Gaussian Splatt Research

From photos to a point cloud

- Start with photos
- Go to structure-from-motion
- COLMAP in our case

COLMAP guesses where the cameras were and constructs a sparse point cloud, which already kind of represents the scene.

What is Gaussian splatting?

First: what is a Gaussian?

We can visualise a Gaussian in 3D as a **blob**.
At first, I thought they were just points, but:

- In **1D**, a Gaussian is a graph with a bump (a bell curve).
- In **3D**, that bump translates to a blob that we can stretch, rotate, scale, etc.

Each Gaussian blob has:

- a position
- a size / shape
- an orientation
- a color / appearance model

Gaussian splatting represents the scene as **a lot of these blobs** instead of polygons or voxels.

Why is this so good / better?

- **Works with reflections**
- **Much faster than NeRF**
  - NeRF = Neural Radiance Fields – a neural network model that reconstructs a 3D scene from a group of 2D pics
- Gaussian splatting is still a radiance field, but **without the heavy neural rendering** that makes NeRF slower.

Other important points:

- Gaussian splatting uses **spherical harmonics**
  - Surfaces change color when viewed from different angles (glints, veneer, etc.) without needing a full neural network.
- Gaussian splatting gives you editing support (you can manipulate splats/points).
- All you need is to get the **sparse point cloud from photogrammetry**.

Through **iterative optimization**, these Gaussians are adjusted to match the original images (using gradient descent):

- You have more Gaussians where you need detail
  - e.g. trees with foliage
- And **fewer Gaussians where detail is low**
  - e.g. sky

Basic workflow

1. Capture pics from different angles of the object / room / scene
2. Reconstruct cameras using a point cloud (COLMAP in my case)
3. Create a 3D Gaussian "splat" centered at each point in the point cloud
4. Run an optimization process:
   - position
   - size
   - orientation
   - color / appearance

Because you cannot have an even distribution of Gaussians that perfectly matches the complexity of reality, here comes:

Adaptive density control

- More Gaussians where you need to model detail
  - again: a tree with lots of leaves
- Fewer Gaussians for simple or flat areas
  - like a pathway or the sky

The difference between Photogrammetry, NeRF and Gaussian Splatting

1. Photogrammetry: how it works

- Uses overlapping photos
- Reconstructs a 3D mesh with textures
- The light and detail are baked into the texture
- Produces a polygon mesh

Good for:

- real-world measurements
- game-ready assets
- accurate and editable results (if you have very good photos)

2. Neural Radiance Fields (NeRFs)

- Uses AI to learn a volumetric radiance field of the scene
- Needs photos (not necessarily strongly overlapping)
- Captures full scene context, including sky and horizon
- Can generate smooth flythrough videos from viewpoints that were never actually photographed

Limitations:

- Doesn’t produce a mesh (it’s a **volumetric field**)
- Not accurate for measuring real sizes
- Heavier and slower to train / render compared to Gaussian Splatting

3. Gaussian Splatting

Third and finally:

- Transforms the trained radiance field into a **point cloud of splats** (elliptical blobs)
- **Way faster than NeRF** (and we’re not even talking about photogrammetry)
- Works in **Unreal / Unity / Blender**
- Still **volumetric-like**, not a true mesh
- Still not accurate for measurement
- Harder to share than classic photogrammetry models (which are just meshes + textures)

Summary

Photogrammetry result:

- Mesh + texture
- Best for:
  - real-world measurements
  - game-ready assets
  - accurate, editable, relatively light (if you have *very* good photos)

NeRF result:

- Volumetric radiance field
- Best for:
  - virtual production
  - camera flythroughs
  - novel views
  - full-scene reconstructions
- Not measurable, pretty heavy

Gaussian Splatting result:

- Real-time splat cloud
- Best for:
  - real-time rendering in engines
  - fast, sharp, Unity/Unreal-friendly
- Not a mesh, not measurable

Conclusion

Photogrammetry is **excellent for accuracy** and measurement.
NeRFs and Gaussian Splatting provide **highly immersive and flexible visual results**, especially for view synthesis and real-time rendering, with Gaussian splatting bringing much higher speed than classic NeRF.

Reference

- https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/3d_gaussian_splatting_high.pdf