ADICT: Accurate Direct and Inverse Color Transformation

Behzad Sajadi, Maxim Lazarov, and Aditi Majumder

University of California, Irvine
{bsajadi,mlazarov,majumder}@uci.edu

Abstract. A color transfer function describes the relationship between the input and the output colors of a device. Computing this function is difficult when devices do not follow traditionally coveted properties like channel independency or color constancy, as is the case with most commodity capture and display devices (like projectors, camerass and printers). In this paper we present a novel representation for the color transfer function of any device, using higher-dimensional Bézier patches, that does not rely on any restrictive assumptions and hence can handle devices that do not behave in an ideal manner. Using this representation and a novel reparametrization technique, we design a color transformation method that is more accurate and free of local artifacts compared to existing color transformation methods. We demonstrate this method's generality by using it for color management on a variety of input and output devices. Our method shows significant improvement in the appearance of seamlessness when used in the particularly demanding application of color matching across multi-projector displays or multi-camera systems. Finally we demonstrate that our color transformation method can be performed efficiently using a real-time GPU implementation.

1 Introduction

A color transfer function matches colors in a device-dependent RGB space to those in the device-independent CIE XYZ space. For capture devices, the input is in the XYZ space and the output in the RGB space and vice-versa for display devices (like projectors and printers). When the domain and the range of this function are the RGB and the XYZ spaces respectively, it is a *direct transfer function*. When the domain and the range are switched, the function is an *inverse transfer function*. Let us consider two devices—a source and a target—with direct transfer functions T_s and T_t. The color given by (r_s, g_s, b_s) in the source device can be achieved by the input $(r_t, g_t, b_t) = T_t^{-1}T_s(r_s, g_s, b_s)$ in the target device. Here T_s is the direct transfer function and T_t^{-1} is the inverse transfer function. Accurate computation of $T_t^{-1}T_s$ is the goal of any color management system. For ideal devices with channel color constancy (constant chromaticity across all channel inputs) and no channel interdependencies, both the direct and inverse transfer functions, T_s and T_t, are linear 3×3 matrices. This allows easy inversion and concatenation to compute desired target inputs that create the

same image as in the source. However, current commodity devices like projector, cameras and printers deviate considerably from the ideal properties of channel independency and color constancy, making it difficult to compute $T_t^{-1}T_s$.

In this paper, we represent the color transfer functions of any non-ideal device using multiple higher-dimensional Bézier patches. Augmenting this with a novel reparametrization of colors in the device independent space, we design a new color transformation method (Section 3) with the following advantages.

Generality: The Bézier based representation of the color transfer function does not depend on ideal behavior of the device. Hence, it can be applied to devices with significant channel interdependencies (often due to the use of more than three primaries that do not form a basis), no color constancy, and non-monotonic channel transfer functions. It is also backward compatible to ideal devices and can be applied to both capture and display devices alike (Section 5).

Quality: Unlike existing color transformation methods (Section 2), our method assures (C^2) continuity, resulting in smoother transition from one color to another. It also handles non-monotonicity in the transfer function elegantly. Consequently, our method consistently shows greater accuracy and less local artifacts compared to the existing methods when tested on a variety of devices including projectors, cameras and printers (Section 5).

Application: The quality of any color matching method is particularly challenged in applications where images from multiple devices are placed in a spatially contiguous manner, where human ability to detect color differences increases [1]. This occurs when tiling multiple display devices (e.g. tiled display walls) or when stitching different parts of a scene captured with different cameras in a panorama (e.g. surveillance application). In these applications, our method shows significant improvement in color matching, especially when the devices differ significantly in color properties (Section 5). Thus, such applications no longer are restricted to use devices of same model or architecture to avoid large variations in color.

Efficiency: Our color transformation method is amenable to real-time implementation on GPU(Section 4).

2 Previous Work

Let $i_l, l \in \{r, g, b\}$, $0 \leq i_l \leq 1$, be a channel input in the device-dependent RGB space. The color gamut of an ideal device (exhibiting channel independence and color constancy) in the XYZ space is a parallelepiped spanned by three vectors (X_l, Y_l, Z_l), one per channel l. The color in the XYZ space corresponding to (i_r, i_g, i_b) in the RGB space is then given by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} i_r \\ i_g \\ i_b \end{pmatrix} = M \begin{pmatrix} i_r \\ i_g \\ i_b \end{pmatrix} \tag{1}$$

Here, the forward transfer function T is the matrix M and can be estimated accurately by measuring the color output at only three points, namely $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Also, T can be easily inverted by computing M^{-1}.

To match the response of the ideal device with that of the human eye, each channel usually has a non-linear channel transfer function, given by $h_l, l \in \{r, g, b\}$. Accounting for h_l, Equation 2 can be written as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \begin{pmatrix} h_r(i_r) \\ h_g(i_g) \\ h_b(i_b) \end{pmatrix} \tag{2}$$

Here, to estimate T we need to reconstruct h_l in addition to M [2]. This involves measuring color output at k uniformly sampled channel inputs i_l when inputs to the other two channels are zero. So, for accurate representation of T, we need $3k$ samples. Inverting T involves two steps: (a) Applying M^{-1}; (b) Applying the inverse channel transfer functions h_l^{-1}. To assure invertibility, h_l is assumed to be monotonic, a property satisfied by most traditional devices like CRT displays. Most work on color management assume channel independence and additivity common in traditional displays [3, 4].

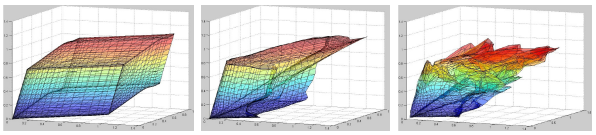

Fig. 1. Visualization of color gamuts of different projection technologies in CIE XYZ space. From left to right: LCD, LCoS and DLP

Current commodity devices are far from such ideal devices. First, the chroma (defined by the chromaticity coordinates) is not constant across a channel. This means that the boundaries of the parallelepiped (and even iso-contours within it) are no longer straight lines, but wiggly curves. Second, the channels may not be independent. This means that the color transformation is non-additive, i.e. $T(I_1) + T(I_2) \neq T(I_1 + I_2)$, where I and $T(I)$ denote the colors in RGB and XYZ space. In this case, the color gamut ceases to be a parallelepiped. Usually, this is due to the fact that many consumer devices have more than 3 primaries that do not form a basis in the XYZ space [5]. This includes the DLP projectors that use an additional white channel and the printers that use an additional black ink. More recently multi-primary cameras and LED/laser projectors are being introduced [6, 7]. Figure 1 shows the effect of such non-ideal behavior on the shape of the 3D gamuts in the XYZ space.

Though some attempts has been made to reconstruct T in non-ideal systems being within the domain of linear matrices [8], they can only tolerate small deviations from ideal devices. The simplest way to reconstruct the function T for non-ideal systems is to sample the three dimensional RGB space uniformly to create a piecewise linear approximation of T. The inversion of the function would then involve a linear interpolation of the function from its neighbors [9]. However, this assumes monotonic iso-contours in T and a dense sampling of the domain. Assuming k samples per channel, k^3 samples are required for accurate estimation. However, k has to be relatively large ($k = 16$) when compared to our method ($k = 9$) to provide a dense sampling assuring more accurate local interpolation. Even for a relatively small $k = 16$, 16^3 measurements can be very time consuming. Most importantly, linear interpolation assures only C^0 continuity creating considerable visual artifacts (Section 5).

Hence, many current systems custom tailor T for specific devices. Windows Color Management System (WCS), designed jointly by Canon and Microsoft, is a good example [10] and uses different techniques to compute the transfer functions of different devices. For projectors, first the input RGB space is sampled uniformly and the corresponding XYZ values are measured. Let the sampled input values be $(r_i, g_i, b_i), 1 \leq i \leq n$ and the XYZ values (X_i, Y_i, Z_i). For each (X_i, Y_i, Z_i), the input (r'_i, g'_i, b'_i) that would result in (X_i, Y_i, Z_i) is predicted assuming an ideal device described by Equation 2. Next, the vector deviation $\bar{d}_i$ of the actual input (r_i, g_i, b_i) from the predicted input (r'_i, g'_i, b'_i) is computed and associated with the corresponding sampled input. To compute the input for desired XYZ values (X_d, Y_d, Z_d), first input (r'_d, g'_d, b'_d) is predicted assuming an ideal device using Equation 2. Then the deviation is linearly interpolated from the sampled $\bar{d}_i$s and added to (r'_d, g'_d, b'_d) to generate the final input (r_d, g_d, b_d). Unfortunately, in addition to C^1 discontinuity, it can result in non-monotonic output even though there is no non-monotonic iso-parametric curves or surfaces in the input. Further, it assumes channel color constancy that is not true in most commodity devices and results in severe color anomalies (Section 5).

Note that all the above related work, including our work, focus on color management techniques that are content agnostic, i.e. does not depend on the image content. Hence, the device once calibrated, can correct any image. However, our work is orthogonal to a body of literature on content-dependent color management schemes. These determine the best possible color mappings for a specific image based on the particular spatial distribution of the colors to achieve the most perceptually pleasing appearance in the new device [11–15]. Hence, in a content-dependent scheme the color mapping is unique for each image and needs to be recomputed for every image. Content dependent methods cannot achieve interactive rates for videos unless special hardware is used.

3 Algorithm

We present a new general way to represent the color transfer functions of a non-ideal device, both direct and inverse, using a set of Bézier patches. A critical aspect of this representation is a non-linear parametrization of the Bézier patches in the XYZ space (Section 3.1). Using this representation we design a new color transformation method for converting the RGB colors in a source device to those in a target device significantly different than the source device (Section 3.2).

3.1 Color Transfer Function

We do not assume channel chrominance constancy, channel independence, or monotonic channel transfer functions. However, we assume T to be a smooth function. Let us sample T at n different RGB points, i.e., for these n samples – $(r_1, g_1, b_1), (r_2, g_2, b_2) \ldots (r_n, g_n, b_n)$ – we know the corresponding outputs – $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2) \ldots (X_n, Y_n, Z_n)$. Our direct transfer function T is represented by a set of Bézier functions $\mathcal{F}_c(r, g, b)$, one for each XYZ channel,

$c, c \in \{X, Y, Z\}$. Our inverse transfer function T^{-1} is represented by another set of Bézier functions $\mathcal{B}_l(X, Y, Z)$, one for each RGB channel, $l, l \in \{r, g, b\}$.

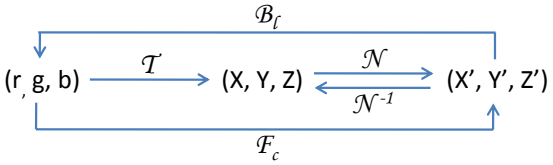

Fig. 2. Representation of color transfer functions.

We first describe the reconstruction of $\mathcal{B}_l$. $\mathcal{F}_c$ is computed similarly, but the domain and range color spaces are switched (Figure 2). We reconstruct $\mathcal{B}_l$ for each RGB channel by *fitting* a Bézier to the measured n samples. This consists of two steps. We measure the X, Y, Z values for various input values of r, g, and b uniformly distributed in the RGB space. For every input $\{r_i, g_i, b_i\}$ let the measured output color be $\{X_i, Y_i, Z_i\}$. Using these samples, we build three (one for each channel) 3D Bézier volumes in 4D – $B_r(X, Y, Z)$, $B_g(X, Y, Z)$, and $B_b(X, Y, Z)$. In other words, our Bézier surfaces are height fields in the XYZ space, one for each of the three input channels.

Although the data points we collect are uniform in the RGB space, they need not be uniformly distributed in the XYZ space, limiting our ability to fit a Bézier surface to the XYZ colors. Hence we apply a non-linear function $\mathcal{N} : (X, Y, Z) \rightarrow (X', Y', Z')$ in order to make the distribution in the XYZ space uniform. We fit the Bézier surface in this modified XYZ space. The reparametrized XYZ space is sampled in a relatively uniform fashion, and the control points of the Bézier are placed in a regular grid (Figure 3), thus fixing three of the four coordinates of the control points. We use a linear least square method to fix the fourth coordinate so that the computed Bézier height field smoothly passes through the data set. Details of the non-linear reparametrization is available in Section 3.1.

The Bézier representation suits non-ideal devices due to the following:

- Since the Bézier is just a polynomial representation, we can represent non-constant higher order variations in the channel chrominance.
- Since we use a separate Bézier function to represent each of the different input channels, non-additive color transformations due to channel interdependencies can be easily handled.
- Higher order Bézier functions can be reconstructed from fewer samples than required for a piecewise linear representation.
- Bézier aids elegant handling of non-monotonicity in the color transfer function(Section 3.1).

Non-Linear Reparametrization In this section, we describe the non-linear transformation $(X', Y', Z') = \mathcal{N}(X, Y, Z)$. $\mathcal{N}_l$ is a 3D non-linear function that can be complex and difficult to design. However, note we do not need perfect uniform parametrization, but a function that will yield close to uniform parametrization. Hence, we approximate the function by a channel dependent non-linear function, $\mathcal{N}_l$, resulting from a concatenation of a channel independent 3D linear transformation $\mathcal{L}$, and a channel dependent 1D non-linear function $\mathcal{K}_l$. Figure 3 shows the result of our reparametrization. This results in more accurate Bézier fitting and better interpolation at unsampled points.

3D Linear Function: We propose a 3×3 linear matrix whose columns are given by the XYZ values corresponding to the inputs $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ respectively. $\mathcal{L}$ is the inverse of this matrix. Thus, this is very similar to a standard gamut transformation matrix M^{-1} that allows us to align the XYZ basis with the RGB basis. This transformation yields an intermediate space (X_m, Y_m, Z_m) such that $(X_m, Y_m, Z_m)^T = \mathcal{L}(X, Y, Z)^T$. Following the linear transformation, the primary contribution to X_m, Y_m and Z_m is from r, g, and b respectively.

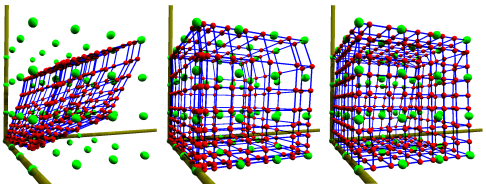

Fig. 3. The samples in XYZ (in red) space and the corresponding control points (in green) generated for B_r before applying $\mathcal{N}_l$ (left), after applying $\mathcal{L}$ (middle), and after applying $\mathcal{K}_l$ (right).

1D Non-Linear Function: Consider function c_r relating r to X_m, i.e. $X_m = c_r(r)$. Similarly we have $Y_m = c_g(g)$ and $Z_m = c_b(b)$. From the measured XYZ values we fit a function of the form $c_l = x^\gamma$ and choose it to be our 1D non-linear function $\mathcal{K}_l$, i.e. $\mathcal{K}_l = c_l^{-1}$.

Handling Non-Monotonicity Let us first consider monotonicity in a 1D function $y = f(x)$ sampled n times and the (x_i, y_i) samples are in increasing order of x_i. Common way to make these samples monotonically increasing is to apply $y_i = max(y_{i-1}, y_i)$, $2 \leq i \leq n$. We expanded this simple idea to three dimensional sampling to make sure that the iso-contour of the sampled outputs are monotonic. Suppose we have the RGB values in the form of (r_i, g_j, b_k) where all of r_i, g_j, and b_ks are in the increasing order and their corresponding XYZ values after application of $\mathcal{F}$ are $(X_m(i,j,k), Y_m(i,j,k), Z_m(i,j,k))$. The following pseudo code makes the iso-parametric curves of this grid monotonic.

for $i = 1 : n$
  for $j = 1 : n$
    for $k = 1 : n$ {
      if $i > 1$ $\{X_m(i,j,k) = max(X_m(i-1,j,k), X_m(i,j,k))\}$
      if $j > 1$ $\{Y_m(i,j,k) = max(Y_m(i,j-1,k), Y_m(i,j,k))\}$
      if $k > 1$ $\{Z_m(i,j,k) = max(Z_m(i,j,k-1), Z_m(i,j,k))\}$ }

3.2 Color Transformation

The color transfer function of a non-ideal device comprises of $\mathcal{F}_c$, $\mathcal{B}_l$, and $\mathcal{N}$. $\mathcal{N}$ comprises of a matrix and inverse channel transfer functions and is hence similar the ideal device parameters. Hence, the nonlinearities are encoded in $\mathcal{F}_c$ and $\mathcal{B}_l$.

$$(r_s, g_s, b_s) \xrightarrow{\mathcal{F}_{c_s}} (X'_s, Y'_s, Z'_s) \xrightarrow{\mathcal{N}_s^{-1}} (X, Y, Z) \xrightarrow{\mathcal{N}_t} (X'_t, Y'_t, Z'_t) \xrightarrow{\mathcal{B}_{l_t}} (r_t, g_t, b_t)$$

Fig. 4. Color Transformation Method

After the Bézier patches are computed, the color transformation from a source RGB color (r_s, g_s, b_s) to the device RGB color (r_t, g_t, b_t) is achieved using the pipeline in Figure 4. (r_s, g_s, b_s) is first converted to the source reparametrized space (X'_s, Y'_s, Z'_s) using $\mathcal{F}_{cs}$. This denotes the desired color in the device independent XYZ color space. To find the target (r_t, g_t, b_t) that produces this color, N_t is applied followed by evaluation of $\mathcal{B}_{lt}$ at the reparametrized (X'_t, Y'_t, Z'_t).

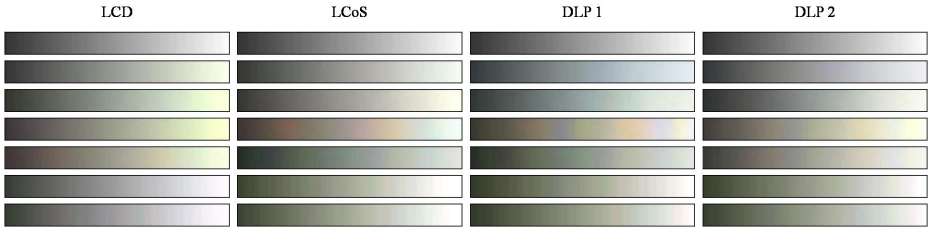

Fig. 5. Comparison of black-to-white gradient reproduction quality using, from top to bottom, target image, our method, method for ideal devices, WCS, Adobe CMM, and linear interpolation (for 9x9x9 and 16x16x16 RGB space samples) in different projection technologies. Note that our method provides the closest match and the smoothest color transitions. Please zoom-in to see the differences.

When matching colors across multiple devices, often the desired color is provided in the XYZ space – say (X_d, Y_d, Z_d). In this situation, achieving a color matching involves evaluating the $\mathcal{B}_l$s of the different devices at the same desired (X_d, Y_d, Z_d) after reparametrization. Hence the input to reproduce (X_d, Y_d, Z_d) is given by $(\mathcal{B}_r(X'_d, Y'_d, Z'_d), \mathcal{B}_g(X'_d, Y'_d, Z'_d), \mathcal{B}_b(X'_d, Y'_d, Z'_d)$.

Note that the focus of our algorithm is to compute a target color when within the device gamut. Out-of-gamut colors can be first mapped to an in-gamut color using gamut mapping techniques [16, 17] before applying our method.

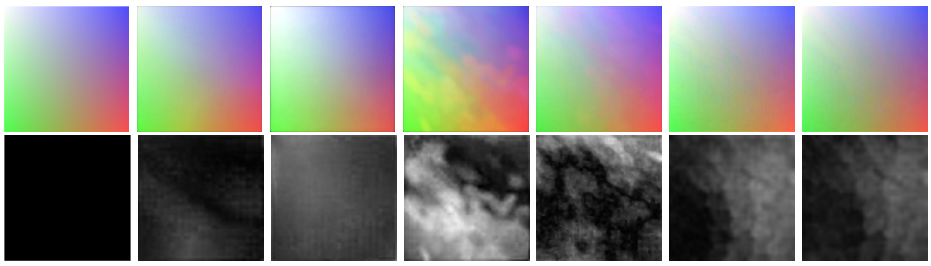

Fig. 6. Comparison of color gradient reproduction quality using different methods – from left to right: target image, our method, ideal devices method, WCS, Adobe CMM, linear interpolation using 9x9x9 and 16x16 samples. The difference of each method from the target image is shown in the second row. Look for color mismatches and artifacts in the top row in the brighter areas. Please zoom-in to see the differences.

Fig. 7. Color uniformity on a display made of 3 heterogeneous projectors (LCD, LCoS and DLP from left to right): Before any color matching (left), after color matching using ideal devices method (middle) and our method (right). Note that the method for ideal devices introduces a color mismatch in the middle projector by changing the color temperature. Our method balances the brightness better, making the left projector of comparable brightness as the other two. Please zoom-in to see the differences.

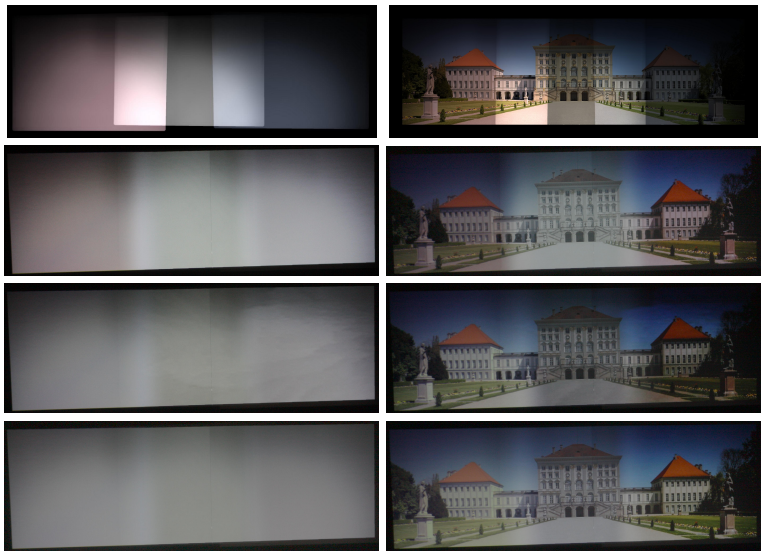

Fig. 8. Comparison of observed color uniformity of a white image and a natural image in a display made of 3 heterogeneous projectors (LCD and two DLPs from left to right). From top to bottom: before color matching, after matching using method for ideal devices, using linear interpolation, and using our method. Zoom in to see banding artifacts in the white image and the severe color temperature change on the right side of the road in the natural image for linear interpolation. Note that the method for ideal device fails to achieve the balancing of the color temperature.

4 Implementation

In our implementation we use a digital SLR camera (Canon Rebel XSi), that has an sRGB color gamut, as measuring device. [18, 9] show that most devices have gamuts that lie within the sRGB gamut and hence our measurements are accurate. We use the RAW images assuring linearity of the camera channel transfer function and no color processing. Hence, to convert the measured colors in the sRGB space to the CIE XYZ space, we apply a standard 3×3 linear transformation. For reconstructing T, we use $k = 9$ (total of $9^3 = 729$ measurements) to sample the input space uniformly. Empirically, this provides good results even for devices with significant deviation from ideal devices – like the ones illustrated in Figure 1. Also, for all devices, a 3D Bézier of degree 3 assures adequate C^2 continuity (details in Section 5.1).

Real-time GPU implementation: The Bézier functions can be stored in a compact manner by storing the $(N+1)^3$ control points for a degree N 3D Bézier (64 control points per channel for our degree 3 Bézier, 192 points in total).

The Bézier is evaluated in runtime to achieve the color correction, implemented in real-time on the GPU using CUDA. We achieve about 70 fps on a 2GHz Xeon workstation with a mid-range GeForce 9600 GT GPU for a XGA (1024 x 768) image.

An alternative to evaluating a Bézier is to compute a 3D LUT for all color values by evaluating the Bézier at all these values which are then indexed by the pixel colors to achieve the color correction. This uses significantly greater storage ($256^3 \times 3$ bytes) but can be implemented on any graphics hardware.

5 Results

We have applied our method to correct colors across many different devices including projectors, cameras, and printers. We have matched colors of multiple devices to a desired image in sRGB space. To evaluate the error between the original (desired or source) images and the corrected image captured by a camera, we compute the pixelwise ΔE color difference (Euclidian distance in the CIE LAB space) between these two images and encode it as a gray image (higher gray values indicate higher deviation from the original image). We also summarize the mean, maximum and the standard deviation of these color differences in Tables 1 and 2. To normalize the brightness of the captured and original image, we scale the captured image by $\frac{M_d}{M_c}$ where M_d and M_c are the mean of the desired and captured image respectively, calculated over all the channels. M_d and M_c provide a measure of the overall brightness of the two images. We use a homography to geometrically align the captured images with the original image since our camera did not show any significant radial distortion [19, 20]. Note that the error reports can be improved by using a more accurate color measuring device, such as colorimeter or spectroradiometer.

We compare our method with three different existing methods. First, we compare with the correction suitable for ideal devices as in Equation 2 achieved by simply applying an inverse matrix multiplication followed by an inverse channel transfer function. Second, we compare with WCS, a commonly used color management system for Windows, described in Section 2. Third, we compare with Adobe CMM, another commonly used commercial color management system. Underlying principles of this system is not available in public domain, however, we can still compare the results of it with that of our method. Finally, we compare with a linear interpolation of the color transfer function as proposed in [9]. For all these methods, we use a sparse sampling of colors, 9^3, as used in our method. However, for the linear interpolation method, we also compare with a much denser sampling, 16^3, as proposed in [9]. Please note that capturing 9^3 and 16^3 images in our experiments with a high-end camera took about 2 hours and 11 hours respectively. For projectors in addition to the longer calibration time heating issues also come into picture when we capture 16^3 images.

To show the effectiveness of our non-linear parametrization (Section 3.1), we show that a version of our method where the reparametrization is not applied yields less accurate results. The advantage of our reparametrization is further emphasized by the reduction of error when it is applied for the linear interpolation method [9].

Note that after combining linear interpolation with our nonlinear reparameterization and with using 6 times more images we achieved statistically similar

Table 1. The average, maximum, and standard deviation of ΔE color differences of the color-matched image from the original image using different methods. For our method (ADICT) and linear interpolation we also mention the number of samples used ($9 \times 9 \times 9 = 9^3$ or $16 \times 16 \times 16 = 16^3$) and if the reparametrization has been used or not (w R and wo R respectively).

Algorithm	LCD			LCoS			DLP 1			DLP 2		
Gray (Fig. 6)												
Our Method(9^3 w R)	2.96	8.22	1.69	3.45	7.59	1.98	4.91	9.08	1.93	5.12	10.32	2.03
Our Method(9^3 wo R)	3.65	10.41	1.76	3.98	8.26	2.12	5.56	13.47	2.01	5.96	14.83	2.46
Ideal Devices Method	5.28	14.86	1.89	4.41	11.37	2.56	5.02	11.70	1.95	10.61	29.48	3.52
WCS	6.29	17.61	1.97	7.76	19.17	3.50	8.68	23.04	2.32	10.25	23.97	3.16
Adobe CMM	4.93	14.77	1.86	6.73	14.24	2.99	6.97	18.63	2.02	9.46	19.28	2.82
Lin Interp.(9^3 wo R)	3.94	11.07	1.84	4.56	10.17	2.15	6.73	13.29	2.08	6.89	15.35	2.52
Lin Interp.(9^3 w R)	3.28	11.21	1.80	3.76	9.32	2.01	5.48	12.86	1.98	5.93	13.47	2.32
Lin Interp.(16^3 wo R)	3.06	9.85	1.76	3.52	8.75	1.95	5.05	11.43	1.96	5.58	11.48	2.13
Lin Interp.(16^3 w R)	2.93	8.75	1.71	3.47	8.50	1.92	4.81	9.93	1.95	5.18	11.07	2.11
Color (Fig. 5)												
Our Method(9^3 w R)	3.79	8.96	1.77	4.14	10.03	2.14	5.84	18.32	2.34	5.43	19.23	2.54
Our Method(9^3 wo R)	4.97	10.53	2.02	5.54	14.12	2.51	7.45	20.25	2.87	7.04	22.14	2.97
Ideal Devices Method	7.65	16.81	2.46	6.33	18.64	2.73	6.83	28.13	2.48	10.32	29.17	3.43
WCS	8.47	17.00	2.42	10.28	21.86	3.76	13.92	30.94	3.38	12.78	24.53	3.23
Adobe CMM	6.29	14.73	2.29	8.67	17.83	3.12	10.86	23.47	3.10	10.77	22.61	3.01
Lin Interp.(9^3 wo R)	5.12	11.04	2.12	5.58	13.79	2.55	8.25	19.64	2.45	7.94	23.12	2.94
Lin Interp.(9^3 w R)	4.48	11.52	2.01	4.86	11.69	2.36	7.10	18.67	2.38	7.12	20.98	2.84
Lin Interp.(16^3 wo R)	3.85	9.04	1.80	4.73	11.02	2.23	6.02	18.98	2.32	5.98	19.45	2.37
Lin Interp.(16^3 w R)	3.67	8.98	1.79	4.53	10.84	2.18	5.92	18.47	2.30	5.48	19.30	2.36

results compared to our method. However, we still see severe local artifacts in the linear interpolation results due to the lack of smoothness constraints. This is more pronounced in the whitish images (Figures 5 and 8).

Matching Different Projectors: We have used our method on three types of projectors: LCD (Epson EMP 74), LCoS (Canon Realis X700) and DLP (Sharp XG-PH50X and InFocus Screenplay 4800). Unlike three-primary LCD and LCoS, DLP projectors use a four-color filter wheel. Thus, they exhibit the greatest devation from an ideal monotonic parallelepiped gamut due to strong contribution from the fourth 'white' primary. We use two desired images in sRGB space: (a) a smooth linear gradient from black to white (Figure 5) and (b) a color gradient image which shows smooth transition of colors from red, green, blue and white (Figure 6). Note that the existing methods show significant deviation from the desired image and also visual artifacts like blotches while our method yields smoother and more accurate colors; the error statistics in Table 1 emphasize this. In particular, the color matching is significantly improved by our method compared to the method for ideal devices when applied to devices that deviate significantly from ideal additive gamut (like the InFocus DLP projector).

The results of any color transformation method is best illustrated when used to match colors across spatially contiguous devices when humans are more sensitive in detecting color differences [1]. We used our method to achieve color matching across displays made of multiple projectors. We built a three-projector display with projectors of different technology and balanced their color using different methods for comparison. The remaining spatial variation of intensity after the color matching is corrected using methods by Majumder and Stevens [21].

In the first setup, we use an LCoS, an LCD and a DLP projector (Figure 7) and compare it with color matching method for ideal devices. In the second setup, we use an LCD projector with two DLP projectors and compare with the method for ideal devices and the linear interpolation technique in [9]. In both cases, existing methods show color mismatches or visual artifacts like blotching and banding while our method shows a seamless result, especially for flat white, the most testing pattern for demonstrating color matching. Note that multi-projector displays are usually never built using projectors of different technologies to avoid the difficult color matching problem. Also, LCD projectors that are close to ideal are the most common choice for multi-projector displays so that method for ideal devices can be used to achieve the color matching. Our result demonstrates that seamless displays made of projectors of different technologies are possible if a sophisticated color management algorithm as ours is applied.

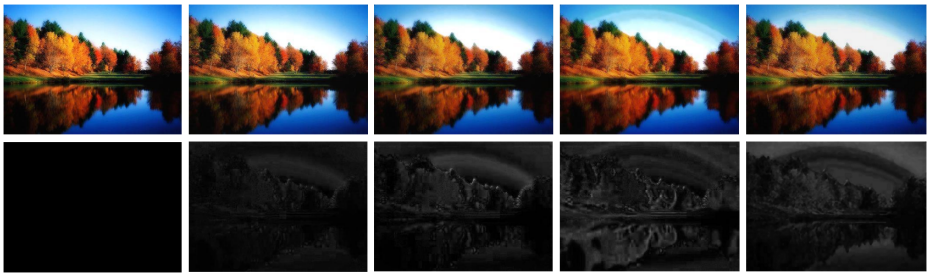

Fig. 10. Comparison of printer-to-projector color matching for a natural image. Difference from input image is shown in the second row of images. From left to right are the printed image, our method, method for ideal devices, WCS, and Adobe CMM. Note the banding artifact and less vibrancy in color due to inaccurate color match in WCS, Adobe CMM and the method for ideal devices when compared to our method. Please zoom in to see the differences.

Fig. 9. Comparison of printer-to-projector color matching for a black-to-white gradient. Note that our method provides the closest match and the smoothest color transitions.

Matching a Projector to a Printer: We demonstrate the use of our method for matching a target projected image to its source printed counterpart. To sample the color transfer function of the printer, we capture with the measurement camera a printed color chart of the 9x9x9 samples. Then apply the different methods to match the color (Figure 4). For this, we use a gray gradient (Figure 9) and a natural image (Figure 10). We find our method to provide the closest match devoid of any banding artifacts. The error statistics on the deviation from the original is summarized in Table 2.

Matching Different Cameras: We use our method to match color across three cameras (Sony DSC-W1, Sony DSC-F707, and Canon SLR 30D) that to-

gether capture a panorama (Figure 11). To find the correspondence between the RGB space of each camera and XYZ space, we capture multiple images using our measurement camera and the camera whose color has to be characterized. Following a homography based registration, this provides us millions of correspondences. We choose the appropriate ones to assure a close to uniform sampling in the RGB space.

Fig. 11. Panorama generated from images captured by three different cameras, when using no color matching (left), inverse transformation for ideal devices (middle) and color matching using our method (right). The image on the left shows severe color mismatch. This is not corrected by the middle one – note the greenish tinge in the white of the floor and ceiling near the center and also lower saturation of the green color of the ping-pong table. These artifacts are completely removed by our method on the right. Please zoom in to see the differences.

5.1 Discussion

Generality: Since our method works directly with the three channel input in which media is usually formatted, it can handle any device irrespective of the actual number of primaries used and the exact method of combining them within the device. Since, our method can be used for both direct and inverse color transformations, it can be used for both capture and display devices alike.

Table 2. The average, maximum, and standard deviation of ΔE color difference of the captured image from the original image of our linear gradient (Fig. 9), in the first row, and natural scene (Fig. 10), in the second row, for projector-to-printer color matching.

Our			Ideal			WCS			Adobe CMM		
2.71	6.71	1.54	3.73	10.22	2.13	6.35	12.90	3.48	5.31	9.97	2.71
2.66	9.41	1.72	4.58	15.16	2.34	5.08	16.96	3.63	4.60	14.89	2.98

Backward Compatibility: When handling devices that are ideal or close to ideal, instead of sampling the output color at all k^3 samples, we can just measure the output color at k values for each channel, i.e. $3k$ measurements. Rest of the samples can be predicted using the additivity assumption. The rest of the our method remains unchanged. Hence, our method is backward compatible to ideal devices, as is demonstrated by the superior results on the near ideal LCD projector (Figure 6 and 5).

Superior Color Management: To illustrate the significantly better results of our method, we choose projectors since they are good examples of commodity

devices with all kinds of anomalies (Figure 1). Almost all different projector technologies (LCD, LCoS, DLP) show channel color non-constancy. LCoS and DLP projectors are severely non-additive in nature. [18] presents extensive studies on projectors that show non-monotonic color responses. Same is shown in [22]. Non-monotonicity is also common in cameras [23]. Unlike linear interpolation that assumes monotonicity and WCS that does not preserve monotonicity, our Bezier based method handles non-monotonicity better. Further, unlike existing methods that can handle only additive gamuts or assure only C^0 continuity while handling non-additive gamuts, our method assures C^2 continuity for both additive and non-additive gamuts. Hence, our method yields superior results that all existing methods consistently.

Degree of the Bezier: We experimented with Beziers of degree up to 6. Cubic Beziers provided a good fitting that was improved marginally by using degree 4. We chose the cubic Bezier for faster GPU implementation. Degrees 5 and 6 showed some visual noise due to over fitting. However, the user can choose the degree that works best for the particular device.

Perceptual Plausibility: Please note that Even though statistically in some of the experiments the non-linear interpolation method with a higher sampling rate achieved similar results to our method still it shows severe local artifacts especially for white as can be seen in Figures 5 and 8. This is due to the fact that our method uses a smooth non-linear interpolation which makes it devoid of these local artifacts. Also please note that these statistical results achieved in combination with our reparameterization and with 6 times more samples.

Sparse Sampling: The sparse sampling for reconstructing the color transfer function is also an additional advantage of our method. Our method requires $9^3 = 729$ samples, an order of magnitude smaller than the $16^3 = 4056$ samples required for the linear interpolations that provide somewhat comparable results. In case of some devices such as projectors we need to capture one image per sample. With a high-end camera it took about 2 hours to capture the 729 images while it takes 11 more than hours for 4056 samples.

6 Conclusion

We have presented a new general method for computing the direct and inverse color transformations for non-ideal devices. This can be extremely useful for addressing color management demands of commodity devices. Our Bézier representation of these functions is general, can be stored compactly, and evaluated in real-time using a GPU. Since our method does not make any assumptions on the nature of the color properties of the device, we have shown that it can be used to match colors across heterogeneous display and capture devices. In the future this work can be used as a foundation to explore color seamlessness algorithms for multi-camera or multi-projector systems.

References

1. Valois, R.L.D., Valois, K.K.D.: Spatial Vision. Oxford University Press (1990)

2. Berns, R., Motta, R., Gorzynski, M.: Crt colorimetry, part i and ii: Theory and practice. Color Research and Application **18** (1992) 299–325
3. Bala, R., Braun, K.: A camera-based method for calibrating projection color displays. 14th Color Imaging Conference (2006)
4. Bastani, B., Ghaffari, R., Funt, B.: Optimal linear rgb-to-xyz mapping for color display calibration. 12th Color Imaging Conference (2004)
5. Heckaman, R.L., Fairchild, M.D., Wyble, D.: The effect of dlp projector white channel on perceptual gamut. 13th Color Imaging Conference (2005)
6. Niven, G., Mooradian, A.: Low cost lasers and laser arrays for projection displays. (2006) 1904–1907
7. Kishimoto, J., Yamaguchi, M., Ohyama, N.: Evaluation of tone mapping for multiband high dynamic range images. ACM SIGGRAPH Talks (2008)
8. Wyble, D.R., Rosen, M.R.: Color management of dlp projectors. 12th Color Imaging Conference (2004)
9. Wallace, G., Chen, H., Li, K.: Color gamut matching for tiled display walls. Immersive Projection Technology Workshop (2003)
10. Tin, S.K.: Color characterization of projectors. US Patent 7148902 (2006)
11. Balasubramanian, R., deQueiroz, R., Eschbach, R.: Gamut mapping to preserve spatial luminance variations. Journal of Image Science and Technology **45** (2001) 436–482
12. Horiuchi, T., Tominaga, S.: Color gamut mapping algorithm for preserving spatial ratios. 16th Color Imaging Conference (2008)
13. Nakauchi, S., Hatanaka, S., Usui, S.: Color gamut mapping based on a perceptual image difference measure. Color Research and Application **24** (1999) 280–290
14. Kimmel, R., Shaked, D., Elad, M., Sobel, I.: Space dependent color gamut mapping: A variational approach. IEEE Transactions on image processing (2005) 796803
15. McCann, J.J.: Lessons learned from mondrian applied to real images and color gamuts. 7th Color Imaging Conference (1999)
16. Montag, E.D., Fairchild, M.D.: Psychophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries. IEEE TIP **6** (1997) 977–989
17. Morovic, J., Ronnier, L.M.: The fundamentals of gamut mapping: a survey. The Journal of Imaging Science and Technology **45** (2001) 283–290
18. Majumder, A., Stevens, R.: Color nonuniformity in projection-based displays: Analysis and solutions. IEEE TVCG **10** (2003)
19. Sukthankar, R., Stockton, R., Mullin, M.: Smarter presentations: Exploiting homography in cameraprojector systems. IEEE ICCV (2001)
20. Raskar, R.: Immersive planar displays using roughly aligned projectors. In: IEEE VR. (1999)
21. Majumder, A., Stevens, R.: Perceptual photometric seamlessness in tiled projection-based displays. ACM TOG **24** (2005)
22. Nayar, S.K., Peri, H., Grossberg, M.D., Belhumeur, P.N.: A projection system with radiometric compensation for screen imperfections. IEEE PROCAMS (2003)
23. Grossberg, M., Nayar, S.: Determining the camera response from images: What is knowable? IEEE PAMI **25** (2003) 1455–1467