Title: Gaussian splatting

URL: https://en.wikipedia.org/wiki/Gaussian_splatting

PageID: 74947242

Categories: Category:3D animation software, Category:3D computer graphics, Category:3D rendering, Category:Carl Friedrich Gauss, Category:Machine learning algorithms



---

Gaussian splatting is a volume rendering technique that deals with the direct rendering of volume data without converting the data into surface or line primitives . [ 1 ] The technique was originally introduced as splatting by Lee Westover in the early 1990s. [ 2 ] [ 3 ]

This technique was revitalized and exploded in popularity in 2023, when a research group from Inria proposed the seminal 3D Gaussian splatting that offers real-time radiance field rendering. Like other radiance field methods, it can convert multiple images into a representation of 3D space, then use the representation to create images as seen from new angles. [ 4 ] Multiple works soon followed, such as 3D temporal Gaussian splatting that offers real-time dynamic scene rendering. [ 5 ]

3D Gaussian splatting

3D Gaussian splatting ( 3DGS ) is a technique used in the field of real-time radiance field rendering. [ definition needed ] [ 4 ] It enables the creation of high-quality real-time novel-view scenes by combining multiple photos or videos, addressing a significant challenge in the field.

The method represents scenes with 3D Gaussians that retain properties of continuous volumetric radiance fields, integrating sparse points produced during camera calibration. It introduces an Anisotropic representation using 3D Gaussians to model radiance fields, along with an interleaved optimization and density control of the Gaussians. A fast visibility-aware rendering algorithm supporting anisotropic splatting is also proposed, catering to GPU usage. [ 4 ]

Method

Training

The method involves several key steps:

Input: A set of images of a static scene along with camera positions, expressed as a sparse point cloud .

3D Gaussians: Definition of mean, covariance matrix, and opacity for each Gaussian.

Color representation: Using spherical harmonics to model view-dependent appearance.

Optimization algorithm: Optimizing the parameters using stochastic gradient descent to minimize a loss function combining L1 loss and D-SSIM, inspired by the Plenoxels work. [ 6 ]

Rasterizer: Implementing a tile-based rasterizer for fast sorting and backward pass, enabling efficient blending of Gaussian components.

The method uses differentiable 3D Gaussian splatting, which is unstructured and explicit, allowing rapid rendering and projection to 2D splats. The covariance of the Gaussians can be thought of as configurations of an ellipsoid, which can be mathematically decomposed into a scaling matrix and a rotation matrix. The gradients for all parameters are derived explicitly to overcome any overhead due to autodiff . [ citation needed ]

Each step of rendering is followed by a comparison to the training views available in the dataset. The optimization uses the difference to create a dense set of 3D Gaussians that represent the scene as accurately as possible. [ citation needed ]

Using

An optimized set of 3D Gaussians is saved onto the computer. Like in the training step, a renderer creates a view from these Gaussians. [ citation needed ]

Several sets of Gaussians can be composed together into larger scenes. [ citation needed ]

Results and evaluation

The authors [ who? ] tested their algorithm on 13 real scenes from previously published datasets and the synthetic Blender dataset. [ 7 ] They compared their method against state-of-the-art techniques like Mip-NeRF360, [ 8 ] InstantNGP, [ 9 ] and Plenoxels. [ 6 ] Quantitative evaluation metrics used were PSNR, L-PIPS, and SSIM.

Their fully converged model (30,000 iterations) achieves quality on par with or slightly better than Mip-NeRF360, [ 8 ] but with significantly reduced training time (35–45 minutes vs. 48 hours) and faster rendering (real-time vs. 10 seconds per frame). At 7,000 iterations (5–10 minutes of training), their method achieves comparable quality to InstantNGP [ 9 ] and Plenoxels. [ 6 ]

For synthetic bounded scenes (Blender dataset [ 7 ] ), they achieved state-of-the-art results even with random initialization, starting from 100,000 uniformly random Gaussians.

Limitations

Some limitations of the method include: [ citation needed ]

Elongated artifacts or "splotchy" Gaussians in some areas.

Occasional popping artifacts due to large Gaussians created by the optimization, especially in regions with view-dependent appearance.

Higher memory consumption compared to NeRF-based solutions, though still more compact than previous point-based approaches.

May require hyperparameter tuning (e.g., reducing position learning rate) for very large scenes.

Peak GPU memory consumption during training can be high (over 20 GB) in the current unoptimized prototype.

The authors [ who? ] note that some of these limitations could potentially be addressed through future improvements like better culling approaches, antialiasing, regularization, and compression techniques.

3D Temporal Gaussian splatting

Extending 3D Gaussian splatting to dynamic scenes, 3D Temporal Gaussian splatting incorporates a time component, allowing for real-time rendering of dynamic scenes with high resolutions. [ 5 ] It represents and renders dynamic scenes by modeling complex motions while maintaining efficiency. The method uses a HexPlane to connect adjacent Gaussians, providing an accurate representation of position and shape deformations. By utilizing only a single set of canonical 3D Gaussians and predictive analytics, it models how they move over different timestamps. [ 10 ]

It is sometimes referred to as "4D Gaussian splatting"; however, this naming convention implies the use of 4D Gaussian primitives (parameterized by a 4×4 mean and a 4×4 covariance matrix). Most work in this area still employs 3D Gaussian primitives, applying temporal constraints as an extra parameter of optimization. [ citation needed ]

Achievements of this technique include real-time rendering on dynamic scenes with high resolutions, while maintaining quality. It showcases potential applications for future developments in film and other media, although there are current limitations regarding the length of motion captured. [ 10 ]

Applications

3D Gaussian splatting has been adapted and extended across various computer vision and graphics applications, from dynamic scene rendering to autonomous driving simulations and 4D content creation:

Text-to-3D using Gaussian Splatting: Applies 3D Gaussian splatting to text-to-3D generation. [ 11 ]

End-to-end Autonomous Driving: Mentions 3D Gaussian splatting as a data-driven sensor simulation method for autonomous driving, highlighting its ability to generate realistic novel views of a scene. [ 12 ]

SuGaR: Proposes a method to extract precise and fast meshes from 3D Gaussian splatting. [ 13 ]

SplaTAM: Applies 3D Gaussian-based radiance fields to Simultaneous Localization and Mapping (SLAM), leveraging fast rendering and optimization capabilities to achieve state-of-the-art results. [ 14 ]

Align Your Gaussians: Uses dynamic 3D Gaussians for 4D content creation from text. [ 15 ]

See also

Ambisonics

Computer graphics

Neural radiance field

Volume rendering

References

v

t

e

Datasets

Digital geometry

Commercial systems

Feature detection

Geometry

Image sensor technology

Learning

Morphology

Motion analysis

Noise reduction techniques

Recognition and categorization

Research infrastructure

Researchers

Segmentation

Software

Computer stereo vision

Motion capture

Object recognition 3D object recognition

3D object recognition

3D reconstruction from multiple images

2D to 3D conversion

Gaussian splatting

Neural radiance field

Shape from focus

Simultaneous localization and mapping

Structure from motion

View synthesis

Visual hull

4D reconstruction Free viewpoint television Volumetric capture

Free viewpoint television

Volumetric capture

3D pose estimation

Activity recognition

Audio-visual speech recognition

Automatic image annotation

Automatic number-plate recognition

Automated species identification

Augmented reality

Bioimage informatics

Blob detection

Computer-aided diagnosis

Content-based image retrieval Reverse image search

Reverse image search

Eye tracking

Face recognition

Foreground detection

Gesture recognition

Image denoising

Image restoration

Landmark detection

Medical image computing

Object detection Moving object detection Small object detection

Moving object detection

Small object detection

Optical character recognition

Pose tracking

Remote sensing

Robotic mapping

Autonomous vehicles

Video content analysis

Video motion analysis

Video surveillance

Video tracking