Color Saturation: Upper and Lower Percentage Histogram Manipulation

Kyra Obert[*1], *Maria Schudt*[1], *& Ian Bentley*[1,2]

[1]*Department of Chemistry and Physics, Saint Mary's College, Notre Dame, IN*
[2]*Department of Engineering Physics, Florida Polytechnic University, Lakeland, FL*

https://doi.org/10.33697/ajur.2023.080

Students: kobert01@saintmarys.edu, mschudt01@saintmarys.edu*
Mentor: ibentley@floridapoly.edu

ABSTRACT
There are various color correction techniques that can be applied to digital photographs to account for environmental lighting variations. This manuscript contains a proposed method for such color correction. The method involves saturating an image by a specified percentage of its pixels via upper and lower percentage histogram manipulation using the image's RGB histograms. Variations of this new technique, the white balance (WB) correction method, and a multi-variable fit are used to test its performance against common color correction techniques. The findings demonstrate that the upper and lower percentage histogram manipulation method is not only more applicable to photos because it doesn't require calibration regions to be sampled but it is also more consistent in its correction of photos when there are substantial gray scale features (e.g. a black and white grid or text). Our motivation for testing these techniques is to find the most robust color correction technique that is broadly applicable (not requiring a color checker chart) and is consistent across different lighting.

KEYWORDS
Color Correction; Histogram Manipulation; Saturation; White Balance; Scientific Image Analysis; Color Comparisons; Euclidean Distance; Standard Deviation; Color Difference

INTRODUCTION
The RGB values for each pixel in a digital image range between 0 and 255 and combine to define each pixel's color. Color correction is used in digital imaging to edit an image to achieve a desired effect or remove effects due to environmental factors, such as lighting. Variations in lighting can distort colors and change the RGB values interpreted and stored in by a camera for pixels in an image.

The importance of color correction extends to general photography, film-making, and photo analyses for scientific studies. In particular, the color correction techniques compared and discussed throughout this paper are of interest to be applied to the Scientific Image Analysis (SIA) application.[1] The SIA application is currently being used for biological studies of motion and has also been produced as a standalone tracking application called BatCount[2] which also relies on the color correction technique described in this manuscript to increase the contrast on images.

Previous work in color correction

Due to its applications in both the scientific and entertainment communities, several color correction techniques have been extensively developed and compared for their intended uses. Some of the alternative correction types consider or focus on one or two specific issues in image production due to limitations in photography, such issues include flares, illumination settings, and weather.[3–10] Although white balance is commonly used in the camera industry, several other methods have been proposed and advocated for in favor over white balance. Nonetheless, white balance is frequently used as the metric to compare proposed techniques against.[4, 6, 8]

MATLAB was used to test our color correction technique. MATLAB contains a variety of functions for correction methods. One of the correction techniques used in MATLAB is *imadjust*. This function, like the technique studied in this manuscript, adjusts the intensity of image input values so that a percentage of the data is saturated to low and high intensities.[11] The default of the correction method with a grayscale image is 1%. One of the variations we make in the analysis described in this manuscript is to change this saturation percentage. When considering the various MATLAB contrast enhancement techniques in grayscale, *imadjust* is also compared to the *histeq* and *adapthisteq* functions. The function *histeq* adjusts the histogram values of the image so that the histogram of the resulting image approximates a specified histogram, with the default setting being a uniform distribution.[11] This adjustment is referred to as "histogram equalization." The other function, *adapthisteq*, applies a similar technique to *histeq* but by analyzing smaller regions across the image to compute and make appropriate adjustments.[11] This approach is known as "contrast-limited adaptive histogram equalization" and is used to reduce the amount of noise that may otherwise be amplified through *histeq*.

The variations in these styles is used to address specific image types and issues. The *imadjust* method, in particular, is effective in having a clear effect on images whose histogram distributions are concentrated in the center of the historgram. Images that are more uniformly distributed already have RGB values closer to the ends of the intensity spectrum, reducing the effectiveness of *imadjust* creating contrast in an image.[11]

Similarly, *histeq* maintains difficulties with adjusting images with both concentrated and uniformly distributed histograms. While this method effectively reveals details hidden in an image, it often also over-saturates areas, especially those that are lighter in the original image.[11] The adaptability of *adapthisteq* makes it the most preferable method in most cases of an equally distributed histogram, as it computes for appropriate adjustments across the image while also allowing the user to limit the amount of contrast enhancement in the image.[11] The latter of these characteristics is particularly helpful in reducing the over-saturation demonstrated by *histeq*.

These techniques can also be applied to color images. In order to do this, colored (RGB) images are converted to a color space that has luminosity as one of its channels.[11] In each of the methods, only the values in the luminosity layer are adjusted before the image is converted back to RGB space. This method allows the pixels colors to be maintained while their intensity is appropriately adjusted.

Methods of white balance used in MATLAB and ones similar to these methods have also formally been tested. One report compared various automatic white balance techniques: the gray world assumption, the perfect reflector algorithm, and a proposed Gaussian distribution technique.[12] The gray world assumption concludes that if there is enough color in an image, then the average values of each R, G, and B tend to be the same. However, if the color in an image does not follow this specification in variety, this method often fails. In the perfect reflector algorithm, the brightest area in the image is considered to be standard white, so the average value of this area is used for cast correction. In this case, this method fails when the brightest point in the image cannot be concluded or the image lacks a white area. The proposed method is to superimpose multiple Gaussian distributions to represent the histogram for an image. This particular method is preferred for nature scenes, as opposed to object scenes. The results of this study demonstrate that this proposed method effectively removes color casts and brings the image closer to what the human eye perceives than the other tested methods because it is able to overcome the limitations described in the other methods.[12]

In addition to the various approaches to white balance techniques, there are also multiple approaches that involve histogram manipulation. One technique focuses on an idea referred to as "histogram matching."[13] In this method, one image is adjusted previously using tools such as PhotoShop. This image is intended to act as a reference image so that it is the only image that needs to be corrected using these tools, which can often be time-consuming. Once this is done, a mapping relation is developed between a source image and the reference image, which results in the images having a similar color characteristic. This is the point where the source image is said to be corrected. The final results showed that the visual effect of the source images were maintained while any color differences between these images and the reference image were effectively removed. However, the process of developing a reference image can be difficult without experience, which may make this method difficult for some users.

Another approach to histogram matching has involved the inclusion of a color matching card.[14] These cards vary in design, but they contain a wide variety of shades and hues of colors. In one demonstration of this method, a color matching card with a hole is placed over a solid colored card. The card is detected in an output image, whose color is changed via histogram matching using a reference image which contains the same matching card and solid colored card. While comparing the solid colors, it was found that this technique effectively visually replicates the solid color between images, but some shadowing due to lighting in the environment is still present in some corrected images. While this approach does not require the labor of a previously corrected image, as this method allows for any color similarities between the images to be detected, it maintains the issue of requiring a color matching card in both images.

Another method of histogram manipulation stretches the actual cumulative distribution function (CDF) of an image to a target CDF.[15] This method is demonstrated to work well with images taken in the dark because it takes the values skewed toward the lower end of the spectrum and stretches them towards higher intensity values, thus lighting the image. The difficulty with this method is determining what type of distribution to use for the target CDF. For the tested image, it was found that using a logistic target CDF worked better than a Cauchy or linear function. The upper values of the logistic function closer matched the original cdf, so it was mostly the lower intensity values that were corrected. This was the desired effect. However, since each of the other functions had their own benefits in the way they adjusted the image (particularly the Cauchy function), it was concluded that no one method fits all images. Thus, it should be up to the person correcting the image to determine which method works best for the particular image that is being processed.[15]

Histogram manipulation, as mentioned previously, is also used in MATLAB functions. Some of the automatic techniques using histograms in MATLAB were tested against a Multi-scale Retinex with color restoration (MSRCR) method. One of such histogram manipulation techniques is RGB histogram equalization, which uses histogram equalization for each channel in the RGB space.[16] The other method is the RGB to HSI (Hue-Saturation-Intensity model) method. This technique only uses histogram equalization for the H heft, while leaving S and I unchanged. The aforementioned MSRCR approach uses three sub-images that determines the reflection ability of different waves, calculates the relationship between every two pixels (to determine their colors), and linearly maps from Retinex space to RGB space. This method is demonstrated to maintain the color of an image while strengthening the image's details so that it is the most consistent method in replicating human vision, but it comes at the expense of time. In contrast, RGB histogram equalization takes less time and makes the images more colorful, but some details of the image are less clear. The RGB to HSI method remains between these two methods in both time and its ability to replicate human vision. Ultimately, the difference in run time experienced between the RGB histogram equalization and MSRCR methods was, on average, two seconds.[16] However, this is something to consider with slower processing computers or images with a higher pixel count.

Some methods and comparisons emphasize and compare measured illuminates, rather than the colors in images. One suggestion notes that, under colored illumination, one color channel in an image will have a significantly different standard deviation than at least one other.[17] Using this, the illuminate in an image can be estimated, which can be used to correct the image accordingly. This estimated illumination color can be compared against the ground truth illumination color by finding the root mean squared. While this technique succeeds in some images, the presence of noise and other illuminates negatively affect the accuracy of this method based on the assumptions made (such as that with the standard deviation) in this process.[17]

Many of the aforementioned studies of color rely, at least partially, on visual comparisons of colors. However, this method of comparison can be subjective and inconsistent, often taking time to achieve less accuracy.[18] As such, quantitative approaches have been explored to correct this issue. In addition to the methods formally mentioned, colors in images have been compared with methods such as the Euclidean color distance.[18, 19, 20] While there are multiple approaches to using Euclidean color distance, this work will focus on using the RGB color space as the values for this comparison. This method has been proposed for use in medical fields, including the detection of jaundice in babies. In one instance, the bilirubin levels in babies were detected by comparing images of babies' complexion to color calibration cards.[19] The application of Euclidean approximate distance of RGB values in the digital image processing led to a 90% accuracy in detecting jaundice.[19]

Other research involving Euclidean distance and Bayesian methods has been rooted in cloud and sky detection.[20] Again, the application and results of Euclidean distance proved successful with a correlation of 97.9% for clouds and 98.4% for sky.[20] The application of this method proved to not only be comparable to other cloud detection software, but it also contained additional attributes that stem from the assessment of color attributes in the clouds. This data collection and comparisons also extended to the classification of Rayleigh-scattering patterns.

Despite its uses, other research suggests that there are some limitations in the application of Euclidean color distance. One source notes that a large color distance in one of either the red, green, or blue values and a small distance in the others may skew interpretations of results.[21] Such differences in the distances between values will lead the increase in the overall distance calculated using the Euclidean method to be less than the difference one can visually perceive. For example, if the distance between the blue values is 70 while the red and green distances are very small, then the calculated color distance will only be around 70. In some cases, if that distance is determined with an even spread across the red, green, and blue values, this distance may be acceptable. However, with this distance being solely concentrated in the blue region of the RGB color space, the visual difference may be more significant. This is an issue that one needs to consider when making image comparisons that may contain significant color differences.

Issues with accuracy have been demonstrated in some studies that use this method of color comparison. In one case where the color of beef to determine its quality was examined, this method proved to have a 60% success rate.[18] While this is argued to be a potential improvement upon visual comparisons, this rate of accuracy is low with the consideration of the purposes of this study. This is particularly important when considering the fact that, like the beef, the images being compared are textured with color variation. As such, a combination of correction comparisons is used to account for the shortcomings of the techniques used in this study.

The work described in this manuscript follows its own quantitative comparisons with the use of standard deviation and Euclidean distance comparisons, both described in more detail in the following subsection and the Methodology section. Like the previous work, this research also compares multiple images and correction techniques, with an emphasis on the white balance technique, to provide a general understanding of how the proposed method performs overall and in relation to the commonly used technique. The proposed method also takes the importance of the histogram distribution in reducing color cast into consideration, as it relies on the stretching of histogram values to correct the image, which is later described in the Methodology.

Color comparisons in this paper

Using computer software, the exact RGB values present in a pixel of an image can be determined. For this reason, it is possible to quantitatively compare images to determine if they match using some of the aforementioned methods. One method by which one could do this is by taking a mean value coincidence of each RGB value of a particular color region in an image as well as the standard deviation of the RGB values of that area and compare that with the same color region in another image. If the colors of the two images match within an chosen number of standard deviations, then they may be considered a true positive. If they do not match and are not supposed to be the same color, then one would consider that a true negative comparison. Any other result would either result in a false positive, where the colors match and should not, or a false negative, where the colors do not match but should.

The Euclidean distance provides another means of comparison and the idea is to calculate the distance between two different colors. The 3-dimensional Pythagorean theorem is used where the values R, G, and B in place of x, y, and z in the formula. For a source 1, e_1, and a source 2, e_2, this is demonstrated in **Equation 1:**[22]

$$d_{Euc.}(e_1, e_2) = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}. \qquad \textbf{Equation } 1.$$

From there, on can determine if this distance falls within an acceptable range for comparisons. Then, the same true-false positive-negative labeling is used as described previously.

An effective way to determine the precision and accuracy of a color correction method is through the use of a color checker chart. A color checker chart is broken into four rows, each with its own variety of colors in a particular color type: grayscale, primary and secondary, miscellaneous, and natural colors.[24] Each row has six colors, which all contain their own set of known RGB values, as demonstrated in **Figure 1**.

Figure 1. Layout of the Color Checker Chart Used with Known RGB Values[24]**.** This figure provides the color distribution in the color checker chart used for this research. The values within each square correspond to their measured and accepted red, green, and blue values, defined in that order. From top to bottom, the rows consist of natural tones, miscellaneous colors, primary and secondary colors, and grayscale colors. The color checker chart has fewer colors than the color matching card and often is often larger in size.

Not only does the chart provide several colors that can be cross-compared across images under multiple lightnings to determine the precision of a correction technique using the methods described previously, but it also presents the RGB values that should be present for each square in an image of the chart. Thus, color correction techniques can be checked for accuracy by comparing the corrected image values against the known chart values. This combination makes it possible to determine if a technique is correcting images both consistently and accurately, which would also make the pros and cons of different color correction techniques more visible and quantifiable for a better understanding of the effectiveness of each method.

METHODS AND PROCEDURES

Prior to testing each color correction method, a set of eighteen photos were taken to provide variety for comparisons. All of the photos contained the color checker chart, presented in **Figure 2**, but were taken with three different cameras under six different light bulbs. The three cameras consisted of an iPhone, an Android phone, and an iPad camera. The six lighting's included: a overhead light, a Reveal LED bulb at 50 Watts, a Sylvania bulb at 60 Watts, a Zilotek bulb at 100 Watts, a Hungary soft white bulb at 70 Watts, and a Solar Spectra bulb.

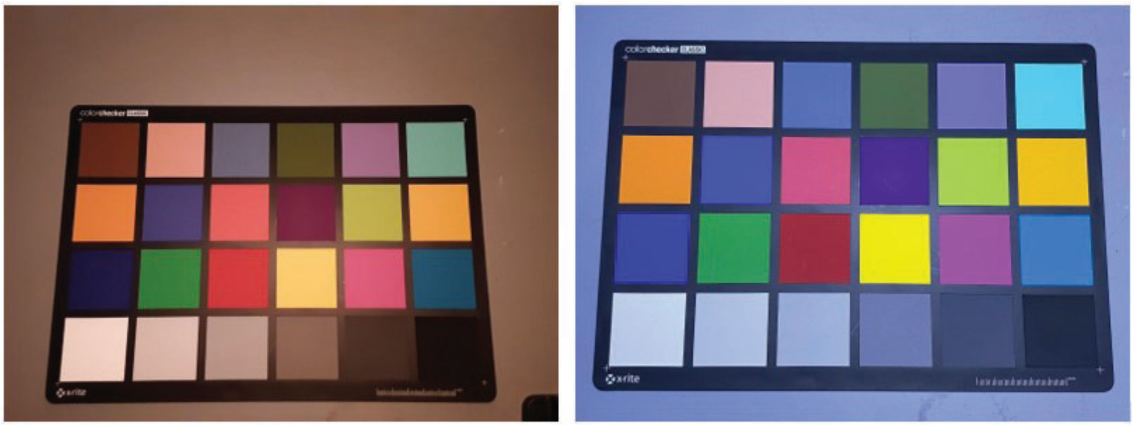

Figure 2. Two Examples of Images Taken and Used for Comparisons. The left image was taken with an Android phone under the Hungary soft white bulb while the right image was taken with an iPhone under the Zilotek bulb. Such variations enabled each of the color corrections to be tested under multiple conditions.

To include a set of real-life photo comparisons, 500 more photos taken in 20 locations with a variety of lightings. Twenty of these photos were taken of a chart created to imitate the color checker chart in its color variety, but each square consisted of a photo of a real-life object. The other 480 photos consist of 20 photos of each individual square from the chart, placed in the scenery without any other squares. These photos were all taken with an iPhone under the natural lighting of the scenery. Examples of each of these are presented in **Figure 3**.

Figure 3. Examples of Photos Taken of the Textured Chart and the Individual Squares. The first photo (left) contains one of the 20 images taken of the textured real-life photos chart. On the right, one of the 20 photos for one of the 24 squares, a zoomed-in image of a fire hydrant, is also shown. These images were used to demonstrate how color correction techniques perform with textured photos containing a single (right) or combination (left) of textured objects.

An additional set of 500 photos also followed the same scheme, but with the addition of a black and white checkered chart or paper with text taking up 20-100% of the background. Half of the colored chart and square photos included the black and white checkered chart and half included the text. Like the previous set, these photos were taken with iPhone camera under the natural lighting of the scenery. Examples of these photos are illustrated in **Figure 4**.

Figure 4. Examples of Textured Photos taken with Black and White Backgrounds. The left image consists of a textured chart photo example taken with a checkered background, while the right one is an individual square photo with a textual background. Both photo sets consist of 50% photos taken with the checkered background and 50% taken with the textual background. In each photo, the percentages of the frame that these black and white backgrounds take up vary.

White balance

The first correction method to be compared is the white balance correction. To achieve the best possible results for this method, a manual white balance method that involved choosing a grayscale was used. For this white balance method, the user is required to select a grayscale portion of the image. This allows the white balance to be automated with fiducial markers, which the code is written to recognize and use to achieve the most accurate results with this correction method.

Using the *drawrectangle* function, an area of the photo can be selected to have the coordinates of that area saved[26]. Then, using the *imcrop* function and these saved coordinates, the image can be cropped down to this selected area.[27] While using the white balance method, an area area of the image is selected and set as the grayscale. Since there are six colors in the grayscale of the color checker, each square is individually selected. Each time the user selects an area, the RGB values for each pixel of the image are saved to be appropriately used.

The average RGB value for each square is found. Then, the average for each red, green, and blue value is found between the six squares. The image is then corrected via white balance using the *chromadapt* function, which takes the average RGB values and the original image as inputs before outputting the white balance image.[28] This image is then displayed.

Since the photos of the individual squares in the first set of 500 photos do not include a grayscale for selection, an automatic white balance function, *lin2rgb*, is used to correct the image instead.[29] The second set of 500 photos allowed for the selection of a grayscale from the text or checkered chart backgrounds. However, this scale could be selected in one area instead of multiple steps. For this reason, the user only selects an area to gather the RGB values from once before using the *chromeadapt* function. In the case of the chart background, four consecutive squares (two white and two black) were selected at once. For the textual background, a single word was selected.

Multi-variable fit method

In order to determine the standard deviation and Euclidean distance that would be used for comparisons in a 3-dimensional distribution, a “best case scenario” color correction technique is necessary for comparison. This correction technique involves a third order multi-variable fit function, which follows the format demonstrated in **Equation 2**:[30]

$$\begin{aligned} R_s = {} & r_0 + r_1 R_e + r_2 B_e + r_3 G_e + r_4 R_e^2 + r_5 G_e^2 + r_6 B_e^2 + r_7 R_e G_e + r_8 R_e B_e + r_9 G_e B_e + r_{10} R_e^2 G_e \\ & + r_{11} R_e^2 B_e + r_{12} G_e^2 R_e + r_{13} G_e^2 B_e + r_{14} B_e^2 R_e + r_{15} B_e^2 G_e + r_{16} R_e^3 + r_{17} G_e^3 + r_{18} B_e^3 + r_{19} R_e G_e B_e. \end{aligned}$$ **Equation 2.**

This is just one of the equations generated for the RGB values for a pixel, specifically the red value. Here, any variable with the subscript e is an RGB value collected from the original image for a particular pixel, as described later in this methodology. Each r represents a coefficient in the expression that is calculated using the multi-variable fit function, also described later in this methodology. While these coefficients remain the same for calculating all red values in a single image, they will vary between each red, green, and blue equations. The value calculated out of this particular expression is the "corrected" red value for a single pixel, which is then used to correct that pixel in the image. Apart from the varying coefficient values, the general format of this equation remains the same for each red, green, and blue calculation.

The accuracy of this method was verified through multivariable goodness of fit plots. This was done for each the red, green, and blue corrections by comparing their adjusted values against the true color values for each square in the color checker chart. An example of these plots for the red values of the color checker squares in an image is provided in **Figure 5**.

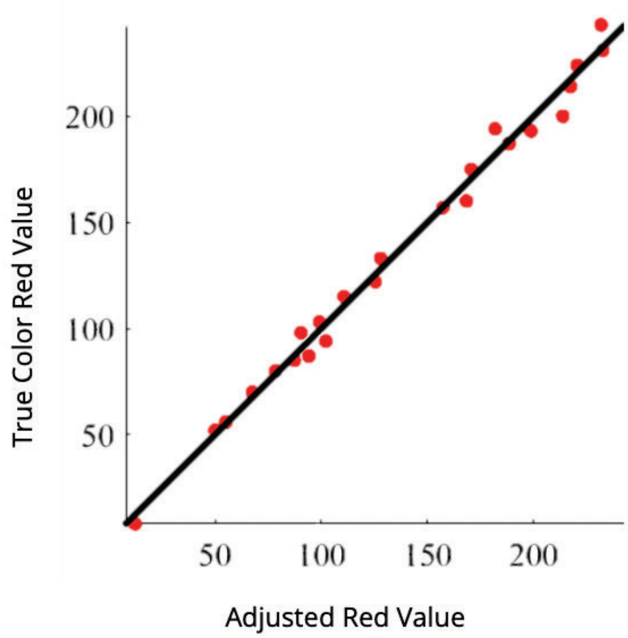

Figure 5. Goodness of Fit Plot for Adjusted Red Values. This figure demonstrates the goodness of fit of the average corrected red values for the pixels of each square in the color checker chart. This information is plotted against the true chart red values, illustrating the close fit of the corrected values to the actual values. Similar plots were developed for both the green and blue values in this and other corrected images.

In order to create this function, both the photographic and actual RGB values for each color checker square need to be known. The actual values are provided by literature, which is demonstrated back in **Figure 1**. The RGB values for each square in the photo, in contrast, needed to be collected from cropping and saving the RGB values for each square. An example of a cropped square and the chart that it belongs to, for comparison, was demonstrated in **Figure 3**.

This data is then used in an outside function, *MultiPolyRegress*.[30] Here, the polynomial fit for the red values is found by taking the first column of the experimental values array, the true red values, and and telling the function to make a third order polynomial fit. This is then repeated for both the green and blue polynomial fits with their respective experimentally recorded RGB values.

These polynomial fit functions are then used to correct the RGB values for each pixel. From there, the values for respective pixels are assigned to their appropriate location to make up a new, corrected image. This is then displayed for visual and data-based comparisons, which RGB values are collected for. This function could not be used for either of the sets of 500 photos because none of the true RGB values in those photos were known.

Upper and lower percentage histogram manipulation

Our chosen color correction technique is a upper and lower percentage histogram manipulation method. We will test this technique against the white balance methods.

This method focuses on saturating each the red, green, and blue value histograms by a given percentage. In order to do so, the method finds the first occurrences of pixel intensities in the histogram and continues to adjust those pixel intensities to a value of zero until the desired percentage of pixels in the image are corrected to this point. The same is done starting from the other side of the histogram (with the maximum values), but these values are corrected to the maximum intensity value of 255. This is done for each the red, green, and blue value histograms. The remaining histogram values are then used to fill in the intensity values between these extremities. This new distribution is then used to correct and update the image. Examples of the original and adjusted histograms is provided in **Figure 6**. Again, once the image is corrected, the RGB values of each square are saved for quantitative comparisons.

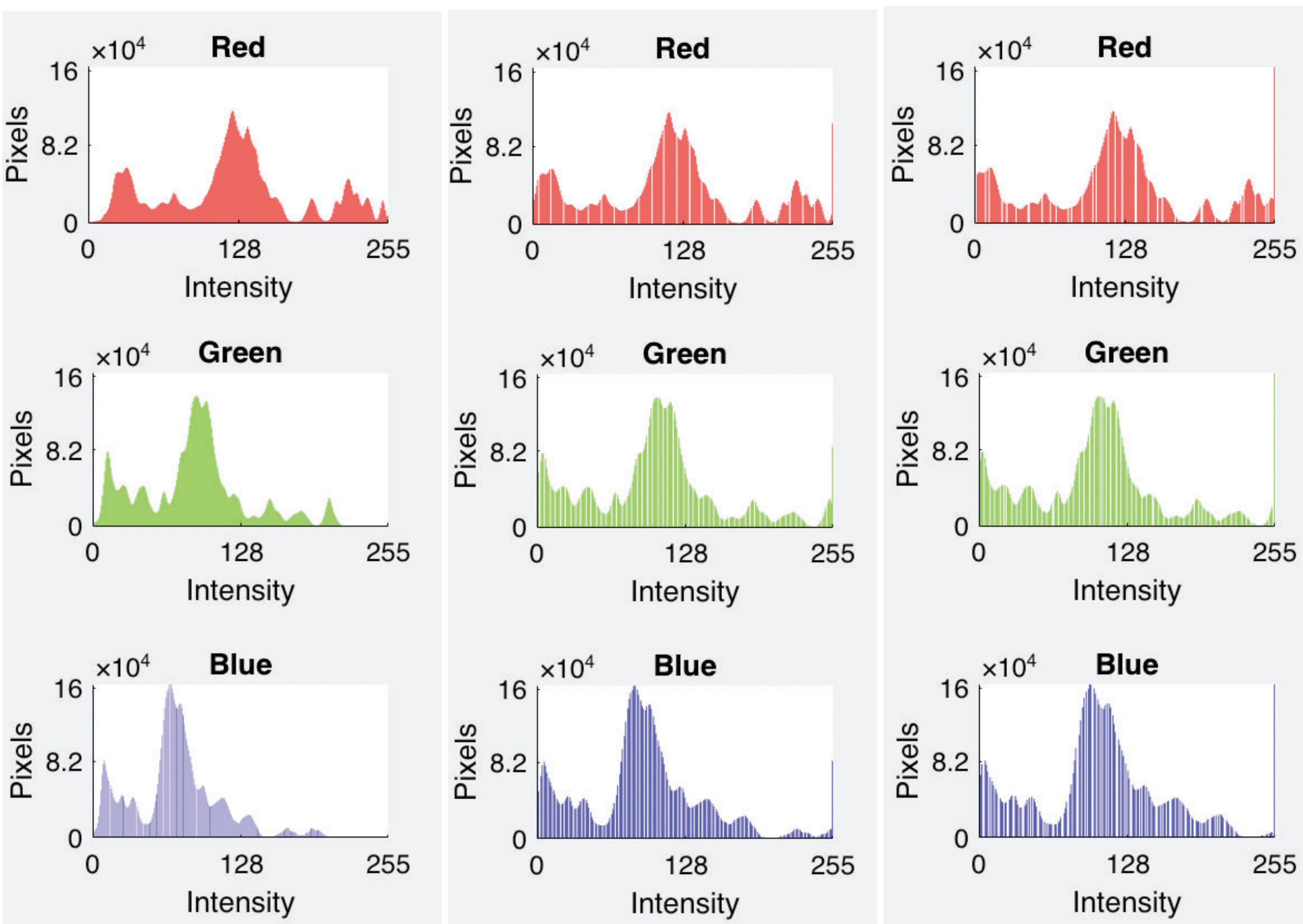

Figure 6. Histograms of RGB Values for Various Saturated Images. These RGB histograms are collected from the original (left), 1% saturated (middle), and 2% saturated (right) image of the color correction chart, provided in Fig. 2 (left). Note how increasing the saturation pulls more pixel values to the extremities of the individual RGB histograms while the remaining pixel values fill in the space in-between.

Comparing images

A visual comparison of an image of the color checker chart taken with an Android phone under the Hungary soft white bulb after using the various color corrections is provided in **Figure 7**.

In order to accurately and quantitatively compare these images, the RGB values from each of them needed to be recorded for comparison. This was done by "sampling" each square in an image. This code follows the same format as the cropping code that was used for the multivariable fit.

Figure 7. Corrections of an Image from Fig. 2 (left) Using Various Correction Techniques. This figure consists of the original (top left), multi-variable fit (top middle), manual white balance (top right), automatic white balance (bottom left), and 1% saturation (bottm right) methods. Note that, due to its use of the actual chart RGB values, the multi-variable fit image is the most accurate and can, therefore, be used when visually comparing the images for accuracy.

Once the RGB values for each correction method and the original images are collected, the comparisons of the values can begin. A separate code is written for the cross-comparisons. The average values are sorted by correction type and into RGB components so they can be appropriately compared. This is also done for the standard deviation values, σ, which act as one of the primary metrics for comparison. The equation for σ is provided in **Equation 3**:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}. \quad \textbf{Equation 3.}$$

The standard deviation for red, green, and blue is automatically calculated by MATLAB. Here, x_i is one of the RGB values corresponding to a singular pixel in the image. If the standard deviation is for red, then the red value for this pixel will be used. The same can be said for the green and blue standard deviations. The variable μ is the average RGB value, again correlating to the color that the standard deviation is being calculated for. This difference is calculated, squared, and summed with other squared differences for all of the pixels in the selected image area. This sum is divided by N, which corresponds to the number of pixels in the cropped image, or sampled square. The square root of this quotient provides the standard deviation, σ, as seen in **Equation 3**.

All other squares from images with the same color correction are compared against one another. If the RGB values of one square fall into the range of maximum and minimum RGB values for another set by the standard deviations (with red being compared against red, green against green, and blue against blue), then the squares are considered a match. If one or more of the RGB values of the first do not fall into the respective ranges of the second, then the squares do not match. The next step is to then check if the colors should match. This is done by comparing the assigned specific color that is saved for the square that the data is taken from. If they are the same, then the colors should match and this indicates a false negative. If not, then the colors should not match and are a true negative. If the RGB values of the first square fall into the range for the second, then a matching specific color comparison would come out to be a true positive. If the values match but the specific color does not, then the result is a false positive.

This process is repeated until all squares within a correction type have been compared with one another. Again, this overall cycle of comparisons is completed for within each correction type. At the end, the percentage of true positive is calculated for each type by dividing the true positive count by the sum of the true positive and false negative counts. The percentage of true negative is then calculated by dividing the true negative count by the sum of the true negative and false positive. Overall, all of these comparisons and data saving were repeated for various standard deviations, or sigma.

The true value comparisons, calculated separately, follows a similar format as the cross comparisons. The data is imported and saved into arrays in the same manner as in the cross-comparison code. The code is also ran for the same range of sigma. However, instead of comparing data from squares against other squares with the same color correction type, each square is only compared with the actual RGB chart values. In addition, the true value comparisons could not be used for either set of 500 photos since their true values are unknown.

Another code follows the same format as the standard deviation comparison code but is written for the Euclidean color difference comparison described earlier in the Methods and Procedures section. Instead of seeing if one data set for an image square falls within a specific number of standard deviations of another, the code calculates the Euclidean difference of the squares' colors using **Equation 1**. After checking if the calculated difference falls within a prescribed maximum difference, the remaining code follows that of the standard deviation comparisons, determining if the results of the squares should and do match to calculate the true/false positive/negative rates. This is also initially tested for multiple Euclidean distance values.

RESULTS

To determine the number of standard deviations (σ) that will be used for the comparisons of each set of photos, the results at various sigma from the gold standard, or multi-variable fit, method must first be considered. Since the gold standard data represents the "best case scenario," the opposite end of the spectrum, the original photo data, must also be considered when choosing an appropriate number of σ. This data helps gauge at what point too many standard deviations are being used because this would result in unreasonably high percentages for the original photos, particularly in the true positive percentages. A summary of the cross-comparisons of original photos and photos corrected using the multi-variable fit at various sigma is provided in **Table 1**.

Table 1. Standard Deviation True Positive and True Negative Results for the Original Photos and the Gold Standard Multivariable Color Correction of the Pure Color Chart

	Original Images				Gold Standard			
	True Positive		True Negative		True Positive		True Negative	
σ	Cross	True Value	Cross	True Value	Cross	True Value	Cross	True Value
2	2.69	0.35	99.98	91.61	81.60	85.76	100.00	91.67
3	6.41	3.82	99.93	91.58	91.54	93.40	99.99	91.65
4	11.01	6.94	99.84	91.47	95.61	97.22	99.86	91.64
5	16.32	14.24	99.67	91.35	97.41	97.92	99.49	91.34
6	22.76	20.14	99.45	91.06	98.52	99.31	98.83	90.70
7	29.26	26.04	99.12	90.79	99.12	99.65	98.00	89.99
8	35.73	33.33	98.67	90.41	99.50	99.65	96.86	88.98

Considering the results found for the gold standard color correction in **Table 1**, it would appear that around 4σ-5σ is necessary to achieve the 95% accuracy desired, which equates to the accuracy of 2σ in a 1-dimensional normal distribution. This also follows with the results for the original photos in **Table 1**, where these numbers of σ correlate to data prior to when the original data results begins to rapidly improve. However, assuming that the RGB values are all independent, one could also argue that 3σ, alone, should encompass $99.7\%^3 = 99.1\%$ of the data. Therefore, going beyond this number of standard deviations would be unreasonable, even if there is a small dependence on one another between the RGB values. This also agrees with the overall trend of both data sets, as the average of the true positive and true negative cross-comparison results for the gold standard at 3σ remains around 95% while the gold standard's true color average is about 92.5%. This is also reasonable for the original data set because this number of σ still falls before the major improvements in results. Thus, three was chosen for the number of standard deviations to be used in comparisons of any following data.

While using standard deviation comparisons is a valid method to compare data, an issue arises when using this method alone for the textured photos. The data provided in **Table 1** result from pure color chart photos, which do not have large standard deviations. However, the added texture in the real-life photos, as demonstrated in **Figure 3**, naturally increases the standard deviation of the RGB values within a photo due to variations in color across each square. With this in mind, the Euclidean color distance comparison, described in the Methods and Procedures section, was also used as a comparison metric using the average RGB values collected for each photo and correction type. **Table 2** provides the results of the Euclidean distance comparison for the original photos and gold standard method.

Table 2. Euclidean Color Distance True Positive and Negative Results for the Original Photos and the Gold Standard Multivariable Color Correction of the Pure Color Chart

	Original Images				Gold Standard			
	True Positive		True Negative		True Positive		True Negative	
$d_{Euc.}$	Cross	True Value	Cross	True Value	Cross	True Value	Cross	True Value
20	17.35	12.73	99.77	99.68	96.57	96.99	99.99	100.00
30	33.88	29.63	99.15	99.09	99.48	99.31	99.99	99.69
40	48.39	52.08	97.79	97.80	99.97	100.00	97.48	97.56
50	59.42	67.36	95.81	95.79	99.97	100.00	95.15	95.12
60	68.66	78.01	93.23	93.03	100.00	100.00	92.43	92.63
70	76.44	86.57	89.93	89.53	100.00	100.00	86.67	86.81
80	83.17	90.74	86.03	85.37	100.00	100.00	81.06	81.11

As seen in **Table 2**, the gold standard method has the highest average true positive and true negative rate for both the cross-comparisons and true color comparisons when the Euclidean color distance is 30. This color distance is also the point where the original photos true positive rates are consistent between the cross-comparison and true color comparison results. The same can be said about the true negative results across both comparison types. For this reason, a Euclidean color distance of 30 was initially considered for the following corrected photo comparisons.

To get a sense of how each method would perform with real-life, or textured, images, the second set of photos mentioned in the methodology was taken to make such comparisons. However, since the actual average RGB values for each square in the images are unknown, both the gold standard method and true value comparisons are not used. In addition, only the 3σ determined using white balance and the original photos in the previous data set was used for this new set of standard deviation comparisons. Furthermore, an automatic white balance was added to comparisons since a grayscale is not initially provided for the manual white balance in photos for individual squares. A range of 0.0%-3.0% saturation via histogram manipulation is provided in these comparisons to consider how their performances vary with real-life photos and determine the best performance.

The cross-comparison results for the real-life photos with and without a black and white background at 3 sigma are provided in **Table 3**. Looking at the squares without the black an white background, alone, the only white balance that can be considered is the automatic white balance. This method performs a minimum of 5.5% worse than any of the other methods, including the original photo, in the true positive comparisons. In contrast, its true negative percentage is greater than any of the histogram manipulation percentages and is second only to the original image. While this does appear to reflect poorly on the histogram manipulation technique, it is important to remember that the textured squares have higher standard deviations than solid colored squares. As a result, there is more likely to be a significant false positive count, resulting in a lower true negative rate. This is one of the justifications for later using the Euclidean distance comparison technique.

Looking at the overall trends between in these comparisons, 0.5% maintained one of the highest true positive rates, while 1.5% retained one of the highest true negative rates of the tested percentages. This factored into the choice to make 1.0% histogram manipulation the main focus when comparing against other correction methods while including the black and white background, as it is the middle-ground of these two percentages. However, since this choice is based on general trends, the Euclidean distance comparisons was also considered before making this decision. As explained later in this section, the Euclidean comparisons also affirmed the 1% saturation choice.

Despite the apparent shortcomings of white balance without the black and white background, it is important to note that this is an incomplete comparison because manual white balance could not be used in the individual square photos without a grayscale present. Taking this issue into consideration, along with the purposes of applying these methods to the image analysis of birds, the final set of photos mentioned in the Methods and Procedures were taken with black and white backgrounds for comparison. This is also to the benefit of the histogram manipulation method, which works on the assumption that there are pixels to the extremities of the RGB value range. Since black and white lie at these points in the RGB value range, this would provide a metric in the image to aid the histogram manipulation method in its color correction. This data for 3σ is also included in **Table 3**.

Table 3. Summary Table of Standard Deviation Cross-Comparisons of Real-Life Square Photos at 3 Sigma

	True Positive		True Negative	
Correction Type	**W/O Black and White**	**W/ Black and White**	**W/O Black and White**	**W/ Black and White**
Original	59.01	84.98	68.13	64.20
Auto WB	53.43	82.93	68.01	70.92
Manual WB	N/A	88.67	N/A	63.46
0.0% Saturation	61.13	86.78	66.71	61.82
0.5% Saturation	65.06	94.42	60.27	61.37
1.0% Saturation	64.11	93.08	60.36	62.95
1.5% Saturation	63.04	92.88	60.61	62.61
2.0% Saturation	62.14	92.41	61.11	62.35
2.5% Saturation	61.36	92.19	61.64	62.57
3.0% Saturation	60.43	91.78	62.05	62.61

The set of photos with black and white backgrounds provides a more complete understanding of how the white balance methods perform in comparison to the original and saturated photos. With the black and white background, the automatic white balance performed about 6% worse than the manual white balance in the true positive rates and 7% better in the true negative rate comparisons. This method also performs worse than the original photos in both rates for the square photos, with the exception of the true negative with a black and white background. The manual white balance, on the other hand, generally performs better in comparisons with the true positive percentages and worse with the true negative rate.

Across the table, 1.0% saturation generally performs among the best of the methods presented. While its true negative results are up to 7% lower than the automatic white balance photos, which did either the best or the second best in both the true negative comparisons, the saturation via histogram manipulation method consistently maintained at least a 4.3% gain on the other non-saturation methods in the true positive rates. The low true negative rates are particularly understandable when considering the large standard deviations with textured photos, which would lead to more false positives. While 0.5% saturation does maintain a higher true positive rate in these photos than the other saturations, it remains 1.5% from the 1% saturation rate. This is a similar difference to that between the two for the true negative rate, where 1% saturation has the highest value. This, combined with the trends seen without the black and white background, adds to the justification for using 1% saturation.

It is also important to note that manual white balance does not perform the best in a single category, despite it never having the worst rate at any point. Regarding the standard deviation comparisons alone, 1.0% saturation is notably the best correction method of those compared. One may notice that in addition to the improvement in white balance and 1.0% saturation methods, the original photos also improve from the previous set. This is because this set of photos was taken separately from the previous, which resulted in variations in lighting and other conditions between the sets of photos. This, however, is not the sole cause of improvements in any of the color correction methods. While the original photos improved by a rate up to about 15%, the saturation and automatic white balance methods improved on a scale of about 20% - 30%. In addition, the original photos worsened in true negative rates, while the correction methods, with the exception of the 0.0% saturation, improved in those rates. A similar trend can be seen with the Euclidean distance comparisons.

Although the results of **Table 2** initially suggested that a Euclidean distance of 30 should be used for other comparisons, low rates in the positive and negative comparisons at this distance for textured photos suggested that this choice was flawed. Like with the standard deviation choice, the Euclidean distance for these comparisons must also take into account the large standard deviations in the real-life photos. For this reason, a larger Euclidean distance should be considered to provide a more fair comparison of the correction techniques. A Euclidean distance of 60 was chosen for the real-life comparisons because this is the point where the average of the true positive and true negative rates were at the highest for each correction type. The results from the Euclidean distance comparisons at a distance of 60 are provided in **Table 4**.

Again, the best histogram manipulation results remains in the 0.5%-1.5% range. The 0.5% saturation results have one of the highest true positive rates in both comparisons of the saturation method, remaining behind only to automatic white balance and 0.0% saturation in the square photos without a black and white background. The results of the higher percent histogram manipulations are higher in the true negative comparisons, but the true negative rate of the 1.5% saturation is within 0.5% of that of the 3.0% saturation.

While 0.5% saturation performed well in the Euclidean difference calculations, 1.0% saturation was still chosen for the final comparison set. This is because the results of the standard deviation comparisons, as explained before, suggest that 1.0% saturation would be a good compromise between 0.5% and 1.5% saturation. In addition, as seen in **Table 4** for the data without the black and white background, 1.0% saturation performs as consistently, although at lower true positive rates, as 0.5% saturation. Therefore, 1.0% remains a reasonable choice for the histogram manipulation method.

Table 4. Summary Table of Real-Life Square Cross-Comparisons with a Euclidean Distance of 60

	True Positive		True Negative	
Correction Type	**W/O Black and White**	**W/ Black and White**	**W/O Black and White**	**W/ Black and White**
Original	42.54	72.81	85.50	82.99
Auto WB	37.87	76.27	84.26	85.61
Manual WB	N/A	75.20	N/A	82.29
0.0% Saturation	43.30	70.30	85.24	82.96
0.5% Saturation	37.36	81.82	86.78	84.89
1.0% Saturation	35.02	80.44	87.27	86.07
1.5% Saturation	33.64	78.68	87.58	86.71
2.0% Saturation	32.30	76.69	87.66	87.09
2.5% Saturation	31.58	75.53	87.76	87.66
3.0% Saturation	30.64	74.12	87.70	88.02

With Euclidean distance, the 1.0% saturation method also maintains the best results across the comparisons with the black and white background. However, unlike in the standard deviation comparisons, 1.0% saturation performs the best across all comparisons and percentage rates, with the exception of other saturation methods which differ in success rates between each percentage type. While the manual white balance has higher true positive rates than the original photos by at least 2.0%, it also consists of the lowest true negative rates of the three methods. The difference between the original and manual white balance true negative rates is less than 1.5% across both comparisons, but the results demonstrate that the 1.0% saturation method is the most effective and consistent of the three methods in this photo set. Automatic white balance does perform better than either of these correction types but maintains lower values than the 1% saturation.

While the code used to produce the images from which **Tables 3** and **4** were the same throughout, the addition of the black and white background made a significant difference in the results. This particular difference is dependent on the processes behind the correction techniques. As described previously, in the histogram manipulation technique, a given percentage of the first and last incidences of an RGB value is assumed to be 0 and 255, respectively. When black and white are not present in the image, this may lead to over-saturating the image, leading to the poorer results seen in **Table 3** and **4**. Conversely, when black and white are present, the portions of the image containing these colors should be adjusted to these extremes. Although not all whites and blacks are perfectly (0, 0, 0) and (255, 255, 255) in the RGB color space, the presence of these colors provides a more accurate metric that the correction method may adjust the image in relation to.

As mentioned in the analysis of the individual square photos in **Tables 3** and **4**, manual white balance is also dependent on a grayscale being present to accurately adjust the image. The lack of success in the manual white balance results demonstrates that the manual white balance correction is considerably less viable and precise than the histogram manipulation method, even when attempts to make this option more applicable is made. Again, while these changes do improve the automatic white balance method, the difference is not enough to make this method more consistent than the saturation method.

Although both methods of statistical comparison reach the same conclusion, it is worth noting that the standard deviation method does so biasing towards true positive rates while the Euclidean distance method biases towards true negative rates. This is likely a result of the difference in how average and standard deviation values of the RGB values are accounted for in each comparison type. The standard deviation method includes a wide band, as a result of the large standard deviations in textured photos, to compare the RGB values between two squares. As a result of this, there is going to be a higher rate of positive results when using this comparison. This means that both the true positive and false positive rates are likely to increase, resulting in a higher true positive rate and a lower true negative rate. On the other hand, while there is a range determined for the Euclidean distance method, this range it not dominated by the high standard deviation values of the textured photos. As a result, in comparison to the standard deviation method, there is more likely to be a decrease in positive rates for the Euclidean distance approach. This, as a result, decreases the true positive rates while increasing the true negative rates.

CONCLUSIONS

Our motivation for testing color techniques is to find a technique or multiple techniques that could be used in SIA[1] to aid the eye by increasing the contrast, and to consistently adjust colors so that color comparisons can be made independent of lighting. Our preferred technique involving color saturation was used not only for these purposes but it was also used to increase the contrast of frames in the standalone application BatCount[2] thereby aiding foreground object detection in videos.

Although the results between data sets vary, the final set that was taken using photos based on the purposes of these color correction comparisons in SIA demonstrated that 1.0% saturation via histogram manipulation performed the best of the presented methods, excluding other saturations. The gains in the percent true positive for 1.0% saturation were more significant than the losses in its percent true negative when comparing it to white balance and the original photos in the standard deviation comparisons. With the Euclidean difference comparisons in the final data set, however, a saturation value of 1.0% consistently performs among the best across these comparisons.

It is important to note, however, that the success of the 1.0% saturation is partially dependent on its intended use. **Tables 3** and **4** illustrate that, when a chart is present without a black and white background, manual white balance does significantly better in the true positive comparisons and within 1.0% of the true negative results of 1.0% saturation. Therefore, the saturation method is not always the “best” method for color correction — it simply best serves the purposes of color correction in SIA. Additionally, please note that our comparison of lighting sources was intentionally treated in a statistical manner. Any subsequent analysis of color comparisons in different lighting using the same techniques and metrics will presumably reach similar conclusions but will likely not find our identical findings.

Despite its shortcomings, the histogram manipulation method does maintain a few advantages over other methods. The multi-variable fit function is impractical in its application to everyday photos due to its reliance on known RGB values in an image. One may consider taking an image with a color checker chart to achieve this level of accuracy, but this makes the process more tedious and reduces one’s ability to take action shots. In a similar manner, the difficulties illustrated in the second set of photos points to the issue of how limited the manual white balance method is. Again, one could take photos with a color checker chart present, but this it impractical and tedious, as explained with the multi-variable fit function. The automatic white balance method is applicable, but results in the second data set also demonstrates that it is less consistent than other methods, including both the 1.0% saturation method and, in some instances, the original image. Additionally, the 1% saturation works best if there is some black and white in the background. Without this colors can saturate in an unnatural way as discussed in[1]. But unlike with white balance a gray scale region doesn’t have to be identified.

In the future, a wider data set may also be considered. In the first set with the pure color charts, a variety of cameras were used to test the overall and more universal performance of correction techniques. In the photos for the following sets, however, all of the photos were taken with the same iPhone camera. Since cameras vary in quality across different devices, including various cell phones, it would be worth comparing how these corrections compare across different cameras — both in general and for specific camera types.

Furthermore, comparisons between the proposed technique and methods described from previous literature may also be compared. Since the best performance of the upper and lower percentage histogram manipulation method occurred when black and white aspects were added to the background, adding a color matching card into images for histogram manipulation using this card would be a reasonable change, especially since it is smaller than a color checker chart. However, the difference in run time in using this method may outweigh any improvements in color correction from the proposed method. Other approaches may include using a neural network. While this may also seem like a favorable option with color correction, this approach is not as frequently used because it takes more time to train. Thus, it is also more expensive. There any many other techniques that may be worth testing in the future, but careful considerations must be made before doing so. Each method has its own advantages and disadvantages and it is important to focus on and emphasize ideas that would best work for the purposes of SIA.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation Infrastructure Innovation for Biological Research: Intuitive Sampling, Tracking and Counting Software for Biological Sciences (Award Number 1916850).

REFERENCES

1. Bentley, I., Ralston, J., Garman, S., Hershberger, O., Probst, C.M. and Washer, C. (2023) Introducing the Scientific Image Analysis Application: A Free and User-Friendly Program for Extracting Bioinformatics From Digital Images. IEEE 2023 CCWC Conference Proceedings, *https://ieeexplore.ieee.org/document/10099135*.
2. Bentley, I., Kuczynska, V., Eddington, V., Armstrong, M., and Kloepper, L.(2023) Bat-Count A software program to count moving animals. PLoS ONE 18(3) e0278012,

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0278012.

3. Bibikov, S.A., Zakharov, R. K., Nikonorov, A. V., Fursov, V. A., and Yakimov, P. Y. (2011) Detection and color correction of artifacts in digital images, *Optoelectronics, Instrumentation, and Data Processing*, 47(3), 226-232, *https://doi.org/10.3103/S8756699011030046*.
4. Noordin, M. J., Mat Isa, N. A., and Lim, W. H. (2016) Saturation avoidance color correction for digital color images, *Multimedia Tools and Applications*, 76(7), 10279-10312, *https://doi.org/10.1007/s11042-016-3620-y*.
5. Gasparini, and Schettini, R. (2003) Color correction for digital photographs, *12th International Conference on Image Analysis and Processing, 2003.Proceedings*, 646-651. *https://doi.org/10.1109/ICIAP.2003.1234123*.
6. Kwok, N.M, Shi, H. Y., Ha, Q. P., Fang, G., Chen, S. Y., and Jia, X. (2013) Simultaneous image color correction and enhancement using particle swarm optimization, *Engineering Applications of Artificial Intelligence*, 26(10), 2356-2371, *https://doi.org/10.1016/j.engappai.2013.07.023*.
7. H. Niu, Lu, Q., and Wang, C. (2018) Color Correction Based on Histogram Matching and Polynomial Regression for Image Stitching, *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 257-261. *https://doi.org/10.1109/ICIVC.2018.8492895*.
8. Bilcu, R. C. (2011) Multiframe Auto White Balance, *IEEE Signal Processing Letters*, 18(3), 165-168. *https://doi.org/10.1109/LSP.2011.2105476*.
9. Feng, L., Jian, S., and Yongzhao, Z. (2004) A fast color correction method based on image analysis, *Third International Conference on Image and Graphics (ICIG'04)*, 108-111. *https://doi.org/10.1109/ICIG.2004.3*.
10. Sengupta, D. Biswas, A., and Gupta, P. (2020). Non-linear weight adjustment in adaptive gamma correction for image contrast enhancement, *Multimedia Tools and Applications*, 80(3), 3835-3862. *https://doi.org/10.1007/s11042-020-09583-1*.
11. Mathworks (2021) Contrast Enhancement Techniques, *MathWorks Help Center*, *https://www.mathworks.com/help/images/contrast-enhancement-techniques.html*.
12. Hu, Y. (2012) Automatic White Balance Based on Gassian Decomposition, *Applied Mechanics and Materials, 263-266*, 2542, *doi:https://doi.org/10.4028/www.scientific.net/AMM.263-266.2542*.
13. Zhang, J., and Chen, G. X. (2013) Research on the color correction algorithm of images based on histogram matching, *Applied Mechanics and Materials, 469*, 256, *doi:https://doi.org/10.4028/www.scientific.net/AMM.469.256*.
14. Rosebrock, A. (2021) Automatic color correction with OpenCV and Python, *pyimagesearch*, *https://pyimagesearch.com/2021/02/15/automatic-color-correction-with-opencv-and-python/*.
15. Manansala, J. (2021) Image Processing with Python: Color Correction using Histogram Manipulation, *The Startup*, *https://medium.com/swlh/image-processing-with-python-histogram-manipulation-on-digital-images-d4fb426d3513*.
16. Xie, X. M., Wang, C. M., and Zhang, A. J. (2013) Color image enhancement methods based on matlab, *Applied Mechanics and Materials, 300-301*, 1664, *doi:https://doi.org/10.4028/www.scientific.net/AMM.300-301.1664*.
17. Choudhury, A. and Medioni, G. (2010) Color Constancy Using Standard Deviation of Color Channels, *2010 20th International Conference on Pattern Recognition*, 1722-1726, *https://doi.org/10.1109/ICPR.2010.426*.
18. Handayani, H. H., and Wahiddin, D. (2018) Digital Image Analysis of Beef Color Using Euclidean Distance Method, *2018 Third International Conference on Informatics and Computing (ICIC)*, 1âĂŞ5, *https://doi.org/10.1109/IAC.2018.8780457*.
19. Rahayu, Widyawati, M. N., and Suryono, S. (2021) Euclidean distance digital image processing for jaundice detect, *IOP Conference Series. Materials Science and Engineering, 1108*(1), 12022, *https://doi.org/10.1088/1757-899X/1108/1/012022*.
20. Neto, M., Wangenheim, A. von, Pereira, E. B., and Comunello, E. (2010) The Use of Euclidean Geometric Distance on RGB Color Space for the Classification of Sky and Cloud Patterns, *Journal of Atmospheric and Oceanic Technology, 27*(9), 1504-1517, *https://doi.org/10.1175/2010JTECHA1353.1*.
21. Abasi, Amani Tehran, M., and Fairchild, M. D. (2020) Distance metrics for very large color differences, *Color Research and Application, 45*(2), 208-223, *https://doi.org/10.1002/col.22451*.

22. Gijsenij, Gevers, T., and Lucassen, M. (2009) Perceptual analysis of distance measures for color constancy algorithms, *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 26(10), 2243-2256, *https://doi.org/10.1364/JOSAA.26.002243*.

23. Nikon, DSLLR Camera Basics: White Balance, *Nikon: DSLR Camera Basics*, *https://imaging.nikon.com/lineup/dslr/basics/17/index.htm*.

24. Image Analyst (2012), Color Checker Chart, *MathWorks File Exchange*, *https://www.mathworks.com/matlabcentral/fileexchange/38236-color-checker-chart*.

25. The MathWorks Inc., MATLAB and Image Toolbox Release 2021a, (accessed Nov 2021).

26. MathWorks (2021) Create Customizable Rectangular ROI, *MathWorks Help Center*, *https://www.mathworks.com/help/images/ref/drawrectangle.html*.

27. MathWorks (2021) Crop Image, *MathWorks Help Center*, *https://www.mathworks.com/help/images/ref/imcrop.html*.

28. Mahmoud Afifi, chromadapt: Adjust color balance of RGB image with chromatic, *MATLAB Central File Exchange*, *https://www.mathworks.com/matlabcentral/fileexchange/66682-chromadapt-adjust-color-balance-of-rgb-image-with-chromatic* (accessed Jul 2021).

29. Mathworks (2021), Comparison of Auto White Balance Algorithms, *MathWorks Help Center*, *https://www.mathworks.com/help/images/comparison-of-auto-white-balance-algorithms.html*.

30. C. Ahmet, Multivariable Polynomial Regression, *Github*, *https://www.mathworks.com/matlabcentral/fileexchange/34918-multivariate-polynomial-regression?s_tid=FX_rc3_behav* (accessed Jul 2021).

31. MathWorks (2021), Adjust Image Intensity Values or Colormap, *Mathworks Help Center*, *https://www.mathworks.com/help/images/ref/imadjust.html*.

ABOUT THE STUDENT AUTHORS

Kyra Obert recently graduated with a B.S. in Physics and a minor in Mathematics in May 2022.

Maria Schudt will graduate with a B.S. in Physics in May 2023. Following this, she will transfer to the University of Notre Dame as part of a 4+1 program to receive a B.S.E. in Mechanical Engineering in May 2024.

PRESS SUMMARY

Color correction is used to adjust digital photographs in order to reduce the effect of environmental factors, such as lighting. This manuscript analyzes the precision and accuracy of upper and lower percentage histogram manipulation, which involves saturating an image by a specified percentage of its pixels using its RGB histogram. It is compared against the commonly used white balance method, corrections using generated multi-variable equations for RGB values, and original images to determine when improvements or deteriorations in corrections are made in using this proposed method. The findings demonstrate that the histogram manipulation method is effective in comparison to these other methods because it does not require sampling of calibration regions and it is the most consistent method in the correction of photos with significant gray scale regions.