Refractive-Aware Gaussian Splatting for Geometrically Accurate and Photorealistic 3D Reconstruction of Bathymetry from UAV-based Imagery

Taiki Uno[1*] and Sohei Kobayashi[2]

[1]Kyoto University, Graduate School of Engineering, Kyoto, Japan
[2]Kyoto University, Disaster Prevention Research Institute, Uji, Japan
[*]uno.taiki.65r@st.kyoto-u.ac.jp

Abstract *The 3D reconstruction of geometry in shallow water has a wide range of applications, such as monitoring bed morphology changes and conducting hazard simulations. In addition, the detailed texture of the reconstructed data is useful for bed surface classification and aquatic habitat quantification. In this context, bathymetric surveying using photogrammetry from aerial images captured by Unmanned Aerial Vehicles (UAVs) is gaining attention as a particularly efficient method. However, it is fundamentally challenged by light refraction at the air-water interface, which invalidates the geometric principles of photogrammetry. Existing methods either rely on iterative post-processing or employ deep learning models that lack physical guarantees and explainability. We solve this challenge by introducing a refraction-aware 3D Gaussian Splatting framework that incorporates an optically accurate model of two-media refraction directly into the reconstruction pipeline. Our key innovation is a differentiable coordinate transformation that analytically models light refraction, mapping 3D Gaussians from their true underwater positions to their apparent space for each aerial view. This enables unified optimization, simultaneously solving for dense scene geometry and detailed appearance while maintaining the efficiency of standard 3D Gaussian Splatting. We evaluated our method on a simulated UAV dataset of a riverbed, rendered with physically-based ray tracing to isolate refractive effects from other optical phenomena. Our approach achieved a geometric F1-score of 96% (with a 10 cm error threshold at a depth scale of 10 m). Furthermore, in novel view synthesis, we obtained photorealistic views with a Peak Signal-to-Noise Ratio (PSNR) of 25.9 dB and a Structural Similarity Index Measure (SSIM) of 0.93. By creating 3D models that are both photorealistic in appearance and dense and geometrically precise in structure, our method addresses a key challenge in aquatic remote sensing from aerial imagery. This work enables cost-effective, high-frequency monitoring of riverbeds, lakeshores, and seashores under calm surface conditions.*

Keywords: *Bathymetry, Gaussian Splatting, 3D Reconstruction, Refraction Correction, UAV Imagery*

Introduction

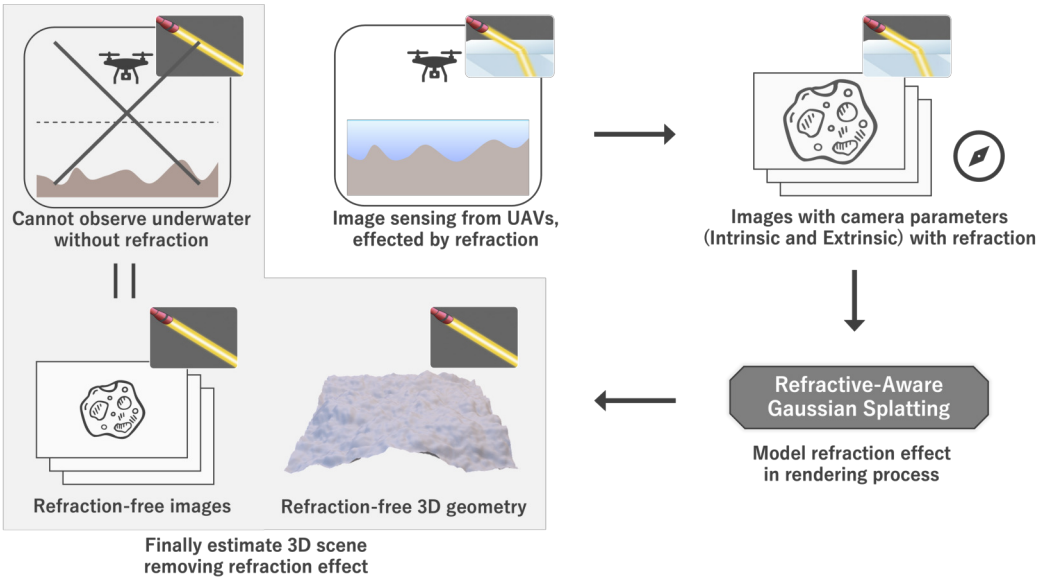

Figure 1. Overview of our proposed method for photogrammetric bathymetry. Our Refractive-Aware Gaussian Splatting takes multi-view images distorted by refraction (top right) as input. By explicitly modeling refraction within the rendering pipeline, it reconstructs a geometrically accurate and refraction-free 3D scene (bottom left).

Bathymetry is critically important. Accurate topographic information serves as fundamental data for river management, including flood prediction and monitoring geomorphological changes, and it also has potential applications in marine environments as well. However, sonar-based hydrographic surveying, which is widely used from vessels, faces significant limitations in shallow waters due to the risk of grounding and equipment damage, and narrow and inefficient survey width. More recently, Airborne Light Detection and Ranging Bathymetry (ALB) has been employed, but both the equipment and data acquisition costs remain prohibitively high. On land, UAV-based photogrammetry has become a widely adopted technique, offering an efficient, inexpensive, safe, and accessible solution. Nonetheless, when applied to underwater targets from aerial platforms, photogrammetry encounters fundamental challenges. These include specular reflections from the water surface, distortions by waves, and light scattering and attenuation within the water medium. Among these factors, optical refraction is particularly detrimental, as it fundamentally breaks the assumption that light travels in a straight line, which is the foundation of photogrammetry.

A variety of methods have been proposed to mitigate refraction effects, but most rely on post-hoc or iterative corrections applied to the outputs of conventional photogrammetric pipelines. Since they lack explicit physical modeling of refraction, such approaches fail to generalize

to scenarios involving oblique or close-range imaging, leaving the core problem unresolved. Recently, refraction-aware structure-from-motion (R-SfM) methods have emerged (Makris et al., 2024), explicitly incorporating refraction into the SfM pipeline. Although promising, these methods typically yield only sparse point clouds and rely on deep learning–based densification techniques trained on ALB-derived datasets (Agrafiotis et al., 2019). Given the scarcity of high-quality training data, applying these approaches across diverse regions remains infeasible.

Concurrently, 3D Gaussian Splatting (3DGS) has recently gained significant attention as an emerging paradigm for 3D reconstruction (Kerbl et al., 2023). Building on this advancement, we propose a novel foundation for photogrammetric bathymetry: Refractive-Aware Gaussian Splatting, which directly integrates an optically accurate two-media refraction model into the Gaussian Splatting pipeline. Our methods not only enables the recovery of geometrically accurate bathymetry but also provides photorealistic renderings for a clear appearance with refraction artifacts effectively removed. The key idea is to incorporate a differentiable parameter transformation that adjusts each Gaussian's parameters with respect to the camera ray according to appearance depth, embedding this directly into the optimization process of Gaussian Splatting.

Under the assumption of a calm, planar water surface without reflection and attenuation underwater, Our method reconstructs accurate 3D Gaussians free of refraction effect from multi-view images with known camera poses and refractive surface position, initialized with only a coarse prior of the depth of the floor. On multiple synthetic CG datasets, our method achieved image quality metrics exceeding PSNR 25 and demonstrated shape reconstruction with F1-scores above 90% after 3DGS to point cloud conversion. This establishes a foundation for a next generation of photogrammetric bathymetry.

Literature Review

Bathymetry in shallow water area has been explored through a variety of methodologies (He et al., 2024). Traditional fields measurements methods with bathymetric rods or hammers, and global positioning systems (GPS) are extremely time-consuming and dangerous. Shipborne sonar systems are a representative method of bathymetry, however, the operation is restricted due to its narrow scanning swath. Therefore, remote sensing that reconstructs underwater morphology from data observed in the air is valuable in this field. As remote sensing for shallow bathymetry, platforms equipped with sensors are generally classified into satellite, airborne, and, UAV-based systems. The sensors employed range from RGB cameras and multi/hyperspectral cameras, to LiDAR and even Synthetic Aperture Radar (SAR). As an efficient, inexpensive, safe, and accessible solution, Our work focuses on methods based on UAV-based RGB imagery.

To reconstruct 3D information from UAV imagery, previous works have employed geometric approaches such as Structure-from-Motion (SfM) (Schönberger & Frahm, 2016) or Multi-View-Stereo (MVS) (Furukawa & Hernández, 2015; Furukawa & Ponce, 2010), which have been developed in the computer vision community. In a typical incremental SfM pipeline, keypoints are firstly detected and matched across frames using feature detectors and descriptors such as SIFT (Lowe, 2004). Fundamental matrix F between two images is then calculated, commonly via eight-points algorithm (Hartley, 1997) combined with random sample consensus (RANSAC) (Fischler & Bolles, 1987), which lead to camera pose recovery through singular value decomposition. New camera poses are iteratively registered via Perspective-n-Point algorithms, which align estimated 3D points with 2D features in new frames. Triangulation is subsequently applied to obtain additional 3D points from feature correspondences, and bundle adjustment (BA) (Triggs et al., 2000) is finally performed to minimize reprojection error, refining both camera parameters and 3D points. MVS builds upon the calibrated cameras estimated from SfM, computing dense per-pixel depth and normal maps across views, which are then fused into a dense 3D point cloud or mesh.

This entire pipeline assumes straight-line traveling of light, while in reality, refraction at water-air interface significantly degrades both SfM and MVS reconstructions. This refractive distortion represents the key challenge in photogrammetric bathymetry. Woodget et al., 2014 demonstrated the potential of photogrammetric bathymetry from UAV imagery, applying a

simple refraction correction by multiplying refractive index to SfM software outputs. Dietrich, 2016 proposed multi-angle refraction correction based on the pixel ray angle and estimating 3D points, which consider the view-angle dependence of each apparent points. However, this method only corrects for vertical displacement, ignoring horizontal distortions. Makris et al., 2024 successfully integrated refraction modeling directly into the SfM pipeline, avoiding post-hoc or iterative correction. This R-SfM method enables highly accurate camera pose estimation and point clouds generation. They combine Deep Learning method (Alevizos et al., 2022) to compensate the sparseness of SfM output, however, Deep Learning based method needs extensive training datasets and remain highly site-dependent.

Preliminaries

a. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) is a recent method that has achieved state-of-the-art results in novel view synthesis (Kerbl et al., 2023). It is distinguished by its remarkable capability for high-fidelity 3D scene capturing with photorealism, rapid training times, and real-time rendering. As an explicit 3D representation, 3DGS has been successfully applied to a wide range of tasks, including Visual-SLAM (Matsuki et al., 2024; Yan et al., 2024; Zheng et al., 2025), human avatar creation (Moreau et al., 2024; Shao et al., 2024), and feedforward 3D reconstruction (Chen et al., 2024). Its potential further extends to various real-world applications such as digital surface model (DSM) generation from satellite imagery (Aira et al., 2025), UAV-based surveying, autonomous driving, and underwater 3D reconstruction (Li et al., 2025).

The 3DGS pipeline consists of two primary stages: a forward pass for rendering and a backward pass for optimization. In the forward pass, a collection of 3D Gaussians is rasterized to synthesize an image. Each Gaussian is defined by a set of optimizable parameters: a center position $\boldsymbol{p} \in \mathbb{R}^3$, an opacity $\alpha \in [0, 1]$, view-dependent color coefficients represented by Spherical Harmonics (SH) $\boldsymbol{c}(\boldsymbol{p}, \boldsymbol{t}_i) \in \mathbb{R}^3$, and a 3D covariance matrix $\boldsymbol{\Sigma}^{3D} \in \mathbb{R}^{3\times3}$. The covariance matrix $\boldsymbol{\Sigma}^{3D}$ is composed by a scaling diagonal matrix $\boldsymbol{S} \in \mathbb{R}^{3\times3}$ composed of a scaling vector $\boldsymbol{s} \in \mathbb{R}^3$ and a rotation quaternion (represented as a rotation matrix $\boldsymbol{R} \in SO(3)$) as:

$$\boldsymbol{\Sigma}^{3D} = \boldsymbol{R}\boldsymbol{s}\boldsymbol{s}^\top\boldsymbol{R}^\top \tag{1}$$

The corresponding unnormalized Gaussian distribution function for a point $\boldsymbol{x} \in \mathbb{R}^3$ is given by:

$$G(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{p})^T \left(\boldsymbol{\Sigma}^{3D}\right)^{-1}(\boldsymbol{x}-\boldsymbol{p})\right) \tag{2}$$

During the forward pass, to render an image from a given camera view, these Gaussians are first transformed from world to camera coordinates using the extrinsic matrix $[\boldsymbol{W}|\boldsymbol{t}]$ where $\boldsymbol{R}_{\text{view}} \in \mathbb{R}^{3\times3}$ is a viewing rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector. The position of a Gaussian center $\boldsymbol{p}$ and 3D covariance matrix are updated as:

$$\boldsymbol{p}_{\text{cam}} = \boldsymbol{R}_{\text{view}}\boldsymbol{p} + \boldsymbol{t}$$
$$\boldsymbol{\Sigma}^{3D}_{\text{cam}} = \boldsymbol{R}_{\text{view}}\boldsymbol{\Sigma}^{3D}\boldsymbol{R}^{\top}_{\text{view}}$$

Following the projection method from (Zwicker et al., 2001), the 3D covariance in camera space, Σ^{3D}_{cam} ,is then projected onto the 2D image plane. This is achieved using the Jacobian $\boldsymbol{J}$ of the affine approximation of the perspective projection, yielding a 2D covariance matrix $\boldsymbol{\Sigma}^{\text{2D}}$:

$$\Sigma^{\text{2D}} = \boldsymbol{J}\Sigma^{\text{3D}}_{\text{cam}}\boldsymbol{J}^{\top} \tag{3}$$

The final RGB value $\boldsymbol{\Gamma} \in \mathbb{R}^3$ for each pixel is rendered by alpha-blending the projected Gaussians. The set of Gaussians that overlap with the pixel are first sorted front-to-back based on their depth, and the view-dependent color is then accumulated as:

$$\boldsymbol{\Gamma}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{c}_k \alpha_k^{\text{pixel}} \prod_{j=1}^{k-1}(1-\alpha_j^{\text{pixel}}) \tag{4}$$
$$\text{where} \quad \alpha_k^{\text{pixel}} = \alpha_k G_k^{\text{2D}}$$

here k is the ordered set of Gaussians overlapping the pixel.

During the backward pass, the optimization minimizes a photometric loss, which is a weighted sum of a $\mathcal{L}_1(\Gamma, \Gamma_{gt})$ Loss and a D-SSIM loss $\mathcal{L}_{\text{D-SSIM}}(\Gamma, \Gamma_{gt})$ (Wang et al., 2004):

$$\mathcal{L} = (1-\lambda)\mathcal{L}1 + \lambda\mathcal{L}_{\text{D-SSIM}} \tag{5}$$

Thus, the optimization problem is formulated as follows:

$$\underset{p,R,s,c,\alpha}{\text{argmin}} \quad \mathcal{L} = \mathcal{L}(\mathbf{\Gamma}, \mathbf{\Gamma}_{\boldsymbol{gt}}) \tag{6}$$

These formulations ensure that the entire pipeline is fully differentiable, allowing the parameters $\{\boldsymbol{p}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{c}, \alpha\}$ to be optimized via gradient descent using the Adam optimizer (Kingma & Ba, 2014).

The collection of 3D Gaussians obtained through this process captures the 3D scene with high fidelity. However, this entire pipeline relies on the pinhole camera model and perspective projection, which fundamentally assume that light travels in a straight line. This assumption is violated in environments with multi-medium interfaces, such as where refraction at the air-water boundary causes severe geometric inconsistency, leading to a failure in the reconstruction.

Despite this limitation, 3DGS is uniquely suited for addressing this challenge compared to implicit representations like NeRF. This is because the explicit nature of the Gaussian primitives that is highly open to direct physical modeling. It allows us to mathematically formulate and apply the laws of refraction directly to the geometric parameters of the scene representation itself.

Methodology

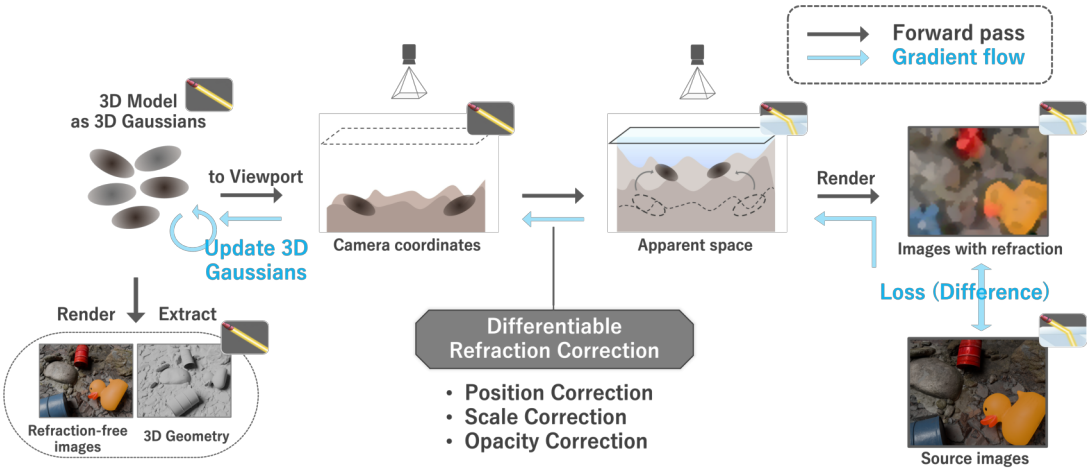

Figure 2. Overview of Refractive-aware Gaussian Splatting. This figure illustrates our proposed pipeline. A 3D scene, represented by 3D Gaussians, is transformed into an ”apparent space” for each camera view by applying a differentiable refraction correction, which consists of position, scale, and opacity correction. This transformation models the physical effects of light refraction at the air-water interface. The resulting Gaussians are rendered to produce an image that includes refractive distortions. The core of our method is a fully differentiable process. A loss is computed by comparing the rendered image with the input source image (which also contains refraction). This loss is then backpropagated to optimize the parameters of the original 3D Gaussians. Through this optimization, the model learns a refraction-free 3D representation of the scene, enabling the generation of both photorealistic, undistorted novel views and a geometrically accurate 3D model.

As illustrated in Fig. 2, our method explicitly models refraction by applying a corrective transformation to the parameters of each Gaussian primitive. This transformation accounts for the refractive effect of bending light rays at the air-water interface. This refractive transformation is composed of three components: (a) position correction, (b)scale correction, and (c)opacity correction.

a. Position Correction

We assume a Cartesian coordinate system where the camera is positioned on the positive z-axis at $(0, 0, H)$, and the water surface corresponds to the xy-plane ($z = 0$). Following prior work, we assume the water surface is flat and calm, resulting in a constant surface normal $\mathbf{n} = (0, 0, 1)$. A 3D Gaussian primitive located underwater has a center position $\boldsymbol{p} = (x, y, z)$, where $z < 0$.

To simplify the geometry, we project the 3D position onto the 2D *rz-plane* using the radial distance $r = \sqrt{x^2 + y^2}$. The apparent position of the Gaussian's center, after accounting for refraction, is denoted as $\boldsymbol{p'} = (x', y', z')$, which projects to (r', z') on this plane.

From Snell's law and geometric optics, it follows that the apparent position of a submerged point, as seen from a fixed camera center, is uniquely determined(Missailidis & Guilhon, 2025; Nassar, 1994). The ray from the apparent position $\boldsymbol{p'}$ to the camera intersects the water surface at a point we define as $(s, 0)$ on the *rz-plane*, where $0 \leq s < r$. Let θ_r be the angle of incidence (in water) and θ_i be the angle of refraction (in air), both measured with respect to the surface normal. Let n be the refractive index of water, and we assume the refractive index of air is 1. Snell's Law gives:

$$\sin\theta_i = n \sin\theta_r \tag{7}$$

From the geometry of the system, these angles can be related to the positions, leading to a quartic equation in s (triangle similarity between $\triangle OAI$ and $\triangle IPC$):

$$\begin{aligned}(1-n^2)s^4 + 2(n^2-1)rs^3 \\ + ((1-n^2)r^2 + H^2 - n^2z^2)s^2 \\ + 2n^2z^2rs - n^2z^2r^2 = 0\end{aligned} \tag{8}$$

This equation can be solved numerically for s, for instance, using Newton's method or directly solved by Ferrari's method. To find the apparent position (r', z'), we analyze the differential change in ray position. From the geometry of triangles formed by the rays and positions, we can establish the following relationships. ($\triangle IPC$, $\triangle IP'C'$, and $\triangle I'P'C'$):

$$r - s = -z\tan\theta_r \tag{9}$$

$$r' - s = z'\tan\theta_i \tag{10}$$

$$\begin{aligned}-z'\tan(\theta_i + \Delta\theta_i) &= (r'-s) + (-\Delta s) \\ -z'(\sin\theta_i + \Delta\theta_i\cos\theta_i) &= (r'-s-\Delta s)(\cos\theta_i - \Delta\theta_i\sin\theta_i)\end{aligned} \tag{11}$$

With the derivative of Snell's law from Eq. (7):

$$\frac{d\theta_r}{d\theta_i} = \frac{1}{n}\frac{\cos\theta_i}{\cos\theta_r} \tag{12}$$

By differentiating these geometric relationships Eqs. (9) to (11) and combining them with equations of Snell's law Eqs. (7) and (12), we can represent the apparent position (r', z') with θ_i and

θ_r.

$$\begin{cases} r' = r + (n^2 - 1) \cdot z \cdot \tan^3 \theta_r & \text{(13a)} \\ z' = \dfrac{1}{n} \cdot \dfrac{\cos \theta_i{}^3}{\cos \theta_r{}^3} \cdot z & \text{(13b)} \end{cases}$$

This is the center position correction transformation. Since θ_i and θ_r are determined by the camera position and the Gaussian's mean position $\boldsymbol{p}$, the apparent position $\boldsymbol{p'}$ is uniquely determined. The transformation is applied by first calculating r and then finding the new coordinates $x' = x\frac{r'}{r}$ and $y' = y\frac{r'}{r}$. Note that since $z < 0$, the term $(n^2 - 1) \cdot z \cdot \tan^3 \theta_r$ is negative, ensuring $r' < r$. Similarly, since $\theta_i > \theta_r$ for $n > 1$, we have $\cos \theta_i < \cos \theta_r$, which ensures $|z'| < |z|$. This confirms that the apparent Gaussian $G'(\boldsymbol{p'})$ always appears shallower and closer to the central axis than the actual Gaussian $G(\boldsymbol{p})$. In the special case where the Gaussian is directly below the camera ($r = 0$, thus $\theta_i = \theta_r = 0$), the equations simplify to $r' = r$ and $z' = z/n$. This special case, where apparent depth is the true depth divided by n, is a common approximation used in prior work (e.g., Woodget et al., 2014). However, this approximation is inaccurate for most rays in a perspective projection which are off-nadir.

For use in Gaussian Splatting, this transformation must be differentiable. We therefore require the Jacobian of the transformation:

$$J_{app} = \frac{\partial \boldsymbol{p'}}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} & \frac{\partial x'}{\partial z} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} & \frac{\partial y'}{\partial z} \\ \frac{\partial z'}{\partial x} & \frac{\partial z'}{\partial y} & \frac{\partial z'}{\partial z} \end{bmatrix} \tag{14}$$

While this Jacobian can be derived analytically, we compute it using numerical differentiation in our implementation for simplicity. Specifically, we use finite differences; for example, the partial derivative $\partial x'/\partial x$ is approximated as $(p'_x(x + \Delta x) - p'_x(x - \Delta x))/(2\Delta x)$.

This correction is the key proposal of our paper, which models the effect of refraction through a direct analytical transformation of the Gaussian means. This enables the reconstruction of an accurate, refraction-free 3D scene from posed images of an underwater environment. The only required inputs are the refractive index of the water and the location of the water surface relative to the cameras.

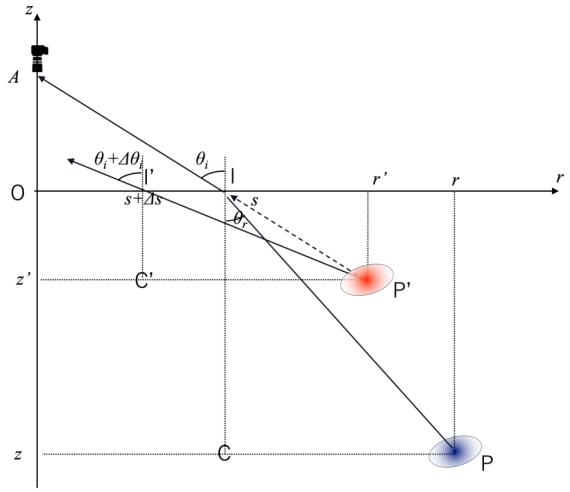

Figure 3. Illustration of the apparent position P' versus the true position P in the rz-plane. Even though the actual object is located at the blue Gaussian position P, it appears shifted toward the red Gaussian position P' when observed through the refractive interface.

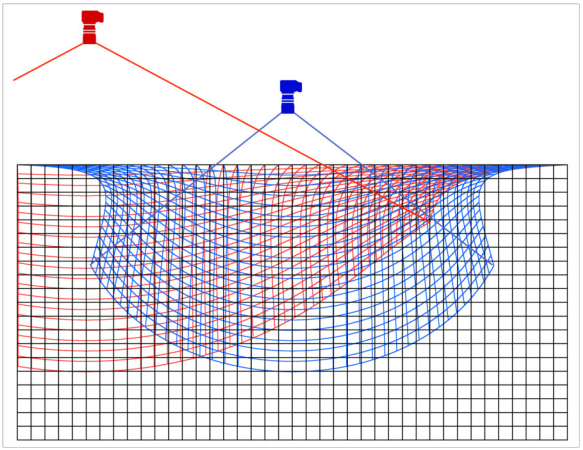

Figure 4. Space compression from the camera view. The Space is distorted by the camera position. Each colored grid means the corresponding compressed space of entire grid in camera coordinates (shown as black.)

b. Scale Correction

The transformation of the Gaussians center position $\boldsymbol{p} \rightarrow \boldsymbol{p'}$, which models the core effect of refraction, distorts the Cartesian coordinates system of camera coordinates. This space is named *the apparent space*, where the axes are no longer orthogonal as illustrated in Fig. 4. This distortion is particularly pronounced for large angles of incidence (in the air), causing the space to be significantly compressed. Without a corresponding correction to the scale parameter $\boldsymbol{s}$, individual Gaussians are perceived as being both densely packed and dilated. This leads to severe visual artifacts, such as blurring and unrealistic expansion of surfaces, as shown in Fig. 5.

To mitigate these artifacts, we propose to downscale each Gaussian according to the local space compression ratio. We quantify this compression by analyzing the transformation of the Cartesian basis vectors. For instance, the basis vector $\mathbf{e}_x = (1, 0, 0)^\top$ is to be mapped by given by the first column of J_{app}:

$$\mathbf{e}'_x = J_{app}\mathbf{e}_x = \begin{bmatrix} \frac{\partial x'}{\partial x} \\ \frac{\partial y'}{\partial x} \\ \frac{\partial z'}{\partial x} \end{bmatrix}. \tag{15}$$

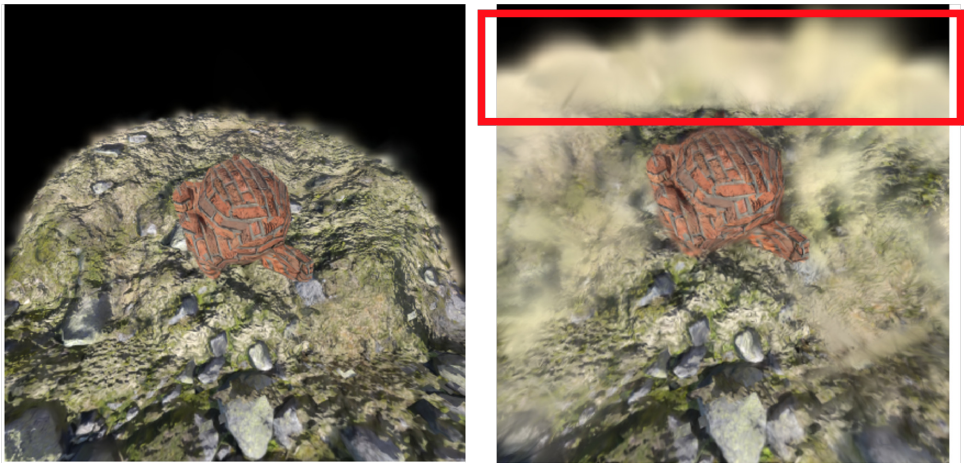

Figure 5. Artifact without scale correction.
Left: The image is rendered image of 3DGS model trained by Ground Truth (GT) images without refractive plane. **Right**: The image is rendered of same model using center position correction, but without scale correction. The scene in area bounded by the red is expanded, where the ray of incident angle is larger.

The local scaling factor along the x-axis is quantified by the norm of this vector:

$$s_x = \|\mathbf{e}'_x\| = \sqrt{\left(\frac{\partial x'}{\partial x}\right)^2 + \left(\frac{\partial y'}{\partial x}\right)^2 + \left(\frac{\partial z'}{\partial x}\right)^2}. \tag{16}$$

Similarly, we can compute the scaling factor s_y and s_z along the y-axis and z-axis.

It seems better in order to calculate the volume compression ratio V by applying to a determinant of Eq. (14) $|\det J_{app}|$. However, this value overestimates downscaling especially the ray of incident angle is larger, which makes Gaussians too small.

While one might consider using the determinant of the Jacobian $|\det J_{app}|$ as a direct measure of the volumetric compression, we empirically found that this approach overestimates the required downscaling, particularly for large incidence angles, resulting in overly diminished Gaussians. Instead, we define an isotropic scale correction factor S derived from the geometric mean of the directional scaling factors. This factor is computed as the cubic root of the product of the individual scaling factors:

$$V = s_x \cdot s_y \cdot s_z \tag{17}$$

This formulation provides a stable measure of the average spatial compression that is robust to the extreme shear in the transformation. While it intentionally disregards the shear components, it is precisely this property that makes it effective for isotropically rescaling the Gaussians,

preventing the visual artifacts without eliminating the primitives themselves.

$$S = V^{1/3} \tag{18}$$

We then apply this factor to the original scale vector $\boldsymbol{s}$ to obtain the corrected scale $\boldsymbol{s'}$

$$\boldsymbol{s'} = S\boldsymbol{s} \tag{19}$$

We apply this correction during the rendering phase, dynamically adjusting the scale of each Gaussian based on its position relative to the camera and the refractive interface.

c. Opacity Correction

In this section, we introduce the regularization that increases the opacity α of each Gaussians through the refractive surface. This is not a physically accurate regularization as refraction, however, it contributes to the fidelity of both appearance and geometry. As illustrated in Figure 6, observing a scene through a refractive interface causes a significant reduction in apparent radiance. This is a direct consequence of etendue conservation, which dictates that for a light bundle traveling from a medium with refractive index n (e.g., water) to air, the radiance is scaled:

$$L_{air} = \frac{1}{n^2} L_{water} \tag{20}$$

This radiometric principle poses a challenge for 3D Gaussian Splatting. To reconstruct the darker target images, the optimizer often converges to a trivial solution composed of Gaussians with low opacity α. This reliance on low-opacity primitives results in a geometrically ambiguous or "blurry" reconstruction, as the scene is represented by a sparse, semi-transparent Gaussians, failing to capture fine details. To counteract this, we introduce a regularization technique inspired by the radiometric relationship. We preemptively scale the opacity of underwater Gaussians to account for the expected radiance loss:

$$\alpha' = \frac{1}{n^2}\alpha \tag{21}$$

By intentionally reducing the opacity budget for each Gaussian, we force the optimizer to explain the target appearance by utilizing a larger number—and therefore a denser configura-

tion—of primitives. This process regularizes the optimization, inducing the model to learn a more detailed and geometrically faithful scene representation. While this is a heuristic, not a physically exact correction for refraction, it yields significant improvements in both final appearance and geometric fidelity.

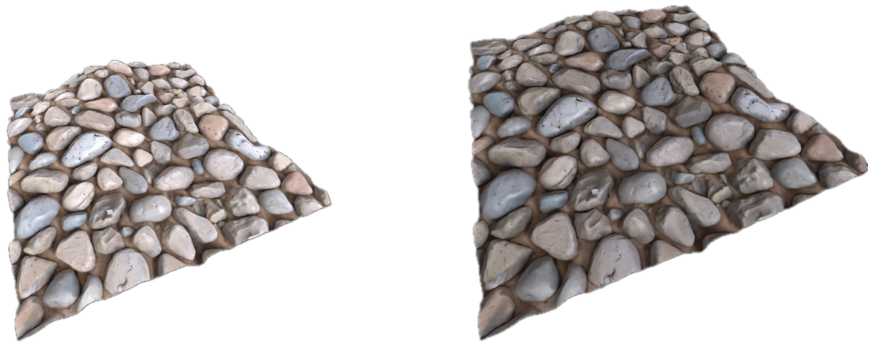

Figure 6. This is a scene rendered using the path tracer in Blender Cycles. The image on the right, viewed through a refractive water surface, is significantly darker than the one on the left (which is shown without refraction). Opacity correction serves as regularization to avoid ambiguities, such as blurriness or floaters.

Experiments

a. Dataset

To focus on the effects of refraction, we conduct our experiments in a simulation-based environment, thereby isolating the influences of other phenomena such as surface reflections and light attenuation in water. We acknowledge that these factors are non-negligible in real-world scenarios, and thus constitute a limitation of our current study. First, we create a detailed mesh representing a riverbed by applying a displacement modifier to a gravel texture within Blender. This dataset is then used to generate physically accurate rendered images using Blender Cycles, a path-tracing renderer. We assume a flat, distortion-free refractive surface (the water surface). The average

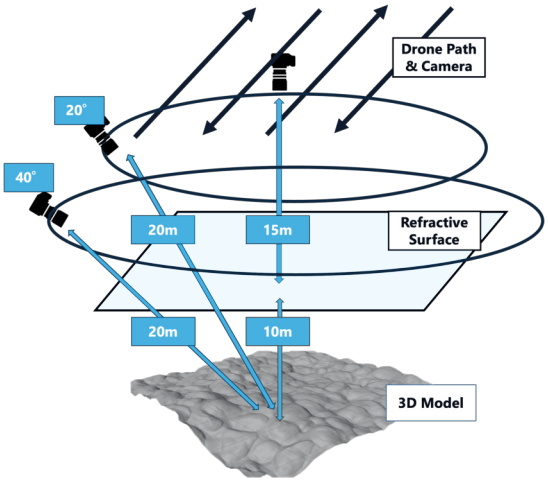

Figure 7. Overview of our simulated data capture setup.

water depth is set to 10 meters. The scene is captured from multiple viewpoints. We acquire images with 85% side and forward overlap, primarily focusing on the riverbed from a nadir view. Additionally, we capture images by circling around the center of the riverbed at off-nadir angles of 20° and 40°. While these multi-view conditions are advantageous for geometry estimation methods like Gaussian Splatting, they also introduce challenges. As illustrated in Fig. 4, larger observation angles to the refractive surface increase both object distortion and the discrepancy between the apparent and true positions, making the removal of refractive effects more difficult. Conversely, from a practical standpoint, preparing multi-view, multi-angle images for a wide area can be inefficient. The images are captured using a distortion-free pinhole camera model with a 70° field of view and a resolution of 800x800 pixels. The total number of images is 90, except images for the evaluation from novel views. Overview of a condition for capturing is shown in Fig. 7.

b. Implementation

We implement our Refractive-Aware Gaussian Splatting by extending the open-source gsplat (Ye et al., 2025) framework, a highly GPU-optimized library for various Gaussian Splatting methods. The parameter transformation that models refraction from the camera's viewpoint is implemented as a custom autograd function in PyTorch (Paszke et al., 2019). This allows for seamless integration into gsplat's differentiable rasterizer and parameter optimization pipeline. To solve the quartic equation (Eq. (8)) for the center correction described in Section a, we employ the Newton-Raphson method. This process is parallelized on the GPU using PyTorch, resulting in a modest computational overhead (at most 3x slower than the original gsplat implementation of 3DGS Kerbl et al., 2023).

For the optimization process, we adopt the adaptive density control from the vanilla 3DGS (Kerbl et al., 2023) and use its default hyperparameters. The optimization is run for 30,000 iterations. To simplify the implementation of explicit gradient calculations, we constrain the spherical harmonic (SH) degree of the Gaussians to 0, meaning they only store a base color and are view-independent. While standard 3DGS typically initializes the Gaussians from a sparse point cloud generated by an SfM pipeline like COLMAP (Schönberger & Frahm, 2016), it may be difficult to detect enough keypoints for a textureless beds. Therefore, we initialize the Gaussians as a planar point cloud positioned at the approximate water depth of 10 meters. All

experiments are conducted on a single NVIDIA RTX 3090 GPU.

c. Geometry Extraction

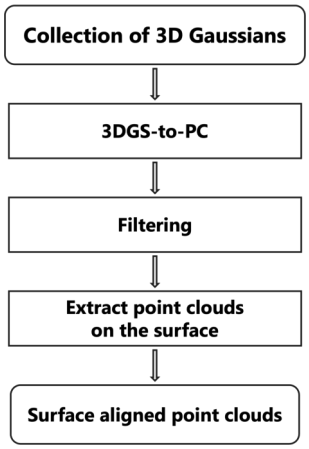

Figure 8. Flow Chart of Point Cloud Extraction.

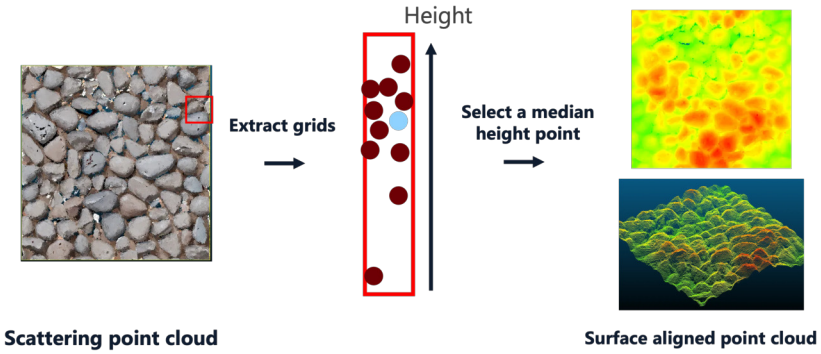

Figure 9. Our geometry extraction pipeline. Left: The initial noisy point cloud converted from 3DGS. Center: The point cloud is discretized into a 2D grid, and the median height is computed for each cell. Top-right: The resulting 2.5D height map. Bottom-right: The final surface-aligned point cloud generated from the height map.

To quantitatively evaluate geometric accuracy, the 3D Gaussian Splatting (3DGS) model must be converted into a standard 3D representation. For this study, we aim to extract a dense point cloud that is precisely aligned with the surface, which is appropriate for our target scenes, such as seabeds and riverbeds. While several sophisticated methods have been proposed to enhance the geometric fidelity of 3DGS, including SuGaR (Guédon & Lepetit, 2024), 2DGS (Huang et al., 2024), and TrimGS (Fan et al., 2024), we adopt a simple extraction method due to its ease of implementation.

First, we convert the 3DGS data into a raw point cloud using 3DGS-to-PC (Stuart & Pound, 2025). The core idea of this method is to sample points from each Gaussian ellipsoid in proportion to its volume, which generates a dense point cloud. However, this point cloud often contains significant noise and floaters artifacts inherent to volumetric sampling. To address this, we perform outlier removal by fitting a local plane to the neighbors of each point within a specified radius and culling points that are distant from this plane. This process is implemented using open-source point cloud library CloudCompare (Girardeau-Montaut et al., 2016) (v2.6.2).

Next, as illustrated in Fig. 9, the filtered point cloud is discretized into a 2D grid on the horizontal plane. The grid resolution is intentionally set finer than that of our quantitative evaluation

metrics to preserve high-frequency surface details. For each grid cell, we determine its elevation by calculating the median height of all points falling within the corresponding vertical column. This procedure yields a height map. Finally, a surface-aligned point cloud is generated by placing a point at the center of each grid cell at its computed median elevation. It is crucial to note that the quality of the final output is sensitive to parameters like the outlier filtering radius and the grid size, which must be carefully tuned for each specific dataset.

Evaluation

The training images for our evaluation are those generated as described in Section a, which exhibit distortions caused by a refractive plane.

a. Appearance Evaulation

Table 1. Quantitative results of appearance results, compared to an ablation of proposed components. Each model is trained from images with a refractive scene.

	Correction Components			Render w/ Refraction			Render w/o Refraction			Stats	
Method	Position	Scale	Opacity	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	# Gaussians (K)	Time (min)
3DGS	✗	✗	✗	-	-	-	0.682	14.12	0.292	40.0	6.8
Position	✓	✗	✗	0.980	37.56	0.023	0.953	21.49	0.039	23.1	10.9
+ Scale	✓	✓	✗	0.964	37.21	0.023	0.936	22.97	0.043	28.0	21.0
+ Opacity	✓	✗	✓	0.984	38.44	0.018	0.954	25.27	0.033	38.8	11.5
Ours	✓	✓	✓	0.981	38.42	0.017	0.933	25.97	0.046	41.3	21.9

Following the standards for Novel View Synthesis (NVS) evaluation, we assess the similarity between rendered images of our 3D model and the corresponding ground-truth images from unseen viewpoints. We employ three widely-used metrics: the Peak Signal-to-Noise Ratio (PSNR), which measures pixel-wise accuracy; the Structural Similarity Index Measure (SSIM) (Wang et al., 2004), which considers local image structures; and the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), which leverages deep features from a Convolutional Neural Network (CNN) to better approximate human perception. Our method aims to reconstruct a 3DGS model that is free of refractive artifacts. To comprehensively evaluate this, we report metrics under two conditions, as shown in Table 1. First, we render the scene with the refractive plane (”Render w/ Refraction”) and compare against the distorted ground-truth images. Second, we render the model without the refractive plane (”Render w/o Refraction”) and compare against the undistorted ground-truth images.

The results demonstrate that correcting the Gaussian center positions plays the most crucial

role, leading to a significant improvement across all metrics. For applications like bed surface classification, preserving structural details is paramount, making SSIM a key metric. Notably, our full method maintains an SSIM score above 0.93 even in the challenging ”w/o Refraction” scenario. However, the PSNR for our full method in this scenario is nearly 13 dB lower than in the ”w/ Refraction” case. We attribute this to the physical incorrectness as discussed in Section c. Our opacity correction uniformly reduces the opacity of all Gaussians, causing background Gaussians to be composited rather than revealing a black background. While the opacity regularization slightly contributes the increase of PSNR, the effect is insufficient to model the reduction of radiance at a refractive surface. This results in a subtle change in luminance, a discrepancy that is heavily penalized by the pixel-wise PSNR metric. Therefore, we need the correction method that directly models the change of the luminance through water to air.

Furthermore, the inclusion of the scale correction slightly degrades the SSIM score. This is likely because our method isotropically corrects the Gaussian volumes regardless of their shape or orientation, which can cause a blurring effect that harms fine structural details in the non-refractive rendering. Finally, while the training time for our method is over three times that of the baseline 3DGS, yet it remains within a practical range. We anticipate that this could be significantly accelerated with a dedicated CUDA implementation.

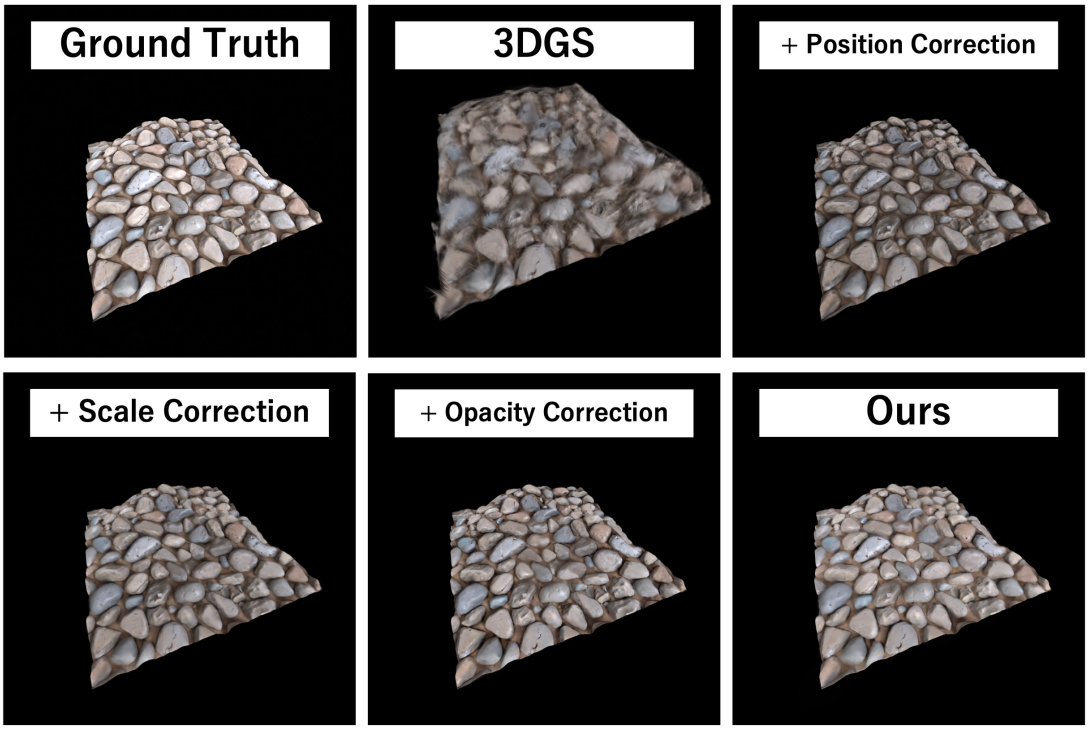

Figure 10. Visual comparisons between our method, Ground Truth, ablation of each correction. We also apply center position correction with scale correction and opacity correction.

b. Geometry Evalation

Table 2. Quantitative results of geometry, compared to an ablation of proposed components. Each model is trained from images with a refractive scene. We report the averaged chamfer distance, F1 score for each threshold (10 cm, 30 cm)

Method	Correction Components			$CD^{\downarrow}$ (m)	10cm			30cm		
	Position	Scale	Opacity		$Precise^{\uparrow}$ (%)	$Recall^{\uparrow}$ (%)	$F1^{\uparrow}$ (%)	$Precise^{\uparrow}$ (%)	$Recall^{\uparrow}$ (%)	$F1^{\uparrow}$ (%)
3DGS	✗	✗	✗	8.477	0.03	0.20	0.06	0.12	2.52	0.23
Position	✓	✗	✗	0.110	69.80	84.65	76.51	82.80	99.95	90.57
+ Scale	✓	✓	✗	0.033	81.10	91.06	85.79	93.03	99.97	96.37
+ Opacity	✓	✗	✓	0.054	79.29	90.80	84.66	90.64	99.99	95.09
Ours	✓	✓	✓	0.011	91.56	96.58	94.00	98.26	100.00	99.12

Accurately reconstructing the true geometry of a terrain observed through a refractive interface poses a significant challenge. In this section, we quantitatively evaluate our method's geometric accuracy by comparing the extracted point cloud against the ground truth from the simulator. We employ two standard metrics for evaluating similarity between point clouds: Chamfer Distance (CD) and F1 Score. The Chamfer Distance measures the average squared distance between nearest neighbors in two point clouds S_{gt} and S_{est}. It is defined symmetrically as:

$$\begin{aligned} \mathrm{CD}(S_{gt}, S_{est}) = &\frac{1}{|S_{gt}|} \sum_{x \in S_{gt}} \min_{y \in S_{est}} \|x - y\|_2^2 \\ &+ \frac{1}{|S_{est}|} \sum_{y \in S_{est}} \min_{x \in S_{gt}} \|x - y\|_2^2 \end{aligned} \tag{22}$$

A lower CD value indicates a closer match between the point clouds.

The F1 score assesses the overall quality by balancing precision and recall against a distance threshold τ.

- Precision: the percentage of points in the estimated cloud S_{est} that are within the threshold τ of the ground-truth cloud S_{gt}.
- Recall: the percentage of points in S_{gt} that are within τ of S_{est}.

The F1 score is the harmonic mean of Precision and Recall, defined as:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{23}$$

As shown in Table 2, the baseline 3DGS performs extremely poorly on the refractive scene, res-

ulting in a very high Chamfer Distance (8.477 m) and near-zero F1 scores, as it fails to account for the physics of refraction. Our ablation study demonstrates the incremental contribution of each proposed component. Introducing position correction yields the most substantial improvement, significantly reducing the CD to 0.110 m and boosting the 10 cm threshold F1 score to 76.51%. Adding scale correction further refines the geometry, lowering the CD to 0.033 m and increasing the F1 score to 85.79%. While adding scale correction and opacity correction each provides incremental improvements, the synergistic integration of all three components in our full method (”Ours”) achieves the best performance. This comprehensive approach achieves the lowest Chamfer Distance of 0.011 m and the highest F1 scores of 94.00% (at 10 cm) and 99.12% (at 30 cm). These results are achieved on a challenging dataset featuring water depths of up to 10 meters, highlighting the robustness of our method. This demonstrates that our combined correction strategy is crucial for achieving high-fidelity geometric reconstruction, complementing the appearance enhancements shown previously.

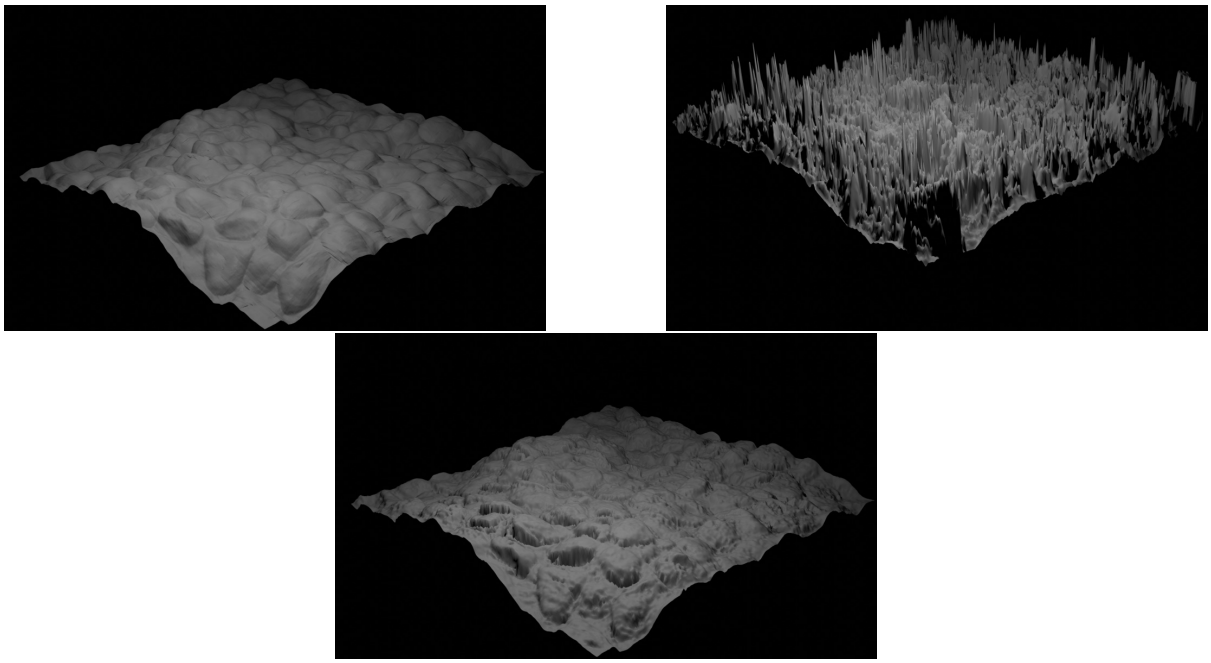

Figure 11. Qualitative results of geometry. Top-left: The ground truth geometry rendered in simulator. Top-right: The extracted geometry without correction. Bottom-center: The extracted geometry with all correction. Each image is rendered from an identical viewpoint and under the same lighting conditions. The ground truth and the corrected geometry are very similar in both position and shape. In contrast, the uncorrected geometry appears completely noisy and is incorrectly positioned at a shallower depth.

Conclusion

We presented refractive-aware Gaussian Splatting, which accurately models effects of refraction by leveraging the radiance field representation and the optimization framework of 3D Gaussian Splatting(Kerbl et al., 2023). Our differentiable transformation into apparent appearance enables refraction-free 3D reconstruction from images of a refractive scene. Our experiments demonstrate superior reconstruction quality, achieving high SSIM scores when our corrected model is compared against the pristine, non-refractive ground-truth images. Furthermore, the extracted 3D geometry attains an F1 Score of over 99% and a Chamfer Distance of 0.011 m, confirming its high geometric accuracy.

a. Limitations and Future Work

Our study primarily focuses on refractive effects, making our method best suited for idealized environments. In real-world scenarios, however, several other physical phenomena must be considered. Our model does not account for surface reflections, underwater light attenuation (i.e., absorption and scattering), or the Fresnel effect. This is a notable limitation, as our radiometric loss is minimized against observed images that inherently capture these phenomena.

Future work should address these effects. Surface reflections, for instance, could be mitigated during image capture using polarizing filters. Light attenuation could be tackled by incorporating recent advances, such as WaterSplatting (Li et al., 2025) and SeaSplat (Yang et al., 2025), which explicitly model and estimate these underwater characteristics. Furthermore, incorporating the Fresnel effect, which governs the view-dependent interplay between reflection and refraction, is crucial for higher fidelity. This could be implemented by applying a radiance reduction to each Gaussian's color parameter $\boldsymbol{c}$ based on the ray's angle of incidence.

Moreover, our geometry extraction pipeline currently relies on handcrafted heuristic priors. Alternative Gaussian Splatting methods with a stronger emphasis on geometric fidelity (Fan et al., 2024; Guédon & Lepetit, 2024; Huang et al., 2024) could further improve geometric accuracy. Introducing regularization terms that encourage Gaussians to align with surfaces could enable more faithful reconstructions of bed topography. Such regularization could also mitigate wave-induced distortions by prioritizing geometric plausibility over per-pixel fidelity.

Overall, our method represents an important step toward tackling refraction, one of the key challenges in photogrammetric bathymetry, and opens a new avenue for accurate shallow-water depth estimation.

References

Agrafiotis, P., Skarlatos, D., Georgopoulos, A., & Karantzalos, K. (2019). Shallow water bathymetry mapping from uav imagery based on machine learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2/W10*, 9–16. https://doi.org/10.5194/isprs-archives-xlii-2-w10-9-2019

Aira, L. S., Facciolo, G., & Ehret, T. (2025). Gaussian splatting for efficient satellite image photogrammetry. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5959–5969. https://doi.org/10.1109/cvpr52734.2025.00559

Alevizos, E., Nicodemou, V. C., Makris, A., Oikonomidis, I., Roussos, A., & Alexakis, D. D. (2022). Integration of photogrammetric and spectral techniques for advanced drone-based bathymetry retrieval using a deep learning approach. *Remote Sensing*, *14*(17), 4160. https://doi.org/10.3390/rs14174160

Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.-J., & Cai, J. (2024, October). Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *Computer vision – eccv 2024* (pp. 370–386). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72664-4_21

Dietrich, J. T. (2016). Bathymetric structure-from-motion: Extracting shallow stream bathymetry from multi□view stereo photogrammetry. *Earth Surface Processes and Landforms*, *42*(2), 355–364. https://doi.org/10.1002/esp.4060

Fan, L., Yang, Y., Li, M., Li, H., & Zhang, Z. (2024). Trim 3d gaussian splatting for accurate geometry representation. *arXiv preprint arXiv:2406.07499*.

Fischler, M. A., & Bolles, R. C. (1987). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision* (pp. 726–740). Elsevier. https://doi.org/10.1016/b978-0-08-051581-6.50070-2

Furukawa, Y., & Hernández, C. (2015). *Multi-view stereo: A tutorial*. now Publishers Inc. https://doi.org/10.1561/9781601988379

Furukawa, Y., & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(8), 1362–1376. https://doi.org/10.1109/tpami.2009.161

Girardeau-Montaut, D., et al. (2016). Cloudcompare. *France: EDF R&D Telecom ParisTech*, *11*(5), 2016.

Guédon, A., & Lepetit, V. (2024). Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5354–5363. https://doi.org/10.1109/cvpr52733.2024.00512

Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(6), 580–593. https://doi.org/10.1109/34.601246

He, J., Zhang, S., Cui, X., & Feng, W. (2024). Remote sensing for shallow bathymetry: A systematic review. *Earth-Science Reviews*, *258*, 104957. https://doi.org/10.1016/j.earscirev.2024.104957

Huang, B., Yu, Z., Chen, A., Geiger, A., & Gao, S. (2024). 2d gaussian splatting for geometrically accurate radiance fields. *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, 1–11. https://doi.org/10.1145/3641519.3657428

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, *42*(4), 1–14. https://doi.org/10.1145/3592433

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, H., Song, W., Xu, T., Elsig, A., & Kulhanek, J. (2025). WaterSplatting: Fast underwater 3D scene reconstruction using gaussian splatting. *3DV*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

Makris, A., Nicodemou, V. C., Alevizos, E., Oikonomidis, I., Alexakis, D. D., & Roussos, A. (2024). Refraction-aware structure from motion for airborne bathymetry. *Remote Sensing*, *16*(22), 4253. https://doi.org/10.3390/rs16224253

Matsuki, H., Murai, R., Kelly, P. H. J., & Davison, A. J. (2024). Gaussian splatting slam. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18039–18048. https://doi.org/10.1109/cvpr52733.2024.01708

Missailidis, G. T., & Guilhon, I. (2025). Determination of the position of an image formed by flat refracting surfaces at large angles of incidence: Three different didactic approaches. *Revista Brasileira de Ensino de Física*, *47*. https://doi.org/10.1590/1806-9126-rbef-2024-0417

Moreau, A., Song, J., Dhamo, H., Shaw, R., Zhou, Y., & Pérez-Pellitero, E. (2024). Human gaussian splatting: Real-time rendering of animatable avatars. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 788–798. https://doi.org/10.1109/cvpr52733.2024.00081

Nassar, A. B. (1994). Apparent depth. *Physics Teacher*, *32*(9), 526–29.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. https://arxiv.org/abs/1912.01703

Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shao, Z., Wang, Z., Li, Z., Wang, D., Lin, X., Zhang, Y., Fan, M., & Wang, Z. (2024). Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1606–1616. https://doi.org/10.1109/cvpr52733.2024.00159

Stuart, L. A. G., & Pound, M. P. (2025). 3dgs-to-pc: Convert a 3d gaussian splatting scene into a dense point cloud or mesh. https://arxiv.org/abs/2501.07478

Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment — a modern synthesis. In *Vision algorithms: Theory and practice* (pp. 298–372). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44480-7_21

Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Woodget, A. S., Carbonneau, P. E., Visser, F., & Maddock, I. P. (2014). Quantifying submerged fluvial topography using hyperspatial resolution uas imagery and structure from motion photogrammetry. *Earth Surface Processes and Landforms*, *40*(1), 47–64. https://doi.org/10.1002/esp.3613

Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., & Li, X. (2024). Gs-slam: Dense visual slam with 3d gaussian splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19595–19604. https://doi.org/10.1109/cvpr52733.2024.01853

Yang, D., Leonard, J. J., & Girdhar, Y. (2025). Seasplat: Representing underwater scenes with 3d gaussian splatting and a physically grounded image formation model. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 7632–7638. https://doi.org/10.1109/icra55743.2025.11128502

Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., et al. (2025). Gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, *26*(34), 1–17.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00068

Zheng, J., Zhu, Z., Bieri, V., Pollefeys, M., Peng, S., & Iro, A. (2025). Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zwicker, M., Pfister, H., van Baar, J., & Gross, M. (2001). Ewa volume splatting. *Proceedings Visualization, 2001. VIS '01.*, 29–538. https://doi.org/10.1109/VISUAL.2001.964490