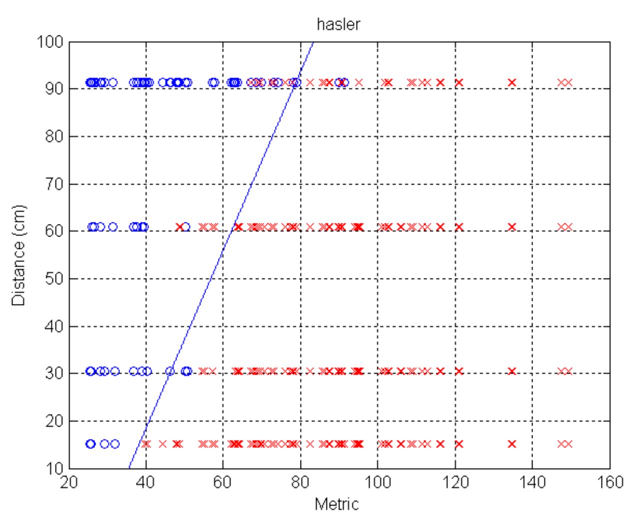


Figure 5. Logistic Regression hyper-plane using Hasler values

b. Support Vector Machine

Figures 6-8 show the result of the support vector machine algorithm. The dotted lines represent the "support vectors"—or extent of the margin. Again, every point is labeled as not colored (blue 'o') or colored (red 'x') according to the results of the behavioral experiments. Remember that this is a two-dimensional plot of a hyper-plane in four-dimensions, so the dividing lines may seem slightly out of place. One can interpret the "range" between the dotted lines as the "uncertain" area. If a point falls into this area, the algorithm is less "certain" that it is colored (or un-colored) than if it fell outside this range. Nonetheless, we indiscriminately consider all points that fall on the left of the line as non-colored and those on the right as colored for purposes of our classification (and for simplicity's sake).

The best separating hyper-plane for the support vector machine is the following.

$$\text{Set } dist^2 + freq^2 + orien^2 + error^2 = 1, then:$$

$$-2.7202 \times dist + 2.1072 \times freq + 0.0475 \times orien + 4.01 \times error - 0.2472 = 0$$

Again, one can gauge the relative importance of each parameter by looking at the absolute value of the coefficients.

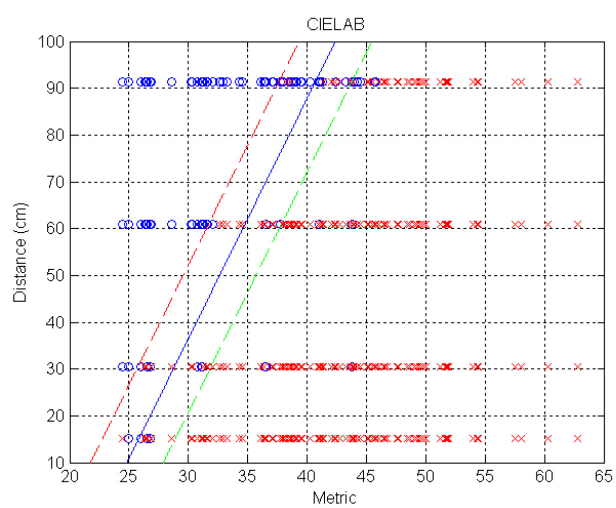


Figure 6. SVM hyper-plane and support vectors using CIELAB values

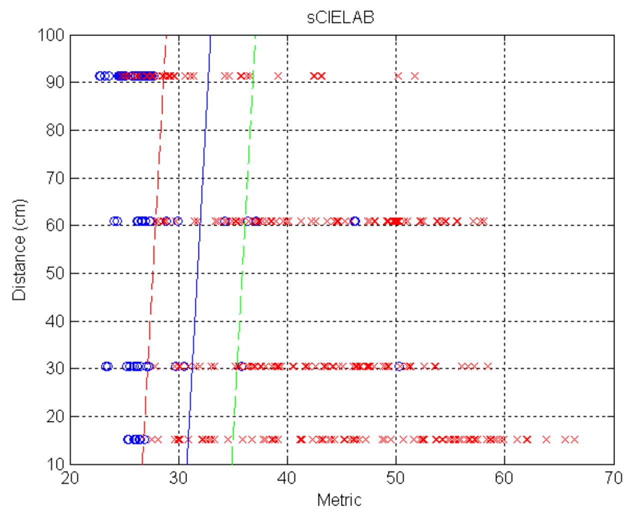


Figure 7. SVM hyper-plane and support vectors using sCIELAB values

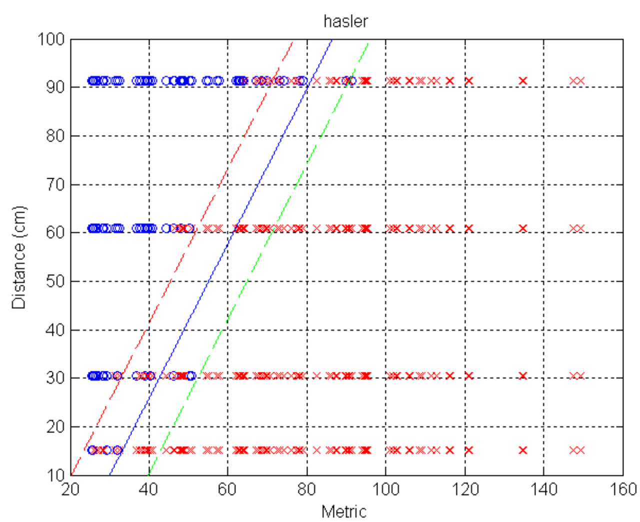


Figure 8. SVM hyper-plane and support vectors using Hasler values

c. Naïve Bayes

The results from running the three metrics with the Naïve Bayes classifier are displayed in Figures 9-11. As can be seen, the Naïve Bayes classifier does not generate a dividing hyper-plane like the binary classifiers. Instead, it takes advantage of the multivariable nature by classifying each point as being colored, uncolored, or unknown. The blue circles represent uncolored, the green triangles are unknown, and the red X's are colored. This classifier appears to be relatively conservative, as many points end up being classified as being unknown, especially at the smaller distances.

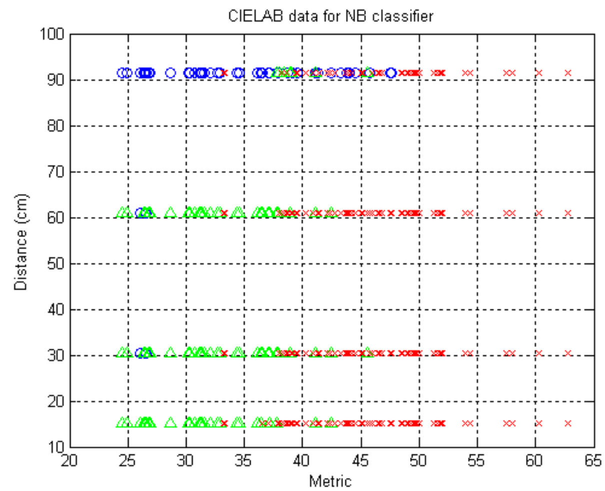


Figure 9. Naïve Bayes classifier results using CIELAB values

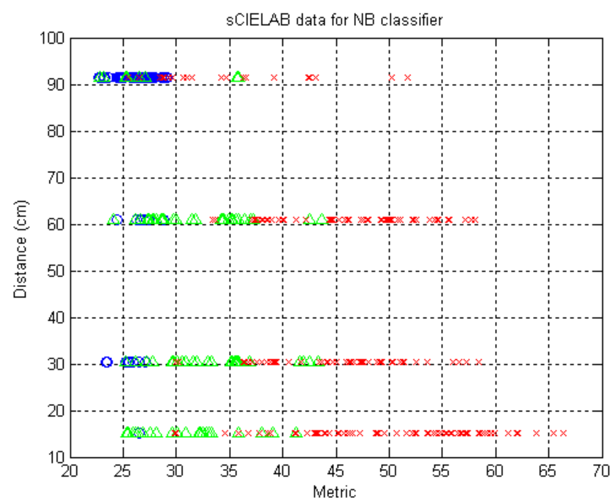


Figure 10. Naïve Bayes classifier results using sCIELAB values

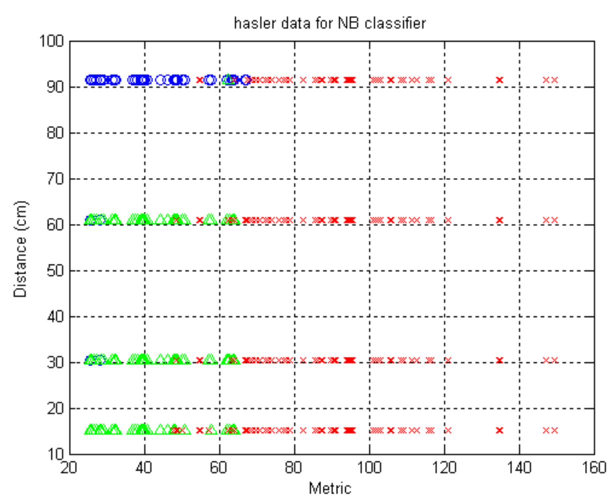


Figure 11. Naïve Bayes classifier results using Hasler values

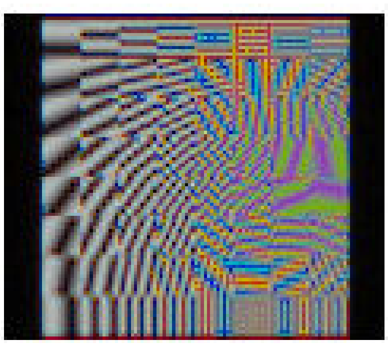
ii. Metric Results

To get a sense of how "good" each error metric is in helping us determine whether a harmonic is colored or not, we unleashed the classifier on a general scene composed of harmonics. The metric that gave the least error was most likely to be the "better" metric. One can apply this general idea to measure the effectiveness of particular metrics (not just these specific ones)—devise a metric to measure "colorfulness," train a classifier using the behavioral data and the new metric, and test the classifier on general data. The best metric will help the classifier correctly label the most harmonics.

Initially, one would expect CIELAB to perform the worst, as it provides no spatial frequency and distance information. However, we have included that information in our classifier, so it is possible that the extra information will help our distinction between colored and non-colored harmonics. Spatial CIELAB, because of all the information it contains, is expected to perform the best; nonetheless, there is plenty of redundant information included in our classifier, such as frequency, distance, and even the blurring of the error image may have unknown consequences on the spatial CIELAB values. Hasler, et al. claimed that their metric had a 94% correlation with experimental, perceptual data (Hasler), but, it, too, did not take spatial frequency and viewing distance into account. As with CIELAB, the extra information is accounted for in our other dimensions, and perhaps that may improve the results of the Hasler metric.

iii. General Test Results

In order to test the classifiers and metrics on a general image, the frequency-orientation scene in ISET was used, as shown in Figure 12 below. The reference image was created as mentioned in Methods, and a truth table at 24 inches declaring each block to be colored or uncolored was agreed upon by all three group members. The data was run through all nine combinations of classifiers and metrics, resulting in the errors in Table 1.



```
[ 0  0  0  1  1  1  1  1;
  0  0  0  1  1  1  1  1;
  0  0  0  0  1  1  1  1;
  0  0  0  0  1  1  1  1;
  0  0  0  1  1  1  1  1;
  0  0  0  1  1  1  1  1;
  0  0  0  1  1  1  1  1;
  0  0  0  1  1  1  1  1 ]
```

Figure 12. Frenquency-orientation scene from ISET and truth table

	CIELAB	**sCIELAB**	Hasler	*Average*
Logistic Regression	0.0313	0.0313	0.0625	***0.0417***
SVM	0.0625	0.0938	0.0938	*0.0834*
Naïve Bayes	0.1797	0.1250	0.1719	*0.1589*
Average	*0.0912*	***0.0834***	*0.1094*	

Table 1. Test results from different combinations of classifiers and metrics

From these numbers, it is apparent that our initial data averaging to create a general observer seems to have worked. Our classifiers are correctly predicting whether a harmonic appears colored or not when looked at by any observer—or when agreed upon by a certain number of observers. Furthermore, logistic regression is the best classifier out of the three in classifying the data. It averages a 4.17% error, while SVM and Naïve Bayes average 8.34% and 15.89% errors, respectively. Moreover, spatial CIELAB has the best overall performance, with an average of an 8.34% error, while CIELAB has a slightly higher error of 9.12% and Hasler has the highest average of 10.94%.

However, it can be observed that with logistic regression, CIELAB and spatial CIELAB actually have the same error of 3.13%, which is unexpected because spatial CIELAB is supposed to have a much better performance than CIELAB, as it takes frequency and distance into account. This phenomenon can be explained by the fact that the error metric includes an extra Gaussian-blurring of the error images as well as frequency and distance parameters, which is already taken into account in spatial CIELAB. Perhaps, by adding the blurring, sCIELAB's success was diminished, while CIELAB was improved, equalizing their error numbers. In the next section, we test the relative importance of each parameter in deciding whether a harmonic is colored or not.

Follow-Up Results

After the presentation, there were several suggestions made regarding the parameters used in our classifications. The code was modified to test the effects of removing the error parameter as well as trying different variations of existing parameters. Since the original results found that the logistic regression classifier was the most successful of the three classifiers, the variations were only tested with this classifier on the general frequency-orientation image. Table 2 below shows the different variations as compared with the previous results.

	CIELAB	sCIELAB	Hasler	Average
Previous results	0.0313	0.0313	0.0625	0.0417
Using only error values (Control)	0.0625	0.1250	0.0938	0.0938
Removing error parameter	0.0156			**0.0156**
Removing distance	0.2500	0.2969	0.2813	0.2761
Removing frequency	0.1406	0.3438	0.1094	0.1979
Removing orientation	0.0469	0.0625	0.0625	0.0573

Table 2. Correlation results from removing individual parameters

This table contains a dense amount of information. The top row holds the results from the previous tests, which took into account distance, frequency, orientation, and error. Next, the error values were classified independently, demonstrating that the previous metric was an improvement on simply using CIELAB, sCIELAB, or the Hasler metric by themselves. The most significant improvement occurred when the error parameter was completely removed, and the classification was just based on distance, frequency, and orientation. In fact, this classification only made one mistake out of 64 points, for an error of only 1.56%!

In order to more easily see the effect of each parameter on the success of the classification, each parameter was successively removed, and the error was calculated. From these numbers, it appears that distance and frequency are the most important parameters, since the error skyrockets when either is removed. Removing the orientation only slightly increased the error in comparison. However, it is surprising that the error for spatial CIELAB increases so much when spatial frequency is removed, because this value should already be factored into the sCIELAB error calculation, which is still included in the classification. Perhaps the small image size causes problems for spatial CIELAB in calculating the spatial frequency. It is also possible that the luminance difference played a large role in the spatial CIELAB calculations, which we overlooked earlier.

To examine the effect of the “luminance” information in our error metrics for CIELAB and sCIELAB, we removed the delta-L values from deltaEab calculations. Table 3 displays the same results as in Table 1, but with luminance values zero-ed out in the deltaEab calculations for CIELAB and spatial CIELAB.

	CIELAB	**sCIELAB**	Hasler	*Average*
Logistic Regression	0.0313	0.0313	0.0781	***0.0469***
SVM	0.2500	0.1094	0.0781	*0.1453*
Naïve Bayes	0.1953	0.1172	0.1641	*0.1589*
Average	*0.1589*	***0.0860***	*0.1068*	

Table 3. Test results from different combinations of classifiers and metrics ignoring the delta-L values.

A quick comparison with Table 1 reveals some interesting information. First, our conclusions about the classifiers remain unchanged—logistic regression still performs the best overall. Furthermore, our conclusions about the performance of spatial CIELAB also hold. Not only that, but by removing the luminance data, we see that spatial CIELAB far outperforms CIELAB, especially for the support vector machine classification (scoring almost 15% more accurately than CIELAB). Hence, while we have different numerical results, the general conclusions about the classifiers and the metrics still stand. In addition, we have quantitative proof that spatial CIELAB reflects perceived color better than the other two metrics, which do not take spatial frequency and distance directly into account.

These calculations were very helpful in exposing the roles of different parameters in the classification metric. Removing the error parameter completely proved to be a good suggestion, since it decreased the error significantly. Like the coefficients of the logistic regression suggested, distance and frequency played major roles in classification, while orientation did not. Furthermore, the inclusion of luminance in the error images did not greatly degrade our performance, but did mask the effectiveness of spatial CIELAB in distinguishing between colored and non-colored harmonics.

Conclusion

After training our system with experimental data using CIELAB, spatial CIELAB, and the Hasler metric, followed by classifying the data with logistic regression, support vector machine, and Naives Bayes algorithms, and finally testing our results with reference images, we were able to quantitatively classify perceived color with at least 90% correlation. Logistic regression turned out to be the quickest and best classifier, as the coefficients could be calculated iteratively and then stored efficiently to calculate the separating hyper-plane for future classification. The support vector machine allowed for a range of hyper-planes while the Naïve Bayes algorithm enabled a range of response other than just “colored” and “not-colored.”

In terms of error calculations, CIELAB and spatial CIELAB performed similarly, with the Hasler metric close behind. Theoretically, spatial CIELAB is the best metric for measuring color perception, but experimentally, Gaussian blurring during our process could have enhanced CIELAB and diminished advantages of spatial CIELAB. Furthermore, the inclusion of extra dimensions into the CIELAB classification may have improved it, while the redundant dimensions introduced to spatial CIELAB may not have contributed at all. The Hasler metric was close behind CIELAB and spatial CIELAB, but did not require a reference image to begin with. Thus, it is important to consider applications and resources when deciding which metric to use. If one is willing to take a hit on accuracy for simplicity, the Hasler metric may perform quite well instead of CIELAB or spatial CIELAB. However, with our follow-up results, it becomes apparent that the error values are not actually necessary to create the metric. In fact, the best results occurred with only frequency, distance, and orientation were taken into account with logistic regression.

Although our experiments produced good results, there are several sources of error to keep in mind. First is the skewed behavioral data. Our sample size was small and not enough to account for different thresholds for colorfulness that the participants had. There was also the effect of different displays since we did not conduct all experiments using one monitor. Furthermore, there was aliasing in spatial frequency that could have easily been mistaken for colorfulness. Second is the accuracy of reference images. Certain assumptions, such as a 5000K white point, were made and might not have been reflective of actual situations. Third is the assumption that a linear relationship exists between errors and distance, frequency, and orientation. We calculated a hyper-plane based on this assumption but the data could have also been nonlinear, especially if we consider adding an additional parameter. And, fourthly, the inclusion of the luminance difference in CIELAB and spatial CIELAB calculations may have contributed to some errors in the classifiers; though it appears to have affected CIELAB less and sCIELAB more as can be shown in the results in our follow-up section.

The next logical step is to consider how to determine whether certain areas of the image will appear colored or not, given a general black and white scene that one is trying to reproduce. To apply this metric, using the frequency and orientation information, it seems natural to take the two-dimensional Fourier Transform of the image and apply the metric on the harmonics individually. The details of this have yet to be worked out; but the general framework is to convert the image into its constituent harmonics and determine colorfulness with this classifier and reconstruct the image with the appropriate "colored" harmonic information.

Moreover, there are some ways that this colorfulness metric can be improved. First, the classification could be made more accurate by testing the effect of other parameters, such as the field of view of the scene, brightness, luminance, or contrast. Another possibility is to try a nonlinear classification, since the raw behavioral data appeared to have the potential of an exponential relationship. The SVM package provides the capability of a nonlinear relationship, so it would not be too hard to apply.

The experiment could also be repeated in depth with the error terms containing only the delta-a and delta-b norms and classification based on the error values alone. Theoretically, spatial CIELAB should outperform the others here, as it will contain distance information. From our follow-up experiments, we have shown that spatial CIELAB does, indeed, outperform the others. More experiments can be done to compare the classifier based on the error value alone to the classifier using only distance, frequency, and orientation. Since color perception is non-linear, it is likely that the model using only distance, frequency, and orientation alone may fall apart for large values of distance and frequency—and its current accuracy may be attributed to the fact that our test images operated in a small, linear region of classification. Information from (spatial) CIELAB would improve the classifier by providing uniform color information, allowing the linear model to hold true for more general cases (instead of the more narrow one).

All in all, we have shown that in the perception of colorfulness, distance and spatial frequency have a great effect. Removing the error values didn't affect our classifiers because the error information was redundant with the distance and spatial frequency information. Nonetheless, as we are using linear models to determine color perception, distance and spatial frequency alone may prove inadequate to correctly classify colored and non-colored harmonics for larger values. Hence, spatial CIELAB (using only its delta-a and delta-b values) may be the better metric after all, for a more general case of colorfulness, as it provides linear data about color perception and includes distance and spatial frequency information. One may also try to improve the Hasler metric (which can be computed quickly from sRGB values) by including spatial filters (as in spatial CIELAB) to get a quick estimate of the colorfulness of pictures. And, finally, we can evaluate the "accuracy" of a metric by using classifiers (or machine learning) to evaluate its correlation with perceptual data. Eventually, machine learning algorithms may be able to determine (without any CIELAB calculations) the colorfulness of an image without having to have human users evaluate the image.

References

1. Hasler, D. and Süsstrunk, S. Measuring Colourfulness in Natural Images. Proc. of SPIE v. 5007, p. 87-95.

2. Joachims, Thorsten. SVM-light Support Vector Machine. ver. 6.01, 9 Feb 2004 <http://www.cs.cornell.edu/People/tj/svm_light/>

3. "Naive Bayes classifier." *Wikipedia, The Free Encyclopedia*. 12 Mar 2007, 18:28 UTC. Wikimedia Foundation, Inc. 17 Mar 2007 <http://en.wikipedia.org/w/index.php?title=Naive_Bayes_classifier&oldid=114597297>.

4. Ng, Andrew. Lecture. CS 229, Stanford University, Stanford, CA. Oct 2006. Classification and Logistic Regression. < http://www.stanford.edu/class/cs229/notes/cs229-notes1.pdf>

5. Ng, Andrew. Lecture. CS 229, Stanford University, Stanford, CA. Oct 2006. Support Vector Machines. < http://www.stanford.edu/class/cs229/notes/cs229-notes3.pdf>

6. Palus, Henryk. Colorfulness of the image: definition, computation, and properties. Proc. of SPIE v. 6158, April 20, 2006.

7. "Support vector machine." *Wikipedia, The Free Encyclopedia*. 13 Mar 2007, 14:43 UTC. Wikimedia Foundation, Inc. 17 Mar 2007 <http://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=114804148>.

8. Wandell, B. and Zhang, X . A spatial extension of CIELAB for digital color image reproduction. Society for Information Display. 1996.

Appendix I

In this section, there are links to the code that was written, the SVM_light package, as well as a link to the data files we produced. We assume that the user of these scripts have the ISET toolkit installed on their computers.

Scripts – These are scripts we produced to ease the collection of error data and behavioral data.

> **Tester.m** – This script was used to produce 100 harmonics for subjects to evaluate as "colored" or "not colored." Subjects press any key to proceed to the next harmonic.
>
> **measError.m** – This script was used to calculate the CIELAB, sCIELAB, and Hasler error values for each harmonic. The error values were saved into a data file called errors_final.mat, which can be found in the data zip file.
>
> **freqOrientTest.m** – This script was used to evaluate the classifiers by running the classifiers against perceived color evaluations. The percentage error is shown at the end of the function.

Classifiers – These functions were used to (a) create the classifiers and (b) use the models to classify data.

> **logisticRegression.m** – This function is called without any inputs and creates the best separating hyper-plane for the data according to logistic regression. It does this for all three types of error values.
>
> **SVM.m** – This function is called with a single input (which describes the name of the model) and creates the best separating hyper-plane for the data according to the support vector machine algorithm. It implicitly calls SVM_light. It also does this for all three error types.
>
> **naiveBayes.m** – This function is called without any inputs and generates the naïve Bayes model for all three error types.
>
> **classifyLR.m** – This is the logistic regression classifier. It takes a string 'type' as an input (describing which error type to use for classification) and a vector or matrix that contains the data in the proper format. It returns whether it thinks the given data describes "colored" or "not colored" harmonics.
>
> **classifySVM.m** – This is the support vector machine classifier. It also takes a string 'type' as an input (describing which error type to use for classification) and a vector or matrix

that contains the data in the proper format. It returns whether it thinks the given data describes “colored” or “not colored” harmonics.

classifyNB.m – This is the naïve Bayes classifier. . It also takes a string ‘type’ as an input (describing which error type to use for classification) and a vector or matrix that contains the data in the proper format. It returns whether it thinks the given data describes “colored”, “unknown”, or “not colored” harmonics.

Data preparation – We collected our data in 10x10 matrices, where the columns represented varying frequencies and the rows represented varying orientations. Each element had a corresponding 1 (colored) or 0 (not colored) label as well as its error value. For each distance, a different 10x10 matrix was found.

prepareData.m – This takes all the behavioral data and the error data and reformats them into a matrix of 100 rows (for each distance). Each row contains four columns, where each column corresponds to “distance”, “frequency”, “orientation”, and “error value.” To see the relative effects of removing a particular feature (such as distance), one can change the formatting in this function. It also returns a column vector whose elements describe whether a particular row in the returned matrix is colored or not.

reformatData.m – This takes the data we are trying to classify and returns it as an appropriate matrix for classification. For example, this function was called in the freqOrientTest.m script in order to re-arrange the data for classification. This function must return data in the same format that the function prepareData returns, as it essentially performs the same operations without producing the labels.

Miscellaneous files – Here you will find links to the miscellaneous data files and the SVM_light program.

SVM_light – This program is free for non-commercial use, as specified by Joachims on his website. It contains the executables for running the support vector machine in Matlab.

Data files – This zip file contains all the behavioral data we collected in addition to the models generated by the classifiers. One may need to change the directories that SVMmodels.mat points to for the SVM to work properly.

Appendix II

Eric Chu

- Data preparation (CIELAB, sCIELAB) code
- Logistic Regression Classifier code
- Support Vector Machine Classifier code
- Presentation development and delivery
- Report write-up

Erin Hsu

- Behavioral experiments and code
- Naïve Bayes Classifier code
- Hasler metric research and code
- Presentation development and delivery
- Report write-up

Sandy Yu

- Background research
- Behavioral experiments
- Reference image development code
- Presentation development and delivery
- Report write-up