Shape and Reflectance Reconstruction in Uncontrolled Environments by Differentiable Rendering

Rui Li
KAUST
arthurlirui.com

Guangmin Zang
KAUST
guangming.zang@kaust.edu.sa

Miao Qi
KAUST
miao.qi@kaust.edu.sa

Wolfgang Heidrich
KAUST
wolfgang.heidrich@kaust.edu.sa

Abstract

Simultaneous reconstruction of geometry and reflectance properties in uncontrolled environments remains a challenging problem. In this paper, we propose an efficient method to reconstruct the scene's 3D geometry and reflectance from multi-view photography using conventional hand-held cameras. Our method automatically builds a virtual scene in a differentiable rendering system that roughly matches the real world's scene parameters, optimized by minimizing photometric objectives alternatingly and stochastically. With the optimal scene parameters evaluated, photo-realistic novel views for various viewing angles and distances can then be generated by our approach. We present the results of captured scenes with complex geometry and various reflection types. Our method also shows superior performance compared to state-of-the-art alternatives in novel view synthesis visually and quantitatively.

1. Introduction

An object's appearance is affected by many factors, including 3D geometry, surface reflection, and transmission, environmental lighting conditions, viewing angle, or camera position. We call all of these the scene parameters. To estimate the scene parameters from photographs is a very challenging task. It involves several inter-connected sub-problems. To name a few: 3D shape reconstruction, spatially-varying bidirectional reflectance distribution functions (SVBRDF) acquisition, environment lighting estimation, structure from motion, and multi-view stereo.

Previous work either requires other parameters to recover the desired ones or relies on specific constraints to narrow down the parameter search space. Thus, these methods have limitations for general scene parameter reconstruction. Specifically, there are several technical challenges: first, accurate 3D geometry is unknown or requires an expensive 3D scanner. Second, specular reflection is shape-sensitive having a significant influence on object appearance. Finally, the natural environment contains multiple direct light sources and an indirect ray path, which is unknown and hard to direct.

To overcome the above technical challenges and enable reconstruction in a typical user scenario, we propose a systematic scene parameter reconstruction method that jointly estimates 3D geometry, surface reflectance, specular coefficients, camera pose, position, and lighting condition using a differentiable inverse rendering framework. Note that there are no specific requirements for the the photography acquisition process, i.e., it only requires several multi-view photos or surround video, without extra constraints such as controlled lighting or exposure, or a specific environment setup, thus, our method enables in-the-wild reconstruction.

We assume that our scenes contain different objects with diffuse and specular reflection and distant environmental lighting. This scene is observed from multiple views from a wide range of angles, for example, by moving a single camera such as a cell phone around the scene. By taking a set of photos without a controlled light source or flashlight, our system can reconstruct object diffuse and specular reflectance, 3D geometry, environment light source. Our contributions are listed as follows,

- We propose a general parameterized framework to describe typical object appearance, enabling the direct reconstruction of a realistic scene from real-world multi-view photography for uncontrolled lighting conditions and general diffuse and specular scene. Thus, it can work entirely in the wild.

- We propose a memory-efficient solution for differentiable forward rendering and backward propagation.

Our framework can optimize real-world scene parameters in an iterative and stochastical fashion.

- Our method can also enable several photo-realistic applications such as novel view synthesis, environment light editing.

2. Related Work

Our approach lies at the intersection of several active research areas, namely image-based rendering (IBR), view synthesis, structure from motion (SfM), multi-view stereo (MVS), as well as 3D reconstruction. In this section, we give a brief review of the above inter-connected topics.

Image-based rendering. To create photo-realistic images, traditional pipelines rely on obtaining high-quality appearances and geometry models, on which global illumination is applied for the rendering process. Directly acquiring 3D models or surface appearance is time-consuming and challenging, especially for complex scenes containing transparent objects, thin structures, or human gestures. Image-based rendering [33, 4, 47, 17] is then developed to generate novel views by leveraging a sufficient number of input images for view interpolation and vision-based modeling. However, IBR methods require a large number of images from different viewpoints, which becomes a heavy burden for storage. Also, IBR results are highly scenes-specific, making it difficult to generalize to other scenes or edit scene parameters.

View synthesis. To tackle different problems, researchers have developed methods using point clouds [20, 1, 21, 6], and textured meshes [35, 16] as input, then novel views are rendered based on the input information. View synthesis approaches with 3D geometry and texture are proved to work efficiently in applications such as human bodies. Eslami *et al*. [8] propose generative query networks to render novel views by learning features embedding 3D scenes and geometrical properties, achieve successful novel view rendering when scene representation and camera extrinsic are given. However, since only a simplified feature vector is applied to represent the scene, acquired results are too coarse to be adopted in relatively complicated scenes. To overcome these limitations, multiplane images (MPIs) [10, 9, 22, 32] based view synthesis methods have been developed, aiming to improve image quality by learning the 3D structure representations. However, only novel views in limited angles can be generated with MPIs-based techniques. Nguyen-Phuoc *et al*. [25] propose RenderNet to represent scenes with sparse voxel grids and generate images with CNN decoder. There is a potential issue for view consistency since convolutional kernels are applied in this type of method. Therefore, tight voxel grids [30, 18] are proposed to improve the quality of generated novel images for view consistency. However, these methods require more storage space. Instead, representing the 3D scene with implicit functions [23, 31, 41, 16, 43, 34, 15] has gain popularity recently. For example, Mildenhall *et al*. [23] propose neural radiance fields (NeRF) that represent radiance field by a trained neural network. NeRF achieves good performance for generating spiral style synthetic view (limited view angle), but fail to inference large angle change or zoom in/out.

Structure from motion. Engel *et al*. [7] propose a direct monocular SLAM algorithm without feature point matching, which allows building large-scale, consistent maps of the environment. Schoenberger *et al*. [28] propose a systematical framework for incremental Structure-from-Motion pipeline, which improves the robustness and efficiency of correspondence search and incremental reconstruction of the large scene. A Multi-View Stereo (MVS) system [29] is introduced for robust and efficient dense modeling from unstructured image collections and jointly estimate depth and surface normal.

3D reconstruction. 3D reconstruction is a vital task in computer vision and graphics. Various data representations are used for different applications and tasks, such as light-field reconstruction [44, 48], tomographic reconstruction [36, 46, 45], polarization [24, 2, 3], shape reconstruction [11, 42], and feature-point based reconstruction [37, 38, 39]. In general, forward rendering and 3D reconstruction can be considered as a pair of forward-backward problems. Forward rendering simulates light traveling and imaging processes, which generates virtual images. By comparing against the real captured images, rendering error can be backward propagated as a gradient for each scene parameter (i.e., 3D geometry, BRDF, reflectance, etc.) by the gradient-based method to optimize the target parameter. [19, 26] propose decent solutions for estimating gradient based on Monte-Carlo sampling, to name a few.

3. Overview

We illustrate our proposed framework in Fig. (1): It takes a set of RGB images $I = \{I_k\}$ of the scene from arbitrary viewpoints and camera position as input. Then, our method automatically builds a virtual scene that roughly matches real world via parameter initialization, and iteratively optimize scene parameters for 3D geometry, diffuse and specular reflectance, camera pose and position, and environment lighting map by pushing the photometric consistency between rendering images and real photography in the same camera intrinsic and extrinsic. Finally, a set of optimized scene parameters can enable several photo-realistic applications: view synthesis, lighting editing, etc. Mathematically, our system can be described as a function of desired parameters as,

$$I = \Phi(\theta_g, \theta_d, \theta_s, \theta_l), \tag{1}$$

where Φ is the physically-based rendering process, which simulates the light rays traveling in a virtual scene. $\theta_g = \{x_0, \ldots, x_N\}$ is the set of 3D positions corresponding to the mesh vertices. $\theta_d = \{\theta_d(x_0), \ldots, \theta_d(x_N)\}$ is the set of diffuse per-vertex color reflectance, with $\theta_d(x) \in \mathbb{R}^3$. Similarly, θ_s is the set of specular reflectances. The illumination in the scene is modeled as environmental lighting l_0, approximated as a set of isotropic point sources $\theta_l = \{l_0, l_1, \ldots, l_M\}$, l_n is the intensity for n-th light sources. We use Mitsuba2 [26] as our physically-based rendering engine since it supports the auto-diff operation. Therefore, it enables an iterative optimization pipeline that back propagates rendering error to update scene parameters.

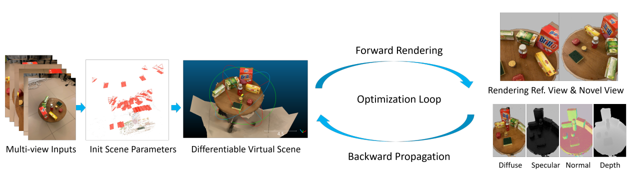

Figure 1. Inverse parameter optimization pipeline.

Initialization Our proposed method requires rough parameters for 3D mesh, camera extrinsic, and intrinsic in the initialization step. We apply Structure-from-Motion (SfM) methods [28] to obtain camera intrinsic, pose, and position for each view. Then, we obtain a rough 3D dense mesh using multi-view stereo methods [29] via a dense pixel-wise matching. In general, there are no special requirements for SfM and MVS; most of the state-of-the-art methods could give satisfying results as an initial guess of our method.

Differentiable and Inverse Reconstruction Initialized by rough parameters for camera intrinsic and extrinsic and a coarse 3D mesh from MVS, we build a virtual scene composed of a 3D mesh object, virtual camera, with a rough initial environment lighting map (e.g., constant white ambient light) in Mitsuba2. We render images for each virtual camera in Mitsuba2 and then minimize our objective by pushing virtual rendering images similar to real camera observations with proper regularization. The rendering error then back propagates to update scene parameters for 3D geometry, diffuse and specular reflectance, environment lighting map by auto-diff mechanism, and automatically gradient estimation for non-differentiable parameters. Specifically, we start from initial estimated 3D mesh with vertex reflectance, our method optimize diffuse reflectance θ_d, and optimize 3D geometry $\theta_g$relying on θ_d. After obtaining θ_d and θ_g, our method update the camera position and pose θ_c and environment light map θ_l by pushing photometric consistency, which allow us further update specular reflection parameter θ_s. We repeat this iterative optimization of inverse rendering and geometric reconstruction until the error converges.

Photo Realistic View Synthesis When the optimized parameters for our real scene are ready, we can synthesize virtual novel and edit current lighting conditions for photo-realistic performance by replacing interested parameters in Eqn. (1) and running the forward rendering process.

4. Method

In this section, we describe the details of system design and consideration.

Initial Geometry and Reflectance Estimation. To generate initial 3D geometry to describe our scene, we first apply the structure-from-motion (SfM) method [28] to generate sparse feature points in each image and then find the pairs of correspondence between those feature points in multiple overlapping images, point cloud, and 3D mesh can be further reconstructed via those matched feature points. Since the SfM method only generates a sparse point cloud and a rough mesh, we further generate a dense pixel-wise matching using a multi-view stereo method [29], which takes a sparse point cloud as input to compute per-pixel matching in each view of images, and fuse the geometry and surface reflectance by the screened Poisson method [13], i.e., dense vertex with a color channel.

In the data acquisition stage, we take photographs surrounding the scene target to ensure each feature point will appear in multiple views to reduce isolated mesh from the background. After initializing 3D geometry from images, a 3D mesh may contain inconsistent parts or isolated point clouds. Manual cropping 3D mesh by tools is required to avoid possible light path occlusion.

Camera Parameter Estimation In our virtual rendering scene, we set up virtual cameras for each real scene photo. The camera's position and orientation are estimated by SfM [28]. They are first represented as a 7-dimensional vector $Q = [Qw, Qx, Qy, Qz, Tx, Ty, Tz]$ consisting of a quaternion for the rotation and a translation vector quaternion for the rotation. We then convert this vector to a $R^{4\times4}$ matrix as initial camera parameters. We assume that all camera sensors have the same focus distance and field of view in the real and virtual scene.

Image Formation Model Our image formation model for each 2D pixel coordinate u can be formulated as

$$I(u) = L(w_o; x)\Delta t, \tag{2}$$

where $x \in \mathbb{R}^3$ is the 3D coordinate, and $u \in \mathbb{R}^2$ is the 2D image coordinate. $L(w_o; x)$ is the radiance from reflected from scene point x with outgoing direction w_o. Δt is the exposure time. According to rendering equation [12], the radiance of a non-emitting object can be written as

$$L_o(w_o; x) = \int_\Omega f_r(x, w_i, w_o) L_i(w_i; x)(w_i \cdot n(x))\, \mathrm{d}w_i, \tag{3}$$

where $L_o(w_o; x)$ and $L_i(w_i; x)$ are the incoming and outgoing radiance functions with direction w_o and w_i respectively for a 3D physical point x. $n(x)$ is the normal function.

In our virtual scene, an environment light source emits light to the virtual scene and bounces when hitting a 3D object. This environment light is represented as a set of isotropic point sources. Let x' be the 3D position of one of the environment light point sources. The ray tracing can be described by an iterative process. First, light rays originate at an environment light source as

$$L^0(w_i; x) = \theta_l(x'), \tag{4}$$

where the superscript notes the 0-*th* bounce of light. We assume that environment light is anisotropic light. Thus, the intensity is identical for any incoming direction w_i, i.e., $x' \to x$. When the light ray hitting the 3D object in the scene, $L_i^t(w_o; x)$ is the *t-th* bounce of outgoing light ray that directly hits camera aperture, and other reflected light with a different direction than w_o will start a new bounce until reaching a maximal bounce limit.

$$L_o^t(w_o; x) = \int_\Omega f_r(x, w_i, w_o) L_i^{t-1}(w_i; x)(w_i \cdot n(x))\, \mathrm{d}w_i, \tag{5}$$

the overall radiance received by the camera is the sum over all outgoing light with the direction w_o, and a maximum of T bounces:

$$L_o(w_o; x) = \sum_{t=0}^{T} L_o^t(w_o; x), \tag{6}$$

Reflection Model We assume that our scene contains a rough surface with diffuse and specular reflection without transmission, and Cook-Torrance (CT) model [5] with an optional microfacet distribution function, e.g., Beckmann [5], GGX [40], can describe a broad class of general real-world objects in reflection. Our reflectance model f_r can be expressed as follows:

$$\begin{aligned}
f_r(x, w_i, w_o) &= \theta_d(x) + \theta_s(x, w_i, w_o), && (7)\\
\theta_d(x) &= \frac{\rho_d(x)}{\pi}, && (8)\\
\theta_s(x, w_i, w_o) &= \rho_s(x) \frac{D(h,\alpha) G(n(x), w_i, w_o) F(h, w_i)}{4(n(x) \cdot w_i)(n(x) \cdot w_o)} && (9)
\end{aligned}$$

where θ_d and θ_s are diffuse and specular reflectance respectively, ρ_d and ρ_s are diffuse and specular albedos, h is the halfway vector, which is computed by normalizing the sum of the light direction w_i and view direction vectors w_o. Our $D(h, \alpha)$ is the microfacet distribution function. It contains several optional analytic distributions: Beckmann, Phong, GGX, etc. α specifies the roughness of surface micro-geometry along with the tangent and bitangent directions. We could also use a non-parametric distribution such as [24] to replace the analytic distribution as long as the reflectance for each halfway angle h can be calculated. G is a shadowing-masking function, and F is the Fresnel term similar in [40]. Thus, we reach our rendering function by combining Eqn. (2), Eqn. (3), Eqn. (7) as,

$$\begin{aligned}
I(u) &= \Phi(\theta_g, \theta_d, \theta_s, \theta_l)(x) && (10)\\
&= \Delta t \sum_{t=0}^{T} L_o^t(w_o; x) && (11)\\
&= \Delta t \sum_{t=0}^{T} \int_\Omega (\theta_d(x) + \theta_s(x)) L_i^{t-1}(w_i; x)(w_i \cdot n(x))\, \mathrm{d}w_i && (12)
\end{aligned}$$

θ_g exists at the 3D position of vertex x, $L_i^{t-1}(w_i; x)$ is incoming radiance from every possible direction with a bounce number of $t-1$, which is also an integral over the bounce number of $t-2$. For the case of $t = 0$ in Eqn. (4), $L^0(w_i; x)$ is the initial environment lighting θ_l, .

Objective Our objective function aims at jointly reconstructing the diffuse reflectance θ_d and specular reflectance θ_s, 3D geometry θ_g for each vertex and environment light source map θ_l.

$$\mathcal{O} = \sum_{k=1}^{K} \| M_k I_k - \Phi_k(\theta_g, \theta_d, \theta_s, \theta_l) \|_2^2, \tag{13}$$

where k is the index for the cameras. Rendering results only contain the target object without background, and thus the error of the background pixels will dominate the value of the objective function. To alleviate this effect, M_k is a precomputed binary mask for each view that removes background pixel contribution in the objective calculation, generated by binary segmentation methods with proper post-processing that preserves the main boundary of objects. M_k only needs to compute once. In our implementation, we apply GrabCut [27] to generate a binary mask for foreground and background, and [14] for view consistency.

Differentiable Optimization We implement our differentiable optimization pipeline in Mitsuba2 by using forward rendering and backward propagation manner. We first initialize scene parameters and assign an optimizer with a different learning rate for each parameter. The objective function is calculated as Eqn. (13) via multi-view photometric error. In the optimization stage, due to the limited GPU memory, we alternate between updating each parameter while keeping the others unchanged. By choosing one scene parameters $\theta \in \{\theta_d, \theta_g, \theta_s, \theta_l\}$, we iteratively update chosen θ in the inner loop. In the inner loop, we first render an image by current parameters and calculate the objective by comparing photometric error E between real observation and rendering results. Since the physically-based rendering system is non-differentiable, therefore, direct gradient calculation is unavailable analytically. Fortunately, the gradient for the target parameter can be estimated by Monte-Carlo sampling [26, 19]. Thus, modern optimizer can optimize non-differentiable parameters (e.g., Adam) by giving an estimated gradient. The overall pipeline is shown as 1

Algorithm 1 Alternating Differentiable Pipeline

```
procedure OPTIPARAM({I_0, ··· , I_k})
    init. parameters: {θ_d, θ_g, θ_s, θ_l}
    init. optimizer: opt
    Designed objective: O
    for i do                                         ▷ outer loop
        for k ∈ {0, ··· , K} do                      ▷ view loop
            for θ in {θ_d, θ_g, θ_s, θ_l} do
                for ii do                            ▷ inner loop
                    fix other {θ_d, θ_g, θ_s, θ_l} \ θ
                    Î_k = Φ(θ)
                    E = O(I_k, Î_k)
                    est. grad. ∇_θ E
                    opt(∇_θ E)                       ▷ update parameter θ
                end for
            end for
        end for
    end for
end procedure
```

5. Experiments

Camera setup. We demonstrate our method and evaluation by using an off-the-shelf mobile camera: iPhone 11pro. When using a mobile phone, we take a photo for the target scene from multiple viewpoints with an auto-focus setup. Our system requires 20-40 images and may take around 1-2 min for data capture. Alternatively, we can also use a video clip as input and decomposing it into 2D images. In this case, it takes only several seconds to acquire the data. Fig. (2) shows our data acquisition setup.

Figure 2. Our data acquisition setup. Left: we use a hand-held camera to capture the image in a natural lighting environment. Right: our system is flexible without specific lighting conditions and camera setup, enabling direct reconstruction from uncontrolled scenes in the wild.

Lighting Control. Our method does not require extra controlled lighting or flash in the scene. An ordinary ambient light or direct, diffuse light is sufficient. Therefore, our method is practical and easy to apply in the wild. At the initial stage, we set the radiance of the light source to be 0.5. Fig. (3) shows the reconstruction results of diffuse and specular reflectance, novel views, generated depth, and surface normal.

System Setup. Physically-based rendering is a GPU memory-consuming task, and we adjust several parameters for the sake of memory saving. We set the maximum number of bounce $T = 3$ for ray tracing bounce number, and raw images are downsampled for $8\times$, the number of sampling per pixel $spp = 1$ in iterative optimization stage (contain dense Monte-Carlo noise), and $spp = 16$ for final output rendering results with higher quality. We use Adam as our main optimizer, with dynamic learning for different task, as $\lambda_d = 0.1$, $\lambda_g = 0.5$, $\lambda_s = 0.01$, $\lambda_l = 0.05$. We set $\alpha = 0.1$ for general surface roughness.

Reconstruction of Scene Parameters We show the reconstructed results of scenes and novel view rendering in Fig. (4). We capture multiple images of the scene, reconstruct its 3D shape and diffuse and specular reflectance, and render several novel views from optimized scene parameters. Our novel view rendering can successfully recover true 3D geometry of the scene, accurate texture and details of objects, the photo-realistic glossy reflection of the surface.

5.1. Evaluation and Comparison

The quantitative evaluation with PSNR and SSIM measurements between synthetic novel view and the captured image is shown in Tab. 1. Our input images contain

Figure 3. Reconstruction Results. First row shows real scene photography, initial rendering result, diffuse reflectance θ_d, specular reflectance θ_s. The second row shows our rendering scene, novel view image, the depth map, and shading normal.

significant viewpoint changes and various captured distances, which explains NeRF [23]'s failure in these cases. Colmap [28, 29] directly reconstructed 3D geometry and vertex reflectance. Thus, as anticipated, rendering results with decent 3D structures but less accurate surface reflectance can be acquired. In contrast, our differentiable pipeline directly optimizes the scene parameter to match the real scene image. Therefore, it can accurately achieve photo-realistic high-quality performance in these viewpoint synthesis scenarios. We also notice that most virtual view synthesis methods will fail in zoom-in or zoom-out cases. We compare our solution with other state-of-the-arts[23][29] in the cases of changing viewing angles and captured distances. Visual comparison is shown in Fig. (5).

Objective Evolution. We show in Fig. (6) the objective evolution during the multiple optimization stages by using our real scene dataset. There are four stages to optimize our scene parameters iteratively: $\{\theta_d, \theta_g, \theta_s, \theta_l\}$. θ_d stage has a significant objective decrease for around 200 iterations since the diffuse reflectance has significant impact on the appearance in most scenes. θ_g continues optimizing geometry by updating the 3D vertex position of the mesh and re-compute surface normal. We also notice that the gradient of θ_g is comparatively smaller than the gradient of θ_d since geometry's gradient mainly exists in silhouettes edge[19]. Thus only a few vertexes near the view's silhouettes will contain a compelling value, and all other vertexes will only have almost zero gradients.

	Fruit		Table 1	
	PSNR	SSIM	PSNR	SSIM
NeRF	14.35	0.28	15.65	0.32
Colmap	17.51	0.70	18.67	0.75
Initial	15.79	0.68	17.97	0.69
Diff Opt.	26.31	0.82	27.21	0.85
Geo Opt.	27.42	0.93	27.45	0.86
Spe Opt.	27.45	0.94	27.95	0.87
Light Opt.	28.78	0.95	28.03	0.87
Ours	29.02	0.96	28.45	0.88

Table 1. PSNR and SSIM evaluations for each approach and stage.

Multi-view images contain more silhouettes edge, which can help geometry optimization but requires dense sampling of viewpoints. θ_s has less contribution to decrease the objective but significantly improve visual performance. θ_l stage recovers rough environment lighting. Since scattering diffuse reflection will eliminate light bounce information, θ_l will only preserve low-resolution or ambient lighting.

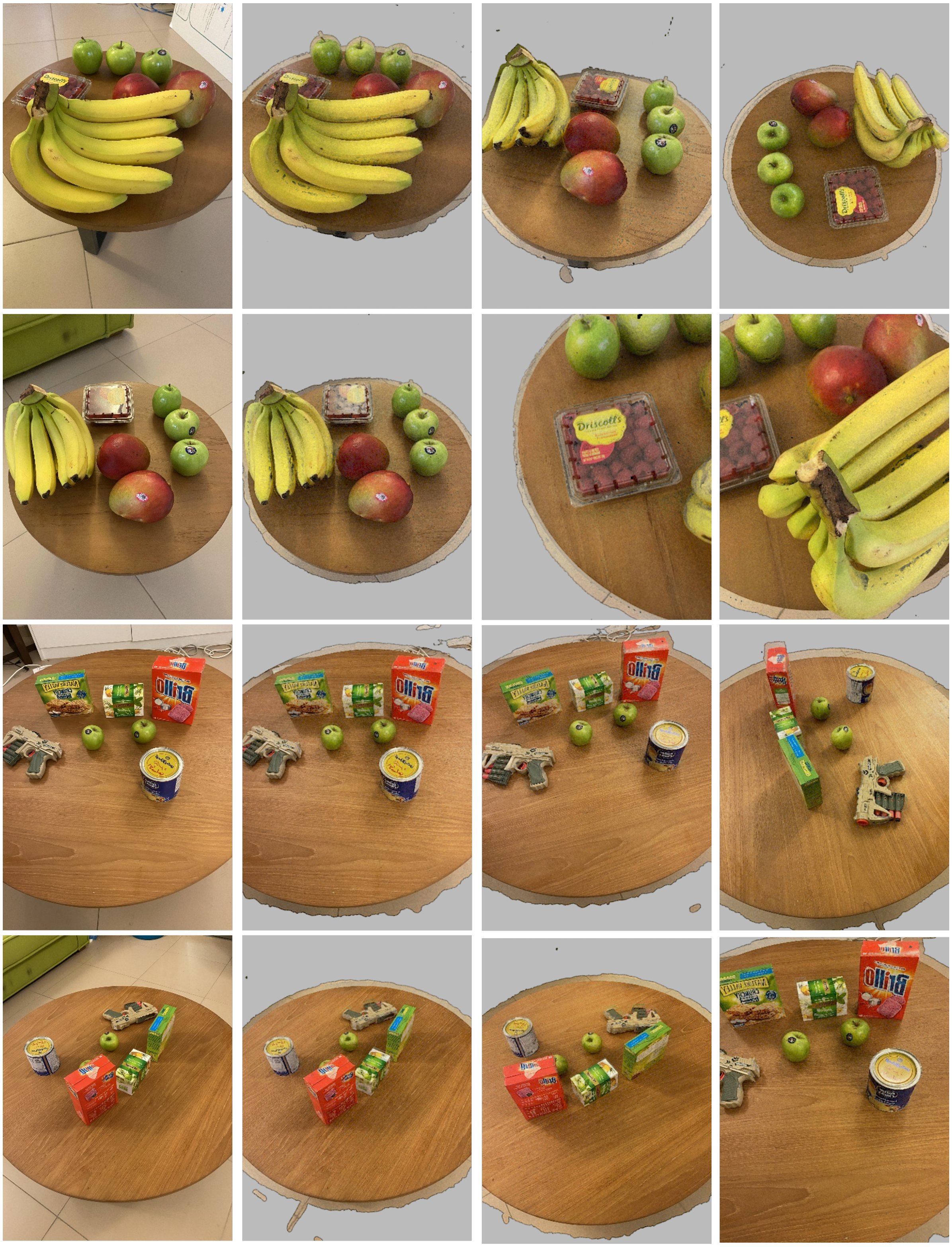

Figure 4. Full scene reconstruction. Two scenes are shown: `Fruit` (Row 1, 2) and `Table 2` (Row 3, 4). Column 1 is real captured images, column 2 is corresponding virtual view, column 3 and 4 are rendering synthetic novel views.

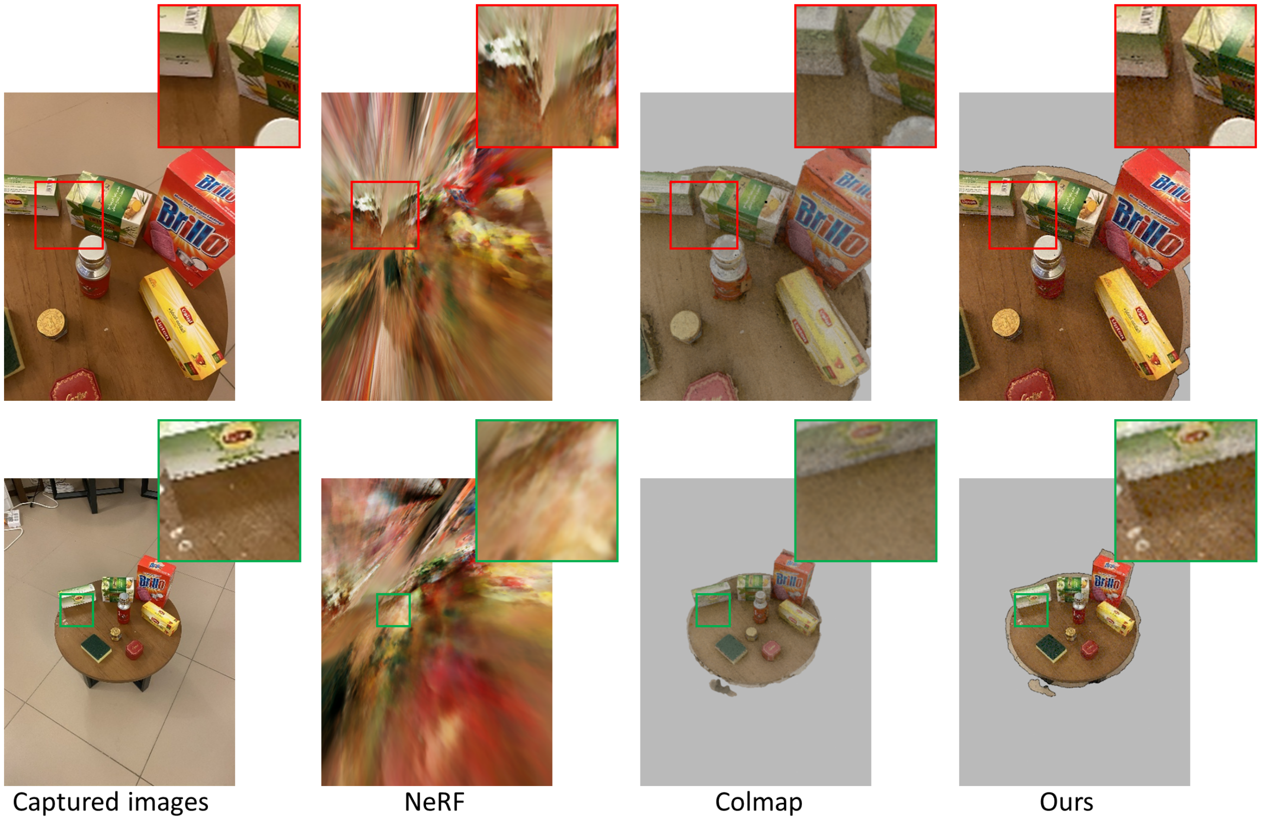

Figure 5. Various viewpoints and camera distance synthesis. From left to right: The captured images (`Table 1`), NeRF results [23], Colmap [29], and Ours

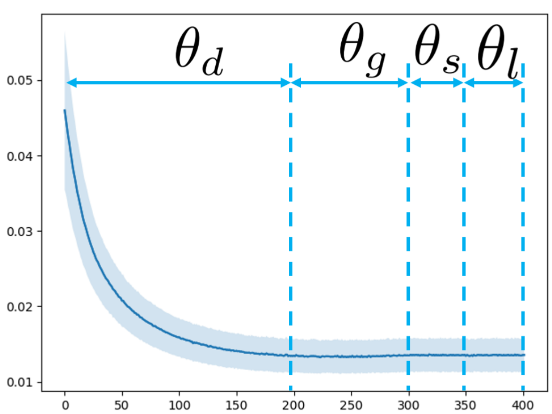

Figure 6. Multi-phrase objective evaluation. Our curve contains 4 stage iterations for θ_d (0-200), θ_g (201-300), θ_s (301-350), θ_l (351-400) respectively, and center dark blue curve is the mean objective of all the viewpoints, light blue area is the range of objective value.

Computational Speed. Our computational platforms are Intel Xeon(R) Gold 6242 CPU @ 2.80GHz $\times$ 32, GeForce RTX 2080 Ti with 11GB GDDR6 memory and support hardware ray tracing, 250GB RAM. Our proposed method runs around $140ms$ per image/iteration, 400 iterations to optimize a viewpoint, and 10-80 images per scene.

6. Conclusion and Future Work

We propose a novel differentiable optimization framework that simultaneously reconstructs scene parameters: diffuse and specular reflectance, geometry, environment lighting using the hand-held camera from an uncontrolled environment. Unlike previous works that require expensive hardware or carefully design lighting, our method can handle a wide range of materials and general ambient lighting, offers an attractive and efficient solution, facilitating in-the-wild scene reconstruction for a wider public, enables a photo-realistic view synthesis.

Differentiable scene reconstruction still has the potential to achieve significant progress. For example, current mesh optimization mainly focuses on optimizing vertex position, where a more advanced mesh operation, e.g., edge collapses, is not supported by gradient-based optimization. To chose proper regularization or prior of mesh or reflectance is another direction to explore, replacing simple photometric objective for specific optimization purposes, e.g., simplifying mesh or topology, capturing SVBRDF. Recovering environment lighting is a highly ill-posed problem because diffuse reflection will significantly erase ray tracing information of each bounce. However, we can take multiple lighting photos with a fixed viewpoint to narrow the search space.

References

[1] Kara-Ali Aliev, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2(3):4, 2019. 2

[2] Seung-Hwan Baek, Daniel S. Jeon, Xin Tong, and Min H. Kim. Simultaneous acquisition of polarimetric svbrdf and normals. *ACM Trans. Graph.*, 37(6), Dec. 2018. 2

[3] Seung-Hwan Baek, Tizian Zeltner, Hyun Jin Ku, Inseung Hwang, Xin Tong, Wenzel Jakob, and Min H. Kim. Image-based acquisition and modeling of polarimetric reflectance. *ACM Trans. Graph.*, 39(4), 2020. 2

[4] SC Chan, Heung-Yeung Shum, and King-To Ng. Image-based rendering and synthesis. *IEEE Signal Processing Magazine*, 24(6):22–33, 2007. 2

[5] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, Jan. 1982. 4

[6] Peng Dai, Yinda Zhang, Zhuwen Li, Shuaicheng Liu, and Bing Zeng. Neural point cloud rendering via multi-plane projection. In *Proc. CVPR*, pages 7830–7839, 2020. 2

[7] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proc. ECCV*, 2014. 2

[8] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proc. CVPR*, pages 2367–2376, 2019. 2

[10] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proc. CVPR*, pages 5515–5524, 2016. 2

[11] Matheus Gadelha, Rui Wang, and Subhransu Maji. Shape reconstruction using differentiable projections and deep priors. In *Proc. ICCV*, pages 22–30, 2019. 2

[12] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 143–150, New York, NY, USA, 1986. ACM. 4

[13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013. 3

[14] Rui Li and Wolfgang Heidrich. Hierarchical and view-invariant light field segmentation by maximizing entropy rate on 4d ray graphs. In *ACM Trans. Graph.*, 2019. 4

[15] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. *arXiv preprint arXiv:2012.01714*, 2020. 2

[16] Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhoefer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. Neural human video rendering by learning dynamic textures and rendering-to-video translation. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 2

[17] Zhong Liu, Zhouchi Lin, Xiguang Wei, and Shing-Chow Chan. A new model-based method for multi-view human body tracking and its application to view transfer in image-based rendering. *IEEE transactions on multimedia*, 20(6):1321–1334, 2017. 2

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[19] Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.*, 38(6), Dec. 2019. 2, 5, 6

[20] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. Lookingood: Enhancing performance capture with real-time neural re-rendering. *arXiv preprint arXiv:1811.05029*, 2018. 2

[21] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proc. CVPR*, pages 6878–6887, 2019. 2

[22] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):1–14, 2019. 2

[23] 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. In *Proc. ECCV*, 2020. 2, 6, 8

[24] 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):267:1–12, 2018. 2, 4

[25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proc. ICCV*, pages 7588–7597, 2019. 2

[26] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph.*, 38(6), Dec. 2019. 2, 3, 5

[27] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 4

[28] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 2, 3, 6

[29] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 2, 3, 6, 8

[30] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. CVPR*, pages 2437–2446, 2019. 2

[31] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019. 2

[32] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proc. CVPR*, pages 175–184, 2019. 2

[33] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2

[34] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020. 2

[35] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 38(4):1–12, 2019. 2

[36] Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic reconstruction of transparent objects. In *ACM SIGGRAPH 2006 Sketches*, pages 55–es. 2006. 2

[37] Benjamin Ummenhofer and Thomas Brox. Dense 3d reconstruction with a hand-held camera. In Axel Pinz, Thomas Pock, Horst Bischof, and Franz Leberl, editors, *Pattern Recognition*, pages 103–112, 2012. 2

[38] Benjamin Ummenhofer and Thomas Brox. Point-based 3d reconstruction of thin objects. In *Proc. ICCV*, December 2013. 2

[39] B. Ummenhofer and T. Brox. Global, dense multiscale reconstruction for a billion points. In *Proc. ICCV*, 2015. 2

[40] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Eurographics Conference on Rendering Techniques*, page 195–206, 2007. 4

[41] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proc. CVPR*, pages 7467–7477, 2020. 2

[42] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *arXiv preprint arXiv:1805.03482*, 2018. 2

[43] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020. 2

[44] Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. *ACM Trans. Graph.*, 35(3), Mar. 2016. 2

[45] Guangming Zang, Mohamed Aly, Ramzi Idoughi, Peter Wonka, and Wolfgang Heidrich. Super-resolution and sparse view ct reconstruction. In *Proc. ECCV*, pages 137–153, 2018. 2

[46] Guangming Zang, Ramzi Idoughi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. Space-time tomography for continuously deforming objects. *ACM Trans. Graph.*, 37(4), July 2018. 2

[47] Cha Zhang and Tsuhan Chen. A survey on image-based rendering—representation, sampling and compression. *Signal Processing: Image Communication*, 19(1):1–28, 2004. 2

[48] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2018. 2