Differentiable Point-based Inverse Rendering

Hoon-Gyu Chung Seokjun Choi Seung-Hwan Baek
POSTECH

Abstract

We present differentiable point-based inverse rendering, DPIR, an analysis-by-synthesis method that processes images captured under diverse illuminations to estimate shape and spatially-varying BRDF. To realize this idea, we devise a hybrid point-volumetric representation for geometry and a regularized basis-BRDF representation for reflectance. The hybrid geometric representation enables fast rendering through point-based splatting while retaining the geometric details and stability inherent to SDF-based representations. The regularized basis-BRDF mitigates the ill-posedness of inverse rendering stemming from limited light-view angular samples. We also propose an efficient shadow detection method using point-based shadow map rendering. Our extensive evaluations demonstrate that DPIR outperforms prior works in terms of reconstruction accuracy, computational efficiency, and memory footprint. Furthermore, our explicit point-based representation and rendering enables intuitive geometry and reflectance editing.

1. Introduction

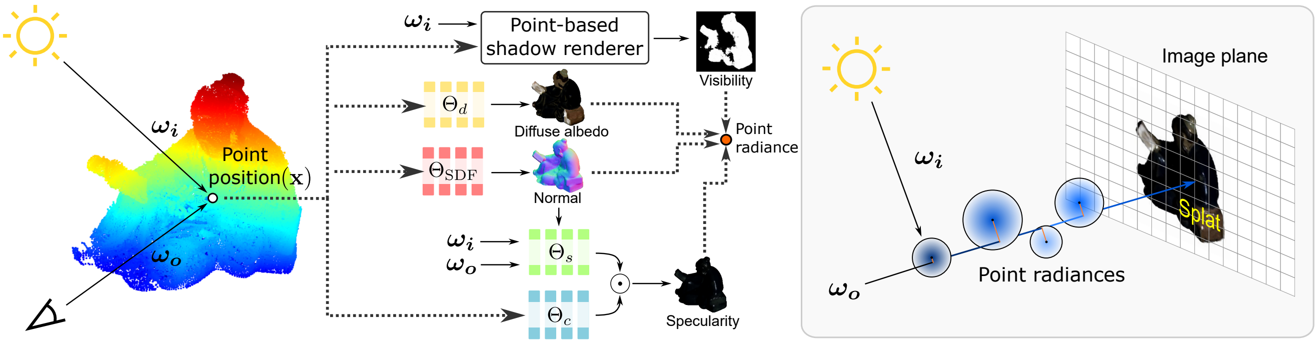
Inverse rendering aims to estimate geometry and reflectance of real-world objects from a set of images, with applications including relighting, augmented and virtual reality, and object digitization. We present differentiable point-based inverse rendering, DPIR, that exploits point-based forward rendering for inverse rendering. DPIR processes either multi-view multi-light images or photometric images captured by multi-view photometric setups [13] and flash photography, respectively. Using point-based forward rendering confronts several challenges critical for inverse rendering: (1) The discrete point representation hinders the reconstruction of both smooth and detailed surface. (2) The inherent difficulty in jointly reconstructing geometry and spatially-varying BRDF from limited light-view samples remains. (3) There is a need for efficient shadow consideration to ensure precise inverse rendering.

To address these challenges, we develop a hybrid point-volumetric geometry representation and a regularized basis-BRDF representation. The hybrid geometric representation enjoys the benefits of both point-based and volumetric geometry representations, ensuring efficient rendering through point-based splatting. The regularized basis-BRDF representation consists of a per-position diffuse albedo and a weighted specular-basis BRDFs, overcoming the ill-posedness of inverse rendering under limited light-view angular samples. Also, we tackle the shadow detection problem through an efficient point-based shadow detection method, bypassing the volumetric integration often used in learning-based inverse rendering methods [25, 30, 36, 37]. DPIR then jointly optimizes point locations, point radii, diffuse albedo, specular basis BRDFs, specular coefficients, and SDF, by leveraging point-based splatting as a forward renderer in the analysis-by-synthesis framework. Our extensive evaluations show that DPIR outperforms previous state-of-the-art inverse rendering methods [8, 30, 33, 34], in accuracy, training speed, and memory footprint. Furthermore, the explicit point representation and rendering of DPIR enable convenient scene editing.

2. Related Work

Inverse Rendering Inverse rendering is a long-standing problem in computer vision and graphics [1, 3, 4, 6, 11, 19, 23, 29, 38]. Learning-based single-image inverse rendering enables accurate reconstruction for planar samples [2, 14–16, 22, 24, 27, 32]. For multi-view inputs captured under a constant lighting, learning-based inverse rendering methods [17, 25, 33, 34, 36, 37] have recently shown promising results by exploiting volumetric rendering as in NeRF [18]. However, these methods struggle with low reconstruction accuracy of spatially-varying reflectance and excessive training time due to the multiple sampling per each ray in the volumetric rendering process. Using multi-view and multi-light inputs improves reconstruction quality as demonstrated in PS-NeRF [30], suffering from excessive training time and pre-trained photometric stereo network.

Point-based Rendering Point-based rendering uses points as a compact scene representation for rendering [5]. Splatting with circular disks, ellipsoids, or surfels [20, 39–41] enables high-quality and efficient point-based rendering. Zhang et al.[35] employ efficient



(a) Point radiance computation

(b) Point splatting-based rendering

Figure 1. **Overview of differentiable forward rendering.** (a) For each 3D point, its position is used as a query for the diffuse-albedo MLP Θ_d, SDF MLP Θ_{SDF}, and specular-basis coefficient MLP Θ_c. The specular-basis BRDF MLP Θ_s models specular-basis reflectance, given the incident and outgoing directions $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_o}$. The point-based shadow renderer estimates the point visibility from a light source per each image. By using the diffuse albedo, normals, specular reflectance, and visibility, we compute the radiance for each point. (b) The radiance is then projected onto a camera plane to render the pixel color through splatting-based differentiable forward rendering.

point-based rendering and spherical-harmonics point radiance for efficient novel view synthesis. Kerbl et al.[9] use anisotropic Gaussians and tile-based optimization method for accurate novel-view synthesis of in-the-wild scenes.

In summary, Our DPIR provides highest reconstruction quality, fastest training time, and lowest memory footprint. To this end, we use points as rendering primitives, resulting in fast training with splatting-based rendering. To handle the limited light-view angular samples, we gather multiple samples from points with similar specular appearance by using regularized basis BRDFs. Also, we handle shadow using efficient point-based shadow rendering.

3. Method

The proposed DPIR estimates points, surface normals, spatially-varying BRDF, and visibility from the images of a static object captured either by flash photography [33] or multi-view multi-light imaging [30]. Figure 1 shows a detailed overview of our differentiable forward rendering process, which is pivotal for achieving efficient and high-quality inverse rendering.

3.1. Scene Representation

Hybrid Point-volumetric Geometric Representation We use a set of 3D points, where each point possesses two parameters: position $\mathbf{x} \in \mathbb{R}^{3\times1}$ and radius $r \in \mathbb{R}^{1\times1}$. Using the points as geometric primitives allows for fast splatting-based rendering by bypassing the per-ray integration used in volumetric rendering. However, using the points only often makes its surface normals noisy as the discrete points are non-uniformly distributed through space. To represent both detailed and smooth geometry with accurate surface normals $\mathbf{n}$ for each point at position $\mathbf{x}$, we use SDF, represented as a coordinate-based MLP, Θ_{SDF}, as follows:

$$\mathbf{n} = \nabla_{\mathbf{x}}\Theta_{\text{SDF}}(\mathbf{x})\,, \tag{1}$$

where $\nabla_{\mathbf{x}}$ is the differentiation operator [26, 31].

Our hybrid point-volumetric representation allows *discrete points* to move in space while the surface normals of the points can be sampled from the *continuous SDF*.

Regularized Basis BRDF Representation Reconstructing per-point BRDF from limited light-view angular samples is an ill-posed problem. Thus, we propose to exploit spatial coherency of specular reflectance by using the basis BRDF representation [10, 12, 19]. Specifically, we use three MLPs for diffuse albedo, specular coefficients, and specular basis BRDFs, describing the total reflectance f_r as

$$f_r(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}, \mathbf{x}, \mathbf{n}) = \Theta_d(\mathbf{x}) + \Theta_s(\boldsymbol{h}, \boldsymbol{n})\,\Theta_c(\mathbf{x}), \tag{2}$$

where Θ_d, Θ_s, and Θ_c are the MLPs for the diffuse albedo, specular basis, and regularized specular-basis coefficients, respectively. The outputs of the MLPs are in the following dimensions: $\Theta_d(\mathbf{x}) \in \mathbb{R}^{3\times1}$, $\Theta_s(\boldsymbol{h}, \boldsymbol{n}) \in \mathbb{R}^{3\times K}$, $\Theta_c(\mathbf{x}) \in \mathbb{R}^{K\times1}$. K is the number of basis BRDFs. $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_o}$ are the incident and outgoing light directions. $\mathbf{h}$ is the half-way vector: $\boldsymbol{h} = (\boldsymbol{\omega_i} + \boldsymbol{\omega_o})/\|\boldsymbol{\omega_i} + \boldsymbol{\omega_o}\|$.

Unlike previous inverse rendering methods utilizing basis BRDFs [12, 19, 30], we found that enforcing a positive constraint in the *specular basis coefficients* and optimizing under an $\boldsymbol{l_1}$-norm with a lower bound of ϵ significantly enhances the accuracy of reflectance estimation. We set $\epsilon = 1$ for highly glossy objects, $\epsilon = 0.5$ otherwise.

3.2. Point-based Visibility Test

Departing from computationally extensive volumetric-rendering approaches for visibility test in recent learning-based methods [12, 30, 36], we use a simple method compatible with our point representation: a shadow map technique used in rasterization-based graphics [7, 21, 28]. Figure 2 shows the schematics of our visibility test. For each

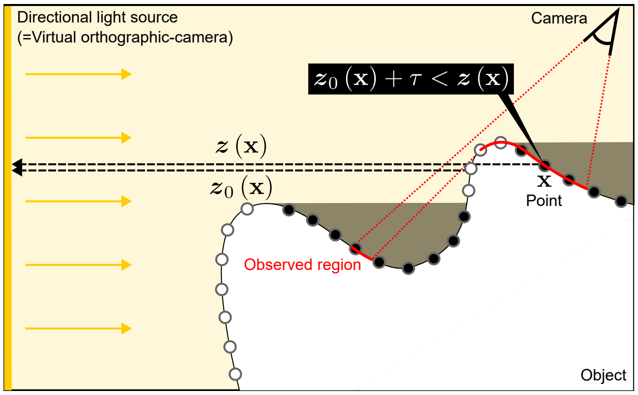


Figure 2. **Point-based visibility test.** To determine the visibility of each point, we compute the depth using the z-buffer from a virtual camera positioned at the light source, and compare the depth with the distance from each point to the virtual camera.

input image and a known light source, we place a virtual orthographic camera at the location of the directional light source. We then perform splatting-based rendering of the points to the virtual camera. The resulting z-buffer stores the depth z of each point with respect to the virtual camera. The visibility function f_v then can be estimated from the z-buffer as

$$f_v(\boldsymbol{\omega_i}, \mathbf{x}) = \sigma(\tau + z_0(\mathbf{x}) - z(\mathbf{x})), \tag{3}$$

where $\sigma(\cdot)$ is the step function that returns 1, meaning visible from the light source if the input is positive, and 0, which means invisible, otherwise. $z_0(\mathbf{x})$ denotes the depth value of the first intersection point and τ is the threshold, set to be 0.1 in our experiments.

3.3. Optimization via Point Splatting

Point-based Rendering For rendering each image, we use the point location $\mathbf{x}$, normals $\mathbf{n}$ sampled from the SDF, reflectance from the learned BRDFs $f_r(\cdot)$, and visibility from the point-based test $f_v(\cdot)$. We compute the radiance R for each point $\mathbf{x}$ along the direction $\boldsymbol{\omega_o}$ towards a camera pixel as follows:

$$R(\boldsymbol{\omega_o}, \mathbf{x}) = L(\boldsymbol{\omega_i}, \mathbf{x}) f_v(\boldsymbol{\omega_i}, \mathbf{x}) f_r(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}, \mathbf{x}, \mathbf{n})(\boldsymbol{\omega_i} \cdot \mathbf{n}), \tag{4}$$

where $L(\boldsymbol{\omega_i}, \mathbf{x})$ is the incident radiance from the direction $\boldsymbol{\omega_i}$ to the point $\mathbf{x}$.

Once the radiance of each point is computed, we project the points to the camera viewpoints from which images are captured, using point splatting. The rendered pixel intensity of a camera pixel u amounts to the results of α-blending of the projected point radiance R:

$$I(u) = \sum_{i=1}^{N} R_i \alpha_i(u) \prod_{j=1}^{i-1} (1 - \alpha_j(u)), \tag{5}$$

$$\alpha_i(u) = 1 - \frac{(p_i - u)^2}{r_i^2}, \tag{6}$$

d where N is the number of points, R_i is the radiance of i-th point computed from Equation (4), p_i is the projected pixel location, and r_i is the radius of the point. α_i is the corresponding weight.

Optimization We propose to optimize point positions $\mathbf{x}$, point radii r, and MLPs for SDF Θ_{SDF}, diffuse albedo Θ_d, specular coefficients Θ_c, and specular-basis BRDFs Θ_s by minimizing the following loss:

$$\mathcal{L}_2 + \lambda_{\text{ssim}} \mathcal{L}_{\text{ssim}} + \lambda_{\text{SDF}} \mathcal{L}_{\text{SDF}} + \lambda_{\text{c}} \mathcal{L}_{\text{c}} + \lambda_{\text{m}} \mathcal{L}_{\text{m}}, \tag{7}$$

where $\mathcal{L}_2$ and $\mathcal{L}_{\text{ssim}}$ are the $\boldsymbol{l_2}$ loss and the SSIM loss for the rendered image I and the captured image I'. $\mathcal{L}_{\text{SDF}}$ promotes the zero-level set of the SDF lies near the point positions: $\mathcal{L}_{\text{SDF}} = \|\Theta_{\text{SDF}}(\mathbf{x})\|^2$. $\mathcal{L}_{\text{c}}$ regularizes $\boldsymbol{l_1}$-norm of per point specular coefficient to be ϵ: $\mathcal{L}_{\text{c}} = \|\Theta_c(\mathbf{x})\|_1 - \varepsilon$. $\mathcal{L}_{\text{m}}$ is defined as the $\boldsymbol{l_2}$ loss between the estimated mask and the ground-truth mask, aligning the rendered mask from the points to the input masks. For stable optimization, we employ the mask-based initialization of the point locations, pruning, and upsampling schemes following Zhang et al. [35].

4. Results

Multi-view Multi-light Images We evaluate DPIR on DiLiGenT-MV [13], a multi-view multi-light dataset, compared to state-of-the-art neural inverse rendering methods: PhySG [34], TensoIR [8] and PS-NeRF [30]. Table 1 and Table 2 show quantitative evaluations of novel-view relighting, normal accuracy, training time, and memory footprint. DPIR not only outperforms the compared methods in rendering and normal accuracy, but also offers 10× faster training and 8× lower memory footprints than PS-NeRF, which is the only competitive method in rendering accuracy. Figure 3 shows novel-view rendering images and estimated normals. For the above experiments, we have two comparison configurations. First, since PhySG and TensoIR assume constant environment illumination, we use the multi-light averaged image for each view to simulate a virtual environment map [30]. Second, we make a comparison only between PS-NeRF and DPIR to directly compare the rendered images under a novel view and lighting without any averaging. This comparison protocol is adopted from the PS-NeRF [30].

		Bear			Buddha			Cow			Pot2			Reading		
Config.	Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS↓
Single light	PhySG	24.52	0.9590	0.041	20.92	0.9229	0.0624	23.64	0.9604	0.0342	25.21	0.9609	0.0241	19.05	0.9056	0.0817
	TensoIR	24.81	0.9597	0.0510	25.40	0.9521	0.0374	27.09	0.9766	0.0260	26.89	0.9741	0.0309	27.02	0.9607	0.0277
	PS-NeRF	35.19	0.9925	0.0061	32.85	0.9783	0.0074	36.57	0.9942	0.0026	38.40	0.9943	**0.0019**	33.73	0.9792	0.0077
	Ours	**43.21**	**0.9944**	**0.0037**	**37.62**	**0.9858**	**0.0048**	**38.23**	**0.9946**	**0.0022**	**39.32**	**0.9951**	0.0022	**35.75**	**0.9843**	**0.0063**
Multi light	PS-NeRF	34.27	0.9802	0.0127	31.58	0.9637	0.0114	36.03	0.9871	0.0066	37.76	0.9851	0.0041	31.16	0.9736	**0.0202**
	Ours	**39.78**	**0.9821**	**0.0083**	**34.88**	**0.9726**	**0.0090**	**37.64**	**0.9890**	**0.0041**	**38.86**	**0.9885**	**0.0034**	**32.50**	**0.9788**	0.0204

Table 1. Quantitative comparison of novel-view rendering and relighting on DiLiGenT-MV dataset.

	Bear	Buddha	Cow	Pot2	Reading	Train	Mem.
Method	MAE ↓	MAE ↓	MAE ↓	MAE ↓	MAE ↓	time↓	↓
PhySG	11.35	27.20	16.10	11.98	26.67	20h	13MB
TensoIR	35.93	37.54	30.10	25.68	33.67	6h	68MB
PS-NeRF	4.68	11.92	5.89	7.55	10.64	22h	40MB
Ours	**4.35**	**11.10**	**4.61**	**6.71**	**9.03**	**2h**	**5MB**

Table 2. Quantitative comparison of normal accuracy, average training time, and memory footprint on DiLiGenT-MV dataset.

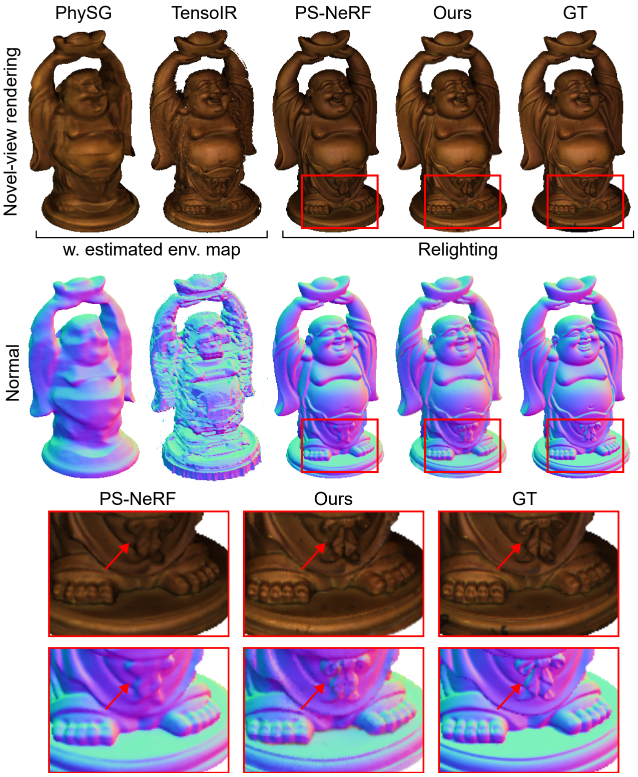


Figure 3. **Comparison of novel view rendering and estimated normal on DiLiGenT-MV dataset.** Our DPIR recovers detailed surface normals and reproduces accurate appearance.

Photometric Images We then evaluate DPIR on a photometric dataset, which we render using four objects at 300 views with co-located point lights, following the configuration of mobile flash photography [19]. Table 3 and Figure 4 show that DPIR outperforms IRON [33], the-state-of-the-art inverse rendering method for photometric images, with higher novel-view relighting quality, 6× faster training, and 5× lower memory footprint.

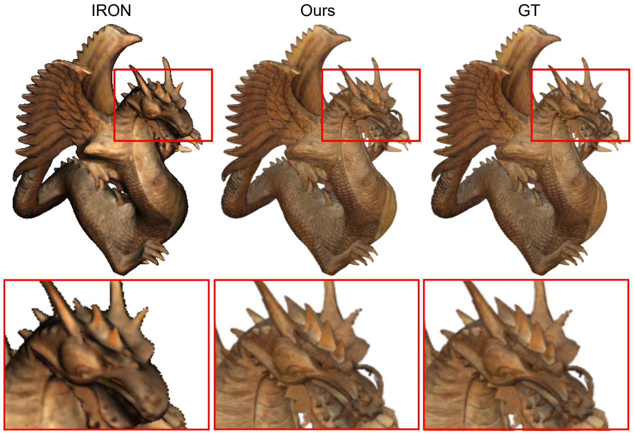


Figure 4. **Comparison of novel-view flashlight relighting.** DPIR successfully reproduces the ground-truth appearance, outperforming IRON.

	PSNR↑	SSIM↑	LPIPS↓	MAE↓	Train↓	Mem↓
IRON	31.91	0.9557	0.0446	9.38	12h	28MB
Ours	**35.56**	**0.9734**	**0.0285**	**8.74**	**2h**	**5MB**

Table 3. Quantitative comparison of novel-view relighting, normal accuracy, training time, and memory footprint on the photometric dataset.

5. Conclusion

In this paper, we introduced DPIR, a point-based inverse rendering method that integrates efficient differentiable point-splatting forward rendering into the analysis-by-synthesis inverse rendering framework. We have developed a hybrid point-volumetric geometry representation, introduced regularized basis BRDFs, and used a point-based visibility detection method. DPIR jointly optimizes the point locations, radii, surface normals, and reflectance in a single stage without using any pre-trained network. Through evaluations, we demonstrate DPIR outperforms state-of-the-art inverse rendering methods in accuracy, training speed, and memory footprint.

Limitations First, DPIR only models direct reflection without considering global-illumination effects such as inter-reflections. Modeling the global-illumination effects with an efficient global rendering method would be an interesting future work. Second, DPIR does not explicitly model transmission. Generalizing inverse rendering with bidirectional scattering distribution functions may open up the applicability of DPIR for more diverse materials.

References

[1] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1670–1687, 2014. 1

[2] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. Dib-r++: learning to predict lighting and material with a hybrid differentiable renderer. *Adv. Neural Inform. Process. Syst.*, 34:22834–22848, 2021. 1

[3] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Trans. Graph.*, 33(6):1–12, 2014. 1

[4] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1060–1071, 2009. 1

[5] Jeffrey P Grossman and William J Dally. Point sample rendering. In *Rendering Techniques' 98: Proceedings of the Eurographics Workshop in Vienna, Austria, June 29—July 1, 1998 9*, pages 181–192. Springer, 1998. 1

[6] Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3):548–554, 2008. 1

[7] J Charles Hourcade and Alain Nicolas. Algorithms for antialiased cast shadows. *Computers & Graphics*, 9(3):259–265, 1985. 2

[8] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2023. 1, 3

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):1–14, 2023. 2

[10] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph.*, 25(3):735–745, 2006. 2

[11] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient brdf importance sampling using a factored representation. *ACM Trans. Graph.*, 23(3):496–505, 2004. 1

[12] Junxuan Li and Hongdong Li. Neural reflectance for shape recovery with shadow handling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16221–16230, 2022. 2

[13] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Trans. Image Process.*, 29:4159–4173, 2020. 1, 3

[14] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2475–2484, 2020. 1

[15] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37(6):1–11, 2018.

[16] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6123–6133, 2021. 1

[17] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *arXiv preprint arXiv:2305.17398*, 2023. 1

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[19] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Trans. Graph.*, 37(6):1–12, 2018. 1, 2, 4

[20] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000. 1

[21] William T Reeves, David H Salesin, and Robert L Cook. Rendering antialiased shadows with depth maps. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 283–291, 1987. 2

[22] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *Eur. Conf. Comput. Vis.*, pages 85–101. Springer, 2020. 1

[23] Carolin Schmitt, Simon Donne, Gernot Riegler, Vladlen Koltun, and Andreas Geiger. On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3493–3503, 2020. 1

[24] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Int. Conf. Comput. Vis.*, pages 8598–8607, 2019. 1

[25] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7495–7504, 2021. 1

[26] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[27] Xin Wei, Guojun Chen, Yue Dong, Stephen Lin, and Xin Tong. Object-based illumination estimation with rendering-aware neural networks. In *Eur. Conf. Comput. Vis.*, pages 380–396. Springer, 2020. 1

[28] Lance Williams. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, 1978. 2

[29] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Trans. Graph.*, 35(6):1–12, 2016.

1

[30] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In *Eur. Conf. Comput. Vis.*, pages 266–284. Springer, 2022. 1, 2, 3

[31] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Adv. Neural Inform. Process. Syst.*, 33:2492–2502, 2020. 2

[32] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3155–3164, 2019. 1

[33] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5565–5574, 2022. 1, 2, 4

[34] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5453–5462, 2021. 1, 3

[35] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–12, 2022. 1, 3

[36] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6):1–18, 2021. 1, 2

[37] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18643–18652, 2022. 1

[38] Zhenglong Zhou, Zhe Wu, and Ping Tan. Multi-view photometric stereo with spatially varying isotropic materials. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1482–1489, 2013. 1

[39] Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. Pointshop 3d: An interactive system for point-based surface editing. *ACM Trans. Graph.*, 21(3):322–329, 2002. 1

[40] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

[41] Matthias Zwicker, Jussi Rasanen, Mario Botsch, Carsten Dachsbacher, and Mark Pauly. Perspective accurate splatting. In *Proceedings-Graphics Interface*, number CONF, pages 247–254, 2004. 1