Computational Visual Media
https://doi.org/10.1007/s41095-024-0436-y Vol. 10, No. 4, August 2024, 613–642

Review Article

Recent advances in 3D Gaussian splatting

Tong Wu[1], Yu-Jie Yuan[1], Ling-Xiao Zhang[1], Jie Yang[1], Yan-Pei Cao[2,3], Ling-Qi Yan[4], and Lin Gao[1] (✉)

Abstract The emergence of 3D Gaussian splatting (3DGS) has greatly accelerated rendering in novel view synthesis. Unlike neural implicit representations like neural radiance fields (NeRFs) that represent a 3D scene with position and viewpoint-conditioned neural networks, 3D Gaussian splatting utilizes a set of Gaussian ellipsoids to model the scene so that efficient rendering can be accomplished by rasterizing Gaussian ellipsoids into images. Apart from fast rendering, the explicit representation of 3D Gaussian splatting also facilitates downstream tasks like dynamic reconstruction, geometry editing, and physical simulation. Considering the rapid changes and growing number of works in this field, we present a literature review of recent 3D Gaussian splatting methods, which can be roughly classified by functionality into 3D reconstruction, 3D editing, and other downstream applications. Traditional point-based rendering methods and the rendering formulation of 3D Gaussian splatting are also covered to aid understanding of this technique. This survey aims to help beginners to quickly get started in this field and to provide experienced researchers with a comprehensive overview, aiming to stimulate future development of the 3D Gaussian splatting representation.

Keywords 3D Gaussian splatting (3DGS); radiance field; novel view synthesis; 3D editing; scene generation

1 Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: T. Wu, wutong19s@ict.ac.cn; Y.-J. Yuan, yuanyujie@ict.ac.cn; L.-X. Zhang, zhanglingxiao@ict.ac.cn; J. Yang, yangjie01@ict.ac.cn; L. Gao, gaolin@ict.ac.cn (✉).

2 Tencent AI Lab, Beijing 100089, China. E-mail: caoyanpei@gmail.com.

3 VAST, Beijing 100000, China.

4 Department of Computer Science, University of California, Santa Barbara, CA 93106, USA. E-mail: lingqi@cs.ucsb.edu.

Manuscript received: 2024-03-07; accepted: 2024-04-24

1 Introduction

With the development of virtual reality and augmented reality, the demand for realistic 3D content is increasing. Traditional 3D content creation methods include 3D reconstruction from scanner data or multi-view images and 3D modeling with professional software. However, traditional 3D reconstruction methods are likely to produce less faithful results due to imperfect capture and noise in camera estimation. 3D modeling methods yield realistic 3D content but require professional user training and interaction, which is time-consuming.

To create realistic 3D content automatically, neural radiance fields (NeRFs) [1] model a 3D scene's geometry and appearance with a density field and a color field, respectively. NeRFs greatly improved the quality of novel view synthesis results but still suffered from low training and rendering speeds. While significant effort [2–4] has been made to accelerate NeRFs to facilitate their application using common devices like cellphones and laptops, it is still hard to find a robust method that can both train a NeRF quickly enough (under 1 h) on a consumer-level GPU and render a 3D scene at an interactive frame rate (about 30 FPS) on common devices. To resolve the speed issues, 3D Gaussian splatting (3DGS) [5] rasterizes a set of Gaussian ellipsoids to approximate the appearance of a 3D scene, which not only achieves comparable novel view synthesis quality but also allows fast convergence (in about 30 min) and real-time rendering (at least 30 FPS) at 1080p resolution, making low-cost 3D content creation and real-time applications possible.

Based on the 3D Gaussian splatting representation, a number of research works have come out and more are on the way. To help readers become familiar with

3D Gaussian splatting quickly, we survey 3D Gaussian splatting, covering both traditional splatting methods and recent neural-based 3DGS methods. Two existing literature reviews [23, 24] summarize recent work on 3DGS; these also serve as good references for this field. As Fig. 1 shows, we divide the surveyed work into three categories according to functionality. We first introduce how 3D Gaussian splatting allows realistic scene reconstruction in various scenarios (Section 2); we further present scene editing techniques using 3D Gaussian splatting (Section 3) and how 3D Gaussian splatting makes downstream applications like digital humans possible (Section 4). Finally, we summarize recent research on 3D Gaussian splatting at a higher level and suggest future work remaining to be done in this field (Section 5). A timeline of representative works can be found in Fig. 2.

2 Gaussian splatting for 3D reconstruction

2.1 Point-based rendering

The point-based rendering technique aims to generate realistic images by rendering a set of discrete geometry primitives. Grossman and Dally [25] proposed the point-based rendering technique based on a purely point-based representation, where each point only influences one pixel on the screen. Instead of rendering points, Zwicker et al. [26] proposed to render splats (ellipsoids). Each splat can occupy multiple pixels; their mutual overlap can generate hole-free images more easily than a purely point-based representation. Later, a series of splatting methods aim to enhance the basic approach by: introducing a texture filter for anti-aliased rendering [27], improving rendering efficiency [28, 29], and resolving discontinuous shading [30]. For more details of traditional point-based rendering techniques, please refer to Ref. [31].

Traditional point-based rendering methods focus on how to render results with high quality from a given geometry. With the development of recent implicit representations [32–34], researchers have started to explore point-based rendering using the neural implicit representation without any given geometry for the 3D reconstruction task. One representative work introduces NeRFs [1], which model geometry

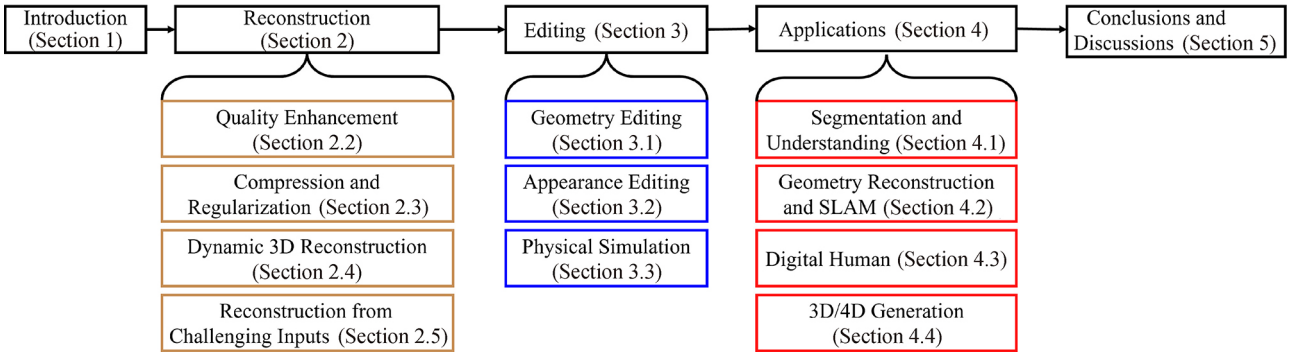


Fig. 1 Structure of our literature review and taxonomy of current 3D Gaussian splatting methods.

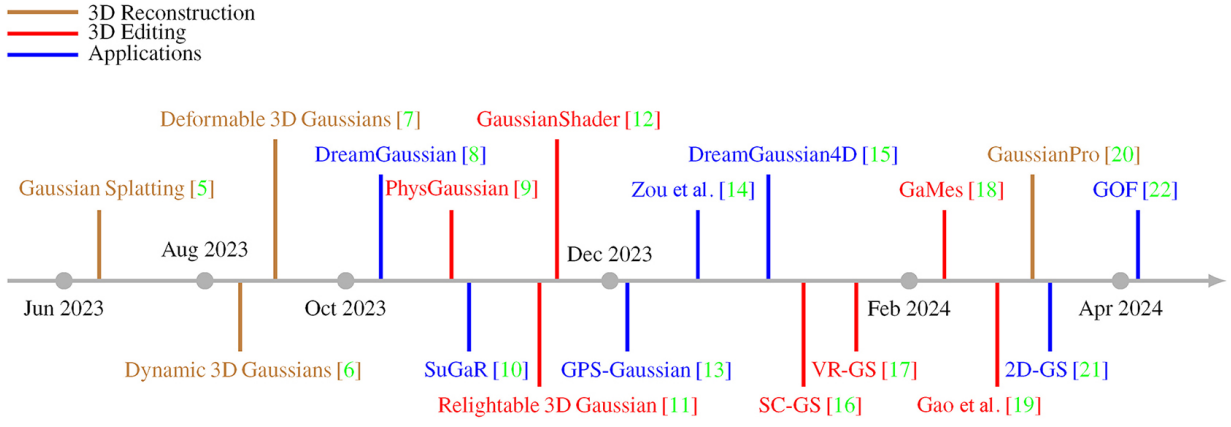


Fig. 2 A brief timeline of representative work using the 3D Gaussian splatting representation.

with an implicit density field and predicts view-dependent color c_i with another appearance field. Point-based rendering combines all sample points' colors along a camera ray to produce a pixel color C using:

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

where N is the number of sample points along a ray and $\alpha_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ is the view-dependent color and opacity for the ith point on the ray. σ_j is the jth point's density. $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$ is the accumulated transmittance. To ensure high-quality rendering, NeRFs [1] typically require 128 sample points along a single ray, which unavoidably takes a long time to train and render.

To speed up both training and rendering, instead of predicting density values and colors for all sample points with neural networks, 3D Gaussian splatting [5] abandons the neural network and directly optimizes Gaussian ellipsoids to which attributes like position P, rotation R, scale S, opacity α, and spherical harmonic (SH) coefficients representing view-dependent color are attached. The pixel color is determined by Gaussian ellipsoids projected onto it from a given viewpoint. The projection of 3D Gaussian ellipsoids can be formulated as

$$\Sigma' = JW\Sigma W^{\mathrm{T}} J^{\mathrm{T}} \tag{2}$$

where Σ' and $\Sigma = RSS^{\mathrm{T}}R^{\mathrm{T}}$ are the covariance matrices for 3D Gaussian ellipsoids and projected Gaussian ellipsoids on the 2D image from a viewpoint with viewing transformation matrix W. J is the Jacobian matrix for the projective transformation. 3DGS shares a similar rendering process with NeRFs but there are two major differences:

(1) 3DGS models opacity values directly while NeRF transforms density values to opacity values.
(2) 3DGS uses rasterization-based rendering which does not require sampling points while NeRF requires dense sampling in 3D space.

Without sampling points and querying the neural network, 3DGS is extremely fast and achieves about 30 FPS on a common device with comparable rendering quality to NeRFs.

2.2 Quality enhancement

Though producing high-quality reconstruction results, improvement is still possible to 3DGS 's rendering. In Mip-Splatting [35], it is observed that changing the sampling rate, for example, the focal length, can greatly influence the quality of rendered images, by introducing high-frequency Gaussian shape-like artifacts or strong dilation effects. To eliminate the high-frequency Gaussian shape-like artifacts, Mip-Splatting [35] constrains the frequency of the 3D representation to be below half the maximum sampling frequency determined by the training images. In addition, to avoid the dilation effects, it introduces another 2D mip filter to the projected Gaussian ellipsoids to approximate a box filter, similar to that in EWA-splatting [27]. MS3DGS [36] also aims to solve the aliasing problem in the original 3DGS and introduces a multi-scale Gaussian splatting representation; when rendering a scene at a new resolution, it selects Gaussians from different scales to produce aliasing-free images. Analytic-Splatting [37] approximates the cumulative distribution function of Gaussians with a logistic function to better model each pixel's intensity response for anti-aliasing. SA-GS [38] utilizes an adaptive 2D low-pass filter at inferencing time according to the rendering resolution and camera distance.

Apart from the aliasing problem, the capability of rendering view-dependent effects also needs to be improved. To produce more faithful view-dependent effects, VDGS [39] proposes to model the 3DGS to represent 3D shapes and predict attributes like view-dependent color and opacity with a NeRF-like neural network instead of the spherical harmonic coefficients in the original 3DGS. Scaffold-GS [40] initializes a voxel grid and attaches learnable features to each voxel point; all attributes of Gaussians are determined from interpolated features and lightweight neural networks. Octree-GS [41] takes this further and uses a level-of-detail strategy to better capture details. Instead of changing the view-dependent appearance modeling approach, StopThePop [42] points out that 3DGS tends to cheat view-dependent effects by popping 3D Gaussians due to the per-ray depth sorting, which leads to less faithful results when the viewpoint is rotated. To mitigate the potential of popping 3D Gaussians, StopThePop [42] replaces per-ray depth sorting with tile-based sorting to ensure consistent sorting order in a local region. To better guide the growth of 3D Gaussian splatting, GaussianPro [20] introduces a progressive propagation strategy, updating Gaussians by considering normal consistency between neighboring views

and adding plane constraints. GeoGaussian [43] densifies Gaussians on their tangent planes and encourages smoothness of geometric properties between neighboring Gaussians. RadSplat [44] initializes 3DGS with a point cloud derived from a trained neural radiance field and prunes Gaussians with a multi-view importance score. To deal with more complex shading like specular and anisotropic components, Spec-Gaussian [45] utilizes anisotropic spherical Gaussians to approximate 3D scene appearance. TRIPS [46] attaches a neural feature to Gaussians and renders pyramid-like image feature planes according to the projected Gaussian's size, similar to ADOP [47], to resolve blurring issues in the original 3DGS. Handling the same issue, FreGS [48] applies frequency-domain regularization to the rendered 2D image to encourage recovery of high-frequency detail. GES [49] utilizes the generalized normal distribution (NFD) to sharpen scene edges. To resolve the problem that 3DGS is sensitive to initialization, RAIN-GS [50] initializes Gaussians sparsely with large variance from the SfM point cloud and progressively applies low-pass filtering to avoid 2D Gaussian projections smaller than a pixel. Pixel-GS [51] takes into account the number of pixels that a Gaussian covers from all input viewpoints, during the splitting process and scales the gradient according to the distance to the camera to suppress floaters. Bulò et al. [52] also use pixel-level error as a densification criterion and revise the opacity setting during the cloning process, leading to a more stable training process. Quantitative results for different reconstruction methods can be found in Table 1. 3DGS -based methods and NeRF-based methods are comparable but 3DGS-based methods render faster.

2.3 Compression and regularization

Although 3D Gaussian splatting achieves real-time rendering, there is still room to improve computational requirements and point distribution. Some methods focus on changing the original representation to reduce computational load.

Table 1 Quantitative comparison of novel view synthesis results on the MipNeRF 360 dataset [53] using PSNR, SSIM, and LPIPS metrics

Method	PSNR ↑	SSIM ↑	LPIPS ↓
MipNeRF [54]	24.04	0.616	0.441
MipNeRF 360 [53]	27.57	0.793	0.234
ZipNeRF [55]	28.54	0.828	0.189
3DGS [5]	27.21	0.815	0.214
Mip-Splatting [35]	27.79	0.827	0.203
Scaffold-GS [40]	28.84	0.848	0.220
VDGS [39]	27.66	0.809	0.223
GaussianPro [20]	27.92	0.825	0.208

Vector quantization, a traditional compression method in signal processing, which involves clustering multi-dimensional data into a finite set of representations, is mainly utilized in Gaussians [56–60]. C3DGS [56] adopts residual vector quantization (R-VQ) [61] to represent geometric attributes, including scaling and rotation. SASCGS [58] utilizes vector clustering to encode color and geometric attributes into two codebooks, with a sensitivity-aware k-means method. EAGLES [59] quantizes all attributes including color, position, opacity, rotation, and scaling; quantization of opacity leads to fewer floaters and visual artifacts in the novel view synthesis task. Compact3D [57] does not quantize opacity and position, because sharing them results in overlapping Gaussians. LightGaussian [60] prunes Gaussians with a smaller importance score and adopts octree-based lossless compression in G-PCC [62] for the position attribute due to the sensitivity of the subsequent rasterization accuracy to the position. Based on the same importance score calculation, Mini-Splatting [63] samples Gaussians instead of pruning points to avoid artifacts caused by pruning. SOGS [64] adopts a different method to vector quantization. They arrange Gaussian attributes into multiple 2D grids. These grids are sorted and a smoothness regularization is applied to penalize all pixels that have very different values compared to their local neighborhood on the 2D grid. HAC [65] adopts the idea of Scaffold-GS [40] to model the scene with a set of anchor points and learnable features on these anchor points. It further introduces an adaptive quantization module to compress the features of anchor points with a multi-resolution hash grid [2]. Jo et al. [66] propose to identify unnecessary Gaussians to compress 3DGS and accelerate the computation. Apart from 3D compression, 3D Gaussian splatting has also been applied to 2D image compression [67], where the 3D Gaussians degenerate to 2D Gaussians.

In terms of disk storage, SASCGS [58] utilizes the Deflate entropy encoding method, which combines the LZ77 algorithm and Huffman coding to compress the data. SOGS [64] compresses the RGB grid with

JPEG XL and stores all other attributes as 32-bit OpenEXR images with zip compression. Quantitative reconstruction results and sizes of 3D scenes after compression are shown in Table 2.

2.4 Dynamic 3D reconstruction

Like NeRF representation, 3DGS can also be extended to reconstruct dynamic scenes. The crux of dynamic 3DGS lies in how to model variations of Gaussian attribute values over time. The most straightforward way is to assign different attribute values to 3D Gaussians at different time steps. Luiten et al. [6] regard the center and orientation (expressed as a quaternion) of the 3D Gaussian as variables that change over time, while other attributes remain constant over all time steps, thus achieving 6-DOF tracking by reconstructing dynamic scenes. However, a frame-by-frame discrete definition lacks continuity, which can cause poor results in long-term tracking. Therefore, physically-based constraints are introduced, as three regularization losses: short-term local-rigidity and local-orientation similarity losses and a long-term local-isometry loss. However, this method still lacks inter-frame correlation and has a high storage overhead for long sequences. Therefore, decomposing spatial and temporal information and modeling them with a canonical space and a deformation field, respectively, has become another exploration direction. The canonical space is the static 3DGS, and then the problem becomes how to model the deformation field. One way is to use an MLP network to implicitly fit it, as is done for dynamic NeRF [68]. Yang et al. [7] follow this idea and input the positional-encoded Gaussian position and time step t to the MLP, which outputs offsets of position, orientation, and scaling of the 3D Gaussian. However, inaccurate poses may affect rendering quality. This is not significant in the continuous modeling of NeRFs, but discrete 3DGS can amplify this problem, especially in the time interpolation task. So, they add a linearly decaying Gaussian noise to the encoded time vector to improve temporal smoothing without additional computational overhead.

Table 2 Comparison of different compression methods on the MipNeRF360 [53] dataset

Method	SSIM↑	PSNR↑	LPIPS↓	Size (MB)↓
3DGS [5]	0.815	27.21	0.214	750
C3DGS [56]	0.798	27.08	0.247	48.8
Compact3D [57]	0.808	27.16	0.228	—
EAGLES [59]	0.810	27.15	0.240	68
SOGS [64]	0.763	25.83	0.273	18.2
SASCGS [58]	0.801	26.98	0.238	28.80
LightGaussian [60]	0.857	28.45	0.210	42.48

4D-GS [69] adopts multi-resolution hexplane voxels [70] to encode the temporal and spatial information of each 3D Gaussian rather than positional encoding and utilizes different compact MLPs for different attributes. For stable training, it first optimizes the static 3DGS and then optimizes the deformation field represented by an MLP. GauFRe [71] applies exponential and normalization operations to the scaling and rotation respectively after adding the delta values predicted by an MLP, ensuring convenient and reasonable optimization. As dynamic scenes contain large static parts, it randomly initializes the point cloud into dynamic point clouds and static point clouds, optimizes them accordingly, and renders them together to achieve decoupling of the dynamic part and the static part. 3DGStream [72] allows online training of 3DGS in dynamic scene reconstruction by modeling the transformation between frames as a neural transformation cache and adaptively adding 3D Gaussians to handle emerging objects. 4DGaussianSplatting [73] turns 3D Gaussians into 4D Gaussians and slices the 4D Gaussians into 3D Gaussians for each time step. The sliced 3D Gaussians are projected onto the image plane to reconstruct the corresponding frame. DG-Mesh [74] builds up a mesh for each frame with a differentiable Poisson solver from which locations of Gaussians can be refined. Guo et al. [75] and GaussianFlow [76] introduce 2D flow estimation results into the training of dynamic 3D Gaussians, which supports deformation modeling between neighboring frames and enables superior 4D reconstruction and 4D generation results. TOGS [77] constructs an opacity offset table to model changes in digital subtraction angiography. Zhang et al. [78] leverage the diffusion prior to enhance dynamic scene reconstruction and propose a neural bone transformation module for reconstructing animatable objects from monocular video.

Unlike NeRF, 3DGS is an explicit representation; implicit deformation modeling requires many parameters which may lead to overfitting, so some

explicit deformation modeling methods are also proposed, ensuring fast training. Katsumata et al. [79] use a Fourier series to fit changes in the Gaussian's position, inspired by the fact that the motion of human and articulated objects is sometimes periodic. The rotation is approximated by a linear function. Other attributes remain unchanged over time. Thus, dynamic optimization is used to optimize the parameters of the Fourier series and the linear function; the number of parameters is independent of time. These parametric functions are continuous functions of time, ensuring temporal continuity, thus ensuring the robustness of novel view synthesis. In addition to the image losses, a bidirectional optical flow loss is also introduced. Polynomial fitting and Fourier approximation have advantages in modeling smooth motion and violent motion, respectively. So Gaussian-Flow [80] combines these two methods in time and frequency domains to capture time-dependent residuals of the attribute, in the dual-domain deformation model (DDDM). The position, orientation, and color are considered to change over time. To prevent optimization problems caused by uniform time division, this work adopts adaptive time step scaling. Overall, the optimization iterates between static optimization and dynamic optimization, and introduces a temporal smoothness loss and a kNN rigid loss. Li et al. [81] introduce a temporal radial basis function to represent temporal opacity, which can effectively model any scene content that appears or vanishes. A polynomial function is exploited to model the motion and orientation of the 3D Gaussians. They also replace the spherical harmonics with features to represent view- and time-related color. These features consist of three parts: a base color, a view-related feature, and a time-related feature. The latter two are translated into a residual color through an MLP added to the base color to obtain the final color. During optimization, new 3D Gaussians will be sampled at the under-optimized positions based on training error and coarse depth. The explicit modeling methods used in the above methods are all based on commonly used functions.

DynMF [82] assumes that each dynamic scene is composed of a finite, fixed number of motion trajectories and argues that a learned basis of the trajectories will be smoother and more expressive. All motion trajectories in the scene can be linearly represented in this learned basis and a small temporal MLP is used to generate the basis. The position and orientation change over time and both share the motion coefficients with different motion bases. The regularization, sparsity, and local rigidity terms of the motion coefficients are introduced during optimization.

There are also some other possibilities to explore. 4DGS [87] regards the spacetime of the scene as an entirety and transforms the 3D Gaussians into 4D Gaussians, that is, transforming the attribute values defined on the Gaussian to 4D space. For example, the scaling matrix is diagonal, so adding a scaling factor for the time dimension on the diagonal forms the scaling matrix in 4D space. The 4D extension of the spherical harmonics (SH) can be expressed as a combination of SH with 1D-basis functions. SWAGS [88] divides the dynamic sequence into different windows based on the amount of motion and trains separate dynamic 3DGS models for different windows, with different canonical spaces and deformation fields. The deformation field uses a tunable MLP [89], which focuses more on modeling the dynamic part of the scene. Finally, fine-tuning ensures temporal consistency between windows using the overlapping frame to add constraints. The MLP is fixed and only the canonical representation is optimized during fine-tuning.

These dynamic modeling methods can be further applied in the medical field, such as for markerless motion reconstruction for motion analysis of infants and neonates [90], which introduces additional mask and depth supervision, and monocular endoscopic reconstruction [91–94]. Quantitative reconstruction results by representative NeRF-based and 3DGS-based methods are reported in Table 3. 3DGS-based methods have clear advantages compared to NeRF-based methods due to their explicit representation

Table 3 Quantitative comparison of novel view synthesis results on the D-NeRF [68] dataset using PSNR, SSIM, and LPIPS metrics

Method	PSNR ↑	SSIM ↑	LPIPS ↓
D-NeRF [68]	31.69	0.975	0.057
TiNeuVox [83]	33.76	0.984	0.044
Tensor4D [84]	27.72	0.945	0.051
K-Planes [85]	32.32	0.973	0.038
CoGS [86]	37.90	0.983	0.027
GauFRe [71]	34.80	0.985	0.028
4D-GS [69]	34.01	0.989	0.025
Yang et al. [7]	39.51	0.990	0.012
SC-GS [16]	43.30	0.997	0.0078

of geometry that can model dynamics more easily. The efficient rendering of 3DGS also avoids densely sampling and querying the neural fields of NeRF-based methods and makes downstream applications of dynamic reconstruction, like free-viewpoint video, more feasible.

2.5 3D reconstruction from challenging inputs

While most methods experiment on regular input data with dense viewpoints and relatively small scenes, other work targets reconstructing 3D scenes with challenging inputs like sparse-view input, data without camera parameters, and larger scenes like urban streets. FSGS [95] was the first to explore reconstructing 3D scenes from sparse view input. It initializes sparse Gaussians from structure-from-motion (SfM) methods and identifies them by unpooling existing Gaussians. To allow faithful geometry reconstruction, an extra pre-trained 2D depth estimation network helps to supervise the rendered depth images. SparseGS [96], CoherentGS [97], and DNGaussian [98] also target 3D reconstruction from sparse-view inputs by introducing depth inputs estimated by a pre-trained 2D network. It further removes Gaussians with incorrect depth values and utilizes the score distillation sampling (SDS) loss [99] to encourage results rendered from novel viewpoints to be more faithful. GaussianObject [100] instead initializes Gaussians with the visual hull and fine-tunes a pre-trained ControlNet [101] to repair degraded rendered images generated by adding noise to each Gaussians' attributes, outperforming previous NeRF-based sparse-view reconstruction methods. Moving a step forward, PixelSplat [102] reconstructs 3D scenes from single-view input without any data priors. It extracts pixel-aligned image features in a similar way to PixelNeRF [103] and predicts attributes for each Gaussian with neural networks. MVSplat [104] brings the cost volume representation into sparse view reconstruction, using it for input to the attribute prediction network for Gaussians. SplatterImage [105] also works on single-view data but instead utilizes a U-Net [106] network to translate the input image into attributes on Gaussians. It can be extended to multi-view inputs by aggregating predicted Gaussians from different viewpoints via warping operations.

For urban scene data, PVG [107] makes the Gaussians' mean and opacity values time-dependent functions centered at the corresponding Gaussian's life peak (maximum prominence over time). Driving-Gaussian [108] and HUGS [109] reconstruct dynamic driving data by first incrementally optimizing static 3D Gaussians and then composing them with dynamic objects' 3D Gaussians. This process is also assisted by the Segment Anything Model [110] and input LiDAR depth data. StreetGaussians [111] models the static background with a static 3DGS and dynamic objects by a dynamic 3DGS where Gaussians are transformed by tracked vehicle poses and their appearance is approximated using time-related spherical harmonic coefficients. SGD [112] incorporates diffusion priors with street scene reconstruction to improve the novel view synthesis results in a similar way to ReconFusion [113]. HGS-Mapping [114] separately models a textureless sky, ground plane, and other objects for more faithful reconstruction. VastGaussian [115] divides a large scene into multiple regions based on the camera distribution projected onto the ground and learns to reconstruct a scene by iteratively adding more viewpoints into training based on visibility criteria. In addition, it models appearance changes with an optimizable appearance embedding for each view. CityGaussian [116] also models large-scale scenes with a divide-and-conquer strategy and further introduces level-of-detail rendering based on the distance between the camera and each Gaussian. To facilitate comparisons on urban scenes for 3DGS methods, GauU-Scene [117] provides a large-scale dataset covering over 1.5 km^2.

Apart from the works mentioned above, other methods focus on special input data including images without camera [118–121], blurred inputs [122–125], unconstrained images [126, 127], mirror-like inputs [128, 129], CT scans [130, 131], panoramic images [132], and satellite images [133].

3 Gaussian splatting for 3D editing

3DGS allows for efficient training and high-quality real-time rendering using rasterization with point-based rendering techniques. Editing in 3DGS has been investigated in a number of fields. We have summarized the editing on 3DGS into three categories: geometry editing, appearance editing, and physical simulation.

3.1 Geometry editing

On the geometry side, GaussianEditor [134] controls the 3DGS using text prompts and semantic information from Gaussian semantic tracing, which enables 3D inpainting, object removal, and object composition. Gaussian Grouping [135] simultaneously rebuilds and segments open-world 3D objects under the supervision of 2D mask predictions from SAM and 3D spatial consistency constraints, which further enables diverse editing applications including 3D object removal, inpainting, and composition with high-quality visual effects and time efficiency. Furthermore, Point'n Move [136] combines interactive scene object manipulation with exposed region inpainting. Thanks to the explicit representation of 3DGS, the dual-stage self-prompting mask propagation process is able to transfer the given 2D prompt points to 3D mask segmentation, resulting in a user-friendly editing experience with high-quality effects. Feng et al. [137] propose a new Gaussian splitting algorithm to avoid inhomogeneous 3D Gaussian reconstruction and makes the boundary of 3D scenes after removal operation sharper. Although the above methods realize the editing on 3DGS, they are still limited to some simple editing operations (removal, rotation, and translation) for 3D objects. SuGaR [10] extracts explicit meshes from the 3DGS representation by regularizing Gaussians over surfaces. Further, it relies on manual adjustment of Gaussian parameters based on deformed meshes to realize desired geometry editing but struggles with large-scale deformation. SC-GS [16] learns a set of sparse control points for 3D scene dynamics but faces challenges with intense movements and detailed surface deformation. GaMeS [18] introduces a new GS-based model that combines a conventional mesh with plain GS. The explicit mesh is utilized as input and parameterizes Gaussian components using the vertices, which can modify Gaussians in real time by altering mesh components during inferencing. However, it cannot handle significant deformations or changes, especially deformation on large faces, since it cannot change the mesh topology during training. Although the above methods can provide some simple rigid transformations and non-rigid deformation, they still face challenges in their editing effectiveness and large-scale deformation. Gao et al. [19] also adapt mesh-based deformation to 3DGS by harnessing the priors of explicit representation (surface properties like normals of the mesh, and gradients generated by explicit deformation methods) and learning the face split to optimize the parameters and number of Gaussians, which provides adequate topological information to 3DGS and improves the quality of both reconstruction and geometry editing results. GaussianFrosting [138] shares a similar idea with Gao et al. [19] by constructing a base mesh but further develops a frosting layer to allow Gaussians to move in a small range near the mesh surface.

3.2 Appearance editing

On the appearance side, GaussianEditor [139] first modifies 2D images with language input using a diffusion model [140] in the masked region generated by a recent 2D segmentation model [110] and updates attributes of Gaussians again as in the previous NeRF editing work Instruct-NeRF2NeRF [141]. Other independent research, GaussianEditor [134], operates similarly but further introduces hierarchical Gaussian splatting (HGS) to allow 3D editing like object inpainting. GSEdit [142] takes a texture mesh or pre-trained 3DGS as input and utilizes Instruct-Pix2Pix [143] and SDS loss to update the input mesh or 3DGS. To alleviate inconsistency issues, GaussCtrl [144] introduces the depth map as conditional input to the ControlNet [101] to encourage geometric consistency. Wang et al. [145] also aim to solve this inconsistency issue by introducing multi-view cross-attention maps. Texture-GS [146] disentangles the geometry and appearance of 3DGS and learns a UV mapping network for points near the underlying surface, thus enabling manipulation such as texture painting and texture swapping. 3DGM [147] also represents a 3D scene with a proxy mesh with fixed UV mapping where Gaussians are stored on the texture map. This disentangled representation also allows animation and texture editing. Apart from local texture editing, other works [148–150] focus on stylizing 3DGS using a reference style image.

To allow more tractable control over texture and lighting, researchers have started to disentangle texture and lighting to enable independent editing. GS-IR [151] and RelightableGaussian [11] separately model texture and lighting. Additional material parameters are defined on each Gaussian to represent texture, and lighting is approximated by a learnable environment map. GIR [152] and GaussianShader [12]

share the same disentanglement paradigm by binding material parameters to 3D Gaussians, but to deal with more challenging reflective scenes, they add normal orientation constraints to Gaussians in a similar way to Ref-NeRF [153]. DeferredGS [154] observes that geometry attributes of 3D Gaussians like opacity overfit to the input lighting condition and exhibit blending artifacts when they are relit. To resolve this issue, DeferredGS distills geometry attributes of a signed distance function to 3D Gaussians and introduces the deferred shading technique into the rendering of 3D Gaussians to avoid blending artifacts caused by multiple shading calculations.

3.3 Physical simulation

Considering physically-based 3DGS editing, Phys-Gaussian [9] employs discrete particle clouds from 3DGS for physically-based dynamics and photo-realistic rendering through continuum deformation [155] of Gaussian kernels. Gaussian Splashing [156] combines 3DGS and position-based dynamics (PBD) [157] to manage rendering, view synthesis, and solid/fluid dynamics cohesively. In a similar way to Gaussian shaders [12], the normal is applied to each Gaussian kernel to align it with the surface normal and improve PBD simulation, also allowing the physically-based rendering to enhance dynamic surface reflections on fluids. VR-GS [17] is a physical dynamics-aware interactive Gaussian Splatting system for VR, tackling the difficulty of editing high-fidelity virtual content in real time. VR-GS utilizes 3DGS to close the quality gap between generated and manually crafted 3D content. By utilizing physically-based dynamics, which enhance immersion and offer precise interaction and manipulation controllability, Spring-Gaus [158] applies the spring–mass model to the modeling of dynamic 3DGS and learns physical properties like mass and velocity from input video, allowing them to be edited for real-world simulation. Feature Splatting [159] further incorporates semantic priors from pre-trained networks and makes object-level simulation possible.

4 Applications of Gaussian splatting

4.1 Segmentation and understanding

Open-world 3D scene understanding is an essential challenge in robotics, autonomous driving, and VR/AR environments. With the remarkable progress in 2D scene understanding brought by SAM [110] and its variants, existing methods have tried to integrate semantic features, such as CLIP [160]/DINO [161] into NeRF, to deal with 3D segmentation, understanding, and editing.

NeRF-based methods are computationally intensive because of the implicit and continuous representation. Recent methods try to integrate 2D scene understanding methods with 3D Gaussians to produce a real-time and easy-to-edit 3D scene representation. Most methods utilize pre-trained 2D segmentation methods like SAM [110] to produce semantic masks of input multi-view images [135, 136, 162–167], or extract dense language features, CLIP [160]/DINO [161], for each pixel [168–170].

LEGaussians [168] adds an uncertainty value attribute and semantic feature vector attribute for each Gaussian. It then renders a semantic map with uncertainties from a given viewpoint, to compare with the quantized CLIP and DINO dense features of the ground truth image. To achieve 2D mask consistency across views, Gaussian Grouping [135] employs DEVA to propagate and associate masks from different viewpoints. It adds identity encoding attributes to 3D Gaussians and renders the identity feature map for comparison to the extracted 2D masks.

4.2 Geometry reconstruction and SLAM

Geometry reconstruction and SLAM are important subtasks in 3D reconstruction.

4.2.1 Geometry reconstruction

In the context of NeRF, a series of works [171–174] has successfully reconstructed high-quality geometry from multi-view images. However, due to the discrete nature of 3DGS, only a few works have stepped into this field. SuGaR [10] pioneered building up 3D surfaces from multi-view images using the 3DGS representation. It introduced a simple but effective self-regularization loss to constrain that the distance between the camera and the closest Gaussian should be as similar as possible to the corresponding pixel's depth value in the rendered depth map, which encourages alignment between the 3DGS and the authentic 3D surface. Instead, NeuSG [175] chooses to incorporate the previous NeRF-based surface reconstruction method NeuS [171] in the 3DGS representation to transfer

the surface property to 3DGS. More specifically, it encourages Gaussians' signed distances to be zero and the normal directions of 3DGS and the NeuS method to be as consistent as possible. 3DGSR [176] and GSDF [177] also encourage consistency of SDF and 3DGS to enhance the quality of reconstructed geometry. DN-Splatter [178] utilizes depth and normal priors captured from common devices or predicted from general-purpose networks to enhance the reconstruction quality of 3DGS. Wolf et al. [179] first train a 3DGS to render stereo-calibrated novel views and apply stereo depth estimation to the rendered views. The estimated dense depth maps are fused by the truncated signed distance function (TSDF) to form a triangular mesh. 2D-GS [21] replaces 3D Gaussians with 2D Gaussians for a more accurate ray-splat intersection and employs a low-pass filter to avoid degenerate line projection. The Gaussian Opacity Fields method [22] calculates a randomly sampled point's opacity value from the opacity of 3D Gaussians and converts the discrete 3D Gaussians into a continuous opacity field, which can be transformed into an explicit surface. While attempts have been made in the field of geometry reconstruction, due to the discrete nature of 3DGS, current methods achieve results no better or worse than implicit representation-based methods with the continuous field assumption where the surface can be easily determined.

4.2.2 SLAM

Further 3DGS methods target simultaneously localizing the camera and reconstructing the 3D scene. GS-SLAM [180] proposes an adaptive 3D Gaussian expanding strategy to add new 3D Gaussians into the training stage and delete unreliable ones with captured depths and rendered opacity values. To avoid duplicate densification, SplaTAM [181] uses view-independent colors for Gaussians and creates a densification mask to determine whether a pixel in a new frame needs densification by considering current Gaussians and the captured depth of the new frame. To stabilize localization and mapping, GaussianSplattingSLAM [182] and Gaussian-SLAM [183] use an extra Gaussian scale regularization loss to encourage isotropic Gaussians. For easier initialization, LIV-GaussMap [184] initializes Gaussians with a LiDAR point cloud and builds up an optimizable size-adaptive voxel grid for the global map. SGS-SLAM [185], NEDS-SLAM [186], and SemGauss-SLAM [187] further consider Gaussians' semantic information in the simultaneous localization and mapping process by distilling 2D semantic information which can be obtained using 2D segmentation methods or provided by the dataset. Deng et al. [188] avoid redundant Gaussian splitting based on a sliding window mask and use vector quantization to further encourage compact 3DGS. CG-SLAM [189] introduces an uncertainty map into the training process based on the rendered depth; this greatly improves the reconstruction quality. Based on the map reconstructed by SLAM-based methods, tasks in robotics like relocalization [190], navigation [191–193], 6D pose estimation [194], multi-sensor calibration [195, 196], and manipulation [197, 198] can be performed efficiently. We report quantitative results of different SLAM methods on the reconstruction task in Table 4. The explicit geometry representation provided by 3DGS enables flexible reprojection to alleviate misalignment of viewpoints, thus leading to better reconstruction than NeRF-based methods. The real-time rendering feature of 3DGS also makes neural-based SLAM methods more applicable: training previous NeRF-based methods requires more hardware and time.

4.3 Digital humans

Learning virtual humans via implicit representation has been explored in various ways, especially using NeRF and SDF representations, which exhibit high-quality results from multi-view images but suffer from heavy computational costs. Thanks to the high efficiency of 3DGS, research has flourished and pushed 3DGS into digital human creation.

Table 4 Quantitative comparison of novel view synthesis results by different SLAM methods on the Replica [199] dataset using PSNR, SSIM, and LPIPS metrics

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NICE-SLAM [200]	24.42	0.81	0.23
Vox-Fusion [201]	24.41	0.80	0.24
Co-SLAM [202]	30.24	0.94	0.25
GS-SLAM [180]	31.56	0.97	0.094
SplaTAM [181]	34.11	0.97	0.10
GaussianSplattingSLAM [182]	37.50	0.96	0.07
Gaussian-SLAM [183]	38.90	0.99	0.07
SGS-SLAM [185]	34.15	0.97	0.096

4.3.1 Body

Full-body modeling aims to reconstruct dynamic humans from multi-view videos. D3GA [203] first creates animatable human avatars using drivable 3D Gaussians and tetrahedral cages to provide promising geometry and appearance modeling. To capture more dynamic details, SplatArmor [204] leverages two different MLPs to predict large motions built upon the SMPL and canonical space and allows pose-dependent effects through use of SE(3) fields, enabling more detailed results. HuGS [205] creates a coarse-to-fine deformation module using linear blending skinning and local learning-based refinement for constructing and animating virtual human avatars based on 3DGS. It achieves state-of-the-art human neural rendering performance of 20 FPS. Similarly, HUGS [206] utilizes the tri-plane representation [207] to factorize the canonical space, which can reconstruct a person and scene from monocular video (of 50–100 frames) within 30 min. Since 3DGS learns a huge number of Gaussians ellipsoids, HiFi4G [208] combines 3DGS with the non-rigid tracking offered by its dual-graph mechanism for high-fidelity rendering, successfully preserving spatial-temporal consistency in a more compact manner. To achieve higher rendering speeds with high resolution on consumer-level devices, GPS-Gaussian [13] introduces Gaussian parameter maps on the sparse source view to regress the Gaussian parameters jointly with a depth estimation module without any fine-tuning or optimization. GART [209] extends human models to other articulated models (e.g., animals) based on the 3DGS representation.

To make full use of the information from multi-view images, Animatable Gaussians [210] incorporates 3DGS and 2D CNNs for more accurate human appearance and realistic garment dynamics using a template-guided parameterization and pose projection mechanism. Gaussian Shell Maps [211] (GSMs) combine CNN-based generators with 3DGS to recreate virtual humans with sophisticated details such as clothing and accessories. ASH [212] projects 3D Gaussian learning into a 2D texture space using mesh *UV* parameterization to capture appearance, enabling real-time, high-quality animated humans. Furthermore, for reconstructing rich details on humans, such as clothing, 3DGS-Avatar [213] introduces a shallow MLP instead of SH to model the color of 3D Gaussians and regularizes deformation with geometry priors, providing the photorealistic rendering with pose-dependent cloth deformation which generalizes to novel poses effectively.

For dynamic digital human modeling based on monocular video, GaussianBody [214] further leverages physically-based priors to regularize the Gaussians in the canonical space to avoid artifacts in the dynamic cloth from monocular video. GauHuman [215] re-designs the prune/split/clone of the original 3DGS to achieve efficient optimization and incorporates pose refinement and weight field modules for fine detail learning. It can be trained in minutes, and allows real-time rendering (166 FPS). GaussianAvatar [216] incorporates an optimizable tensor with a dynamic appearance network to better capture the dynamics, allowing dynamic avatar reconstruction and realistic novel animation in real time. Human101 [217] further pushes the speed of high-fidelity dynamic human creation, taking 100 s, using a fixed-perspective camera. Similar to Ref. [19], SplattingAvatar [218] and GoMAvatar [219] embed Gaussians onto a canonical human body mesh. The position of a Gaussian is determined by the barycenter and displacement along the normal direction. To resolve the unbalanced aggregation of Gaussians caused by densification and splitting operations, GVA [220] uses a surface-guided Gaussian re-initialization strategy to make the trained Gaussians better fit the input monocular video. HAHA [221] also attaches Gaussians to the surface of a mesh but combines rendered results from a textured human body mesh and Gaussians, to reduce the number of Gaussians.

4.3.2 Head

For human head modeling with 3DGS, Mono-GaussianAvatar [222] first applies 3DGS to dynamic head reconstruction using canonical space modeling and deformation prediction. Further, PSAvatar [223] introduces the explicit Flame face model [224] to initialize the Gaussians, which can capture high-fidelity facial geometry and even complicated volumetric objects (e.g., glasses). Tri-plane representation and motion fields are used in GaussianHead [225] to simulate geometrically changing heads in continuous movements and to render rich textures, including skin and hair. For easier head expression controllability, GaussianAvatars [226] introduces geometric priors

(the Flame parametric face model [224]) into 3DGS, which binds the Gaussians onto the explicit mesh, and optimizes the parameters of the Gaussian ellipsoids. Rig3DGS [227] employs a learnable deformation to provide stability and generalization to novel expressions, head poses, and viewing directions to achieve controllable portraits on portable devices. In another approach, HeadGas [228] attributes the 3DGS with a base of latent features that are weighted by the expression vector from 3DMMs [229], which allows reconstruction in real-time of animatable heads. FlashAvatar [230] further embeds a uniform 3D Gaussian field in a parametric face model and learns additional spatial offsets to capture facial details, successfully pushing the rendering speed to 300 FPS. To synthesize high-resolution results, Gaussian Head Avatar [231] adopts a super-resolution network to achieve high-fidelity head avatar learning. To synthesize high-quality avatars from few input views, SplatFace [232] first initializes Gaussians on a template mesh and jointly optimizes Gaussians and the mesh with a splat-to-mesh distance loss. GauMesh [233] uses a hybrid representation containing both tracked textured meshes and canonical 3D Gaussians together with a learnable deformation field to represent dynamic human heads. Other works use 3DGS for text-based head generation [234], deep fakes [235], and relighting [236].

Hair and hands. Other specific human parts have also been explored, such as hair and hands. 3D-PSHR [237] combines hand geometry priors (MANO) with 3DGS, to provide the first real-time hand reconstruction. MANUS [238] further explores the interaction between hands and objects using 3DGS. In addition, GaussianHair [239] combines the Marschner Hair Model [240] with UE4's real-time hair rendering to create the Gaussian Hair Scattering Model. It captures complex hair geometry and appearance for fast rasterization and volumetric rendering, enabling applications including editing and relighting.

4.4 3D/4D generation

4.4.1 Need

Cross-modal image generation has achieved stunning results using the diffusion model [140]. However, due to the lack of 3D data, it is difficult to directly train a large-scale 3D generation model. The pioneering work DreamFusion [99] exploits a pre-trained 2D diffusion model and proposes the score distillation sampling (SDS) loss, which distills the 2D generative priors into 3D without requiring 3D data for training, achieving text-to-3D generation. However, the NeRF representation brings a heavy rendering overhead. The optimization time for each case takes several hours and the rendering resolution is low, leading to poor-quality results. Although some improved methods extract a mesh representation from trained NeRFs for fine-tuning to improve the quality [241], this further increases optimization time. 3DGS representation can render high-resolution images at high FPS using little memory, so it replaces NeRFs as the 3D representation in some recent 3D/4D generation methods.

4.4.2 3D generation

DreamGaussian [8] replaces the MipNeRF [54] representation in the DreamFusion [99] framework with 3DGS, using SDS loss to optimize 3D Gaussians. The splitting process of 3DGS is very suitable for optimization in a generative setting, allowing the efficiency of 3DGS to be brought to text-to-3D generation based on the SDS loss. To improve the final quality, this work follows the idea of Magic3D [241] which extracts a mesh from the generated 3DGS and refines the texture details by optimizing UV textures through a pixel-wise mean squared error (MSE) loss. In addition to 2D SDS, GSGEN [242] introduces a 3D SDS loss based on Point-E [243], a text-to-point-cloud diffusion model, to mitigate the multi-face (Janus) problem. It adopts Point-E to initialize the point cloud as the initial geometry for optimization and also refines the appearance with only the 2D image prior. GaussianDreamer [244] also combines the priors of 2D and 3D diffusion models. It utilizes Shap-E [245] to generate the initial point cloud and optimizes 3DGS using 2D SDS. However, the generated initial point cloud is relatively sparse, so noisy point growth and color perturbation are further proposed to densify it. However, even if the 3D SDS loss is introduced, the Janus problem may still exist during optimization as the view is sampled one by one. Some methods [246, 247] fine-tune the 2D diffusion model [140] to generate multi-view images at once, thereby achieving multi-view supervision during SDS optimization. Alternatively, the multi-view SDS proposed by BoostDream [248] directly creates large 2×2 images by stitching rendered images from 4 sampled views and calculates the gradients under

the condition of the multi-view normal map. This is a plug-and-play method that can first convert a 3D asset into differentiable representations including NeRF, 3DGS, and DMTet [249] through rendering supervision, and then optimize them to improve the quality of the 3D asset.

Some methods have made improvements to SDS loss. LucidDreamer [250] uses interval score matching (ISM), to replace DDPM in SDS with DDIM inversion and introduces supervision from interval steps of the diffusion process to avoid large error in one-step reconstruction. GaussianDiffusion [251] incorporates structured noise from multiple viewpoints to alleviate the Janus problem and variational 3DGS for better generation results by mitigating floaters. Yang et al. [252] point out that the differences between the diffusion prior and the training process of the diffusion model impair the quality of 3D generation, so they propose iterative optimization of the 3D model and the diffusion prior. Specifically, two additional learnable kinds of parameters are introduced in the classifier-free guidance formula, a learnable unconditional embedding, and additional parameters added to the network, such as LoRA [253] parameters. These methods are not limited to 3DGS, and other originally NeRF-based methods including VSD [254] and CSD [255] which aim to improve the SDS loss can be used with 3DGS generation. GaussianCube [256] instead trains a 3D diffusion model based on a GaussianCube representation that is converted from a constant number of Gaussians with voxelization via optimal transport. GVGEN [257] also works in 3D space but is based on a 3D Gaussian volume representation.

As a special category, human body modeling can introduce a model prior, such as SMPL [258], to assist in generation. GSMs [211] build multi-layer shells from the SMPL template and bind 3D Gaussians to the shells. By utilizing the differentiable rendering of 3DGS and the generative adversarial network of StyleGAN2 [259], animatable 3D humans can be efficiently generated. GAvatar [260] adopts a primitive-based representation [261] attached to SMPL-X [262] and attaches 3D Gaussians to the local coordinate system of each primitive. The attribute values of 3D Gaussians are predicted by an implicit network and the opacity is converted to the signed distance field through a NeuS-like method [171], providing geometry constraints and extracting detailed textured meshes. The generation is text-based and mainly guided by the SDS loss. HumanGaussian [263] initializes 3D Gaussians by randomly sampling points on the surface of the SMPL-X [262] template. It extends Stable Diffusion [140] to generate RGB and depth simultaneously and constructs a dual-branch SDS for optimization guidance. It also combines the classifier score provided by the null text prompt and the negative score provided by the negative prompt to construct negative prompt guidance to address the over-saturation issue.

The above methods focus on the generation of an individual object, while scene generation requires consideration of interactions and relationships between different objects. CG3D [264] inputs a text prompt manually deconstructed by the user into a scene graph, and the textual scene graph is interpreted as a probabilistic graphical model in which the tail of a directed edge is an object node and the head is an interaction node. Scene generation becomes ancestor sampling by first generating the objects and then their interactions. Optimization is divided into two stages; gravity and normal contact forces are introduced in the second stage. LucidDreamer [265] and Text2Immersion [266] are both based on a reference image (user-specified or text-generated) and extend outward to achieve 3D scene generation. The former utilizes stable diffusion (SD) [140] for image inpainting to generate unseen regions in the sampled views and incorporates monocular depth estimation and alignment to establish a 3D point cloud from these views. Finally, the point cloud is used as the initial value, and a 3DGS is trained using the projected images as ground truth to generate a 3D scene. The latter method has a similar idea, while having a process to remove outliers in the point cloud and the 3DGS optimization has two stages: coarse 3DGS training, and refinement. For 3D scene generation, GALA3D [267] leverages both the object-level text-to-3D model MVDream [246] to generate realistic objects and a scene-level diffusion model to compose them. DreamScene [268] uses multi-timestep sampling for multi-stage scene-level generation to synthesize a surrounding environment, ground, and objects to avoid complicated object composition. Instead of separately modeling each

object, RealmDreamer [269] utilizes diffusion priors from inpainting and depth estimation models to generate different viewpoints of a scene iteratively. DreamScene360 [270] instead generates a 360-degree panoramic image and converts it into a 3D scene with depth estimation.

Text-to-3D generation methods can be applied to image-to-3D generation, or monocular 3D reconstruction, with some simple modifications. For example, the pre-trained diffusion model used in SDS loss can be replaced by Zero-1-to-3 XL [271] for image condition [8]. We can also add losses between the input image and the corresponding rendered image in the input view to make generation more consistent with the input image. Based on the image-to-3D generation of DreamGaussian [8], Repaint123 [272] has a progressive controllable repainting mechanism to refine the generated mesh texture. During the process of repainting the occlusions, it incorporates textural information from the reference image through attention feature injection [273] and uses a visibility-aware repainting process to refine overlap regions with different strengths. Finally, the refined images are used as ground truths to directly optimize the texture through MSE loss, achieving fast optimization. Other methods explore utilizing existing 3D datasets [274, 275] and constructing large models to directly generate a 3DGS representation from a single image. TriplaneGaussian [14] proposes a hybrid representation combining tri-planes and 3DGS. It generates a point cloud and a tri-plane encoding 3DGS's attribute information from the input image features through a transformer-based point cloud decoder and a tri-plane decoder, respectively. The generated point cloud is densified through an upsampling method and then projected onto the tri-plane to query features. The queried features are augmented by the projected image features and are translated into 3D Gaussian attributes using an MLP, thereby achieving generation of 3DGS from a single image. LGM [276] first exploits off-the-shelf models to generate multi-view images from text [246] or a single image [247]. Then it trains a U-Net-based network to generate 3DGS from multi-view images. The U-Net is asymmetric, which allows for the input of high-resolution images while limiting the number of output Gaussians. AGG [277] also introduces a hybrid generator to obtain the point cloud and tri-plane features. However, it first generates a coarse 3DGS and then upsamples it through a U-Net-based super-resolution module to improve the fidelity of the generated results. Instead of predicting a point cloud from an image, BrightDreamer [278] predicts deviations of a set of fixed anchor points to determine the centers of Gaussians. GRM [279] utilizes existing multi-view generation models to train a pixel-aligned Gaussian representation for faithful 3D generation using a single feed-forward pass. IM-3D [280] fine tunes an image-to-video model based on Emu [281] to generate a turn-table like video rotating around an object, which is taken as the input for 3DGS reconstruction. Gamba [282] predicts Gaussian attributes using the recent Mamba [283] network to better capture the relationship between Gaussians. MVControl [284] extends ControlNet [101] to the 3D generation task and allows extra conditioning inputs like edge, depth, normal, and scribbles to be fed into existing multi-view generation models. Hyper-3DG [285] has a geometry and texture refinement module to improve generation quality based on hypergraph learning where each node is a patch image of a coarse 3DGS. For the same purpose, DreamPolisher [286] utilizes a ControlNet-based network for texture refinement and ensures consistency between different viewpoints with view-consistent geometric guidance. FDGaussian [287] injects tri-plane features into the diffusion model for better geometry generation.

4.4.3 4D generation

Based on the current progress in 3D generation, preliminary exploration has also been performed on 4D generation with 3DGS representation. AYG [288] endows 3DGS with dynamics with a deformation network for text-to-4D generation. It is divided into two stages, static 3DGS generation with SDS losses based on Stable Diffusion [140] and MVDream [246], and then dynamic generation with a video SDS loss based on a text-to-video diffusion model [289]. In the dynamic generation stage, only the deformation field network is optimized, and some frames are randomly selected to add image-based SDS to ensure generation quality. DreamGaussian4D [15] achieves 4D generation given a reference image. A static 3DGS is first generated using an improved version of DreamGaussian [8]. Off-the-shelf Stable Diffusion Video is utilized to generate a video from the given image. Then dynamic generation is also realized by

optimization of a deformation network added to the static 3DGS, and the generated video is used as supervision, along with a 3D SDS loss based on Zero-1-to-3 XL [271] from sampled views. Finally, this method also extracts a mesh sequence and optimizes the texture with an image-to-video diffusion model. Last, for video-to-4D generation, 4DGen [290] and Efficient4D [291] both utilize SyncDreamer [292] to generate multi-view images from the input frames as pseudo ground truth to train a dynamic 3DGS. The former introduces HexPlane [70] as the dynamic representation and constructs point clouds using generated multi-view images as 3D deformation pseudo ground truth. The latter directly converts 3D Gaussian into 4D Gaussian and enhances the temporal continuity of SyncDreamer [292] by fusing spatial volumes of adjacent timestamps, achieving synchronization to generate better cross-time multi-view images for supervision. SC4D [293] migrates the idea of sparse control points in SC-GS [16] to model deformation and appearance more efficiently. STAG4D [294] proposes a temporally consistent multi-view diffusion model to generate multi-view videos for 4D reconstruction from monocular video or 4D generation. To overcome previous 4D generation models' object-centric limitation, Comp4D [295] first generates individual 4D objects and later composes them subject to trajectory constraints. To enable more realistic 3D/4D generation, most methods utilize priors from diffusion models, which operate on 2D images and require a rendered viewpoint of the generated object. The fast rasterization-based rendering in 3DGS allows the priors to be more efficiently applied than when using NeRF-based methods with slow rendering speed.

5 Discussion

5.1 Summary

This survey has presented an overview of the recent 3D Gaussian splatting (3DGS) technique, not only illustrating how it originated from traditional point-based rendering methods but also how its fast rendering and explicit geometry facilitate a series of works targeting different tasks like 3D reconstruction and 3D editing, with representative works shown in the diagrams in Fig. 3.

Although 3DGS has greatly improved efficiency and result quality on a few tasks, it is not a perfect 3D representation that can satisfy the needs of all tasks. Here we first discuss the advantages and disadvantages of commonly used 3D representations including meshes, SDFs, NeRFs, and 3DGS. We then summarize the challenges that remain for 3D Gaussian splatting and how these challenges might be resolved in the future.

5.2 Representations

5.2.1 Meshes

A mesh is made of a set of vertices, edges, and faces, which can express detailed geometry at a relatively low storage cost. As the most widely used 3D representation in industry, it can create high-quality visual effects with the help of physically-based materials in real time. However, most meshes are made by artists or creators, which is time consuming.

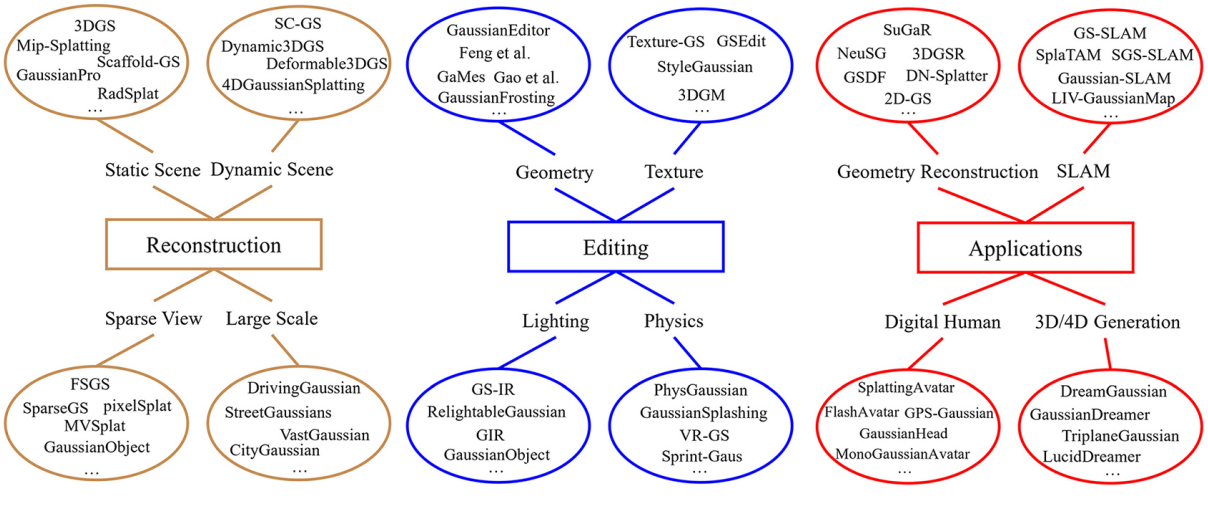


Fig. 3 Representative works based on the 3DGS representation, for different tasks.

Even if some works [296–299] attempt to generate meshes automatically with neural networks, their generation capability is still constrained by the scale and scope of existing datasets.

5.2.2 SDFs and NeRFs

Both SDFs and NeRFs are based on implicit neural fields, which can be learned automatically from a set of multi-view images. Explicit geometry or a mesh can be extracted from SDF or NeRF representations with the marching cube algorithm. Thus, SDFs and NeRFs have an advantage in tasks like inverse rendering that rely on a good surface representation. However, due to the dense sampling in 3D space, rendering is inefficient, limiting their applicability on consumer-level devices. Moreover, SDFs and NeRFs are less successful for dynamic scene reconstruction due to their implicit representation.

5.2.3 3DGS

3DGS also has explicit geometry but unlike meshes they have no edges or faces, to connect different Gaussians. To make up for the missing connection information, each Gaussian has anisotropic scales to avoid a gap between neighboring Gaussians and synthesize realistic novel views. With a rasterization-based renderer, 3DGS allows real-time visualization of 3D scenes on consumer-level devices, making applications like large-scene reconstruction, SLAM, and generation that have efficiency requirements more possible. Apart from efficient rendering, the explicit geometry representation of 3DGS enables flexible point reprojection from one viewpoint to another viewpoint, making simultaneous geometry reconstruction and camera pose optimization and dynamic reconstruction easier, and tasks like SLAM and large scene reconstruction more efficient. However, due to the discrete geometry representation, the geometry reconstruction quality of current 3DGS-based methods is only comparable to previous SDF-based methods like NeuS [171]. It would be promising to combine other 3D representations with 3DGS similar to Refs. [19, 175] to build up a high-quality geometry or surface, facilitating downstream applications like automated vehicles and animation.

5.3 Challenges

5.3.1 Robust and generalizable novel view synthesis

Although 3D Gaussian splatting can achieve realistic novel view synthesis results, its reconstruction quality degrades as indicated by Ref. [42] when dealing with challenging inputs like sparse-view inputs, complex shading effects, and large-scale scenes. While attempts [12, 96, 108] have been made to improve results, there remains room for further improvement. Making reconstruction more robust across different inputs is an important problem. In addition, developing a generalizable reconstruction pipeline with or without data priors, like Refs. [102, 105, 277], would significantly reduce training costs.

5.3.2 Geometry reconstruction

Despite the efforts on rendering quality, few methods [10, 175] have tackled geometry and surface reconstruction with the 3DGS representation. Compared to the continuous implicit representations like NeRFs and SDFs, 3DGS's geometry quality still suffers from its discrete geometry representation.

5.3.3 Independent and efficient 3D editing

A few methods have dived into the field of editing 3D Gaussian splatting's geometry [10, 16, 18, 19], texture [134, 139], or lighting [11, 12, 151, 152]. However, they cannot decompose geometry, texture, and lighting accurately or need re-optimization of Gaussians' attributes. As a result, these methods still lack independent editing capabilities or lack efficiency in the editing process. A promising challenge is to extract geometry, texture, and lighting with more advanced rendering techniques to facilitate independent editing and to build the connection between 3DGS and mesh-based representation to enable efficient editing.

5.3.4 Realistic 4D generation

With the help of SDS loss based on SD [140], generative models [8, 14, 276] using the 3DGS representation have produced faithful results. However, 4D generation results of current methods [15, 289, 290] still lack realistic geometry, appearance, and physics-aware motion. Integrating data priors like results produced by video generative models and physical laws might boost the quality of generated 4D content.

5.3.5 Platforms

Most implementations of methods and frameworks like GauStudio [300] for the 3D Gaussian splatting representation are written in Python with the CUDA-supported PyTorch [301] framework, which may limit its future applicability to a wider range of platforms. Reproducing the results with deep learning

frameworks like Tensorflow [302] and Jittor [303] could facilitate their usage on other platforms.

Author contributions

Tong Wu conducted an extensive literature review and drafted the manuscript. Yu-Jie Yuan, Ling-Xiao Zhang, and Jie Yang provided critical insights, analysis of the existing research, and part of the manuscript writing. Yan-Pei Cao, Ling-Qi Yan, and Lin Gao conceived the idea and scope of the survey and improved the writing. All authors read and approved the final manuscript.

Availability of data and materials

As the paper does not involve the generation or analysis of specific datasets, there is no associated data or materials.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62322210), Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (JQ21013), Beijing Municipal Science and Technology Commission (Z231100005923031), and 2023 Tencent AI Lab Rhino-Bird Focused Research Program. We would like to thank Jia-Mu Sun and Shu-Yu Chen for their suggestions concerning the timeline figure.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Ling-Qi Yan is the Associate Editor of this journal.

References

[1] Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J-M. Eds. Springer Cham, 405–421, 2020.

[2] Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics* Vol. 41, No. 4, Article No. 102, 2022.

[3] Chen, Z.; Funkhouser, T.; Hedman, P.; Tagliasacchi, A. MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16569–16578, 2023.

[4] Yariv, L.; Hedman, P.; Reiser, C.; Verbin, D.; Srinivasan, P. P.; Szeliski, R.; Barron, J. T.; Mildenhall, B. BakedSDF: Meshing neural SDFs for real-time view synthesis. In: Proceedings of the SIGGRAPH Conference, Article No. 46, 2023.

[5] Kerbl, B.; Kopanas, G.; Leimkuehler, T.; Drettakis, G. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* Vol. 42, No. 4. Article No. 139, 2023.

[6] Luiten, J.; Kopanas, G.; Leibe, B.; Ramanan, D. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint* arXiv:2308.09713, 2023.

[7] Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; Jin, X. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint* arXiv:2309.13101, 2023.

[8] 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.

[9] Xie, T.; Zong, Z.; Qiu, Y.; Li, X.; Feng, Y.; Yang, Y.; Jiang, C. PhysGaussian: Physics-integrated 3D Gaussians for generative dynamics. *arXiv preprint* arXiv:2311.12198, 2023.

[10] Guédon, A.; Lepetit, V. SuGaR: Surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. *arXiv preprint* arXiv:2311.12775, 2023.

[11] Gao, J.; Gu, C.; Lin, Y.; Zhu, H.; Cao, X.; Zhang, L.; Yao, Y. Relightable 3D Gaussian: Real-time point cloud relighting with BRDF decomposition and ray tracing. *arXiv preprint* arXiv:2311.16043, 2023.

[12] Jiang, Y.; Tu, J.; Liu, Y.; Gao, X.; Long, X.; Wang, W.; Ma, Y. GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces. *arXiv preprint* arXiv:2311.17977, 2023.

[13] Zheng, S.; Zhou, B.; Shao, R.; Liu, B.; Zhang, S.; Nie, L.; Liu, Y. GPS-Gaussian: Generalizable pixel-wise 3D Gaussian splatting for real-time human novel view synthesis. *arXiv preprint* arXiv:2312.02155, 2023.

[14] Zou, Z. X.; Yu, Z.; Guo, Y. C.; Li, Y.; Liang, D.; Cao, Y. P.; Zhang, S. H. Triplane meets Gaussian splatting: Fast and generalizable single-view 3D reconstruction with transformers. *arXiv preprint* arXiv:2312.09147, 2023.

[15] Ren, J.; Pan, L.; Tang, J.; Zhang, C.; Cao, A.;

Zeng, G.; Liu, Z. DreamGaussian4D: Generative 4D Gaussian splatting. *arXiv preprint* arXiv:2312.17142, 2023.

[16] Huang, Y. H.; Sun, Y. T.; Yang, Z.; Lyu, X.; Cao, Y. P.; Qi, X. SC-GS: Sparse-controlled Gaussian splatting for editable dynamic scenes. *arXiv preprint* arXiv:2312.14937, 2023.

[17] Jiang, Y.; Yu, C.; Xie, T.; Li, X.; Feng, Y.; Wang, H.; Li, M.; Lau, H.; Gao, F.; Yang, Y.; et al. VR-GS: A physical dynamics-aware interactive Gaussian splatting system in virtual reality. *arXiv preprint* arXiv:2401.16663, 2024.

[18] Waczyńska, J.; Borycki, P.; Tadeja, S.; Tabor, J.; Spurek, P. GaMeS: Mesh-based adapting and modification of Gaussian splatting. *arXiv preprint* arXiv:2402.01459, 2024.

[19] Gao, L.; Yang, J.; Zhang, B. T.; Sun, J. M.; Yuan, Y. J.; Fu, H.; Lai, Y. K. Mesh-based Gaussian splatting for real-time large-scale deformation. *arXiv preprint* arXiv:2402.04796, 2024.

[20] Cheng, K.; Long, X.; Yang, K.; Yao, Y.; Yin, W.; Ma, Y.; Wang, W.; Chen, X. GaussianPro: 3D Gaussian splatting with progressive propagation. *arXiv preprint* arXiv:2402.14650, 2024.

[21] Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; Gao, S. 2D Gaussian splatting for geometrically accurate radiance fields. *arXiv preprint* arXiv:2403.17888, 2024.

[22] Yu, Z.; Sattler, T.; Geiger, A. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint* arXiv:2404.10772, 2024.

[23] Chen, G.; Wang, W. A survey on 3D Gaussian splatting. *arXiv preprint* arXiv:2401.03890, 2024.

[24] Fei, B.; Xu, J.; Zhang, R.; Zhou, Q.; Yang, W.; He, Y. 3D Gaussian as a new vision era: A survey. *arXiv preprint* arXiv:2402.07181, 2024.

[25] Grossman, J. P.; Dally, W. J. Point sample rendering. In: *Rendering Techniques '98. Eurographics Workshop on Rendering Techniques*. Drettakis, G.; Max, N. Eds. Springer Cham, 181–192, 1998.

[26] Zwicker, M.; Pfister, H.; van Baar, J.; Gross, M. Surface splatting. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 371–378, 2001.

[27] Zwicker, M.; Pfister, H.; van Baar, J.; Gross, M. EWA volume splatting. In: Proceedings of the Visualization, 29–538, 2001.

[28] Botsch, M.; Wiratanaya, A.; Kobbelt, L. Efficient high quality rendering of point sampled geometry. In: Proceedings of the 13th Eurographics Workshop on Rendering, 53–64, 2002.

[29] Botsch, M.; Kobbelt, L. High-quality point-based rendering on modern GPUs. In: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, 335–343, 2003.

[30] Rusinkiewicz, S.; Levoy, M. QSplat: A multiresolution point rendering system for large meshes. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 343–352, 2000.

[31] Kobbelt, L.; Botsch, M. A survey of point-based techniques in computer graphics. *Computers & Graphics* Vol. 28, No. 6, 801–814, 2004.

[32] Chen, Z.; Zhang, H. Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5939–5948, 2019.

[33] Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 165–174, 2019.

[34] 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, 4455–4465, 2019.

[35] Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; Geiger, A. Mip-splatting: Alias-free 3D Gaussian splatting. *arXiv preprint* arXiv:2311.16493, 2023.

[36] Yan, Z.; Low, W. F.; Chen, Y.; Lee, G. H. Multi-scale 3D Gaussian splatting for anti-aliased rendering. *arXiv preprint* arXiv:2311.17089, 2023.

[37] Liang, Z.; Zhang, Q.; Hu, W.; Feng, Y.; Zhu, L.; Jia, K. Analytic-splatting: Anti-aliased 3D Gaussian splatting via analytic integration. *arXiv preprint* arXiv:2403.11056, 2024.

[38] Song, X.; Zheng, J.; Yuan, S.; Gao, H.; Zhao, J.; He, X.; Gu, W.; Zhao, H. SA-GS: Scale-adaptive Gaussian splatting for training-free anti-aliasing. *arXiv preprint* arXiv:2403.19615, 2024.

[39] Malarz, D.; Smolak, W.; Tabor, J.; Tadeja, S.; Spurek, P. Gaussian splatting with NeRF-based color and opacity. *arXiv preprint* arXiv:2312.13729, 2023.

[40] Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; Dai, B. Scaffold-GS: Structured 3D Gaussians for view-adaptive rendering. *arXiv preprint* arXiv:2312.13729, 2023.

[41] Ren, K.; Jiang, L.; Lu, T.; Yu, M.; Xu, L.; Ni, Z.; Dai, B. Octree-GS: Towards consistent real-time rendering with LOD-structured 3D Gaussians. *arXiv preprint* arXiv:2403.17898, 2024.

[42] Radl, L.; Steiner, M.; Parger, M.; Weinrauch, A.; Kerbl, B.; Steinberger, M. Stop ThePop: Sorted Gaussian splatting for view-consistent real-time rendering. *arXiv preprint* arXiv:2402.00525, 2024.

[43] Li, Y.; Lyu, C.; Di, Y.; Zhai, G.; Lee, G. H.; Tombari, F. GeoGaussian: Geometry-aware Gaussian splatting for scene rendering. *arXiv preprint* arXiv:2403.11324, 2024.

[44] Niemeyer, M.; Manhardt, F.; Rakotosaona, M. J.; Oechsle, M.; Duckworth, D.; Gosula, R.; Tateno, K.; Bates, J.; Kaeser, D.; Tombari, F. RadSplat: Radiance field-informed Gaussian splatting for robust real-time rendering with 900+ FPS. *arXiv preprint* arXiv:2403.13806, 2024.

[45] Yang, Z.; Gao, X.; Sun, Y.; Huang, Y.; Lyu, X.; Zhou, W.; Jiao, S.; Qi, X.; Jin, X. Spec-Gaussian: Anisotropic view-dependent appearance for 3D Gaussian splatting. *arXiv preprint* arXiv:2402.15870, 2024.

[46] Franke, L.; Rückert, D.; Fink, L.; Stamminger, M. TRIPS: Trilinear point splatting for real-time radiance field rendering, *Computer Graphics Forum* Vol 43, No. 2, e15012, 2024.

[47] Rückert, D.; Franke, L.; Stamminger, M. ADOP: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics* Vol. 41, No. 4, Article No. 99, 2022.

[48] Zhang, J.; Zhan, F.; Xu, M.; Lu, S.; Xing, E. FreGS: 3D Gaussian splatting with progressive frequency regularization. *arXiv preprint* arXiv:2403.06908, 2024.

[49] Hamdi, A.; Melas-Kyriazi, L.; Qian, G.; Mai, J.; Liu, R.; Vondrick, C.; Ghanem, B.; Vedaldi, A. GES: Generalized exponential splatting for efficient radiance field rendering. *arXiv preprint* arXiv:2402.10128, 2024.

[50] Jung, J.; Han, J.; An, H.; Kang, J.; Park, S.; Kim, S. Relaxing accurate initialization constraint for 3D Gaussian splatting. *arXiv preprint* arXiv:2403.09413, 2024.

[51] Zhang, Z.; Hu, W.; Lao, Y.; He, T.; Zhao, H. Pixel-GS: Density control with pixel-aware gradient for 3D Gaussian splatting. *arXiv preprint* arXiv:2403.15530, 2024.

[52] Bulò, S. R.; Porzi, L.; Kontschieder, P. Revising densification in Gaussian splatting. *arXiv preprint* arXiv:2404.06109, 2024.

[53] Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; Hedman, P. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5470–5479, 2022.

[54] Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P. P. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5855–5864, 2021.

[55] Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; Hedman, P. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 19697–19705, 2023.

[56] Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; Park, E. Compact 3D Gaussian representation for radiance field. *arXiv preprint* arXiv:2311.13681, 2024.

[57] Navaneet, K.; Meibodi, K. P.; Koohpayegani, S. A.; Pirsiavash, H. Compact3D: Compressing Gaussian splat radiance field models with vector quantization. *arXiv preprint* arXiv:2311.18159, 2023.

[58] Niedermayr, S.; Stumpfegger, J.; Westermann, R. Compressed 3D Gaussian splatting for accelerated novel view synthesis. *arXiv preprint* arXiv:2401.02436, 2023.

[59] Girish, S.; Gupta, K.; Shrivastava, A. EAGLES: Efficient accelerated 3D Gaussians with lightweight EncodingS. *arXiv preprint* arXiv:2312.04564, 2023.

[60] Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; Wang, Z. LightGaussian: Unbounded 3D Gaussian compression with 15x reduction and 200+ FPS. *arXiv preprint* arXiv:2311.17245, 2023.

[61] Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; Tagliasacchi, M. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* Vol. 30, 495–507, 2023.

[62] MPEGGroup. mpeg-pcc-tmc13. Available at https://github.com/MPEGGroup/mpeg-pcc-tmc13

[63] Fang, G.; Wang, B. Mini-Splatting: Representing scenes with a constrained number of Gaussians. *arXiv preprint* arXiv:2403.14166, 2024.

[64] Morgenstern, W.; Barthel, F.; Hilsmann, A.; Eisert, P. Compact 3D scene representation via self-organizing Gaussian grids. *arXiv preprint* arXiv:2312.13299, 2023.

[65] Chen, Y.; Wu, Q.; Cai, J.; Harandi, M.; Lin, W. HAC: Hash-grid assisted context for 3D Gaussian splatting compression. *arXiv preprint* arXiv:2403.14530, 2024.

[66] Jo, J.; Kim, H.; Park, J. Identifying unnecessary 3D Gaussians using clustering for fast rendering of 3D Gaussian splatting. *arXiv preprint* arXiv:2402.13827, 2024.

[67] Zhang, X.; Ge, X.; Xu, T.; He, D.; Wang, Y.; Qin,

H.; Lu, G.; Geng, J.; Zhang, J. GaussianImage: 1000 FPS image representation and compression by 2D Gaussian splatting. *arXiv preprint* arXiv:2403.08551, 2024.

[68] Pumarola, A.; Corona, E.; Pons-Moll, G.; Moreno-Noguer, F. D-NeRF: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10318–10327, 2021.

[69] Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; Wang, X. 4D Gaussian splatting for real-time dynamic scene rendering. *arXiv preprint* arXiv:2310.08528, 2023.

[70] Cao, A.; Johnson, J. HexPlane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 130–141, 2023.

[71] Liang, Y.; Khan, N.; Li, Z.; Nguyen-Phuoc, T.; Lanman, D.; Tompkin, J.; Xiao, L. GauFRe: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv preprint* arXiv:2312.11458, 2023.

[72] Sun, J.; Jiao, H.; Li, G.; Zhang, Z.; Zhao, L.; Xing, W. 3DGStream: On-the-fly training of 3D Gaussians for efficient streaming of photo-realistic free-viewpoint videos. *arXiv preprint* arXiv:2403.01444, 2024.

[73] Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; Chen, B. 4D Gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. *arXiv preprint* arXiv:2402.03307, 2024.

[74] Liu, I.; Su, H.; Wang, X. Dynamic Gaussians mesh: Consistent mesh reconstruction from monocular videos. *arXiv preprint* arXiv:2404.12379, 2024.

[75] Guo, Z.; Zhou, W.; Li, L.; Wang, M.; Li, H. Motion-aware 3D Gaussian splatting for efficient dynamic scene reconstruction. *arXiv preprint* arXiv:2403.11447, 2024.

[76] Gao, Q.; Xu, Q.; Cao, Z.; Mildenhall, B.; Ma, W.; Chen, L.; Tang, D.; Neumann, U. GaussianFlow: Splatting Gaussian dynamics for 4D content creation. *arXiv preprint* arXiv:2403.12365, 2024.

[77] Zhang, S.; Zhao, H.; Zhou, Z.; Wu, G.; Zheng, C.; Wang, X.; Liu, W. TOGS: Gaussian splatting with temporal opacity offset for real-time 4D DSA rendering. *arXiv preprint* arXiv:2403.19586, 2024.

[78] Zhang, T.; Gao, Q.; Li, W.; Liu, L.; Chen, B. BAGS: Building animatable Gaussian splatting from a monocular video with diffusion priors. *arXiv preprint* arXiv:2403.11427, 2024.

[79] Katsumata, K.; Vo, D. M.; Nakayama, H. An efficient 3D Gaussian representation for monocular/multi-view dynamic scenes. *arXiv preprint* arXiv:2311.12897, 2023.

[80] Lin, Y.; Dai, Z.; Zhu, S.; Yao, Y. Gaussian-flow: 4D reconstruction with dynamic 3D Gaussian particle. *arXiv preprint* arXiv:2312.03431, 2023.

[81] Li, Z.; Chen, Z.; Li, Z.; Xu Y. Spacetime Gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint* arXiv:2312.16812, 2023.

[82] Kratimenos, A.; Lei, J.; Daniilidis, K. DynMF: Neural motion factorization for real-time dynamic view synthesis with 3D Gaussian splatting. *arXiv preprint* arXiv:2312.00112, 2023.

[83] Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; Tian, Q. Fast dynamic radiance fields with time-aware neural voxels. In: Proceedings of the SIGGRAPH Asia Conference Papers, Article No. 11, 2022.

[84] Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; Liu, Y. Tensor4D: Efficient neural 4D decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16632–16642, 2023.

[85] Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; Kanazawa, A. K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12479–12488, 2023.

[86] Yu, H.; Julin, J.; Milacski, Z. A.; Niinuma, K.; Jeni, L. A. CoGS: Controllable Gaussian splatting. *arXiv preprint* arXiv:2312.05664, 2024.

[87] Yang, Z.; Yang, H.; Pan, Z.; Zhang, L. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. *arXiv preprint* arXiv:2310.10642, 2023.

[88] Shaw, R.; Song, J.; Moreau, A.; Nazarczuk, M.; Catley-Chandar, S.; Dhamo, H.; Perez-Pellitero, E. SWAGS: Sampling windows adaptively for dynamic 3D Gaussian splatting. *arXiv preprint* arXiv:2312.13308, 2023.

[89] Maggioni, M.; Tanay, T.; Babiloni, F.; McDonagh, S.; Leonardis, A. Tunable convolutions with parametric multi-loss optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20226–20236, 2023.

[90] Cotton, R. J.; Peyton, C. Dynamic Gaussian splatting from markerless motion capture reconstruct infants movements. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, 60–68, 2024.

[91] Zhu, L.; Wang, Z.; Cui, J.; Jin, Z.; Lin, G.;

Yu, L. EndoGS: Deformable endoscopic tissues reconstruction with Gaussian splatting. *arXiv preprint* arXiv:2401.11535, 2024.

[92] Chen, Y.; Wang, H. EndoGaussians: Single view dynamic Gaussian splatting for deformable endoscopic tissues reconstruction. *arXiv preprint* arXiv:2401.13352, 2024.

[93] Huang, Y.; Cui, B.; Bai, L.; Guo, Z.; Xu, M.; Islam, M.; Ren, H. Endo-4DGS: Endoscopic monocular scene reconstruction with 4D Gaussian splatting. *arXiv preprint* arXiv:2401.16416, 2024.

[94] Wang, K.; Yang, C.; Wang, Y.; Li, S.; Wang, Y.; Dou, Q.; Yang, X.; Shen, W. EndoGSLAM: Real-time dense reconstruction and tracking in endoscopic surgeries using Gaussian platting. *arXiv preprint* arXiv:2403.15124, 2024.

[95] Zhu, Z.; Fan, Z.; Jiang, Y.; Wang, Z. FSGS: Real-time few-shot view synthesis using Gaussian splatting. *arXiv preprint* arXiv:2312.00451, 2023.

[96] Xiong, H.; Muttukuru, S.; Upadhyay, R.; Chari, P.; Kadambi, A. SparseGS: Real-time 360° sparse view synthesis using Gaussian splatting. *arXiv preprint* arXiv:2312.00206, 2023.

[97] Paliwal, A.; Ye, W.; Xiong, J.; Kotovenko, D.; Ranjan, R.; Chandra, V.; Kalantari, N. K. CoherentGS: Sparse novel view synthesis with coherent 3D Gaussians. *arXiv preprint* arXiv:2403.19495, 2024.

[98] Li, J.; Zhang, J.; Bai, X.; Zheng, J.; Ning, X.; Zhou, J.; Gu, L. DNGaussian: Optimizing sparse-view 3D Gaussian radiance fields with global-local depth normalization. *arXiv preprint* arXiv:2403.06912, 2024.

[99] Poole, B.; Jain, A.; Barron, J. T.; Mildenhall, B.; Feng, L.; Wang, M.; Wang, M.; Xu, K.; Liu, X. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint* arXiv:2209.14988, 2022.

[100] Yang, C.; Li, S.; Fang, J.; Liang, R.; Xie, L.; Zhang, X.; Shen, W.; Tian, Q. GaussainObject: Just taking four images to get a high-quality 3D object with Gaussian splatting. *arXiv preprint* arXiv:2402.10259, 2024.

[101] Zhang, L.; Rao, A.; Agrawala, M. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3813–3824, 2023.

[102] Charatan, D.; Li, S.; Tagliasacchi, A.; Sitzmann, V. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. *arXiv preprint* arXiv:2312.12337, 2023.

[103] Yu, A.; Ye, V.; Tancik, M.; Kanazawa, A. pixelNeRF: Neural radiance fields from one or few images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4578–4587, 2021.

[104] Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T. J.; Cai, J. MVSplat: Efficient 3D Gaussian splatting from sparse multi-view images. *arXiv preprint* arXiv:2403.14627, 2024.

[105] Szymanowicz, S.; Rupprecht, C.; Vedaldi, A. Splatter image: Ultra-fast single-view 3D reconstruction. *arXiv preprint* arXiv:2312.13150, 2023.

[106] Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint* arXiv:1505.04597, 2015.

[107] Chen, Y.; Gu, C.; Jiang, J.; Zhu, X.; Zhang, L. Periodic vibration Gaussian: Dynamic urban scene reconstruction and realtime rendering. *arXiv preprint* arXiv:2311.18561, 2023.

[108] Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; Yang, M. DrivingGaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint* arXiv:2312.07920, 2023.

[109] Zhou, H.; Shao, J.; Xu, L.; Bai, D.; Qiu, W.; Liu, B.; Wang, Y.; Geiger, A.; Liao, Y. HUGS: Holistic urban 3D scene understanding via Gaussian splatting. *arXiv preprint* arXiv:2403.12722, 2024.

[110] Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W. Y.; et al. Segment anything. *arXiv preprint* arXiv:2304.02643, 2023.

[111] Yan, Y.; Lin, H.; Zhou, C.; Wang, W.; Sun, H.; Zhan, K.; Lang, X.; Zhou, X.; Peng, S. Street Gaussians for modeling dynamic urban scenes. *arXiv preprint* arXiv:2401.01339, 2024.

[112] Yu, Z.; Wang, H.; Yang, J.; Wang, H.; Xie, Z.; Cai, Y.; Cao, J.; Ji, Z.; Sun, M. SGD: Street view synthesis with Gaussian splatting and diffusion prior. *arXiv preprint* arXiv:2403.20079, 2024.

[113] Wu, R.; Mildenhall, B.; Henzler, P.; Park, K.; Gao, R.; Watson, D.; Srinivasan, P. P.; Verbin, D.; Barron, J. T.; Poole, B.; Holynski, A. ReconFusion: 3D reconstruction with diffusion priors. *arXiv preprint* arXiv:2312.02981, 2023.

[114] Wu, K.; Zhang, K.; Zhang, Z.; Yuan, S.; Tie, M.; Wei, J.; Xu, Z.; Zhao, J.; Gan, Z.; Ding, W. HGS-mapping: Online dense mapping using hybrid Gaussian representation in urban scenes. *arXiv preprint* arXiv:2403.20159, 2024.

[115] Lin, J.; Li, Z.; Tang, X.; Liu, J.; Liu, S.; Liu, J.; Lu, Y.; Wu, X.; Xu, S.; Yan, Y.; Yang, W. VastGaussian: Vast 3D Gaussians for large scene reconstruction. *arXiv preprint* arXiv:2402.17427, 2024.

[116] Liu, Y.; Guan, H.; Luo, C.; Fan, L.; Peng, J.; Zhang, Z. CityGaussian: Real-time high-quality large-scale scene rendering with Gaussians. *arXiv preprint* arXiv:2404.01133, 2024.

[117] Xiong, B.; Li, Z.; Li, Z. GauU-scene: A scene reconstruction benchmark on large scale 3D reconstruction dataset using Gaussian splatting. *arXiv preprint* arXiv:2401.14032, 2024.

[118] Fu, Y.; Liu, S.; Kulkarni, A.; Kautz, J.; Efros, A. A.; Wang, X. COLMAP-free 3D Gaussian splatting. *arXiv preprint* arXiv:2312.07504, 2023.

[119] Sun, Y.; Wang, X.; Zhang, Y.; Zhang, J.; Jiang, C.; Guo, Y.; Wang, F. iComMa: Inverting 3D Gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint* arXiv:2312.09031, 2023.

[120] Fan, Z.; Cong, W.; Wen, K.; Wang, K.; Zhang, J.; Ding, X.; Xu, D.; Ivanovic, B.; Pavone, M.; Pavlakos, G.; Wang, Z.; Wang, Y. InstantSplat: Unbounded sparse-view pose-free Gaussian splatting in 40 seconds. *arXiv preprint* arXiv:2403.20309, 2024.

[121] Li, H.; Gao, Y.; Wu, C.; Zhang, D.; Dai, Y.; Zhao, C.; Feng, H.; Ding, E.; Wang, J.; Han, J. GGRt: Towards pose-free generalizable 3D Gaussian splatting in real-time. *arXiv preprint* arXiv:2403.10147, 2024.

[122] Lee, B.; Lee, H.; Sun, X.; Ali, U.; Park, E. Deblurring 3D Gaussian splatting. *arXiv preprint* arXiv:2401.00834, 2024.

[123] Peng, C.; Tang, Y.; Zhou, Y.; Wang, N.; Liu, X.; Li, D.; Chellappa, R. BAGS: Blur agnostic Gaussian splatting through multi-scale kernel modeling. *arXiv preprint* arXiv:2403.04926, 2024.

[124] Zhao, L.; Wang, P.; Liu, P. BAD-Gaussians: Bundle adjusted deblur Gaussian splatting. *arXiv preprint* arXiv:2403.11831, 2024.

[125] Seiskari, O.; Ylilammi, J.; Kaatrasalo, V.; Rantalankila, P.; Turkulainen, M.; Kannala, J.; Rahtu, E.; Solin, A. Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion. *arXiv preprint* arXiv:2403.13327, 2024.

[126] Dahmani, H.; Bennehar, M.; Piasco, N.; Roldao, L.; Tsishkou, D. SWAG: Splatting in the wild images with appearance-conditioned Gaussians. *arXiv preprint* arXiv:2403.10427, 2024.

[127] Zhang, D.; Wang, C.; Wang, W.; Li, P.; Qin, M.; Wang, H. Gaussian in the wild: 3D Gaussian splatting for unconstrained image collections. *arXiv preprint* arXiv:2403.15704, 2024.

[128] Meng, J.; Li, H.; Wu, Y.; Gao, Q.; Yang, S.; Zhang, J.; Ma, S. Mirror-3DGS: Incorporating mirror teflections into 3D Gaussian splatting. *arXiv preprint* arXiv:2404.01168, 2024.

[129] Comi, M.; Tonioni, A.; Yang, M.; Tremblay, J.; Blukis, V.; Lin, Y.; Lepora, N. F.; Aitchison, L. Snap-it, tap-it, splat-it: Tactile-informed 3D Gaussian splatting for reconstructing challenging surfaces. *arXiv preprint* arXiv:2403.20275, 2024.

[130] Li, Y.; Fu, X.; Zhao, S.; Jin, R.; Zhou, S. K. Sparse-view CT reconstruction with 3D Gaussian volumetric representation. *arXiv preprint* arXiv:2312.15676, 2023.

[131] Cai, Y.; Liang, Y.; Wang, J.; Wang, A.; Zhang, Y.; Yang, X.; Zhou, Z.; Yuille, A. Radiative Gaussian splatting for efficient X-ray novel view synthesis. *arXiv preprint* arXiv:2403.04116, 2024.

[132] Bai, J.; Huang, L.; Guo, J.; Gong, W.; Li, Y.; Guo, Y. 360-GS: Layout-guided panoramic Gaussian splatting for indoor roaming. *arXiv preprint* arXiv:2402.00763, 2024.

[133] Nguyen, V. M.; Sandidge, E.; Mahendrakar, T.; White, R. T. Characterizing satellite geometry via accelerated 3D Gaussian splatting. *Aerospace* Vol. 11, No. 3, Article No. 183, 2024.

[134] Chen, Y.; Chen, Z.; Zhang, C.; Wang, F.; Yang, X.; Wang, Y.; Cai, Z.; Yang, L.; Liu, H.; Lin, G. GaussianEditor: Swift and controllable 3D editing with Gaussian splatting. *arXiv preprint* arXiv:2311.14521, 2023.

[135] Ye, M.; Danelljan, M.; Yu, F.; Ke, L. Gaussian grouping: Segment and edit anything in 3D scenes. *arXiv preprint* arXiv:2312.00732, 2023.

[136] Huang, J.; Yu, H. Point'n Move: Interactive scene object manipulation on Gaussian splatting radiance fields. *arXiv preprint* arXiv:2311.16737, 2023.

[137] Feng, Q.; Cao, G.; Chen, H.; Mu, T. J.; Martin, R. R.; Hu, S. M. A new split algorithm for 3D Gaussian splatting. *arXiv preprint* arXiv:2403.09143, 2024.

[138] Guédon, A.; Lepetit, V. Gaussian frosting: Editable complex radiance fields with real-time rendering. *arXiv preprint* arXiv:2403.14554, 2024.

[139] Fang, J.; Wang, J.; Zhang, X.; Xie, L.; Tian, Q. GaussianEditor: Editing 3D Gaussians delicately with text instructions. *arXiv preprint* arXiv:2311.16037, 2023.

[140] 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, 10674–10685, 2022.

[141] Haque, A.; Tancik, M.; Efros, A. A.; Holynski, A.; Kanazawa, A. Instruct-NeRF$_2$NeRF: Editing 3D scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 19683–19693, 2023.

[142] Palandra, F.; Sanchietti, A.; Baieri, D.; Rodolà, E. GSEdit: Efficient text-guided editing of 3D objects via gaussian splatting. *arXiv preprint* arXiv:2403.05154, 2024.

[143] Brooks, T.; Holynski, A.; Efros, A. A. InstructPix2Pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18392–18402, 2023.

[144] Wu, J.; Bian, J. W.; Li, X.; Wang, G.; Reid, I.; Torr, P.; Prisacariu, V. A. GaussCtrl: Multi-view consistent text-driven 3D Gaussian splatting editing. *arXiv preprint* arXiv:2403.08733, 2024.

[145] Wang, Y.; Yi, X.; Wu, Z.; Zhao, N.; Chen, L.; Zhang, H. View-consistent 3D editing with Gaussian splatting. *arXiv preprint* arXiv:2403.11868, 2024.

[146] Xu, T. X.; Hu, W.; Lai, Y. K.; Shan, Y.; Zhang, S. H. Texture-GS: Disentangling the geometry and texture for 3D Gaussian splatting editing. *arXiv preprint* arXiv:2403.10050, 2024.

[147] Wang, X. E.; Sin, Z. P. T. 3D Gaussian model for animation and texturing. *arXiv preprint* arXiv:2402.19441, 2024.

[148] Liu, K.; Zhan, F.; Xu, M.; Theobalt, C.; Shao, L.; Lu, S. Style-Gaussian: Instant 3D style transfer with Gaussian splatting. *arXiv preprint* arXiv:2403.07807, 2024.

[149] Saroha, A.; Gladkova, M.; Curreli, C.; Yenamandra, T.; Cremers, D. Gaussian splatting in style. *arXiv preprint* arXiv:2403.08498, 2024.

[150] Zhang, D.; Chen, Z.; Yuan, Y. J.; Zhang, F. L.; He, Z.; Shan, S.; Gao, L. StylizedGS: Controllable stylization for 3D Gaussian splatting. *arXiv preprint* arXiv:2404.05220, 2024.

[151] Liang, Z.; Zhang, Q.; Feng, Y.; Shan, Y.; Jia, K. GS-IR: 3D Gaussian splatting for inverse rendering. *arXiv preprint* arXiv:2311.16473, 2023.

[152] Shi, Y.; Wu, Y.; Wu, C.; Liu, X.; Zhao, C.; Feng, H.; Liu, J.; Zhang, L.; Zhang, J.; Zhou, B.; et al. GIR: 3D Gaussian inverse rendering for relightable scene factorization. *arXiv preprint* arXiv:2312.05133, 2023.

[153] Verbin, D.; Hedman, P.; Mildenhall, B.; Zickler, T.; Barron, J. T.; Srinivasan, P. P. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5481–5490, 2022.

[154] Wu, T.; Sun, J. M.; Lai, Y. K.; Ma, Y.; Kobbelt, L.; Gao, L. DeferredGS: Decoupled and editable Gaussian splatting with deferred shading. *arXiv preprint* arXiv:2404.09412, 2024.

[155] Bonet, J.; Wood, R. D. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge, UK: Cambridge University Press, 2008.

[156] Feng, Y.; Feng, X.; Shang, Y.; Jiang, Y.; Yu, C.; Zong, Z.; Shao, T.; Wu, H.; Zhou, K.; Jiang, C.; et al. Gaussian splashing: Dynamic fluid synthesis with Gaussian splatting. *arXiv preprint* arXiv:2401.15318, 2024.

[157] Macklin, M.; Müller, M.; Chentanez, N. XPBD: Position-based simulation of compliant constrained dynamics. In: Proceedings of the 9th International Conference on Motion in Games, 49–54, 2016.

[158] Zhong, L.; Yu, H. X.; Wu, J.; Li, Y. Reconstruction and simulation of elastic objects with spring-mass 3D Gaussians. *arXiv preprint* arXiv:2403.09434, 2024.

[159] Qiu, R. Z.; Yang, G.; Zeng, W.; Wang, X. Feature splatting: Language-driven physics-based scene synthesis and editing. *arXiv preprint* arXiv:2404.01223, 2024.

[160] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In: Proceedings of the International conference on Machine Learning, 8748–8763, 2021.

[161] Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9650–9660, 2021.

[162] Cen, J.; Fang, J.; Yang, C.; Xie, L.; Zhang, X.; Shen, W.; Tian, Q. Segment any 3D Gaussians. *arXiv preprint* arXiv:2312.00860, 2023.

[163] Zhou, S.; Chang, H.; Jiang, S.; Fan, Z.; Zhu, Z.; Xu, D.; Chari, P.; You, S.; Wang, Z.; Kadambi, A. Feature 3DGS: Supercharging 3D Gaussian splatting to enable distilled feature field. *arXiv preprint* arXiv:2312.03203, 2023.

[164] Qin, M.; Li, W.; Zhou, J.; Wang, H.; Pfister, H. LangSplat: 3D language Gaussian splatting. *arXiv preprint* arXiv:2312.16084, 2023.

[165] Hu, X.; Wang, Y.; Fan, L.; Fan, J.; Peng, J.; Lei, Z.; Li, Q.; Zhang, Z. SAGD: Boundary-enhanced segment anything in 3D Gaussian via Gaussian decomposition. *arXiv preprint* arXiv:2401.17857, 2024.

[166] Guo, J.; Ma, X.; Fan, Y.; Liu, H.; Li, Q. Semantic Gaussians: Open-vocabulary scene understanding with 3D Gaussian splatting. *arXiv preprint* arXiv:2403.15624, 2024.

[167] Lyu, W.; Li, X.; Kundu, A.; Tsai, Y. H.; Yang, M. H. Gaga: Group any Gaussians via 3D-aware memory bank. *arXiv preprint* arXiv:2404.07977, 2024.

[168] Shi, J. C.; Wang, M.; Duan, H. B.; Guan, S. H. Language embedded 3D Gaussians for open-vocabulary scene understanding. *arXiv preprint* arXiv:2311.18482, 2023.

[169] Zuo, X.; Samangouei, P.; Zhou, Y.; Di, Y.; Li, M. FMGS: Foundation model embedded 3D Gaussian splatting for holistic 3D scene understanding. *arXiv preprint* arXiv:2401.01970, 2024.

[170] Dou, B.; Zhang, T.; Ma, Y.; Wang, Z.; Yuan, Z. CoSSegGaussians: Compact and swift scene segmenting 3D Gaussians with dual feature fusion. *arXiv preprint* arXiv:2401.05925, 2024.

[171] Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 27171–27183, 2021.

[172] Liu, Y. T.; Wang, L.; Yang, J.; Chen, W.; Meng, X.; Yang, B.; Gao, L. NeUDF: Leaning neural unsigned distance fields with volume rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 237–247, 2023.

[173] Zhuang, Y.; Zhang, Q.; Feng, Y.; Zhu, H.; Yao, Y.; Li, X.; Cao, Y. P.; Shan, Y.; Cao, X. Anti-aliased neural implicit surfaces with encoding level of detail. In: Proceedings of the SIGGRAPH Asia Conference Papers, Article No. 119, 2023.

[174] Ge, W.; Hu, T.; Zhao, H.; Liu, S.; Chen, Y. C. Ref-NeuS: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4228–4237, 2023.

[175] Chen, H.; Li, C.; Lee, G. H. NeuSG: Neural implicit surface reconstruction with 3D Gaussian splatting guidance. *arXiv preprint* arXiv:2312.00846, 2023.

[176] Lyu, X.; Sun, Y. T.; Huang, Y. H.; Wu, X.; Yang, Z.; Chen, Y.; Pang, J.; Qi, X. 3DGSR: Implicit surface reconstruction with 3D Gaussian splatting. *arXiv preprint* arXiv:2404.00409, 2024.

[177] Yu, M.; Lu, T.; Xu, L.; Jiang, L.; Xiangli, Y.; Dai, B. GSDF: 3DGS meets SDF for improved rendering and reconstruction. *arXiv preprint* arXiv:2403.16964, 2024.

[178] Turkulainen, M.; Ren, X.; Melekhov, I.; Seiskari, O.; Rahtu, E.; Kannala, J. DN-Splatter: Depth and normal priors for Gaussian splatting and meshing. *arXiv preprint* arXiv:2403.17822, 2024.

[179] Wolf, Y.; Bracha, A.; Kimmel, R. Surface reconstruction from Gaussian splatting via novel stereo views. *arXiv preprint* arXiv:2403.17822, 2024.

[180] Yan, C.; Qu, D.; Wang, D.; Xu, D.; Wang, Z.; Zhao, B.; Li, X. GSSLAM: Dense visual SLAM with 3D Gaussian splatting. *arXiv preprint* arXiv:2311.11700, 2023.

[181] Keetha, N. V.; Karhade, J.; Jatavallabhula, K. M.; Yang, G.; Scherer, S. A.; Ramanan, D.; Luiten, J. SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM. *arXiv preprint* arXiv:2312.02126, 2024.

[182] Matsuki, H.; Murai, R.; Kelly, P. H. J.; Davison, A. J. Gaussian splatting SLAM. *arXiv preprint* arXiv:2312.06741, 2023.

[183] Yugay, V.; Li, Y.; Gevers, T.; Oswald, M. R. Gaussian-SLAM: Photo-realistic dense SLAM with Gaussian splatting. *arXiv preprint* arXiv:2312.10070, 2023.

[184] Hong, S.; He, J.; Zheng, X.; Zheng, C.; Shen, S. LIV-GaussMap: LiDAR-inertial-visual fusion for real-time 3D radiance field map rendering. *arXiv preprint* arXiv:2401.14857, 2024.

[185] Li, M.; Liu, S.; Zhou, H. SGS-SLAM: Semantic Gaussian splatting for neural dense SLAM. *arXiv preprint* arXiv:2402.03246, 2024.

[186] Ji, Y.; Liu, Y.; Xie, G.; Ma, B.; Xie, Z. NEDS-SLAM: A novel neural explicit dense semantic SLAM framework using 3D Gaussian splatting. *arXiv preprint* arXiv:2403.11679, 2024.

[187] Zhu, S.; Qin, R.; Wang, G.; Liu, J.; Wang, H. SemGauss-SLAM: Dense semantic Gaussian splatting SLAM. *arXiv preprint* arXiv:2403.07494, 2024.

[188] Deng, T.; Chen, Y.; Zhang, L.; Yang, J.; Yuan, S.; Wang, D.; Chen, W. Compact 3D Gaussian splatting for dense visual SLAM. *arXiv preprint* arXiv:2403.11247, 2024.

[189] Hu, J.; Chen, X.; Feng, B.; Li, G.; Yang, L.; Bao, H.; Zhang, G.; Cui, Z. CG-SLAM: Efficient dense RGB-D SLAM in a consistent uncertainty-aware 3D Gaussian field. *arXiv preprint* arXiv:2403.16095, 2024.

[190] Jiang, P.; Pandey, G.; Saripalli, S. 3DGS-ReLoc: 3D Gaussian splatting for map representation and visual ReLocalization. *arXiv preprint* arXiv:2403.11367, 2024.

[191] Chen, T.; Shorinwa, O.; Zeng, W.; Bruno, J.; Dames, P.; Schwager, M. Splat-Nav: Safe real-time robot navigation in Gaussian splatting maps. *arXiv preprint* arXiv:2403.02751, 2024.

[192] Lei, X.; Wang, M.; Zhou, W.; Li, H. GaussNav: Gaussian splatting for visual navigation. *arXiv preprint* arXiv:2403.11625, 2024.

[193] Liu, G.; Jiang, W.; Lei, B.; Pandey, V.; Daniilidis, K.; Motee, N. Beyond uncertainty: Risk-aware active view acquisition for safe robot navigation and 3D scene understanding with FisherRF. *arXiv preprint* arXiv:2403.11396, 2024.

[194] Cai, D.; Heikkilä, J.; Rahtu, E. GS-pose: Cascaded

framework for generalizable segmentation-based 6D object pose estimation. *arXiv preprint* arXiv: 2403.11247, 2024.

[195] Sun, L. C.; Bhatt, N. P.; Liu, J. C.; Fan, Z.; Wang, Z.; Humphreys, T. E.; Topcu, U. MM3DGS SLAM: Multi-modal 3D Gaussian splatting for SLAM using vision, depth, and inertial measurements. *arXiv preprint* arXiv:2404.00923, 2024.

[196] Herau, Q.; Bennehar, M.; Moreau, A.; Piasco, N.; Roldao, L.; Tsishkou, D.; Migniot, C.; Vasseur, P.; Demonceaux, C. 3DGSCalib: 3D Gaussian splatting for multimodal spatiotemporal calibration. *arXiv preprint* arXiv:2403.11577, 2024.

[197] Lu, G.; Zhang, S.; Wang, Z.; Liu, C.; Lu, J.; Tang, Y. ManiGaussian: Dynamic Gaussian splatting for multi-task robotic manipulation. *arXiv preprint* arXiv:2403.08321, 2024.

[198] Zheng, Y.; Chen, X.; Zheng, Y.; Gu, S.; Yang, R.; Jin, B.; Li, P.; Zhong, C.; Wang, Z.; Liu, L.; et al. GaussianGrasper: 3D language Gaussian splatting for open-vocabulary robotic grasping. *arXiv preprint* arXiv:2403.09637, 2024.

[199] Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C. Y.; Verma, S.; et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint* arXiv:1906.05797, 2019.

[200] Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; Pollefeys, M. NICE-SLAM: Neural implicit scalable encoding for SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12776–12786, 2022.

[201] Yang, X.; Li, H.; Zhai, H.; Ming, Y.; Liu, Y.; Zhang, G. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, 499–507, 2022.

[202] Wang, H.; Wang, J.; Agapito, L. Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13293–13302, 2023.

[203] Zielonka, W.; Bagautdinov, T.; Saito, S.; Zollhöfer, M.; Thies, J.; Romero, J. Drivable 3D Gaussian avatars. *arXiv preprint* arXiv:2311.08581, 2023.

[204] Jena, R.; Iyer, G. S.; Choudhary, S.; Smith, B.; Chaudhari, P.; Gee, J. SplatArmor: Articulated Gaussian splatting for animatable humans from monocular RGB videos. *arXiv preprint* arXiv: 2311.10812, 2023.

[205] Moreau, A.; Song, J.; Dhamo, H.; Shaw, R.; Zhou, Y.; Pérez-Pellitero, E. Human Gaussian splatting: Real-time rendering of animatable avatars. *arXiv preprint* arXiv:2311.17113, 2023.

[206] Kocabas, M.; Chang, J. H R.; Gabriel, J.; Tuzel, O.; Ranjan, A. HUGS: Human Gaussian splats. *arXiv preprint* arXiv:2311.17910, 2023.

[207] Chan, E. R.; Lin, C. Z.; Chan, M. A.; Nagano, K.; Pan, B.; de Mello, S.; Gallo, O.; Guibas, L.; Tremblay, J.; Khamis, S.; et al. Efficient geometry-aware 3D generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16123–16133, 2022.

[208] Jiang, Y.; Shen, Z.; Wang, P.; Su, Z.; Hong, Y.; Zhang, Y.; Yu, J.; Xu, L. HiFi4G: High-fidelity human performance rendering via compact gaussian splatting. *arXiv preprint* arXiv:2312.03461, 2023.

[209] Lei, J.; Wang, Y.; Pavlakos, G.; Liu, L.; Daniilidis, K. GART: Gaussian articulated template models. *arXiv preprint* arXiv:2311.16099, 2023.

[210] Li, Z.; Zheng, Z.; Wang, L.; Liu, Y. Animatable Gaussians: Learning pose-dependent Gaussian maps for high-fidelity human avatar modeling. *arXiv preprint* arXiv:2311.16099, 2023.

[211] Abdal, R.; Yifan, W.; Shi, Z.; Xu, Y.; Po, R.; Kuang, Z.; Chen, Q.; Yeung, D. Y.; Wetzstein, G. Gaussian shell maps for efficient 3D human generation. *arXiv preprint* arXiv:2311.17857, 2023.

[212] Pang, H.; Zhu, H.; Kortylewski, A.; Theobalt, C.; Habermann, M. ASH: Animatable Gaussian splats for efficient and photoreal human rendering. *arXiv preprint* arXiv:2312.05941, 2023.

[213] Qian, Z.; Wang, S.; Mihajlovic, M.; Geiger, A.; Tang, S. 3DGSAvatar: Animatable avatars via deformable 3D Gaussian splatting. *arXiv preprint* arXiv:2312.09228, 2023.

[214] Li, M.; Yao, S.; Xie, Z.; Chen, K. GaussianBody: Clothed human reconstruction via 3D Gaussian splatting. *arXiv preprint* arXiv:2401.09720, 2024.

[215] Hu, S.; Liu, Z. GauHuman: Articulated Gaussian splatting from monocular human videos. *arXiv preprint* arXiv:2312.02973, 2023.

[216] Hu, L.; Zhang, H.; Zhang, Y.; Zhou, B.; Liu, B.; Zhang, S.; Nie, L. GaussianAvatar: Towards realistic human avatar modeling from a single video via animatable 3D gaussians. *arXiv preprint* arXiv:2312.02134, 2023.

[217] Li, M.; Tao, J.; Yang, Z.; Yang, Y. Human101: Training 100+FPS human Gaussians in 100s from 1 view. *arXiv preprint* arXiv:2312.15258, 2023.

[218] Shao, Z.; Wang, Z.; Li, Z.; Wang, D.; Lin, X.; Zhang, Y.; Fan, M.; Wang, Z. SplattingAvatar: Realistic real-time human avatars with mesh-embedded Gaussian splatting. *arXiv preprint* arXiv:2403.05087, 2024.

[219] Wen, J.; Zhao, X.; Ren, Z.; Schwing, A. G.; Wang, S. GoMAvatar: Efficient animatable human modeling from monocular video using Gaussians-on-mesh. *arXiv preprint* arXiv:2404.07991, 2024.

[220] Liu, X.; Wu, C.; Liu, J.; Liu, X.; Zhao, C.; Feng, H.; Ding, E.; Wang, J. GVA: Reconstructing vivid 3D Gaussian avatars from monocular videos. *arXiv preprint* arXiv:2404.07991, 2024.

[221] Svitov, D.; Morerio, P.; Agapito, L.; Bue, A. D. HAHA: Highly articulated Gaussian human avatars with textured mesh prior. *arXiv preprint* arXiv:2404.01053, 2024.

[222] Chen, Y.; Wang, L.; Li, Q.; Xiao, H.; Zhang, S.; Yao, H.; Liu, Y. MonoGussianAvatar: Monocular Gaussian point-based head avatar. *arXiv preprint* arXiv:2404.01053, 2024.

[223] Zhao, Z.; Bao, Z.; Li, Q.; Qiu, G.; Liu, K. PSAvatar: A point-based morphable shape model for real-time head avatar animation with 3D Gaussian splatting. *arXiv preprint* arXiv:2401.12900, 2024.

[224] Li, T.; Bolkart, T.; Black, M. J.; Li, H.; Romero, J. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics* Vol. 36, No. 6, Article No. 194, 2017.

[225] Wang, J.; Xie, J. C.; Li, X.; Xu, F.; Pun, C. M.; Gao, H. GaussianHead: High-fidelity head avatars with learnable Gaussian derivation. *arXiv preprint* arXiv:2312.01632, 2023.

[226] Qian, S.; Kirschstein, T.; Schoneveld, L.; Davoli, D.; Giebenhain, S.; Nießner, M. GaussianAvatars: Photorealistic head avatars with rigged 3D Gaussians. *arXiv preprint* arXiv:2312.02069, 2023.

[227] Rivero, A.; Athar, S.; Shu, Z.; Samaras, D. Rig3DGS: Creating controllable portraits from casual monocular videos. *arXiv preprint* arXiv:2402.03723, 2024.

[228] Dhamo, H.; Nie, Y.; Moreau, A.; Song, J.; Shaw, R.; Zhou, Y.; Pérez-Pellitero, E. HeadGaS: Real-time animatable head avatars via 3D Gaussian splatting. *arXiv preprint* arXiv:2312.02902, 2023.

[229] Blanz, V.; Vetter, T. A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 187–194, 2023.

[230] Xiang, J.; Gao, X.; Guo, Y.; Zhang, J. FlashAvatar: High-fidelity digital avatar rendering at 300FPS. *arXiv preprint* arXiv:2312.02214, 2023.

[231] Xu, Y.; Chen, B.; Li, Z.; Zhang, H.; Wang, L.; Zheng, Z.; Liu, Y. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic Gaussians. *arXiv preprint* arXiv:2312.03029, 2023.

[232] Luo, J.; Liu, J.; Davis, J. SplatFace: Gaussian splat face reconstruction leveraging an optimizable surface. *arXiv preprint* arXiv:2403.18784, 2024.

[233] Xiao, Y.; Wang, X.; Li, J.; Cai, H.; Fan, Y.; Xue, N.; Yang, M.; Shen, Y.; Gao, S. Bridging 3D Gaussian and mesh for freeview video rendering. *arXiv preprint* arXiv:2403.11453, 2024.

[234] Zhou, Z.; Ma, F.; Fan, H.; Yang, Y. HeadStudio: Text to animatable head avatars with 3D Gaussian splatting. *arXiv preprint* arXiv:2402.06149, 2024.

[235] Stanishevskii, G.; Steczkiewicz, J.; Szczepanik, T.; Tadeja, S.; Tabor, J.; Spurek, P. ImplicitDeepfake: Plausible face-swapping through implicit deepfake generation using NeRF and Gaussian splatting. *arXiv preprint* arXiv:2402.06390, 2024.

[236] Saito, S.; Schwartz, G.; Simon, T.; Li, J.; Nam, G. Relightable Gaussian codec avatars. *arXiv preprint* arXiv:2312.03704, 2023.

[237] Jiang, Z.; Rahmani, H.; Black, S.; Williams, B. M. 3D points splatting for real-time dynamic hand reconstruction. *arXiv preprint* arXiv:2312.13770, 2023.

[238] Pokhariya, C.; Shah, I. N.; Xing, A.; Li, Z.; Chen, K.; Sharma, A.; Sridhar, S. MANUS: Markerless grasp capture using articulated 3D Gaussians. *arXiv preprint* arXiv:2312.02137, 2023.

[239] Luo, H.; Ouyang, M.; Zhao, Z.; Jiang, S.; Zhang, L.; Zhang, Q.; Yang, W.; Xu, L.; Yu, J. GaussianHair: Hair modeling and rendering with light-aware Gaussians. *arXiv preprint* arXiv:2402.10483, 2024.

[240] Marschner, S.; Jensen, H.; Cammarano, M.; Worley, S.; Hanrahan, P. Light scattering from human hair fibers. *ACM Transactions on Graphics* Vol. 22, No. 3, 780–791, 2003.

[241] Lin, C. H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M. Y.; Lin, T. Y. Magic3D: High-resolution text-to-3D content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 300–309, 2023.

[242] Chen, Z.; Wang, F.; Liu, H. Text-to-3D using Gaussian splatting. *arXiv preprint* arXiv:2309.16585, 2023.

[243] Nichol, A.; Jun, H.; Dhariwal, P.; Mishkin, P.; Chen, M. Point-E: A system for generating 3D point clouds from complex prompts. *arXiv preprint* arXiv:2212.08751, 2022.

[244] Yi, T.; Fang, J.; Wang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; Wang, X. GaussianDreamer: Fast generation from text to 3D Gaussians by bridging 2D and 3D diffusion models. *arXiv preprint* arXiv:2310.08529, 2023.

[245] Jun, H.; Nichol, A. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint* arXiv:2305.02463, 2023.

[246] Shi, Y.; Wang, P.; Ye, J.; Long, M.; Li, K.; Yang, X. MVDream: Multi-view diffusion for 3D generation. *arXiv preprint* arXiv:2308.16512, 2023.

[247] Wang, P.; Shi, Y. ImageDream: Image-prompt multi-view diffusion for 3D generation. *arXiv preprint* arXiv:2312.02201, 2023.

[248] Yu, Y.; Zhu, S.; Qin, H.; Li, H. BoostDream: Efficient refining for high-quality text-to-3D generation from multi-view diffusion. *arXiv preprint* arXiv:2401.16764, 2024.

[249] Shen, T.; Gao, J.; Yin, K.; Liu, M. Y.; Fidler, S. Deep marching tetrahedra: A hybrid representation for high-resolution 3D shape synthesis. *arXiv preprint* arXiv:2111.04276, 2021.

[250] Liang, Y.; Yang, X.; Lin, J.; Li, H.; Xu, X.; Chen, Y. LucidDreamer: Towards high-fidelity text-to-3D generation via interval score matching. *arXiv preprint* arXiv:2311.11284, 2023.

[251] Li, X.; Wang, H.; Tseng, K. K. GaussianDiffusion: 3D Gaussian splatting for denoising diffusion probabilistic models with structured noise. *arXiv preprint* arXiv:2311.11221, 2023.

[252] Yang, X.; Chen, Y.; Chen, C.; Zhang, C.; Xu, Y.; Yang, X.; Liu, F.; Lin, G. Learn to optimize denoising scores for 3D generation: A unified and improved diffusion prior on NeRF and 3D Gaussian splatting. *arXiv preprint* arXiv:2312.04820, 2023.

[253] Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint* arXiv:2106.09685, 2021.

[254] Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; Zhu, J. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. *arXiv preprint* arXiv:2305.16213, 2023.

[255] Yu, X.; Guo, Y. C.; Li, Y.; Liang, D.; Zhang, S. H.; Qi, X. Text-to-3D with classifier score distillation. *arXiv preprint* arXiv:2310.19415, 2023.

[256] Zhang, B.; Cheng, Y.; Yang, J.; Wang, C.; Zhao, F.; Tang, Y.; Chen, D.; Guo, B. GaussianCube: Structuring Gaussian splatting using optimal transport for 3D generative modeling. *arXiv preprint* arXiv:2403.19655, 2024.

[257] He, X.; Chen, J.; Peng, S.; Huang, D.; Li, Y.; Huang, X.; Yuan, C.; Ouyang, W.; He, T. GVGEN: Text-to-3D generation with volumetric representation. *arXiv preprint* arXiv:2403.12957, 2024.

[258] Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M. J. SMPL: A skinned multi-person linear model. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. ACM, 851–866, 2023.

[259] Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8110–8119, 2020.

[260] Yuan, Y.; Li, X.; Huang, Y.; De Mello, S; Nagano, K.; Kautz, J.; Iqbal, U. GAvatar: Animatable 3D Gaussian avatars with implicit mesh learning. *arXiv preprint* arXiv:2312.11461, 2023.

[261] Lombardi, S.; Simon, T.; Schwartz, G.; Zollhoefer, M.; Sheikh, Y.; Saragih, J. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics* Vol. 40, No. 4, Article No. 59, 2021.

[262] Pavlakos, G.; Choutas, V.; Ghorbani, N.; Bolkart, T.; Osman, A. A.; Tzionas, D.; Black, M. J. Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10975–10985, 2019.

[263] Liu, X.; Zhan, X.; Tang, J.; Shan, Y.; Zeng, G.; Lin, D.; Liu, X.; Liu, Z. HumanGaussian: Text-driven 3D human generation with gaussian splatting. *arXiv preprint* arXiv:2311.17061, 2023.

[264] Vilesov, A.; Chari, P.; Kadambi, A. CG3D: Compositional generation for text-to-3D via Gaussian splatting. *arXiv preprint* arXiv:2311.17907, 2023.

[265] Chung, J.; Lee, S.; Nam, H.; Lee, J.; Lee, K. M. LucidDreamer: Domain-free generation of 3D Gaussian splatting scenes. *arXiv preprint* arXiv:2311.13384, 2023.

[266] Ouyang, H.; Heal, K.; Lombardi, S.; Sun, T. Text2Immersion: Generative immersive scene with 3D Gaussians. *arXiv preprint* arXiv:2312.09242, 2023.

[267] Zhou, X.; Ran, X.; Xiong, Y.; He, J.; Lin, Z.; Wang, Y.; Sun, D.; Yang, M. H. GALA3D: Towards text-to-3D complex scene generation via layout-guided generative Gaussian splatting. *arXiv preprint* arXiv:2402.07207, 2023.

[268] Li, H.; Shi, H.; Zhang, W.; Wu, W.; Liao, Y.; Wang, L.; Lee, L.; Zhou, P. DreamScene: 3D Gaussian-based text-to-3D scene generation via formation pattern sampling. *arXiv preprint* arXiv:2404.03575, 2024.

[269] Shriram, J.; Trevithick, A.; Liu, L.; Ramamoorthi, R. Realm-dreamer: Text-driven 3D scene generation with inpainting and depth diffusion. *arXiv preprint* arXiv:2404.07199, 2024.

[270] Zhou, S.; Fan, Z.; Xu, D.; Chang, H.; Chari, P.; Bharadwaj, T.; You, S.; Wang, Z.; Kadambi, A. DreamScene360: Unconstrained textto-3D scene generation with panoramic Gaussian splatting. *arXiv preprint* arXiv:2404.06903, 2024.

[271] 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, 9264–9275, 2023.

[272] Zhang, J.; Tang, Z.; Pang, Y.; Cheng, X.; Jin, P.; Wei, Y.; Ning, M.; Yuan, L. Repaint123: Fast and high-quality one image to 3D generation with progressive controllable 2D repainting. *arXiv preprint* arXiv:2312.13271, 2023.

[273] Cao, M.; Wang, X.; Qi, Z.; Shan, Y.; Qie, X.; Zheng, Y. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint* arXiv:2304.08465, 2023.

[274] Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsanit, K.; Kembhavi, A.; Farhadi, A. Objaverse: A universe of annotated 3D objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13142–13153, 2023.

[275] Deitke, M.; Liu, R.; Wallingford, M.; Ngo, H.; Michel, O.; Kusupati, A.; Fan, A.; Laforte, C.; Voleti, V.; Gadre, S. Y.; et al. Objaverse-XL: A universe of 10M+ 3D objects. *arXiv preprint* arXiv:2307.05663, 2023.

[276] Tang, J.; Chen, Z.; Chen, X.; Wang, T.; Zeng, G.; Liu, Z. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. *arXiv preprint* arXiv:2402.05054, 2024.

[277] Xu, D.; Yuan, Y.; Mardani, M.; Liu, S.; Song, J.; Wang, Z.; Vahdat, A. AGG: Amortized generative 3D Gaussians for single image to 3D. *arXiv preprint* arXiv:2401.04099, 2024.

[278] Jiang, L.; Wang, L. BrightDreamer: Generic 3D Gaussian generative framework for fast text-to-3D synthesis. *arXiv preprint* arXiv:2403.11273, 2024.

[279] Xu, Y.; Shi, Z.; Yifan, W.; Chen, H.; Yang, C.; Peng, S.; Shen, Y.; Wetzstein, G. GRM: Large Gaussian reconstruction model for efficient 3D reconstruction and generation. *arXiv preprint* arXiv:2403.14621, 2024.

[280] Melas-Kyriazi, L.; Laina, I.; Rupprecht, C.; Neverova, N.; Vedaldi, A.; Gafni, O.; Kokkinos, F. IM-3D: Iterative multiview diffusion and reconstruction for high-quality 3D generation. *arXiv preprint* arXiv:2402.08682, 2024.

[281] Dai, X.; Hou, J.; Ma, C. Y.; Tsai, S.; Wang, J.; Wang, R.; Zhang, P.; Vandenhende, S.; Wang, X.; Dubey, A.; et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint* arXiv:2309.15807, 2023.

[282] Shen, Q.; Yi, X.; Wu, Z.; Zhou, P.; Zhang, H.; Yan, S.; Wang, X. Gamba: Marry Gaussian splatting with Mamba for single view 3D reconstruction. *arXiv preprint* arXiv:2403.18795, 2024.

[283] Gu, A.; Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint* arXiv:2312.00752, 2023.

[284] Li, Z.; Chen, Y.; Zhao, L.; Liu, P. Controllable text-to-3D generation via surface-aligned Gaussian splatting. *arXiv preprint* arXiv:2403.09981, 2024.

[285] Di, D.; Yang, J.; Luo, C.; Xue, Z.; Chen, W.; Yang, X.; Gao, Y. Hyper-3DG: Text-to-3D Gaussian generation via hypergraph. *arXiv preprint* arXiv:2403.09236, 2024.

[286] Lin, Y.; Clark, R.; Torr, P. DreamPolisher: Towards high-quality text-to-3D generation via geometric diffusion. *arXiv preprint* arXiv:2403.17237, 2024.

[287] Feng, Q.; Xing, Z.; Wu, Z.; Jiang, Y. G. FDGaussian: Fast Gaussian splatting from single image via geometric-aware diffusion model. *arXiv preprint* arXiv:2403.10242, 2024.

[288] Ling, H.; Kim, S. W.; Torralba, A.; Fidler, S.; Kreis, K. Align your Gaussians: Text-to-4D with dynamic 3D Gaussians and composed diffusion models. *arXiv preprint* arXiv:2312.10242, 2023.

[289] Blattmann, A.; Rombach, R.; Ling, H.; Dockhorn, T.; Kim, S. W.; Fidler, S.; Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 22563–22575, 2023.

[290] Yin, Y.; Xu, D.; Wang, Z.; Zhao, Y.; Wei, Y. 4DGen: Grounded 4D content generation with spatial-temporal consistency. *arXiv preprint* arXiv:2312.17225, 2023.

[291] Pan, Z.; Yang, Z.; Zhu, X.; Zhang, L. Fast dynamic 3D object generation from a single-view video. *arXiv preprint* arXiv:2401.08742, 2024.

[292] Liu, Y.; Lin, C.; Zeng, Z.; Long, X.; Liu, L.; Komura, T.; Wang, W. SyncDreamer: Generating multiview-

consistent images from a single-view image. *arXiv preprint* arXiv:2309.03453, 2023.

[293] Wu, Z.; Yu, C.; Jiang, Y.; Cao, C.; Wang, F.; Bai, X. SC4D: Sparse-controlled video-to-4D generation and motion transfer. *arXiv preprint* arXiv:2404.03736, 2024.

[294] Zeng, Y.; Jiang, Y.; Zhu, S.; Lu, Y.; Lin, Y.; Zhu, H.; Hu, W.; Cao, X.; Yao, Y. STAG4D: Spatial-temporal anchored generative 4D Gaussians. *arXiv preprint* arXiv:2403.14939, 2024.

[295] Xu, D.; Liang, H.; Bhatt, N. P.; Hu, H.; Liang, H.; Plataniotis, K. N.; Wang, Z. Comp4D: LLM-guided compositional 4D scene generation. *arXiv preprint* arXiv:2312.13763, 2023.

[296] Gao, L.; Wu, T.; Yuan, Y. J.; Lin, M. X.; Lai, Y. K.; Zhang, H. TM-NET: Deep generative networks for textured meshes. *arXiv preprint* arXiv:2010.06217, 2020.

[297] Gao, L.; Yang, J.; Wu, T.; Yuan, Y.; Fu, H.; Lai, Y.; Zhang, H. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 243, 2019.

[298] Nash, C.; Ganin, Y.; Ali Eslami, S. M.; Battaglia, P. W. PolyGen: An autoregressive generative model of 3D meshes. *arXiv preprint* arXiv:2002.10880, 2020.

[299] Siddiqui, Y.; Alliegro, A.; Artemov, A.; Tommasi, T.; Sirigatti, D.; Rosov, V.; Dai, A.; Nießner M. MeshGPT: Generating triangle meshes with decoder-only transformers. *arXiv preprint* arXiv:2311.15475, 2023.

[300] Ye, C.; Nie, Y.; Chang, J.; Chen, Y.; Zhi, Y.; Han, X. GauStudio: A modular framework for 3D Gaussian splatting and beyond. *arXiv preprint* arXiv:2403.19632, 2024.

[301] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; De-Vito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems, 2017.

[302] Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 265–283, 2016.

[303] Hu, S. M.; Liang, D.; Yang, G. Y.; Yang, G. W.; Zhou, W. Y. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* Vol. 63, No. 12, Article No. 222103, 2020.

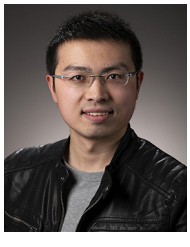
Tong Wu received his bachelor degree in computer science from Huazhong University of Science and Technology in 2019. He is currently a Ph.D. candidate in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics and computer vision.

Yu-Jie Yuan received his bachelor degree in mathematics from Xi'an Jiaotong University in 2018. He is currently a Ph.D. candidate in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics and neural rendering.

Ling-Xiao Zhang received his master of engineering degree in computer technology from the Chinese Academy of Sciences in 2020. He is currently an engineer at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics and geometric processing.

Jie Yang received his bachelor degree in mathematics from Sichuan University and his Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, where he is currently an assistant professor. His research interests include computer graphics and geometric processing.

Yan-Pei Cao received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2013 and 2018, respectively. He is currently the head of research and founding team at VAST. His research interests include computer graphics and 3D computer vision.



Ling-Qi Yan is an assistant professor of computer science at UC Santa Barbara, co-director of the MIRAGE Lab, and affiliated faculty in the Four Eyes Lab. Before joining UCSB, he received his Ph.D. degree from the Department of Electrical Engineering and Computer Sciences at UC Berkeley.

Lin Gao received his bachelor degree in mathematics from Sichuan University and his Ph.D. degree in computer science from Tsinghua University. He is currently a professor in the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded a Newton Advanced Fellowship from the Royal Society and an Asia Graphics Association young researcher award. His research interests include computer graphics and geometric processing.