The Development of a Color-based Musical Instrument

Samuel Stone
Scott Craver
Department of Electrical and Computer Engineering
Binghamton University
Binghamton, NY
sstone2@binghamton.edu

Abstract

MIDI controllers can be expensive and offer limited configuration options. In this paper, we propose an electronic music instrument based on color. This instrument allows the user to turn any surface into an instrument through colored cells or on any object through surface color mapping using MLE and color-space transforms.

Index Terms

Image color analysis, Computer generated music, Matlab

I. Introduction

This project grew out of a recurring senior project at Binghamton University, to produce a novel MIDI controller. One proposed design was a device like a xylophone mallet, containing a color sensor; it would then play a pitch depending on the color of the object struck.

Such an instrument would have a fascinating utility: one could use it to play an arbitrary ”keyboard” of color swatches or colored objects, or instead use it to play a pre-printed melody, consisting of color patterns printed on paper. It could also provide a musical experience between these two extremes: one could print out a piece of music containing color sequences representing phrases, along with scales and arpeggios in certain keys relevant to the piece, allowing a musician to play them improvisationally.

Initial experiments with this design determined that such an instrument is not simple to build. A device must be able to sense the color of an object as it is striking it, which introduces several problems. One is illumination of the target object, which is occluded by the device. Another is the swift and imprecise tapping of the target object; a sensor must have a visual field wide enough to capture the target object, while excluding colors of the surrounding environment. The target object itself may exhibit specular reflection, and variations in materials, even variations in printing processes, can produce a challenge for consistent color classification.

For use in a musical instrument, a key requirement is consistency. Absolute accuracy in color measurement is desirable, but less important: a user can calibrate a color sensor to classify a certain object as a musical note, for example, but the instrument is not usable if this classification occurs with a macroscopic error rate.

II. Color Analysis

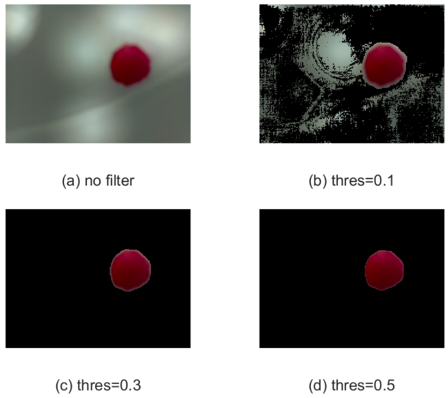
An important factor in the way the color classifier works is the correspondence between the computer and user’s interpretations of surface color. The basic principle of the device is that a person can select two colored surfaces which, to them, seem very similar or identical in color and the device will also recognize this similarity. To mitigate the human-computer perception difference and maintain program speed, we selected the HSV color space for image analysis.

A. Color Spaces and human perception

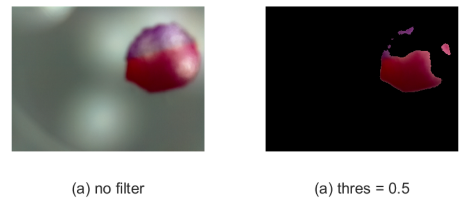
According to [2], the HSV color space was developed to replicate the way humans perceive color, while maintaining fast processing speeds. Although spaces like CIE L*A*B* would more closely represent color in the way humans perceive it, the linearity of hue makes probability calculations simpler. Future work may show a significant increase if effectiveness with CIE L*U*V*.

B. Color Saturation Filtering

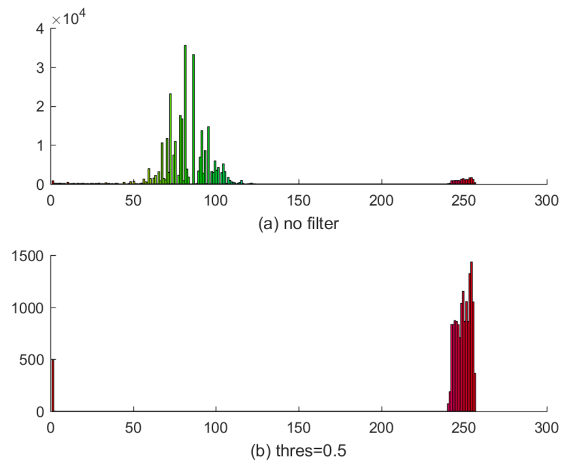
After the image is captured and transformed to the HSV color-space, we need to filter the resulting image for the correct sample area. Using the color diffusers ead., the majority of the image does not represent the sample. By only capturing pixels where the saturation is above a certain threshold, we can easily both isolate the sample region and eliminate any glare caused by the led lights. Figure 1(a) shows the effect of the saturation threshold filter on the output image. Figure 1(b) illustrates the



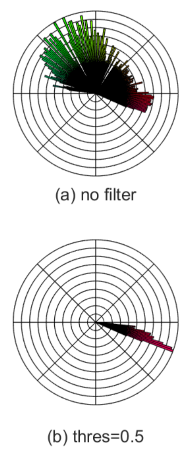
(a) Effect of saturation filter



(b) Saturation filter glare elimination



(c) Linear Color Histogram



(d) Circular Color Histogram with log scale

saturation filter's effectiveness on removing glare from a glossy surface. The analyzed image was taken on a glossy print that would negatively affect the classifiers ability to correctly identify the color.

We also created a number of computer visualization tools to not only tune the performance of the filters and classifiers, but also diagnose problems with the analysis. Figure 1(c) shows a linear histogram of the a captured image both before (a) and after (b) filtering. Unlike existing histogram tools, our tool uses the corresponding colors for each segment of the histogram to better show its results.Figure 1(d) shows this same data plotted on a polar axis, making it easier to see proximity in the circular hue values. The general threshold value of 0.5 was selected through an analysis of color spanning the hue spectrum. For each color, we increased the threshold until all of the non-surface pixels were eliminated. A graphical computer tool was developed in Matlab to accelerate this process.

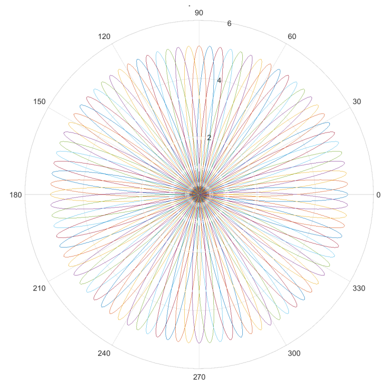
III. Mathematical Methods

A. Color Classifier

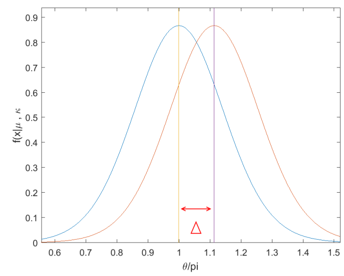
The in order to select the number of notes the instrument can play, we must adjust the width of each note's selection range. Because the hue value is circular, a traditional normal distribution could not be used. Instead, we opted for the Von Mises distribution [4].

$$f(x|\mu,\kappa) = \frac{e^{\kappa\cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{1}$$

The Von Mises distribution allows us to determine the likelihood of any hue value falling within a a particular predetermined classification. By employing Maximum Likelihood Estimation [3], we can determine the most likely distribution for an entire data set.



(e) 88 classification circular MLE



(f) Adjacent classification overlap

The log likelihood ratio test greatly simplifies calculation [3].. We assign each classification a median angle from the set Λ. For each class, we find the probability that all of the m data-points in Γ, the hue values, fall within its distribution. The estimated probable mean is the one with the highest probability. Utilizing the log ratio [3], comparing the probability of any point to the maximum (eq.3), greatly simplifies this calculation by eliminating the first order Bessel function.

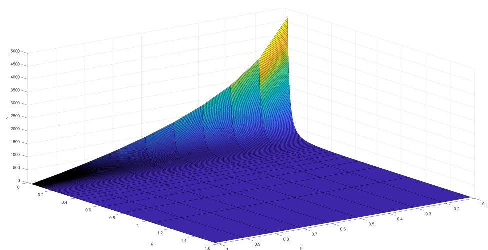
$$p_\mu(x) = \frac{f(x|\mu,\kappa)}{f_{max}} = e^{\kappa(\cos{(x-\mu)}-1)} \tag{2}$$

$$\log p_\mu(x) = \kappa(\cos{(x-\mu)} - 1) \tag{3}$$

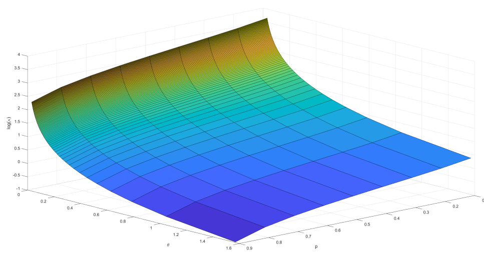
$$\hat{\Theta}_{ML} = \max_{\mu\in\Lambda} \sum_{j=1}^{m} \log p_\mu(\Gamma_j) \tag{4}$$

In figure 1(e), we show a classifier that could capture each note on a standard 88 key piano. Each classification overlaps an amount determined by a minimum difference for detection ead. The number of categories can be adjusted to a limit determined by the minimum difference.

B. Optimizing P and κ for optimal detection



(g) Relationship between p and curve width on κ



(h) Relationship between p and curve width on $\log \kappa$

$$\kappa = \frac{\log p}{\cos x - 1} \tag{5}$$

As the value of p becomes greater, the adjacent curves become increasingly overlapped. For any θ value on the curve, the greatest likelihood would always be for the closest curve. Tuning this overlap precisely would allow us to have a lower κ value and a more accurate classifier. In figure 1(h), we see that the range o κ is much lower when the overlap is large. The problem with significant overlap is that it makes it more difficult to detect the difference between two adjacent classifications. In figure 1(f), we pick a value of θ, ie zero, the difference between the likelihoods of the two adjacent classifications should be significant enough for detection.

$$f(0,0,\kappa) - f(0,\Delta,\kappa) > 0.05 \tag{6}$$

$$\Delta_{min} = \arccos\left(\frac{\log\left(2\pi I_0(\kappa)0.05 + e^{\kappa}\right)}{\kappa}\right) \quad (7)$$

The minimum difference depends greatly on the value of κ. For low values of κ, the Von Mises Distribution approximates the uniform distribution [2]. This forces the minimum difference between each category to be very large. At values of κ above 100, the minimum angle becomes very small.

C. Mapped Color Categorization

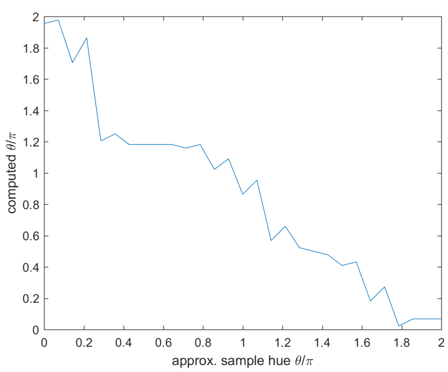
After the color category for the image is determined, each category is mapped to an output note. Each category can be individually assigned a unique note through a note-color calibration program or the entire set can be linearly mapped to a specific octave.

IV. Device Construction and Testing

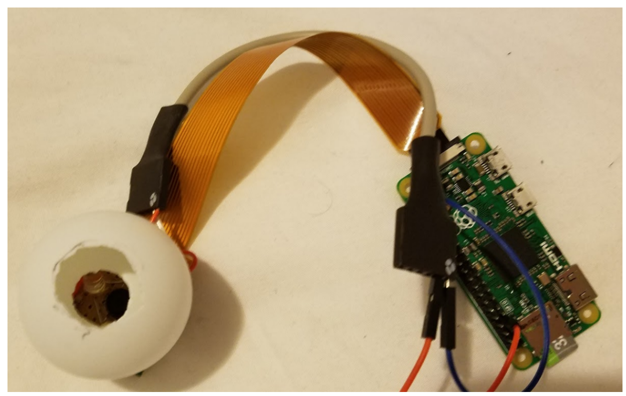
A. Construction

Computation is done by a Raspberry Pi zero with a camera. The light coming into the camera and its field of view are carefully controlled through an led light ring and series of diffusers. The inner and outer diffusers are made from sections of ping pong balls. By mitigating noise from the outside environment and the highly directional aspects of the LED's, we improve the accuracy of the classifier.

To activate the color classifier/sound output, the user presses a small button. This signals the program to begin collecting and analyzing image sensor data. In its current configuration, the settings of the camera can be automatically adjusted to eliminate the possibility of sensor differences between sessions.



(i) Color Sample Test



(j) Picture of Device

B. Verification

In order to verify the accuracy of the classifier, we implemented a matching test between known hue value surfaces, printed with maximum saturation and value, and the identified result. One of the most important aspects of this device is its ability to repeatedly identify colors, regardless of the media.

As one can see from the results of figure 1(i), the device is able to repeatedly identify surface colors. In the test, we sampled a printed grid of colors ranging from zero to 255 in hue value. Ideally, each sample would correspond directly to a color selection. We believe the error seen is due to inaccuracy of the sample selection and not due to the classifier itself due to the relative inconstancy.

V. Conclusion

By implementing Maximum Likelihood Estimation on the HSV transformed data-set collected from a camera with an optically diffused light source, we were able to successfully implement a general surface color estimator. The output of this estimator can be re-mapped by the user and its output can be applied to a MIDI controller or used to synthesize sound by the device. Future work includes optimization of the processing to minimize any delay.

References

[1] Thorpe, Simon, Denis Fize, and Catherine Marlot. "Speed of processing in the human visual system." nature 381.6582 (1996): 520.

[2] Ebner, Fritz, and Mark D. Fairchild. "Development and testing of a color space (IPT) with improved hue uniformity." Color and Imaging Conference. Vol. 1998. No. 1. Society for Imaging Science and Technology, 1998.

[3] Poor, H. Vincent. An introduction to signal detection and estimation. Springer Science & Business Media, 2013.

[4] Forbes, Catherine, et al. "von Mises Distribution." Statistical Distributions, Fourth Edition (2011): 191-192.