HandLight: Light Estimation from Hand Interaction in Mixed Reality

David Mandl, Denis Kalkofen, Peter Mohr, Dieter Schmalstieg, and Alexander Plopski

Fig. 1: HandLight estimates the surrounding illumination from hand appearances during gesture-based interaction. For example, while exploring furniture options for an office (a). After a furniture piece is selected, (b) its initial shading does not match the environment (c). As the user interacts with the model to adjust its position and orientation, the system estimates the surrounding illumination with HandLight, resulting in a coherent appearance (d).

Abstract— Correctly estimating the surrounding illumination is essential for creating visually coherent Mixed Reality (MR) experiences. The most accurate results can be achieved by utilizing a light probe, a dedicated object with known reflectance parameters that is placed into the scene. However, the need for a dedicated object placed in the area where the illumination is estimated presents a severe limitation. Building on the increasing popularity of gestural interaction in MR, we present *HandLight*, an approach to estimating the illumination from the user's hands during interaction. Contrary to static light probes, HandLight does not require preparation of the environment and generates an atlas of light probes while the user moves in the world, thus reflecting variable illumination. Our system utilizes a neural network that learns the environment lighting from images of the hand. We train the network on a dataset depicting three common gestures (pinch, fist, bloom) under varying light conditions. We show that our approach can provide believable illumination estimations for a variety of illuminations on a dataset of real hand images.

Index Terms—Mixed Reality, Illumination Estimation, Deep Learning, Interaction

1 Introduction

With the success of consumer-grade head-mounted display (HMD) devices, such as the HoloLens 2, Meta Quest 3, or Apple Vision Pro, Mixed Reality (MR) has become accessible to end users. While advances in object and user tracking support geometric registration well, photometric registration in unprepared environments and at real-time update rates remains challenging. Existing approaches often require time-consuming processing [10, 33, 40] or depend on often difficult to achieve prior knowledge of the user's environment.

Mimicking the real-world scene illumination requires instantaneous and accurate estimation of the light parameters for an unknown number of light sources across the scene, with unknown shapes, colors, and intensities. While some MR devices have built-in light sensors, these provide only a rough estimate of the surrounding intensity with no information about the location or direction of the light sources.

A widely used approach to illumination estimation is placing a light probe, traditionally a sphere with known appearance and reflectance parameters, in the scene and recovering illumination from its appearance [7, 19]. More recent methods replace the reflective sphere with an arbitrary object. By training a deep learning model with illuminated images of that object, the model becomes able to predict its appearance changes under different light situations at runtime [23, 43]. However, while such approaches can recover the surrounding illumination in real-time, they require a known physical object in the scene with known reflectance parameters. Additionally, these systems can only estimate the illumination at the position of the object. This limitation makes them less suitable for mid- to large-scale environments, where multiple such light probes would be required to capture the entire scene illumination.

To address the practical limitations of placing object-based light probes in a scene, several approaches have been proposed for estimating the illumination of unprepared and unaltered environments. These include approaches based on 2D images [17, 30, 36] and approaches based on 3D reconstructions of the environment [13, 34, 47]. Although more versatile, these methods cannot recover local illumination and fail in scenes with limited detail. In addition, approaches based on scene reconstruction are resource-intensive and include errors from an incorrect registration of the reconstruction.

In this paper, we present *HandLight*, a novel approach to estimating illumination in mobile MR applications on an HMD. HandLight builds on the observation that gestural interaction with virtual content is becoming more common in MR applications. During user interaction with virtual content, the cameras on a recent HMD often observe the hands of the MR user for gesture detection. We exploit this data

- *David Mandl, Denis Kalkofen, Peter Mohr, and Alexander Plopski are with Graz University of Technology.*
  E-mail: mandl;kalkofen;pmohr;alexander.plopski@tugraz.at
- *Dieter Schmalstieg is with the University of Stuttgart and Graz University of Technology. E-mail: dschmaldr@visus.uni-stuttgart.de.*

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

and use the detected hand gestures for a subsequent light estimate. In particular, we use a neural network that has been trained on virtual renderings of hand gestures under varying illuminations. In our prototype, we focus on the bloom, pinch, and fist gestures, which are commonly used in MR interaction. By inferring the illumination conditions from images of the hands, we create localized light probes that reflect changes in illumination throughout the scene. As hand interactions are not restricted to a certain place, our approach is well-suited for mobile applications.

Figure 1 shows an example application, in which user hand interaction is monitored to detect environmental lighting. In this application, the user explores different furniture in an MR environment. After selecting a virtual piece of furniture, the user places the corresponding 3D model within the real environment by performing a certain gesture. The system detects the placement gestures, which trigger HandLight to estimate the lighting at the position of the user's hand. We use the position of the hand at the time of light estimation to spatially anchor the light estimation, i.e., we interpret the hand at the time of estimation as a light probe at the position of the hand. As the user walks around and places additional pieces in the environment, we further detect the illumination locally at the position of the user's hands.

To evaluate our approach, we conducted extensive evaluations on captures of synthetic and real hands performing the target gestures. We compare the results obtained with our approach to ground truth data captured with a camera having a wide field of view. We show that our method can capture the surrounding illumination, presenting a practical approach for unconstrained coherent MR experiences.

In addition, we demonstrate the impact of our design choices on the result in an ablation study. In summary, our work makes the following contributions:

- We introduce HandLight, the first approach for mobile light estimation based on hand interaction in MR applications. Using HandLight, we create localized, i.e., spatially anchored, light probes during interaction with MR content.
- We present a comprehensive evaluation of HandLight on rendered and real hands.
- We built a prototype implementation of HandLight on two HMD types (HoloLens 2 and SNAP Spectacles'24) to verify its practicability and to demonstrate possible use cases.

2 Related work

Scene illumination can be recovered from images, videos, 3D reconstructions of the environment, or the appearance of a known object. In the following, we provide an overview of previous work. For more extensive background information, we refer the interested reader to reviews of existing illumination estimation methods by Alhakamy and Tuceryan [1] and Einabadi et al. [8].

2.1 Light estimation from physical objects

Debevec and Malik showed that the incoming illumination can be recovered from observations of a mirror ball [7] that can be applied to illuminating virtual scenes [6]. This approach became widely adopted [19, 38], as it is both accurate and simple. However, relying on a mirror ball for light estimation requires inserting it directly into the scene, which alters the way the environment is perceived. This intrusion not only affects the integrity of the scene but also becomes impractical in large-scale environments, where placing multiple mirror balls would be required.

Jachnik et al. [18] proposed scanning an arbitrary but specular object to estimate its reflectance and texture. After the parameters have been estimated, the object can be used as a light probe that is less intrusive. However, since specular objects are not naturally available in many environments, the light may still be perceived as unnatural. Mandl et al. [23] overcome this limitation by generating a dataset of synthetic renderings of a 3D scan of a real object that is not necessarily specular. They represent the illumination by spherical harmonics (SH) and use a convolutional neural network (CNN) to estimate the SH coefficients from images of the object. Spherical harmonics provide a compact and computationally efficient representation of functions over a spherical domain. In an SH representation [31], radiance can be computed as a weighted sum of the basis functions instead of solving a complex integral, which is infeasible for real-time rendering. Directional illumination can be represented by just a few coefficients that weigh a set of hierarchical basis functions [11].

Similar approaches that allow the use of an arbitrary real object as a light probe have been explored in recent years [41, 43]. However, although these approaches do not alter the perception of the scene, they require preparing the scene, which is often impractical for mobile applications.

2.2 Light estimation from 3D reconstructions

Approaches that avoid introducing additional objects for illumination detection often rely on a 3D reconstruction of the environment. For example, Gruber et al. [13] compute the SH coefficients for each vertex of a reconstructed scene mesh and recover illumination at run-time by solving an equation system that describes the diffuse illumination observed at the vertices. While this approach produces good results, generating the 3D reconstruction can be time-consuming, whereas the reconstruction and its real-world registration may be prone to errors. Gruber et al. [12] presented an extension for dynamic scenes that introduces an improved radiance transfer sampling scheme to achieve real-time frame rates [14]. However, generating the 3D reconstruction of the user's environment remains an obstacle for its practical use.

Rohmer et al. [34] introduced a more lightweight approach that uses an initial 3D point cloud reconstruction instead. The point cloud is used in a server-side simulation and streamed to a mobile device to apply the result at runtime. While offloading computationally intensive tasks is a practical approach, creating the sparse point-cloud reconstruction requires additional time, and the result may be unreliable in feature-deprived environments.

Zhao et al. [46] propose to recover a sparse point cloud of the environment and to estimate the illumination from the recovered points using machine learning. They later extended their work by separating the reconstruction into near-field and far-field components [47]. By recovering the local model and appearance as a dense mesh, the system could estimate the near-field lighting and appearance of high quality. Illumination components and scene appearance farther away were recovered as a sparse point cloud.

While approaches to light estimation from 3D reconstructions avoid the need for preparing the environment, obtaining a reconstruction is typically a computationally expensive task. Using a sparse point cloud lowers the computational demand, but makes the system less reliable in unprepared environments.

2.3 Light estimation from images

In scenes where a light probe or 3D reconstruction is not available, the direction of incoming light can be inferred directly from images. Traditionally, shadow analysis provided insight into the direction of dominant light sources. The illumination can be estimated from the shadows in the scene [29, 35]. This approach is especially effective in outdoor environments, where a single distant light source (the sun) is present. In recent years, rather than analyzing specific aspects of the image, various neural network architectures have been explored to infer the illumination of the scene. Although some methods parameterize illumination [17, 24], a more common approach is to estimate environment maps without any restrictions [10, 37, 41]. Somanath and Kurz [36] utilize a GAN-style framework to train an encoder-decoder network that generates a panorama from a single image and extracts the illumination from it. Most recently, stable diffusion models have been utilized to predict the surrounding panoramas from which HDR illumination can be extracted [30, 45].

While most of these methods work with images, they heavily depend on the scene showing information to infer lighting from. For example, a scene with uniform texture or few or subtle shadows will present a challenge for these approaches. We overcome these problems by relying on the user's hand that is always available.

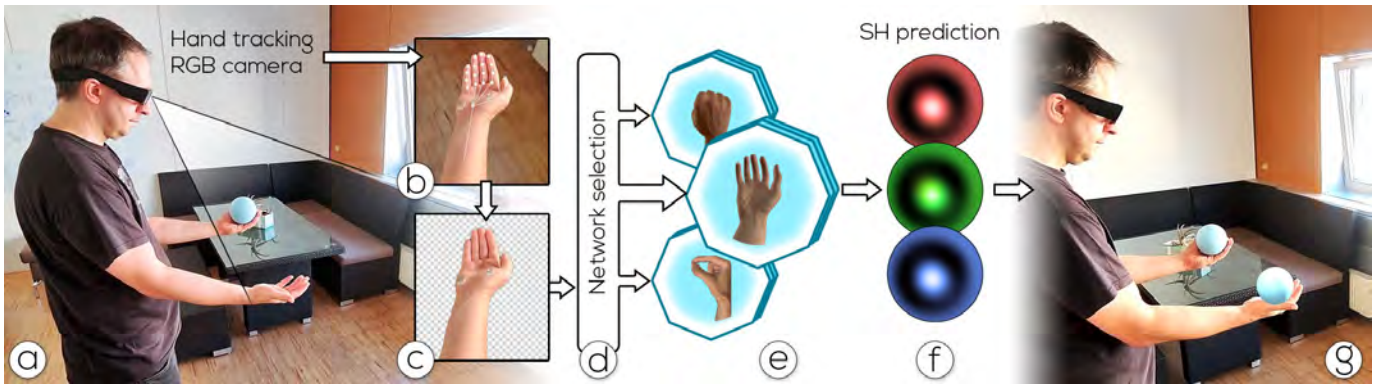


Fig. 2: Overview. HandLight utilizes the device's tracking capabilities to (a) detect and localize the gesture performed by the user. (b) From the camera feed of the scene camera on the head-mounted display, we (c) segment the user's hand and (d) select the network trained to estimate the illumination for the detected gesture. (e) The selected network estimates the (f) spherical harmonics representation of the environment illumination. Finally, we combine the estimated gesture position, the device pose, and the estimated spherical harmonics to (g) compute the scene illumination and apply it to virtual content.

2.4 Light estimation from humans as a light probe

We investigate an approach using images of a known object class, the user's hands, that is always available to a mobile system. Approaches using humans present in a scene for light estimation have been considered before [39]. Knorr et al. [20] relied on appearance changes of several points on the user's face to estimate the surrounding lighting. Nishino and Nayar [28] showed that the reflections in the user's eyes can be used to determine surrounding lights and apply them to relight the scene. Recently, neural networks have been used to recover scene illumination from facial images [3, 22, 42]. Whereas existing approaches rely on capturing other people in the scene or pointing a camera at a specific body part, our method uses images of the user's hands, which we acquire from observing interactions with MR content. Therefore, we do not require any changes to an existing MR system. Marques et al. [24, 25] proposed using the user's hands for illumination estimation in virtual reality. Their work is conceptually closest to ours, but they operate in virtual environments and did not investigate the suitability of the method for MR applications in the real world. Furthermore, they did not provide any information on the size of their dataset, details of the evaluation, and provide few results. As we show in our evaluation, our approach significantly outperforms their results.

3 Method

HandLight takes advantage of the fact that hands are often visible to the system during interactions. Therefore, we can use the appearance of the hand as a reliable and readily available light probe (see Figure 2 for an illustration). Rather than estimating the illumination from arbitrary hand poses, we take advantage of common gestures to detect and segment the user's hands. Thus, HandLight relies implicitly on user interaction with MR content. However, the required interactions are already part of most MR applications.

For each gesture, HandLight utilizes a dedicated CNN to estimate illumination. Hand tracking simultaneously determines the location of the hand, indicating where in the MR environment the SH parameters have been estimated. If multiple closely spaced SH estimations are available, we use these light probes for outlier detection and removal. In the following, we provide details of each component of our pipeline.

3.1 Neural illumination estimation

HandLight assumes that the skin on the user's hand can be approximated as a diffuse surface and that the light is emitted by a distant source, e.g., ceiling light, windows, etc. We represent the global and directional illumination aspects through the first three SH bands (a total of 27 coefficients for nine basis functions in RGB).

We use a CNN to estimate these coefficients from segmented images of the user's hand. Our network, shown in Figure 3, consists of two main building blocks, a ConvBlock and ResNetBlock [16], which we combine into a ConvRes block (Figure 3(right)). A ConvBlock consists of a convolutional layer with a 3×3 kernel, followed by a max-pooling layer with a kernel size and stride of two, and a ReLU activation function. A ResNetBlock consists of two convolutional layers, each of which is followed by batch normalization and ReLU.

The overall architecture of HandLight (Figure 3(left)) starts with a 300×400 image of a segmented hand performing one of the gestures. We chose this resolution to be memory efficient, while keeping training times short. The image is processed by five ConvResBlocks with increasing channel size, followed by a single ConvBlock and finally two fully connected layers with linear output. The output of the last ConvBlock is flattened to generate the input for the first linear layer. We use ReLU for all layers as activation functions. Only the final layer has no activation function and directly outputs the 27 SH coefficients. Note that we considered smaller alternative architectures, such as the one proposed in Mandl et al. [23]. However, we noticed that such small models have insufficient capacity to represent the illuminations and the required variations in hand poses of the same gesture.

3.2 Training

The error function $\mathscr{L}$ consists of two parts, a general loss term that uses a mean squared error (MSE) on all the resulting coefficients (left sum in $\mathscr{L}$), and a weighted MSE that emphasizes errors for some coefficients (right sum in $\mathscr{L}$). The weights are computed on the basis of an exponential fall-off that penalizes high-frequency coefficients more than low-frequency coefficients. This formula ensures a balance between the overall illumination estimate and directional components. For all images in the dataset, we normalize the intensities. The error function $\mathscr{L}$ is defined as

$$\mathscr{L} = \sum_{i=1}^{N} \frac{1}{N}(y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{N} \frac{1}{N}\sqrt{\frac{i}{N-1}}(y_i - \hat{y}_i)^2,$$

and y_i and $\hat{y}_i$ are the ground-truth and predicted SH coefficients, respectively. The term λ (empirically set to 0.1) balances the general loss and the coefficient-weighted loss.

We divide the training data into a training set of 80% and a validation set of 20%. We use a standard optimizer, i.e., the Adam optimizer with a learning rate of 1/1000 and weight decay. The model is trained for 2000 epochs with early stopping to select the best model. We empirically found that these parameters work best in our experiments.

3.3 Data generation

We use renderings of an illuminated hand model together with ground truth SH coefficients for training. We chose synthetic data as the basis

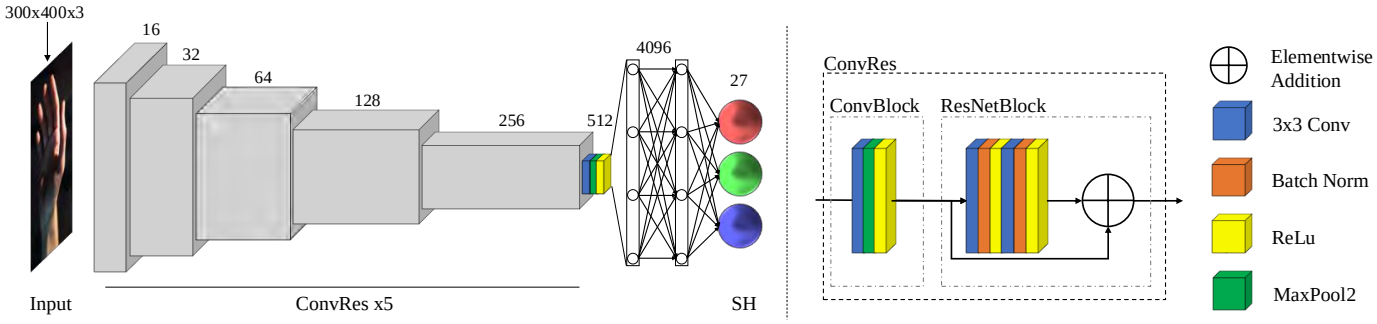


Fig. 3: Network architecture. (left) Our network processes the segmented image of the hand with five ConvRes blocks with an increasing number of channels, followed by a Convolution with a 3x3 filter, a Max pool, a ReLu layer, and two fully convolutional layers. The outcome is the first 9 spherical harmonics components for each color channel. Thus, the output is 27 parameters, nine parameters times three color channels. (right) An illustration of the layout of the building blocks.

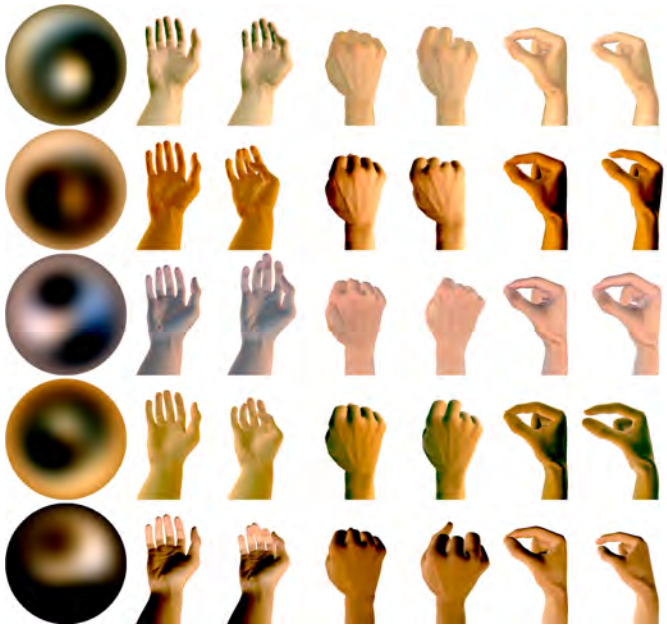


Fig. 4: Training data. Sample images from the datasets used for training light detection. Our prototype used three gestures. Each row is a different light variations and every two columns are without and with pose variations.

for our training dataset because it allows us to easily produce images of hands under controlled lighting conditions.

While there is a large number of hand datasets with variable illumination [26,27], these do not include egocentric hand captures performing gestures interacting with virtual content. As capturing a sufficient number of egocentric hand captures under variable illumination for machine learning presents a challenge in itself, we generate a custom synthetic dataset.

The training data is generated in an OpenGL framework that loads the synthetic hand model and manipulates the hand joints to create the targeted hand gestures. Illumination is computed using random SH coefficients sampled from a range of natural color temperatures. These coefficients are set in a shader to compute the diffuse irradiance. The framework allows setting all parameters needed for both random poses of the hand and random light settings, generating any number of images for a given gesture. The images are annotated with the SH coefficients used in the image synthesis.

To capture controlled images of hands under different environmental lighting conditions, we use an articulated virtual hand model from TurboSquid[1]. This model resembles how the HMD camera would capture a human hand. An example of the generated images is shown in Figure 4. To account for differences in how users perform gestures, we randomly rotate the hand model in three dimensions by 5-15° and apply a random rotation by 0-10° to each finger joint. The illumination is generated using three bands of SH coefficients, and we randomly change these coefficients to generate different light conditions.

To make the network more robust to skin variations, we use a variation of dark and light colored skin textures on the hand model we use. We compute each material texture by blending a dark and light base texture with varying alpha values in the range of [0,1]. The two base skin textures achieve a Fitzpatrick scale [15] of Type I (light skin) and Type VI (dark skin). We computed the scale on all non-black texels using the individual Typology Angle (ITA) [4].

To emulate natural light sources, we randomly set the color temperature in the range of 3000 to 8000 Kelvin and convert it to RGB values using the correlated color temperature [21]. Next, we select an exposure value in the range [-2.0, 2.0] to adjust the intensity of the final color. For more variation, we add uniformly distributed offsets in the range [-0.1, 0.1] to each color channel. These RGB values correspond to the first SH coefficient that represents ambient light. The other coefficients are randomly selected from a uniform distribution in the range [-1.0, 1.0] that is multiplied by the color temperature. We apply tonemapping using a global Reinhard operator [32] and Gamma correction with a value of 2.2 before storing the image in sRGB color space. For each gesture, we render 10K images using the above approach and store them together with the corresponding SH coefficients.

3.4 Hand tracking and model selection

Once a known gesture has been detected, we use the corresponding frame for light detection. We use the device's hand tracking to determine the position of the hand. For each image received on the server, we define a region of interest based on the bounding box of the fingers after projecting the known 3D joint positions into the image. Inside this region of interest, we segment the hand by applying a color threshold in HSV space. Using the mask obtained by this segmentation step, we reformat the image by cropping it to the hand dimension and resizing the result to the input size of the network. The detected gesture is used to select a gesture-specific network version and to run the inference on the reformatted image. The SH coefficients are sent back to the client.

3.5 Light probe placement and rendering

Once HandLight has estimated the illumination for a given hand pose, the system places a new light probe at the location of the hand and aligns it with the existing lightprobes in the environment. We represent the environment as a voxel grid and add the new probe to the corresponding voxel. If the voxel has prior observations, we exclude

[1] https://www.turbosquid.com

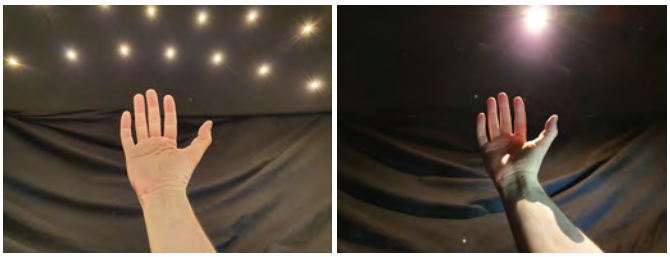


Fig. 5: We illuminate real hands using a light stage that consists of 61 LEDs, which enables capturing hand gestures under varying illumination conditions. Here we show two examples from our dataset.

estimates older than ten seconds to account for changing light conditions.

From the remaining observations, we calculate a single representative for the currently altered voxel by computing the median of each SH component and channel. This approach enables providing an irradiance volume that represents the observations and that allows either selecting the closest available illumination or to interpolate between light probes, for example, by using the approach of Cupisz et al. [5].

Additionally, we use the estimated probe to estimate the main light direction as the peak in the SH basis. The main light direction is used to compute direct shading [44], which is added to indirect shading [2]. Moreover, we use the main light direction to apply shadow mapping (Figure 8).

3.6 Application integration

HandLight is split into two parts: an offline training system that is implemented using the PyTorch[2] framework and an online component that uses the trained model in an MR application to infer the lighting from the user's hand. We demonstrate the feasibility of our approach on the Microsoft HoloLens 2 and the Snap Spectacles'24, both of which already provide hand tracking. Because both headsets lack the computational power to run CNN inference, we employ a client/server framework with a GPU-accelerated stationary computer as the server and the headset as the client.

To invoke the computation, the client streams a camera image over WiFi. We annotate the image stream with the headset pose, camera pose, and the currently detected hand gesture. The server receives the annotated images, runs the inference, and then sends the resulting SH coefficients back to the headset for rendering. The network communication runs asynchronously so that rendering is not affected, and the computed light probes are cached on the HMD to minimize client-server communication. Thus, the latest illumination estimation is available for MR rendering without affecting the frame rate on the client device.

4 Evaluation

We evaluated how accurately HandLight can estimate the surrounding illumination for the three selected gestures on synthetic and real datasets (Figures 4 and 5). We show qualitative results by comparing the rendered scenes with the ground truth and estimated illumination. We determined multiple error metrics for the estimation results and created visualizations to better understand the impact of our design choices. Additionally, to compare HandLight with the results of Marques et al. [25], who, to our knowledge, were the only prior work that targeted hands for relighting. We implemented their network to the best of our understanding and trained the network on our training dataset.

We evaluate HandLight on a desktop computer with an Intel i7-12700K with 3.6 Ghz, 128 GB RAM, and a NVIDIA GeForce 4090 RTX with 24 GB VRAM. The same machine was used for training and all evaluations. We measured the average inference time over 1000 runs at 9.06 ms.

[2] https://pytorch.org

4.1 Synthetic dataset

To evaluate the performance of our model, we use 92 panoramic images from the Laval Indoor HDR dataset [9]. These images represent a wide variety of indoor illumination scenarios, offering insight into how HandLight performs in a wide range of environments. We used panoramic images because they provide a good approximation of all incident light directions. To apply the illumination captured in these panoramic images to a virtual environment, we first convert the equirectangular images to a spherical representation. We generate low-discrepancy samples on the unit sphere to uniformly sample from the panorama. These samples are in spherical coordinates and can be converted to Cartesian coordinates, resulting in the normal direction. With this normal, we sample the SH basis functions. Finally, we fetch the color from the panorama using the normal and accumulate the resulting coefficients. With the resulting illuminations, we render the hand model with the three gestures to get images for our test set.

4.2 Real dataset

For the real dataset, we capture the users' hands in our light stage (Figure 5) with 61 individually controlled LED lights. To capture a user's hand under varying viewpoints quickly enough to avoid tiring the user, we mounted nine camera units in close proximity to each other. Each camera unit consisted of a Raspberry Pi Zero W and Raspberry Camera Module 3, allowing us to capture images with a resolution of 2592×4608. Synchronous frame capture was triggered from a remote computer.

We captured the hands of 10 participants (1 asian female, 1 caucasian female, 1 asian male, 7 caucasian male) showing the three gestures. Each gesture is captured with each individual LED turned on, resulting in 61 different illumination conditions. We used nine different viewpoints per condition, resulting in a total of 5490 images per gesture (61 conditions $\times$ 9 viewpoints $\times$ 10 participants).

To acquire ground-truth illumination, we captured an HDR environment map inside the light stage using a Garmin VIRB 360 camera. We captured five images with exposure values in the range $[-2, 2]$ and combined these into an HDR panorama using Picturenaut[3]. To calibrate the panoramic camera with the capture rig, we used a calibration target to determine the intrinsics of both cameras. We then used an image target to obtain the extrinsic calibration between the panorama camera and the light stage. This calibration allowed us to align the captured panorama with the captured hand images. We applied intensity-based thresholding to segment the hand without the background. Thresholding is straightforward because the light stage background is mostly black or very dark.

4.3 Qualitative results

To visualize the estimated illumination and compare it with the ground truth, we rendered the SH environment probes as spherical images. The rendering determines a direction for each texel and samples the SH basis to determine the texel color. We compared the resulting illumination by computing the per-pixel difference and color-coded it using a jet colormap.

The predicted illumination was used to render the hands again for comparison. This comparison allows us to determine the impact of the illumination on the rendered 3D model of the hand. We also compute the same colormap to visualize the error between the estimation and the ground truth. See Figure 4 for sample results of the bloom, fist, and pinch gestures.

To evaluate the results, we computed the error between the rendered object and the real object. To accommodate the error introduced by the 3D reconstruction of the virtual object (geometry and material), we compared the resulting light probes directly. We used spherical projection and compared the light probes to the ground-truth illumination captured using a panoramic camera. The captured panorama is projected to the SH basis (Section 4.1).

[3] https://www.picturenaut.de

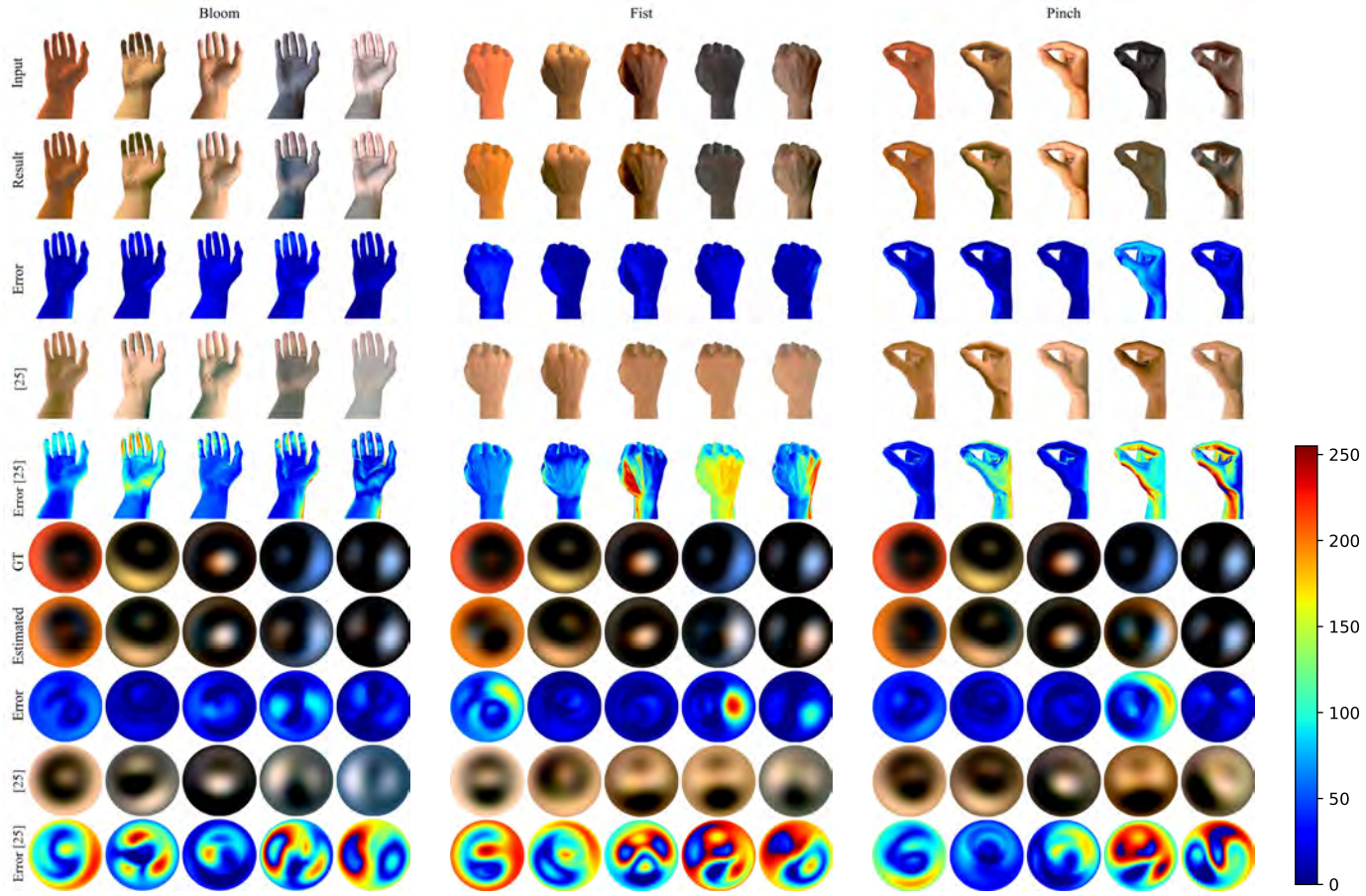


Fig. 6: Qualitative comparison of the three gestures using synthetic data. The first row is the input to the network. The second row shows the estimated illumination applied to the hand model. The third row shows the per-pixel error between the ground-truth rendering of the hand and the predicted rendering. The fourth and fifth rows show the results of Marques et al. [25]. The sixth to eighth row shows a visualization of the ground truth and predicted SH environment maps for HandLight and the per-pixel differences between them. The ninth and tenth rows show the SH environment and the per-pixel difference to the ground truth for Marques et al. [25].

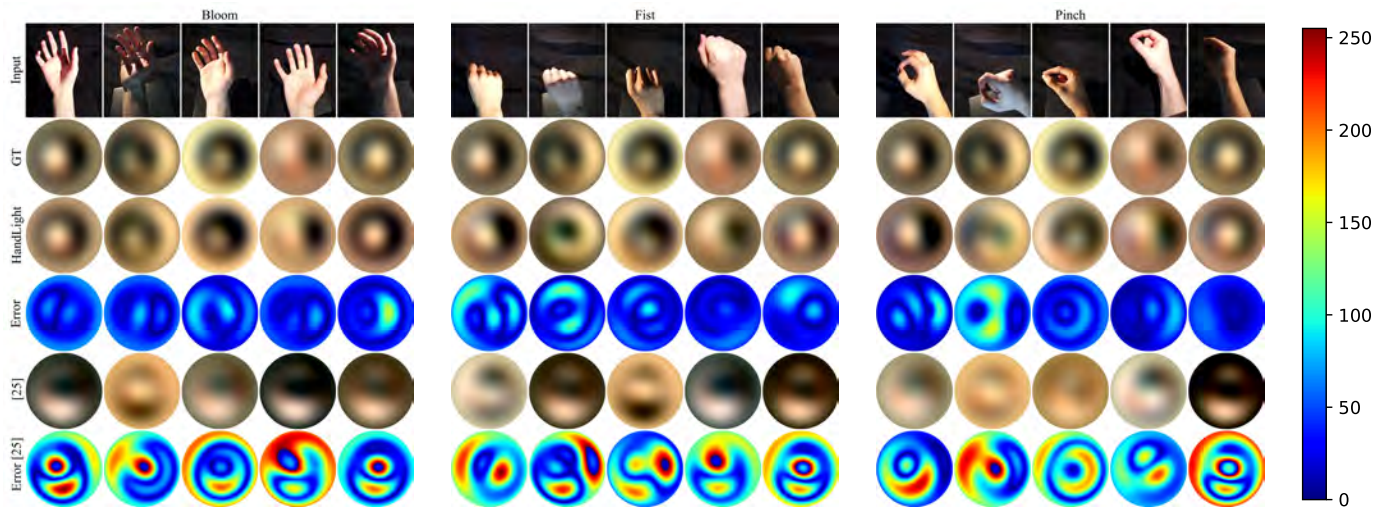


Fig. 7: Qualitative comparison of the three gestures using captured real data. The first row is the input to the network. The second row column shows the ground truth illumination that was captured inside the lightstage using a 360° camera. The third row shows the estimated illumination, and the fourth row shows the per-pixel error between ground truth and estimated SH environment maps. The fifth and sixth rows show the SH environment and the per-pixel difference to the ground truth for Marques et al. [25].

Fig. 8: Comparison of lighting a virtual object with light estimated by different methods. The first column shows hands illuminated in the light stage; the second shows the object being illuminated by the ground truth illumination that was captured inside the lightstage using a 360° camera; the third shows the illumination estimated by HandLight; the fourth shows the illumination estimated with Marques et al. [25].

HandLight achieves superior performance compared to the method of Marques et al. [24] on images of synthetic (Figure 6) and real (Figure 7) hands. To further evaluate the results on the real data set, we used the resulting illumination to illuminate a 3D model. We illuminate the model with the captured ground truth illumination as well as the estimation results with HandLight. Finally, we also illuminate the object with illumination estimated with the approach of Marques et al. [25]. As can be seen in Figure 8, our method correctly recovers the direction of the light while the results of Marques et al. lead to incorrect object illumination and degraded appearance.

4.4 Quantitative results

To quantify our performance, we compared the resulting illumination by computing MSE on the SH coefficients. We also compared the rendered images using the estimated illumination with ground-truth images. The metrics used are PSNR, SSIM, and LPIPS. See Table 1 for results on the synthetic dataset. The table shows that the bloom gesture has the lowest error with respect to the light coefficients on the LPIPS measure, while fist and pinch show a lower PSNR error. This observation can be explained by the bloom gesture covering the largest area in image space among the tested gestures, resulting in the lowest PSNR.

Since there are several works related to light estimation, we selected the one closest to our method, which is the work of Marques et al. [25] to compare to. We achieve better results in all metrics on the synthetic dataset as well as on the real data. Since Marques trains a ResNet on all gestures instead of having one for each interaction gesture, the overall performance is worse compared to our specialized networks.

4.5 Ablation study

To evaluate our design choices, we compared different parameters for both data generation and augmentation. First, we tested the impact of different parameters in light randomization and compared it to the performance when using real-world data in terms of panoramic images. We ran three tests: (1) random color variations, (2) random pose variations, and (3) mixing real with synthetic data.

For the first study, we always used the same set of real-life illumination tests applied to the hand. We tested three different random color variations: First, we tried uniform colors, i.e., the color of the SH probe coefficients is chosen randomly from a uniform distribution. Second, we used a normal distribution while still choosing the value for each color channel independently. Third, we use random color temperature from a range of 3k - 10k Kelvin, which we convert to RGB colors. We report the errors in all three variations using the same error metrics as in the quantitative evaluation. See Table 2 for the values, and Figure 9a for a qualitative example of the variations.

In the second ablation study, we compared different parameters to establish the pose variations of our gestures. For this study, we set the amount of rotational offset of the hand to 5°, 10°, and 15°. For the fingers, we use slight random rotations by 5° along the primary axis for all segments and 2° left and right for the first finger segment. We used the bloom gesture because it induces the strongest hand variations among the tested gestures. We trained three networks with the aforementioned rotations and computed the errors on the test set for each. See Table 2 for the results.

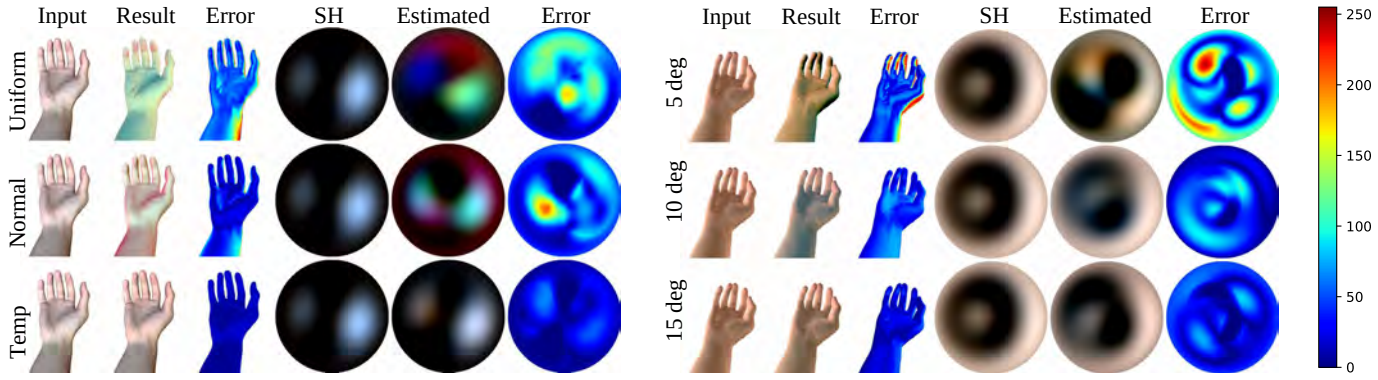
To emphasise the impact of the pose variation on the performance, we also conducted an experiment where we changed the hand rotation around the y-axis in one-degree steps while testing all illuminations from our test set in each rotation. Subsequently, we use the resulting renderings to run the estimation with our model and compare the result to the ground truth illumination. To understand the error based on the rotation, we compute the MSE between the estimation and the groundtruth illumination. The plot in Figure 10 shows for each rotation the average error from all lighting conditions in the dataset. The plot indicates that our system is providing stable results for up to 20 degrees of rotation. At approximately 20 degrees, the error starts increasing. This is expected as we train with variations of 15 degrees. Interestingly, the error decreases for the pinch gesture at rotations of about 70–90 degrees. At such large rotations, the fingers are not visible in the pinch gesture anymore, so the system relies on information from the back of the hand only. Thus, we believe that false rotations of the finger have a higher impact on the error than misinterpretations of the back of the hand. This hypothesis is further supported by the fact that the pinch gesture shows higher error for all rotations compared to the fist gesture. We also observe that the pinch gesture shows higher error at all rotations below 70 degrees. This may result from the pinch gesture's smaller surface area and greater structural variation, which makes it more sensitive to differences between the training data and the images seen by the system at runtime. Thus, we suggest designing MR interactions around a fist or similar gesture that provides a larger

Table 1: Errors computed on the test set with panoramic images.

		Synthetic				Real
System	Gesture	PSNR↑	SSIM↑	LPIPS↓	MSE↓	MSE↓
Ours	Bloom	**31.92**	**0.97**	**0.018**	**0.186**	**0.891**
	Fist	**33.69**	**0.97**	**0.021**	**0.205**	**0.946**
	Pinch	**33.53**	**0.97**	**0.019**	**0.199**	**0.764**
[25]	Bloom	20.59	0.91	0.058	0.255	1.171
	Fist	19.93	0.89	0.056	0.22	1.076
	Pinch	22.86	0.9	0.056	0.232	1.175

Table 2: Errors computed on the test set with panoramic images

Color variation	PSNR↑	SSIM↑	LPIPS↓	MSE↓
Normal	25.44	0.923	0.055	0.281
Temperature	31.12	0.958	0.04	0.197
Uniform	23.87	0.9	0.054	0.298
Pose variation				
5 degree	28.96	0.97	0.025	0.203
10 degree	27.66	0.96	0.032	0.196
15 degree	29.96	0.96	0.023	0.184



(a) Color variation (b) Pose variation

Fig. 9: Comparison of (a) light and (b) pose variations in the data generation process. The first three columns show ground-truth rendering, rendering with estimated lighting, and per-pixel error. The last three columns show the ground truth illumination, estimated illumination, and per-pixel error between them.

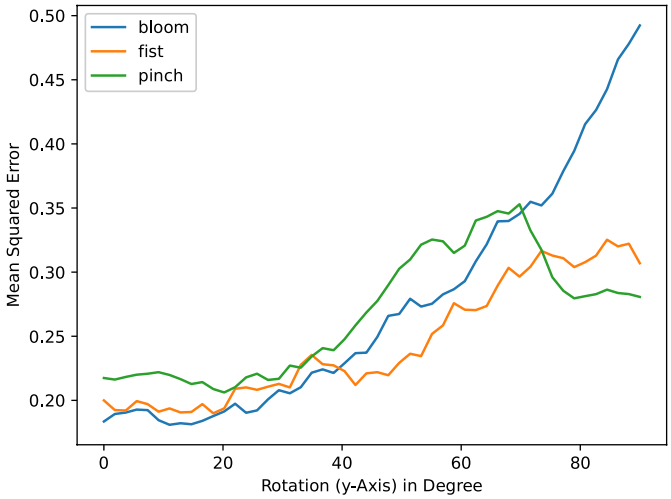


Fig. 10: Impact of pose variation on the estimation performance. Here, we rotate the Hand around the y-Axis from 0 to 90 degrees and run the inference each time.

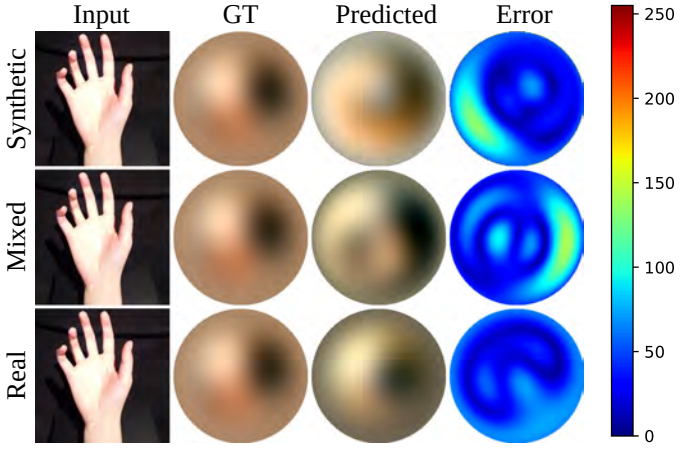


Fig. 11: Comparison of different data augmentations. The first column shows the input image, followed by the ground truth, predicted illumination, and the error map between them. The first row shows the result with a model trained on synthetic data only. In the second row, the data is augmented with real training images from the light stage. The last row shows the result when using only real data.

surface area and fewer risks of structural variations than relying on pinch gestures.

The last ablation study compares different training data augmentations. We use the real test set here for comparison. We train three different models, one with synthetic, one with real data, and a final one with mixed real and synthetic data. We chose 1000 images from the synthetic dataset and 450 images from the real dataset. Then we test these models on the real test set taken from the light stage. See Figure 11 for example results. We also compute the MSE between estimated and ground-truth coefficients on the whole test set. With purely synthetic images, we get an MSE value of 0.818, on par with the results on the real dataset. Mixed images achieved an MSE of 0.556, and with only real images, the error was 0.333. This result seems quite good for a synthetic dataset, but still shows errors in some regions. When using mixed real and synthetic data, the quality already increases. Using real training data works best, but has the disadvantage of cumbersome data capturing.

Overall, the ablation showed that our design choices make sense. The natural illumination improves the performance on real-world light settings compared to randomized coefficients. We also showed that changes in hand pose impact the performance depending on the difference from the trained poses. Thus, we improved the performance by adding pose variations in the training set for each gesture.

5 Limitations

While HandLight enables unconstrained illumination estimation, it requires the user's hand to be present and visible during gestures. As such, if the user is faced with an AR experience that requires no interaction, the system will not be able to recover the illumination of the scene. Furthermore, the generated light probes are created at the location of the hand. If users interact with distant objects, the estimated illumination may not correctly represent the conditions at the object's location.

Our evaluation focused on comparing HandLight with existing works focusing on hand-based illumination estimation. Other methods, such as inverse rendering-based illumination estimation or recovering the scene illumination from the estimated hand mesh, are feasible but present additional challenges. These methods rely on accurate modelling and recovery of the user's hand shape, pose, and reflectance. Additionally, assumptions such as light color, number of lights in the scene, or Lambertian surfaces are needed. In the future, HandLight could be compared with such methods as well as estimations from scene appearance.

Our system requires the illumination to result in the appearance changes of the gesture. When the light is directly behind the hand, the entire visible hand is in shadow, and we cannot predict the direction of the incoming light accurately. Here, combining HandLight with scene illumination estimation from camera images can improve the results.

While our system addresses most use cases, it can sometimes have problems with hand poses that diverge too much from the trained poses. This also applies to poses HandLight was not trained for. The quality of the estimation for new hand poses will depend on how close it matches one of the trained poses. For similar poses, we may still get usable estimations; however, for robust estimations of recurring new gestures, an additional model should be trained on such poses.

Skin tones HandLight was not trained on, i.e., Type V and VI on Fitzpatrick scale, could pose a challenge. In addition, strong skin variations (such as scars or burns) can pose problems with estimation. Embellishments such as rings or nail polish can introduce errors because they are not part of our data model. Such shiny areas could even be a chance to further improve the system by using the reflective properties to extract more information from specular highlights. Personalizing HandLight by incorporating information about the user's hand appearance and detecting embellishments could be incorporated in the future to further improve the performance.

To account for differences in how precisely the user performs a gesture, the dataset includes slight variations of hand poses (please see Section 3.3). However, if during interaction the user's hand pose changes slightly outside of the range of trained variations it may happen that the light estimation becomes unstable. In an interactive application, this can result in visible flickering of the estimated lights in the virtual scene. This can be addressed by estimating the confidence of the current estimation, so that we can scale down the impact of possibly wrong estimations. For example, since we know the base pose from the training set, we can compute the distance of each hand joint to the current pose. This information might be used as a confidence value of the corresponding light estimate and thus, allow rejecting or scaling down the impact of bad estimates.

6 Conclusion and Future Work

We presented HandLight, a method for estimating illumination in real environments from the user's hand interaction in MR. Our results show that it can believably approximate the current lighting, and we evaluated it on real and synthetic images.

This system can be integrated into consumer-grade MR devices to support light estimation in real environments. We have demonstrated HandLights' feasibility using an AR furniture application. Other interesting applications exist, for example, in an AR gaming context. When the user interacts with a virtual character with hand gestures, the shading on the character and other game objects can be adapted using the illumination obtained from the hands. A similar situation involves MR training of a worker who operates a complex machine using an AR headset. Here, we could use the estimated light probes to better blend virtual decorations with the machine parts.

Our work has several interesting directions for future research. Combining HandLight with illumination estimated from objects in the environment can improve the robustness of the system. Such a hybrid light estimation could deal with light sources occluded by the hand or continue to work if the hand is not visible.

Another option is to combine input from multiple cameras, which might become an option in multi-user applications in situations where the cameras of several users observe a single hand gesture from several viewpoints.

Moreover, we consider a simplified approach for extending HandLight so that it can handle arbitrary hand poses. This will especially require developing an approach for capturing new datasets that does not rely on complex hardware installations, such as the light stage.

Acknowledgments

This work was supported by Snap Inc. and the Alexander von Humboldt Foundation, funded by the German Federal Ministry of Research, Technology, and Space.

References

[1] A. Alhakamy and M. Tuceryan. Real-time illumination and visual coherence for photorealistic augmented/mixed reality. *ACM Computing Surveys*, 53(3):1–34, May 2020. doi: 10.1145/3386496 2

[2] M. Bunnell. Dynamic ambient occlusion and indirect lighting. *Gpu gems*, 2(2):223–233, 2005. 5

[3] D. A. Calian, J.-F. Lalonde, P. Gotardo, T. Simon, I. Matthews, and K. Mitchell. From faces to outdoor light probes. *Computer Graphics Forum*, 37(2):51–61, may 2018. doi: 10.1111/cgf.13341 3

[4] A. CHARDON, I. CRETOIS, and C. HOURSEAU. Skin colour typology and suntanning pathways. *International Journal of Cosmetic Science*, 13(4):191–208, Aug. 1991. doi: 10.1111/j.1467-2494.1991.tb00561.x 4

[5] R. Cupisz. Light probe interpolation using tetrahedral tessellations. In *Game Developers Conference (GDC)*, 2012. 5

[6] P. Debevec. Rendering synthetic objects into real scenes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*. ACM Press, 1998. doi: 10.1145/280814.280864 2

[7] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, SIGGRAPH '97, pp. 369–378. ACM Press, 1997. doi: 10.1145/258734.258884 1, 2

[8] F. Einabadi, J.-Y. Guillemaut, and A. Hilton. Deep neural models for illumination estimation and relighting: A survey. In *Computer Graphics Forum*, vol. 40, pp. 315–331. Wiley Online Library, 2021. 2

[9] M.-A. Gardner, Y. Hold-Geoffroy, K. Sunkavalli, C. Gagne, and J.-F. Lalonde. Deep parametric indoor lighting estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. doi: 10.1109/iccv.2019.00727 5

[10] M.-A. Gardner, K. Sunkavalli, E. Yumer, X. Shen, E. Gambaretto, C. Gagné, and J.-F. Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics*, 36(6):1–14, nov 2017. doi: 10.1145/3130800.3130891 1, 2

[11] R. Green. Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference (Vol. 56, p. 4–51)*, 2003. 2

[12] L. Gruber, T. Langlotz, P. Sen, T. Hoellerer, and D. Schmalstieg. Efficient and robust radiance transfer for probeless photorealistic augmented reality. In *2014 IEEE Virtual Reality (VR)*. IEEE, mar 2014. doi: 10.1109/vr.2014.6802044 2

[13] L. Gruber, T. Richter-Trummer, and D. Schmalstieg. Real-time photometric registration from arbitrary geometry. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, nov 2012. doi: 10.1109/ismar.2012.6402548 1, 2

[14] L. Gruber, J. Ventura, and D. Schmalstieg. Image-space illumination for augmented reality in dynamic environments. In *2015 IEEE Virtual Reality (VR)*. IEEE, mar 2015. doi: 10.1109/vr.2015.7223334 2

[15] V. Gupta and V. K. Sharma. Skin typing: Fitzpatrick grading and others. *Clinics in Dermatology*, 37(5):430–436, Sept. 2019. doi: 10.1016/j.clindermatol.2019.07.010 4

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. doi: 10.1109/cvpr.2016.90 3

[17] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde. Deep outdoor illumination estimation. In *Proceedigns of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. doi: 10.1109/cvpr.2017.255 1, 2

[18] J. Jachnik, R. A. Newcombe, and A. J. Davison. Real-time surface light-field capture for augmentation of planar specular surfaces. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, nov 2012. doi: 10.1109/ismar.2012.6402544 2

[19] M. Kanbara and N. Yokoya. Real-time estimation of light source environment for photorealistic augmented reality. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pp. 911–914 Vol.2. IEEE, 2004. doi: 10.1109/icpr.2004.1334407 1, 2

[20] S. B. Knorr and D. Kurz. Real-time illumination estimation from faces for coherent rendering. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, sep 2014. doi: 10.1109/ismar.2014.6948416 3

[21] M. Krystek. An algorithm to calculate correlated colour temperature. *Color Research and Application*, 10(1):38–40, Mar. 1985. doi: 10.1002/col.5080100109 4

[22] C. LeGendre, W.-C. Ma, R. Pandey, S. Fanello, C. Rhemann, J. Dourgarian, J. Busch, and P. Debevec. Learning illumination from diverse portraits. In *SIGGRAPH Asia 2020 Technical Communications*. ACM, nov 2020. doi: 10.1145/3410700.3425432 3

[23] D. Mandl, K. M. Yi, P. Mohr, P. M. Roth, P. Fua, V. Lepetit, D. Schmalstieg, and D. Kalkofen. Learning lightprobes for mixed reality illumination. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, oct 2017. doi: 10.1109/ismar.2017.25 1, 2, 3

[24] B. A. Marques, R. R. Drumond, C. N. Vasconcelos, and E. Clua. Deep light source estimation for mixed reality. In *VISIGRAPP (1: GRAPP)*, 2018. 2, 3, 7

[25] B. A. D. Marques, E. W. G. Clua, and C. N. Vasconcelos. Deep spherical harmonics light probe estimator for mixed reality games. *Computers and Graphics*, 76:96–106, Nov. 2018. doi: 10.1016/j.cag.2018.09.003 3, 5, 6, 7

[26] G. Moon, S. Saito, W. Xu, R. Joshi, J. Buffalini, H. Bellan, N. Rosen, J. Richardson, M. Mize, P. De Bree, et al. A dataset of relighted 3d interacting hands. *Advances in Neural Information Processing Systems*, 36:17689–17701, 2023. 4

[27] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*, pp. 548–564. Springer, 2020. 4

[28] K. Nishino and S. K. Nayar. Eyes for relighting. *ACM Transactions on Graphics*, 23(3):704–711, Aug. 2004. doi: 10.1145/1015706.1015783 3

[29] A. Panagopoulos, D. Samaras, and N. Paragios. Robust shadow and illumination estimation using a mixture model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 651–658. IEEE, 2009. 2

[30] P. Phongthawee, W. Chinchuthakun, N. Sinsunthithet, V. Jampani, A. Raj, P. Khungurn, and S. Suwajanakorn. Diffusionlight: Light probes for free by painting a chrome ball. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 98–108, 2024. 1, 2

[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*. ACM, aug 2001. doi: 10.1145/383259.383317 2

[32] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, July 2002. doi: 10.1145/566654.566575 4

[33] T. Richter-Trummer, D. Kalkofen, J. Park, and D. Schmalstieg. Instant mixed reality lighting from casual scanning. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, sep 2016. doi: 10.1109/ismar.2016.18 1

[34] K. Rohmer, J. Jendersie, and T. Grosch. Natural environment illumination: Coherent interactive augmented reality for mobile and non-mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2474–2484, nov 2017. doi: 10.1109/tvcg.2017.2734426 1, 2

[35] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):290–300, Mar. 2003. doi: 10.1109/tpami.2003.1182093 2

[36] G. Somanath and D. Kurz. Hdr environment map estimation for real-time augmented reality. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2021. doi: 10.1109/cvpr46437.2021.01114 1, 2

[37] S. Song and T. Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019. doi: 10.1109/cvpr.2019.00708 2

[38] T. Takai, K. Niinuma, A. Maki, and T. Matsuyama. Difference sphere: an approach to near light source estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. 98–105. IEEE, 2004. doi: 10.1109/cvpr.2004.1315019 2

[39] L. Wang, R. Li, X. Shi, L.-Q. Yan, and Z. Li. Foveated instant radiosity. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, nov 2020. doi: 10.1109/ismar50242.2020.00017 3

[40] H. Weber, D. Prevost, and J.-F. Lalonde. Learning to estimate indoor lighting from 3d objects. In *2018 International Conference on 3D Vision (3DV)*. IEEE, sep 2018. doi: 10.1109/3dv.2018.00032 1

[41] X. Wei, G. Chen, Y. Dong, S. Lin, and X. Tong. Object-based illumination estimation with rendering-aware neural networks. In *Computer Vision – ECCV 2020*, pp. 380–396. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58555-6_23 2

[42] R. Yi, C. Zhu, P. Tan, and S. Lin. *Faces as Lighting Probes via Unsupervised Deep Highlight Extraction*, pp. 321–338. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01240-3_20 3

[43] H.-X. Yu, S. Agarwala, C. Herrmann, R. Szeliski, N. Snavely, J. Wu, and D. Sun. Accidental light probes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12521–12530. IEEE, June 2023. doi: 10.1109/cvpr52729.2023.01205 1, 2

[44] S. Zainali, S. Ma Lu, B. Stridh, A. Avelin, S. Amaducci, M. Colauzzi, and P. E. Campana. Direct and diffuse shading factors modelling for the most representative agrivoltaic system layouts. *Applied Energy*, 339:120981, 2023. doi: 10.1016/j.apenergy.2023.120981 5

[45] Y. Zhao, M. Dasari, and T. Guo. Clear: Robust context-guided generative lighting estimation for mobile augmented reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–26, 2025. 2

[46] Y. Zhao and T. Guo. *PointAR: Efficient Lighting Estimation for Mobile Augmented Reality*, pp. 678–693. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58592-1_40 2

[47] Y. Zhao, C. Ma, H. Huang, and T. Guo. Litar: Visually coherent lighting for mobile augmented reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–29, Sept. 2022. doi: 10.1145/3550291 1, 2