A Space-Variant Lighting Representation for Photorealistic Rendering of Augmented Content

Hung-Hsiang Chiu*, Yu Fu*, and Homer H. Chen

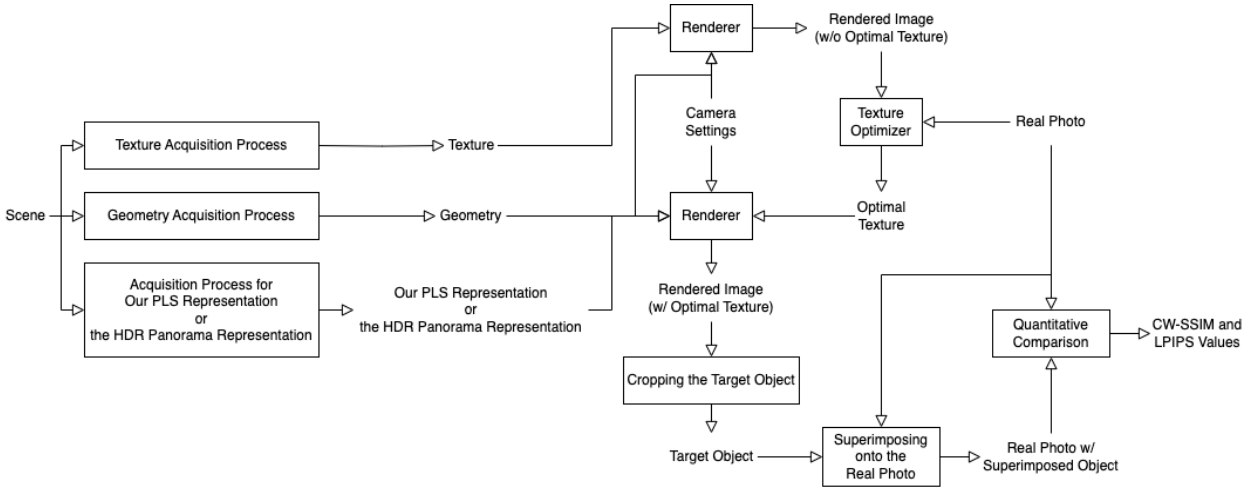

Fig. 1: An overview of our processing pipeline. We first acquire the scene information (texture, geometry, and lighting) of the scene. The lighting is represented by the proposed PLS representation or the HDR panorama representation. Then, the renderer renders images from the scene information and the camera settings. The texture is optimized to minimize the difference between the rendered images and the real photos. Finally, we render the images again, crop the target objects from the rendered images, and superimpose them onto the real photos.

Abstract—The illumination of virtual objects must be consistent with that of real objects to achieve photorealistic rendering for augmented reality. Since objects may appear at arbitrary positions in a scene, the lighting needs to be represented accurately regardless of the object position. Most lighting representations, however, cannot satisfy this fundamental requisite. In this paper, we propose a novel representation that models the lighting as an aggregate of parametric light sources. The illumination of a virtual object depends on its position within the visual field. This spatial variance property is verified by comparing the proposed representation with the high dynamic range panorama representation, a space-invariant representation, and the comparison is performed by evaluating the quality of the rendered objects against the ground truth. The results show that our proposed representation leads to more natural and high-quality images than the high dynamic range panorama representation. Besides spatial variance, our lighting representation is also able to model various kinds of lighting conditions. Both quantitative and qualitative comparisons against the ground truth demonstrate that the proposed representation generates high-fidelity images under various lighting conditions.

Index Terms—Lighting representation, photorealistic rendering, AR display

1 Introduction

Augmented reality (AR) [8, 13] is a technology that superimposes computer-generated virtual images, sounds, and other sensory enhancements onto the real world in real time. It involves a near-eye display device [14] to overlay computer-generated digital information onto the real objects in the audiovisual field of a user. Various applications in entertainment, gaming [36], education [24, 44], marketing, manufacturing [28, 29], and medicine [18, 43] can leverage such a new layer of information and the associated interactivity provided by augmented reality to increase efficacy and productivity. For instance, AR can make education more engaging and effective, offer businesses a new way to promote and showcase products, and assist medical doctors to have a better view of the patient's anatomy during a surgical procedure.

• *Hung-Hsiang Chiu, Yu Fu, and Homer Chen are with National Taiwan University. E-mail: b08901039, b08901095, homer@ntu.edu.tw.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Photorealistic rendering, which makes the augmented virtual objects appear as an integral part of the real world, is critically important to an AR system because it offers convincing and immersive experiences for users. When realistic geometric appearance, surface colors, brightness, and shading for the virtual objects are perceived through an AR system, the virtual objects match the real objects with minimum visual abruptness.

Illumination and hence lighting representations play a crucial role in photorealistic rendering. A lighting representation describes the direction, intensity, and distribution of light rays in the visual field and governs the lighting of virtual objects. To achieve a realistic appearance, the lighting of virtual objects must be coherent with that of real objects; otherwise, unnatural shadows, colors, and exposure can make the virtual objects mismatch the real objects, as shown in Fig. 2. Hence, a lighting representation faithfully modeling real-world lighting is essential to photorealistic rendering.

Obtaining a lighting representation requires collecting lighting data from the real world and estimating the lighting from the collected data. The lighting data can be collected by photographing the scene or measuring the scene's luminosity. Many AR devices [8, 34] are

(a) A real bunny. (b) A virtual bunny lit coherently. (c) A virtual bunny lit incoherently.

Fig. 2: Visual comparison of (a) a real bunny with (b) a virtual counterpart lit coherently and (c) a virtual counterpart lit incoherently.

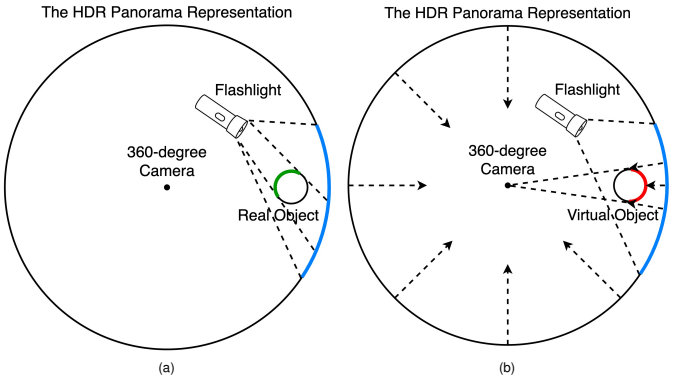

Fig. 3: Illustration of the problem of the HDR panorama representation. (a) The only light source here is a flashlight that emits no ray into a 360-degree camera. The emitted light rays are reflected by the objects in the scene and recorded on the camera. In this record, the region where these objects are located is projected onto a sphere surface at infinity. The lighting information of the flashlight is included in this projected region shown by the blue arc. (b) The dotted arrows towards the center represent the light rays used in rendering. The lit area of the rendered virtual object shown by the red arc is different from the real lit area shown by the green arc in (a).

equipped with sensors such as cameras, LiDARs, or photometers to capture such lighting data and estimate the lighting by inverse rendering techniques [16, 25–27, 33, 40]. We consider the application scenario where an AR display device is able to estimate the lighting of a real scene in real time. The lighting may change itself, for example, due to power on or off, during the operation of the AR display device.

To perform photorealistic rendering, a lighting representation should meet two requirements. First, it should be space-variant, meaning it should be able to accurately describe the difference in direction and intensity of light at different locations in the scene. If a representation is not space-variant, an object rendered at certain locations may appear unrealistic in brightness, color, and shadow position. For example, since the HDR panorama representation, a space-invariant representation, only describes the lighting at the location where the panorama is photographed by a 360-degree camera, the illuminated area of the rendered object would be incorrect when it is rendered at a different location, as illustrated in Fig. 3. Being space-variant is essential to a lighting representation for AR since the position where an augmented object is rendered is most likely to be different from the sensor position, which is near the glasses. Second, a lighting representation should be able to model different lighting conditions.

The parametric light source (PLS) representation developed in this work meets the above requirements. In this representation, we model the lighting as an aggregation of light sources, each characterized by five parameters describing the physical properties of a light source. The results have shown that our PLS representation can represent area lights and spotlights with various shapes and brightness. Besides, our PLS representation outperforms the high dynamic range (HDR) panorama representation in both qualitative and quantitative performance comparisons, showing that our PLS representation is indeed space-variant.

The contributions of our work are highlighted as follows:

- We propose a novel lighting representation and associated light sampling process and coordinate transformation to describe real-world lighting conditions for photorealistic rendering for AR.
- The proposed representation meets the essential requirements for photorealistic rendering of virtual objects.

2 Related Work

Photorealistic rendering is the process of generating photorealistic images, and ray tracing is a widely used technique for photorealistic rendering. It traces the light along its path in a virtual environment and determines how light interacts with objects using various lighting representations, surface texture models, camera models, and scene geometry models.

2.1 Ray Tracing

In a ray tracing process, light is emitted from the camera with its initial distribution determined by the camera model. Then, it travels through the virtual environment by following the laws in geometrical optics and by considering scene geometry. As the light hits an object surface, its direction and color intensity change according to the rules given by the surface texture models. Finally, when the light reaches a light source represented by a lighting representation, an RGB value is generated on the camera. Early ray tracing algorithms [7, 42] only consider a small number of reflected rays for one incident ray, which is sufficient for specular object surfaces but not diffuse ones. Monte-Carlo methods [23, 39], which consider a large number of reflected rays, were proposed to solve the problem, at the expense of computational overhead. Therefore, importance sampling [37] was proposed to reduce the computational cost of Monte-Carlo ray tracing.

2.2 Lighting Representation

The panorama representations [6, 31, 41] and the cubic map representations [6, 41] capture environmental lighting with a 360-degree camera. However, they are space-invariant, unable to describe the lighting at any position other than that of the sensor. The spherical harmonic representation [47] addresses this problem by capturing multiple panoramas at several locations in the scene. The lighting at any position can be further generated by interpolating the lighting information captured at these panorama locations. The volumetric spherical Gaussian representation [26] also achieves spatial variance by dividing the environment into voxels, each of which is considered a light source. These space-variant representations require a large memory for data storage and powerful processors for extensive computation, making them impractical for real-time AR applications.

Parametric lighting representations achieve spatial variance while requiring less data by characterizing lighting as a collection of light sources and associated parameters. The amount of light sources and parameter are usually of two orders of magnitude. The spatial variance property of parametric representations is achieved by using a realistic light propagation model, which ensures correct lighting at any position in the virtual environment provided that accurate light source parameters are given. A recent parametric representation [16] has been developed to model lighting conditions with various kinds of light sources, such as point lights and area lights. However, it can only represent lighting for light sources captured by a sensor. This significantly lowers its generalizability as light rays emitted from light sources in the real world may be blocked by objects, unable to reach the sensor.

2.3 Surface Texture Model

The Lambertian model [22] is one of the surface texture models most widely used for rendering due to its simplicity. It assumes that the surface reflects light uniformly in all directions, which approximates

the behavior of light on diffuse surfaces, not specular ones. The Blinn-Phong model [11] represents both types of behavior by assigning a higher intensity to the specular reflection than to the diffuse reflection. To model light-surface interactions more realistically, the Cook-Torrance model utilizes the distribution of microfacet normals on a surface to determine the direction of light rays after reflection. Commonly used distributions include the Beckmann distribution [10] and the GGX (Trowbridge-Reitz) distribution [38].

2.4 Camera Model

The pinhole model [35] and thin lens model [46] are two common camera models used in computer graphics for modeling a camera with a small field of view. The pinhole model uses a single pinhole or aperture through which light rays pass to form an image on a 2D plane, while the thin lens model incorporates a lens that focuses light onto the image plane. For wide fields of view, the fisheye model [20] and dual-fisheye model [12] are used to capture hemispherical images and panoramas, respectively.

2.5 Scene Geometry Model

Polygonal modeling is a straightforward method to model scene geometry, connecting triangles, quadrangles, and other polygons to represent object surfaces. Point cloud modeling is another commonly used method, representing objects by a collection of points in 3D space. These two methods are often used to construct models of simple scenes. To build a complex one, procedural modeling is often adopted. This method generates objects through a set of rules, such as fractal algorithms [15, 30].

3 Parametric Light Source Representation

3.1 Mathematical Formulation

To make a virtual object appear photorealistic under various lighting conditions, we represent the lighting $\mathscr{L}$ by a collection of light sources,

$$\mathscr{L} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_N\}, \tag{1}$$

where $\mathbf{l}_i \in \mathscr{L}$ denotes a light source and N denotes the number of light sources. Each light source $\mathbf{l}_i$ is modeled as a planar surface with its direction $\mathbf{n}_i \in \mathbb{R}^3$ defined by the unit normal vector of the surface. Its sign is determined artificially. The light source's position, shape, divergence angle, color, and brightness should also be specified to achieve photorealistic rendering. Specifically, each $\mathbf{l}_i$ is mathematically described by

$$\mathbf{l}_i = (\mathbf{n}_i, \mathbf{g}_i, a_i, \mathbf{c}_i, b_i), \tag{2}$$

where $\mathbf{g}_i$ denotes the geometry, including the position and shape, of the light source, a_i denotes the divergence angle of light beams emitted from the light source, $\mathbf{c}_i$ denotes the RGB value of the light beams, and b_i denotes the brightness of the light beams.

We assume that the shape of any light source has an axis of symmetry. It divides the surface boundary of each light source into two halves, which are mirror reflections of each other. We represent each half by a cubic Bézier curve [17] using four anchor points. Specifically, the two cubic Bézier curves, $B_1(t)$ and $B_2(t)$, are given by

$$\begin{aligned} B_1(t) &= (1-t)^3\mathbf{c}_1^i + 3(1-t)^2 t\mathbf{c}_2^i \\ &\quad + 3(1-t)t^2\mathbf{c}_3^i + t^3\mathbf{c}_4^i, \\ B_2(t) &= (1-t)^3\mathbf{c}_5^i + 3(1-t)^2 t\mathbf{c}_6^i \\ &\quad + 3(1-t)t^2\mathbf{c}_7^i + t^3\mathbf{c}_8^i, \end{aligned} \tag{3}$$

where $\mathbf{c}_j^i, j \in \{1, 2, \ldots, 8\}$ denote the anchor points. Note that for each $j \in \{1,2,3,4\}$, $\mathbf{c}_j^i$ is a mirror reflection of $\mathbf{c}_{j+4}^i$ with respect to the axis of symmetry. These two curves determine the geometry of the light source $\mathbf{g}_i$.

Instead of specifying four pairs of symmetric anchor points, which require 24 parameters in $\mathbf{g}_i$, we describe the two cubic Bézier curves in a more storage-efficient way based on the observation that, due to the symmetry, the projections of the four pairs of anchor points onto the axis of symmetry yield a total of four projection points, $\mathbf{e}_1^i$, $\mathbf{a}_1^i$, $\mathbf{a}_2^i$, and $\mathbf{e}_2^i$. The two outer projection points, $\mathbf{e}_1^i$ and $\mathbf{e}_2^i$, are taken as the endpoints of the axis of symmetry and the other two projection points, $\mathbf{a}_1^i$ and $\mathbf{a}_2^i$, are the intermediate points between the two endpoints.

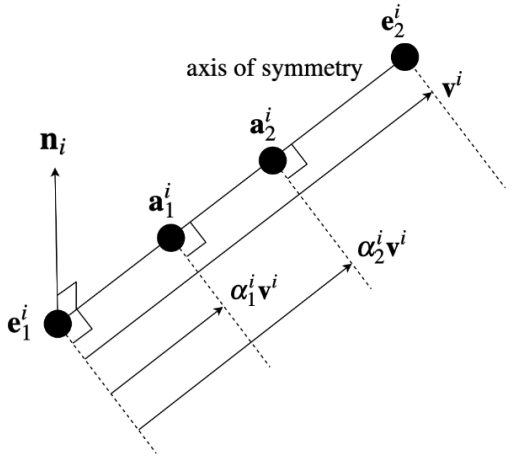

Fig. 4: Geometric relationship between projection points.

Let $\mathbf{v}^i$ denote the vector that starts from $\mathbf{e}_1^i$ and ends at $\mathbf{e}_2^i$. Given the two endpoints with $\mathbf{e}_1^i$ being the reference endpoint, each intermediate point can be calculated by adding a scaled $\mathbf{v}^i$ to the reference point, with the scaling factor denoted by $\alpha_j^i, j \in \{1,2\}$. In particular,

$$\begin{aligned} \mathbf{a}_1^i &= \mathbf{e}_1^i + \alpha_1^i\mathbf{v}^i, \\ \mathbf{a}_2^i &= \mathbf{e}_1^i + \alpha_2^i\mathbf{v}^i, \end{aligned} \tag{4}$$

where $\mathbf{v}^i = \mathbf{e}_2^i - \mathbf{e}_1^i$, as shown in Fig. 4. Note that

$$0 \le \alpha_1^i \le \alpha_2^i \le 1. \tag{5}$$

Moreover, given the light source's direction, the four projection points, and the distances from the projection points to the anchor points, all eight anchor points of the two cubic Bézier curves can be obtained by

$$\begin{aligned} \mathbf{c}_1^i &= \mathbf{e}_1^i + d_1^i\mathbf{p}_i, \\ \mathbf{c}_2^i &= \mathbf{a}_1^i + d_2^i\mathbf{p}_i, \\ \mathbf{c}_3^i &= \mathbf{a}_2^i + d_3^i\mathbf{p}_i, \\ \mathbf{c}_4^i &= \mathbf{e}_2^i + d_4^i\mathbf{p}_i, \\ \mathbf{c}_5^i &= \mathbf{e}_1^i - d_1^i\mathbf{p}_i, \\ \mathbf{c}_6^i &= \mathbf{a}_1^i - d_2^i\mathbf{p}_i, \\ \mathbf{c}_7^i &= \mathbf{a}_2^i - d_3^i\mathbf{p}_i, \\ \mathbf{c}_8^i &= \mathbf{e}_2^i - d_4^i\mathbf{p}_i, \end{aligned} \tag{6}$$

where

$$\mathbf{p}_i = \frac{(\mathbf{e}_2^i - \mathbf{e}_1^i) \times \mathbf{n}_i}{|(\mathbf{e}_2^i - \mathbf{e}_1^i) \times \mathbf{n}_i|} \tag{7}$$

and d_1^i, d_2^i, d_3^i, and d_4^i are the distances between the four anchor points on one cubic Bézier curve and the axis of symmetry. The geometric relationship between these points, vectors, and distances are shown in Fig. 5.

Therefore, we describe the two cubic Bézier curves by specifying the two endpoints of the axis of symmetry, the scaling factors related to

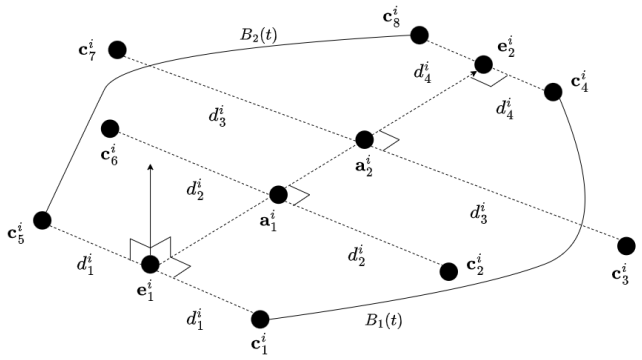

Fig. 5: Relationship between points and distances of two Bézier curves $B_1(t)$ and $B_2(t)$.

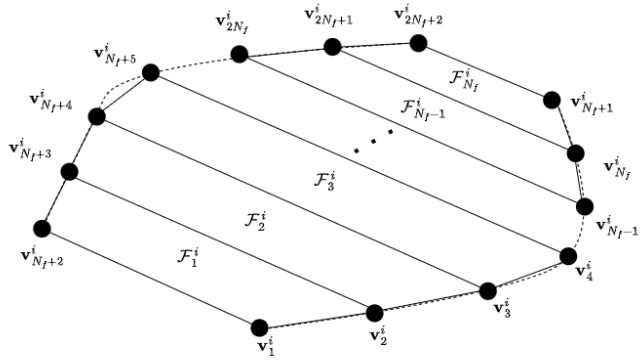

Fig. 6: The quadrilateral approximation of a surface represented by two cubic Bézier curves.

the intermediate points, and the distances between the anchor points on one cubic Bézier curve and the axis of symmetry. In particular,

$$\mathbf{g}_i = (\mathbf{e}_1^i, \mathbf{e}_2^i, \alpha_1^i, \alpha_2^i, d_1^i, d_2^i, d_3^i, d_4^i). \tag{8}$$

This alternative specification reduces the number of parameters for $\mathbf{g}_i$ from 24 to 12, a significant reduction of storage requirement.

3.2 Boundary Approximation

Representing and manipulating surfaces with curved boundaries can be challenging in computer graphics. Simple polygons, such as triangles or quadrilaterals, are easier to process, render, and calculate than complex surface representations. Moreover, they can be tessellated to approximate a curved surface. Therefore, the cubic Bézier curves represented by $\mathbf{g}_i$ are first approximated by polygons before rendering takes place due to computational considerations.

The approximation divides the surface of each light source, denoted by $\mathscr{S}_i$, into a sequence of quadrilaterals along its axis of symmetry, as depicted in Fig. 6. Each quadrilateral consists of four vertices, with two vertices sampled from $B_1(t)$ and the other two from $B_2(t)$. The vertex $\mathbf{v}_j^i$ can be obtained by

$$\mathbf{v}_j^i = \begin{cases} B_1(\frac{j-1}{N_f}) & j \in \{1,...,N_f+1\} \\ B_2(\frac{j-N_f-2}{N_f}) & j \in \{N_f+2,...,2N_f+2\}, \end{cases} \tag{9}$$

where N_f is the number of quadrilaterals. Given the vertices sampled from $B_1(t)$ and $B_2(t)$, the quadrilateral $\mathscr{F}_j^i$ can be represented by

$$\mathscr{F}_k^i = \{\mathbf{v}_k^i, \mathbf{v}_{k+1}^i, \mathbf{v}_{k+N_f+1}^i, \mathbf{v}_{k+N_f+2}^i\},\ k \in \{1,...,N_f\}, \tag{10}$$

and the surface $\mathscr{S}_i$ by

$$\mathscr{S}_i = \{\mathscr{F}_1^i, \mathscr{F}_2^i, ..., \mathscr{F}_{N_f}^i\}. \tag{11}$$

3.3 Light Sampling

In ray tracing, the rays either dissipate in space or encounter a light source in the end. When a ray encounters a light source, light sampling is performed to obtain the light source's color and brightness, which is further used to generate RGB values on the camera. Consider a ray r and a light source $\mathbf{l}_i$. Let o be the last virtual object where r undergoes a reflection. Suppose r intersects the surfaces of o and $\mathbf{l}_i$ at $\mathbf{p}_r^o$ and $\mathbf{p}_r^{l_i}$, respectively. We sample the color $\mathbf{c}_i^s$ and brightness b_i^s of $\mathbf{l}_i$ by

$$\begin{aligned} \mathbf{c}_i^s(\mathbf{p}_r^o,\ \mathbf{p}_r^{l_i}) &= W_i(\mathbf{u}_r)\mathbf{c}_i \\ b_i^s(\mathbf{p}_r^o,\ \mathbf{p}_r^{l_i}) &= W_i(\mathbf{u}_r)b_i, \end{aligned} \tag{12}$$

where $\mathbf{u}_r$ is the vector from $\mathbf{p}_r^{l_i}$ to $\mathbf{p}_r^o$,

$$\mathbf{u}_r = \frac{\mathbf{p}_r^o - \mathbf{p}_r^{l_i}}{|\mathbf{p}_r^o - \mathbf{p}_r^{l_i}|}, \tag{13}$$

and $W_i(\mathbf{u}_r)$ is the window function that determines whether the light source $\mathbf{l}_i$ emits light rays in the direction $\mathbf{u}_r$,

$$W_i(\mathbf{u}_r) = \begin{cases} 1, & \arccos(\mathbf{u}_r \cdot \mathbf{n}_i) \le a_i \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

3.4 Coordinate Transformation

Transforming the mesh of a light source is a common operation in AR applications, which stems from the coordinate system difference between the sensor and the display. In AR applications, the mesh of a light source should be transformed from the sensor coordinate system, where the mesh is acquired, to the display coordinate system, where images are rendered. The transformation can be decomposed into a series of scaling, rotation, and translation, to name a few.

Usually, the mesh's vertices are multiplied by a matrix representing the transformation, introducing a computational overhead due to the large number of vertices. Our PLS representation provides a time-efficient solution. Note that the lighting in our PLS representation is represented by a collection of light sources characterized by multiple parameters. Therefore, instead of transforming the vertices calculated from the parameters of the light source, we transform the parameters before calculating the vertices. Let $M \in \mathbb{R}^{4\times4}$ be the homogeneous transformation matrix. Then

$$M = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix}, \tag{15}$$

where $R \in \mathbb{R}^{3\times3}$ denotes the matrix of rotation, scaling, and shearing, and $T \in \mathbb{R}^3$ denotes the translation vector. Consider the case where a light source $\mathbf{l}_i$ is transformed to $\mathbf{l}_i'$ by M. Since the divergence angle, color, and brightness do not change after transformation,

$$\mathbf{l}_i' = (\mathbf{n}_i', \mathbf{g}_i', a_i, \mathbf{c}_i, b_i), \tag{16}$$

where $\mathbf{n}_i'$ and $\mathbf{g}_i'$ denote the direction and geometry of $\mathbf{l}_i'$, respectively. Given $\mathbf{n}_i$, the transformed direction $\mathbf{n}_i'$ is obtained by

$$\mathbf{n}_i' = \frac{R\mathbf{n}_i}{|R\mathbf{n}_i|}. \tag{17}$$

Given $\mathbf{g}_i$, the transformed geometry $\mathbf{g}_i'$ is described by

$$\mathbf{g}_i' = ((\mathbf{e}_1^i)', (\mathbf{e}_2^i)', \alpha_1^i, \alpha_2^i, (d_1^i)', (d_2^i)', (d_3^i)', (d_4^i)'), \tag{18}$$

where $(\mathbf{e}_1^i)'$ and $(\mathbf{e}_2^i)'$ are obtained by

$$\begin{bmatrix} (\mathbf{e}_1^i)' \\ 1 \end{bmatrix} = M \begin{bmatrix} \mathbf{e}_1^i \\ 1 \end{bmatrix}, \quad \begin{bmatrix} (\mathbf{e}_2^i)' \\ 1 \end{bmatrix} = M \begin{bmatrix} \mathbf{e}_2^i \\ 1 \end{bmatrix}, \tag{19}$$

and $(d_1^i)'$, $(d_2^i)'$, $(d_3^i)'$ and $(d_4^i)'$ can be calculated by

$$\begin{aligned} (d_1^i)' &= sd_1^i, \\ (d_2^i)' &= sd_2^i, \\ (d_3^i)' &= sd_3^i, \\ (d_4^i)' &= sd_4^i, \end{aligned} \tag{20}$$

with

$$s = \frac{|R[(\mathbf{e}_2^i - \mathbf{e}_1^i) \times \mathbf{n}_i]|}{|(\mathbf{e}_2^i - \mathbf{e}_1^i) \times \mathbf{n}_i|}. \tag{21}$$

4 Experiments

We conducted two experiments, the first one for verifying that our PLS lighting representation is space-variant and the second one for verifying that our PLS lighting representation is able to represent various lighting conditions. In the first experiment, we tested our PLS lighting representation against the HDR panorama lighting representation on an object placed at various positions as shown in Fig. 7 by comparing the ground truth with the images rendered by these two lighting representations. In the second experiment, two light sources shown in Fig. 8 were used to create three different lighting conditions by turning on either light source or both. We tested the performance of our PLS lighting representation by comparing the ground truth with the images rendered under the three lighting conditions. To conduct these experiments, we constructed a virtual counterpart of a real scene, including its geometry model, surface texture model, camera model, and PLS lighting representation. For comparison purpose, an HDR representation of the lighting of the virtual scene was created as well. In all tests, we used Mitsuba 3 [19] as our renderer.

4.1 Geometry Model Construction

We constructed the geometry model of eight kinds of objects in the scene, including wall, floor, window, drawn curtain, opened curtain, armchair, folding chair, and coffee table. The geometry models of the wall and floor was constructed by using the polygonal modeling in Blender 3.1.2 [2] with the required length information provided using rulers. The geometry models of the other objects were obtained from TurboSquid [1] and Free3D [3] and adjusted in Blender 3.1.2 to match their real counterparts.

4.2 Surface Texture Model Construction

The modeling of surface textures was based on a bidirectional scattering distribution function [9]. The initial values of the parameters of each surface were chosen according to the surface material. Then, the key parameters including the base color, roughness, metallic, and anisotropic were refined by an Adam optimizer [21] with learning rate, sample-per-pixel, and total iteration number set to 0.05, 16, and 100, respectively. Since the parameters have no ground truth, the optimization was performed to minimize the mean square error between the rendered images and corresponding real photos. The real scene and its geometry and surface texture models are shown in Fig. 9.

4.3 Camera Model Construction

We constructed a pinhole camera model with settings listed in Tab. 1. To verify the correctness of shading across various parts of the objects, the camera was placed at seven positions with different orientations, as shown in Fig. 10.

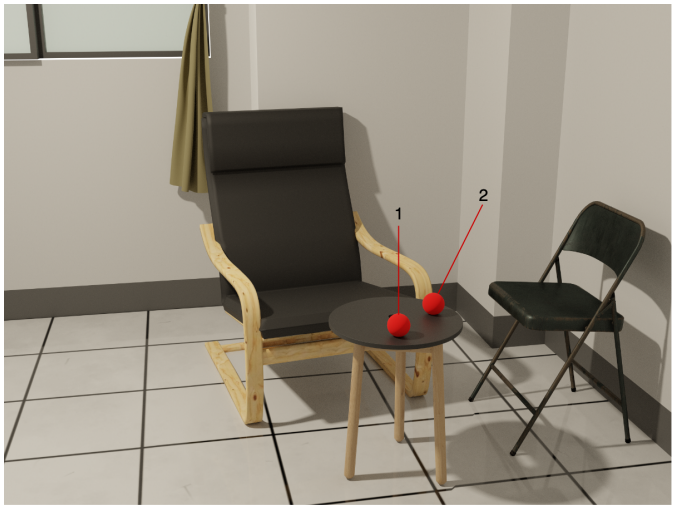

Fig. 7: Two object positions.

Fig. 8: The real light sources.

(a) The real scene. (b) The virtual scene.

Fig. 9: Visual comparison between (a) the real scene and (b) the constructed virtual scene.

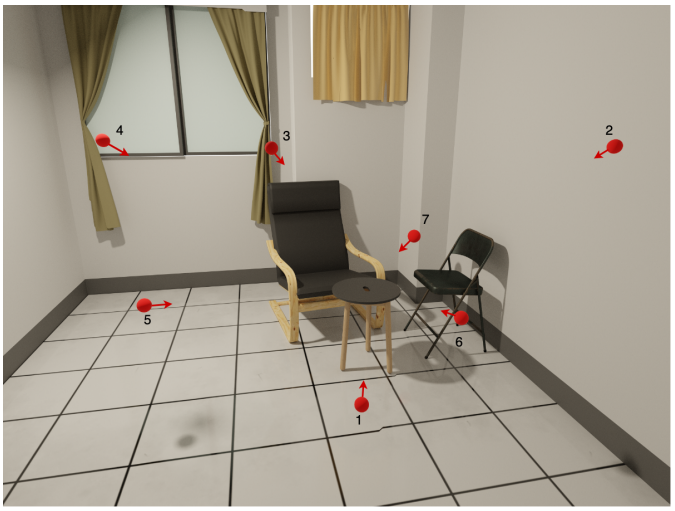

Fig. 10: Seven camera positions and orientations used in the experiments.

Table 1: Camera settings

Camera Parameter	Value
Resolution	4000 x 3000
Color temperature	4400K
Shutter speed	0.25 sec
ISO	400
Field of View	43.6°
Focal length	50mm
Zooming ratio	1.0 (camera 1 $\sim$ 6), 1.8 (camera 7)

4.4 Lighting Representation Construction

We obtained the five parameters of each light source in our PLS representation as follows. The color $\mathbf{c}$ and brightness b were measured by the OHSP-350 spectral irradiance colorimeter. The measured color values in the CIE 1931 color space were further transformed into the RGB space. The direction $\mathbf{n}$ was obtained by normalizing $\overrightarrow{CC'}$, and the divergence angle a was set to the angle between $\overrightarrow{PP'}$ and $\overrightarrow{PP''}$, as shown in Fig. 11. The geometry $\mathbf{g}$ itself contains eight sub-level parameters: the coordinates of two endpoints, two ratios, and four distances. The coordinates of the two endpoints were measured using a ruler, and the two ratios were set to values satisfying (5). The four distances were determined differently for rectangular and circular surfaces. For rectangular surfaces, the distances were obtained by a ruler; for circular surfaces, the distances were obtained as follows:

$$\begin{aligned} d_1 &= d_4 = 0, \\ d_2 &= d_3 = 1.35 d_e, \end{aligned} \tag{22}$$

where d_e is the distance between the two endpoints. The parameters of our PLS representation for the two light sources used in the experiments are listed in Tab. 2 and Tab. 3.

The HDR panorama representation was obtained by merging 10 LDR panoramic photographs taken by a 360-degree camera (Ricoh Theta Z1 [5]) with 10 different exposure time: 1/30, 1/15, 1/8, 1/4, 1/2, 1, 2, 4, 8, and 15 seconds. The merging process was done by using the "HDR merge" function in Adobe Lightroom Classic, with "auto tone" and "auto align" disabled.

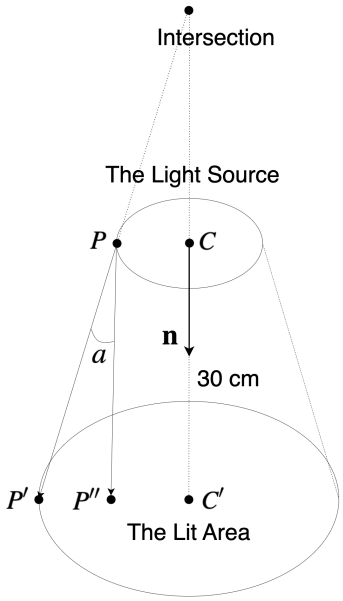

Fig. 11: Obtaining the direction $\mathbf{n}$ and divergence angle a of a light source. The surface of the lit area is parallel to that of the light source. C and C' are the centers of the light source and the lit area, respectively. P is an arbitrary point chosen from the boundary of the light source. P' is a point on the boundary of the lit area such that $\overleftrightarrow{PP'}$ intersects $\overleftrightarrow{CC'}$. P'' is a point on the surface of the lit area such that $\overrightarrow{PP''} \parallel \mathbf{n}$.

5 Results

In the two experiments, qualitative and quantitative comparisons were performed between photos of real and virtual objects. The virtual objects were extracted from the rendered images and superimposed onto the real photos to match practical AR applications where computer-generated content is inserted into a real scene. For quantitative comparisons, two metrics were used to provide different assessments of image similarity. In addition, we displayed rendered objects on AR glasses to verify the applicability of our PLS representation to practical AR scenarios.

5.1 Metrics

The complex-wavelet structural similarity index measure (CW-SSIM) [32] and the learned perceptual image patch similarity (LPIPS) [45] are the two metrics used in our experiments to measure the similarity between real and virtual objects. Traditional image similarity metrics, such as mean square error or peak signal-to-noise ratio, are deprecated due to their high sensitivity to the positional offset between images. In our experiments, a small yet inevitable positional offset could occur due to the manual alignment of virtual objects (including virtual cameras) with their real counterparts. Therefore, we opt for CW-SSIM, which is less sensitive to the positional offset of virtual objects, to focus on object structure instead of subtle pixel difference. The second metric is the AlexNet-based LPIPS, which is good for evaluating the perceptual similarity between real and virtual objects. Note that, unlike CW-SSIM, a lower LPIPS value indicates a higher degree of similarity.

5.2 Results of Spatial Variance Test

The images rendered by our PLS representation are shown in Fig. 12 and Fig. 13. We can see that our results are similar to the ground truth regardless of object position, indicating the space-variant nature of our representation. In contrast, the images rendered by the HDR panorama representation at object position 1 bear no resemblance to the ground truth. That is, the HDR representation suffers from the problem illustrated in Fig. 3.

The quantitative results in Tab. 4 show that our PLS representation outperforms the HDR panorama representation under both metrics.

Table 2: The directions, divergence angles, colors, and brightness of the two light sources used in the experiments.

Light Source	**n** (m)	a (deg)	**c** (RGB)	b (W/sr)
Flashlight	(-0.697, -0.339, -0.632)	10.983	(0.271, 0.239, 0.412)	7.011
Lamp	(-0.117, 0.371, -0.921)	85.227	(0.753, 0.588, 0.306)	8.928

Table 3: The Bézier curve parameters of the two light sources used in the experiments.

Light Source	**g**							
	$\mathbf{e}_1$ (m)	$\mathbf{e}_2$ (m)	α_1	α_2	d_1 (m)	d_2 (m)	d_3 (m)	d_4 (m)
Flashlight	(2.398, 2.564, 1.249)	(2.401, 2.536, 1.260)	0.000	1.000	0.000	0.021	0.021	0.000
Lamp	(1.276, 1.159, 2.311)	(1.275, 1.392, 2.405)	0.333	0.666	0.038	0.038	0.038	0.038

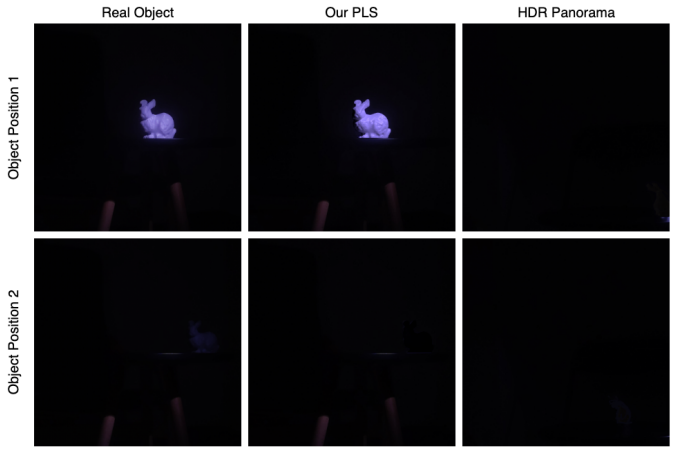

Fig. 12: Results of the spatial variance test. The objects were placed at two positions, and the images were photographed at camera position 1.

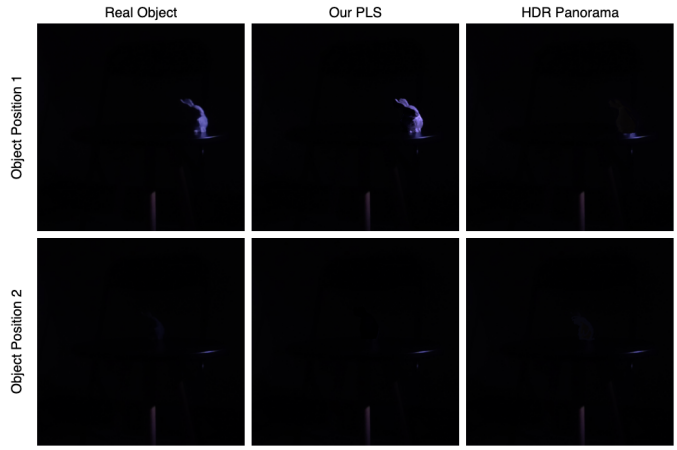

Fig. 13: Results of the spatial variance test. The objects were placed at two positions, and the images were photographed at camera position 5.

Table 4: Quantitative results of the spatial variance test.

Lighting Representation	**CW-SSIM (obj. pos. 1 / 2)**	**LPIPS (obj. pos. 1 / 2)**
our PLS	0.9828 / 0.9481	0.0279 / 0.0444
HDR panorama	0.9085 / 0.9394	0.0712 / 0.0571

Table 5: Quantitative results of the versatility test.

Lighting Condition	**CW-SSIM**	**LPIPS**
Flashlight only	0.9515	0.0252
Lamp only	0.9847	0.0246
Flashlight + Lamp	0.9867	0.0291

At object position 1, where the HDR panorama representation cannot accurately represent lighting, both metrics give rise to a large difference between the two representations. The results clearly demonstrate the spacial variance property of our PLS representation.

5.3 Results of Versatility Test

Figures 14 and 15 show images of objects photographed from different viewing directions. Under all three lighting conditions, the rendered objects are similar to their real counterparts, demonstrating that our PLS representation is able to work for diverse lighting conditions. There are subtle differences in brightness, color, and contrast stemming from the mediocrity of the rendering engine. Despite the differences, the superimposition of rendered objects on natural objects induces no visual abruptness, indicating that our PLS representation can perform photorealistic rendering under various lighting conditions.

The quantitative results in Tab. 5 also support the conclusion. Under all three lighting conditions, good CW-SSIM and LPIPS values are obtained, suggesting a high similarity between the real photos and the rendered images. The high similarity obtained for the third lighting condition (both the flashlight and the lamp are on) underscores the capability of our PLS representation to model the lighting with multiple light sources and verifies that lighting can indeed be represented by a collection of light sources.

5.4 Results on AR Display

To see if our PLS representation can work for an AR device, we crop the virtual objects from the rendered images and display them on the Jorjin J7EF AR glasses [4]. The results in Fig. 16 show that the virtual object can blend well with the real scene regardless of the lighting condition and the shape and texture of the object. The accuracy of brightness, color, and shadow positions ensures a smooth insertion of the virtual object. We notice a minor visual abruptness in the right column of Fig. 16, which in our view is due to the relatively low intensity of the lamp. The dim light makes the virtual object unable to fully block the light from the background. As a result, the background (an armrest in this case) slightly appears, as shown in Fig. 17. Since the problem pertains to a common challenge in AR applications, we consider it outside the scope of our research, which mainly focuses on lighting representation.

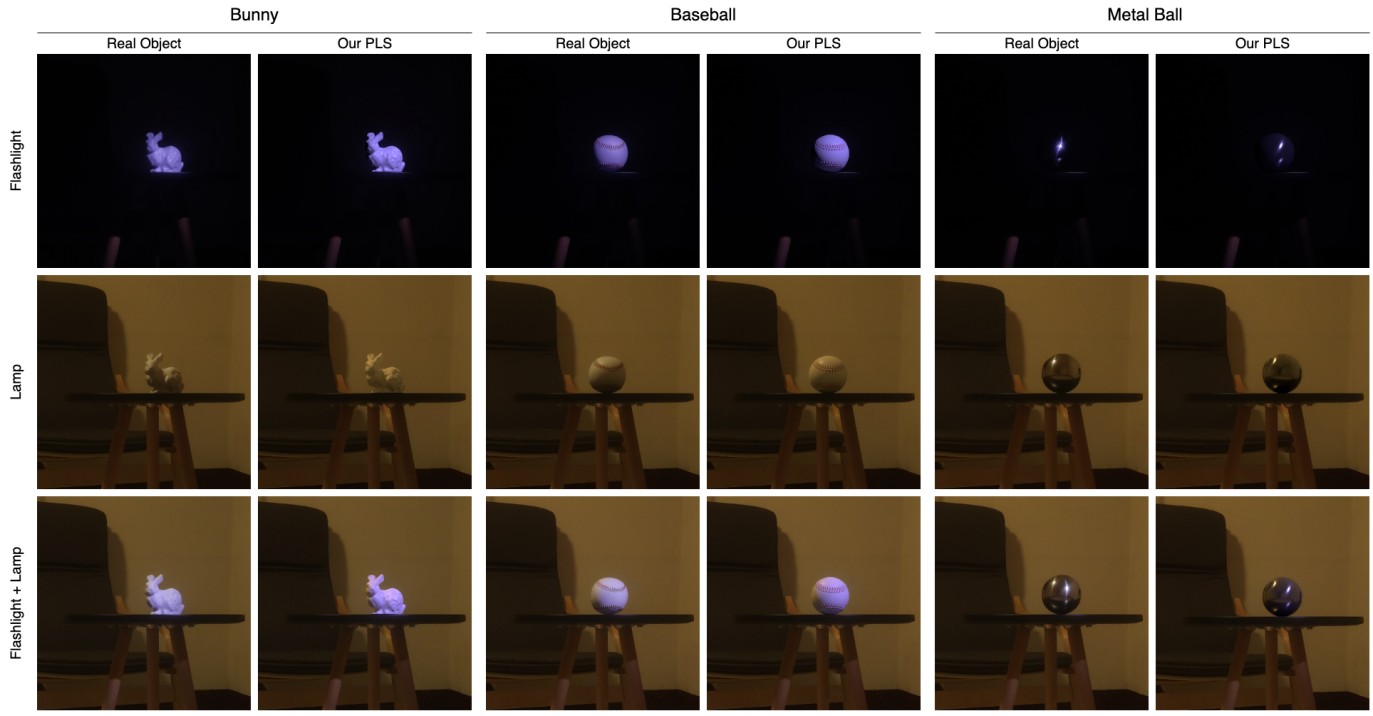

Fig. 14: Results of the versatility test. Three different objects (a bunny, a baseball, and a metal ball) were placed at object position 1, and the images were photographed at camera position 1 under three different lighting conditions.

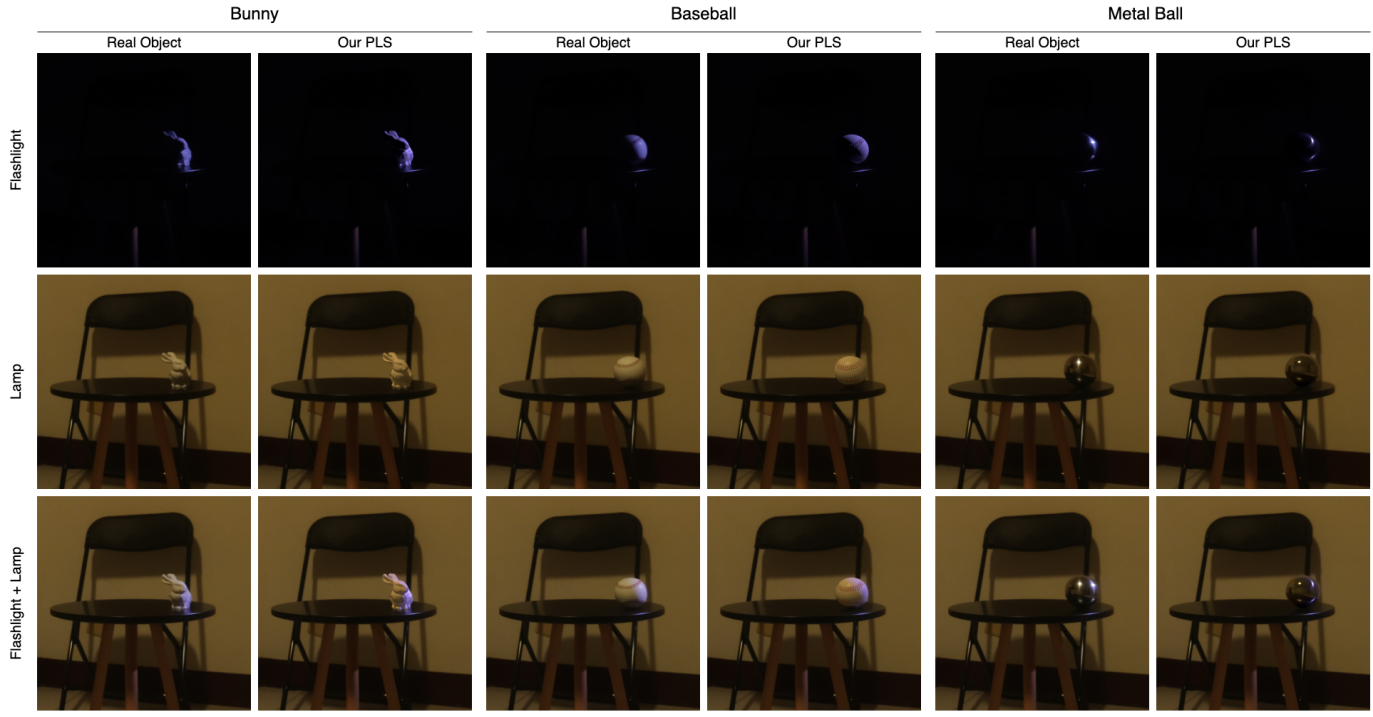

Fig. 15: Results of the versatility test. Three different objects (a bunny, a baseball, and a metal ball) were placed at object position 1, and the images were photographed at camera position 5 under three different lighting conditions.

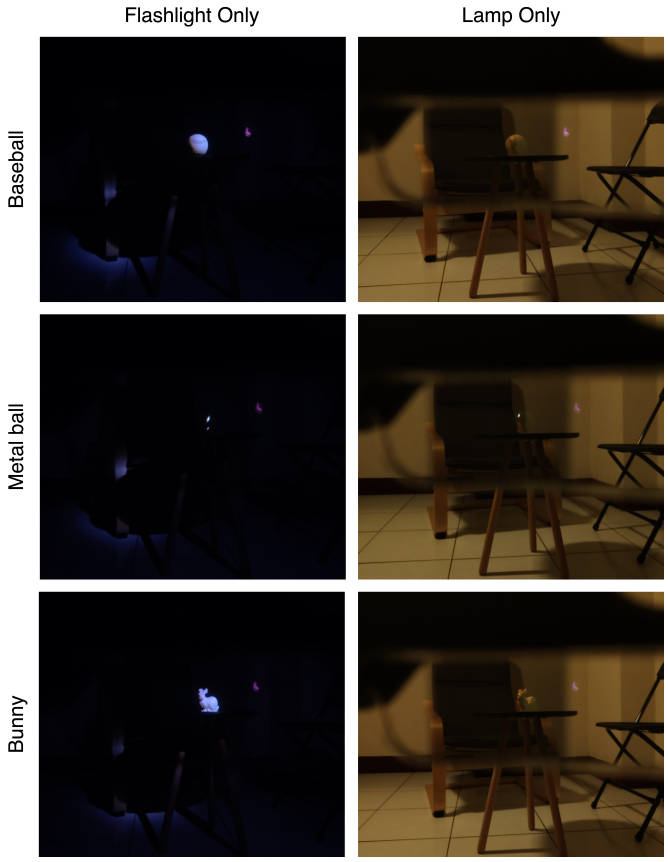

Fig. 16: Appearances of rendered objects displayed on an AR display. The AR glasses were placed at camera position 1, and the smartphone was placed at the right human eye position.

6 Limitations

Our PLS representation assumes uniform light distribution and planar surfaces for all light sources. Therefore, it may fail to model light sources with different light distributions or light sources that are three-dimensional. The applicability of our PLS representation can be enhanced by incorporating additional parameters to characterize the light distribution and 3D surfaces of the light sources.

7 Conclusion

In this paper, we have presented a novel PLS representation of the lighting of a scene to enable photorealistic rendering for AR applications. Our PLS representation characterizes the lighting as a collection of light sources, each requiring only five parameters (the geometry parameter includes eight sub-level parameters). This memory-efficient representation is good for edge-computing devices, such as AR glasses, that have limited memory resource. In addition, our PLS representation is space-variant and able to represent the lighting anywhere in a scene for various lighting conditions. The versatility test has verified this characteristic of our representation for different objects and camera viewing directions. We have also checked the applicability of our PLS representation to real AR glasses. The displayed virtual objects appear realistic, leading to immersive user experience.

Fig. 17: A background leakage may occur at the low intensity areas (encircled), meaning that the background may leak through the virtual object and result in an unrealistic visual experience.

Acknowledgments

The authors would thank Professor Yung-Yu Chuang and Dr. Wan-Chun Ma for their invaluable advice throughout this research project. The support and resources provided by PetaRay were essential in conducting the experiments and achieving the results presented in this paper.

References

[1] 3D Models for Professionals. https://www.turbosquid.com/. 5
[2] Blender 3.1. www.blender.org/download/releases/3-1/. 5
[3] Blender Free 3D Models. free3d.com/3d-models/blender. 5
[4] J-Reality J7EF. www.jorjin.com/products/ar-vr-glasses/j-reality/j7ef/?fbclid=IwAR18xl_6UqHZkkZ_bzNb8GVbsBqcXONE-2E1Dze-p7OorNtzUo1bzFaCIjo. 7
[5] Ricoh Theta Z1. theta360.com/cn/about/theta/z1.html. 6
[6] K. Agusanto, L. Li, Z. Chuangui, and N. W. Sing. Photorealistic rendering for augmented reality using environment illumination. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 208–216. IEEE Computer Society, USA, 2003. doi: 10.1109/ISMAR.2003.1240704 2
[7] A. Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, AFIPS '68 (Spring), pp. 37–45. Association for Computing Machinery, New York, NY, USA, Apr. 1968. doi: 10.1145/1468075.1468082 2
[8] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Aug. 1997. doi: 10.1162/pres.1997.6.4.355 1
[9] F. O. Bartell, E. L. Dereniak, and W. L. Wolfe. The Theory And Measurement Of Bidirectional Reflectance Distribution Function (BRDF) And Bidirectional Transmittance Distribution Function (BTDF). In G. H. Hunt, ed., *Radiation Scattering in Optical Systems*, vol. 0257, pp. 154–160. International Society for Optics and Photonics, SPIE, Mar. 1981. doi: 10.1117/12.959611 5
[10] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. Pergamon Press, 1963. 3
[11] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '77, pp. 192–198. Association for Computing Machinery, New York, NY, USA, 1977. doi: 10.1145/563858.563893 3
[12] M. B. Campos, A. M. G. Tommaselli, J. Marcato Junior, and E. Honkavaara. Geometric model and assessment of a dual-fisheye imaging system. *The Photogrammetric Record*, 33(162):243–263, May 2018. doi: 10.1111/phor.12240 3
[13] J. Carmigniani and B. Furht. Augmented reality: An overview. In B. Furht, ed., *Handbook of Augmented Reality*, pp. 3–46. Springer New York, New York, NY, 2011. doi: 10.1007/978-1-4614-0064-6_1 1
[14] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, Jan. 2011. doi: 10.1007/s11042-010-0660-6 1
[15] A. Fournier, D. Fussell, and L. Carpenter. *Computer Rendering of Stochastic Models*, pp. 189–202. Association for Computing Machinery, New York, NY, USA, June 1998. 3
[16] M. Gardner, Y. Hold-Geoffroy, K. Sunkavalli, C. Gagne, and J. Lalonde. Deep parametric indoor lighting estimation. In *2019 IEEE/CVF International Conference on Computer Vision*, pp. 7174–7182. IEEE Computer Society, Los Alamitos, CA, USA, Nov. 2019. doi: 10.1109/ICCV.2019.00727 2
[17] M. Hazewinkel. *Encyclopaedia of Mathematics: Supplement Volume I*, p. 119. Springer Dordrecht, 1997. doi: 10.1007/978-94-015-1288-6 3
[18] T.-K. Huang, C.-H. Yang, Y.-H. Hsieh, J.-C. Wang, and C.-C. Hung. Augmented reality (AR) and virtual reality (VR) applied in dentistry. *The Kaohsiung journal of medical sciences*, 34(4):243–248, Apr. 2018. doi: 10.1016/j.kjms.2018.01.009 1

[19] W. Jakob, S. Speierer, N. Roussel, and D. Vicini. Dr.jit: A just-in-time compiler for differentiable rendering. *ACM Trans. Graph.*, 41(4), July 2022. doi: 10.1145/3528223.3530099 5

[20] J. Kannala and S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, June 2006. doi: 10.1109/TPAMI.2006.153 3

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. doi: 10.48550/arXiv.1412.6980 5

[22] S. J. Koppal. *Lambertian Reflectance*, pp. 441–443. Springer US, Boston, MA, Feb. 2014. doi: 10.1007/978-0-387-31439-6_534 2

[23] E. P. Lafortune and Y. D. Willems. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques*, pp. 145–153. Alvor, Portugal, Dec. 1993. 2

[24] K. Lee. Augmented reality in education and training. *TechTrends*, 56:13–21, Mar. 2012. doi: 10.1007/s11528-012-0559-3 1

[25] C. LeGendre, W. Ma, G. Fyffe, J. Flynn, L. Charbonnel, J. Busch, and P. Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5911–5921. IEEE Computer Society, Los Alamitos, CA, USA, June 2019. doi: 10.1109/CVPR.2019.00607 2

[26] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481. IEEE Computer Society, Los Alamitos, CA, USA, June 2020. doi: 10.1109/CVPR42600.2020.00255 2

[27] Y. Liu, W. Lai, Y. Chen, Y. Kao, M. Yang, Y. Chuang, and J. Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1648–1657. IEEE Computer Society, Los Alamitos, CA, USA, June 2020. doi: 10.1109/CVPR42600.2020.00172 2

[28] A. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis. Augmented reality applications in design and manufacturing. *CIRP Annals*, 61(2):657–679, 2012. doi: 10.1016/j.cirp.2012.05.010 1

[29] S. K. Ong, M. L. Yuan, and A. Y. C. Nee. Augmented reality applications in manufacturing: a survey. *International Journal of Production Research*, 46(10):2707–2742, Mar. 2008. doi: 10.1080/00207540601064773 1

[30] K. Perlin. An image synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pp. 287–296. Association for Computing Machinery, New York, NY, USA, July 1985. doi: 10.1145/325334.325247 3

[31] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 497–500. Association for Computing Machinery, New York, NY, USA, Aug. 2001. doi: 10.1145/383259.383317 2

[32] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey. Complex wavelet structural similarity: A new image similarity index. *Trans. Img. Proc.*, 18(11):2385–2401, Nov. 2009. doi: 10.1109/TIP.2009.2025923 6

[33] S. A. Sharif, R. A. Naqvi, M. Biswas, and S. Kim. A two-stage deep network for high dynamic range image reconstruction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 550–559. IEEE Computer Society, Los Alamitos, CA, USA, June 2021. doi: 10.1109/CVPRW53098.2021.00067 2

[34] T. Siriborvornratanakul. Enhancing user experiences of mobile-based augmented reality via spatial augmented reality: Designs and architectures of projector-camera devices. *Advances in Multimedia*, 2018, Apr. 2018. doi: 10.1155/2018/8194726 1

[35] P. Sturm. *Pinhole Camera Model*, pp. 610–613. Springer US, Boston, MA, Feb. 2014. doi: 10.1007/978-0-387-31439-6_472 3

[36] B. H. Thomas. A survey of visual, mixed, and augmented reality gaming. *Computers in Entertainment*, 10(1):1–33, Oct. 2012. doi: 10.1145/2381876.2381879 1

[37] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pp. 419–428. Association for Computing Machinery, New York, NY, USA, Sept. 1995. doi: 10.1145/218380.218498 2

[38] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pp. 195–206. Eurographics Association, Goslar, DEU, 2007. doi: 10.5555/2383847.2383874 3

[39] G. J. Ward, F. M. Rubinstein, and R. D. Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pp. 85–92. Association for Computing Machinery, New York, NY, USA, 1988. doi: 10.1145/54852.378490 2

[40] X. Wei, G. Chen, Y. Dong, S. Lin, and X. Tong. Object-based illumination estimation with rendering-aware neural networks. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds., *Computer Vision – ECCV 2020*, pp. 380–396. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-58555-6_23 2

[41] Z. Wen, Z. Liu, and T. Huang. Face relighting with radiance environment maps. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2, pp. II–158. IEEE Computer Society, USA, 2003. doi: 10.1109/CVPR.2003.1211466 2

[42] T. Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980. doi: 10.1145/358876.358882 2

[43] J. W. Yoon, R. E. Chen, E. J. Kim, O. O. Akinduro, P. Kerezoudis, P. K. Han, P. Si, W. D. Freeman, R. J. Diaz, R. J. Komotar, S. M. Pirris, B. L. Brown, M. Bydon, M. Y. Wang, R. E. W. Jr, and A. Quinones-Hinojosa. Augmented reality for the surgeon: systematic review. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 14(4):e1914, Aug. 2018. doi: 10.1002/rcs.1914 1

[44] S. C.-Y. Yuen, G. Yaoyuneyong, and E. Johnson. Augmented reality: An overview and five directions for AR in education. *Journal of Educational Technology Development and Exchange*, 4(1):119–140, 2011. doi: 10.18785/jetde.0401.10 1

[45] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595. IEEE Computer Society, Los Alamitos, CA, USA, June 2018. doi: 10.1109/CVPR.2018.00068 6

[46] Z. Zhang. *Camera Model*, pp. 77–80. Springer US, Boston, MA, Feb. 2014. doi: 10.1007/978-0-387-31439-6_165 3

[47] Y. Zhu, Y. Zhang, S. Li, and B. Shi. Spatially-varying outdoor lighting estimation from intrinsics. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12829–12837. IEEE Computer Society, Los Alamitos, CA, USA, June 2021. doi: 10.1109/CVPR46437.2021.01264 2