Doctoral Dissertation
Doctoral Program in Eletrical, Electronic and Telecommunications (38^{th} cycle)

Deep Learning for 3D World Representations

Resolution, Reliability, and Remote Sensing

Luca Savant Aira

* * * * *

Supervisors
Prof. Enrico Magli, Supervisor
Prof. Diego Valsesia, Co-Supervisor
Prof. Giulia Fracastoro, Co-Supervisor

Doctoral examination committee
Prof. Marco Grangetto, Università di Torino
Prof. Enzo Tartaglione, Télécom Paris, Institut Polytechnique de Paris
Prof. Pietro Zanuttigh, Università di Padova
Prof. Lia Morra, Politecnico di Torino
Prof. Alessandro Rizzo, Politecnico di Torino

Politecnico di Torino
January 30, 2026

I hereby declare that the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.
This dissertation is presented in partial fulfillment of the requirements for Ph.D. degree in the Graduate School of Politecnico di Torino (ScuDo).

. . . . . .

Luca Savant Aira
Torino, January 30, 2026

Summary

This dissertation studies volumetric representations of 3D scenes that can be reconstructed from real sensor data and rendered from arbitrary viewpoints. It concentrates on radiance-field-style models such as Neural Radiance Fields and 3D Gaussian Splatting, which encode geometry and appearance as continuous fields of density and color and are optimized end-to-end via differentiable volumetric rendering. A preliminary chapter revisits light transport and the volume rendering integral, and proves analytically how these classical concepts map to modern neural formulations and fixing notation for radiance, color, density, cameras and rendering operators.

On this foundation, the dissertation addresses four main challenges. The first is multi-image super-resolution (MISR) with large viewpoint changes, where traditional methods based on optical flow in the image plane break down. The proposed EpiMISR deep neural network replaces optical flow with explicit epipolar geometry, implicitly building 3D feature fields akin to Neural Radiance Fields but without per-scene optimization. This geometry-aware design handles arbitrary numbers of views, gracefully falls back to single-image SR, and yields clear gains over prior MISR approaches in settings with strong parallax.

The second challenge is reliability, in particular how to attach meaningful uncertainty estimates to reconstructions obtained with Gaussian Splatting. The dissertation introduces Stochastic Gaussian Splatting, which turns each Gaussian primitive into a Bayesian random variable. Rendering becomes stochastic, and Monte-Carlo evaluation produces both an expected image and a per-pixel predictive variance. A new loss term encourages these variances to correlate with true errors, leading to calibrated uncertainty maps, while preserving the speed and quality advantages of 3D Gaussian Splatting.

The third challenge is to cope with missing or extremely sparse viewpoints. To regularize volumetric models in such regimes, the thesis explores physics-based priors on motion via MotionCraft, a zero-shot video generator that operates in the latent space of a pretrained image diffusion model. The method is parameter-free, works from a single input image, and improves over existing zero-shot baselines on complex motions. Within the broader thesis, this line of work shows how physics-based latent warping can serve as a strong prior for imagining plausible dynamics

and unseen viewpoints, complementing static volumetric reconstructions.

The fourth challenge is efficiency and scalability in real application, such as in satellite remote sensing, where multi-view high-resolution image collections must be processed under tight computational budgets. The Earth-Observation Gaussian Splatting (EOGS) framework adapts Gaussian Splatting to satellite photogrammetry by combining remote-sensing-specific ingredients such as radiometric corrections and physically motivated shadow modeling. Experiments show that EOGS attains reconstruction accuracy comparable to state-of-the-art NeRF-based Earth-observation methods while requiring orders of magnitude less training time, making volumetric radiance fields practical for high-throughput satellite pipelines.

Altogether, the dissertation shows that radiance-field-based scene representations can be adapted to address accuracy, reliability, data efficiency and scalability in a range of settings. Through four complementary but distinct methods, it advances our understanding of how physically grounded rendering, geometric constraints, Bayesian modeling and generative priors can be combined with volumetric models. The resulting contributions improve multi-image super-resolution, provide uncertainty estimates for Gaussian Splatting, enable plausible dynamics from sparse observations, and make radiance-field-style approaches more practical for large-scale remote-sensing pipelines.

Acknowledgements

Il lavoro che state leggendo è il risultato di un percorso triennale che non sarebbe mai potuto avvenire se molte persone a me care non mi fossero state accanto. Innanzi tutto ringrazio, ma soprattuto abbraccio, i miei genitori per essermi stati accanto anche durante questi tre anni e per avermi ascoltato e supportato sempre. Loro mi hanno introdotto, tra le tante altre cose, alla curiosità, che è ciò che mi ha guidato, e non spinto, durante questi tre anni. Ringrazio Vale che, oltre ad avermi supportato, che mi ha anche sopportato. Che fortuna aver incontrato una persona con lo stesso tipo di pazzie: gli svuotatasche di tappi riciclati, il poster data-viz della nostra chat, il porta-rubinetto stampato in 3d. Dopo 9 anni, penso davvero che tu meriteresti una laurea in ingegneria ad-honorem! Ringrazio i miei nonni che non hanno mai smesso, e mai smetteranno, di pregare per me. Ringrazio anche i miei cugini, che considero fratelli, e zia Pippi per essermi stati vicini ogni weekend. Grazie ai miei amici che, con la scusa dell'appuntamento settimanale di D&D, sono sempre stati presenti durante questo mio percorso. In particolare ringrazio Marco, collega e amico caro, con cui ho discusso gli avanzamenti della mia vita lavorativa e personale; nonchè coinquilino a Londra, battendo Vale sul tempo! Non per ultimi vorrei ringraziare i miei professori e tutori Enrico e Diego per avermi introdotto nel mondo della ricerca e per essere stati sempre disponibili a confrontarsi con me, anche nelle pause pranzo! Grazie anche a Gabriele e Thibaud per avermi ospitato calorosamente e guidato durante il mio periodo di ricerca all'École normale supérieure di Parigi. Grazie anche ad Alessandro e ai due Andrea che mi hanno seguito durante il periodo a Londra.

Infine, questa tesi di dottorato è stata redatta a conclusione del percorso di dottorato finanziato a valere sul PNRR, Missione 4, componente 2 "Dalla Ricerca all'Impresa" - Investimento 3.3 "Introduzione di dottorati innovativi che rispondono ai fabbisogni di innovazione delle imprese e promuovono l'assunzione dei ricercatori dalle imprese", tramite il Decreto Ministeriale n. 352 del 9 aprile 2022.

Contents

List of Tables

List of Figures

Nomenclature

α	Opacity of a Gaussian Primitive
β	Diffusion Parameter
γ	a Gaussian Primitive
Γ	the set of all Gaussian Primitives
θ	parameters
κ	color field
λ	a constant fixed scaling factor
μ	center position of a Gaussian Primitive
ρ	density field
Σ	Shape of a Gaussian Primitive

Chapter 1

Imagining 3D Worlds

A central question in this thesis is how to represent and reconstruct a three-dimensional scene. By representation we mean an internal encoding that makes a scene computationally accessible: it specifies which quantities are defined in space (*e.g.*, geometry, reflectance, volumetric density, radiance) and provides the operators that allow us to query and manipulate them. Informally, a good scene representation is one that allows us to *imagine* the scene from arbitrary viewpoints, this is called the *Novel View Synthesis* task in the literature. A 3D scene reconstruction pipeline then seeks to solve the inverse problem of inferring such a representation from a finite set of sensor measurements, most commonly images, depth maps, or point clouds. The design of the representation is driven by the intended tasks (*e.g.*, rendering, editing, simulation and navigation) and by the constraints of available hardware. The increasing demand for high-fidelity 3D models in computer graphics, virtual and augmented reality, scientific visualization, and autonomous systems makes the problem of choosing appropriate scene representations both practically urgent and scientifically interesting.

Historically, the evolution of scene representations has been tightly coupled with the development of input/output hardware. The roots of computer graphics are usually traced back to the 1960s, when William Fetter and Verne L. Hudson at Boeing coined the term *computer graphics* while exploring the use of computers for cockpit design and human-figure studies [92]. In this period, images were generated on vector displays, which traced parametric curves directly on a cathode ray tube. These devices were naturally suited to line drawings and wireframe models, and early scene representations followed: constructive solid geometry (CSG) and other analytic primitives, often embedded in emerging computer-aided design (CAD) systems, described objects as combinations of idealized volumes. A core challenge at the time was the visibility problem: determining which parts of a geometric model should be seen from a given viewpoint. This was studied in pioneering works such as [128, 168] and eventually formalized and solved in the context of interactive 3D systems by Sutherland and colleagues [150]. In this early phase, 3D

scene representation was largely synonymous with sets of geometric primitives and their Boolean combinations, tailored to the capabilities and limitations of vector hardware.

The introduction of raster displays in the 1970s and 1980s radically changed both the graphics pipeline and the prevalent scene representations. Raster devices rendered images as arrays of pixels, which naturally encouraged polygonal approximations of surfaces. Representing a scene as a collection of polygonal meshes, equipped with per-vertex attributes and textures, became the dominant paradigm. This shift enabled the development of key techniques such as shading [53, 120], texture mapping [20], Z-buffering [26], and anti-aliasing [35]. In this era, a typical pipeline consisted of modeling, animation, and rendering: artists and engineers constructed polygonal models, animated them via kinematic rigs or keyframes, and finally rasterized and shaded them onto the screen. The underlying 3D scene representation (*i.e.*, polygonal meshes plus material parameters) was explicitly designed to feed this pipeline efficiently, and to match the memory and compute characteristics of emerging graphics hardware.

As requirements for realism grew, it became clear that local illumination models and simple surface descriptions could not capture complex lighting phenomena such as soft shadows, indirect illumination, and caustics. This led to the formulation of global illumination methods in the late 1970s and 1980s. Whitted's introduction of recursive ray tracing [170] provided a first physically motivated framework for specular reflections and refractions and Kajiya's rendering equation [79] unified these ideas as a formal description of light transport in scenes with arbitrary geometry and reflectance. Within this framework, a 3D scene representation is not only a collection of surfaces and materials, but also the domain of a light transport operator: it must support visibility queries, surface interaction models, and integration of radiance along paths. In practice, this pushed representations towards richer descriptions of geometry (*e.g.*, smooth surfaces, displacement maps) and appearance (*e.g.*, bidirectional reflectance distribution functions), while still largely staying within a surface-based view of the world.

In parallel, an alternative family of representations emerged that treat images themselves, rather than geometry, as the primary object. Light field cameras, plenoptic sampling, and related work in image-based rendering represent scenes as samples of the plenoptic function: a high-dimensional function that gives radiance for each ray in space. Techniques such as view interpolation, plenoptic modeling, and volume rendering can synthesize novel views directly from captured images or discretized radiance fields, sometimes with only implicit or coarse geometric structure. Here, the 3D scene representation shifts from explicit surfaces to higher-dimensional functions over ray space or volumetric grids, and rendering reduces to resampling and interpolating these functions. These ideas anticipate many of the modern neural representations of scenes, in which learned functions approximate radiance fields rather than explicit meshes.

More recently, the landscape has been reshaped by new classes of sensors and interaction devices. Virtual and augmented reality headsets, RGB-D cameras, LiDAR-equipped mobile devices, multi-view camera rigs, and high-resolution 3D scanners provide dense, multi-modal observations of the physical world. At the same time, motion capture systems and inertial sensors capture the dynamics of bodies and objects at high temporal resolution. As a consequence, the graphics pipeline has expanded from the traditional sequence of modeling, animation, and rendering to a richer loop that includes 3D scanning, motion capture, and image-based rendering. Scenes are increasingly "acquired" rather than hand-modeled, and downstream tasks (*e.g.*, view synthesis, scene understanding, interaction planning) depend on how effectively we can convert sensor data into usable scene representations. In this context, we must design representations that are not only expressive and efficient for rendering, but also amenable to being inferred from noisy, incomplete measurements.

The current wave of artificial intelligence, and in particular deep learning, offers tools to "learn" scene representations directly from images and other sensor data. Instead of relying solely on analytic reconstruction pipelines, we can parametrize scene representations with neural networks or other flexible function approximators, and fit them end-to-end to large collections of observations. This perspective recasts 3D scene reconstruction as a learning problem in which the representation, the rendering operator, and sometimes even the sensor model are jointly optimized. However, it also raises new questions: which representations are most compatible with gradient-based optimization? How can we ensure physical plausibility and generalization? How should we balance compactness, interpretability, and rendering efficiency?

In this thesis, we focus on *volumetric* representations of scenes, in which the entire three-dimensional space is modeled by continuous or discretized fields that "fill the volume". Such representations typically assign to each point in space (or each cell in a partition) quantities such as density, color, and possibly additional attributes, and define view-dependent appearance via the *volumetric integral*. This integral formalizes how a radiation field interacts with a medium that absorbs and emits radiation along a ray. Within this framework, both geometry and appearance are encoded implicitly in the parameters of the underlying fields, and rendering amounts to integrating these fields along camera rays.

From a computational standpoint, volumetric representations have been found empirically easier to learn from images: they offer a continuous, differentiable parameterization of both geometry and appearance and integrate seamlessly with gradient-based optimization. In contrast, traditional polygonal or mesh-based representations are inherently discrete, with visibility, rasterization, and shading implemented through piecewise operations and lookups. While recent differentiable rendering frameworks (*e.g.*, Mitsuba [74] and Nvdiffrast [88]) have significantly

improved our ability to optimize mesh-based scenes, they must still face discontinuities in visibility and topology changes. Volumetric scene representations avoid many of these issues, making them particularly attractive as a backend for 3D scene reconstruction.

The remainder of this thesis is structured as follows. Chapter 2 introduces the mathematical notation and theoretical foundations that underpin neural scene representations based on volumetric rendering. Its goal is to equip the reader with a clear understanding of the symbols and physical quantities used throughout the manuscript, and to make explicit the connection between classical light transport theory and modern neural radiance field formulations. Building on this common language, the subsequent chapters study volumetric scene representations under a single perspective: as learned *world models* inferred from incomplete observations and evaluated by their ability to support downstream visual tasks. The four main contributions can be seen as addressing four complementary desiderata of such models: quality, reliability, completeness, and efficiency. Concretely, one chapter is dedicated to leveraging volumetric representations for super-resolution, one to modeling their reliability and uncertainty, one to handling missing views, and one to achieving efficient, large-scale remote-sensing deployment.

The first challenge, addressed in Chapter 3, concerns *quality*: how to exploit volumetric scene representations in the inverse problem of super-resolution. In many practical settings we may only have access to low-resolution observations of a scene. We therefore investigate how a volumetric world model can be estimated from multiple such observations and then used to synthesize high-resolution images that are both geometrically consistent across views and photometrically faithful. Casting multi-image super-resolution as the estimation of a joint 3D volumetric representation allows us to go "beyond optical flow" formulations that operate purely in the image plane.

The second challenge concerns *reliability* of learned volumetric representations: how much can we trust the predictions they produce? As these models are increasingly used in safety-critical or scientifically demanding applications, it becomes essential to quantify their uncertainty. In Chapter 4, we focus in particular on uncertainty modeling for Gaussian Splatting, a popular volumetric representation that describes scenes as collections of anisotropic Gaussian primitives. We propose probabilistic extensions that endow these primitives with uncertainty estimates, allowing us to capture aleatoric components of the reconstruction error and to propagate them to downstream tasks.

The third challenge, addressed in Chapter 5, concerns *completeness*: how to deal with missing or severely sparse viewpoints, a regime that is common whenever acquisition is expensive, constrained, or partially corrupted. Volumetric representations, while expressive, can overfit the observed views and behave unpredictably in unobserved regions if not properly regularized. We therefore study how to incorporate physically grounded priors on motion and dynamics into the volumetric

modeling framework, so that the representation can plausibly *imagine* the scene evolution and viewpoints that were never directly observed, while remaining consistent with the measurements that are available. In particular, we explore physics-based, zero-shot generative models that synthesize temporally coherent videos and novel views from minimal input, effectively compensating for missing data through a learned prior over plausible motions and interactions.

The fourth challenge concerns *efficiency*: how to make volumetric representations and their associated rendering and learning procedures fast and scalable enough to cope with the ever growing rate of data production and the large spatial extent of real-world scenes. This is particularly critical in domains such as remote sensing, where satellites continuously acquire massive amounts of imagery over wide geographic areas. We investigate how volumetric techniques can be adapted to large-scale photogrammetry pipelines for satellite images, striking a balance between reconstruction accuracy, rendering quality, and computational cost. These contributions are presented in Chapter 6.

1.1 Publications

The following publications were developed during the course of this PhD and form the basis of the chapters in this thesis:

- [136] Luca Savant Aira, Diego Valsesia, Andrea Bordone Molini, Giulia Fracastoro, Enrico Magli, and Andrea Mirabile. "Deep 3D World Models for Multi-Image Super-Resolution Beyond Optical Flow". In: *IEEE Access* 12 (2024).

- [135] Luca Savant Aira, Diego Valsesia, and Enrico Magli. "Modeling Uncertainty for Gaussian Splatting". In: *IEEE Transactions on Neural Networks and Learning Systems* 36.6 (2025).

- [112] Antonio Montanaro, Luca Savant Aira, Emanuele Aiello, Diego Valsesia, and Enrico Magli. "MotionCraft: Physics-Based Zero-Shot Video Generation". In: *Advances in Neural Information Processing Systems – NeurIPS* 37 (2024).

- [134] Luca Savant Aira, Gabriele Facciolo, and Thibaud Ehret. "Gaussian Splatting for Efficient Satellite Image Photogrammetry". In: *Proceedings of the Computer Vision and Pattern Recognition Conference – CVPR* (2025).

Chapter 2

A Primer on Volume Rendering

In this chapter we introduce the mathematical notation and the theoretical foundations that underpin neural scene representations based on volumetric rendering. The objective is to provide the reader with a clear understanding of the symbols and physical quantities used throughout the thesis, as well as to establish the link between classical light transport and modern neural radiance field formulations. In particular, we provide here a unified mathematical treatment that bridges classical radiative transfer, Neural Radiance Fields (NeRF), and 3D Gaussian Splatting (3DGS). The notation and derivations are chosen to make the connections explicit and to highlight the approximations and numerical choices that distinguish each method. To the best of our knowledge, this unified presentation is not commonly stated in this compact form in the literature; presenting it helps compare algorithms and transfer insights between families of methods.

Throughout this chapter we will use this unified framework to derive the rendering equations used by NeRF and 3DGS, and to point out the specific assumptions (approximations, sampling, discretization) that lead from the general radiative-transfer integral to each practical algorithm.

2.1 Notation

We denote vectors by underlined lowercase letters, for example $\underline{x} \in \mathbb{R}^3$ or $\underline{\mu} \in \mathbb{R}^n$, and matrices by double-underlined uppercase letters, for example $\underline{\underline{A}} \in \mathbb{R}^{3,4}$ or $\underline{\underline{\Sigma}} \in \mathbb{R}^{n,n}$. The transpose of a matrix $\underline{\underline{A}}$ is indicated with the superscript $'$, for example $\underline{\underline{A}}'$. We will also take the "transpose of a vector", meaning that $\underline{x}' \in \mathbb{R}^{1,n}$ is a row vector.

With a small abuse of notation, the symbol L^1 will indicate the mean absolute error between two vectors, *i.e.*, for $\underline{x}, \underline{y} \in \mathbb{R}^n$:

$$L^1(\underline{x}, \underline{y}) = \frac{||\underline{x} - \underline{y}||_1}{n} = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|.$$

When reporting experimental results in tables, we will indicate the best result in **bold** and the second best with underline.

2.1.1 Images

As we deal with scene reconstruction methods, we will always works with images and associated cameras. From the mathematical standpoint, images are defined as functions $\mathbb{R}^2 \to \mathbb{R}^c$. The domain $\mathbb{R}^2$ represents the coordinates of points in the image plane, while the codomain $\mathbb{R}^c$ represents the color of each point, where c is the number of channels. For example, a grayscale image has $c = 1$, while an RGB image has $c = 3$. With an abuse of notation, we will call images also the "discrete" version of the function, theoretically obtained by integrating the continuous function over small square areas called *pixels*. We will denote an image with the uppercase letter I, so that the color of a point $\underline{u} \in \mathbb{R}^2$ or a pixel $\underline{u} \in \mathbb{Z}^2$ is given by $I(\underline{u}) \in \mathbb{R}^c$.

2.1.2 Camera Models

A camera model is a function $\mathbb{R}^3 \to \mathbb{R}^2$, *i.e.*, it maps a point from the 3D world to a point on the 2D *image plane*. We will denote a camera model with the calligraphic uppercase letter $\mathcal{C}$, so that a 3D point $\underline{x} \in \mathbb{R}^3$ is mapped to a 2D point $\underline{u} \in \mathbb{R}^2$ as $\underline{u} = \mathcal{C}(\underline{x})$.

In the rest of the thesis, we will mostly work with three different camera models: pinhole cameras, affine cameras, and pushbroom cameras.

A pinhole camera is the most common camera model in computer vision and graphics. It is defined by its intrinsic parameters $(\phi_x, \phi_y, \zeta, \delta_x, \delta_y)$, that define the camera internal characteristics, and its extrinsic parameters $\left(\underline{\tau} \in \mathbb{R}^3, \underline{\underline{\Omega}} \in \mathbb{R}^{3,3}\right)$ that define the camera position and orientation in the 3D world. The pinhole camera model is defined as:

$$\mathcal{C}(\underline{x}) = \pi \left(\begin{pmatrix} \phi_x & \zeta & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \underline{\underline{\Omega}} & \underline{\tau} \end{pmatrix} \begin{pmatrix} \underline{x} \\ 1 \end{pmatrix} \right) \tag{2.1}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the perspective division defined as:

$$\pi\left(\underline{x}\right) = \pi \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \frac{1}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{2.2}$$

It is useful to define the *camera center* $\underline{o} \in \mathbb{R}^3$ of a pinhole camera, that can be tought as the position of the "pinhole" in the 3D world. It is defined $\underline{o} = -\underline{\underline{\Omega}}\underline{\tau} \in \mathbb{R}^3$.

An affine camera is the simplest camera model, and it is defined by a full-rank matrix $\underline{\underline{A}} \in \mathbb{R}^{2,3}$ and a translation vector $\underline{a} \in \mathbb{R}^2$, so that the affine camera model is defined as:

$$\mathcal{A}(\underline{x}) = \underline{\underline{A}}\underline{x} + \underline{a}. \tag{2.3}$$

The *view direction* of an affine camera is the one of a unitary vector $\underline{d} \in \mathbb{R}^3$ such that $\underline{\underline{A}}\underline{d} = \underline{0}$.

A pushbroom camera is a camera that is often used in remote sensing applications, for example in satellite imaging. Its camera model is defined as a rational function, *i.e.*, a function that can be written in the form of a ratio of two polynomials. The two polynomials are taken to be cubic polynomials in the coordinates of the 3D point $\underline{x} = (x_1, x_2, x_3)'$. This camera model is also known as the *Rational Polynomial Camera* (RPC) model, and it is commonly used in satellite imaging because it can accurately model the complex distortions that are introduced by the satellite's motion and the Earth's curvature.

Inverse Camera Models

When dealing with scene reconstruction methods, inverse camera models allow to map a pixel from the image plane to a set of points in the 3D world. Since all camera models are not invertible functions by definition, as they map a 3D space to a 2D space, we can only consider the *full-inverse* $\mathcal{C}^{-1}$ of a camera model $\mathcal{C}$, that is a multivalued function that maps a point $\underline{u} \in \mathbb{R}^2$ in the image plane to its preimage in the 3D space:

$$\mathcal{C}^{-1}(\underline{u}) = \left\{\underline{x} \in \mathbb{R}^3 : \mathcal{C}(\underline{x}) = \underline{u}\right\}. \tag{2.4}$$

Note that commonly used camera models are such that preimages form simple one-dimensional manifolds in 3D space. For example, the preimage of a point $\underline{u} \in \mathbb{R}^2$ in a pinhole camera or in an affine camera is a straight line in 3D space, while the preimage of a point $\underline{u}$ in a pushbroom camera is a open non-self-intersecting curve in 3D space. This enable us to easily parameterize the preimage of a point $\underline{u}$ using a single real parameter $t \in \mathbb{R}$, so that:

$$\mathcal{C}^{-1}(\underline{u}) = \left\{\underline{r}_{\underline{u}}(t) : t \in \mathbb{R}\right\}, \tag{2.5}$$

where $\underline{r}_{\underline{u}} : \mathbb{R} \to \mathbb{R}^3$ is called the *ray* of the point $\underline{u}$.

As shown in Fig. 2.1, the set of rays of a pinhole camera is the set of half-lines starting from the camera center and passing through the image plane. Similarly, for affine cameras, the set of rays is the set of lines parallel to the view direction of the camera. Instead, the rays of a pushbroom camera are curves, even if light propagates in straight lines in a homogeneous medium, due to the motion of the camera during the image acquisition.

Depth and localization are two important concepts when dealing with inverse camera models. The depth of a 3D point $\underline{x}$ is a scalar value that represents its

distance from the camera. Different camera models use different definitions of distance. For example, in pinhole cameras, depth is defined as the Euclidean distance from the camera center to the point in 3D space. In pushbroom cameras or affine cameras, depth is often defined as the distance from a reference plane to the 3D point.

The depthmap is a function $\mathbb{R}^2 \to \mathbb{R}_+$ that maps a pixel $\underline{u}$ in the image plane to the depth of its imagined 3D point $\underline{x}$.

2.2 Light Transport in Participating Media

In this section we give a clear and concise mathematical derivation of the optical phenomena described in [30] and [105], and we refer the reader to these sources for a complete treatment of the topic. The physical process under consideration is called *Radiative Transfer* and it addresses the question of how light interacts with a medium that fills the 3D space. More precisely, the goal is to model how a radiation field evolves as it propagates through a medium that both absorbs and emits radiation.

To fix ideas, consider a narrow "pencil" of radiation traversing a small volume element of the medium along a given direction. As the radiation travels, its intensity is reduced if matter absorbs it and increased if it is emitted into the same pencil. Over an infinitesimal segment of the path, the net variation in intensity is therefore obtained by "counting" the losses due to absorption and the gains due to emission. Aggregating these infinitesimal changes along the ray leads to the classical radiative transfer formulation.

We denote the physical light intensity field with I, which increases or decreases

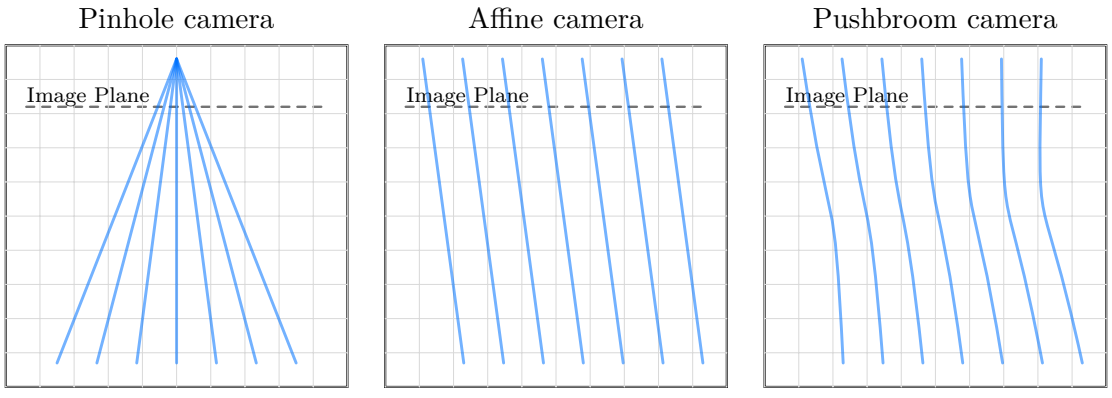


Figure 2.1: Rays of different camera models. From left to right: pinhole camera, affine camera, pushbroom camera.

along a ray segment, parameterized by x, by interacting with particles in a particle-filled volume. These interactions are described using the radiance field (ρ, κ), composed by non-negative density field ρ (also called the absorption coefficient) and the color field κ (also called the emission coefficient), with the following Cauchy problem:

$$\begin{cases} \dfrac{dI}{dx} = \rho(x)I(x) - \rho(x)\kappa(x) \quad \forall x \in [x_n, x_f] \\ I(x_f) = I_f \end{cases} \tag{2.6}$$

where $x_n < x_f$ are two points along the ray (near and far), and I_f is the boundary condition that represents the intensity of light coming from points outside the considered domain, *i.e.*, from $x > x_f$, for example from a background light source.

Proposition 1 Volumetric Integral

Consider the Cauchy problem defined by the differential Eq. (2.6) with the boundary condition $I(x_f) = I_f$ for some $x_f \in \mathbb{R}$ and $I_f \in \mathbb{R}$. The solution of the Cauchy problem at any point $x_n < x_f$ is given by the volumetric integral:

$$I(x_n) = e^{-\int_{x_n}^{x_f} \rho(y)dy} I_f + \int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(y)dy} \rho(x)\kappa(x)dx \tag{2.7}$$

Proof of Proposition 1

We start from the differential Eq. (2.6) and we multiply both sides by the quantity $\exp(\int_x^{x_f} \rho(y)dy)$:

$$\begin{aligned} &\iff e^{\int_x^{x_f} \rho(y)dy} \frac{dI(x)}{dx} = e^{\int_x^{x_f} \rho(y)dy} \left(\rho(x)I(x) - \rho(x)\kappa(x)\right) \\ &\iff e^{\int_x^{x_f} \rho(y)dy} \frac{dI(x)}{dx} - e^{\int_x^{x_f} \rho(y)dy} \rho(x)I(x) = -e^{\int_x^{x_f} \rho(y)dy} \rho(x)\kappa(x) \\ &\iff \frac{d}{dx}\left(e^{\int_x^{x_f} \rho(y)dy} I(x)\right) = -e^{\int_x^{x_f} \rho(y)dy} \rho(x)\kappa(x) \end{aligned}$$

where in the last step we used the product rule of the derivative.
Now integrate both side from x_n to x_f:

$$\begin{aligned} &\implies \int_{x_n}^{x_f} \frac{d}{dx}\left(e^{\int_x^{x_f} \rho(y)dy} I(x)\right) dx = -\int_{x_n}^{x_f} e^{\int_x^{x_f} \rho(y)dy} \rho(x)\kappa(x)dx \\ &\iff \left[e^{\int_x^{x_f} \rho(y)dy} I(x)\right]_{x=x_n}^{x=x_f} = -\int_{x_n}^{x_f} e^{\int_x^{x_f} \rho(y)dy} \rho(x)\kappa(x)dx \\ &\iff e^{\int_{x_f}^{x_f} \rho(y)dy} I(x_f) - e^{\int_{x_n}^{x_f} \rho(y)dy} I(x_n) = -\int_{x_n}^{x_f} e^{\int_x^{x_f} \rho(y)dy} \rho(x)\kappa(x)dx \end{aligned}$$

By rearranging and using exponential properties, we have:

$$\implies I(x_f) - e^{\int_{x_n}^{x_f} \rho(y)dy} I(x_n) = -\int_{x_n}^{x_f} e^{\int_{x}^{x_f} \rho(y)dy} \rho(x)\kappa(x)dx$$

$$\iff e^{-\int_{x_n}^{x_f} \rho(y)dy} I(x_f) - I(x_n) = -\int_{x_n}^{x_f} e^{-\int_{x_n}^{x_f} \rho(y)dy} e^{\int_{x}^{x_f} \rho(y)dy} \rho(x)\kappa(x)dx$$

$$\iff e^{-\int_{x_n}^{x_f} \rho(y)dy} I(x_f) - I(x_n) = -\int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(y)dy} \rho(x)\kappa(x)dx$$

Rearranging the equation and making use of the boundary condition $I(x_f) = I_f$ concludes the proof.

□

An intuitive explanation of Eq. (2.7) can now be given. The quantity $I(x_n)$ is the intensity of light at a point x_n along a fixed ray, which we can think of as the point where the ray intersects the camera sensor. The expression in Eq. (2.7) decomposes this intensity into two contributions. The first term, $e^{-\int_{x_n}^{x_f} \rho(y)dy} I_f$, represents the light that originates outside of the considered domain with intensity I_f (for example, a very distant source such as the sun) and then travels through the medium towards the camera. As it propagates from x_f to x_n, it is attenuated according to the cumulative density $\int_{x_n}^{x_f} \rho(y)dy$ encountered along the ray.

The second term, $\int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(y)dy} \rho(x)\kappa(x)\,dx$, accounts for the light that is *emitted* by the medium itself. Each point x between x_n and x_f contributes an amount of light proportional to the local emission $\rho(x)\kappa(x)$, which is then attenuated as it travels from x to the camera, with a factor $e^{-\int_{x_n}^{x} \rho(y)dy}$. The integral therefore sums up all such emitted contributions along the ray segment between x_n and x_f.

We remark that Eq. (2.7) contains two nested integrals: an outer integral that sums up the contributions from each point x along the ray, and an inner integral that computes the cumulative density from x_n to x for the attenuation factor. This is computationally challenging and motivates the following manipulations to simplify its numerical evaluation. The underlying radiative-transfer formulation is classical [30, 105], while the connection with Gaussian-based rendering follows ideas related to [184]. The partitioned formulation below and its use as a common lens to describe both NeRF [109] and 3DGS [82] are presented here as a unifying novel exposition.

Proposition 2 Partitioning the volumetric integral

Let's consider a partition of the interval $[x_n, x_f]$ into K subintervals:

$$[x_n, x_f] = [t_0, t_1] \cup [t_1, t_2] \cup \cdots \cup [t_{K-1}, t_K]$$

where $x_n = t_0 \leq t_1 \leq \cdots \leq t_{K-1} \leq t_K = x_f$. Supposing the color field κ can be

approximated as constant in each subinterval, *i.e.*, $\kappa(x)|_{x\in[t_{k-1},t_k]} \approx \kappa_k$, then the volumetric integral in Eq. (2.7) can be approximated as:

$$I(x_n) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} e^{-\mathcal{T}_j} \right) \left(1 - e^{-\mathcal{T}_k}\right) \kappa_k + I_f \prod_{k=1}^{K} e^{-\mathcal{T}_k} \tag{2.8}$$

where

$$\mathcal{T}_k := \int_{t_{k-1}}^{t_k} \rho(s) ds$$

is the *transmittance* of the medium in the k-th subinterval.

Proof of Proposition 2

Let's consider the volumetric integral in Eq. (2.7):

$$I(x_n) = e^{-\int_{x_n}^{x_f} \rho(t)dt} I_f + \int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(t)dt} \rho(x)\kappa(x)dx$$

Exploiting the linearity of the integrals and paying special attention to the "last bit" of the inner integral domain, we split all the integrals using the partition:

$$I(x_n) = e^{-\sum_{k=1}^{K} \int_{t_{k-1}}^{t_k} \rho(s)ds} I_f + \sum_{k=1}^{K} \left[\int_{t_{k-1}}^{t_k} e^{-\sum_{j=1}^{k-1} \int_{t_{j-1}}^{t_j} \rho(s)ds - \int_{t_{k-1}}^{t} \rho(s)ds} \kappa(t)\rho(t)dt \right]$$

Now we apply the properties of the exponential function:

$$I(x_n) = \sum_{k=1}^{K} \left[\int_{t_{k-1}}^{t_k} \left(\prod_{j=1}^{k-1} e^{-\int_{t_{j-1}}^{t_j} \rho(s)ds} \right) e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \kappa(t)\rho(t)dt \right] + I_f \prod_{k=1}^{K} e^{-\int_{t_{k-1}}^{t_k} \rho(s)ds}$$

We make use of the "constant-color" approximation, that assumes that the color is almost constant in a given interval, *i.e.*, $\kappa(t)|_{t\in[t_{k-1},t_k]} \approx \kappa_k$. Then we extract all the terms that do not depend on t from the integral:

$$I(x_n) = \sum_{k=1}^{K} \left[\kappa_k \left(\prod_{j=1}^{k-1} e^{-\int_{t_{j-1}}^{t_j} \rho(s)ds} \right) \int_{t_{k-1}}^{t_k} e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \rho(t)dt \right] + I_f \prod_{k=1}^{K} e^{-\int_{t_{k-1}}^{t_k} \rho(s)ds}$$

Now, recalling the long-gone calculus classes, we can notice that:

$$-\frac{d}{dt}\left(e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \right) = e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \rho(t),$$

so that, thanks to the *fundamental theorem of calculus*[2], it holds that:

$$\begin{aligned}\int_{t_{k-1}}^{t_k} e^{-\int_{t_{k-1}}^{t}\rho(s)ds}\rho(t)dt &= -\int_{t_{k-1}}^{t_k}\frac{d}{dt}\left(e^{-\int_{t_{k-1}}^{t}\rho(s)ds}\right)dt \\ &= -\left[e^{-\int_{t_{k-1}}^{t_k}\rho(s)ds} - e^{-\int_{t_{k-1}}^{t_{k-1}}\rho(s)ds}\right] \\ &= 1 - e^{-\int_{t_{k-1}}^{t_k}\rho(s)ds}.\end{aligned}$$

Hence, plugging this result back into the previous equation, we get:

$$I(x_n) = \sum_{k=1}^{K}\left[\kappa_k\left(\prod_{j=1}^{k-1} e^{-\int_{t_{j-1}}^{t_j}\rho(s)ds}\right)\left(1 - e^{-\int_{t_{k-1}}^{t_k}\rho(s)ds}\right)\right] + I_f\prod_{k=1}^{K} e^{-\int_{t_{k-1}}^{t_k}\rho(s)ds}$$

Making use of the symbol $\mathcal{T}_k = \int_{t_{k-1}}^{t_k}\rho(s)ds$ we conclude the proof.

□

Remarkably, Eq. (2.8) is analytically exact, except for the per-interval constant color approximation. We deem this approximation reasonable, as we are free to choose the partition granularity (that can be even non-constant and adaptive). Moreover Eq. (2.8) requires only a numerical scheme for the 1D integrals $\mathcal{T}_k$. This is a great simplification with respect to the original volumetric integral in Eq. (2.7), that requires a double integral computation.

We also remark that the volumetric integral in Eq. (2.8) has been derived for "one ray only", meaning that the variables x_n, x_f, ρ, κ are defined along a single ray. To extend this formulation to all the pixels of a camera, we can use the ray parameterization introduced in Section 2.1.2. For example, in an affine camera $\mathcal{C}$, given a pixel $\underline{u} \in \mathbb{R}^2$, we can denote its ray as $\underline{r}_{\underline{u}} : \mathbb{R} \to \mathbb{R}^3$. Let's also consider the unitary direction $\underline{d}_{\underline{u}}$ parallel to the ray. Then, we can rewrite the volumetric integral in Eq. (2.8) as:

$$I(\underline{u}) = \sum_{k=1}^{K}\left(\prod_{j=1}^{k-1} e^{-\mathcal{T}_j(\underline{u})}\right)\left(1 - e^{-\mathcal{T}_k(\underline{u})}\right)\kappa_k(\underline{d}_{\underline{u}}) + I_f\prod_{k=1}^{K} e^{-\mathcal{T}_k(\underline{u})} \tag{2.9}$$

where

$$\mathcal{T}_k(\underline{u}) = \int_{t_{k-1}}^{t_k}\rho\left(\underline{r}_{\underline{u}}(s)\right)ds$$

is the transmittance of the medium along the ray of pixel $\underline{u}$ in the k-th subinterval and, similarly,

$$\kappa_k(\underline{d}_{\underline{u}}) \approx \kappa\left(\underline{r}_{\underline{u}}(t), \underline{d}_{\underline{u}}\right) \quad \forall t \in [t_{k-1}, t_k]$$

is the color of the medium along the ray of pixel $\underline{u}$ in the k-th subinterval and in the direction $\underline{d}_{\underline{u}}$, approximated as a constant over the whole subinterval. We

remark that we extended the color field κ to depend also on the view direction $\underline{d}_{\underline{u}}$, to model view-dependent effects such as specular reflections. Although $\underline{d}_{\underline{u}}$ is introduced explicitly only at this stage, in the single-ray derivation we implicitly considered a fixed view direction associated with that ray, and thus we wrote κ as a 1D function along the ray. Indeed, for a given pixel $\underline{u}$ the ray $\underline{r}_{\underline{u}}$ has a constant direction. When we extend the formulation from one ray to all pixels, however, different pixels correspond to different rays (hence different $\underline{d}_{\underline{u}}$), so it becomes important to make this dependence explicit. In this sense the rendering equation is separable across pixels: each $I(\underline{u})$ is obtained by evaluating the same 1D volumetric integral independently along its own ray $\underline{r}_{\underline{u}}$ with its associated direction $\underline{d}_{\underline{u}}$.

2.3 Neural Radiance Fields

3D reconstruction from posed images is a long-standing problem in visual computing. Given a set of images of a static scene acquired from different viewpoints, together with their known camera models, the goal is to recover a 3D representation that faithfully captures the scene's geometry and appearance. Classical approaches typically rely on explicit geometric primitives, such as point clouds or meshes, and reconstruct surfaces via multi-view stereo or structure-from-motion pipelines. These methods, however, often struggle with complex materials, view-dependent effects, and incomplete or noisy observations. In parallel, the task of *Novel View Synthesis* (NVS) aims at rendering photorealistic images of the scene from previously unseen viewpoints. While 3D reconstruction and NVS are traditionally treated as separate problems, they are tightly coupled: a good 3D representation should enable accurate rendering from arbitrary views, and a method designed for NVS implicitly induces a 3D structure consistent with the observed images.

"NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis" [109] has arguably been the main driver of recent advances in both 3D reconstruction and NVS. The *Neural Radiance Fields* (NeRF) framework casts scene reconstruction as a statistical learning problem grounded in the volumetric rendering integral introduced in Proposition 1. NeRFs model a continuous volumetric representation of the scene, namely the two components of the radiance field (ρ, κ), and learn it directly from a set of posed images, by optimizing the parameters of a neural network so that the rendered views match the input photographs. Concretely, given N posed input views (which may be photos, video frames, satellite acquisitions, and so on), NeRF seeks a radiance field that explains all observations simultaneously. We remark that the input views should be posed, meaning that the camera models (for example, the intrinsic and extrinsic parameters for a pinhole camera) should be known in advance for each image, at least up to a certain accuracy. This radiance field can be queried from arbitrary viewpoints at test time, thereby unifying 3D reconstruction and NVS in a single differentiable rendering framework.

Formally, a scene is represented as a radiance field defined over 3D space and view directions. Since the input images are posed, the camera models are known, hence, we can associate to each pixel $\underline{u}_i$ its parametrized ray $\underline{r}_{\underline{u}_i}(t)$. Moreover, the corresponding pixel color $I(\underline{u}_i) = y_i \in \mathbb{R}^c$ is also known, as this is merely the color of the selected pixel in the input images. We consider the set of all available pixel-color pairs $\mathcal{D} = \{(\underline{u}_i, y_i)\}_i$ and we will use it as the training data for the NeRF statistical learning problem.

The learnable parameters of the NeRF model are the weights of two Multi-Layer Perceptrons (MLPs), briefly reviewed in Refresher 1, that represent the volumetric fields $\hat{\rho} \approx \rho : \mathbb{R}^3 \to \mathbb{R}_+$ and $\hat{\kappa} \approx \kappa : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^c$. Along each ray $\underline{r}_{\underline{u}}(t)$, the observed pixel color is modeled via the volumetric integral in Eq. (2.9), with two important assumptions.

- First, the integrals $\mathcal{T}_k(\underline{u})$ are approximated via Monte Carlo sampling, usually by drawing a single sample τ_k in each subinterval $[t_{k-1}, t_k]$.

$$\mathcal{T}_k(\underline{u}) \approx \hat{\rho}\left(\underline{r}_{\underline{u}}(\tau_k)\right)(t_k - t_{k-1}) \quad \text{with } \tau_k \sim \text{Uniform}(t_{k-1}, t_k)$$

- Second, the per-interval constant color $\kappa_k(\underline{d}_{\underline{u}})$ is approximated by querying $\hat{\kappa}$ at each sampled point along the ray.

$$\kappa_k(\underline{d}_{\underline{u}}) \approx \hat{\kappa}\left(\underline{r}_{\underline{u}}(\tau_k), \underline{d}_{\underline{u}}\right) \quad \text{with } \tau_k \sim \text{Uniform}(t_{k-1}, t_k)$$

Hence, NeRFs can be mathematically seen as a numerical scheme to compute the volumetric integral in Eq. (2.9).

We remark that $\hat{\rho} : \mathbb{R}^3 \to \mathbb{R}_+$ is the (non-negative) volume density field network and $\hat{\kappa} : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^c$ is the color field network. The dependence of $\hat{\kappa}$ on the view direction $\underline{d}_{\underline{u}}$ allows NeRFs to model view-dependent appearance phenomena, such as specular highlights and non-Lambertian reflections, within the same volumetric formulation.

On the whole, this yields a fully differentiable rendering process that maps the learnable parameters of the MLPs to the predicted pixel colors $\hat{I}(\underline{u})$ along each ray $\underline{r}_{\underline{u}}$, where $\hat{I}$ denotes the volumetric integral approximation using the MLPs $\hat{\rho}$ and $\hat{\kappa}$:

$$\hat{I}(\underline{u}) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} e^{-\Delta_j \hat{\rho}\left(\underline{r}_{\underline{u}}(\tau_j)\right)} \right) \left(1 - e^{-\Delta_k \hat{\rho}\left(\underline{r}_{\underline{u}}(\tau_k)\right)}\right) \hat{\kappa}\left(\underline{r}_{\underline{u}}(\tau_k), \underline{d}_{\underline{u}}\right) + I_f \prod_{k=1}^{K} e^{-\Delta_k \hat{\rho}\left(\underline{r}_{\underline{u}}(\tau_k)\right)} \tag{2.10}$$

where $\Delta_k := t_k - t_{k-1}$ and $\tau_k \sim \text{Uniform}(t_{k-1}, t_k)$ for all $k = 1, \ldots, K$.

Learning a NeRF thus amounts to fitting the parameters of these MLPs so that the rendered colors match the observed pixel colors in the training set, minimizing the following *reconstruction loss*:

$$\mathcal{L}_{\text{rec}} = \sum_i \mathcal{L}\left(\hat{I}(\underline{u}_i), y_i\right) \tag{2.11}$$

where $\mathcal{L}$ is a per-pixel regression loss, typically the L^1 or L^2 distance between predicted and ground truth colors. Because the rendering process is fully differentiable with respect to the parameters of the MLPs $\hat{\rho}$ and $\hat{\kappa}$, gradients can be backpropagated through the volumetric integral approximation and used to update the MLP weights via stochastic gradient descent.

Once trained, the NeRF can synthesize novel views by evaluating Eq. (2.10) for each pixel $\underline{u}$ on the new desired image plane. This yields highly detailed and photorealistic novel views that are consistent with the observed training images and with the inferred continuous 3D radiance structure.

To effectively represent high-frequency details in the scene, the input coordinates are typically mapped through positional encoding functions before being fed to the MLPs, enabling the networks to approximate complex radiance fields with relatively modest depth and width.

Refresher 1 Multilayer Perceptrons (MLPs)

Multi-Layer Perceptrons (MLPs) are a class of feedforward artificial neural networks that consist of multiple layers of neurons, where each layer is fully connected to the next one. MLPs are capable of learning complex functions by composing simple linear transformations with non-linear activation functions. Mathematically, an MLP with L layers can be represented as a composition of functions:

$$f(\underline{x}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(\underline{x})$$

where each layer $f^{(l)}$ is defined as:

$$f^{(l)}(\underline{z}) = g^{(l)}\left(\underline{\underline{W}}^{(l)}\underline{z} + \underline{b}^{(l)}\right)$$

Here, $\underline{\underline{W}}^{(l)}$ is the weight matrix, $\underline{b}^{(l)}$ is the bias vector, and $g^{(l)}$ is a non-linear activation function, such as ReLU, sigmoid, or tanh. The input $\underline{x}$ is transformed through each layer, allowing the MLP to learn complex mappings from input to output.

Building on the original NeRF formulation [109], a first practical limitation that emerged is its strong dependence on accurate camera poses. In many realistic acquisition settings (*e.g.*, handheld videos or casual photo collections), off-the-shelf structure-from-motion can be noisy or fail, and NeRF's reconstruction quality degrades significantly under pose errors. A line of research (BARF [98], NeRF - - [167], NoPe-NeRF [16], SCnERF [76]) directly addresses this issue by jointly optimizing the camera parameters and the radiance field.

A second major limitation of the original NeRF is the assumption of a static scene captured under consistent illumination and exposure. This assumption is often violated in *in-the-wild* settings, where images come from heterogeneous sources,

time periods, and imaging conditions, and may contain transient objects. Extensions such as NeRF-W [104], RAWNeRF [108], and Ha-NeRF [33] address these effects by modeling appearance variation, exposure changes, and other nuisance factors.

A related but distinct limitation is that the original NeRF formulation is restricted to static scenes. When the scene contains non-rigid motion or articulated objects, one must extend the representation to account for time or deformation. Representative examples include D-NeRF [124] and Nerfies [118].

From a rendering perspective, another inherent weakness of vanilla NeRF is aliasing: pixels are treated as infinitesimal ray samples, disregarding their finite footprint in the image plane. This leads to artifacts when zooming or rendering at resolutions different from those seen in training. Scale-aware variants (such as Mip-NeRF [10], Mip-NeRF360 [9], Zip-NeRF [11], ExactNeRF [72]) address this by integrating radiance over frustums instead of points, thus incorporating built-in anti-aliasing and enabling robust reconstruction of unbounded scenes. Other works, such as RegNeRF [116], mitigates aliasing artifacts in sparse and unbounded settings by introducing regularization priors on geometry and appearance.

Another line of work tackles a more structural limitation of standard NeRFs, namely the fact that they represent geometry implicitly via a volumetric density field, which makes it difficult to extract precise surfaces and to enforce geometric regularity. To obtain more faithful and controllable geometry, several methods replace the density field with a signed distance function (SDF) and derive a volume rendering formulation that is consistent with this implicit surface representation. Examples include NeuS [162], VolSDF [177], UNISURF [117], MonoSDF [179].

Finally, a practical challenge lies in the computational cost of per-scene NeRF optimization, which can require hours of training for a single scene in the original formulation. This has motivated a line of work (Plenoxels [44], DVGO [148], Instant-NGP [113]) on accelerating both representation and optimization. In this line of work, a central developement is the 3D Gaussian Splatting framework [82], which we describe in the next section.

2.4 3D Gaussian Splatting

The 3D Gaussian Splatting (3DGS) framework [82] is a recent approach introduced to directly address the computational inefficiencies of NeRF-like models, in terms of both training time and rendering speed. The similarities between 3DGS and NeRF are many: both methods address the problem of reconstructing a 3D scene from a set of posed images, they rely on the volumetric rendering integral introduced in Proposition 1, hence they can be viewed as numberical schemes to compute Eq. (2.9). However, the two methods differ significantly in how they represent the radiance field (ρ, κ): 3DGS represents it explicitly using a set of 3D

Gaussian-shaped profiles, as opposed to the implicit MLP representation of NeRF.

Before describing why it leads to significant computational advantages, we borrow notation from [24] and present a unified derivation of 3DGS that highlights its connections with the volumetric integral in Proposition 1. In particular, we describe mathematically how 3DGS numerically solves the volumetric integral in Eq. (2.9), that we rewrite here for clarity:

$$I(\underline{u}) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} e^{-\mathcal{T}_j(\underline{u})} \right) \left(1 - e^{-\mathcal{T}_k(\underline{u})}\right) \kappa_k(\underline{d}_{\underline{u}}) + I_f \prod_{k=1}^{K} e^{-\mathcal{T}_k(\underline{u})}$$

where $\mathcal{T}_k(\underline{u}) = \int_{t_{k-1}}^{t_k} \rho\left(\underline{r}_{\underline{u}}(s)\right) ds$ and $\kappa_k(\underline{d}_{\underline{u}})$ is a per-interval color that may depend on the view direction $\underline{d}_{\underline{u}}$.

While NeRF approximates the integrals $\mathcal{T}_k(\underline{u})$ using Monte Carlo sampling, 3DGS uses a different approach, based on the approximation of the radiant field using Gaussian primitives. A *Gaussian primitive* is a tuple $\gamma_k = \left(\underline{\mu}_k, \underline{\underline{\Sigma}}_k, \alpha_k, \underline{f}_k\right)$ representing a single Gaussian-shaped volume element in the scene, where $\underline{\mu}_k \in \mathbb{R}^3$ is the primitive center (mean), $\underline{\underline{\Sigma}}_k \in \mathbb{R}^{3,3}$ its 3D shape and orientation (covariance), $\alpha_k \in [0,1]$ its opacity, and $\underline{f}_k \in \mathbb{R}^d$ its feature vector, that can be decoded into color, usually via spherical harmonics, as we will discuss later.

We remark that the choice of using k to index both the Gaussian primitives and the partition intervals is not casual. In fact, in 3DGS the partition is "defined" by the Gaussian primitives themselves, meaning that each interval $[t_{k-1}, t_k]$ is such that the corresponding integral $\mathcal{T}_k(\underline{u})$ captures the one and only contribution of the k-th Gaussian primitive along the ray of pixel $\underline{u}$. In other words, 3DGS associates each partition interval to a Gaussian primitive, whose position and shape can be optimized, so that an adaptive partitioning scheme of the ray is obtained. Note that this is an approximation, as the contribution of a single Gaussian primitive should be non-zero in all the intervals along the ray. In other words, 3DGS assumes that primitive supports overlapping is negligible. However, this approximation is reasonable as the contribution decays exponentially with the distance from the primitive center.

Mathematically, 3DGS assumes that:

$$\rho\left(\underline{r}(t)\right)|_{t\in[t_{k-1},t_k]} \approx \alpha_k \mathcal{G}(\underline{r}(t); \underline{\mu}_k, \underline{\underline{\Sigma}}_k)$$

where

$$\mathcal{G}(\underline{x}; \underline{\mu}, \underline{\underline{\Sigma}}) := \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu})' \underline{\underline{\Sigma}}^{-1} (\underline{x} - \underline{\mu})\right).$$

Thanks to the marginalization property of the Gaussian distribution, we get $\mathcal{T}_k(\underline{u}) \approx \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u})$, with:

$$\mathcal{G}_k^{\mathcal{C}}(\underline{u}) := \mathcal{G}\left(\underline{u}; \mathcal{C}(\underline{\mu}_k), \underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right) \underline{\underline{\Sigma}}_k \left(\underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right)\right)'\right) \tag{2.12}$$

where $\mathcal{C} : \mathbb{R}^3 \to \mathbb{R}^2$ is the camera model and $\underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right) \in \mathbb{R}^{2,3}$ is the Jacobian of the camera model at point $\underline{\mu}_k$.

Moreover, 3DGS assumes that the contribution of each Gaussian is so small that the exponential terms in Eq. (2.9) can be approximated using a first-order Taylor expansion, leading to the following rendering equation:

$$\hat{I}(\underline{u}) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{C}}(\underline{u}) \right) \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \tilde{\underline{f}}_k + I_f \prod_{k=1}^{K} 1 - \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \tag{2.13}$$

where the color field κ has been replaced by the color contribution $\tilde{\underline{f}}_k$ of the k-th Gaussian primitive. This color contribution is typically defined as a view-dependent function of the feature vector $\underline{f}_k$ of the Gaussian primitive, *i.e.*,

$$\tilde{\underline{f}}_k = \sum_{l=0}^{L} \sum_{m=0}^{l} c_{klm} Y_{lm}(\underline{d}_{\underline{u}}), \tag{2.14}$$

where Y_{lm} are the spherical harmonics basis functions (following standard practices [44, 113]), L is the maximum degree of the spherical harmonics expansion, and c_{klm} are typically just the components of the feature vector $\underline{f}_k$ of the Gaussian primitive γ_k reshaped appropriately.

The intuitive description of the operations in Eq. (2.13) is the following. To render a view, 3DGS "splats" each 3D Gaussian kernel onto the camera image plane, defined by the camera model $\mathcal{C}$. This process is called the *splatting* operation and it associates a Gaussian primitive γ_k to the 2D Gaussian kernel $\mathcal{G}_k^{\mathcal{C}} : \mathbb{R}^2 \to \mathbb{R}$. During the splatting operation, each Gaussian is projected according to the first-order approximation of the camera model $\mathcal{C}$ computed at $\underline{\mu}_k$, $\underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right)$. In this way, the mean vector and covariance matrix of the 2D Gaussian kernel are:

$$\underline{\mu}_k^{\mathcal{C}} = \mathcal{C}(\underline{\mu}_k) \qquad \underline{\underline{\Sigma}}_k^{\mathcal{C}} = \underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right) \underline{\underline{\Sigma}} \left(\underline{\underline{J}}^{\mathcal{C}}\left(\underline{\mu}_k\right)\right)'. \tag{2.15}$$

Once all the primitives are splatted, they are sorted front-to-back with respect to the camera reference. This is needed because the outer sum in the original Eq. (2.9) was done on a sorted partition of the ray. Then, they are aggregated using the traditional alpha compositing, described in more detail in Refresher 2, accounting also for the Gaussian kernel decay, obtaining Eq. (2.13).

Similar to NeRF, now that we have an image formation model via Eq. (2.13), 3DGS learns its parameters (*i.e.*, the set of Gaussian primitives $\{\gamma_k\}_{k=1}^{K}$) by minimizing a reconstruction loss between the rendered images and the ground truth input images. Thanks to the efficient splatting operation, the entire estimated image $\hat{I}$ is available during 3DGS training phase, along with the real observed image I. We want to stress how much more computationally efficient this formulation with respect to the NeRF one, where only random batches of pixels are rendered at each training step. This efficiency gain is mainly due to the fact that 3DGS can

leverage highly optimized GPU rasterization pipelines for the splatting operation while NeRF needs to render each pixel independently via ray marching. From a learning perspective, 3DGS can exploit the spatial correlations between neighboring pixels in the loss function, while NeRF typically treats each pixel independently. Hence the reconstruction loss used in 3DGS has an extra term with respect to the NeRF one (Eq. (2.11)), leading to the following formulation:

$$\mathcal{L}_{\text{GS}}(\hat{I}, I) = \frac{4}{5} L^1(\hat{I}, I) + \frac{1}{5}\left(1 - \text{SSIM}(\hat{I}, I)\right) \tag{2.16}$$

where L^1 is the mean absolute error between the rendered and ground truth images, and $\text{SSIM}(\hat{I}, I) \in [-1,1]$ is the Structural Similarity Index Measure (SSIM) between the rendered and ground truth images, which captures perceptual differences more effectively than pixel-wise losses alone [166].

Refresher 2 Alpha Compositing

Alpha compositing is a technique used in computer graphics to combine multiple images or layers into a single image, taking into account the transparency (alpha) of each layer. The alpha value represents the opacity of a pixel, where an alpha of 1 means fully opaque and an alpha of 0 means fully transparent. The basic formula for alpha compositing two layers, A and B, is given by:

$$C = A \cdot \alpha_A + B \cdot (1 - \alpha_A)$$

where C is the resulting color after compositing, A is the color of the top layer, B is the color of the bottom layer, and α_A is the alpha value of layer A. This formula can be extended to composite multiple layers by iteratively applying the compositing operation from the topmost layer to the bottommost layer.

Recent literature has extended the original 3DGS [82] in several directions. A first line of work revisits the rendering model itself. Since the original method assumes pinhole cameras and relies on a first-order local approximation of the projection, subsequent works consider alternative camera models (*e.g.*, fisheye [96]) and more accurate splatting or projection schemes [69]. A second line of work studies more challenging input regimes. In particular, several methods target sparse-view reconstruction [174], while others relax the assumption of known camera parameters and jointly address pose estimation and scene reconstruction [45]. These directions mirror, in the Gaussian-splatting setting, some of the robustness concerns that had previously emerged in the NeRF literature. A third active direction concerns appearance control and physically richer rendering. Recent methods investigate editable or controllable texture, relighting, and illumination-aware Gaussian representations [95, 49, 140, 77]. More broadly, the literature is rapidly expanding toward dynamic scenes, semantics, editing, and generation. Further references and taxonomies can be found in recent surveys [173, 7].

Chapter 3

Resolution

Super-resolution is a classic pop culture trope, but it is also a challenging inverse problem in imaging.

From the Blade Runner 1982 film, where the protagonist Deckard examines a photo and "enhances" it to partially see a plot-revealing detail in a mirror reflection, everybody is at least familiar with the idea of super-resolution. In practice, single-image super-resolution (SISR) consists in recovering a high-resolution image from a low-resolution one. However, the problem is highly ill-posed, as many high-resolution images can correspond to the same low-resolution observation. Modern deep learning approaches have made significant progress in this field, by exploiting prior knowledge about natural images (*i.e.*, the typical pattern of hairs, grass, textures *etc.*) in order to "choose" the most plausible high-resolution image.

Fittingly, after super-resolving the image, Deckard navigates inside the 3D space of the image to change the viewpoint and see the entirety of the mirror reflection, solving the case. This opens the door to the use of 3D geometry in super-resolution. Multi-image super-resolution (MISR) methods seek to recover a high-resolution image from multiple low-resolution images of the same scene, by understanding the 3D geometry of the scene and the acquisition process.

In this chapter, we present EpiMISR, a novel multi-image super-resolution method that can handle images acquired from arbitrary camera positions and orientations by leveraging epipolar geometry and transformer-based processing of radiance feature fields. Our method significantly improves over state-of-the-art techniques in scenarios with large disparities among low-resolution images.

3.1 Introduction

Image super-resolution (SR) is the task of recovering a high-resolution (HR) version of an image from degraded low-resolution (LR) observations. It is a longstanding inverse problem in the imaging field and has numerous practical applications

due to camera limitations and image acquisition conditions. Most of the literature focuses on estimating the HR image from a single input image (SISR). While recent deep learning approaches [94, 86, 32] have tremendously advanced the state of the art, SISR remains highly ill-posed due to the limited high-frequency information available in a single image. Multi-image SR methods (MISR), on the other hand, are presented with multiple samplings of a given scene, carrying complementary information at a sub-pixel level. MISR techniques seek to accurately fuse the multiple LR images to obtain SR images with significantly higher quality than what is achievable by SISR methods. Only recently the deep learning literature has started exploring the multi-image setting due to increased difficulty in creating benchmark datasets as well as developing effective methods that can handle accurate image registration.

MISR can be seen as a generalization of the classic Stereo-SR setting [34], in which a pair of images is captured, often with a tightly controlled geometry to simplify the fusion process. At the moment, the most studied MISR settings are in the context of video [165] where successive frames provide the multiple images, remote sensing images [111] where satellite revisits of the same scene are exploited, and burst photography, where a set of photos is acquired in rapid succession such in [14], [91] or [100]. All these settings present a common denominator in that variations in the acquisition geometry among the multiple images are relatively small, resulting in relatively small disparities in the image pixels. For example, in burst SR, geometric variations are mostly due to natural hand shaking. This is desirable because the SR process requires subpixel shifts in the sampling grid, and obtaining them with minimal overall movement only simplifies the fusion process. For this reason, works in this field resort on using forms of optical flow estimation between LR images to accurately register them. Optical flow estimates a translation vector for each pixel of an image in order to warp it to a target image. Such a transformation between flat camera planes may struggle in presence of complex 3D transformations.

It is thus clear that the aforementioned small-parallax settings that have been currently studied are restrictive and do not allow to account for many interesting scenarios for super-resolution where the LR images come from cameras with wildly different positions and orientations. As examples, one can think of sets of security cameras which image a scene from significantly different vantage points, or sets of images of a scene collected in the wild with no control over the acquisition process.

In this chapter, we present EpiMISR, a new method designed for the general MISR setting, where a set of LR images are acquired by cameras with arbitrary positions and orientations, and our task is to super-resolve one (or more) of them. We move away from the optical flow based models, in favour of an explicit use of epipolar geometry with techniques inspired by recent works in the NeRF literature [109]. However, contrary to the NeRF literature, we are not concerned with novel view synthesis, but rather follow the standard SR approach of restoring one of

the observed LR images. Our proposed method, called EpiMISR, leverages strong spatial priors necessary for the SR task and transfomer-based processing of radiance feature fields to achieve effective fusion of images with large discrepancies in acquisition geometries. We show that EpiMISR substantially improves over the state-of-the-art SR techniques developed for the more restrictive scenarios.

3.2 Related Work

3.2.1 Single-Image Super-Resolution

Single-image super-resolution (SISR) is a long-standing problem in the field of computer vision, aiming at recovering a high-resolution (HR) image I^{HR} given its degraded version I^{LR}. In its simplest form, the forward model of the problem is:

$$I^{\mathrm{LR}} = (D * I^{\mathrm{HR}}) \downarrow_s \tag{3.1}$$

where $\downarrow_s$ denotes decimation by a factor s and $*$ denotes a convolution with degradation kernel D.

Note that this problem is ill-posed as the degradation process is non-injective. To overcome this challenge, two main families of approaches have been proposed: regularization methods and data-driven methods. Regularizers such as total variation impose handcrafted a-priori knowledge to establish a criterion in order to choose a plausible SR image, as done by [4, 93]. Data-driven approaches, instead, extract this knowledge directly from data. Modern deep learning approaches to SISR [66, 68, 67] descend from the pioneer works of [38] and [97]. A recent state-of-the-art neural network design is SwinIR [94] which leverages a windows-attention-based architecture. It is also worth mentioning that some works [71] tackle the blind SISR problem, *i.e.*, when the degradation process is not known and hence should be estimated. Finally, a branch of the literature is concerned with lightweight architectures, such as the one by [86].

3.2.2 Multi-Image Super-Resolution

The ill-posedness of SISR is intuitively reduced if extra images of the same scene are available. This MISR approach can be further specialized in the multiframe-SR if these extra images comes from adjacent frames of a video, burst-SR if they comes from a photo-burst, stereo-SR if the single extra image is the stereo companion of the target one.

Multiframe-SR and burst-SR assume small geometric disparity as there are small camera movements between successive acquisitions. Exploiting this fact, the first step in algorithms for these settings is typically to register the images to each other using optical flow models [5]. Recent works in the context of the burst-SR challenge

by [15], such as [14], [91], and [100] follow this approach, relying on neural networks modules estimating optical flow. However, optical flow models geometric relations as locally translational on the camera plane, and, as such, is limited in its expressive power. This is fine when the geometric disparity is small, but a general setting may benefit for a more accurate account of the 3D geometry.

Similarly, lightfield SR [182] employs a familiar grid-like arrangement of multiple cameras with minimal disparities. Consequently, it facilitates simpler image fusion techniques and does not impose as stringent robustness requirements as a setting with large disparities. For instance, our scenario necessitates addressing potential occlusions and non-Lambertian surfaces. Unlike light field SR, which can comfortably rely on Lambertian approximations due to its small disparities, this approach does not exhibit clear generalizability to the large-disparity setting explored in our study.

The stereo-SR setting, instead, assumes only the presence of two cameras (*i.e.*, just one extra image) and the acquisition setting is typically controlled so that camera poses only differ by an horizontal shift. Recently, [183, 152, 34] developed methods for stereo-SR that utilizes an attention mechanism to perform image alignment implicitly. Finally, a branch of the literature is concerned with SISR or stereo-SR of omnidirectional images [29, 28].

To the best of our knowledge, EpiMISR is the first method that tackles the problem of generic multi-image super-resolution, *i.e.*, there are no assumptions about the number of images or the relative poses of the cameras. Hence, we move away from 2D image alignment processes and leverage a full deep 3D world model.

3.2.3 NeRF and Image Fusion

As described in Section 2.3, NeRF architectures are neural world models, as they encode information from posed images in the weights of a neural network in a 3D-geometrically consistent way. In their original formulation by [109], a multilayer perceptron encodes the 5D radiance field of a given scene. An alternative 3D scene representation, based on Gaussian-shaped primitives, is proposed in [82], and explained in Section 2.4. Further NeRF evolutions, such as [178, 163] aim to avoid per-scene training, learning general priors by introducing a feature extractor and exploiting constraints from epipolar geometry in an explicit way. Some works move away from the physically-grounded volumetric rendering integral by replacing it with transformers acting on a feature space, and address the novel view synthesis task using both per-scene training [159, 146] or using an inductive approach [147]. [58] uses a similar architecture to perform 3D human joints localization and [164] to perform point cloud reconstruction. Also other works, such as [70], are concerned with multi-image fusion leveraging transformers in their pipelines. However, they differ from EpiMISR in that they do not deal with a super-resolution problem and are often limited by processing images in pairs and then aggregating the results

with non-parametric processes. Recently, NeRF-like models have also been used to address inverse problems in imaging, of which super-resolution is an example. [119] and [108] address the case where the input views are noisy, discovering outstanding denoising performance. [161], [57], instead, tackle the problem of superresolving the NeRF 3D geometry model, hence being capable of generating novel-views at a higher resolution. EpiMISR differs significantly from them in that we are concerned with super-resolution of existing views only and we do not optimize on a per-scene basis, but rather leverage a training set to train an image fusion model that can be then used for an arbitrary scene with an arbitrary number of views with an arbitrary geometry.

3.3 Proposed Method

We address the setting in which a number of images of a given scene are acquired from arbitrary vantage points, possibly with large geometric disparity. These images have low resolution and we seek to super-resolve one of them by suitably combining the complementary information carried by the other images. Our proposed method, called **EpiMISR**, is a MISR neural network which explicitly accounts for the epipolar geometry by exploiting camera poses and processing 3D feature fields in a NeRF-like manner. Given $V+1$ LR views of a static scene, and the corresponding intrinsic and extrinsic camera parameters, our task is to obtain a HR version of one of them, which we will call the *target view*, by also leveraging information from the V extra views. In the parlance of NeRF models, this is referred to as *not-novel* view synthesis.

EpiMISR is not optimized on a per-scene basis, but rather uses a training set to learn the function needed to perform image fusion with an arbitrary geometry for the SR task in a supervised way. As shown in the high-level overview in Fig. 3.1, EpiMISR consists of three main modules, named SISR-FE, CAP and MIFF, which create SR features, sample them along epipolar lines and fuse them, and will be detailed in the following sections. Notice that EpiMISR also computes a super-resolved image from only the target view, called I^{SISR}. We found that a balanced loss function optimizing the fidelity of both the SISR and MISR outputs with respect to the HR ground truth, such as

$$L^1\left(I^{\mathrm{MISR}}, I^{\mathrm{HR}}\right) + L^1\left(I^{\mathrm{SISR}}, I^{\mathrm{HR}}\right) \tag{3.2}$$

provided more stable performance over a variable range of available views and ensured that the degenerate case of a single view ($V = 0$) recovers the performance of the SISR backbone.

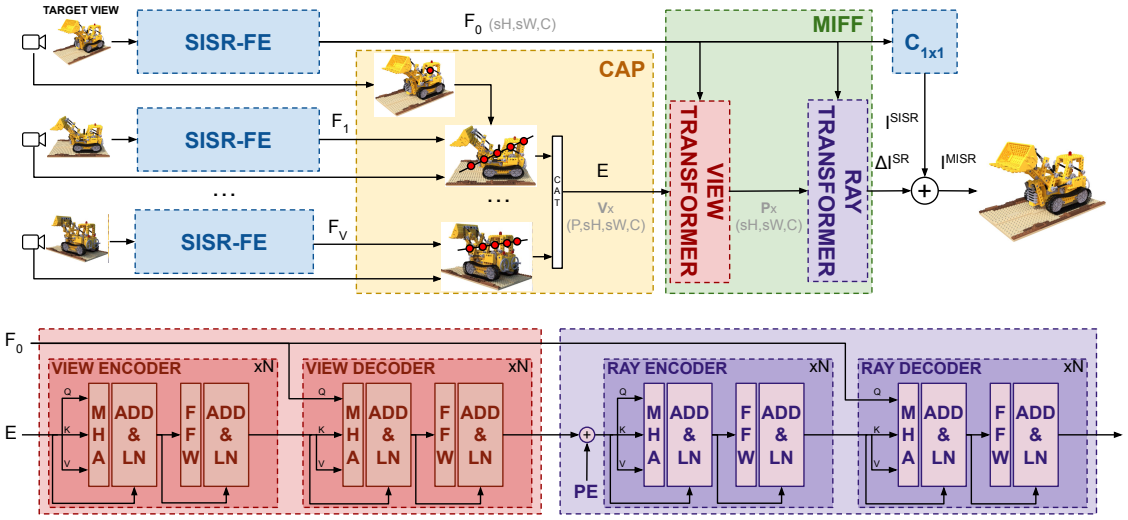


Figure 3.1: EpiMISR Architecture. From the LR target view and the extra views super-resolved features are obtained by any single-image SR network (SISR-FE), sampled along epipolar lines associated to pixels in the target view (CAP) and fused (MIFF) to produce a residual correction to single-image SR.

3.3.1 SISR-FE Module

The single-image super-resolution feature extractor (SISR-FE) module is shared across views and its purpose is to capture strong spatial priors (local correlation and, possibly, non-local self-similarity) to extract features supported on a super-resolved image grid. Each pixel in this super-resolved grid is geometrically positioned on the camera plane associated to each particular view, but its feature vector captures the information of a neighborhood. The increased resolution with respect to the original allows finer processing by the other modules. Being part of a modular approach, SISR-FE can leverage any state-of-the-art SISR architecture by truncating the final projection to RGB space. More formally, let I_v^{LR} be the v-th view as input of the module, its output will be a set of C feature maps at s times the resolution:

$$\text{SISR-FE} : I_v^{\mathrm{LR}} \in \mathbb{R}^{H,W,3} \rightarrow F_v \in \mathbb{R}^{sH,sW,C} \tag{3.3}$$

where $v = 0$ denotes the target view. We also remark that a SISR image prediction I^{SISR} is obtained from F_0 via projection of features to RGB values, and it is used as a basis for the multi-image residual correction estimated by the other modules.

3.3.2 CAP Module

In order to handle potentially large geometric disparities in camera poses, epipolar geometry is employed instead of the optical flow modules commonly used in the burst SR literature. A deterministic, non-learnable module called Cast-and-Project

(CAP) is used to implement epipolar geometry with an approximate pinhole camera model. Given a pixel on the SR target view grid, there exists an associated straight line, called the epipolar line, for each of the extra views, such that the line will intersect with the object imaged by the target pixel. The CAP module is shared across the extra views, and receives as input the camera model (as defined in Section 2.1.2) of the target view $\mathcal{C}_0$, the camera model of the v-th view $\mathcal{C}_v$ and the super-resolved feature map of the v-th view F_v to compute the epipolar features E_v.

$$\mathrm{CAP}_{\mathcal{C}_0 \to \mathcal{C}_v} : F_v \in \mathbb{R}^{sH,sW,C} \to E_v \in \mathbb{R}^{P,sH,sW,C} \tag{3.4}$$

The epipolar features tensor E_v denotes the epipolar lines for view v sampled at P locations, for each pixel and feature in the target view.

The purpose of this module is to build the tensor E_v so that the following MIFF module can efficiently scan the epipolar line in search of features in the extra views that match the feature in the target view at each target pixel position, thus effectively exploiting inter-view information. For each pixel in the target view, CAP casts a ray (as defined in Section 2.1.2) in the 3D space passing through the center of the target camera and the selected pixel (using $\mathcal{C}_0^{-1}$). Along this ray, P points are sampled. For each sampled point, the module computes the projection point onto the image plane of the extra view (using $\mathcal{C}_v$). As the obtained coordinates can be non-integer, the module bicubically resamples the super-resolved feature maps F_v at the correct coordinates. This also highlights the importance of having features F_v on a super-resolved grid to properly account for fine details. The module also generates a boolean mask to flag invalid projected points that are outside the feature map or behind the extra camera. We also note that CAP samples points hyperbolically along the ray, so that the points are equally spaced when projected on the image planes.

3.3.3 MIFF Module

The MIFF (Multi-Image Feature Fusion) module receives as input the epipolar feature tensors $E_1, \ldots, E_V$ returned by the CAP module, containing features from the extra views, warped and aligned to the target view. Its task is to aggregate them to return a residual correction to the SISR image of the target view that accounts for the information of the other views.

$$\mathrm{MIFF} : (F_0, \{E_1, \ldots, E_V\}) \to \Delta I^{\mathrm{SR}} \in \mathbb{R}^{sH,sW,3} \tag{3.5}$$

The final super-resolved version of the target view is then obtained by:

$$I^{\mathrm{MISR}} = I^{\mathrm{SISR}} + \Delta I^{\mathrm{SR}}. \tag{3.6}$$

Similarly to [159], we drop the classical physics-based volume integral formulation, replacing it with two transformers that aggregate the information from the extra views directly in a feature space. The two transformers work in a cascade fashion, with the first transformer aggregating the views (*view transformer*) and the second transformer aggregating the points along the ray (*ray transformer*). Using the notation from [160], each transformer is formed by an encoder and a decoder module. We refer the reader to Fig. 3.1 for a detailed block diagram of the following explanation.

The encoder for the view transformer considers the sequence of V epipolar feature tensors E_v as input and derives joint features by means of a stack of several multihead self-attention layers, feed-forward layers and LayerNorm layers [3]. This operation is crucial as it allows for the fusion of independently computed features E_v from each view. By leveraging self-attention layers we enable the network to derive more intricate and integrated joint features. Also notice that this operation is equivariant to the ordering of the views and does not depend on the specific number of views V available. The output of the view transformer encoder is a sequence of length V of joint features. This is provided as input to the decoder together with the super-resolved features F_0 of the target view. The decoder uses multiple cross-attention layers to correlate the features of the target view with those extracted from the other views. Its output summarizes the content of the views in a feature field, equivalent to the radiance field in the physics-based approach of NeRF.

Next, the ray transformer replaces the physics-backed volumetric integral to integrate the feature field over the ray. Again an encoder-decoder structure is used. The encoder performs self-attention over the sequence of P ray points to mix the ray features. Then the decoder uses cross-attention between the super-resolved features F_0 of the target view and the output of the encoder to estimate the RGB residual image correction ΔI^{SR} that is added to the SISR image.

Notice that performing the aggregation along the ray and then along the views is not optimal. However, performing both aggregations together in a single step is too computationally demanding, hence we perform first the aggregation along the views and then along the ray.

Figure 3.2: DTU scene 3 with 4× scale factor. From left to right: LR nearest neighbours interpolation (19.31 dB), NeRF-SR (19.75 dB), BSRT (23.60 dB), EpiMISR (24.43 dB), HR ground truth.

3.4 Experimental Results

3.4.1 Experimental Setting

In this chapter, we address the MISR task with a supervised learning paradigm. In order to properly characterize the proposed method from an experimental standpoint, we need a setting with multiple images having relatively large disparity compared to the more conventionally studied burst SR setting. Consequently, we use the DTU dataset ([75]), which is already known in the NeRF literature, for this new SR setting. In particular, we utilize the rectified DTU dataset (in particular, its third light setting, as it is the most uniform), comprising 124 different scenes, with 49 posed views per scene, each view having 1600×1200 pixels. For reasons of computational efficiency, we first bicubically downsample the original images by a factor of 4 obtaining the 400×300 HR images from which degraded LR images are derived. We split the dataset into train, validation and test. Validation set is formed by only scene 47 while the test set is formed by scenes 3, 10, 13, 18, 30, 63, 77, 99, 103. All the other 114 scenes form the train split. From each scene, multiple input sets are extracted by selecting as the target view a random image among the 49 and then choosing the nearest V images as extra views, with respect to camera centers. The number of extra views during training is $V = 7$ and, unless otherwise stated,the same number is also used for testing. The angle between the target view and the other views ranges between 11 and 33 degrees, averaging around 15 degrees, which is in line with our large disparity setting.

In our experiments, the SISR-FE module is based on the SwinIR architecture ([94]) in order to be comparable with recent methods in the burst SR literature. We also present some ablations with simpler designs for SISR-FE in Section 3.4.5. The number of points sampled by the CAP module along the ray during training

is $P = 256$, and, unless otherwise stated, the same number is used during testing. Finally, regarding the MIFF module, we set the number of encoder and decoder layers to 4 for both transformers.

The training pipeline of EpiMISR for the following experiments consists of two steps. First, we pretrain the SISR-FE module and its RGB projection as a SISR neural network on the DIV2K dataset from [1], and finetune it on the DTU dataset. Then the whole EpiMISR architecture is trained end-to-end for the MISR task using the loss in Eq. (3.2).

We employ the Adam optimizer for the end-to-end optimization of EpiMISR. The SISR-FE module is frozen to the pretrained weights for the first 350 iterations to train the sole MIFF module and stabilize the training, followed by an additional 150 iterations to finetune the whole network. The learning rate is linearly warmed up for the first 60 epochs starting with 10^{-6} up to 10^{-4}. A multi-step scheduler halves it at epochs 150,250. For the final 150 epochs, the learning rate is set to 10^{-5} and further halved at epochs 80,120. We train on four A100 GPUs for about 7 days.

We compare the proposed technique to a number of state-of-the-art approaches for multi-image super-resolution in the literature. However, we remark that our setting with relatively large parallax and free camera positions is new and different from existing settings in the super-resolution literature. The closest match is the burst SR literature, which however only considers small disparities and does not use camera poses. We consider BSRT ([100]) as the state-of-the-art for the burst SR literature, and DBSR ([14]) as additional baseline. The NeRF literature has recently published the NeRF-SR method by [161]. We consider this method as an interesting additional point of reference which follows the NeRF methodologies and explicitly uses camera poses. However, NeRF-SR follows a different settings as it is concerned with novel view synthesis at a higher resolution rather than not-novel view enhancement and it does not follow the supervised learning paradigm. A recent preprint by [57] proposes Super-NeRF, but it has not been tested due to the lack of publicly available code. Besides, its setting is also different because, similarly to NeRF-SR, it does not follow the supervised learning paradigm, it focuses on novel view synthesis and, moreover, it optimizes for perception metrics and not for distortion. All methods in our comparisons have been retrained using the authors' code and following the same pretraining procedure of EpiMISR. The number of epochs for their training has been chosen to maximize their performance on a validation set. A minor modification has been made to the burst methods to use RGB images instead of RAW mosaiced images.

3.4.2 Main Experiment

Table 3.1 reports our main results on the DTU dataset for a 4× SR factor. To quantitatively assess the super-resolved image against the ground truth, we use PSNR as distortion metric and SSIM and LPIPS as perceptual metrics. We remark that all methods, except NeRF-SR, optimize for distortion rather than perception, see [19] for distortion vs. perception tradeoff. Metrics are computed after cropping 16 pixels on each side to avoid border effects. It can be noticed that some multi-image methods with weak geometric priors struggle to improve over the SISR result of SwinIR, indicating that naive multi-view fusion is not sufficient in this setting. As a sanity check, we tested but not reported in the table the SISR performance of EpiMISR after all the finetuning procedures, and saw that it is just marginally above the reference SwinIR results (26.96 dB), confirming that improvements actually come from the use of multiple images. Instead, the state-of-the-art from the burst SR literature (Burstormer) is not able to handle the high disparities present in the DTU dataset, highlighting the difference between the burst SR task and the generic multi-image SR task. We found that a previous burst SR method (BSRT) performs better than Burstormer but still shows a significantly lower PSNR of about 0.8 dB compared to EpiMISR, highlighting the importance of explicitly modeling the problem geometry at the core of our model rather than relying on optical flow. NeRF-SR does not show competitive performance, which is expected for several reasons: i) it targets the novel view synthesis setting; ii) it is optimized on a per-scene basis, thus not being able to learn powerful image priors from training data; iii) it is a much smaller model. We also compare EpiMISR with the 3DGS [82] method, which, in this task, does not show competitive performance as it shares the same drawbacks of NeRF-SR. Figure 3.2 shows a qualitative comparison between the proposed method and the other baselines. It can be noticed that EpiMISR provides more accurate details and recovers sharper structures than the competing methods, although a mild residual blurring is still visible with respect to the HR ground truth in some regions. Additional qualitative examples and failure cases are reported later in Figs. 3.6 and 3.7.

	No. Params	PSNR ↑	SSIM ↑	LPIPS ↓
EpiMISR	23.30M	**28.60**	**0.87**	**0.11**
BSRT [100]	20.56M	<u>27.84</u>	<u>0.85</u>	<u>0.13</u>
Burstormer [39]	3.27M	26.76	0.84	0.16
DBSR [14]	12.91M	26.36	0.80	0.20
NeRF-SR [161]	1.19M	23.17	0.64	0.32
SwinIR [94]	14.70M	26.87	0.82	0.17
3DGS [82]	≈ 1.88M	22.73	0.73	0.30

Table 3.1: Supervised quantitative comparison on the DTU dataset.

3.4.3 Zero-Shot Experiments

In this section we report our results on the 1023 scenes from the Google Scanned Objects dataset [163] (512×512 pixels) and on the 8 scenes from LLFF dataset [107] ($\approx 2100 \times 1600$) pixels), for a $4\times$ SR factor. Table 3.2 reports the evaluation results of EpiMISR, BSRT and SwinIR methods on the Google Scanned Objects dataset and on the LLFF dataset. All the methods are trained only on DTU dataset as previously described and are not finetuned on the GSO dataset nor on the LLFF dataset, hence these results shows that EpiMISR out-performs baselines even in the difficult zero-shot setting (*i.e.*, on an unseen data distribution).

	GSO			LLFF		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
EpiMISR	**31.50**	**0.96**	**0.04**	**23.07**	**0.74**	**0.20**
BSRT [100]	<u>30.09</u>	<u>0.95</u>	<u>0.05</u>	<u>22.85</u>	<u>0.72</u>	<u>0.24</u>
SwinIR [94]	29.29	<u>0.95</u>	0.07	22.27	0.69	0.29

Table 3.2: Zero-shot quantitative comparison on the GSO and LLFF datasets.

3.4.4 Wider Baseline Experiment

In this section we present an experiment where views are taken very far apart and asymmetrically with respect to the target view in order to challenge the method and the state-of-the-art BSRT. Table 3.3 reports the PSNR obtained by BSRT and EpiMISR when compared to the SISR PSNR. It can be noticed that in this challenging setting, BSRT degrades to the SISR performance, while EpiMISR still provides an improvement. However, the gain of the proposed approach over SISR is also limited in this regime. This is expected because, with a very wide baseline, the overlap between views is reduced and the corresponding image regions become harder to match reliably along the epipolar lines, due to the effect of occlusions, non-Lambertian surfaces and local appearance changes. This more challenging geometry is created by taking the $V-1$ extra views that are at median distance (out of all the views available in the dataset) with respect to the distance to the target view camera center.

	No. Params	PSNR ↑	SSIM ↑	LPIPS ↓
EpiMISR	23.30M	**27.00**	<u>0.82</u>	**0.15**
BSRT [100]	20.56M	26.82	**0.83**	<u>0.16</u>
SwinIR [94]	14.70M	<u>26.87</u>	<u>0.82</u>	0.17

Table 3.3: Quantitative comparison on the DTU dataset with challenging geometry.

3.4.5 SISR-FE Ablation

The EpiMISR modular design allows to decouple the fusion of multiple images using the 3D geometry from the super-resolved feature extraction, which can leverage advances in SISR methods or be tuned for the desired complexity. In this section, we present some MISR results using different SISR-FE modules in order to study its impact on overall performance. Results are shown in Table 3.4. Unsurprisingly, the SwinIR architecture used in the main experiment provides the best performance but it is also a relatively large model. However, it is interesting to notice that the RLFN architecture by [86] from the NTIRE 2022 challenge on Efficient Super-Resolution is able to still improve over BSRT with a fraction of the parameters. We also notice that bicubic upsampling followed by 1×1 RGB-to-features convolution is not sufficient to provide reasonable performance, highlighting the need for operations that capture a local context larger than 1 pixel. In fact, when bicubic upsampling is followed by 3×3 convolution the subsequent MIFF module is able to successfully exploit the local context as the overall performance increases by 1.17dB while the SISR performance stays almost the same. We also notice that the PSNR difference between the single-image and multi-image results is stable around 1.6 dB, proving that the MIFF module is relatively robust to the single-image processing.

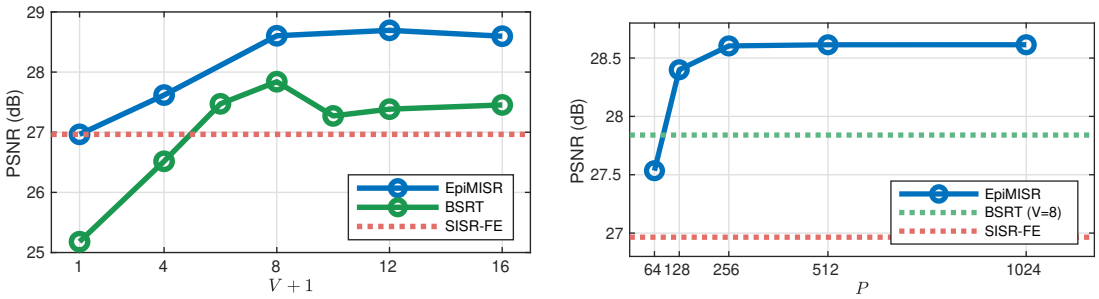
SISR-FE Module	No. Params	PSNR (SISR) ↑	SSIM ↑	LPIPS ↓
SwinIR	14.85M	**28.60** (**26.96**)	**0.87**	**0.11**
RLFN [86]	0.86M	<u>28.05</u> (<u>26.38</u>)	<u>0.86</u>	<u>0.12</u>
Bicubic + conv3×3	7.94k	25.73 (24.13)	0.79	0.23
Bicubic + conv1×1	1.80k	24.56 (24.04)	0.76	0.27

Table 3.4: Quantitative comparison of different SISR-FE modules in terms of MISR and SISR performance on the DTU dataset.

3.4.6 Hyperparameter Ablations

In this section, we study the impact of two important parameters of the proposed method, namely V, the number of extra views, and P the number of points along the ray.

It can be expected that increasing the number of views V allows to integrate extra information and increase the quality of the SR image. However, diminishing returns are expected, especially for extra views with very large disparity. Figure 3.3a reports the PSNR of the SR image for different number of views used by the super-resolution process. Images are added by expanding the neighborhood of available views around the target, so they are progressively farther or more angled



(a) PSNR varying V.

(b) PSNR varying P.

Figure 3.3: PSNR with respect to V and P.

with respect to the target. We notice that only a marginal improvement is obtained increasing from 8 to 16 views. Regarding views, we also remark that EpiMISR can process an arbitrary number of input views with an arbitrary ordering, as its operations are invariant in that dimension.

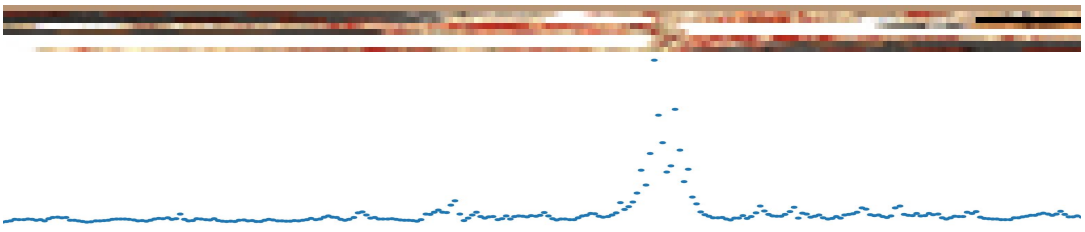
The number of ray points P determines the density of the feature field that takes the place of the radiance field in our model. This parameter is strictly tied to the resolution of the images and the scene characteristics, and its sampling should be fine enough to capture the fine details of the scene. Figure 3.3b shows that a too small value of P has a significant impact on SR quality, while performance saturates beyond the chosen value of $P = 256$.

3.4.7 Analysis of Ray Attention

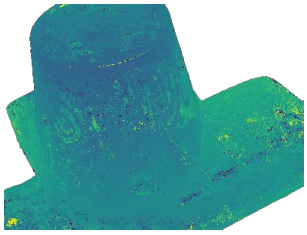
In this section, we present an interpretation of the attention map generated by the ray transformer within the MIFF module as a depth map.

Figure 3.4c illustrates a typical input image set. The first image is the target view, while the subsequent $V = 7$ images are the extra views. Let us fix the pixel to be superresolved in the target image. The CAP modules casts a ray through this pixel and projects it onto the other views. This process yields samples along the epipolar lines, which are collected to form a "strip" of dimensions $P \times (V + 1)$, depicted in Fig. 3.4a (depiction is in RGB space instead of feature vectors). There are P columns because the CAP module samples P points along the epipolar lines, and there are $V + 1$ rows because there are $V + 1$ epipolar lines. It is worth noting that the first row comprises repeated instances of the same pixel, as the epipolar line collapses to a single point in the target view.

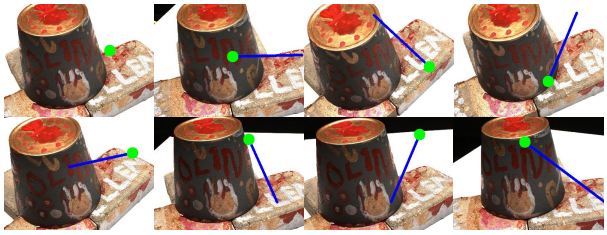
Thanks to the property of epipolar geometry, there is a region along the strip, which we will call "strip alignment region", where all the views are imaging the same 3D point, hence the sampled feature map should report similar information. The attention weights generated by the ray transformer are also visualized in Fig. 3.4a and we can see they reach their maximum in the alignment region, meaning that the



(a) Strip aligned against Ray Transformer attention.



(b) Depth map.



(c) Input image set with epipolar lines.

Figure 3.4: An example of depth map generation.

MIFF module has identified the correspondences across all extra images. Moreover, the position of the maximum attention weight provides an estimate of the depth of the object imaged by the selected pixel in the target view. A noisy depth map for all pixels can be extracted in this unsupervised way and is visualized in Fig. 3.4b.

3.4.8 Sensitivity Analysis to Camera Parameter Estimation

In this section we present two experiments that show EpiMISR performance when camera parameters are not available (first experiment) or noisy (second experiment).

Camera parameters in the DTU dataset are highly accurate, as they have been obtained from a calibration procedure. The availability of calibrated poses is not guaranteed in practical application. Hence, in this experiment we use the state-of-the-art HLOC algorithm [133] to infer poses from the LR images alone. We then input these inferred poses to EpiMISR and obtain the predicted HR images. Comapring with the ground truth HR images, we report a PSNR of 28.10 dB, which is degraded from the result with accurate poses but still superior

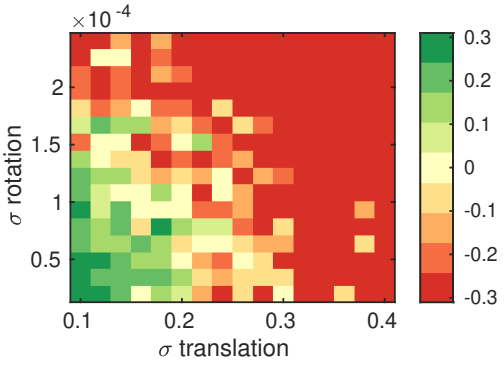


Figure 3.5: EpiMISR PSNR gain (dB) over BSRT for different noise regimes on camera poses for a single test image.

to BSRT which does not need that information, confirming that, while improvements in 3D geometry modeling lead to improved performance, poses retrieved from a practical algorithm are sufficient to obtain state-of-the-art super-resolution performance.

As a second experiment, a sensitivity analysis to perturbations of the extrinsic camera parameters is shown in Fig. 3.5. It shows the PSNR achieved when the 6-D DTU pose is perturbed to simulate uncertainty. A diagonal zero-mean Gaussian with parameter $\sigma_{\text{translation}}$ is used to perturb the translational components. A simple symmetric distribution over $SO(3)$ with parameter σ_{rotation} is used to perturb the rotational component. As Fig. 3.5 shows, the performance of EpiMISR degrades in higher noise poses regime, but it is still superior to BSRT in a lower noise regime and, overall, it exhibits a stable trend.

Finally, we remark that camera parameter estimation from LR images performed disjointly from the SR process is clearly suboptimal. Future work may significantly improve the results by designing joint methods that correct an initial pose estimation while performing super-resolution, similarly to what is done by NeRF methods for in-the-wild images ([104]).

3.4.9 Failure Cases and More Qualitative Results

To complement the quantitative discussion in Tables 3.1 and 3.3, we report here additional qualitative results and failure cases.

The fraction of the DTU dataset test split where BSRT outperforms EpiMISR is only about 2%. Figure 3.6 shows an example of such rare cases while Fig. 3.7 reports some DTU scenes results where the proposed method outperforms the baselines.

Figure 3.6: A qualitative example of a failure case (DTU dataset, scan 63). This is an example where BSRT outperforms EpiMISR. From left to right: LR nearest neighbours interpolation, NeRF-SR, BSRT, EpiMISR, HR ground truth.

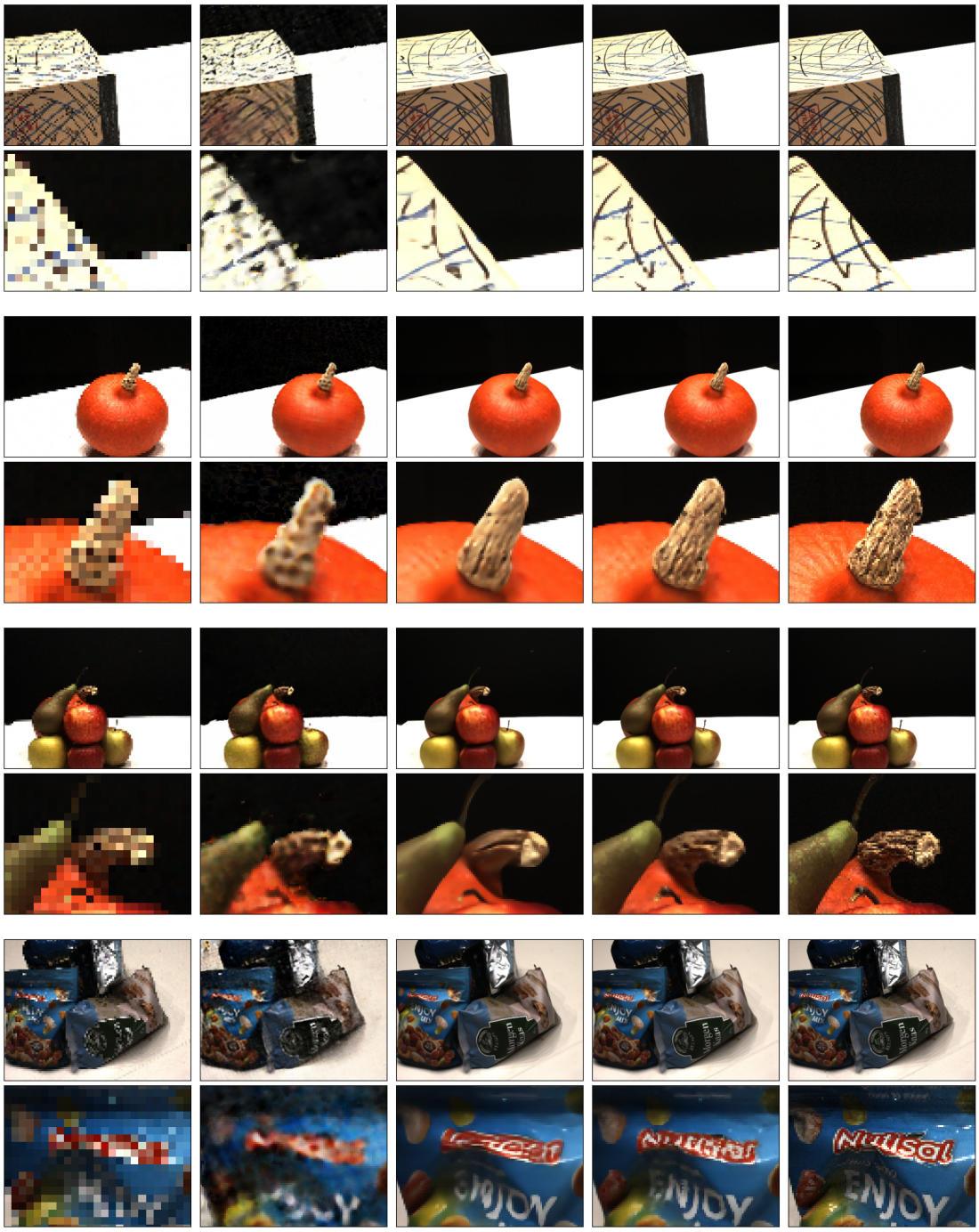


Figure 3.7: Qualitative results of some DTU test scenes with 4× scale factor. From left to right: LR nearest neighbours interpolation, NeRF-SR, BSRT, EpiMISR, HR ground truth.

3.5 Conclusions & Future Works

In this chapter we introduced EpiMISR, a new formulation of multi-image super-resolution for image sets with arbitrary camera placements and potentially large disparities. By replacing image-plane optical flow with explicit epipolar geometry, the method can aggregate information along rays consistent with the calibrated cameras and outperform existing flow-based approaches in challenging wide-baseline settings. More broadly, EpiMISR shows that geometry-aware 3D reasoning can provide a stronger prior for super-resolution than purely 2D correspondence models.

At the same time, the current formulation still depends on reasonably accurate pre-estimated camera parameters and on the assumption of static scenes. As pose noise increases, reconstruction quality degrades, and the present CAP module is tailored to pinhole cameras, limiting its applicability to more complex imaging models such as fish-eye or wide-angle. A natural direction for future work is therefore to couple super-resolution and camera refinement more tightly, so that pose errors can be corrected during reconstruction rather than treated as fixed upstream noise.

Chapter 4

Reliability

In the previous chapter, we treated volumetric scene representations as powerful tools for super-resolving images: once a world model has been reconstructed, the information from different low resolution images can be fused together, and we can render high-resolution views that were never directly observed. Implicit in this story is a silent assumption: that the reconstructed model is *correct enough* wherever we choose to use it. In practice, however, no reconstruction is perfect. Cameras are noisy, calibration is imperfect, parts of the scene are poorly seen or not seen at all, and learning-based models can extrapolate in surprising ways. A system that only outputs a single, best-guess image offers no clue as to where it might be hallucinating structure or misrepresenting appearance. For applications that rely on these models (*e.g.*, robotics, autonomous driving, scientific measurement, or safety-critical visualization) this is not just a matter of aesthetics: it is a matter of risk. Knowing *how wrong we might be* can be as important as knowing *what we think is there*.

Uncertainty is a familiar concept in other domains. Weather forecasts report not only tomorrow's temperature but also the probability of rain; navigation systems consider confidence in road positions when planning routes; experimental sciences accompany every measurement with an error bar. By contrast, many modern 3D reconstruction and novel view synthesis pipelines still act as if uncertainty did not exist: they compress all the complexity of the inverse problem into a single deterministic output. Neural radiance fields have started to challenge this view by incorporating Bayesian formulations and outputting per-pixel uncertainty estimates, but these advances have so far remained largely confined to slow, network-based representations. At the same time, Gaussian Splatting has emerged as an attractive alternative: it offers a volumetric, radiance-field-like description of a scene in terms of simple Gaussian primitives, and it enables real-time rendering with high visual fidelity. Yet in its original form, Gaussian Splatting is entirely deterministic: every pixel is a fixed function of a fixed set of Gaussians, with no attached notion of confidence.

In this chapter, we ask how to endow Gaussian Splatting with the ability to say "I am not sure" - and to do so without sacrificing its efficiency. Rather than changing the rendering pipeline itself, we reinterpret the underlying primitives as stochastic objects: their positions, opacities, and colors are no longer single values, but random variables with distributions that can be learned from data. By sampling these distributions, we obtain multiple plausible realizations of the same scene, whose variability translates into uncertainty maps over the synthesized images. Crucially, we do not treat uncertainty as an afterthought: we explicitly train the model so that high predicted uncertainty correlates with large actual errors, allowing downstream systems to trust or discount different regions of a rendered view. The result is a *stochastic* extension of Gaussian Splatting that preserves performance while providing meaningful, spatially resolved confidence estimates. This bridges the gap between fast volumetric rendering and risk-aware decision-making, and addresses the second challenge outlined in the introduction: making learned 3D world models not only accurate, but also reliable.

4.1 Introduction

Novel-view synthesis, the task of generating images of a scene from viewpoints not observed during data collection, is a fundamental problem in computer vision with numerous applications, including virtual reality, augmented reality, and robotics. Traditionally, this task has been addressed using methods such as Structure from Motion [132] which rely on geometric reconstruction techniques. However, recent advances in deep learning, particularly with the introduction of Neural Radiance Fields (NeRF), have revolutionized the field by enabling high-fidelity synthesis of novel views directly from the underlying scene representation.

NeRF [109] has recently enjoyed great success by representing a scene as a continuous volumetric function that maps 3D spatial coordinates and view directions to radiance values. By learning this function from a set of posed images, NeRF can generate photorealistic images from novel viewpoints, as explained in Section 2.3. However, while NeRF achieves impressive results, its computational complexity and memory requirements limit its practicality for real-time applications. This is the focus of the emerging 3D Gaussian Splatting (3DGS) technique [82] which offers a more computationally efficient alternative to NeRF while maintaining high-quality novel-view synthesis. 3DGS learns to approximate the radiance field by using a set of Gaussian primitives, enabling real-time rendering with competitive visual fidelity, as detailed in Section 2.4.

At the same time, research in novel view synthesis has started addressing the problem of estimating the epistemic or aleatoric uncertainty in order to understand the reliability of the generated views. Indeed, any practical downstream task that involves taking actions in the real world (such as robotics and autonomous systems)

must consider not only the newly synthesized views but also their corresponding uncertainties, in order to potentially discard too uncertain yet promising actions. As an example, a robot moving in the real world could use a 3DGS representation of the environment around itself, and plans movement according to some goal. It is very important that this goal takes into account also the uncertainty of the representation of the environment, as in application riskier moves (moving too close to a too-much uncertain obstacle) should be penalized. This scenario was first addressed in the seminal work of Shen et al. [139], where they proposed a deep architecture, based on NeRF, called S-NeRF to also estimate meaningful (aleatoric) uncertainty maps for each generated view.

At the moment, 3DGS lacks a mechanism for estimating uncertainty in the synthesized views. In this chapter, we seek to address this limitation by proposing a novel framework for uncertainty estimation in 3DGS. We extend the traditional deterministic 3DGS framework to incorporate stochasticity, allowing us to predict uncertainty alongside synthesized views. Our approach leverages Variational Inference (VI) to learn the parameters of the 3DGS radiance field in a Bayesian framework, enabling us to accurately estimate uncertainty without sacrificing computational efficiency. As done in S-NeRF and the subsequent work CF-NeRF, our method SGS primarily focuses on aleatoric uncertainty, not epistemic uncertainty. Aleatoric uncertainty refers to the irreducible uncertainty inherent in the data due to stochastic processes, such as noise in image acquisition or variability in camera pose estimation, while epistemic uncertainty refers to the model's lack of knowledge, *e.g.*, due to limited training data or incomplete view coverage. This is an important limitation: in novel-view synthesis, the available training views are inevitably only a subsample of the scene's light-field information, and the missing information is often the main source of uncertainty in downstream use.

We can summarize our novel contributions as follows:

- we introduce a novel framework for uncertainty estimation in 3DGS, called Stochastic Gaussian Splatting (SGS), enabling real-time synthesis of high-quality images with accurate uncertainty predictions;
- we propose a VI-based approach to learn the parameters of the 3DGS radiance field, allowing us to incorporate uncertainty prediction seamlessly into the rendering pipeline. Moreover, we innovate this learning process by augmenting Empirical Bayes with a loss function dependent on the area under the sparsification curve;
- we demonstrate the effectiveness of our approach through experiments on three different datasets (LLFF, Blender and Mip-NeRF360), showing significant improvements in both rendering quality and uncertainty estimation metrics compared to state-of-the-art methods.

4.2 Background

4.2.1 Gaussian Splatting

In this chapter, our focus is on 3DGS [82], which achieves state-of-the-art visual quality while maintaining competitive training times and, importantly, enabling high-quality real-time novel-view synthesis ($\geq$ 30fps at 1080p resolution). As summarized in Section 2.4 and Section 2.3, 3DGS replaces the deep neural network in NeRF with an explicit set of Gaussian primitives and exploits the *splatting* operation instead of ray marching. This explicit formulation is also what makes it a natural starting point for the stochastic extension proposed in this chapter.

4.2.2 Structure from Motion Uncertainty Estimation

Uncertainty estimation is a long standing problem in the context of machine learning and computer vision, for example semantic segmentation and depth regression tasks, and it is addressed in many works[80, 81, 99]. Some broader works such as [47, 78] provides an excellent starting point for uncertainty estimation, Bayesian learning and variational inference applied to deep learning.

An emerging research direction for the Structure from Motion field is the quantification of the uncertainty of the synthesized novel-views or of the 3D radiance field itself. This task was first addressed by [139], where it is cast as a Bayesian learning problem and solved with the Variational Inference (VI) framework [78], applied to NeRF. Subsequently, the same authors proposed an evolution of their method [137], which differs from the previous work by dropping the independence assumptions between the color and density fields, hence recovering a more complex stochastic dependency graph using the Conditional Normalizing Flows framework [176]. Following these works, [169] drops all independence assumptions by using a generative Flow-GAN model [55]. Another approach is addressed in [52] where the Laplace Approximation framework [78] is used. Finally, two similar works, [138] and [149], focus on developing methods that actively estimate high uncertainty in spatial areas that are not covered by the input views, using VI and Ensemble Learning frameworks, respectively. It is worth noting that all these methods rely on the use of NeRF, while, to the best of our knowledge, there was no prior work, to the best of our knowledge, addressing uncertainty estimation in the 3DGS framework. Filling this void in literature is interesting, as 3DGS is both more computational efficient than NeRF and more easily interpretable. Moreover, it is not possible to apply the same techniques used in some previous works on NeRF to 3DGS, as they exploit the presence of a deep neural network, for example [48] or [104].

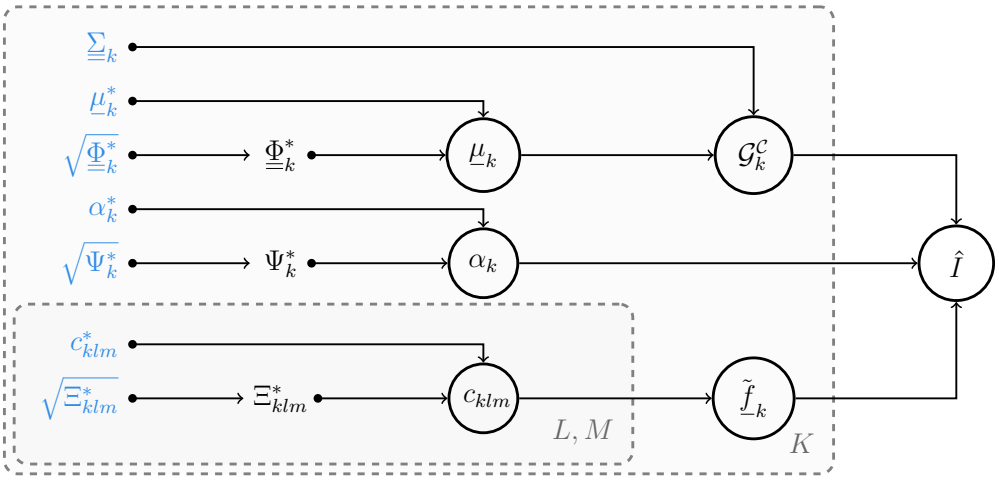


Figure 4.1: Bayesian Network Graphical Model of SGS. Learnable variables are depicted in blue, while stochastic variables are circled. Gray dashed rectangles are used for the plate notation, *i.e.*, variables repetitions.

4.3 Method

In this section, we present the proposed method, called Stochastic Gaussian Splatting (SGS), to enable uncertainty quantification in the Gaussian Splatting framework. Figure 4.1 provides an overview of the probabilistic dependencies in SGS.

4.3.1 Stochastic Gaussian Splatting

The main difference between NeRF and 3DGS lies in the fact that in the original NeRF formulation [109], the color and density fields are regressed with a multilayer perceptron neural network. Instead, in 3DGS [82], the radiant field is approximated with a discrete set of *Gaussian primitives*. See Section 2.4 for an in-depth explaination. Let $\underline{u}$ be a pixel in the image plane, $\underline{r}_{\underline{u}}(t)$ the associated ray, and let $\underline{d}_{\underline{u}}$ be the view direction of that ray. The predicted pixel color $\hat{I}(\underline{u})$ is obtained combining Eqs. (2.12) to (2.14), that we report here for clarity, assuming the background color $I_f = 0$:

$$\begin{cases} \hat{I}(\underline{u}) = \sum_{k=1}^{K} \tilde{f}_k \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{C}}(\underline{u}) \right) \\ \mathcal{G}_k^{\mathcal{C}}(\underline{u}) = \exp\left(-\frac{1}{2} (\underline{u} - \mathcal{C}(\underline{\mu}_k))' \left(\underline{\underline{J}}_k^{\mathcal{C}} \underline{\underline{\Sigma}}_k \underline{\underline{J}}_k^{\mathcal{C}\prime} \right)^{-1} (\underline{u} - \mathcal{C}(\underline{\mu}_k)) \right) \\ \tilde{f}_k = \sum_{l=0}^{L} \sum_{m=0}^{l} c_{klm} Y_{lm}(\underline{d}_{\underline{u}}) \end{cases} \tag{4.1}$$

where:

- K is the number of primitives used to approximate the radiance field;
- $\underline{\tilde{f}}_k$ is the color of the k-th Gaussian primitive along view direction $\underline{d}_{\underline{u}}$;
- α_k is the opacity of the k-th Gaussian primitive;
- $\mathcal{C}$ is the camera projection function;
- $\mathcal{G}_k^{\mathcal{C}}(\underline{u})$ is the splatting coefficient of the k-th Gaussian primitive for pixel $\underline{u}$;
- $\underline{\mu}_k$ is the k-th primitive center (mean);
- $\underline{\underline{\Sigma}}_k \in \mathbb{R}^{3,3}$ is the k-th primitive 3D shape and orientation (covariance matrix);
- $J_k^{\mathcal{C}}$ is the Jacobian of the camera projection function $\mathcal{C}$ evaluated at $\underline{\mu}_k$;
- c_{klm} are the spherical harmonics coefficients of the k-th Gaussian primitive associated to the spherical harmonics basis functions Y_{lm};
- l, m are degree and order of the used spherical harmonics Y_{lm}.

In 3DGS, the learnable parameters for each Gaussian primitive are $\underline{\mu}_k, \underline{\underline{\Sigma}}_k, c_{klm}$, and α_k.

Aligning with previous works in the field, the proposed SGS method predicts an uncertainty value for each pixel in the novel-synthesized view. It is natural to define this uncertainty as the standard deviation of the predicted pixel color, requiring the predicted colors to be random variables. Hence, we need to inject stochasticity into the otherwise deterministic process of Eq. (4.1). We propose to use a Monte Carlo method (sampling the model $S \gtrsim 1$ times) to approximate the variance of pixel colors, so that the expression (4.1) should be evaluated multiple times, each time by sampling some random variables.

Following the Variational Inference framework, we directly expand upon 3DGS by imposing a prior distribution on each of the following parameters:

- $\underline{\mu}_k \sim \mathcal{N}\left(\underline{\mu}_k^*, \underline{\underline{\Phi}}_k^*\right)$
- $\ln \frac{\alpha_k}{1-\alpha_k} \sim \mathcal{N}\left(\alpha_k^*, \Psi_k^*\right)$
- $c_{klm} \sim \mathcal{N}\left(c_{klm}^*, \Xi_{klm}^*\right)$

so that the original 3DGS parameters are no longer learned but are sampled from the above distributions, whose parameters are the new learning variables. In order to keep the model simple, we decided not to model the uncertainty of the rotation and scaling of the Gaussian primitives (hence their covariance matrices), but

to model their positions, colors and opacities. This choice avoids introducing additional matrix-valued random variables and keeps both sampling and optimization significantly simpler. However, it is also a limitation: uncertainty in the covariance matrices would capture ambiguity in the spatial extent and anisotropy of each primitive, so neglecting it may lead SGS to underestimate uncertainty near object boundaries, in elongated structures, or whenever reconstruction errors are better explained by geometric support than by primitive position, color, or opacity alone.

Thanks to the reparameterization trick [85], gradients can flow from the loss in Eq. (2.16), which is a function of the samples, to the new distribution parameters $\underline{\mu}_k^*$, $\underline{\underline{\Phi}}_k^*$, α_k^*, Ψ_k^*, c_{klm}^* and Ξ_{klm}^*, enabling learning with standard backpropagation techniques.

4.3.2 Learning with Variational Inference

The variational framework introduced in [78] and used in [139] and [137] is required for optimization, since direct optimization of just the loss would otherwise incur in underestimation of the pixel variance as the model could reduce to a deterministic one.

The VI framework stems from an approximation of Bayes' Theorem. Let γ_k be a Gaussian primitive of a Gaussian Splatting Radiance Field $\Gamma = \{\gamma_k\}_{k=1}^K$, and let $\mathcal{D} = \{(\underline{u}_i, y_i)\}_i$ be the pixels dataset as explained in Section 2.4, where $\underline{u}_i$ is the i-th pixel and y_i is the associated observed color. Then the Bayes' Theorem reads as:

$$\mathbb{P}(\Gamma|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\Gamma)\,\mathbb{P}(\Gamma)}{\mathbb{P}(\mathcal{D})}. \tag{4.2}$$

Since this is generally intractable, the VI framework prescribes to approximate the true posterior $\mathbb{P}(\Gamma|\mathcal{D})$ by introducing a parametric distribution q_θ over all 3DGS radiance fields Γ and to learn these parameters θ in order to minimize the Kullback-Leibler (KL) divergence between the approximate posterior and the true one:

$$\min_\theta \mathrm{KL}(q_\theta(\Gamma) \,||\, \mathbb{P}(\Gamma|\mathcal{D})) \tag{4.3}$$

Now, this problem is further manipulated to get a tractable expression. Let us start with the following manipulation, where only properties of the logarithm and linearity on the expectation are used:

$$\begin{aligned}
&\mathrm{KL}(q_\theta(\Gamma) \,||\, \mathbb{P}(\Gamma|\mathcal{D})) = \\
&= \mathbb{E}_{q_\theta}\left[\log \frac{q_\theta(\Gamma)}{\mathbb{P}(\Gamma|\mathcal{D})}\right] = \mathbb{E}_{q_\theta}\left[\log \frac{q_\theta(\Gamma)\mathbb{P}(\mathcal{D})}{\mathbb{P}(\mathcal{D}|\Gamma)\,\mathbb{P}(\Gamma)}\right] \\
&= -\mathbb{E}_{q_\theta}[\log \mathbb{P}(\mathcal{D}|\Gamma)] + \mathbb{E}_{q_\theta}\left[\log \frac{q_\theta(\Gamma)}{\mathbb{P}(\Gamma)}\right] + \mathbb{E}_{q_\theta}[\log \mathbb{P}(\mathcal{D})] \\
&= -\mathbb{E}_{q_\theta}[\log \mathbb{P}(\mathcal{D}|\Gamma)] + \mathrm{KL}(q_\theta(\Gamma) \,||\, \mathbb{P}(\Gamma)) + \log \mathbb{P}(\mathcal{D})
\end{aligned} \tag{4.4}$$

Assuming independence, the first term in the last expression is equivalent to:

$$\begin{aligned}
&\mathbb{E}_{q_\theta}[\log \mathbb{P}(\mathcal{D}|\Gamma)] = && (4.5)\\
&= \mathbb{E}_{q_\theta}\left[\log \prod_i \mathbb{P}\left((\underline{u}_i, y_i)\,|\Gamma\right)\right] && (4.6)\\
&= \sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left((\underline{u}_i, y_i)\,|\Gamma\right)] && (4.7)\\
&= \sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)\mathbb{P}\left(\underline{u}_i|\Gamma\right)] && (4.8)\\
&= \sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)\mathbb{P}\left(\underline{u}_i\right)] && (4.9)\\
&= \sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)] + \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(\underline{u}_i\right)] && (4.10)\\
&= \sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)] + \log \mathbb{P}\left(\underline{u}_i\right) && (4.11)
\end{aligned}$$

Plugging (4.5) into (4.4), we get:

$$\begin{aligned}
&\mathrm{KL}(q_\theta(\Gamma)\;||\;\mathbb{P}(\Gamma|\mathcal{D})) = \\
&= -\sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)] + \\
&- \sum_i \log \mathbb{P}\left(\underline{u}_i\right) + \\
&+ \mathrm{KL}(q_\theta(\Gamma)\;||\;\mathbb{P}(\Gamma)) + \log \mathbb{P}(\mathcal{D})
\end{aligned} \tag{4.12}$$

As the optimization variable is the parameters θ, all the terms that do not depend on θ in the latter equation can be discarded. Finally, we have the optimization problem:

$$\min_\theta -\sum_i \mathbb{E}_{q_\theta}[\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)] + \mathrm{KL}(q_\theta(\Gamma)\;||\;\mathbb{P}(\Gamma)) \tag{4.13}$$

The first term in the loss function of problem (4.13) is the expected negative log-likelihood. This term forces θ to maximize the expected log-likelihood by matching the observations in the dataset $\mathcal{D}$. So, in spirit, it replaces the standard loss (2.16). It is estimated with the Monte Carlo method (with S samples) and, for simplicity, by defining the conditional probability distribution $\mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right)$ to be a normal distribution that is pixel-wise independent, i.e.:

$$\log \mathbb{P}\left(y_i|\underline{u}_i, \Gamma\right) \propto \left(y_i - \hat{I}(\underline{u}_i)\right)^2 \tag{4.14}$$

Note that this requirement is not too strict as the independence is required in the conditional probability distribution and not for the unconditional probability distribution.

The second term in Eq. (4.13), instead, limits θ from moving too far away from the prior distribution of the 3DGS radiance field Γ. For it to be efficiently tractable, we suppose independence among the K Gaussian primitives γ_k, for both the prior and the approximate posterior distributions, i.e.:

$$q_\theta(\Gamma) = \prod_{k=1}^{K} q_{\theta_k}(\gamma_k) \tag{4.15}$$

$$\mathbb{P}(\Gamma) = \prod_{k=1}^{K} \mathbb{P}(\gamma_k) \tag{4.16}$$

This independence assumption between Gaussian primitives is more general with respect to previous work. For instance, Shen et al. [139] prescribed independence between the values of the density fields for every pair of points in the radiance field, even if they are very close to each other. Meanwhile, in the proposed method, the independence is prescribed at the Gaussian primitive level and not at the infinitesimal 3D point level.

Thanks to the assumptions in Eqs. (4.15) and (4.16), the second term in Eq. (4.13) can be expanded to become:

$$\mathrm{KL}(q_\theta(\Gamma) \,||\, \mathbb{P}(\Gamma)) = \sum_{k=1}^{K} \mathrm{KL}(q_{\theta_k}(\gamma_k) \,||\, \mathbb{P}(\gamma_k)) \tag{4.17}$$

If we suppose that the prior distributions are themselves multivariate normal, each KL term in the right-hand side of Eq. (4.17) has the following general closed form expression:

$$\begin{aligned}\mathrm{KL}(\mathcal{N}_{\underline{\mu}_0,\underline{\underline{\Sigma}}_0} \,||\, \mathcal{N}_{\underline{\mu}_1,\underline{\underline{\Sigma}}_1}) = &-\frac{3}{2} + \frac{1}{2}\,\mathrm{Tr}\left(\underline{\underline{\Sigma}}_1^{-1}\underline{\underline{\Sigma}}_0\right) \\ &+ \frac{1}{2}(\underline{\mu}_1 - \underline{\mu}_0)^T \underline{\underline{\Sigma}}_1^{-1}(\underline{\mu}_1 - \underline{\mu}_0) \\ &+ \frac{1}{2}\ln\left(\frac{\det\underline{\underline{\Sigma}}_1}{\det\underline{\underline{\Sigma}}_0}\right)\end{aligned}$$

Hence, we define the total KL contribution to the training loss $\mathcal{L}_{\mathrm{KL}}$ as the sum over all Gaussians and over all the learnable parameters of the KL divergence between the prior (daggered variables) and the posterior (starred variables) of that parameter:

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}} = \sum_{k=1}^{K} \Big[&\mathrm{KL}(\mathcal{N}_{\underline{\mu}_k^\dagger,\underline{\underline{\Phi}}_k^\dagger} \,||\, \mathcal{N}_{\underline{\mu}_k^*,\underline{\underline{\Phi}}_k^*}) \\ &+\mathrm{KL}(\mathcal{N}_{\alpha_k^\dagger,\Psi_k^\dagger} \,||\, \mathcal{N}_{\alpha_k^*,\Psi_k^*}) \\ &+\mathrm{KL}(\mathcal{N}_{c_{klm}^\dagger,\Xi_{klm}^\dagger} \,||\, \mathcal{N}_{c_{klm}^*,\Xi_{klm}^*})\Big]\end{aligned} \tag{4.18}$$

4.3.3 Learning with AUSE

In order to assess the accuracy of uncertainty estimation, a quantitative approach involves examining its correlation with the true error map using the Sparsification Curve [56]. First, the predicted values are sorted based on decreasing predicted uncertainty and then progressively removed, starting from those with high predicted uncertainty. By keeping track of a quality metric applied to the remaining values, the Sparsification Curve is generated. The area under this curve is called Area Under Sparsification Curve (AUSC). The Area Under the Sparsification Error (AUSE) metric is defined as the difference between the AUSC of the method and the AUSC of the oracle, *i.e.*, the curve obtained by sorting the predicted values according to the true error.

The AUSE can be formally described as in the following. We consider a statistical model that predicts a value $\hat{y}_i$ along a predicted uncertainty $\hat{\sigma}_i$. Let $e_i = E(\hat{y}_i, y_i)$ be the error committed by the model, where y_i is the ground truth and $E(\cdot, \cdot)$ is an error function. Consider a collection of predictions of the model, with the associated ground truth $\{\hat{y}_i, \hat{\sigma}_i, y_i\}_{i=1}^{N}$. Also consider two permutations of $\{1, \dots, N\}$, namely P and Q, such that $\hat{\sigma}_{P(i)} \leq \hat{\sigma}_{P(j)}$ and $e_{Q(i)} \leq e_{Q(j)}, \forall j \geq i$ respectively. These two permutations sort the aforementioned collection according to the predicted uncertainty and the error, respectively. Given a permutation R, the Sparsification Curve is:

$$\mathcal{S}_R(n) = \frac{1}{n} \sum_{i=1}^{n} e_{R(i)} \qquad \forall n \in \{1, \dots, N\}$$

The Area Under the Sparsification Curve (AUSC) is:

$$\text{AUSC}(R) = \sum_{n=1}^{N} \mathcal{S}_R(n)$$

Finally, the AUSE is defined as the difference of AUSC(P) and AUSC(Q):

$$\text{AUSE} = \sum_{n=1}^{N} \left(\frac{1}{n} \sum_{i=1}^{n} e_{P(i)} - e_{Q(i)} \right)$$

If the uncertainty prediction was random, the percolation process would also be random, resulting in a sparsification flat curve and a high AUSE. Otherwise, if the uncertainty prediction was positively correlated with the prediction error, improvements in the tracked quality metric would be observed. As the 3DGS technique has significantly lower memory requirements compared to the NeRF used in previous works, it is capable of sampling the whole view multiple times in a single forward pass. This enables us to directly compute the AUSE metric applied to all the pixels in a view, taking the standard deviation of the samples of each pixel as the uncertainty map.

As we will show in the experiments section, using only the standard VI framework is suboptimal. Indeed, 3DGS inherently has low memory requirements and high rendering efficiency, leading to underutilized GPU memory and computational resources, even when using a standard VI framework. Hence, in this work, we propose to augment the VI loss of Eq. (4.3) with the AUSE metric. Introducing the AUSE loss enhances the framework by providing an additional optimization signal that fully leverages these available resources. Although it introduces computational overhead, this overhead is limited to operations performed on top of the S Monte Carlo renderings already required by SGS, namely the computation of per-pixel sample statistics and the sparsification-ranking procedure. Therefore, the AUSE term does not change the dominant linear dependence on S of the method, but adds an extra per-iteration cost that is moderate in practice and buys a substantial improvement in uncertainty quality, as shown in Table 4.3.

4.3.4 End-to-End SGS Training

We now have all the necessary ingredients to define the overall loss used to train the proposed SGS method. In particular, the overall loss is the following combination:

$$\mathcal{L}_{\text{SGS}} = \mathcal{L}_{\text{GS}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{AUSE}}\mathcal{L}_{\text{AUSE}} \tag{4.19}$$

where $\mathcal{L}_{\text{GS}}$ is defined in (2.16), $\mathcal{L}_{\text{KL}}$ is the Kullback-Leibler divergence with the prior from Eq. (4.18), and $\mathcal{L}_{\text{AUSE}}$ is the loss induced by the AUSE RMSE metric. The 3DGS and AUSE losses augment the conventional KL loss in order to more explicitly enforce the training tradeoff between distortion, perceptual quality and uncertainty estimation.

Finally, one more aspect in which SGS training differs from previous works is the approach to learning the distribution of the priors. In previous works, stochasticity was introduced in the weights of neural networks, which are typically randomly initialized [114]. However, in 3DGS, the parameters have a more direct physical meaning in the 3D space. For example, minimizing the KL-divergence in 3DGS would tend to fix the center of a Gaussian primitive fixed in a randomly initialized position in 3D space. Instead, we tackle this convergence issue, by taking inspiration from Empirical Bayes. We thus first learn an informative prior with some iterations of classic 3DGS, which serves as an initialization before switching to the SGS formulation.

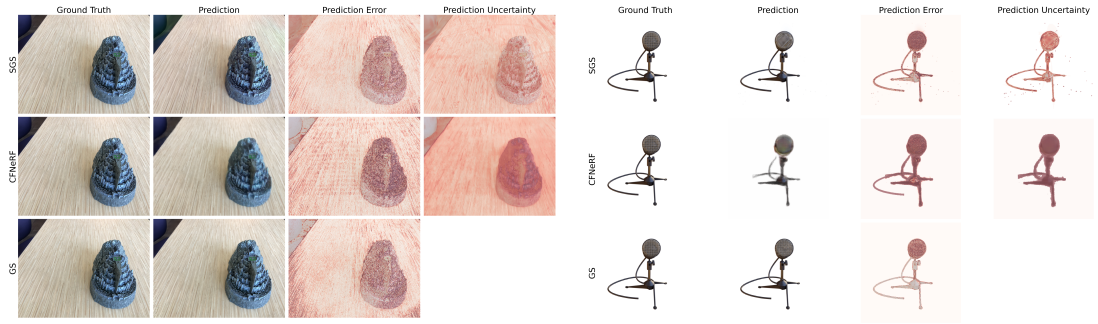


Figure 4.2: Two qualitative examples of our method SGS with CF-NeRF [137]. The last column is a visualization of the predicted uncertainty map.

4.4 Experimental Results

4.4.1 Experimental Setting

This work addresses the task of synthesizing novel views and associated uncertainty maps from multiple posed views of a static scene. These views consist of RGB photos captured from various camera positions and orientations. The spatial positions of the cameras are referred to as extrinsic parameters, while the intrinsic parameters include the camera's focal length and central point. Similar to prior research, we assume that both extrinsic and intrinsic camera parameters are known for all input views. The objective is to perform standard novel view synthesis using the provided views while also generating an uncertainty map for each synthesized view. Consequently, for every pixel in the novel view, both its color and uncertainty are predicted. It is crucial that the predicted uncertainty correlates with the true error.

We remark that this is the first method addressing uncertainty estimation for the 3DGS setting, while current literature focuses on NeRF models. This makes direct comparisons challenging, as both absolute image quality as well as metrics for measuring the effectiveness of uncertainty estimation need to be reported. In particular, we remark that the AUSE metric benchmarks uncertainty estimation against the oracle method, so it is relatively insensitive to the absolute image quality generated by each method.

We compare our proposed method with the current literature on NeRF: the state-of-the-art CF-NeRF [137], the pioneering work of S-NeRF [139], and also with NeRF-W [104], Deep-Ensembles (D.E.) [89] and MC-Dropout [48], as done in previous works. We remark that some methods [139, 137] use extra information in the form of depth maps while we do not, resulting in a setting that is slightly unfair towards SGS. Nevertheless, we report improvements over such methods. We stress that the D.E. method is defined simply as training from scratch 5 vanilla NeRF networks [109], testing them and averaging the results. This method may reach

high levels of uncertainty prediction accuracy at the cost of being 5 time slower of the "backbone" method.

Moreover, we report also the results of the vanilla 3DGS algorithm, even if it does not enable any uncertainty prediction.

As common practice, the experiments are conducted on the LLFF dataset from the original NeRF paper [109]. Moreover, while works in previous related literature [139, 137] report results only for the LLFF dataset, we also report results on the Blender and Mip-NeRF360 datasets in order to validate the robustness and generalization of SGS and to ease the comparison of SGS with other methods. In fact these datasets are a de-facto standard benchmark for general-purpose novel view synthesis methods. Note that the Mip-NeRF360 dataset is much more challenging than the other two datasets, and all the methods based on the original NeRF are not able to correctly represent the scenes therein. This highlights also the added benefit of using 3DGS as the backbone for SGS, as the vanilla 3DGS can correctly reproduce the Mip-NeRF360 dataset.

All the experiments are performed at a fraction of the original resolution, so that all synthesized images and uncertainty maps are composed by approximately 400×400 pixels.

The hyperparameters in the final loss function are: $\lambda_{\text{KL}} = 10^{-3}$ and $\lambda_{\text{AUSE}} = 5$. We use the default 3DGS hyperparameters for each dataset, except for fixing the highest spherical harmonics degree to 1.

At iteration 16000, the current learned 3DGS is fixed and taken as the prior, and the Bayesian regime is introduced. All the prior covariance matrices are initialized as identity matrices of correct dimensions scaled by 10^{-2}. The learning rate for all the posterior learnable parameters is set to 10^{-4}. Then we continue the training until iteration 30000. During both training and testing, Monte Carlo sampling from the posterior is performed for $S = 8$ times.

4.4.2 Main Results

Table 4.1 reports our results on the LLFF, Blender and Mip-NeRF360 datasets, evaluating the quality of the rendered images, as well as the reliability of the associated uncertainty maps. The rendered images are quantitatively evaluated with three metrics: PSNR as a distortion metric and LPIPS and SSIM as perceptual metrics. It can be noticed that our method improves by a large margin all these metrics, hence our method has a much lower test prediction bias with respect to previous work. The uncertainty maps are quantitatively evaluated with two metrics: AUSE RMSE and AUSE MAE. Both metrics are obtained as the area under the sparsification curve, but considering as the error metric the Root Mean Square Error and the Mean Absolute Error, respectively. As shown in Table 4.1, our method improves the AUSE RMSE metric, while keeping an AUSE MAE metric

		Rendering Metrics			Uncertainty Metrics (AUSE)	
		PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓	MAE ↓
LLFF	3DGS [82]	26.85	0.880	0.055	N.A.	N.A.
	SGS	**24.20**	**0.842**	**0.121**	**0.0147**	0.0092
	D.E. [89]	22.32	0.788	0.236	0.0254	0.0122
	Drop. [48]	21.90	0.758	0.248	0.0316	0.0162
	NeRF-W [104]	20.19	0.706	0.291	0.0268	0.0113
	S-NeRF* [139]	20.27	0.738	0.229	0.0248	0.0101
	CF-NeRF* [137]	21.96	0.790	0.201	0.0177	**0.0078**
Blender	3DGS [82]	34.50	0.986	0.011	N.A.	N.A.
	SGS	**31.13**	**0.965**	**0.021**	0.0069	0.0027
	D.E. [89]	23.72	0.909	0.070	**0.0029**	**0.0007**
	Drop. [48]	23.59	0.882	0.153	0.0564	0.0250
	NeRF-W [104]	25.90	0.912	0.092	0.0345	0.0122
	S-NeRF* [139]	N.A.	N.A.	N.A.	N.A.	N.A.
	CF-NeRF* [137]	23.26	0.860	0.160	0.0085	0.0031
Mip-NeRF360	3DGS [82]	30.66	0.923	0.033	N.A.	N.A.
	SGS	**26.30**	**0.873**	**0.091**	**0.0139**	**0.0084**
	D.E. [89]	20.72	0.487	0.533	0.0379	0.0231
	Drop. [48]	19.21	0.402	0.628	0.0536	0.0338
	NeRF-W [104]	19.40	0.457	0.503	0.0649	0.0421
	S-NeRF* [139]	N.A.	N.A.	N.A.	N.A.	N.A.
	CF-NeRF* [137]	19.51	0.422	0.633	0.0292	0.0171

Table 4.1: Quantitative results on the LLFF, Blender and Mip-NeRF360 datasets. * denotes extra depth information. The N.A. entries in the 3DGS rows are due to the method being deterministic. Instead, the N.A. entries in the S-NeRF rows are due to non-availability of public code implementation.

comparable with the state of the art, which however also exploits depth information. Figure 4.2 shows a qualitative result for a novel view generated by SGS and CF-NeRF together with the predicted uncertainty maps. We can notice that SGS is capable of producing a sharp view as well as prediction uncertainty which correlates well with the true rendering error.

4.4.3 Components Ablation

SGS imposes a prior distribution on the position, color, and opacity of the Gaussian primitives. Table 4.2 compares the full proposed SGS method with versions of it where only one attribute at a time is modeled as a stochastic variable, showing that the best uncertainty maps are obtained when all the parameters considered are stochastic.

	PSNR ↑	SSIM ↑	LPIPS ↓	AUSE RMSE ↓
SGS	28.61	0.949	**0.034**	**0.0090**
Position	27.93	<u>0.950</u>	0.043	<u>0.0105</u>
Color	<u>28.94</u>	0.947	<u>0.040</u>	0.0121
Opacity	**29.05**	**0.954**	0.043	0.0112

Table 4.2: Components ablation on *kitchen* scene of the Mip-NeRF360 dataset.

4.4.4 AUSE Loss Ablation

	PSNR ↑	SSIM ↑	LPIPS ↓	AUSE RMSE ↓
SGS	24.20	0.842	0.121	**0.0147**
$\lambda_{\text{AUSE}} = 0$	**26.65**	**0.869**	**0.082**	0.0291

Table 4.3: AUSE Loss Term ablation of LLFF dataset.

Table 4.3 compares our SGS method with an ablated version, where the proposed AUSE loss term $\mathcal{L}_{\text{AUSE}}$ in equation (4.19) is removed in order to verify its effectiveness. As reported in the table, the removal of this loss term improves the photometric reconstruction (measured by the three quality metrics: PSNR, SSIM, and LPIPS), while deteriorating the model's ability to predict accurate uncertainty maps. Hence, this ablation study proves that one of our key contribution, that is to incorporate the AUSE loss term, improves the quality of the predicted uncertainty maps. Moreover, the hyperparameter λ_{AUSE} provides a natural way to control the impact of this loss term, so that an application-specific trade-off between reconstruction quality and accurate uncertainty prediction can be found for downstream tasks.

4.4.5 Effect of Sample Size on AUSE Performance

In this section, we evaluate the quality of uncertainty estimation in SGS as a function of the number of Monte Carlo samples, S, utilized in the variational inference process. We selected the *kitchen* scene from the Mip-NeRF360 dataset as our test case. For each value of S in $\{2, 4, 8, 16, 32\}$, we ran the model and computed the AUSE metric across eight trials, averaging the results for each value of S. The average AUSE values are plotted in Fig. 4.3. The results reveal an almost linear decrease in AUSE as S doubles, indicating that increasing the number of Monte Carlo samples improves the quality of uncertainty estimation. However, this improvement comes with a tradeoff: higher values of S lead to more accurate uncertainty estimates but also result in longer runtimes due to the increased computational load required for additional sampling. Therefore, SGS allows for a balance between achieving better uncertainty estimation and managing computational efficiency, depending on the choice of S.

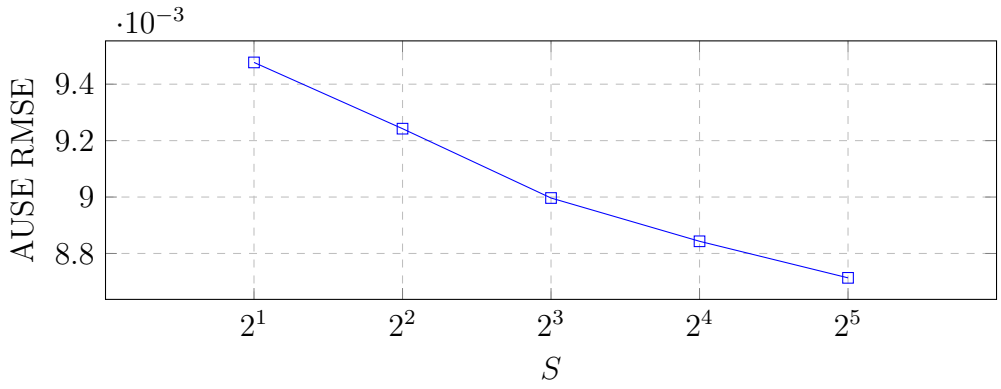


Figure 4.3: SGS trend between the number of samples S used in the Monte Carlo estimation and the resulting AUSE, in the *kitchen* scene of the Mip-NeRF360 dataset.

4.4.6 Runtime Complexity

We analyzed the runtime performance of our SGS method to compare it with baselines. From a theoretical perspective, the proposed method is S times slower than conventional 3DGS, where S is the number of Monte Carlo samples used in the variational inference process. The AUSE loss adds a further overhead during training, but this is secondary with respect to the cost of the repeated renderings: once the S samples have been produced, AUSE only requires computing the empirical uncertainty and sorting pixels according to uncertainty and error. Therefore, the dominant runtime factor remains the Monte Carlo sampling parameter S. Note that S is a dynamic parameter, *i.e.*, it can be changed even after the training phase

is completed, depending on the application needs. In the limit case $S = 1$, one recovers the classic 3DGS, losing the ability to estimate uncertainty.

We experimentally compared the runtime of the proposed SGS method with CF-NeRF. In order to have a comparison as fair as possible, we run the original evaluation code for both methods on the test split of the *hotdog* scene of Blender dataset ($N = 200$ views to be rendered) but all the disk accesses are disabled, GPU-CPU transfers are reduced as much as possible, and the same Nvidia Quadro P6000 GPU is used. Moreover, we compare both methods at an equal value of the parameter $S = 8$. We report the results measures as seconds per view per sample ($\frac{sec}{NM}$). SGS achieves $\approx 0.002\frac{sec}{NM}$, while CF-NeRF is substantially slower at approximately $4.7\frac{sec}{NM}$.

4.5 Conclusions & Future Works

In this chapter we presented Stochastic Gaussian Splatting (SGS), a Bayesian extension of 3D Gaussian Splatting for uncertainty-aware novel view synthesis. By treating the Gaussian parameters stochastically and rendering through Monte Carlo sampling, SGS predicts not only expected pixel intensities but also per-pixel predictive variances. This makes it possible to obtain uncertainty maps that correlate with rendering errors while preserving the speed and reconstruction quality that make 3DGS attractive.

Nevertheless, SGS inherits some of the main limitations of vanilla 3DGS. Like the underlying representation, it is designed for static scenes and requires scene-specific retraining, which reduces practicality when the environment changes over time. Moreover, the current formulation models only aleatoric uncertainty, while epistemic uncertainty remains largely unaddressed. This is a particularly important limitation in novel-view synthesis, since the training views only subsample the scene light field, and the uncertainty in poorly observed or entirely unseen regions is precisely epistemic in nature. In its present form, SGS can only express uncertainty where Gaussian primitives have already been placed, and cannot provide meaningful predictions for views lying outside the reconstructed scene extent. However, epistemic uncertainty can be reduced by obtaining training views that effectively enlarge the scene extent.

An interesting direction for future work is how to use the uncertainty map predicted by SGS to aid and inform the reconstruction process. Even if the uncertainty is represented as a variance on the image plane, making it non-trivial to directly map this back to the 3D space for adjusting the Gaussian parameters, it can potentially be used as a more principled densification criterion for vanilla 3DGS. Using uncertainty information, future research could focus on optimizing adaptive density control methods (densification/pruning strategies) in 3DGS to improve reconstruction quality and computational efficiency.

Chapter 5

Missing Data

So far, our discussion has largely focused on static scenes: given a fixed environment and a collection of images, we build a volumetric representation and use it to synthesize new views. The physical world, however, is rarely still. People walk through streets, clouds drift across the sky, fluids flow and splash, cameras move along complex trajectories. In many practical settings what we ultimately care about are *videos*: coherent sequences of frames that obey the laws of physics over time. At the same time, our observations are almost always incomplete. We only see a handful of viewpoints, perhaps at a few time instants, and yet we would like to imagine how the scene might have evolved in between, or under different conditions. This brings us to a complementary challenge: how to *hallucinate missing views and times* in a way that respects both data and physics.

One tempting answer is to let large video diffusion models do all the work. By training on massive text-video datasets, these models can generate strikingly realistic clips from short prompts. However, this approach comes with important limitations for our purposes. First, such models are extremely expensive to train and to deploy, making them ill-suited as general-purpose tools inside 3D reconstruction pipelines. Second, control is coarse: describing a precise motion trajectory or a specific physical process in natural language is, at best, awkward. Finally, the underlying dynamics are learned implicitly from data, which makes it difficult to guarantee physical plausibility or to impose domain-specific constraints. If our goal is to fill in missing viewpoints or time steps according to known physical laws - say, rigid-body motion, fluid dynamics, or flocking behaviour - then relying solely on black-box temporal priors seems wasteful.

In this chapter, we explore a different route with *MotionCraft*, a physics-based, zero-shot framework for video generation built on top of pre-trained image diffusion models. Rather than training a dedicated video model, we assume that we have access to a powerful still-image generator - capable of producing high-quality frames consistent with a text prompt - and we ask how to *animate* its outputs. The key idea is to let physics drive the motion while the diffusion model fills in the visual

details. Concretely, we use a physical simulator (for fluids, rigid bodies, or multi-agent systems) to generate a sequence of optical flow fields describing how scene elements should move across the image plane. Instead of applying these flows directly to pixels, we warp the internal noise latents of the diffusion model and let the denoising process reconstruct each new frame. This coupling has two important consequences: the prescribed motion is faithfully followed, and the model is free to invent new content where needed - revealing previously occluded surfaces, extending objects that move into view, or updating reflections and illumination consistently over time.

From the perspective of this thesis, MotionCraft addresses the third challenge highlighted in the introduction: handling missing views and dynamics in a principled way. By separating "what the world looks like" (handled by the image diffusion model) from "how it moves" (handled by explicit physics), we obtain a controllable and interpretable video generator that can extrapolate beyond the observed data while remaining grounded in physical laws. Even if it is now explored in this chapter, MotionCraft could be used to supply missing views to a 3D reconstruction pipeline.

5.1 Background

Denoising Diffusion models [142, 144, 61] represents a generative modeling approach that leverage a noise diffusion process to model a data distribution starting from random noise. These models are based on a predefined Markovian forward noising chain that progressively adds Gaussian noise to the data $\underline{x}_0$ in an iterative procedure of T steps. The reverse diffusion process traverses back the Markov Chain and can be written as:

$$p_\theta(\underline{x}_{0:T}) = p(\underline{x}_T)\prod_{t=1}^{T} p_\theta(\underline{x}_{t-1} \mid \underline{x}_t) \qquad p_\theta(\underline{x}_{t-1} \mid \underline{x}_t) = \mathcal{N}(\underline{x}_{t-1} \mid \nu_\theta(\underline{x}_t, t), \sigma_t^2 \mathbb{I}) \ . \tag{5.1}$$

The training phase optimizes the parameters of the reverse process p_θ maximising an evidence lower bound (ELBO) over the target data. The work of [143] shows that is possible to construct a non-Markovian process defining a faster sampler (DDIM) that is compatible with the pretrained model. So starting from $p_\theta(\underline{x}_{0:T})$, it is possible to sample $\underline{x}_{t-1}$ using:

$$x_{t-1} = \sqrt{1-\beta_{t-1}}\left(\frac{x_t - \sqrt{\beta_t}\hat{\epsilon}_t}{\sqrt{1-\beta_t}}\right) + \sqrt{\beta_{t-1} - \sigma_t(\eta)^2}\cdot\hat{\epsilon}_t + \sigma_t(\eta)\varepsilon_t \tag{5.2}$$

where β_t determines how much noise is added at each step t, $\sigma_t(\eta) = \eta\sqrt{\frac{\beta_{t-1}}{\beta_t}}\sqrt{\frac{\beta_t}{1-\beta_{t-1}}}$ and $\eta \in (0,1)$ is a parameter controlling the forward process. When $\eta = 0$, the sampling becomes deterministic, when $\eta = 1$, the process result in Denoising Diffusion

Probabilistic Models (DDPM) sampling. $\hat{\epsilon}_t$ is the estimated noise present in x_t, typically estimated with a UNet architecture [130]: $\epsilon_t(\cdot)$. Finally, ε_t is an independent normal stochastic variable. In this work we employ a Latent Diffusion Model [129] that perform the diffusion process over a compressed latent space, reducing the computational burden of training in pixel space, while keeping high perceptual quality. Before the diffusion process, a VQ-VAE [158] is trained; the input image is then encoded by the VQ-VAE Encoder that reduces the spatial dimension. The generated features are decoded back to the image space when generating images by means of th VQ-VAE Decoder. The UNet architecture is tipically composed by convolutional layers followed by spatial self-attention layers and cross-attention conditioning layers. Recent works [83, 59, 22] propose to reprogram this mechanism to enhance consistency between frames by letting the currently generated frame to attend to the first frame by swapping the original attention keys (K) and values (V) with the keys and values of the first frame, leading to the Cross-Frame Attention (CFA) mechanism:

$$\text{Cross-Frame-Attn(Q,K,V)} = \text{Softmax}\left(\frac{Q^f \cdot K^1}{\sqrt{d_k}}\right) V^1 \tag{5.3}$$

where V^1 and K^1 represent the keys and values of the first frame, while Q^f represents the queries of the current frame, and d_k is the channel dimension of the keys. In this work we will use the notation $\epsilon_t(z, \mathcal{P}; \{a, b, c, \dots\})$, where z is a latent, $\mathcal{P}$ is the prompt, and $\{a, b, c, \dots\}$ is a *list* of latents to attend to, as MCFA enables to attends to a list of latents and not only to a single one.

Classifier-Free Guidance (CFG) [62] is a widely used technique to guide the conditional generation process using a linear combination of conditional and unconditional estimated scores:

$$\hat{\epsilon} = \epsilon_t(z, \mathcal{P}_\emptyset, \{\dots\}) + \lambda\left[\epsilon_t(z, \mathcal{P}, \{\dots\}) - \epsilon_t(z, \mathcal{P}_\emptyset, \{\dots\})\right] \tag{5.4}$$

where λ is the guidance scale, $\mathcal{P}_\emptyset$ represents the null condition and $\mathcal{P}$ is the target text prompt. Intuitively, the unconditional term preserves the natural image prior of the model, while the difference term pushes the sample towards better agreement with the text condition. Larger values of λ typically increase prompt adherence, although overly large values may also amplify artifacts or reduce diversity.

5.2 Introduction

As human beings, we have always exploited our creativity to generate art, in different forms such as visual art, music or poetry. In vision, we are often inspired by the natural world since our visual system continuously acquire images perceived as a video sequence. Indeed, videos or movies are one of the best visual stimuli since they contain images, motion and audio.

Recent generative models for still images based on diffusion models [142, 143, 129] achieved remarkable results with quality almost indistinguishable from real images. It is therefore clear that the next big goal is video generation. However, it seems that including the dimension of time remains challenging. Some works such as Sora [23] achieve astonishing temporal consistency and photorealism at the expense of enormous computational and data requirements. Moreover, we argue that fine-grained control over the motion dynamics is impossible with a simple text prompt. If one wants to synthesize a video according to some precise physical dynamics, they would not be able to do it with current models. Interestingly, explicitly controlling the motion dynamics also allows to decouple temporal evolution from content generation. Indeed, explicitly injecting the physics of the real world as motion dynamics allows to develop more parsimonious models, that do not need to brute-force learn them from data.

For this reason, in this chapter, we investigate the possibility to create a zero-shot video generation model that only requires a pretrained still image generator and knowledge of physical laws regarding motion. Indeed, since videos are temporal sequences of images correlated by physical laws, we only need to devise a way to include physical laws in the diffusion prior to animate a starting image. We thus advocate for physics simulators as appropriate sources of motion, output as a sequence of optical flows, while also being completely user-controllable, plausible, and explainable.

We propose MotionCraft, a physics-based zero-shot video generator that uses optical flow extracted from a physical simulation to warp the noise latent space of a pretrained image diffusion model to generate videos with complex dynamics without the need to train anything. While using a projection of motion onto the camera plane as a pixelwise displacement field (optical flow) may seem limiting due to the fact that, if applied in the pixel space, it would not be able to synthesise novel coherent content but only displace pixels, the trick lies in its application in the noise latent domain. Backed by evidence that motion vectors correlate between pixel and noise space, warping of the latter by means that MotionCraft allows to simultaneously apply the desired motion and exploit the powerful image prior of the generative model. This is capable of adapting the scene to the prescribed motion without significant artefacts, generate novel content and shows impressive global consistency (reflections, illumination, etc., consistent with the desired evolution).

We present quantitative and qualitative experimental results where we show that MotionCraft is capable of synthesising realistic videos with finely controlled temporal evolution governed by fluid-dynamics equations, rigid body physics, and multi-agent interaction models, while zero-shot state-of-art techniques cannot.

We remark that synthetic video generation is a powerful technology that can be misused to create fake videos, hence it is important to limit and safely deploy these models. From a safety perspective, we emphasize that MotionCraft does not add any new restrictions nor does it relax any existing ones with respect to our base

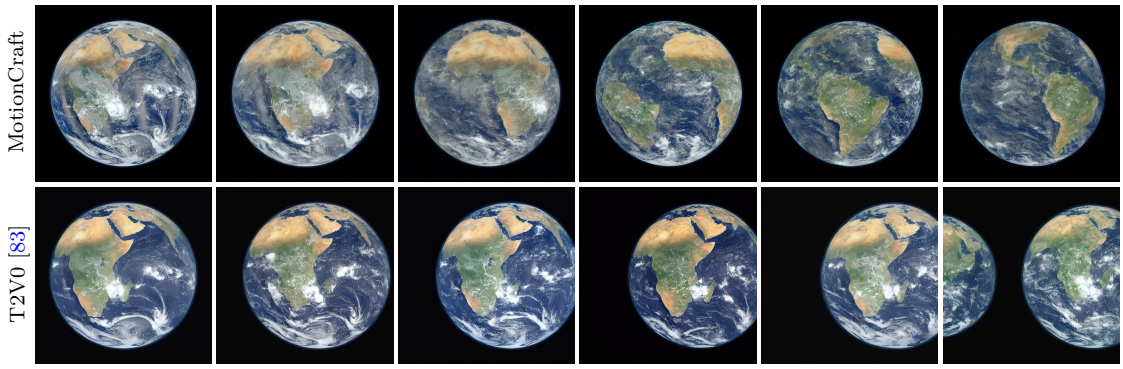


Figure 5.1: Rigid motion simulation: *Earth*. MotionCraft uses a fluid dynamics simulation to warp noise latents and synthetize video frames. T2V0 [83] is unable to simulate the evolution of the rotating Earth and simply moves the object towards the right of the frame.

text-to-image model. Moreover MotionCraft, using existing text-to-image diffusion models, does not need extra training or adjustments. This means we avoid the large environmental costs associated with training new models. One possible broader impact of MotionCraft is its usage by scientists across various fields to visualize their simulations, thereby offering AI-based visualization of physical processes to a wider scientific audience.

5.3 Related work

Diffusion Based Video Generation Video Generation [13] is a longstanding problem in computer vision aiming to learn the distribution of and synthesise realistic videos. Recently, text-based Denoising Diffusion Probabilistic Models (DDPM) [142, 145] have been studied to tackle this challenge delivering impressive results. These approaches include Sora [23], Video Diffusion models [64], Imagen-video [63] and Align your Latents [18]. They require sophisticated spatio-temporal denoising architectures at the expense of huge computational requirements and large amounts of paired text-video data for training. To reduce the data requirements, different approaches investigate few-shot and unsupervised learning techniques. Make-a-Video [141] proposes an unsupervised training with only videos, coupled with a retrieval strategy to sample using text. On the other hand, Ni et al. [115] train a diffusion-based optical flow generator that outputs a flow conditioned on a reference image and a textual prompt, that reduces the computational burden of generating videos by training the diffusion process on small flow fields. Differently from them, MotionCraft is zero-shot and does not require any training of additional components.

To the best of our knowledge, Text-to-video-Zero [83] and Generative Rendering [25] are the only zero-shot video generators. However, Generative Rendering (concurrent work, with no code available) has significant extra requirements beyond Stable Diffusion (SD) as image generator, in the form of a depth-conditioned ControlNet [181], and a 3D mesh manually animated, leveraging UV maps to render the scene. Moreover, Generative Rendering cannot render fluids, since they are difficult to represent as 3D meshes.

In this chapter, we compare MotionCraft to Text-to-video-Zero (T2V0), as zero-shot video generator baseline. T2V0 applies a constant shift (with a fixed direction) to the initial latent noise of SD, sampling each frame sequentially by means of DDPM. As shown in Section 5.4, since the motion in the noise latent space directly translates into the motion of the pixel space, the generated videos result in a overall shift in the same fixed direction. The largest part of the motion is caused by the stochastic fluctuations of the DDPM sampling strategy leading to unnatural motion and inconsistency of the objects in the different frames. On the contrary, in this work, we avoid the use of a constant warping operation derived from physics simulation flows in the latent space in order to incorporate complex motion dynamics.

Diffusion Based Video and Image Editing Recently, different methods exploit the prior of text-to-image diffusion models for video editing. In particular, Tune-A-Video [172] finetunes a text-to-image diffusion model to edit a video. They start from the inverted frames in the latent space and use the text prompt as an editing tool. Pix2Video [27] employs a self-attention injection mechanism to edit videos using a pretrained image diffusion model.

Other methods use the optical flow to edit reference images or videos. Motion Guidance [50] leverages a user defined optical flow that allow zero-shot image editing. It works by guiding the diffusion sampling process with the gradient from a pretrained optical flow network via a guidance loss. LatentWarp [8] and TokenFlow [51], use an optical flow estimated from a reference video to warp the latent space of the diffusion model to achieve consistent editing. These methods leverage both diffusion models priors and other components such as ControlNet for structural control, and trained flow estimators such as RAFT [154]. Alternatively, we propose MotionCraft, a zero-shot video generation method, using only vanilla SD. This means that MotionCraft does not require a reference video but it can animate an image, generated by the SD model or obtained by inverting a real one. Moreover, the physics simulations allow to generate different videos from the same starting image.

5.4 Optical Flow is Preserved in Latent Space

MotionCraft stems from a key observation: the optical flow estimated between two frames in the pixel space is correlated with the flow estimated between the corresponding noise latent representations of SD. We conjecture that this is related to the specific design of the SD variational auto-encoder and denoiser architectures. In fact, by largely using convolution operations, they enforce a locality prior which preserves spatial information to some extent.

In order to empirically investigate this phenomenon, we conducted a quantitative experiment using the MSU Video Frame Interpolation Benchmark dataset [54], considering only real videos. For each pair of consecutive video frames, the following steps have been taken. We first estimate the optical flow in the RGB space by using a well-established method, based on the Gunnar Farneback's algorithm, provided by OpenCV [73]. Then, we compute the noise latent representations of the two frames, first encoding the image in the variational autoencoder (VAE) of SD at timestep $\tau = 0$, followed by DDIM inversion [143] up to timestep $\tau = 400$ (same value for all experiments in this work, empirically determined). Finally, a correlation coefficient based on cosine similarity is computed between the optical flows estimated in the RGB and noise latent spaces. The resulting correlations are then averaged across all pairs of consecutive frames in the dataset, obtaining an average value of 0.727, which indicates a strong correlation between the optical field in the RGB and noise latent domains. An example of this experiment is presented in Fig. 5.2, showcasing the two estimated flows in the image and latent space and their correlation.

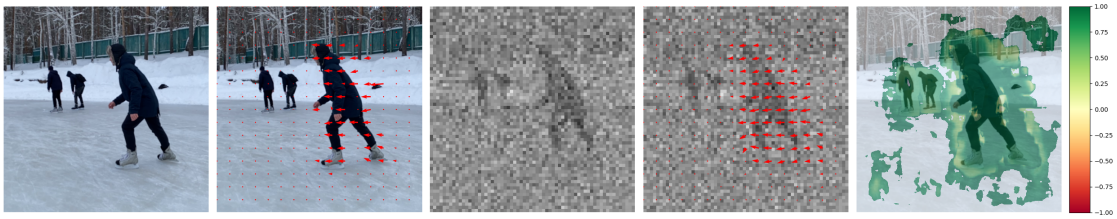


Figure 5.2: A qualitative example of the image and latent flows correlation. This figure shows, from left to right, (a) the first RGB frame, (b) the second RGB frame superimposed with the estimated flow in the RGB domain, (c) the first latent frame, (d) the second latent frame superimposed with the estimated flow in the latent domain and (e) the correlation map of the two non-zero flows.

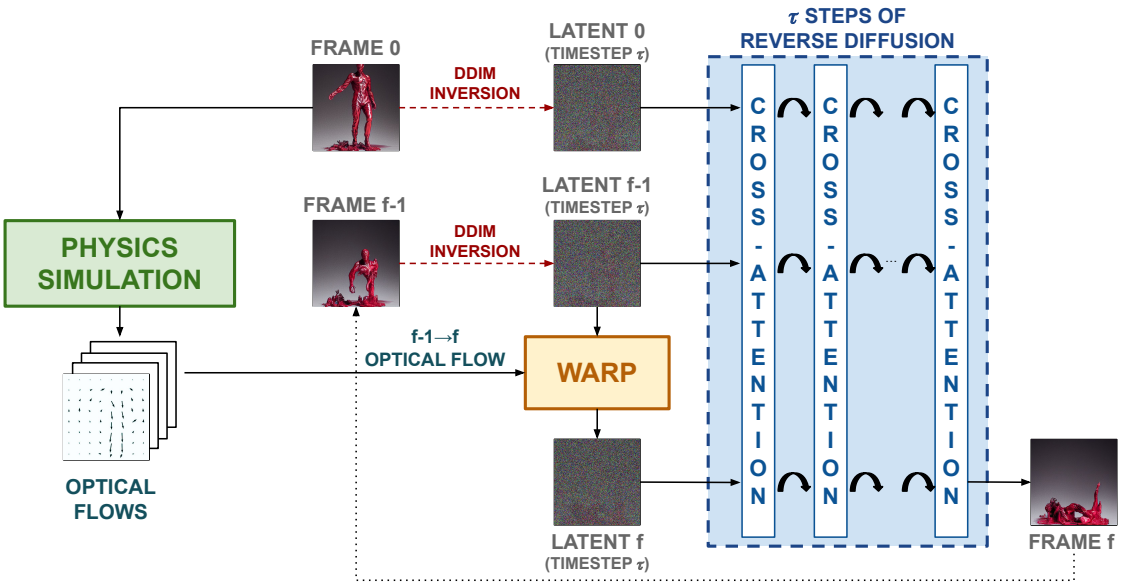


Figure 5.3: MotionCraft overview. A video is generated from a starting image using a pretrained still image generative model by warping noise latents according to an optical flow description of the motion to be synthesised.

Algorithm 1: Pseudocode of MotionCraft

Input : $I^0, \mathcal{W}, \eta, \mathcal{P}, \mathcal{P}_\emptyset$
Output: $I^0, \ldots, I^{N-1}$

for $f = 1$ **to** $N-1$ **do**
  $z_0^{f-1} = \mathrm{Enc}(I^{f-1})$; // Encode the frame
  for $t = 0$ **to** $\tau - 1$ **do** // Inversion loop
    $\hat{\epsilon} \leftarrow \epsilon_t(z_t^{f-1}, \mathcal{P}; \{z_t^{f-1}\})$; // Self-Attention, No MCFA
    $z_{t+1}^{f-1} \leftarrow \mathrm{DDIMInversion}_{t \to t+1}(z_t^{f-1}, \hat{\epsilon}, 0)$; // $\eta = 0 \iff$ DDIM
  end
  $\zeta_\tau^f = \mathcal{W}^{f-1 \to f}(z_\tau^{f-1})$; // Warp the latent
  for $t = \tau - 1$ **to** 0 **do** // Generation loop
    $\hat{\epsilon}_\mathcal{P} \leftarrow \epsilon_t(\zeta_{t+1}^f, \mathcal{P}; \{z_t^0, z_t^{f-1}\})$; // MCFA with I^0 and I^{f-1}
    $\hat{\epsilon}_\emptyset \leftarrow \epsilon_t(\zeta_{t+1}^f, \mathcal{P}_\emptyset; \{z_t^0, z_t^{f-1}\})$; // MCFA with I^0 and I^{f-1}
    $\hat{\epsilon} \leftarrow \hat{\epsilon}_\emptyset + \lambda(\hat{\epsilon}_\mathcal{P} - \hat{\epsilon}_\emptyset)$; // Classifier-free guidance
    $\zeta_t^f \leftarrow \mathrm{DDIM}_{t+1 \to t}(\zeta_{t+1}^f, \hat{\epsilon}, \eta^f)$; // Perform Spatial-η
  end
  $I^f \leftarrow \mathrm{Dec}(\zeta_0^f)$; // Decode the latent
end
return $I^0, \ldots, I^{N-1}$

5.5 Method

Based on the analysis presented in the previous section, we propose a novel zero-shot video generation method, named MotionCraft, where an image (real or generated), serving as a starting frame I^0, is animated according to a physical simulation, by means of a (possibly time-varying) optical-flow generator $\mathcal{W}$ in the noise latent space. The outcome is a video made of N frames $I^0, \dots, I^{N-1}$ that follows the motion prescribed by the physical simulation and evolves the content of the first frame coherently. Inspired by the previous observation, this animation is obtained by warping the noisy latent representation of an image in the latent diffusion space. Regarding the physics simulation for the optical flow generation, we use different libraries to simulate different physics, as explained in the experimental section, such as fluid dynamics, rigid motion and multi-agent systems. It is also possible, albeit not shown in this work, to use animation software to generate the required optical flows.

We remark that the meaning of "zero-shot" is twofold: we do not train or finetune any component of the text-to-image diffusion model, nor do we use a reference video or optical flow estimator as a starting point. In the following, we use Stable Diffusion as the pretrained text-to-image model.

Figure 5.3 illustrates an overview of MotionCraft highlighting the autoregressive generation of the video. At each iteration $f \geq 1$, the frame I^f is generated using only the information contained in the first frame I^0 and the previous frame I^{f-1}. Given this Markovian structure, MotionCraft is characterized by $\mathcal{O}(1)$ space complexity and $\mathcal{O}(N)$ time complexity with respect to the total number N of frames to be generated. More in detail, first, the two RGB frames I^0, I^{f-1} are encoded into the latent space and they are independently inverted with the reversed DDIM sampling scheme up to a fixed diffusion timestep τ, obtaining z_τ^0, and z_τ^{f-1}, respectively. Then, the optical flow warping operator $\mathcal{W}^{f-1 \to f}$ prescribed by the physical simulation is applied to z_τ^{f-1}, obtaining ζ_τ^f. Finally, the next RGB frame I^f is generated by performing τ steps of reverse diffusion using the DDIM sampling scheme with a novel cross-frame attention mechanism and a novel spatial noise map η^f weighting technique, explained below. Furthermore, we exploit the classifier-free guidance (CFG) technique for generation proposed in [62]. In practice, at each reverse diffusion step we evaluate the UNet twice on the same latent: once with the text condition $\mathcal{P}$ and once with the null condition $\mathcal{P}_\emptyset$, and then combine the two predictions with guidance scale $\lambda > 1$, as reported in Section 5.1. In our setting, CFG is used only during frame generation (lines 9–12 of Algorithm 1) and not during inversion, since we empirically found the latter to be less stable.

Algorithm 1 reports the pseudocode of MotionCraft. Lines $2-6$ include the DDIM inversion up to timestep τ. Starting current frame I^{f-1} that was previously generated, in line 2 we embed it with the VAE encoder, obtaining z_0^{f-1}. Then we apply DDIM inversion on z_0^{f-1} for τ timesteps (line $3-6$). This involves the

UNet with the standard self-attention (note the repetition of the noisy latent z_t^{f-1}) and the positive prompt $\mathcal{P}$. As briefly reported in [110], we have also experienced that DDIM inversion is not compatible with CFG; hence, during the inversion, we do not use the negative prompt $\mathcal{P}_\emptyset$. The resulting estimated noise is used in line 5 for applying the DDIM inversion step (note that the $\eta = 0$, so pure DDIM is performed). Upon completion of the DDIM inversion process, we obtain z_τ^{f-1}, the noisy latent corresponding to the frame I^{f-1}.

In line 7, the optical flow warping operator $\mathcal{W}^{f-1\to f}$ is applied to the noisy latent of the current frame z_τ^{f-1} to obtain a new noisy latent ζ_τ^f that will generate the successive frame. Finally, in lines $8-14$, the frame is generated. During this generation phase we use CFG to increase the quality of the generated images; therefore, for each denoising step we compute both the conditional prediction $\hat{\epsilon}_\mathcal{P}$ and the unconditional one $\hat{\epsilon}_\emptyset$, and combine them through the guidance scale λ, as shown in lines 9–11 of Algorithm 1. This lets us preserve the motion and structure induced by the warped latent while strengthening consistency with the prompt. To create new content while preserving the original image, we propose two direct generalization of two known techniques: the multiple cross-frame attention (MCFA) mechanism and a spatial noise map weighting (Spatial-η).

The MCFA technique generalizes the Cross Frame Attention (CFA) [83], as it enables the to-be-generated frame to attend to an arbitrary number of frames. We choose to attend to the first frame and the previous frame (as shown in lines $9-10$ of Algorithm 1 and Fig. 5.3) to ensure long-range and short-range temporal consistency, respectively. MCFA intervenes in all the self-attention blocks of the SD UNet, by replacing the keys and values, that are originally computed from projections of the generating frame features, with the ones computed from the attended frames.

We also propose Spatial-η (line 12), that is a novel technique that enables to choose, on a pixel-by-pixel basis, whether to use DDIM or DDPM as a sampling scheme. This enables the usage of DDPM in regions of the images where novel content should be created (for example, when a new part of an object is entering the scene), while using DDIM in the other regions to ensure consistency and determinism where the already-present content is just moving. Note that this spatial map η^f can be obtained in multiple ways from the physical simulation. For example, η^f can be set to 1 in regions of the image where the flow is not well-defined (pointing outside of the image boundaries) or in regions where the optical flow field has discontinuities.

5.6 Experimental Results

5.6.1 Experimental Setting

In this section, we show examples of video generation based on different physics simulations: rigid body motion, fluid dynamics and multi-agent systems. Given an optical flow, we apply it on the SD latent space using MotionCraft. We used the following hyperparameters throughout the work if not explicitely said otherwise. We set $\tau = 400$, the number of inference steps (both for DDIM inversion and for inverse diffusion) is set to 200 and the used latent space diffusion model is `runwayml/stable-diffusion-v1-5` (license CreativeML Open RAIL-M). The choice of the inversion timestep τ reflects a trade-off. If τ is too small, the latent remains too close to the encoded image and the subsequent warping does not provide enough flexibility to synthesize newly revealed content. If τ is too large, the inversion moves the sample too far into the noisy regime, making it harder to preserve fine appearance details and temporal consistency during reconstruction. We found $\tau = 400$ to be a good compromise across all experiments, providing enough noise for content completion while still preserving the structure of the previous frame. Similarly, we use 200 inversion and denoising steps as a practical balance between quality and runtime. All our experiments are done on a single NVIDIA A6000 (48GB); video generation runs in minutes (1-5min) on a single GPU. We compare MotionCraft to Text2Video-Zero (T2V0) [83] that, to the best of our knowledge, is the only diffusion-based zero-shot method for video generation.

We show qualitative results in Figs. 5.1, 5.4 to 5.6 and 5.8, which we separately describe in the following sections. Table 5.1 reports two metrics to evaluate the quality of the generated videos. As done in previous works, we use the *Frame Consistency* metric, defined as the average cosine similarity of the CLIP embeddings of consecutive frames. However, this metric presents some limitations, as CLIP focuses on high-level semantic features and not on low-level details, resulting in high correlations even if the content changes but its semantics do not (as an example, the T2V0 simulation shown in Fig. 5.6 has a *Frame Consistency* of 0.97 even if the dragons are not the same dragons in each frame). To overcome some of these limitations, we propose a novel metric, named *Motion Consistency*, that measures how similar two frames are while accounting for the motion between them. We start from the observation that, if an object moves through the scene, its textures should remain almost the same, and, if we know its flow, we can bring back that object to overlap with its starting position. Then we can apply a similarity distance between the initial image and the next frame brought back by the reversed flow. Given two consecutive frames, we use a high-quality flow estimator (RAFT [154]) to estimate the optical flow between them and apply it on the second frame to reverse the motion. Then we compute the SSIM metric [166] on the first frame and the registered one.

Physics	Name	Frame Consistency ↑		Motion Consistency ↑	
		T2V0	**MotionCraft**	T2V0	**MotionCraft**
Rigid Body	Earth	**0.9812**	0.9696	**0.7213**	0.6783
	City	0.9588	**0.9875**	0.2852	**0.9219**
Fluids	Statue	0.9463	**0.9566**	0.7817	**0.8252**
	Dragons	0.9664	**0.9991**	0.6846	**0.9637**
Agents	Birds	0.9765	**0.9968**	0.8973	**0.9385**
	Average	0.9658	**0.9819**	0.6740	**0.8655**

Table 5.1: Quantitative results comparing MotionCraft to T2V0 [83] on different physics-based video generation tasks. We report the Frame Consistency and the proposed Motion Consistency metrics (higher is better).

Figure 5.4: Rigid motion simulation: *city*.

5.6.2 Rigid Body

Figure 5.4 shows a pivotal example where MotionCraft can be directly compared to the state-of-the-art T2V0, as in this case we use an optical flow equivalent to a their proposed shift along the vertical axis. This example shows a video generated starting from a satellite view of a city, and, by simulating the rectilinear motion of the satellite, new portions of the city appear from the top of the image. While T2V0 struggles with keeping temporal consistency, even with large structural elements (*e.g.*, the river), MotionCraft is able to coherently scroll down the already present part of the city, while also generating new plausible content in the top part of the frames.

A similar case study is the Earth rotation in Fig. 5.1. Here, the optical flow is obtained by simulating a rotating sphere that was fitted to the first frame while keeping track of the starting and ending position of each point. As the Earth rotates, a slice disappears from one side and a new one needs to be generated on the opposite side. Thanks to the powerful natural image prior of SD, MotionCraft

Figure 5.5: Fluid simulation: *Statue*.

is able to autonomously generate other continents in the correct position, even if the text prompt contains no reference about them (see Section 5.7 for all the text prompts used in this chapter). On the contrary, T2V0 is not able to rotate the Earth consistently while creating new content, as visible in the same Fig. 5.1.

5.6.3 Fluids

In this set of experiments, we use the Φ-flow library [65] to simulate fluid dynamics based on the shape and position specified in the first frame I^0. The simulation can be configured in several ways depending on the numerical solver employed, Eulerian (grid-based) or Lagrangian (particle-based). We can also introduce rigid obstacles into the scene or define initial velocity and force fields. From each simulation, we extract the resulting velocity field and use it as a proxy for optical flow.

We found the Eulerian simulations to be more stable and easier to configure. Figures 5.5 and 5.6 illustrate two examples of fluid-dynamics-based motion generated with MotionCraft. The system exhibits compelling emergent behavior not explicitly encoded in the optical flow. For example, in Fig. 5.6, the global scene illumination appears to respond to the fire dynamics, while in Fig. 5.5 the fluid shows realistic bouncing and splashing interactions with the ground.

To also demonstrate the capabilities of Lagrangian simulations, Fig. 5.7 presents an example of liquid being poured from a jug into a glass, modeled with interacting particles and two rigid obstacles. The same figure compares this result with optical flow applied directly in image space, without the use of a diffusion model. As shown, applying optical flow in image space introduces noticeable artifacts - such as distortions in the glass and unnaturally smoothed liquid surfaces caused by pixel stretching. In contrast, applying the same flow within MotionCraft's noisy latent space yields a more realistic video, avoiding these deformations.

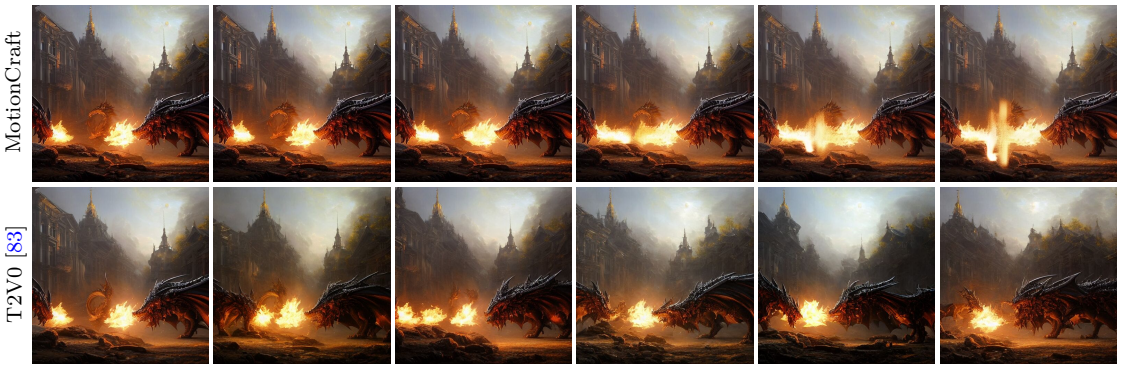


Figure 5.6: Fluid simulation *Dragons*.

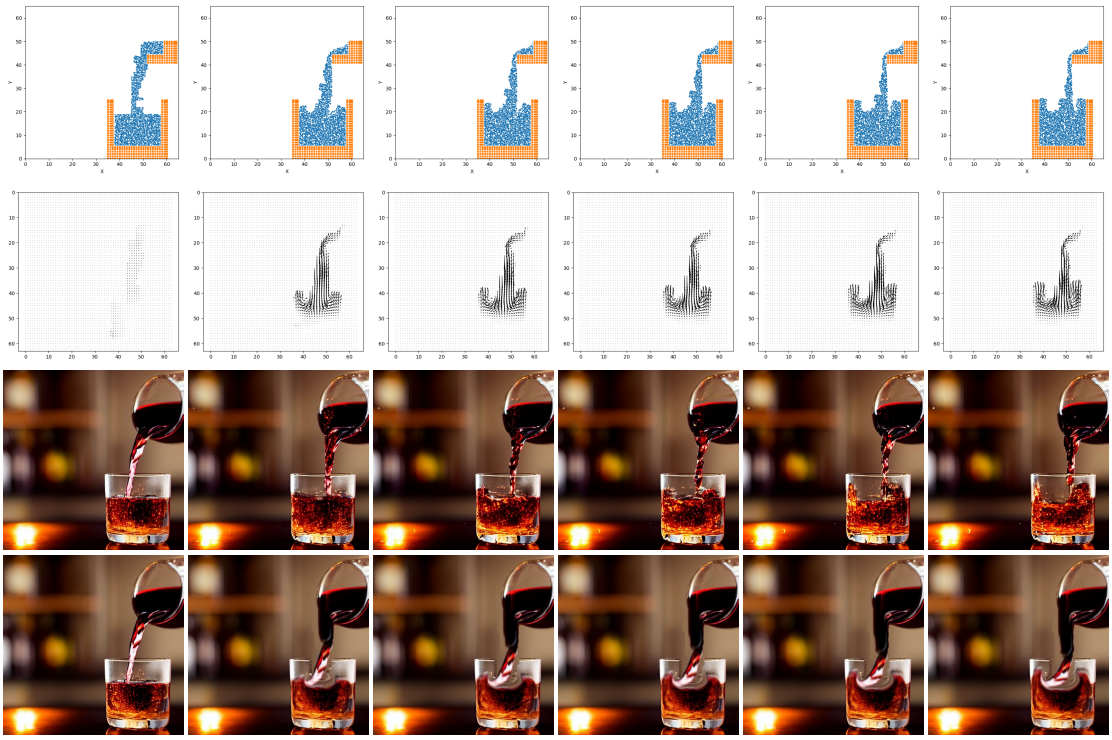


Figure 5.7: Lagrangian fluid simulation: *drink*. From top row to bottom row: particle positions at different time-steps; optical flow derived from the particle simulation; video generated with MotionCraft; video generated by applying the optical flow directly to the pixel-space.

5.6.4 Multi-Agent Systems

Multi-agent systems are another interesting family of simulated dynamics. A simple agents model is the *Boids* model [127], consisting of a set of point-like agents (named boids) that move according to three steering behaviour rules: separation,

Figure 5.8: Multi-agent system simulation: *birds*.

as boids avoid collisions with nearby agents by steering away from them, alignment, as boids align their direction with that of nearby agents, and cohesion, as boids move towards the average position of nearby agents to stay together as a group. To simulate this system we used the agentpy [42] library, in which the number of agents, the simulation time-steps and different physical parameters related to the steering rules can be chosen.

An example is shown in Fig. 5.8, generating the temporal evolution of a flock of birds. As SD is not able to generate images with a controllable number of agents in specified positions, we start from an image where there is a single agent (a bird in the example). Then, we extract the corresponding latent vector patch with the attention map [40] related to the CLIP token containing the word "bird", and clone it to the simulated positions of the other agents. At this point, we evolve the frames according to the optical flow derived from the simulation velocity field. While MotionCraft produces a realistic flock motion, T2V0 motion is not consistent and the number of birds changes in each frame.

5.6.5 Ablations

In this section, we ablate the contribution of the most important components and hyperparameters on the *Earth*, *city* and *dragons* scenes.

First, we start from investigating the impact of the cross-attention mechanism by comparing four different variants: i) each frame attends to itself (no MCFA); ii) each frame attends to the previous frame; iii) each frame attends to the first frame; iv) each frame attends to both the previous frame and the first frame (proposed MCFA). Visual results are shown in Fig. 5.9. As can be seen, the MCFA mechanism is necessary to generate plausible frames; moreover, attending only to the first frame reduces the overall motion, (*e.g.*, always showing Africa as in the first frame), while only attending to the previous frame reduces color consistency. Overall, we demonstrate that the proposed MCFA, attending to both the first and the previous

frame, represents the optimal solution to keep global consistency with the initial image and local consistency with the preceding frame.

Secondly, we ablate the Spatial-η weighting technique by comparing two variants: without Spatial-η, *i.e.*, setting $\eta = 0$ everywhere resulting in no DDPM sampling, and with Spatial-η. Visual results are shown in Fig. 5.10. As shown, being able to sample with DDPM in some parts of the image is crucial in order to generate novel plausible content. Indeed, DDPM adds, during each reverse diffusion step, random white noise to the latent. We suppose that this allows to better sample from the real distribution, avoiding artefacts other components of the method, such as the warping operator or the MCFA, would otherwise introduce.

Finally, we ablated the partial inversion process, *i.e.*, lines 2-6 in Algorithm 1. Without the DDIM inversion, textures and details generated by SD cannot be brought into the next frame, resulting in corrupted videos. Visual results can be found in Fig. 5.11

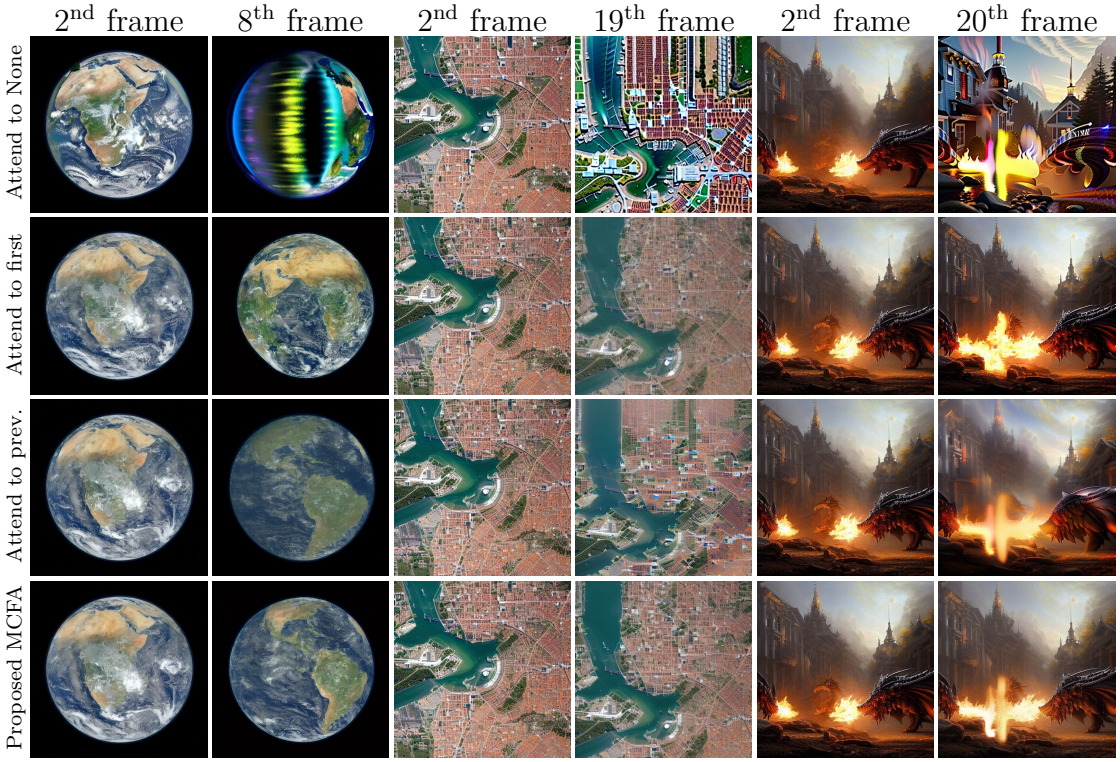


Figure 5.9: Cross-Frame attention ablation. We show the extremal frames generated under different cross-frame attention configurations (from top to bottom): no cross frame attention, attend only to the initial frame, attend only to the previous frame, attend to the initial and preceding frame (as the proposed MCFA mechanism).

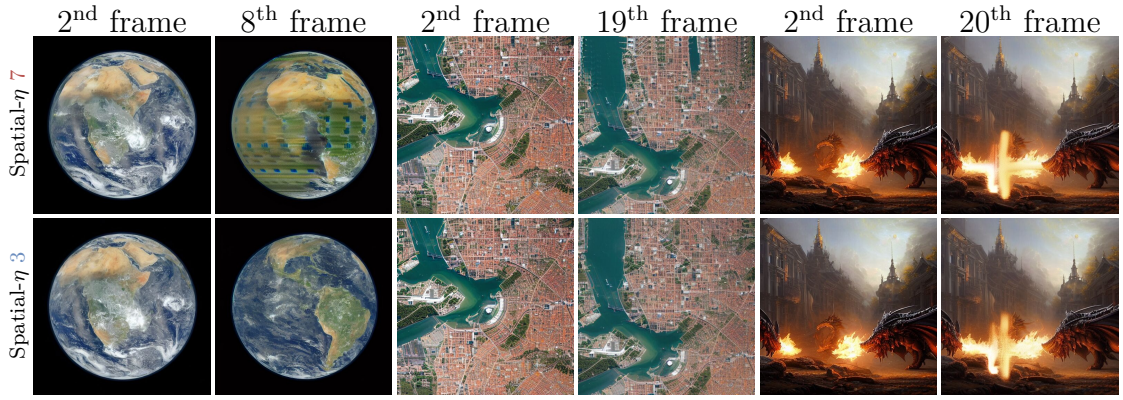


Figure 5.10: Spatial-η ablation. We show the extremal frames generated without (first row) and with (second row) the proposed Spatial-η conditioning mechanism.

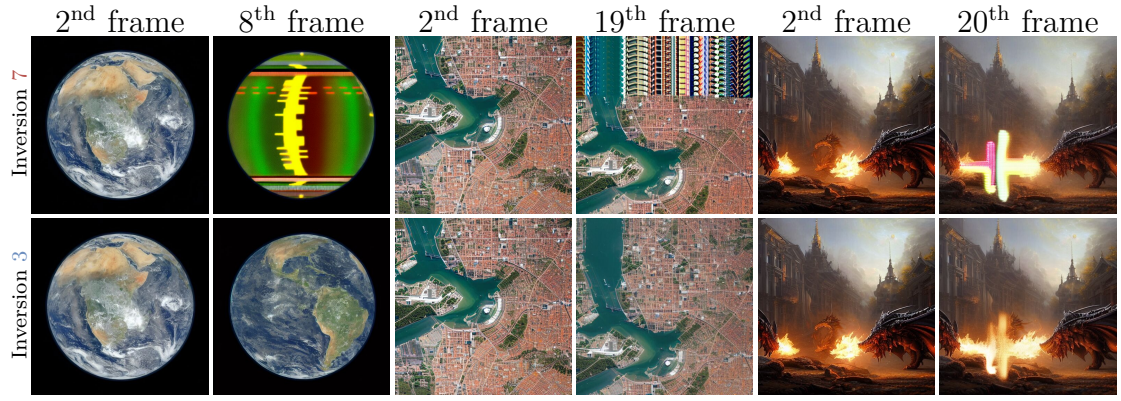


Figure 5.11: Inversion mechanism ablation. We show the extremal frames generated without (first row) and with (second row) the proposed inversion mechanism.

5.6.6 Additional Qualitative Results

Figure 5.12 shows some additional results of MotionCraft. The first row shows a tree growing. This video was obtained using a simple constant outward-facing radial optical flow applied only on the foliage. Note that while the tree grows, its shadow evolution is coherent. As the input flow is zero in this part of the image, the shadow consistency is recovered only by Stable Diffusion. The last row show a video obtained by applying MotionCraft to SDXL. This shows the generalizability of MotionCraft to different diffusion models, with also different resolutions. Hence, MotionCraft is able to produce high-resolution videos with a high level of detail.

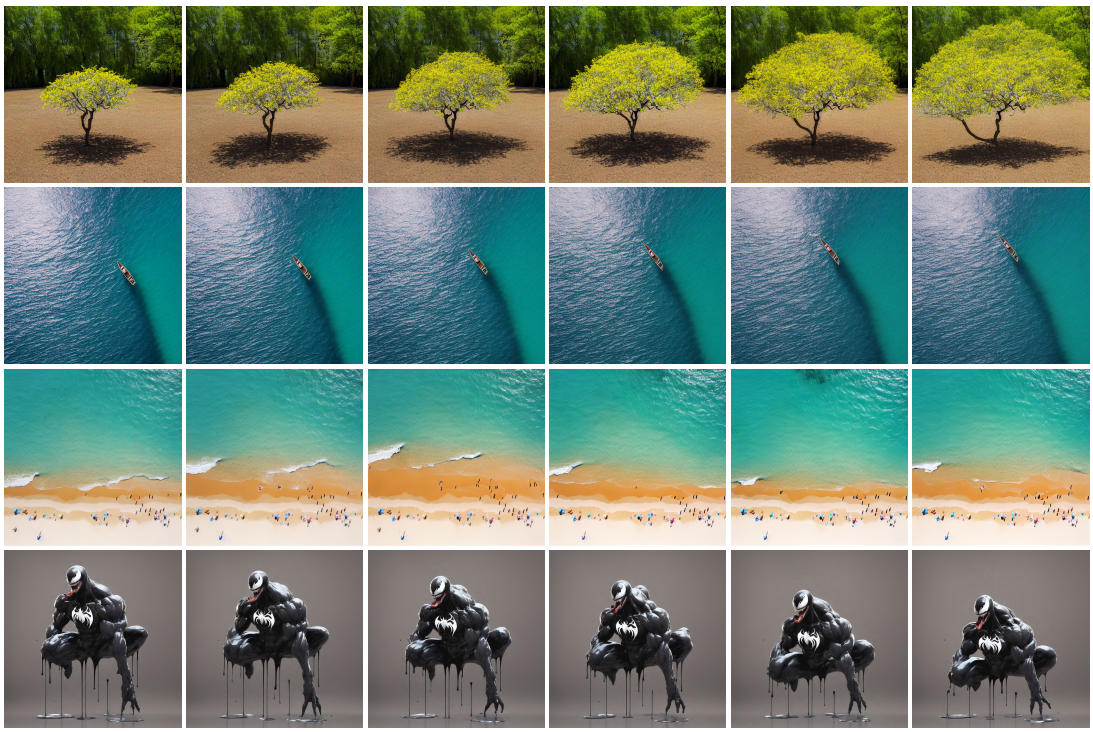


Figure 5.12: Additional video frames from MotionCraft. The last row is obtained by applying MotionCraft to SDXL [122]

5.7 Text Prompts

In this section we state the text prompts used in the generated videos for both MotionCraft and T2V0. Note that while MotionCraft is able to start from a real or generated image (with almost zero error for the real image reconstruction), T2V0 needs a hyper-parameters tuning due to a high guidance scale (not supporting direct inversion of real images).

Name	Text Prompt
Earth	"a close up of a picture of the earth from space."
City	"a satellite image of a city"
Dragons	"Two dragons fighting while breathing fires to each other. The flames are blazing and majestic light. Theatrical, character concept art by ruan jia, thomas kinkade, and trending on Artstation."
Statue	"transparent man made by water and smoke, in style of Yoji Shinkawa and Hyung-tae Kim, trending on ArtStation, dark fantasy, great composition, concept art, highly human made of water and foam, in the style of Pierre Koenig, red pigment, pastel paint, pink color scheme"
Drink	"wine falling on a empty glass"
Birds	"a small flock bird flying in the sky at the sunset"

For the text prompts of *Dragons* and *Statue* we leveraged MagicPrompt (for which we credit Gustavo Santana), a tool for rewriting simple text prompts to create more appealing starting images with Stable Diffusion.

For each example, the negative prompt $\mathcal{P}_\emptyset$ is equal to "poorly drawn, cartoon, 2d, disfigured, bad art, deformed, poorly drawn, extra limbs, close up, b&w, weird colors, blurry"

5.8 Conclusions & Future Work

In this chapter we presented MotionCraft, a zero-shot framework for video generation that combines a pretrained text-to-image diffusion model with optical flows derived from physics simulations. By warping the latent noise space according to simulated motion and reconstructing each frame through a modified denoising process, MotionCraft can generate realistic and temporally coherent videos without any additional training. This shows that it is possible to decouple appearance and motion, relying on the image prior of Stable Diffusion for content generation while using physically derived flows to impose controllable dynamics.

The method, however, also inherits limitations from the underlying image generator. In particular, the use of Stable Diffusion introduces imperfections such as inexact DDIM inversion, and in practice we observed a global color shift that tends to accumulate in later frames. The proposed MCFA strategy mitigates this effect only partially; a stronger solution could involve attending to a larger history of previously generated frames, at the cost of increased memory usage and run time.

A second limitation is that the method depends on optical flows obtained from physics simulators, which restricts it to motions that can be prescribed or simulated reliably. While this is effective for fluids, rigid bodies, or multi-agent systems, it is less suitable for complex motions such as human dancing or other difficult-to-model dynamics. Future work could address this by learning a generative model of optical flow conditioned on the initial frame and a prompt, while still constraining it with a physics simulator. An even more ambitious direction would be to close the loop between simulator and generator, allowing the generated frames to feed back into the physical process. More generally, extending this idea to explicit dynamic volumetric representations, such as latent 4D radiance fields or dynamic Gaussian models, could turn MotionCraft from a 2D video generator into a component of richer 3D world models capable of hallucinating missing views and time steps in a physically grounded way.

Chapter 6

Remote Sensing

The final challenge we consider takes us from laboratory-scale scenes to the scale of the Earth. Modern satellites continuously image our planet, producing an ever-growing archive of high-resolution optical data. These images are more than just pictures: they are raw material for digital surface models, land-cover maps, and change detection products that inform climate science, urban planning, disaster response, and many other societal applications. Turning stacks of satellite images into accurate 3D reconstructions is the domain of *remote-sensing photogrammetry*. Classical pipelines, based on stereo matching and multi-view geometry, have been refined over decades and can deliver high-quality digital elevation models, but they typically process each stereo pair in isolation, and they struggle to fully exploit large, multi-date image collections. Recent NeRF-based approaches bring the benefits of volumetric scene representations to Earth observation, achieving impressive reconstruction quality and shadow handling. However, they do so at a steep computational cost: training can take many hours or days per area of interest, which is difficult to reconcile with the scale and update frequency of modern satellite archives.

In a sense, we face an efficiency bottleneck similar to the one that motivated Gaussian Splatting in the context of novel view synthesis for everyday scenes. We would like to represent satellite scenes volumetrically, with accurate geometry and appearance, but we must also respect practical constraints: limited compute on operational systems, large numbers of scenes to process, and the need to re-run pipelines as new imagery arrives. Simply porting generic Gaussian Splatting to satellite data is not enough, though. The imaging geometry is more complex than a pinhole camera; shadows are cast by tall structures under varying sun angles; and the available views are sparse and irregular in time. Any method that aims to be both fast and accurate must therefore adapt to the specifics of remote sensing while preserving the efficiency that makes Gaussian Splatting attractive in the first place.

This chapter introduces *EOGS* (Earth-Observation Gaussian Splatting), a specialization of Gaussian Splatting tailored to satellite photogrammetry. The central idea is to retain the lightweight, explicit Gaussian primitive representation, but to embed it into a pipeline that respects the peculiarities of satellite imagery. We approximate the pushbroom sensor geometry with efficient affine camera models, incorporate physically motivated shadow rendering using the known sun direction, and regularize the Gaussian primitives so that the representation remains sparse, view-consistent, and predominantly opaque. These design choices allow EOGS to recover digital surface models and appearance at a level of detail comparable to NeRF-based Earth observation methods, while reducing optimization time from days to minutes per scene. In doing so, this chapter addresses the fourth and final challenge outlined in the introduction: making volumetric scene representations and their learning procedures efficient and scalable enough to keep up with real-world data streams at planetary scale.

6.1 Introduction

Photogrammetry from remote sensing images aims to recover the 3D geometry (*e.g.*, a Digital Surface Model, DSM) and appearance (*e.g.*, an albedo map) of the Earth's surface from satellite observations. In this chapter, we focus on digital surface modeling from multi-date images acquired from arbitrary satellite positions.

Historically, binocular stereovision and tri-stereo methods have been used for this purpose. However, these methods rely on image acquisitions being nearly simultaneous and with specific relative positions, which is often impractical, with limited acquisition opportunities, and/or costly.

More recently, multi-view methods developed for novel-view synthesis (NVS) have been applied to this task because they naturally handle diverse camera positions. In particular, NeRF-based approaches such as EO-NeRF [102] achieve state-of-the-art performance for digital surface modeling, notably thanks to improved shadow handling, but they remain computationally expensive. In recent years, 3D Gaussian Splatting (3DGS) [82] has emerged as a faster alternative to NeRF while maintaining comparable reconstruction accuracy.

In this work, we introduce EOGS, the Earth-observation Gaussian Splatting, the first method for digital elevation modeling based on 3DGS. EOGS achieves accuracy comparable to previous state-of-the-art approaches while being approximately 300× faster. Keys to the success of EOGS are the following contributions, all of which are compatible with the original 3DGS framework's efficiency:

- Approximating locally the pushbroom satellite sensors as affine cameras.
- Introducing a shadow-mapping-based pipeline for rendering the shadows in a physically accurate manner.

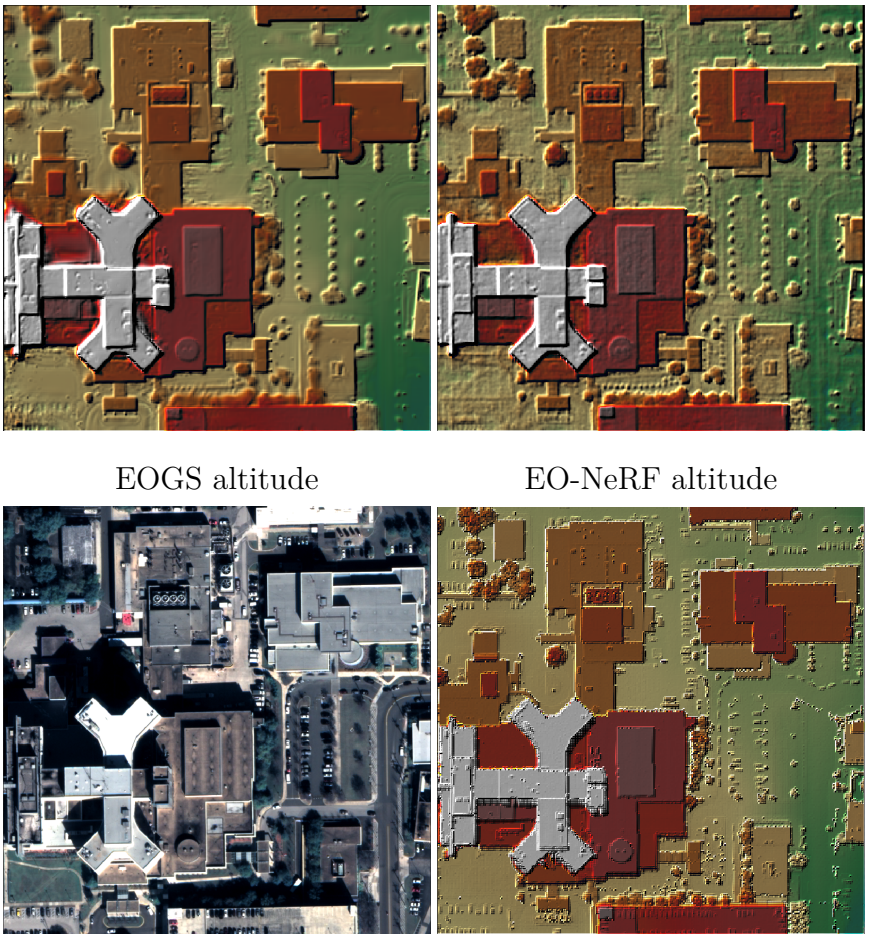


Figure 6.1: Using a limited number of satellite images of a given scene, the proposed EOGS method estimates the appearance and geometry of the scene. It achieves the same level of detail as EO-NeRF [102], such as the group of fans or the thin structures on top of the tall building on the left. However, EOGS requires only a few minutes of optimization, compared to the day-long training time required by EO-NeRF [102].

- Adding a new regularization term that promotes sparsity in the Gaussians opacities in order to reduce the training time.
- Adding a new regularization term that promotes view consistency by ensuring the rendered scene geometry remains consistent under small camera perturbations.
- Adding a new regularization term that promotes completely opaque objects by forcing them to cast non-translucent shadows.

6.2 Related Work

6.2.1 Stereovision for Earth Observation

Stereovision is at the heart of many tools for 3D estimation from series of satellite images. Examples of such pipelines are Ames stereo pipeline [12], MicMac [131], CARS [106], S2P [43], or CATENA [87].

Traditionally, these multi-view stereo methods are applied to well-chosen (either manually or automatically) image pairs. Since they process each pair independently (for example in the dense stereo matching step), a crucial step is the fusion of all the generated pairwise 3D models into a single one.

The recent trend has been to replace classic dense matching methods, such as semi-global matching (SGM) [60] or more global matching (MGM) [41], with deep learning based methods such as PSM [31], HSM [175] or GA-Net [180]. A review of these methods and a comparative study for satellite images is performed in [101].

6.2.2 NeRF for Earth Observation

As explained in Section 2.3, [109] have shown that it is possible to learn a volumetric model of a scene, called neural radiance fields (NeRF), using differentiable inverse rendering. Given a sparse set of views of the scene, NeRF learns in a self-supervised manner by maximizing the photoconsistency across the predicted renderings corresponding to the available viewpoints. After convergence, the volumetric model can then be used to render realistic novel views of the scene. In practice, this volumetric model is represented by an MLP that predicts, for each position $\mathbf{x}$ of the space, the local density of the scene $\rho(\mathbf{x})$ as well as its appearance (*i.e.*, color) $\kappa(\mathbf{x})$. The rendering is performed using an approximation of the volumetric integral Eq. (2.10) from optical physics estimated using ray casting.

NeRF-based methods were then extended to the remote sensing case, and in particular to perform multi-view and multi-date satellite photogrammetry, namely S-NeRF [37], Sat-NeRF [103], and EO-NeRF [102]. S-NeRF [37] exploits the solar direction, information typically available in the metadata of each observation or that can be easily retrieved knowing the location of the scene as well as the acquisition hour and date, to predict the direct sun light reaching each point in the scene. This is done by adding the solar direction as an input to the MLP and predicting the amount of sun light reaching a point as a new output. In this way, the shadows cast by buildings can be learned by the MLP and generated accordingly during the novel-view rendering step. Sat-NeRF [103] extends S-NeRF by modeling the transient parts of the scenes (*e.g.*, cars, construction sites, or foliage) as done in NeRF-in-the-wild [104] and improves the camera representation of S-NeRF, from pinhole to RPC [153, 6]. EO-NeRF [102] improves the handling of shadows of S-NeRF and Sat-NeRF by defining physically plausible shadows directly from

the geometry. These shadows are then rendered by additional raycasting from the surface in the direction of the sun. More recent work focuses on modeling difficult seasonal effects [46], extending the proposed volumetric models to surface models [125], using the raw pre-pansharpened data provided directly by the satellite operators [121], and accelerating the training step by taking advantage of faster NeRF versions [17].

6.3 Method

The proposed Earth-observation Gaussian splatting (EOGS) method specializes and adapts 3DGS, explained in detail in Section 2.4, for the satellite photogrammetry task. Given N non-orthorectified satellite images and their corresponding RPC camera model coefficients, a set of Gaussian primitives $\Gamma = \{\gamma_k\}_{k=1}^{K}$ is optimized to recover both the 3D geometry and appearance of the scene.

The general learning problem is to find the set of K Gaussian primitives that best approximates the N satellite images, with the rendering process of Eq. (2.13). This can be formulated as:

$$\underset{(\gamma_k)_{[\![1,K]\!]}}{\arg\min} \sum_{i=1}^{N} \mathcal{L}_{\text{GS}}(\hat{I}^{\mathcal{A}_i}, I^{\mathcal{A}_i}), \tag{6.1}$$

The predicted pixel color is obtained combining Eqs. (2.12) to (2.14), that we report here for clarity, assuming the background color $I_f = 0$:

$$\begin{cases} \hat{I}^{\mathcal{A}}(\underline{u}) = \sum_{k=1}^{K} \underline{\tilde{f}}_k \alpha_k \mathcal{G}_k^{\mathcal{A}}(\underline{u}) \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{A}}(\underline{u}) \right) \\ \mathcal{G}_k^{\mathcal{A}}(\underline{u}) = \exp\left(-\frac{1}{2} (\underline{u} - \mathcal{A}(\underline{\mu}_k))' \left(\underline{\underline{J}}_k^{\mathcal{A}} \underline{\underline{\Sigma}}_k \underline{\underline{J}}_k^{\mathcal{A}\prime} \right)^{-1} (\underline{u} - \mathcal{A}(\underline{\mu}_k)) \right) \end{cases}$$

where $\mathcal{A}$ is the camera model associated with the image containing pixel $\underline{u}$, $\underline{d}_{\underline{u}}$ is its view direction, $\underline{\underline{J}}_k^{\mathcal{A}}$ is the Jacobian of $\mathcal{A}$ computed at $\underline{\mu}_k$, and $\underline{\tilde{f}}_k$ represents the color of the k-th Gaussian primitive.

In the following sections we highlight the differences between EOGS and previous 3DGS and NeRF-based approaches.

6.3.1 Projections and Coordinate Systems

We define the coordinate system in which the Gaussian primitive centers and shapes are expressed as *world-space coordinates*. This coordinate system is a uniformly rescaled and recentered version of the *Universal Transverse Mercator (UTM) coordinate system* [156], such that the center of the scenes coincides with the origin, the scene is contained in a unit cube, and it is east-north-up aligned similarly to EO-NeRF [102]. At the other end of the transformation pipeline lies the *2D NDC-space*, where the Gaussian primitives are splatted.

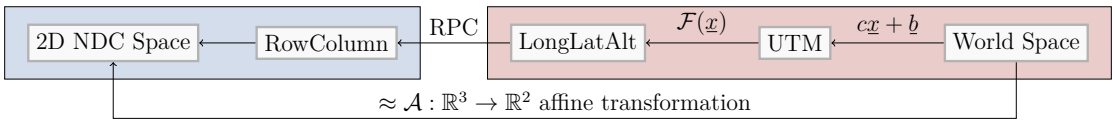


Figure 6.2: Summary of the transformation from world-space to NDC-space and its affine approximation. The affine approximation is computationally efficient, compatible with the Gaussian splatting formulation, and well-suited for satellite images. The coordinate systems in the right red box represent 3D world coordinates (camera-independent), while the left blue box shows 2D coordinates (camera-dependent).

The correct mapping between these two spaces is a composition of transformations: world-space to UTM to longitude-latitude-altitude. Using the RPC coefficients (that model the satellite position and 3D attitude) associated with each observation, the latter coordinate system is mapped onto the image row-column coordinates. Finally, the coordinates are normalized to range in $[-1,1]$ to get to NDC-space. As shown in Fig. 6.2, we instead compute a per-scene affine approximation of the whole transformation, introducing a negligible mean error of ≈ 0.012 pixels while being more computationally efficient than previous works and compatible with a Gaussian Splatting formulation. Specifically, for an affine camera model $\mathcal{A} : \underline{x} \in \mathbb{R}^3 \to \underline{\underline{A}}\underline{x} + \underline{a} \in \mathbb{R}^2$, the Eq. (2.15) simplifies to $\underline{\mu}_k^{\mathcal{A}} = \underline{\underline{A}}\underline{\mu}_k + \underline{a} \in \mathbb{R}^2$ and $\underline{\underline{\Sigma}}_k^{\mathcal{A}} = \underline{\underline{A}}\underline{\underline{\Sigma}}\underline{\underline{A}}' \in \mathbb{R}^{2,2}$. This simplification eliminates the need for the local first-order approximation used in the original 3DGS method, as we moved the approximation to the camera models.

6.3.2 Shadow Mapping

As in EO-NeRF, we want to explicitly model the shadow phenomena in the images, as the solar direction is available for each image in the scene. Unlike previous literature, our method uses a custom variant of shadow mapping to cast geometrically consistent shadows. Introduced in [171], Shadow Mapping is a well-known technique in the field of 3D graphics for adding shadows to a computer graphic rendering. It is particularly suited for EOGS, as it requires just the ability to render the scene from different points of view, as opposed to the shadow casting technique used in EO-NeRF that requires ray marching (which is not defined in Gaussian splatting).

Before introducing our variant of Shadow Mapping, we define the *elevation render*, *localization* function, and the *homologous point* function.

Given a camera model/projection $\mathcal{A}$, the elevation render is defined as the 3DGS

rendering Eq. (2.13) using the real elevations instead of colors

$$E^{\mathcal{A}}(\underline{u}) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{C}}(\underline{u}) \right) \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \mathcal{E}(\underline{\mu}_k) + E_f \prod_{k=1}^{K} 1 - \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \tag{6.2}$$

where $E_f << 0$ is a fixed constant representing a very low elevation, $\mathcal{E} : \mathbb{R}^3 \to \mathbb{R}$ is an affine operation mapping 3D points expressed in the "native" world coordinates to the corresponding real altitude, expressed in meters. We remark that this is not the depth nor the inverse depth, typically found in the MVS literature. Using the real altitude is advantageous here for two practical reasons. First, altitude is expressed in a global physical reference frame shared by all satellite views and by the sun camera, so elevations can be compared directly across cameras without depending on the viewing direction. Second, the target product in remote sensing is a DSM in metric geodetic coordinates, making altitude a more natural quantity than view-dependent depth.

Given a camera model/projection $\mathcal{A}$, the localization function that maps a pixel of the camera and a given absolute altitude to its associated point in the native 3D world is defined as:

$$\begin{aligned} &\mathrm{loc}^{\mathcal{A}} : \left(\mathbb{R}^2 \times \mathbb{R}\right) \to \mathbb{R}^3, \quad (\underline{u}, h) \mapsto \underline{x} \\ &\text{s.t.} \quad \mathcal{A}(\underline{x}) = \underline{u} \quad \text{and} \quad \mathcal{E}(\underline{x}) = h. \end{aligned} \tag{6.3}$$

Given two cameras $\mathcal{A}$ and $\mathcal{B}$, the homologous point function maps a pixel of the first camera to the corresponding pixel of the second camera, taking into consideration the 3D geometry:

$$\begin{aligned} &\mathrm{hom}^{\mathcal{A},\mathcal{B}} : \mathbb{R}^2 \to \mathbb{R}^2, \quad \underline{u} \mapsto \tilde{\underline{u}} \\ &\text{s.t.} \quad \tilde{\underline{u}} = \mathcal{B}\left(\mathrm{loc}^{\mathcal{A}}\left(\underline{u}, E^{\mathcal{A}}(\underline{u})\right)\right). \end{aligned} \tag{6.4}$$

In our shadow mapping approach (depicted in Fig. 6.3), we assume that the sun is the only *directional* light source present in the scene. This assumption is appropriate for the daytime optical satellite imagery considered in this chapter. Non-directional illumination is partially absorbed by the per-camera ambient light term $\underline{\psi}^{\mathcal{A}}$, while scenes dominated by artificial light sources (*e.g.*, nighttime city imagery) fall outside the scope of the proposed model. Moreover, since it is far from the scene, it can be approximated as a directional light. Following the classic shadow mapping approach, we construct a camera $\mathcal{S}$, called *sun camera*, placed at and aligned with the light source. As the camera model corresponding to a directional light is the affine camera, we can handle uniformly the sun cameras and the satellite cameras.

Then, we consider a second camera, $\mathcal{A}$, from which we want to synthesize a novel view and apply shadows according to the sun direction. Given a point $\underline{u}$ in the $\mathcal{A}$

NDC-space, and its corresponding altitude $E^{\mathcal{A}}(\underline{u})$, we first localize it, obtaining a 3D point in world-space. We then project this point according to $\mathcal{S}$, obtaining the homologous point of $\underline{u}$ in $\mathcal{S}$. We then resample the elevation rendering of $\mathcal{S}$ at this projected point and compare it with $E^{\mathcal{A}}(\underline{u})$. Mathematically, this corresponds to

$$\Delta h^{\mathcal{A},\mathcal{S}}(\underline{u}) = E^{\mathcal{S}}\left(\hom^{\mathcal{A},\mathcal{S}}(\underline{u})\right) - E^{\mathcal{A}}(\underline{u}). \tag{6.5}$$

If these two elevations, $E^{\mathcal{A}}(\underline{u})$ and $E^{\mathcal{S}}(\tilde{\underline{u}})$ are the same, it means that both the camera and the sun camera are imaging the same 3D point, hence this point is in light. If the two elevations are not the same, then the sun camera is not able to "see" the 3D point, hence it is in shadows. To represent this shading, the color of points in the shadows is multiplied by a darkening coefficient computed from $\Delta h^{\mathcal{A},\mathcal{S}}(\underline{u})$ as

$$s^{\mathcal{A},\mathcal{S}}(\underline{u}) = \min\left\{\exp\left\{-\rho\Delta h^{\mathcal{A},\mathcal{S}}(\underline{u})\right\},1\right\}. \tag{6.6}$$

We argue that this formulation is physically plausible as this would be the correct equation for a homogeneous medium of density ρ, as shown in [105].

Following [102], we also model a per-camera ambient light $\underline{\psi}^{\mathcal{A}}$ so that in-shadow objects do not appear completely black. The shading to be applied to a given pixel $\underline{u}$ is given by the following *lighting coefficient*

$$l^{\mathcal{A},\mathcal{S}}(\underline{u}) = s^{\mathcal{A},\mathcal{S}}(\underline{u}) + (1 - s^{\mathcal{A},\mathcal{S}}(\underline{u}))\underline{\psi}^{\mathcal{A}}. \tag{6.7}$$

Finally, EOGS image formation equation is:

$$I^{\mathcal{A},\mathcal{S}}(\underline{u}) = l^{\mathcal{A},\mathcal{S}}(\underline{u}) \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{C}}(\underline{u})\right) \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \underline{\phi}^{\mathcal{A}}(\underline{f}_k) \tag{6.8}$$

where $\underline{\phi}^{\mathcal{A}}(\cdot)$ is a camera-specific affine color correction applied to the intrinsic primitive colors $\underline{f}_k$. We remark that, differently from 3DGS, we drop the view-direction dependencies of the primitive colors and introduce a camera-dependent color correction.

It is useful to define the *albedo rendering*, where we do not use the shadows or the camera-specific color correction:

$$I^{\mathcal{A}}(\underline{u}) = \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} 1 - \alpha_j \mathcal{G}_j^{\mathcal{C}}(\underline{u})\right) \alpha_k \mathcal{G}_k^{\mathcal{C}}(\underline{u}) \underline{f}_k \tag{6.9}$$

While the image formation model defined in Eq. (6.8) is equivalent to EO-NeRF, the shadow definition is quite different. In EO-NeRF case, shadows are defined as the sun visibility for all points on the surface. Because of possible occlusions, two points of the scene can correspond to the same point seen from the sun direction. Therefore, it is not possible to define the sun visibility as an "image" that could be

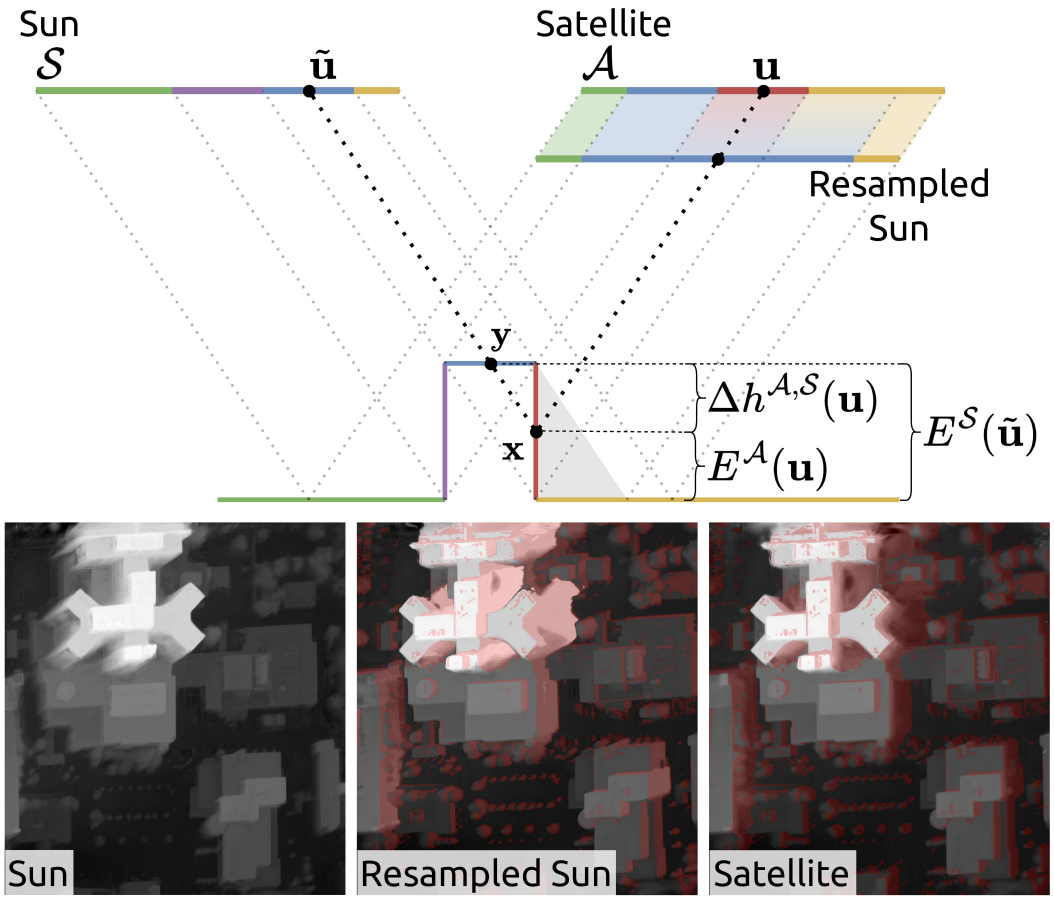


Figure 6.3: Shadow mapping illustration. The point $\underline{u}$ in the satellite image (affine camera $\mathcal{A}$) corresponds to the 3D point $\underline{x} = \mathrm{loc}^{\mathcal{A}}(\underline{u})$ on the vertical wall. Projecting $\underline{x}$ to the sun camera (affine camera $\mathcal{S}$), $\underline{\tilde{u}} = \mathcal{S}\underline{x}$ is obtained. Then $\underline{y} = \mathrm{loc}^{\mathcal{S}}(\underline{\tilde{u}})$ is obtained localizing $\underline{\tilde{u}}$. The point $\underline{x}$ and its pixel $\underline{u}$ are in shadow because the elevation of $\underline{y}$ is greater than the elevation of $\underline{x}$. Indeed, all and only the points where the satellite elevation and the resampled sun elevation do not match should be shaded. On the bottom of the illustration are shown examples of the sun elevation, the resampled sun elevation, and the satellite elevation renderings, with shadows highlighted in red.

estimated using a Gaussian splatting-like process. Trying to compute an irregularly sampled “image" corresponding to these points would break the locality assumption used in Gaussian splatting during the rasterization step and thus reduce the computational efficiency. On the contrary, the proposed shadow mapping verifies all the assumptions made by Gaussian splatting.

6.3.3 Regularizers

It is well-known that deep neural networks are implicitly regularized [126, 123, 151, 157], meaning that despite being used in the overparametrized regime, they show generalization capabilities.

On the other hand, we found out that primitives in 3DGS-based methods are almost independently optimized one from the other. This is probably due to the fact that the primitives in 3DGS are initialized as small spheres, spread out in the entire scene. This results in 3DGS being less regularized than NeRF-based methods and "lacking" constraints during the optimization phase.

Hence we are free to add additional regularization constraints to the general optimization problem Eq. (6.1) that induce smoother and more regular solutions. In particular, we introduce constraints that promote our solution to be sparse (*i.e.*, we encourage solutions that require fewer Gaussian primitives), view consistent, and mostly composed of completely opaque objects.

As common ML pipelines are specialized for unconstrained optimization problems, we argue to use a Lagrangian relaxation approach and re-formulate each constraint as a new loss term, each with its own experimentally-found Lagrangian multiplier.

Promoting Sparsity. Training time is directly proportional to the number of Gaussian primitives considered during the optimization process. As we want to recover the geometry of the scene as fast as possible, we want as few Gaussian primitives as possible, hence a sparse solution.

Inspired by the well-known LASSO regularization in linear regression [155] that promotes a sparse solution, we consider a L^1 regularization of the opacities

$$\mathcal{L}_o = \frac{1}{K} \sum_{k=1}^{K} \alpha_k. \tag{6.10}$$

This regularization promotes sparsity in the primitive opacities distribution, hence only "useful" primitives will be visible at the end of the optimization. We pair this regularization with a simple thresholding pruning procedure that discards any primitive with $\alpha < \alpha_{\min}$. In this way, unused primitives are actually discarded, yielding faster splatting and overall faster training (specifically, we recorded speedups of up to 2× on the considered datasets).

We remark that many works [24, 84] have proposed replacements to the original 3DGS densification/pruning procedure. Here, instead, we aim only at lowering the number of primitives, so we do not need a densification strategy as long as we instantiate enough of them at the beginning of the optimization. Moreover, we set $\alpha_{\min} = 0.0025$ as primitives with lower opacities are already discarded in the original 3DGS implementation of the front-to-back splatting procedure.

Promoting View Consistency. Differently from the classical NVS context, in remote sensing the available views are low-count and sparse, resulting in Eq. (6.1)

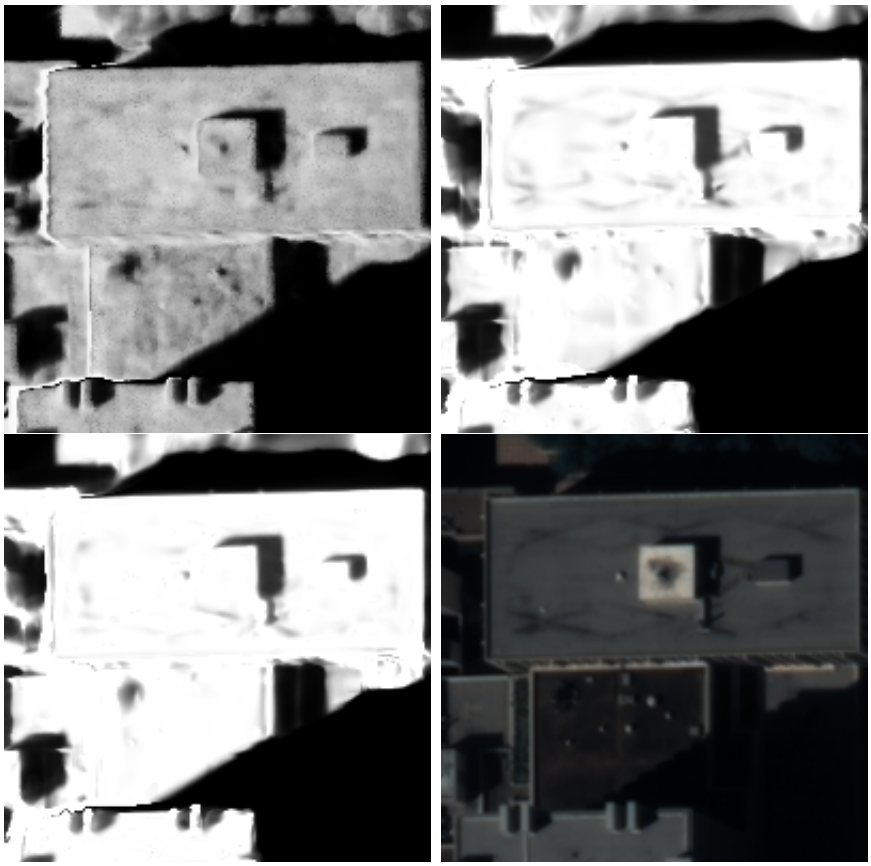


Figure 6.4: From top-left to bottom-right, shadow maps of EO-NeRF, EOGS without the $\mathcal{L}_s$ penalizer, EOGS with the $\mathcal{L}_s$ penalizer, and the corresponding satellite image. Textures corresponding to the image content can be observed in the shadow map of EO-NeRF and EOGS without the $\mathcal{L}_s$ penalizer, but not in EOGS.

being even less constrained. Paired with the fact that 3DGS does not benefit from the implicit regularization of NeRF, we argue that an additional constraint promoting view consistency is needed.

We propose a "local view consistency" loss based on the intuition that if the same 3D point is visible from two cameras and the cameras are close to each other, then the color and elevation resampled at the corresponding pixels should be the same. Otherwise the object is occluded or outside the camera boundaries.

Mathematically, given a camera $\mathcal{A}$ we randomly perturb it and obtain a camera $\mathcal{B}$. Assuming that there is no view-direction dependent color effect, this constraint reads:

$$\Delta h^{\mathcal{A},\mathcal{B}}(\underline{u}) < \Delta h_{\min} \Rightarrow \begin{cases} I^{\mathcal{A}}(\underline{u}) = I^{\mathcal{B}}\left(\hom^{\mathcal{A},\mathcal{B}}(\underline{u})\right) \\ E^{\mathcal{A}}(\underline{u}) = E^{\mathcal{B}}\left(\hom^{\mathcal{A},\mathcal{B}}(\underline{u})\right), \end{cases} \tag{6.11}$$

where we reused the same notation of the shadow mapping explanation.

This constraint results in two loss terms, the color (albedo) consistency and the altitude consistency:

$$\mathcal{L}_{cc} = \sum_{\underline{u}} M^{\mathcal{A},\mathcal{B}}(\underline{u}) \left| I^{\mathcal{A}}(\underline{u}) - I^{\mathcal{B}}\left(\hom^{\mathcal{A},\mathcal{B}}(\underline{u})\right)\right| \tag{6.12}$$

$$\mathcal{L}_{ac} = \sum_{\underline{u}} M^{\mathcal{A},\mathcal{B}}(\underline{u}) \left| E^{\mathcal{A}}(\underline{u}) - E^{\mathcal{B}}\left(\hom^{\mathcal{A},\mathcal{B}}(\underline{u})\right)\right|, \tag{6.13}$$

where $M^{\mathcal{A},\mathcal{B}}(\cdot)$ is a binary mask that selects all pixels $\underline{u}$ such that $\Delta h^{\mathcal{A},\mathcal{B}}(\underline{u}) < \Delta h_{\min}$ and $\hom^{\mathcal{A},\mathcal{B}}(\underline{u})$ is inside the image boundaries. We remark that we always choose $\mathcal{A}$ from the input posed images and we set $\Delta h_{\min} = 30$cm. Moreover, we obtain $\mathcal{B}$ by independently sampling $q_1, q_2 \in \mathbb{R}$ from a ± 1-truncated standard distribution and defining

$$\mathcal{B}(\underline{x}) = \mathcal{A}(\underline{x}) + 0.05\ \mathcal{E}(\underline{x}) \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}. \tag{6.14}$$

In the generic NVS literature, many works [116, 36] proposed different methods for increasing the view consistency. RegNeRF [116] is the first work that deals with sparse camera poses by introducing a loss term that maximizes the likelihood of rendered RGB patches from virtual cameras with a pre-trained deep normalizing-flow model, while also adding a total variation regularization on the rendered depth. Furthermore, [36] introduces a reprojection mechanism such that only the geometry needs to be learned from the NeRF, as the colors are resampled from the input images. Note that EOGS differs from [116] as we ask for consistency (RGB and depth) between two cameras (one real and one virtual), hence we do not need any pre-trained model for the RGB renders nor prior on the elevation renders. EOGS also differs from [36] as we learn the colors and do not resample input images that may contain transients or color shifts.

Promoting Opacity. Looking at the output from SatNeRF and EO-NeRF (see Fig. 6.4), we can see that much of the texture of the scene is embedded in the geometric shadows. Additional examples of the illumination decomposition into generated image, albedo, and shadow maps are shown later in Figs. 6.11 and 6.12, which help visualize the effect of the proposed lighting model. This geometry misuse is caused by semi-transparent objects casting semi-transparent shadows. In order to lessen this effect, we propose to add an entropy-based penalty $\mathcal{L}_s$ for incorrect use of the shadows. This penalty is defined as

$$\mathcal{L}_s = \sum_{\underline{u}} H\left(s^{\mathcal{A},\mathcal{S}}(\underline{u})\right), \tag{6.15}$$

where $H(x) = -\left(x \log_2(x) + (1-x)\log_2(1-x)\right)$. Indeed the shadow map $s^{\mathcal{A},\mathcal{S}}$ should contain only 0 or 1 values. This is the case for $\rho \to +\infty$ in Eq. (6.6), as a

building should not cast a semi-transparent shadow. Hence we add this entropy-based penalizer to discourage the use of semi-transparent shadows, which in turn encourage objects to be either completely transparent or fully opaque. Note that choosing a large ρ during training is not an option since it would make the training unstable as Eq. (6.6) would be close to a non-differentiable step function.

6.3.4 Implementation Details

The implementation of EOGS is based on the original 3DGS code base. Other than the aforementioned novel contributions, the main differences lie in disabling the per-Gaussian view-direction color dependency and initializing all the Gaussians with white color and as low as possible opacity (1%). Moreover, we reduce the number of iterations to 5000 and enable the shadow mapping and all three regularizations at iteration 1000. Furthermore, the Gaussians centers are initialized uniformly in the 3D scene such that the initial density is 0.13 Gaussians per m^3.

We use the same optimizer and scheduler of 3DGS for the primitives and use a second Adam scheduler with 10^{-2} learning rate for learning the camera-dependent parameters: the affine color-correction $\underline{\phi}^{\mathcal{A}}$ and the ambient color $\underline{\psi}^{\mathcal{A}}$.

The Lagrangian coefficients of the regularization constraints have been found experimentally on a single scene, rounded to the nearest power of ten, and applied to all scenes. This highlights the robustness of EOGS to the specific values of these coefficients. The final loss is:

$$\min \sum_{i=1}^{N} \mathcal{L}_{\text{GS}}(\hat{I}_i, I_i) + 0.1\mathcal{L}_o + 0.1\mathcal{L}_{cc} + 0.01\mathcal{L}_{ac} + 0.01\mathcal{L}_s, \tag{6.16}$$

where $\hat{I}_i$ is now, differently from 3DGS in Eq. (6.1), a shorthand notation for $I^{\mathcal{A}_i,\mathcal{S}_i}$ from Eq. (6.8), which also depends on the sun camera $\mathcal{S}_i$.

	mask	method	004	068	214	260	Mean ↓	Time ↓
JAX dataset	✗	**EOGS**	1.45	1.10	1.73	1.55	1.46	**3 minutes**
		EO-NeRF [102]	**1.37**	**1.05**	**1.61**	1.37	**1.35**	15 hours
		Sat-Mesh [125]	1.55	1.15	2.02	**1.36**	1.52	8 minutes
		SAT-NGP [17]	1.63	1.27	2.18	1.79	1.72	25 minutes
	✓	**EOGS**	**0.89**	**1.01**	1.63	**1.24**	**1.19**	**3 minutes**
		EO-NeRF [102]	1.02	1.03	**1.55**	**1.24**	1.21	15 hours
		SAT-NGP [17]	1.03	1.26	2.17	1.43	1.47	25 minutes

	mask	method	001	002	003	Mean ↓	Time ↓
IARPA dataset	✗	**EOGS**	1.58	2.00	**1.27**	1.62	**3 minutes**
		EO-NeRF [102]	**1.43**	**1.79**	1.31	**1.51**	15 hours
		Sat-Mesh [125]	N.A.	N.A.	N.A.	N.A.	8 minutes
		SAT-NGP [17]	1.54	2.11	1.69	1.78	25 minutes
	✓	**EOGS**	1.38	1.70	**1.03**	**1.37**	**3 minutes**
		EO-NeRF [102]	**1.32**	**1.63**	1.18	1.38	15 hours
		SAT-NGP [17]	1.34	1.85	1.62	1.60	25 minutes

Table 6.1: Mean absolute error on the elevation [meters] and the corresponding training time for various baseline methods, when considering the whole AOI (no mask) or when ignoring foliage areas (foliage mask). Results for Sat-Mesh are reported from the paper since the authors did not share their code.

6.4 Experiments

We evaluate EOGS in the same experimental setting as the most recent related work in the literature, EO-NeRF.

We are using datasets provided in the 2019 IEEE GRSS Data Fusion Contest (DFC2019) [21, 90] and 2016 IARPA Multi-View Stereo 3D Mapping Challenge (IARPA2016). These datasets, comprising a total of 7 areas of interest (AOI), contain cropped non-orthorectified multidate WorldView-3 observations, along with metadata such as the 3D satellite attitude (encoded in the RPC coefficient) and the local sun direction. We use the bundled-adjusted version of the RPC coefficient used in EO-NeRF. Each image covers approximately 256×256 meters squared of terrain with a resolution of $30 \sim 50$ cm per pixel, while each AOI is imagined by $10 \sim 20$ crops.

6.4.1 Main Experiment Results

Table 6.1 show the main experimental results of EOGS. To assess the accuracy of EOGS we report the mean absolute error (MAE) between a lidar scan included

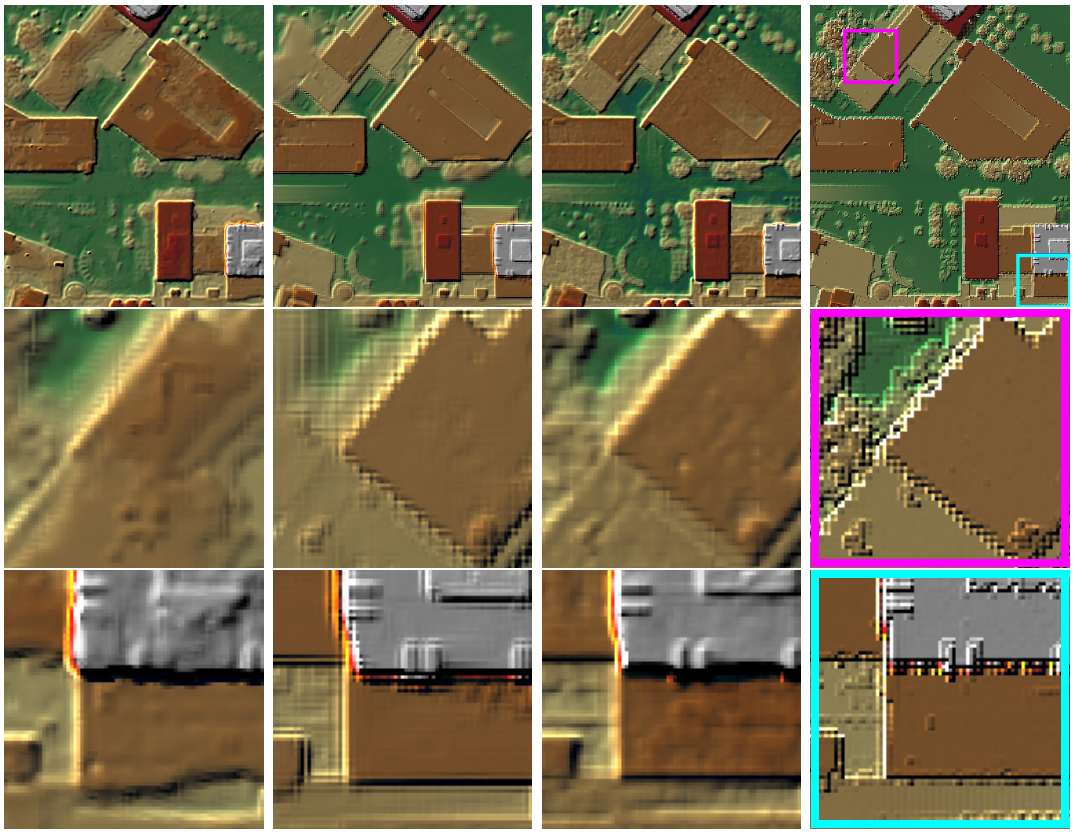


Figure 6.5: From left to right: visual results on JAX_214 comparing SAT-NGP [17], EOGS, EO-NeRF [102] and the ground truth.

in the dataset and the elevation render aligned to this nadir view. We argue that the volume of these data will grow in the near future, so we are also interested in the time required to recover the geometry from the input images. Hence we also report the training time.

If the entire AOIs are considered, as reported in Table 6.1 (top), EOGS performs slightly worse than the state of the art EO-NeRF but it is approximately 300× faster. For reference, we also report all available results of other methods from the literature (EO-NeRF [102], SAT-NGP [17], and Sat-Mesh [125]). We see that EOGS is pareto-optimal with respect to elevation MAE and training time. If instead we use available ground truth semantic maps to ignore prediction in the foliage areas, EOGS performance is equivalent to EO-NeRF, showing higher accuracy for structural objects, as reported in Table 6.1 (bottom). We present visual results in Fig. 6.5 as well as in the supplementary material.

Shadowmap	✗	✓	✓	✓	✓	✓	✓	✓	✓
Sparsity	✗	✗	✓	✗	✗	✗	✓	✓	✓
Consistency	✗	✗	✗	✓	✗	✓	✗	✓	✓
Opacity	✗	✗	✗	✗	✓	✓	✓	✗	✓
MAE [m] ↓	5.03	1.86	1.83	1.69	1.79	1.57	1.76	1.64	1.54

Table 6.2: Ablation study of each proposed component of EOGS.

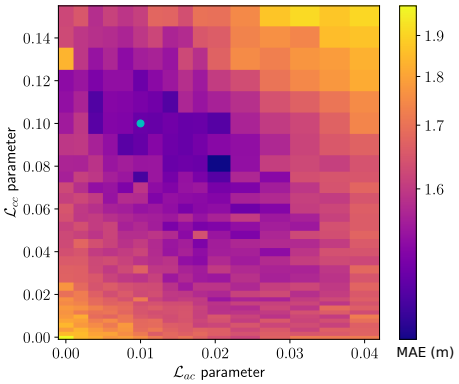
6.4.2 Ablation and Parameter Studies

Impact of the Different Losses. Table 6.2 reports an ablation study of the loss terms in EOGS. Each column corresponds to a different ablation experiment, while each row corresponds to a different component of EOGS being ablated. The first row indicates whether the Shadow Mapping technique is enabled or not. The following three rows indicate, respectively, the presence of the sparsity, consistency, and opacity regularizers. We remark that the first column is equivalent to 3DGS with affine cameras, learnable per-camera affine color correction, and different primitives initialization. For each column, we report the grand mean elevation MAE of JAX and IARPA scenes.

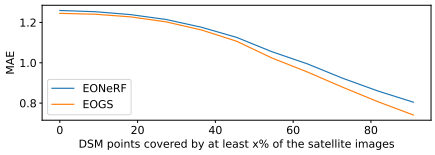
To quantitatively measure the impact of each single component, we linearly regress the MAE from the presence of the components, obtaining a coefficient for each component that expresses an elevation MAE gain with respect to the base case (reported in the first column of Table 6.2). The introduction of shadow mapping is the most impactful component, gaining 3.16 meters of accuracy. Then, the consistency regularizer and the opacity regularizer further improve the accuracy of EOGS by 0.20 and 0.09 meters, respectively. Lastly, the sparsity regularizer, while being necessary for achieving efficient training, also reports an improvement of 0.04 meters. Hence, all components independently contribute to the quality of the geometry reconstruction.

Regularization Parameters. Figure 6.6a shows the results of a grid search on the coefficients of $\mathcal{L}_{cc}$ and $\mathcal{L}_{ac}$ in Eq. (6.16). It shows that both the altitude regularization and the color regularization are necessary to achieve the best performance. We remark that, in order to reduce overfitting to a particular scene, we choose the same "round coefficients" for all scenes.

Impact of visibility. Figure 6.6b shows the impact of the visibility (*i.e.*, the number of cameras that can see a given point of the scene) on the performance. While EOGS and EO-NeRF are comparable on average, this test shows that EOGS performs better for regions that are visible in most images but struggles in the regions observed in only a few images.



(a) View consistency regularization parameter ablation study. Selected parameter set is shown with the cyan dot. Estimation performed on the JAX_260 sequence.



(b) Impact of the visibility on the performance (using foliage mask).

6.5 Additional Visual Results

We present additional visual results in Figs. 6.7 to 6.10.

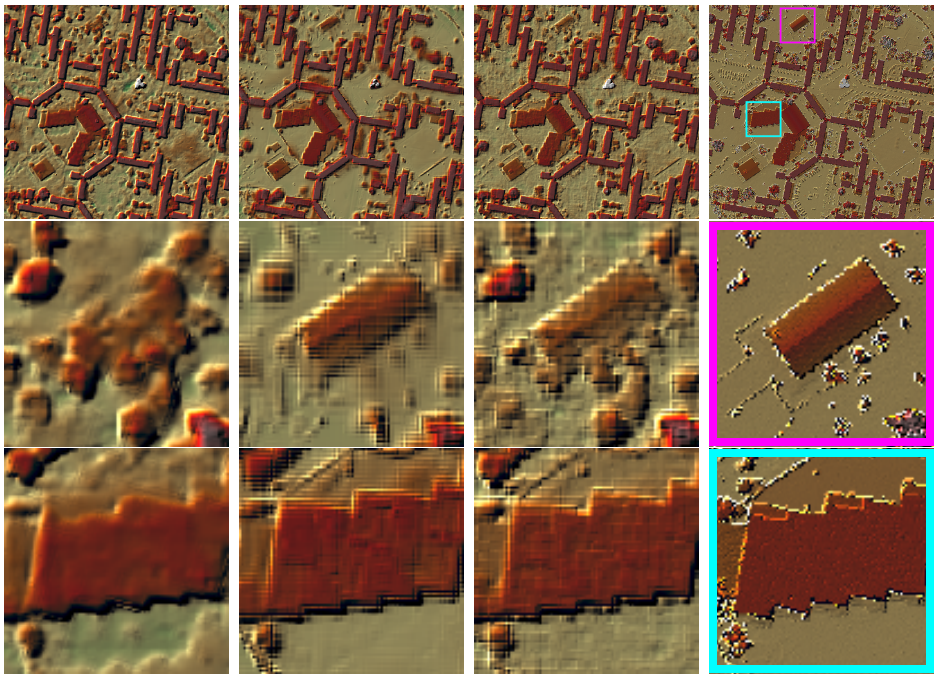


Figure 6.7: From left to right: visual results on IARPA_001 comparing SAT-NGP [17], EOGS, EO-NeRF [102] and the ground truth.

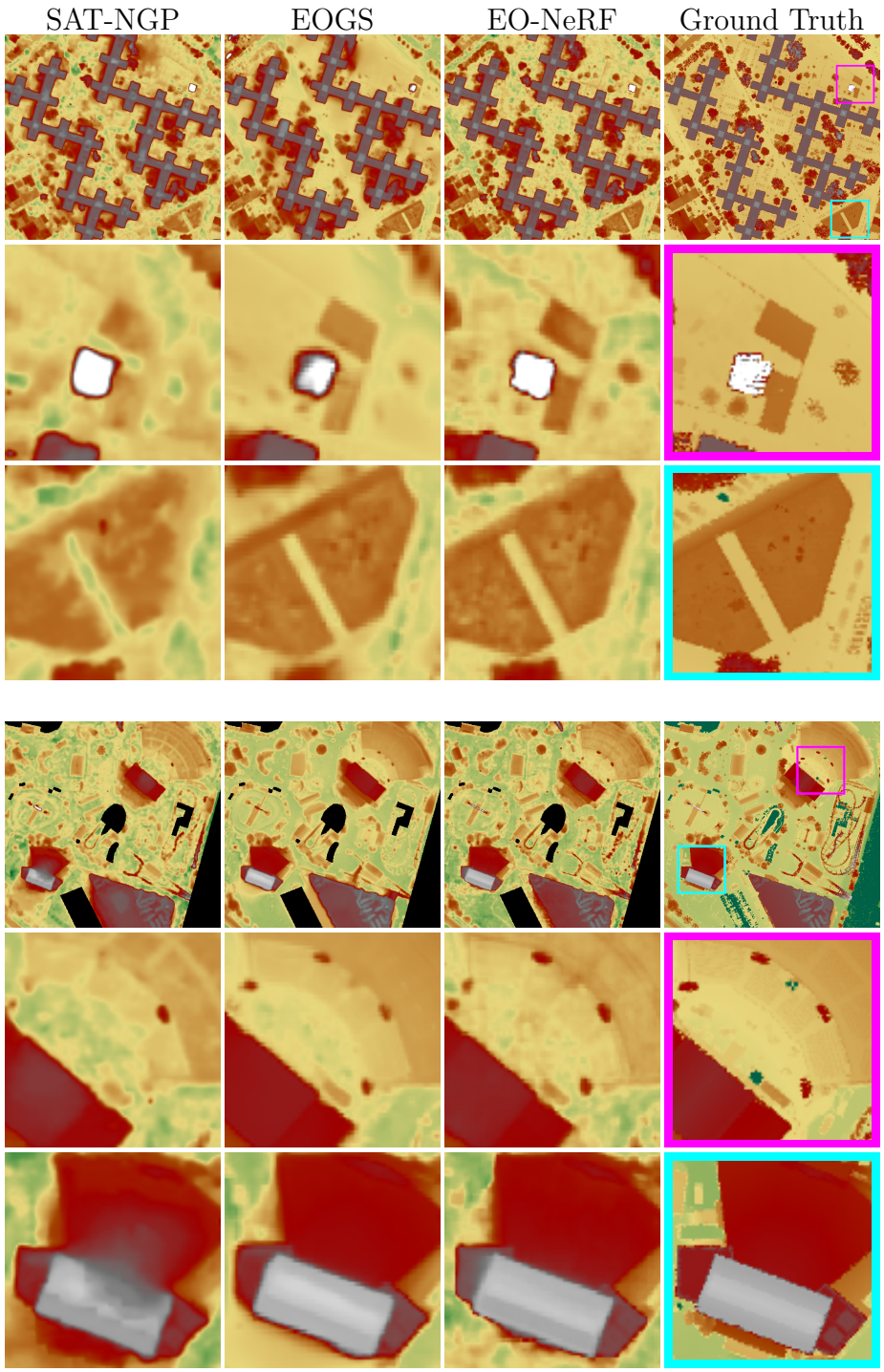


Figure 6.8: Visual results on IARPA_002 and IARPA_003.

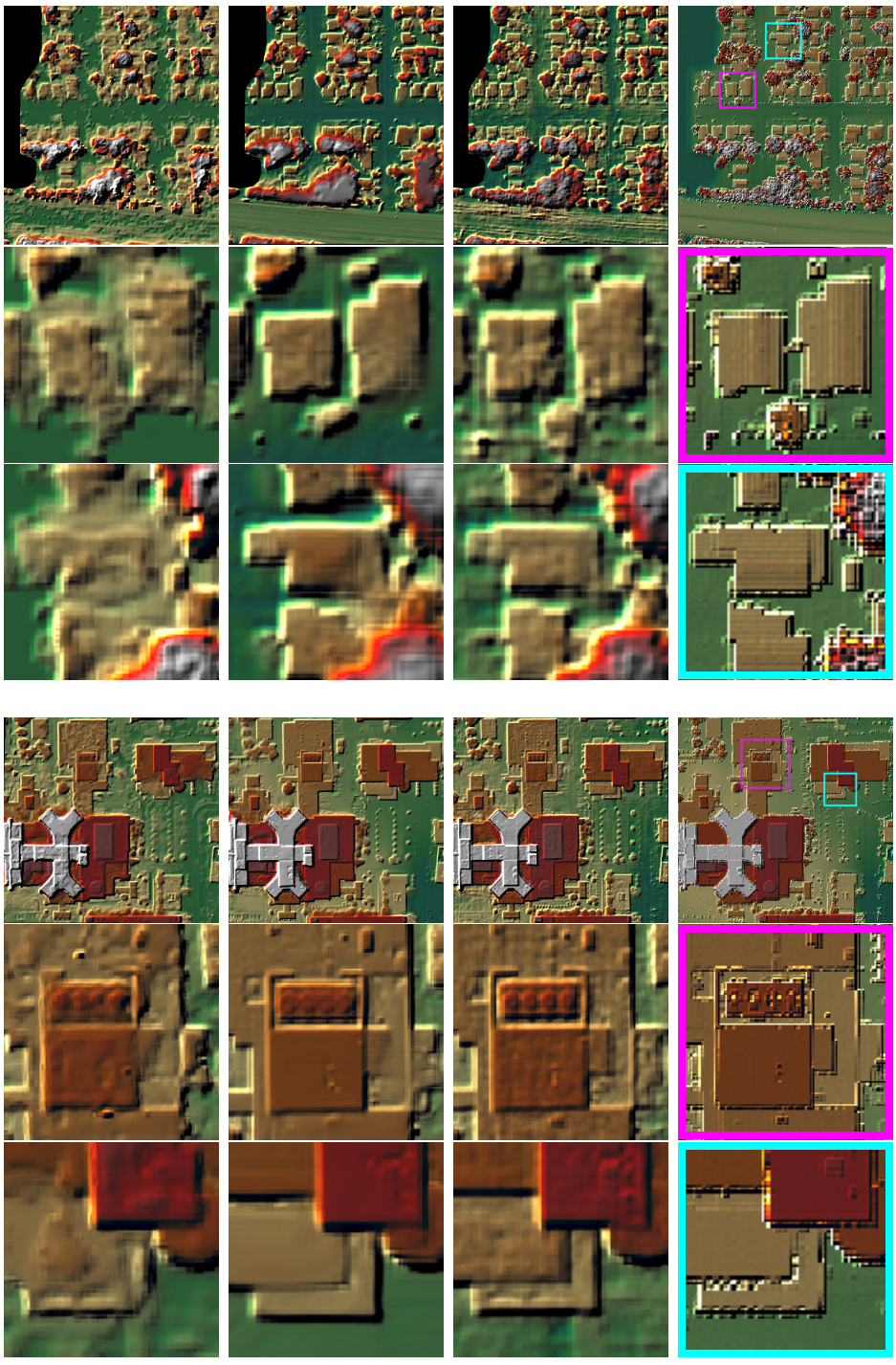


Figure 6.9: From left to right: visual results on JAX_004 and JAX_068 comparing SAT-NGP [17], EOGS, EO-NeRF [102] and the ground truth.

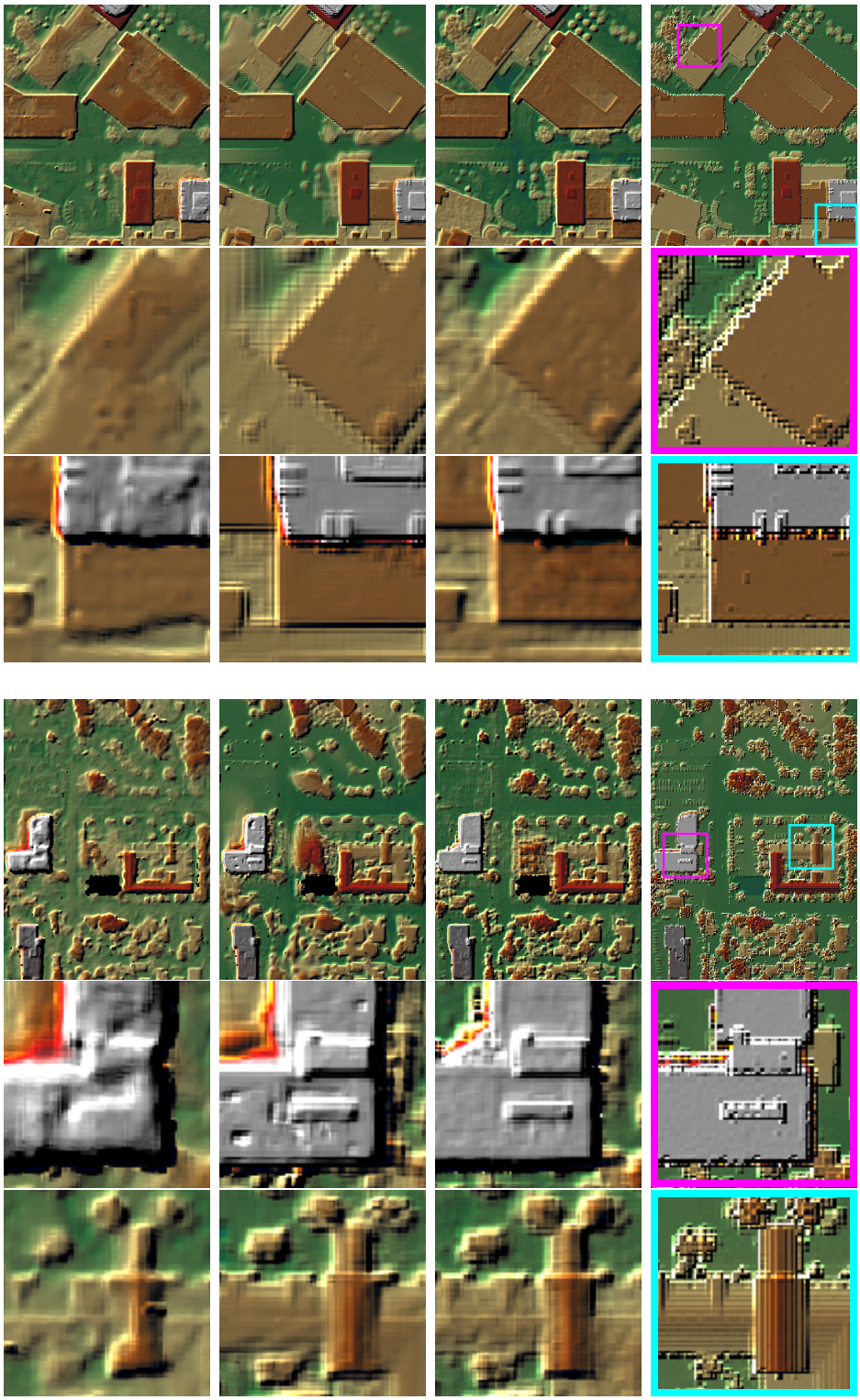


Figure 6.10: From left to right: visual results on JAX_214 and JAX_260 comparing SAT-NGP [17], EOGS, EO-NeRF [102] and the ground truth.

6.6 Albedo and Shadow Visualization

We present in Figs. 6.11 and 6.12 examples of albedos and shadows generated by EO-NeRF and the proposed EOGS for multiple scenes.

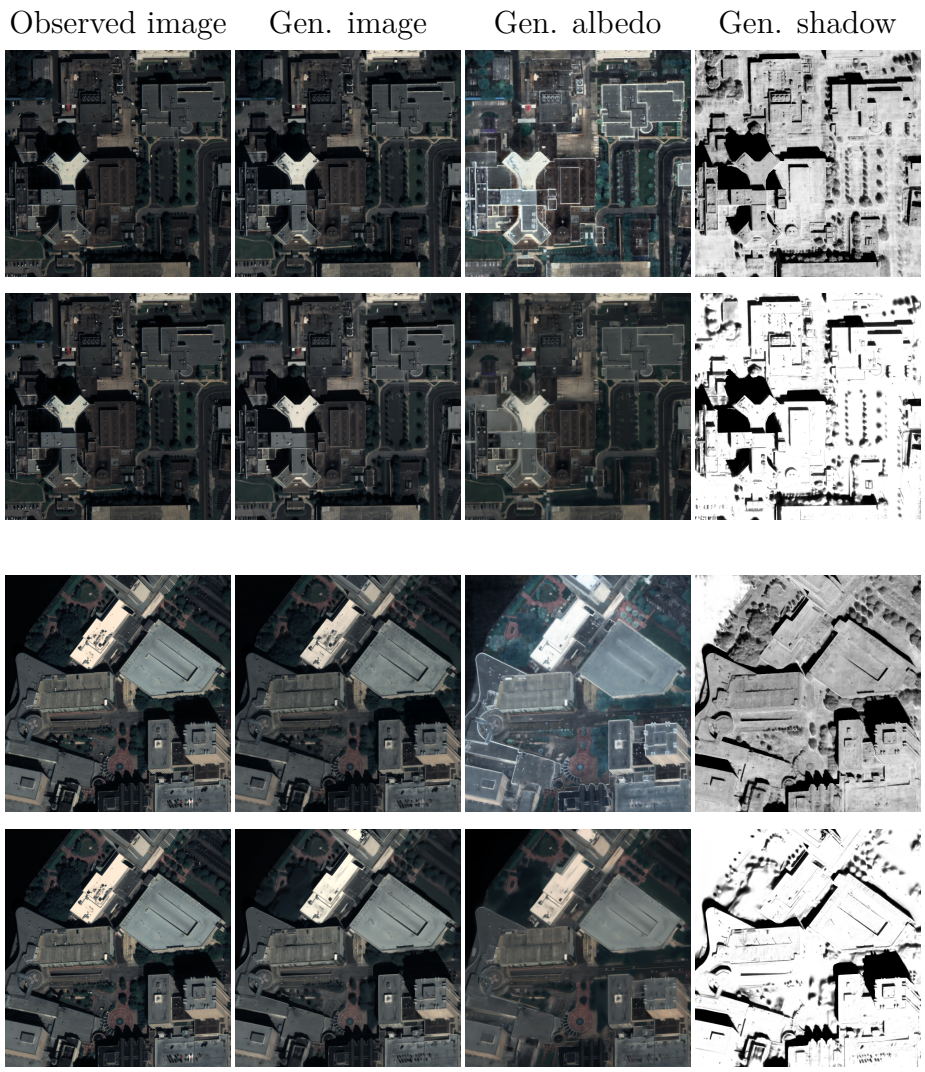


Figure 6.11: Visual comparison of the scene albedo and shadows generated by EO-NeRF (top) and EOGS (bottom) for JAX_068 and JAX_214. Note that the scaling for all images is the same except for the albedo that is rescaled independently to show the entire dynamic.

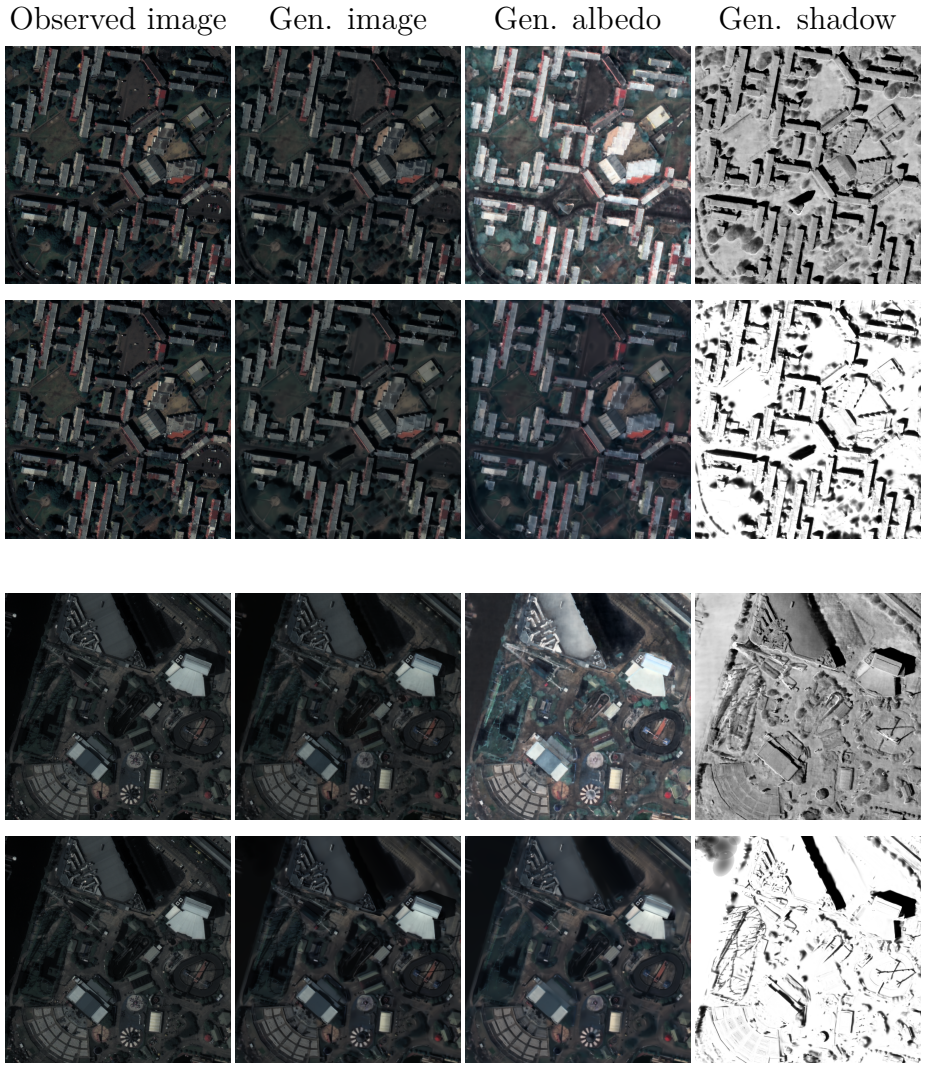


Figure 6.12: Visual comparison of the scene albedo and shadows generated by EO-NeRF (top) and EOGS (bottom) for IARPA_001 and IARPA_003. Note that the scaling for all images is the same except for the albedo that is rescaled independently to show the entire dynamic.

6.7 Conclusions & Future Work

In this chapter we introduced EOGS, the first Gaussian-Splatting-based framework for Earth observation. By adapting 3DGS to the requirements of satellite imagery, including shadow modeling and camera-specific corrections, EOGS achieves digital surface model reconstruction quality comparable to the current state of the art, EO-NeRF, while reducing optimization time by more than two orders of magnitude. This makes Gaussian-splatting-based volumetric representations a practical solution for large-scale and potentially high-throughput remote-sensing pipelines.

Our analysis also highlights the current limits of the method. EOGS performs especially well in regions with high image coverage, where it can reconstruct fine details at a fraction of the computational cost, but it is less robust in areas with low coverage, strong vegetation, or other challenging visibility conditions. These cases suggest that stronger priors, better initialization schemes, and additional regularization may be necessary to improve performance in unevenly observed regions. More generally, handling seasonal changes and heterogeneous land-cover types remains an open challenge for volumetric Earth observation.

Looking ahead, an important direction is to combine EOGS with richer sources of information and more scalable architectures. Multi-modal fusion with SAR, LiDAR, or multispectral imagery could provide complementary cues where optical coverage is insufficient, while hierarchical or streaming formulations could allow local Gaussian or radiance-field patches to be linked into larger global reconstructions. Integrating uncertainty-aware extensions would further improve the practical value of the framework, allowing operators not only to reconstruct large scenes efficiently, but also to quantify confidence in the derived products and prioritize future acquisitions where the model is least certain.

Chapter 7

Conclusions and Outlook

In this dissertation we have explored how deep learning and volumetric scene representations can be used to build *3D world models* that are both expressive and practical. Starting from the classical theory of light transport and the volume rendering integral, we studied how modern radiance-field-style models can be specialized and extended to address four concrete challenges: enhancing resolution from limited viewpoints, quantifying reliability and uncertainty, hallucinating missing data in a principled way, and deploying these models at the scale and constraints of satellite remote sensing. This chapter summarizes the main contributions, discusses their limitations, and outlines possible directions for future work and for the broader field of 3D scene representation.

7.1 Summary of Contributions

Theoretical Foundations for Volumetric Representations

The thesis begins by revisiting light transport in participating media and the volume rendering integral, and by showing how these classical concepts specialize to neural scene representations. The derivations in Chapter 2 make explicit the connection between radiance, density, and color in the continuous setting, and the corresponding quantities used in Neural Radiance Fields (NeRF), 3D Gaussian Splatting (3DGS), and related models. This provides a unified notation for rays, cameras, and rendering operators, and clarifies how modern neural methods can be seen as specific parameterizations of the same underlying physical integral. Beyond serving as a reference, this perspective helps to reason about what is gained and lost when introducing approximations (*e.g.*, constant-color approxmation, alpha compositing, or explicit splats) into the rendering pipeline.

Resolution: Deep 3D World Models for MISR

Chapter 3 tackles the problem of multi-image super-resolution (MISR) in the regime of large viewpoint changes and arbitrary camera placements. Traditional MISR methods are often built around optical flow in the image plane, which works well when parallax is small but breaks down in the presence of strong 3D effects. The EpiMISR model proposed in this thesis replaces optical flow with explicit epipolar geometry: features are sampled along rays consistent with the calibrated cameras, and aggregated by a transformer that implicitly builds a 3D radiance-like feature field.

This geometry-aware design allows EpiMISR to handle arbitrary numbers of views, to gracefully fall back to single-image SR when only one observation is available, and to outperform flow-based methods in scenarios with large disparities. In the context of this thesis, EpiMISR illustrates how volumetric world models can be used not only for novel view synthesis, but also as powerful priors for solving classical inverse problems such as super-resolution.

Reliability: Uncertainty for Gaussian Splatting

Chapter 4 turns to the question of *reliability*: how much can we trust the predictions of a learned volumetric model, and how can that trust be quantified? Focusing on 3D Gaussian Splatting, the Stochastic Gaussian Splatting (SGS) framework extends each Gaussian primitive with a Bayesian treatment of its parameters, turning rendering into a stochastic process. Monte Carlo sampling from the learned posterior yields not only expected pixel intensities but also per-pixel predictive variances.

A dedicated loss term encourages these variances to correlate with true rendering errors, yielding well-calibrated uncertainty maps while preserving the speed and reconstruction quality that make 3DGS attractive in the first place. SGS is, to the best of our knowledge, the first uncertainty-aware extension of Gaussian Splatting, and shows that radiance-field-style models can provide *probabilistic* predictions rather than just single best guesses.

Missing Data: Physics-based Priors for Dynamics

While the preceding chapters focus on static scenes, Chapter 5 addresses the challenge of missing viewpoints and missing time: in practical applications we rarely observe the full spatio-temporal evolution of a scene, yet we would like to imagine plausible dynamics that are consistent with both the data and the laws of physics. Rather than training a large video diffusion model, the MotionCraft framework assumes access to a powerful still-image generator and asks how to animate its outputs.

The key idea is to decouple appearance and motion. A physics simulator (for

fluids, rigid bodies, or multi-agent systems) produces a sequence of optical flows in the image plane, which encode how scene elements should move. These flows are applied not to pixels, but to the internal noise latents of an image diffusion model, and the denoising process reconstructs each frame. This latent-space warping faithfully imposes the prescribed motion while letting the model invent new content where needed (previously occluded regions, extended objects, consistent reflections and lighting), resulting in realistic, controllable videos in a zero-shot manner.

Within the narrative of the thesis, MotionCraft plays a dual role. On the one hand, it is a practical zero-shot video generator that competes favorably with prior text-to-video methods. On the other hand, it exemplifies how *physics-based priors on motion* can regularize otherwise underconstrained problems. We speculate that MotionCraft could help volumetric and generative models to hallucinate missing views and time steps in a principled way.

Remote Sensing: Efficient Gaussian Splatting for Earth Observation

Chapter 6 investigates how to adapt volumetric representations to the specific requirements of satellite remote sensing, where we must process large collections of multi-date, multi-view images under tight computational budgets. The Earth-Observation Gaussian Splatting (EOGS) framework specializes 3DGS to the characteristics of satellite imagery, achieving digital surface models with quality comparable to NeRF-based Earth observation methods ones, while reducing optimization time from days to minutes per scene. This makes volumetric radiance fields a realistic option for high-throughput, potentially planetary-scale satellite pipelines, addressing the efficiency and scalability challenge highlighted in the introduction.

A Unifying Perspective

Taken together, these contributions show that radiance-field-style representations are not just a curiosity for photorealistic view synthesis, but a versatile substrate for *world modeling*. The thesis demonstrates how geometric constraints, probabilistic reasoning, physics-based priors, and application-specific adaptations can be layered on top of volumetric models to address resolution, reliability, missing data, and scalability in diverse settings. This suggests a broader view of 3D world models as a common language between imaging, graphics, machine learning, and applications.

7.2 Outlook: Towards Learned World Models

Stepping back from the individual methods, the arc of this thesis points towards a broader vision: learned 3D world models as a central abstraction at the intersection of perception, simulation, and decision-making.

In this context, a *3D world model* should be understood as more than a tool for novel view synthesis. It is an internal representation that organizes what an agent or system knows about the geometry, appearance, dynamics, and uncertainty of its environment, and that can be queried to predict observations, infer missing information, or evaluate hypothetical outcomes. Radiance fields and Gaussian Splatting are compelling building blocks for such models because they already provide a structured, spatially grounded, and differentiable description of the world.

On the one hand, radiance fields and Gaussian Splatting are rooted in classical physics: they describe how light interacts with matter along rays, and their rendering equations are direct descendants of the volume rendering integral. On the other hand, their parameters are learned from data using flexible function approximators, making them capable of representing complex real-world scenes. This fusion of physical structure and statistical learning is, in many ways, the defining feature of modern 3D representation research.

Looking ahead, several trends seem likely:

- **From scenes to environments.** Today's models typically reconstruct one scene at a time; tomorrow's models will represent entire environments, with persistent 3D memory and dynamic updates as new data arrives. In robotics and AR/VR, such models could serve as a shared substrate for localization, navigation, interaction, and simulation.

- **From pixels to actions.** As uncertainty-aware radiance fields and dynamic generative models mature, their outputs will increasingly feed into downstream tasks: planning a robot's path, assessing risk in autonomous driving, or estimating confidence in Earth observation products. Representations that explicitly encode uncertainty and physical constraints will be key to making such pipelines robust.

- **From reconstruction to imagination.** Generative priors like MotionCraft hint at a future where 3D world models do not merely reconstruct what was observed, but can also imagine counterfactuals: how a scene would look under different lighting, weather, or interventions; how a city might evolve over years; how a physical system would respond to hypothetical forces. Combining realistic generative capabilities with physical plausibility and uncertainty quantification is an ambitious but increasingly tangible goal.

- **From isolated tools to shared infrastructure.** As robotics, remote sensing, medical imaging, and entertainment all adopt volumetric representations,

there is an opportunity to converge on common abstractions and software stacks. Just as convolutional networks became a ubiquitous building block for 2D perception, radiance-field-style models and their variants may become a standard interface between sensors, simulators, and decision systems.

These trends come with challenges: scaling models while keeping them interpretable and energy-efficient; ensuring datasets are representative and do not encode harmful biases; designing evaluation protocols that go beyond PSNR and SSIM to capture the usefulness of a representation in downstream tasks. Yet they also outline an exciting trajectory.

In this context, the contributions of this thesis can be seen as small but concrete steps towards richer world models. EpiMISR shows how explicit geometry can unlock new regimes of super-resolution. SGS demonstrates that uncertainty can be brought to fast, explicit volumetric representations. MotionCraft illustrates how physics can be woven into generative priors to fill in missing dynamics. EOGS shows that volumetric representations can be made efficient and practical in demanding real-world applications such as satellite photogrammetry.

If 2D convolutional networks were the lingua franca of the previous decade in computer vision, volumetric and radiance-field-based models are strong candidates for the next. As they continue to evolve from scene representations into fuller 3D world models, incorporating geometry, physics, uncertainty, and generative capabilities, they bring us closer to the long-standing goal of endowing machines with a useful internal model of the three-dimensional, dynamic world we inhabit.

Bibliography

[1] Eirikur Agustsson and Radu Timofte. "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[2] Tom M. Apostol. *Calculus / Vol. 1, One-variable calculus, with an introduction to linear algebra.* eng. 2nd ed. New York: John Wiley, 1967. ISBN: 9780471000068.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[4] S Derin Babacan, Rafael Molina, and Aggelos K Katsaggelos. "Total variation super resolution using a variational approach". In: *2008 15th IEEE International Conference on Image Processing*. IEEE. 2008, pp. 641–644.

[5] Simon Baker and Takeo Kanade. *Super Resolution Optical Flow*. Tech. rep. CMU-RI-TR-99-36. Pittsburgh, PA: Carnegie Mellon University, Oct. 1999.

[6] Emmanuel P Baltsavias and Dirk Stallmann. "Metric information extraction from SPOT images and the role of polynomial mapping functions". In: *XVII ISPRS Congress, Commission IV*. Swiss Federal Institute of Technology, Institute of Geodesy and Photogrammetry. 1992.

[7] Yanqi Bao et al. "3d gaussian splatting: Survey, technologies, challenges, and opportunities". In: *arXiv preprint arXiv:2407.17418* (2024).

[8] Yuxiang Bao et al. "LatentWarp: Consistent Diffusion Latents for Zero-Shot Video-to-Video Translation". In: *arXiv preprint arXiv:2311.00353* (2023).

[9] Jonathan T Barron et al. "Mip-nerf 360: Unbounded anti-aliased neural radiance fields". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5470–5479.

[10] Jonathan T Barron et al. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5855–5864.

[11] Jonathan T Barron et al. "Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields". In: *arXiv preprint arXiv:2304.06706* (2023).

[12] Ross A Beyer, Oleg Alexandrov, and Scott McMichael. "The Ames Stereo Pipeline: NASA's open source software for deriving and processing terrain data". In: *Earth and Space Science* 5.9 (2018), pp. 537–548.

[13] Rishika Bhagwatkar et al. "A Review of Video Generation Approaches". In: *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*. 2020, pp. 1–5. DOI: 10.1109/PICC51425.2020.9362485.

[14] Goutam Bhat et al. "Deep burst super-resolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9209–9218.

[15] Goutam Bhat et al. "NTIRE 2022 burst super-resolution challenge". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1041–1061.

[16] Wenjing Bian et al. "NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 4160–4169. DOI: 10.1109/CVPR52729.2023.00405.

[17] Camille Billouard et al. "SAT-NGP : Unleashing Neural Graphics Primitives for Fast Relightable Transient-Free 3D Reconstruction From Satellite Imagery". In: *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*. 2024, pp. 8749–8753. DOI: 10.1109/IGARSS53475.2024.10641775.

[18] Andreas Blattmann et al. "Align your latents: High-resolution video synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22563–22575.

[19] Yochai Blau and Tomer Michaeli. "The perception-distortion tradeoff". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6228–6237.

[20] James F Blinn and Martin E Newell. "Texture and reflection in computer generated images". In: *Communications of the ACM* 19.10 (1976), pp. 542–547.

[21] Marc Bosch et al. "Semantic stereo for incidental satellite images". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019, pp. 1524–1532.

[22] Tim Brooks, Aleksander Holynski, and Alexei A Efros. "Instructpix2pix: Learning to follow image editing instructions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18392–18402.

[23] Tim Brooks et al. “Video generation models as world simulators”. In: (2024). URL: https://openai.com/research/video-generation-models-as-world-simulators.

[24] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. “Revising densification in gaussian splatting”. In: *arXiv preprint arXiv:2404.06109* (2024).

[25] Shengqu Cai et al. “Generative Rendering: Controllable 4D-Guided Video Generation with 2D Diffusion Models”. In: *arXiv preprint arXiv:2312.01409* (2023).

[26] Edwin Earl Catmull. *A subdivision algorithm for computer display of curved surfaces*. The University of Utah, 1974.

[27] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. “Pix2video: Video editing using image diffusion”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 23206–23217.

[28] Xiongli Chai et al. “Super-Resolution Reconstruction for Stereoscopic Omnidirectional Display Systems via Dynamic Convolutions and Cross-View Transformer”. In: *IEEE Transactions on Instrumentation and Measurement* (2023).

[29] Xiongli Chai et al. “Tccl-net: Transformer-convolution collaborative learning network for omnidirectional image super-resolution”. In: *Knowledge-Based Systems* 274 (2023), p. 110625.

[30] Subrahmanyan Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.

[31] Jia-Ren Chang and Yong-Sheng Chen. “Pyramid Stereo Matching Network”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 5410–5418.

[32] Xiangyu Chen et al. “Activating more pixels in image super-resolution transformer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22367–22377.

[33] Xingyu Chen et al. “Hallucinated neural radiance fields in the wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12943–12952.

[34] Xiaojie Chu, Liangyu Chen, and Wenqing Yu. “NAFSSR: Stereo image super-resolution using NAFNet”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1239–1248.

[35] Franklin C. Crow. “The aliasing problem in computer-generated shaded images”. In: *Seminal Graphics: Pioneering Efforts That Shaped the Field, Volume 1*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 57–63. ISBN: 158113052X. URL: https://doi.org/10.1145/280811.280976.

[36] François Darmon et al. “Improving neural implicit surfaces geometry with patch warping”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6260–6269.

[37] Dawa Derksen and Dario Izzo. “Shadow neural radiance fields for multi-view satellite photogrammetry”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1152–1161.

[38] Chao Dong et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[39] Akshay Dudhane et al. “Burstormer: Burst Image Restoration and Enhancement Transformer”. In: *arXiv preprint arXiv:2304.01194* (2023).

[40] Dave Epstein et al. “Diffusion self-guidance for controllable image generation”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 16222–16239.

[41] Gabriele Facciolo, Carlo de Franchis, and Enric Meinhardt. “MGM: A Significantly More Global Matching for Stereovision”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. 90. 2015, pp. 1–12.

[42] Joël Foramitti. “AgentPy: A package for agent-based modeling in Python”. In: *Journal of Open Source Software* 6.62 (2021), p. 3065.

[43] Carlo de Franchis et al. “An automatic and modular stereo pipeline for pushbroom images”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 2-3 (2014), pp. 49–56.

[44] Sara Fridovich-Keil et al. “Plenoxels: Radiance fields without neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5501–5510.

[45] Yang Fu et al. “COLMAP-Free 3D Gaussian Splatting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 20796–20805.

[46] Michael Gableman and Avinash Kak. “Incorporating season and solar specificity into renderings made by a NeRF architecture using satellite images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[47] Yarin Gal et al. “Uncertainty in deep learning”. PhD thesis. University of Cambridge, 2016.

[48] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[49] Jian Gao et al. “Relightable 3D Gaussian: Real-time Point Cloud Relighting with BRDF Decomposition and Ray Tracing”. In: *arXiv:2311.16043* (2023).

[50] Daniel Geng and Andrew Owens. "Motion Guidance: Diffusion-Based Image Editing with Differentiable Motion Estimators". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=WIAO4vbnNV.

[51] Michal Geyer et al. "TokenFlow: Consistent Diffusion Features for Consistent Video Editing". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=lKK50q2MtV.

[52] Lily Goli et al. "Bayes' Rays: Uncertainty Quantification for Neural Radiance Fields". In: *arXiv preprint arXiv:2309.03185* (2023).

[53] Henri Gouraud. "Computer display of curved surfaces". PhD thesis. 1971.

[54] MSU Graphics and Media Lab. *MSU Video Frame Interpolation Benchmark dataset*. 2022. URL: https://videoprocessing.ai/benchmarks/video-frame-interpolation-dataset.html.

[55] Aditya Grover, Manik Dhar, and Stefano Ermon. "Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11829.

[56] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. "Evaluating scalable bayesian deep learning methods for robust computer vision". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 318–319.

[57] Yuqi Han et al. "Super-NeRF: View-consistent Detail Generation for NeRF super-resolution". In: *arXiv preprint arXiv:2304.13518* (2023).

[58] Yihui He et al. "Epipolar transformers". In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2020, pp. 7779–7788.

[59] Amir Hertz et al. "Prompt-to-Prompt Image Editing with Cross-Attention Control". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=_CDixzkzeyb.

[60] Heiko Hirschmuller. "Stereo processing by semiglobal matching and mutual information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2007), pp. 328–341.

[61] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[62] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021. URL: https://openreview.net/forum?id=qw8AKxfYbI.

[63] Jonathan Ho et al. "Imagen video: High definition video generation with diffusion models". In: *arXiv preprint arXiv:2210.02303* (2022).

[64] Jonathan Ho et al. "Video diffusion models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 8633–8646.

[65] Philipp Holl, Nils Thuerey, and Vladlen Koltun. "Learning to Control PDEs with Differentiable Physics". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HyeSin4FPB.

[66] Yanting Hu et al. "Channel-Wise and Spatial Feature Modulation Network for Single Image Super-Resolution". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.11 (2020), pp. 3911–3927. DOI: 10.1109/TCSVT.2019.2915238.

[67] Detian Huang et al. "CLSR: Cross-Layer Interaction Pyramid Super-Resolution Network". In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.11 (2023), pp. 6273–6287. DOI: 10.1109/TCSVT.2023.3266222.

[68] Han Huang et al. "Differentiable Neural Architecture Search for Extremely Lightweight Image Super-Resolution". In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.6 (2023), pp. 2672–2682. DOI: 10.1109/TCSVT.2022.3230824.

[69] Letian Huang et al. "On the Error Analysis of 3D Gaussian Splatting and an Optimal Projection Strategy". In: *arXiv preprint arXiv:2402.00752* (2024).

[70] Qian Huang, Minghao Hu, and David Jones Brady. "Array camera image fusion using physics-aware transformers". In: *arXiv preprint arXiv:2207.02250* (2022).

[71] Yan Huang et al. "Unfolding the alternating optimization for blind super resolution". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5632–5643.

[72] Brian KS Isaac-Medina, Chris G Willcocks, and Toby P Breckon. "Exact-NeRF: An Exploration of a Precise Volumetric Parameterization for Neural Radiance Fields". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 66–75.

[73] Itseez. *Open Source Computer Vision Library*. https://github.com/itseez/opencv. 2015.

[74] Wenzel Jakob et al. "Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering". In: *Transactions on Graphics (Proceedings of SIGGRAPH)* 41.4 (July 2022). DOI: 10.1145/3528223.3530099.

[75] Rasmus Jensen et al. "Large scale multi-view stereopsis evaluation". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2014, pp. 406–413.

[76] Yoonwoo Jeong et al. “Self-calibrating neural radiance fields”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 5846–5854.

[77] Yingwenqi Jiang et al. “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5322–5332.

[78] Laurent Valentin Jospin et al. “Hands-on Bayesian neural networks–A tutorial for deep learning users”. In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48.

[79] James T. Kajiya. “The rendering equation”. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’86. New York, NY, USA: Association for Computing Machinery, 1986, pp. 143–150. ISBN: 0897911962. DOI: 10.1145/15922.15902. URL: https://doi.org/10.1145/15922.15902.

[80] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems* 30 (2017).

[81] Alex Guy Kendall. “Geometry and uncertainty in deep learning for computer vision”. PhD thesis. University of Cambridge, 2017.

[82] Bernhard Kerbl et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering”. In: *ACM Transactions on Graphics* 42.4 (July 2023), pp. 1–14. URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[83] Levon Khachatryan et al. “Text2video-zero: Text-to-image diffusion models are zero-shot video generators”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 15954–15964.

[84] Shakiba Kheradmand et al. “3D Gaussian Splatting as Markov Chain Monte Carlo”. In: *arXiv preprint arXiv:2404.09591* (2024).

[85] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[86] Fangyuan Kong et al. “Residual local feature network for efficient super-resolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 766–776.

[87] Thomas Krauß et al. “The fully automatic optical processing system CATENA at DLR”. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 40-1/W1 (2013), pp. 177–181.

[88] Samuli Laine et al. “Modular Primitives for High-Performance Differentiable Rendering”. In: *ACM Transactions on Graphics* 39.6 (2020).

[89] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30 (2017).

[90] Bertrand Le Saux et al. "2019 data fusion contest [technical committees]". In: *IEEE Geoscience and Remote Sensing Magazine* 7.1 (2019), pp. 103–105.

[91] Bruno Lecouat, Jean Ponce, and Julien Mairal. "Lucas-Kanade Reloaded: End-to-End Super-Resolution from Raw Image Bursts". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[92] Marc Levoy. *CS 248 - Introduction to Computer Graphics*. 2006. URL: https://graphics.stanford.edu/courses/cs248-06/.

[93] Yan-Ran Li, Dao-Qing Dai, and Lixin Shen. "Multiframe Super-Resolution Reconstruction Using Sparse Directional Regularization". In: *IEEE Transactions on Circuits and Systems for Video Technology* 20.7 (2010), pp. 945–956. DOI: 10.1109/TCSVT.2010.2045908.

[94] Jingyun Liang et al. "Swinir: Image restoration using swin transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 1833–1844.

[95] Zhihao Liang et al. "Gs-ir: 3d gaussian splatting for inverse rendering". In: *arXiv preprint arXiv:2311.16473* (2023).

[96] Zimu Liao et al. "Fisheye-GS: Lightweight and Extensible Gaussian Splatting Module for Fisheye Cameras". In: *arXiv preprint arXiv:2409.04751* (2024).

[97] Bee Lim et al. "Enhanced deep residual networks for single image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[98] Chen-Hsuan Lin et al. "Barf: Bundle-adjusting neural radiance fields". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 5741–5751.

[99] Yan Lu et al. "Geometry uncertainty projection network for monocular 3d object detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3111–3121.

[100] Ziwei Luo et al. "BSRT: Improving burst super-resolution with swin transformer and flow-guided deformable alignment". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 998–1008.

[101] Roger Marí, Thibaud Ehret, and Gabriele Facciolo. "Disparity Estimation Networks for Aerial and High-Resolution Satellite Images: A Review". In: *Image Processing On Line* 12 (2022), pp. 501–526.

[102] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. "Multi-date earth observation NeRF: The detail is in the shadows". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2035–2045.

[103] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. "Sat-nerf: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using rpc cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1311–1321.

[104] Ricardo Martin-Brualla et al. "Nerf in the wild: Neural radiance fields for unconstrained photo collections". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 7210–7219.

[105] N. Max. "Optical models for direct volume rendering". In: *IEEE Transactions on Visualization and Computer Graphics* 1.2 (1995), pp. 99–108. DOI: 10.1109/2945.468400.

[106] Julien Michel et al. "A new satellite imagery stereo pipeline designed for scalability, robustness and performance". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 5-2-2020 (2020), pp. 171–178.

[107] Ben Mildenhall et al. "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.

[108] Ben Mildenhall et al. "Nerf in the dark: High dynamic range view synthesis from noisy raw images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16190–16199.

[109] Ben Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[110] Ron Mokady et al. "Null-text inversion for editing real images using guided diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 6038–6047.

[111] Andrea Bordone Molini et al. "Deepsum: Deep neural network for super-resolution of unregistered multitemporal images". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.5 (2019), pp. 3644–3656.

[112] Antonio Montanaro et al. "Motioncraft: Physics-based zero-shot video generation". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 123155–123181.

[113] Thomas Müller et al. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM transactions on graphics (TOG)* 41.4 (2022), pp. 1–15.

[114] Meenal V Narkhede, Prashant P Bartakke, and Mukul S Sutaone. "A review on weight initialization strategies for neural networks". In: *Artificial intelligence review* 55.1 (2022), pp. 291–322.

[115] Haomiao Ni et al. "Conditional image-to-video generation with latent flow diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18444–18455.

[116] Michael Niemeyer et al. "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5480–5490.

[117] Michael Oechsle, Songyou Peng, and Andreas Geiger. "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 5589–5599.

[118] Keunhong Park et al. "Nerfies: Deformable Neural Radiance Fields". In: *ICCV* (2021).

[119] Naama Pearl, Tali Treibitz, and Simon Korman. "Nan: Noise-aware nerfs for burst-denoising". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12672–12681.

[120] Bui Tuong Phong. "Illumination for computer generated pictures". PhD thesis. 1974.

[121] Emilie Pic et al. "Pseudo pansharpening nerf for satellite image collections". In: *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2024, pp. 2650–2655.

[122] Dustin Podell et al. "Sdxl: Improving latent diffusion models for high-resolution image synthesis". In: *arXiv preprint arXiv:2307.01952* (2023).

[123] Tomaso Poggio et al. "Theory IIIb: Generalization in deep networks". In: *arXiv preprint arXiv:1806.11379* (2018).

[124] Albert Pumarola et al. "D-NeRF: Neural Radiance Fields for Dynamic Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[125] Yingjie Qu and Fei Deng. "Sat-mesh: Learning neural implicit surfaces for multi-view satellite reconstruction". In: *Remote Sensing* 15.17 (2023), p. 4297.

[126] Sameera Ramasinghe, Lachlan E MacDonald, and Simon Lucey. "On the frequency-bias of coordinate-mlps". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 796–809.

[127] Craig W Reynolds. "Flocks, herds and schools: A distributed behavioral model". In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 1987, pp. 25–34.

[128] Lawrence G Roberts. "Machine perception of three-dimensional solids". PhD thesis. Massachusetts Institute of Technology, 1963.

[129] Robin Rombach et al. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[130] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[131] Ewelina Rupnik, Mehdi Daakir, and Marc Pierrot-Deseilligny. "MicMac–a free, open-source solution for photogrammetry". In: *Open Geospatial Data, Software and Standards* 2.14 (2017).

[132] Ullman S. "The interpretation of structure from motion". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426.

[133] Paul-Edouard Sarlin et al. "From coarse to fine: Robust hierarchical localization at large scale". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12716–12725.

[134] Luca Savant Aira, Gabriele Facciolo, and Thibaud Ehret. "Gaussian Splatting for Efficient Satellite Image Photogrammetry". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 5959–5969.

[135] Luca Savant Aira, Diego Valsesia, and Enrico Magli. "Modeling Uncertainty for Gaussian Splatting". In: *IEEE Transactions on Neural Networks and Learning Systems* 36.6 (2025), pp. 11657–11663. DOI: 10.1109/TNNLS.2025.3553582.

[136] Luca Savant Aira et al. "Deep 3D World Models for Multi-Image Super-Resolution Beyond Optical Flow". In: *IEEE Access* 12 (2024), pp. 188902–188913. DOI: 10.1109/ACCESS.2024.3514188.

[137] Jianxiong Shen et al. "Conditional-flow NeRF: Accurate 3D modelling with reliable uncertainty quantification". In: *European Conference on Computer Vision*. Springer. 2022, pp. 540–557.

[138] Jianxiong Shen et al. "Estimating 3D Uncertainty Field: Quantifying Uncertainty for Neural Radiance Fields". In: *arXiv preprint arXiv:2311.01815* (2023).

[139] Jianxiong Shen et al. "Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations". In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 972–981.

[140] Yahao Shi et al. "Gir: 3d gaussian inverse rendering for relightable scene factorization". In: *arXiv preprint arXiv:2312.05133* (2023).

[141] Uriel Singer et al. "Make-A-Video: Text-to-Video Generation without Text-Video Data". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=nJfylDvgzlq.

[142] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[143] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2020.

[144] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems* 32 (2019).

[145] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2020.

[146] Mohammed Suhail et al. "Generalizable patch-based neural rendering". In: *European Conference on Computer Vision*. Springer. 2022, pp. 156–174.

[147] Mohammed Suhail et al. "Light field neural rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8269–8279.

[148] Cheng Sun, Min Sun, and Hwann-Tzong Chen. "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5459–5469.

[149] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. "Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 9370–9376.

[150] Ivan E Sutherland and Gary W Hodgman. "Reentrant polygon clipping". In: *Communications of the ACM* 17.1 (1974), pp. 32–42.

[151] Matthew Tancik et al. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains". In: *NeurIPS* (2020).

[152] Jun Tang et al. "CTVSR: Collaborative Spatial-Temporal Transformer for Video Super-Resolution". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2023), pp. 1–1. DOI: 10.1109/TCSVT.2023.3340439.

[153] C Vincent Tao and Yong Hu. "A comprehensive study of the rational function model for photogrammetric processing". In: *Photogrammetric engineering and remote sensing* 67.12 (2001), pp. 1347–1358.

[154] Zachary Teed and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 402–419.

[155] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288.

[156] U.S. Geological Survey. *The Universal Transverse Mercator (UTM) Grid*. ENGLISH. Tech. rep. Report. Reston, VA: U.S. Geological Survey, 2001. DOI: 10.3133/fs07701. URL: https://doi.org/10.3133/fs07701.

[157] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9446–9454.

[158] Aaron Van Den Oord, Oriol Vinyals, et al. "Neural discrete representation learning". In: *Advances in neural information processing systems* 30 (2017).

[159] Mukund Varma et al. "Is Attention All That NeRF Needs?" In: *The Eleventh International Conference on Learning Representations*. 2022.

[160] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[161] Chen Wang et al. "NeRF-SR: High Quality Neural Radiance Fields using Supersampling". In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 6445–6454.

[162] Peng Wang et al. "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction". In: *arXiv preprint arXiv:2106.10689* (2021).

[163] Qianqian Wang et al. "Ibrnet: Learning multi-view image-based rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4690–4699.

[164] Xiaofeng Wang et al. "MVSTER: Epipolar transformer for efficient multi-view stereo". In: *European Conference on Computer Vision*. Springer. 2022, pp. 573–591.

[165] Zhihao Wang, Jian Chen, and Steven CH Hoi. "Deep learning for image super-resolution: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3365–3387.

[166] Zhou Wang. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[167] Zirui Wang et al. "NeRF−−: Neural Radiance Fields Without Known Camera Parameters". In: *arXiv preprint arXiv:2102.07064* (2021).

[168] Gary S Watkins. "A real time visible surface algorithm". PhD thesis. 1970.

[169] Songlin Wei et al. "FG-NeRF: Flow-GAN based Probabilistic Neural Radiance Field for Independence-Assumption-Free Uncertainty Estimation". In: *arXiv preprint arXiv:2309.16364* (2023).

[170] Turner Whitted. "An improved illumination model for shaded display". In: *Commun. ACM* 23.6 (June 1980), pp. 343–349. ISSN: 0001-0782. DOI: 10.1145/358876.358882. URL: https://doi.org/10.1145/358876.358882.

[171] Lance Williams. "Casting curved shadows on curved surfaces". In: *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*. 1978, pp. 270–274.

[172] Jay Zhangjie Wu et al. "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7623–7633.

[173] Tong Wu et al. "Recent advances in 3d gaussian splatting". In: *Computational Visual Media* 10.4 (2024), pp. 613–642.

[174] Haolin Xiong et al. "Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting". In: *arXiv preprint arXiv:2312.00206* (2023).

[175] Gengshan Yang et al. "Hierarchical Deep Stereo Matching on High-resolution Images". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5510–5519.

[176] Guandao Yang et al. "Pointflow: 3d point cloud generation with continuous normalizing flows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4541–4550.

[177] Lior Yariv et al. "Volume rendering of neural implicit surfaces". In: *Advances in neural information processing systems* 34 (2021), pp. 4805–4815.

[178] Alex Yu et al. "pixelnerf: Neural radiance fields from one or few images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4578–4587.

[179] Zehao Yu et al. "MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

[180] Feihu Zhang et al. "GA-Net: Guided aggregation net for end-to-end stereo matching". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 185–194.

[181] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3836–3847.

[182] Shuo Zhang, Song Chang, and Youfang Lin. "End-to-end light field spatial super-resolution network using multiple epipolar geometry". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5956–5968.

[183] Zhe Zhang et al. "Recurrent Interaction Network for Stereoscopic Image Super-Resolution". In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.5 (2023), pp. 2048–2060. DOI: 10.1109/TCSVT.2022.3220412.

[184] M. Zwicker et al. "EWA volume splatting". In: *Proceedings Visualization, 2001. VIS'01.* IEEE. 2001, pp. 29–538. DOI: 10.1109/VISUAL.2001.964490.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.