Reconstructing 3D Reconstruction: A Graphical Taxonomy of Current Techniques

by

Cynthia Chang

A Thesis Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(COMPUTER SCIENCE)

August 2024

Dedication

To my family.

Acknowledgements

I extend my deepest gratitude to my advisor and committee chair, Professor Saty Raghavachary, for our numerous enlightening and creative discussions both prior to and throughout the writing of this thesis. This work could not have been possible without his depth of knowledge in all things generative AI-, ML-, animation-, and graphics-related (there's more but I must cut it off at some point!). Thank you for your unwavering belief in me from the very first day we met.

I am also grateful to my committee members, Professor Victor Adamchik and Professor Aiichiro Nakano, for their review and critique of this work in order to ensure that it is of my best quality.

A very special thanks to my friends and loved ones, for always uplifting me with their friendship and encouragement.

Finally, I thank my parents for their consistent and unconditional support throughout my life. I am the person I am today because of their dedication.

Table of Contents

List of Figures

Abstract

3D reconstruction has rapidly advanced in recent years as both industry and research communities work towards (re)-creating our changing world around us - discovering new techniques while improving the old. Recent breakthroughs such as Neural Radiance Fields and 3D Gaussian Splatting have paved the way for an influx of new methodologies in 3D reconstruction. With dozens of papers on this topic published weekly to academic archives at the time of writing this thesis, it is ever more crucial that there is a systematic overview of recent methods and applications to provide both established researchers and newcomers with up-to-date information. This thesis aims to contribute a review of recent papers on 3D reconstruction by comparing and analyzing the different techniques used. Of the many approaches, we focus on research produced within the past 5 years, constructively organize the literature, and ultimately produce several taxonomies, each based on a different aspect of 3D reconstruction.

Chapter 1

Introduction

The human experience on Earth occurs in many sensory dimensions, and is often remembered in any given combination: the smells, tastes, and sounds of the summer festival; the bone-chilling temperatures and visual spectacle of an aurora borealis; or the softness felt under fingertips gliding through the fur of a furry companion. However, given that up to 80% of all human perception occurs through sight [1], it is no surprise that both our technology and creative arts focus on capturing, remembering, and envisioning the scenes around us. Simply seeing images, artwork, and videos is often enough to evoke deep emotion, and even transport us through time.

The topic of 3D reconstruction has been a long-standing, ill-posed problem in the field of computer vision and computer graphics. Many different objects in three dimensions can be reduced in a multitude of ways into the same two-dimensional rendered representation; thus the reverse problem can present itself in significantly different ways depending on the perspective. Despite this, the process to recreate these digitized versions of ourselves and the world around us has drastically improved since its inception - especially so in recent years with advances in hardware and GPU technologies allowing deep learning techniques to replace all or parts of classical methods. This has blossomed into an ever-increasing demand for 3D assets, and subsequently large efforts in both research and industry work on 3D generation and reconstruction. The field continues to advance at staggering speeds, with techniques using neural radiance fields (NeRF) [2] and Gaussian splatting (GS) [3] currently trending for reconstructing depth and dimension from flat, 2D inputs.

Taxonomies are, by nature, pieces of on-going work - timestamped on the current latest-and-greatest. In nascent fields where research development is fast-paced and new advancements are discovered often, it can be expected that such surveys quickly become outdated and therefore require updating. While the topic of 3D reconstruction itself is not novel, recent advances with deep learning techniques are. Thus, most taxonomies on non-generative/classical methods of 3D reconstruction date back to the early 2000s, whereas learning-based generative methods have only begun within the past decade. Due to the brisk pace at which deep-learning-based reconstruction literature is appearing, current taxonomies have naturally become narrower in scope, often focusing on specific areas of the broader 3D reconstruction research.

1.1 Objectives

This thesis aims to contribute to existing taxonomy work regarding 3D reconstruction methods. We do so by defining the following objectives:

- summarize the details and methods of applications and advancements in 3D reconstruction within the past 5 years, and their respective key contributions; the publication date cutoff for papers considered for taxonomy in this thesis is May 24, 2024
- present multiple taxonomies based on various classifications, providing a conceptual framework for users to interpret results as needed by their research requirements (i.e. input, output, and scale of result)
- infer the future direction of 3D reconstruction based on trends seen in this thesis work

1.2 Outline

The remainder of this thesis is organized as such: Chapter 2 reviews previous taxonomies; Chapter 3 provides background information on 3D representations relevant to this thesis; Chapter 4 reviews 3D reconstruction literature included in this taxonomy and their key contributions; Chapter 5

presents the taxonomy, and discusses the process behind creating the visualizations; Chapter 6 summarizes this thesis and offers ideas for potential applications and iterations of this work.

Chapter 2

Previous Work

In 2019, Han et al. [4] provided the first comprehensive taxonomy on image-based object reconstruction using state-of-the-art deep learning methods. They compared over a hundred image-based 3D reconstruction methods that used convolutional neural networks, a trend which began in 2015. Khan et al. [5] followed later in 2022 with a broader review of deep-learned 3D reconstruction utilizing single RGB images only. Much of their literature selections focused on methods which produce and use depth maps, one of the many ways 3D geometry can be represented.

Phang et al. [6] presented one of the more current general reviews on 3D reconstruction in 2021, and included statistical, deterministic, and generative reconstruction methods. Research has shifted towards methods centered around deep-learning since then, and most 3D reconstruction taxonomies have followed in focusing on these areas.

In the era of deep-learning-based 3D reconstruction, neural fields and point clouds have become key methods of representing in three dimensions, with hundreds of papers applying methods to each, respectively. Huang et al. [7] provide a survey and benchmarking comparison of various methods to reconstruct 3D surfaces from point clouds. Xie et al. [8] have a very comprehensive review from 2022 focused on neural fields and the vast number of ways to reconstruct the objects and scenes they represent. Mildenhall et al.'s NeRF [2] enabled fast and robust reconstruction from neural fields, further spurring on research. Gao et al. [9] released a survey in late 2023 comparing the different extensions of NeRF.

Following NeRF is 3D GS [3], a new transformative technique. A plethora of research extensions on 3D GS have been pursued since its debut barely a year ago. Earlier this year, Chen et al. [10] conducted a focused survey on the many practical applications and novel frameworks of 3D GS extensions. Their survey is the first comprehensive overview of 3D GS, and to our knowledge the most recent in the general field of 3D reconstruction. Nevertheless, the focal point on 3D GS does not place their survey in the broader taxonomy of 3D reconstruction as a whole, but rather a single branch. The extent to which 3D GS has revolutionized high-fidelity 3D reconstruction using deep-learnable Gaussians indicates a need for updated conceptual frameworks on 3D reconstructions.

All previous taxonomies here, with the exception of [9], present their classifications in a tabular format. Tables are widely useful for lower-level details in each row and serve their purpose for various benchmarking comparisons in these taxonomies, but redundant columns and a lack of graphical diagrams make it difficult to visualize and understand the patterns, trends, and relationships between the different methods. This misses critical information, as many of the methods developed in 3D reconstruction iterate on some aspect of an existing technique or another. Figure 2.1 shows an example of column redundancy, and Figure 2.2 shows the comparatively direct usefulness of tables for benchmarking.

Model	Dataset	Data Representation
Neural renderer [35]	ShapeNet [10]	Meshes
Residual MeshNet [36]	ShapeNet [10]	Meshes
Pixel2Mesh [37]	ShapeNet [10]	Meshes
CoReNet [38]	ShapeNet [10]	Meshes

Figure 2.1: **Tabular Redundancies.** Table 3 of Vinodkumar et al. [11] comparing ”3D reconstruction models using mesh data representation”. Notice how the ”Dataset” and ”Data Representation” columns consist of the same values down the whole column.

Upon first glance, Figure 2 of [10] (not included here) seems to be a graphical representation of their survey, however it is actually just an overview of the structure of the survey itself. Its usefulness in quickly providing an understanding of the relationships between different 3D GS uses and applications underscores what the other tables could have been in the rest of the paper.

Method	GS	PSNR↑	SSIM↑	LPIPS↓
D-NeRF [122] [CVPR21]		30.50	0.95	0.07
TiNeuVox-B [229] [SGA22]		32.67	0.97	0.04
KPlanes [37] [CVPR23]		31.61	0.97	-
HexPlane-Slim [230] [CVPR23]		32.68	0.97	0.02
FFDNeRF [125] [ICCV23]		32.68	0.97	0.02
MSTH [231] [NeurIPS23]		31.34	0.98	0.02
3D GS* [12] [TOG23]	✓	23.19	0.93	0.08
4DGS [97] [ICLR24]	✓	34.09	0.98	-
CoGS [133] [CVPR24]	✓	37.90	0.98	0.02
4D-GS [99] [CVPR24]	✓	34.05	0.98	0.02
D-3DGS [98] [CVPR24]	✓	39.51	0.99	0.01

Figure 2.2: **Fine-Grain Details with Tables.** Table 3 of Chen et al. [10]. A standard table format useful for benchmarking performance results.

Tables are useful for detail, but graphs allow for deeper relational understanding rather than the rote memorization that tables would require. We address this by representing our taxonomy as a hierarchical tree.

Chapter 3

3D Representations

Various ways to represent 3D objects and scenes have been developed in computer graphics. A general understanding of how 3D objects are represented is key to then being able to understand how these objects play into the different methods of 3D reconstruction, and why one representation may be chosen over another. The following sections provide concise information on many widely used representations in 3D object and scene modelling, and we follow with an additional explanation in the context of 3D reconstruction.

3.1 Meshes

Similar to its real-world material made of wire or thread, a mesh in computer graphics is intuitively a collection of 2D polygons connected together in 3D space by shared edges, faces, and vertices. Triangle meshes are composed of many, well, triangles, and are the most commonly used polygon type. Triangle meshes are also the only polygon shape used by graphics renderers for rasterization (though Pixar's RenderMan renderer is an exception and uses quadrilateral micropolygons [12]). Figure 3.1 from [13] shows several examples of 3D wireframe meshes, and their optimized variations.

Meshes provide explicit geometry and topology details about its represented structure. Thus, they are widely used for surface reconstruction and traditional 3D modelling, animation, and rendering. The ability to directly utilize the data structure of a mesh in 3D applications makes it

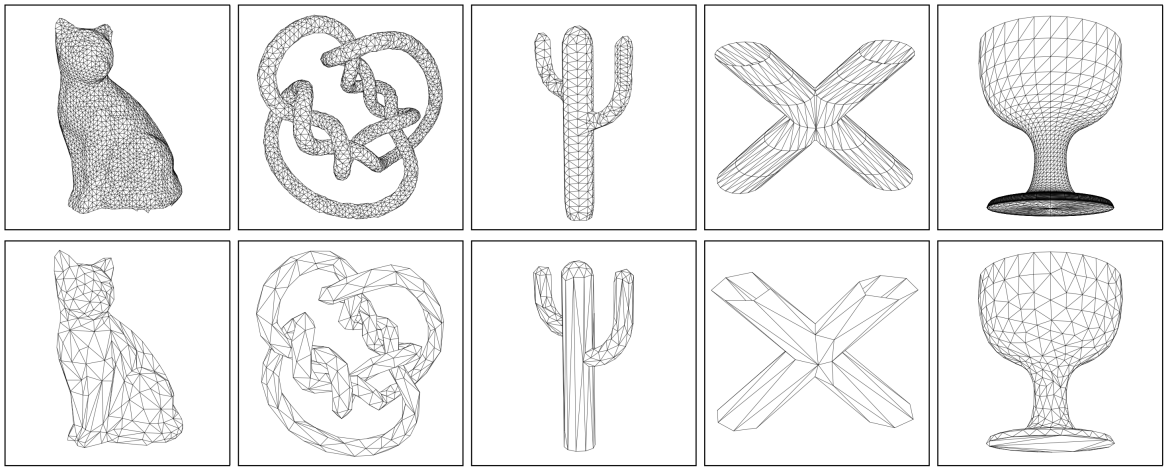

Figure 3.1: **Mesh Representation and Optimization.** Figure 1 from Hoppe et al. [13] showing several 3D meshes in wireframe format. The top row is the original mesh with a denser set of polygons. The bottom row is the same mesh with a smaller number of polygons per mesh (i.e., a coarser representation), while preserving overall object detail.

the most widely used format; since the ultimate goal of 3D reconstruction tasks is to (re)-create the virtual object in 3D, it is crucial for output representations to be meshes. This is particularly important for 3D objects where non-rigid deformations, animations, and/or editing is desired.

3.2 Voxel Grids

Voxel grids (Figure 3.2) are array-like structures composed of many smaller voxels, or singular 3D unit-cubes. These unit-cubes are laid out along Cartesian XYZ axes and axis-aligned with them. Each voxel can contain one or more pieces of information about the object at that voxel's location, such as RGB color, opacity, signed distance field values, and the surface normal vector.

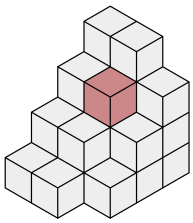

Figure 3.2: **Voxel Grid Representation.** Multiple voxels stacked together in a voxel grid, with a singular voxel shaded pink. Source: Wikimedia Commons.

An example of such structured storage lies in one of the world's most popular video games, Minecraft [14], a voxel grid-based 3D sandbox game. Player worlds consist of cubic representations of terrain, atmosphere, and items (Figure 3.3) - called "blocks" - and all block data is stored together in a large voxel-grid.

Figure 3.3: **Voxels in Minecraft.** A screenshot of one of the author's Minecraft worlds, depicting blocky terrain and cubic tree and bee hive objects. Additional shapes such as the flowers each occupy a cube, with the air texture in the space above the flower texture.

The uniformity of a structured voxel grid makes it useful for filtering techniques in deep-learning methods such as convolutional neural networks [15]. Since data is explicitly stored for application, voxel grids significantly reduce computation costs and thus are an excellent representation for reconstruction methods. However, a densely represented surface is highly memory inefficient [16], as voxel grids must store all voxels regardless if a voxel space is empty (which, arguably, re-increases computation cost a bit). To that end, researchers have recently developed hybrid representations of voxel grids with neural fields to utilize sparse voxel grids [17], [18], [19], eliminating the need to store empty voxels in memory.

3.3 Point Clouds

Point clouds comprise of a discrete set of unstructured points in 3D space, and are commonly used for surface modelling and reconstruction. This data type requires either a 3D scanning device, laser (like LiDAR), or computer program to generate. They are often used for 3D scene reconstruction tasks, and prevalent in industries such as robotics, CAD, and architectural engineering and restoration (Figure 3.5).

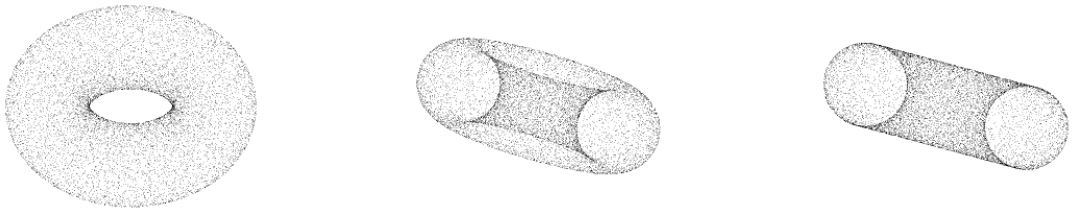

Figure 3.4: **Point Cloud Torus.** Different angles of a point cloud representation of a torus, a donut-like 3D object. Source: Wikimedia Commons.

Figure 3.5: **Point Cloud Applications.** *Left:* Point cloud of the Notre Dame, with over a billion data points. Source: National Geographic. *Right:* Point cloud of the Roman Colosseum. Source: GRAIL University of Washington, Seattle.

While the mesh representation described earlier consists of 3D points and their connected edges and faces, point clouds have only the points in space themselves. Thus, similar to the overarching inconsistency of 2D-to-3D mapping, many different objects can be represented by the same point cloud depending on how the cloud was generated. Without the explicit description of an object

by way of defined edges, there are infinitely many ways to create a surface of polygons from an unorganized collection of 2D points in 3D space [20].

3.4 Depth Maps

Depth maps are representations of an image's distance to a particular viewpoint, either as an image (often in greyscale) or an image channel. The value at each pixel location of a depth map indicates how close that pixel lies to the camera viewpoint - in greyscale images, this translates to whiter pixels being closer and darker pixels being farther. Depth maps are commonly referred to in traditional 3D computer graphics pipelines as "depth buffers" or "z-buffers," where the z component is the z-axis of the xyz-plane.

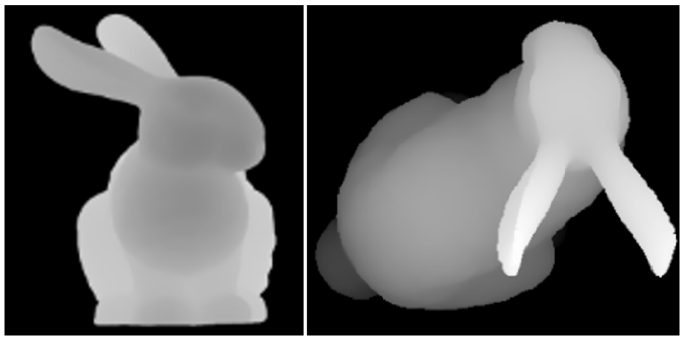

Figure 3.6: **Depth Maps.** Depth map images of the Stanford Bunny from [21].

Depth maps are beneficial in the 3D reconstruction pipeline for extracting dimension from a singular 2D plane. They are often used in conjunction with flat images to reconstruct the 3D scene, such as in VisionGPT-3D [21]. If multiple views of the same object are extracted into 3D space with their respective depth maps, a full 3D representation (such as a point cloud) can be quickly generated.

3.5 Signed Distance Fields

A signed distance field (SDF) is a continuous representation in 3D space, where point x_i indicates its orthogonal distance from the surface of the object. Each point further takes on a positive or negative value depending on whether it is inside or outside of the 3D object - conventionally, points within the object are negatively-signed. The surface itself then implicitly exists at the 0-value boundary of the SDF and can be easily reconstructed into a mesh or other representation. SDFs are also commonly used for 3D texture mapping and other surface manipulation techniques. They are additionally helpful in determining object space and orientation within point clouds and other representations.

Due to its implicit representation of surface geometry, SDFs have remained steadfast as a 3D representation. It has also become a popular implicit encoding (either in whole or in part) for neural rendering methods in 3D reconstruction pipelines, as seen in [22], [23], [24], [25]. These works are further discussed in Chapter 4.

3.6 Occupancy Fields/Networks

Whereas points in a SDF depict distance from the surface of an object, an occupancy field is the discrete binary representation of where in space an object "occupies". Occupancy networks [26] then, are neural networks which implicitly encode this discrete field into a continuous decision boundary to represent the 3D geometry. Because of their point-based primitive, occupancy fields and networks are a memory-efficient representation of 3D geometries. They are able to translate between different geometry representations, and can reconstruct 3D objects with classic volumetric rendering techniques.

3.7 Neural Fields

Xie et al. defines a neural field as "a field that is parameterized fully or in part by a neural network." Most often, neural fields are parameterized as Multi-Layer Perceptron (MLP) networks with well-defined gradients [8]. In other words, for each sampled coordinate point in space, its relevant data (ex: RGB value) is encoded within a neural network and its activation function(s). Figure 3.7, adapted from [8], shows a general neural fields pipeline.

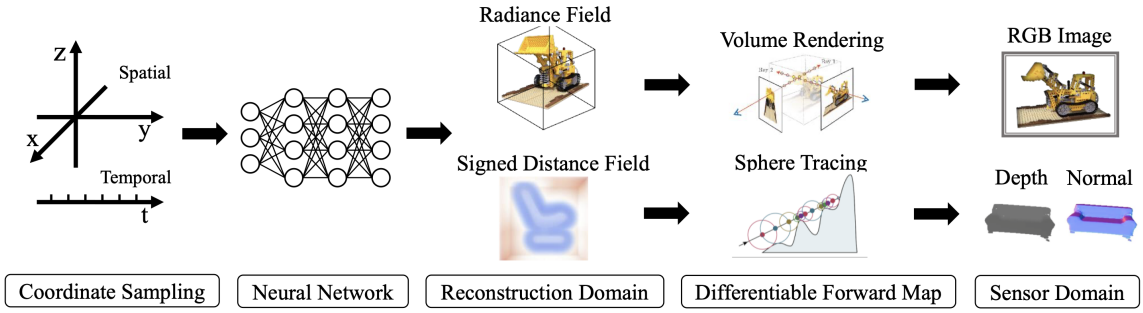

Figure 3.7: **Neural Fields.** Figure 3 (adapted [2], [27]) from Xie et al. [8]. Adapted for this thesis.

Neural fields circumvent a key issue present with dense voxel grids - as object and/or scene detail increases so does the memory requirement for maintaining storage of all these details. Instead, neural fields limit the complexity of data to the number of parameters/features of the network. Thus, neural fields are able to implicitly store details of an arbitrary resolution as long as there are sufficient network features able to encode everything. The revolutionary 3D reconstructing method NeRF [2] and its many extensions [18], [28], [29], utilized this representation. By querying 5D coordinates (spatial x, y, z, and camera view direction θ, ϕ), NeRF's neural algorithm synthesizes novel views of 3D scenes and objects with the expected radiance of a spatial location. We go into more detail on NeRF and related methods in Chapter 4.

3.8 3D Gaussians

A very recent and novel representation for 3D geometry, 3D Gaussians have become a trending focus of computer vision and graphics research. When neural radiance fields emerged in 2020, and

following NeRF-based techniques consistently proved to be state-of-the-art (SOTA) for volumetric rendering, it was widely believed that implicit representations were the path forward. Kerbl et. al. [3] challenged this belief in late 2023. Their discrete, explicit representation of points expressed as individual 3D Gaussians has launched the industry into a new path of research, and 3D Gaussians have proved to reconstruct precise, scalable results so far.

First, a point cloud representation is estimated given few images of an object or a scene. Each point is then converted to a Gaussian distribution with a positional coordinate for location, covariance matrix for scale, and RGBA values for color and opacity. By representing each point in space as a 3D Gaussian and using splatting on distributions to render locally, this enables a near real-time rendering of scenes in much larger size than its predecessors. Encoding these local spatial details in each Gaussian also allows much finer-grained detail alongside its scalability. The many applications of 3D Gaussians and the nascence-yet-promise of this area of research makes it a very interesting topic for concurrent and future work in computer graphics.

3.9 Hybrid

In the 3D reconstruction pipeline, there is often a need for translating between data representations to achieve the desired output [18], or for having a new data representation which incorporates two or more different data representations simultaneously [17]. This may be through the merging of an implicit and explicit representation, or developing an explicit representation of an implicit field, and so on.

Neural-based implicit representations in particular often blend multiple representations together. In fact, as the field advances, fewer and fewer approaches rely purely on a single representation - the majority of literature in this thesis uses some combination of geometry representation. This work will refer to these non-singular representation uses as hybrid representations, similar to [20].

3.10 Other Representations

While the following representations are not highlighted in the literature we review for the taxonomy, the inclusion of these primitives provides a comprehensive overview of 3D representations used throughout the history of 3D reconstruction and its applications.

Blobby Metaball Also referred to as a blobby mesh, this representation consists of a field centered around a point in 3D space [30]. Each field has a force that is strongest at the center of the point and weakest at the edge of the point field. Designed to coexist with many other metaballs, the field begins as a symmetric, circular field around its point. When two or more metaball fields interact with each other, a "neck region" is formed at their midpoint to create a smooth deformation of spherical objects. As done in [30], the combined fields can later be extracted into a blobby mesh for further simulation and/or processing. Blobby metaballs are mostly used in fluid simulation, as it easily simulates surface tension between fluid molecules. However, despite the efficiency with which fluids can be modelled, it lacks accuracy and cannot replicate intricate details or highly dynamic fluids.

3D Curves and Spline Patches Spline curves and spline patches are very commonly used throughout the animation industry for representing curvature and smoothness in 3D. Because splines are interpolated curves derived from piecewise functions, they allow for a smoother surface modelled with fewer points [31]. This makes splines more efficient in modelling these smooth surfaces than using a mesh surface, the latter of which would require many more polygons to replicate the same curve. However, splines are difficult to edit. Despite this, spline curves are very commonly used to emulate complex, large systems of individually curved components, such as hair or fur strands. Each hair strand is modelled as a spline curve, which allows for precise control over each strand in both movement and metadata.

Subdivision Surface Sometimes a polygonal mesh has a coarser initialization than desired. Specifically, there is a poor representation of smooth edges or curves. Subdivision surfaces rectify

this issue via mesh processing algorithms such as Catmull-Clark [32] which recursively splits the mesh faces at rendering time for a smoother and denser mesh. By operating on the base mesh at render time, subdivision surfaces enable the smoothness of splines while maintaining the memory-efficiency and operative capabilities of a mesh (without storing the denser, smoothened mesh). It is commonly used in computer graphics applications such as Computer Aided Design (CAD) for modelling smooth, synthetic objects [33], and more recently in computer animation [34].

Light Fields A light field is a 4D data structure storing the varying amounts and directions of light that flow through every possible point in space [35]. These fields are parameterizable by two parallel planes in 3D space, and can be acquired in a variety of ways. Direct scene acquisition typically requires multiple calibrated image sensors, however time-sequential capture with a single image sensor greatly reduces system cost and is often used for static scenes. The detailed depth information that specialized light field cameras are able to capture additionally enables automatic post-capture techniques such as image refocusing. Light fields can also be easily generated from multiple virtually rendered 2D images - here, sampling method will heavily influence the outcome. While the image-in, image-out pipeline has no explicit 3D reconstruction, one can be produced if needed for additional 3D reconstruction and view synthesis.

Particle Systems Particle systems and their simulations are excellent for providing the individual dynamic information of a large quantity of particles, i.e. points in space. Although also based on discrete 3D points, the similarities between particle systems and point clouds/other point-based representations end there. These points are camera-view rendered, and cannot be oriented differently from facing the camera. Additionally, particle systems are not texturizable, and instead store variable data such as trajectory and/or acting forces [36]. However, they are immensely useful in replicating high intensity phenomena - such as explosions, liquids, flames, and so on - due to their dynamic nature. This “fuzzy” phenomena is difficult to render without using this primitive, since other representations are typically glued together in one way or another.

Chapter 4

3D Reconstruction

The rapidly-evolving nature of this field renders it nearly impossible to present a truly comprehensive, full review of each and every method available. For the scope of this thesis, we focus on 40 approaches from within the past 5 years (2019 - May 24, 2024). The majority of papers fall within the past 2 years, with the bulk of literature from throughout 2023. This is due to the recent trends in applying NeRF and Gaussian-based techniques, both of which have shown stellar results. Many approaches selected in this report are from top conferences and journals in computer vision, graphics, and machine learning; several others consist of work completed in top technology companies foraying into the universe of AI and AR/VR. Some very recent technical papers from arXiv are also included in this report. See Figure 4.1 for a tree-like visualization. Each vertical column of the tree indicates a year, and colors indicate the method node's main 3D representation.

Tatarchenko et al. [37] defined reconstruction itself as a "per-pixel reasoning about the 3D structure of the object shown in the input." They added that for modern 3D reconstruction with deep-learning, reconstruction has become blended with image recognition tasks - indeed, many recent 3D reconstruction papers provide a fully contained, end-to-end framework which takes input and both recognizes what object is contained and reconstructs its 3D form represented in some primitive for further manipulation.

3D reconstruction methods can be generically categorized as static or dynamic; static reconstruction does not incorporate a time feature, whereas dynamic reconstructions often build upon existing static methods and spatio-temporal accuracy becomes a key component. We loosely follow

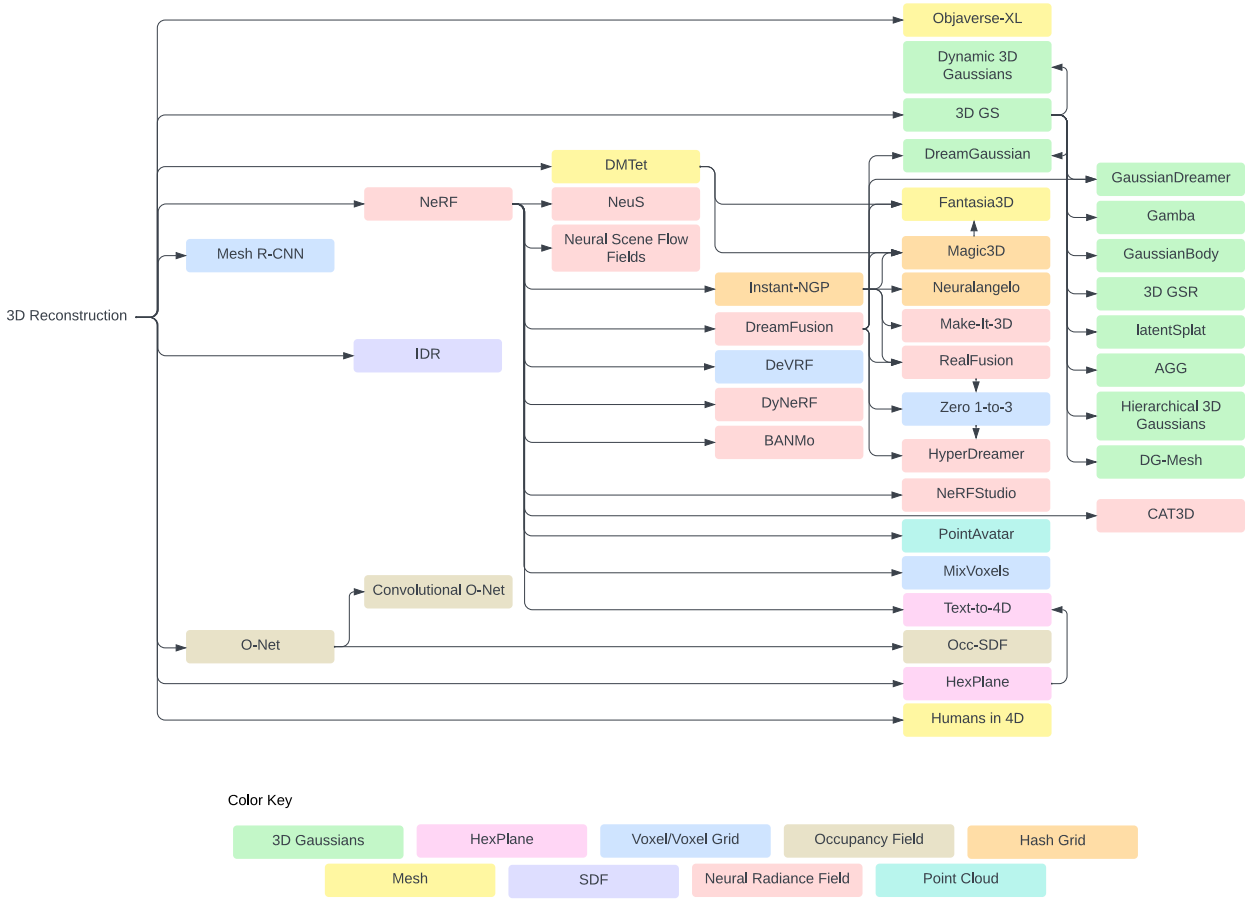

Figure 4.1: **Literature Overview.** A visualization of all literature reviewed for this thesis. Tree column 1 indicates 2019, column 2 indicates 2020, and so on until 2024. Arrows indicate direct or heavy influence. Node colors indicate its method's main 3D representation, as shown in the legend at the bottom. For hybrid representations, we have chosen to categorize using the most prevalent representation.

the structure of Chapter 3 and divide the literature by their choice of geometry primitive, reviewing each of its developments from static to dynamic.

4.1 Mesh-based

Figure 4.2: **Mesh-based Reconstruction.** Figure 1 from [38] showcasing their human mesh recovery results from static and dynamic input.

Deep **M**arching **Tet**rahedra (**DMTet**) [39] utilizes the tetrahedral grid to encode underlying object surfaces, modelled as an implicit SDF. It is easily extracted into a mesh via a differentiable marching tetrahedral algorithm which enables joint optimization of surface geometry and appearance. Starting at the unit tetrahedral with a coarse voxel as a guide, DMTet predicts the SDF and deforms the grid to best refine the object. DMTet is able to generatively synthesize complex 3D topologies and is trained on various animal shapes. Its simplicity and efficient marrying of explicit and implicit geometry representation made it a popular choice for many later works.

One such work inspired by the high-quality results of DMTet is **Fantasia3D** [25], a text-to-3D mesh framework for objects. Fantasia3D separates geometry and appearance learning into individual networks, and is the first to incorporate reflectance lighting into a text-to-3D framework as a result of this disentanglement. They follow DMTet in parameterizing the 3D geometry, but change initialization to either a default 3D ellipsoid or a user-provided 3D model. Fantasia3D is able to produce high quality, photorealistic meshes compatible with graphics engines for model relighting and editing.

Goel et al.'s **Humans in 4D** [38] reconstructs humans and tracks the meshes over time as seen in Figure 4.2, taking 3D human reconstruction (a well-studied topic) a step further into dynamic human mesh recovery. The reconstructed model is used as input to a 3D tracking system to train body model keypoint predictions, resulting in high quality single-image 3D reconstructions even when contorted into unusual positions as seen in athletic movements. They further link the reconstructed model over time with the 3D tracking, and are able to jointly reconstruct and track models in video inputs. Their system is able to maintain identities while dealing with multiple human subjects and occlusions, achieving SOTA results for tracking and action predictions. This approach makes a significant improvement over prior works in human pose and shape estimation in unique poses, and is a foray into more accurate human body animations using deep-learning.

4.2 Voxel-based

Figure 4.3: **Voxel-based Reconstruction.** Adapted from Figure 1 of [40]. From left to right: 2D image recognition of objects, to their predicted coarse voxels, and finally to its refined mesh output.

Mesh R-CNN [40] (Figure 4.3) achieved a first attempt at 3D shape prediction in-the-wild (ITW), and is able to detect multiple complex objects within a single 3D scene. They outperform previous benchmarks that relied on deforming an a priori mesh, whereas Mesh R-CNN is able to reconstruct arbitrary topologies. Mesh R-CNN does so through feeding an input image into a 2D object prediction network, after which coarse voxelization is applied to estimate the object depth and shape. The voxels are then transformed into a 3D triangle mesh to be further refined into the final output, which contains individual meshes for every object within the complete scene and likely gives the method its final name.

MixVoxels [41] proposed using a mix of static and dynamic voxel grids to achieve rapid reconstruction of dynamic 4D scenes. The two different voxel grids are processed in separate networks, and a novel variational field is incorporated to estimate the temporal variance of each voxel. By separating the static and dynamic grids, the dynamic features are encoded in the voxel corners, and the full network is able to focus learning on the shape and distinct boundaries of high-motion regions. The two voxel grids are then merged to render the final ray color per timestep. Although complex lighting remains a challenge for MixVoxels, it is able to produce comparable or better results with a large speed up in multiview 3D video synthesis.

4.3 Occupancy Networks

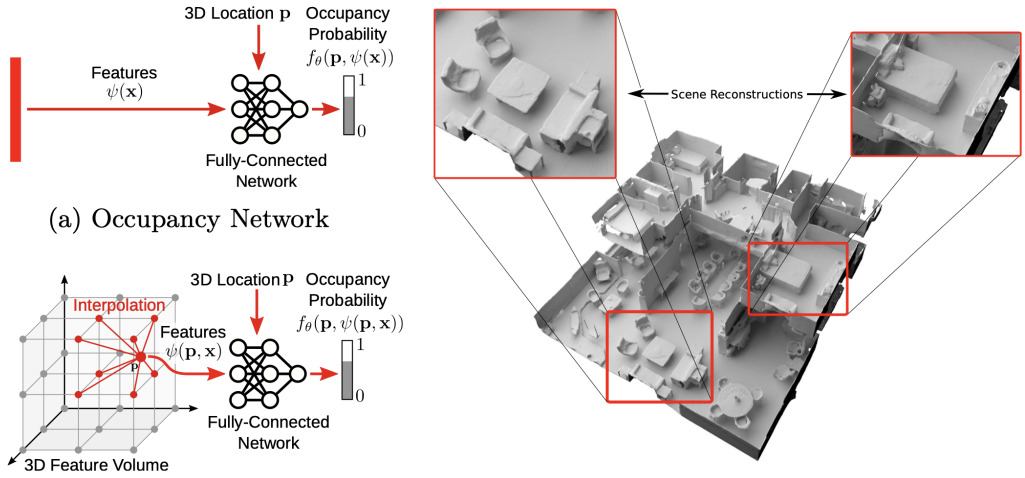

Figure 4.4: **ONet-based Reconstruction.** Figure 1 from [42] depicting the difference between a) the occupancy network approach and that of b) the convolutional o-net approach with c) the resulting complex, large indoor scene reconstruction using convolutional occupancy networks.

In 2019, Mescheder et al. proposed a novel implicit representation of 3D geometries as the continuous decision boundary function of a deep neural occupancy network. **O-Net** [26] achieved 3D surface reconstruction from single images, point clouds, or coarse discrete voxel grids. Up until then, 3D representations had mainly consisted meshes, point clouds, or voxels - all of which are difficult to train neural networks with. O-Net drastically reduces training memory requirements, as

the new representation relies on a binary occupancy field to determine the decision boundary. The complete occupancy function then becomes a binary classification problem with an emphasis on the decision boundary specifically, which implicitly represents the surface geometry.

Peng et al. followed O-Net with **Convolutional O-Net** [42] (Figure 4.4), adding flexibility to the implicit representation for finely detailed reconstructions of 3D objects and scenes. By combining a convolutional encoder and implicit occupancy decoder, Peng incorporated structural reasoning into the 3D space. This representation preserves fine geometry details previous lost in O-Net, and enables scalable complex reconstruction; large indoor scenes can be reconstructed in the "sliding-window" fashion paramount to convolutional neural networks. Despite requiring 3D inputs such as noisy point clouds or voxel grids, the method generalizes well from synthetic to real-world datasets.

4.4 Point Clouds and SDFs

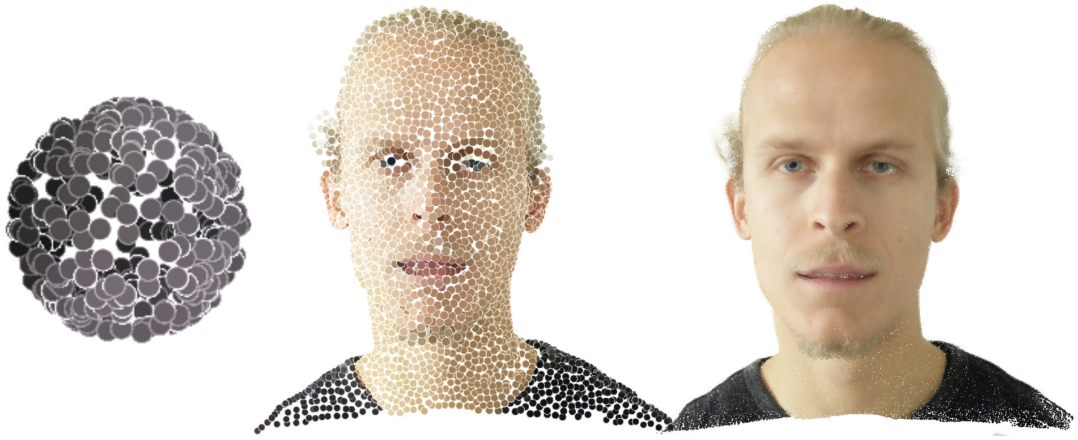

Figure 4.5: **Point Cloud-based Reconstruction.** Image adapted from [23]. PointAvatar utilizes a coarse-to-fine refining method on the point cloud to create the final avatar.

PointAvatar [23] exercises attribute disentanglement through a point-based human avatar representation. Their end-to-end framework creates realistic animatable and relightable head avatars from many casual videos, and bridges the gap between explicit and implicit representations. They do this by utilizing a point cloud (Figure 4.5 for their geometry and a continuous deformation

field of FLAME blendshapes (learned deformations and poses of a pre-trained 3D facial mesh model) [43]. They separate pixel color into its intrinsic albedo and a surface normal-dependent shading, achieving relightability even with novel view synthesis (NVS). A coarse-to-fine geometry optimization strategy produces SOTA results in complex structures like thin hair strands and detailed facial geometries.

One multiview surface reconstruction method used fully differentiable renderers learned through a neural network alongside geometry and lighting attributes. **I**mplicit **D**ifferentiable **R**enderer (**IDR**) [44] presented a zero-level set of a neural network, in which the neural function modeled surface geometry as a SDF. The choice to model the geometry as a SDF provides efficient ray casting, and thus efficient ways to get the appearance and lighting of the surface through the ray casting. The pixel colors are represented as a differentiable function on geometry, appearance, and lighting - creating a fully differentiable renderer. They additionally apply 2D masks to supervise during 3D training, helping to improve the neural network. IDR achieved smooth, realistic rendering, however required "reasonable" camera settings and does not work with ITW data.

SDFs have shown to work well in large-scale methods since SDFs represent the scene as a whole. But in areas with complex, large-and-small structures, the smaller ones are often removed to preserve larger objects. Occupancies do not have this issue as they represent all objects as points, and are excellent in discovering intersections. Thus, Lyu et al. introduced an occupancy-SDF hybrid (**Occ-SDF**) [45] in an implicit neural representation to improve reconstruction of low-intensity, small and complex (i.e. thin) structures in a room-level scene, offering finer details where large-scale reconstruction models miss the mark. They implement a feature-based color rendering scheme to overcome vanishing gradients, and outperform SOTA methods on particularly small and dark areas of room-level scenes. This marks an important step in accurately replicating small-yet-complex environments, a common occurrence in AR/VR applications.

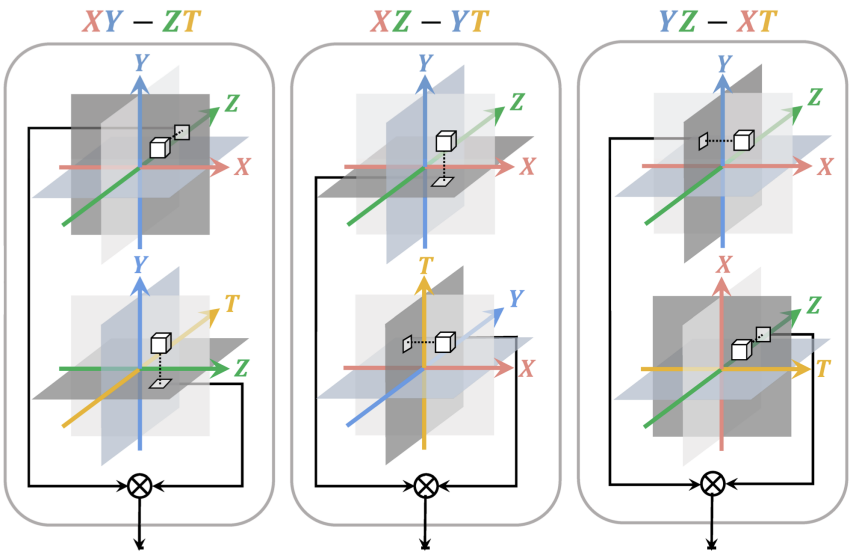

Figure 4.6: **HexPlanes for Dynamic Reconstruction.** Adapted from [46]. HexPlane proposes to parameterize dynamic 3D scenes into 6 planes of learnable features in 3D space, which are then concatenated at the end as input to a tiny MLP.

4.5 HexPlanes

HexPlane [46] uses 6 array-like planes to encode dynamic 3D scenes (Figure 4.6). Cao et al. realized that 4D volumes could be decomposed into only 6 planes of learnable features conditioned on time. Fusing together a vector from each plane is sufficient to restore all information from a tiny MLP, and because the planes must share information across all timesteps, this also reduces the memory usage and training time of dynamic scenes. This new hybrid representation, consisting of an explicit scene representation in combination with a tiny implicit MLP, was shown to be quite efficient and lightweight for dynamic NVS. In their benchmarking reports, HexPlane showed to be 100 times faster than DyNeRF. Additionally HexPlane is designed to easily convert to a mesh, making it even more applicable towards fast, dynamic computer graphics.

Singer et al. took advantage of this new representation to generate dynamic 3D scenes from text input [47]. Their framework, **Text-to-4D**, produces any-angle-viewable scenes. Trained on only text-image pairs from a dynamic camera, Text-to-4D first renders static 3D scenes matching the text input, then stacks this sequence into a video. This is then optimized to 4D via HexPlane NeRFs.

They are able to generate animated 3D assets and visual effects for video games, and further show that HexPlane is a useful representation for dynamic scene research.

4.6 Neural Radiance Field-based

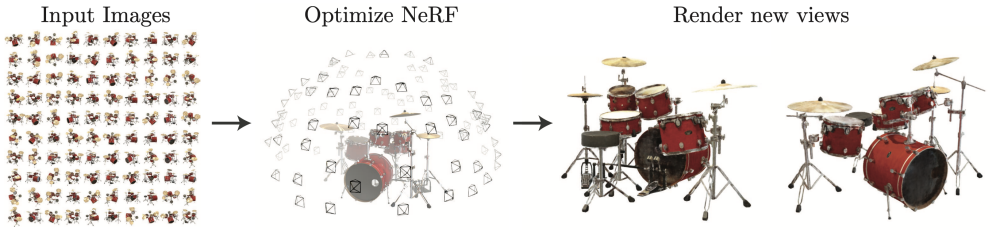

Figure 4.7: **NeRF-based Reconstruction.** Adapted from [2]. Given a set of input images with known camera poses, the NeRF model is optimized to produce novel views of the object. A similar pipeline is followed for the majority of NeRF-based methods presented in this section.

One of the most influential papers in this taxonomy is **NeRF** [2] (Figures 3.7 and 4.7). In 2020, Mildenhall et al. optimized an underlying continuous scene function to enable NVS of complex scenes. Given a partial/sparse set of input scene images and their camera poses, NeRF is able to reconstruct the 3D scene in its entirety. To do so, NeRF casts rays along camera views into the scene and samples spatial coordinates, which are passed into a neural network. This MLP then returns the corresponding volume density and view-dependent radiance of the sampled location. These values are composited back towards the camera and rendered as the RGB-value of the pixel. NeRF's ability to perform NVS was a first for its time, and inspired many other approaches and extensions towards faster volumetric rendering.

Gao [48] et al. recently posed the reconstruction problem as an object generation problem instead, and used multi-view diffusion for 3D content generation in their framework **CAT3D**. Given any number of input images, CAT3D achieves quick generation of entire scenes in under 1 minute, using a parallel sampling strategy while learning a NeRF representation. Their diffusion model is trained specifically for NVS, and generates large sets of synthetic views. They achieve highly consistent novel views of 3D scenes, used as input to a 3D reconstruction pipeline for object

reconstruction and rendering. Despite a wide array of successful generative content, their method is still inconsistent for many-camera setups, leaving it suitable only for tasks of a smaller scale.

Inspired by NeRF and IDR, **NeuS** [49] also represented surfaces as the zero-level set of a neural implicit SDF encoded by a MLP as in IDR; to which they developed a new neural volume rendering method for improved training. They maintain the hierarchical sampling strategies of NeRF. Their network learns the weights of the neural SDF as a density field, with higher density values closer to the object surface. Wang et al.'s algorithm achieved high definition surface reconstruction of objects and scenes given 2D images - compared to NeRF, this approach is especially good at dealing with complex and thin structures, and self-occlusions.

An industry-applicable work combining NeRFs and neural SDF implicit representation was Yang et al.'s **B**uilder of **An**imatable 3D **Mo**dels (**BANMo**) [50]. BANMo learns implicit representations of 3D shape, appearance, and movement of non-rigid subjects from many casual videos broken down into 2D images. To deal with the dynamic nature, neural blend skinning is applied for smoother mesh deformations. Using a dense 2D-to-3D correspondence mapping, BANMo consolidates thousands of unsynchronized images from unknown camera parameters into the same canonical space to create a target object, forgoing the need for a predefined shape template. Motion deformation is made possible via forward and backward mapping from a root position. BANMo successfully models humans and quadrupeds, as well as other creatures as the number of video captures increase. Optimization, however, leaves much to desire and is addressed in their limitations.

Despite NeRF's revolutionary success, its main concern is training and rendering time. **Instant-NGP** [51] accelerated this through a novel multi-resolution hash encoding in the MLP structure confined by only two values: the number of parameters and the final resolution. By mapping a cascade of grids to a corresponding fixed-size array of feature vectors for different resolution levels, Instant-NGP enables a near constant lookup time akin to that of hash tables and is structured to work seamlessly with modern GPU technology. The cascading factor also prevents hash collisions, since different resolution levels contain different values for the same hash. Thus, when aggregating the values of various hash lookups, no collisions occur. Instant-NGP's speed up in both training

and rendering time over NeRF made it a popular adaptation for NeRF-based representations in later research.

One such work utilizing the multi-resolution hash grids is **NeuralAngelo** [22]. Li et al. combined the encoding of Instant-NGP with a coarse-to-fine optimization on the hash grids to better control the different levels of details. The values encoded in these hash grids are input to both a SDF MLP and a second MLP for color prediction, after which the results are composited and volume rendered. By combining the improvements of Instant-NGP while modelling geometry as a SDF (and thus improving on NeuS results), they are able to effectively recover dense 3D surfaces from multiview images, and large-scale scene reconstructions of both indoor and outdoor spaces given RGB videos.

In the world of text-to-3D content generation, **DreamFusion** [52] made a significant impact with their method, using an existing, pre-trained text-to-image 2D diffusion model to generate 3D models parameterized as NeRFs. These results are any-angle viewable, relightable, and can be composited into any 3D environment/scene. No 3D data is required for training, and they use a random camera pose and lighting when rendering. They also introduce a novel Score Distillation Sampling (SDS) loss later used by many works. Despite requiring high computation cost and generally being not very efficient, DreamFusion's impressive results pioneered a chain of directly related research in input-to-3D methods utilizing powerful pre-trained diffusion priors, including Fantasia3D described earlier in Section 4.1.

Three works following DreamFusion's results occur nearly simultaneously: **RealFusion** [53], **Magic3D** [24], and **Make-It-3D** (**MI3D**) [54]. All three methods optimize DreamFusion with a coarse-to-fine model fitting and accelerate with Instant-NGP. RealFusion and MI3D adapt DreamFusion to take single-view image input instead for 3D reconstruction, while Magic3D retains text as input. RealFusion adds texture inversion, fusing the given input views into a NeRF to train. They re-engineer the diffusion prompt from random arguments of the input image for improvement and have no supervision in training. With additional text guidance alongside the image, RealFusion is also able to successfully reconstruct arbitrary objects. Magic3D optimizes the textured 3D mesh

model (extracted via DMTet) with an efficient differentiable renderer and a high-resolution latent diffusion model - achieving much higher-quality 3D meshes in half the computing time of DreamFusion. Added camera close ups help Magic3D recover high-frequency details in geometry and appearance, and fine-tuning their network enables prompt-based editing and a personalized text-to-3D experience. MI3D relaxes the SDS loss constraints, and performs neural texture enhancements with textured point clouds and texture priors in the second stage. They additionally leverage a depth prior to improve single-view depth estimates, and achieve realistic textures in their results. In comparisons with a modified image-input version of DreamFusion, MI3D outperforms the former in texture synthesis when its predecessor suffers from smoothing issues.

Zero 1-to-3 [55] later took inspiration from DreamFusion and its extensions in their method for NVS of a single RGB image from a specified camera viewpoint, achieving strong zero-shot performance on objects despite complex geometries and artistic styles. With the target object parameterized as a voxelized radiance field, Liu et al. exploited the geometry priors of StableDiffusion [56] and conditioned on viewpoint, using a synthetic dataset to control the viewpoint camera. They outperform contemporary SOTA single-view reconstruction methods due to the internet-scale pre-training provided by large-scale diffusion models.

The last in the "dream line" from 2023, **HyperDreamer** [57] used Zero 1-to-3 as a guidance model to generate hyper-realistic 3D content from a single image, parameterizing scenes as NeRFs and extracting the mesh via DMTet's marching tetrahedral algorithm. They introduce a semantic-aware material estimation for improved textures, and produce full-range viewable, renderable, and texture-editable meshes alongside a novel interactive 3D texture editing interface.

Moving into dynamic reconstructions using NeRF-based methods, **Neural Scene Flow Fields** [58] built upon NeRF with a time component and incorporated dense vector fields for forward and backward scene flows to estimate 3D motion. This new representation of 3D geometry, appearance, and motion requires only monocular video and known camera poses, and uses volume tracing to render results. They successfully represent sharp motion continuities, and integrates static scenes

to improve rendering results over a static background. The method showed to be useful for various ITW scenes and successful with complexity such as thin structures and view-dependent effects.

DyNeRF [59] further expanded on dynamic NeRF work, achieving near-photorealistic dynamic 3D NVS. Their method extends NeRFs into the space-time domain by time-conditioning the radiance fields with new, compact latent codes. They additionally implement a coarse-to-fine hierarchy sampling in their video frames, a strategy that later becomes quite commonplace among NVS methods. This strategy exploits time-variant radiance changes, focusing learning efforts on areas of the scene with large variation across time. Naturally, this idea led their neural networks to be trained upon the keyframes of video inputs. Using keyframes for training input, however, creates a coarse model capture of the scene - DyNeRF falls short in synthesizing videos with fast motion, and results for highly dynamic scenes appear blurry.

Liu et al. [60] was interested in accelerating the learning of dynamic scenes in their approach, **DeVRF**. Rather than fully dynamic-based learning, they proposed a static-to-dynamic learning paradigm, in which static 3D learning is used to bias the 4D (3D plus time) dynamic learning. Specifically, they utilize 3D volumetric data from multiview static images to influence a hybridized, 4D voxelized deformation radiance field. They achieved results with quality comparable to contemporary SOTA methods, at 100x faster speeds.

4.7 Gaussian-based

3D GS [3] made a significant impact in visual computing with its new discrete representation of scenes and objects. By representing radiance fields as anisotropic 3D Gaussians and optimizing splatting performance for captured scene rendering, 3D GS is able to achieve real-time, high quality radiance field rendering and NVS. It has quickly replaced NeRF as SOTA in rendering complex static scenes, and its unstructured, GPU-friendly optimization makes it handy for a diverse set of applications. A key part of their high-quality free-view synthesis rendering is attributed to the Gaussian processing - 3D GS clones small Gaussians into missing camera view sections and splits

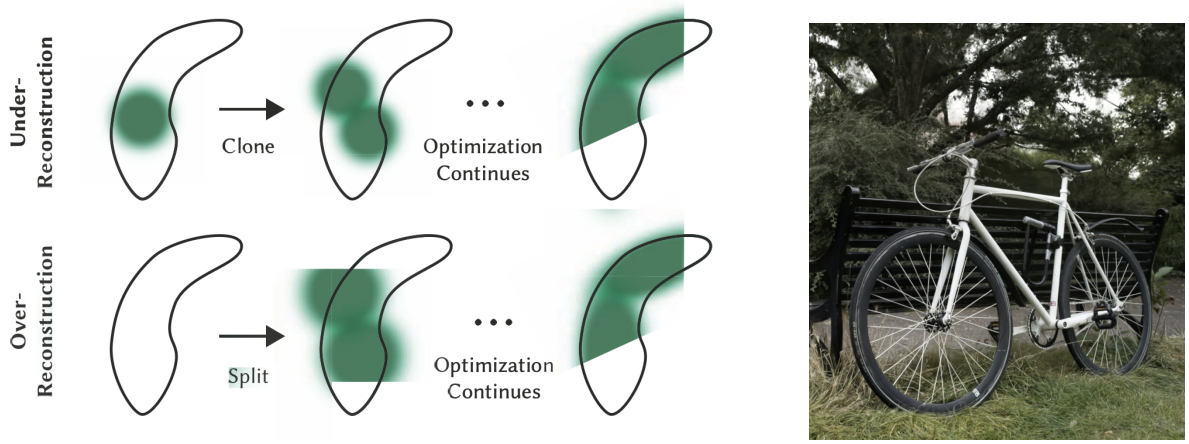

Figure 4.8: **Gaussian-based Reconstruction.** Figure (left) and result image (right) from [3]. The figure illustrates the Gaussian optimization method: if the Gaussian is too small, it is cloned (top), and if it is too large, it is split (bottom).

the Gaussians that are too large (Figure 4.8). This allows for seamless patching across scenes while using a lower number of overall parameters.

Incorporating an implicit differentiable SDF as the input encoding before transforming it to a Gaussian opacity, Lyu et al. achieved accurate 3D Gaussian surface reconstruction (**3DGSR**) with rich details [61]. Gaussian points are bound to the SDF throughout training to ensure geometries are aligned with the Gaussian values (i.e. depth, normals), which optimizes the SDF learning given Gaussian learning results. While achieving highly accurate surface reconstructions, 3DGSR still has a trade-off between quality and smoothness, and areas with complex textures do not perform well with this algorithm.

Kerbl et al. have since built upon their 3D GS work with **Hierarchical 3D Gaussians** [62], introducing a hierarchy for efficient rendering of large data and optimizing visual quality. The hierarchy uses a divide-and-conquer approach based on a Gaussian's level of detail. The primitives are subdivided into chunks for parallel processing, then reconsolidated for the final render. This top-down, tree-based method is real-time efficient even for very large scenes with massive amounts of data, such as street-level scenes spanning thousands of kilometers, and is promising in many applications such as AR/VR, city-wide planning, and other large-scale endeavors.

Another method for scalable generalizable 3D reconstruction includes autoencoding a 3D latent space with variational Gaussian features for fast NVS. **LatentSplat** [63] combines both regression- and generative-based approaches, separating the feature Gaussians into two parts. The variational Gaussians consist of the distribution of all possible Gaussians and encodes varying uncertainty. The semantic Gaussians are specific sampled features from the complete distribution, and used for efficient rendering via splatting and a generative decoder network. The network is trained on readily available video data processed into two-view images, and particularly observes spaces with low variance. LatentSplat achieves better efficiency than their baselines and outperforms other two-view reconstruction methods. There is high detail in object reconstruction, and is cleaner with less artifacts at the scene level.

Given the popularity of DreamFusion and its extensions (Section 4.6), it is no surprise that Gaussian-infused variations were prompted by the success of 3D GS. **DreamGaussian** [64] adapts 3D Gaussians splatting into generative contexts for both text and image input, and significantly reduces 3D content generation time compared to existing methods in lifting the 2D image into 3D space. They also introduce an efficient mesh extraction algorithm from 3D Gaussians, and a texture refinement stage to further improve the generated quality. **GaussianDreamer** [65] focuses solely on text input, bridging 2D and 3D diffusion models with splatting for improved 3D consistency while exploiting the speed that comes with 3D Gaussians. They incorporate noisy point and color perturbation to further enrich output. GaussianDreamer generates realistic and interesting results after training for several minutes - much faster than existing methods such as DreamFusion, Magic3D, and Fantasia3D, showing a promising direction in the fast generation of 3D assets.

As Gaussian splatting has proved to be near real-time, amortizing Gaussian rendering has also become an area of interest. **AGG** [66] produces instant 3D Gaussians from a single image without requiring per-instance optimization; they do so with a coarse-to-fine representation. Their intermediate hybrid representation follows a cascading generative framework, with two different transformers for geometry prediction and texture information. Intermediate layers of the neural network use coarse point-voxel layers, and later upsample resolutions via a Gaussian super-resolution

model for high quality output. Their method remains limited on complex or highly occluded geometries, and initialization of variables is non-trivial and highly impactful.

Shen et al. improved on AGG with **Gamba** [67], a network to model 3D Gaussians from a single image at millisecond speed. Gamba is a Mamba-based network, the latter a new scalable linear-time sequential model [68]. Shen et al. leverages the linear complexity of Mamba's state-space model for amortization and derives a radial mask constraint for network loss based on multiview masking. Their end-to-end feedforward single-view reconstruction pipeline outperforms SOTA when looking at context-dependent reasoning. AGG and Gamba's achievements indicate a promising direction towards instantaneous 3D reconstruction and rendering, with heavy implications towards AR/VR applications.

Luiten et al. extended 3D GS to perform dynamic NVS [69], along with a 6 degree of freedom (DOF) tracking in the dynamic scene. Their **Dynamic 3D Gaussians** consist of Gaussians which move and rotate over time through rigid-body transformations. All other attributes of the Gaussians (i.e. color, opacity, size, etc.) are restricted to persist over time. This achieves high-accuracy, long term point tracking and dynamic reconstruction of scenes. Furthermore, by tracking any particular Gaussian over time, Luiten et al. generate novel views such as first-person view synthesis - highly applicable in AR/VR. Unfortunately, their approach requires an extensive setup with multiple cameras and known angles and captured timesteps, and fails with monocular video.

GaussianBody [70] introduced dynamic, clothed human 3D models from monocular videos using 3D GS as a representation. With point clouds as the mechanism for deformation control, GaussianBody deforms the Gaussians with forward linear blend skinning in each frame guided by an articulated human model. Their approach is limited to the pre-existing model's parameters, and falls short in novel pose synthesis, but presents an interesting opportunity for modelling highly dynamic motions with Gaussians for efficient reconstruction.

On the other hand, Liu [71] recently exploited the advancements of 3D GS to reconstruct high-fidelity, time-consistent meshes from monocular video input in her work, **DG-Mesh**. This method tracks mesh vertices over time to produce textured 3D Gaussian surface reconstruction meshes.

Liu separates the mesh tracking into two parts: a mesh-to-Gaussians and a Gaussians-to-canonical-points translation. In order to yield uniformly distributed Gaussians in each world frame, they introduce a Gaussian-mesh anchoring procedure, creating uniform 3D spatiotemporal points in the output geometry. These anchored Gaussians corresponding to the video input are projected onto the correlating Gaussian at that timestep, correcting and improving the geometry for stability over time. Deformed Gaussians are transformed back into the mesh via explicit geometry reconstruction methods.

4.8 Datasets and APIs

Although outside the time-frame of this taxonomy, it is important to mention the popular ShapeNet [72] dataset from 2015, commonly used in a majority of 3D object training tasks. ShapeNet's three million 3D models are richly-annotated and span 55 common object categories, making it ideal for training neural models requiring 3D input. However, a critical flaw of ShapeNet's data is the lack of variation in its object views, and that ShapeNet designs require professional work to create. And unsurprisingly, the need for diverse and varied data has only increased since ShapeNet's inception. Thus, we include **Objaverse-XL** [73] in this taxonomy, created in 2023 from web crawling a variety of 3D asset sources. It is currently the largest and most diverse dataset of deduplicated 3D assets, with over 10 million rendered 3D files each with various associated metadata. This makes Objaverse-XL 2 orders of magnitude larger than ShapeNet. Their scale in size and diversity has achieved strong zero-shot generation abilities in testing, and furthers the field of generative 3D graphics by a large margin.

Given the number of NeRF-based methods published in recent years, we also include **NeRF-Studio** [28] in this taxonomy. An API created in 2023 for implementing NeRF-based methods, the plug-and-play compatibility of this modular PyTorch framework supports extensive real-time visualization tools and has pipelines for importing captured ITW data. The program also has tools to export results to mesh, point cloud, and video formats. It remains an open-source project, and

is released under the Apache2 license. Its creation can facilitate more research and learning of NeRF-based methods, and enable users to combine components from multiple approaches to find different combinations.

Chapter 5

The Taxonomies

There are a variety of ways to interpret and understand the methods present in 3D reconstruction. Exploration of literature in Chapter 4 was sectioned by representation primitive to best understand the mechanisms of 3D reconstruction, accompanied with a tree-like figure in rough chronological ordering. While not a taxonomy itself, Figure 4.1 may prove useful for understanding recent and potential trends. We now present our taxonomies, each of which are rooted in a different research objective. We justify the choices for each view, and suggest usage interpretations for the user. All colorings of the methods' nodes retain the same purpose as in Figure 4.1 and denote the method's main choice of 3D representation. Child nodes inherit the parent node's categorizations.

5.1 Input-based Hierarchy

We consider the input-based classification to be a fundamental categorization. It is the main method in prior taxonomies [4], [5], [10], [11], particularly with single- and/or multi-image based reconstruction tasks. Therefore, it would be remiss to perform a taxonomy without this widely-used sorting method. Figure 5.1 depicts the taxonomy. We further categorize the image and video inputs into single- and/or multiple-count inputs, and those further yet into monocular or multiple-camera setups. Additionally, some approaches have also been tested to work well with unconstrained, ITW input - a non-trivial accomplishment. These are categorized in an additional depth level for

distinction. Methods within nodes are organized by geometry representation, and not any particular performance statistic.

5.2 Output-based Hierarchy

In the output-based classification, we present a hierarchy delineating the methods which achieve desired goals and outputs. This format logically makes sense as a sort of "reverse-engineered classification", but to the best of our knowledge is not included in taxonomies. Perhaps it is implicit; however, we believe that an explicit visualization showing this relationship provides a more comprehensive understanding for those unfamiliar with this field - after all, that is the goal of taxonomy work. Users may additionally use this variation in conjunction with the input-based hierarchy to form a better end-to-end understanding of these methods.

To that end, Figure 5.2 depicts the output-based taxonomy. The initial hierarchy level is based on the goal of the output - the achievements of the literature in this taxonomy can be broadly categorized into either surface reconstruction or NVS. Surface reconstruction methods are further sorted into their output geometry representations, and NVS methods sorted into either static or dynamic synthesis results. Additionally, several methods in the NVS sub-hierarchy are categorized with an "extractable" primitive. Contrary to the surface reconstruction-oriented methods, these methods do not acquire the explicit geometry reconstruction as the main achievement and use isosurface extraction algorithms such as Marching Cubes [74] to retrieve them. Similarly to the input-based taxonomy, methods within nodes are organized by geometry representation and not by any particular performance statistic.

5.3 Scale-based Hierarchy

The scale-based classification is dependent on a method's suitability towards two aspects: level of detail in reconstruction, and level of scale achievable. It is widely acknowledged that coarser, lesser detailed objects (i.e. smooth, man-made surfaces) are easier to reconstruct and/or generate than

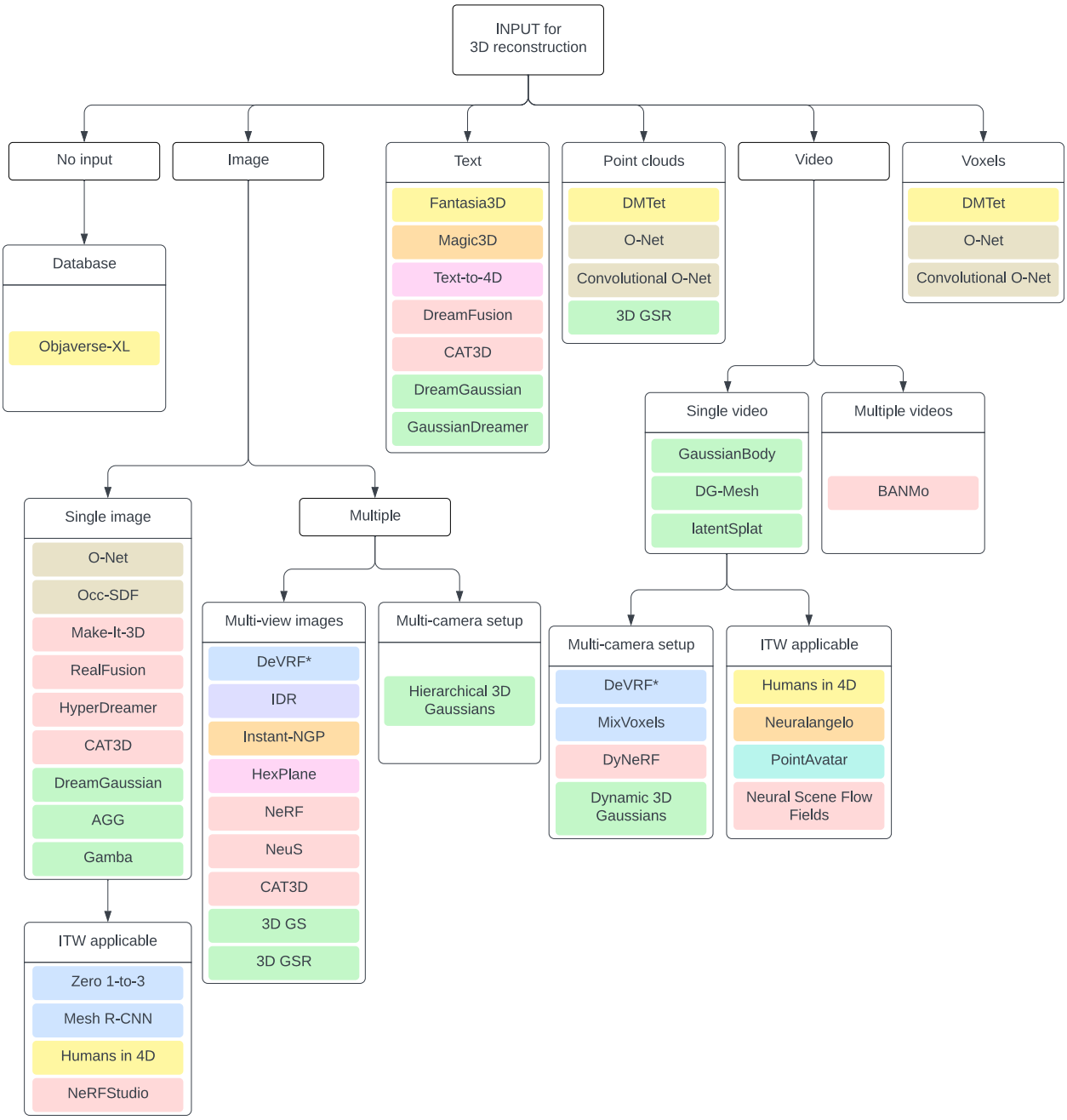

Figure 5.1: **Input-based Taxonomy.** The input-based categorization is a classic taxonomy structuring. Approaches with multiple input types are replicated under their respective input branches. Titles marked with an asterisk (*) indicate that the method requires input consisting of *all* other input types the title is categorized under.

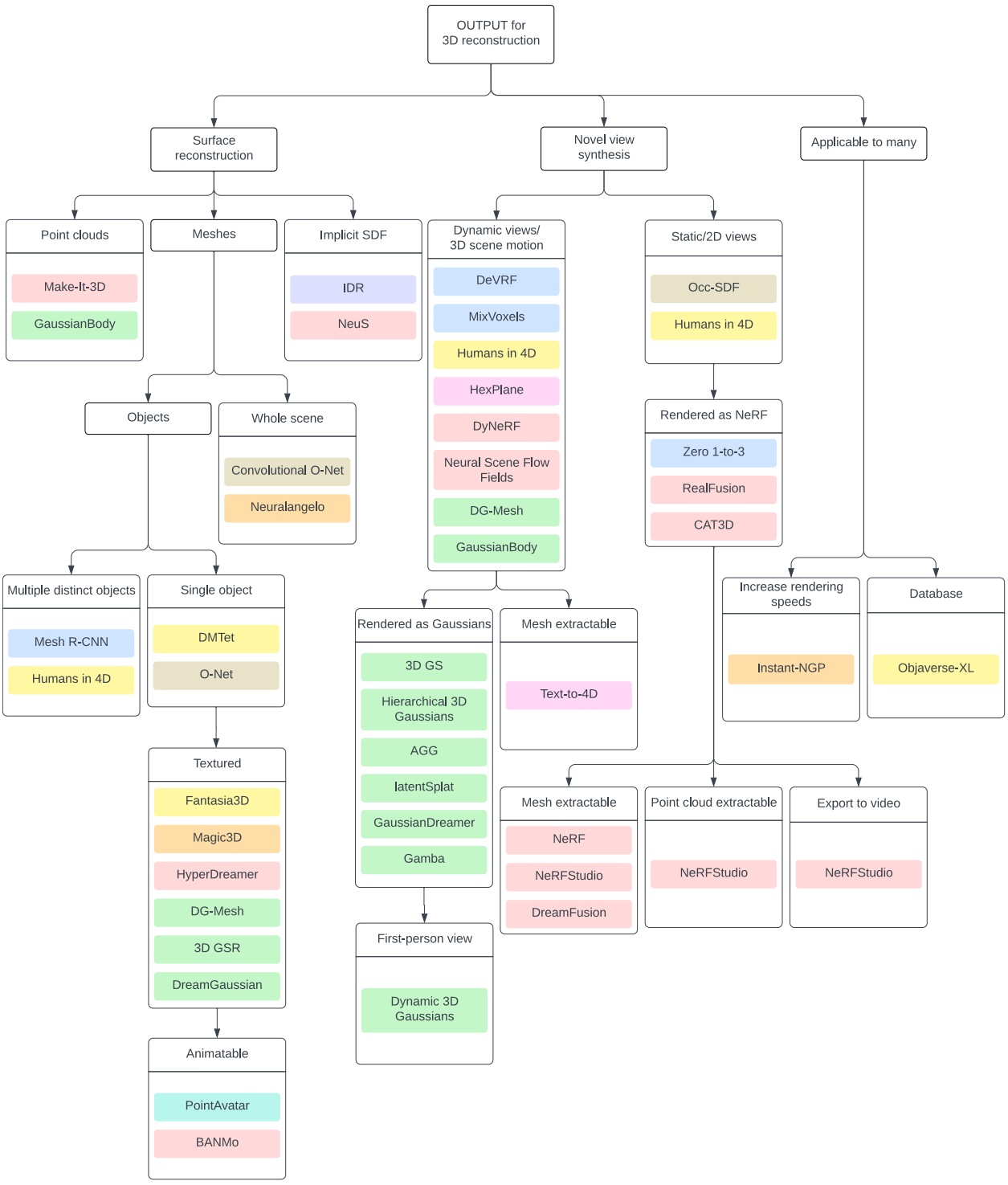

Figure 5.2: **Output-based Taxonomy.** The output-based taxonomy is first organized on the methods' output goal and secondly on their respective output geometry representation. 'Extractable' primitive categories indicate the NVS methods whose results can further be processed with existing surface extraction algorithms to obtain the respective 3D geometry.

those with very detailed, complex shapes (i.e. organic materials and photorealistic details). In fact, many techniques in generative 3D reconstruction work quite well for synthetic, human-developed objects, while there is still a lack of techniques that produce adequate results on natural materials such as animal fur and wings. It is also interesting to note that different scales of scenes (i.e. rooms versus cityscapes) currently require different techniques to best capture both level of detail and size, and results from generalizing a single method still fall short of the more specialized.

Figure 5.3 presents the scale-based taxonomy. The hierarchy order is structured such that performing a depth-first search from left to right produces an ordering of least to most complex and smallest to largest scale. We split the static and dynamic reconstruction methods into separate trees, as the methods achieve very different outcomes. Complexity and scale is then measured individually for static methods and for dynamic methods. Unlike the two previous taxonomies, methods contained within nodes ***are*** listed in order of scale and complexity. The methods closer to the left or to the root of the tree have less accuracy in detail, or handle less complex or smaller scaled scenes than those in a rightmost branch or towards the bottom of a node.

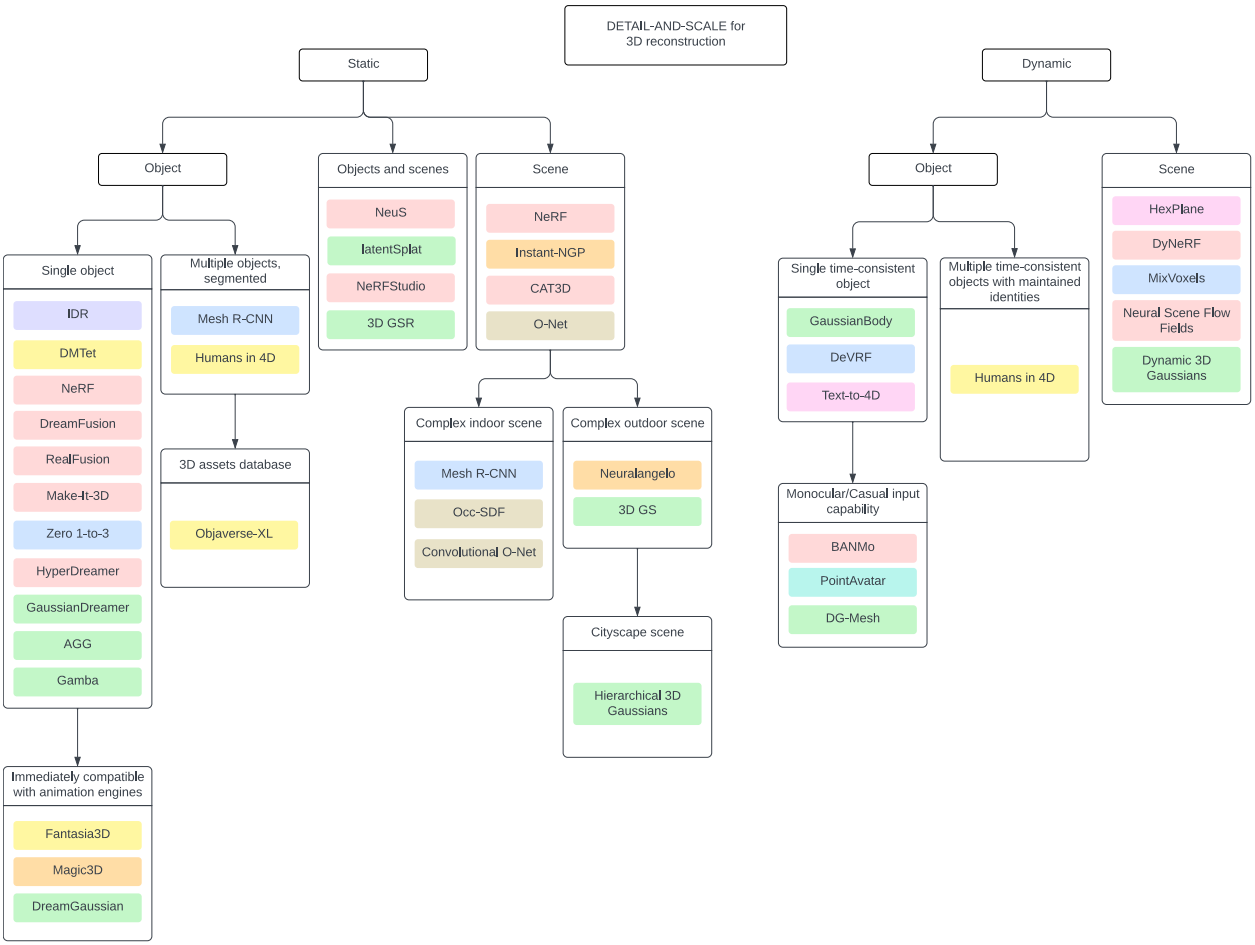

Figure 5.3: **Scale-based Taxonomy.** The scale-based taxonomy is rooted in the methods' suitability in scale of reconstruction. We separate the methods into static and dynamic reconstruction scale and sort each tree individually. The ordering from least to most complex/large in scale is obtained with a top-down depth-first search prioritizing leftmost nodes on each tree, respectively.

Chapter 6

Conclusion

This master's thesis has multi-dimensionally discussed the current state of 3D object and scene recovery from varying input sources. We summarize the details and methods of recent 3D reconstruction research advancements from within the past five years, categorizing literature on primitive representation. The collection is then organized and curated into three tree-like taxonomy visualizations rooted in method input, method output, and reconstruction scale, respectively. Utilizing a tree structure for a hierarchy lends itself to the idea of inheritance, and creates distinctions between different methods. We believe this multi-factored classification facilitates a more complete understanding of the field of 3D reconstruction, and may inspire additional innovation and improvement. These "multi-view" taxonomies further allow the reader to determine which categorization best fits their usage criteria. We conclude this thesis with a discussion on future trends of related applications and ideas for future iterations on this work.

6.1 Discussion

The addition of AI/ML into 3D reconstruction is still in its relatively early stages, and as such there exist many more avenues of research that can be pursued to better this field. As a result, the field of 3D reconstruction appears to be heavily influenced by AI/ML at the moment. However, as researchers continue to improve the ML techniques involved in 3D reconstruction, we believe the

field will begin shifting towards an emphasis in computer graphics instead, taking advantage of the sheer amount of untapped potential in highly graphical applications.

Augmented/Virtual Reality With the steady rise of consumer-level AR/VR across the industry, advancements in 3D reconstruction will provide consumers with higher resolution and more photorealistic results. NVIDIA's Omniverse is able to incorporate all inputs which use Pixar Studios' Universal Scene Description (USD) format [75]. This creates an environment in which 3D scenes are scalable and interchangeable, regardless of the different sources, assets, and animations. In medical research, Anatomage [76] provides photoreal human cadaver dissection simulations. These models must be as accurate as possible in order to foster high quality training.

3D GS has shown to be influential for real-time high-definition rendering, and current trends in 3D Gaussian research are showing promise in taking a large step towards consistent high resolution results. Additionally, recent efforts to combine 3D GS and other reconstruction techniques with existing representations such as meshes [77] further encourages the advancement of techniques relevant to the AR/VR community and computer graphics as a whole, where meshes still dominate the computer graphics pipeline.

Animal Reconstruction The majority of reconstructive work surrounding "sentient beings" has focused on human reconstruction, and rightly so. However, there are several research efforts as well in animal reconstruction, and accuracy in the 3D modelling and rendering of the millions of species we coexist with is highly applicable in many fields, including but not limited to: animal husbandry, veterinary medicine, zoology and ecological studies, evolutionary and extinction preservation, and general entertainment.

Several works [78], [79], [80] have focused on and found success with shape, pose, and texture estimation of animals, but rely heavily on artist generated models and fall short in comparison to the advancements made in human reconstruction. Others have focused on "man's best friend" and developed end-to-end frameworks for shape and pose estimation of dogs [81], [82], [83]. However, the vast number of dog species and their various unique traits make this a non-trivial and

non-generalizable task. This is even more so when considering the sheer diversity of the animal kingdom as a whole. Portraying non-mesh textures and extremely fine details (fur, hair, feather, etc.) is an additional area of research that still remains unexplored. Achieving photorealism in near real-time appears to be a distant goal on top of it all, leaving much to be worked on.

Text-to-3D Video The rising popularity of large language models (LLMs) urged us to include several text-to-3D generative models in this taxonomy, dubbed the "dream line." Naturally, the future direction of these research endeavors lies in its dynamic counterparts. Technologies in text-to-3D video (TTV) have already emerged in industry and more are being developed: OpenAI's Sora [84], Luma's Dream Machine [85], and Kuaishou's Kling [86] are just a few names to consider. Interestingly, neither Sora nor Kling are publicly available software at the time of this writing; however, Kling's ability to synthesize longer videos than Sora while following the latter's technical route already makes it a competitor upon release. As when LLMs first became publicly available, commercialized TTV models will likely face similar ethics decisions to ensure training data is accurately representative and diverse.

6.2 Future Work

Taxonomies are always an on-going process, and we encourage any interested reader to update these visuals as deemed appropriate - this work is limited in scope and by no means a complete edition. As the field of 3D reconstruction remains a very hot topic and at the forefront of visual computing and computer graphics research, a live version of the results of this taxonomy will be transferred online to c-chang.github.io. It will continue to be updated as new research on 3D reconstruction appears. An interactive interface is also being developed.

References

1. Man, D. & Olchawa, R. in, 30–37 (2018). doi:10.1007/978-3-319-75025-5_4.
2. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. & Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *CoRR* **abs/2003.08934** (2020).
3. Kerbl, B., Kopanas, G., Leimkühler, T. & Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* **42** (2023).
4. Han, X.-F., Laga, H. & Bennamoun, M. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence* **43,** 1578–1604 (2019).
5. Khan, M. S. U., Pagani, A., Liwicki, M., Stricker, D. & Afzal, M. Z. Three-Dimensional Reconstruction from a Single RGB Image Using Deep Learning: A Review. *Journal of Imaging* **8,** 225 (2022).
6. Phang, J. T. S., Lim, K. H. & Chiong, R. C. W. A review of three dimensional reconstruction techniques. *Multimedia Tools and Applications* **80,** 17879–17891 (2021).
7. Huang, Z., Wen, Y., Wang, Z., Ren, J. & Jia, K. *Surface Reconstruction from Point Clouds: A Survey and a Benchmark* 2022.
8. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., *et al.* Neural Fields in Visual Computing and Beyond. *Computer Graphics Forum.* doi:10.1111/cgf.14505 (2022).
9. Gao, K., Gao, Y., He, H., Lu, D., Xu, L. & Li, J. *NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review* 2023.
10. Chen, G. & Wang, W. *A Survey on 3D Gaussian Splatting* 2024.
11. Vinodkumar, P. K., Karabulut, D., Avots, E., Ozcinar, C. & Anbarjafari, G. Deep Learning for 3D Reconstruction, Augmentation, and Registration: A Review Paper. *Entropy* **26,** 235 (2024).
12. Raghavachary, S. *Rendering for Beginners: Image Synthesis Using RenderMan* (Focal Press, 2004).
13. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. & Stuetzle, W. *Mesh optimization* in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), 19–26.
14. Microsoft Corporation. *Minecraft* Accessed: 2024-07-06. https://www.minecraft.net/.
15. Liu, Z., Tang, H., Lin, Y. & Han, S. *Point-Voxel CNN for Efficient 3D Deep Learning* 2019.
16. Nießner, M., Zollhöfer, M., Izadi, S. & Stamminger, M. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* **32,** 1–11 (2013).
17. Liu, L., Gu, J., Lin, K. Z., Chua, T.-S. & Theobalt, C. Neural Sparse Voxel Fields. *NeurIPS* (2020).

18. Hedman, P., Srinivasan, P. P., Mildenhall, B., Barron, J. T. & Debevec, P. Baking Neural Radiance Fields for Real-Time View Synthesis. *ICCV* (2021).
19. Pixar Studios. *The Brick Map Geometric Primitive* Accessed: 2024-07-06. https://renderman.pixar.com/resources/RenderMan_20/brickmapgprim.html.
20. Shi, Z., Peng, S., Xu, Y., Geiger, A., Liao, Y. & Shen, Y. *Deep Generative Models on 3D Representations: A Survey* 2023.
21. Kelly, C., Hu, L., Hu, J., Tian, Y., Yang, D., Yang, B., *et al. VisionGPT-3D: A Generalized Multimodal Agent for Enhanced 3D Vision Understanding* 2024.
22. Li, Z., Müller, T., Evans, A., Taylor, R. H., Unberath, M., Liu, M.-Y., *et al. Neuralangelo: High-Fidelity Neural Surface Reconstruction* in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 8456–8465. doi:10.1109/CVPR52729.2023.00817.
23. Zheng, Y., Yifan, W., Wetzstein, G., Black, M. J. & Hilliges, O. *PointAvatar: Deformable Point-based Head Avatars from Videos* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
24. Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., *et al. Magic3D: High-Resolution Text-to-3D Content Creation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 300–309.
25. Chen, R., Chen, Y., Jiao, N. & Jia, K. *Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 22246–22256.
26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. & Geiger, A. *Occupancy networks: Learning 3d reconstruction in function space* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), 4460–4470.
27. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M. & Cui, Z. *DIST: Rendering Deep Implicit Signed Distance Function with Differentiable Sphere Tracing* 2020.
28. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., *et al. Nerfstudio: A modular framework for neural radiance field development* in *ACM SIGGRAPH 2023 Conference Proceedings* (2023), 1–12.
29. Haque, A., Tancik, M., Efros, A., Holynski, A. & Kanazawa, A. Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions (2023).
30. Kommareddy, S., Siripun, J. & Sum, J. *3D Object Morphing with Metaballs* 2014.
31. Mortenson, M. E. in. Chap. 5 (John Wiley & Sons, Inc., 1997).
32. Catmull, E. & Clark, J. in *Seminal graphics: pioneering efforts that shaped the field* 183–188 (1998).
33. Ma, W. Subdivision surfaces for CAD—an overview. *Computer-Aided Design* **37,** 693–709 (2005).
34. DeRose, T., Kass, M. & Truong, T. in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2* 801–810 (2023).
35. Levoy, M. & Hanrahan, P. *Light field rendering* in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (Association for Computing Machinery, 1996), 31–42. doi:10.1145/237170.237199.
36. Zhu, H., Zhou, Z., Yang, R. & Yu, A. Discrete particle simulation of particulate systems: a review of major applications and findings. *Chemical Engineering Science* **63,** 5728–5770 (2008).

37. Tatarchenko, M., Richter, S. R., Ranftl, R., Li, Z., Koltun, V. & Brox, T. *What do single-view 3d reconstruction networks learn?* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), 3405–3414.
38. Goel, S., Pavlakos, G., Rajasegaran, J., Kanazawa*, A. & Malik*, J. *Humans in 4D: Reconstructing and Tracking Humans with Transformers* in *International Conference on Computer Vision (ICCV)* (2023).
39. Shen, T., Gao, J., Yin, K., Liu, M.-Y. & Fidler, S. *Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis* in *Advances in Neural Information Processing Systems (NeurIPS)* (2021).
40. Gkioxari, G., Malik, J. & Johnson, J. *Mesh r-cnn* in *Proceedings of the IEEE/CVF international conference on computer vision* (2019), 9785–9795.
41. Wang, F., Tan, S., Li, X., Tian, Z., Song, Y. & Liu, H. *Mixed Neural Voxels for Fast Multi-view Video Synthesis* in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 19649–19659. doi:10.1109/ICCV51070.2023.01805.
42. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M. & Geiger, A. *Convolutional Occupancy Networks* in *European Conference on Computer Vision (ECCV)* (2020).
43. Li, T., Bolkart, T., Black, M. J., Li, H. & Romero, J. Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph.* **36,** 194–1 (2017).
44. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., *et al.* Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems* **33** (2020).
45. Lyu, X., Dai, P., Li, Z., Yan, D., Lin, Y., Peng, Y., *et al. Learning a Room with the Occ-SDF Hybrid: Signed Distance Function Mingled with Occupancy Aids Scene Representation* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 8940–8950.
46. Cao, A. & Johnson, J. *HexPlane: A Fast Representation for Dynamic Scenes* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 130–141.
47. Singer, U., Sheynin, S., Polyak, A., Ashual, O., Makarov, I., Kokkinos, F., *et al.* Text-To-4D Dynamic Scene Generation. *arXiv:2301.11280* (2023).
48. Gao*, R., Holynski*, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P. P., *et al.* CAT3D: Create Anything in 3D with Multi-View Diffusion Models. *arXiv* (2024).
49. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T. & Wang, W. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS* (2021).
50. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A. & Joo, H. *BANMo: Building Animatable 3D Neural Models From Many Casual Videos* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 2863–2873.
51. Müller, T., Evans, A., Schied, C. & Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* **41,** 1–15 (2022).
52. Poole, B., Jain, A., Barron, J. T. & Mildenhall, B. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).
53. Melas-Kyriazi, L., Laina, I., Rupprecht, C. & Vedaldi, A. *Realfusion: 360deg reconstruction of any object from a single image* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), 8446–8455.

54. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., *et al. Make-It-3D: High-fidelity 3D Creation from A Single Image with Diffusion Prior* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 22819–22829.
55. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S. & Vondrick, C. *Zero-1-to-3: Zero-shot One Image to 3D Object* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 9298–9309.
56. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. *High-Resolution Image Synthesis With Latent Diffusion Models* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 10684–10695.
57. Wu, T., Li, Z., Yang, S., Zhang, P., Pan, X., Wang, J., *et al. HyperDreamer: Hyper-Realistic 3D Content Generation and Editing from a Single Image* in (2023).
58. Li, Z., Niklaus, S., Snavely, N. & Wang, O. *Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes* in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 6494–6504. doi:10.1109/CVPR46437.2021.00643.
59. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., *et al. Neural 3d video synthesis from multi-view video* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 5521–5531.
60. Liu, J.-W., Cao, Y.-P., Mao, W., Zhang, W., Zhang, D. J., Keppo, J., *et al.* DeVRF: Fast Deformable Voxel Radiance Fields for Dynamic Scenes. *Neural Information Processing Systems (NeurIPS)* (2022).
61. Lyu, X., Sun, Y.-T., Huang, Y.-H., Wu, X., Yang, Z., Chen, Y., *et al. 3DGSR: Implicit Surface Reconstruction with 3D Gaussian Splatting* 2024.
62. Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A. & Drettakis, G. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics* **44** (2024).
63. Wewer, C., Raj, K., Ilg, E., Schiele, B. & Lenssen, J. E. *latentSplat: Autoencoding Variational Gaussians for Fast Generalizable 3D Reconstruction* 2024.
64. Tang, J., Ren, J., Zhou, H., Liu, Z. & Zeng, G. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv preprint arXiv:2309.16653* (2023).
65. Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., *et al. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), 6796–6807.
66. Xu, D., Yuan, Y., Mardani, M., Liu, S., Song, J., Wang, Z., *et al. AGG: Amortized Generative 3D Gaussians for Single Image to 3D* 2024.
67. Shen, Q., Wu, Z., Yi, X., Zhou, P., Zhang, H., Yan, S., *et al. Gamba: Marry Gaussian Splatting with Mamba for single view 3D reconstruction* 2024.
68. Gu, A. & Dao, T. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* 2024.
69. Luiten, J., Kopanas, G., Leibe, B. & Ramanan, D. *Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis* in *3DV* (2024).
70. Li, M., Yao, S., Xie, Z. & Chen, K. *GaussianBody: Clothed Human Reconstruction via 3d Gaussian Splatting* 2024.
71. Liu, I., Su, H. & Wang, X. Dynamic Gaussians Mesh: Consistent Mesh Reconstruction from Monocular Videos (2024).

72. Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., *et al.* Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).
73. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., *et al. Objaverse-XL: A Universe of 10M+ 3D Objects* 2023.
74. Lorensen, W. E. & Cline, H. E. in *Seminal graphics: pioneering efforts that shaped the field* 347–353 (1998).
75. *Universal Scene Description (USD) 3D Framework — NVIDIA* Accessed: 2024-07-06. https://www.nvidia.com/en-us/omniverse/usd/.
76. Anatomage Incorporated. *Anatomage* Accessed: 2024-07-06. https://anatomage.com/.
77. Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., *et al. MeshAnything: Artist-Created Mesh Generation with Autoregressive Transformers* 2024.
78. Zuffi, S., Kanazawa, A., Jacobs, D. & Black, M. J. *3D Menagerie: Modeling the 3D Shape and Pose of Animals* in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2017).
79. Kanazawa, A., Tulsiani, S., Efros, A. A. & Malik, J. *Learning Category-Specific Mesh Reconstruction from Image Collections* in *ECCV* (2018).
80. Zuffi, S., Kanazawa, A., Berger-Wolf, T. & Black, M. J. *Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images "In the Wild"* in *The IEEE International Conference on Computer Vision (ICCV)* (2019).
81. Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A. & Cipolla, R. *Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop* in *ECCV* (2020).
82. Rüegg, N., Zuffi, S., Schindler, K. & Black, M. J. Barc: Breed-augmented regression using classification for 3d dog reconstruction from images. *International Journal of Computer Vision* **131,** 1964–1979 (2022).
83. Rüegg, N., Tripathi, S., Schindler, K., Black, M. J. & Zuffi, S. *BITE: Beyond Priors for Improved Three-D Dog Pose Estimation* in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (2023), 8867–8876.
84. OpenAI. *Sora — OpenAI* Accessed: 2024-07-06. https://openai.com/index/sora/.
85. Luma Labs. *Luma Dream Machine* Accessed: 2024-07-06. https://lumalabs.ai/dream-machine.
86. Kuaishou. *Kling-AI — Kuaishou Sora-Like Text-to-Video Generation Model* Accessed: 2024-07-06. https://kling-ai.com.