Color Correction Methods with Applications to Digital Projection Environments

Jianfeng Yin
Centre for Intelligent Machines
McGill University
3480 University Street
Montreal, QC H3A 2A7 Canada

jfyin@cim.mcgill.ca

Jeremy R. Cooperstock
Centre for Intelligent Machines
McGill University
3480 University Street
Montreal, QC H3A 2A7 Canada

jer@cim.mcgill.ca

ABSTRACT

Due to differing optics, sensor characteristics, and hardware processing employed by video cameras, the resulting colors produced by two cameras can be very different, thus complicating the task of computer vision applications. While various color correction methods exist to deal with this problem, most involve strong assumptions, such as constant illumination, that are, in general, unsatisfied in complex environments. To address the problem of color correction in a less restrictive manner, we propose the use of neural networks, which can easily be trained and which produce excellent results. We compare these results with other methods.

Keywords
Color Correction, BPNN.

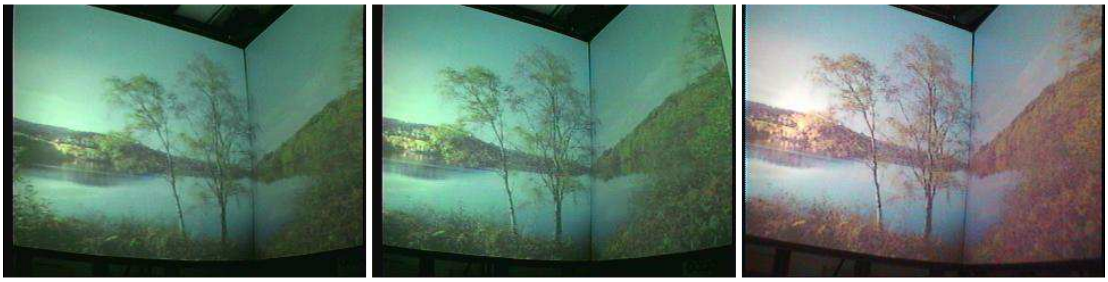
1 INTRODUCTION

Color provides useful information for many vision applications. However, the video acquisition process is dependent on a number of factors including illumination, optics, sensor characteristics, and hardware processing. As a result, different cameras typically produce different color values for the same objects or scenes, as illustrated in Fig. 1. These differences complicate the task of computer vision applications involving the use of more than one camera. An approach is thus required to correct the images so that colors of the same object appear to be similar in the output from each camera.

This correction typically takes one of two forms. In the first, the mapping is found between the true color values (which may be unavailable in many cases) and the observed colors for each camera, while in the second, the transform between each camera and one reference camera is found. The latter approach is generally simpler, as it is determined solely by the camera parameters and the mapping can be characterized by a relatively small number of data samples. In this paper, we will focus more on the second case, that is, we concern about improving the color consistency among cameras. We assume, in either case, that the cameras focus on approximately the same portion of the scene, and thus receiving very similar visual information. However, we do not wish to impose additional constraints, such as an assumption of uniform illumination or matte objects.

A closely related but different problem is that of color constancy [Bar02a, Bar02b], in which a relationship is sought between surface colors and illumination, in order to map the observed color to the correct one under some canonical illumination [For90]. Common solutions include the *gray world* approach, which assumes that the average color in an image is gray, the *white patch* approach, derived from retinex theory [Lan77], which assumes that the maximum value of each channel is white, and neural network methods [Car00], which estimate the illuminant chromaticity of an image using a neural network, which usu-

Journal of WSCG, Vol.12, No.1-3., ISSN 1213-6972
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.



a b c

Figure 1: images captured by three different cameras.

ally needs a large database of illuminants and reflectances(surfaces) for training. They also assume each image is taken under one uniform illuminant which is not valid in our environment. [Fin95] considers varying illumination, but it assumes a difference in illumination can be identified. Other approaches involving gamut mapping methods [For90, Fin96] and Bayesian methods [Bra97] require either large datasets of reflectance spectra from a wide variety of common objects or knowledge of the camera sensor responses, both of which are generally difficult to obtain. Some other related works on color calibration and color reproduction can be found in [Jac97, Kan92, Vrh99, Tom99]. In this paper, we focus on finding the relationship between cameras, that is, we try to make color appear consistent between them. Thus, we do not need to estimate the chromaticity of the actual illuminant, nor do we require a large training set as the problem can be addressed sufficiently with relatively sparse data and a simple method.

For our specific application, we are interested in color correction for an immersive environment employing digital rear projection (see Fig. 4), in which the output of several, possibly heterogeneous cameras, must be correlated. In such an environment, the color of any pixel as registered by each camera is affected by many factors, including non-uniform background lighting conditions, projector color gamut, uneven intensity distribution over the screen, and differing camera poses and sensitivities. As a result, the appearance of colors obtained by the cameras can vary widely, as pictured in Fig. 1. While an inter-projector calibration that produces a uniform color response across projectors would help reduce these effects, the problem of different camera responses to these pixels remains. It is this problem on which we focus here, leaving for future work the question of projector calibration.

Following an overview of other color constancy methods, we investigate several options for dealing with the color correction problem. We first examine linear methods in Section 2 and then compare these with our proposed approach of a neural network in Section 3, concluding the paper with a summary of experimental results in Section 4.

2 LINEAR COLOR CORRECTION METHODS

In this section, several methods based on linear models are discussed. The RGB color space is used in the paper because it is the most popular space used in sensor and display devices. Most methods in the paper should be applicable to other color spaces, but a comparison of different color spaces is beyond the scope of the current paper. The color transfer method in section 2.4 converts the RGB space to an $l\alpha\beta$ space first, then works on that space, and converts back to RGB at the final step. The least squares approximation method (section 2.3) requires the estimation of a transform matrix, which is similar in approach to the training of a neural network, as described in section 3. However, the remaining methods are based only on single camera models, and as such, do not undergo any training or estimation step.

2.1 Gray world (GW)

The *gray world* approach assumes the average color of an image is some predefined value of "gray," for example, half the value of the maximum intensity for each color component, i.e., (128,128,128). Based on this assumption, image colors are corrected through the following normalization:

$$\begin{aligned} R_n &= R_o * 128/\bar{R} \\ G_n &= G_o * 128/\bar{G} \\ B_n &= B_o * 128/\bar{B} \end{aligned} \quad (1)$$

where, (R_o, G_o, B_o) is the original color, $(\bar{R}, \bar{G}, \bar{B})$ is the average color, and (R_n, G_n, B_n) is the corrected color. One might also consider the use of the average color components $(\bar{R}, \bar{G}, \bar{B})$ from an arbitrary reference camera, and use these, rather than the fixed value (e.g. 128) as the normalizing term. However, this may suffer problems if the reference camera's color distribution is not well balanced.

2.2 White patch (WP)

The *white patch* approach is similar to the *gray world* method but assumes that the maximum value of each channel should correspond to full white, i.e. (255, 255, 255). Image colors are corrected through the following normalization:

$$\begin{aligned} R_n &= R_o * 255/R_m \\ G_n &= G_o * 255/G_m \\ B_n &= B_o * 255/B_m \end{aligned} \quad (2)$$

where, (R_o, G_o, B_o) is the original color, (R_n, G_n, B_n) is the corrected color, and R_m, G_m, and B_m are the maximum observed color components in the three channels, respectively.

Again, we may consider using one camera as a reference, with the same caveats as earlier.

2.3 Least squares (LS) approximation

The gray world and white patch approaches use diagonal matrix transforms, assuming that the different channels are independent. While various research suggests that diagonal transforms should suffice [Fin93], or suffice with sensor sharpening [Fin94], this is not the case in general with complex scenes. Worse still, sensor sharpening techniques may be unstable [Bar01].

Instead, we consider the use of a full matrix transform, i.e.,

$$(\mathbf{C_1'}, \mathbf{C_2'}, \ldots, \mathbf{C_n'}) = \mathbf{T} \cdot (\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_n}) \quad (3)$$

in which $\mathbf{C_i'}$ and $\mathbf{C_i}$ $(i = 1, \ldots, n)$ are colors from two different cameras and $\mathbf{T}$ is the transformation matrix between them. From a set of corresponding colors from two cameras, $\mathbf{T}$ can be estimated by least squares approximation methods. Here, we use the color from the first row of Fig. 2 to estimate $\mathbf{T}$. The image from Fig. 2c) is taken as the standard color or reference, from which we estimate transforms between it and the images produced by the other two cameras. These transforms are then used to correct the colors.

2.4 Color transfer between images

Reinhard *et al.* [Rei01] proposed a color transfer method that can be applied to color correction. It first decorrelates the RGB values to an $l\alpha\beta$ color space and then computes the statistics (mean and standard deviation) of source and target images. The source colors are corrected by scaling and offsetting according to the mean and standard deviation of the target image, as follows:

$$\begin{aligned} l_s' &= (l_s - \overline{l_s}) * \frac{\sigma_t^l}{\sigma_s^l} + \overline{l_t} \\ \alpha_s' &= (\alpha_s - \overline{\alpha_s}) * \frac{\sigma_t^\alpha}{\sigma_s^\alpha} + \overline{\alpha_t} \\ \beta_s' &= (\beta_s - \overline{\beta_s}) * \frac{\sigma_t^\beta}{\sigma_s^\beta} + \overline{\beta_t} \end{aligned} \quad (4)$$

Where, $\bar{c}_i, \sigma_i^c, c = l, \alpha, \beta$ is the mean and standard deviation of an image for each channel, respectively, and $i = s, t$ refers to the source and target, respectively. Following this transform, the image is converted back to RGB space. The conversions between $l\alpha\beta$ and RGB spaces can be found in [Rei01].

We also consider the use of color transfer, followed by the gray world method, in order to normalize the results of the transfer. The results of this combination, as illustrated later in Fig 5 and Fig 6 appear to be superior than the color transfer itself.

3 NEURAL NETWORK COLOR CORRECTION

The previous methods are all based on linear models[1], which, for complex scenes, sometimes prove inadequate to correct colors from different cameras. While it is often difficult to find a suitable, explicit, nonlinear representation neural network methods [Car00] have been shown capable of performing similar tasks, such as estimating the illumination of an image, given a large database of known illuminations and surface colors.

3.1 Neural Network Architecture

The network architecture used was a simple two layer backpropagation network (BPNN) with 10 hidden layer neurons, as shown in Fig. 3. The inputs are the source RGB values, and the outputs are the corrected RGB values. The network tries to minimize errors between the estimated colors and the true colors. The general theory about BPNN training can be found in many neural network books, e.g., [Hay99]. Since we need only find the relation between colors from different cameras, assuming the same lighting is applied to the views of each camera, a simpler training set proves to be sufficient. The training data consists of 216 color checkers, uniformly distributed in the RGB

[1] Color transfer is linear in $l\alpha\beta$ space.

a b c

Figure 2: 1st row: training data; 2nd row: validation data.

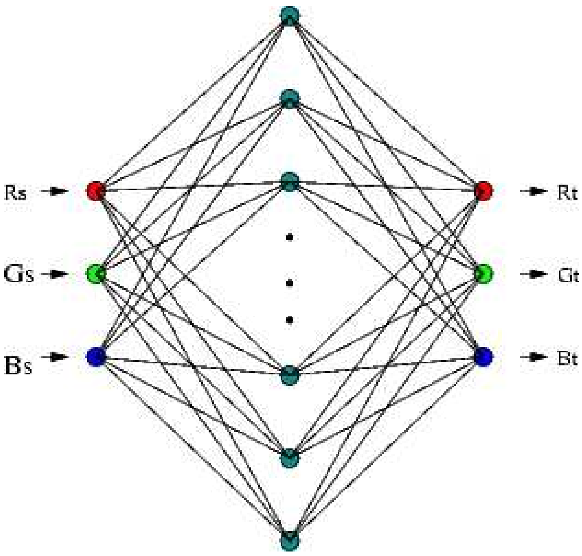


Figure 3: the BPNN architecture

space while the validation data is provided by the Macbeth 24-color checkers,[2] as shown in Fig. 2, . The samples are collected manually by clipping a rectangle in each color square and then computing its mean value. The values obtained from one of the cameras (Fig. 2c) are taken as the reference and the color values from the remaining cameras are corrected accordingly.

[2]The industry-standard color checking chart.

3.2 Empirical Results

To evaluate the performance of these methods empirically, we measure the error both as absolute differences of the individual color components, δ_r, δ_g, and δ_b, as well as the Euclidean distance between the components of the true object color, $\mathbf{x_t}$, and the estimated color, $\mathbf{x_e}$, as follows:

$$Err = \frac{\sqrt{(r_t - r_e)^2 + (g_t - g_e)^2 + (b_t - b_e)^2}}{255} * 100\% \tag{5}$$

where 255 is the maximal value for each color component.

The error statistics for a corrected camera corresponding to the image of Fig. 1a are provided in Table 1. Since both corrected camera images exhibited similar results, we only list one of these here. For comparison purposes, we include the results obtained by the other methods described in section 2. Since the least squares approximation method involves the estimation of a transform matrix based on observed data, this is similar in approach to the training session of the neural network, so it is also meaningful to compare performance on independent test data. However, the other methods do not include such training steps, so no comparison with test data is relevant.

The results obtained demonstrate, both quantitatively and qualitatively, the superiority of the backpropagation neural network. Complex images, such as those shown in Fig. 5, exhibit significantly better correction by the BPNN method than with the other approaches.

	training set		validation set	
	average error	standard deviation	average error	standard deviation
GW	6.53 (4.43,2.05,3.31)	3.26 (3.17,1.62,2.39)	-	-
WP	8.89 (6.25,3.16,3.86)	4.65 (4.61,2.30,3.17)	-	-
COL_TRANS	7.33 (4.84,2.62,3.67)	3.39 (3.65,2.28,2.67)	-	-
COL_TRANS+GW	6.14 (3.39,2.43,3.03)	3.30 (3.82,1.72,2.92)	-	-
LS	7.22 (5.02,1.87,3.76)	3.58 (3.35,1.65,2.87)	15.75 (12.13,5.37,6.43)	5.53 (5.46,3.81,4.28)
BPNN	3.69 (1.89,1.38,2.25)	2.15 (1.63,1.11,1.95)	14.03 (9.94,6.50,5.89)	7.02 (5.30,4.62,4.68)

Table 1: Error statistics for color correction applied to one of the cameras. The top row of each cell corresponds to the Euclidean error metric, whereas the bottom row corresponds to the individual component differences, $\delta_r, \delta_g, \delta_b$.

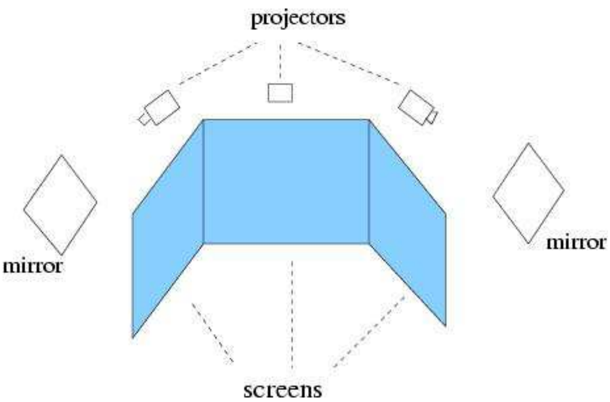


Figure 4: Illustration of the rear projection environment used for our application.

3.3 Digital Projection Results

We applied the various color correction methods described previously to a set of sample images taken by video cameras in our environment. The results are shown in Fig. 5 and Fig. 6. Since some of the simple methods are based on strong assumptions such as constant illumination, which are not satisfied in our environment, these often fail to perform adequately.

Again, the neural network method outperforms other strategies. This is particularly evident in Fig. 6, in which the presence of a person standing in the scene changes the illumination level from that used during training. In this example, all of the methods apart from the neural network approach exhibit noticeable degradation. While the transform matrix, T, used for the least squares and the neural network methods were estimated or trained using the data of Fig. 2, i.e. independent of both test scenarios, only the neural network method proved to be robust to the change in illumination.

4 CONCLUSIONS

We have considered the problem of color correction for a set of heterogeneous cameras in a general environment, in which constant illumination cannot be assumed. Various methods, based on solutions to the color constancy problem, were applied to this task and their results compared. We found that under non-idealized conditions, our proposed use of a simple backpropagation neural network achieves results that are superior to other methods for correcting images from different cameras to produce results that appear similar to each other in color. The neural network method allows for simple training and proves to be robust to significant scene variations.

Although we have only evaluated these approaches within our rear projection environment described previously, we see no reason why the neural network approach would not succeed equally well in other environments or on natural scenes, provided suitable training data, such as the Munsell color checkers, can be used.

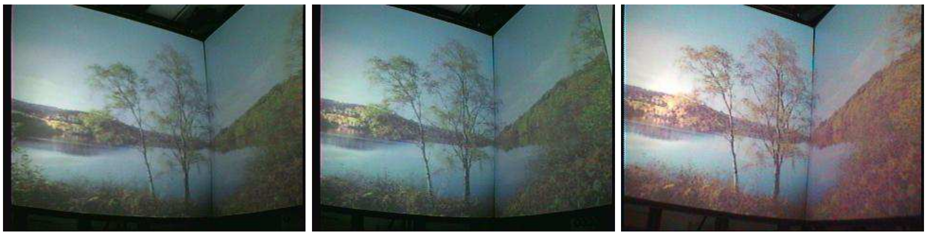
An interesting avenue for ongoing research that are now considering is how to extend these results to the far more challenging problem of color correction for heterogeneous projection equipment and cameras that are no longer viewing the same portion of the scene.

References

[Bar02a] Barnard, K., Cardei V., and Funt B. A Comparison of Computational Color Constancy algorithms—Part 1: Methodology and Experiments with Synthesized Data. IEEE Trans. Image Processing, Vol. 11, No. 9, pp. 972-983,2002.

[Bar02b] Barnard, K., Martin, L., Coath, A., and Funt, B. A Comparison of Computational Color Constancy algorithms—Part 2: Experiments with Image Data. IEEE Trans. Image Processing, Vol. 11, No. 9, pp. 985-996, 2002.

[Bar01] Barnard, K., Ciurea, F., and Funt, B. Sensor sharpening for computational color constancy. J. Opt. Soc.

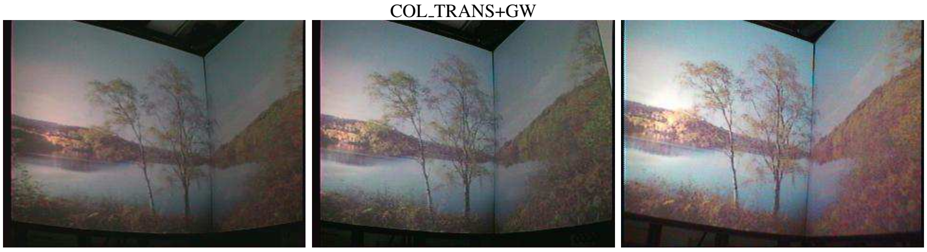


WP

GW

COL_TRANS

COL_TRANS+GW



LS

BPNN

Figure 5: Comparison of color correction results for the original scene.

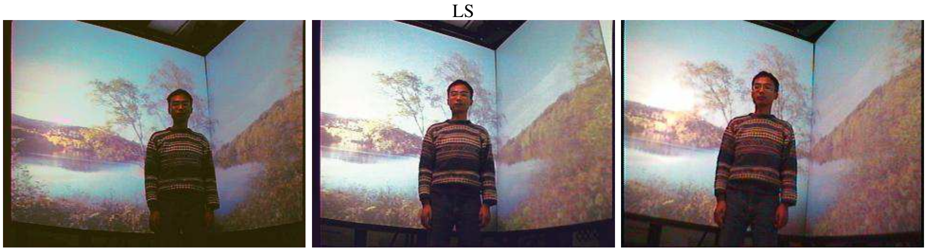
Original images

GW

COL_TRANS

COL_TRANS+GW

LS



BPNN

Figure 6: Color correction results including a person standing in the scene.

Amer. A, vol. 18, pp. 2728-2743, 2001.

[Bra97] Brainard, D. H. and Freeman, W. T. Bayesian color constancy. J. Opt. Soc. Amer. A, vol. 14, pp. 1393-1411,1997

[Car00] Cardei, V. A neural network approach to color constancy. PhD thesis, Simon Fraser Univ., Burnaby, BC, Canada, 2000.

[Fin93] Finlayson, G. D., Drew, M. S., and Funt, B. V. Diagonal transforms suffice for color constancy. International Conference on Computer Vision, pp. 164-171, 1993.

[Fin94] Finlayson, G. D., Drew, M. S., and Funt, B. V. Spectral sharpening: Sensor transformations for improved color constancy. J. Opt. Soc. Amer. A, vol. 11, pp. 1553-1563, 1994.

[Fin96] Finlayson, G.D. Color in perspective. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, pp. 1034-1038, 1996.

[Fin95] Finlayson, G.D., Funt, B.V., and Barnard, K. Color constancy under varying illumination. Proc. 5th Int. Conf. Computer Vision, pp. 720-725, 1995

[For90] Forsyth, D. A novel algorithm for color constancy, International Journal of Computer Vision, vol. 5, pp. 5-36, 1990.

[Hay99] Haykin, S. Neural Networks – a Comprehensive Foundation. Prentice Hall, New Jersey, 2nd edition, 1999.

[Jac97] Jackowski, M., Goshtasby, A., Bines, S., and Roseman, D. Correcting the Geometry and Color of Digital Images. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No. 10, pp. 1152-1158, 1997.

[Kan92] Kang H.R. and Anderson, P.G. Neural network applications to the color scanner and printer calibrations. Journal of Electronic Imaging 1(2), pp. 125-135, 1992.

[Lan77] Land, E.H. The Retinex theory of color vision. Scientific American, vol. 237, pp. 108-129, 1977.

[Rei01] Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. Color Transfer between Images. IEEE Graphics and Applications,Vol. 21, No. 5, pp. 34-41, 2001.

[Tom99] Tominaga, S. Color coordinate conversion via neural networks. Colour Imaging: Vision and Technology, *ed.* Lindsay W. MacDonald and M. Ronnier Luo, Hohn Wiley & Sons Ltd., pp. 166-178, 1999.

[Vrh99] Vrhel, M. J. and Trussell, H. J. Color Scanner Calibration via a Neural Network. IEEE ICASSP'99, Vol. 6, pp. 3465-3468, 1999.