Efficient Multi-View Inverse Rendering Using a Hybrid Differentiable Rendering Method

Xiangyang Zhu\textsuperscript{1,2}, Yiling Pan\textsuperscript{1,2}, Bailin Deng\textsuperscript{3} and Bin Wang\textsuperscript{1,2}\textsuperscript{*}

\textsuperscript{1}School of Software, Tsinghua University, China
\textsuperscript{2}Beijing National Research Center for Information Science and Technology (BNRist), China
\textsuperscript{3}School of Computer Science and Informatics, Cardiff University, UK
{zhuxy20, pyl16}@mails.tsinghua.edu.cn, DengB3@cardiff.ac.uk, wangbins@tsinghua.edu.cn

Abstract

Recovering the shape and appearance of real-world objects from natural 2D images is a long-standing and challenging inverse rendering problem. In this paper, we introduce a novel hybrid differentiable rendering method to efficiently reconstruct the 3D geometry and reflectance of a scene from multi-view images captured by conventional hand-held cameras. Our method follows an analysis-by-synthesis approach and consists of two phases. In the initialization phase, we use traditional SfM and MVS methods to reconstruct a virtual scene roughly matching the real scene. Then in the optimization phase, we adopt a hybrid approach to refine the geometry and reflectance, where the geometry is first optimized using an approximate differentiable rendering method, and the reflectance is optimized afterward using a physically-based differentiable rendering method. Our hybrid approach combines the efficiency of approximate methods with the high-quality results of physically-based methods. Extensive experiments on synthetic and real data demonstrate that our method can produce reconstructions with similar or higher quality than state-of-the-art methods while being more efficient.

1 Introduction

Inverse rendering, which recovers 3D geometry, reflection properties and illumination from 2D images, is a long-standing and challenging problem in computer graphics and vision [Patow and Pueyo, 2003]. Benefiting from the advances in deep neural networks, many deep learning-based methods learn to obtain the materials and plane normals of near-flat objects in a data-driven way [Li et al., 2018b; Aittala et al., 2016; Gao et al., 2019]. However, these methods struggle to deal with complex geometry, resulting in limited application in practice. Some other deep learning-based methods use 3D geometric representations such as signed distance fields [Yariv et al., 2020; Zhang et al., 2021b], tetrahedral meshes [Munkberg et al., 2021], and occupancy functions [Mildenhall et al., 2020; Zhang et al., 2021c; Boss et al., 2021; Srinivasan et al., 2021]; Zhang et al., 2021a; Zhang et al., 2022], to handle more complex geometries. However, such geometry representations may require post-processing in order to be used in traditional graphics pipelines [Lorensen and Cline, 1987; Remelli et al., 2020], which may cause material information loss and affect the rendering image quality.

The emergence of differentiable rendering techniques [Kato et al., 2020] means that image loss can be back-propagated along the rendering pipeline to solve inverse rendering problems, which promotes the development of multi-view inverse rendering tasks using triangular meshes as the geometry representation. Approximate differentiable rendering methods [Loper and Black, 2014; Kato et al., 2018; Liu et al., 2019; Chen et al., 2019] utilize simplified rendering processes and can solve the inverse rendering problem efficiently, but the simple material models they use usually lead to poor visual effects. On the contrary, physically-based differentiable rendering methods [Li et al., 2018a; Li et al., 2021] can reconstruct high-quality physically-based rendering (PBR) materials in the path tracing manner, but their realistic results come at a high computational cost. Although some methods have been proposed to accelerate the reconstruction process [Luan et al., 2021], they often impose strong assumptions on the lighting conditions to narrow down the parameter search space, which hinders their wide application.

In this paper, we propose a hybrid differentiable rendering method to reconstruct triangular meshes and PBR materials from multi-view real-world images captured by conventional hand-held cameras (e.g. mobile phones) with uncontrolled lighting conditions. Our method consists of two phases. In the initialization phase, we use traditional methods to reconstruct a rough triangular mesh for the scene. Then in the optimization phase, we take a hybrid approach to optimize the scene geometry along the rendering pipeline to solve inverse rendering problems, which promotes the development of multi-view inverse rendering tasks using triangular meshes as the geometry representation. Approximate differentiable rendering methods [Loper and Black, 2014; Kato et al., 2018; Liu et al., 2019] can reconstruct high-quality physically-based rendering (PBR) materials in the path tracing manner, but their realistic results come at a high computational cost. Although some methods have been proposed to accelerate the reconstruction process [Luan et al., 2021], they often impose strong assumptions on the lighting conditions to narrow down the parameter search space, which hinders their wide application.

In this paper, we propose a hybrid differentiable rendering method to reconstruct triangular meshes and PBR materials from multi-view real-world images captured by conventional hand-held cameras (e.g. mobile phones) with uncontrolled lighting conditions. Our method consists of two phases. In the initialization phase, we use traditional methods to reconstruct a rough triangular mesh for the scene. Then in the optimization phase, we take a hybrid approach to optimize the scene geometry along the rendering pipeline to solve inverse rendering problems, which promotes the development of multi-view inverse rendering tasks using triangular meshes as the geometry representation. Approximate differentiable rendering methods [Loper and Black, 2014; Kato et al., 2018; Liu et al., 2019] can reconstruct high-quality physically-based rendering (PBR) materials in the path tracing manner, but their realistic results come at a high computational cost. Although some methods have been proposed to accelerate the reconstruction process [Luan et al., 2021], they often impose strong assumptions on the lighting conditions to narrow down the parameter search space, which hinders their wide application.

In summary, our contributions include:

- We propose a novel hybrid differentiable rendering optimization method based on triangular meshes, which utilizes an...
approximate differentiable rendering method to optimize the geometry, and a physically-based differentiable rendering method to get the PBR materials.

- The proposed pipeline is user-friendly and can work end-to-end. Furthermore, the optimized scene parameters can be easily deployed on mainstream commercial graphics engines without the need for conversion, making it applicable to a wide range of applications such as virtual reality and augmented reality.

- We conduct extensive experiments on both synthetic and real-world data, which demonstrate that our method outperforms or achieves comparable results to the state-of-the-art methods while being significantly more efficient.

2 Related Work

Shape Reconstruction. Reconstructing object geometry is a long-standing problem in computer vision and graphics. In traditional methods, the Structure-from-Motion (SfM) method [Schonberger and Frahm, 2016] is first applied to generate sparse feature points and find their correspondence to further generate a sparse point cloud and a rough mesh. Then, the Multi-View-Stereo (MVS) method [Schönberger et al., 2016] is leveraged to generate dense pixel-level matching. Finally, a dense mesh with vertex color is generated by the Poisson reconstruction method [Kazhdan and Hoppe, 2013]. A few recent learning-based methods assume that the target object mesh is homeomorphic with the sphere. The method in [Wang et al., 2018] uses an image feature network (2D CNN) to extract perceptual features from the input image, and a cascaded mesh deformation network (GCN) to progressively deform an ellipsoid mesh into the desired 3D model. These methods can only reconstruct rough geometry, which is insufficient for downstream tasks such as VR and AR.

Reflectance Reconstruction. Reflectance model, i.e., spatially-varying bidirectional reflectance distribution function (SVBRDF), describes how light interacts with surfaces in the scene. Traditional SVBRDF reconstruction methods [Matusik, 2003; Lensch et al., 2003; Holroyd et al., 2010; Dong et al., 2010; Chen et al., 2014; Dong et al., 2015; Kang et al., 2018] rely on dense input images measured using auxiliary equipment, e.g., gantry. Some other works focus on [Zhou et al., 2016; Kim et al., 2017; Park et al., 2018] exploiting the structure of SVBRDF parameter spaces to reduce the requirement for the number of input images. Additionally, some data-driven works [Li et al., 2018b; Aittala et al., 2016; Gao et al., 2019] have been introduced recently to produce plausible SVBRDF estimations for near-flat objects using a small number of input images. Despite their ease of use, these methods struggle to handle more complex objects.

Differentiable Rendering. Differentiable methods are reviewed in detail in [Kato et al., 2020]. Here we focus on methods closely related to our work. Traditional rendering pipelines, e.g., rasterization and ray-tracing, are not differentiable due to some discrete parts, which means that they cannot work with the gradient descent method just like neural networks. The emergence of differentiable rendering has changed all. Some approximately-based methods have been proposed recently. [Loper and Black, 2014; Kato et al., 2018; Kato and Harada, 2019] compute approximated gradients to optimize scene parameters. Besides, [Liu et al., 2019; Chen et al., 2019] replace the z-buffer-based triangle selection of a vanilla rasterization process with a probabilistic manner. Unfortunately, the result is not so good due to inaccuracies introduced by these methods. On the contrary, Monte Carlo edge sampling based methods [Li et al., 2018a; Zhang et al., 2019] provide unbiased gradient estimates capable of producing plausible results. But these methods are resource-consuming because of their path tracing module, hindering their generalization.

Differentiable Rendering based Multi-view Inverse Rendering. The emergence of differentiable rendering boosts the development of the inverse rendering techniques. Several prior works leverage differentiable rendering methods to solve the inverse problem. Wu et al. [2020] propose a completely unsupervised method for face reconstruction from just one face image using the approximately-based differentiable method, which does not rely on existing 3D morphable face models. Luan et al. [2021] leverage the Monte Carlo edge sampling based methods to reconstruct fine geometry and plausible material. However, they assume that the camera and light are collocated. In addition, a rough geometry scanned by professional equipment is also needed. Although these settings could narrow down the solution space, they also reduce the generalization of the method. The method proposed by Li et
al. [2021] reduces the requirements for input. It takes as input multi-view wild scene images, and reconstructs the initial geometry through MVS, then use the general Monte Carlo path tracing differentiable renderer to optimize the geometry, illumination, and geometry. Although it can achieve good results, it is resource-consuming and time-consuming.

3 Our Method

3.1 Overview

Our inverse rendering pipeline, shown in Figure 2, takes as input a set of RGB images of some object obtained by conventional hand-held cameras from several viewpoints with uncontrolled lighting conditions. Then, we utilize SfM and MVS method to reconstruct a virtual scene roughly matching the real scene in the initialization phase. In the optimization phase, we first optimize the geometry using the approximately-based differentiable rendering method, then optimize reflection using the physically-based differentiable rendering method. Both methods work iteratively by pushing the loss between rendering images and the ground truth in the camera pose.

3.2 Initialization Phase

Our reconstruction pipeline is designed to work in complex, unstructured scenes where the input images have no additional information, such as depth. To tackle these challenges, we leverage traditional methods to obtain an initial geometry. Specifically, as shown in Figure 2(a), we use the SfM method from [Schonberger and Frahm, 2016] to generate sparse feature points in each image. These points are then used to reconstruct a sparse point cloud and a rough mesh by matching the feature points across images. Next, we use the MVS reconstruction method from [Schönberger et al., 2016] to generate dense pixel-level matching. Finally, a dense mesh is then obtained from Poisson surface reconstruction [Kazhdan et al., 2006], with vertex colors derived from the input images.

In terms of lighting, we use an environment map as our lighting source. The resolution of the environment map is 512 × 128. The environment map is a learnable parameter that is inferred during the material optimization phase. In the beginning, we assume that the light is white and the three channels of our environment map are set to (0.5, 0.5, 0.5).

3.3 Optimization Phase

In the optimization phase, starting from the initial geometry obtained in the previous step, we adopt a hybrid differentiable rendering method to further optimize the geometry and reflectance. Specifically, we utilize an approximate differentiable rendering method to optimize the geometry, and a physically-based differentiable rendering method to optimize the reflectance. Our hybrid approach is motivated by the following observations in experiments:

- Although the geometry reconstructed by SfM and MVS in the initialization phase provides a good approximation of the scene, there can be some defects in the boundary regions.
- Approximate differentiable rendering methods dedicated to...
We leverage a modified Cook-Torrance (CT) model [Cook et al., 1982] based on [Zeltner et al., 2021] as our reflectance model, which contains a rough surface with diffuse and specular reflection without refraction. Mathematically, our reflectance model can be described as the following,

\[ f_r(\omega_o, \omega_i; x) = \theta_d(x) + \theta_s(\omega_o, \omega_i; x), \]

where \( \theta_d \) and \( \theta_s \) are the diffuse and specular reflectance:

\[ \theta_d(x) = \rho_d(x)/\pi, \]
\[ \theta_s(\omega_o, \omega_i; x) = \rho_s(x)D(h, \alpha)G(\omega_o, \omega_i, n(x)) \frac{(n(x) \cdot \omega_o)(n(x) \cdot \omega_i)}{\pi}. \]

Specifically, \( \rho_d \) is diffuse albedos, \( \rho_s \) is specular albedos, and \( h \) is the halfway vector. \( D(h, \alpha) \) is the microfacet distribution function which is GGX [Walter et al., 2007] used in our work. \( \alpha \) and \( n(x) \) denote the surface’s roughness and normal, respectively. \( G \) is a shadowing-masking function. Like other works, we also ignore the Fresnel effect which cannot be observed in our scene. To optimize the reflectance parameters \( \theta_i = (\theta_d, \theta_s) \), we minimize the following reflectance loss:

\[ L_{\text{ref}} = \lambda_{\text{rgb}} L_{\text{rgb}}(\mathcal{I}_{\text{rgb}}(\theta_i); \mathcal{I}_{\text{rgb}}) + \lambda_{\text{reg}} R(\theta_i), \]

where \( L_{\text{rgb}} = \| \mathcal{I}_{\text{rgb}} - \mathcal{I}_{\text{rgb}}(\mathcal{I}_{\text{rgb}}(\theta_i)) \|_1 \) measures the \( \ell_1 \) norm of the difference between the rendered color image \( \mathcal{I}_{\text{rgb}}(\theta_i) \) and the ground truth \( \mathcal{I}_{\text{rgb}} \). \( \mathcal{R}(\theta_i) \) is a regularizer for the diffuse albedo \( \theta_d \) and specular albedo \( \theta_s \). Similar to [Schmitt et al., 2020], we assume that nearby pixels with similar diffuse albedos have similar specular albedos. So we choose

\[ \mathcal{R}(\theta_i) = \sum_p \| \theta_s[p] - \left( \frac{\sum_q \theta_s[q] k_{p,q}}{\sum_q k_{p,q}} \right) \|_1, \]

where \( k_{p,q} = \exp \left( -\frac{|p-q|^2}{2\sigma^2} - \frac{(\theta_d[p] - \theta_d[q])^2}{2\sigma^2} \right) \) and \( p, q \) are two specific mesh vertices.

4 Experiments

We perform quantitative and qualitative evaluation of our method using extensive experiments on both synthetic and real data, and compare it with state-of-the-art methods.

Implementation Details. We implement the geometry reconstruction module in the initialization phase using COLMAP\footnote{https://github.com/colmap/colmap} [Schönberger et al., 2016; Schonberger and Frahm, 2016], a general-purpose Structure-from-Motion and Multi-View Stereo pipeline. In the optimization phase, we implement the approximate differentiable rendering based on Pytorch3d [Ravi et al., 2020], and the physically-based differentiable rendering based on Mitsuba 2 [Nimier-David et al., 2020]. The rest of our optimization pipeline is implemented with PyTorch using the Adam optimizer. For geometry optimization, we use the weights \((\lambda_{\text{sil}}, \lambda_{\text{lap}}, \lambda_{\text{edge}}, \lambda_{\text{normal}}) = (1.0, 1.0, 1.0, 0.01)\) for Eq. 1, and 0.001 for the learning rate. The ground-truth silhouette images required for Eq. 1 are obtained either from rendering (for synthetic data) or using a background removal tool\footnote{https://www.remove.bg/} (for real data). For reflectance optimization, we use the weights \((\lambda_{\text{rgb}}, \lambda_{\text{reg}}) = (0.1, 1.0)\) for
We train and test our method on one NVIDIA RTX 3090 GPU. For geometry optimization, our implementation uses 200 iterations per view and takes about 0.5 hours per example. For reflectance optimization phase, our implementation uses 400 iterations per view and takes 1.5~2.5 hours per example.

**Synthetic Data.** Our synthetic data is created using meshes and textures from [Zhou et al., 2016] and the Internet, covering different materials and geometries. Concretely, for each object, we render 400 images with colored environmental lighting using graphics engines, 300 for training, and the left 100 for novel view synthesis testing, whose viewpoints are evenly distributed inside the upper hemisphere. In addition, we also render the same object with two other environment lighting maps as the ground truth for the following relighting performance testing. To quantitatively evaluate our method, we use three image quality metrics—LPIPS [Zhang et al., 2018], SSIM and PSNR—to compare the rendering results with the corresponding ground truth images. Figure 3 shows examples of results from our method and their ground truth, as well as their evaluation metric values. Both the quantitative metrics and the qualitative visualization show that our novel views and relighting results match the ground truth closely. Fig. 6 shows the diffuse and specular albedo for the lemon model.

**Real Data.** We evaluate our method on multiple real-world images from the DTU datasets [Aanæs et al., 2016], where the objects are glossy and the illumination is static across different views. We use two objects from the dataset, *shiny scan114 buddha* and *scan110 ghost*, and discard photos with strong shadows. Our inverse rendering results are shown in Figure 4. We can see that our pipeline generates photo-realistic novel views, and results of relighting and material editing.

**Comparison with State-of-the-art Methods.** As far as we are aware, there is no prior work tackling exactly the same problem as our work: reconstructing triangle meshes and PBR materials from multi-view real-world object images with uncontrolled lighting conditions. The methods closest to our method are [Luan et al., 2021; Li et al., 2021], but no source code or data is released. It is worth noting that our method addresses the shortcomings of [Li et al., 2021] in dealing with geometry, while overcoming the limitation of [Luan et al., 2021] which requires the camera and point lighting to be in the same position and the images to be taken in dark scenes.

We compare our method to the most related neural rendering approaches, including PhySG [Zhang et al., 2021b], NeRF [Mildenhall et al., 2020] and IDR [Yariv et al., 2020], in terms of novel view synthesis. These approaches are different from our method in the model of light transport: NeRF uses the occupancy function to describe geometry and appearance maps and gets the pixel color according to location and viewing direction in a ray-marching way, while PhySG and IDR use SDF to represent geometry and material. In addition, unlike the aforementioned two methods that focus solely on novel view synthesis, PhySG also performs inverse rendering tasks, using Spherical Gaussian functions to describe materials and illumination. Qualitative and quantitative comparisons between different methods on our synthetic data are depicted in Figure 5 and Table 1. The unsatisfactory performance of PhySG may be due to the presence of strong specular highlights in our...
Figure 4: With our pipeline, we can synthesize novel views and edit the materials and lighting of the real-world captures.

Figure 5: We qualitatively compare our results with PhySG [Zhang et al., 2021b], NeRF [Mildenhall et al., 2020] and IDR [Yariv et al., 2020].

Figure 6: Our diffuse and specular albedo. We have adjusted the tonal range to make the variation of specular albedo more visible.

synthetic data, which are difficult to describe using the Spherical Gaussian functions employed by PhySG. NeRF performs relatively poorly in view synthesis because its volumetric representation does not concentrate colors around surfaces as well as surface-based approaches. IDR does a better job in view synthesis because of its view-dependence model. However, it still struggles to synthesize specular highlights due to the non-physical model of appearance. In contrast, our method models highlight well, benefiting from our PBR materials. We also compare the running time, where our training time is the sum of initialization time and optimization time. Thanks to the combination of efficiency from the approximate method and the accuracy from the physically-based method, our hybrid approach runs significantly faster than the two baselines both in training and testing. In addition, our optimized geometry and material can be deployed directly on mainstream graphics engines.

We also compare our method with the latest related work IRON [Zhang et al., 2022] which adopts neural representations and also leverages a hybrid optimization scheme. Note that IRON assumes a point light source, while our method allows for uncontrolled lighting conditions. As a result, IRON performs poorly in complex lighting situations. Figure 9 com-
Figure 7 shows our ablation study on the number of input images, which depicts how the number of input images affects our reconstruction quality. With too few input images, the optimization may become highly under-constrained, making it difficult to produce accurate synthesis results. In our experiments, 150 images are sufficient to produce a quite good result. With 250 or more input images, our results will closely match the ground truth. More ablation studies can be found in the supplementary materials.

**Ablation Study.** Figure 7 shows our ablation study on the number of input images, which depicts how the number of input images affects our reconstruction quality. With too few input images, the optimization may become highly under-constrained, making it difficult to produce accurate synthesis results. In our experiments, 150 images are sufficient to produce a quite good result. With 250 or more input images, our results will closely match the ground truth. More ablation studies can be found in the supplementary materials.

Figure 8 compares the two methods in different lighting conditions. Our method can achieve similarly good results as IRON on point-lighting data, while being superior with more complex lighting. In particular, IRON cannot handle the highlights on the surface of objects under environment lighting because their light source cameras are in the same position, while our method can handle it well. In addition, for some flat data like the hotdog in Figure 8, IRON struggles to obtain appropriate geometry, while our method can produce an accurate reconstruction.

Figure 8: Comparison between our method and IRON [Zhang et al., 2022] on the hotdog data.

**Table 1: Comparison between PhySG [Zhang et al., 2021b], NeRF [Mildenhall et al., 2020], IDR [Yariv et al., 2020] and our method. #train and #test denote the size of training set and test set, respectively. Training time is in hours and rendering time is in seconds.**

<table>
<thead>
<tr>
<th>Objects</th>
<th>Training LPIPS</th>
<th>Training SSIM</th>
<th>Training PSNR</th>
<th>Testing LPIPS</th>
<th>Testing SSIM</th>
<th>Testing PSNR</th>
<th>Training Time (h)</th>
<th>Testing Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic Pig</td>
<td>0.2200</td>
<td>0.784</td>
<td>17.39</td>
<td>0.425</td>
<td>0.928</td>
<td>23.98</td>
<td>~15</td>
<td>~4</td>
</tr>
<tr>
<td>Synthetic Pomegranate</td>
<td>0.8578</td>
<td>0.859</td>
<td>27.19</td>
<td>0.0600</td>
<td>0.918</td>
<td>31.91</td>
<td>~15</td>
<td>~10</td>
</tr>
<tr>
<td>Synthetic Hotdog</td>
<td>0.0183</td>
<td>0.955</td>
<td>29.94</td>
<td>0.0211</td>
<td>0.942</td>
<td>26.85</td>
<td>~18</td>
<td>~5</td>
</tr>
<tr>
<td>Ours</td>
<td>0.0124</td>
<td>0.975</td>
<td>35.01</td>
<td>0.0230</td>
<td>0.903</td>
<td>34.28</td>
<td>~3</td>
<td>~1</td>
</tr>
</tbody>
</table>

GTIRON Ours

Table 1: Comparison between PhySG [Zhang et al., 2021b], NeRF [Mildenhall et al., 2020], IDR [Yariv et al., 2020] and our method. #train and #test denote the size of training set and test set, respectively. Training time is in hours and rendering time is in seconds.

**Figure 7: Ablation study on the number of input images. We show a novel view synthesis result of the reconstruction for each object, given a different number of input images, whose viewpoints are evenly distributed inside the upper hemisphere. We recommend that readers zoom in to see the details of the picture, especially the folds at the top and the specular highlights on the face.**

**Figure 8:** Comparison between our method and IRON [Zhang et al., 2022]. The first and third rows are point lighting, and the second and fourth rows are environment lighting. The numbers above each result show the LPIPS, SSIM and PSNR in 512×512 pictures, respectively.

**Figure 9:** Comparison between our method and IRON [Zhang et al., 2022]. The first and third rows are point lighting, and the second and fourth rows are environment lighting. The numbers above each result show the LPIPS, SSIM and PSNR in 512×512 pictures, respectively.
5 Conclusion

We introduce a novel efficient hybrid differentiable rendering method to reconstruct triangular object mesh and PBR material from multi-view real-world images with uncontrolled lighting conditions. Unlike prior works that require massive resource consumption or approximated rendering process, we utilize an approximate method to optimize the geometry and a physically-based method to estimate the reflectance so that we could benefit from both the efficiency of the former and the high quality of the latter. In general, our method can handle a wide range of real-world scenes, providing an attractive and efficient solution and enabling photo-realistic novel view synthesis and relighting applications.

Limitations and future work. Our method can have difficulties with very thin geometry, which is a common problem of mesh-based methods. In addition, our method optimizes geometry and material separately. A potential future work is a unified pipeline to optimize geometry and material simultaneously, which should further improve the result quality.

Acknowledgments

This work was supported by the NSFC under Grant 62072271.

References


