Truthful Color Reproduction in Spatial Augmented Reality Applications

Christoffer Menk and Reinhard Koch

Abstract—Spatial augmented reality is especially interesting for the design process of a car, because a lot of virtual content and corresponding real objects are used. One important issue in such a process is that the designer can trust the visualized colors on the real object, because design decisions are made on basis of the projection. In this paper, we present an interactive visualization technique which is able to exactly compute the RGB values for the projected image, so that the resulting colors on the real object are equally perceived as the real desired colors. Our approach computes the influences of the ambient light, the material, the pose and the color model of the projector to the resulting colors of the projected RGB values by using a physically based computation. This information allows us to compute the adjustment for the RGB values for varying projector positions at interactive rates. Since the amount of projectable colors does not only depend on the material and the ambient light, but also on the pose of the projector, our method can be used to interactively adjust the range of projectable colors by moving the projector to arbitrary positions around the real object. We further extend the mentioned method so that it is applicable to multiple projectors. All methods are evaluated in a number of experiments.

Index Terms—Computer graphics, picture/image generation, display algorithms, raytracing, augmented reality, image processing and computer vision, radiometry, color

1 Introduction

The automotive industry uses a lot of virtual content for designing, developing and assessing components in the design process of a new car. This virtual content is traditionally visualized by using a monitor, projection wall or CAVE. However, real objects, e.g., hardware mockups, are still used and preferred in most steps of the design process, because the designer has a more realistic assessment of the different components in reality. Spatial augmented reality, where a projector is used to visualize the virtual content directly on the real object, is therefore interesting for this process, because the designer can assess the content in a well-known environment where he additionally has the variety and flexibility of the virtual world. Spatial augmented reality can significantly reduce the number of required hardware mockups as well as the number of design iterations.

If spatial augmented reality is used in such a process, it is crucial that the designer can rely on the colors which are projected onto the real object. In an ideal situation, these colors should be equally perceived as the corresponding real desired colors which would be used, e.g., to create a hardware mockup. Since the resulting colors of the projected RGB values are influenced by the ambient light, the material, the pose of the projector and its color model, these influences have to be taken into account (Fig. 1). The real objects used in the design process are typically made of a diffuse material with no specular reflections. Otherwise, the position of the observer has also an influence on the perceived color.

However, even if all of these influences are considered, then not all of the desired colors will be projectable, since a projector has a limited range of projectable colors (gamut) and cannot produce negative light. One possible solution to this problem would be to adjust the desired colors till they are projectable. Such an adjustment is only a last option for the design process where decisions are made on basis of the projection. In a first step, it would be more appropriate to modulate the gamut of the projector by changing the position and orientation or by combining the light of multiple projectors.

This paper is an extended version of a previous publication on an interactive visualization technique for spatial augmented reality applications [13]. This paper expands the original work by presenting an extension of the technique for multiple projectors. Furthermore, new results and an application where the technology is used in the interior design process are presented.

1.1 Related Work

The idea of using a projector to augment a real object with additional graphical content was presented by Raskar et al. [17]. In this work, the projected RGB values were radiometrically compensated by considering the surface reflectance of the projection surface and the pose of the projector, but no ambient light or color model was taken into account. Nayar et al. [14] described the relation between the RGB values of the projected image and the resulting colors on the real object by using a radiometric model which describes the relation with a 3×3 color mixing matrix. This matrix was captured for every single projector pixel by a camera. It can be represented unnormalized and expanded with a

- *C. Menk is with the Volkswagen AG, Letterbox 011/1511, 38446 Wolfsburg, Germany. E-mail: christoffer.menk@volkswagen.de.*
- *R. Koch is with the Institut fuer Informatik, Albrechts-Universität zu Kiel, Herman-Rodewald-Str. 3, D-24098 Kiel, Germany. E-mail: rk@mip.informatik.uni-kiel.de.*

Manuscript received 30 Jan. 2012; revised 14 May 2012; accepted 31 May 2012; published online 12 June 2012.
Recommended for acceptance by J. Park, G. Reitmayr, and G. Welch.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-2012-01-0025.
Digital Object Identifier no. 10.1109/TVCG.2012.146.

Fig. 1. (a) A projector is to be used to project virtual content onto a real object whereas the resulting color of the projected content should be equal to the desired color on the chart. Since the resulting color of the projected image is influenced by the ambient light, the material, the pose and color model of the projector, the RGB values of the projected image have to be adjusted to these influences. (b) If the influences are not taken into account then the resulting color on the real object is not equal to the desired color. (c) The resulting color is equal to the desired color if our interactive visualization technique is applied.

fourth row to account for the ambient light [25]. The radiometric model was enhanced by Grossberg et al. [6] by using the fact that the nonlinear projector response is the same for all projector pixels. The described techniques create clipping artifacts if a color is outside the gamut [4] and additionally, no indirect illumination between the projector pixels was considered.

One way to extend the range of projectable colors is to use multiple projectors or a transparent film as proposed by Bimber et al. [3], [4]. Other techniques consider besides the projection surface also the content for the radiometric compensation of the projected image. Wang et al. [22] applied an error metric according to the human perception to compensate gray projection images. A more complex photometric adaptation was proposed by Ashdown et al. [2] where a luminance and chrominance fitting was applied according to certain specified properties. This work considers both, a radiometric model of the system and the content of the image, but was not applicable to real-time applications.

A radiometric compensation which can be applied in real time and considers the global illumination was introduced by Wetzstein and Bimber [24] where the full light transport between projector and camera was computed. They used a global scaling factor which was derived from the average luminance of the projected image. Light transport matrices, between multiple projectors and a camera, and an energy minimization were also applied to change the physical appearance of deteriorated objects [1]. The light transport was captured in both applications with the technique proposed by Sen et al. [18]. The drawback is that it takes several hours to capture all information.

The techniques described so far assume a static scene and projection system. This assumption was relaxed by Fujii et al. [5] where a radiometric compensation technique was proposed, which dynamically adapts the color mixing matrix for changing scenes. Sheng et al. [20] presented a method which interactively minimizes the difference between the desired and actual illumination for a setup with multiple projectors, but no ambient light or color model of the projectors was considered.

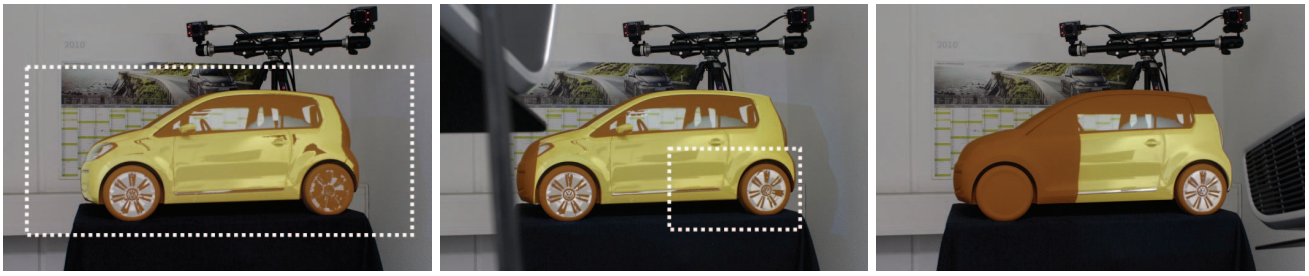
1.2 Our Contribution

The techniques which were described in the previous section use color mixing matrices and the inverse of the nonlinear projector response to compute the RGB values for the projected image which visualizes a desired color on the real object. There are two main drawbacks introduced by using such matrices: 1) the relation between RGB values and the resulting color on the real object cannot be accurately modeled by such matrices, especially for devices which do not exhibit chromaticity constancy [21], [9]. 2) All influences of the ambient light, the material and the pose of the projector are mixed in these matrices and thus, if the model is exchanged or the pose of the projector is changed, then a camera has to be used to capture these matrices again [14], [6], [22], [2], [24]. The second drawback was relaxed by Fuji et al. [5] where a coaxial projector-camera system with a dynamic adaptation of the color mixing matrices was proposed, but the first drawback still remains and has to be relaxed for color critical applications.

In this paper, we present a new approach which extends the idea of using a physically based computation [12] by applying a 3D lookup table (3D LUT) for the color model of the projector. The proposed method can exactly compute the RGB values for the projected image so that the visualized colors on the real object are identical to the real desired colors (Fig. 1). Since a 3D LUT cannot be captured in real time and is different for every single projector pixel, we propose the idea to measure the 3D LUT only once and for one single pixel and then use a physically based computation to create the 3D LUTs for every other pixel. Since the physically based computation determines the influences of the ambient light, the material and the pose of the projector, the 3D LUT does not need to be recaptured even if the projector changes position or the real object is exchanged. Therefore, the proposed method relaxes both mentioned drawbacks.

A 3D LUT is also used for the characterization of monitors [9] or projectors which display content onto a planar screen [21]. However, for these traditional displays, it is sufficient to measure the 3D LUT for a limited number of pixels over the monitor or screen and to interpolate the 3D LUT for all other pixels. This procedure is, therefore, not applicable to our scenario, because the projector displays the content onto a complex real object. Moreover, the procedure is not applicable to a scenario with a moving projector.

For this purpose, we describe an implementation of the approach which directly computes the RGB values for every single projector pixel at interactive rates and does not need any dynamic adaptation in contrast to [5]. Since the projectable colors at a point on the real object do not only depend on the ambient light and the material, but also on the pose of the projector, our proposed method allows us to



(a) (b) (c)

Fig. 2. A desired color can only be exactly represented by a projected RGB value if it is inside the gamut. Areas where desired colors are outside the gamut are marked with a white square in the images (a,b). The gamut does not only depend on the material and the ambient light, but also on the pose of the projector. Our method runs at interactive rates and therefore a new projector pose can be interactively chosen where all desired colors are inside the gamut of the projector as shown in (c).

interactively adjust the range of projectable colors by changing the position and orientation of the projector (Fig. 2).

1.3 Overview

Section 2 describes the calibration and geometric registration of the projector. The idea of using a physically based computation and a 3D LUT to determine the RGB values of the projected image for visualizing desired colors is described in Section 3. This idea is then expanded in Section 4 for enabling interactive rates. The results for using our implemented system in different scenarios and a comparison to alternative approaches are presented in Section 5. The extension to multiple projectors and first results of an application are shown in Section 6. Finally, Section 7 summarizes the results and describes possible future work.

2 Geometric Registration

For the described visualization technique the geometric relationships between the projector, the real object and the observer must be known. This information is used to create an image of the desired virtual content which appears undistorted to the observer, when it is projected onto the real object. Furthermore, the information is necessary to adjust the RGB values of the projected image according to the projector pose. Note that the described geometric registration technique can be replaced with any other technique that computes the projector pose.

The optical tracking system and target[1] used by us are shown in Fig. 1a and are calibrated to the coordinate system of the real object. Therefore, we only need to compute the transformation from the tracking target to the projector. In a first step, we calibrate the projector by computing its intrinsic parameters as described in [12] where a checkerboard pattern is projected onto a planar geometry for varying projector positions (Fig. 3a). The distortion coefficients of the intrinsic parameters are used to derive a predistortion map which is applied with a pixel shader to the projected image. The calibrated projector is in a second step geometrically registered to the real object by shifting a predistorted cross-hair to known 3D coordinates [17], which can be seen in Fig. 3b.

These correspondences between 2D and 3D coordinates are used to compute the pose of the projector to the real object. The pose of the projector with rotation R_P and translation C_P and the corresponding pose of the tracking target (R_{Tar}, C_{Tar}) can then be used to compute the transformation $(R_{Tar\to P}, C_{Tar\to P})$ from tracking target to projector as shown

$$C_{Tar\to P} = R_{Tar}^T \cdot (C_P - C_{Tar}) \qquad R_{Tar\to P} = R_{Tar}^T \cdot R_P. \quad (1)$$

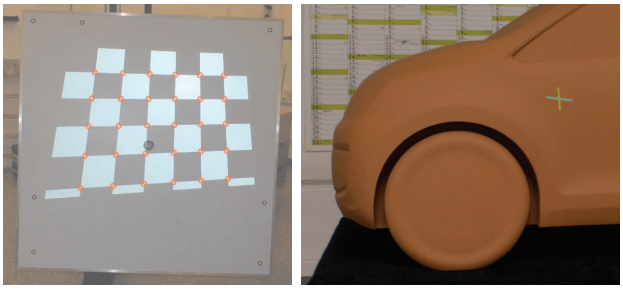
This transformation has only to be computed once and can be used to determine the current pose of the projector (R'_P, C'_P) by using (2) where (R'_{Tar}, C'_{Tar}) is the currently tracked pose of the target on the projector

$$C'_P = C'_{Tar} + R'_{Tar} \cdot C_{Tar\to P} \qquad R'_P = R'_{Tar} \cdot R_{Tar\to P}. \quad (2)$$

The viewing position of the observer can also be tracked with an optical measurement system like the one used in our scenario or determined by placing a calibrated camera into the later viewing position. Since this is a well-discussed topic in research, we do not further describe it in this paper and the interested reader is referred to [4]. An evaluation of geometric registration methods was proposed by Menk et al. [10] and we use their method to evaluate the accuracy of the tracked projector in Section 5.

3 Radiometric Compensation

We want to compute a compensation for the RGB values of the projected image so that the compensated RGB values result in the specified desired colors on the real object. We describe these desired colors by using the device-independent CIE XYZ color space, because we use the rendering

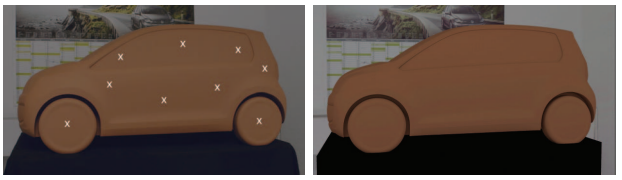


(a) (b)

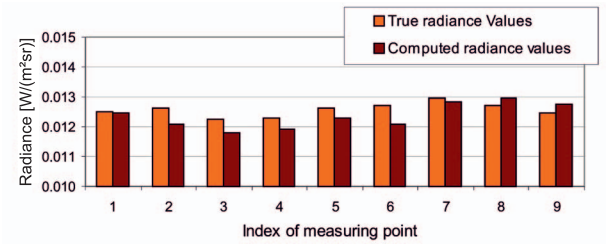
Fig. 3. (a) Calibration of intrinsic parameters. (b) Geometric registration by projecting a predistorted cross-hair to known 3D coordinates.

1. A.R.T. TrackPack: http://www.ar-tracking.de/.

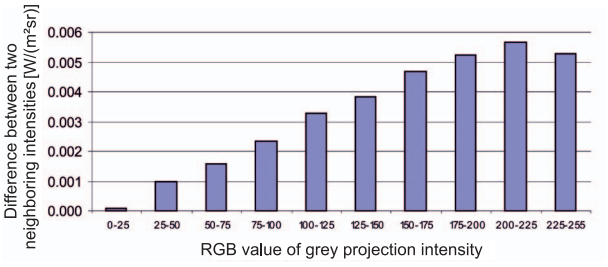
(a)



(b) (c)



(d)



(e)

Fig. 4. (a) Captured HDR image of the ambient light. (b) Image captured with a real camera from the real object. (c) Physically based rendering using the HDR image and the measured material. (d) Comparison between measured and rendered radiance values on nine positions on the real object which are marked with an x in (b). (e) The differences in radiance between two neighboring projection intensities.

software RADIANCE [23]. This physically based rendering software does not compute the complete radiance spectrum for a desired color, but uses three primary color channels. This is less precise than a computation with all spectral values, but spectral data, in contrary to XYZ values, for materials is nowadays not generally available.

The result of the rendering is an image of the virtual content from the viewpoint of the observer. Since the viewpoint of the projector is different to that of the observer, the image has to be transformed so that it appears undistorted to the observer, when it is projected onto the real object. The transformed images for applications where virtual content should be projected onto a real object with known geometry, can be created by using a two-pass rendering approach with projective textures [16]. The resulting images contain for every projector pixel q the value $\mathrm{XYZ}_{des}(q)$ of the desired color on the real object. This section describes how an appropriate RGB value $I(q)$ for a given value $\mathrm{XYZ}_{des}(q)$ of a pixel q can be computed.

For this purpose, the ambient light and the material of the real object have to be taken into account (Section 3.1) and have to be used with the pose of the projector to compute the projectable values at every projector pixel (Section 3.2). These projectable values are represented by a 3D LUT and are used to compute the RGB values for the desired colors inside and outside the gamut (Section 3.3). Note that only if $\mathrm{XYZ}_{des}(q)$ is inside the gamut of the projector, then the computed RGB value is equally perceived as the desired color when it is projected onto the real object as shown in Fig. 1c.

3.1 Material and Ambient Light

In this paper, we consider the real object which is shown in Fig. 4b as a projection surface. This real object is used in a design process of a car and is made of a brown diffuse material which has a color corresponding to that of clay. The spectral reflectance of the surface is determined by taking measurements on different positions on the surface of the real object with a spectrophotometer.[2] The average of these measurements is used as the overall surface reflectance m. The ambient light is measured and integrated into the computation as proposed in [12]: a high dynamic range (HDR) image of the environment is taken and the result is scaled to match the true radiance values of the scene (Fig. 4a). Additionally, the HDR image is mapped onto the local geometry and used as a light source in our rendering. This method allows us to render images which describe the ambient light values XYZ_{amb} on the real object.

It can be seen that there are only small deviations between the measured and rendered radiance values on the real object (Fig. 4d). If we take a look at the distance between the radiance values of two neighboring projection intensities on the real object (Fig. 4e), then it can be seen that the average distance for neighboring intensities above an RGB value of $(25, 25, 25)$ is even greater. Since projection intensities below an RGB value of $(25, 25, 25)$ create radiance values less than $0.003\ \mathrm{W}/(\mathrm{m}^2\mathrm{sr})$, the computed ambient light values can be used for the adjustment of the projection and the deviations to the true radiance values are neglected. This ambient light is also used to render the desired virtual content as shown in Fig. 2.

3.2 Computation of 3D LUTs

The relation between radiance values on the real object and RGB values of the projected image can be described by using a color mixing matrix for every single pixel [14], [6], [5]. As mentioned before, the use of such a matrix assumes that the primaries exhibit chromaticity constancy which is not the case for most projectors as shown in [9] and [21]. Thus, we use more complex 3D LUTs to describe the relation between radiance values on the real object and RGB values of the projected image. These 3D LUTs are built from measurements of intensity combinations of the RGB channels. Since the measurement of values for such a table, e.g., using a $12 \times 12 \times 12$ grid for every pixel, results in a very large number of measurement, this table cannot be

2. X-Rite i1: http://www.xrite.com.

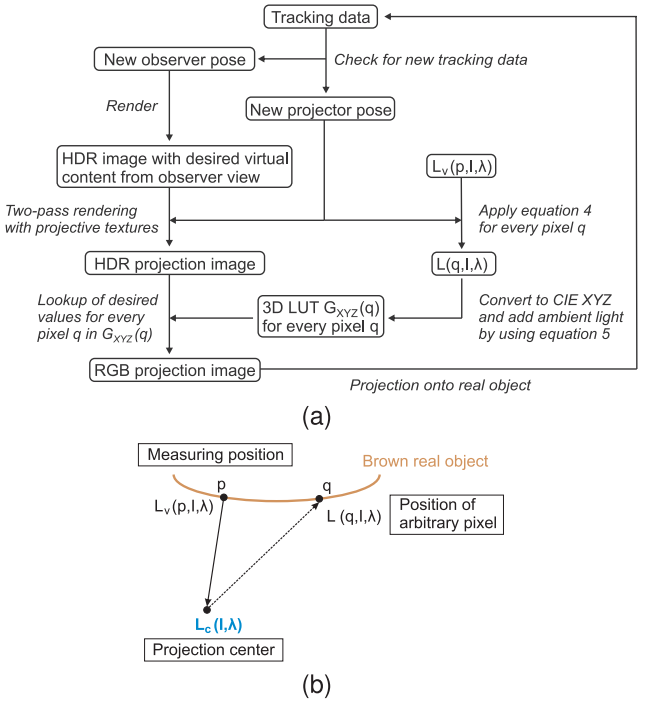


Fig. 5. (a) Idea of using a physically based computation to determine the RGB value for a given desired radiance value. (b) Transformation of measured values.

captured in real time. Our idea is to use one basic 3D LUT which is captured in a calibration step for only one single pixel and a physically based computation to compute the 3D LUTs for all other pixels of the projector by incorporating the material, the ambient light and the pose of the projector. This idea is sketched in Fig. 5a, extended for an interactive deployment in Section 4 and further expanded for multiple projectors in Section 6.

The projector, whose intrinsic parameters are calibrated in advance, is geometrically registered to the real object by using the tracking system. Then, a spectroradiometer is used to measure the radiance spectrum $L_v(p, I, \lambda)$ of different RGB values $I \in \{0, \ldots, 255\}^3$ of the projector at a specific projector pixel p for certain wavelengths λ. The projected values are generated by dividing the RGB intensity axes into arbitrary steps. We use the registered pose of the projector and the geometry of the real object to transform the measurements $L_v(p, I, \lambda)$ from the measuring position to the projection center, which results in the spectrum $L_c(I, \lambda)$ (Fig. 5b). This is shown in (3) where $\vec{n}_p$ denotes the normal of the surface at pixel p, $\vec{r}_p$ is the normalized vector from pixel p toward the real object with distance d_p and π accounts for the fact that $m(p, \lambda)$ is the reflectance of a diffuse material.

$$L_c(I, \lambda) = \frac{L_v(p, I, \lambda) \cdot \pi \cdot d_p^2}{(-\vec{r}_p^T \cdot \vec{n}_p) \cdot m(p, \lambda)}. \quad (3)$$

The measurements are taken without ambient light, because otherwise an additional measurement of the ambient light spectrum has to be taken and subtracted from $L_v(p, I, \lambda)$. The radiance values of the computed ambient light cannot be used for this purpose, because they are represented in a 3D space (Section 3.1).

It is important that the computed values $L_c(I, \lambda)$ are independent of the material, the ambient light and the pose of the projector, because they describe the radiance spectrum for the RGB value I in the projection center. If the projector moves to another position or the real object is exchanged, then these values do not change and the values $L_c(I, \lambda)$ do not need to be measured again. However, $L_c(I, \lambda)$ describes the property of the projector and thus, has to be measured again after a certain time interval to account for the aging of the projector lamp. The values $L_c(I, \lambda)$ can now be used to compute the spectrum $L(q, I, \lambda)$ for arbitrary projector poses and projector pixels q on the real object by extending (3) to (4)

$$L(q, I, \lambda) = \frac{L_c(I, \lambda) \cdot \left(-\vec{r}_q^T \cdot \vec{n}_q\right) \cdot m(q, \lambda)}{\pi \cdot d_q^2}. \quad (4)$$

So far, we have computed the projectable spectrum for different RGB values of the projector at specific positions on the real object. These values are now converted into the CIE XYZ color space, because the values of the desired color $\mathrm{XYZ}_{des}(q)$ and the ambient light $\mathrm{XYZ}_{amb}(q)$ are not in the spectral domain. Since the projectable values also depend on the present ambient light, these values have to be added (5). The values for the ambient light were computed in Section 3.1 and $\overline{x}$, $\overline{y}$, and $\overline{z}$ are the CIE standard observer functions

$$\begin{aligned} X(q, I) &= \int_\lambda \overline{x}(\lambda) L(q, I, \lambda) d\lambda + X_{amb}(q), \\ Y(q, I) &= \int_\lambda \overline{y}(\lambda) L(q, I, \lambda) d\lambda + Y_{amb}(q), \\ Z(q, I) &= \int_\lambda \overline{z}(\lambda) L(q, I, \lambda) d\lambda + Z_{amb}(q). \end{aligned} \quad (5)$$

The computed values $(X(q, I), Y(q, I), Z(q, I))$ for varying RGB values I of the projected image represent a 3D grid of XYZ values for every projector pixel q on the real object. These grids are different for every pixel q and represent the projectable values. We denote these grids as $G_{\mathrm{XYZ}}(q)$. Additionally, we have for every projector pixel a desired color with the value $\mathrm{XYZ}_{des}(q) = (X_{des}(q), Y_{des}(q), Z_{des}(q))$.

3.3 Computation of RGB Values for Desired Colors

The desired colors, which are represented by the values $\mathrm{XYZ}_{des}(q)$ for the pixels q of the HDR projection image, are only producible by an RGB value of the projector, if they are included in the computed 3D grid $G_{\mathrm{XYZ}}(q)$. One possible solution to this problem would be to adjust all desired colors till they are projectable which is only a last option for the design process in which decision are made on basis of the projection.

Therefore, we developed the following procedure: in a first step, the desired colors which are inside the gamut are computed (Section 3.3.1), because this colors can be exactly represented by an RGB value of the projector. If not all desired colors are inside the gamut, then the designer can interactively change the pose of the projector to adjust the gamut and to maximize the amount of projectable colors which can be seen in Fig. 2. If not all of the desired colors are projectable after this step, then the designer can allow to compute the RGB values for colors outside the gamut with

the method proposed in Section 3.3.2. This method preserves colors that are inside the gamut. For colors outside the gamut, an RGB value is determined which has a minimum perceptual distance, in terms of the ΔE^*-metric, to the desired color, when it is projected onto the real object.

Since values inside the gamut are preserved, this method will lead to a change toward the relative lightness and chromatic overall appearance of the desired virtual content. Therefore, we implemented an adjustment of all desired values before the values inside or outside the gamut are computed. This adjustment was presented in [11] and was developed by conducting a user study where automotive design experts were asked to adjust colors outside the gamut. The proposed method transforms the values $\mathrm{XYZ}_{des}(q)$ of the desired colors into the CIE LAB color space and adjusts the lightness value with a user defined global scaling S_g and additive factor A_g. This is shown in (6) where $\overline{L}^*_{des}$ is the average and $L'^*_{des}(q)$ the adjusted lightness value

$$L'^*_{des}(q) = (L^*_{des}(q) - \overline{L}^*_{des}) \cdot S_g + \overline{L}^*_{des} + A_g. \tag{6}$$

Note that the desired colors can also be adjusted by methods proposed in [7] or [2], but it is crucial that the designer is aware of the adjustment he makes toward the desired values.

3.3.1 Handling Desired Colors Inside the Gamut

The computation of RGB values is described for our new method using a 3D LUT. The existing method using a color mixing matrix which is adapted for our physically based computation is also explained, because it is compared to our new method in Section 5.2.

3D LUT. If a color is inside the gamut, then its $\mathrm{XYZ}_{des}(q)$ value is included in the computed 3D grid $G_{\mathrm{XYZ}}(q)$ of a pixel q. The coordinates of a value inside the grid $G_{\mathrm{XYZ}}(q)$, can be determined by using tetrahedral division and barycentric coordinates after the model of Hung [8]. The tetrahedral division is applied to the 3D grid $G_{\mathrm{XYZ}}(q)$ of XYZ values $(X(q,I), Y(q,I), Z(q,I))$. Every point of this grid corresponds to a certain RGB value $I = (R, G, B)$ and a tetrahedron of this grid is formed by four values $(X(q,I_i), Y(q,I_i), Z(q,I_i))$ with $i \in \{0,1,2,3\}$. The barycentric coordinates α, β, and γ of a value $\mathrm{XYZ}_{des}(q)$ corresponding to a tetrahedron can be computed by using (7) where $X_i = X(q,I_i)$, $Y_i = Y(q,I_i)$, and $Z_i = Z(q,I_i)$

$$\begin{pmatrix}\alpha\\ \beta\\ \gamma\end{pmatrix} = M^{-1}_{XYZ} \cdot \begin{pmatrix}X_{des}(q) - X_0\\ Y_{des}(q) - Y_0\\ Z_{des}(q) - Z_0\end{pmatrix}, \tag{7}$$

$$M_{XYZ} = \begin{pmatrix}X_1 - X_0 & X_2 - X_0 & X_3 - X_0\\ Y_1 - Y_0 & Y_2 - Y_0 & Y_3 - Y_0\\ Z_1 - Z_0 & Z_2 - Z_0 & Z_3 - Z_0\end{pmatrix}. \tag{8}$$

If the barycentric coordinates α, β, and γ are inside $[0,1]$, then the value of the desired color is inside the tetrahedron and (9) can be used to determine the final RGB value $I_{des}(q) = (R_{des}(q), G_{des}(q), B_{des}(q))$ for the projector pixel q

$$I_{des}(q) = M_{RGB} \cdot \begin{pmatrix}\alpha\\ \beta\\ \gamma\end{pmatrix} + \begin{pmatrix}R_0\\ G_0\\ B_0\end{pmatrix}, \tag{9}$$

$$M_{RGB} = \begin{pmatrix}R_1 - R_0 & R_2 - R_0 & R_3 - R_0\\ G_1 - G_0 & G_2 - G_0 & G_3 - G_0\\ B_1 - B_0 & B_2 - B_0 & B_3 - B_0\end{pmatrix}. \tag{10}$$

Note that the interpolation between the values of the grid $G_{\mathrm{XYZ}}(q)$ is an approximation, because the values are nonlinearly related. The error caused by this approximation gets smaller as more measurements are made which results in a finer resolution of the grid $G_{\mathrm{XYZ}}(q)$.

Matrix. The computation of the RGB values $I_{des}(q)$ using a matrix M and the inverse of the nonlinear projector response f_p is described in (11). The nonlinear response can be determined from the measurements of gray intensities. This function is assumed to be monotonic and can thus be inverted by using a 1D LUT. The following intensities are used in the equation: $I_{rmax} = (255, 0, 0)$, $I_{gmax} = (0, 255, 0)$, $I_{bmax} = (0, 0, 255)$, and $I_{black} = (0, 0, 0)$. Note that this approach assumes a fixed relation of the different projector channels with matrix M

$$I_{des}(q) = f_p^{-1}[M^{-1} \cdot \begin{pmatrix}X_{des}(q) - X(q, I_{black})\\ Y_{des}(q) - Y(q, I_{black})\\ Z_{des}(q) - Z(q, I_{black})\end{pmatrix}], \tag{11}$$

$$M = \begin{pmatrix}X_{bc}(q, I_{rmax}) & X_{bc}(q, I_{gmax}) & X_{bc}(q, I_{bmax})\\ Y_{bc}(q, I_{rmax}) & Y_{bc}(q, I_{gmax}) & Y_{bc}(q, I_{bmax})\\ Z_{bc}(q, I_{rmax}) & Z_{bc}(q, I_{gmax}) & Z_{bc}(q, I_{bmax})\end{pmatrix}, \tag{12}$$

$$\begin{pmatrix}X_{bc}(q,I)\\ Y_{bc}(q,I)\\ Z_{bc}(q,I)\end{pmatrix} = \begin{pmatrix}X(q,I)\\ Y(q,I)\\ Z(q,I)\end{pmatrix} - \begin{pmatrix}X(q,I_{black})\\ Y(q,I_{black})\\ Z(q,I_{black})\end{pmatrix}. \tag{13}$$

3.3.2 Handling Desired Colors Outside the Gamut

For a value $\mathrm{XYZ}_{des}(q)$ of the desired color outside of the grid $G_{\mathrm{XYZ}}(q)$, no exact RGB value for the projected image can be computed. We visualize such colors by choosing an RGB value which has projected onto the real object the smallest ΔE^* distance toward the desired color. For this purpose, we convert the value of the desired color $\mathrm{XYZ}_{des}(q)$ to the CIE LAB color space which results in the values $LAB(q,I) = ((L^*(q,I), a^*(q,I), b^*(q,I)))$, where (L^*, a^*, b^*) are the coordinates of the CIE LAB color space. The value of the desired color is also transformed to this color space and is denoted as $LAB_{des}(q) = (L^*_{des}(q), a^*_{des}(q), b^*_{des}(q))$. Since we want to find an RGB value for the projector pixel q which has projected onto the real object the smallest distance to the desired value $LAB_{des}(q)$, we have to solve

$$\min_I \Delta E^*(q,I) = \min_I \|LAB(q,I) - LAB_{des}(q)\|_2. \tag{14}$$

Since the CIE LAB color space has the advantage that it is perceptually uniform, the euclidean distances between the converted values can be used to determine the RGB value with the smallest perceptual distance. Note that colors inside the gamut are preserved and computed with the method described in the previous section.

4 Implementation for Interactive Visualization

This section presents an efficient implementation of the idea, which uses 3D lookup tables for every single projector

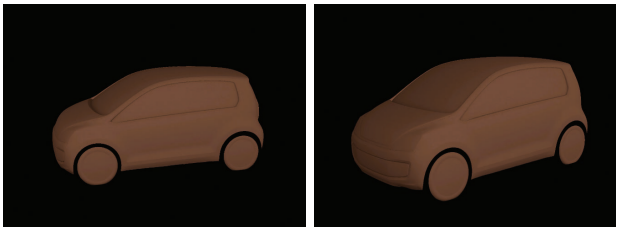


Fig. 6. Interactive visualization of the ambient light for two projector poses by using offline rendered images as shown in Fig. 4c.

pixel to compute the RGB values of the projected image for arbitrary desired colors.

4.1 Ambient Light

The ambient light is used to compute the projectable values at every projector pixel as shown in (5). We take advantage of the fact that the real object is made of a diffuse material. This means that the ambient light for a specific point on the real object is independent of the viewing position. Therefore, we render five high-dynamic range images in an offline rendering step which cover the views from the right, front, left, back and top. An example of such a rendering can be seen in Fig. 4c. An interactive visualization is achieved by using these images and the technique of projective textures to compute the ambient light $\mathrm{XYZ}_{amb}(q)$ of an arbitrary projector pixel q. Two results of such a visualization are shown in Fig. 6. Note that also more offline renderings from various viewpoints can be used to compute the ambient light for a more complex real object than the one used in our scenario. In this case, texture baking should be used instead of projective textures so that a computation at interactive rates is still possible.

4.2 Colors Inside the Gamut

In a second step, the RGB values for desired colors with values $\mathrm{XYZ}_{des}(q)$ inside the gamut are computed. The idea is to apply the transformation from (4) directly to the values $\mathrm{XYZ}_{des}(q)$ of the desired color instead of computing a grid $G_{\mathrm{XYZ}}(q)$ for every pixel q as proposed in Section 3.2. This results in the adjusted values $\mathrm{XYZ'}_{des}(q)$ as shown

$$\mathrm{XYZ'_{des}(q)} = (\mathrm{XYZ_{des}(q)} - \mathrm{XYZ_{amb}(q)}) \cdot \mathrm{F(q)}, \tag{15}$$

$$F(q) = \frac{\left(-\vec{r}_p^{\,T} \cdot \vec{n}_p\right) \cdot m(p) \cdot d_q^2}{\left(-\vec{r}_q^{\,T} \cdot \vec{n}_q\right) \cdot m(q) \cdot d_p^2}. \tag{16}$$

This procedure has the advantage that for all transformed values of the desired colors the same 3D LUT can be used and thus, can be efficiently stored in the memory of CPU or GPU. Note that the storage of a 3D LUT for every single pixel would require a lot of memory, e.g., using a $12 \times 12 \times 12$ grid, a resolution of $1{,}400 \times 1{,}050$ pixel and a precision of 16 bit, would require 5.1 GB. And these tables would all have to be updated if a component of the scenario is modified which is not the case for our method.

The information for (15) is computed with our ray tracing software which uses the NVIDIA OptiX engine [15]. We now propose an efficient way how the corresponding RGB values for the transformed desired values $\mathrm{XYZ'}_{des}(q)$ can be computed by implementing the equations of Section 3.3 on the GPU and CPU. We first store all values $\mathrm{XYZ'}_{des}(q)$ for valid pixels q in the $validDesiredValues$ list where valid means that the corresponding projector pixel lies on the real object. These valid values and the 3D LUT are then used in our proposed Algorithm 1 which computes the final RGB values of the projected image. The idea of the algorithm is that the computation steps for every single pixel q are done in parallel on the GPU while coordination work, which is equal for all pixels q, is done on the CPU at negligible costs. The bounding box of each subcube in the 3D LUT is computed in advance. Each part of the algorithm which is computed on the GPU is denoted with the keyword *in parallel* and is implemented by using the NVIDIA CUDA Framework.

Algorithm 1. Interactive computation of the RGB values

```
procedure ComputeRGBValues (
    in validDesiredValues:list; 3dlut:3DLUT
    out RGBValues:list)
begin
  for each subcube c in 3dlut do
    // Check if values of desired colors are inside
    // bounding box of c
    valueIsInside ← false
    for each value v in validDesiredValues in parallel
    do
      if v is inside bounding box of c then
        valueIsInside ← true
      end if
    end for
    // Check if at least one value of a desired color
    // was inside c
    if not valueIsInside then
      continue
    end if
    Divide subcube c in tetrahedrons t
    for each tetrahedron t in subcube c do
      Create matrices M_XYZ and M_RGB
      Invert M_XYZ
      // Compute RGB value
      for each value v in validDesiredValues in parallel
      do
        if v is outside bounding box of c then
          return
        end if
        Compute barycentric coordinates α, β, γ for
        value v and tetrahedron t with (7)
        if α, β, γ ∈ [0, 1] then
          Compute RGB value I_des with (9)
          RGBValues[v] ← I_des
        end if
      end for
    end for
  end for
end
```

4.3 Colors Outside the Gamut

For handling colors outside the gamut of the projector we proposed to find the RGB values which have projected onto the real object a minimum perceptual distance to the corresponding desired color. This problem corresponds to

TABLE 1
Results for Computing an RGB Value Which Results Projected onto a Real Object in a Specific Desired Color

	Method	Value								Overall
		1	2	3	4	5	6	7	8	
Average ΔE^*	Compensated (3D LUT)	1.39	1.56	1.76	1.03	1.45	1.17	1.13	1.14	1.33
	Compensated (Matrix)	5.55	5.67	5.37	5.82	6.07	4.54	5.75	5.93	5.63
	Uncompensated	6.46	13.64	5.10	8.01	10.01	8.95	6.79	9.61	8.57
Maximum ΔE^*	Compensated (3D LUT)	2.30	2.66	2.07	1.88	2.17	1.97	1.84	2.08	2.66
	Compensated (Matrix)	6.70	6.41	6.55	6.90	7.49	6.49	6.89	6.83	7.49
	Uncompensated	20.92	47.53	16.45	27.13	34.33	31.96	22.71	34.08	47.53

The RGB values for every desired color are computed and projected from 11 different poses for each method.

the task of finding a nearest neighbor in a given point cloud. First of all, we can make the crucial observation that the nearest neighbor can only be a value at the boundary of the computed 3D LUT. Since this 3D LUT only represents a grid of values, we take all values at the boundary and use these values to interpolate the complete boundary of the 3D LUT to get all XYZ values of the corresponding RGB values. This results in overall $256 \cdot 256 \cdot 6$ values which can be precomputed offline. We use a binary space partitioning technique to find the nearest neighbor and decided to build a kd-tree of the interpolated XYZ values before the application is started. This tree is stored on the graphics cards and is used to compute the XYZ values which have a minimal perceptual distance to the desired colors. The corresponding RGB values are then stored in the projected image at interactive rates.

5 Results

We use for all experiments a Canon XEED SX6 projector which has a resolution of $1{,}400 \times 1{,}050$ pixels. All computations are performed on an Intel Core 2 Quad Q9550 processor and a NVIDIA GeForce GTX 580 graphics card.

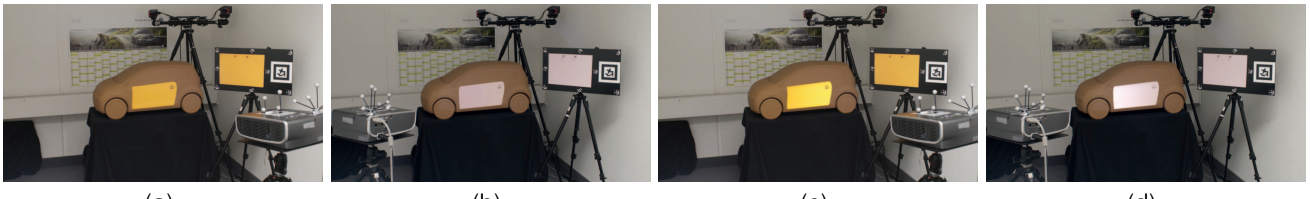
5.1 Accuracy of Geometric Registration

We evaluate the accuracy of our geometric registration by using the technique which was proposed by Menk et al. [10]: a virtual 3D model, with 12 feature points at specific positions, is projected onto the real object and the real 3D coordinates of the projected features are measured with a camera. Since the feature points are virtually aligned on top of the real object the measured centers of the projected feature points can be directly compared to their virtual position. The deviation between the virtual and real positions are computed for each feature point and averaged for all 11 projector poses. Images from two of these projector poses are shown in Fig. 8. The results are listed in Table 2 and it can be seen that the virtual content can be projected with an average deviation of 2.0 mm onto the real object.

The size of the projector pixels varies over the complete real object from 0.84 mm to 1.4 mm, since it depends on the position and orientation of the projector to the real object and therefore, differs between every single projector pixel and projector pose. If this overlay accuracy is not sufficient to fulfill the geometric requirements of a specific scenario, then this result may be improved by using more than two cameras for the optical tracking system. Another possibility would be to geometrically register the projector, e.g., by shifting a predistorted cross-hair to known 3D coordinates [17], [12], if a final pose was identified with the proposed method. Such a registration technique achieves a pixel-precise alignment, but cannot be applied in real time. An overview and evaluation of the overlay accuracy of different registration techniques can be found in [10].

5.2 Accuracy of Projected RGB Values

In this section, the accuracy of our physically based 3D LUT approach to visualize colors inside the gamut is evaluated. As mentioned before, the gamut of the projector depends on the material, the ambient light and the projector pose and thus, we determine eight, randomly selected, desired colors which are inside the gamut for 11 arbitrarily chosen projector poses around the real object. We move the projector to all of these poses and use a spectroradiometer to measure the result of the eight computed and projected RGB values on the real object. The comparison between the measured and desired values is shown in Table 1. It can clearly be seen that our proposed method achieves a high accuracy, since the deviation of ΔE^* is always less than 2.3 which is denoted as the just noticeable difference [19]. Therefore, the projected RGB values are equally perceived as the desired colors which can be seen in Fig. 1 and Fig. 7. Also the maximum deviations are except for one value below this threshold. Such a result is not achieved if no compensation or a color mixing matrix is used to compute the RGB values of the projected image for the desired colors, because the ΔE^* is always greater than 4.5. A comparison of the techniques for real objects with



(a) (b) (c) (d)

Fig. 7. (a,b) A door is projected onto the real object which has a color equivalent to the desired color on the chart, if the projection image is computed with the proposed efficient implementation of the radiometric compensation using a 3D LUT ($12 \times 12 \times 12$). (c,d) The color of the projected door is not equal to the desired color if the influences of the ambient light, the material and the projector pose are neglected.

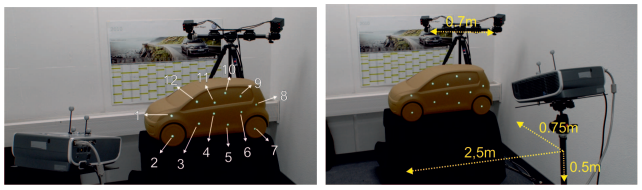


Fig. 8. The accuracy of the projected content on the real object is verified by projecting 12 feature points for 11 different projector poses and comparing their coordinates.

different materials and an evaluation in a user study can be found in [11].

5.3 Interactive Visualization

We evaluate our proposed method in three different scenarios which are shown in Fig. 9 and compare it to implementations where the computations are entirely performed on the CPU or GPU. In each of these scenarios the projector is moved around the real object for overall 120 different frames and the projected image for the specified observer position is computed. The RGB values of the projected image are computed for colors inside the gamut with tetrahedral interpolation (Section 4.2) and for colors outside the gamut by using a kd-tree (Section 4.3).

The results are shown in Table 3. It can be seen that our proposed method achieves interactive rates from 3 to 7 fps for all of the three scenarios and even achieves acceptable frame rates if additionally colors outside the gamut are computed with a kd-tree. The implementation of our approach which is entirely performed on the GPU or CPU has a 1.5 to 10 times lower frame rate. The overall times include the ray tracing process for determining all parameters like the ambient light of (15), the predistortion of the projected image and memory management between GPU and CPU. Note that the HDR observer image with the values of the desired colors is equal for all projector poses and has therefore only to be recomputed, if the position of the observer changes. The rendering of a new HDR observer image will cause additional computational time which depends on the used rendering system.

5.4 Discussion and Limitations

Fig. 10 shows results where different virtual content is projected onto the real object. The results are captured with a camera and show the effect of different computations and adjustments. It can be seen that the adjustment is only a last option for the design process, especially if decisions about the color are made on basis of the projection (Fig. 10, second column). In this scenario, it would be more appropriate to change the projector pose till all colors are inside the gamut of the projector (Fig. 10, fourth column). The computation of

TABLE 2
Accuracy of Geometric Registration: Deviation of Each Feature Point Which Is Averaged over 11 Projector Poses

	Deviations for feature points						Overall
Average [mm]	1.8	1.8	1.8	1.9	2.0	2.1	2.0 mm
	2.4	2.2	2.1	2.1	1.9	1.8	
Maximum [mm]	2.9	3.7	2.8	2.9	3.8	3.9	3.9 mm
	3.6	3.4	3.7	3.1	3.0	3.9	

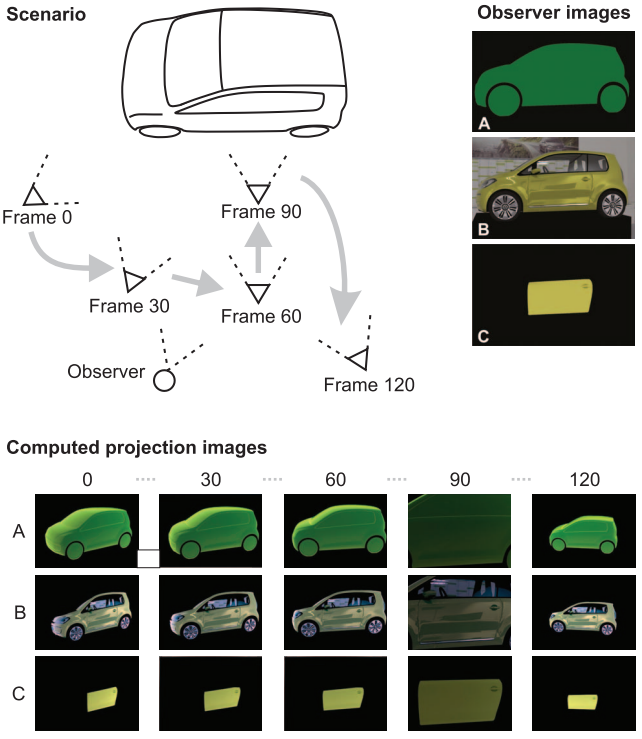


Fig. 9. Experiment for evaluating our proposed interactive visualization technique: the projector is moved for every frame to a new position and projects three images which are computed from the observer images with our proposed method. The observer images are HDR images for the three scenarios A, B, and C which are tone-mapped for visualization in this paper.

RGB values of the projected image for colors outside the gamut works especially well for colors which have only a small perceptual distance to the desired colors. For colors with a larger distance a slight color shift can occur, e.g., for the color of the car painting near the back light (Fig. 10, third column). The drawback of moving the projector till most of the desired colors are inside the gamut is that the projection may not cover the complete real object which could be resolved by using multiple projectors (Section 6). Furthermore, it may be difficult to find the optimal projection

TABLE 3
Average Computation Times for Using Different Algorithms for Computing Colors inside the Gamut with Tetrahedral Interpolation (Section 4.2) and Colors outside the Gamut by Using a kd-Tree (Section 4.3)

	Method	Tetrahedral interpolation	kd-tree	Overall	Fps
A	CPU	0.78s	-	0.84s	1.19
	GPU	1.85s	-	1.91s	0.52
	Our method	0.16s	-	0.23s	4.43
	Our method	0.16s	0.09s	0.32s	3.13
B	CPU	0.91s	-	0.97s	1.03
	GPU	2.69s	-	2.76s	0.36
	Our method	0.22s	-	0.28s	3.61
	Our method	0.22s	0.12s	0.40s	2.50
C	CPU	0.19s	-	0.25s	4.03
	GPU	0.23s	-	0.29s	3.43
	Our method	0.08s	-	0.14s	7.04
	Our method	0.08s	0.05s	0.19s	5.11

The computations are performed for the three different scenarios which are shown in Fig. 9.

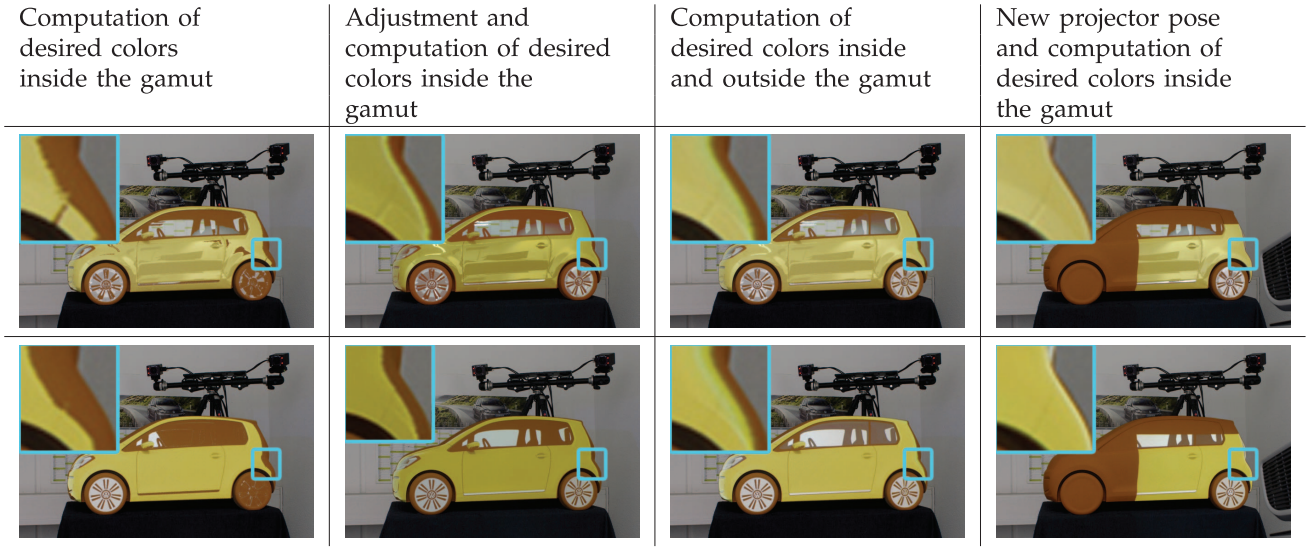


Fig. 10. Two virtual contents are projected onto the real object by using different methods. The results are captured with a camera and the influences of only computing the desired colors inside the gamut (first column), using an additional adjustment of the desired colors before the projected images are computed (second column) or computing also the desired colors outside the gamut (third column) can be seen. All desired colors are inside the gamut by changing the pose of the projector (fourth column). Note that only desired colors which are inside the gamut can be exactly represented by an RGB value of the projected image.

position where most of the desired colors are inside the gamut. Therefore, future work should incorporate a computation of such a position and a guidance for the designer.

The range of projectable colors can also be extended by choosing a darker ambient light or a new material for the real object. Since the real objects are viewed under the influence of ambient light in the design process, an interesting topic of study would be to find a material which optimizes the number of projectable colors. However, there will often be values which are outside the gamut of the projection and therefore in [11] an user study was conducted to see how design experts working in the automotive industry would adjust such values. Altogether, it is important that the designer is aware of these effects and can decide based on the current scenario if he only wants to assess colors which can be exactly represented or also adjusted colors.

Section 5.2 showed that the proposed 3D LUT approach is able to reproduce desired colors on a real object with a high accuracy. In contrast to the other techniques which use a color mixing matrix or no compensation, the 3D LUT approach is thus suitable for color critical spatial augmented reality applications like the design process. Depending on the requirements of the considered scenario, also a smaller 3D LUT, e.g., consisting of $5 \times 5 \times 5$ values or even a matrix (Section 3.3.1), may be sufficient which would reduce the necessary measurement time. Our proposed method cannot be applied to scenarios where the object geometry and ambient light is unknown, e.g., if a video is projected onto an arbitrary surface. In this case, the approach using camera-based color mixing matrices as proposed by Nayar et al. [14] should be used.

5.5 Interior Design Process

The application which was presented throughout this paper showed the projection of exterior designs. One area of the design process, which was so far not considered, is the interior design. An example for an application in this part of the design process is shown in Fig. 11. As for the exterior design, the virtual content has to be projected with a high geometric and visual quality to meet the requirements of an automotive design process. Since the radio and other components are viewed from a short distance, the mentioned

Fig. 11. Two projectors are used to project different customizations and contents onto a real object in the interior design process: (a) Setup; (b,c) Projection of two different customizations of a *Passat B6* onto the real object; (c) An iPad application can be used to switch interactively between different variants, customizations and colors.

requirements are mandatory, but can be met by using the visualization techniques of this paper. The visual quality is further increased by using two projectors for displaying the virtual content onto the real object. Since the proposed method does so far not consider an overlap of the projection, the projectors simply visualize disjunctive parts of the cockpit. Furthermore, an iPad application was implemented for this scenario so that the designer is able to switch between different variants and customizations from the driver seat (Fig. 11c).

6 Extension to Multiple Projectors

To increase the range of projectable colors, this section describes the extension of the proposed 3D LUT method to multiple projectors. For this purpose, let us consider a 3D point Q which is illuminated without loss of generality by two projectors. The corresponding pixels of the projectors are denoted with q_1 for the first and with q_2 for the second projector. Furthermore, let p_i be the projector pixel where the basic 3D LUT of the projector $i \in \{1,2\}$ was measured. All other used notations were introduced in Section 3.2. According to (15), we can define for each of the projectors $i \in \{1,2\}$ a form factor $F_i(q)$ which converts the XYZ value of a desired color for an arbitrary pixel q from the actual tracked projector pose to the position where the 3D LUT of projector i was measured

$$F_i(q) = \frac{(-\vec{r}_{p_i}^{T} \cdot \vec{n}_{p_i}) \cdot m(p_i) \cdot d_q^2}{(-\vec{r}_q^{T} \cdot \vec{n}_q) \cdot m(q) \cdot d_{p_i}^2}, \tag{17}$$

$$F_i(q)^{-1} = \frac{1}{F_i(q).} \tag{18}$$

This conversion is necessary so that Algorithm 1 can be used to compute the RGB values for arbitrary pixels by using the same 3D LUT. However, in contrast to the method in Section 4, we have to consider the light of two projectors. First of all, it is checked, if the 3D point Q is illuminated by both projectors. If this is not the case, then the proposed method of Section 4 can be used and we are finished. If both projectors illuminate the 3D point Q, then we have to find two RGB values $I_1(q_1)$ and $I_2(q_2)$ which create the value $\mathrm{XYZ}_{des}(Q)$ of the desired color on the real object.

In a first step, the ambient light $\mathrm{XYZ}_{amb}(Q)$ is computed with the method described in Section 4.1 and subtracted from the value of the desired color as shown

$$\mathrm{XYZ'}_{des}(Q) = \mathrm{XYZ}_{des}(Q) - \mathrm{XYZ}_{amb}(Q). \tag{19}$$

In a second step, we compute the black level $\mathrm{XYZ}_{black}(q_2)$ of the second projector which is present on the 3D point Q. This can be done by converting the measured spectrum $L_v(p_2, I_{black}, \lambda)$ for the projection intensity $I_{black} = (0,0,0)$ to the CIE XYZ color space and multiplying the resulting XYZ value with the form factor $F_2(q_2)^{-1}$. Note that the form factor $F_2(q_2)^{-1}$ converts the XYZ value of the measured black level from the measuring position of the 3D LUT to the tracked pose of the second projector. The black level and the ambient light are then used to compute the value $XYZ''_{des}(Q)$ as shown

$$XYZ''_{des}(Q) = \mathrm{XYZ'}_{des}(Q) - \mathrm{XYZ}_{black}(p_2) \cdot F_2(q_2)^{-1}. \tag{20}$$

To compute an RGB value which creates $XYZ''_{des}(Q)$ on the real object, we have to multiply the value with the factor $F_1(q_1)$ (21). This transformation is necessary so that Algorithm 1, the measured 3D LUT of the first projector at pixel p_1 and the value $XYZ''_{des}(p_1)$ can be used to compute the RGB value for the desired color

$$XYZ''_{des}(p_1) = XYZ''_{des}(Q) \cdot F_1(q_1). \tag{21}$$

If the value $XYZ''_{des}(p_1)$ is inside the 3D LUT of the first projector, then the final RGB value is found and can be displayed by the first projector while the second projector displays a black projection intensity.

If the value is not inside the 3D LUT, then we must try to combine the RGB values of both projectors to create the desired color on the real object. For this purpose, we use the kd-tree which was so far used to find an RGB value with a minimal perceptual distance in Section 4.3. This time, we use the kd-tree to find a value $\mathrm{XYZ}_{I_1}(q_1)$ which can be created by the first projector on the real object and which is less than $\mathrm{XYZ'}_{des}(Q)$. To find the value $\mathrm{XYZ}_{I_1}(q_1)$, we multiply $\mathrm{XYZ'}_{des}(Q)$ with the form factor $F_1(q_1)$ and search for a corresponding RGB value I_1 with the value $\mathrm{XYZ}_{I_1}(p2)$. Note that it is important that we use $\mathrm{XYZ'}_{des}(Q)$ for this computation, because the second projector should display an RGB value greater than zero. The remaining value $\mathrm{XYZ'}_{rem}(Q)$ of the desired color, which should be created by the second projector, is computed as shown

$$\mathrm{XYZ'}_{rem}(Q) = \mathrm{XYZ'}_{des}(Q) - \mathrm{XYZ}_{I_1}(p2) \cdot F_1(q_1)^{-1}. \tag{22}$$

An RGB value I_2 for the remaining value $\mathrm{XYZ'}_{rem}(p_2)$ (23) is again found by using Algorithm 1 with the 3D LUT of the second projector.

$$\mathrm{XYZ'}_{rem}(p_2) = \mathrm{XYZ'}_{rem}(Q) \cdot F_2(q_2). \tag{23}$$

If an RGB value $I_2(q_2)$ is found, then it can be used to create the remaining value $\mathrm{XYZ'}_{rem}(Q)$ on the real object. Therefore, the computed RGB values $I_1(q_1)$ and $I_2(q_2)$ create combined the desired color on the real object.

If no corresponding RGB value $I_2(q_2)$ can be found, then we have two options: 1) both projector pixels are set to a black projection intensity or 2) an RGB value with a minimal perceptual distance to the value $\mathrm{XYZ'}_{rem}(Q)$ of the desired color is computed by using the method described in Section 4.3. If decisions about the color are made on basis of the projection, then this choice should be up to the designer.

6.1 Results

The scenario for the final result is shown in Fig. 12a. Two projectors are used to display one desired color onto the real object by using the method proposed in the previous section. The desired color is created on the real object by overlaying the projected RGB values of both projectors as seen in Fig. 12.

According to Section 5.2, we also evaluate the accuracy of our approach by projecting ten randomly, selected desired colors which are inside the gamut of both projectors. We achieved an average ΔE^* of 1.35 and a

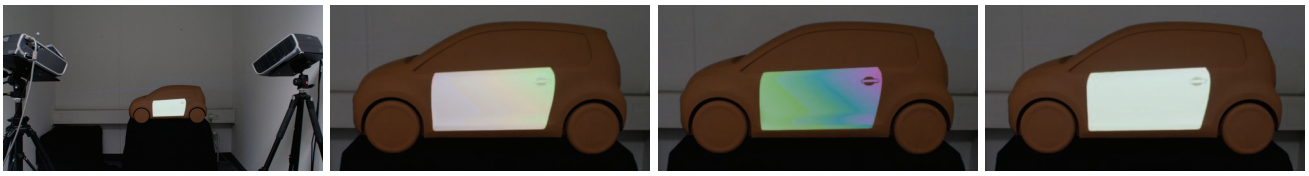


Fig. 12. (a) Two projectors are used to project a color which is equivalent to a desired color onto the real object; (b) The projected colors of the first and (c) of the second projector result in (d) the final desired color.

maximum of 2.38. This result is comparable to the results of Section 5.2. Since we use two projectors, the deviation would be higher if we would move the projector to different positions around the real object as in the previous experiment so that the measuring position of the 3D LUT is different to the actual pose of the projector. However, the result showed that the method is suitable for color critical applications, since the average ΔE^* is below the just noticeable difference of 2.3. Since optimal poses for multiple projectors, so that most of the desired colors are inside the overlapping gamuts, can hardly be found by moving the projectors manually, these positions should be computed. This computation can be done by using the information of our physically based computation and offers an interesting topic of study.

6.2 Limitations

The presented method which computes the RGB values for multiple projectors runs at interactive rates, but has the drawback that it does not check all combination of RGB values. Therefore, there may be situations where a desired color can be created by two RGB values on the real object, but these RGB values are not found by the algorithm. The algorithm could therefore be formulated as an optimization problem so that such RGB values are also found. However, such a problem can hardly be computed at interactive rates for all projector pixels which were the scope of our algorithm. It is important that if our algorithm finds two RGB values, then they are optimal in such way that they are able to create the desired color on the real object with a high accuracy.

Another limitation is that the proposed solution has the assumption that the projector pixels q_1 and q_2 overlap with their full size which is generally not the case. Therefore, the method has to be extended so that also neighboring projector pixels of q_1 and q_2 are taken into account. This problem offers an interesting topic of study, because the projected image of both projectors should generate a visualization with a high quality. Therefore, a simple interpolation technique cannot be applied, because the complete visualization would get blurred.

7 Conclusions and Future Work

We presented, in this paper, a new method which achieves color confidence in spatial augmented reality applications. For this purpose, we used instead of a color mixing matrix a 3D LUT for every single projector pixel and a physically based computation to determine the influences of ambient light, material, pose of the projector and its color model at interactive rates. This method is able to exactly compute the RGB values of the projected image so that they are equally perceived when they are projected onto a real object as specified desired colors. Our experiments showed that the deviation between desired and projected color is for arbitrary projector poses with an average deviation of $1.3\ \Delta E^*$ less than the just noticeable difference. This accuracy which is crucial for color critical applications like the design process is not achieved by other methods.

However, not all of the desired colors may be inside the gamut of the projector. The proposed implementation runs at interactive rates and does not need any recalibration if the pose of the projector changes. Therefore, it allows us to interactively adjust the gamut and subsequently also the amount of projectable colors by moving the projector to different poses. The proposed approach including all computations achieves frame rates from 3 to 7 fps depending on the scenario. The method was additionally expanded so that also multiple projectors could be used to extend the range of projectable colors at a specific point on the real object. The deviations of the resulting colors to the desired colors in terms of the ΔE^*-metric were below the just noticeable difference.

Since a manual selection of optimal positions for the projectors so that most of the desired colors are inside the gamut of overlapping projectors is hardly possible, these positions should be computed. Furthermore, the resulting visual quality should be preserved even if the images of multiple projectors are overlaid.

References

[1] D.G. Aliaga, A.J. Law, and Y.H. Yeung, "A Virtual Restoration Stage for Real-World Objects," *Proc. ACM SIGGRAPH Asia papers,* pp. 1-10, 2008.

[2] M. Ashdown, T. Okabe, I. Sato, and Y. Sato, "Robust Content-Dependent Photometric Projector Compensation," *Proc. Conf. Computer Vision and Pattern Recognition Workshop (CVPRW '06),* p. 6, 2006.

[3] O. Bimber, A. Emmerling, and T. Klemmer, "Embedded Entertainment with Smart Projectors," *Computer,* vol. 38, no. 1, pp. 48-55, 2005.

[4] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer, "The Visual Computing of Projector-Camera Systems," *Computer Graphic Forum,* vol. 27, no. 8, pp. 2219-2245, 2008.

[5] K. Fujii, M.D. Grossberg, and S.K. Nayar, "A Projector-Camera System with Real-Time Photometric Adaptation for Dynamic Environments," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* p. 1180, 2005.

[6] M.D. Grossberg, H. Peri, S.K. Nayar, and P.N. Belhumeur, "Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 452-459, 2004.

[7] A. Grundhofer and O. Bimber, "Real-Time Adaptive Radiometric Compensation," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 1, pp. 97-108, Jan./Feb. 2008.

[8] P.-C. Hung, "Tetrahedral Division Technique Applied to Colorimetric Calibration for Imaging Media," *Ann. Meeting IS and T,* pp. 419-422, May 1992.
[9] J. Koch, N. Henrich, and S. Müller, "Spatial Color Confidence for Physically Based Rendering Settings on LC Displays," *Proc. Int'l Conf. Computer Graphics Theory and Applications (GRAPP '10),* 2010.
[10] C. Menk, E. Jundt, and R. Koch, "Evaluation of Geometric Registration Methods for Using Spatial Augmented Reality in the Automotive Industry," *Proc. 15th Int'l Workshop Vision, Modeling and Visualization,* 2010.
[11] C. Menk, E. Jundt, and R. Koch, "Visualization Techniques for Using Spatial Augmented Reality in the Design Process of a Car," *Computer Graphics Forum,* vol. 30, pp. 2354-2366, 2011.
[12] C. Menk and R. Koch, "Physically-Based Augmentation of Real Objects with Virtual Content under the Influence of Ambient Light," *Proc. IEEE Int'l Workshop Projector-Camera Systems,* 2010.
[13] C. Menk and R. Koch, "Interactive Visualization Technique for Truthful Color Reproduction in Spatial Augmented Reality Applications," *Proc. IEEE 10th Symp. Mixed and Augmented Reality,* 2011.
[14] S.K. Nayar, H. Peri, M.D. Grossberg, and P.N. Belhumeur, "A Projection System with Radiometric Compensation for Screen Imperfections." *Proc. Int'l Workshop Projector-Camera Systems,* 2003.
[15] S.G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "Optix: A General Purpose Ray Tracing Engine," *ACM Trans. Graphics,* vol. 29, Aug. 2010.
[16] R. Raskar, G. Welch, and H. Fuchs, "Spatially Augmented Reality," *Proc. First Int'l Workshop Augmented Reality,* Nov. 1998.
[17] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay, "Shader Lamps: Animating Real Objects with Image-Based Illumination," *Proc. 12th Eurographics Workshop Rendering Techniques,* pp. 89-102, 2001.
[18] P. Sen, B. Chen, G. Garg, S.R. Marschner, M. Horowitz, M. Levoy, and H.P.A. Lensch, "Dual Photography," *Proc. ACM SIGGRAPH Papers,* pp. 745-755, 2005.
[19] G. Sharma and H.J. Trussell, "Digital Color Imaging," *IEEE Trans. Image Processing,* vol. 6, no. 7, pp. 901-932, July 1997.
[20] Y. Sheng, T.C. Yapo, and B. Cutler, "Global Illumination Compensation for Spatially Augmented Reality," *Computer Graphics Forum,* vol. 29, pp. 387-396, May 2010.
[21] J.-B. Thomas, P. Colantoni, J.Y. Hardeberg, I. Foucherot, and P. Gouton, "An Inverse Display Color Characterization Model Based on an Optimized Structure," *Color Imaging XIII: Processing, Hardcopy, and Applications,* vol. 6807, pp. 68070A-68070A, Jan. 2008.
[22] D. Wang, I. Sato, T. Okabe, and Y. Sato, "Radiometric Compensation in a Projector-Camera System Based Properties of Human Vision System," *Proc. IEEE CS Int'l Workshop Projector-Camera Systems,* p. 100, 2005.
[23] G.J. Ward, "The Radiance Lighting Simulation and Rendering System," *Proc. SIGGRAPH '94,* pp. 459-472, 1994.
[24] G. Wetzstein and O. Bimber, "Radiometric Compensation through Inverse Light Transport," *Proc. 15th Pacific Conf. Computer Graphics and Applications (PG '07),* pp. 391-399, 2007.
[25] T. Yoshida, C. Horii, and K. Sato, "A Virtual Color Reconstruction System for Real Heritage with Light Projection," *Proc. Int'l Conf. Virtual Systems and Multimedia (VSMM '03),* pp. 161-168, 2003.

Christoffer Menk received the degree in computer science from the Christian-Albrechts-University of Kiel in February 2009 and the PhD (Dr-Ing) degree in computer science from the Christian-Albrechts-University of Kiel in 2012. He wrote the PhD thesis at the Volkswagen Group Research on the topic of Spatial Augmented Reality. Currently, he is working as a project manager for the Department of Virtual Technologies at the Volkswagen Group Research in Wolfsburg. His research interests include virtual and augmented reality, computer graphics, computer vision, 3D modeling and tracking algorithms.

Reinhard Koch received the PhD (Dr-Ing) degree in electrical engineering in the field of 3D scene modeling from the University of Hannover in 1996, and from 1996-1999 he lead the 3D modeling team at the KU Leuven, Belgium, in the vision group of Prof. Luc van Gool. He is the head of the Multimedia Information Processing Group at the Christian-Albrechts-University Kiel since 1999. His research interests include 3D modeling from video and images, fast camera tracking and calibration algorithms, the confluence of computer graphics and computer vision in mixed and augmented reality, and real-time image processing.