



Minecraft Novel View Synthesis

BRIAN XU

Personal Motivation

- 3D reconstruction for downstream applications
- NeRFs/Splats are not generally usable downstream
- Optimization model \neq downstream model
- Optimization model = downstream model (?)



Minecraft

- World is composed of voxels
 - Highly discrete and simplified setting
- Optimize position + appearance of voxels from multi-view images
- Optimization model = downstream model



Approach

- Voxels
 - Voxel carving / differentiable voxel rendering
 - Not a super active field of research
- Gaussian Splats
 - Discrete
 - Active field of research
- Final approach combines different lines of works in Gaussian Splatting

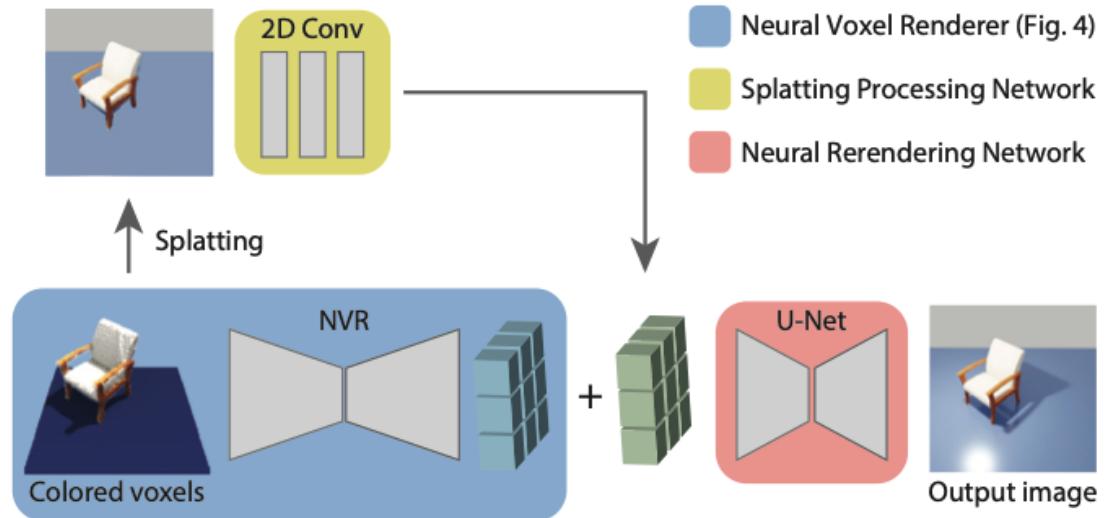
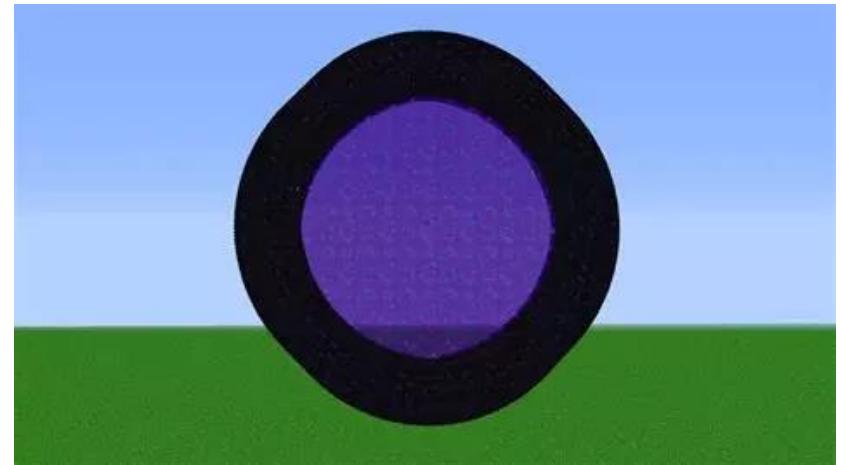


Figure 5. NVR+ network architecture.

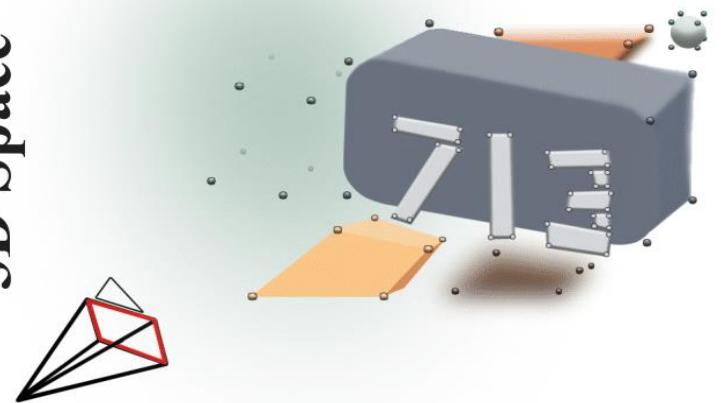
Different Gaussian Shapes

- There are no circles in Minecraft!
- Not everything in the real world is oblate (and volumetric) either!
- Papers exploring alterations of the gaussian model:
 - 2D Gaussian Splatting, Deformable Radial Kernel Splatting, Generalized Exponential Splatting, Triangle Splatting
- We adopt Convex Splatting

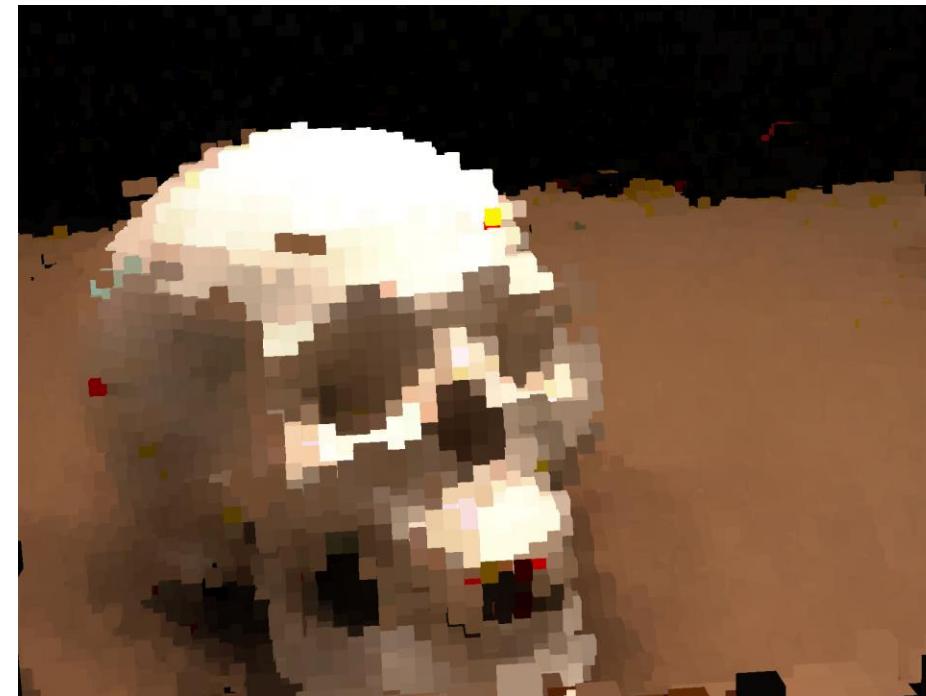


3D Smooth Convexes

3D Space



Results (shape)



Different Gaussian Appearances

- Minecraft blocks are textured!
- Things in the real world are textured too!
- Texture individual gaussians / entire scene
 - Individual Gaussians: GSTex, Textured Gaussians, TextureSplat, Textured-GS, BBSplat
 - Entire Scene: Texture-GS, HDGS, NeST Splatting
- We texture individual primitives

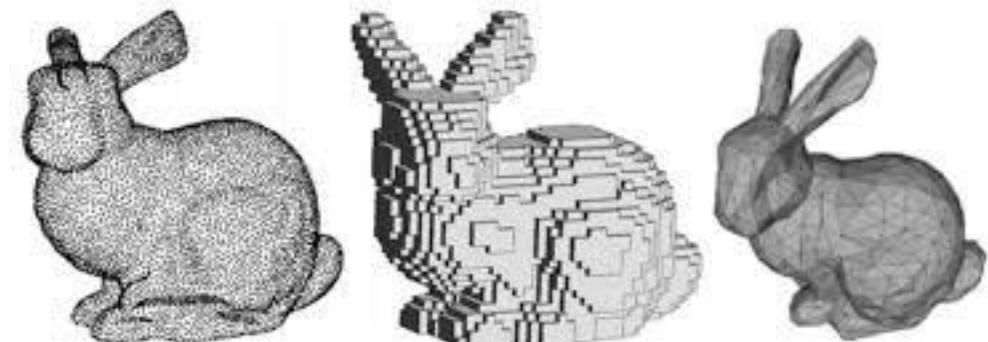
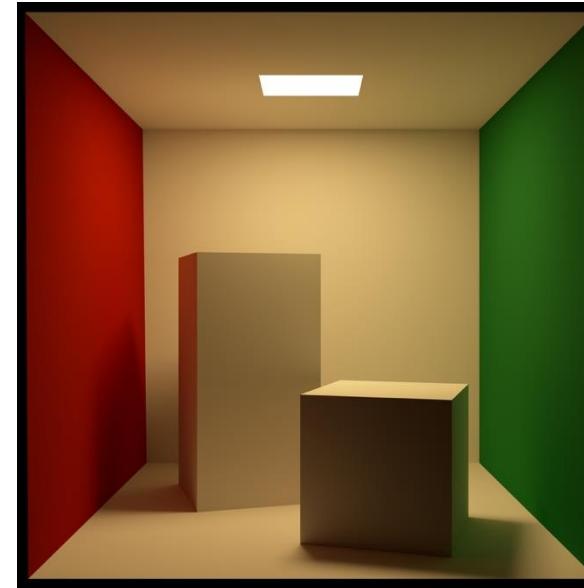


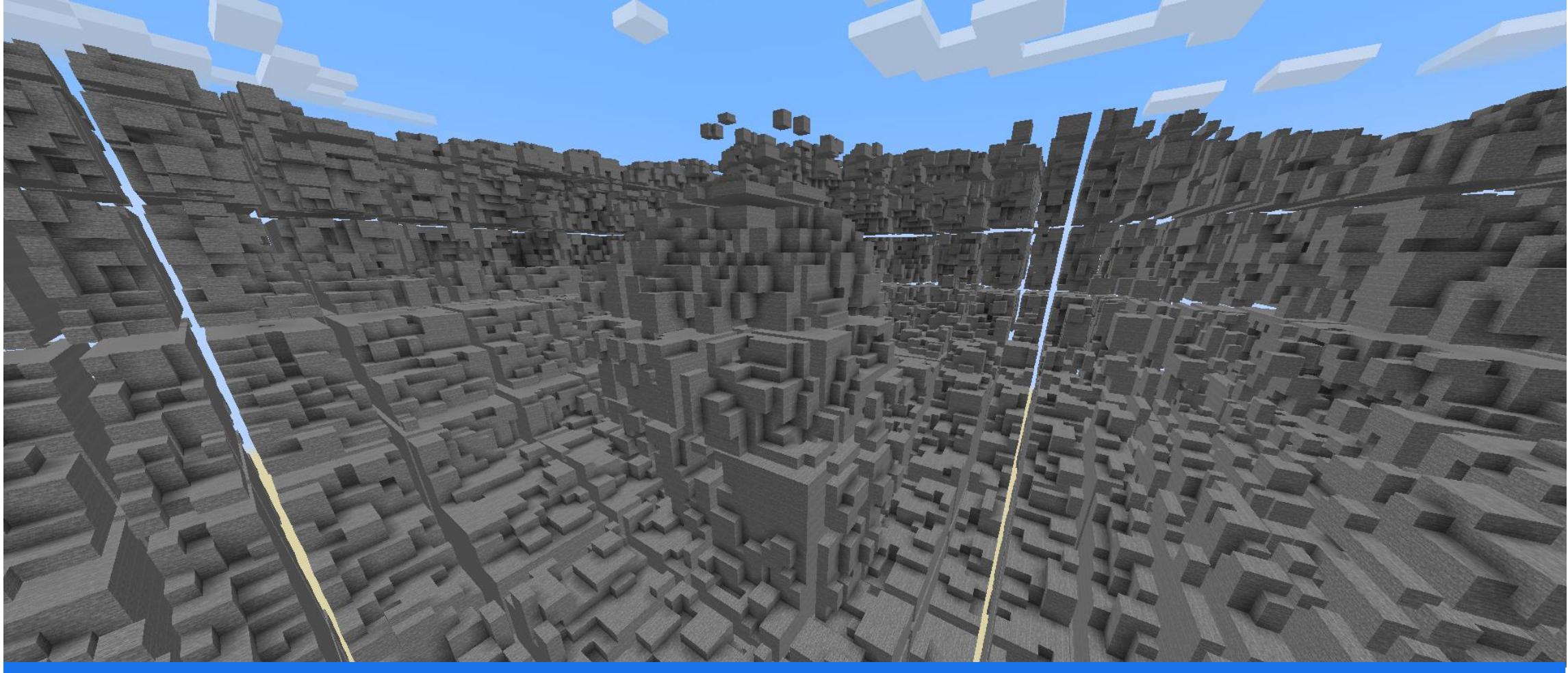
Results (texture)



Next steps

- Finish texturing
- Export to Minecraft
- Evaluate on famous graphics scenes
- Create and evaluate on a Minecraft NVS dataset
- Write a cheeky paper (?) and release to public
 - The first work to combine shape + texture





Takeaways

- Toy problem, but surprisingly complex
- If you work on something silly enough it is guaranteed to be novel