Computers & Graphics 76 (2018) 96–106

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on SBGames 2018

Deep spherical harmonics light probe estimator for mixed reality games

Bruno Augusto Dorta Marques*, Esteban Walter Gonzalez Clua, Cristina Nader Vasconcelos

Instituto de Computação - UFF, Av. Gal. Milton Tavares de Souza, Niterói, RJ, Brazil

ARTICLE INFO

Article history:
Received 3 May 2018
Revised 3 September 2018
Accepted 4 September 2018
Available online 12 September 2018

Keywords:
Mixed reality
Augmented reality
Virtual reality
Games
Lighting estimation
Deep learning

ABSTRACT

The recent developments in virtual and mixed reality by the video game and entertainment industries are responsible for increasing user's visual immersion and provide a better user experience in games and other interactive simulations. However, the interaction between the user and simulated environment still relies on game controllers or other unnatural handheld devices. In the mixed reality context, the usage of more natural and immersive alternative to the game controllers, such as the user's hands, may drastically increase the game interface experience, allowing a personalized visual feedback of the user's interactions in the real-time simulation. There are basically two approaches for including the user's hand: a 3D reconstruction based method, typically based on depth cameras, or an image-based approach, composing the virtual scene with the real images of the user's hands. In the composition of the user's hands and virtual elements, perceptual discrepancies in the illumination of objects may occur, generating an inconsistency in the illumination of the mixed reality environment. A consistent illumination of the environment greatly improves the user's immersion in the mixed reality application. One way to ensure consistent illumination is by estimating the real-world illumination and use this information to adapt the virtual world lighting setting. We present the Spherical Harmonics Light Probe Estimator, a deep learning based technique that estimates the lighting setting of the real-world environment. The method uses a single RGB image and does not requires prior knowledge of the scene. The estimator outputs a light probe of the real-world lighting, represented by 9 spherical harmonics coefficients. The estimated light probe is used to create a composite image containing both real and virtual elements in an environment with a consistent illumination. We validate the technique through synthetic tests achieving an RMS error of 0.0573. We show the usage of the method in an augmented virtuality application.

1. Introduction

Mixed Reality (MR) is the mixture of virtual reality environment and the real world. Several applications can benefit from mixed reality because of the increased user's immersion in the simulated environment. This increased immersion can be achieved due to improvements in the realism of the simulated environment and the creation of a personalized experience.

The mixed reality spectrum defines a range of environments between a complete real scene and a totally virtual environment. This term is defined in the reality-virtuality continuum by Milgram et al. [1]. In this spectrum, it is possible to define the Augmented Reality (AR) environment, where most of the environment is composed of objects from the real world. Virtual objects are inserted in the real environment, allowing some kind of interaction with them.

At the other end of the mixed reality spectrum, there is the Augmented Virtuality (AV), where most of the environment is composed of virtual objects. Real objects are inserted in this virtual environment and can interact with the users and the virtual world. An additional form of augmented virtuality is the usage of real-world information, such as movement sensors, GPS location, and weather information.

In the extrema of the reality-virtuality continuum, there are the real environments consisting exclusively of real objects. In the other end of the continuum, there are the virtual environments consisting exclusively of virtual objects, being these virtual environments the essential part of the Virtual Reality (VR) applications.

AR and VR applications usually make use of the user's information, such as the user's location, movements, and image to bring a personalized experience to the users. The personalized experience has been explored by the entertainment industry to attract new users to video-games (Nintendo Wii®, Microsoft Kinect®) and im-

* Corresponding author.
E-mail addresses: brunodorta@id.uff.br (B.A.D. Marques), esteban@ic.uff.br (E.W.G. Clua), crisnv@ic.uff.br (C.N. Vasconcelos).

https://doi.org/10.1016/j.cag.2018.09.003
0097-8493/

prove the user's interaction in home cinema. The advances in MR and VR technology have a huge potential to increase even more the personalization of user's experience by introducing the user as an active character in the simulation.

Recent developments in virtual reality technologies are making low-cost Head-Mounted Displays (HMDs) available to the public. The HMDs manufacturers are investing significant technical and monetary resources to create and distribute immersive content for home entertainment, including video games and cinema. The increasing popularity of VR and MR is reflected by the worldwide revenues for mixed reality and virtual reality market that are expected to grow from US$5.2 billion in 2016 to more than US$162 billion in 2020 [2].

1.1. User's immersion

Video game and other simulation contents for VR and MR treat the user as the main character of the retreated history. The usual representation of the user is a virtual character or an animated 3D model, typically named as the avatar.

For VR and MR applications, the avatar is almost exclusively seen from the first-person point of view. Meaning that a camera is positioned in the eyes of the avatar, and what is seen by the camera is the representation of what the avatar is observing in the environment. The first-person point of view also means that for most of the time, the only visible part of the avatar is the upper limbs (hands and arms).

A big improvement in the user's immersion in MR applications can be achieved by substitution of a 3D avatar's reconstruction of the upper limbs by real images of the user's body, captured by a camera positioned at the HMD. This substitution would mean that the user can see the exact real appearance and movement of his arms and hands, including skin color, geometry, and lighting conditions in the mixed reality simulation, increasing dramatically the user's personalization of the experience.

This substitution of the avatar's synthetic upper limbs to the real user's upper limbs is not straightforward. The projection of a real-world 2D footage containing the user's body, captured from a color camera, into the mixed reality environment is a possible approach. The captured footage containing the user's upper limb is projected into the virtual camera plane, being necessary to preprocess the captured footage to remove unwanted objects from the real scene environment.

This approach is capable of accurately representing the user's physical attributes in the virtual environment but fails to merge the real world and virtual world appearance due to the different lighting condition. We call this difference in the lighting condition between the real and virtual environment as the illumination mismatch problem.

1.2. Illumination mismatch

The illumination mismatch problem can affect the way that the user perceives the scene, due to the distinct lighting conditions in the real objects and the virtual scene. This may cause the user perception to loses the sensation of belonging to the scene, leading to a decrease in his self-presence sensation. The illumination mismatch problem can be solved by adjusting the illumination of the real, or virtual, or both environment.

The illumination of the virtual world is known a priori in the virtual environments. The information of the light sources position and their properties are required to correctly render the virtual environment. Since every light source is well known, adjusting their properties is also straightforward.

The illumination in the real environment is unknown to the simulation. There is no readily available information about lighting condition such as light source position, direction, intensity or color in the real environment. This may be particularly critical in cases of dynamic changes of the lighting conditions.

In a typical mixed reality scenario, the knowledge of the user's real environment, including the lighting conditions, needs to be extracted from images captured by an RGB camera. This extraction is not straightforward. Likewise, changing the lighting configuration of a color image with unknown illumination is also a challenging task because the image does not provide explicit information about the geometry of the scene.

1.3. Deep lighting estimation

One important step to solve the illumination mismatch problem is to recognize or estimate the lighting condition of the real environment. Based on this information, it is possible to match the virtual and real environment lighting by changing the lighting setup of the virtual environment, with an adequate level design project.

The matching of lighting conditions in both real and virtual environment is beneficial to the mixed reality spectrum on both ends. On the augmented reality end, virtual objects inserted in the real scenario can have a realistic appearance and behave like a real object. On the augmented virtuality end, a real object can seamlessly be inserted in the virtual environment.

In a practical mixed reality application, the lighting estimation process can't be onerous for the user. Hence, it should not require complicated setup procedures or additional hardware. By the nature of such applications, which implies the user's movement in an interactive simulation, the lighting estimation process should be computationally fast enough to achieve interactive rate and recognize changes in the illumination.

The lighting estimation is a pattern recognition task that can be treated by machine learning algorithms. Among machine learning approaches that could tackle this task, deep learning algorithms [3] have been responsible for most of the success on techniques to classify or recognize patterns in images and videos.

Artificial Neural Networks (ANN) [4] are specialized algorithms for pattern recognition. These algorithms are inspired by the physiological structure of the human brain, where a pattern is learned by a complex connection between cells called neurons. In the computational neural network, data processing cells called artificial neurons are connected in layers. In the past, ANN algorithms made use of architectures containing few layers of neurons.

Advances in processing power, in particular by the development of Graphics Processing Units (GPUs) and the high availability of data have driven the ANN researchers to increase the number of layers in the ANN architectures. Deep learning is a concept that defines techniques whose an architecture of artificial neural network with multiple hidden layers is used to solve a machine learning task.

Given the nature of the lighting estimation problem as a pattern recognition task, we explore the deep learning algorithms to solve the lighting estimation problem. We developed a strategy to estimates the lighting in the real environment based on a color image. The key contributions of this paper are:

- Novelty strategy based on deep learning to estimate lighting from a raw image. This method is particularly more convenient than others since it does not require special devices such as depth cameras, fish eyes lenses or passive probes inserted in the scene. Furthermore, the method does not require previous knowledge of the scene's geometry;
- A method for lighting estimation that is suitable for indoor and outdoor environments;
- Fast inference for interactive environments. While training a deep learning based model can be computationally intensive,

during inference a trained model demands only a fixed number of floating point operations making it feasible to be processed even on low cost embedded devices;
- An input/output interface that can be easily integrated with current popular game engines such as Unreal® Engine and Unity® software.

2. Related works

The lighting estimation problem is not limited to mixed reality applications. The lighting settings of a scene is an important information for a variety of tasks, including scene editing, video and cinema post process effects, environment design, and scene reconstruction. Previous authors have been working on lighting estimation techniques for a variety of applications with different constraints and available resources.

2.1. Physical light probes for lighting estimation

The use of physical objects as lighting probes is a possible approach to the lighting estimation problem. It is possible to represent the lighting condition of a scene by the scene's radiance. An image-based rendering method introduced by Debevec et al. [5,6] made use of a physical spherical probe placed directly into the scene to measure scene radiance. Their method captures high dynamic images of the lighting probe and uses this information to render a virtual scene. The process to capture the high dynamic images of the lighting probe presents in the scene is laborious and requires a proper setup of the scene. Thus, it is not suitable for practical usage in mixed reality applications, especially for games.

Using the same idea of a physical lighting probe, Calian et al. [7] propose the usage of a specialized 3D printed lighting probe that is capable of capturing the shading of a scene, consequently, estimating the lighting condition of the scene.

A common issue associated with the usage of physical objects as lighting probes into mixed reality application is the requirement of the lighting probe to be present and visible in the real scene. This constraint is not achievable in every mixed reality application.

2.2. Lighting estimation for outdoor scenes

Some methods were developed specifically for outdoor environments. LaLonde et al. [8] present a method to estimate the lighting condition in the outdoor environment by estimating the parameters of a sky lighting model [9,10]. The sky lighting model can't be applied to indoor environments thus is not suitable for most MR applications. The parameters of the sky lighting model are estimated by analyzing shading and shadow cues on the 2D image. The method of Hold et al. [11] also focused on the outdoor environment, but used a different approach, where the parameters of the sky lighting model are inferred by a convolutional neural network.

2.3. Lighting estimation with known scene geometry

Another possible approach for lighting estimation is to make the assumption of a prior knowledge of the scene geometry. A specialized depth camera (RGB-D camera) can be used to capture the scene geometry.

Boom et al. [12] proposed a method to estimate a single point light source based on the geometry of the scene. An RGB-D camera is used to capture the geometry of the scene. They use an image segmentation to find regions of the image with similar albedo and this segmentation provides the necessary information of the object's material in the scene. The geometry and material information allows the reconstruction of the original scene under different lighting configurations. They search for the best position of the light source by a minimization process between the reconstructed scene and the captured image of the real scene. Jiddi et al. [13] proposed a similar method for lighting estimation that could handle multiple light sources in the scene. The multiple light sources estimation is accomplished by analyzing the specular reflections in scene images. Some approaches rely on cameras with a fish-eye lens to capture the surrounding environment and approximate the lighting condition of the scene.

Richter-Trummer et al. [14] proposed a technique to recover the incident lighting from a 3D scanned object. The method presented a series of procedures to recover the diffuse and specular material of a 3D scanned geometry. They estimate the lighting on the surface of the scanned object by an inverse-rendering process based on the object's geometry and materials.

Choe and Shim [15] proposed a method to estimate the incident light of an object based on an inverse rendering technique. The approach takes an RGB-D image as input and employs a segmentation based estimator to predict the low-frequency and high-frequency lighting. Their method generates a lightmap of the scene through an optimization process that estimates the lighting and albedo of the image based on the depth image and the diffuse region of the RGB image. The runtime for whole estimation process takes 6 min. Thus it's not suitable for real-time environments.

Knetch et al. [16] made use of a fish-eye lens camera to approximate the lighting condition of the scene, their approach also requires previous knowledge of the scene geometry, thus requiring an RGB-D camera for the environment reconstruction. Their method estimates the lighting using Virtual Point Lights (VPL) and a combination of differential rendering and instant radiosity to render mixed reality scenarios. Pessoa et al. [17] present a method to dynamically generate and update an environment map of the scene using a fish-eye lens camera and an appropriate camera set up to capture High Dynamic Range (HDR) images. The environment map is employed in a rendering pipeline to render virtual objects into real scenes.

Mandl et al. [18] present a method to generate a radiance map that estimates the lighting configuration in a real scene. The method can use any object present in the scene as a lighting probe. Prior knowledge of the object geometry and texture is required. This method uses deep learning to estimate the radiance map in a 2D image. They generate train data for the neural networks by rendering the lighting probe with the camera positioned in different poses. They train several convolutional neural networks, one for each camera pose. They use an algorithm to estimate the camera pose in the scene of the real environment. This information is used to select which CNN to use in real time. The method requires computationally expensive pre-processing steps, including the 3D scanning of the lighting probe, rendering and generation of the dataset for the specific lighting probe, and training of the neural network.

The method proposed by Mandl et al. [18] uses a bounding sphere around the light probe to represent camera poses. They triangulate the bounding sphere to generate a discretized space of camera poses. They proposed an interpolation technique to select the correct pose, this method requires the training of 6 CNN's for each vertex of the bounding sphere. Additionally, a data augmentation procedure called disk sampling was applied to reduce the number of CNN required for the lighting estimations. With the disk sampling, one CNN training was used for each vertex of the bounding sphere.

The training of a CNN is the most time-consuming procedure in a deep learning technique. Our proposed method uses one CNN for the entire lighting estimation providing a clear advantage over the time-consuming process of training multiple CNN's in the Mandl et al. method. Additionally, Mandl et al. work requires at run-time a pose estimation process to select which CNN to use; this process

can negatively impact the performance of the technique on real-time environments. Moreover, our method does not require extra information of the scene while Mandl et al. demands a 3D scanned object with the respective material.

Gardner et al. [19] proposed a method to estimate the indoor illumination from a single Low Dynamic Range (LDR) image. The method uses a convolutional neural network to generate an HDR environment map of the scene. They adopt a three-step process that consists in construct and train two logistic regression classifiers to identify and annotate the location of light sources in the LDR panoramas. The annotated light sources are used in the training of a CNN to predicts the light source positions in an LDR image. The trained CNN is fine-tuned with an HDR panorama dataset to produce an HDR environment map that represents the lighting in the scene. The method proposed by Gardner et al. requires an HDR panorama dataset to train the CNN. The construction of this dataset is time-consuming and requires specialized equipment and resources, such as panoramic heads, high-quality Digital Single-Lens Reflex (DSLR) cameras, and lenses. The preprocessing of the input images that estimate light sources positions depends on the result of the logistic regression classifiers. They noted that the algorithm has difficulties in finding the exact position of small light sources and have a high error associated with very large area light sources. Our method encodes all the light sources in a set of Spherical Harmonics (SH) coefficients. This implies that our method does not require the exact location of the light sources in the scene thus not requiring preprocessing of the input image prior to the training step.

Marques et al. [20] proposed a method that estimates a point light source position in a scene based on a single LDR image. Their method uses a deep learning approach to predict the main light source position in the real scene. Although their method accurately estimates the main light source of the scene when a single light source is present, it has limitations in the lighting representation of complex lighting scenarios. The method assumes that a single point light source can represent the lighting of the real environment. This assumption is unrealistic. In the real environment, the illumination comes from different light sources placed in distinct locations of the scene, each light source contributes to the lighting of the environment with different intensities. A single light source position cannot accurately represent the lighting in realistic scenarios.

The proposed method, called Spherical Harmonics Light Probe Estimator (SHLPE), uses a convolutional neural network that estimates the parameters (coefficients) of spherical harmonics basis functions that are employed as a light probe of the real scene. The representation of the lighting by a spherical harmonics light probe allows the lighting to be an arbitrary complex area light, thus overcoming the main limitation of Marques et al. method.

We improved on Marques et al. [20] work by changing the representation of the lighting. This change leads to a series of benefits when compared to the previous method:

- The lighting is an area light that can represent multiple light sources of distinct intensities and directions;
- The training dataset does not require an explicit discretization in the placement of the light sources. The light sources are naturally distributed in the scene by spherical harmonics functions;
- The training dataset can incorporate real captured data (HDR environment mappings) to generate realistic lighting configurations;
- The method output is a set of SH coefficients that can be used directly by the game engine as an environment lighting. In the previous work, the predicted light position could not be a valid position for a specific virtual scene (for example, light position inside an item of furniture, or occluded by walls), thus needing to be adjusted by the application.

The previous related works share the constraints that limit the usage of those methods in mixed reality applications applied to games. Those constraints are the usage of obtrusive and not practical lighting probe, prior knowledge of the scene geometry or RGB-D cameras to reconstruct the scene geometry, special equipment such as fish-eye lens camera to capture the environment, and complicated prior setup of the real environment such a properly calibrated camera placement to capture HDR images of the entire scene.

Unlike the previous related methods, the work presented in this paper does not rely on any special equipment, intrusive physical probes, or prior knowledge of the scene. We present a lighting estimation method that uses a single RGB image as input. Our method works on both indoor and outdoor environment and does not require any special scene setup.

The Table 1 summarizes the discussed methods for lighting estimation. Our method, listed in the first row, is the only one that does not require any special equipment or laborious scene setup and is capable of estimate the lighting in both indoor and outdoor environments with multiple light sources in real time.

3. Convolutional neural network

In our method, the lighting estimation is performed by a Convolutional Neural Network (CNN), which is a type of ANN where the main layers are composed of convolution operations. This type of ANN is specialized in recovering features of images. We based our technique on the Residual Network architecture (ResNet) [21].

The neural network is a learning algorithm that works in two separated phases, the network's training, and the inference. The training of the neural network is a phase that is executed a single time and works in any application scenario. In the training process, the training data is fed to the network and weights associated with the layers of the neural network architecture are updated to correctly predict the provided training answers.

The inference process consists of a single feedforward operation of data through the layers of a trained network. The inference is a computationally fast process since no weights updates or complex calculations are necessary. Thus it is possible to execute the inference process in the run-time of a mixed reality application.

As the depth of a deep neural network increases, the network begins to exhibit saturation in the accuracy. This saturation is rapidly followed by a degradation in the accuracy that leads to higher training error. This problem is called the degradation problem.

In deep learning, the degradation problem arises as a consequence of the solvers having difficult in approximating identity mappings from a set of nonlinear layers. The insertion of a preconditioning residual learning function can help the solver to deal with the degradation problem. The residual learning framework [21] introduces residual learning functions that are closer to identity mappings. The residual learning functions are modeled by shortcuts between the layers in the network.

The identity shortcut connects two different layers and produces new output feature maps by an element-wise addition operation. There are two or more layers between the shortcut connected layers. The projection shortcut connection is similar to the identity shortcut, but an additional linear projection is applied to match the dimensions of the connected layers.

The construction of a CNN that follows the residual learning framework can be accomplished by the introduction of residual blocks. A residual block has the following characteristics:

Table 1
Lighting estimation methods comparison.

Authors	Input	Output	No RGB- D required	No fish-eye lens required	No scene setup required	Indoor env.	Outdoor env.
Our, SHLPE	RGB image	Spherical harmonics lighting coefficients	✓	✓	✓	✓	✓
Marques et al. [20]	RGB image	Single point light source position	✓	✓	✓	✓	✓
Gardner et al. [19]	RGB Image	HDR environment mapping	✓	✓		✓	✓
LaLonde et al. [8]	RGB image	Sky lighting model parameters	✓	✓	✓		✓
Hold et al. [11]	RGB image	Sky lighting model parameters	✓	✓	✓		✓
Calian et al. [7]	RGB image containing the lighting probe	Custom PRT Shading parameters	✓	✓		✓	✓
Debevec et al. [5]	HDR images of the lighting probe	Radiance map	✓	✓		✓	✓
Mandl et al. [18]	Lighting probe 3D geometry and material. Scene's RGB image	Spherical harmonics lighting coefficients		✓		✓	✓
Richter-Trummer et al. [14]	3D geometry and material captured by an RGB-D camera	Radiance transfer function		✓		✓	✓
Choe and Shim [15]	RGB-D image	Light map		✓		✓	✓
Boom et al. [12]	RGB-D image	Single point light source position		✓		✓	✓
Jidi et al. [13]	RGB-D image	Multiple point light source position		✓		✓	✓
Knetch et al. [16]	Fish eye camera and RGB-D image	Point light source position			✓	✓	✓
Pessoa et al. [17]	Precomputed environment map and virtual objects geometry	Updated environment map				✓	✓

- A shortcut connection is made to connect the input of the first layer in the building block to the output of the last layer in the building block. The identity shortcut connection is used when the dimension of the input and output are the same. Otherwise, the projection shortcut connection is employed to match the dimensions in the feature maps. The projection shortcut connection is implemented as 1×1 convolution layer with a stride value of 2;
- A bottleneck design is used to decrease the required training time. In the ResNet architecture, this design consists in replacing two 3×3 convolution layers by a stack of three layers following the convention: 1×1, 3×3, and 1×1 convolution layers. The 1 x 1 layers are used to create a bottleneck by reducing the input and output dimensions in the 3×3 layer. Due to the matching dimensions in the first and last layers of the building block, the bottleneck design allows an identity shortcut to be used instead of a projection shortcut. The identity shortcuts result in smaller model size and consequently lower time complexity.

Fig. 1 illustrate the basic bottleneck residual building block for a building block with an input of n feature maps.

The design of a ResNet CNN consists of stacking the building blocks following two rules:

1. Consecutive building blocks that have the same feature map output size must maintain the same number of filters in the 3×3 convolution layer;

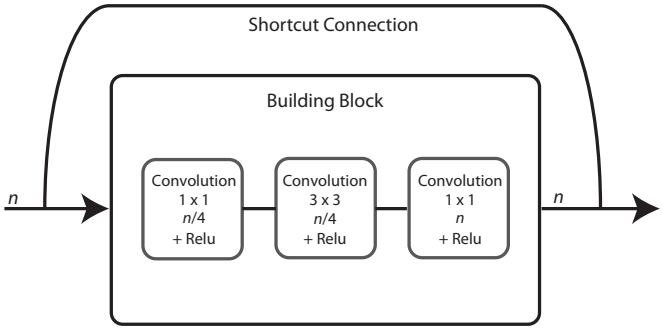

Fig. 1. The basic building block of a Residual Network. The building block has a shortcut connection and follows the bottleneck design [20].

2. The time complexity per layer is preserved by doubling the number of filters every time that a feature map size is halved.

Fig. 2 illustrates a valid design for residual building block stackings in the ResNet architecture. The element-wise addition and the Rectified Linear unit (ReLu) [22] functions present in the output end of the shortcut connection were omitted for a better visualization. In this example, the building block 1 and 2 output feature maps of the same size (in this example, 256). Therefore, according to the first rule, they must have the same number of filters in the 3×3 convolution layer (in this example, 64). The second and

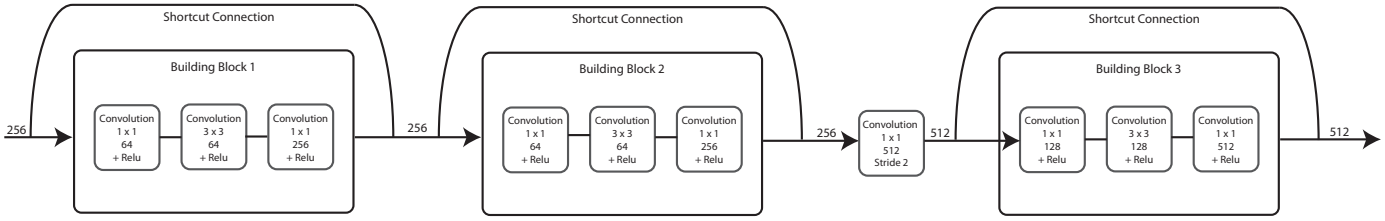

Fig. 2. Residual network design: stacking residual building blocks.

third building block have different feature map output size, a 1 × 1 convolution layer with a stride of 2 was used to halve the feature map size. According to the second rule, the number of filters in the building block 2 layers were doubled (in this example, from 64 to 128 in the 3 × 3 convolution layer). Note that the second rule is also applied to the 1 × 1 convolution layer responsible for downsampling the output of the second building block. We follow the residual network design to construct a 50 layer ResNet for the lighting estimation problem. Our implementation of the ResNet architecture is discussed in Section 8.

4. Spherical harmonics lighting

The spherical harmonics lighting is a technique for calculating the illumination of an area light source on a 3D object. This technique is based on spherical harmonics special functions, that is a basis function capable of representing the lighting over all possible directions in spherical coordinates.

Basis functions are pieces of a signal that can be combined to reconstruct an approximation of the original signal. This reconstruction is based on the sum of pieces, where each piece is composed of a function and a constant scalar.

Given a signal $f(x)$ and a set of basis functions $B_i(x)$, it is possible to calculate the constants C_i that approximate the original signal by integrating the signal by each basis function over the entire domain of the signal. This process of estimating the contribution of each basis function in the original signal is called **projection**. The projection process is shown in Eq. (1):

$$C_i = \int f(x)B_i(x)dx. \tag{1}$$

The context of lighting an 3D environment allows for representing an arbitrary light configuration as a light function. It is possible to represent this light function by a set of coefficients obtained from the project operation on predefined basis functions.

The estimated reconstruction $f'(x)$ of the function $f(x)$ can be obtained by the sum of the basis functions $B_i(x)$ scaled by the associated constant C_i. This operation is shown in Eq (2):

$$f'(x) = \sum_i B_i(x)C_i. \tag{2}$$

The reconstruction of the light function defined on predefined basis functions can be used in the rendering process of a 3D scene. During the rendering, instead of evaluating the light function, it is possible to use only the coefficients that represent the light function on the predefined basis function. This process can speed up the rendering process and allows the usage of a complex light function in real time.

A particularly interesting subfamily of basis function is the orthonormal polynomial basis function. An orthonormal basis is defined by basis functions where the integration of any two of them results in 0 or 1:

$$\int_{-1}^{1} b'_m(x)b'_n(x)dx = \begin{cases} 0 & \text{for } m \neq n \\ 1 & \text{for } m = n \end{cases}. \tag{3}$$

The main idea of the orthonormal basis function is that the contribution of each basis function does not overlap each other, similarly to the Fourier transform that breaks a signal into components sine-waves.

The Spherical Harmonics (SH) functions are a set of polynomial functions that define an orthonormal basis across the surface of a sphere [23]. The general spherical harmonics functions are defined for complex numbers. For lighting purposes, we use only real functions, so we consider only the real spherical harmonics functions (referred here as spherical harmonics functions).

The spherical harmonics are defined in bands, parametrized with the numbers l and m. The positive integer l is the band index (or degree in the polynomial functions notation), the signed integer m is the band order and have values in the range $[-l, l]$. The spherical harmonics functions y_l^m of the spherical angular coordinates (θ, ϕ) are expressed as the following expression:

$$y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}K_l^m \cos(m\phi)P_l^m(\cos\theta) & \text{for } m > 0, \\ \sqrt{2}K_l^m \sin(-m\phi)P_l^{-m}(\cos\theta) & \text{for } m < 0, \\ K_l^0 P_l^0(\cos\theta) & \text{for } m = 0, \end{cases} \tag{4}$$

where K is a scaling factor defined by

$$K_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}}. \tag{5}$$

The term P_l^m in Eq. (4) is the associated Legendre polynomial of degree l and order m. The associated Legendre polynomials [23] are recursively defined by the recurrence relations in the following equations:

$$(l-m)P_l^m(x) = x(2l-1)P_{l-1}^m(x) - (l+m-1)P_{l-2}^m(x), \tag{6a}$$

$$P_m^m(x) = (-1)^m(2m-1)!!(1-x^2)^{m/2}, \tag{6b}$$

$$P_{m+1}^m(x) = x(2m+1)P_m^m(x), \tag{6c}$$

$$P_0^0(x) = 1. \tag{6d}$$

Eq. (6a) generates a higher degree polynomial based on the previous two degrees ($l-1$ and $l-2$) functions. The Eq. (6b) requires no previous values so is suitable to raise the order m from the starting point P_0^0, described in Eq. (6d). The Eq. (6c) can be used to lift a degree l based on the value of a previous function P_{l-1}^m. The process of evaluating the Legendre polynomial function P_l^m consist of generating P_0^m with the Eq. (6b) starting from Eq. (6d). Then use the Eq. (6c) to generate P_1^m and then iterate Eq. (6a) to generate P_l^m.

The SH functions in Eq. (4) are defined in spherical coordinates, the relation between spherical and cartesian coordinates can be

Table 2
First 3 bands SH functions in cartesian coordinates [24].

	$m=-2$	$m=-1$	$m=0$	$m=1$	$m=2$
$l=0$			$\frac{1}{2}\sqrt{\frac{1}{\pi}}$		
$l=1$		$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{y}{r}$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{z}{r}$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{x}{r}$	
$l=2$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{yx}{r^2}$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{yz}{r^2}$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}\frac{2z^2-x^2-y^2}{r^2}$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{zx}{r^2}$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{x^2-y^2}{r^2}$

expressed as:

$$(x, y, z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta). \quad (7)$$

Table 2 shows the first 3 bands of SH functions converted from spherical to cartesian coordinates using the described relation, the term r ensures the normalization of the vector and it is defined as follows:

$$r(x, y, z) = \sqrt{x^2 + y^2 + z^2}. \quad (8)$$

A light distribution function projected in a 3 bands spherical harmonics basis results in a set of 9 coefficients $c_{l,m}$, one coefficient for each SH function. We can use those coefficients in the rendering equation as the diffuse light contributions in a given normal direction n.

The light contribution $L(n)$ in the irradiance function can be expressed as:

$$L(n) = \sum_{l,m} c_{l,m} y_{l,m}(n). \quad (9)$$

For a normal vector in cartesian coordinates, we can use the spherical harmonics functions $y_{l,m}(x, y, z)$ of the Table 2. It is convenient to precalculate the constant terms of the functions $y_{l,m}(x, y, z)$ for usage in the shader step of the rendering process.

The code for the spherical harmonics evaluation is listed in Algorithm 1. The array C contains 9 values resulting from the multiplication of the constant terms of the functions $y_{l,m}(x, y, z)$ and the coefficients $c_{l,m}$ of the SH light function.

Algorithm 1 Spherical harmonics evaluation algorithm.

```
1: function SHResolve(normal, C)        ▷ C is the premultiplied coefficients array.
2:     result ← C[0]
3:     result ← result + C[1] * normal.y
4:     result ← result + C[2] * normal.z
5:     result ← result + C[3] * normal.x
6:     squared ← normal * normal        ▷ the * operator is the component-wise product of two vectors.
7:     result ← result + C[4] * normal.x * normal.y
8:     result ← result + C[5] * normal.z * normal.y
9:     result ← result + C[6] * squared.z
10:    result ← result + C[7] * normal.x * normal.z
11:    result ← result + C[8] * (squared.x − squared.y)
12:    return result
13: end function
```

We use the SH to represent the light probe of the real environment in the SHLPE method. In the Section 7 we apply an SH lighting algorithm to render a scene containing a 3D model of human hands under different lighting settings.

5. Lighting estimation on mixed reality

The calculation of a physically accurate light model requires a huge computational effort and time that is not available in real-time applications. The lighting configuration of the 3D environment in an interactive simulation can be expressed by many possible representations.

Light source models are used to represent the light setup in the 3D scene. The light source models are heuristics that simulates how the real light source works. Those heuristics are computationally efficient and extensively used for interactive and real-time simulations, where a time constraint is present. The usual target for the frame rendering time in an interactive simulation such as games is in the range of 16 to 33 ms [25].

A point light source model is a light model that gives an equal amount of light in all directions, the point light is defined by the position, color, intensity and an attenuation function of the light. The attenuation function is a heuristic function that gives the intensity of the lighting hitting an object, based on the distance between the point light source and the object position. Although point lights being extensively used by games and real-time rendering applications, the area light source model [26] produces smoother shadows and are more suitable for realistic graphics.

Another way to represent the light setup in a 3D scene is the usage of image-based lighting techniques [6]. Those techniques make use of the information contained in radiance images to create a light probe image capable of representing the incident illumination conditions at different points in the space. The light probe can represent arbitrary complex area light of the environment.

In this work, we present a deep learning based method for lighting estimation in mixed reality environments: The Spherical Harmonics Light Probe Estimator (SHLPE). Our method estimates a light probe of the real scene, the light probe is represented by 9 spherical harmonics coefficients. The method use as input a single RGB image, provided by a video camera positioned in the HMD device.

The proposed framework consists of acquiring the input image, processing the image to extract the desired information, feed the processed image to a trained artificial neural network, and output the lighting estimation information for a mixed reality application. The mixed reality application can use this information to adapt the virtual environment in real time to match the illumination in both real and virtual environments. The method assumes the hypothesis that the user's hands are visible in the input image and that the user's hands contain sufficient information about the lighting environment. The Flowchart presented in Fig. 3 illustrates the overview of the mixed reality framework, details of each step of the framework are discussed in the following sections.

6. Input processing

The input processing step in the framework consists in segmenting the input image to isolate the user's upper limbs from any other objects from the image. Since every frame of the camera feed should be processed, the segmentation process should be computationally efficient and suitable for real-time applications.

For the segmentation process, we use a threshold based skin segmentation algorithm. This algorithm makes use of images under different color spaces to identify the pixels in the image that contain a color intensity characteristic of the human skin. The color

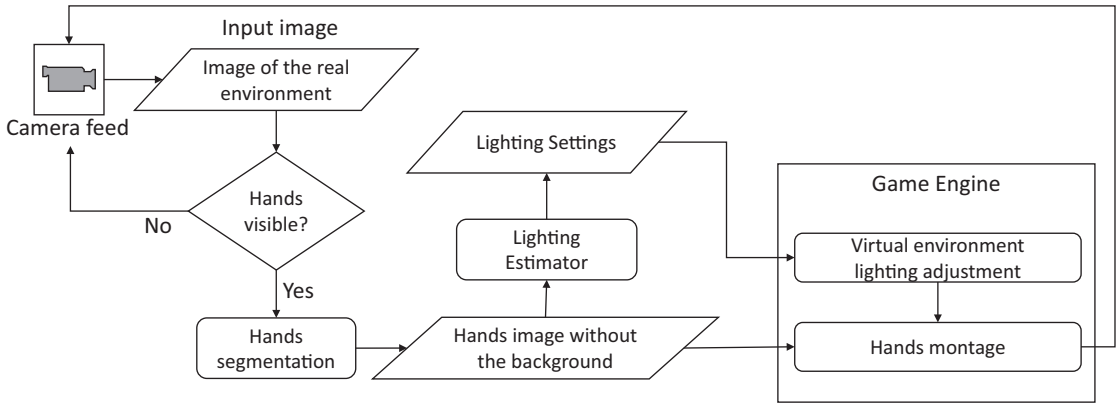

Fig. 3. Mixed reality framework overview: the real environment is acquired by a camera feed, the input image is an RGB color image, we check if the hand is visible in the image to proceed with the lighting estimation. A preprocessing step is performed to separate the hands from the background of the real environment, the resulting segmented image is fed to the lighting estimator and to the hands montage process. The lighting estimator receives the segmented image and outputs the lighting settings that describe the real environment lighting. The game engine receives the lighting settings and adjusts the virtual environment to match the real environment. The hands' montage process overlay the segmented image in the virtual world image to produce the mixed reality scenario.

spaces used in this algorithm are the RGB, HSV and YCbCr color space [27]. The threshold values are based on empirical test results presented in the work of Kolkur et al. [28].

To improve the segmentation results, we employ a closing morphological image processing operation on the image to fill small holes in the segmented skin area of the image. A normalized box filter blur effect is applied to the image to smooth the edges of the segmented area. The next step is to find the contours of the segmented area and flood fill the interior of the segmented area to obtain the final segmented area in the image.

The training phase of a CNN requires a rich dataset of images and their corresponding labels. In the Section 7, we explain a method to create a synthetic dataset for illumination estimation of a spherical harmonics' light probe. The training process of the CNN is described in the Section 8.

7. SH Light Probe Dataset synthesis

The SHLPE treats the lighting estimation process as a regression task. The regression task estimates continuous values for a set of outputs. The regression CNN is the core of the SHLPE. The CNN outputs a set of spherical harmonics coefficients that represent a light probe in the real environment. Each coefficient has a value in the range $[-1, 1]$.

The training process of the CNN requires a dataset containing the images and the values for the SH light probe coefficients. There is no public available dataset suitable for the training process of the CNN in the SHLPE method. For this reason, we decided to create a complete synthetic dataset (The SHLP Dataset) containing human hands under different lighting conditions. The lighting conditions are represented by a single SH encoded light probe.

We authored a 3D human hands model to represent the user in various situations under the usage of the mixed reality application. There are two geometry meshes of human hands to represent both male and female characteristics. We also account for user's racial variations by creating three different skin materials. The 3D model is placed in a game engine to simulate the user's different interactions during the real MR application. We have created animations that simulate the most common interactions performed by the user in mixed reality games and simulations; those animations include walk, push, grab, jump and punch actions. The 3D model uses a screen-space skin shader that approximates the diffuse subsurface scattering of human skin [29]. The 3D hand model is positioned right in front of the virtual camera to simulate the user's placement in a first-person view. The Fig. 4 illustrates the scene setup.

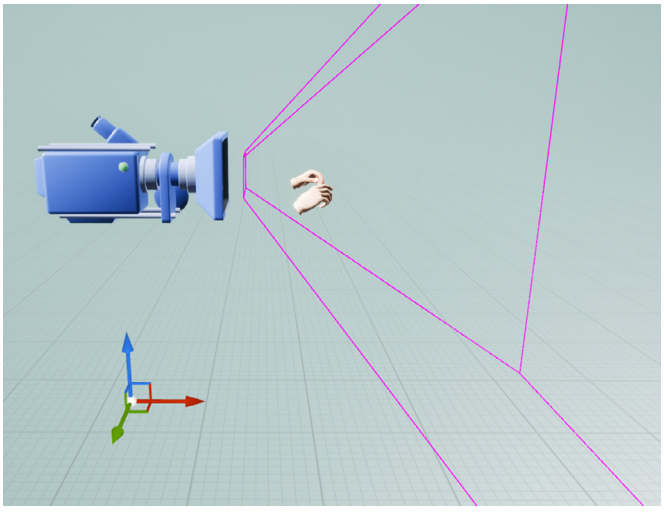

Fig. 4. The SH Light Probe Dataset scene setup: the scene is composed of a virtual camera, a 3D hand model, and a set of 9 SH coefficients that represents the lighting of the scene.

Each entry in the SHLP Dataset is composed of the image containing the rendered scene and the 9 SH coefficients used during the rendering process. We use a second order spherical harmonics basis to approximate the environment lighting. A light probe composed by a set of 9 SH coefficients are sufficient to approximate the low frequency, diffuse, lighting condition of a real light probe [30]. This representation allows multiple light sources with distinct intensities in the resulting lighting probe.

To create representations of various lighting conditions, we generate two SH light probes formed of 9 coefficients each. Then we combine all coefficient values in the light probes to create 2^9 potential lighting configurations. The coefficients are generated by randomly sampling real numbers in the range $[-1, 1]$.

We change the lighting settings in the virtual environment using the SH coefficients in the rendering pipeline [24]. We use the Unreal™Engine 4 for 3D rendering the scene. We discard any rendered frame which the hands are not visible by the virtual camera. Samples of the SHLP Dataset are shown in the Fig. 5.

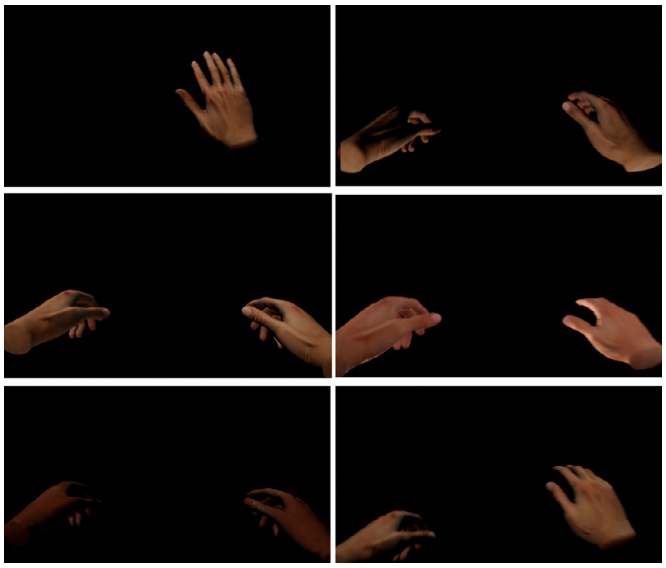

Fig. 5. The SHLP Dataset sample images: resulting sample images of the Dataset. Different skin colors and mesh geometry were used to create the dataset. There is a label describing the 9 SH coefficients lighting settings associated with each sample in the dataset.

Table 3
The ResNet 50 network architecture.

Kernel size	Stride	Pad	Output	Rpt
7 × 7 Convolution	2	3	64	1
3 × 3 Max Pooling	2	0		
1 × 1 Convolution	1	0	64	
3 × 3 Convolution	1	1	64	3
1 × 1 Convolution	1	0	256	
1 × 1 Convolution	1	0	128	4
3 × 3 Convolution	1	1	128	
1 × 1 Convolution	1	0	512	
1 × 1 Convolution	1	0	256	6
3 × 3 Convolution	1	1	256	
1 × 1 Convolution	1	0	2048	
1 × 1 Convolution	1	0	512	3
3 × 3 Convolution	1	1	512	
1 × 1 Convolution	1	0	2048	
7 × 7 Avg. Pooling	1	0	2048	1
Fully Connected Mean squared error (MSE)	–	–	9	1

8. Experiments: convolutional neural network training

The network adopted has 50 layers, the architecture of the network is shown in Table 3. The first layer is a 7 × 7 convolution layer with stride 2, zero padding of 3, and 64 filters. This layer is followed by a max pooling with a stride 2. The next layers are created by stacking ResNet building blocks (described in the Section 3). The first building block has a 3 × 3 convolution layer with 64 filters, the last layer of this building block outputs a feature map of size 256. We repeat this block 3 times (as shown in column Rpt of Table 3). The next building blocks use 3 × 3 convolutions of 128, 256 and 512, respectively. The last portion of the network is composed of a 7 × 7 average pooling layer and a fully connected layer with a hyperbolic tangent function activation.

The training process was executed on top of the Caffe Framework [31]. The 50 layers ResNet convolutional neural network receives as input images with the size of 455 pixels width and 256 pixels height. The network is trained for 80 epochs, with a learning rate of 0.0001. The training batch size was defined as 32, this number was determined by the available GPU memory in our training system. The standard Stochastic Gradient Descent (SGD) [32] algorithm was used for the minimization of the loss function.

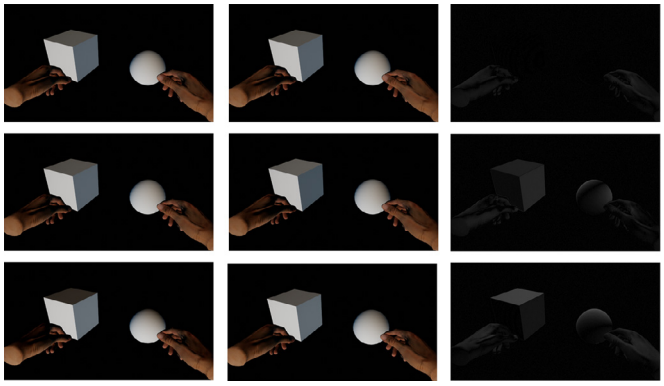

Fig. 6. SHLP estimator on synthetic input. Left Column: Ground truth image generated from the 3D hands and two 3D objects (3D cube and 3D sphere) illuminated by a known spherical harmonics coefficients. Center Column: Scene illuminated by the SHLPE. Right Column: The difference image of the ground truth and the estimated result.

The loss function used was the standard mean squared error (MSE), resulting in a linear regression task. The output of the network is an array of size 9 containing the spherical harmonics coefficients of the lighting configuration.

9. Lighting estimation results

The training, inference, and software prototypes were executed on a machine with the following specification: Intel™Core™i7 4790, 3.6GHz CPU, 24 GB of DDR3 RAM memory with two NVIDIA™Geforce™Titan X, 12GB of GDDR5 memory.

The training time of the CNN's for the SHLPE took 40 h. It is important to note that the training phase is a process that must be executed only once. In the run-time, only the inference phase is executed. The inference time in the CPU took 0.53 s, in the GPU the process took 13 ms. The CNN learned approximately 24 million parameters. The output of the last layer in the CNN is an array of 9 SH coefficients. The RMS error of the trained networks was 0.0573 against the test images.

Fig. 6 shows the resulting image of the SHLPE lighting estimation in a scene containing the 3D human hands and two objects. To generate each sample, a randomly selected pose of the 3D hands were selected and placed in the virtual environment. we generate a random spherical harmonics light probe and render the scene to obtains the ground truth image. The image containing only the 3D human hands are fed to the SHLP estimator, the output coefficients are used to generate the resulting image of the predicted light probe. To illustrate the error in the estimation process, the difference image (right column in Fig. 6) is created by subtracting pixels intensity in the resulting image from the pixels intensity in the ground truth image, the resulting difference image is converted to grayscale for better visualization.

The SHLPE was adopted to generate the images in Fig. 6. The use of a random light probe generated from random sampled spherical harmonics coefficients implies that the values chosen for the light probe do not match the predefined light probe in the SHLP dataset. The estimator was capable of outputting a classification that generates an image with plausible lighting configuration. The first row of images in the Fig. 6 shows lighting estimation that is close to the ground truth (RMSE error: 495.127), while the second and the third rows of images show the results of the experiment with a higher associated error (RMSE error: 975.335, 1367.84, respectively). Even in the cases that the estimation differs from the ground truth, the general appearance of both images are similar and present a plausible lighting and shading. The Root

Table 4
RMSE and NCC statistics for Fig. 6.

Row in Fig. 6	RMSE	NCC
1	495.127	0.865148
2	975.335	0.862725
3	1367.84	0.859666

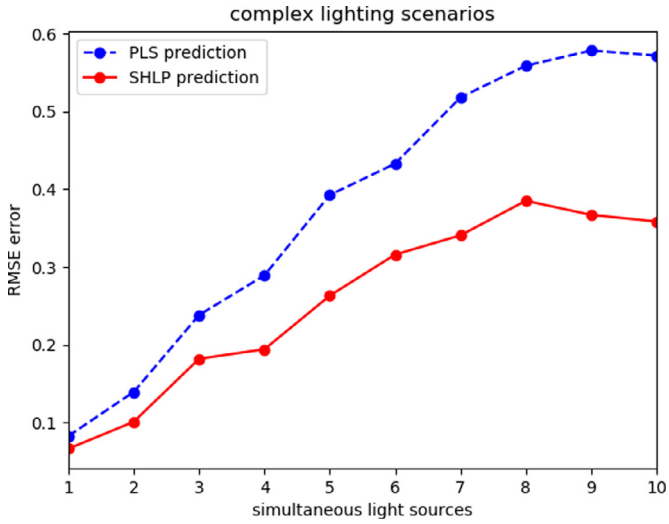

Fig. 7. SHLPE and PLSE [20] lighting estimation on complex lighting scenarios.

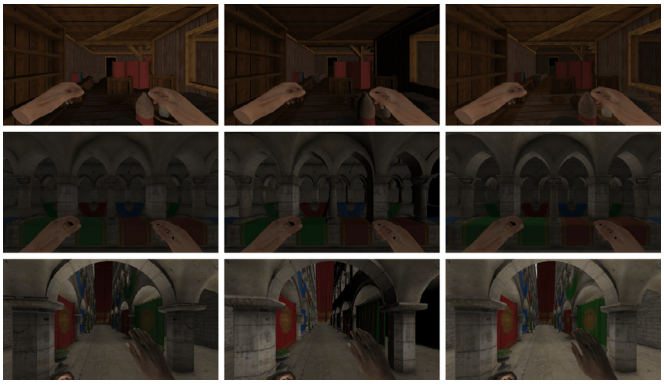

Fig. 8. SHLPE and PLSE lighting estimation comparison. Left Column: Ground truth scene illuminated by a known spherical harmonics coefficients. Center Column: Scene illuminated by the PLSE [20] method. Scene illuminated by our SHLPE method.

Mean Square Error (RMSE) and the Normalized Cross-Correlation (NCC) for all the images in the Fig. 6 are shown in the Table 4.

We compared the estimated prediction of our method with the Point Light Source (PLS) estimation method [20]. Increasing the complexity of the lighting in the scene, by adding additional light sources, results in larger estimation error as seen in Fig. 7. Our method outperforms the PLS method, resulting in a more accurate lighting estimation for any number of additional light sources. In fact, our method estimations are significantly better for lighting complex scenes (RMS error of 0.358 vs 0.572 for 10 additional light sources). We also made a visual comparison of both methods as shown in the Fig. 8. We compare both techniques estimations to a scene rendered with known illumination. For complex lighting scenarios, our method estimates more believable lighting settings than the PLS method, note that our method estimations resemble the ground truth images while the PLS method produces inaccurate hard shadows and a darker ambient light.

The visual perception of the correct lighting setting provided by the SHLPE greatly improves the immersion by correctly blending the hands and the virtual environment.

10. Conclusions

We applied the residual network framework to build a CNN for the proposed SHLPE method. We made use of basis function, in particular, the spherical harmonics basis functions to represent a light function in the real world. The SH functions were used to represent the light probe in the SHLPE method.

We developed a method for lighting estimation in mixed reality applications. The SHLPE method is capable of estimate the light probe that represents the lighting of a real environment. We have shown how the proposed lighting estimation methods could be used in a framework for mixed reality applications and how they can improve the user immersion by mitigating the lighting mismatch problem.

In the Section 9, the experimental results of the usage of the SHLPE method were presented. The results include the performance time, qualitative visual images, and estimation's error comparisons where we demonstrated that the method efficiently estimates the lighting in mixed reality applications. We also show that the SHLPE method outperforms the previous related state of the art method for lighting estimation producing more convincing lighting settings under complex lighting scenarios.

The key results and contributions of this paper are listed below:

- Novel deep learning based method that estimates the lighting condition of the real scene in interactive time from a raw image and does not require any special equipment or prior knowledge of the scene;
- A mixed reality framework that incorporates the lighting estimation process to mitigate the lighting mismatch problem in real time.

Other contribution of this paper includes:

- A public dataset for lighting estimations that output an SH encoded light probe.

Future works includes further investigation of temporal issues. The temporal coherence is a particularly important subject, we plan to create solutions for the detection of abrupt changes in the lighting conditions and methods to adapt the virtual environment to those changes.

Acknowledgments

The authors thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support of this work and NVIDIA® for providing GPUs.

References

[1] Milgram P, Takemura H, Utsumi A, Kishino F. Augmented reality: a class of displays on the reality-virtuality continuum. In: Telemanipulator and telepresence technologies, 2351. SPIE; 1995. p. 282–93.
[2] Gaggioli A. An open research community for studying virtual reality experience. Cyberpsychol Behav Soc Netw 2017;20(2):138–9.
[3] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436.
[4] Bishop CM. Neural networks for pattern recognition. Oxford University Press; 1995.
[5] Debevec P, Graham P, Busch J, Bolas M. A single-shot light probe. In: Proceedings of the ACM SIGGRAPH talks. New York, NY, USA: ACM; 2012 10:1.
[6] Debevec P. Image-based lighting. In: Proceedings of the ACM SIGGRAPH 2005 Courses. ACM; 2005. p. 3.
[7] Calian DA, Mitchell K, Nowrouzezahrai D, Kautz J. The shading probe: Fast appearance acquisition for mobile ar. In: Proceedings of the SIGGRAPH Asia technical briefs. ACM; 2013. p. 20.
[8] Lalonde J-F, Efros AA, Narasimhan SG. Estimating the natural illumination conditions from a single outdoor image. Int J Comput Vis 2012;98(2):123–45.

[9] Lalonde J-F, Narasimhan SG, Efros AA. What do the sun and the sky tell us about the camera? Int J Comput Vis 2010;88(1):24–51.
[10] Hosek L, Wilkie A. An analytic model for full spectral sky-dome radiance. ACM Trans Graph 2012;31(4):1–9.
[11] Hold-Geoffroy Y, Sunkavalli K, Hadap S, Gambaretto E, Lalonde J-F. Deep outdoor illumination estimation. In: Proceedings of the conference on computer vision and pattern recognition (CVPR), 1. IEEE; 2017. p. 6.
[12] Boom BJ, Orts-Escolano S, Ning XX, McDonagh S, Sandilands P, Fisher RB. Interactive light source position estimation for augmented reality with an RGB-D camera. Comput Anim Virt Worlds 2017;28(1):1686.
[13] Jiddi S, Robert P, Marchand E. Reflectance and illumination estimation for realistic augmentations of real scenes. In: Proceedings of the IEEE international symposium on mixed and augmented reality (ISMAR-Adjunct). IEEE; 2016. p. 244–9.
[14] Richter-Trummer T, Kalkofen D, Park J, Schmalstieg D. Instant mixed reality lighting from casual scanning. In: Proceedings of the IEEE international symposium on mixed and augmented reality (ISMAR); 2016. p. 27–36.
[15] Choe J, Shim H. Robust approach to inverse lighting using RGB-D images. Inf Sci 2018;438:73–94.
[16] Knecht M, Traxler C, Mattausch O, Wimmer M. Reciprocal shading for mixed reality. Comput Graph 2012;36(7):846–56.
[17] Pessoa SA, Moura Gd S, Lima JPSd M, Teichrieb V, Kelner J. Rpr-sors: real-time photorealistic rendering of synthetic objects into real scenes. Comput Graph 2012;36(2):50–69.
[18] Mandl D, Yi KM, Mohr P, Roth P, Fua P, Lepetit V, et al. Learning lightprobes for mixed reality illumination. In: Proceedings of the international symposium on mixed and augmented reality (ISMAR); 2017. p. 82–9.
[19] Gardner M-A, Sunkavalli K, Yumer E, Shen X, Gambaretto E, Gagné C, et al. Learning to predict indoor illumination from a single image. ACM Trans Graph 2017;36(6) 176:1–176:14.
[20] Marques BAD, Drumond RR, Vasconcelos CN, Clua E. Deep light source estimation for mixed reality. In: Proceedings of the 13th international joint conference on computer vision, imaging and computer graphics theory and applications, vol. 1: GRAPP,. SciTePress; 2018. p. 303–11.
[21] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
[22] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 807–14.
[23] Weber HJ, Arfken GB. Essential Mathematical Methods for Physicists, ISE. Elsevier; 2003.
[24] Green R. Spherical harmonic lighting: The gritty details. In: Proceedings of the archives of the game developers conference, 56; 2003. p. 4.
[25] Shi S, Hsu C-H, Nahrstedt K, Campbell R. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. In: Proceedings of the 19th ACM international conference on multimedia MM 11. ACM Press; 2011. p. 103–12.
[26] Snyder JM. Area light sources for real-time graphics. Microsoft Research, Redmond, WA, USA, Tech Rep MSR-TR-96–111996.
[27] Zarit BD, Super BJ, Quek FKH. Comparison of five color models in skin pixel classification. In: Proceedings international workshop on recognition, analysis, and tracking of faces and gestures in real-time systems. In conjunction with ICCV99. IEEE Comput. Soc; 1999. p. 58–63.
[28] Kolkur S, Kalbande D, Shimpi P, Bapat C, Jatakia J. Human Skin Detection Using RGB, HSV and YCbCr Color Models. In: Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016). Atlantis Press; 2017 Available from: http://dx.doi.org/10.2991/iccasp-16.2017.51.
[29] Jimenez J, Gutierrez D. GPU Pro: Advanced Rendering Techniques. AK Peters Ltd.; 2010. p. 335–51.
[30] Ramamoorthi R, Hanrahan P. An efficient representation for irradiance environment maps. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM; 2001. p. 497–500.
[31] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, et al. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22Nd ACM international conference on multimedia. New York, NY, USA: ACM; 2014. p. 675–8.
[32] Bottou L. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT. Springer; 2010. p. 177–86.