Photometric Stereo Using Gaussian Splatting And Inverse Rendering

Matéo Ducastel, David Tschumperlé, Yvain Quéau

PHOTOMETRIC STEREO USING GAUSSIAN SPLATTING AND INVERSE RENDERING

Matéo Ducastel, Yvain Quéau, David Tschumperlé

Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC, Caen, France

ABSTRACT

Recent state-of-the-art algorithms in photometric stereo rely on neural networks and operate either through prior learning or inverse rendering optimization. Here, we revisit the problem of calibrated photometric stereo by leveraging recent advances in 3D inverse rendering using the *Gaussian Splatting* formalism. This allows us to parameterize the 3D scene to be reconstructed and optimize it in a more interpretable manner. Through a proof-of-concept on the Lambertian reflectance case, our approach demonstrates the potential of the Gaussian Splatting rendering engine for the single view photometric stereo problem by accurately estimating the shape of the object.

***Index Terms*—** Photometric Stereo, Shape-from-X, 3D Reconstruction, Gaussian Splatting, Inverse Rendering, Normal Map Estimation

1. INTRODUCTION

One of the goals of photometric stereo [1] is to estimate a 3D normal map describing the surface of an object from several photographs taken with a fixed camera but under different illumination angles. In the present work, we propose to explore a new approach to solve this problem, based on the recent formalism of *Gaussian Splatting* (GS) [2]. This formalism consists of modeling a scene as a field of Gaussians in a 3D space, to which different attributes are attached (notably orientation, size, and color). A dedicated rendering engine then allows for the rapid regeneration of a scene rendering from any viewpoint. What is remarkable is that this engine is differentiable, and can therefore be used in classical optimization methods (such as gradient descent) to estimate the parameters of the Gaussians so that the renderings of the resynthesized scenes are as close as possible to the input images (solving an inverse problem). We therefore propose to revisit the problem of photometric stereo in light of this formalism. The key insight is that Gaussian Splatting's differentiable rendering pipeline allows for end-to-end optimization of an explicit 3D scene representation, offering a compelling alternative to deep learning: Gaussian splatting is both lighter and more interpretable than optimizing black-box neural networks.

2. RELATED WORKS

A major advantage of photometric stereo is its ability to reconstruct fine details of imaged scenes, with applications in 3D digitization of cultural heritage artifacts [3] or defect detection on machined parts [4]. The historical approach to photometric stereo is that of Woodham [1], which defines a model to estimate the normal at each point of an object from several observations under varying illumination. This model assumes that the object has a Lambertian reflectance, so its appearance depends only on the angle between the normal and the light direction:

$$I_{(x,y)} \propto l^T \cdot n_{(x,y)}, \tag{1}$$

where I is the grayscale image, $l \in \mathbb{S}^2$ is the light direction, and $n_{(x,y)} \in \mathbb{S}^2$ is the normal of the observed object at each point (x, y) in the image. This model can be inverted using a least-squares optimization, but this has several limitations, notably the inability to handle cast shadows and specular reflections, which makes it impractical for complex materials and scenes. Following this pioneering work, most subsequent model-driven methods [5, 6, 7] have treated cast shadows and specular reflections as outliers, while developing more complex and robust optimization approaches to stabilize convergence. Such approaches following the inverse problem paradigm are usually computationally efficient, and interpretable by construction. Still, the scope of materials they can handle remains limited by the underlying model.

More recently, approaches using deep learning [8, 9, 10] have emerged to overcome this limitation. These approaches are often data-driven, enabling them to robustly handle challenging effects like specular highlights, cast shadows, and inter-reflections that typically cause classic algorithms to fail. Other works aim for more generality in terms of acquisition setup, such as the universal photometric stereo model by Ikehata et al. [11] and extensions thereof by Hardy et al. [12], which is designed to handle a wide variety of unknown illumination conditions. However, the performance of learning-based methods is highly dependent on the quality and diversity of the training dataset, which can limit their generalization capabilities to materials not seen during training. Most importantly, they require powerful machines to train and run which can limit their use in practice, and they lack the interpretability of physical model-driven methods, since the design of the underlying loss function is not guided

by physics.

Somewhere at the interface lie recent approaches leveraging neural networks within an inverse rendering framework featuring an explicit specular reflectance model. For instance, Li et al. [13] use several small, specialized neural networks to estimate intermediate parameters, such as light direction and the depth map. Their approach relies on a differentiable inverse rendering model, allowing the use of iterative optimization schemes to adjust network weights and minimize the difference between the reconstructed and observed images. A large part of the optimization is, however, dedicated to estimating cast shadows via ray tracing from each pixel, which significantly increases optimization time.

Gaussian Splatting has recently emerged as a promising alternative to neural networks for inverse rendering, as demonstrated by works like GS-IR [14] and Relightable 3D Gaussians [15], which decompose a scene into geometry, materials, and illumination from multiple views. Further extensions like IRGS [16] even tackle challenging global effects like inter-reflections. PS-GS by Chen et al. [17] leverages multi-view photometric stereo (MVPS) and GS to reconstruct full 3D objects using a cast shadow estimation with specular and diffuse reflection from Disney's BRDF [18]. These approaches, however, are fundamentally designed for multi-view reconstruction, distinguishing them from the single-view challenge posed by classical photometric stereo. Notably, multi-view methods such as PS-GS heavily rely on geometry initialization using Structure-from-Motion, which is unavailable in single-view PS. Besides, PS-GS still rely on traditional PS methods to estimate the normals used to regularize multi-view GS optimization. Instead, we question the feasability of end-to-end optimization of GS parameters within a PS model, without any supervision.

Our motivation for solving PS using Gaussian Splatting and inverse rendering is therefore threefold: providing a framework that is lightweight, in comparison with neural networks; setting up a flexible paradigm (the Lambertian model used in this proof of concept can easily be replaced by more advanced ones); and guarantee the interpretability of results (GS being easier to analyze than network weights). The rest of this paper aims at showing that the GS parameterization is a promising avenue for meeting these requirements.

3. PROPOSED FORMULATION

Gaussian Splatting. Our objective is to retrieve normal maps of objects with Lambertian materials by inverting the image formation model (1). To this end, we explore the Gaussian Splatting approach developed by Kerbl et al. [2], in order to leverage the advantages of associated differentiable inverse rendering methods. This technique, initially developed for multi-view 3D reconstruction, is based on a compact and efficient representation of the scene's geometry and appearance using fields of Gaussians placed in a 3D space.

Gaussian Splatting uses a set of spatially located Gaussians, each defined by several fundamental parameters (orientation, size, color). The rendering of a scene modeled in this way is then performed by projecting these Gaussians in 2D onto the final image, according to a viewing angle defined by the user. Each Gaussian is defined by a center $p_k \in \mathbb{R}^3$ and a covariance matrix Σ such that for any point $p \in \mathbb{R}^3$:

$$\mathcal{G}_k(p) = \exp\left(-\frac{1}{2}(p-p_k)^T\Sigma_k^{-1}(p-p_k)\right). \qquad (2)$$

The covariance matrix $\Sigma = RSS^TR^T$ is defined by a rotation matrix R and a scaling matrix S to ensure it is positive semi-definite.

To perform the splatting of the Gaussians at each pixel x of the image, one must first obtain the 3D point to evaluate on each Gaussian. For this, we retrieve the 3D coordinates of the projection of the center p_k onto the ray $r_x(t) = o + td_x$ cast from the camera position o in the viewing direction d_x. The position on the ray that is chosen is the one that minimizes the orthogonal distance to the center p_k, denoted t_{min}:

$$\widehat{\mathcal{G}}(x) = \mathcal{G}(r_x(t_{min})). \qquad (3)$$

A color is also assigned to each Gaussian and is defined using spherical harmonics, which allows the perceived color to vary dynamically with the viewing angle. Finally, an opacity is associated with each Gaussian to modulate its influence in the rendering process, controlling its relative importance in the blending of contributions through an alpha blending process by taking the k Gaussians from front to back until the opacity saturates at $\alpha = 1$. The color $c(x)$ of the pixel x is obtained by the following equation:

$$c(x) = \sum_{k=1}^{K} c_k\alpha_k\widehat{\mathcal{G}}_k(x)\prod_{j=1}^{k-1}(1-\alpha_j\widehat{\mathcal{G}}_j(x)), \qquad (4)$$

where α_k is the opacity parameter and c_k is the color for the given viewpoint. A key advantage of *Gaussian Splatting* is that the entire process is differentiable almost everywhere, allowing the parameters of the Gaussian models to be optimized by gradient descent. This property makes the method particularly suitable for approaches based on machine learning and scene parameter optimization. Typically, optimization is carried out using an L_1 loss function that measures the difference between the rendered image and a reference one.

In addition to the *splatting* performed for colors, these steps can be reproduced to obtain a depth map. From this depth map, it is naturally possible to estimate a normal map by calculating its gradient. However, this map will not necessarily be linked to the object's geometry because the volumetric nature of 3D Gaussians conflicts with the representation of surfaces. To solve this problem, Huang et al. [19] propose an extension of GS that constrains Gaussians to a parameterized

tangent space such that the point P on the plane of the 2D Gaussian with coordinates (u, v) satisfies the equation:

$$P(u,v) = p_k + s_u t_u u + s_v t_v v = H(u,v,1,1)^T, \quad (5)$$

$$\text{with } H = \begin{bmatrix} RS & p_k \\ 0 & 1 \end{bmatrix}, \quad (6)$$

where $p_k \in \mathbb{R}^3$ is still the center of the Gaussian in 3D space, $R = [t_u, t_v, t_w]$ is the rotation matrix defined by the two tangent vectors t_u and t_v and the orthogonal vector $t_w = t_u \times t_v$ to the tangent plane, and $S = (s_u, s_v)$ is the scaling matrix where the last component is 0 to form a 2D Gaussian. The point considered is the one on the Gaussian's plane. Thanks to this constraint, it becomes possible to obtain a normal map by performing a *splatting* of the normals in the same way as for the color. This is a crucial advantage over standard 3DGS, where normals can only be derived indirectly from depth gradients, a process that often results in noisy normals. The explicit normals of 2DGS provide a more stable geometric foundation for our inverse rendering task. We have therefore chosen to adapt the general idea of this work to solve the photometric stereo problem, by modeling and exploiting the normal map to obtain the final rendering.

Initialization. Normally, the initialization of Gaussian Splatting is done by Structure-from-Motion, which allows obtaining a sparse point cloud from several viewpoints of the same scene. However, in PS only one viewpoint is available. Our experiments have shown that GS can smoothly be incorporated into PS, even in the absence of such precise initialization. In our method, we simply place a Gaussian for each pixel on a plane with coordinates $z = 0$ and orient the Gaussians towards the camera placed along the Z axis. The color associated with each Gaussian is set as the maximum color observed across all images I_i under varying illumination:

$$c(x,y) = \max\{I_1(x,y), I_2(x,y), \ldots, I_n(x,y)\}. \quad (7)$$

This provides a reasonable, shadow-free upper bound for the object's albedo, which we use as initial color estimate.

Rendering. Given the GS parameters, it is possible render a normal map N and an albedo map A, by splatting their respective attributes. In the original Gaussian Splatting model, illumination is implicitly captured by spherical harmonics. To incorporate explicit lighting for photometric stereo, we render the final image I'_i for a given light direction l_i at pixel coordinate (x, y) using the Lambertian reflectance model:

$$I'_i = A(x,y) \odot \max(0, l_i^T \cdot N(x,y)) \quad (8)$$

where $A(x, y)$ is the rendered albedo map, $N(x, y)$ is the rendered normal map, the $\odot$ operator denotes element-wise multiplication, and the $\max(0, \cdot)$ term encodes self-shadows where the surface normal faces away from the light source.

Optimization. Once the model has been set, optimization is carried out to refine the set of parameters of the Gaussians (position, covariance matrix, color, and opacity). As mentioned earlier, the rendering engine exclusively uses Gaussians to represent the scene, which allows for efficient calculation of gradients to optimize them via gradient descent.
The loss function $\mathcal{L}$ we used consists of two terms. First, a photometric loss term measures the difference between the rendered image and the real image using an L_1 loss function. Second, a normal regularization term is added to ensure that the Gaussians are aligned with the true surface of the object. We compute a normal map using the gradient of the depth map by calculating the 3D coordinates of the points, then we obtain the derivatives d_x and d_y of the points in the x and y directions. We use the cross product to get the orthogonal direction and then normalize it:

$$N_D = \frac{d_x \times d_y}{||d_x \times d_y||}, \quad (9)$$

and then compare with the normals obtained by splatting in the L_1 norm sense, yielding the following loss function:

$$\mathcal{L} = \underbrace{\sum_i \|I'_i - I_i\|_1}_{\mathcal{L}_c \text{ (photometric)}} + \lambda \underbrace{\|N_D - N_S\|_1}_{\mathcal{L}_n \text{ (regularisation)}} \quad (10)$$

with I'_i and I_i the rendered and input image for the i-th light source, respectively, and N_S the normal obtained by splatting the explicit 2D Gaussian normals. Therein, all Gaussians' parameters are involved in the photometric loss, while regularization involves only the covariance matrices, positions and opacities. Let us also remark that the number of Gaussians will also evolve during optimization. In particular, Gaussians that are too large or have too low opacity will be removed. They can also be cloned or split when more detail is needed. The details of this process are described in [2].

4. RESULTS

Our GS-based PS method was tested first on synthetic images comprising simple geometric shapes, rendered using the *Blender* [20] rendering software. In these simple cases, our algorithm finds coherent normals (Figure 1).
Second, we evaluated our methods on real data from the *DiliGenT* dataset [21], specifically working on the *Bear*, *Cat*, and *Pot1* objects, which are objects from the dataset with Lambertian materials (Figure 2). The angular error is close to zero, except in regions where a cast shadow is present in a large number of input images. This is because our model does not yet account for this type of shadow, which is the most complex to estimate due to its global nature. We illustrate this phenomenon in Figure 3, where the cat's left hind leg is hidden by the cast shadow of the right front paw. Specularities are another source of errors since they are neglected in our model, as illustrated in Figure 4.

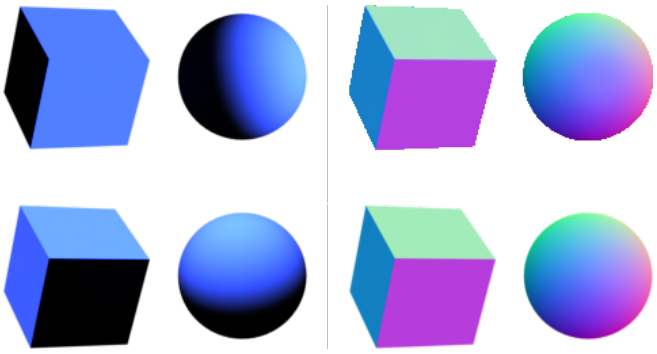

Fig. 1: Examples of two object renderings via Blender (left, the two rows account for different light directions), estimated normals (top right, after optimization) and ground truth (bottom right).

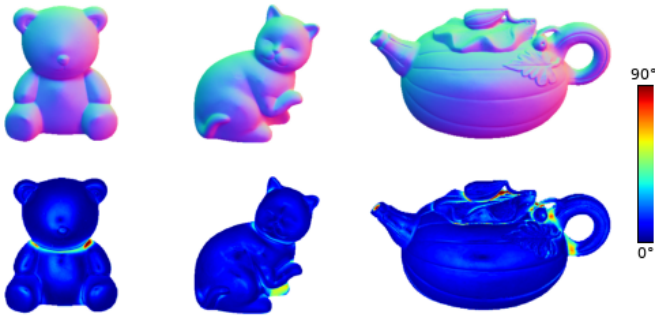

Fig. 2: Estimated normals and angular error map for the *Bear*, *Cat*, and *Pot1* objects from the *DiliGenT* dataset. The normal orientations are correctly estimated, except in cast shadow regions.

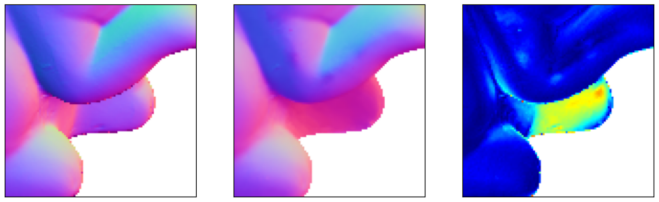

Fig. 3: Ground truth normals, estimated normals and angular error map. High errors are concentrated on the cast shadows areas.

We also quantitatively compared in Table 1 our results against state-of-the-art PS methods, notably a neural network-based inverse rendering one [13] and a deep learning one [12], using the mean angular error (in degrees). Our results are in par with the baseline [1], which was to be expected since we consider the exact same image formation model (only the parameterization differs). The performance gap with state-of-the-art neural methods is primarily due to our current model's simplified assumptions (strict Lambertian reflectance, absence of a cast shadow model).

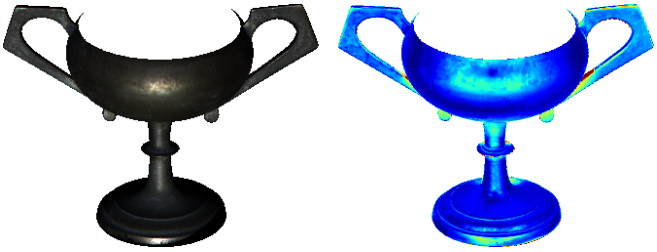

Fig. 4: Example image of a specular object (left), and angular error map. Specularities induce high estimation errors, since they are not accounted by the model yet.

	Cat	Bear	Pot1
Baseline [1]	8.41	8.39	8.89
DANINet [13]	4.73	4.11	6.41
Uni MS-PS [12]	3.45	3.14	4.12
GS-PS (Ours)	8.26	8.13	10.11

Table 1: Performance on the DiliGenT test set, measured by the mean angular error (in degrees).

However, our approach offers significant advantages over neural networks. Unlike learning-based methods that require extensive, varied datasets for training and may fail on out-of-distribution materials, our method is optimized per-scene and requires no prior training. Besides, the parameters being optimized (position, orientation, scale of Gaussians) have a direct physical and geometric interpretation, making the model highly explainable compared to the opaque internal weights of a neural network.

5. CONCLUSIONS AND FUTURE WORK

This work has demonstrated the feasability of the Gaussian Splatting paradigm for solving single-view photometric stereo, yielding a computationally efficient and interpretable framework. In the future, we will first seek to refine our model to handle non-Lambertian materials, taking inspiration from GS rendering engines that handle specular reflections [22, 23] using more complex BRDF models like Disney's [18]. In addition, we will account for cast shadows, using Shadow Mapping [24]. This involves rendering a depth map from the light's point of view, then comparing this rendering with the coordinates of the points obtained via the depth map generated from the camera's point of view. Shadow mapping has recently been used in the multi-view GS context by Savant Aira et al. [25]: adapting such an approach to single view PS seems a promising direction for future research.

6. REFERENCES

[1] Robert J. Woodham, "Photometric Method For Determining Surface Orientation From Multiple Images," *OptEng*, vol. 19, no. 1, pp. 191139, 1980.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, "3D Gaussian Splatting for real-time radiance field rendering," *ToG*, 2023.

[3] Antoine Laurent, Benjamin Coupry, Baptiste Brument, Jean Mélou, Yvain Quéau, Carole Fritz, and Jean-Denis Durou, "Combining geometric and photometric 3d reconstruction techniques for cultural heritage," *JCH*, vol. 73, 2025.

[4] Shun Wang, Ke Xu, Baohua Li, and Xiangyu Cao, "Online micro defects detection for ductile cast iron pipes based on twin light photometric stereo," *Case Studies in Construction Materials*, vol. 19, 2023.

[5] Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma, "Robust photometric stereo via low-rank matrix completion and recovery," in *ACCV*, 2010.

[6] Satoshi Ikehata, David Wipf, Yasuyuki Matsushita, and Kiyoharu Aizawa, "Robust photometric stereo using sparse regression," in *CVPR*, 2012.

[7] Yvain Queau, Tao Wu, Francois Lauze, Jean-Denis Durou, and Daniel Cremers, "A non-convex variational approach to photometric stereo under inaccurate lighting," in *CVPR*, 2017.

[8] Yakun Ju, Kin-Man Lam, Wuyuan Xie, Huiyu Zhou, Junyu Dong, and Boxin Shi, "Deep learning methods for calibrated photometric stereo and beyond," *TPAMI*, 2024.

[9] Xiaoyao Wei, Zongrui Li, Binjie Ding, Boxin Shi, Xudong Jiang, Gang Pan, Yanlong Cao, and Qian Zheng, "Revisiting supervised learning-based photometric stereo networks," *TPAMI*, 2025.

[10] Satoshi Ikehata, "Ps-transformer: Learning sparse photometric stereo network using self-attention mechanism," in *BMVC*, 2021.

[11] Satoshi Ikehata, "Scalable, detailed and mask-free universal photometric stereo," in *CVPR*, 2023.

[12] Clément Hardy, Yvain Quéau, and David Tschumperlé, "Uni MS-PS: A multi-scale encoder-decoder transformer for universal photometric stereo," *CVIU*, vol. 248, 2024.

[13] Zongrui Li, Qian Zheng, Boxin Shi, Gang Pan, and Xudong Jiang, "DANI-Net: Uncalibrated photometric stereo by differentiable shadow handling, anisotropic reflectance modeling, and neural inverse rendering," in *CVPR*, 2023.

[14] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia, "Gs-ir: 3d gaussian splatting for inverse rendering," in *CVPR*, 2024.

[15] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao, "Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing," *ECCV*, 2024.

[16] Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang, "Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing," in *CVPR*, 2025.

[17] Yijun Chen, Bowen Liang, Hong Guo, Yuchen Cheng, Jianzhuang Zhao, and Ruofeng Tong, "PS-GS: Gaussian splatting for multi-view photometric stereo," 2024.

[18] Brent Burley and Walt Disney Animation Studios, "Physically-based shading at disney," in *SIGGRAPH*, 2012.

[19] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao, "2D Gaussian Splatting for geometrically accurate radiance fields," in *SIGGRAPH*, 2024.

[20] Blender-Foundation, *Blender - a 3D modelling and rendering package*.

[21] Boxin Shi, Zhe Wu, Zhipeng Mo, Dinglong Duan, Sai-Kit Yeung, and Ping Tan, "A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo," in *CVPR*, 2016.

[22] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma, "Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces," in *CVPR*, 2024.

[23] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia, "GS-IR: 3d gaussian splatting for inverse rendering," in *CVPR*, 2024.

[24] Lance Williams, "Casting curved shadows on curved surfaces," *SIGGRAPH Comput. Graph.*, vol. 12, no. 3, 1978.

[25] Luca Savant Aira, Gabriele Facciolo, and Thibaud Ehret, "Gaussian splatting for efficient satellite image photogrammetry," in *CVPR*, 2025.