

Dense Sampling of 3D Color Transfer Functions using HDR photography

Marcel Heinz and Guido Brunnett
Technische Universität Chemnitz
09111 Chemnitz, Germany

{heilm,brunnett}@informatik.tu-chemnitz.de

Abstract

To apply brightness and color adjustments to projected images, the color transfer function (CTF) of the projector has to be known. We propose a novel approach to determine the CTF using a high sampling density, which is suitable for modern DLP projectors working with color wheels with additional primaries. Our approach is based on the principle of measuring patterns consisting thousands of color samples at once, using a DSLR camera and high dynamic range photography. To ensure high accuracy, additional correction patterns are introduced to compensate for the influence of the dynamic background light caused by displaying the patterns itself. Furthermore, several permutations of the samples in the patterns are captured to address spatial variances of both the projector and the camera. We show that our method achieves comparable accuracy to existing methods, but is one to two orders of magnitude faster. A 64^3 sampling of the CTF can be acquired in a few hours, compared to several weeks that sequential spot measurements would take. Additionally, we demonstrate that a different configuration of our method can be used to capture 17^3 samples extremely fast, indicating the applicability for cases where sparse sampling is sufficient.

1. Introduction

Applying photometric corrections to the output of projectors is a challenging task since adjusting brightness and color requires knowledge of the CTF, which describes the output color produced for each input RGB triplet in some device-independent color space (usually CIE XYZ). Using the inverse CTF, the device-specific RGB value to achieve a desired output can be determined. In the process of building stereoscopic powerwalls, we experimented with different methods for measuring the CTFs. We are using off-the-shelf projectors not specifically designed for such use cases. Moreover, we focus on “ultra short throw” devices to build very space-efficient installations. Such projectors are developed for office use and are typically implemented as single-

chip DLP devices with 6-segment color wheels (RGB + cyan, yellow and white). We found that none of the existing color calibration methods achieved compelling results in such scenarios. This is especially apparent when the projector’s native 3D modes are activated. In that modes, the CTF is altered to incorporate the time needed for the shutter glasses to switch (and often also, for the additional white flash for synchronization with DLP-Link glasses [10]). The color reproduction of these devices is generally quite poor and deviates significantly from standard RGB models. We even observed *discontinuities* in the CTF (see the abrupt use of the cyan segment in fig. 5), while most existing calibration methods assume smooth and continuous functions.

In this paper, we present a novel method to measure the CTF of display devices to address the aforementioned issues. The applicability of existing methods is limited by the measurement time. In practice, at most a couple of hours per projector is tolerable, so that a sparse sampling consisting of a few thousand samples can be taken. To adequately measure the local deviations and discontinuities, a much higher density is desirable. Our method is based upon the fact that DSLR cameras consist of millions of separate sensor elements, and displays consist of millions of independent pixels, so that in principle, a high number of measurements can be taken *in parallel* with a single photo.

2. Related work

Stone characterizes the color transfer of RGB projectors based on two assumptions [8]: *color constancy* (the color primaries’ chromacity coordinates are constant at all intensity levels) and *channel independence* (each primary’s intensity is independently controlled by one of the input channels). In such a scenario, the color transfer can be described by the XYZ coordinates of each primary (at maximum intensity) as well as the black offset, and a linear-combination of three *one-dimensional* intensity transfer functions (ITFs), which can be measured with varying sampling density (Stone uses spectroradiometers and colorimeters). Exact colorimetric measurements are only required for determining the primaries and the black offset, while output intensi-

ties can be measured using standard cameras. Raji et. al use a black-and-white digital video camera [6] and rely on *high dynamic range* (HDR) photography techniques as described by Debevec and Malik [2]. Since projectors typically work with 8 bit color depth, measuring all three ITFs will require at most 768 samples.

Unfortunately, many projectors’ color transfer functions cannot be characterized this way. Especially single-chip DLP devices with a color wheel include additional color primaries. Regarding the RGB input space, the color constancy and channel independence assumptions do not hold any more. Wallace, Chen and Li address this problem by directly measuring the 3D color transfer function of DLP projectors with RGBW color wheels [9]. They use a sparse sampling of 13^3 measurements (with a higher density at lower intensities to accommodate for the nonlinear nature of the function) and linearly interpolate in between them. Sajadi, Lazarov and Majumder suggest using 3D Bézier volumes [7] for interpolation and report that using their method with 9^3 sample points yields comparable results to 16^3 samples with linear interpolation. They further report that high-end DSLR cameras’ color gamuts are wide enough to directly capture the color transfer function of most projectors, so that a 3×3 matrix may be applied to convert the camera’s (RAW) sRGB images to XYZ. Measuring 9^3 samples took about 2 hours with that approach, while 16^3 took 11 hours.

3. Camera-based measurement

To achieve a much higher sampling density, we suggest to divide the set of input color samples into patterns, each of which can be displayed and captured at once.

3.1. Aggregation of camera pixels

For optimal quality and reduction of noise, the number of sensor pixels per color sample should be maximized. Since most digital cameras use color filters in a Bayer pattern [1], the actual number of available measurements per (camera) color channel is further reduced to $\frac{1}{2}$ (G) or $\frac{1}{4}$ (R and B).

We render each color sample as an axis-aligned rectangle in projector image space and assume that a geometric registration between the camera and projector image space has been established, so that each camera pixel can be assigned the ID of a color sample, denoted by the function $m(x, y)$. This mapping is calculated by transforming the four corner points of the camera pixel into projector space. If all four points lie in the same sample, the camera pixel is assigned to that sample, otherwise it is marked invalid.

To compensate for inaccuracies of the geometric registration, unsharpness of the projection or cameras as well as slight variations over time (e.g. due to change in temperature), a weighting factor $w(x, y)$ is applied to each camera pixel, smoothly fading out to the borders of the sample, so that crosstalk between neighboring samples is minimized.

Since we are not interested in reconstructing an image, the expensive interpolation to determine an RGB triplet per pixel can be skipped. Instead, we define a function $h(x, y, z)$ which assigns each pixel (x, y) and raw input channel z a camera color channel¹ and calculate a weighted average of all of the camera pixel values assigned to a single color sample *and* image channel, resulting in single measurement vector $\tilde{\mathbf{q}}$ per sample i , in the camera’s native color space as

$$\tilde{q}_j(i) = \frac{\sum_{(x,y,z) \in V_j(i)} w(x, y) \cdot \hat{q}_z(x, y)}{\sum_{(x,y,z) \in V_j(i)} w(x, y)} \quad (1)$$

where (x, y) denotes the camera pixel location, j the camera color channel, $V_j(i) = \{(x, y, z) \mid m(x, y) = i \wedge h(x, y, z) = j\}$ the set of pixels for sample i and \hat{q}_z the *linearized* sensor value for color channel z .

3.2. Exposure parameters and dynamic range

We build upon the HDR approach by Debevec and Malik and take a series of photos with varying exposure parameters. This is especially relevant when further constraining the range of the sensor values. We define an interval for the *preferred* range, and always try to get exposures for each sample in that range (although we resort to using values outside of that range if no better exposure is possible due to the constraints described below). For our experiments, we chose $[0.03, 0.90]$ as the preferred interval, which limits the dynamic range of a single photo to $30 : 1$. This implies that even for patterns containing just a subset with a quite limited range in color and brightness, several different exposures may be required.

As noted in [2], the aperture should not be changed in the HDR series, because this changes vignetting characteristics of the image and breaks the fundamental HDR assumption that the linearized sensor values at each pixel location of two different exposures are linearly related by the relative factors of the exposure values. Debevec and Malik generally recommend varying only the exposure time. Since modern cameras offer a range of exposure times from fractions of milliseconds to many seconds or even minutes, the achievable dynamic range is very high.

However, for our use case, additional constraints have to be taken into account. The display devices we want to measure might create the image in a time-sequential manner. E.g., taking photos with exposure times less than one revolution of the color wheel will produce meaningless data. As the camera and display are not synchronized to each other,

¹Note that with cameras with Bayer pattern the number of raw input channels is one, so that z is always 0 and h just maps each sensor pixel to a color channel. However, the chosen model can also represent Foveon sensors, where three different values are available per sensor location [3].

there is no guarantee that an integral number of revolutions or frames is captured, so that the exposure time should be chosen at least an order of magnitude higher than the frame time to minimize the error introduced by incorporating an incomplete frame. Additionally, photos can be repeated with the same exposure parameters to further cancel out this effect. We suggest selecting the number of photos per exposure setting as a function of the exposure time Δt as

$$\text{cnt}(\Delta t) = \max\left(\frac{n \cdot \text{frametime}}{\Delta t}, k\right) \quad (2)$$

so that in total at least n full frames are captured and at least k photos are taken. For devices which continuously display the complete image, $\text{frametime} = 0$ can be assumed. From empirical experiments, we recommend $n = 100$ and $k = 2$ to always repeat each measurement at least once, and $\Delta t > 30\text{frametime}$.

On the other hand, the upper limit for Δt should not be chosen too high, because doing so would increase the overall duration and would also introduce additional noise. We limit ourself to the available exposure time steps offered by the camera and avoid long-term ‘‘bulb’’ exposures, so a typical upper limit is $\Delta t \leq 30s$. We found that with those constraints, the resulting dynamic range is too low to capture the full range of the projector outputs. Therefore, we also vary the *sensitivity* (ISO value) S during the HDR series. Modifying the ISO value does not change the light reaching the sensor, but it controls the amplification *before* the digitizing operation, so it reduces quantization noise. In general, the sensitivity should be kept as low as possible and only increased for very dark contents. However, modern cameras deal quite well with moderate ISO values < 1000 , so that slightly increasing S early can be beneficial. To incorporate the changed sensitivity value S into the HDR calculations, we extend the concept of the exposure value to a generalized exposure factor $b = \frac{A^2}{\Delta t \cdot S}$.

The exposures are selected on a predefined path in the 2D parameter space defined by $(\Delta t, S)$. This path represents series of points, since the parameters can only be chosen in discrete steps. Each point is characterized by the corresponding b value. Tuning this path allows to make a trade-off between speed and quality by earlier/later resorting to higher ISO values.

The exposure series must capture the full dynamic range of each native camera channel. An iterative algorithm is used for that. An initial guess for the exposure is made by calculating the average of the input color samples and predicting the display’s output assuming a simple RGB model. Then, the point on the exposure path which maps the expected value closest to the center of the measurement range is selected and the cnt photos are taken. An interval $[b_{\min}, b_{\max}]$ is used to keep track of the already covered exposure range. As long as overexposed (underexposed) samples

are present, a new exposure is searched along the path to decrease b_{\min} (increase b_{\max}), and the photos taken accordingly. The distance to walk on the path is controlled by a user parameter. This method is repeated separately for each color channel. However, the intervals of previous channels might be re-used if the exposure ranges are not totally disjoint. To do so, the number of under- and overexposed samples per color channel is stored and updated for each interval border. This is always done for all color channels, not only currently processed one. Before a new photo is taken, the intervals of previous channels are searched for that exposure factor. If it is already covered, the current interval and the previous one can be united.

The standard HDR approach assumes that the exposure value of each photo is exactly known, so that for two exposures α, β the equation $\frac{\hat{q}_\alpha}{\hat{q}_\beta} = \frac{b_\beta}{b_\alpha} = f$ holds for every sensor location (correctly exposed in both photos). We modify this approach by not calculating f from the exposure parameters, but by relying on the presence of many samples correctly exposed in both photos to find an optimized factor f' using a least-squares minimification strategy. This corrects for slight variations in the data as well as uncertainties in the exposure parameters.² Since we work on the aggregated sensor elements and not directly on the millions of raw camera pixels, this optimization step is not very expensive.

Reconstructing the HDR data starts by selecting the exposure with the most not over- or underexposed samples. Then, iteratively, of the remaining photos, the one with the most overlapping samples is added. When adding a new image and determining f' , the data accumulated up to this point is used. The values are normalized so that for the accumulated image, the exposure factor $b = 1$ is assumed. This results in a measurement value \hat{s}_i per sample, which is independent of the exposure parameters.

3.3. Emulating spot measurements

In the existing approaches, cameras are primarily used as replacements for spot measurements with colorimeters or spectrophotometers. Majumder found that projectors mostly vary in luminance, while chrominance and especially the nonlinear characteristics of the ITFs remain spatially constant [4], so an arbitrary location can be chosen. But when measuring many samples at different locations, the individual measurements must be related to each other, so that the spatial variations cannot be neglected any more.

A naive approach would be capturing the output of a uniform color across the whole display and determining a scale factor for each camera pixel. This could be seen as a modification of Majumder and Steven’s LAM method [5], just

²Debevec and Malik did report that the actual exposure times of digital cameras deviated from the displayed ones, with the actual ones strictly organized in an $\frac{1}{2}$ or $\frac{1}{3}$ f-stop raster. We already corrected for this effect, but the actual ISO values are also not exactly known.

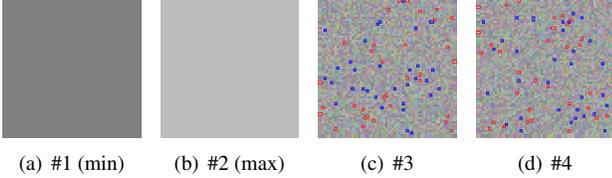


Figure 1. Sub-patterns for pattern #43 out of a set of 64. The minimum- and maximum reference pattern and two permutations are shown. The reference samples are marked with red (max) and blue (min) borders.

applied to the camera image (containing the *composition* of the projector’s and the camera’s spatial luminance variations) instead of the projector image. However, we found that this strategy does not work well in practice. It turns out that the background light cannot be neglected. The irradiance E at some sensor pixel can be characterized as $E = E_b + E_x$, where E_b is caused by background light (both of the room as well as the black offset of the display) and E_x represents the actual output of the projector which is to be measured. Assuming that the measurement take place in a darkened room and capturing a map of E_b is not sufficient, though. Displaying the pattern itself illuminates the whole room, and the light gets diffusely reflected back onto the display. Conceptually, this can be seen as a *dynamic* background light, so that the actual irradiance incident at the sensor is $E_b + E_d(E_x) + E_x$. Assuming that the reflective characteristics of the room stay constant, the dynamic background light might be characterized as $E_d(E_x) = \tau E_x$. This does not pose a problem for traditional spot measurements, because after correction for the static black level, the measured irradiance is $(1 + \tau)E_x$, so this only changes the already arbitrary scale factor. But in our use case, E_x varies per sample, but E_d will be a function of all samples of the pattern, so that the samples influence each other.

To address this issue, we are capturing correction data specifically for each pattern. Two reference samples are created per pattern, one contains the minimum (input) intensity (per channel), the other the maximum. For each pattern, a set of *sub-patterns* is created. Each sub-pattern is measured with the method described in the previous sections. The first sub-pattern consists only of recurrences of the minimum reference sample, the second one uses the maximum reference sample. Finally, θ sub-patterns consisting of the actual samples are created. To each of these color sub-patterns, n recurrences of the minimum and maximum samples are added. Furthermore, the sample positions in each sub-pattern will be *permuted* in a pseudo-random manner, so that for each color sample, θ measurements will be made at (very likely) different locations of the display surface. Using more than one permutation provides more data points per sample, so that noise can be further reduced. It also mitigates the influence of spatially varying color re-

production of the projector.

Furthermore, the sub-patterns are enlarged by duplicating the border samples. Those additional fields are ignored during the measurement and are added to ensure a similar neighborhood for all samples in the pattern. Fig. 1 illustrates the layout of the sub-patterns which are actually displayed during the measurement process.

Let i be the index of the sample, j the index of the sub-pattern (1 and 2 denote the min/max sub-pattern), $l_j(i)$ the location of sample i in sub-pattern j and $\dot{s}_j(l)$ the measured value of sub-pattern j at the location l . We also capture the static background $\dot{s}^{(0)}(l)$ when displaying black to calculate corrected values $\tilde{s}_j(l) = \dot{s}_j(l) - \dot{s}^{(0)}(l)$. To reconstruct a location-independent value for sample i , a relative factor is determined as

$$r_{i,j} = \frac{\tilde{s}_j(l_j(i)) - (\alpha_{1,j}\tilde{s}_1(l_j(i)) + \beta_{1,j})}{\alpha_{2,j}\tilde{s}_2(l_j(i)) + \beta_{2,j} - (\alpha_{1,j}\tilde{s}_1(l_j(i)) + \beta_{1,j})} \quad (3)$$

The coefficients α and β allow estimating the value that the reference sample would have had at the sample’s location in sub-pattern j . These coefficients are found by a least-squares optimization using the reference samples of the sub-pattern and the corresponding values of the reference patterns. In other words, we “project” the surface of the reference sub-patterns into the color sub-patterns using a linear model $f(x) = \alpha x + \beta$ and determine the relative position of each sample in-between those surfaces. Note that r is not limited to the range $[0, 1]$, because the min and max samples were chosen with respect to the input color space, and do not take the actual properties of the display’s CTF into account.

To emulate the spot measurement, an arbitrary sample location l_{ref} is chosen. The data is *reinterpreted* for the new sample location as

$$s'_{i,j} = r_{i,j}(\tilde{s}_2(l_{\text{ref}}) - \tilde{s}_1(l_{\text{ref}})) + \tilde{s}_1(l_{\text{ref}}) \quad (4)$$

using the uncorrected reference sub-patterns, which provide a global frame of reference for the measurement data matching the traditional spot-measurement methods. In figure 2, the process of the reconstructing these position-independent values from the sub-patterns is illustrated. The final value s_i for each sample is calculated as the average of the $s'_{i,j}$ for $j = 3, \dots, \theta + 2$. The method is applied to each color channel separately, so that for each input color sample, the vector \mathbf{s}_i representing the display’s output in the camera’s native color space is determined.

3.4. Partitioning of the color samples

The gamut of the a display is defined by its outputs for all possible RGB input combinations. The sampling density is controlled by the step size in the input color range. For example, when using a step size of 4, the resulting number

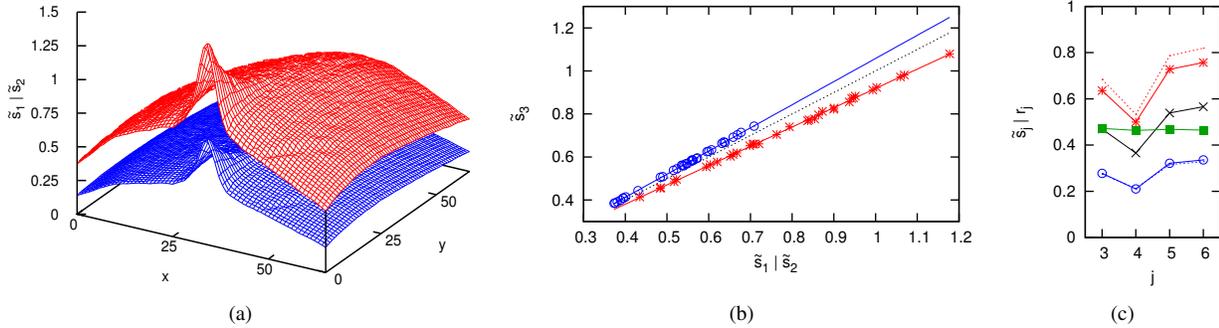


Figure 2. Measurement of the pattern from fig. 1. Only data of the green channel is shown for illustration. In all graphs, the minimum reference samples are shown blue (circle symbol), maximum reference samples in red (asterisk symbol), color samples in black (cross symbol). (a) The reference sub-patterns. The ultra-short-throw projector uses a strong off-axis projection so that the hot spot is at the border. (b) Fitting the reference sub-pattern into the color sub-pattern using the reference samples. The dashed line represents the uncorrected case ($y = x$). (c) Relative interpretation of a single color sample in 4 different permutations (green line, square symbol). The dashed lines represent the uncorrected min- and max-samples for that location. The lines between the data points were only added to highlight the proportions, they bear no direct meaning.

of color samples will be 64^3 or 65^3 (with the highest intensity clamped to the maximum of 255). Since the number of samples is much higher than what can be captured with a single pattern, the samples are partitioned into several patterns.

To improve overall speed, the dynamic range of each pattern should be minimized. Furthermore, samples in a pattern should not differ too much in color and brightness to minimize crosstalk between the virtual sensor elements, and to ensure that the dynamic background light created by the pattern is uniform and does not deviate much from the individual samples' colors.

Although the display might significantly deviate from an ideal RGB model, that is still valid as a rough approximation for the device's CTF, because otherwise the image produced for typical RGB image content would be not tolerable. Therefore, it appears reasonable to partition the RGB cube in input color space into axis-aligned sub-cubes. Within each sub-cube, the dynamic range is limited, and the color and brightness of the samples will appear similar.

We recommend choosing the number of samples per pattern to be at least as high as the number of samples per sub-cube (otherwise, the sub-cube is splitted into several patterns). Using a step size of 4 requires 4096 to 4913 (for the 65^3 case) samples per pattern (not including the reference samples and the duplicated border). In a typical setup with a 12 megapixel camera, this results in 1000 to 2000 camera pixels (with non-zero weight) per sample. If a pattern provides more sample locations than samples present in the sub-cube, the actual samples are repeated until the pattern is filled. This appears preferable to increasing the sample size, since more distinct locations across the whole image are captured at once.

4. Post-processing and LUT generation

To use the data for color transformations, a 3D LUT mapping each input color sample to the projector's response s is built. First, the data is transformed into the projector's color space, defined by the red, green and blue output and the black point. This basis is modified to ensure that no negative sample values arise. Then, a *pseudo-linearization* operation is applied by assuming an ideal RGB model and finding the best fitting gamma value using the Levenberg-Marquardt nonlinear least-squares optimization algorithm. The data is gamma-corrected using the inverse of the found gamma value. A 3D gauss filter is applied to further reduce measurement noise. The rationale for filtering the data in this space is that for an ideal RGB projector and perfect measurements, the transformed LUT would represent the identity mapping and the sample points would lie on an uniform grid. Applying a gauss filter would not change the data at all, contrary to applying it before the linearization. The LUT transformations up to this point are illustrated in fig. 3.

After the data is filtered, the missing samples are interpolated. For each cell formed by the $2 \times 2 \times 2$ neighborhood, a piecewise linear tetrahedron interpolation is performed. Since this interpolation is applied in the pseudo-linearized space as well, this effectively results in a gamma-corrected interpolation.

As the last step, the LUT is transformed back by reversing the pseudo-linearization. However, the original basis is not restored, but a new basis is chosen to convert the data from the camera's native color space into CIE XYZ using the camera's 3×3 color matrix.

The inversion of the CTF is implemented by finding the nearest neighbor to a target value in the point cloud represented by the forward LUT, using CIE Lab color space

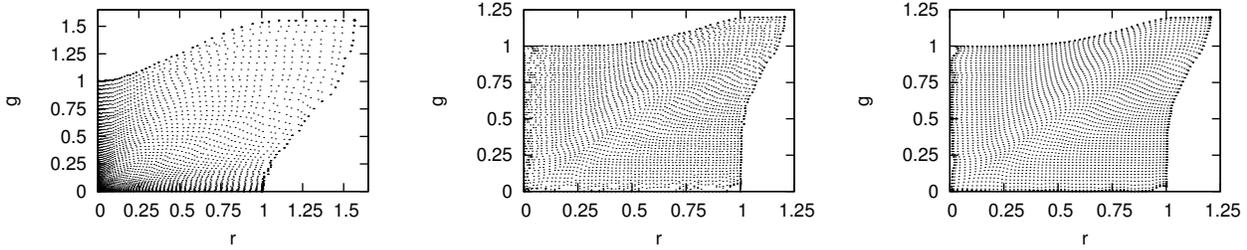


Figure 3. Postprocessing of the LUT. For better illustration, only one slice (blue= 0) is shown in 2D. Left: LUT in device-specific basis, middle: after pseudo-linearization, right: after filtering. Values > 1 represent points which lie outside of the gamut formed by the RGB basis of the device. In this case, the extrusion of the gamut is the result of a separate yellow segment on the color wheel of a DLP projector.

to minimize the ΔE color distances. To speed up the calculations, a k-d-tree is used for the neighborhood queries. For interactive use, the inverse CTF is pre-calculated as another LUT as follows: An enclosing parallelepiped in XYZ space is sampled non-uniformly (according to the gamma value), and for each sample, the closest point of the forward LUT is determined and the corresponding input color value is stored in the inverse LUT. This does not assume monotonous CTFs, but may lead to discontinuities in the inverse function in this case, so that the LUT should only be accessed with nearest point sampling. Hence, the size of the inverse LUT should be chosen high enough; we recommend 384^3 .

5. Experimental results

For our experiments, we used a Nikon D3s high-end DSLR and a XRite i1pro spectrophotometer (abbreviated as *SP* in the following) as reference. Our target was measuring the CTF of a Sanyo PDG DWL-2500 projector with enabled 3D mode.³ We used two different setups of our measurement method. The dense configuration represents 64^3 samples in 64 patterns, and the fast configuration 17^3 samples in 8 patterns. The number of samples per pattern was the same for both. In the fast setup, the samples were repeated at least 7 times per pattern. In either case, 64 additional reference samples were added per pattern. We used ΔE in CIE Lab color space as metric for the quality of the results. For conversions between XYZ and Lab, we used the projector’s native white point as Lab reference point.

In a first experiment, we determined the influence of the parameter θ on quality and measurement time. We selected 2 patterns from the dense set. We chose the first pattern (containing the darkest samples in the input range $[0, 60]$) and #43. Furthermore, we measured 8^3 samples from pattern #1 and all 16^3 samples from pattern #43 with the SP.⁴

³The mode is called “Nvidia 3D Vision”, but it only differs from mode “On” in that DLP-Link is deactivated.

⁴The lower number of reference samples for the first pattern is due to the fact that the SP needs a higher exposure time for dark samples too, which took too long for all 4096 samples.

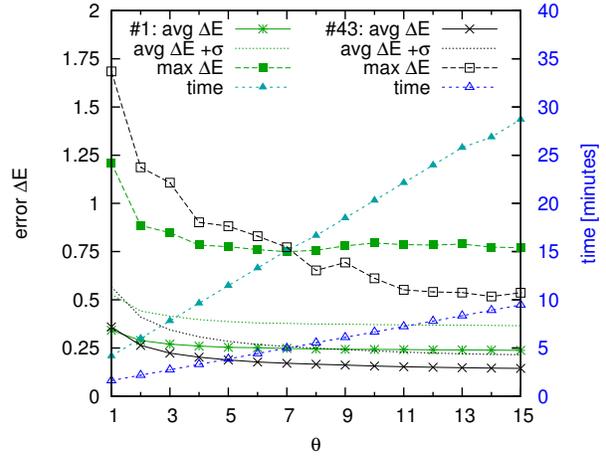


Figure 4. Accuracy and measurement time as a function of θ .

For this experiment, we wanted to minimize the influence of the color matrix of the camera and only focus on the error introduced by our measurement, and by nonlinear distortions in the camera’s color-space. To do so, we used an optimized 3×3 matrix to convert the raw data to XYZ. This was obtained by minimizing the residuals between the corresponding camera and SP data (resulting in a slightly different matrix for each pattern).

In fig. 4, the ΔE error is shown for varying values of θ . The results indicate that using multiple permutations can improve the quality considerably, but more than 5 permutations do not reduce the average error significantly any more. Only the maximum error can be slightly improved. However, the time requirements increase linearly with θ , so that 2 to 5 permutations seems a reasonable range.

In another experiment, we measured the CTF completely (see fig. 5). We compared our results to a 17^3 sampling taken with the SP. The results are shown in table 1. Since the color matrices for most DSLRs are not openly documented, we extracted the matrix for our camera from the open source utility *dcraw*. However, the results achieved with this matrix are not very good. It is not clear how this

	$\Delta E M_{\text{dcraw}}$			$\Delta E M_{\text{opt}}$			speed		
	\emptyset	σ	max	\emptyset	σ	max	time	$\frac{\text{smp}}{\text{sec}}$	
f1	7.89	3.84	18.73	0.67	0.41	3.99	19.1	4.29	
f2	7.87	3.84	18.28	0.64	0.38	3.52	24.4	3.36	
f3	7.87	3.84	18.10	0.61	0.37	3.46	29.9	2.74	
f4	7.87	3.84	18.02	0.61	0.36	3.49	35.6	2.30	
d1	8.09	4.04	20.02	0.59	0.39	3.59	132.6	32.95	
d2	8.08	4.03	20.13	0.53	0.35	3.27	178.8	24.43	
d3	8.07	4.03	20.78	0.51	0.36	3.09	223.5	19.55	
d4	8.07	4.03	20.02	0.50	0.35	3.29	273.7	15.96	
							SP (9^3)	57.2	0.21
							SP (16^3)	307.2	0.23
							SP (17^3)	362.7	0.23

Table 1. Quality (average, standard deviation and maximum ΔE), time consumption and average speed (samples per seconds) of different configurations (d for dense 64^3 and f for fast 17^3 , each for $\theta = 1 \dots 4$) of our method.

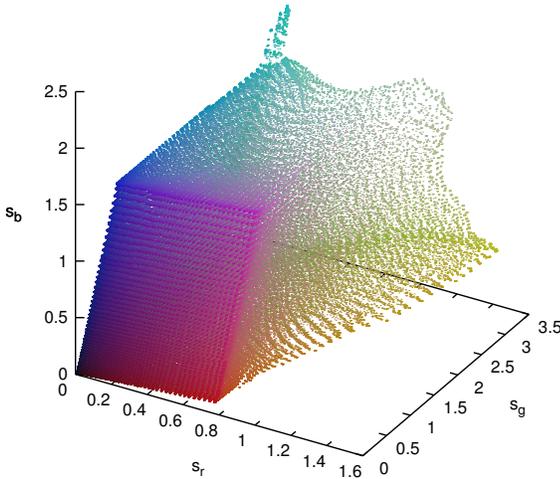


Figure 5. Measured color gamut of the Sanyo PDG-DWL 2500 projector with 64^3 samples in the camera’s native color space.

matrix was created and how good it is in general. It is probably optimized for a wide range of different spectral stimuli. However, the gamut of the projector is quite restricted and the spectral power distributions it is able to produce are just linear combinations of 6 primaries (and ultimately created by one lamp). So it should be possible to create an optimized matrix using just the corresponding measurements between the camera and the SP. This step can be seen as a calibration for the camera itself. It must be noted that this calibration data needs to be acquired only once per camera and can be reused for repeated measurements of the same or similar projectors.

Using M_{opt} confirms that the high ΔE errors observed so far were primarily caused by the linear color space transformation. The maximum error could be reduced from 20 to 4, the average error from 8 to 0.5 – 0.7. In other words, for

method	$\Delta E \text{ abs}$			$\Delta E \text{ rel}$		
	\emptyset	σ	max	\emptyset	σ	max
linear 9^3	1,55	1,15	9,98	1,90	1,35	14,41
linear 17^3	0,99	0,77	4,92	0,94	0,67	6,53
Gamma 9^3	1,90	1,14	10,81	1,98	1,41	14,42
Gamma 17^3	1,19	0,61	6,21	0,90	0,62	6,55
f2 M_{dcraw}	7,19	3,01	20,08	0,95	0,70	7,38
d4 M_{dcraw}	5,93	3,75	18,19	0,66	0,51	3,78
d1 M_{opt}	0,93	0,58	5,78	0,71	0,60	8,78
d4 M_{opt}	0,61	0,48	3,48	0,65	0,51	3,76
hybrid	0,65	0,38	3,13	0,49	0,33	3,81

Table 2. Accuracy of the inverse CTF for 825 XYZ target colors comparing different methods. The relative and absolute error metrics are explained in the text.

most samples, the error introduced by our method is below the perception threshold of a human. The maximum error is still in the noticeable range, though.

It can be seen that in the fast configuration, the errors are somewhat higher than when using dense sampling. This might seem counter-intuitive at first, because in the fast setup, the number of raw measurements per sample is 7 times as high. However, since only a $2 \times 2 \times 2$ subdivision was used, the variations in luminance and color inside each pattern was much higher, which lead to a higher dynamic range and more background light per pattern.⁵

The effect of θ appears smaller than our first experiment suggested. Working with 4 permutations decreased the average error only by 15%. We think that the nonlinear distortions of the camera color space might superpose the actual measurement noise addressed by repetitions. In the fast setup, the effect of θ is even smaller. This was to be expected, because of the presence of repeated samples in each pattern.

Our method is orders of magnitude faster than spot measurements. With our fast mode, we are able to gather 17^3 samples in half an hour, which is an improvement of factor 10 over previous methods. In the dense configuration, 1971 photos were taken in 4.5 hours ($\theta = 4$), so an improvement of factor 70 is achieved (factor 143 for $\theta = 1$).

To evaluate our method in a real-world scenario, we did another experiment to test the inverted CTFs. We selected 35 colors, distributed roughly uniform over the gamut of the projector in the CIE xy chromacity plane (7 of these representing the corners of the input RGB cube). For each color, we chose 25 luminance steps uniformly distributed from black to maximum output intensity. We transformed the 875 samples to RGB input colors for the projector using the inverse CTF and measured the projector’s actual re-

⁵This is also the reason why we did not capture all 17^3 samples in just one pattern. In that case, the dynamic background light created by the pattern is so high that the dark color samples cannot be measured with reasonable precision.

sponse with the SP. We used two different metrics for our evaluation: the absolute ΔE error between the actual and the requested output, and a relative ΔE error. For the latter, the achieved xy coordinates for each color at maximum intensity are chosen as the reference coordinates for all luminance steps of that color. This metric cancels out the effects of linear color transformations. It is motivated by our use case of photometric calibration for segmented displays, where we want to achieve uniform colors across all segments and need to be able to independently control the luminance, but do not necessarily need absolute color calibration exactly reproducing a specified XYZ value.

We compared different variants of our method to linear and gamma-corrected interpolation directly operating on SP data with 9^3 and 17^3 samples. The results (see table 2) show that 9^3 samples are far too few for this projector. We consider the relative error metric first. Using the *fast* setup, our method is comparable in quality to the SP, the average error and standard deviation is almost identical, only the maximum observed error is 10% higher. Using the dense configuration, the average error can be reduced by 30% compared to the best result achieved with the SP alone, and the maximum error can be reduced from 6.5 to 3.8. Using only one permutation cannot be recommended, since this leads to a noticeable higher maximum error. We also tested a *hybrid* method by combining the SP data with the camera samples. The results show that this further improves the average error but it also increases the measurement time to 9.5 hours. In the absolute metric, variants using M_{dcrav} result in high errors, as already expected. When using M_{opt} , the results are very similar to the relative metric. This shows that our method is eligible for absolute color calibration, as long as a calibrated camera color matrix is available.

6. Conclusions and future work

In this paper, we proposed a new method to measure the CTF of projectors. We address the characteristics of the CTFs of modern DLP projectors by using a very high sampling density. We have shown that our method is dramatically faster than sequential spot measurements, without sacrificing the measurement quality. For the first time, a 64^3 sampling of a CTF can be acquired in a reasonably time frame.

Our method can be used without a colorimeter or spectrophotometer at all, as the results with the relative ΔE metric show. In that case, instead of using CIE XYZ as device-independent color space, the camera's native color space can directly be used as the reference. This is especially useful for multi-projector display systems, where photometric uniformity is desired.

If the goal is absolute color calibration, the camera cannot replace a colorimeter or spectrophotometer. However, an optimized camera color matrix can be created, which can

be re-used for recurring measurements of the same or similar projectors. Furthermore, a hybrid approach might be used, where the high-precision, sparsely sampled SP measurements are enriched by the densely sampled camera data.

Although our method was specifically developed for dense sampling, our experiments with the *fast* configuration indicate that it can also be applied to cases where sparse sampling is sufficient. Therefore, our method can also be used in conjunction with previous approaches, for example ADICT [7].

Our method could be extended to work with multiple cameras at once. We imagine a very interesting configuration consisting of using several *black-and-white* cameras and external color filters, so that each camera could be assigned to a color channel. This would allow to further speed up the measurement, as well as using more than 3 channels, possibly improving the accuracy. It might be possible to transfer this approach to inexpensive industry cameras.

References

- [1] B. Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.
- [2] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [3] A. El Gamal. Trends in CMOS image sensor technology and design. In *Electron Devices Meeting, 2002. IEDM'02. International*, pages 805–808. IEEE, 2002.
- [4] A. Majumder. Properties of color variation across a multi-projector display. In *Proceedings of SID Eurodisplay*, 2002.
- [5] A. Majumder and R. Stevens. LAM: luminance attenuation map for photometric uniformity in projection based displays. In *Proceedings of the ACM symposium on Virtual reality software and technology, VRST '02*, pages 147–154, New York, NY, USA, 2002. ACM.
- [6] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. Pixelflex2: A comprehensive, automatic, casually-aligned multi-projector display. In *In Proc. IEEE International Workshop on Projector-Camera Systems*, 2003.
- [7] B. Sajadi, M. Lazarov, and A. Majumder. ADICT: accurate direct and inverse color transformation. In *Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10*, pages 72–86, Berlin, Heidelberg, 2010. Springer-Verlag.
- [8] M. C. Stone. Color and brightness appearance issues in tiled displays. *IEEE Comput. Graph. Appl.*, 21(5):58–66, Sept. 2001.
- [9] G. Wallace, H. Chen, and K. Li. Color gamut matching for tiled display walls. In *Proceedings of the workshop on Virtual environments 2003, EGVE '03*, pages 293–302, New York, NY, USA, 2003. ACM.
- [10] A. Woods and J. Helliwell. A Survey of 3D Sync IR Protocols. whitepaper, Curtin University, Perth, Australia, 2011.