PhyIR: Physics-based Inverse Rendering for Panoramic Indoor Images Supplemental Material

Zhen Li[1] Lingli Wang[1] Xiang Huang[1] Cihui Pan[1,*] Jiaqi Yang[2,*]
[1]Realsee [2]Northwestern Polytechnical University
yodlee@mail.nwpu.edu.cn,{wanglingli008,huangxiang003,pancihui001}@ke.com,jqyang@nwpu.edu.cn

In this supplementary material, we provide more details of our modules (Sec. A), implementation (Sec. B), proposed datasets (Sec. C), experimental settings (Sec. D) and additional results (Sec. E).

A. Details of Modules

A.1. GMNet

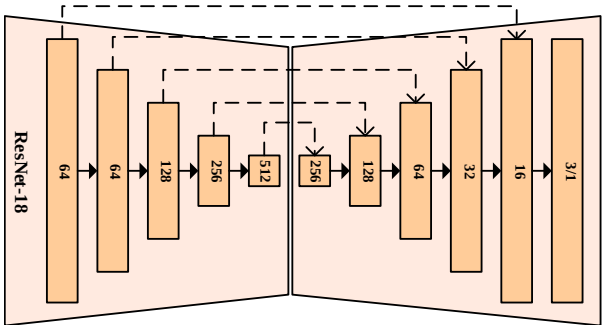


Figure 1. The architecture of GMNet. The encoder is ResNet-18. The number denotes the output channel of each layer or block.

The GMNet consists of an encoder and five decoders. It is similar to UniFuse [7]. The encoder is ResNet-18 and the decoder consists of 11 convolution layers with skip-connection. The detailed architecture is shown in Figure 1. Each convolution layer is followed by an activate layer except for last two layers. The activate layer is ELU. All of five decoders have similar architectures. The number of the last output channel at five branches are different. In particular, one for depth, roughness and metalness; three for normal, base color. The decoder network is described as:

$i512-o256-k3$, $(i512-o256-k3+i256-o128-k3)$, $(i256-o128-k3+i128-o64-k3)$, $(i128-o64-k3+i64-o-32-k3)$, $(i96-o32-k3+i32-o16-k3)$, $(i16-o16-k3+i16-o3/o1-k3)$.

*Co-corresponding authors. The project page is at https://lzleejean.github.io/PhyIR

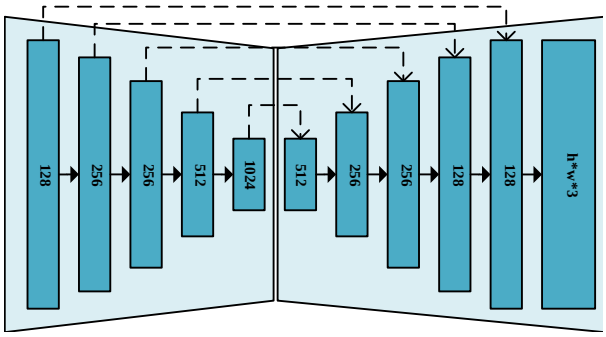


Figure 2. The architecture of LNet. It is similar to [11]. The number denotes the output channel of each layer or block. Number of the last output channel is the total size of a light probe.

The term i denotes the input channel, o is the output channel, k is the kernel size and $()$ represents the convolution block consisting of two convolution layers.

A.2. LNet

The architecture of LNet is similar to InvIndoor [11]. In InvIndoor [11], the LightNet has three branches for spherical Gaussian parameters. For our LNet, we directly predict HDR environment map with an encoder and a decoder. The detailed architecture is shown in Figure 2. Channels of group in group normalization layer are 16. Each convolution layer is followed by a ReLU activate function. The whole network is described as:

$i64-o128-k4-s2-g16$, $i128-o256-k4-s2-g16$, $i256-o256-k4-s2-g16$, $i256-o512-k4-s2-g16$, $i512-o1024-k3-s1-g16$, $i1024-o512-k1-s1-g16$, $i1024-o256-k3-s1-g16$, $i512-o256-k3-s1-g16$, $i512-o128-k3-s1-g16$, $i256-o128-k3-s1-g16$, $(i128-o512-k3-s1+i512-oh\times w\times 3-k3-s1)$.

Here, the term g is the channel of a group, and s denotes the stride of a convolution layer.

Figure 3. Comparison of the rerendering module. From left to right, source image from panorama, re-rendered image, re-rendered diffuse image, re-rendered specular image, our re-rendered specular image using high-resolution and denser light probes. Note that we bright the specular image for a better visualization. We observe that our module produces realistic details, even in glossy surface and metal surface.

A.3. Rendering layer

BRDF model. We use a physics-based BRDF representation in our network named microfacet BRDF. Although InvIndoor [11] also applies microfacet BRDF, it does not model metalness, which is essential in current material assets. In Eq. 9 in the main paper, f_d and f_s are defined as:

$$f_d = \frac{B(1-M)}{\pi}, \tag{1}$$

$$f_s = \frac{DFG}{4(n\cdot v)(n\cdot l)}, \tag{2}$$

where B is base color; M is metalness; l denotes light direction; n denotes normal; v denotes view direction; D denotes Normal Distribution Function (NDF); F denotes Fresnel function and G is the Geometry Factor. We adopt UE4's specular shading model [8].

The specular D:

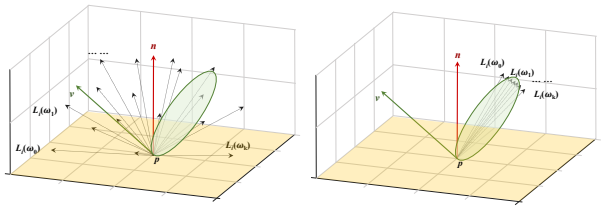
$$\begin{aligned} D &= \frac{\alpha^2}{\pi((n\cdot h)^2(\alpha^2-1)+1)^2}, \\ h &= bisector(v,l), \\ \alpha &= R^2. \end{aligned} \tag{3}$$

The specular F:

$$\begin{aligned} F &= F_0 + (1-F_0)2^{(-5.55473(v\cdot h)-6.98316)(v\cdot h)}, \\ F_0 &= 0.04(1-M)+MB. \end{aligned} \tag{4}$$

The specular G:

$$\begin{aligned} G &= G_1(l)G_1(v), \\ G_1(v) &= \frac{n\cdot v}{(n\cdot v)(1-k)+k}, \\ G_1(l) &= \frac{n\cdot l}{(n\cdot l)(1-k)+k}, \\ k &= \frac{(R+1)^2}{8}. \end{aligned} \tag{5}$$



(a) Uniform sampling (b) Importance sampling

Figure 4. Visualization of importance sampling. The yellow plane represents the surface and the green denotes the f_s lobe of point p. The uniform sampling method is unable to cover main meaningful direction, leading large variance results; however, importance sampling only computes the important direction according to known BRDF of surface leading reflectance with sharper details.

Importantce sampling. As described in Sec. 3.2 in the main paper, we calculate Monte Carlo numerical integration with importance sampling to render detailed specular reflectance. Specifically, we define the p of Eq.10 in the main paper as:

$$p = \begin{cases} \frac{n\cdot l}{\pi} & \text{diffuse} \\ \frac{D(n\cdot h)}{4(v\cdot h)} & \text{specular} \end{cases}, \tag{6}$$

where D is the specular D defined in Eq. 3.

As shown in Figure 4, for a surface point p, if sampled directions are randomly or uniformly selected, most samples cannot be fully employed. Therefore, the re-render result has a large variance.

As shown in Figure 3, our re-render module can effectively render realistic specular reflectance. Thus, our model is able to provide meaningful physical constraints on all components.

Table 1. The inference time of each sub-module. Our entire framework can be trained efficiently.

	GMNet	LNet	Rendering layer	GSNet
Time (ms)	18	4	109	8

B. Details of Implementation

B.1. Training

We first use Adam [9] to train the GMNet for 120 epochs with a learning rate as 1e-4. The batch size is 8. We set β_B, β_R, β_M, β_D to 1.0 and β_N, $\beta_{gradient}$ to 0.1. The resolution of the input panorama, geometry and material is 256×512.

Second, we frozen the GMNet and use Adam [9] to train the LNet for 90 epochs with a learning rate as 1e-4. The batch size is 4. We set β_L, β_{SC}, β_{render} to 1.0. The resolution of the re-rendered image is 128×256. The resolution of each light probe is 16×32.

Third, we jointly finetune the GMNet and LNet for 10 epochs with a learning rate as 1e-5. The batch size is 4. We set β_L, β_{render} to 1.0 and β_{SC} to 0.1.

Last but not least, we frozen the GMNet and use Adam [9] to train the GSNet for 80 epochs with a learning rate as 5e-4. The batch size is 16. We set β_B, β_R to 1.0, β_N to 0.1 and $radius$ of the guided filter to 2.

B.2. Inference

The inference time of each sub-module is averaged over 2000 images with a batch size of 1, which is clocked on a Tesla V100 GPU. The results are summarized in Table 1. Thus, our framework can be trained end-to-end efficiently.

With a batch size of 1, our framework consumes less 6G GPU memory without quantization.

C. Details of Proposed Datasets

C.1. FutureHouse

As described in Sec. 3.1 in the main paper, our artist-designed dataset named *FutureHouse* is very close to real-world data thanks to expensive assets and powerful rendering technologies. As shown in Table 1 in the main paper, our dataset provides comprehensive annotations that aid research on multiple topics. We introduce the production of dataset in the following.

We first design massive and diverse high-resolution models by a large number of professional designers. The category of models includes common furnitures and essential decorative ornaments, as shown in Figure 5. The changeable style of models is capable of simulating a variety of house types. Then, to reduce the gap with the real-world, excellent layouts are designed by over 100 professional artists.

Figure 5. Examples of our high-quality objects. More than 70,000 models with high-resolution meshes and material significantly improve the realism of rendered images and the diversity of our dataset.

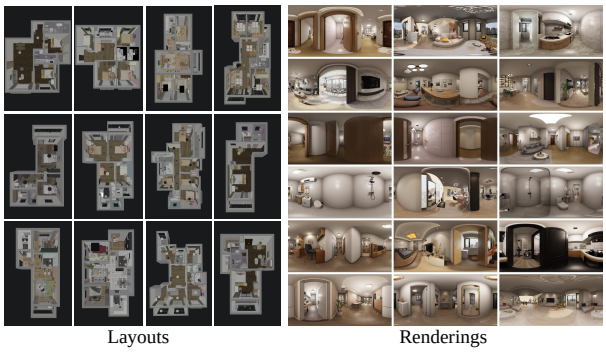


Figure 6. Our *FutureHouse* dataset.

As shown in Figure 6, our indoor scenes are very close to the real-house in layout, which greatly reduces the divergence between our data and the real-world data. Lastly, we use a GPU cluster consisting of 32 Quadro RTX 8000 GPUs and a real-time ray tracing rendering engine, UE4 [2], to efficiently render high-quality images. Rendering this dataset spends almost one month.

We provide more detailed examples for all renderings, including final image, depth, normal, base color, roughness, metalness, mask of light source and transmission, and per-pixel illumination in Figure 7.

Rendering color, geometry and material images with 480 × 640 resolution costs total 600 seconds per image and rendering per-pixel SV environment maps costs 100 seconds per image in OpenRooms [12]. In our *FutureHouse*, rendering color, geometry and material images with 512 × 1024 resolution costs total less than 1 second and rendering per-pixel SV environment map costs almost 9 hours. Our lighting annotation is a denser high-resolution per-pixel HDR illumination map with (3, 128 × 128, 256 × 256) resolution while the shape of OpenRooms [12] is (3, 120 × 16, 160 × 32). The comparison of quality between selected examples from OpenRooms [12] and our *FutureHouse* is shown in Figure 8. The noise decreases greatly in our renderings.

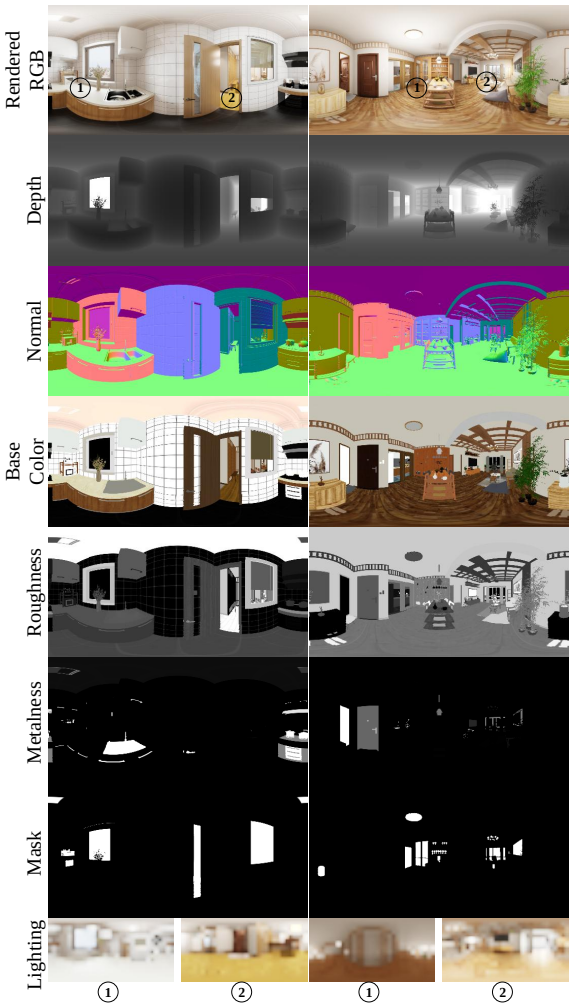
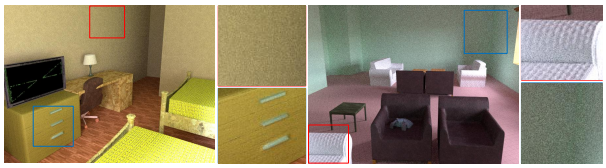


Figure 7. Detailed examples of annotations. Our GT annotations include depth, normal, base color, roughness, metalness, mask of emissive material and transparent material, and per-pixel lighting. For a better visualization, we only show two selected light probes.

Note that our light probe images also use the same rendering parameters as color images. As shown in Figure 9, our light probes are sharper with more details of full-spherical environment, which is important for SC loss proposed in Sec. 3.2 in the main paper.

C.2. The SC illumination dataset

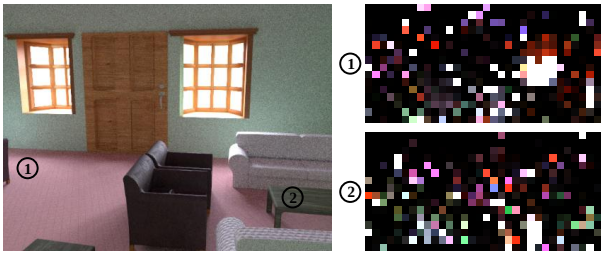
As described in Sec. 4 in the main paper, we capture a panoramic dataset including 7 indoor scenes and 72 local high-resolution HDR light probes. Compared to [5], ***the SC light probe*** is the most critical difference. We encourage readers to view SC lighting video in supplementary videos. Another important difference is that all of our images are ***high-resolution and panoramic*** while the input image of [5] is perspective and the light probe is low-resolution without details of whole environment. More ex-



(a) Examples from OpenRooms [11]

(b) Examples from *FutureHouse*

Figure 8. Qualitative comparison of rendering quality between OpenRooms [12] and *FutureHouse*. Our dataset is more photo-realistic with less noise.



(a) Examples from OpenRooms [11]

(b) Examples from *FutureHouse*

Figure 9. Qualitative comparison of light probes between OpenRooms [11] and ours. For a fair comparison, the resolution of our shown light probes is equal to OpenRooms, (16×32). Our light probes are sharper with more details of whole environment, which is important for proposed SC loss.

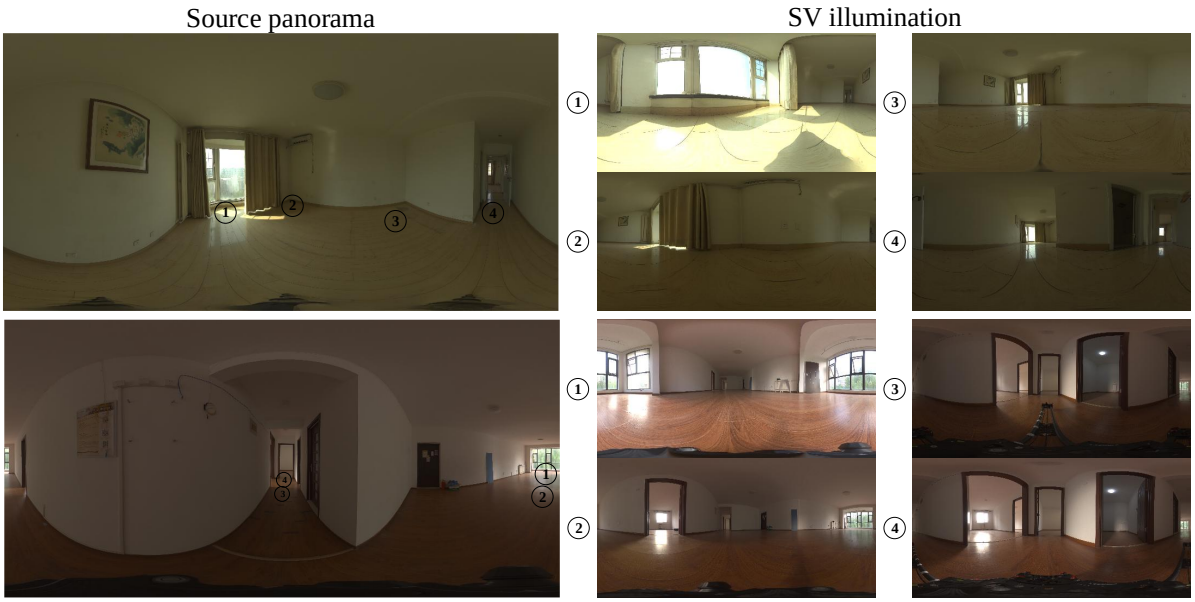


Figure 10. Examples of our captured SV illumination. All of our panoramas, including source input and light probes, are fully HDR and high-resolution (8K).

Figure 11. The virtual object insertion of our captured SC illumination dataset. The virtual object shows realistic complex lighting effects. Please zoom in for details.

amples of captured SV lighting are shown in Figure 10. We also insert some virtual objects into these scenes based on captured high-quality illumination in Figure 11. The virtual object shows realistic complex lighting effects, *e.g.*, soft shadows and highlight.

D. Details of Experiments

D.1. Our microfacet BRDF renderer based Mitsuba

To calculate the relighting error of virtual spheres with different material, we realize the BRDF model introduced in Sec. A.3 using a physics-based renderer named Mitsuba [6], which is licensed under GNU 3.0. It can handle complex material, *e.g.* metal material and mirror material, in a uniform microfacet model. In our experiments, we render spheres with predicted illumination or GT illumination using image-based lighting.

D.2. Virtual object insertion

To render the virtual object into real image, two methods are used to fuse them. One is similar with InvIndoor [11], rendering two images, i.e., I_{all} and I_{pl}. I_{all} is the rendered image containing both the virtual object and the virtual plane. I_{pl} is the rendered image containing only the virtual plane. Rendering object and plane together can ensure inter-reflectances between them are properly simulated. Detailed formulation can be found in InvIndoor [11]. However, this fusion method generates strong artifacts in object with specular material, because the virtual plane is inconsistent with the real plane in the texture detail. Therefore, we propose the other one, rendering three images, i.e., I_{all}, I_{pl} and I_{obj}. I_{obj} is the rendered image containing only the virtual object. For the object region of final image, we only use the value in I_{obj}:

$$I_{new} \odot M_{obj} = I_{obj} \odot M_{obj}, \tag{7}$$

where M_{obj} is binary mask covering only the virtual object. This fusion method does not consider the inter-reflectance result on virtual object. Note that it also generates the inter-reflectance result on virtual plane, *e.g.*, shadows and specular reflectance caused by the object. It can ensure the bottom region of specular virtual object has detailed texture that is

Table 2. The microfacet parameters of three spheres for rendering. Three spheres have different material, including absolute diffuse, matte sliver and mirror sliver.

	Diffuse	Matte Sliver	Mirror Sliver
Base color	(0.5, 0.5, 0.5)	(0.972, 0.960, 0.915)	(0.972, 0.960, 0.915)
Roughness	1.0	0.5	0.0
Metalness	0.0	1.0	1.0

consistent with real images.

The selection of these methods depends on the material of the object and the quality of albedo and lighting. Specifically, the albedo prediction of InvIndoor [11] has more details but their predicted illumination lacks high-frequency details. The former fusion method is more suitable. In contrast, for the projection-based method [3, 4, 10], it can generate high-quality illumination with high-frequency details from the input panorama. However, these method lack albedo estimation or predict albedo with less details. Therefore, the latter fusion method is more suitable.

D.3. Light comparison

As described in Sec.4.2 in the main paper, we use the widely used metric, the relighting error, to evaluate the performance of different approaches. To achieve a more comprehensive comparison, we relight three virtual spheres with different material, pure diffuse, matte sliver and mirror sliver. The diffuse sphere and matte sliver will evaluate the total radiance and HDR, and the mirror sliver will evaluate the high-frequency detail of predicted illumination.

For the quantitative result of Table 5 in the main paper, microfacet parameters of three spheres rendered by our renderer (Sec. D.1), are shown in Table 2. The *base color* parameter of the glossy sphere for qualitative results of Figure 7 in the main paper is (0.8, 0.8, 0.8), which equals to the setting in InvIndoor [11].

D.4. Depth comparison

In Table 7 in the main paper and Table 3, all approaches are evaluated on standard metrics, including mean absolute error (MAE), absolute relative error (Abs Rel), square relative error (Sq Rel), root mean square error (RMSE), root mean square error in log space (RMSE log), and relative accuracy metrics δ^n, which represents the ratio of pixels with a relative error lower than 1.25^n.

E. Additional Results

E.1. Ablation study

We verify the validity of CirP [14] and joint training on depth estimation in Table 3, the CirP can extracts robust 3D features from panoramas and the joint training including our GMNet, LNet and physics-based renderer provides more physical constraints to assist depth estimation.

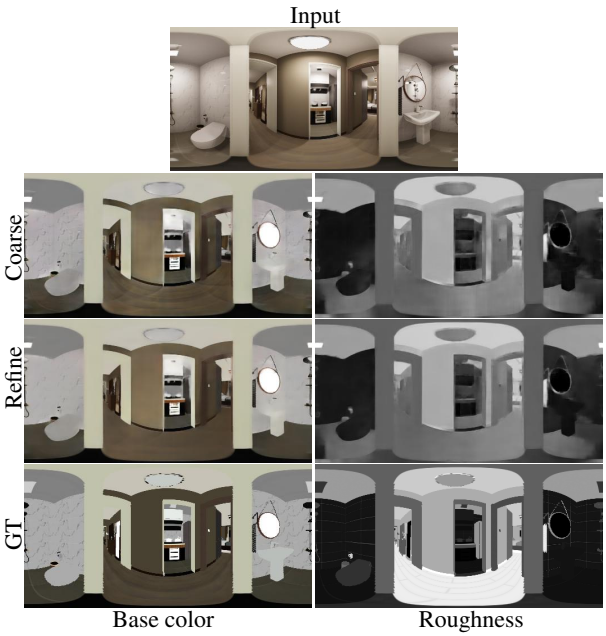


Figure 12. Ablation study of the GSNet on *FutureHouse*.

Additionally, we provide several qualitative results for ablation study of the GSNet. As shown in Figure 12, the GSNet can significantly generate smoother results.

E.2. Qualitative results of geometry and material

Comparison in virtual data. As described in the Sec. 4.1 in the main paper, we provide more qualitative results on *Futurehouse* and synthetic data provided by LRG360 [10]. More examples on *Futurehouse* in Figure 13 and more examples on synthetic data provided by LRG360 [10] in Figure 14. Moreover, we provide more examples of re-rendered images in Figure 16. The proposed method can reproduce realistic specular reflectance on glossy material and even in mirror material.

Comparison in real data. We show the qualitative result on real data provided by LRG360 [10] in Figure 5 in the main paper. In addition, we provide more examples on real images in Figure 15 and Figure 17.

E.3. Qualitative results of illumination

We show more results of illumination on our unseen synthetic data in Figure 18 and Figure 19. We observe that our method can recover the illumination that is similar to GT in structure. Moreover, we provide qualitative results of dynamic virtual object insertion using our predicted illumination in Figure 20. Our method generates coherent virtual object insertion results without any temporal constraints. More animations in supplementary videos.

As described in Sec. 4.2 in the main paper, we provide more virtual object insertion results for the lighting com-

Table 3. Ablation study of CirP and joint training on depth estimation. The performance evaluated on standard metrics are shown in below.

	MAE	Abs Rel	Sq Rel	RMSE	RMSElog	Log_{10}	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Baseline	0.0905	0.0675	0.0266	0.1915	0.0510	0.0300	0.9468	0.9818	0.9910
+CirP	0.0865	0.0642	0.0255	0.1876	0.0490	0.0286	0.9506	0.9831	0.9916
+CirP+Joint	**0.0846**	**0.0638**	**0.0255**	**0.1859**	**0.0485**	**0.0279**	**0.9516**	**0.9833**	**0.9917**

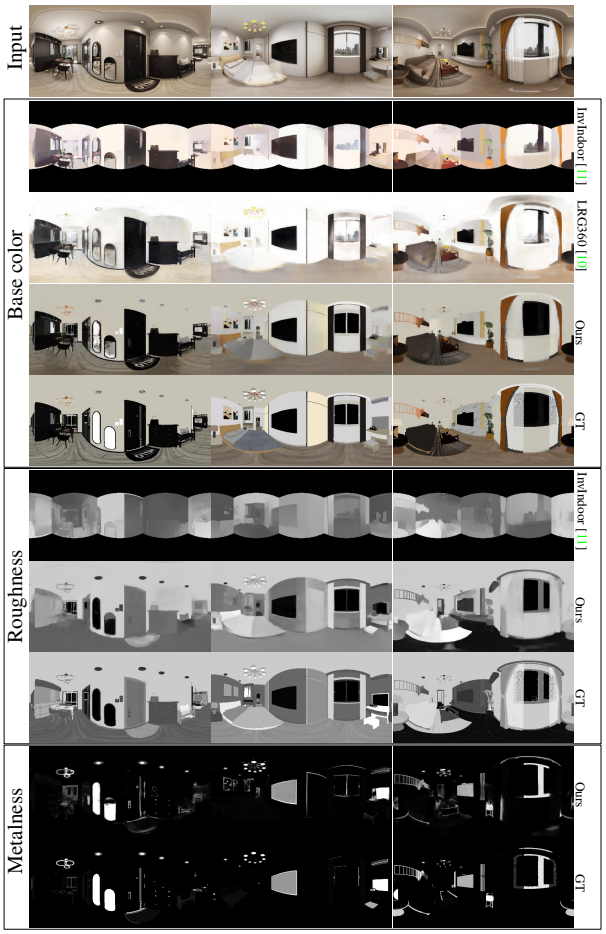


Figure 13. Qualitative comparison of material estimation on *FutureHouse*.

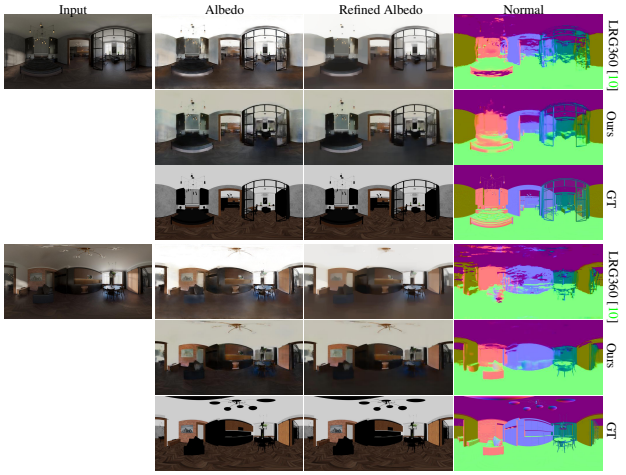


Figure 14. Qualitative comparison on synthetic data provided by LRG360 [10].

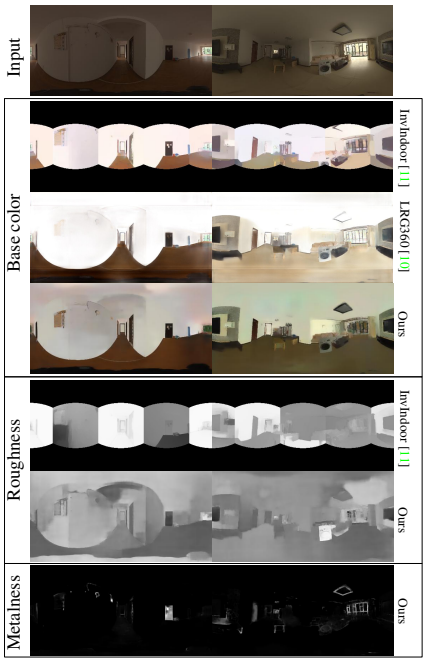


Figure 15. Qualitative comparison of material on real-world data.

parison in Figure 21. In addition, we also show predicted or GT illumination at each spatial position. Our method can recover more detailed illumination with correct spatial structure compared to InvIndoor [11]. More dynamic animations in supplementary videos. We use a video frame interpolation method named DAIN [1] to generate high frame-rate videos on our SC illumination dataset.

E.4. Limitation and future work

The proposed SC loss, as shown in Eq.2 in the main paper, is based on the assumption that discontinuities of nearby light probes mainly occur where the gradient of the global depth map is large. However, this assumption is a simplification for visibility calculation, which will be limited in the shadow boundary.

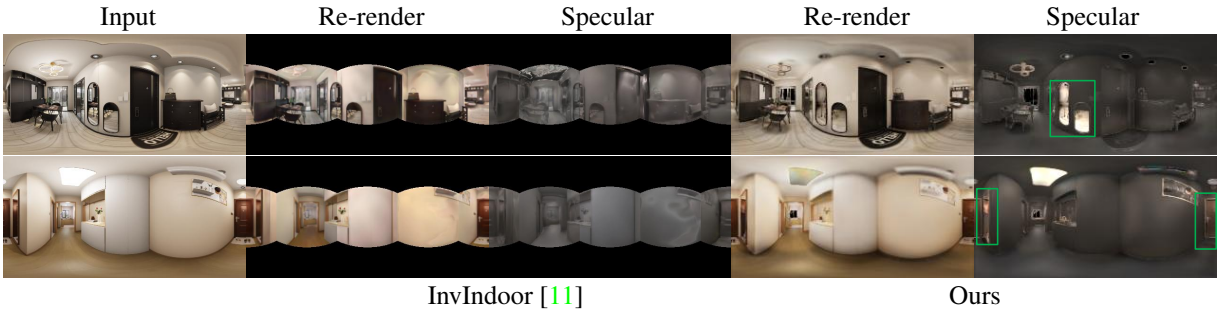


Figure 16. Qualitative comparison of re-rendered images.

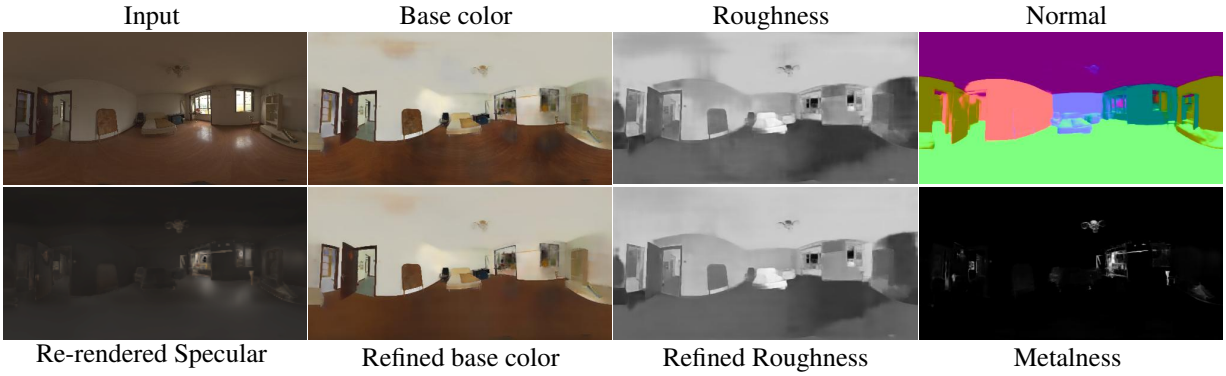


Figure 17. Qualitative results on real-world data.

Figure 18. Qualitative results of virtual object insertion and illumination on unseen synthetic data.

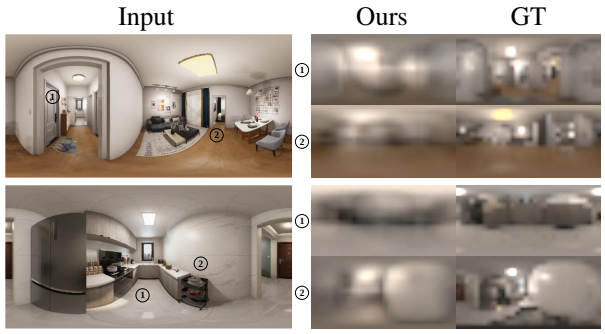


Figure 19. Qualitative results of illumination on unseen synthetic data.

Although our method can recover more detailed illumination maps than previous per-pixel lighting approach [11], the prediction is still not detailed enough which only has a coarse 3D structure of the scene. Recently, the projection-based lighting [10] and the volumetric lighting [13,15] show great potential in detailed illumination. Incorporating these representations into our physics-based in-network rendering module is challenging yet meaningful.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7

[2] Epic Games. Unreal engine. 3

[3] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7175–7183, 2019. 6

[4] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 9(4), 2017. 6

[5] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[6] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 5

Figure 20. Qualitative results of dynamic virtual object insertion. Our approach generate coherent results without any temporal constraints. More animations in our supplementary videos.

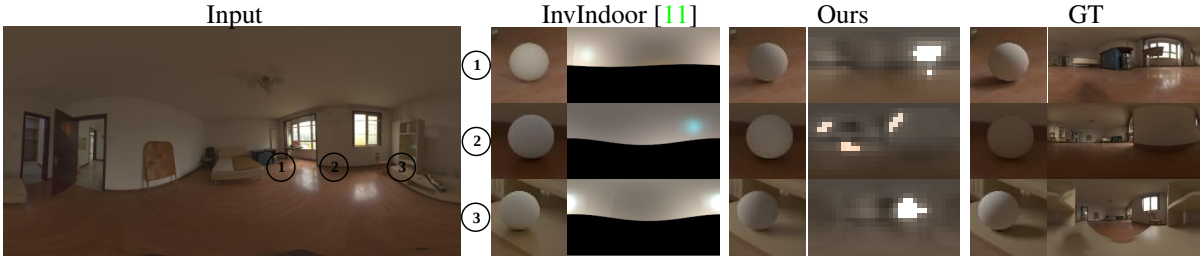


Figure 21. Qualitative comparison of illumination on real-world data. We provide results of virtual object insertion and illumination for each method.

[7] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 2021. 1

[8] Brian Karis and Epic Games. Real shading in unreal engine 4, 2013. 2

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 3

[10] Junxuan Li, Hongdong Li, and Yasuyuki Matsushita. Lighting, reflectance and geometry estimation from 360° panoramic stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6, 7, 8

[11] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 1, 2, 4, 5, 6, 7, 8, 9

[12] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Hong-Xing Yu, Zexiang Xu, Kalyan Sunkavalli, Milos Hasan, Ravi Ramamoorthi, and Manmohan Chandraker. Openrooms: An open framework for photorealistic indoor scene datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7190–7199, June 2021. 3, 4

[13] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2020. 8

[14] Tsun-Hsuan Wang, Hung-Jui Huang, Juan-Ting Lin, Chan-Wei Hu, Kuo-Hao Zeng, and Min Sun. Omnidirectional cnn for visual place recognition and navigation. *arXiv preprint arXiv:1803.04228*, 2018. 6

[15] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 8