Minecraft to 3D: A Pipeline for High-Fidelity Reconstruction of Minecraft Worlds

Sean Hardesty Lewis
shl225@cornell.edu
Cornell Tech
New York, NY, USA

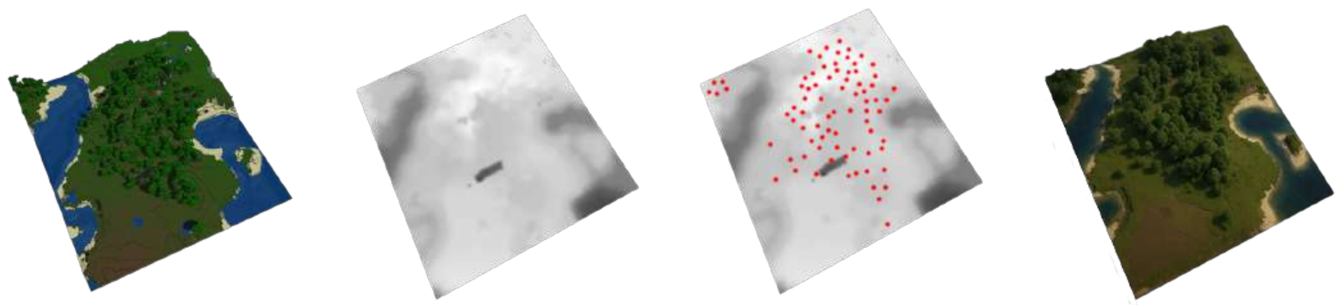


Figure 1: A Minecraft survival world (left) reconstructed as a high-quality 3D scene (right) with *Minecraft to 3D.* The stepped block surface is smoothed into a continuous height-map and identified objects are replaced with user-selectable 3D assets.

ABSTRACT

We introduce *Minecraft to 3D,* a novel pipeline that automatically converts any Minecraft world into a high-quality polygonal scene. A 3D convolutional network recognises Minecraft's default objects, the block surface is resampled into a smooth height-map, and each recognised object is substituted with a high-quality 3D model chosen from an external library. Object locations, orientations, and tags are preserved, a separate water plane is exported for engine-level ocean rendering, and the final scene opens natively in modern 3D engines. The pipeline processes a one-square-kilometre world in under three minutes on a single consumer GPU, enabling educators, indie developers, and artists to move rapidly from voxel sketches to fully lit environments.

ACM Reference Format:
Sean Hardesty Lewis. 2025. Minecraft to 3D: A Pipeline for High-Fidelity Reconstruction of Minecraft Worlds. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH Posters '25), August 10–14, 2025*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641236.3664XYZ

1 INTRODUCTION

Minecraft's block-based sandbox empowers millions of players to create sprawling worlds, but its coarse cubes leave those sketches locked inside the game—no direct path previously existed to turn a Minecraft world into a polished, engine-ready 3D environment.

Minecraft to 3D[1] closes that gap by converting any Minecraft world into a fully textured 3D scene with physically-based materials while faithfully preserving layout and semantics. Related work on layout-guided scene synthesis [Yang et al. 2024], large-scale sparse-voxel generation [Ren et al. 2024], and procedural building creation inside Minecraft itself [Green et al. 2019] shows the community's appetite for methods that translate coarse input into detailed 3D content, yet none provides a turnkey path from arbitrary Minecraft worlds to high-quality polygonal scenes. Starting from a Sponge Schematic export that retains block IDs and NBT tags, our system recognises default Minecraft structures, rebuilds the ground as a continuous surface, and swaps each structure for a higher-fidelity counterpart while keeping the original layout intact. The resulting scene imports directly into Blender, Unreal, Unity, and Godot without manual clean-up.

2 PIPELINE OVERVIEW

Voxel data are streamed in 256 × 256-block tiles with a 20-block overlap so that structures crossing region boundaries retain context. A 3D U-Net trained on Minecraft's canonical assets—oak trees, villager houses, desert temples, and related variants—assigns a semantic label to every object. The network achieves 97.8 % mean intersection-over-union (mIoU) on isolated structures and 88.6 % where multiple structures intersect.

Each label is written to a lightweight object list that stores block-accurate position, three-axis orientation, and any auxiliary tags (for example, house orientation). After segmentation, non-terrain voxels are hidden, the stepped surface is up-sampled with trilinear interpolation, and an anisotropic Gaussian filter removes staircase artefacts. Tiles are cropped to their original bounds, then merged with a Poisson blend; the blend suppresses seams without disturbing coarse relief.

[1] https://mc3d.org

SIGGRAPH Posters '25, August 10–14, 2025, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0518-2/25/08
https://doi.org/10.1145/3641236.3664XYZ

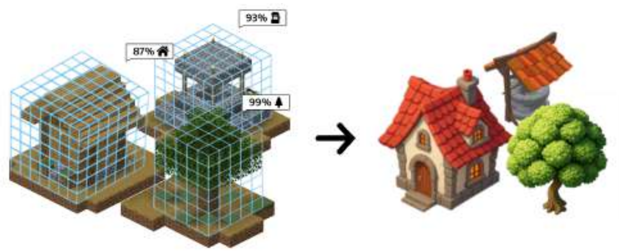


Figure 2: Overview of 3D CNN labeling and substitution.

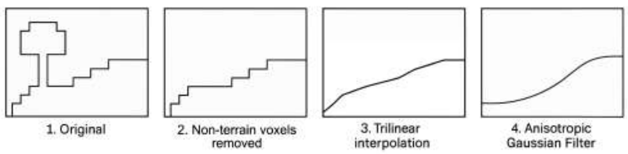


Figure 3: Workflow for terrain upscaling.

Recognised objects are then substituted with the object list driving a query against an external model library, returning high-quality geometry that matches class, scale, and broad proportions. Before placement, every model is rigidly aligned and normalised to the local ground patch so that it sits flush with the terrain. The water layer is exported separately as a flat mesh at the recorded sea level, and engines can replace it with their own ocean shaders or leave it intact for stylised rendering.

3 PERFORMANCE

Processing a 1 km^2 map—about 65 million blocks—takes 147 s on an RTX 4090 and never exceeds 3.2 GB of system memory thanks to a sparse-voxel octree. CNN inference accounts for 84 % of that time; the remainder is height-map synthesis, Poisson blending, and model placement. Exported scenes open in Blender 4.1, Unreal 5.4, Unity 2023 LTS, and Godot 4.3 without missing geometry, inverted normals, or material errors.

4 LIMITATIONS

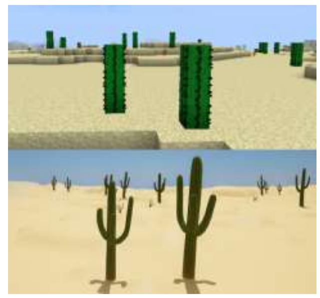
Heavy smoothing erases the blocky staircase that defines Minecraft but also removes fine detail—individual steps and decorative block patterns flatten into subtle slopes. Lowering the filter strength restores detail at the cost of bumpier terrain, which can feel awkward for first-person navigation. The pipeline handles only the default block set; custom or modded blocks fall back to “terrain” and receive no model substitution. Intersecting structures confuse the network, occasionally merging two labels and picking the wrong replacement.

5 APPLICATIONS

Indie studios can prototype a level in Minecraft, run it through our pipeline, and continue in any game engine with their own art assets. A 3D artist who wants a quick starting scene can build the rough layout in Minecraft, press export, and spend the saved

(a) Figure 1

(b) Figure 2



(c) Figure 3

(d) Figure 4

Figure 4: Comparison of results across four scenarios.

hours on lighting and shading instead of manual modelling. Educators can generate cinematic fly-throughs of student projects, and virtual-production teams can move collaboratively designed block sets onto LED stages with minimal hand-off.

6 FUTURE WORK

We are training on community-created structures to extend beyond the default asset set, and we are testing on-demand AI-generated geometry so that a substituted model can match style prompts or concept art. Additional work targets an adaptive smoothing filter that keeps critical block detail while eliminating visible staircases.

ACKNOWLEDGMENTS

Minecraft™ and related assets © Mojang Studios (2025). Minecraft™ screenshots, textures, or other Mojang assets, or derivatives of Mojang assets displayed used here for educational research under Mojang AB © permission per the published Minecraft™ Brand and Asset Usage Guidelines v2.1. This work is not affiliated with or endorsed by Mojang AB.

The author thanks the open-source asset community for their support. All 3D models shown are CC0.

REFERENCES

Michael Cerny Green, Christoph Salge, and Julian Togelius. 2019. Organic Building Generation in Minecraft. arXiv:1906.05094 [cs.HC] https://arxiv.org/abs/1906.05094

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. 2024. XCube: Large-Scale 3D Generative Modeling using Sparse Voxel Hierarchies. arXiv:2312.03806 [cs.CV] https://arxiv.org/abs/2312.03806

Xiuyu Yang, Yunze Man, Jun-Kun Chen, and Yu-Xiong Wang. 2024. SceneCraft: Layout-Guided 3D Scene Generation. arXiv:2410.09049 [cs.CV] https://arxiv.org/abs/2410.09049