ISO/IEC JTC 1/SC29/WG1N101164
107th JPEG Meeting, Brussels, Belgium, 12 to 18 April 2025

ISO/IEC JTC 1/SC 29/WG 1
(& ITU-T SG16)

Coding of Still Pictures

JBIG.
Joint Bi-level Image Experts Group

JPEG
Joint Photographic Experts Group

TITLE: JPEG Radiance Fields State of the Art and Challenges v1.1

SOURCE: JPEG RF

EDITORS: Davi Lazzarotto, Stuart Perry, Antonio Pinheiro

PROJECT: -

STATUS: Draft

REQUESTED ACTION: For information and feedback

DISTRIBUTION: WG1

Contact:
ISO/IEC JTC 1/SC 29/WG 1 Convener – Prof. Touradj Ebrahimi
EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: Touradj.Ebrahimi@epfl.ch

JPEG RF
State of the Art and Challenges

Table of Contents

1. Background and Motivation

The use of images as a representation of visual information has become ubiquitous in modern society, facilitating the sharing of digital media worldwide. Traditional imaging modalities represent visual data as color triplets sampled on a regular two-dimensional (2D) grid and can be easily transposed to be directly rendered in regular screens. At the same time, recent years have seen the rise of advanced systems that allow viewers to navigate and interact with a scene in order to create a deeper sense of immersion than simply visualizing it through a display. Such systems are set to create enhanced experiences when compared to traditional rendering, where users are able to have a much more natural interaction with the digital world.

Immersive applications rely heavily on new technologies for rendering, acquisition, and representation of visual data to increase quality of experience. Modern devices such as virtual reality headsets, augmented reality glasses, light field monitors and others are able to display visual data with depth cues, better taking advantage of the characteristics of the human visual system. However, traditional images and videos cannot fully leverage the advanced display capabilities of such technologies. Alternative imaging modalities representing scenes and objects with higher dimensionality are therefore required to enable immersive representations.

Light fields can represent a scene as a set of regularly arranged 2D views. In contrast, holographic data represents a three-dimensional (3D) environment by encoding the wavefront of light emanating from a scene as an interference pattern. However, these two modalities are restricted in terms of how a user can navigate through a scene since they are not defined in the 3D domain. Point clouds and meshes instead depict objects as sets of unconnected points or connected surfaces, respectively, representing color attributes directly in the volumetric space and allowing for increased navigation and interaction.

Point clouds and meshes representing real-world objects can be obtained from photogrammetry, a process that extracts 3D measures from a set of images depicting the same object or scene from different angles. In most applications, the resulting model obtains constant color values which can correctly represent diffuse reflection, but fail to capture specular effects produced by glossy surfaces. Radiance fields adopt a different approach by modeling the intensity of colored light emanating from each point of the 3D space in all directions. Two main techniques have been proposed to achieve this goal:

Neural Radiance Fields (NeRFs) and 3D Gaussian Splatting (3DGS). NeRFs employ a neural network to produce color and density values from a geometric coordinate and angular direction given as input, which is optimized using backpropagation to produce images that closely match a set of training images captured from known camera positions, generating an implicit representation of the scene and allowing for novel view synthesis. 3DGS adopts instead an explicit approach by representing light as Gaussian distributions employing spherical harmonics to model specular effects and view-dependent color, with parameters learned during an optimization process similar to the one conducted for NeRFs.

Recent research works have proposed a multitude of distinct methods for the representation of radiance fields. While NeRFs originally employed a single neural network to represent the visual information of an entire scene, later works proposed the use of multiple smaller networks or the addition of spherical harmonics to account for angular variations in light incidence rather than a learned function. On the other hand, Gaussian splatting models represent 3D objects and scenes as a collection of Gaussian splats with associated attributes. The initial framework attached spherical harmonics, scale, rotation, and opacity values to each splat, but other methods proposed different sets of attributes. Research has been rapidly evolving, with newly proposed methods adopting techniques incompatible with previous works. This scenario poses a challenge to interoperability between devices for the encoding and decoding of radiance fields. Moreover, the memory footprint of these techniques can be elevated given the size of the neural networks or the number of splats attributes involved, mainly since bitrate is usually not considered during optimization. For these reasons, the JPEG RF activity is being initiated to investigate the role that the JPEG standardization committee may assume to facilitate the widespread adoption of radiance fields for visual media representation.

This document describes the state of the art on radiance fields highlighting the associated challenges according to the following outline:

1. Key terminologies related to radiance field technologies are first defined.
2. State-of-the-art technologies for radiance field representation based on NeRFs and 3DGS are described and compared.
3. Use cases and requirements for radiance field representations relevant to society and commercial interests are outlined.

4. Challenges related to their use for the outlined applications are discussed.
5. Finally, considerations related to the role of JPEG in this scenario are raised.

2. Terminology

To facilitate the reading, this section defines key terms used in the remaining sections of this document. The terms are organized in the following sub-sections:

1. **Fundamentals;**
2. **Photogrammetry;**
3. **Neural Radiance Field (NeRF);**
4. **3D Gaussian Splatting (3DGS);**
5. **Other Relevant Terminology.**

2.1 Fundamentals

This sub-section defines, by alphabetical order, fundamental terms for this document.

Gaussian Splatting: Representation of a radiance field using a collection of 3D Gaussian distributions to model the color and opacity of a region in space.

Light Field: Representation of the plenoptic function that takes the form of an ordered multi-dimensional grid of values modelling the amount of light traveling in every direction through every point in the defined grid.

NeRF: Neural radiance field. A representation of the radiance field of a scene using a fully connected deep network.

Novel View Synthesis: The process of generating novel views of a scene from angles not directly captured with a camera.

Photogrammetry: The process of creating a 3D representation of a scene using images taken from different viewpoints and the localization of points of the scene in 3D space.

Point Cloud: A collection of points in 3D space representing the external surface of an object or terrain.

Radiance Field: A function representing the color and density of light in a region in space as a function of the position and viewing direction.

2.2 Photogrammetry

This sub-section defines, by alphabetical order, key terms related to photogrammetry.

Stereo Pair: Two overlapping images taken from different viewpoints used to derive 3D information.

Multi-View Stereo (MVS): Techniques that use stereo correspondence for more than two images.

Structure-from-Motion (SfM): A photogrammetric method for estimating 3D structures from a set of 2D images.

Image Matching: The process of finding corresponding points in different images.

Tie Point: Common points identified in multiple images used to align and stitch them together.

2.3 Neural Radiance Fields (NeRF)

This sub-section defines, by alphabetical order, key terms related to NeRF.

Density: A measure of how much light is absorbed or scattered as it passes through a point in space.

Implicit Representation: A parametric function that represents 3D geometry and color without an explicit mesh or point cloud.

Positional Encoding: A method used to encode spatial coordinates into a higher-dimensional space to capture fine details.

Ray Marching: A technique for tracing rays through a volume to compute the color of the light received at a viewpoint.

Volume Rendering: A technique used in NeRF to compute 2D images from 3D volumes by integrating colors and densities along rays.

2.4 3D Gaussian Splatting (3DGS)

This sub-section defines, by alphabetical order, key terms related to 3DGS.

Blending Function: A function used to blend overlapping splats to produce an estimate of the color of a pixel in the reconstructed viewpoint.

Densification: The process used to split or clone Gaussians to fill holes or add detail to the representation of the radiance field.

Opacity: The degree to which an object or region reduces how much light passes through it.

Splatting Radius: The radius around each point within which the Gaussian influence is computed.

Weight Function: In Gaussian splatting, a function that determines the influence of each point on the rendered image.

2.5 Other Relevant Terminology

This sub-section defines, by alphabetical order, other terms relevant to this document.

Holography: A technique used to record and reconstruct light waves based on their amplitude and phase information to create 3D images of objects.

Simultaneous Localization and Mapping (SLAM): A method of creating a map of an unknown environment captured with multiple views using photogrammetry while simultaneously keeping track of the location and viewing angle of each view used within that map.

3. Radiance Fields: Basic Concepts and Key Technologies

3.1 Basic Concepts

Traditional 2D images and videos represent the light received at one point in space from a given angle range, usually corresponding to a narrow field of view. Immersive imaging modalities expand the representation of visual data by depicting the scene with a wider angular extent or by incorporating the light received at a broader region in space larger than a single point. A general representation for all immersive modalities is the plenoptic function $P(x, y, z, \theta, \Phi, t, \lambda)$, which models the light intensity received at any point in space (x, y, z), from any angle of view (θ, Φ), at any time (t), and for any wavelength (λ) inside the visible spectrum. Since representing the entire plenoptic function at all possible values for all variables would require a massive amount of data, immersive imaging techniques usually select a bounded domain over which they are defined. Most representation formats take advantage of the limitations of the human visual system which perceives colors with only three types of cones and collapse the wavelength dimension λ into only three color values within a defined colorspace. Moreover, modalities where the visual content is static do not require the representation of the time dimension t.

Many imaging modalities further reduce the domain of the plenoptic function while still allowing for increased immersion when compared to 2D images. Omnidirectional images depict the light received at only one spatial position (x, y, z) but expand the range of angular incidence (θ, Φ), effectively representing the light coming from all directions. Light fields are defined over different positions (x, y, z) sampled from a plane while keeping a restricted field of view defined by the angles (θ, Φ). Holograms do not represent light intensity directly, but instead record interference patterns between the light emitted from the scene and a

reference wave over a plane. These patterns can be used to faithfully reconstruct the incident light at that plane from a wide range of incoming angles.

The common aspect between these image modalities is that they represent the light where it was acquired. The immersiveness enabled by light fields or holograms is derived from the capabilities of the devices employed to obtain them, which employ complex mechanisms to measure the light in a wider range than regular cameras. For this reason, the observer is always restricted to the angle and position where the device was placed during the capture of the image. In use cases where visualization from novel viewpoints is required, the visual information must be modeled by a 3D model defined in the scene space associated with an adequate rendering technique.

Acquiring a 3D model from a set of regular images is a challenging task due to the lack of volumetric information since images only represent a 2D projection of the scene. Traditional techniques can estimate the positions and surfaces of objects if multiple images describing the scene from different angles are available, notably through the use of photogrammetry. Photogrammetric techniques are able to produce 3D models from 2D data by identifying features common to multiple images and estimating their spatial position. Structure-from-motion (SfM) algorithms are often employed for this purpose, producing a sparse point cloud with the feature points of the scene and estimating the extrinsic and intrinsic parameters of the cameras. Such methods can be coupled with multi-view stereo (MVS) and enhanced reconstruction algorithms to produce a denser point cloud or a mesh and enable the visualization of the analysed scene when combined with a rendering method. The quality of the images obtained from the model observed from different viewpoints depends on many factors such as the types of photogrammetry algorithms employed, the representation format for the model (e.g. point cloud or mesh), and the employed rendering techniques.

Radiance fields have recently been proposed as an alternative representation format to model a scene in the 3D space. These models represent the density and color of the scene as a function of the spatial position and viewing direction. In order to obtain images from these models, a simulated camera can be placed in a given spatial position and each pixel of the corresponding image can be regarded as a sample of the plenoptic function. For each sample, a ray is cast onto the scene and the color of the corresponding pixel can be obtained through the integration of the density and color along the ray using volume

rendering techniques. A radiance field can be either computed with an implicit function, which usually produces color and density values with a neural network receiving as input the position of the sample and direction of the ray, or explicitly by assigning radiance field values to specific regions in space.

One of the main advantages of the use of radiance fields rather than photogrammetry techniques to generate novel views from a set of images obtained at different viewpoints is that the employed volume rendering algorithm is differentiable. Therefore, the parameters of the radiance field representation, either implicit or explicit, can be optimized through gradient descent to produce renderings closely matching the available images at the known viewpoints.

The radiance field methods described in this document can be classified as NeRF or 3DGS models. While NeRF models compute the light emitted by each point in the scene with a multi-layer perceptron, 3DGS models simulate multiple points in space emitting colored light with intensity following an anisotropic 3D Gaussian distribution. For this reason, NeRF models are classified as implicit since the spatial variation in the scene properties is implicitly computed by a neural network. On the other hand, 3DGS models are classified as explicit since the radiance at each point is explicitly defined by the parameters of each Gaussian Splat.

3.2 Radiance Field Processing Pipeline

The pipeline for the creation, coding, and synthesation of images using radiance field techniques is illustrated in Figures 1 and 2.

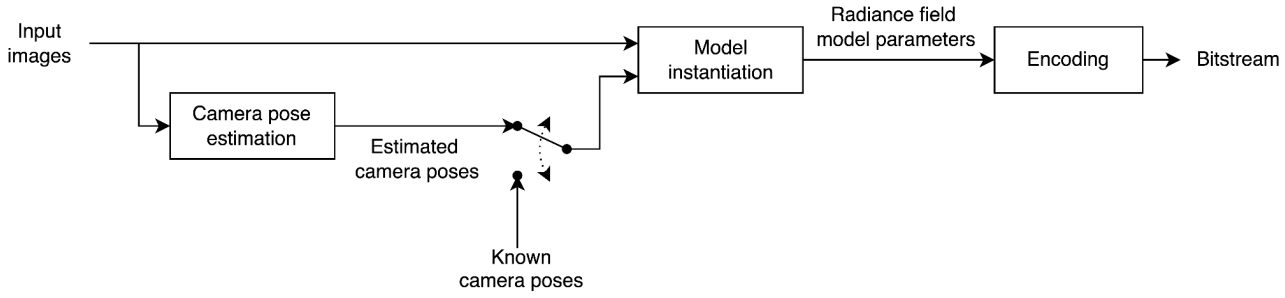

Figure 1 - workflow of the process for instantiating a radiance field model from a set of images and encoding the result into a bitstream

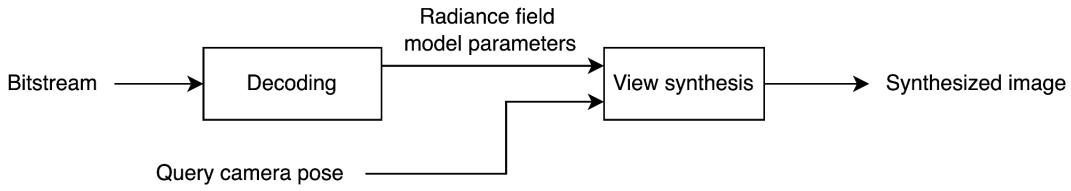

Figure 2 - workflow of the process for decoding a bitstream and reconstructing a viewpoint given camera parameters such as visualization angle and focal length

Camera pose estimation: The training process for radiance field methods such as NeRF and 3DGS requires a set of images with associated camera poses. In some cases, these poses may already be known if the cameras include other mechanisms such as position sensors are fixed in a pre-defined arrangement. However, when this information is lacking, the poses can be estimated using algorithms such as SfM. Correctly evaluating the camera poses is crucial for the generation of an accurate radiance field.

Model instantiation: This process instantiates a radiance field model using either NeRF, 3DGS or other similar techniques. It usually involves the use of stochastic gradient descent or another algorithm to optimize the parameters of the radiance field to be able to accurately represent the scene at the poses present in the training set.

Encoding: This process converts the parameters of a radiance field model into a compact bitstream. The format of the received parameters depends on the adopted method for the instantiation of the radiance field. For instance, if a NeRF is employed, these parameters would correspond to the weights of the neural network implicitly describing the radiance field. In the case where a 3DGS model is used, the parameters refer to the value of the attributes of all the splats of the model.

Decoding: This process reconstructs the parameters of a radiance field model from the corresponding bitstream. If the encoder and decoding processes are lossless, the values of the radiance parameters would be the same as the values sent as input to the encoder. Otherwise, these processes are categorized as lossy.

View synthesis: This process synthesizes an image corresponding to the scene described by a radiance field model viewed from a query camera pose. Usually, this camera pose is different from the poses seen by the model during the model instantiation process. In many use cases, a sequence of camera poses following a continuous trajectory is given as input to generate multiple views that are concatenated into a video.

The remainder of this section explores in detail different techniques to obtain radiance fields. As a point of comparison, traditional photogrammetry methods are first described and discussed. Then, relevant research contributions on NeRF and 3DGS are described in detail, comparing the advantages and drawbacks between different solutions.

3.3 Photogrammetry

Photogrammetry, meaning the act of deriving measurements from photographs, is commonly used to refer to computer algorithms that extract spatial and geometric information about a 3D scene or object from overlapping 2D images, taken from different perspectives. Photogrammetry algorithms analyze image features such as texture, edges, and patterns to determine the relative position of detected feature points and reconstruct a 3D representation of the captured scene or object.

Photogrammetry differs from 3D scanning since it relies only on color images, instead of using structured laser light to measure the locations of points. As a result, photogrammetry techniques can be used with minimal specialized hardware, leveraging consumer-grade cameras and software. Nevertheless, more accurate and high-end photogrammetry systems are very complex and costly, since they rely on the use of many high-resolution cameras and complex lighting systems. Such photogrammetry systems are able to produce detailed models with realistic textures, making them ideal for applications demanding high visual quality. One other important advantage of photogrammetry is the fact that it can be applied to objects of varying scales, from small artifacts to expansive environments.

Despite these advantages, photogrammetry presents some challenges and limitations. For instance, the reconstruction process is computationally intensive, especially for large datasets or high-resolution outputs. Moreover, occluded or obscured surfaces can lead to incomplete models, necessitating manual intervention or complementary capture techniques. Photogrammetry also struggles with moving subjects, as consistency across images is critical for accurate reconstruction.

In the context of 3D video, photogrammetry plays a vital role in capturing real-world subjects or scenes for integration into volumetric workflows, including for VR and AR applications, digital games and the TV and film industry. Key applications include capturing detailed 3D environments that serve as dynamic backdrops or interactive settings for 3D video; generating volumetric assets that preserve fine details such as facial expressions and clothing textures; digitization of artifacts, architecture, and natural landscapes, for cultural preservation; and the construction of 3D maps for geomorphic studies and geographic information systems [1].

3.3.1 View Synthesis

Unlike NeRF and 3DGS, which are techniques that generate models that can be used directly for view synthesis, photogrammetry is a technique that focuses on the creation of a 3D representation, namely a point cloud or a 3D mesh, that can then be used to obtain new viewpoints of the object or scene that is being considered. In this sense, the view synthesis process can use any traditional point cloud or mesh rendering technique independently from the model instantiation process, which will be described in the next section.

The rendering of point cloud data can be performed in a very simple way by using direct point rendering, which displays each point as a pixel. Despite the simplicity and efficiency for real-time applications, direct rendering offers limited accuracy in the representation of smooth surfaces, especially for sparser point clouds. Splatted rendering improves visual continuity by representing points as discs or splats, which can be used when additional point attributes such as normal vectors are available, creating smoother transitions between points. These attributes can be determined during the model instantiation process or computed afterward by using the information about local neighborhoods in a point cloud.

The rendering of 3D meshes is generally able to produce 2D images able to achieve high levels of realism across diverse applications, including the well-established digital games and TV/film industries. Rasterization is one of the most common and simple methods, rendering meshes into 2D frames using vertex projection and fragment shaders, offering real-time efficiency suitable for gaming and VR. Ray tracing techniques rely on casting rays from the perspective of the viewer into the scene, checking for intersections between the ray and the mesh. In photogrammetry applications, enhancement of ray tracing is achieved by estimating light interactions, such as reflections, refractions, and shadows, from the high-resolution textures. While ray tracing provides unparalleled realism when compared

with rasterization, it is computationally intensive, particularly for dense photogrammetric meshes.

3.3.2 Model Instantiation

The photogrammetry pipeline for determining the 3D model of an object or scene typically involves three main steps:

Image Acquisition: During the image acquisition stage, multiple high-resolution overlapping photographs are taken from varying angles around the object or scene. These images may be acquired either by well-positioned fixed cameras, for which the locations and camera parameters are known [2, 3], or by one or more cameras taking overlapping photographs of the object from different points of view, with no information about camera position or parameters [4, 5]. Important factors to be taken into account in the image acquisition stage are: the lighting conditions, which should be as consistent throughout the object as possible; the overlapping between the acquired images, which should be enough to provide good spatial matching; good color contrast between the object and the background (ideally a chroma-key backdrop is employed); image quality, including resolution, sharpness, focus, little depth-of-field, among others. One common option in high-end systems is the use of coded markers that are placed in the object that is being scanned. These markers increase the accuracy and efficiency of the next steps of the photogrammetry process, notably for reflective, transparent, and otherwise featureless surfaces. Figure 3 shows a representation of camera positions for the image acquisition stage of the photogrammetry process.

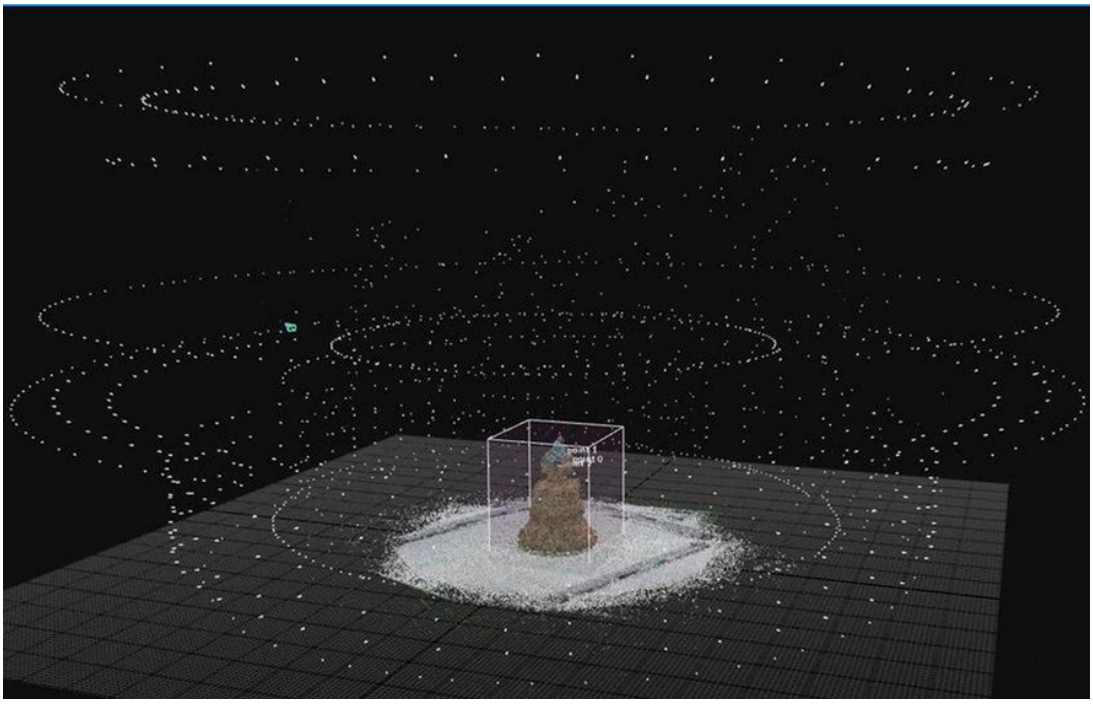

Figure 3 - Acquisition of the volumetric representation of an object, showing camera positions. Source: [6].

Image Matching: The computer algorithm searches the input photographs for overlapping areas and defines how they should be stitched together, in a process similar to the construction of a 3D puzzle of the acquired object. MVS algorithms use the epipolar geometry data of the acquisition system to aid the matching of points across different views. In SfM techniques, feature extraction algorithms are used to determine tie points, which correspond to distinctive image features, such as edge points, lines and corners, or dense textures on objects that can be uniquely recognized across multiple images, enabling accurate alignment and depth estimation [7, 8, 9, 10]. Some solutions further enrich the data using more complex techniques, known as Shape-from-Shading, which are able to improve the image matching process using lighting and shading cues [11, 12]. The final stage of the image mapping step is the definition of a transformation that maps the tie points between images, using algorithms based on projective geometry. Figure 4 shows a representation of the detection and matching of a tie point in the scene.

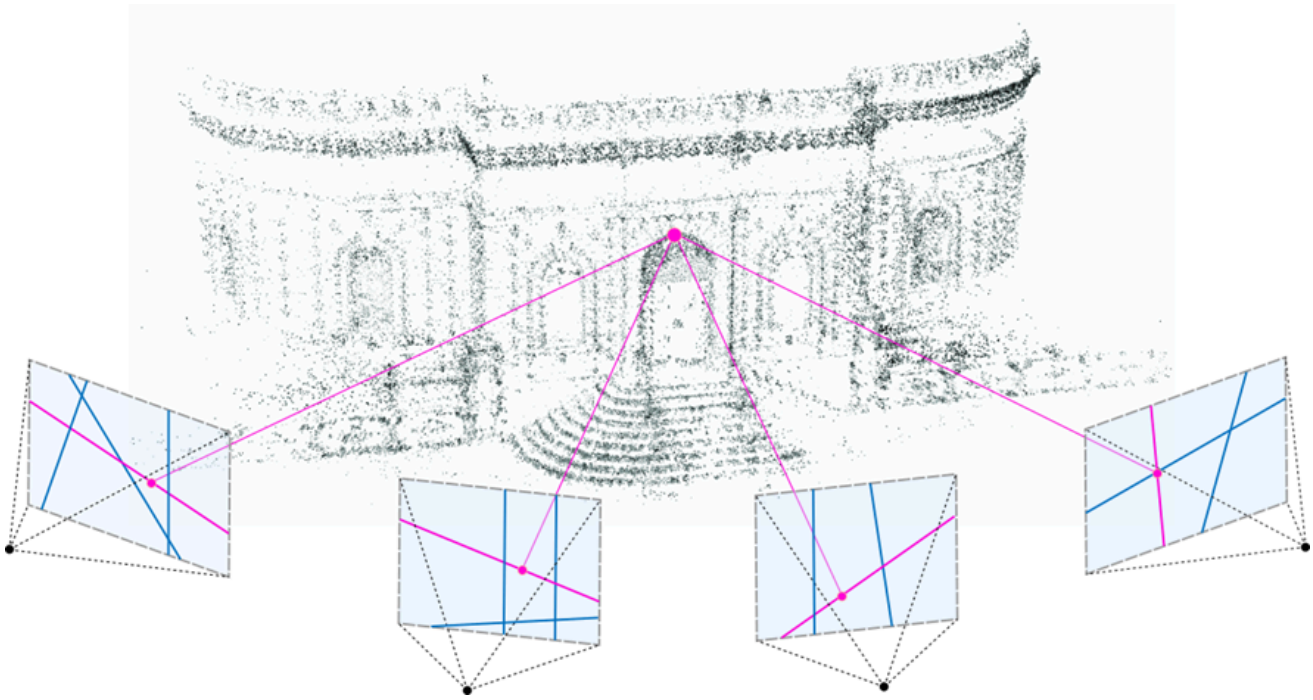

Figure 4 - Feature detection and matching. Source: [13].

3D Reconstruction: A point cloud or mesh is generated, representing the geometry and color of the subject. The 3D coordinates of the surface points are estimated by using projection rays (see Figure 5), corresponding to the lines of sight connecting each camera position to the feature points of the object, that were mapped in the image matching step [14]. The 3D coordinates of the object are positioned at the intersection of the rays that connect the cameras and the detected feature points. These points are then marked as the vertices of a point cloud, which can then be converted to a mesh using triangulation algorithms. After an optional refinement stage, the texture information of the original photographs is mapped to the 3D representation of the object. Advanced de-lighting algorithms can be used to enhance the texture information, improving the representation of lit and shaded areas across the surface of the model, in order to produce photorealistic 3D assets suitable for volumetric applications.

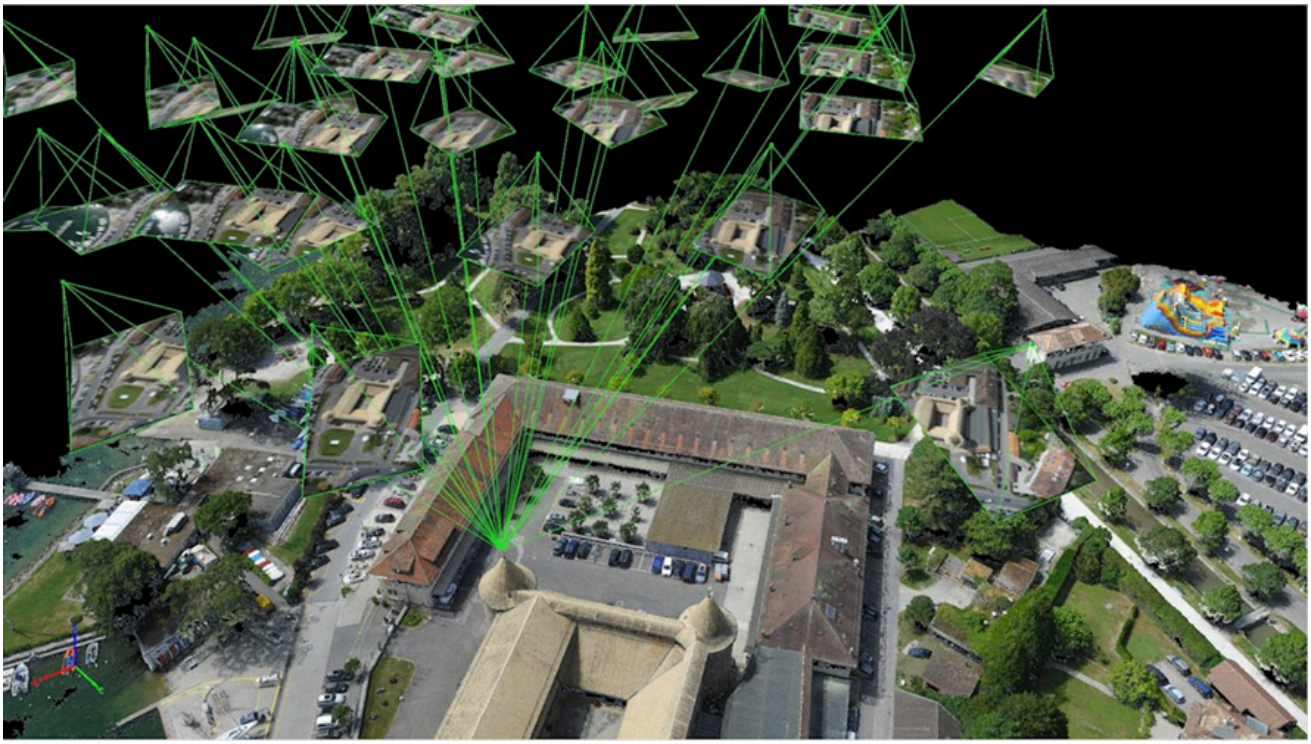

Figure 5 – 3D reconstruction using a ray cloud. Source: [15].

3.4 Neural Radiance Fields (NeRF)

NeRFs were proposed as the first technique to model radiance fields implicitly with neural networks. These techniques offered an alternative to traditional photogrammetry-based methods for 3D-based modeling of real-world scenes. This section provides an overview of the key NeRF technologies, starting by describing the basic algorithm for view synthesis followed by the description of how a NeRF model can be instantiated. While the model instantiation must be conducted prior to view synthesis in the processing pipeline, the former is described first in this section since an understanding of this process is required to describe the model instantiation algorithm. The last subsection compares the methodologies adopted by different variants of NeRF models.

3.4.1 View Synthesis

Ray Projection: The use of NeRFs was first proposed by Mildenhall et al. [16] to model the radiance field describing a scene implicitly with neural networks. The NeRF model can be used to model a plenoptic function using volume rendering to integrate the density and color at different positions and along a ray with a given direction.

The density of the radiance field can be interpreted as the differential probability of a ray terminating at an infinitesimal particle in a given position. Empty space is expected to have a density of zero and opaque surfaces are expected to have higher density values. The color modeled by the NeRF can be understood as the light emitted by an infinitesimal particle at a given direction. While the NeRF model allows the color to change depending on the direction to represent specular effects and view-dependent color, the density is only a function of the sampling position to ensure continuity.

The image from any given viewpoint of the radiance field can be obtained through the integration of the radiance field following volume rendering techniques. For each pixel of the image, a ray is cast in the view direction from the position of a simulated camera and the radiance field is integrated along the ray. This integral is estimated using a discrete set of samples of the NeRF at different positions along the ray. Each sample is computed by querying a neural network with the sample position and ray direction as input. The density and color of the radiance field at the sample are given as the output of this computation.

An illustration of this process can be observed in Figure 6.

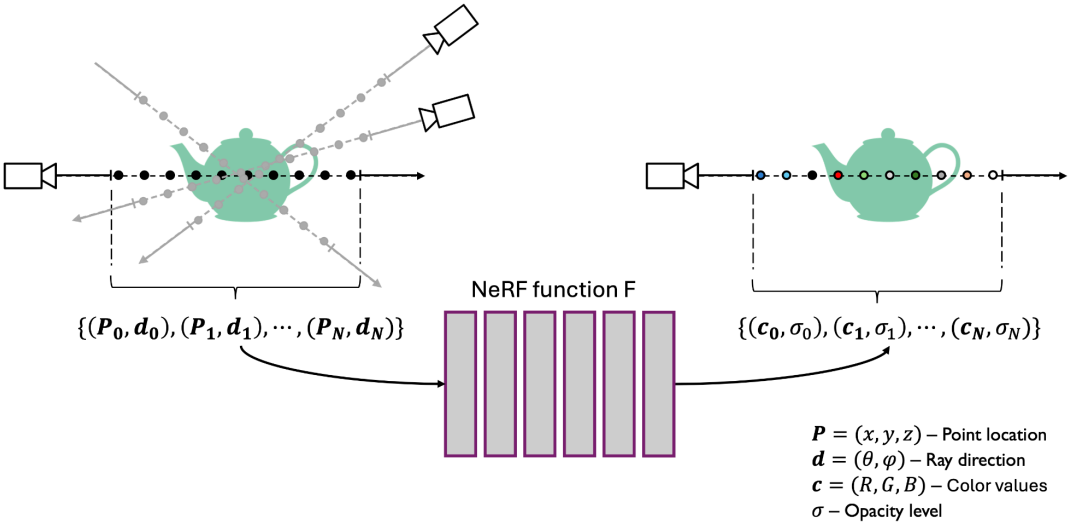

Figure 6 - Illustration of the volume rendering process employed using a NeRF.

Ray Sampling: While employing a high amount of samples approximates the integral required for volume rendering techniques more accurately, it also increases the computational complexity since the density and color functions have to be computed a higher number of times. In order to keep a good balance between these two factors, NeRF-based methods adopt a stratified sampling strategy. Firstly, the ray is sampled uniformly in a given interval. Then, a new operation is conducted by separating the ray into different regions and sampling each region with the weight computed by the first sampling procedure in that region. In this way, empty areas are ignored while sections of the ray with higher density and closer to the origin of the ray are more finely sampled.

Model Function: The original NeRF models the density and color with a neural network. A multi-layer perceptron (MLP) first processes the input coordinates and outputs the density and a feature vector, which is concatenated with the direction to be further processed by another set of layers to produce the color. This architecture ensures that the density is only affected by the position while the color can be influenced by the ray direction as well, allowing for specular effects while maintaining material continuity.

Positional Encoding: While the neural network representing the density and color in the scene space is commonly seen as a universal approximator for any given function,

multi-layer perceptrons have a bias towards low-frequency representations [16]. Feeding directly the geometric coordinates of a position as the input to the network would thus hinder the ability of the NeRF to represent high frequency details. For this reason, a positional encoding is employed where the input to the model is given by Equation (1).

$$\gamma(p) = (sin(2^0\pi p), cos(2^0\pi p), \ldots, sin(2^{L-1}\pi p), cos(2^{L-1}\pi p)) \quad (1)$$

This function is applied separately to each dimension in the input after normalization to the range [-1,1], i.e. the input value *p* is set to the *x*, *y*, and *z* values of the coordinate. The *L* parameter can be configured by the user and determines the size of the positional encoding vector as well as the number of frequencies captured by this representation. The final encoding contains coefficients that vary at different rates according to the position and is adopted to improve the network power to represent fine visual detail. An illustration of the combination of ray sampling and positional encoding can be observed in Figure 7.

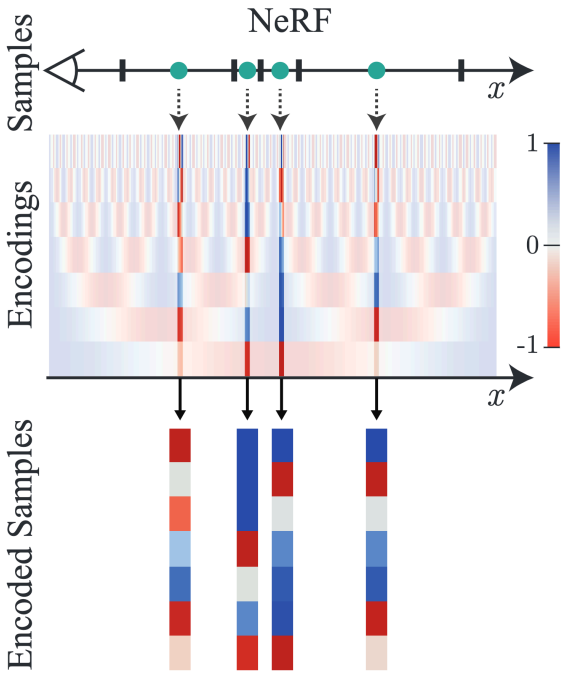

Figure 7 - Illustration of the sampling and positional encoding processes in NeRF. Source: [17]

3.4.2 Model Instantiation

The weights of the neural network describing the NeRF model are randomly initialized and require training to accurately represent the visualized scene. The training data employed for this purpose is a set of images with known camera parameters. When artificial images are used, the camera parameters employed during the generation of the images can be used. For natural images, SfM methods are usually employed to estimate the camera poses and intrinsic parameters. Since each pixel of the set of images corresponds to a sample of the radiance field at a given position and direction, the model is optimized to minimize a loss function related to the difference between samples generated from the model and real samples from the training set.

3.4.3 Alternative Techniques

Despite the advantages demonstrated by implicit radiance field modeling, the original NeRF presents significant drawbacks. For instance, rendering a single image requires a large amount of computing time since the neural network should be queried multiple times for each pixel. Moreover, the visual quality of the renderings at queried viewpoints is highly dependent on the quality of the training images and the model was not robust to fluctuations in lighting and required a large amount of data for the optimization process. Representing the scene with only one network also hinders the ability of the model to represent large scenes with accuracy.

Several variants of the original model have therefore been proposed to mitigate these issues, employing different techniques to improve the quality and reduce the complexity of the model. Relevant NeRF variants are described in this section and classified according to the method employed to handle different stages of the algorithm. A description of the classification criteria for each NeRF variant is given below.

Ray Model: In order to compute the radiance field from the NeRF model, the ray is assumed to be a line with infinitesimal thickness, and only points lying exactly in this line are considered to compute the radiance field. This process is prone to issues such as aliasing and blurring to appear in the rendering images if the camera distance is different from what is present in the training set.

An alternative technique was proposed by Mip-NeRF [17] where the ray is instead modeled as a cone rather than a line. The computation of the input encoding is also adapted to be given by the expected value of the encoding function over a frustum instead of a point.

Model Function: the original NeRF employs two MLPs to model the density and color of the scene: the first for coarse sampling using uniform intervals and the second for fine sampling following an importance-based sampling strategy. Samples derived from both operations are employed to determine the final color value of the corresponding pixel. Mip-NeRF 360 [22] also adopts a coarse-to-fine approach but employs the first network only to provide density values to inform the fine sampling operation, which are then used to compute the corresponding color. The first network is denominated the proposal MLP while the second is called NeRF MLP. kiloNeRF [18] proposed to partition the space into blocks with limited size and representing each block with a different smaller network. This approach reduces the complexity required for each forward pass and thus the total complexity of rendering an image. NeRF++ [19] proposed to model the scene with two networks, splitting the space into a sphere and an inverse sphere to represent foreground and background respectively. Other methods employ a single network to represent the entire space and thus reduce the computational complexity and memory footprint. Plenoctrees [23] proposed instead to represent color and density values explicitly in the 3D space in order to avoid the computation of the neural network and reduce rendering time. Due to the lack of neural-based modeling, Plenoctrees and other related methods cannot be categorized as NeRFs, but are nonetheless included in this section as they are traditionally referred to in the same context.

Positional Encoding: the position served as input to the model network is encoded with the function described in Equation (1) in the original NeRF model. NSVF [20] proposed another method to compute the positional encoding in two stages. The first stage employs a voxel grid where each voxel corner is associated with a feature vector learned during training. The encoding is given by the trilinear interpolation between the feature vectors at the corners of the voxel containing the query point. The second stage employs the same sinusoidal functions from Equation (1). InstantNGP [21] proposes a different approach and defines multiple voxel grids at different resolutions. Instead of assigning different features to each position, the vectors are arranged in a hash map queried by voxel coordinates. The encoding for the point applies a similar process as NSVF for each resolution, interpolating

over neighboring voxel corners, and the final feature vector is the concatenation of the multi-resolution features.

Direction Handling: the direction is fed as input to the neural network to model the color in the original NeRF model. Plenoctrees [23] introduced an alternative technique where the model produces the amplitude for spherical harmonics (SH) coefficients. The final color at any direction can then be obtained by computing the corresponding values of the SH evaluated at that direction and combining them using the estimated amplitudes.

Table 1 orders previous works proposed for the generation of radiance fields using NeRF according to these described categories.

Name	Ray model	Model function	Positional encoding	Direction handling
NeRF [16]	Line	Coarse & Fine MLPs	Sinusoidal	Model input
NSVF [20]	Line	Single MLP	Learned features and sinusoidal	Model input
Mip-NeRF [17]	Cone	Single MLP	Sinusoidal	Model input
NeRF++ [19]	Line	Background & Foreground MLPs	Sinusoidal	Model input
Mip-NeRF 360 [22]	Cone	Proposal and NeRF MLPs	Sinusoidal	Model input
Plenoctrees [23]	Line	Explicit model	-	SH
Plenoxels [24]	Line	Explicit model	-	SH
TensoRF [25]	Line	Explicit model	-	SH
KiloNeRF [18]	Line	Multiple space-bound MLPs	Sinusoidal	Model input

InstantNGP [21]	Line	Single MLP	Spatial hash	Model input

Table 1 - Previous works for radiance field representation with NeRF

3.4.4 Model Coding

In applications where memory requirements are restricted, the storage size needed for the neural networks and feature vectors representing NeRF with naïve encoding may be too high. For this reason, compression methods for NeRF models have been proposed, with the majority enabling the compression of the model parameters from previous works.

Inspired by the original NeRF, cNeRF [26] proposed the joint optimization of the network to minimize the entropy of the parameters of the neural networks representing both coarse and fine components at the same time as the error in the rendered images. However, the majority of compression methods are designed for NeRF models with components defined within the scene space, either explicitly [25] or implicitly as feature vectors [20, 21].

Leveraging a 4D representation combined with tensor decomposition proposed by TensoRF [25], ccNeRF [27] proposes a learning strategy to concentrate scene information on lower tensor ranks and truncate the obtained tensor to reduce the coding rate with scalable levels of detail defined by the rank of truncation. Another work [28] applied the discrete wavelet transform on 2D feature planes combined with a learned binary mask to obtain frequency coefficients encoded with run-length encoding and Huffman coding. NeRFCodec [29] employs convolutional autoencoders trained for image compression and finetunes them on feature planes using a loss function balancing rate and distortion.

Other works focused instead on compressing multi-resolution feature grids such as those proposed by InstantNGP [21], which substantially increased the memory requirements of NeRF techniques. VQAD [30] employed a dictionary with quantized vectors which are soft-indexed during training to ensure differentiability. The optimization process minimized the entropy of the features and thus reduce the coding rate after entropy coding. SHACIRA [31] proposed to represent the feature grids as latents quantized with Gumbel annealing which are decoded for each level to obtain the features employed during rendering. CNC [32] leveraged a learned context model to improve the efficiency of entropy coding and

further reduce coding rate. Binary radiance fields (BiRF) [33] proposed instead a binary feature grid, combining 3D and 2D grids for the representation of the radiance fields.

3.5 3D Gaussian Splatting (3DGS)

Unlike NeRFs, 3DGS models represent radiance fields as sets of anisotropic Gaussian functions with a centroid in the 3D space, covariance matrix, opacity and spherical harmonics to represent the directional aspects of color in the radiance field. Like NeRFs and Photogrammetry, 3DGS can be divided into view synthesis and model instantiation phases.

3.5.1 View Synthesis

In this operation, the 3DGS model of the radiance field is used to synthesize a view of the scene from an arbitrary perspective. This is performed by using the 3DGS model to determine the colour of each pixel in an image of the scene from the perspective of a position in space and a view angle selected by the user. Figure 8 illustrates the computation of the 3DGS model to synthesize an arbitrary viewpoint.

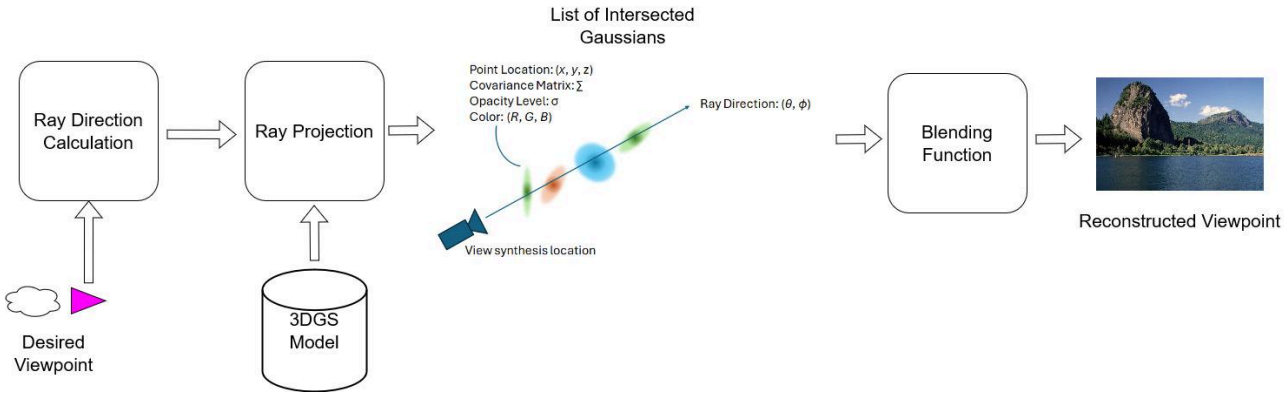

Figure 8: Synthesis of a viewpoint of a scene by use of a Gaussian Splatting model.

The view synthesis process can be divided into the following key stages:

Ray Direction Calculation: For a given viewpoint of the scene to be reconstructed, each pixel in the intended viewpoint image is represented by a ray that passes from the imaging plane through the radiance field away from the observer in the direction of observation. This

stage therefore determines the origin and direction of the ray given the position and angle of the simulated camera as well as the position of the pixel in the reconstructed image. For each ray projected through the radiance field, the model projects all of the 3D Gaussians onto the 2D image plane defined by the ray.

Ray Projection: This is the first step in the evaluation of the 3DGS model. The model outputs a list of Gaussians intersected by the ray recorded in depth order as well as their respective properties, such as covariance, color, opacity and centroid position. It should be noted that a ray does not need to pass directly through the centroid of a Gaussian for that Gaussian to be associated with the ray. The Gaussian is associated with the ray if it passes within the Splatting Radius of the Gaussian, which is defined as the 2D area of the Gaussian containing a set percentage of the probability density of the Gaussian. An area containing 99% of the probability density of the Gaussian distribution is a common choice [34].

Blending Function: The final color of the pixel in the rendered view is based on an integration of the properties of all of the Gaussian functions intercepted by the projected ray. The image is divided into 16×16 regular tiles and all 2D Gaussians touching each tile are sorted based on their original depth. The color of each pixel in the tile is then rasterized by sequentially alpha-blending the 2D Gaussians from front to back. The summation is termed the “Blending Function” and is performed in a depth-ordered fashion to emphasize Gaussians closer to the imaging plane.

3.5.2 Model Instantiation

In order to allow the model to be used for view synthesis, the model must first be initialized and then trained to represent a scene described by a set of images.

In the initialization phase, a preliminary 3DGS model is created from a set of 2D images of the scene. Figure 9 illustrates the initialization of a 3DGS model.

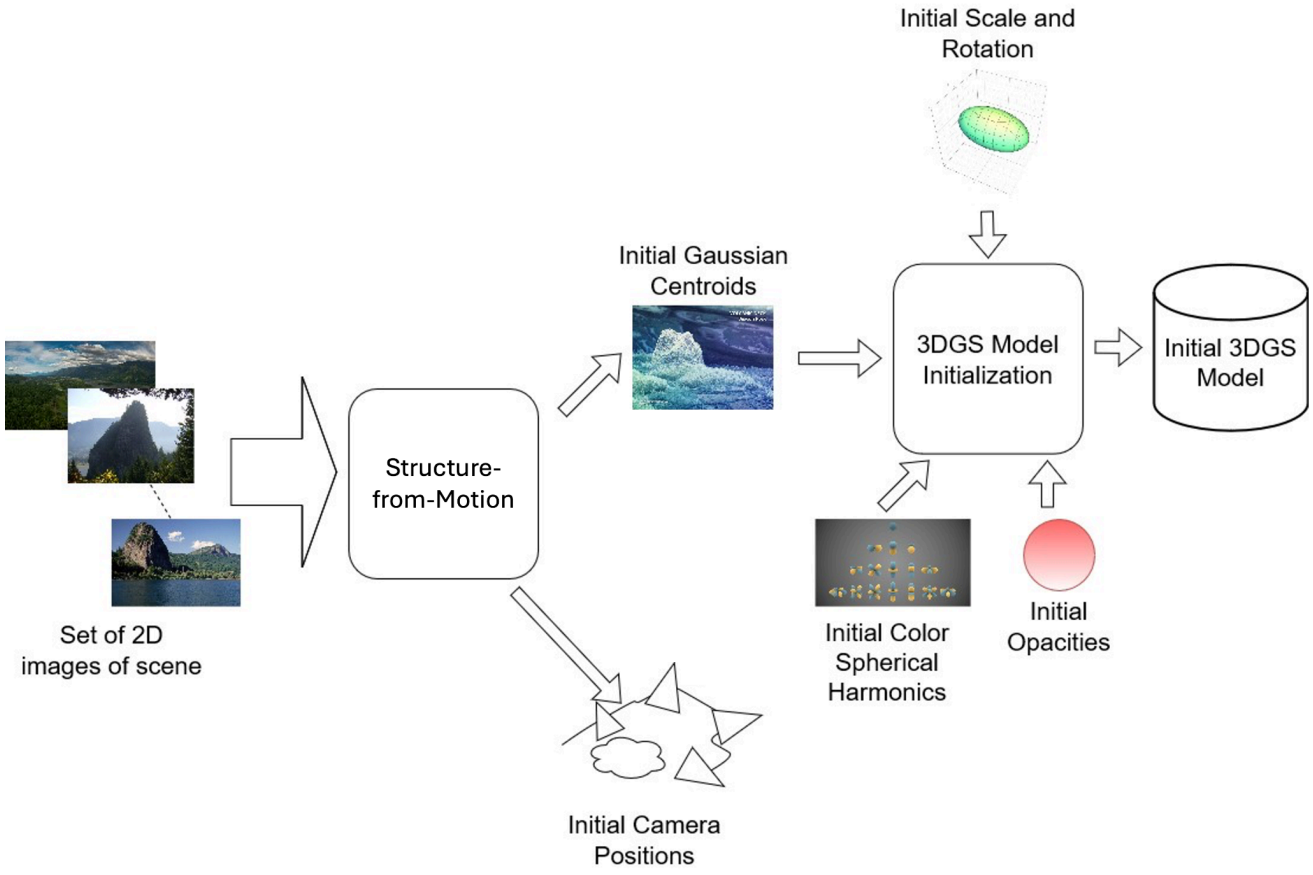

Figure 9: Initialization of a 3DGS radiance field model.

The initialization process can be divided into the following key stages:

Structure-from-Motion: The input information used to create the 3DGS model is a set of 2D images of the scene. In practice, between 200 and 1000 captured images of the scene are typically used, preferably from a wide range of positions and viewing angles. From the input images, SfM [35] is used to match points between pairs of images and, together with a model of the camera used to capture the images, the position of each camera in 3D space used to capture each image in the input set is determined. A side effect of determining the camera positions is the creation of a sparse point cloud describing the scene. The model instantiation stage builds this sparse presentation into a dense 3DGS model to describe the radiance field.

3DGS Model Initialization: The sparse point cloud generated by the SfM module is used as the initial centroids of a set of anisotropic Gaussian functions, one for each point in the initial point cloud.

Each Gaussian has the following attributes:

- A centroid initialized to be the *x*, *y* and *z* coordinates of the corresponding point in the initial point cloud produced by the SfM module.
- An opacity mapped between 0 to 1 to indicate the degree to which light passes through the Gaussian.
- A covariance matrix that defines the scale and orientation of the Gaussian. In practice, the covariance matrix is decomposed into a 3D scaling vector and a quaternion to represent the rotation of the Gaussian. This representation allows the Gaussian function to adapt according to scene structure such as open space or edges.
- A representation of a spherical function that defines the color of the Gaussian function when viewed from an arbitrary viewpoint. The spherical function is represented as a series of spherical harmonics of increasing order. The lowest order defines a view-independent color value, while the higher order harmonics define the change in appearance according to the view direction, enabling the representation of specular effects.

The set of anisotropic Gaussians defined in this step forms the framework by which the 3DGS model represents the radiance field.

The initialization phase is followed by a training phase. In the training phase, the initial 3DGS model is refined by reference to the set of images representing a scene through an iterative process to create a final dense model. Figure 9 illustrates the training process for a 3DGS model.

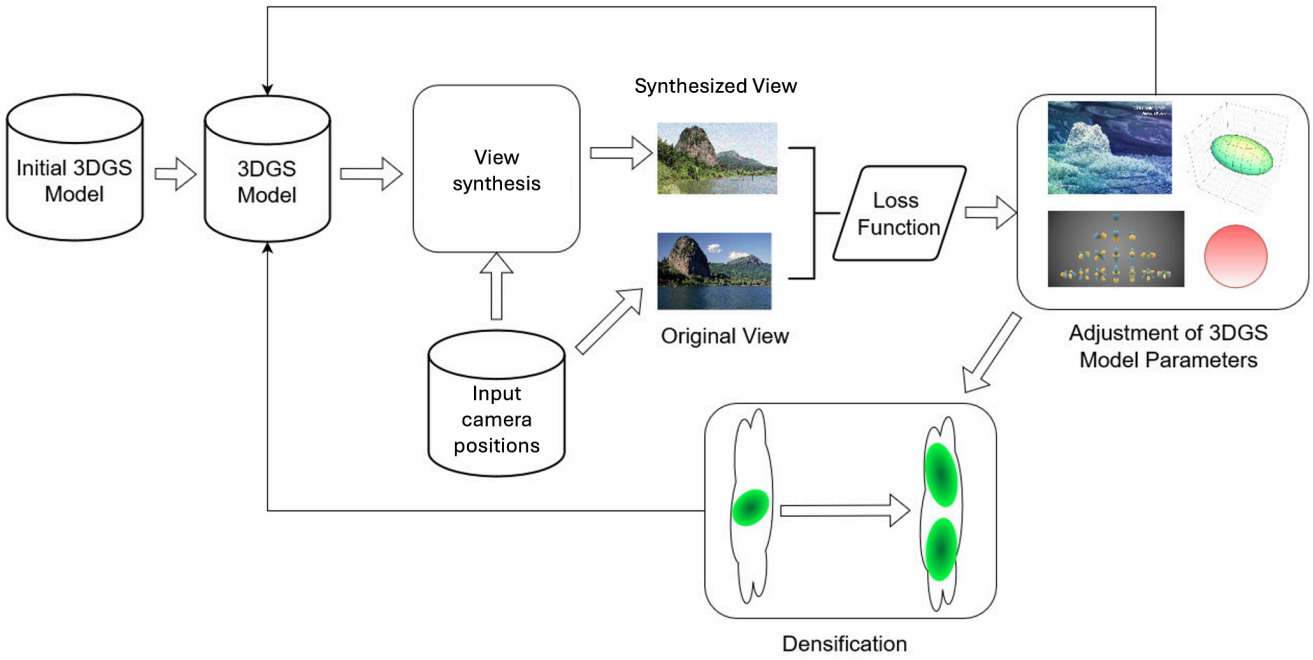

Figure 9: Training process for a 3DGS Model.

The training process can be divided into the following key stages:

View Synthesis: For the purpose of training, the view synthesis method described in Section 3.3.1 evaluates the current iteration of the 3DGS model to synthesize a viewpoint whose camera position matches a randomly selected camera position estimated by the SfM module described above, allowing for a direct comparison with ground-truth input image at the same camera position.

Loss Function Calculation: The loss between the synthesized view and the corresponding input image is computed. There are a variety of loss functions that may be applied, however a commonly used example is the L1 loss combined with the D-SSIM (Data Structural Similarity Index Measure) loss [34]. This loss is then used to adjust the attributes of each Gaussian and the process is repeated until convergence.

Adjustment of 3DGS Model Parameters: The positions, covariance, color and opacity parameters for each Gaussian in the 3DGS model are adjusted using an optimization

process to minimize the loss calculated by the loss function calculation module. An example of a commonly used optimization process is stochastic gradient descent.

Densification: In order to avoid over- or under-representation of the scene, a process of adaptive densification is performed at particular moments (for example, every 100 iterations) during the optimization process [34]. In areas where there are large gradients in the rendered scenes, Gaussians may be split or cloned to allow for the representation of finer details. Gaussians with very low opacities may be removed as these will not be visible in the rendered views.

The result of the training process is an explicit volumetric 3DGS model of the radiance field that can be evaluated to synthesize viewpoints from arbitrary positions.

3.5.3 Alternative Techniques

3DGS models are efficient and rapid to evaluate, making this technique suitable for real-time applications and flexible in terms of compatibility with different 3D representations such as point clouds as well as scaling well with scene complexity. In addition, the explicit volumetric representation of the radiance field implied by the 3DGS model allows for modifications such as the addition of virtual content and relighting. In terms of disadvantages, 3DGS requires careful parameter tuning and initial setup for optimal performance and may struggle with the capture of fine details compared with photogrammetry.

Compared to NeRF techniques, 3DGS greatly reduces the rendering time and increases the quality of the reconstructed views. The basic framework of NeRF involves volumetric rendering and requires a ray marching technique where points within the volume along a ray are computed as queries to an MLP. Although this is a reasonably rapid process for each individual query, the fact that this needs to be performed many times per pixel produces slow rendering times. 3DGS avoids this problem by using an explicit structure of anisotropic Gaussians that can be rapidly projected onto the image plane and rendered quickly.

Several variants of the original 3DGS methodology [34] have been introduced to improve performance or reduce storage demands. These variants focus on the following specific aspects of the 3DGS pipeline. The list below is not intended to give the reader a

comprehensive list of state-of-the-art variations on 3DGS, but rather an overview of techniques used to mitigate issues common to 3DGS-based methods.

View Synthesis: Yu et al. [36] noted that the dilation (scaling) of Gaussians during the model instantiation phase depends on the resolution of the training images. This creates problems when zooming in or out as the reconstructed views can be at much higher or lower resolution than the original training images producing aliasing problems during viewpoint rendering. To mitigate this issue they introduced Mip-splatting [36], which replaced 2D dilation during viewpoint rendering with a 2D Mip filter to address aliasing and dilation artifacts when rendering viewpoints zoomed out relative to the training set. Qu et al. introduced Discontinuity-aware Gaussian Splatting (DisC-GS) [37] based on the observation that the continuous Gaussians used in 3DGS are not suitable to represent discontinuities and boundaries in the 3D scene. To each Gaussian, they added Bézier curves which are used by an indicator function during α-blending to determine if a pixel is affected by a Gaussian in the reconstructed viewpoint. This allows for only portions of each Gaussian to be projected onto the image plane during reconstruction and hence a more accurate representation of edges and discontinuities. With the goal of accurate surface reconstruction, Huang et al. [38] collapsed the 3D volume of each Gaussian into a set of 2D-oriented planar Gaussian disks and introduced a perspective-accurate 2D splatting process to support this representation.

3DGS Model Initialization: Yan et al. [39] noted that when rendering at low resolution or far away, the pixel size falls below the Nyquist frequency of the splatted Gaussians and causes aliasing. This also causes the rendering to be slowed as too many splats are rendered per pixel. They introduced Gaussians at multiple scales to represent the scene at different levels of zoom. Small finer-level Gaussians are aggregated to create larger Gaussians for coarser levels during training . To support better viewpoint rendering of scenes with edges and discontinuities, DisC-GS [37] introduced an additional attribute for each Gaussian, c_{curve} . c_{curve} describes control points of *M* Bézier curves in the coordinates of the two principal axes of each Gaussian. These control points are projected onto the image plane as the control points for the Bézier curves that pre-scissor the Gaussians during view synthesis. *M* is a hyperparameter defined by the user. A Bézier-boundary gradient approximation strategy is used to make the function continuous for training. Huang et al. [38] collapsed the 3D

volume of each Gaussian into a set of 2D oriented planar Gaussian disks to support better surface reconstruction during the view synthesis phase.

Densification and Training: In order to reduce the number of Gaussians produced during the training process, Light Gaussian [40] removed Gaussians with minimal visual impact by a pruning process which was followed by a recovery process to ensure smooth adaptation. GaussianPro [41] was motivated by the observation that for large-scale scenes, initial SfM point clouds do not represent textureless surfaces with sufficient points and hence do not always provide a good initial start for 3DGS algorithms. GaussianPro attempts to mitigate this issue by a modification to the densification process [41]. A progressive propagation strategy is implemented using priors of existing reconstructed scenes. Normal and depth values from neighbours are propagated to the current pixel in the synthesized viewpoint and a search is made for pixels whose propagated depth is very different from the rendered depth. Pixels with depth discrepancies are used to initialise new Gaussians by back-projecting propagated depths into 3D space to add new Gaussians to cover high error regions. Hanson et al. [42] noted that current pruning algorithms made use of heuristics that lose fidelity and foreground details at high compression ratios. The authors introduced PUP 3D-GS, making use of a new sensitivity pruning score computed as a second-order approximation of the reconstruction error on the training views. This work also proposed a multi-round pruning and refinement algorithm that can be applied to any pre-trained 3DGS model. Mip-splatting [36], added a 3D smoothing filter during model instantiation to constrain the size of 3D Gaussians and hence the maximum frequency to allow for better performance when rendering higher-resolution viewpoints. Lee et al. [43] propose a learnable mask strategy applied during training to reduce the number of Gaussians whilst still preserving reconstruction speed and quality. Instead of relying on spherical harmonics for colour presentation, Lee et al. [43] use a grid-based neural field to represent colour. The neural field is based on InstantNGP [21] and is optimized during training. Scaffold-GS [44] is motivated by the observation that during training, redundant Gaussians are created to try to fit every training view, making the model less robust to view changes, textureless areas and lighting effects. To mitigate this issue, Scaffold-GS [44] introduces anchor points to distribute 3D Gaussians and predict attributes on-the-fly based on viewing parameters. This is supplemented with anchor growing and pruning strategies based on Gaussian importance. The anchors provide for hierarchical and region-aware representation. Each anchor is associated with a set of neural Gaussians that learn together with learnable offsets from the anchor. Properties of the neural Gaussians are predicted by individual MLPs based on the

offsets, view angle and direction, and the properties of the anchors. Zhang et al. [45] noted that during densification, there is over-reconstruction where large regions are covered by a few very large Gaussians, leading to blur and artifacts during view synthesis. FreGS [45] was introduced to extract low- to high-frequency components with filters and use them for coarse-to-fine densification. The key idea is to minimise the difference between the frequency spectrum of the rendered image and ground truth. This is performed by a frequency annealing technique to go from low- to high-frequency detail.

Loss Function: As described above, GaussianPro [41] introduces new Gaussians during densification whose scales and rotations, represented as normals, are propagated from pixels in need of improved representation during view synthesis. To support this process, the orientations of the Gaussians are regularised by the propagated normals and by the introduction of a planar loss function into the training process. Huang et al. [38] incorporated two new regularization terms, depth distortion and normal consistency terms to enhance the quality of view synthesis. Morgenstern et al. [46] adds a smoothness loss to the loss function in order to improve the representation and subsequent coding on the Gaussians onto 2D grids. CompGS [47] adds the L1 norm of the opacity to the loss as a regularizer to encourage zero values for opacity. This in turn reduces the number of Gaussians and speeds up rendering time. To support their learnable mask strategy, Lee et al. [43] introduce a masking loss to the overall loss function during training. Scaffold-GS [44] introduces a volume regularization loss to encourage Gaussians to be small with minimal overlapping.

Table 2 describes the variants above in terms of the modifications they make to the various stages of the 3DGS architecture described in sections 3.3.1 and 3.3.2 above.

Method	**View synthesis**	**Model Instantiation**		
		3DGS Model Initialization	**Densification and Training**	**Loss Functions**
Light Gaussian [40]	-	-	Pruning	-
Mip-splatting [36]	2D Mip filter	-	3D Smoothing Filter	-
PUP 3D-GS [42]	-	-	Pruning	-

GaussianPro [41]	-	-	Progressive Propagation	Planar loss added
Yan et al. [39]	-	Multi-scale Gaussians	-	-
DisC-GS [37]	Bézier curve clipping	Bézier curve control points	-	-
Huang et al. [38]	Perspective-accurate 2D splatting	2D oriented planar Gaussians	-	Depth distortion and normal consistency losses added
Morgenstern et al. [46]	-	-	-	Smoothness loss added
CompGS [47]	-	-	-	Opacity loss added
Lee et al. [43]	-	InstantNGP [21] for colour	Learnable mask strategy	Masking loss added
Scaffold-GS [44]	-	-	Hierarchical representation and pruning	Volume loss added
FreGS [45]	-	-	Frequency-based annealing	-

Table 2 - Previous works for radiance field representation with 3DGS. "-" denotes that the variant makes use of prior art for this aspect of the 3DGS.

3.5.4 Model Coding

3DGS models did not originally include any specific coding techniques. Splat attributes were originally stored directly in the memory, resulting in storage requirement usually much higher than those for NeRF. For this reason, later works proposed different compression methods for 3DGS in order to enable practical applications.

In order to reduce the memory requirements for the spherical harmonics, Light Gaussian [40] used a distillation process supported by pseudo-view augmentation to compact the harmonics. This process was combined with vector quantization (VQ), which was used to adaptively encode the scale and rotation attributes into a codebook that further reduced storage requirements. The visual quality of the synthesized views was maintained by quantization-aware fine-tuning of the attributes. Morgenstern et al. [46] developed a coding and compression algorithm for 3DGS models involving the mapping of all of the attributes including the positions of the Gaussians onto a set of 2D grids. This is done during training by storing the attributes on 2D grids and then taking advantage of the unstructured nature of the 3DGS model, arranging the Gaussians with the exploitation of perceptual redundancies by a highly parallel sorting algorithm with enforcement of local smoothness. Since the uncompressed structure is the same as traditional 3DGS, this method can be integrated into existing systems. The authors compressed the RGB grid with lossy JPEG XL and all other attributes with lossless JPEG XL. CompGS by Navaneet et al. [47] developed a VQ method to quantize the Gaussian parameters while optimizing them. The method does not quantize opacity or positions. Opacity is made use of for reducing the number of Gaussians later in the process, while the quantization of the spatial positions is avoided to not degrade the structure of the 3DGS model. Lee et al. [43] also make use of VQ to compactly represent the scale and rotation of each Gaussian, noting that these attributes tend to have little variation with the 3DGS model.

Other methods were based on the neural gaussians proposed by Scaffold-GS, encoding anchor positions and their respective attributes instead of directly compressing the splats. HAC [48] proposed the use of a multi-resolution binary grid to assist in the encoding process. The method employed a feature vector obtained from interpolating the position of the anchor in the hash grid to estimate the entropy of the anchor attributes and entropy code them. ContextGS [49] divided the anchors into disjoint sets each corresponding to a different level of detail. Each level was encoded sequentually and the already encoded attributes were leveraged to model the entropy of the later levels in an autoregressive coding architecture.

4. Challenges

The use of radiance fields for the representation of visual media faces a number of challenges that must be considered in order to achieve widespread adoption. This section

describes and categorizes the main challenges related to the representation and the implementation of the models, as well as their quality assessment.

4.1 Representation Challenges

Diversity of Representation: The large number of techniques proposed in the state of the art for the representation of radiance fields, including several variants of NeRF and 3DGS, demonstrated how high-quality novel view synthesis can be obtained using many different algorithms. This abundant research activity might pose a problem for interoperability between encoding and decoding devices if the employed representation for the radiance field is constantly changing. This issue may also be a problem for coding standards, which may need to select a radiance field representation method that is prone to being replaced in the future. For instance, not only will a coding standard targeting a NeRF method not be compatible with 3DGS, but it may also not be compatible with other NeRF methods. For this reason, the selected coding mechanism should accept an input representation that will stay relevant for the industry for a reasonable amount of time to enable widespread adoption.

Representation of Challenging Real-World Scenes: While radiance field techniques have demonstrated impressive performance for novel view synthesis under controlled conditions, the performance naturally decreases in the presence of features in the input images such as non-opaque surfaces or light perturbations. Despite the high performance of 3DGS techniques, there are still areas for improvement in terms of the representation of challenging real-world scenes. The initial framework for the positions of the Gaussians is provided by SfM techniques and these techniques are known to produce poor representations of textureless areas resulting in large Gaussians that produce blur and artifacts in the reconstructed images.

Aliasing and Scaling Issues: Both NeRF and 3DGS methods are trained on the specific scale of the ground truth images used to capture the scene and may not perform as well when rendering the scene at a higher or lower resolution than the training images. In the case of 3DGS, the scale of each Gaussian is set during the training process and is related to the scale at which the training images are captured. However, when view synthesis occurs at distances substantially closer or further away from the scene than the training images, artifacts can occur due to the mismatch between the pixel size in the reconstructed image and the scale of the splatted Gaussians.

4.2 Implementation Challenges

Large Memory Footprint: The high capacity of accurately representing fine detail with radiance fields is partly due to the large amount of parameters of the obtained model. While NeRF models may rely on large networks with several parameters, this problem is particularly observed for 3DGS models. The explicit storage of the Gaussians can consume a large amount of memory, particularly for large scenes. Even for small scenes, the storage of the millions of Gaussians needed to represent the scene can be prohibitive for implementation on some devices. Each Gaussian is described by a large number of parameters further increasing the memory requirements. The storage problem is exacerbated by the densification process that splits or clones new Gaussians to better represent texture. The process of densification can produce excessive redundant Gaussians that increase memory requirements and produce artifacts during view synthesis. This results in a large memory footprint which is associated with high storage cost. Reducing the size of radiance field models is one of the main challenges to be addressed by a coding standard.

Time-Consuming Model Instantiation: The process required to derive a radiance field model from a set of input viewpoints usually includes the optimization of parameters with stochastic gradient descent or similar algorithms. While great progress has been made in recent years to reduce the complexity of this process, the model instantiation is still time-consuming and several applications would benefit from further improvement of this issue.

4.3 Quality Assessment Challenges

Lack of Defined Subjective Evaluation Protocols: The subjective evaluation of visual quality is an essential part of the activity of standardisation groups interested in the creation of coding standards. However, subjective evaluation of radiance fields involves steps not required for regular images, such as defining camera positions and intrinsic parameters. Despite some recent publications [50], the definition of best practices for the subjective evaluation of radiance fields is still an open problem.

Lack of Objective Metrics for Quality Evaluation: Since subjective quality assessment experiments can be expensive and time-consuming, objective quality metrics can be used to estimate the visual quality of coding methods for radiance fields. While image-based

metrics are usually employed for this purpose, there is little evidence regarding which metrics better correlate with the human visual system for the evaluation of radiance fields. Further work is required to determine which image-based metrics are best suited for this task, as well as to explore the possibility of evaluating radiance fields directly in the 3D domain.

5. Use Cases

Several use cases have been proposed to leverage radiance field techniques such as NeRF and 3DGS for immersive visual imaging. This section lists and describes major use cases for radiance fields.

5.1 Extended Reality (XR)

In Extended Reality (XR), a user sees not only virtual content, but also the environment around the user in real time composited with a virtual environment or virtual objects. Radiance field technologies have applications in this use case to allow for scanned real content from one context to be transplanted into another context seamlessly. Since both the environment and the composited content are scans of real objects, there will be a need for a flexible representation of the content to allow for the correct representation of light material appearance and shadow.

This use case can be associated with the following requirements:

High Visual Quality: In order to allow for natural interactions, the encoded radiance fields must be able to represent the composited content in high detail with sufficient accuracy in material appearance to allow for relighting or shadow production consistent with the target environment.

5.2 Autonomous Driving

In the autonomous driving use case, multiple vehicle cameras capture the surrounding environment, providing a 3D scene reconstruction for perception and computer vision tasks. Radiance field technologies that have application to this use case include perception, 3D reconstruction, simultaneous localization and mapping (SLAM), and simulation. SLAM allows autonomous vehicles to simultaneously build a map and localize within it. SLAM

algorithms enable vehicles to map unknown environments, aiding in tasks like path planning and obstacle avoidance. Radiance fields can enhance relocation and mapping for accurate pose estimation.

Radiance field reconstruction in autonomous driving simulations provides a safer, cost-effective alternative to real-world testing. This technology can be used to create realistic virtual environments for sensor data generation, facilitating diverse driving scenarios and reducing safety risks. The approach enhances realism and reduces manual effort in scene creation and editing, bridging the gap between real and virtual worlds.

This use case can be associated with the following requirements:

Metrological Accuracy: Since accurate distance measurement is crucial to safety at driving speeds. An accurate preservation of distances between elements of the geometry of the environment around the vehicle (either in situ or in simulation) must be ensured by the employed radiance field method in order to enable this use case.

5.3 AEC (Architecture, Engineering and Construction)

Radiance fields can be used to support the display of 3D content for visualisation of architecture and infrastructure in the context of a building site or surrounding area. An example would be to visualize planned construction elements against already completed construction elements or visualize a planned construction in the context of existing surroundings. The content may be generated artificially by CAD design software or created through the scanning of physical objects with 3D scanning equipment or may be a merger of both sources of data. The resolution of the content will be highly dependent on the type of object and industry. Geometrically, radiance fields created using CAD software are likely to be arranged on regular grids and patterns, while data collected using 3D scanning equipment is likely to be arranged in an irregular geometric pattern. There may be examples of this use case that require 3D processing and computer vision for the matching of scanned data from real objects with the computer generated elements for verification of physical elements against a plan without human intervention.

This use case can be associated with the following requirements:

Support for Large-Scale Scenes: The radiance fields used in this application must be able to represent large scenes such as a building or construction site within a single encoded file.

Random Access: The radiance field representing a large area must be able to be only partially decoded to represent only one spatial region of the entire model.

Metrological Accuracy: Since accurate distance measurement is crucial to engineering and architectural applications, an accurate preservation of distances between elements of the geometry of the scene must be ensured by the employed radiance field method in order to enable this use case.

5.4 Medical Imaging

In this application, radiance fields can assist medical practitioners with such tasks as visualising body structures, compositing artificial content (such as indicators of medically significant elements) onto the real world or allowing for remote diagnosis and treatment. An example of this would be the creation of a real-time radiance field of a patient to allow a doctor to remotely operate. In this scenario, the continuous representation of the scene allows the doctor to move freely without being restricted to the particular viewpoints of the cameras used to capture the scene. The ability to represent complex material appearance allows the doctor to recognise tissues and membranes more accurately as transparency and gloss are accurately represented.

This use case can be associated with the following requirements:

High Visual Quality: In order to allow for natural interactions, the encoded radiance fields must be able to represent the composited content in high detail with sufficient accuracy in material appearance to allow for the recognition of subtle differences in appearance of anatomical structures.

Metrological Accuracy: To allow for the precise placement of instruments and medical intervention, accurate distance measurement is crucial. An accurate preservation of distances between elements of the geometry of the scene must be ensured by the employed radiance field method in order to enable this use case.

5.5 Industrial Imaging

Several industrial applications have been already employing 3D-based visualisation with the concept of digital twins, which are virtual instances of physical assets meant to track its behavior during its lifetime. While traditionally digital twins employed point clouds or meshes

for their visual representation, radiance fields may be able to provide better visual quality and photorrealism and thus have the potential of being leveraged in industrial environments.

This use case can be associated with the following requirements:

Metrological Accuracy: An accurate representation of the geometry of industrial assets must be ensured by the employed radiance field method in order to enable this use case.

5.6 Virtual Communication

Communication systems allowing for immersive multimedia experience may employ radiance fields for the representation of users in video communication. In recent years, researchers have proposed methods to produce 3DGS avatars representing faces from a set of images, which can then be animated with facial expressions captured from a video. This technology can be employed for communication by storing the avatars in the endpoints and animating them with expressions captured and transmitted in real-time. An example of this use case can be observed in Figure 11.

This use case can be associated with the following requirements:

High Visual Quality: In order to allow for natural interactions, the encoded radiance fields must be able to maintain the facial features of the avatars with high detail.

Figure 11: 3DGS avatars (bottom) incorporating facial expressions extracted from videos (top) rendered in a mobile device. Source: [51]

5.7 Scientific Modeling

3D-based modeling for scientific purposes already employ point clouds and meshes for data visualisation and physical simulations. Visual data acquired directly from the real world with radiance fields have the potential to be used in this field, as already demonstrated by several recent research works. Figure 12 illustrates an example of this use case in a fluid simulator embedding objects modeled with 3DGS.

This use case can be associated with the following requirements:

Metrological Accuracy: In order to allow reliable modeling of the physical properties of obejcts, the corresponding dimensions must be accurate. The requirements depend heavily on the application and on the scale of the objects being modeles, since the represented dimensions may differ orders of magnitude if applied for chemistry, fluid dynamics, or astronomy.

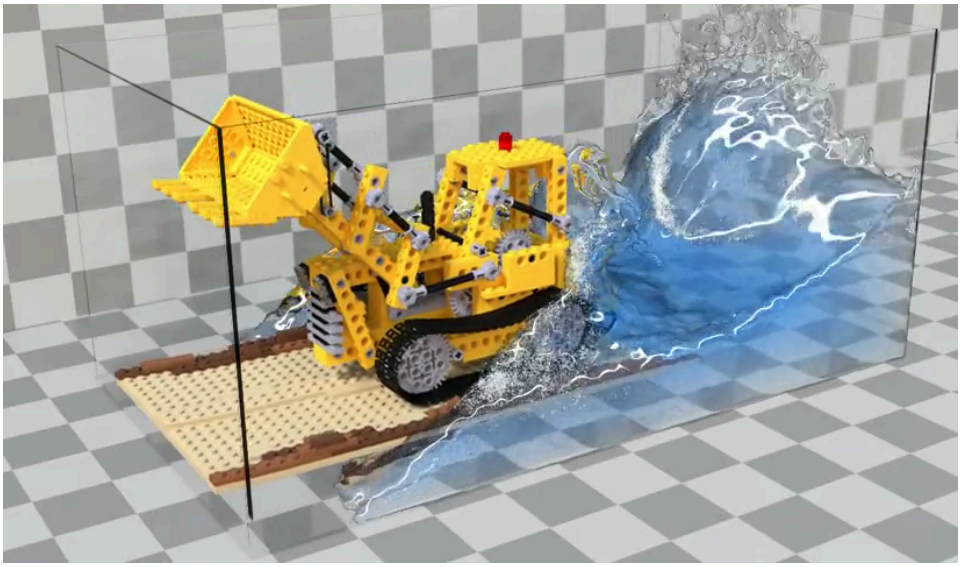

Figure 12: 3DGS model employed inside a fluid dynamics simulation. Source: [52]

5.8 GIS (Geographic Information Systems)

Geographic information systems (GIS) model geographic information creating maps that can be integrated with other types of data. Part of the related information may be represented in the form of radiance fields, allowing for free-viewpoint visualization of wide areas from radiance field models obtained from both ground and aerial images. Figure 13 depicts an example where 3DGS models were used to model a visual representation of a geographical location inside a map.

Figure 13: visualization of a 3DGS model representation within GIS. Source: [53]

This use case can be associated with the following requirements:

Support for Large-Scale Scenes: The radiance fields used in GIS must be able to represent large scenes within a single encoded file.

High Compression ratio: Due to the large extent of the area often represented by GIS, the representation of the bitstream associated with the radiance fields must be as compact as possible in order to enable the representation of large areas.

Random Access: The radiance field representing a large area must be able to be only partially decoded to represent only one spatial region of the entire model.

Progressive Decoding: When rendering large scenes, progressive decoding of the encoded representation must be possible, allowing for the extraction a version of the model without fine detail requiring less memory for its representation.

6. JPEG RF Scope

The scope of the JPEG RF activity is the creation of a **coding standard** for radiance fields, offering a **compact representation** for models enabling the **immersive depiction of 3D scenes** given a set of input images from different viewpoints. This standard shall accommodate modern radiance field generation and rendering techniques, efficiently coding models with high compression performance while minimizing the impact on the visual quality of the view synthesis process, with the goal of supporting a royalty-free baseline.

This activity shall specify a normative decoder capable of converting a codestream with normative syntax and semantics into a set of radiance field parameters from which images depicting a scene from a query camera pose can be synthesized. The remaining processes of the processing pipeline, namely the camera pose estimation, model instantiation, encoder, and view synthesis, may be informatively included in the standard but shall not be a normative component.

References

[1] Eltner, Anette, and Sofia, Giulia. "Structure from motion photogrammetric technique." Developments in Earth surface processes. Vol. 23. Elsevier, 2020. 1-24.

[2] Seitz, Steven et al. “A comparison and evaluation of multi-view stereo reconstruction algorithms.” In IEEE Conference on Computer Vision and Pattern Recognition, pages 519–528, 2006.

[3] Strecha, C. et al. “On benchmarking camera calibration and multi-view stereo for high resolution imagery.” In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, June 2008.

[4] Agarwal, Sameer et al. “Building Rome in a day.” In IEEE International Conference on Computer Vision, 2009.

[5] Schönberger, Johannes et al. "Pixelwise view selection for unstructured multi-view stereo", In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, pages 501–518. Springer, 2016.

[6] https://factumfoundation.org/technology/3d-digitisation/close-range-photogrammetry/, viewed on December 2024

[7] Harris, C. and Stephens, M. “A combined corner and edge detector.” In Alvey Vision Conference, pages 147–151, 1988.

[8] Rosten, Tom. “Machine learning for high-speed corner detection.” In European Conference on Computer Vision, pages 430–443. IEEE, 2006.

[9] Leutenegger, S. et al. “Brisk: Binary robust invariant scalable keypoints.” In IEEE International Conference on Computer Vision, pages 2548–2555, Nov 2011.

[10] Bay, Herbert et al. “Speeded-up robust features (surf).” Computer Vision and Image Understanding, 110(3):346–359, June 2008.

[11] Zhang, Ruo et al. "Shape-from-shading: a survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 8, pp. 690-706, Aug. 1999, doi: 10.1109/34.784284.

[12] Fan, Jiacheng et al. "Solving Shape-From-Shading problem through shape and depth joint optimization", Optik, Volume 270, 2022, ISSN 0030-4026, doi:10.1016/j.ijleo.2022.170009.

[13] https://www.linkedin.com/pulse/structure-from-motion-manish-joshi/, viewed on December 2024

[14] Binh Do, P. N. et al. "A Review of Stereo-Photogrammetry Method for 3-D Reconstruction in Computer Vision," 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, 2019, pp. 138-143, doi: 10.1109/ISCIT.2019.8905144.

[15] https://www.pix4d.com/blog/raycloud-power-understanding-photogrammetry/, viewed on December 2024

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

[17] Barron, Jonathan T., et al. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[18] Reiser, Christian, et al. "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[19] Zhang, Kai, et al. "Nerf++: Analyzing and improving neural radiance fields." arXiv preprint arXiv:2010.07492 (2020).

[20] Liu, Lingjie, et al. "Neural sparse voxel fields." Advances in Neural Information Processing Systems 33 (2020): 15651-15663.

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

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

[23] Yu, Alex, et al. "Plenoctrees for real-time rendering of neural radiance fields." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[24] Fridovich-Keil, Sara, et al. "Plenoxels: Radiance fields without neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[25] Chen, Anpei, et al. "Tensorf: Tensorial radiance fields." European conference on computer vision. Cham: Springer Nature Switzerland, 2022.

[26] Bird, Thomas, et al. "3d scene compression through entropy penalized neural representation functions." 2021 Picture Coding Symposium (PCS). IEEE, 2021.

[27] Tang, Jiaxiang, et al. "Compressible-composable nerf via rank-residual decomposition." Advances in Neural Information Processing Systems 35 (2022): 14798-14809.

[28] Rho, Daniel, et al. "Masked wavelet representation for compact neural radiance fields." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[29] Li, Sicheng, et al. "NeRFCodec: Neural feature compression meets neural radiance fields for memory-efficient scene representation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[30] Takikawa, Towaki, et al. "Variable bitrate neural fields." ACM SIGGRAPH 2022 Conference Proceedings. 2022.

[31] Girish, Sharath, Abhinav Shrivastava, and Kamal Gupta. "Shacira: Scalable hash-grid compression for implicit neural representations." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

[32] Chen, Yihang, et al. "How far can we compress instant-ngp-based nerf?." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[33] Shin, Seungjoo, and Jaesik Park. "Binary radiance fields." Advances in neural information processing systems 36 (2023): 55919-55931.

[34] Kerbl, Bernhard, et al. "3d gaussian splatting for real-time radiance field rendering." ACM Trans. Graph. 42.4 (2023): 139-1.

[35] Remondino, Fabio, et al. "State of the art in high density image matching." The photogrammetric record 29.146 (2014): 144-166.

[36] Yu, Zehao, et al. "Mip-splatting: Alias-free 3d gaussian splatting." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[37] Qu, Haoxuan, et al. "DisC-GS: Discontinuity-aware Gaussian Splatting." arXiv preprint arXiv:2405.15196 (2024).

[38] Huang, Binbin, et al. "2d gaussian splatting for geometrically accurate radiance fields." ACM SIGGRAPH 2024 conference papers. 2024.

[39] Yan, Zhiwen, et al. "Multi-scale 3d gaussian splatting for anti-aliased rendering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[40] Fan, Zhiwen, et al. "Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps." arXiv preprint arXiv:2311.17245 (2023).

[41] Cheng, Kai, et al. "Gaussianpro: 3d gaussian splatting with progressive propagation." Forty-first International Conference on Machine Learning. 2024.

[42] Hanson, Alex, et al. "PUP 3D-GS: Principled Uncertainty Pruning for 3D Gaussian Splatting." arXiv preprint arXiv:2406.10219 (2024).

[43] Lee, Joo Chan, et al. "Compact 3d gaussian representation for radiance field." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[44] Lu, Tao, et al. "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[45] Zhang, Jiahui, et al. "Fregs: 3d gaussian splatting with progressive frequency regularization." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[46] Morgenstern, Wieland, et al. "Compact 3d scene representation via self-organizing gaussian grids." European Conference on Computer Vision. Springer, Cham, 2025.

[47] Navaneet, K. L., et al. "Compgs: Smaller and faster gaussian splatting with vector quantization." European Conference on Computer Vision. Springer, Cham, 2025.

[48] Chen, Yihang, et al. "Hac: Hash-grid assisted context for 3d gaussian splatting compression." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.

[49] Wang, Yufei, et al. "Contextgs: Compact 3d gaussian splatting with anchor level context model." Advances in neural information processing systems 37 (2024): 51532-51551.

[50] Martin, Pedro, et al. "NeRF View Synthesis: Subjective Quality Assessment and Objective Metrics Evaluation." arXiv preprint arXiv:2405.20078 (2024).

[51]https://www.qualcomm.com/developer/blog/2024/12/driving-photorealistic03d-avatars-in-real-time-on-device-3d-gaussian-splatting, viewed on April 2025.

[52] Feng, Yutao, et al. "Gaussian splashing: Dynamic fluid synthesis with gaussian splatting." arXiv e-prints (2024): arXiv-2401.

[53] https://dev.to/gisbox/efficient-rendering-gaussian-splatting-to-3dtiles-with-gisbox-2m49, viewed on April 2025