BrepGaussian: CAD reconstruction from Multi-View Images with Gaussian Splatting

Supplementary Material

Jiaxing Yu[1], Dongyang Ren[1], Hangyu Xu[1], Zhouyuxiao Yang[1],
Yuanqi Li[1†], Jie Guo[1], Zhengkang Zhou[2], Yanwen Guo[1]
[1]State Key Laboratory of Novel Software Technology, Nanjing University
[2]Nanjing Urban Construction Tunnel& Bridge Intelligent Management Co., Ltd.

1. Further Discussion on the Full Pipeline

Another representative method for image to point cloud reconstruction is VGGT: Visual Geometry Grounded Transformer [11], a large feed-forward transformer trained on massive multi-view datasets that directly predicts camera poses, dense depth, and point clouds from sparse or even uncalibrated views. VGGT gives very fast and robust 3D initialization compared to traditional SfM [10]+3DGS pipelines. However, its predicted geometry is generally less accurate and less metrically stable than that of optimized 3DGS. This limitation becomes more pronounced for CAD models. CAD models often exhibit low-texture surfaces, uniform materials, and repetitive geometric patterns. As a result, the reconstructed point clouds from different views are often misaligned or fragmented, producing multiple disconnected local point clouds rather than a single coherent structure. In contrast, our BrepGaussian builds upon Gaussian Splatting, thereby enabling robust recovery of geometry from multi-view images, even for low-texture CAD objects.

Recent studies have also explored end-to-end parametric differentiable renderers [3, 12, 13] that reconstruct geometry directly from multi-view images. These approaches operate on explicit primitives such as meshes, CSG trees, or Bézier surfaces, and allow gradients from losses defined in the image space to propagate back to shape parameters. However, they usually assume a fixed topology and rely on a set of shape parameters with strong initialization priors, which restricts them to local refinement, rather than complete CAD reconstruction from images. In contrast, BrepGaussian leverages Gaussian Splatting to obtain clean point clouds from multi-view images, enabling full B-Rep recovery with surface and edge structures.

2. Parametric Fitting

RANSAC Model. RANSAC (RANdom SAmple Consensus) [2] is a robust model estimation algorithm used to fit a parametric model to data that may contain a large proportion of outliers. It assumes that the input data consist of *inliers*, which can be well fitted by a valid set of model parameters, and *outliers*. Given a dataset $\mathcal{D}$ and a parameterized model $\mathcal{M}(\theta)$, the algorithm iteratively samples a minimal subset $\mathcal{S} \subset \mathcal{D}$ to estimate the model parameters θ, fits the model to $\mathcal{S}$, and evaluates all points in $\mathcal{D}$ using a distance measure ε. Points satisfying $\varepsilon_i < \tau$ are considered as inliers, where τ is a predefined distance threshold. The model with the largest inlier count is selected as the final estimate:

$$\theta^* = \arg\max_{\theta} \sum_i \mathbf{L}(\varepsilon_i < \tau), \tag{1}$$

where $\mathbf{L}(\cdot)$ denotes the indicator function. In our implementation, RANSAC is used to robustly estimate the parameters of plane, cylinder, and sphere models.

Plane. For planar primitive fitting, in each iteration, points are sampled to define a candidate plane:

$$ax + by + cz + d = 0, \tag{2}$$

which can be represented as $\Pi = (n, d)$, where $n = [a, b, c]$ is the normalized plane normal vector and d is the offset from the origin. For a point $\mathbf{p}_i = (x_i, y_i, z_i)$, the perpendicular distance to the plane is given by

$$\varepsilon_i = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}}. \tag{3}$$

Points with $\varepsilon_i < \tau$ (typically $\tau = 0.01$ units) are regarded as inliers. The algorithm iterates up to 1000 times or until 90% of the points are classified as inliers, and the plane with the maximum inlier count is selected as the optimal fit. After fitting, all inlier points are projected onto the plane to form a local 2D coordinate system, and their convex hull is computed to determine the boundary of the planar patch.

Cylinder. A cylinder is mathematically defined by its radius R, the top face center $\mathbf{c_t}$ and the bottom face center $\mathbf{c_b}$. The parameter set is:

$$\mathbf{\Theta}_{\text{cylinder}} = \{\mathbf{c}_t, \mathbf{c}_b, R\} \tag{4}$$

Estimating this primitive requires simultaneous recovery of both the axis orientation and radius. We generate candidate axis directions by combining Principal Component Analysis (PCA) and the Minimal Oriented Bounding Box (MOBB) [9]. PCA provides the direction of least variance

as an initial axis, while MOBB contributes three orthogonal axes from its edges. For each axis candidate, points are projected onto the plane perpendicular to the axis to obtain 2D coordinates (X_i, Y_i). In each iteration, points are randomly sampled to form a circle candidate whose parameters (a, b, R) are estimated by solving:

$$\begin{bmatrix} -X_1 & -Y_1 & -1 \\ -X_2 & -Y_2 & -1 \\ \vdots & \vdots & \vdots \\ -X_n & -Y_n & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} X_1^2 + Y_1^2 \\ X_2^2 + Y_2^2 \\ \vdots \\ X_n^2 + Y_n^2 \end{bmatrix}, \tag{5}$$

where the circle center (x_c, y_c) and radius R are computed as

$$x_c = -\frac{a}{2}, \quad y_c = -\frac{b}{2}, \quad R = \sqrt{x_c^2 + y_c^2 - c}. \tag{6}$$

Points whose radial distance to the circle satisfies $|\sqrt{(X_i - x_c)^2 + (Y_i - y_c)^2} - R| < \tau$ are considered inliers. The candidate with the largest inlier count is selected. Given the selected axis, the cylinder height h is estimated by computing the projection range of inlier points along the axis direction. The fitted circle and height are transformed along the estimated axis to obtain the top and bottom centers $(\mathbf{c}_t, \mathbf{c}_b)$ in world coordinates.

Sphere. In each iteration, points are randomly sampled to define a candidate sphere. The parameters of a sphere, i.e., its center $\mathbf{c} = (x_c, y_c, z_c)$ and radius r, satisfy the general sphere equation

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r^2. \tag{7}$$

Expanding the equation leads to a linear form that can be solved by least squares:

$$\begin{bmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2x_n & -2y_n & -2z_n & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ D \end{bmatrix} = \begin{bmatrix} -(x_1^2 + y_1^2 + z_1^2) \\ -(x_2^2 + y_2^2 + z_2^2) \\ \vdots \\ -(x_n^2 + y_n^2 + z_n^2) \end{bmatrix}, \tag{8}$$

where $D = r^2 - x_c^2 - y_c^2 - z_c^2$. The center and radius are obtained as

$$\mathbf{c} = (x_c, y_c, z_c), \quad r = \sqrt{x_c^2 + y_c^2 + z_c^2 - D}. \tag{9}$$

For each candidate sphere, we compute the residual for every point $\mathbf{p}_i$ as

$$\varepsilon_i = \big|\, \|\mathbf{p}_i - \mathbf{c}\| - r \,\big|, \tag{10}$$

and classify points with $\varepsilon_i < \tau$ (typically $\tau = 0.01$ units) as inliers.

Primitive Intersections. The intersection relations between fitted primitives are computed analytically.

For two planes $\Pi_1 = (n_1, d_1)$ and $\Pi_2 = (n_2, d_2)$, the intersection line has direction

$$d = n_1 \times n_2, \tag{11}$$

The anchor point x_0 on the line is obtained by solving a least-squares system that satisfies the two plane constraints and an additional constraint along d:

$$\begin{bmatrix} n_1^\top \\ n_2^\top \\ d^\top \end{bmatrix} x = \begin{bmatrix} n_1^\top p_1 \\ n_2^\top p_2 \\ 0 \end{bmatrix}. \tag{12}$$

The anchor point x_0 together with the direction d defines the intersection line $x(t) = x_0 + td$. Edge points obtained from Stage1 are projected onto this line to determine the valid parameter range $[t_{\min}, t_{\max}]$.

For a plane $\Pi = (n, d)$ intersecting with a cylinder $C = (a, c_{\text{axis}}, r)$, where a is the axis direction, c_{axis} is a point on the axis, and r is the radius, the resulting curve is a circle embedded in Π. The circle center is computed as

$$o = c_{\text{axis}} - \frac{n^\top (c_{\text{axis}} - p_0)}{n^\top a}\, a, \tag{13}$$

where p_0 is any point on Π. The circle radius remains r, and edge points projected onto Π are retained if they satisfy both plane and radial distance thresholds.

The intersection between a plane $\Pi = (n, d)$ and a sphere $S(c, r)$ is also a circle, whose center and radius are given by:

$$o = c - (n^\top c + d)\, n, \quad r' = \sqrt{r^2 - (n^\top c + d)^2}. \tag{14}$$

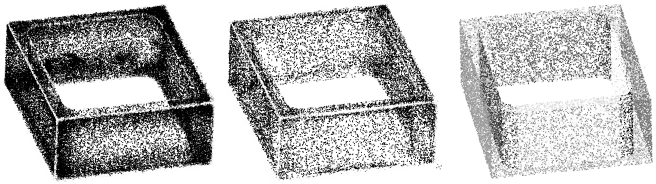
Here, o is the projection of the sphere center onto the plane, and r' is the radius of the intersection circle.

Triangulation. After fitting the primitives (planes, cylinders, and spheres) and computing their pairwise intersections (lines and circles), we generate closed surface patches to assemble the final B-Rep model. For each surface, all intersection lines and curves incident to it are collected as boundary elements and projected to its parametric domain. Planar surfaces $\Pi = (n, d)$ are projected onto a local (u, v) frame orthogonal to n, while cylindrical surfaces $C = (a, c_{\text{axis}}, r)$ are represented in cylindrical coordinates (θ, h), and spherical surfaces $S = (c, r)$ are parameterized by spherical coordinates (θ, ϕ). The intersection curves are then connected according to shared corner points to form closed boundary loops. Each loop is analyzed in the parameter domain to distinguish outer boundaries and inner holes based on their signed areas and containment relations.

Planar surfaces are triangulated using constrained Delaunay triangulation, while cylindrical and spherical surfaces are meshed in their respective parametric domains to produce watertight patches. Finally, all reconstructed patches are merged into a unified CAD mesh, and vertex normals are aligned with the surface orientations.

3. Effect of two stage learning

We provide qualitative comparisons to illustrate the effect of using our two-stage Gaussian training scheme. As shown in Fig.1, the quantitative results in the main paper already demonstrate that the two-stage strategy yields better reconstruction accuracy. In the one-stage setting, excessive loss coupling causes the Gaussian positions to be influenced by competing objectives, resulting in noisy and irregular point distributions that deviate from the true object surfaces. Such artifacts degrade the quality of the reconstructed CAD models. In contrast, our two-stage training decouples geometry learning from feature optimization, producing clean Gaussian distributions that align closely with the underlying surfaces, and thus enable higher-quality CAD reconstruction.

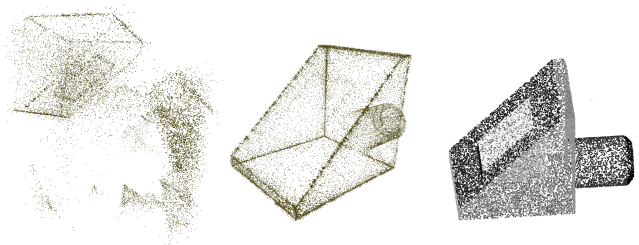


w/o two stage two stage Ground Truth

Figure 1. Effect of two-stage learning.

4. Visual Differences between 3DGS and 2DGS

Fig.2 compares 3DGS, 2DGS, and the ground-truth points. 3DGS models geometry with volumetric ellipsoids, which introduces artificial thickness and noisy distributions. In contrast, 2DGS represents geometry using oriented elliptical disks constrained to the surface, producing cleaner and sharper structures that better match CAD surfaces.

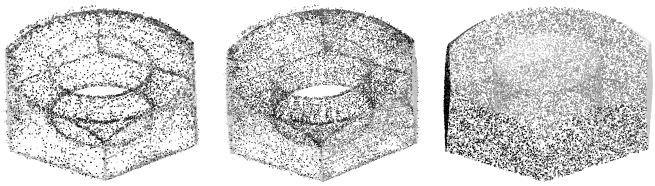


3DGS output 2DGS output Ground Truth

Figure 2. Visual differences between 3DGS and 2DGS.

5. Effect of Densification

Fig.3 compares the point clouds generated with and without the densification step during the Gaussian-to-point conversion. Without densification, the sampled points are sparse and unevenly distributed, failing to capture fine surface details and resulting in incomplete geometric structures. After applying densification, the point distribution becomes more uniform and better aligned with the ground-truth surfaces, providing a higher-quality input for subsequent segmentation and CAD reconstruction.



w/o densification densification Ground Truth

Figure 3. Effect of densification in Gaussian-to-point conversion.

6. Comparisons with Generative Methods

We compare with recent generative baselines, follow their input specs, and report the best result. These methods enable diverse part synthesis, but image conditioning remains under-constrained.

Table 1. Quantitative comparison with generative methods.

Metrics	Hola [8]	GenCAD [1]	CADDreamer [4]	Ours
$\mathbf{D_c(10^{-2})} \downarrow$	9.55	24.22	16.18	**4.90**

7. Runtime

We report the runtime of different methods to demonstrate the efficiency of our method.

Table 2. Runtime analysis of different methods.

Methods	Training time	Prediction time
PcerNet[5]	>30h	3s
Split-and-Fit[6]	>50h	30m
Point2CAD[7]	✗	30s
Our stage1	–	5m
Our stage2	–	11m
Our fit	–	3s

8. Additional Visualization Comparisons.

To further validate the effectiveness of our two-stage Gaussian learning and CAD reconstruction pipeline, we provide

additional qualitative comparisons in Fig.4,5,6. These examples follow the same evaluation standard as the main paper, showing both segmentation results and reconstructed CAD models.

References

[1] Md Ferdous Alam et al. Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. *Trans. Mach. Learn. Res.*, 2025. 3

[2] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1

[3] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. In *ACM Trans. Graph.*, 2020. 1

[4] Yuan Li, Cheng Lin, Yuan Liu, Xiaoxiao Long, Chenxu Zhang, Ningna Wang, Xin Li, Wenping Wang, and Xiaohu Guo. Caddreamer: Cad object generation from single-view images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025. 3

[5] Yuanqi Li, Hongshen Wang, Yansong Liu, Jingcheng Huang, Shun Liu, and Chenyu Huang. Deep point cloud edge reconstruction via surface patch segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 3

[6] Yilin Liu, Jiale Chen, Shanshan Pan, Daniel Cohen-Or, Hao Zhang, and Hui Huang. Split-and-fit: Learning b-reps via structure-aware voronoi partitioning. In *ACM SIGGRAPH Annual Conference*, 2024. 3

[7] Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2cad: Reverse engineering cad models from 3d point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 3

[8] Yilin Liu et al. Hola: B-rep generation using a holistic latent representation. In *SIGGRAPH*, 2025. 3

[9] Paul L Rosin. Measuring rectangularity. *Machine Vision and Applications*, 1999. 1

[10] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1

[11] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025. 1

[12] Markus Worchel and Marc Alexa. Differentiable rendering of parametric geometry. In *ACM Trans. Graph.*, 2023. 1

[13] Haocheng Yuan, Adrien Bousseau, Hao Pan, Chengquan Zhang, Niloy J. Mitra, and Changjian Li. Diffcsg: Differentiable csg via rasterization. In *ACM Trans. Graph.*, 2024. 1

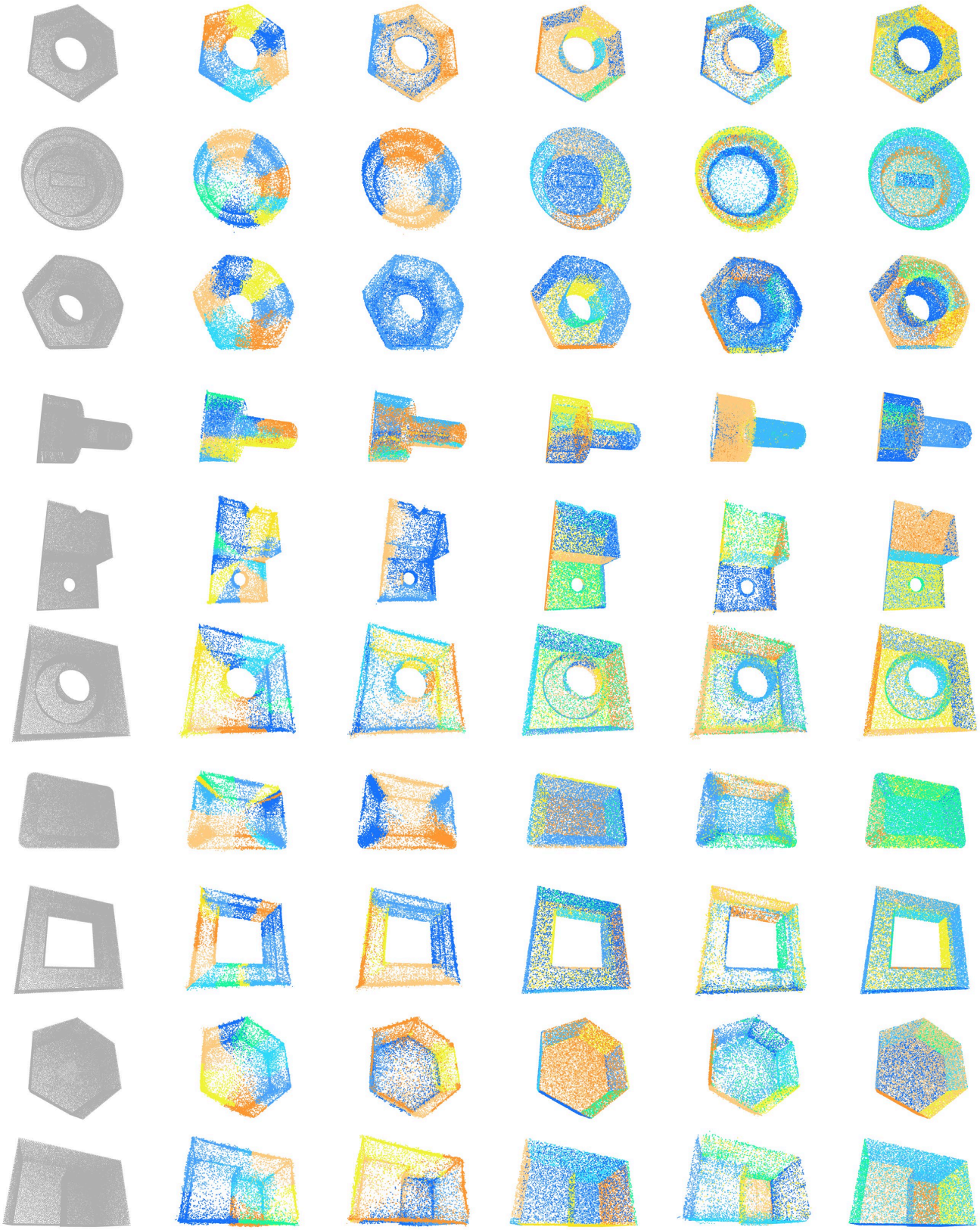


Figure 4. More visualization results.

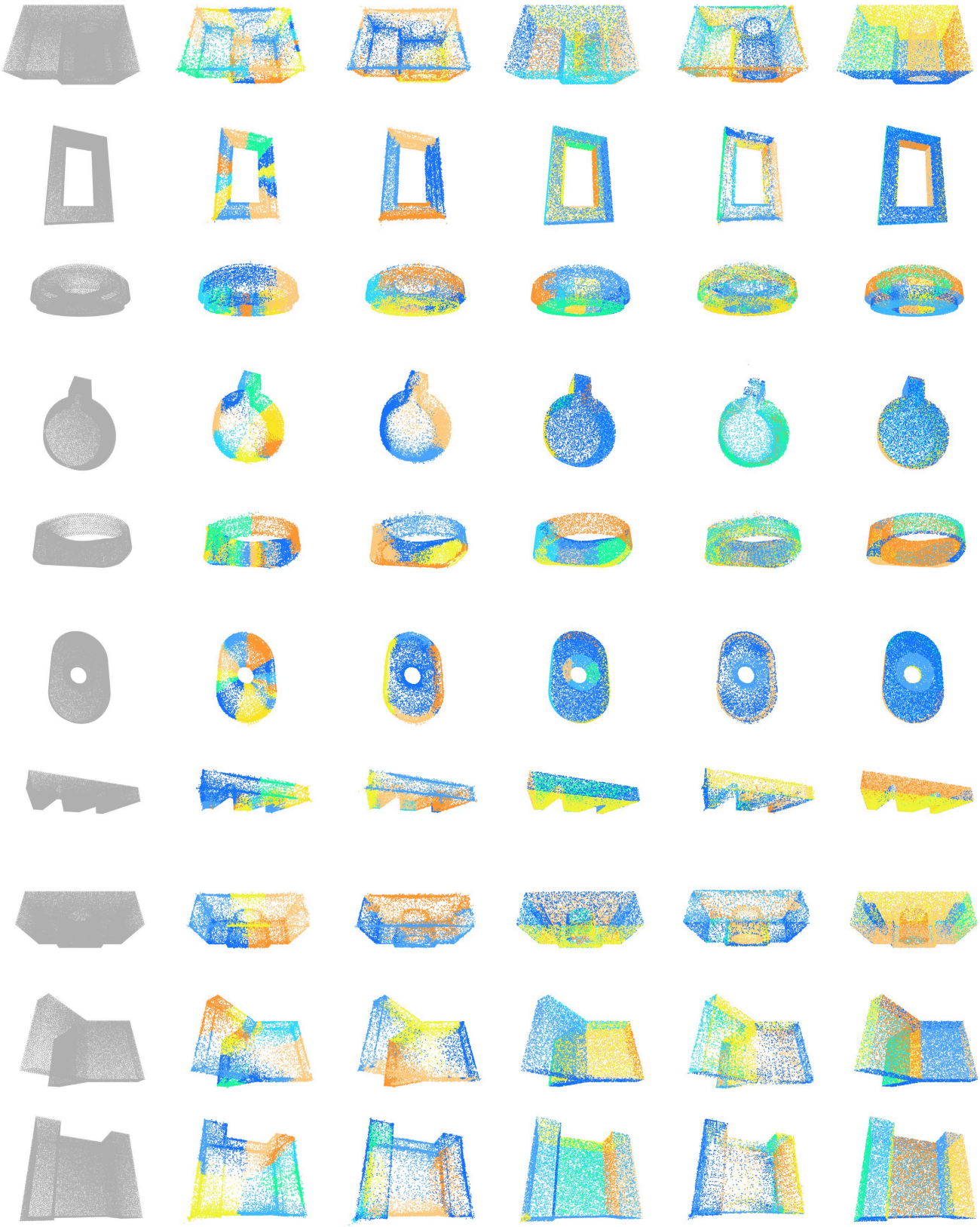


Figure 5. More visualization results.

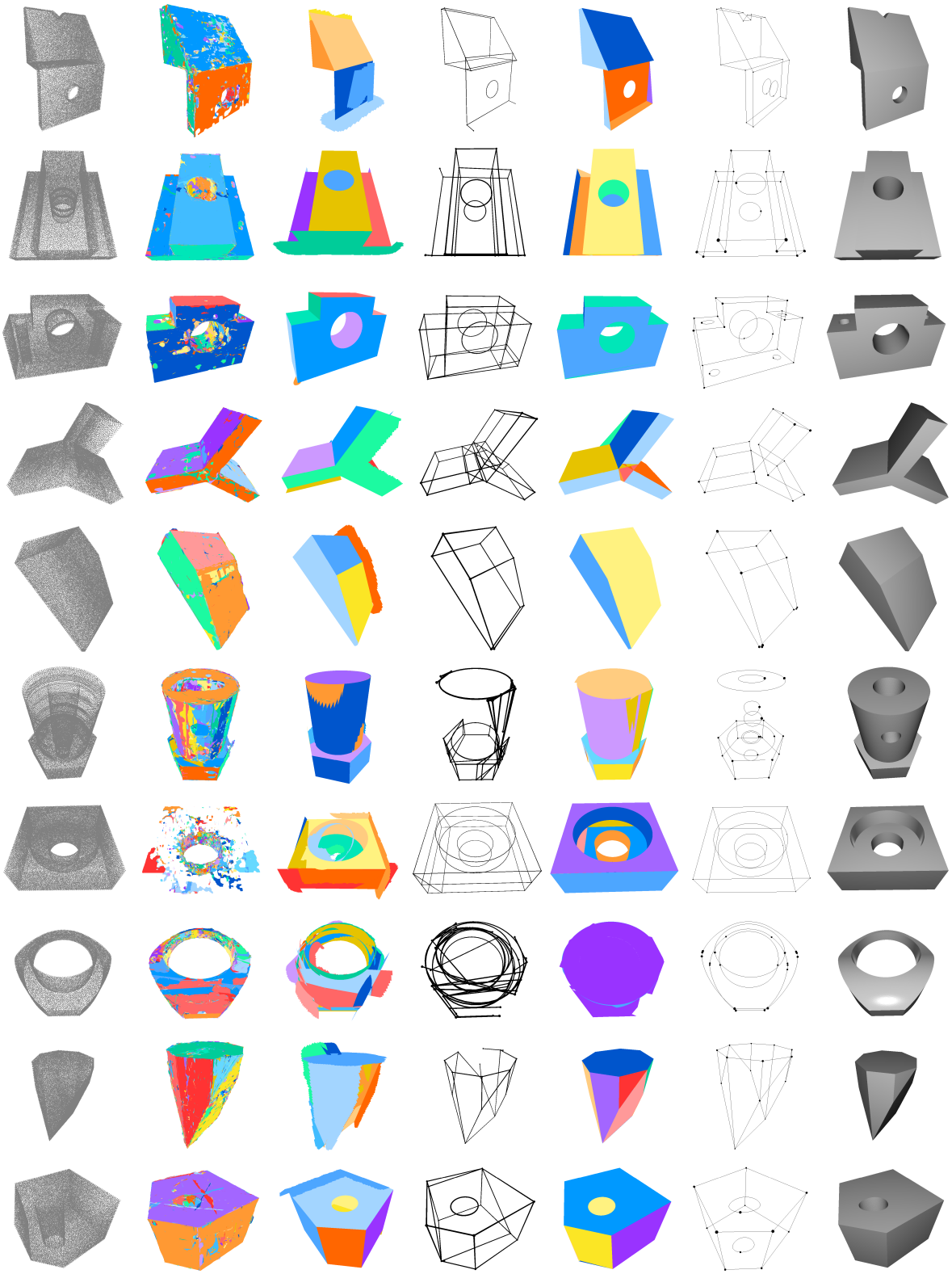


Figure 6. More visualization results.