

# Robust, Error-Tolerant Photometric Projector Compensation

Anselm Grundhöfer, and Daisuke Iwai

**Abstract**—We propose a novel error tolerant optimization approach to generate a high-quality photometric compensated projection. The application of a non-linear color mapping function does not require radiometric pre-calibration of cameras or projectors. This characteristic improves the compensation quality compared to related linear methods if this approach is used with devices that apply complex color processing, such as single-chip DLP projectors. Our approach consists of a sparse sampling of the projector’s color gamut and non-linear scattered data interpolation to generate the per-pixel mapping from the projector to camera colors in real-time. To avoid out-of-gamut artifacts, the input image’s luminance is automatically adjusted locally in an optional off-line optimization step that maximizes the achievable contrast while preserving smooth input gradients without significant clipping errors. To minimize the appearance of color artifacts at high-frequency reflectance changes of the surface due to usually unavoidable slight projector vibrations and movement (drift), we show that a drift measurement and analysis step, when combined with per-pixel compensation image optimization, significantly decreases the visibility of such artifacts.

**Index Terms**—Projector-camera systems, photometric compensation.

## I. INTRODUCTION

PROJECTORS today are used in various application fields, such as home cinema, entertainment, cultural heritage, and augmented reality. Several hardware and software tools are available to guide the user on how to set up a high-quality projection system, even when dealing with complex geometry and color surfaces. Often, cameras are used to ease the calibration process with software that analyzes the surface and calculates an undistorted projection image. In addition to these well-established geometric correction tools, photometric compensation (PC) algorithms have been introduced within the last decade, that generate a modified projection image that, when projected on a colored or even textured surface, reproduces the desired color with impressive quality.

The preparation, however, still is a laborious process. The available algorithms all require the devices to be, at least partially, radiometrically calibrated as well as carefully set up. This can be a cumbersome, time-consuming, and error-prone process, reducing the system’s flexibility. This approach also limits the compensation quality on devices such as certain DLP projectors, which offer complex, non-monotonic color processing algorithms [1], [2], and additional primaries as well as transparent components in their color wheels, since this black box image processing is not accurately described with standard radiometric color calibration methods.

Furthermore, even after carefully setting up the projection, slight misregistrations resulting from temperature changes, unsteady projector mounts, or surfaces, as well as other external influences, such as construction or small earthquakes, will severely reduce the image quality especially in long-term installations, since those misregistrations will lead to a boosting of color errors in high-contrast surface areas where the compensation color becomes strikingly evident.

### A. Motivation

The main motivation of this work is to overcome the limitations of current PC methods so that they can be easily deployed even by non-experts. This includes the development of an algorithm that does not depend on radiometric pre-calibration of the camera and projector. In addition, the method should analyze potential projector motion and generate a visually optimal compensation image even in slight misalignments.

To solve these problems, the proposed method consists of a camera-based analysis of the surface’s reflectance properties and the temporal drift behavior of the projector. First, this method applies a scattered data interpolation approach to model the color mapping of projector to camera pixels in a non-linear manner. Therefore, thin plate splines (TPS) [3] are applied to compute the mapping that, in addition to relatively low computational complexity using radial basis functions (RBF), guarantee an optimally smooth transition between the captured color samples. Second, before the projection image is computed, the input image is automatically scaled in an optimization step that maximizes the luminance and contrast while avoiding significant clipping errors and preserving smooth input gradients. Third, a drift analysis and perceptual error minimization are carried out to avoid the appearance of visually disturbing hard color edges due to misaligned projections.

## II. BACKGROUND AND RELATED WORK

In this Section, we will give a quick introduction to PC and summarize the existing methods in the following sections. A more thorough overview of this topic can be found in Bimber’s *et al.* [4] state-of-the-art-report, for example.

The main purpose of photometric compensated projections is the neutralization of non-perfectly white or textured surface reflectance. This is accomplished by using a projector-camera system to evaluate the reflectance properties for each pixel and to calculate colors that, if projected on the surface pigments, modulate to the expected intensity when captured by the camera. Therefore, a two-dimensional geometric calibration is required to map projector to camera pixels, which is usually

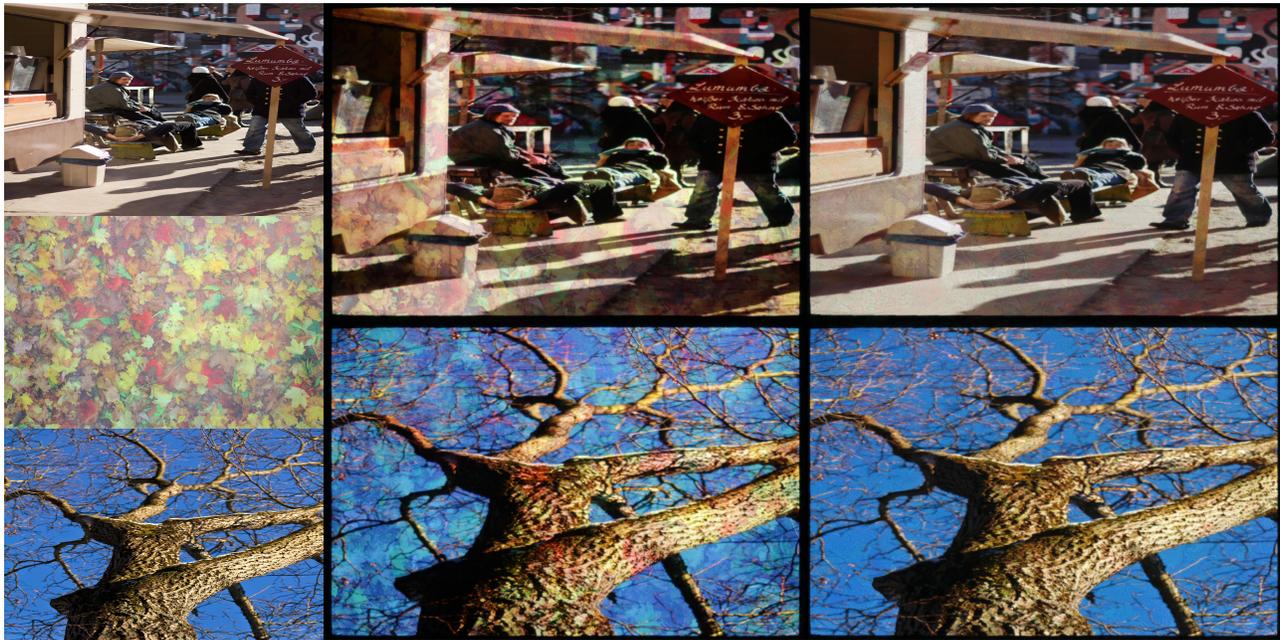


Fig. 1: Results of the proposed method while projecting on the print of colorful leaves shown on the left. The uncompensated projection is shown in the center, while the right shows how our proposed compensation method matches the input images shown on the left.

carried out by projecting a series of well-defined structured light patterns, such as binary gray codes (see [5] for a detailed discussion). Note that this enables accurate mapping for all types of diffuse surfaces, including non-planar ones. Depending on the method, several patterns are also projected to estimate the color mapping needed to compute the compensation images. To our knowledge, all existing methods require the projector or camera to be at least partially radiometrically calibrated to linearly model the light modulation.

#### A. Simple Photometric Compensation Methods

The oldest algorithms focused on modeling the compensation without taking the input image content into account. They varied widely in complexity and prerequisites. In Bimber *et al.* [6], a multi-projector approach was presented in which the devices were assumed to be linearized and their *RGB* color channels to be completely independent. Although this approach compensates for environmental illumination, most real-world setups require a model of the overlapping color channels to achieve high-quality compensation. In Nayar *et al.* [1], a  $3 \times 3$  matrix was used to model color channel mixing that generated more accurate compensation. Yoshida *et al.* [7] extended this approach to a  $3 \times 4$  matrix, which also considered uncontrollable, ambient illumination. [8] showed that the color mixing matrix can also be separated from the spatially varying surface reflection, and thus, the matrix must be stored only once for the setup. An extended approach that compensates global, complex illumination effects, such as caustics, refractions, and scattering was presented in [9] based on the idea of measuring the full light transport between the projector and the camera and inverting it to compute the compensation image. This approach requires, in addition to the device linearization, up to several hours of scanning

time depending on the scene complexity, which makes this approach impractical for real-world applications.

Although these algorithms achieve satisfying results under well-calibrated conditions, image contrast is lost and color artifacts might occur due to the intensity saturation of the projector on dark surface pigments.

#### B. Photometric Compensation with Adaptive Input Image Scaling

Due to artifacts, more recent compensation methods have focused on content-dependent adaptation to increase the visual quality of the projected images by maximizing the perceived contrast and luminance while suppressing saturation artifacts. The methods optimize the projection images with different computational complexities, depending on the application scenarios. However, these methods require a radiometrically calibrated projector-camera setup.

The first adaptive PC method was presented by [10], in which the input image is scaled automatically by a global scaling factor until the saturation artifacts approach the per-pixel visibility threshold [11]. This algorithm compensates grayscale images only, which constrains its applicability. This idea was extended to color images by [12]. They described a compensation framework that operates in the CIE  $L^*u^*v^*$  color space and applies luminance and chrominance rescaling of the input image based on human visual perception. This rescaling is optimized so that the visual impression after the PC is applied is still close to the desired input. This algorithm was further improved [13], enabling smoother chrominance adaptations. Another algorithm presented by [14] focused on a GPU-accelerated real-time adaptation to enable the system to work with real-time content. A sophisticated compensation method was recently presented by [15], which applies globally

optimized compensation using the measured light transport matrix between a camera and multiple projectors.

Although those methods show that high-quality compensated projections can be achieved with radiometrically calibrated devices, they make the usage quite cumbersome for several reasons. Any change in the hardware settings, e.g., adapting the projectors' brightness to a new setup or replacing it with a different model, requires radiometric recalibration; furthermore, most DLP projectors apply complex, multi-primary color processing and thus are difficult to accurately linearize. Although projection-based illumination in public spaces is becoming more popular, the surface colors are mostly not compensated for because the expertise and hardware needed to adequately calibrate the response curves of the projection system are missing.

### C. Photometric Compensation with One-to-Many Pixel Correspondence

The compensation techniques described implicitly assume one-to-one pixel correspondences between the projector and the camera. However, the assumption is not valid in general cases. For example, when projected on a textured surface, some pixels inevitably cover high-frequency areas, such as edges of the texture; i.e., reflectance varies widely within a single pixel area. [16] proposed measuring the reflectance within a single projector pixel area with multiple camera pixels to compute the compensation color more accurately. One-to-many pixel correspondences also must be considered in the case of long-term projector use, in which the projected pixels may move slightly on the surface for various reasons, such as projector housing deformations due to heat variations, unstable projector mounts, or projection surfaces. Consequently, the compensation color of a single projector pixel must be optimized over the multiple camera pixels on the drift trajectory. Conventional PC techniques based on one-to-one correspondence models are not suitable in these cases and will lead to a boosting of color artifacts, particularly in high-frequency areas.

### D. Our Contribution

The adaptive PC methods mainly differ in how the target images are adapted so that they fit into the limited spatially varying dynamic range of the projection. Furthermore, the complexity of illumination effects that can be compensated also varies depending on the approach. They all, however, apply variations of the core color mapping techniques assuming completely known linear system behavior. Since depending on the available hardware achieving this linearity is complicated or even impossible, we focused on developing a novel PC method that does not depend on this linearity and thus has the potential to improve the image quality of recent methods.

Our approach computes a compensation image by generating non-linear per-pixel color mapping between the radiometrically uncalibrated projector-camera pair. Scattered data interpolation based on TPS is used to calculate an accurate color transformation. The advantage of the applied RBF compared to the ones used in other scattered interpolation techniques, such

as Shepard's interpolation method [17] or multiquadrics [18], is the ability to smoothly interpolate between given sample points as well as to adequately extrapolate the colors. This method compensates for non-monotonic responses as well as inter-channel color modulations.

We extend the work presented in [19] with a thorough evaluation of the non-linear compensation approach and a quality comparison with existing methods. Furthermore, the extended method improves the compensation quality by measuring and optimizing projector misalignment in non-perfectly registered and unstable projection setups.

Specifically, two compensation image generation methods are presented: The first one processes images in real-time, while the second additionally optimizes the input image off-line in a global optimization step to further minimize local clipping errors while preserving high overall luminance and contrast. Furthermore, we propose a drift compensation method that considers temporal one-to-many pixel correspondences between the drifting projector and an observing camera. We analyze the temporal behavior of the pixel and compute a projection color that minimizes the perceptual error of the projected result for a long-term installation—even if slight projector misalignments occur.

## III. NON-LINEAR PHOTOMETRIC COMPENSATION

In the following sections, the proposed PC method is described in detail. We start with a brief summary of the algorithm and the prerequisites, followed by an in-depth description of the required data acquisition and mapping function calculation.

### A. Algorithm Overview

In contrast to pre-calibrating the camera and projector to achieve linear system behavior, our method can be applied immediately after the devices are set up and a series of projected color patterns is captured. Radiometrically uncalibrated projectors and cameras, however, require a per-pixel non-linear mapping description to accurately model the unknown color transformation from the camera to the projector via the spatially inhomogeneous reflectance of the surface material.

Although, theoretically, this can be generated using a dense sampling to populate a 3D color look-up table, this would, assuming 8-bit color depth, require several millions of projected images to be captured and stored, which is impractical considering the acquisition time and memory requirements. The number of required samples can be minimized by sparsely sampling the colors and applying sophisticated interpolation methods to compute the remaining values. Straightforward tri-linear interpolation cannot be applied directly since even if the projector projects color intensity samples with uniform spacing, the captured camera values are distributed in an irregular grid. Therefore, more complex scattered data interpolation methods are required which minimize the errors significantly, such as those presented in [20]. This method, however, still requires more than 700 images to be acquired and is therefore not practical, especially if the data has to be calculated and stored for each individual pixel. To further reduce the number

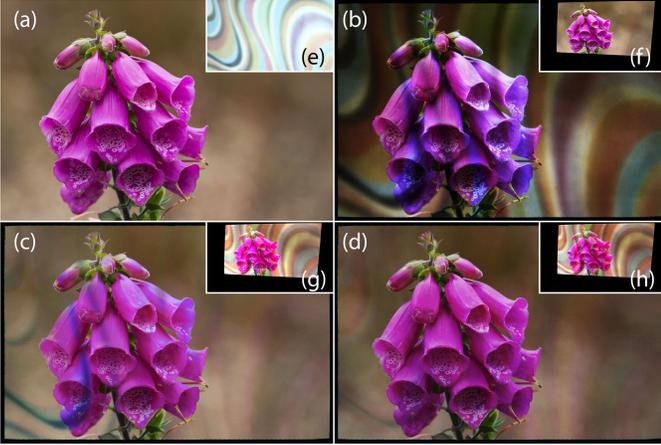


Fig. 2: (a) original input image; (e) non-uniformly colored projection surface (illuminated uniformly white); (b) captured projection of (a) on (e); (c) captured projected compensation image shows local clipping errors; (d) reduced clipping errors after applying a global optimization step. (f-h) geometrically warped projection images generating camera images (b-d).

of color samples while achieving high-quality compensation results, we apply TPS interpolation [3], which guarantees a smooth transition while interpolating between the known projected and captured sample points.

### B. Prerequisites

To acquire the series of projected color samples used to calculate the color mapping transformation, a camera device is required whose image sensor can capture the entire dynamic range of the projector without saturating in bright areas. If that is not possible depending on the hardware and the spatially varying reflectance properties of the surface, high dynamic range images can be captured as well (this implies radiometric calibration of the camera). In the particular case of using time-sequential projectors such as ones based on DLP, care must be taken to set the exposure times to exactly or a multiple of the refresh rate and to apply image averaging to minimize the influence of the time sequential color generation. Except for this precondition, no other calibration is required.

As with other camera-based compensation algorithms, a color channel adjustment is optionally carried out to adjust the projector’s white point to match the desired perceived impression of the human observer. To accurately achieve desired color values that are independent of the camera settings, full radiometric calibration can be applied beforehand, but this is not required for the algorithm to work. However, as in any camera-based photometric projector compensation technique, the resulting visual quality strongly depends on the camera used. Image degradation effects resulting from the lens were not considered in this research. To avoid strong vignetting, the aperture was set at  $f/5.6$ , which seemed to give a good compromise between image quality and light throughput.

### C. Color Mapping Method

As opposed to relying on a linear color relationship between the projector and the camera described by, for example, a per-

pixel  $3 \times 3$  [1] or  $3 \times 4$  [7] matrix multiplication, we define it by a generic per-pixel color mapping function

$$c_c^* = f(c_{in}^*) \quad (1)$$

transforming the individual input colors of each pixel  $c_{in}^*$  into the compensation colors  $c_c^*$  required to generate the desired intensities of  $c_{in}^*$  on the respective locations on the camera’s image plane (\* denotes a  $3 \times 1$  vector that stores the color intensities  $[r, g, b]$ ).

Our algorithm applies TPS interpolation to define this function in a smooth, non-linear manner. To gather the required parameters, the color space of the projector is regularly, but sparsely sampled. Although this regularity does not have to be strict, it should sample the extremes as well as the interior of the full  $RGB$  color cube evenly. This is achieved by selecting colors with  $n$  increasing intensity levels from zero to maximum intensity in all three color channels as well as their combinations, which leads to  $n^3$  color samples. If more detailed knowledge about the shape of the projector’s color response curves exists, the samples can be adjusted accordingly to get more regularized sampling. For our algorithm, we used values of  $n = 4$  up to  $n = 6$  (cf. Section V for an error analysis). These reference color samples are acquired with sequential full screen projection of uniform colors and capturing them with the camera.

Consequently, we acquire  $N = n^3$  correspondences of reference color samples and according captured colors. Hereinafter, we represent the set of reference color samples as  $[p_0^* \dots p_{N-1}^*] \in P^*$  and the corresponding set of captured colors as  $[q_0^* \dots q_{N-1}^*] \in Q^*$ . These values are used to compute the weighting factors for each per-pixel TPS interpolation function:

$$f(c_{in}^*) = \sum_{i=0}^{N-1} \omega_i^* \varphi(\|c_{in}^* - q_i^*\|) + \omega_N^* + \omega_{N+1}^* c_{in}^R + \omega_{N+2}^* c_{in}^G + \omega_{N+3}^* c_{in}^B \quad (2)$$

where  $\omega_i^*$  are the  $N + 4$  TPS weighting coefficients per color channel,  $\|\cdot\|$  the distance in Euclidean space, and  $\varphi$  is chosen to be the TPS RBF:

$$\varphi(d) = \begin{cases} 0, & d = 0 \\ d^2 \log d, & \text{otherwise} \end{cases} \quad (3)$$

This RBF minimizes the integral of the squared second derivative of  $f$  and thus is well suited to the generation of smooth color mapping. The value  $d$  is the Euclidean distance between  $c_{in}^*$  and each individual member of  $Q^*$ .

The per-pixel computation of  $\omega_i^*$  must be carried out only once per projector-camera setup and is calculated using regularized TPS [21]: To compute the values of  $\omega_i^*$ , a linear equation system is solved as described in the following. Suppose the square matrix  $L$  is composed as follows:

$$L = \begin{bmatrix} \mathbf{K} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{O}_{4 \times 4} \end{bmatrix} \quad (4)$$

where the elements of  $\mathbf{K}$ , denoted by  $k_{ij}$  with  $i, j \in [0 \dots N-1]$ , are defined by:

$$k_{ij} = \varphi(\|q_i^* - q_j^*\|) \quad (5)$$

which stores the RBFs of all captured color samples  $Q^*$  to each other.  $Q$  contains the individual color channels of all  $N$  members of  $Q^*$ :

$$Q = \begin{bmatrix} 1 & q_0^R & q_0^G & q_0^B \\ 1 & q_1^R & q_1^G & q_1^B \\ \vdots & \vdots & \vdots & \vdots \\ 1 & q_{N-1}^R & q_{N-1}^G & q_{N-1}^B \end{bmatrix} \quad (6)$$

$O_{x \times y}$  is a  $x \times y$  zero matrix. The weighting coefficients are computed by multiplying the inverted matrix of  $L$  with the reference color values ( $L$  is inverted using LU decomposition):

$$W = L^{-1} \begin{bmatrix} P \\ O_{4 \times 3} \end{bmatrix} \quad (7)$$

where matrix  $W$  stores the desired weights:

$$W = \begin{bmatrix} \omega_0^R & \omega_0^G & \omega_0^B \\ \omega_1^R & \omega_1^G & \omega_1^B \\ \vdots & \vdots & \vdots \\ \omega_{N+3}^R & \omega_{N+3}^G & \omega_{N+3}^B \end{bmatrix} \quad (8)$$

and matrix  $P$  stores the reference color samples:

$$P = \begin{bmatrix} p_0^R & p_0^G & p_0^B \\ p_1^R & p_1^G & p_1^B \\ \vdots & \vdots & \vdots \\ p_{N-1}^R & p_{N-1}^G & p_{N-1}^B \end{bmatrix} \quad (9)$$

Since  $p_i^*$  is the  $i$ -th reference color sample, this results in a vector storing the individual weighting factors  $\omega_i^*$ .

To reduce the influence of measurement inaccuracies resulting from noise and sampling, a regularization term is added to the TPS weights computation as proposed in [21]. This is realized by extending equation 5 to:

$$k_{ij} = \varphi(\|q_i^* - q_j^*\|) + \delta_{ij} \alpha^2 \lambda; \quad i, j \in [0 \dots N-1] \quad (10)$$

where  $\delta_{ij}$  represents the element of an  $(N-1) \times (N-1)$  identity matrix  $I_{N-1}$ , and

$$\alpha = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \|q_i^* - q_j^*\|}{N^2} \quad (11)$$

The weight of the regularization is defined by  $\lambda$  and was set within the range of 0.01 – 0.1 in our experiments.

Depending on the image resolution as well as the chosen number of samples  $n$ , the computation time for all  $\omega_i^*$  values for the individual pixels varies significantly. On the used hardware the mapping calculation took several minutes, but this calculation must be carried out only once per calibration. During run-time, equation 2 must be evaluated once for each pixel to calculate its compensation color, which can be carried out in real-time using the GPU: We implemented a OpenGL Shading Language (GLSL)-based compensation algorithm that achieves real-time frame rates for a 1080p projector resolution<sup>1</sup>, which proves that this method can also be applied to interactive applications.<sup>2</sup> Please refer to V-C for a more

<sup>1</sup>approx 45Hz on a Nvidia Geforce Titan Black using 5<sup>3</sup> color samples.

<sup>2</sup>The calculation can also be applied within other color spaces, such as CIEL\*a\*b\*. Since experiments showed no significant visual improvement using this color space, it was decided to carry out the computations in RGB to avoid the required computational overhead for the color transformations.

detailed performance evaluation as well as to Section V-A, where an evaluation of the sampling parameters of the method is given. Its performance is compared to other state-of-the-art compensation methods in Section V-B.

Having computed the weighting factors for each pixel, arbitrary input images can be transformed from the camera's desired colors into the corresponding values of the projector's color space by computing the transformations using equation 2. Although, in theory, this accurately approximates the desired color values, errors might occur because of out-of-gamut clipping, which arises from too bright or low RGB intensity values that cannot be reproduced on dark or colorful surface pigments or on grounds of ambient illumination. To avoid this, the input image intensities  $c_{in}^*$  can be globally adjusted by adapting the overall brightness and saturation with a uniform offset  $o$  and a scaling factor  $s$  so that the color values required for the compensation can be generated by the projector. Therefore, the input  $c_{in}^*$  is transformed into  $c_{in_{adapt}}^*$  by

$$c_{in_{adapt}}^* = o + s * c_{in}^* \quad (12)$$

before equation 1 is applied:  $c_c^* = f(c_{in_{adapt}}^*)$ . Depending on the variations of the surface reflectance, applying global intensity scaling can lead to a severe contrast and intensity reduction even if only a part of the image is affected by out-of-gamut clipping. In that case, a slight local manipulation of the input image can improve the compensation quality as explained in the next Section.

#### D. Spatially Varying Luminance Scaling

As already described by [12], an adaptive, spatially varying adjustment of the input image has the potential to increase the overall perceived image quality, especially if the surface contains high spatially reflectance variations like the one shown in Figs. 1 and 2. A possible solution for the application of these intensity adjustments to the input image is described in the following.

The perception of regional clipping errors is influenced by two main factors. On the one hand, they depend on the local reflection properties of the surface; on the other hand, the intensities of the image content are crucial. Although it is assumed that the former is static for the chosen projection setup, the latter can be manipulated by optimizing the input image's intensities and slightly changing the content similar to the approaches presented in [12], [14]. We have chosen to smoothly adapt the luminance of the input colors with a set  $S$  of spatially varying luminance scaling values  $s(x, y) \in S$  to avoid clipping. For the current approach, it is assumed that  $o$  is static over time and the luminance offset of the captured ambient illumination conditions is already added to  $c_{in}^*$ . Of course, this should be kept at a minimum, since any source of ambient illumination should be avoided to reduce unnecessary contrast reductions. To achieve the smoothly adapted values for  $S$ , non-linear optimization is applied to the input image to estimate the optimized spatially varying luminance scaling values  $S_{opt}$ :

$$S_{opt} = \arg \min_S err_{opt}(S), \quad (13)$$

which minimizes the error function

$$err_{opt}(S) = \sum_{x=0}^{width-1} \sum_{y=0}^{height-1} \omega_{sat} \cdot err_{sat}(S, x, y) + \omega_{grad} \cdot err_{grad}(S, x, y) + \omega_{int} \cdot err_{int}(S, x, y) \quad (14)$$

consisting of the following three per-pixel error components:

- Saturation error, occurring due to limited maximum projector brightness and generating perceived image artifacts:

$$err_{sat}(S, x, y) = \begin{cases} (f(c_{in}^*(x, y)s(x, y)) - 1.0)^2, & f(c_{in}^*(x, y)s(x, y)) > 1.0 \\ 0, & otherwise \end{cases} \quad (15)$$

penalizing the occurring out-of-gamut errors that should be neutralized.

- Intensity error, resulting from an intensity reduction at the current pixel, lowering the image brightness (an increase in intensity is accepted):

$$err_{int}(S, x, y) = \begin{cases} (1.0 - s(x, y))^2, & s(x, y) < 1.0 \\ 0, & otherwise \end{cases} \quad (16)$$

This error measure ensures that the overall image luminance is maintained at a reasonable high level since human visual perception usually prefers brighter projections within the normal limits of projection systems [22].

- Gradient variation error resulting from the spatially varying intensity adjustments leading to potentially visible local intensity variations:

$$err_{grad}(S, x, y) = (s(x, y) - s(x-1, y))^2 + (s(x, y) - s(x+1, y))^2 + (s(x, y) - s(x, y-1))^2 + (s(x, y) - s(x, y+1))^2 \quad (17)$$

Ensuring a smooth change in the adapted spatial luminance scaling values to minimize the perceived spatial manipulation of the input image is the goal of this error term.

Independent weights are applied to these errors to generate an acceptable tradeoff between the image degradation from clipping errors as well as global and smooth local luminance reduction in the final error term as shown in equation 14. For our experiments, we used error weights of  $\omega_{sat} = 0.797$ ,  $\omega_{grad} = 0.004$ , and  $\omega_{int} = 0.199$ . Since the main goal of this optimization is the neutralization of visible out-of-gamut clipping errors,  $\omega_{sat}$  was assigned the highest weight. The lower weights of  $\omega_{int}$  and  $\omega_{grad}$  seemed to be sufficient to keep the image adaptations minimal while preserving high luminance. In addition, since the scaling factor is optimized at a sub-sampled resolution and then smoothly up-sampled for speeding up the process, the spatial variance of the scaling factor at the original resolution becomes small. Therefore, we assigned the lowest weight to  $\omega_{grad}$ . Please note that automatic adaptation of these values that depend on the system configuration and the human visual system will be part of future evaluations.

Before optimization starts, all members of  $S$  are uniformly initialized by a value that decreases the overall number of

saturated pixels in  $f(c_{in}^*(x, y)s(x, y))$  to  $\leq 1\%$ . To speed up optimization is applied only in a sub-sampled resolution. Note that this requires a new calculation of weights for the mapping functions to ensure that the worst-case scenario is considered in this lower-resolution calculation. Therefore, the original pixel values are not simply averaged during rescaling. Instead, all members of  $Q^*$  are sub-sampled such that of all the pixels collapsing to a single one, the one containing the lowest luminance is chosen, while for the individual images to compensate ( $c_{in}^*$ ), the one with the highest luminance is used.

Currently, we use a sub-sampling of  $40 \times 28$  pixels, which seemed to be a reasonable trade-off in terms of accuracy and computation time for our test setups. To solve this relatively large number of variables within a reasonable amount of time, bound constrained optimization (BOBYQA) [23], which does not require derivatives, was applied. However, as stated in [24], the number of variables in the sub-sampled computation should not be set to a too high value, since this will make the optimization no longer feasible. To further reduce the computation time, the color mapping of the individual sub-sampled pixels was pre-computed and stored in lookup tables (LUT). In a post-process, the computed result is smoothly up-sampled into its original resolution with a Gaussian kernel and the computed per-pixel scaling values  $S$  are then used to adjust the luminance of  $c_{in}^*$  to generate  $c_{in}^{*adapt}$  as shown in equation 12 that will be used as input for equation 1 (cf. Figure 2 for a comparison of global compensation with and without the local adjustment step.).

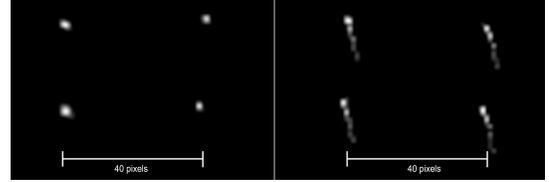


Fig. 3: Close-up views of four pixel drifts during two measurements over a duration of 2.5 hours. The intensity encodes the number of occurrences. Left: Projector has been placed on a concrete surface. Right: Projector has been mounted on a tripod. While the first setup shows a satisfying stability over time, the vertical drift is clearly visible in the latter.

#### IV. DRIFT ERROR MINIMIZATION

Although the results generate the expected image quality under optimal, stable calibration conditions, problems occur due to slight, unavoidable movements if this method is applied in a real-world installation that is required to run for hours or even days without any recalibration. As mentioned in the introduction, there can be various reasons for those movements, such as projector housing deformations due to heat variations, unstable mounts, or surfaces. Since in real-world installations such slight movements can occur unexpectedly and projectors might not be immediately accessible for manual realignments, we propose to adapt the projected compensation image to take the potential projector drift into account.

To avoid the appearance of visually disturbing color edges due to misaligned projections, drift analysis and perceptual



Fig. 4: Comparison of a drift optimized compensation (right) with a conventionally compensated projection (center) in a misaligned projection setup. The input image is shown in the upper left, the lower left shows the projection surface containing hard edges. The other images in the lower row show close ups of the regions highlighted in red. Note that the hard color seams are visually reduced with our approach.

error minimization are carried out to increase the image quality even when the projector becomes slightly misaligned over time by distributing the expected errors depending on the potential movement.

#### A. Algorithm Overview

Minimizing the perceived error resulting from a misaligned projection system requires additional knowledge in addition to the surface reflectance measurements that must be carried out for the PC: The misalignment behavior of the individual projector pixels must be defined as an input of the error minimization calculation. Depending on the requirements and prerequisites, this behavior can either be modeled by some kind of heuristic distribution function or can be directly acquired by the projector-camera system by measuring individual pixel locations over a sufficiently long amount of time. The results of the latter are used to describe a probability distribution function that approximates the temporal projector movement for each individual pixel accurately. In our tests, we applied elliptic Gaussian functions as approximation functions. Although this might be relatively rough approximations, they seemed to offer an acceptable tradeoff between accuracy and generality, since the measurements generate only a relatively small number of samples that include measurement noise as well.

#### B. Drift Measurements

To acquire and estimate the potential projector movement, we measured a series of projector to camera mappings over time with a fixed measurement interval. The mappings were generated using a combination of robust gray codes and an additional grid of blob patterns to acquire sub-pixel accurate measurements.<sup>3</sup> To avoid measurement errors resulting from

<sup>3</sup>For our experiments, we captured one reference measurement and additional 60 measurements with a time offset of 150s after initially heating up the projectors for at least one hour.

camera motion when a DSLR camera is used with a physically moving mirror, which potentially moves the camera slightly, we used a GiG-E camera<sup>4</sup> for these experiments.

After the measurements were captured, corresponding temporal points were collected, and for each point set, a 2D Gaussian ellipsoid was fitted by calculating its mean and covariance matrix. Using this data allows us to calculate localization probabilities for all surrounding pixels describing how likely a specific projection pixel might be located at this position. To speed up the optimization described later, for each pixel the drift probabilities are precomputed for a specific set of  $x, y$  coordinates surrounding the current pixel  $p$  and stored in a set  $P$  of probabilities for all pixels of interest. The size of this 2D region of interest is calculated by searching for the maximum deviation of the current pixel's reference measurement to its mean value. Equation 18 was used to calculate the probabilities for the discrete pixel locations:

$$p(x, y) = \frac{1}{2\pi C(0, 0)C(1, 1)} \cdot \exp\left(-\frac{1}{2 \cdot C(0, 0)^2} \cdot (x - m_x)^2 - \frac{1}{2C(1, 1)^2} \cdot (y - m_y)^2\right), \quad (18)$$

where  $C$  is the  $2 \times 2$  covariance matrix, and  $m_x$  and  $m_y$  are the mean  $x, y$  coordinates of the current pixel. Figure 3 shows the plots of four sample pixels within two different setups. For the left one, the projector was placed on the ground, while in the right one, the projector was placed on a tripod. As can be seen, depending on the setup and the hardware, the movement varies significantly.

#### C. Perceivable Drift Error Minimization

To estimate the projection color that generates the least perceivable error for all possible projector movements, we also

<sup>4</sup>Allied Vision Manta, 4MP

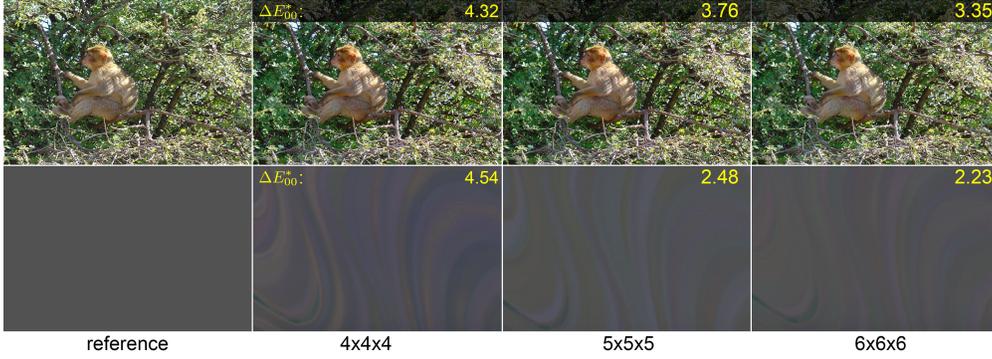


Fig. 5: Compensation accuracy with respect to the number of samples used to compute the TPS weights: The overlaid numbers shows the average  $\Delta E_{00}^*$  of the two sample image series with respect to the reference images shown on the left. While there is a clear quality improvement between  $4^3$  and  $5^3$  samples, the extension to  $6^3$  samples only marginally improves the result, while requiring almost twice as many images. For this setup, an LCD projector (Epson EH-TW3200) has been used.

applied non-linear error minimization that will be described in the following<sup>5</sup>.

Instead of just computing the forward mapping from the input to compensation pixel intensities using the TPS mapping function 1, we also computed the inverse mapping functions for each pixel

$$c_{in}^* = f^{-1}(c_c^*) \quad (19)$$

transforming the compensation intensities back to its original values. This was carried out by calculating the individual TPS weights by switching the input and output color samples.<sup>6</sup> Obviously, applying equation 19 directly to the output of equation 1 for the same coordinate  $x, y$  should lead to a result that is very close to the input image intensities. If the projection has been moved, this prerequisite no longer holds, but the computation can be used to calculate the appearance of the projection image while being moved during a physical drift. Since we do not know the actual position of the projected pixel on the surface, but gathered a probability function from the measurements as described in Section IV-B or with a heuristic function, we applied per-pixel error minimization adjusting the color intensities for each individual projected pixel  $c_{c_{opt}}^*$  such that the perceived error of the misaligned projection is minimized for all possible drift positions weighted by their probabilities:

$$c_{c_{opt}}^* = \arg \min_{c_c^*} err_{drift}(p, c_c^*) \quad (20)$$

where  $c_c^*$  is an individual pixel value at any coordinate  $x, y$ , and  $p$  stores the according drift probabilities. The error function  $err_{drift}$  calculates the expected error for the current input pixel at location  $x, y$  by:

$$\begin{aligned} & err_{drift}(p, c_c^*) \\ &= \sum_{s=-r_x}^{r_x} \sum_{t=-r_y}^{r_y} p(s, t) \cdot \Delta E_{00}^*(c_{est}(x+s, y+t), c_{in}(x, y)) \end{aligned} \quad (21)$$

<sup>5</sup>Please note that, besides the fact that we integrated this method into our proposed TPS mapping function, it can be applied with any of the existing PC methods and does not rely on our proposed TPS mapping.

<sup>6</sup>For the matrix multiplications used in the related work, the inverse mapping can easily be calculated by inverting the color transformation matrix.

where  $r_x$  and  $r_y$  are the range of drift probabilities stored for each individual pixel (this might be constant, spatially varying, or even computed on the fly),  $p(s, t)$  is the probability for the current pixel at the given offset coordinate  $s, t$  and  $\Delta E_{00}^*$  is a CIE standard for perceptual color differences (cf. [25]).  $c_{est}(x+s, y+t)$  is the estimated color value after backward color mapping is applied to the compensation image using the mapping function of the current offset coordinate  $x+s, y+t$  instead of  $x, y$ :

$$c_{est}^*(x+s, y+t) = f_{(x+s, y+t)}^{-1}(f_{(x, y)}(c_{in}^*(x, y))) \quad (22)$$

Since this approach optimizes only three variables and is solved independently for each pixel, it can be easily parallelized and solved using the BOBYQA algorithm [23] in a reasonable amount of time. If no drift is expected at all, the optimization still can be applied to further enhance the compensation image quality generated by applying thin plane spline mapping. This results from the fact that if the pixel offset is set to 0, re-projection error minimization is applied to reduce potential color mapping inaccuracies by applying the inverted color mapping function (cf. equ. 19). Since the latter also contains inaccuracies, the result still might not be the exact compensation color, but with reduced error compared to forward color mapping alone.

In addition to the samples shown in Fig. 4, additional measurements and drift compensation results are presented in Section V-B.

## V. EVALUATION

To evaluate the performance of the proposed methods, various hardware setups were used to compare the image quality based on the chosen parameters as well as in comparison to related compensation algorithms [6], [7]. The projectors were linearized beforehand with a commercial calibration system<sup>7</sup>. This hardware calculates only linear responses for the three primaries and a white channel, and thus fails to accurately generate linearization for all kinds of mixtures of the red, green, and blue color channels. Since the black box processing

<sup>7</sup>Datacolor Spyder 4 Elite

of the projectors varied significantly between the different devices, the algorithms performed with noticeable quality differences. This will be discussed in the following sections. Our proposed method does not require linearization; it was applied for an equal comparison of the different techniques under the same conditions.

We will start with the first evaluation of the image quality regarding the number of samples used for the TPS mapping, followed by an analysis of the image quality using different projectors and color processing settings. Finally, after a comparison of processing times, we also evaluate the effects of the drift error compensation.

### A. Image Quality Regarding Sampling Density

The number of color samples used to generate the mapping function strongly influences the accuracy and thus the resulting image quality. In Section III, we mentioned that for our experiments, the number of samples varied from  $4^3$  to  $6^3$  samples. Figure 5 shows a comparison of the resulting image quality regarding the sampling density for an LCD projector setup containing the strongly saturated projection surface shown in Fig. 2. Here, the  $\Delta E_{00}^*$  error metric was applied. Using more samples decreased the average error. For this particular setup,  $5^3$  samples seemed to be a reasonable tradeoff between image quality, acquisition, and computation time. Note that the image content strongly influences the perceived errors due to the contrast sensitivity and visual masking properties of the human visual system. For other setups, for example, the BenQ projection system described in the following section and shown in Fig. 10, a greater amount of non-linear color channel interdependency was encountered, and therefore  $6^3$  samples were required to achieve satisfactory image quality.

### B. Comparison with Other Photometric Compensation Methods

Two off-the-shelf projectors were used for evaluating the image quality of the developed algorithm and to compare it to related approaches. In the first setup, the goal was to generate a projector-camera system that is calibrated such that a linear response is approximated as much as possible. The results should show whether the proposed method generates results that are adequate or still provide a better image quality compared to linear compensation methods, such as the one presented by [7]. Therefore, we used a 3-chip LCD projector<sup>8</sup> and a machine vision GigE camera<sup>9</sup> to provide linear data. Although this camera has a relatively poor color rendition, it was chosen to avoid a linearization operation that might contain additional calibration inaccuracies and to reduce the overall acquisition and processing time compared to a DSLR camera with RAW image data. Since the application purpose of this projector is home cinema, it offers comparably good color reproduction; additional internal color processing could be deactivated so that the response curves could be calibrated accurately using a colorimeter.

<sup>8</sup>Epson EH-TW3200

<sup>9</sup>Allied Vision Manta G-504C

We used the same surface pattern as the one shown in Fig. 2 for this evaluation, since it contains various different saturated colors and does not lead to severe errors during small misregistrations. In addition to an uncompensated projection, we captured the projected compensation images generated with the following methods:

- Bimber’s linear, non-color mixing method [6]
- Yoshida’s linear,  $3 \times 4$  color mixing matrix method [7]
- Our proposed non-linear TPS method using  $6^3$  input samples

For computing the required  $3 \times 4$  color mixing matrix as well as the TPS weights, the same  $6^3$  input samples were used.

In Fig. 6, the captured images of several compensated projections of the LCD projector are shown. As can be seen, all compensation methods significantly decreased the visibility of the surface structure.

The proposed non-linear compensation approach decreases color errors even more than the already well-compensating linear method and thus better approximates the desired input images. While this is visually shown in the figure, it is also objectively visualized in the comparisons shown in Fig. 7: It shows the  $\Delta E_{00}^*$  and the SSIM of the different methods compared to the input images.

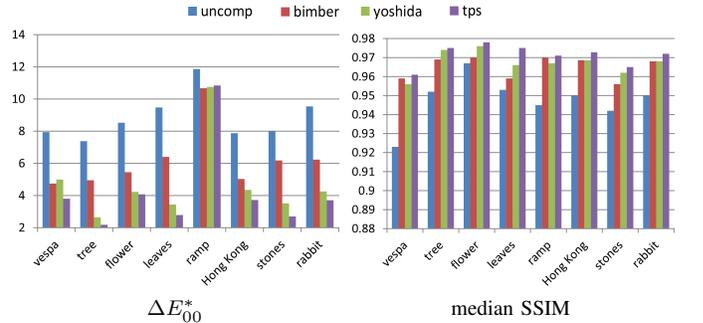


Fig. 7:  $\Delta E_{00}^*$  and median SSIM plots of the captured image with respect to the input images shown in Fig. 6. While all compensation methods can better reproduce the image contents compared to the uncompensated version, our proposed method shows an advantage over Yoshida’s linear color mixing method even if the camera and projector have been carefully linearized beforehand.

Table I lists the median  $\Delta E_{00}^*$  values as well as the median SSIM of the sample images. We used the median values since the color ramp sample generates far worse results than the other samples due to the saturated colors that were out of gamut of the camera (cf. Fig. 8 for a successful compensation of this image using a DSLR). However, our proposed method outperforms the other methods in both metrics.

Method	Median $\Delta E_{00}^*$	SSIM
Uncompensated	8.265	0.948
Bimber	5.814	0.965
Yoshida	4.239	0.967
TPS	<b>3.715</b>	<b>0.971</b>

TABLE I: Averaged median  $\Delta E_{00}^*$  and SSIM measurements for the images shown in Fig. 6.

Since a machine vision camera with relatively low-quality

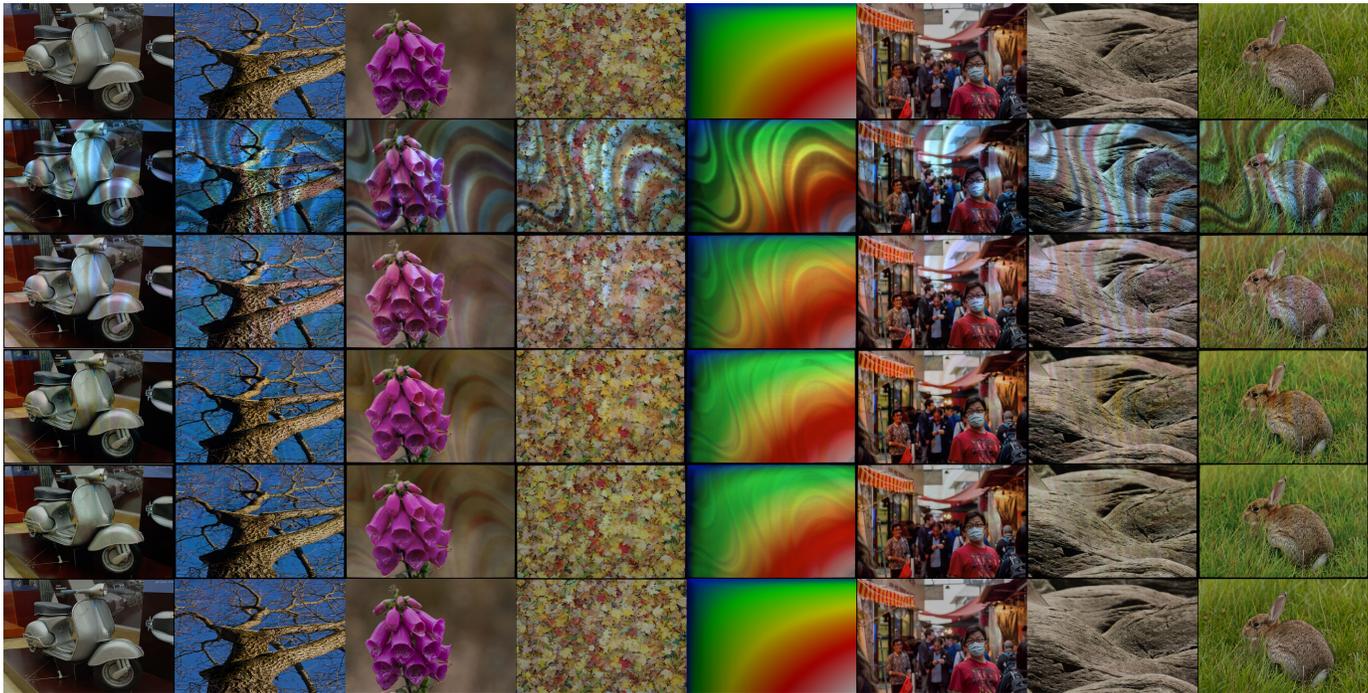


Fig. 6: Comparison between uncompensated projections (2nd row), Bimber’s method (3rd row), Yoshida’s method (4th row), and the proposed method using  $6^3$  samples (5th row). It is clear that the color errors are reduced depending on the complexity of the method. See Fig. 7 for details. The LCD projector was calibrated to a gamma of 1.0, and all special color processing was turned off beforehand. For reference, the input images are shown in the top and bottom rows.

color reproduction was used to avoid response curve linearization errors, even the TPS compensation is not able to fully hide the underlying surface. If a DSLR camera capturing RAW data is used instead, the compensation quality can be further enhanced. See Fig. 8 for this particular setup.

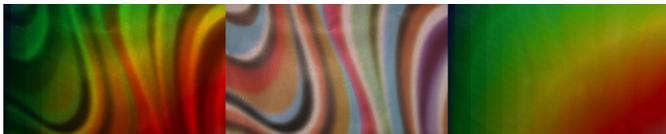


Fig. 8: Sample result using a DSLR camera in RAW image mode (Canon Eos 5D MkII). The left side shows the captured image of the uncompensated projection. The projection surface under uniform illumination is shown in the center. On the right side, a captured image of the projected compensated image using the proposed method is shown.

In the second experiment, a DLP-based presentation projector was used<sup>10</sup> that contains several internal color processing operators. For this setup, the projector again was calibrated to a linearized response; however, the internal, unknown CMY and white color processing made it impossible to achieve accurate linearization for arbitrary input images. This is clearly seen in Fig. 10. Interestingly the linear matrix transformation method [7] generated results that were comparable or even worse than the method of [6] not considering color mixing.

Table II lists the median  $\Delta E_{00}^*$  and SSIM of the sample images. The individual results are visualized in Fig. 9. Again,

<sup>10</sup>BenQ W1100

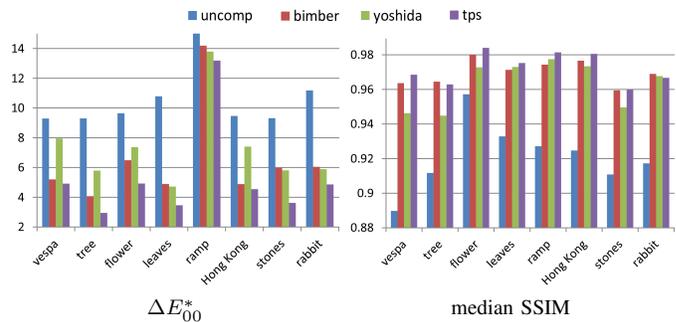


Fig. 9:  $\Delta E_{00}^*$  and median SSIM plots of the captured images for the input images shown in Fig. 10.

our proposed method generated an increased image quality compared to the other methods.

Method	Median $\Delta E_{00}^*$	SSIM
Uncompensated	9.644	0.921
Bimber	5.975	0.969
Yoshida	5.882	0.963
TPS	<b>4.544</b>	<b>0.972</b>

TABLE II: Averaged median  $\Delta E_{00}^*$  and SSIM measurements for the images shown in Fig. 10.

While all three evaluated compensation methods can better reproduce the image contents compared to uncompensated projection, our proposed method shows an improvement in image quality even if the projector is carefully linearized

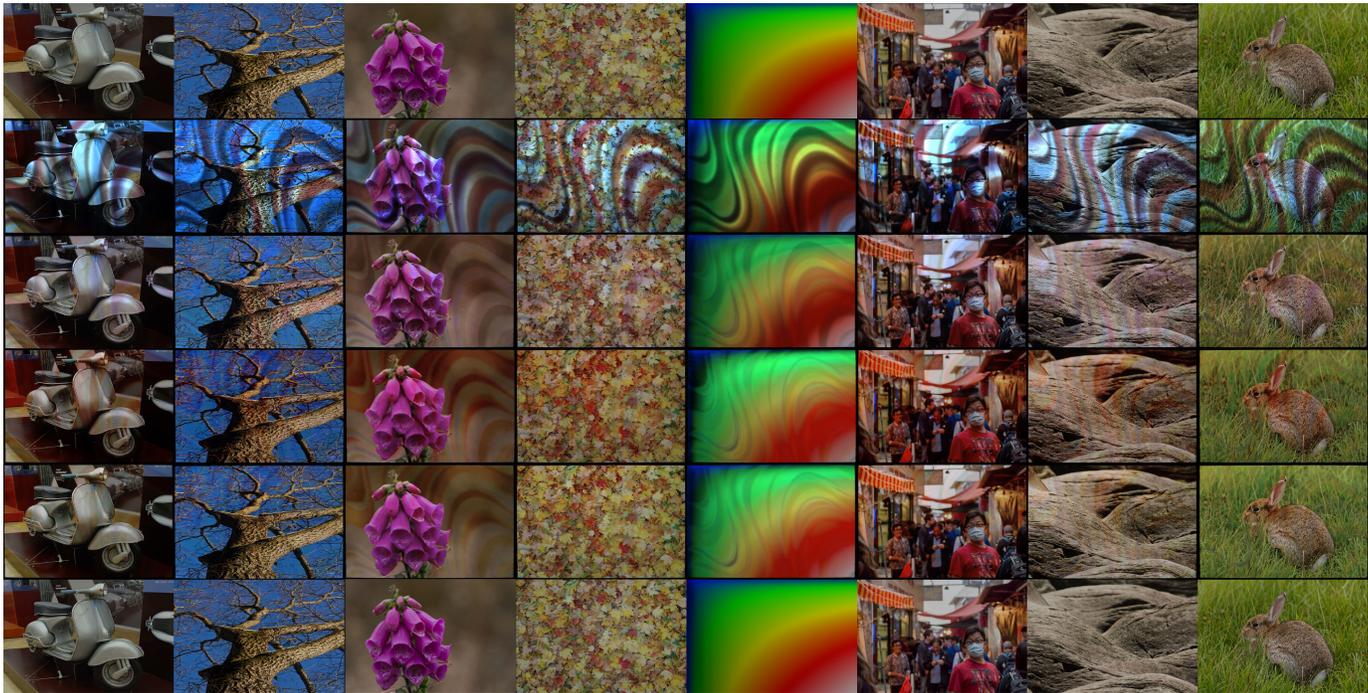


Fig. 10: Comparison between uncompensated projection (2nd row), Bimber’s method (3rd row), Yoshida’s method (4th row), and the proposed method using  $6^3$  samples (5th row). The linear matrix transform, in this case, is not able to coop with the non-linearity of the DLP’s color processing. See Fig. 9 for details. The DLP projector was calibrated to Gamma 1.0, but in this case, white boosting and “brilliant” color processing was turned on during calibration. For reference, the input images are shown in the top and bottom rows.

beforehand.

The improvement of the proposed method was bigger in the case of the DLP projector than that of the LCD projector although the same number of  $6^3$  samples was used in both experiments. Because the DLP projector generally has more non-linear and inconsistent color space than the LCD projector, conventional linear methods are not suitable for DLP-based devices. However, our method compensates for the nonlinearities of the DLP’s internal processing. Therefore, the image quality can be improved further in the case of the DLP projector.

### C. Processing Time Analysis

The improved visual quality comes with the tradeoff of longer computation times. In Table III, the different methods are compared regarding their computation times for the initial mapping parameter calculation that must be carried out only once for each setup and for the compensation image generation that, depending on the application, requires real-time computation in the case of interactive or unknown dynamic content. As can be seen, the pre-computation time scales significantly; however, real-time processing is possible using state-of-the-art GPUs.

### D. Perceptual Error Reduction with Drift Error Compensation

To investigate the potential image quality improvement in a geometrically unstable projection setup, we synthetically generated a drift probability map of a reasonable size ( $5 \times 11$  projector pixels in our test case) with similar parameters to the

	Mapping (s)	Comp. CPU (s)	Comp. GPU (ms)
Ref.[6]	0	0.05	< 1
Ref.[7]	34	0.07	< 1
TPS $4^3$	132	2.6	17
TPS $5^3$	403	4.8	21
TPS $6^3$	1763	7.1	34*

TABLE III: Timing comparison of the proposed methods using  $4^3$ - $6^3$  samples and the methods of [6] and [7]. *Mapping* denotes the initial calculation of the required parameters, while *Comp.* denotes the image dependent calculations which were implemented on the CPU and on the GPU. Note that the starred mapping of  $6^3$  samples could not be calculated in the given  $1080p$  resolution due to memory limitations; therefore, the mapping was calculated with  $720p$  in that particular case. Since other GPUs already offer up to  $16GB$  of RAM and next-gen GPUs were already announced with this amount doubled this limitation will soon become negligible.

ones that were estimated during our test measurements. Using the additionally acquired color samples, two compensation images were generated using the proposed TPS color mapping algorithm: For the first image, only center pixel compensation was directly applied, as explained in Section III-D. For the second, the drift compensation optimization described in Section IV-C was applied afterward. Since the goal is to provide an enhanced image quality for arbitrary slightly offset positions, which might occur during projection, we synthetically shifted both images on the projector’s image plane by  $5 \times 11$  pixels to generate misregistered samples. For all of them, the projected images were captured and compared to the reference input using the same approach described in Section V-B. The results

of the SSIM evaluation are visualized in Fig. 11: The heat maps show the median SSIM for the  $5 \times 11$  shifted pixel positions for any of the four test images. As can be seen, the drift compensated projection is able to increase the SSIM in offset areas as long as it is not too far away from the optimal position. The approach outperforms the center pixel compensation in all of the four test images for all offset values and closely matches the SSIM even in the not-shifted position. The same can be observed by analyzing the  $\Delta E_{00}^*$  values (see Table IV). Again, the drift compensation decreases the values in a significantly wider range than the standard compensation approach alone. As can be seen, on average, the median error can be significantly reduced with our proposed method; however, for a perfectly registered projection, the standard compensation method generates similar or even slightly better results.

Image	DC median $\Delta E_{00}^*$	CC median $\Delta E_{00}^*$
Ship	<b>4.279</b>	4.688
Tree	<b>3.270</b>	3.364
String	<b>4.295</b>	4.610
People	<b>4.747</b>	5.066

TABLE IV: Median  $\Delta E_{00}^*$  measurements of the four images shown in Fig. 11. Left: results of the drift compensation (DC) method. Right: values for a conventional compensation (CC).

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we showed that robust error-tolerant high-quality photometric compensation can be carried out without any radiometric pre-calibration of the device (see Figs. 1, 2, 4, and 8). By projecting and capturing a reasonable number of images, a non-linear color mapping function is generated that calculates an accurately compensated projection image even on strongly textured surfaces. The proposed method offers several advantages compared to existing methods:

- Linearization errors resulting from inaccurate or noisy radiometric calibration do not affect the compensation quality.
- The algorithm compensates for the complex, multi-primary processing of many off-the-shelf consumer projectors.
- No further calibration hardware is required, which makes deploying the compensation straightforward.
- The visibility of color seams at strong surface color gradients can be efficiently reduced by applying the proposed drift measurement and compensation method.

The overall memory requirement is higher compared to the linear mapping algorithms. In our system, we require up to  $436 \times 3 (= (6^3 \times 2 + 4) \times 3)$  floating point values for each pixel (for  $6^3$  color samples) which, depending on the resolution, requires several gigabytes of memory. We could, however, show that even current consumer GPUs can process this data in real-time. The global optimization takes, depending on the setup, 5 to 30 minutes to converge. The current drift compensation implementation takes several minutes to converge as well (5–10 minutes in our examples). A GPGPU-based non-linear optimization might even achieve interactive frame rates including these optimizations.

Currently, we are trying to directly map desired input colors to output color only accepting smooth luminance variations. Incorporating gamut mapping algorithms [26] accounting for the spatially varying gamut as well as extending the work presented in [27] for multi-projector systems to per-pixel gamuts is part of our future work.

Although the proposed drift error compensation method reduces the perceptibility of hard edges that appear during misaligned projections, future work is needed on their evaluation and compensation depending on the actual image content. The relatively simple error metric that is currently applied could be replaced with a more sophisticated error model that takes contrast sensitivity, saliency, and the temporal component into account to further reduce the perceived artifacts. This will be part of future investigations as well.

Finally, we are working on a more thorough evaluation of the method using a series of varying surface textures, projectors, and input images. Since this is a laborious process due to the calibration requirements and processing needs, we are planning to use a physically based rendering system to accurately evaluate the different methods in simulation.

## REFERENCES

- [1] S. K. Nayar, H. Peri, M. D. Grossberg, and P. N. Belhumeur, “A Projection System with Radiometric Compensation for Screen Imperfections,” in *ProCams*, 2003. 1, 2, 4
- [2] A. Majumder and R. Stevens, “Color nonuniformity in projection-based displays: Analysis and solutions,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 10, no. 2, Mar. 2004. 1
- [3] J. Duchon, “Splines minimizing rotation-invariant semi-norms in Sobolev spaces,” in *Constructive Theory of Functions of Several Variables*, ser. Lecture Notes in Mathematics, 1977, vol. 571, ch. 7. 1, 4
- [4] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer, “The visual computing of projector-camera systems,” in *Proc. Eurographics (State-of-the-Art Report)*, 2007. 1
- [5] J. Salvi, J. Pagès, and J. Battle, “Pattern codification strategies in structured light systems,” *Pattern Recognition*, vol. 37, 2004. 2
- [6] O. Bimber, A. Emmerling, and T. Klemmer, “Embedded entertainment with smart projectors,” *IEEE Computer*, vol. 38, no. 1, 2005. 2, 8, 9, 10, 11
- [7] T. Yoshida, C. Horii, and K. Sato, “A virtual color reconstruction system for real heritage with light projection,” in *Proc. of International Conference on Virtual Systems and Multimedia (VSMM)*, 2003. 2, 4, 8, 9, 10, 11
- [8] X. Chen, X. Yang, S. Xiao, and M. Li, “Color mixing property of a projector-camera system,” ser. *ProjCams*. ACM, 2008. 2
- [9] G. Wetzstein and O. Bimber, “Radiometric compensation through inverse light transport,” *Proc. of Pacific Graphics*, 2007. 2
- [10] D. Wang, I. Sato, T. Okabe, and Y. Sato, “Radiometric compensation in a projector-camera system based properties of human vision system,” 2005. 2
- [11] M. Ramasubramanian, S. N. Pattanaik, and D. P. Greenberg, “A perceptually based physical error metric for realistic image synthesis,” 1999. 2
- [12] M. Ashdown, T. Okabe, I. Sato, and Y. Sato, “Robust content-dependent photometric projector compensation,” in *Proc. of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE Computer Society, 2006. 2, 5
- [13] S. H. Park, S. Yang, and B.-U. Lee, “Adaptive chrominance correction for a projector considering image and screen color,” in *Proc. of the 3rd international conference on Advances in visual computing*, ser. ISVC’07, 2007. 2
- [14] A. Grundhöfer and O. Bimber, “Real-time adaptive radiometric compensation,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 14, 2008. 2, 5
- [15] D. G. Aliaga, Y. H. Yeung, A. Law, B. Sajadi, and A. Majumder, “Fast high-resolution appearance editing using superimposed projections,” *ACM Trans. Graph.*, vol. 31, no. 2, 2012. 2

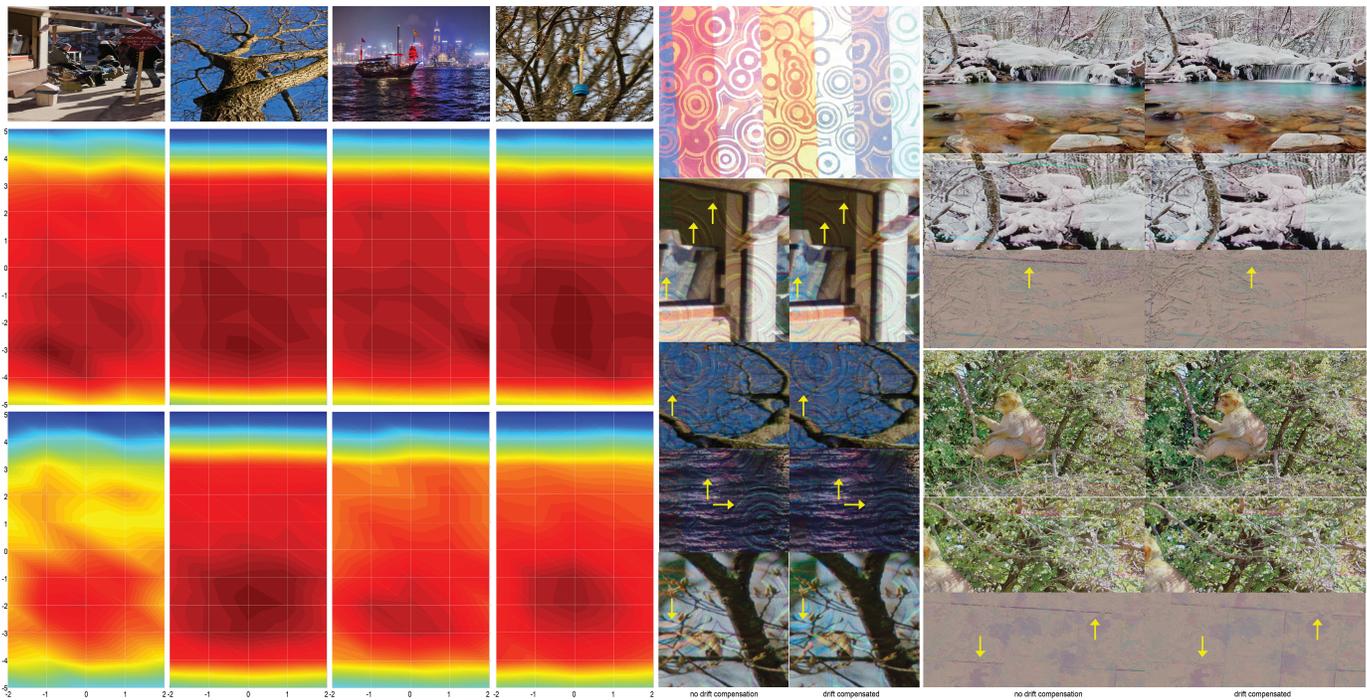


Fig. 11: Left: SSIM heat maps of shifted and projected compensation images compared with the four reference images (top row) are shown. The heat maps in the upper row show results for the  $5 \times 11$  times shifted drift compensated projections, the lower row without optimizing potential projection drift. The SSIM values range from 0.87 (blue) to 0.94 (red). The numbers indicate the pixel shift position from the initial measurements. The projection surface is shown at the center top. Below, four magnified comparisons are shown of uncompensated and drift-compensated projections (cf. red regions). On the right side, further drift compensation results and (inverted) difference images to the input image are given (cf. the surface shown in Fig. 4). Please note that print and display color settings might affect the impression. The SSIM and  $\Delta E_{00}^*$  results, however, clearly prove the measurable quality improvement of our method.

- [16] S. Mihara, D. Iwai, and K. Sato, "Artifact reduction in radiometric compensation of projector-camera systems for steep reflectance variations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1631–1638, 2014. 3
- [17] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," 1968. 3
- [18] R. L. Hardy, "Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968-1988," *Computers & Mathematics with Applications*, vol. 19, no. 8, 1990. 3
- [19] A. Grundhofer, "Practical non-linear photometric projector compensation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013, pp. 924–929. 3
- [20] B. Sajadi, M. Lazarov, and A. Majumder, "Adict: accurate direct and inverse color transformation," in *Proc. of the 11th Europ. conference on Computer vision*, ser. ECCV'10, 2010. 3
- [21] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *Proc. of the 7th Europ. Conference on Computer Vision*, ser. ECCV '02, 2002. 4, 5
- [22] P. S. Guterman, K. Fukuda, L. M. Wilcox, and R. S. Allison, "75.3: Is brighter always better? the effects of display and ambient luminance on preferences for digital signage," *SID Symposium Digest of Technical Papers*, vol. 41, no. 1, 2010. 6
- [23] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," Aug. 2009. 6, 8
- [24] A. Conn, K. Scheinberg, and L. Vicente, *Introduction to Derivative-free Optimization*, ser. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009. 6
- [25] G. Sharma, W. Wu, and E. N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research and Appl.*, vol. 30, no. 1, 2005. 8
- [26] J. Morović, *Color Gamut Mapping*, ser. The Wiley-IS&T Series in Imaging Science and Technology. Wiley, 2008. 12
- [27] B. Sajadi, M. Lazarov, M. Gopi, and A. Majumder, "Color seamlessness in multi-projector displays using constrained gamut morphing," vol. 15, no. 6, Nov 2009, pp. 1317–1326. 12



CV, AR, video processing, and display technologies.

**Dr. Anselm Grundhofer** is a Senior Research Engineer at Disney Research Zürich, Switzerland, where he works in the computer vision group. He graduated with a degree in Media Systems Sciences in 2006 and obtained his doctorate in Engineering in 2010 at Bauhaus University, Weimar, Germany. His work is focused on developing procams systems and assisting in their deployment. He is researching optimization methods to overcome several unresolved problems involved when projecting onto non-trivial surfaces and, aside from this, conducts research on



**Dr. Daisuke Iwai** received his B.S., M.S. and Ph.D. degrees from Osaka University, Japan, in 2003, 2005, and 2007, respectively. He was a visiting scientist at Bauhaus-University Weimar, Germany, from 2007 to 2008, and a visiting associate professor at ETH Zürich, Switzerland, in 2011. He is currently an associate professor at the Graduate School of Engineering Science, Osaka University. His research interests include human-computer interaction, projection-based mixed reality, and projector-camera systems.