



HAL
open science

Intrinsic decomposition and editing of 3D Gaussian splats

Alexandre Lanvin, Simon Lucas, Jeffrey Hu, Adrien Bousseau, George Drettakis

► **To cite this version:**

Alexandre Lanvin, Simon Lucas, Jeffrey Hu, Adrien Bousseau, George Drettakis. Intrinsic decomposition and editing of 3D Gaussian splats. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2026, <10.1145/3804495>. <hal-05602203>

HAL Id: hal-05602203

<https://hal.science/hal-05602203v1>

Submitted on 4 May 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Intrinsic decomposition and editing of 3D Gaussian splats

ALEXANDRE LANVIN, Inria, Université Côte d’Azur, France

JEFFREY HU, Inria, Université Côte d’Azur, France and Cambridge University, United Kingdom

SIMON LUCAS, Inria, Université Côte d’Azur, France

ADRIEN BOUSSEAU, Inria, Université Côte d’Azur, France

GEORGE DRETTAKIS, Inria, Université Côte d’Azur, France

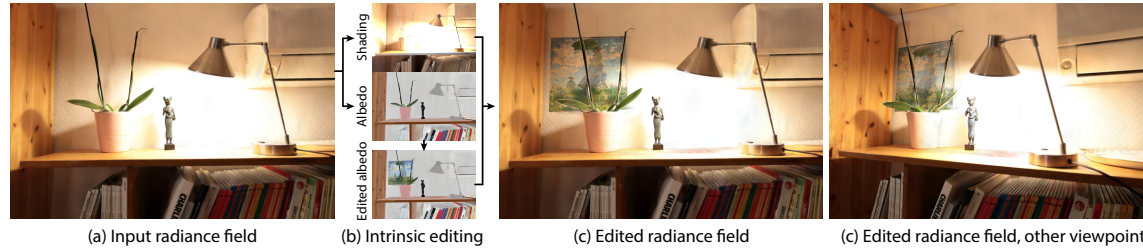


Fig. 1. By decoupling albedo and shading in a radiance field (a-b) our method enables direct edition of albedo in a physically plausible way while remaining multi-view consistent (c-d).

Intrinsic decomposition — which expresses image colors as the product of diffuse albedo and shading, possibly augmented with view-dependent residuals — has a long history in image editing as it enables the modification of object colors and textures without altering lighting. We extend intrinsic decomposition to radiance fields represented with Gaussian splatting by proposing solutions to three key aspects of such decomposition. First, we describe how to model the intrinsic decomposition as independent sets of Gaussian primitives, which allows each set to adapt to the characteristics of the layer it represents. Second, we present an optimization procedure guided by data-driven predictions to disentangle multi-view photographs of a scene into the aforementioned intrinsic sets. Finally, we provide an editing workflow where users modify the texture of planar surfaces simply by modifying the albedo of that surface in one image. Capturing this edit within the intrinsic radiance field allows re-rendering of the edited scene with plausible lighting under arbitrary viewpoints.

CCS Concepts: • **Computing methodologies** → **Rasterization**; *Neural networks*; **Reflectance modeling**.

ACM Reference Format:

Alexandre LANVIN, JEFFREY HU, SIMON LUCAS, ADRIEN BOUSSEAU, and GEORGE DRETTAKIS. 2026. Intrinsic decomposition and editing of 3D Gaussian splats. *Proc. ACM Comput. Graph. Interact. Tech.* 9, 1, Article 10,19 (May 2026), 18 pages. <https://doi.org/10.1145/3804495>

Authors’ Contact Information: Alexandre LANVIN, Inria, Université Côte d’Azur, France, Alexandre.Lanvin@inria.fr; JEFFREY HU, Inria, Université Côte d’Azur, France, huj14@gmail.com; SIMON LUCAS, Inria, Université Côte d’Azur, France, simon.lucas@inria.fr; ADRIEN BOUSSEAU, Inria, Université Côte d’Azur, France, adrien.bousseau@inria.fr; GEORGE DRETTAKIS, Inria, Université Côte d’Azur, France, George.Drettakis@inria.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2026 Copyright held by the owner/author(s).

Manuscript submitted to ACM

Manuscript submitted to ACM

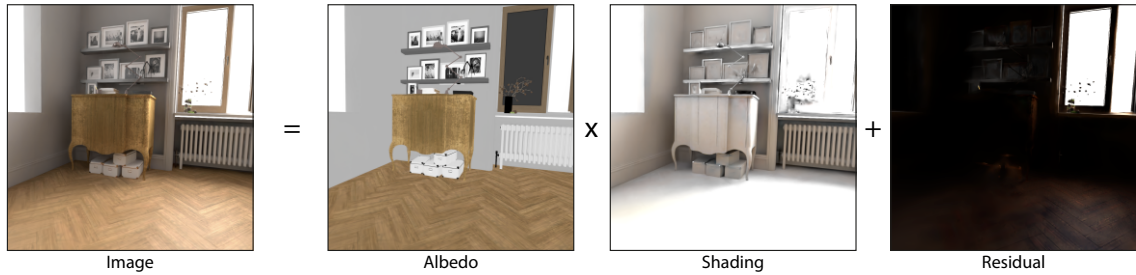


Fig. 2. The intrinsic image model expresses the observed image (left) as the product of diffuse albedo and shading, summed with a view-dependent residual (right).

1 Introduction

Intrinsic decomposition is a fundamental task for image editing as it allows modification of object colors (called *albedo*) without altering shading and shadows [Bonneel et al. 2017]. Numerous algorithms have been proposed to decompose images [Bousseau et al. 2009], image collections [Laffont et al. 2012], videos [Bonneel et al. 2014], lightfields [Garces et al. 2017], and neural radiance fields [Ye et al. 2023]. In this paper, we describe how to *model*, *reconstruct* and *edit* intrinsic decompositions of radiance fields represented as 3D Gaussian splats [Kerbl et al. 2023], allowing real-time navigation in the modified 3D scene.

Modeling intrinsic decompositions of Gaussian splats. 3D Gaussian splatting (3DGS) represents the radiance field of a scene as a collection of 3D Gaussian primitives that store localized view-dependent colors, typically optimized to reproduce the colors observed in multiple photographs of the scene. Prior work proposed to decompose 3D Gaussian splats by augmenting each primitive with intrinsic quantities, i.e. to separate the view-dependent color of each Gaussian into its albedo, shading and view-dependent residual values (or other material-lighting separations) [Jiang et al. 2024; Liang et al. 2024; Ye et al. 2024]. But while intrinsic quantities often exhibit some spatial correlation (for example, object boundaries typically produce discontinuities in both albedo and shading), they can also require significantly different spatial resolution (for example, object textures produce high-frequency details in albedo but not in shading, while hard shadows produce sharp edges in shading that should not appear in albedo). We thus propose to model and reconstruct intrinsic decompositions with *separate sets of Gaussians* for the albedo, shading and residual components, allowing our approach to adapt the number and distribution of Gaussian primitives to the local characteristics of each signal.

Reconstructing intrinsic decompositions of Gaussian splats. Decomposing observed radiance into intrinsic quantities is an ill-posed problem, for which numerous heuristic [Bonneel et al. 2017] and data-driven [Garces et al. 2022] priors have been proposed. We follow the latter approach and leverage a recent diffusion-based video decomposition model [Liang et al. 2025a] to predict the albedo component of all images of a scene. Compared to single-image models, this video model produces consistent predictions across successive images, which is critical to fuse these predictions into a single 3D Gaussian splat. However, being free of shading, albedo images exhibit large uniform areas that hinder 3D reconstruction. We address this challenge by regularizing the 3D reconstruction of albedo Gaussians with per-image depth predictions computed on the original input images. Equipped with a consistent *albedo field*, we next optimize a view-independent *shading field* and a view-dependent *residual field* – each represented by its own set of Gaussians – such that rendering and compositing all three fields reproduces the input images.

Editing intrinsic decompositions of Gaussian splats. Similar to prior work on intrinsic images, we demonstrate the value of our decomposition by modifying object colors and texture while preserving realistic shading and shadows (Figure 1). With our solution, users can orient a virtual camera towards the surface they wish to edit, and paint over the resulting image to modify its albedo. We then update the 3D Gaussians that represent the albedo field to capture these user edits while keeping the shading Gaussians untouched. The main difficulty resides in ensuring that the edits performed in one view stay consistent in other views, which is not guaranteed when a single view is used for Gaussian optimization. To address this difficulty, we reconstruct a proxy surface around the user edits and use it to reproject the edits in nearby views to further constrain the albedo update.

In summary, our contributions are:

- A representation of radiance fields as three separate Gaussian splats representing albedo, shading and view-dependent residuals.
- A method to reconstruct each of these intrinsic fields from multiple photographs, based on a video diffusion prior.
- A method that leverages this intrinsic decomposition to edit the albedo of a scene while keeping the shading untouched, allowing free-viewpoint navigation in the edited scene.

We demonstrate our approach by decomposing and editing radiance field reconstructions of three real-world scenes, and of one synthetic scene for which we have ground truth albedo.

2 Related Work

There is a vast literature on intrinsic image decomposition, typically separating the image into diffuse albedo and lighting layers. In later work, a residual layer was also introduced to better model view-dependent effects. We refer the reader to the excellent surveys covering this body of work [Bonnel et al. 2017; Garces et al. 2022]. Early approaches depended on heuristic priors, for example assuming that the albedo is piecewise-constant while lighting is smooth, with mixed results. Recent approaches employ machine learning to obtain more powerful, data-driven priors [Careaga and Aksoy 2023, 2024]. In particular, diffusion models offer rich priors learned from very large datasets, and recent work has demonstrated that fine-tuning a large diffusion model can achieve impressive intrinsic decompositions [Kocsis et al. 2024; Luo et al. 2024; Zeng et al. 2024]. Since we are interested in the multi-view context, we employ a video-based diffusion model [Liang et al. 2025a] to extract coherent intrinsic layers for multiple, densely-captured photographs of a scene.

Intrinsic decomposition is often a key component for relighting captured scenes. Early work combined forward rendering with heuristic priors [Duchêne et al. 2015], while subsequent methods adopted deep learning for better results [Philip et al. 2021]. The introduction of Neural Radiance Fields (NeRFs) quickly led to exploration of how to jointly perform 3D reconstruction, intrinsic decomposition and relighting [Boss et al. 2021a,b; Srinivasan et al. 2021; Ye et al. 2023; Zhang et al. 2021], but the computational and memory expense of the original method plus the additional terms is prohibitive for all practical purposes. 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] reduces the computational cost, and a large number of methods consider intrinsic decompositions of Gaussian splats. However, most of these (e.g., [Du et al. 2025; Haiyang Bai 2025; Jiang et al. 2024; Liang et al. 2024; Ye et al. 2024]) perform some form of inverse rendering under environment map lighting – often applied to an isolated object – and thus have difficulty handling the more involved context of full scenes with local lighting. We focus on such scenes and avoid the need for full inverse

rendering by relying instead on a pre-trained diffusion model for intrinsic decomposition. While our decomposition does not allow relighting, it does support convincing albedo edits.

Several methods have been developed for editing NeRFs, and in particular modifying the color, e.g., for posterization [Tojo and Umetani 2022] or palette-based color editing [Kuang et al. 2023]. The speed of 3DGS has allowed the development of interactive editing tools for radiance fields, such as SplatShop [Schütz et al. 2025], that allow editing of the geometry and “painting” on top of existing splats, with an emphasis on Virtual Reality interaction. ReCoGS [Rutayisire et al. 2025] allows real-time re-coloring of splats using unprojection and fine-tuning to obtain the final result. With our intrinsic decomposition of 3DGS we can go much further than simple recoloring, since by separating the lighting from the albedo/texture we can perform color and texture edits while preserving the lighting conditions in the scene.

3 Method

Our pipeline is illustrated in Fig. 3. We capture a scene with a video, which we feed to a video diffusion model to estimate albedo for each frame [Liang et al. 2025a]. We also feed the video to a monocular depth estimator to obtain an initial guess of the scene geometry [Yang et al. 2024]. We then subsample the video to create a typical multi-view dataset used to create a radiance field (a few hundred images in most of our examples). Equipped with this multi-view dataset and accompanying albedo estimates, the second step of our method is to create an *albedo field* by optimizing a 3DGS representation from the albedo images. We regularize the optimization using the estimated depth. In the third step, we optimize a *shading field* and a *residual field* that we composite with the albedo field to reproduce the original input images. With this representation, the user can directly edit the albedo layer, e.g., by painting or by adding a texture. We then re-optimize the albedo Gaussians to capture the modified texture, allowing us to render the modified scene from any viewpoint while maintaining the original lighting effects. We next describe each step in detail.

3.1 Modeling intrinsic decompositions of Gaussian splats

We adopt the intrinsic image formation model that expresses each observed image I as the product of a diffuse albedo image A and a diffuse shading image S , summed with a view-dependent residual R [Careaga and Aksoy 2024; Ye et al. 2023] (Fig. 2):

$$I = A \times S + R. \quad (1)$$

Prior work on intrinsic decomposition of Gaussian splats [Jiang et al. 2024; Liang et al. 2024; Ye et al. 2024] (resp. neural radiance fields [Ye et al. 2023]) optimize a single set of Gaussians (resp. a single MLP) to express all intrinsic quantities. Instead, we propose to optimize three separate sets of Gaussians \mathcal{G}_A , \mathcal{G}_S and \mathcal{G}_R which, once splatted and composited according to Eq. 1, reproduce all input images. Using three separate sets allows our approach to better capture the (possibly misaligned) sharp details of each intrinsic component.

Similarly to standard 3DGS [Kerbl et al. 2023], we model each Gaussian primitive within \mathcal{G}_A , \mathcal{G}_S or \mathcal{G}_R with a position p , color c , opacity α and a covariance matrix Σ :

$$g = \{c, \alpha, p, \Sigma\}. \quad (2)$$

However, while the original method uses spherical harmonics to encode view-dependent radiance in c , we only use spherical harmonics for the Gaussians \mathcal{G}_R that model the view-dependent residual, and resort to a simple triplet $c = \{r, g, b\} \in \mathbb{R}^3$ for the Gaussians \mathcal{G}_A and Gaussians \mathcal{G}_S that model diffuse albedo and shading, respectively. For a given viewpoint, we render each set of Gaussians into its corresponding image by using the same rasterizer as in the

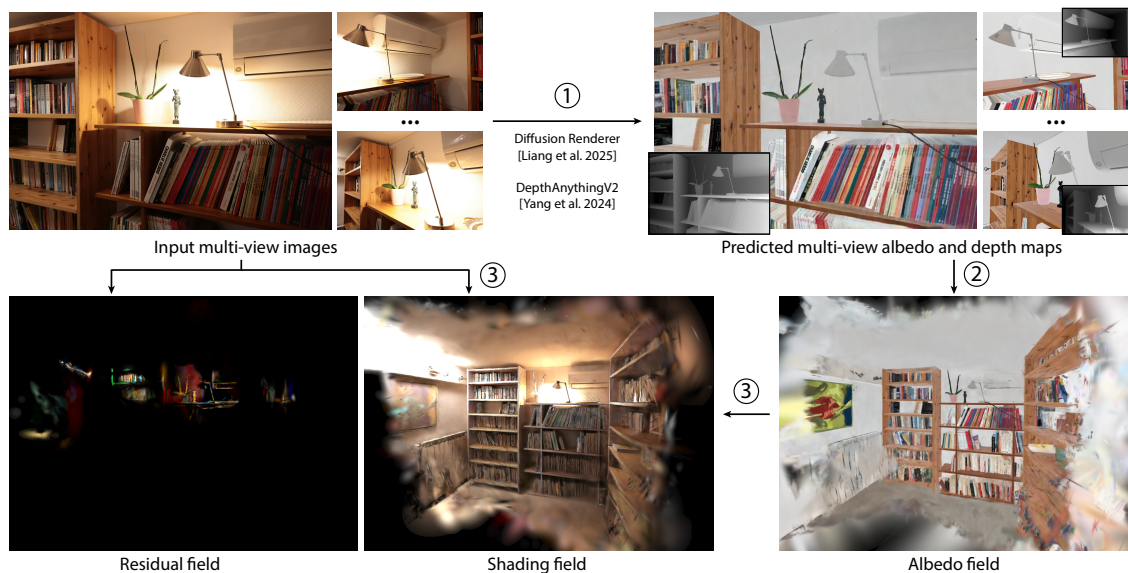


Fig. 3. Overview of the pipeline for intrinsic reconstruction using distincts sets of Gaussians. Our pipeline takes as input multiple photographs of a scene and uses off the shelf methods to predict depth maps and albedo maps for each photograph (1). We first use these albedo and depth maps to reconstruct an albedo field (2, Sec.3.2.2), and then optimize a shading field and a view-dependent residual field such that they best reproduce the input images when combined with the albedo (3, Sec.3.2.3). Users can edit the albedo by placing textured planes in the scene, our method updates the underlying albedo field accordingly (Sec.3.3).

original 3DGS implementation. The three sets of Gaussians \mathcal{G}_A , \mathcal{G}_S and \mathcal{G}_R can be seen as separate 3D fields, which we call *albedo*, *shading* and *residual fields*.

3.2 Reconstructing intrinsic decompositions of Gaussian splats

We now describe how to reconstruct the three sets of intrinsic Gaussians \mathcal{G}_A , \mathcal{G}_S and \mathcal{G}_R from multiple photographs of a scene. Given the ambiguity of the task, we first leverage a pre-trained video diffusion model to estimate the albedo for each input image. This data-driven estimate allows us to reconstruct \mathcal{G}_A , which we then freeze to optimize \mathcal{G}_S and \mathcal{G}_R to best reconstruct the images according to Eq. 1.

3.2.1 Data-driven albedo estimation. To extract albedo for each input image, we employ DiffusionRenderer, a state-of-the-art material estimation method based on video diffusion models [Liang et al. 2025b], which yields temporally consistent albedo estimates across all frames. Our initial experiments with single-image methods (e.g., [Zeng et al. 2024]) revealed significant inter-frame inconsistencies that degrade 3D reconstruction quality.

In practice, video capture produces sequences containing hundreds of frames, whereas DiffusionRenderer is limited to processing clips of 57 frames at 24 FPS. Naïvely partitioning a long video into independent chunks and denoising each chunk separately introduces visible discontinuities in albedo predictions at chunk boundaries. To mitigate this issue, we adopt Brick Diffusion [Yuan et al. 2025] and stagger the denoising process across neighboring chunks, enabling smooth transitions between the chunks.

Cosmos DiffusionRenderer operates in the latent space of the Cosmos Video Tokenizer [NVIDIA et al. 2025]. However, the tokenizer’s asymmetric design complicates the direct application of brick diffusion. DiffusionRenderer denoises

eight latent codes per step, where the first latent decodes to a single frame, and each subsequent latent decodes to a group of eight frames, yielding a total of 57 frames. Effective application of brick diffusion requires shifting the denoising window by both an integer number of latent codes and an integer number of frames simultaneously.

To address this, we process the sequence in chunks of 64 frames, corresponding to one start latent and eight regular latents. At each denoising step, we operate on 57 out of the 64 frames to denoise one start latent and seven of the eight regular latents. The window is then advanced by four latent codes, corresponding to a shift of 32 frames. After denoising, all start and regular latents are decoded to frames, and redundant frames are discarded. This strategy produces temporally consistent albedo predictions over long sequences with minimal boundary artifacts.

The video diffusion model expects and produces video at 24 FPS, which is far too dense for radiance field reconstruction. Hence the set of input images we use in our method is a subset of the video frames, it is obtained by subsampling the video sequence to a few hundred images taking every n th frame, with a preference for frames with low blur. We determined experimentally that sampling every $n = 5$ frame provided a sufficiently dense camera path for high quality novel view synthesis. We then run Structure-from-Motion (SfM) [Schönberger and Frahm 2016] on the extracted images to compute camera poses. Importantly, we apply the undistortion SfM applies on the input images to the albedo images as well.

3.2.2 Albedo reconstruction. As can be seen in Fig. 2, the albedo images typically contain piecewise constant color regions with an insufficient number of geometric cues, making it harder for the 3DGS algorithm to place Gaussians in 3D space. We address this challenge by regularizing the reconstruction against depth maps estimated for each input view [Chung et al. 2024], which we obtain using the Depth Anything V2 monocular depth estimator [Yang et al. 2024].

As in 3DGS, we start from a set of calibrated cameras and an initial point cloud typically produced by SfM. The set of albedo Gaussians \mathcal{G}_A is optimized using the same learning rates and activation functions as in the original implementation except for the albedo color, which we expect to be in $[0; 1]$ so we use a sigmoid activation function. For regularization, we follow [Chung et al. 2024] and render a depth image D by replacing the color of the Gaussians by their depth. Given the predicted depthmap \tilde{D} , we define the regularization term as:

$$\mathcal{L}_{depth} = \|\tilde{D} - D\|_1. \quad (3)$$

This regularization term encourages the placement of Gaussians close to the actual geometry of the scene and greatly improves densification in texture-free regions, such as the constant color patches frequently observed in the input albedo images. Depth regularization also helps snapping the user edits to the scene surface during albedo editing (Sec.3.3).

To improve multi-view consistency, we add another regularization term on the average scale \bar{s} of each Gaussian primitive to encourage the primitives to flatten and form sharp surface boundaries:

$$\mathcal{L}_{scale} = \bar{s}. \quad (4)$$

Considering the predicted albedo images \tilde{A} and rendered albedo images A , the total loss function for reconstructing the albedo radiance field is computed as:

$$\mathcal{L} = (1 - \lambda)\|\tilde{A} - A\|_1 + \lambda\mathcal{L}_{D-SSIM}(\tilde{A}, A) + \lambda_d\mathcal{L}_{depth} + \lambda_s\mathcal{L}_{scale}. \quad (5)$$

An important benefit of our approach is that the albedo field is decoupled from shading and thus does not reconstruct lighting variations and shadows. This results in more uniform distribution of Gaussian primitives across the scene, which in turn allows more efficient editing of albedo and texture.

3.2.3 Shading and Residual Reconstruction. Once the albedo field is reconstructed, the \mathcal{G}_A splats can be rendered from any viewpoint to produce albedo images. Given an input image I and the rendered albedo image A , we invert the image formation model (Eq.1) to optimize the shading and residual Gaussians \mathcal{G}_S and \mathcal{G}_R . Denoting S and R the rendered shading and residual, we aim to minimize the reconstruction loss:

$$\mathcal{L} = (1 - \lambda) \|I - (A \times S + R)\|_1 + \mathcal{L}_{D-SSIM}(I, A \times S + R) + \lambda_d \mathcal{L}_{depth} + \lambda_s \mathcal{L}_{scale} \quad (6)$$

However, since jointly optimizing \mathcal{G}_S and \mathcal{G}_R is a challenging task, we perform this optimization in two phases. First, we ignore the residual term and only optimize \mathcal{G}_S using the diffuse image formation model $I = A \times S$; these shading Gaussians are initialized from SfM points in the same manner as albedo Gaussians. Once this first optimization is completed, we freeze \mathcal{G}_S and optimize \mathcal{G}_R to capture residual non-diffuse effects.

Furthermore, since the shading of a scene is unbounded and can exhibit high dynamic range, we follow the recommendations of Careaga and Aksoy [2023] and compute the photometric loss in the inverse domain, i.e., minimizing the difference between $\frac{1}{1+I}$ and $\frac{1}{1+A \times S + R}$. This transformation bounds the values in $[0; 1]$, alleviating the need for an activation function on the color parameters.

If an albedo channel is dark, the shading can become arbitrarily large, which may lead to incorrect color reproduction. To alleviate this problem, we encourage the shading to be gray with an additional regularization term that computes, for each shading Gaussian, the range of its color parameters:

$$\mathcal{L}_{gray} = \max(c) - \min(c). \quad (7)$$

Residual Gaussians \mathcal{G}_R are created where they are most likely to be needed. Specifically, we iterate through the training views and compute the difference between the input image and the composited diffuse render (i.e., $A \times S$). We randomly select a subset of pixels with high difference and use their predicted depth as positions to spawn residual Gaussians. We model the color of these residual Gaussians with up to 3 bands of spherical harmonics to represent view-dependent effects. We use the same scheduling as in 3DGS to progressively add additional bands to the spherical harmonics and we keep the same scaling and depth regularizations as in albedo and shading reconstruction stages. Optimization then proceeds with the full image formation model (Eq. 6) but only the residual Gaussians are updated.

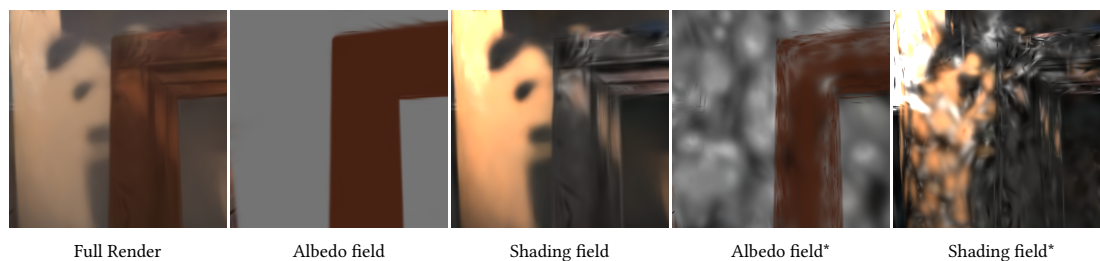


Fig. 4. Each intrinsic field is optimized to represent different quantities. In Albedo field* and Shading field*, we have artificially scaled down the scale of the Gaussians. Notice the different distribution of primitives across albedo and shading field, each capturing details specific to each.

Each optimized field is composed of primitives with positions and shapes adapted to the content of each intrinsic layer. This can be seen in Fig. 4 where in the two rightmost images we have artificially reduced the scale of the Gaussians to better visualize the individual primitives. We can see that small and anisotropic Gaussian primitives are created in the shading layer to model shadows, and equivalently to model texture content in albedo.

3.3 Editing intrinsic decompositions of Gaussian splats

Once we have the full radiance field decomposed in three components, it is easy to edit the albedo separately while maintaining lighting and shadow effects. We provide an interactive interface allowing the user to perform albedo editing. To better comprehend the editing process, please see the supplemental video.

Users of our interface can edit the albedo in any input view. However, while we can minimize Eq. 5 on this view to update the albedo Gaussians accordingly, the resulting albedo field often degrades as soon as we move away from the view under which the editing was performed. Achieving multi-view consistency requires minimizing Eq. 5 over multiple edited views, which would be prohibitive to provide by users. Our solution consists in *synthesizing* these multiple views using depth-based reprojection of the user edits.

Unfortunately, 3DGS does not provide sufficiently reliable depth to perform such reprojection into neighboring views; for this reason we use a proxy plane to approximate the surface to be edited. The user first indicates the surface of interest by dragging a rectangular window in the image. We select, for each pixel in that window, the primitive that contributes most to its rendering. We then use RANSAC to fit the proxy plane over the center of these Gaussians. The plane can be optionally manipulated using standard 3D widgets, allowing the user to adjust its placement in the scene. Since a typical editing session involves adding or replacing the texture of a surface; we render the new texture over the proxy plane for pre-visualization.

Once the user is satisfied with the position of the textured proxy plane, we render a set of edited albedo images from multiple nearby viewpoints by compositing the plane with the albedo field, using the rendered depth to test for occlusions. We sample these virtual viewpoints on an hemisphere centered on the editing viewpoint. Finally, we use these multiple edited images to re-optimize the albedo Gaussians using Eq. 5. Since the user edits might contain details not present in the original albedo, we spawn additional albedo Gaussians on the textured proxy plane by randomly sampling 20% of the pixels covered by the edit in the edited view. This editing process is illustrated in Fig. 5.

4 Implementation

Since we are performing lighting computation, we need to perform all our compositing operations in linear space. We thus apply a simple inverse gamma correction to the sRGB images used as input to obtain “pseudo-linear” RGB images used for processing. We apply the same gamma inversion to the albedo maps produced by DiffusionRenderer, which we observed to be non-linear. In addition, the albedo images can be quite blurry, resulting in floaters. We apply a simple floater reduction approach to improve our results. Our floater removal operates on each input view: We visit each pixel and accumulate all Gaussians until transmittance is saturated (“expected termination” depth). We then compute the standard deviation of the depths of these primitives and increment a counter for those that have depth less than n standard deviations of the average depth; larger floaters typically satisfy this condition for several pixels, resulting in a high value of the counter. We then remove primitives with a high counter value. This operation is performed before every densification step, which happens every 100 iterations from iteration 500 to iteration 15000 of the optimization.

We have implemented our system in an interactive viewer [Shah et al. [n. d.]] shown in Fig. 6. The viewer includes interactive editing tools to select albedo Gaussians to be edited, and allows easy manipulation of the proxy plane to

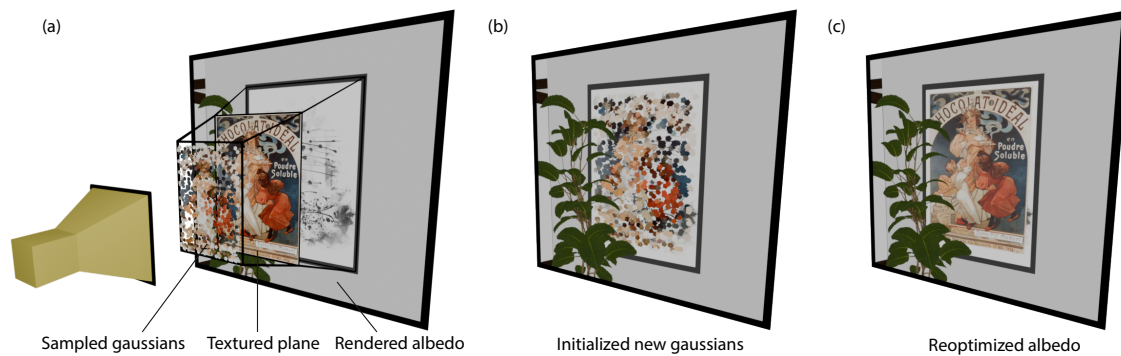


Fig. 5. To capture a user edit, we initialize new Gaussians on the textured proxy plane (a). These Gaussians are added to the set of albedo Gaussians and placed in 3d space using the proxy depth (b). The augmented set of albedo Gaussians is re-optimized to match the edited albedo images rendered from multiple viewpoints (c).

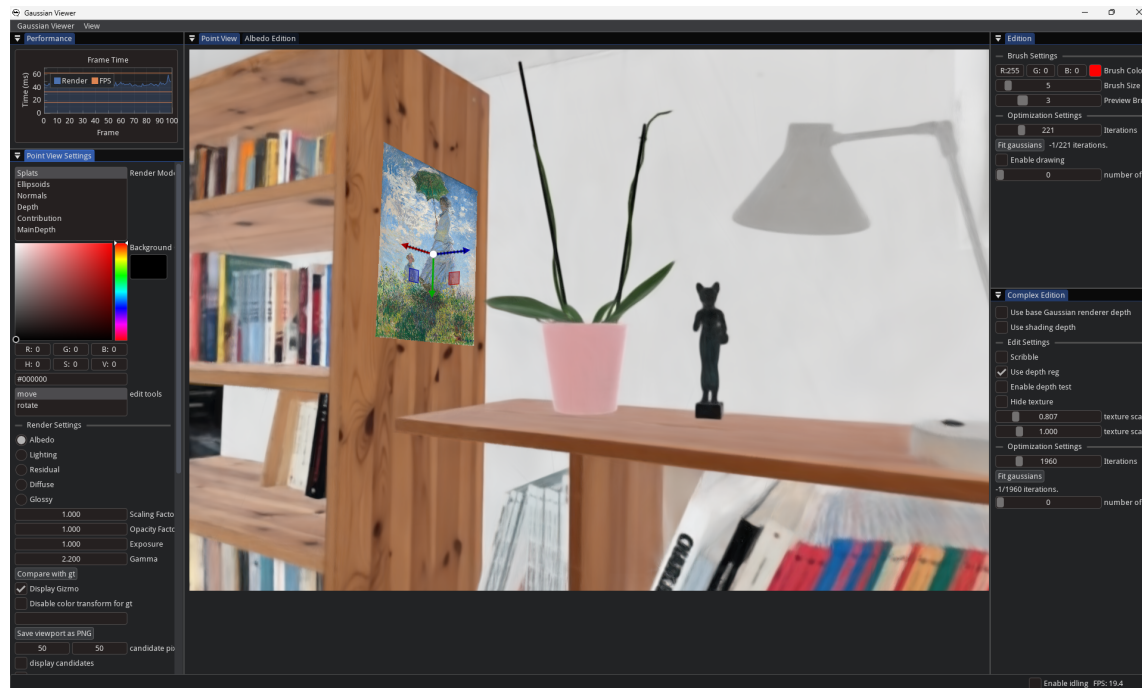


Fig. 6. We added custom widgets to enable the selection of different intrinsic fields and editing of the albedo field. The user can then upload and manipulate any image in an interactive session as well as navigating the 3D reconstruction in real time.

achieve the best results, including 3D widgets (please see supplemental video). We will release all source code of our system on publication.

5 Results and Evaluation

In Fig. 7 we show edits on three real scenes captured with a Canon EOS R6 in continuous shooting mode. The resulting sequence of still frames is dense enough to be processed as a video by DiffusionRenderer, which we use to extract albedo from the input images. Table. 1 provides a quantitative comparison on training times and primitive counts between our method and vanilla 3DGS [Kerbl et al. 2023] across our 4 scenes, all experiments ran on a H100 GPU. Our method requires an additional render call every time a new layer is reconstructed (i.e., two render calls when reconstructing a diffuse scene, three when enabling the residual layer) which yields slower rendering times that inevitably leads to longer training times.

We also show results on a synthetic scene that we use for the quantitative evaluation in the ablations (see Sec. 5.2). We rendered 288 images of this scene that we used as input photos for our intrinsic 3DGS pipeline. Our results on this synthetic scene (Fig. 8) illustrate the best possible outcome, since we use ground truth albedo and depth maps to reconstruct the intrinsic fields.



Fig. 7. Intrinsic decomposition and albedo editing on real scenes. See accompanying video for animated viewpoints.

5.1 Comparisons

5.1.1 Intrinsic decomposition. We compare our intrinsic decomposition both quantitatively and qualitatively with two inverse rendering methods based on 3D Gaussian splatting on the synthetic scene shown as inset in Fig 11. R3GS [Gao et al. 2023] and GI-GS [Chen et al. 2025] both propose a full inverse rendering pipeline using 3DGS, the former uses raytracing directly on Gaussians in 3D space while the latter relies on GBuffers obtained by rasterizing different properties on Gaussians to speed up the computations. Table. 2 shows that on our synthetic indoor scene, our method produces higher quality albedo while being significantly faster than the others at the cost of generating $\times 3.5$ more
 Manuscript submitted to ACM



Fig. 8. Intrinsic decomposition and albedo editing on a synthetic scene, using ground-truth albedo for supervision. By representing albedo and shading as two separate radiance fields, our method allows to introduce new visual details in the albedo without altering the soft shading and shadows.

Table 1. Training time and primitive count across scenes.

| Scene | Method | Layer | # Primitives | ratio (\times 3DGS) | Training Time | FPS |
|-----------------------|--------|----------|--------------|------------------------|---------------|--------|
| Basement 1 | Ours | Albedo | 1.575M | 1.01 | 00:06:56 | – |
| | | Shading | 469K | 0.30 | 00:09:45 | – |
| | | Residual | 141K | 0.09 | 00:11:49 | – |
| | | Full | 2.185M | 1.40 | 00:28:30 | 96.33 |
| | 3DGS | – | 1.558M | 1 | 00:06:14 | 501.17 |
| Basement 2 | Ours | Albedo | 851K | 0.40 | 00:04:28 | – |
| | | Shading | 675K | 0.32 | 00:07:21 | – |
| | | Residual | 70K | 0.03 | 00:09:02 | – |
| | | Full | 1.596M | 0.76 | 00:20:51 | 103.52 |
| | 3DGS | – | 2.113M | 1 | 00:06:19 | 448.95 |
| Basement 3 | Ours | Albedo | 984K | 0.91 | 00:06:05 | – |
| | | Shading | 272K | 0.25 | 00:08:56 | – |
| | | Residual | 130K | 0.12 | 00:10:54 | – |
| | | Full | 1.386M | 1.29 | 00:25:55 | 105.73 |
| | 3DGS | – | 1.076M | 1 | 00:04:55 | 420.12 |
| Synthetic room | Ours | Albedo | 1.285M | 0.82 | 00:06:04 | – |
| | | Shading | 752K | 0.48 | 00:09:44 | – |
| | | Residual | 52K | 0.03 | 00:11:23 | – |
| | | Full | 2.089M | 1.33 | 00:27:11 | 128.12 |
| | 3DGS | – | 1.576M | 1 | 00:05:10 | 831.67 |

primitives than GI-GS [Chen et al. 2025]. R3DG [Gao et al. 2023] also assumes a near natural white lighting so it cannot correctly represent the warmer, more realistic light placed in our synthetic scene; moreover it is designed for object-centric scenes so it fails to reconstruct our extended scene as shown in Fig 9.

Table 2. Quantitative results when comparing our intrinsic decomposition with GI-GS and R3GS.

| Method | Layer | PSNR | SSIM | LPIPS | # Primitives | Training Time |
|-----------|--------|--------|-------|-------|--------------|---------------|
| Ours full | Albedo | 29.419 | 0.932 | 0.095 | 1.285M | 00:06:04 |
| | Full | 31.016 | 0.923 | 0.117 | 2.089M | 00:27:11 |
| GI-GS | Albedo | 13.675 | 0.769 | 0.310 | – | – |
| | Full | 35.284 | 0.951 | 0.106 | 790k | 00:50:49 |
| R3GS | Albedo | 11.680 | 0.683 | 0.445 | – | – |
| | Full | 15.002 | 0.655 | 0.492 | 3.943M | 02:46:20 |

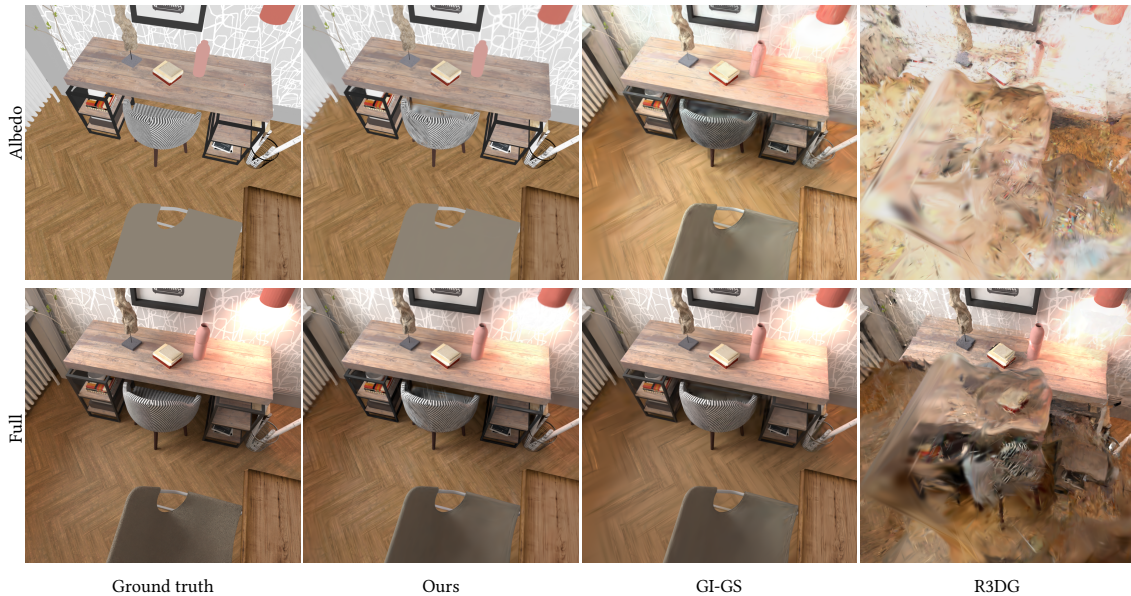


Fig. 9. Compared to our approach, GI-GS [Chen et al. 2025] retains significant shading residuals in the albedo, even though it better reconstructs the input image than R3DG [Gao et al. 2023], which is designed for object-centric scenes and as such fails on our extended scene.

5.1.2 Interactive edition. We choose to compare our work on an interactive editing task against two other 3DGS editing methods that can perform recoloring. ReCoGS [Rutayisire et al. 2025] allows interactive tinting of Gaussians by reoptimizing the colors of the primitives affected by a user-specified mask. The second method we compare to is SplatShop [Schütz et al. 2025], which provides a framework to select, transform and directly paint Gaussians that are intersected with a 3D brush. SplatShop is tailored for VR and the painting does not involve any optimization.

This recoloring application showcases two advantages of our method compared to previous work, as it allows to modify the albedo without altering the shading. This is illustrated in Fig. 10 where we recolored textured surfaces with a uniform red color, for a synthetic scene and a real scene. Setting a red tint in ReCoGS preserves both the shading and the texture information, while painting with a red brush in SplatShop erases both the shading and the texture. In contrast, our method allows to remove the texture without altering the shading and cast shadows. Note however that on the real scene, errors in the albedo prediction makes some albedo details appear in the shading, and as such remain

visible in our edit, albeit much less than with ReCoGS. Our decomposition also yields a sharper boundary around the edited area, which we achieve by updating the parameters of the albedo Gaussians while keeping the shading and residual Gaussians untouched.

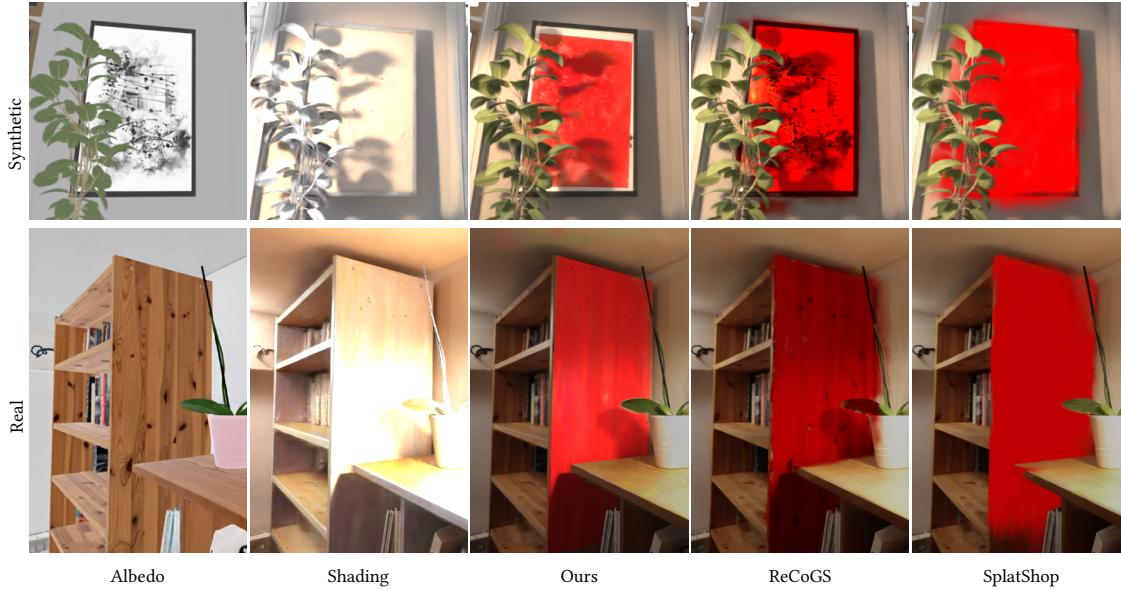


Fig. 10. Our intrinsic decomposition enables realistic editing of colors and textures compared to previous methods where shading and albedo are entangled. Our method correctly preserves shadows while replacing the texture, while ReCoGS incorrectly preserves the texture [Rutayisire et al. 2025], and SplatShop [Schütz et al. 2025] overwrites both texture and shadows. Note however that slight texture details remain visible in our result for the real scene (bottom) due to the albedo predictor that tends to produce blurry albedo maps.

5.2 Ablation Study

We evaluate our method across different configurations on the quality of the reconstructed albedo field, shading field and fully composited renderings using the image formation model of Eq. 1.

We use the synthetic scene shown as inset to perform quantitative evaluation, comparing the rendered results of each reconstructed field with the values of ground truth renderings.

To compute our error metrics for the ablations, we left out every 8th image of the dataset as test views and compute PSNR, SSIM and LPIPS on albedo, shading and fully composited renderings. The results of the quantitative evaluation are shown in Tab. 3.

Our color regularization (Eq. 7) yields a cleaner shading reconstruction that contains fewer artifacts while remaining colorful; this regularization does not degrade the reconstruction quality, as can be seen in the metrics that are essentially unchanged. The qualitative effect of the regularization can be seen in Fig. 12.

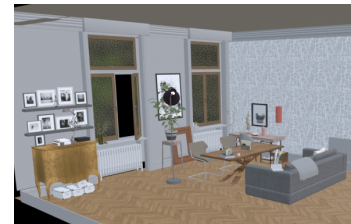


Fig. 11. Our synthetic scene

Table 3. Quantitative results of ablations, namely shading reconstruction without Color Regularization (CR*) and albedo reconstruction using predicted albedo.

| Method | Layer | PSNR | SSIM | LPIPS |
|------------------|---------|--------|-------|-------|
| Ours full | Albedo | 31.827 | 0.945 | 0.078 |
| | Shading | 22.876 | 0.894 | 0.210 |
| | Full | 35.679 | 0.956 | 0.087 |
| Shading w/o CR* | Shading | 22.957 | 0.897 | 0.208 |
| Predicted albedo | Shading | 14.114 | 0.744 | 0.295 |
| | Full | 34.278 | 0.944 | 0.126 |

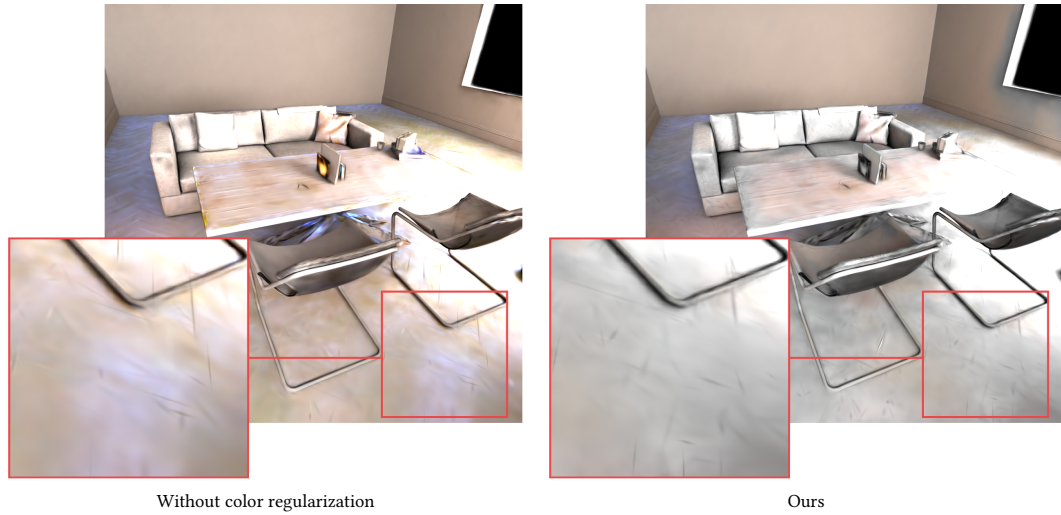


Fig. 12. Shading field reconstruction with color regularization turned off/on. Our shading reconstruction is more constrained while remaining colorful.

Replacing ground truth albedo maps with predicted albedo maps gives a blurrier albedo reconstruction, which makes it harder to reconstruct shading and residual layers free of albedo variations (Fig. 13). We tried several other options, but the multi-view consistent albedo maps from DiffusionRenderer are currently the best option, even if quite blurry. Moreover, we observed that DiffusionRenderer does not produce linear outputs. While the gamma inversion we apply compensates for this nonlinearity, it is only an approximation and some color shift sometimes remains.

6 Limitations

While our method allows a reasonable intrinsic decomposition of 3D Gaussian Splatting, it comes with some limitations.

The first limitation is the “fuzzy” nature of the 3DGS representation itself, that makes editing complicated. We have demonstrated a practical solution for edits on planar surfaces, requiring some user interaction. This could be extended to simple non-planar shapes by fitting suitable proxies (e.g., cylinders or surfaces of revolution), but providing a truly general solution would be more complex. Methods that compute meshes from 3D Gaussian splats (e.g., [Guédon et al. 2025; Guédon and Lepetit 2024]) could potentially be used to initialize proxies, but accurate edits would still be



Fig. 13. Predicted albedo images can lack details and exhibit a slight color shift due to unknown nonlinearities, which yields blurrier reconstructions of the albedo field (left) compared to the reconstruction obtained with exact albedo images (right).



Fig. 14. DiffusionRenderer tends to predict blurry albedo images (b). As a result, the high-frequency details present in the original images (a) are reconstructed by the shading field (d).

challenging. Our method also inherits popping artifacts already present in standard 3DGS reconstruction, most notably seen in darker regions. These could be reduced by sorting the primitives as in StopThePop [Radl et al. 2024]. The second limitation is our dependence on the quality of diffusion models to extract albedo layers. While the layers can be used in a useful manner as shown in our results on real scenes, the predictions are often blurry, and sometimes can assign texture to shading. An example is shown in Fig. 14. Future progress in intrinsic image and video decomposition will directly benefit our method. Finally, as previously mentioned in section 5 our method is slower than 3DGS but a caching mechanism could be implemented to avoid re-rendering pretrained layers when training. At inference time this caching mechanism cannot be used but smarter memory layout of the Gaussians representing different layers could be used to increase rendering speed, since the current implementation consists of calling the renderer 3 times with different sets of Gaussians.

7 Future Work and Conclusion

In future work, we would like to investigate how to build on our ideas to allow full inverse rendering for radiance fields. Our explicit decomposition is promising, but many challenges remain in terms of the actual representation and the capacity to perform light transport.

We have presented a new method for intrinsic decomposition of 3D Gaussian splats, using diffusion model priors for albedo and depth. We introduce a new approach that first predicts an *albedo field*, and then jointly optimizes a *shading* and *residual* field. This intrinsic decomposition of 3D Gaussian splats allows easy editing of albedo and texture, while maintaining the appearance of shading, allowing for realistic edited results.

Acknowledgments

This work was funded by the European Research Council (ERC) Advanced Grant NERPHYS, number 101141721 <https://project.inria.fr/nerphys>. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the EU or the European Research Council. Neither the EU nor the granting authority can be held responsible for them. The authors are grateful to the OPAL infrastructure of the Université Côte d’Azur for providing resources and support, as well as Adobe and NVIDIA for software and hardware donations. Experiments presented in this paper were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

References

- Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. 2017. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of the Art Reports)* 36, 2 (2017).
- Nicolas Bonneel, Kalyan Sunkavalli, James Tompkin, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. 2014. Interactive Intrinsic Video Editing. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 33, 6 (2014).
- Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. 2021a. NeRD: Neural Reflectance Decomposition from Image Collections. In *IEEE International Conference on Computer Vision (ICCV)*.
- Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. 2021b. Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Adrien Bousseau, Sylvain Paris, and Frédo Durand. 2009. User Assisted Intrinsic Images. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 28, 5 (2009).
- Chris Careaga and Yağız Aksoy. 2023. Intrinsic image decomposition via ordinal shading. *ACM Transactions on Graphics* 43, 1 (2023), 1–24.
- Chris Careaga and Yağız Aksoy. 2024. Colorful diffuse intrinsic image decomposition in the wild. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–12.
- Hongze Chen, Zehong Lin, and Jun Zhang. 2025. GI-GS: Global Illumination Decomposition on Gaussian Splatting for Inverse Rendering. In *ICLR*.
- Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. 2024. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 811–820.
- Kang Du, Zhihao Liang, Yulin Shen, and Zeyu Wang. 2025. GS-ID: Illumination Decomposition on Gaussian Splatting via Adaptive Light Aggregation and Diffusion-Guided Material Priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 26220–26229.
- Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez-Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. 2015. Multi-View Intrinsic Images of Outdoors Scenes with an Application to Relighting. *ACM Transactions on Graphics* (2015). <http://www-sop.inria.fr/reves/Basilic/2015/DRCLLPD15>
- Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. 2023. Relightable 3D Gaussian: Real-time Point Cloud Relighting with BRDF Decomposition and Ray Tracing. *arXiv:2311.16043* (2023).
- Elena Garces, Jose I. Echevarria, Wen Zhang, Hongzhi Wu, Kun Zhou, and Diego Gutierrez. 2017. Intrinsic Light Field Images. *Computer Graphics Forum* 36, 8 (2017). doi:10.1111/cgf.13154
- Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. 2022. A Survey on Intrinsic Images: Delving Deep Into Lambert and Beyond. *International Journal in Computer Vision* (2022).
- Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. 2025. MiLo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* (2025).
- Antoine Guédon and Vincent Lepetit. 2024. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. *CVPR* (2024).
- Songru Jiang, Wei Huang, Tao Lu, Yuanqi Li, Jie Guo, Runze Fu, Yanwen Guo, Lijun Chen, Haiyang Bai, Jiaqi Zhu. 2025. GaRe: Relightable 3D Gaussian Splatting for Outdoor Scenes from Unconstrained Photo Collections. *ICCV* (2025).
- Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. 2024. GaussianShader: 3D Gaussian Splatting with Shading Functions for Reflective Surfaces. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 5322–5332. doi:10.1109/CVPR52733.2024.00509
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 42, 4 (July 2023).
- Peter Kocsis, Vincent Sitzmann, and Matthias Niessner. 2024. Intrinsic Image Diffusion for Indoor Single-view Material Estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).
- Zhengfei Kuang, Fujun Luan, Sai Bi, Zhixin Shu, Gordon Wetzstein, and Kalyan Sunkavalli. 2023. PaletteNeRF: Palette-Based Appearance Editing of Neural Radiance Fields. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012). doi:10.1145/2366145.2366221
- Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. 2025b. Diffusion Renderer: Neural Inverse and Forward Rendering with Video Diffusion Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 26069–26080.
- Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. 2025a. DiffusionRenderer: Neural Inverse and Forward Rendering with Video Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. 2024. GS-IR: 3D Gaussian Splatting for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21644–21653.
- Jundan Luo, Duygu Ceylan, Jae Shin Yoon, Nanxuan Zhao, Julien Philip, Anna Frühstück, Wenbin Li, Christian Richardt, and Tuanfeng Wang. 2024. IntrinsicDiffusion: Joint Intrinsic Layers from Latent Diffusion Models. In *ACM SIGGRAPH Conference Papers*. Article 74, 11 pages.

- NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezani, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. 2025. Cosmos World Foundation Model Platform for Physical AI. *arXiv preprint arXiv:2501.03575* (2025).
- Julien Philip, Sébastien Morgenthaler, Michaël Gharbi, and George Drettakis. 2021. Free-viewpoint Indoor Neural Relighting from Multi-view Stereo. *ACM Transactions on Graphics* (2021). <http://www-sop.inria.fr/revs/Basilic/2021/PMGD21>
- Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. 2024. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–17.
- Lorenzo Rutayisire, Nicola Capodieci, and Fabio Pellacini. 2025. ReCoGS: Real-time ReColoring for Gaussian Splatting scenes. *arXiv preprint arXiv:2511.18441* (2025).
- Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Markus Schütz, Christoph Peters, Florian Hahlbohm, Elmar Eisemann, Marcus Magnor, and Michael Wimmer. 2025. Splatshop: Efficiently editing large Gaussian splat models. In *Computer Graphics Forum*. Wiley Online Library, e70214.
- Ishaan Shah, Andreas Meuleman, Alexandre Lanvin, and George Drettakis. [n. d.]. *GraphDeco Viewer*. <https://github.com/graphdeco-inria/graphdecoviewer>
- Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kenji Tojo and Nobuyuki Umetani. 2022. Recolorable Posterization of Volumetric Radiance Fields Using Visibility-Weighted Palette Extraction. *Computer Graphics Forum* 41, 4 (2022), 149–160.
- Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything V2. *arXiv:2406.09414* (2024).
- Keyang Ye, Qiming Hou, and Kun Zhou. 2024. 3D Gaussian Splatting with Deferred Reflection. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*. 1–10. arXiv:2404.18454 [cs] doi:10.1145/3641519.3657456
- Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. 2023. IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Yunlong Yuan, Yuanfan Guo, Chunwei Wang, Hang Xu, and Li Zhang. 2025. Brick-Diffusion: Generating Long Videos with Brick-to-Wall Denoising. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024. RGBX: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH Conference Papers*. Article 75, 11 pages.
- Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. 2021. NeRFactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics* 40, 6 (2021). doi:10.1145/3478513.3480496