LiteAR: A Framework to Estimate Lighting for Mixed Reality Sessions for Enhanced Realism

Chinmay Raut[1], Anamitra Mani[2](✉), Lakshmi Priya Muraleedharan[2], and Raghavan Velappan[2]

[1] Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India
[2] Samsung Research Institute India Bangalore, Bagmane Constellation Business Park, Bengaluru 560037, Karnataka, India
{anam.mani,lakshmi.m,raghavan.v}@samsung.com

Abstract. We propose an end-to-end learning based method to estimate irradiance in real-time given a single input limited field of view image from a mobile phone camera. We further develop a technique inspired by physically based rendering to take advantage of spatially varying environment to illuminate virtual objects in augmented reality sessions to make them look more realistic. We integrate the Inertial Measurement Unit sensor to dynamically estimate illumination, making the mixed reality experience interactive. Our solution runs in real-time on mobile phones, with significantly lower computational requirements and enhanced realism in comparison to state-of-the-art methods.

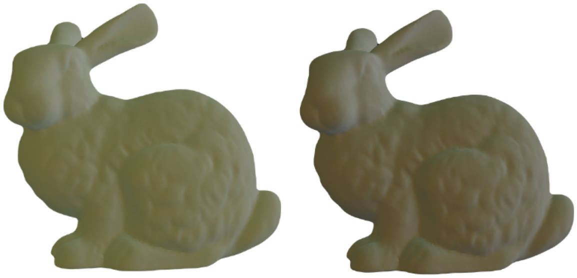
Keywords: Illumination estimation · Augmented reality · Mobile mixed reality

1 Introduction

One of the main challenges in making augmented reality accessible is to make it seem as realistic as possible. Virtual objects should be indistinguishable from the real world in AR sessions. Lighting plays a major role in rendering objects realistically. While direct light is important in rendering shadows, indirect light is essential for realistic renders. The environment surrounding an object acts as an indirect light source and hence contributes to the illumination of the object. It is extremely important to estimate this diffuse lighting accurately for realistic rendering to enhance augmented reality experiences. Image-based lighting is a physically-based rendering method to illuminate objects using an environment map.

In the context of AR, a panorama surrounding the AR object can be used as an environment map. However, on a mobile phone, the camera captures only a small fraction of the panorama. Therefore, it is very difficult to predict the complete environment map from camera images. Recent works propose a learning based method to predict illumination using a single input image from the rear camera. However, most of these methods assume that the mobile phone is at the

N. Magnenat-Thalmann et al. (Eds.): CGI 2022, LNCS 13443, pp. 407–423, 2022.
https://doi.org/10.1007/978-3-031-23473-6_32



(a) Ground truth model

(b) Model with spatially-aware illumination

Fig. 1. The environmentally-aware illumination of the object enhances its realism in an AR scene.

centre of the environment and thus predict an environment map surrounding the phone. While this may work for most cases, it is not necessarily true as often the virtual object to be placed in an AR session is placed away from the mobile phone camera. Sometimes the real-world objects surrounding the virtual object may alter the environment map. Thus, for realistic rendering it is essential to take into account the immediate environment surrounding the virtual object. Figure 1(b) shows how the environmentally aware illumination enhances the realism of a virtual object in Fig. 1(a).

Our contribution includes,

1. A framework to estimate illumination in real-time for augmented reality experiences on mobile phones by representing the dynamically changing irradiance map as a set of spherical harmonics and training a light-weight neural network on the same.
2. Utilizing scene geometry estimation to update the object's local environment and using this information to enhance object illumination for realism.
3. Use of the Inertial Measurement Unit (IMU) sensor present in the phone to update lighting instead of relying on calling the neural network per frame, which in turn reduces computational cost while achieving realistic illumination.

2 Related Work

After demonstrating a way to capture illumination using a mirrored sphere, earlier works in illumination estimation use a light probe to predict scene lighting. Debevec [5] presented a way to construct an omnidirectional HDR using multiple photographs of a mirrored sphere taken under varying exposures. This HDR can in turn be used to render additional objects in the scene. Prakash et al. [17] use a specular sphere to sample radiance for mobile augmented reality. Debevec presented a way to capture illumination using hybrid 3D spheres [6]. Beyond mirrored balls, known 3D objects have also been used to estimate illumination. Mandl [15] used a combination of pose estimation and illumination estimation neural networks to accurately estimate lighting using a light probe. Calian et al. [2] made use of human faces to predict illumination. However, for mobile AR experiences, the necessity of having a known light probe in the environment ruins the user experience. Thus, we need a probeless illumination estimation method.

Apart from using single object probes, scene properties have also been used to explore illumination estimation. The scene appearance is determined by a variety of factors like the scene geometry, material properties, lighting, etc. One way to estimate scene illumination is to optimize these properties to find the best representation of the scene. However, with limited inputs, the problem becomes an under-constrained optimization problem, and thus the probability of the error multiplying is high. Thus, when using an optimization method, the work either makes certain assumptions about the scene or expects the user to manually provide ground truth. Karsch et al. [11] expect user annotations to estimate initial geometry and lighting. Zhang et al. [22] require depth information and expect users to manually provide ground truth for lightsource locations. One more method matches the image to the most similar cropped image from a database of panoramas, assuming that similar images share illumination estimates. Although probe based techniques produce good results, they are not practical for commercial mobile augmented reality since they require the presence of a light probe in the scene. Recent work has explored end-to-end neural network based solutions to estimate illumination based on input images and additional information.

Most recently, learning based methods have been found to produce seamless augmented reality experiences. Gardner et al. [7] proposed a learning-based method that predicts indoor illumination based on a single image. Their network contained global and local branches and was trained on LDR panoramas of indoor scenes. They further used 2100 HDR panoramas to fine tune the model. They do not take into consideration depth data, and therefore they fail to capture spatially varying lighting information. However, their method is considered state-of-the-art for indoor illumination estimation. Cheng et al. [4] utilize views from both the front and rear cameras of mobile devices to train a neural network with two different branches concatenating to produce spherical harmonics. However, the model is not optimized and is not suitable to run in real-time. Legendre et al. [12] create their own dataset by capturing illumination information through a mirrored ball along with the image from the rear camera. They formulate the problem so as to output the HDR image containing lighting information using the cropped input image from the rear camera. They use L2 loss and discriminatory loss to fine tune the network. Deeplight et al. [12] capture illumination using a mirrored sphere placed 60 cm in front of the camera. However, the placement of the virtual objects rarely coincides with that. Often, they are placed on surfaces visible in the scene and are closely surrounded by other real objects. They do not take into consideration spatially varying lighting and therefore fail to capture true illumination at the local position.

Song et al. [19] proposed a fully differential modular network consisting of 4 components, namely: geometry estimation, scene completion, and LDR-to-HDR estimation. By splitting the network into four components, it becomes easier to optimize the modules individually and, consequently, the whole network. However, the complete model becomes bulky and it is difficult to run in real-time on a mobile phone. Zhao et al. [23] calculate spherical harmonics directly from point cloud data [14] inspired from Monte Carlo integration [18]. They expect rgb-d data as input and warp the point cloud data to a global panorama. Although predicting SH directly from point cloud data reduces the complexity of the model, the model requires RGB-D input, which in itself is sparse in nature.

Other recent works for immersion during interaction in VR include tracking and rendering of contacts with tangible objects in VR [20], Recently mixture graphs [1] have been designed to compute correctly pre-filtered volume lighting. An efficient approach for high quality GPU-based rendering of line data with ambient occlusion and transparency effects has been discussed in [9].

Considering the existing methods for estimating illumination, they are still far from solving the illumination estimation problem for augmented reality scenes for mobile environment. The probe base method ruins the user experience for mobile AR users and hence is impractical. The scene-property based illumination estimation methods expect user intervention in terms of initial geometry and light source estimations. Some of the learning based methods do produce seamless augmented reality experiences. However, the majority of them are not suitable for real-time operation or are not practical for mobile augmented reality experiences. Learning based methods rely on neural networks to dynamically update lighting. However, depending on the complexity of the model, using a neural network might not be suitable to run per frame because of the required computational power. When considering mixed reality applications for mobile, we can also make use of other sensors present in the device to make the process computationally lighter. We propose a learning based method that integrates depth sensors and IMU sensors for dynamic lighting.

3 Illumination Estimation

Figure 2 illustrates the complete proposed pipeline for illumination estimation for mobile augmented reality. The input module has three streams, one each for camera input, which is passed through a neural network to produce global spherical harmonics; depth input, which is used to update lighting based on the local environment of the virtual object; and the angle of rotation of the phone about the vertical axis whenever the user rotates the phone. The input data from the camera is processed to estimate global spherical harmonics by passing it through the neural network model trained using the Matterport 3D dataset [3]. A depth image is used to obtain point cloud data, which is further used to update global spherical harmonics based on the immediate local environment surrounding the virtual object. We also keep track of the rotation of the mobile phone to update spherical harmonics using fast spherical harmonics rotation.

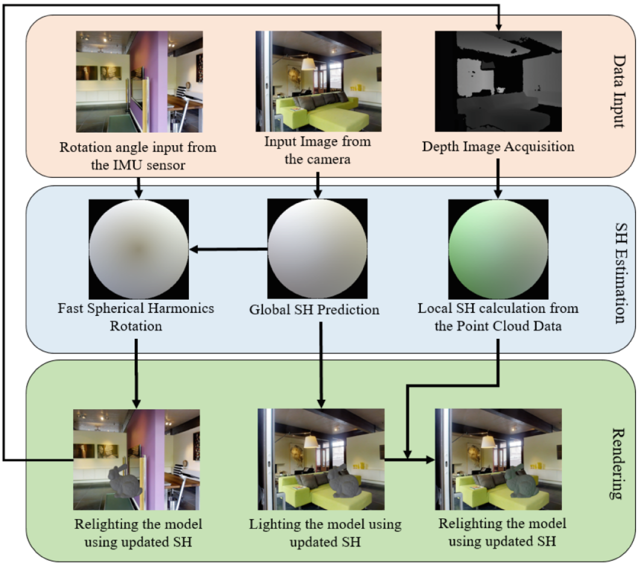

Rotation angle input from the IMU sensor
Input Image from the camera
Depth Image Acquisition
Data Input
Fast Spherical Harmonics Rotation
Global SH Prediction
Local SH calculation from the Point Cloud Data
SH Estimation
Relighting the model using updated SH
Lighting the model using updated SH
Relighting the model using updated SH
Rendering

Fig. 2. Complete data pipeline showcasing all the modules and the flow of the proposed process

Table 1. Standard variables

Symbol	Variable
SH_{lm}	Spherical harmonics coefficient l of band l
L	Radiance at the point
R	Radius of the sphere we query points from
r	Distance of a point from the center of the sphere
$sign(d)$	$sign(d)$ function outputs -1 or 1 depending on which side of the center the point lies along axis d
SH_g	Global Spherical harmonics coefficients
SH_l	Local Spherical harmonics coefficients
D	Maximum distance between any two points in the point cloud dataset
y	SH band 2 with 5 componentes that we want to rotate
P	A function which projects a normal vector into the second band of spherical harmonics. It takes a normalized three dimensional vector as input and outputs a 5 dimensional SH vector
M	3×3 rotation matrix. It's the rotation that we want to somehow apply to our SH vector
U	The 5×5 (unknown) rotation matrix that want to apply to y
N	Set of five three-dimensional normalized vectors

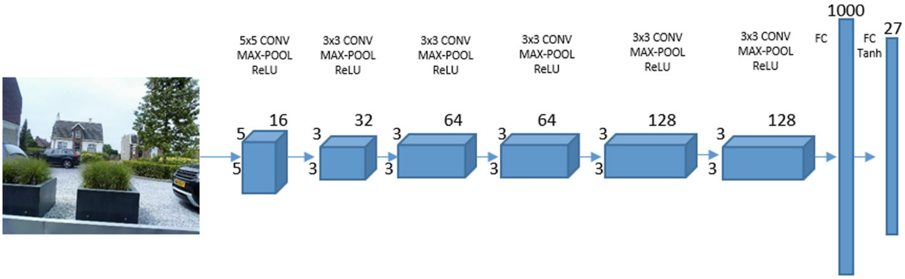


Fig. 3. Network diagram to predict global spherical harmonics. We use blocks of convolutional and max-pool layers along with ReLU activation function followed by two fully connected layers. We use Tanh activation function before the last layer to restrict output between −1 and 1.

We explain each step in detail in this section using four subsections, namely: data preparation, global spherical harmonics prediction, spatially varying environment, and spherical harmonics rotation. We define standard variables and their symbols in Table 1.

3.1 Data Preparation

We use publicly available Matterport 3D dataset [3] consisting of panoramas of indoor scenes and viewpoint images (rear camera images) for the same. One set of observations consists of a rear camera image and a panorama. There are a total of 10,800 panoramic scenes in the dataset. Each panorama contains 18 viewpoint images taken from multiple angles. For training purposes we choose 6 viewpoints separated by 60° each with vertical camera alignment.

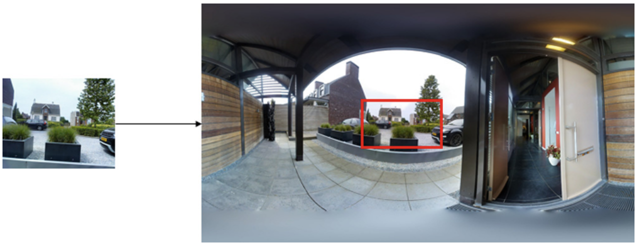
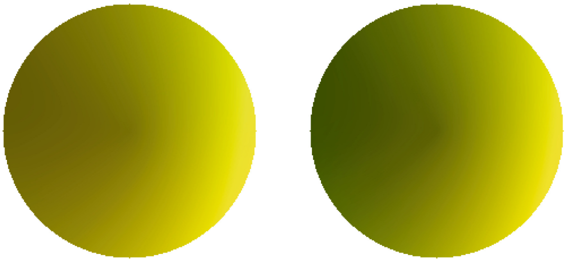


Fig. 4. Input from rear camera and corresponding panorama

Figure 4 depicts one instance of the dataset containing the rear camera input image used as training data and the corresponding panorama from which we calculate spherical harmonics. We split the data into training and validation sets with a ratio of 3:1. To enhance the dataset, we add color tints so that there is enough variance in the dataset.

3.2 Global Spherical Harmonics Prediction

We use an image captured using the rear camera as the input and produce nine spherical harmonics for each channel. We formulate this as a regression problem and utilize convolutional neural network followed by fully connected layers. We begin by resizing the image to 480$*$320 pixels. We design a block of convolutional layer followed by a max-pool layer with ReLU activation function. We increase the depth of each layer to extract more features in subsequent layers. Then, afterward, to reduce the output to 27 components, we use fully connected layers. Finally, we pass the output through the Tanh function to restrict predicted coefficients between -1 and 1. Figure 3 demonstrates the model architecture.



(a) Sphere $\hat{S}$ using predicted SH coefficients

(b) Sphere S rendered using ground truth SH coefficients.

Fig. 5. We calculate l2 loss between these images and aim to minimize it along with minimizing the l2 loss between SH coefficients themselves.

We use mean squared error on predicted SH coefficients as our loss function and ADAM as our optimizer. A small difference in SH coefficients can lead to a significant change in illumination. Using only 27 coefficients might be an under-constrained problem that can lead to errors in illumination estimation. As shown in Fig. 5, we introduce render loss as an additional loss function in which we render a sphere $hatS$ using predicted SH coefficients and calculate $l2$ loss with respect to the sphere S rendered using ground truth SH coefficients.

3.3 Spatially Varying Environment

We must take into account the spatially varying environment, especially around the virtual object, to make the augmented reality experience more realistic. Another popular problem in mixed reality is geometric estimation of the scene. A lot of newer mobile devices have an integrated Lidar sensor to capture depth images. For devices without a depth sensor, several approaches like estimating structure from motion with the help of camera images from multiple angles and sensor data [21], estimating depth from a single image [10,13] have filled in the role of depth estimation. Since most mixed reality sessions devote certain computational power to geometry estimation, we can leverage the same for realistic relighting of the virtual objects placed in the scene.

The inspiration for relighting the object from local point cloud data comes from Monte Carlo integration, wherein we treat every point queried from a sphere of certain radius surrounding the virtual object as a point light source. However, since the distance between these points and the virtual object is less, we approximate integration to summation. We first downsample the data uniformly. We arranged the point cloud data in a K-Dimensional tree (KDTree) data structure [8]. The time complexity for querying neighbours is reduced from N to $logN$. We query all the points lying in a sphere of a certain radius. We experiment with different values of this radius.

We focus on updating the spherical harmonic coefficients of the first two bands. We calculate irradiance in the form of spherical harmonic coefficients using the colour of the point and its distance from the object. To obtain the local SH coefficient, we integrate weighted irradiance based on distance over all of the points in the ball point query. Equations 1–3 use queried points and their radiance values to update the local spherical harmonics of band 1.

$$SH_{10} = \sum (L * (R - r)/R) * sign(x) \tag{1}$$

$$SH_{11} = \sum (L * (R - r)/R) * sign(y) \tag{2}$$

$$SH_{12} = \sum (L * (R - r)/R) * sign(z) \tag{3}$$

where R is the radius of the sphere we query points from. r is the distance of a point from the centre of the sphere. These local coefficients are used to update global SH coefficients based on a distance measure, as shown in Eq. 5. We use alpha as the measure of distance, which is calculated using Eq. 4.

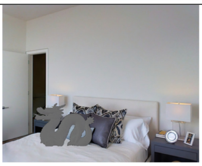
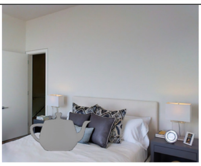
$$Alpha = R/D \tag{4}$$

$$SH_g = alpha * SH_g + (1 - alpha) * SH_l \tag{5}$$

3.4 Spherical Harmonics Rotation

Panoramic images capture more details in the horizontal direction since the distribution of radiance varies more in the horizontal direction. In a mobile mixed reality session, the user places a virtual object in the scene captured by the camera. After placing the object, the user might move around the object, but the surrounding environment would remain the same. Thus, the only way object illumination would change is if the object is moved and placed somewhere else or if there is some change in the environment. To keep track of the scene, we use sparse optical flow. Even if the scene itself does not change, if the user moves around the object, the illumination would change because of the rotation.

Table 2. LiteAR renders of various models (a) bunny, (b) dragon, (c) teapot, and (d) Lucy in various environments. The model takes the image as an input and produces 27 spherical harmonics coefficients as irradiance.

	Bunny	Dragon	Teapot	Lucy
Scene 1				
Scene 2				
Scene 3				
Scene 4				

We track the rotation around the vertical axis and update spherical harmonics accordingly. We utilize the IMU sensor present in mobile phones to track rotation. Thus, instead of calling the neural network every frame, we rotate the environment map whenever the user rotates the phone about the vertical axis triggered by the IMU sensor. We implement zonal harmonics for fast spherical harmonics rotation calculation [16]. Using zonal harmonics, the total number of multiplication operations required for spherical harmonics rotation per channel is 118, which is significantly less than the 120 million multiplication operations required in a neural network. Furthermore, since we do not care about zonal harmonics themselves and only care about spherical harmonics rotation, we can make use of sparse data and formulate the rotation problem as finding the rotation matrices for each spherical harmonics band.

1. The first band does not change with rotation as its value remains constant.
2. The second band can be treated as a vector, which can be rotated by pre-multiplying by a rotation matrix corresponding to the angle of rotation.
3. The third band has five components. To find a rotation matrix for this band, we make use of the fact that rotation followed by projection is the same as projection followed by rotation. We demonstrate this using Eq. 6.

$$U * P(N) = P(M * N) \tag{6}$$

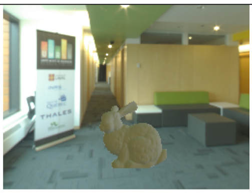
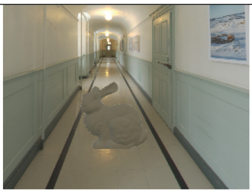
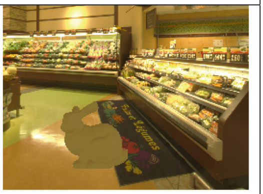
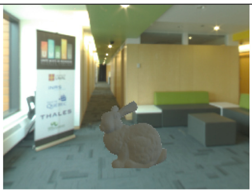
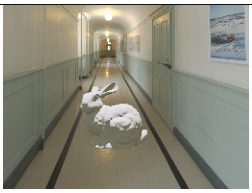
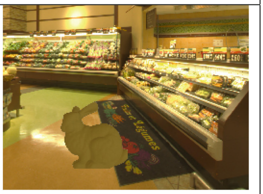
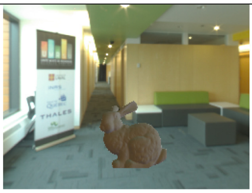
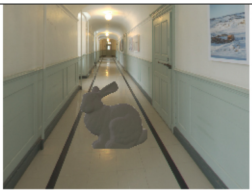
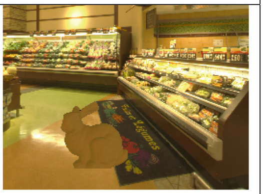
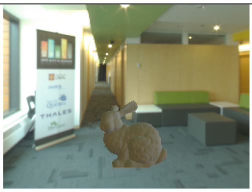
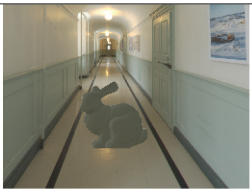
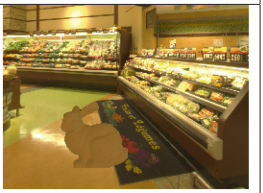
$$U * y = [P(M * N)] * P(N)^{-1} * y \tag{7}$$

We have to solve for U. We can choose N to be a set of five unit vectors as long as the projections of those vectors are linearly independent in order to solve for U. $U * y$ gives us the value of rotated spherical harmonics in the second band as shown in Eq. 7.

3.5 Computational Analysis

The model is designed to have less than 120M multiplication and accumulation functions to make it mobile friendly. Spherical Harmonics rotation only requires 118 multiplication operation thus is very cheap computationally. Updating local lighting based on the immediate environment depends on the density of point cloud data. We first down sample the point cloud data since reducing density eliminates redundancy with negligible change in the results. Our neural network model runs at 30 FPS using Intel(R) Core(TM) i7-6700HQ CPU.

Table 3. Comparison of global illumination estimation by different models with the ground truth. We use various learning based methods to render the Stanford bunny and demonstrate the same to compare realism.

	Scene 1	Scene 2	Scene 3
Input Image			
Ground Truth			
Gardner[7]			
Deeplight[12]			
LiteAR (ours)			

4 Results and Discussion

With 10,800 panoramic scenes and 6 viewpoints for each scene, we have 64,000 distinct observations in our dataset. To improve the illumination variance, we add color tints to produce two more sets of observations for every one set of observations. We split the dataset into training and testing data respectively, with a 75%–25% split. We train our model LiteAR on this data and evaluate the results. Table 2 shows renders of different models rendered using spherical harmonics produced by our model. We compare the results to recent state-of-the-

Table 4. Comparison of rendering based on global illumination and spatially aware illumination. We use depth image to obtain point cloud data which is further used to update global spherical harmonics.

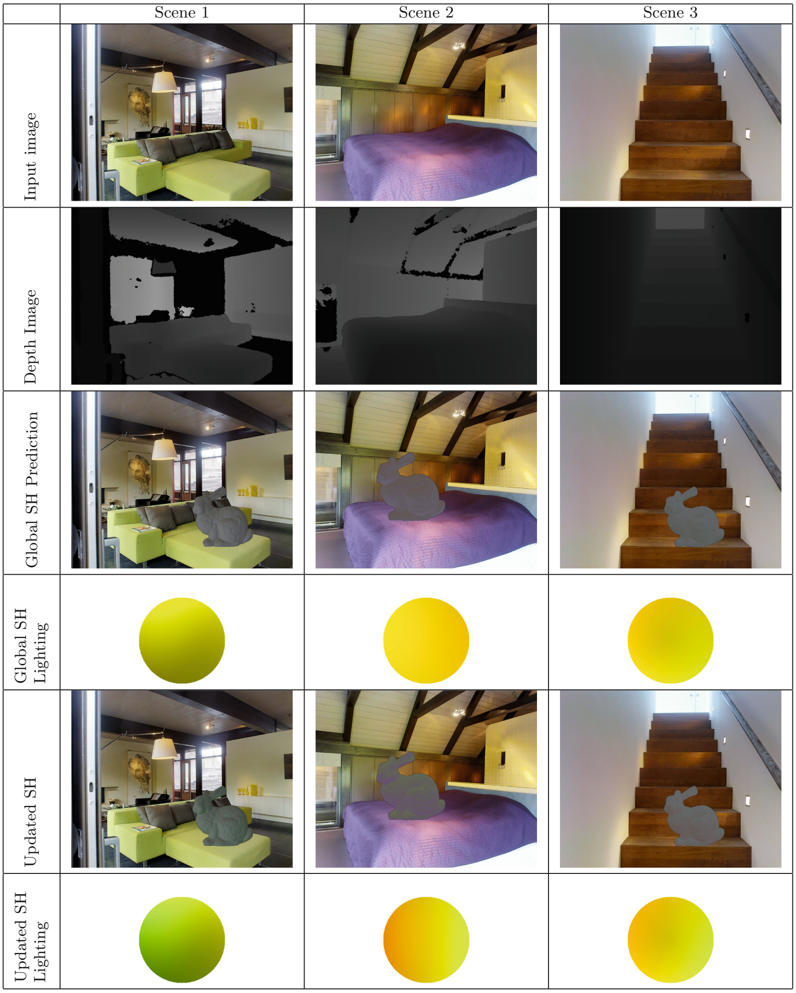
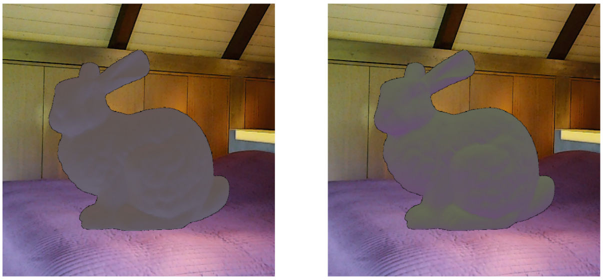


Table 5. L2 loss on global spherical harmonics for different models

Model	l2 loss
Gardner [7]	0.18
Deeplight [12]	0.28
PointAR [23]	0.21
LiteAR (ours)	0.24

Table 6. Computational complexity in terms of the number of parameters and multiply accumulates (MACs) where M stands for million

	Number of parameters (M)	MACs (M)
Gardner [7]	20	3800
Deeplight [12]	3.5	300
PointAR [23]	1.4	790
LiteAR (ours)	2.2	120



(a) Global SH Render

(b) Spatially Aware SH Render

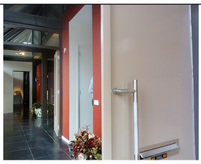
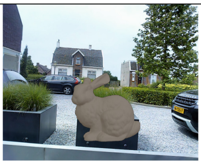
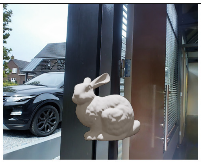
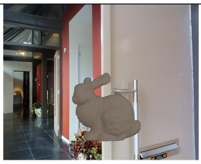
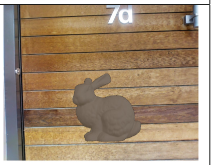
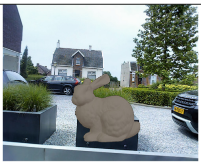
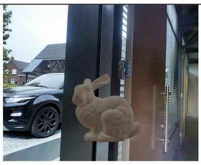
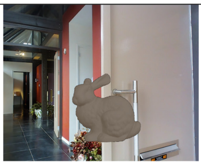
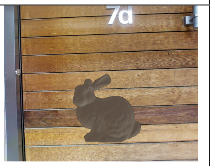
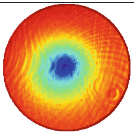
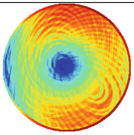
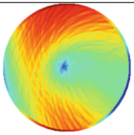
Fig. 6. (a) depicts rendering of Stanford bunny using spherical harmonics predicted by the neural network and (b) depicts the same using updated spherical harmonics after taking into consideration the immediate local environment

art lighting estimation research in Table 3. It is important to note that we target mobile AR applications and, thus, model complexity is as important a metric as accuracy. Furthermore, the objective is to improve realism. Thus, making the model look more realistic is essential as opposed to only improving the model to minimize errors on global lighting information.

Table 3 compares bunny renderings using different models. Each model only uses a single input image to predict illumination. Deeplight and LiteAR produce similar results when getting the ambient lighting right. Gardner's [7] method fails to estimate indirect lighting correctly when there is a light source present in the image, as can be seen in the second example. For evaluation, we use $l2$ loss to compare model accuracy for global lighting information. We calculate $l2$ loss by computing the average $l2$ distance between spherical harmonics coefficients produced by our model (SH_g) and ground truth spherical harmonics coefficients (SH_{gt}).

Our neural network model for global spherical harmonics prediction produces better results, i.e., less l2 loss than deeplight [12] and comparable results to that of Gardner [7] as shown in Table 5 and PointAR [23] while being 40 times less computationally expensive than Gardner's [7] method and 6 times less computationally expensive than PointAR [23]. Table 6 compares the model complexities for our method against the state of the art. The model proposed by Gardner

Table 7. Comparison of illumination estimation predicted by the neural network to those estimated by rotating initially predicted spherical harmonics. We use spherical harmonics predicted using one frame and perform SH rotation on them. We then use rotated frames to predict spherical harmonics using the neural network. We compare renderings obtained by both the methods. We also demonstrate the visual difference in illumination and structural similarity index measure (SSIM)

	0	60	120	180
Input image				
Predicted SH				
Rotated SH				
Difference				
SSIM		0.91	0.94	0.96

et al. [7] has more than 20M parameters, resulting in more than 3800M multiply accumulates (MACs). This makes it unsuitable for mobile augmented reality applications. LiteAR has 3 times and 6 times fewer MACs compared to Deeplight and PointAR [23] respectively, therefore making it suitable to run in real time even on mobile phones. Table 5 demonstrates the l2 loss for each model. LiteAR produces better results than Deeplight for global illumination estimation and comparable results to PointAR. However, the l2 loss on global lighting estimation is not a direct metric of measuring realism as the local environment can greatly influence lighting. Spatially varying environmental lighting modules visibly improve the realism of the model. Thus, even with less accurate global lighting prediction compared to other models, LiteAR produces results that are more realistic.

Table 4 demonstrates lighting estimation after taking into account the spatially variable environment. In the first and second examples, point cloud data samples consists of the green-coloured sofa and purple-coloured bed, respectively. These points are located beneath the model, affecting primarily the SH_1 harmonic. Wherein, in the third example, the stairs and walls affect every spherical harmonic in the first band, more so SH_{10} because of the proximity of the stairs. The visual appearance of the model is enhanced greatly and thus helps make the mixed reality experience feel more realistic. Figure 6 demonstrates the improvement in lighting with a closer look at Scene 2 from Table 4. The purple color of the bedsheet affects the lighting of the bunny from below. The updated lighting demonstrates the purple shade on the chest and legs of the bunny, thus making it more realistic.

With a light neural network combined with spherical harmonics rotation based on the input from the IMU sensor, the whole pipeline is mobile-friendly being able to render models at high frame rates. Instead of calling a neural network every frame, in order to make the pipeline even lighter, we use spherical harmonics rotation based on IMU sensor input. Table 7 demonstrates bunny rendering with spherical harmonics predicted by a neural network for every image and compares it to bunny rendering with spherical harmonics predicted once and then rotated by a given angle. The comparison showed a high structural similarity index between lighting estimated using the neural network directly and using SH rotation after estimating once. The rotation operation only requires less than 120 multiply accumulates compared to millions for calling the neural network, therefore reducing the computational load.

5 Conclusion and Future Work

In conclusion, the LiteAR pipeline operates much faster than the state-of-the-art methods while slightly compromising the global illumination estimation accuracy. However, the dataset used to train the model did not have enough variance with respect to illumination. Therefore, the accuracy could be improved with a more varied dataset. Moreover, after updating lighting based on the local spatial environment, the renders look more realistic. Using integrated sensors like the IMU sensor makes the process much faster with minimal visual compromise in estimating illumination.

The dataset to train the model to predict global spherical harmonics consisted of indoor images taken from multiple angles. Most of the photographed rooms share similar lighting for multiple photos. Thus, there is little variation in labels in the form of spherical harmonics. This may lead to over-fitting, as the model would try to find an optimal solution. We solve this problem using data augmentation by introducing colour tints. However, the dataset could be naturally enriched by introducing pictures taken with different mobile phone cameras and of different places under varying lighting.

For considering the local environment to update lighting, experimenting with different values for the radius to sample points and alpha coefficient to update global spherical harmonics gives varying results. Thus, a method could be developed to dynamically select values for the radius and alpha coefficient.

A confidence score along with the spherical harmonics would be helpful to determine the best set of spherical harmonic coefficients predicted by the model. Thus, the most accurate SH prediction could be used along with the input from the IMU sensor instead of calling the neural network model every few frames.

References

1. Althelaya, K.A., Agus, M., Schneider, J.: The mixture graph-a data structure for compressing, rendering, and querying segmentation histograms. IEEE Trans. Vis. Comput. Graph. **27**, 645–655 (2021)
2. Calian, D.A., Lalonde, J.F., Gotardo, P., Simon, T., Matthews, I., Mitchell, K.: From faces to outdoor light probes. In: Computer Graphics Forum, vol. 37, pp. 51–61. Wiley Online Library (2018)
3. Chang, A., et al.: Matterport3D: learning from RGB-D data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
4. Cheng, D., Shi, J., Chen, Y., Deng, X., Zhang, X.: Learning scene illumination by pairwise photos from rear and front mobile cameras. In: Computer Graphics Forum, vol. 37, pp. 213–221. Wiley Online Library (2018)
5. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: ACM SIGGRAPH 2008 Classes, pp. 1–10 (2008)
6. Debevec, P., Graham, P., Busch, J., Bolas, M.: A single-shot light probe. In: ACM SIGGRAPH 2012 Talks, p. 1 (2012)
7. Gardner, M.A., et al.: Learning to predict indoor illumination from a single image. arXiv preprint arXiv:1704.00090 (2017)
8. Greenspan, M., Yurick, M.: Approximate KD tree search for efficient ICP. In: Fourth International Conference on 3-D Digital Imaging and Modeling 2003, 3DIM 2003. Proceedings, pp. 442–448. IEEE (2003)
9. Groß, D., Gumhold, S.: Advanced rendering of line data with ambient occlusion and transparency. IEEE Trans. Vis. Comput. Graph. **27**, 614–624 (2021)
10. Hambarde, P., Murala, S.: S2DNet: depth estimation from single image and sparse samples. IEEE Trans. Comput. Imaging **6**, 806–817 (2020)
11. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. ACM Trans. Graph. (TOG) **30**(6), 1–12 (2011)
12. LeGendre, C., et al.: DeepLight: learning illumination for unconstrained mobile mixed reality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5918–5928 (2019)
13. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162–5170 (2015)
14. Liu, W., Sun, J., Li, W., Hu, T., Wang, P.: Deep learning on point clouds and its application: a survey. Sensors **19**(19), 4188 (2019)
15. Mandl, D., et al.: Learning lightprobes for mixed reality illumination. In: 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 82–89. IEEE (2017)

16. Nowrouzezahrai, D., Simari, P., Fiume, E.: Sparse zonal harmonic factorization for efficient SH rotation. ACM Trans. Graph. (TOG) **31**(3), 1–9 (2012)
17. Prakash, S., Bahremand, A., Nguyen, L.D., LiKamWa, R.: GLEAM: an illumination estimation framework for real-time photorealistic augmented reality on mobile devices. In: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, pp. 142–154 (2019)
18. Robert, C.P., Casella, G.: Monte Carlo integration. In: Robert, C.P., Casella, G. (eds.) Monte Carlo Statistical Methods, pp. 71–138. Springer, New York (1999). https://doi.org/10.1007/978-1-4757-3071-5_3
19. Song, S., Funkhouser, T.: Neural illumination: lighting prediction for indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6918–6926 (2019)
20. de Tinguy, X., Pacchierotti, C., Lécuyer, A., Marchal, M.: Capacitive sensing for improving contact rendering with tangible objects in VR. IEEE Trans. Vis. Comput. Graph. **27**, 2481–2487 (2021)
21. Zanfir, A., Marinoiu, E., Sminchisescu, C.: Monocular 3D pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2148–2157 (2018)
22. Zhang, E., Cohen, M.F., Curless, B.: Emptying, refurnishing, and relighting indoor spaces. ACM Trans. Graph. (TOG) **35**(6), 1–14 (2016)
23. Zhao, Y., Guo, T.: PointAR: efficient lighting estimation for mobile augmented reality. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 678–693. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_40