Invisible Display: Reverting Color Distortion in a Camera-Display System Using Color Mapping and Barycentric Interpolation

Margaret Cress Connell, Abdel-Rahman Amr Madkour, John Douglas Stone

Abstract

Recent advances in computer vision, specifically virtual reality, have given rise to the need for accurately representing real world colors using digital displays. The difficulty of this endeavor arises from the problem of digital color distortion. Our research specifically addresses such color transformation in a camera-display system, in which both the camera and display distort the real life scene's color. In this paper, we propose a solution for how we can best reverse this distortion. Our method utilizes color mapping and barycentric interpolation to produce restored images within 5% average difference of real world colors. Possible applications for this method include teleprecence, virtual reality and three dimensional modeling using two dimensional images.

1 Introduction

Due to modern advances in virtual reality the need to accurately reproduce realistic colors in an image arises. We seek to examine this issue in the context of a camera-display system, in which both the camera and display distort the image's color values. Our research addresses how we can best reverse that distortion. Our proposed solution to this problem involves gathering mapping data on a subset of the color values, then applying barycentric coordinates to interpolate the remaining values of this preliminary data, and finally encoding this data into a three dimensional color cube. We use this color cube to create an ideal image whose colors match colors that exist in real life.

2 Background

Grossberg, Peri, Nayar, and Belhumeur [1] present research on how to alter the appearance of an object using a projector-camera system. They use a camera to capture the colors and textures present in the scene and create compensation images. They overlay these compensation images onto the object using the projector to obtain the desired appearance. Similarly, we also examine colors in a camera-display system. However, we hope to study color distortion and revert it, rather than attempting to alter the appearance of a three dimensional object.

Hincapie-Ramos, Ivanchuk,Sridharan and Irani's work [2] aims at adjusting color blending in optical see-through displays. The perceived color in an optical see-through display is a mix of the color of the display and the color of the background. The color correction that this work proposes aims to mitigate this mixing effect by having the display produce a color, that when mixed with the background color, produces the desired or intended color. This work relates to our project as it attempts to find a mapping between the colors of the display and the real life scene behind the display. While this work focuses on color blending in an optical see-through display, we seek to address color distortion in a camera-display system.

We were unable to find work that specifically addresses color distortion in a camera-display system in order to imitate real world colors.

3 Method

3.1 Camera Calibration

To properly analyze color distortion and accurately correct it we need to precisely map the camera's pixels to the corresponding pixels of the display. To do so, we utilize work conducted by a previous team of St. Olaf students: Andrew Crocker, Austin Martin, Jon Sandness. This work maps the pixels in an image captured by a camera belonging to the display to the pixels corresponding to that location in the display. This work uses a program that employs a recursive striping method to map the pixels. Their method involves taking images of pure black and pure white stripes on the display which decrease in size by half in each image until the stripes are a single pixel wide. The camera must be put in manual mode, with any automated focus or lighting correction settings turned off. For our research we use a Canon Rebel T5i camera. We process these images to produce a mapping text file that maps each display pixel to the multiple corresponding camera pixels. We then use this information to crop images taken by the camera down to just the pixels that belong to the display.

3.2 Color Calibration

Before we can correct color distortion we must map the inputted color to the outputted, distorted color. For the purposes of this paper we will define the inputted color that belongs to the initial scene as belonging to the ideal color space and the color resulting from camera and display distortion as belonging to the distorted color space. To map the distorted color space to the ideal color space we begin by generating images that contain 60 by 60 pixel color swatches of every possible combination of the ideal red, green, and blue values on an interval of 15, from 0 to 255. With the camera set to RAW mode we capture these images. After cropping these images we average the resulting RGB values in the distorted color space so that they map to the corresponding RGB values in the ideal color space. Averaging a total of 3,600 pixels per value provides us with enough information for further interpolation. Accuracy could be increased by using larger color swatches, however this would require more images to be processed. If we collected data on interval of 1, 28 thousand images would be needed to be processed, each taking 18 seconds to run. We found that 60 by 60 pixels swatches on intervals of 15 provide the best balance between accuracy and efficiency. This provides us with 4,913 data points that are outputted into a new color mapping text file.

3.3 Barycentric Interpolation

The mapping data on intervals of 15 fills less than 10% of the ideal color space thus is insufficient to accurately correct color distortion. We use barycentric coordinates to interpolate between the ideal color space and the distorted color space to fill the remainder of the ideal color space on intervals of 1. More specifically, given mapping data on intervals of v, barycentric coordinates are used to estimate the mapping data on subintervals u, where $v \neq u$ and $v > u$. Barycentric coordinates are coordinates of a point with respect to a simplex, which is the generalized notion of a triangle in n-dimensions, in terms of weights placed on the simplex's vertices. Hence a point r with a simplex of vertices $r_1, r_2, ...r_n$, can be written as

$$r = \lambda_1 r_1 + \lambda_2 r_2 + \ldots \lambda_n r_n \tag{1}$$

,where n is the dimension of the simplex + 1 and $\lambda_1, \lambda_2, ...\lambda_n$ are the weights placed on the vertices. From this equation we can derive a formula of calculating the λs. This formula can be written as the following linear transformation:

$$T \cdot \lambda = r - r_n \tag{2}$$

Where

$$T = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ z_1 & z_2 & z_3 & \dots & z_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_1 & d_2 & d_3 & \dots & d_n \end{bmatrix}$$

where d represents the nth dimension. We can then find the inverse of matrix T, T^{-1} and multiply it to the vector $(r - r_n)$ to find λ, i.e. $\lambda = T^{-1} \cdot (r - r_n)$.

To use the barycentric coordinates to interpolate between the distorted and the ideal spaces we split the spaces into simplexes, in this case tetrahedrons. The ideal space is tiled into cubes of length v, the size of the interval of the gathered data. These cubes are then split into 5 tetrahedrons, the vertices of which we use to find the corresponding points in the distorted space which form a corresponding tetrahedral. Thus for a given cube C_i there will be 5 tetrahedrons in the ideal space, t_i^k and 5 tetrahedrons in the distorted space, t_d^k, where $k = 1, 2, 3, 4, 5$. For a given point p_i in the cube, the barycentric coordinates are calculated with respect to t_i^k.Going through the tetrahedral we check if there are any negative weights, if there are the point lies outside of the simplex. If so, we use the next tetrahedral t_i^{k+1}. If not then those weights and the corresponding vertices of t_d^k in the distorted space are used in formula 1 to obtain the point p_d to which p_i is estimated to be mapped. We repeat this process finding all points p_i that are of subinterval u of each other in the cube C_i and for every C_i in the ideal color space. All of the p_is and p_ds are outputted into a new

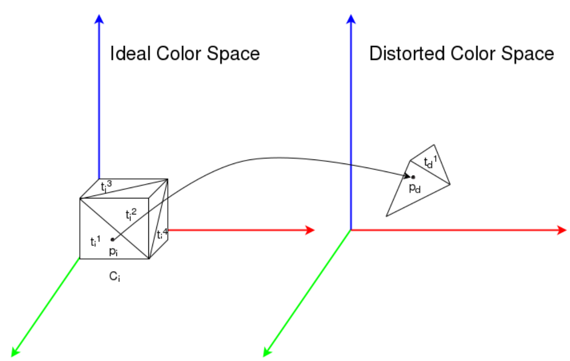


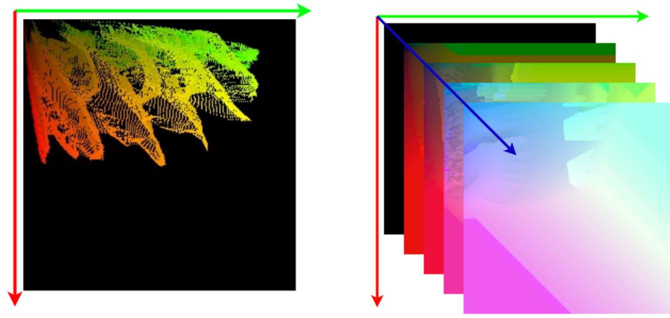
Figure 1: The splitting of cube C_i into tetrahedrons and the mapping of point p_i to p_d

color mapping file. With interval $x = 15$ and subinterval $y = 1$, the output file is over 16 million lines, and maps the entire ideal space.

3.4 Creating the Color Cube

With a complete distorted to ideal color mapping file in place, we are now able to accurately revert the camera-display distortion of a captured image. To organize this data efficiently, we encode it into images. We develop this model initially only considering the red and green color channels, noting that with minor modifications the process can be expanded to include blue. For this two color channel model, we focus on depicting the mapping of the distorted red and green color channels to their corresponding values in the ideal color space. We use the distorted red values as the image's y coordinates and the distorted green values as the x coordinates. The created image is 256 by 256, thus containing all possible unique combinations of red and green. We then display the distorted to ideal relationship by setting the pixel value at each coordinate to the red and green values of the corresponding point in the ideal color space. This method produces a single image with the encoded color mapping data as seen in 2a.

Our strategy for expanding this two dimensional model to three dimensions is to add a depth element to this image. Using the distorted blue value as the z coordinate we create 256 two dimensional mapping images, where each image represents a unique point on the z axis. These images form a color cube in which all possible values in the red, green, and blue distorted space are mapped to a value in the red, green, and blue ideal space. Figure 2b depicts this cube.

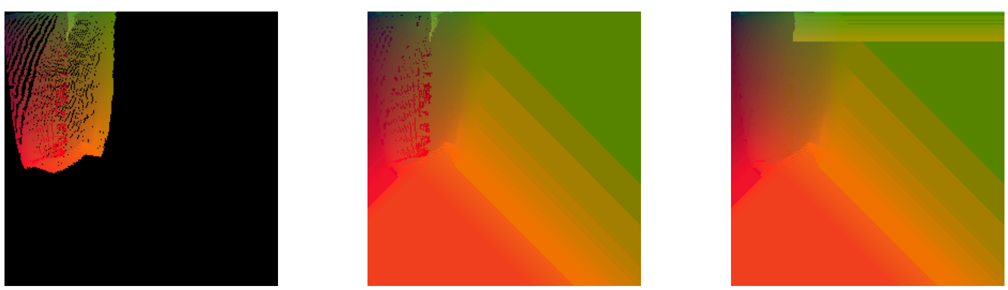


(a) 2 Dimensional Map **(b)** 3 Dimensional Map

Figure 2: The distorted color maps; in both 2 and 3 dimensions

Many pixels in the distorted space have no recorded corresponding point in the ideal space and are thus mapped to pure black. As seen in figure 3a, this forms "holes" in our color cube and decreases accuracy. To counteract this problem, we estimate the red and green pixel values of these holes by setting each black pixel to the value of the nearest colored pixel. The result of this process can be seen in figure 3b.

While this filled color cube provides better results, certain outliers still exist in the images thus causing a speckling effect in the created ideal image. To remove these outliers, we "smooth" out the images by ensuring that each pixel does not differ more than 20 green values from its neighbors. We arrived at this solution by analyzing the average difference between neighboring pixels and found that outliers tended to differ from their neighbor by greater than 20 green values. The results of this smoothing process are shown in figure 3c.



(a) Initial **(b)** Filled **(c)** Smoothed

Figure 3: Process of creating mapping images

This complete process creates a satisfactory color cube 2b which accurately maps distorted color values to perceived real world colors.

4 Results

Using this encoded mapping data we apply a pixel by pixel transformation to a captured image, restoring it to real world colors. To measure this operation's accuracy we compare the captured ideal image with the original captured scene image by calculating the RGB value differences between corresponding pixels. Figure 4 depicts resulting images of applying restoration process to an image of a still life. We expect that the camera distorts both the ideal image and and the scene in the same way. If our colors in the ideal image accurately match real world colors, the colors in the captured ideal image and captured scene image should match. Using this test we determined that we are able to match real world coloring in our ideal image with only a 5% average margin of error throughout the image. The ideal image's colors are on average 13 RGB values off from the captured scene image.

(a) Ideal Image **(b)** Captured Ideal **(c)** Captured Scene

Figure 4: Example 1 of restored still life resulting images

(a) Ideal Image **(b)** Captured Ideal **(c)** Captured Scene

Figure 5: Example 2 of restored still life resulting images

5 Future Work

The source of the inaccuracy in our approach primarily stems from inconsistencies in data collection quality. We suspect that this heavily correlates with variation in camera calibration. The calibration often contains unmapped and mismapped pixels. In under less than ideal conditions this causes holes and smudging in images produced using this data. This error translates to further inaccuracies when correcting color distortion. Therefore we believe the best way to improve accuracy would be to improve the camera calibration, which would in turn improve the color calibration data.

6 Conclusion

We present a new method of reverting color distortion caused in a camera-display system. Using color mapping and barycentric interpolation we are able to produce real world colors in an image within a 5% average margin of error. This method could be applied towards achieving more realistic colors in virtual reality, three dimensional modeling using two dimensional images, and teleprescence [3].

References

[1] M.D. Grossberg, H. Peri, S.K. Nayar, and P.N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–452–I–459 Vol.1, June 2004.

[2] Srikanth Kirshnamachari Sridharan, Juan David Hincapié-Ramos, David R. Flatla, and Pourang Irani. Color correction for optical see-through displays using display color profiles. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, VRST '13, pages 231–240, New York, NY, USA, 2013. ACM.

[3] J. Edwards. Telepresence: Virtual reality in the real world [special reports]. *Signal Processing Magazine, IEEE*, 28(6):9–142, Nov 2011.