3-D Reconstruction Of Scenes from 2-D Images

Abdul Basit
Univeristy of Evry Paris Saclay
Paris, France
basitmal36@gmail.com

Supervisor: Professor Dr Hedi Tabia
University of Evry Paris Saclay
Paris, France

***Abstract*— This paper presents a comprehensive review of recent advances in 3D scene reconstruction using only 2D images. While traditional approaches predominantly rely on geometric principles, such as multi-view stereo and structure-from-motion, they often require dense image inputs and struggle with occlusions, textureless regions, and viewpoint sparsity. Modern techniques, driven by deep learning, offer a paradigm shift by directly learning 3D structure from 2D observations through optimization or generative modeling. Among these, Neural Radiance Fields (NeRF), Gaussian Splatting, and SparseFusion have emerged as prominent frameworks, each introducing unique mechanisms for representing, rendering, and learning 3D scenes. These methods leverage neural networks, volumetric rendering, and generative priors to achieve high-fidelity reconstructions from minimal input data. This review highlights their architectural foundations, learning strategies, and limitations, providing a consolidated understanding of the current state of the art. The paper aims to serve as a foundation for future research in neural 3D reconstruction and its applications in computer vision, graphics, robotics, and immersive technologies.**

Keywords—NeRFs (Neural Radiance Fields)

I. Introduction

3-D reconstruction of a scene or object can be done by using input data as 2-D images from different viewpoints, video, point clouds and LiDAR, etc. Traditionally, 3-D reconstruction has been done through Active or Passive methods. Active methods mostly rely on actively interacting with the object or scene by projecting some form of energy onto it and sensing how that energy has been modified such as depth cameras, laser scanners that beam infrared light to capture depth information in real time. These methods can be costly, difficult to calibrate the sensing system, and with limited range. Alternatively, passive methods do not actively interact with the object. They use the light that is already present in the object or scene, such as using 2-D images of the scene.

Machine learning is playing a crucial role in revolutionizing 3-D reconstruction. In past identifying features (corner, shapes, texture) across multiple images was a time consuming task but now machine learning especially deep neural networks can automatically identify and match features even in the complex scene like view dependent images etc. Machine learning models can also be trained to handle noise in the data. They can fill the gaps, reconstruct more accurate and complete 3-D scenes. Unlike traditional methods, Machine learning models can handle unfamiliar data and scenes more accurately.

II. Scope of this paper

There are different methods for 3-D reconstruction of data from different types of input data such as images, point clouds, depth sensors etc. But in this paper we will be focusing on 3-D reconstruction of scene or an object from 2-D images (passive method) taken from different viewing directions. In this case the availability of the data will not be an issue, compared to other methods like getting data from depth sensors like (LiDAR etc) this is more affordable, and also image capture the color or texture of an image which can be directly mapped onto the 3-D model resulting in visually appealing and more realistic results.

On the other hand, processing the images and reconstructing a 3-D model from them can be computationally very intensive, especially for high resolution images or large datasets, the accuracy of 3-D model will be dependent on the number of images, number of overlapping images, quality of images and lightning conditions. Changing light can be very sensitive in some cases.

III. Neural Radiance Fields

Neural Radiance Fields, also known as NeRF, is a research project completed with the collaboration of researchers from Google and the University of California, Berkeley. In the name NeRF, neural represents a neural network that learns how light behaves in a 3-D scene by mapping spatial positions and viewing direction into color and density, radiance means how much light is emitted by a point in space in each direction and field represents a continuous function over time.

NeRF has a very simple architecture, as simple as a multilayer perceptron (MLP). NeRF as an input takes a 5-Dimensional vector having spatial location (x, y, z) and viewing direction (θ, Φ) and outputs a color (r, g, b), meaning how much light is emitted by radiance and density (σ), where density represent is there any object in that point or not, if there is then how dense it is. 0 if there is nothing and infinity for an object. NeRF has 9 Fully-connected non-convolutional layers, each having 256 channels. So, weights are going to define the subset of the whole world of NeRF, we will optimize the weights on the given images to explain the way we see the world.

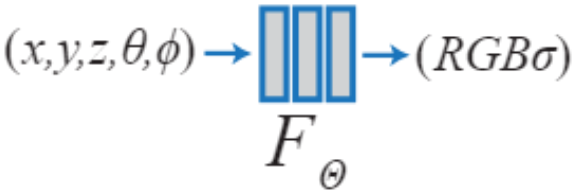

Figure 1: This figure shows the simple MLP used in NeRFs with input and output parameters.

We generate views using **volume rendering**, which is a way to create a 2D image that shows what this 3-D object would look like if you look at it from a certain angle. Through volume rendering, we know how light interacts with the material inside the space. It works like a ray tracer. We shoot rays at each pixel from the camera and drop points evenly along the ray, and each of those point (spatial location and viewing direction) is an input vector to the neural network and get output as an emitted radiance and density.

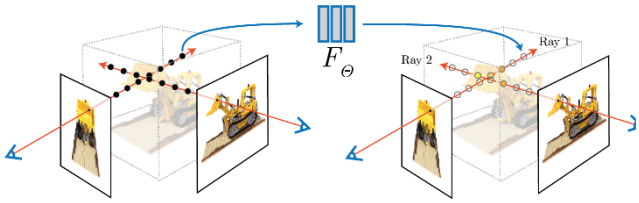

Figure 2: This figure shows the ray marching phenomenon used in NeRFs for volume rendering.

So, now we take the ray and march from back to front or front to back until we reach the back of the camera. While marching back, we alpha-compose (If something is opaque, it will get rid of the stuff behind it, it is actually the modeling of how semi-transparent stuff interacts with each other in a physical way) densities on top of each other and at the end we get radiance. The color of a pixel is a weighted combination of colors in the world and alpha values

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

colors

weights

The one issue is that we have high radiance fields defined by neural network and it is hard to drop sample along all the rays to get good rendering, So we do it by evenly dropping point along the rays which give us the information about where the stuff is and then we lay down more points based on the information from point dropped before. In this case, the (Weight and alpha) acts as a probability distribution for new sample points. This is known as Two-pass rendering.

The way we train this model is that we iterate through pixels, we shoot a ray, we render that array according to our current randomly initialize neural network and then we apply gradient descent and minimize the Mean square error between render pixel value and the observed pixel value and through that we recover the 3-D scene.

Another issue is that while learning a mapping from (x, y) which are the coordinates of an image to (rgb) values which are the color of pixel the neural network is unable to memorize the image and it is unable to reconstruct the image that it has fully observed even if you train it for days.

The solution to this is that we take the coordinates (x, y, z) and just push them through a bunch of sinusoids of increasing frequency and expand the feature vector to a high-dimensional space with a lot of these high-frequency sinusoids that are concatenated and then fed back into the neural network.

NeRF can generate **realistic novel views** with fine details and accurate lighting, even from a sparse set of input images. NeRF **do not suffer from fixed resolution limitations**, allowing for smooth scaling and rendering. NeRF model **view-dependent effects** like specular reflections and transparency, making them more physically accurate. But on the other hand NeRF is very data hungry. It cannot generalize well on limited data.

IV. Instant Neural Graphics Primitives with Multi-resolution Hash Encoding

Neural graphics primitives, parameterized by fully connected neural networks, can be computationally expensive to train and evaluate. This paper introduces a novel input encoding method that significantly reduces the computational cost without sacrificing quality. The approach integrates a small neural network with a multiresolution hash table of trainable feature vectors, optimized through stochastic gradient descent. This structure helps resolve hash collisions and enables efficient parallelization on modern GPUs. By implementing fully-fused CUDA kernels, the system minimizes memory bandwidth usage and computation overhead, resulting in a substantial speedup. The proposed method allows training high-quality neural graphics primitives in seconds and rendering in milliseconds at high resolutions.

The technique mentioned in paper [1] for encoding is not hardware friendly because cosine and sine are complex math operation and it also does not able to reflect sharp details. The MLP used in [1] is large that makes it computationally expensive. To overcome all these issues the researchers at NVIDIA proposed this paper. They used Hash grid encoding where trainable features vectors are stored in multi-resolution hash table.

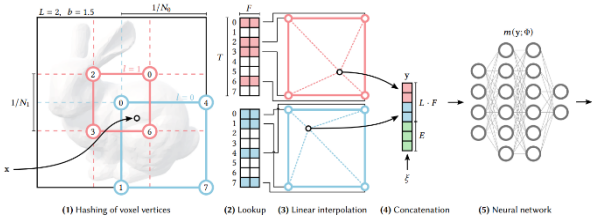

Figure 3: This figure shows the hash grid encoding used in the mentioned paper before sending it into MLP.

The authors added grid structure to 16 different levels of resolutions. In this technique we only look at grid that contains X and hash its vertices that contain input X. Hash function take the vertices (the corner locations) i.e 1,4,0,6. We look-up to corresponding learnable features vectors from hash table and linear interpolate to get the actual feature while considering only the grids that contains X. Then we concatenate all the feature of input X at different resolutions while also concatenating the parameters. After that these features are passed to small fully connected Neural Network. The NN predicts the density and RGB color at given 3-D point from the given viewing direction. These outputs are then used for volume rendering similar to NeRF [1]. The

authors used fully fused cuda kernels where data does not leavae GPU until final output.

V. PixelNeRF: Neural Radiance Field from one or few images

PixelNeRF is a neural rendering approach that improves upon standard NeRF by enabling **single-image 3D reconstruction and novel view synthesis** without requiring per-scene optimization. Unlike traditional NeRF, which needs multiple images and extensive training per scene, PixelNeRF learns a **scene-independent prior** from a large dataset of images. It extracts **pixel-aligned image features** using a convolutional neural network (CNN) and conditions the NeRF model on these features, allowing it to generate accurate 3D representations from just one or a few input images. This makes PixelNeRF significantly more **data-efficient and generalizable**

The techniques mentioned in paper [1] and [2] require Requires **multiple views** of a scene for high-quality results of 3-D reconstruction. They also do not utilize prior common knowledge across scenes. But pixel NeRF learn pixels level information from the data. The author used Convolutinal Neural Network is used as encoder and to extract image features that will be used to infer 3-D structure. The CNN-based encoder learns a **scene-independent prior** from a large dataset. Instead of training a new model for every scene, this prior enables PixelNeRF to generalize across scenes, allowing it to infer 3D information from a single or a few images without retraining.

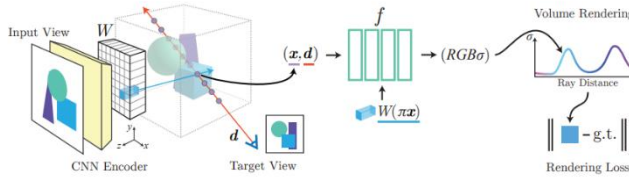

Figure 4: This figure PixelNeRF's architecture and volume rendering process.

The CNN architecture that authors used in ResNet34, that uses feature pyramid to get pixels aligned by up sampling it to the view space. The features extracted and 3-D coordinates are guided to the NeRF model that uses volume rendering for 3-D reconstruction of scenes. The reconstructed 3D scene is compared with ground truth, and the model is trained via gradient descent. Loss functions such as pixel-wise differences are used to optimize the model by ajusting the weights in both the CNN feature extractor and the NeRF model itself.

VI. Neuralangelo: High-Fidelity Neural Surface Reconstruction

Neuralangelo is a neural surface reconstruction method that uses multi-resolution 3D hash grids and neural surface rendering to achieve high-fidelity 3D reconstructions from multi-view images. It employs numerical gradients for smoothing and a coarse-to-fine optimization strategy, enabling the reconstruction of intricate details from 2D video inputs. It samples 3-D points along camera viewing direction and use multiresolution hash encoding to encode the position. The encoded features are input to an SDF MLP and a color MLP.

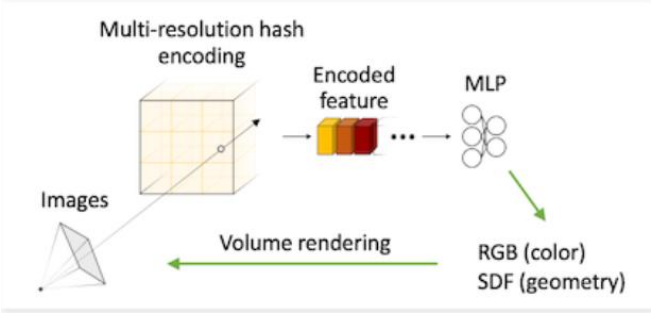

Figure 5: This figure shows the architecture and scene generation process implemented by the Neural Angelo paper.

It is an orthogonal distance of given point x to the boundary of set S, with sign determined by whether or not x is inside or outside of S. It is surface can be represented as zero-level set F(x)=S{x∈R3|f(x)=0}
Given 3D point and SDF value, density value at x is calculated. This let us compute rendering loss. MLPs & feature stored in hash entries are trained jointly

L=Lrgb+Leik+Lcurv

- Lrgb: loss between input and synthesized image
- Leik: Eikonal loss regularize underlying SDF such that the surface normals are unit-norm.
- Curvature loss: regularize underlying SDF such that means curvature is not arbitrary large.

SDF is initialized as approximately a sphere with this initial shape high curvature loss makes concave shapes difficult to form because it prevents topology. Thus instead of applying L-curvfrom beginning of optimisation process, a short warm up period is used that linearly increase the curvature loss weight.

They are features into hash entries, and cell corner is a hash entry. Given input position xi, it maps to correspond grid resolution. Each grid resolution of the feature vector is obtained by linear interpolation of hash entries at corners. Feature vector is consisting of concatenated features from different resolution grids. Then feature vector is passed to shallow MLP. But analytical gradient is not good so authors proposed numerical gradient for continuous space. But it requires to compute 6 more points given xi along x,y,z axes. It allows to simultaneously update adjacent gradient cell given xi, thus become some version of analytical gradient. Use of numerical gradient distributes the back propagation update beyond the local hash grids. For optimization we do progressive activation of grids, finer hash grids are progressively activated when ϵ (step size) decrease to their spatial size. Neuralangelo employs numerical gradients instead of analytical ones to enhance surface reconstruction stability and detail preservation. Analytical gradients often introduce noise, instability, and artifacts, especially in fine details and higher-order derivatives. In contrast, numerical gradients act as a smoothing operation, refining surface

details and ensuring more consistent optimization. This approach, combined with multi-resolution hash grids, allows Neuralangelo to achieve high-fidelity 3D reconstructions with improved accuracy and stability.

VII. SPARSE FUSION

SparseFusion is a_novel 3D reconstruction technique,designed to_work effectively with sparse multi-view images. Introduced in 2023 by researchers from Alibaba DAMO Academy, SparseFusion addresses the limitations of Neural Radiance Fields (NeRF) when the number of available input images is very low. Traditional NeRF models require dense image coverage from many viewpoints to produce accurate 3D reconstructions. In contrast, SparseFusion achieves high-fidelity 3D geometry generation using as few as two to six images. This is made possible by combining sparse-view 3D learning with the guidance of powerful diffusion models. The central innovation of SparseFusion lies in leveraging Score Distillation Sampling (SDS) from a pre-trained text-to-image diffusion model, such as Stable Diffusion, which provides semantic and visual priors to supervise the geometry generation, even in the absence of dense view coverage.

SparseFusion architecture consists of three main components integrated into a cohesive 3D reconstruction_pipeline. It begins with a sparse-view geometry encoder that takes a few posed_RGB images of the target object and constructs a tri-plane feature_representation. The input views are first processed by an encoder network which_extracts spatial features and converts them into three orthogonal feature planes (XY, YZ, and XZ). This tri-plane representation efficiently captures volumetric information in a format that can be decoded into 3D shapes and novel views.

The second core part of the architecture introduces Score Distillation Sampling, where a pre-trained diffusion model is used as a source of supervision. The rendered images from the tri-plane representation are fed into the diffusion model, and its gradient feedback is used to refine the 3D geometry. This process allows SparseFusion to hallucinate realistic and semantically meaningful details that are missing or occluded in the sparse views. The third component is a small decoder or renderer that synthesizes new views of the 3D object from arbitrary camera angles, using the tri-plane as a conditioning representation. The entire architecture is trained in an end-to-end fashion, enabling robust and generalizable 3D reconstructions.

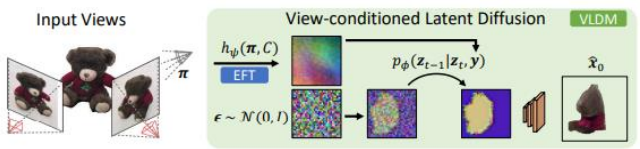

Figure 6: This figure shows the overall architecture and scene generation process used by Sparse Fusion. A view-conditioned latent diffusion model (VLDM) and a diffusion distillation process that optimizes an Instant NGP and used VLDM to model $p(\boldsymbol{x}j\boldsymbol{\pi}; C)$

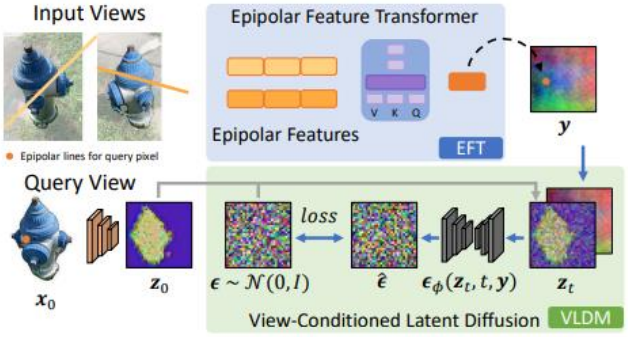

Figure 7: This shows the diagram of view-conditioned latent diffusion model. VLDM is conditioned on features $\boldsymbol{y}$, which is predicted by EFT..

SparseFusion introduces several key innovations that set it apart from earlier approaches such as NeRF. The most significant novelty is the use of Score Distillation Sampling, which replaces traditional photometric loss with semantic supervision from a diffusion model. Instead of matching pixels between observed and rendered views, SparseFusion optimizes 3D geometry by aligning the output with the learned priors of a text-to-image diffusion model. This enables the network to reconstruct plausible 3D shapes even with highly limited visual evidence.
Another innovation is the use of tri-plane representation, which condenses 3D features into three 2D planes, offering a memory-efficient and differentiable structure for volumetric modeling. This representation reduces computational complexity while retaining high-quality spatial detail. SparseFusion also supports both image and text conditioning, allowing it to operate in zero-shot scenarios and perform 3D generation from textual descriptions. This flexibility and efficiency make it suitable for both sparse-view reconstruction and generative modeling, setting a new benchmark in the field of neural 3D rendering.

The training of SparseFusion relies on iterative optimization using gradient descent. The main supervisory signal comes from the Score Distillation Sampling loss, which is computed by passing rendered images through a frozen diffusion model and extracting the guidance signal from its internal representations. Unlike traditional losses that depend on pixel-wise comparison, SDS operates at a semantic level, comparing the structure and realism of rendered images against the expectations of the diffusion model. This enables the model to infer occluded regions and complete missing details in the geometry.
Training proceeds by rendering novel views from the current tri-plane geometry, feeding them into the diffusion model, and computing the gradient of the SDS loss. These gradients are then backpropagated to refine the tri-plane representation. In addition to the SDS loss, auxiliary losses are used to enforce view consistency across the sparse inputs and to regularize the geometry for stability. The combination of semantic supervision and sparse view-based optimization

allows SparseFusion to converge efficiently, even when only a handful of input images are available.

SparseFusion marks a significant leap forward in the domain of neural 3D reconstruction, particularly under constraints of limited data. It achieves high-fidelity view synthesis and geometry reconstruction by combining sparse-view consistency with semantic supervision from diffusion models. The use of Score Distillation Sampling allows it to render physically plausible, semantically coherent 3D shapes from just a few images. The tri-plane representation ensures computational efficiency while preserving spatial detail. Unlike NeRF, which suffers from limited generalization and high data demands, SparseFusion performs well even in low-data regimes and can be extended to conditional 3D generation from text prompts.
However, this comes at the cost of reliance on external generative models and increased training complexity. Despite these trade-offs, SparseFusion offers a promising direction for future research in neural rendering, 3D generative modeling, and applications such as digital twins, AR/VR, and content creation from minimal data.

VIII. Gaussian Splatting

Gaussian Splatting is a real-time, high-fidelity 3D reconstruction and rendering method introduced in 2023 that departs significantly from the volumetric grid and neural field-based representations used in models like NeRF. Instead of modeling scenes with implicit neural networks, Gaussian Splatting directly represents 3D geometry using a set of spatially located 3D Gaussians, each of which possesses attributes such as color, opacity, orientation, and covariance. This explicit point-based representation allows for immediate and efficient rasterization using GPU-friendly operations, enabling real-time novel view synthesis with exceptional photo-realism and detail. The key innovation of Gaussian Splatting lies in treating these 3D Gaussians as volumetric primitives that can be blended together using alpha compositing, offering a continuous and differentiable rendering pipeline that bypasses the need for ray marching.

The architecture of Gaussian Splatting consists of a sparse cloud of 3D Gaussians that act as anisotropic ellipsoidal kernels in space. Each Gaussian is defined by its 3D position, scale (captured by a covariance matrix), orientation, RGB color, and opacity. To synthesize a novel view, the system projects all Gaussians onto the image plane of the virtual camera, rendering them using a splatting algorithm that blends their contributions based on depth and transparency.

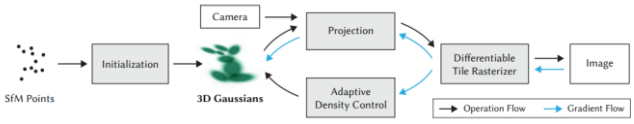

Figure 8: This shows the overall architecture of Gaussian Splatting, where optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians, then optimizes and adaptively controls the density of this set of Gaussians.

At its core, the rendering pipeline sorts visible Gaussians from back to front and composites their projected influence using a weighted accumulation function. A differentiable renderer is used to simulate the view from any camera angle, allowing gradients to flow through the entire pipeline. During optimization, both the positions and attributes of the Gaussians are updated to minimize photometric error with respect to the reference images. Because of this direct rendering and optimization approach, Gaussian Splatting avoids the high computational overhead of neural field evaluations and integrates naturally with classical rendering principles.

The most groundbreaking idea in Gaussian Splatting is the replacement of implicit volumetric fields with an explicit, surface-aware set of Gaussians. This allows for significant improvements in rendering speed and visual quality. Unlike NeRF, which requires volumetric sampling along each ray and neural network evaluations per point, Gaussian Splatting leverages GPU-accelerated 2D splatting techniques, which scale much more efficiently.

Another novel contribution is the use of anisotropic Gaussians, which can stretch and orient themselves to better approximate the underlying 3D geometry. This enables high-fidelity reconstructions with a relatively small number of points compared to point clouds or voxel grids. Gaussian Splatting also introduces a differentiable formulation of alpha compositing over the projected Gaussians, maintaining end-to-end learnability while enabling extremely fast rendering. This unique blend of classical rendering ideas and modern optimization techniques makes Gaussian Splatting a powerful alternative to neural rendering frameworks.

Training Gaussian Splatting involves optimizing the parameters of a set of 3D Gaussians to minimize the photometric reconstruction error between rendered views and the captured images. Initially, a sparse set of 3D points is extracted from the input views using structure-from-motion (SfM) or multi-view stereo (MVS) techniques. These points are then parameterized as 3D Gaussians with initial colors, scales, and opacities.

During training, novel views are rendered from the current set of Gaussians, and a loss is computed based on pixel-wise differences with ground truth images. This loss is backpropagated through the differentiable rasterizer to adjust the positions, sizes, colors, and opacities of the Gaussians. The differentiability of the rendering operation is critical, allowing for efficient gradient-based optimization. To improve convergence, additional regularization terms are sometimes added to encourage spatial smoothness and prevent overfitting to noise in the input images. Importantly, since there is no neural network involved in the rendering loop, training is often significantly faster and more stable than methods like NeRF.

Gaussian Splatting offers a fresh perspective on 3D reconstruction by replacing implicit neural fields with

explicit 3D Gaussian primitives. This approach results in faster training and real-time rendering while maintaining high visual fidelity. It eliminates the need for expensive ray marching and neural network evaluations by relying on direct alpha compositing of projected Gaussians, making it exceptionally efficient for both static and dynamic scene rendering.

Despite certain trade-offs in handling volumetric or translucent materials, Gaussian Splatting excels in reconstructing complex scenes from multi-view images with high accuracy and speed. Its interpretability, differentiability, and performance make it a strong contender in the landscape of neural rendering. As future work integrates learned priors, temporal coherence, and hybrid representations, Gaussian Splatting is likely to remain a foundational technique for real-time 3D scene modeling and view synthesis.

IX. COMPARATIVE ANALYSIS

Model	Strengths	Weakness
NeRF	High-quality rendering and photorealism	Slower training and inference, with high memory requirements
Pixel-NeRF	Works with a few images and fast inferences.	Lower fidelity than NeRF with fewer views
Instant-NeRF	Extremely fast training & rendering	Limited to static scenes, resolution depends on hash levels
Neural Angelo	Produces detailed geometry from video	Computationally very expensive
Gaussian Splatting	Real-time performance with high quality	Struggles with occlusions and dynamic scenes
Sparse Fusion	Real-time and Robust to sparse depth and efficient fusion	Depends on the quality of the sparse input

X. CONCLUSION

Collectively, these methodologies represent significant strides in 3D reconstruction, balancing computational efficiency with high-quality output. They underscore the transformative role of machine learning in processing 2D images to generate detailed and accurate 3D models, with applications ranging from interactive 3D environments to single-image reconstruction. As techniques evolve from fully implicit representations like NeRF to hybrid and explicit forms such as Gaussian Splatting and SparseFusion, the field is moving toward models that not only render photo-realistic views but also generalize across scenes with minimal supervision. This shift opens the door to scalable and real-time 3D content creation, bridging the gap between vision and geometry. Future research will likely focus on improving generalization, integrating semantic understanding, and reducing reliance on large datasets or pretrained models, ultimately enabling broader applicability across AR/VR, robotics, digital twins, and creative industries.

XI. FUTURE WORK

While recent advances have significantly improved 3D reconstruction from 2D images, several challenges remain. Future research should focus on enhancing generalization to novel scenes and objects with minimal data. Real-time rendering methods like Instant-NGP and Gaussian Splatting show promise, but further efforts are needed to reduce inference time and memory usage without compromising quality.

Another direction involves incorporating richer semantics and interactivity into 3D models, moving beyond geometry to enable scene understanding and object-level reasoning. Additionally, integrating multi-modal inputs—such as text and images could enhance reconstruction in sparse or ambiguous settings, as demonstrated by approaches like SparseFusion. These directions open exciting opportunities for more robust, flexible, and intelligent 3D systems
wo

REFERENCES

[1] Jonathan Mildenhall, Pratul P. Srinivasan, Mariusz Klain, et al. **NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. pp. 6346-6355.

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. **Instant Neural Graphics Primitives with a Multiresolution Hash Encoding**. ACM Transactions on Graphics, 41(4):1-12, 2022.

[3] Xiuming Zhang, Liangyu Zhao, Yijun Li, et al. PixelNeRF: **Neural 3D Reconstruction with Pixel-Aligned Implicit Functions**. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. pp. 2616-2625.

[4] Chen-Hsuan Lin, Jun Gao, Luming Tang, Xiuming Zhang, Zhiding Yu, Sergey Tulyakov, Shalini De Mello, Sanja Fidler**. Neuralangelo: High-Fidelity Neural Surface Reconstruction.** Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[5] Z. Zheng et al., "SparseFusion: Distilling View Prior for High-Quality Single-View 3D Reconstruction," *arXiv preprint arXiv:2305.18766*, 2023.

[6] B. Kerbl, G. Kopanas, T. Leimkühler and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," ACM Trans. Graph., vol. 42, no. 4, 2023