Residual Primitive Fitting of 3D Shapes with SuperFrusta

Aditya Ganeshan* Matheus Gadelha† Thibault Groueix† Zhiqin Chen†
Siddhartha Chaudhuri† Vladimir Kim† Wang Yifan† Daniel Ritchie*

*Brown University †Adobe Research

adityaganeshan@gmail.com

Figure 1. (Left) Primitive assemblies inferred by our method capture a wide range of shapes, including hollow forms (vase), curved toroidal parts (bike), intricate geometry (ladder, robot), and smooth organic shapes (crab). (Right) Our approach shifts the reconstruction–parsimony Pareto frontier: compared to state-of-the-art methods, Marching Primitives [18] (MPS) and Primitive Anything [45] (PA), we achieve markedly lower reconstruction error using significantly fewer primitives.

Abstract

We introduce a framework for converting 3D shapes into compact and editable assemblies of analytic primitives, directly addressing the persistent trade-off between reconstruction fidelity and parsimony. Our approach combines two key contributions: a novel primitive, termed SuperFrustum, and an iterative fiting algorithm, Residual Primitive Fitting (ResFit). SuperFrustum is an analytical primitive that is simultaneously (1) expressive, being able to model various common solids such as cylinders, spheres, cones & their tapered and bent forms, (2) editable, being compactly parameterized with 8 parameters, and (3) optimizable, with a sign distance field differentiable w.r.t. its parameters almost everywhere. ResFit is an unsupervised procedure that interleaves global shape analysis with local optimization, iteratively fitting primitives to the unexplained residual of a shape to discover a parsimonious yet accurate decompositions for each input shape. On diverse 3D benchmarks, our method achieves state-of-the-art results, improving IoU by over 9 points while using nearly half as many primitives as prior work. The resulting assemblies bridge the gap between dense 3D data and human-controllable design, producing high-fidelity and editable shape programs.

1. Introduction

Recent breakthroughs in 3D generation have enabled the creation of high-quality assets from simple prompts [15, 38]. However, while visually impressive, these outputs are often structurally unorganized, posing challenges for downstream applications like animation, rigging, and interactive editing. Primitive-based representations offer a compelling alternative by distilling complex geometry into a compact assembly of interpretable, analytic parts. This approach yields editable assets and aligns with cognitive findings that humans perceive objects as compositions of simpler forms [3], providing a structured understanding that dense representations lack. The central challenge is converting these unstructured 3D assets into meaningful, primitive-based designs.

Inferring a primitive assembly from a raw 3D shape, however, presents a fundamental trade-off between reconstruction fidelity and program parsimony. Approaches that prioritize high fidelity often yield dense, redundant assemblies of overlapping primitives [1, 4, 8, 27, 34]. Conversely,

Figure 2. **SuperFrustum— An Expressive, Compact & Differentiable Primitive.** SuperFrustum is a unified analytic SDF primitive with only **8** parameters controlling dilation, taper, bulge, onion-like hollowing, profile roundness, and axial scaling. Its SDF is C^0-continuous and fully differentiable (almost eveywhere) with respect to all parameters, enabling robust inverse modeling and gradient-based optimization. As shown on the right, these parameters allow a single formulation to morph smoothly across common solids—cuboids, cylinders, cones, spheres, and toroidal variants—and to produce more complex shapes such as bent, hollow, or smoothly capped forms.

methods that enforce parsimony may fail to capture fine geometric details or curved structures [18, 26, 39]. Achieving a representation that is simultaneously expressive, compact, and editable remains an open challenge.

This persistent trade-off can be attributed to two factors. First, commonly used primitive families such as cuboids, superquadrics, or ellipsoids [2] may require a large number of instances to model the rich shape variations in 3D assets. Second, the inference procedures themselves have distinct limitations. Methods that first commit to a complete segmentation of the input rely on a fixed partition that may not align with what the primitives can efficiently represent [19, 35, 40, 49]. This makes the process brittle, as any initial segmentation errors propagate directly to the final assembly. On the other hand, optimization-driven approaches that fit a large "soup" of primitives from scratch must navigate a highly non-convex loss landscape [4, 28, 36].

To address these limitations, we introduce a framework that marries a highly expressive primitive with a robust inference strategy. At the core of our approach is the **SuperFrustum**, an analytic primitive that fills a key gap in prior work: existing primitive families typically satisfy only one or two of the critical desiderata—*expressivity*, *editability*, and *optimizability*. In contrast, SuperFrustum (1) spans common solids such as cylinders, cones, spheres, and their tapered or bent variants; (2) is compactly parameterized with just 8 parameters; and (3) admits a signed-distance field that is differentiable with respect to all parameters, enabling smooth blending and effective inverse modeling. Intriguingly, its design builds on analytic functions uncovered by the Shadertoy and Demoscene communities in their pursuit of highly expressive analytic forms with minimal description length [22–24, 29]. We find that, when carefully adapted, these formulations are exceptionally well-suited for inverse modeling.

To achieve parsimonious assemblies, an expressive primitive must be paired with an equally effective inference algorithm. We propose **Residual Primitive Fitting (ResFit)**, an unsupervised procedure that tightly interleaves global shape analysis with local primitive optimization to better navigate the highly non-convex reconstruction loss. Instead of optimizing a large set of primitives jointly from scratch, ResFit first analyzes the input geometry to propose initial structures based on global cues. These primitives are then refined via gradient descent to conform to the local geometry. The resulting assembly is subtracted from the target shape, and the process repeats on the unexplained residual. By alternating between proposing global structure and optimizing local parameters, ResFit allows these two signals to mutually inform each other, producing assemblies that are both compact and high-fidelity.

Our approach sets a new state-of-the-art on diverse 3D benchmarks. It consistently produces higher-fidelity reconstructions—**improving IoU by over 9 points**—while using nearly half the primitives of prior work, demonstrating a fundamental shift in the fidelity-parsimony frontier. These results are enabled by our two primary contributions:

1. **The SuperFrustum:** A single compact analytic primitive that spans a wide range of canonical volumetric forms while remaining differentiable and suitable for gradient-based optimization.
2. **Residual Primitive Fitting (ResFit):** An unsupervised inference procedure that alternates between global shape analysis and local primitive optimization to produce compact and accurate assemblies.

Beyond reconstruction, we demonstrate how this framework enables downstream applications including the generation of editable assets, the inference of structured CSG programs, and the enrichment of semantic part segmentations. **Code will be open-sourced upon acceptance.**

2. Related Works

Inferring primitive assemblies. Existing approaches fall into three main categories. *Shape-analysis–driven methods* [19, 35, 39, 40, 49] partition a shape into regions using geometric cues—such as curvature, thickness, or convexity—and then fit primitives to these regions. They produce structurally coherent decompositions when the partitions match the primitive family, but are often brittle across diverse shapes and sensitive to tuning. Since the decomposition is fixed and independent of what the primitives can represent, these methods struggle to balance fidelity and compactness. *Optimization-driven methods* directly adjust primitive parameters to minimize reconstruction error for a target shape [18, 28] or its renders [21]. While effective on small assemblies, they often require many primitives for high fidelity, as reconstruction loss tends to dominate disentanglement and parsimony without strong initialization. *Learned methods* predict primitive parameters or part layouts using neural networks. Some methods train the network on supervised data [10, 13, 20, 45] while others formulate unsupervised reconstruction-based objectives to infer the assemblies [4–6, 8, 11, 25–27, 32, 41, 50]. Such models achieve high reconstruction accuracy on domains similar to their training data but generalize poorly to novel or complex objects.

Our approach combines the strengths of analysis- and optimization-driven paradigms: rather than committing to a single decomposition or a fixed primitive set, we *interleave* analysis and optimization so that each informs the other. This bidirectional formulation adapts the decomposition to the representational capacity of the primitives, producing assemblies that remain both compact and geometrically faithful.

Primitive representations. Primitive design has progressed from simple analytic forms to more expressive but increasingly complex parameterizations. Early methods used cuboids or cylinders [8, 36], which are interpretable but limited in expressivity. Superquadrics [2, 7, 26] and algebraic surfaces [44] enlarge the shape space but sacrifice editability and cannot exactly reproduce canonical solids such as cubes or cones—common in manufactured objects. Recent generalized-cylinder–based primitives [31, 48] increase flexibility yet still fall short in reconstruction fidelity. Neural implicit part representations [8, 9, 27, 43] offer high expressivity but are opaque, costly, and difficult to control or reuse.

In parallel, the graphics and demoscene communities have explored unified analytic primitives that morph between basic shapes within a single functional form [22–24, 29]. These formulations were developed to minimize scene-description size and enable real-time rendering, not for inverse modeling or differentiable fitting. SuperFrustum draws inspiration from this work, extending it to a broader shape space and demonstrating its effectiveness for high-fidelity primitive assembly inference.

Figure 3. **ResFit** infers parsimonious assemblies by interleaving *shape analysis* and *primitive optimization*. Shape decomposition provides initial primitives, which are refined with decomposition-aware optimization. Residual unexplained volumes are then extracted and seeded with new primitives.

3. Method

We define the primitive assembly inference task as follows: given a 3D shape x, our goal is to infer a primitive assembly z composed of analytic primitives whose execution $E(z)$ reconstructs the input shape. Each program z defines a sequence of primitives $\{f_{\theta_i}\}_{i=1}^{|z|}$ combined through compositional operators to yield a closed surface $E(z)$. Following Occam's razor, we seek programs that are both *accurate* and *compact*. Formally, we aim to maximize the following objective:

$$z^* = \arg\max_z \ \mathcal{O}(x, z), \tag{1}$$

$$\mathcal{O}(x, z) = \mathcal{R}(x, E(z)) - \alpha|z|, \tag{2}$$

where $\mathcal{R}$ measures the reconstruction accuracy between the input shape x and the program execution $E(z)$, $|z|$ denotes the program complexity (or number of primitives in the program), and α controls the trade-off between accuracy and compactness. Maximizing $\mathcal{O}$ thus favors concise programs that explain the geometry with a small set of expressive parts.

We now summarize the components of our method. Section 3.1 introduces ResFit, our iterative fitting procedure. Section 3.2 defines SuperFrustum, the unified analytic primitive used in all assemblies. Section 3.3 describes our MSD-based initialization strategy, and Section 3.4 details the optimization process that balances geometric fidelity with parsimony.

3.1. ResFit : Residual Primitive Fitting

Purely optimization-based methods often produce entangled reconstructions, while analysis-based approaches partition shapes without considering the primitive family's representational capacity. This creates a disconnect between the geometric analysis (top-down) and the primitive representation (bottom-up). Residual Primitive Fitting (ResFit) bridges this divide by interleaving shape analysis and assembly optimization, allowing each phase to inform the other and yielding assemblies that are both compact and geometrically faithful.

Our procedure alternates between analysis and optimization (Fig. 3). The analysis stage decomposes the current residual volume into regions that seed new primitives. The optimization stage then adjusts parameters to maximize $\mathcal{O}$ (Eq. 1), separating explained geometry from remaining residuals. This cycle repeats until $\mathcal{O}$ saturates or a fixed iteration budget K is reached.

Several design choices ensure that the iterative loop can correct both over- and under-parameterization. To prevent over-parameterization, we seed few primitives per round and employ parsimony-aware optimization: a soft regularizer penalizes redundancy during fitting, while hard pruning removes parts that degrade $\mathcal{O}$. To address under-parameterization, primitives are optimized based on their local support, and the full assembly is re-optimized in each round. This enables self-correction as new parts are added, allowing the system to converge toward a compact and coherent structure.

3.2. Expressive, Editable & Optimizable Primitive

Primitive	Expressive	Editable	Optimizable
S. quadrics [26]	✗	✓	✓
Alg. Surf [44]	✓	✗	✓
Multi-type [45]	✓	✓	✗
SuperFrusta	✓	✓	✓

An ideal primitive for inverse graphics must be *expressive* enough for diverse forms, *editable* via intuitive controls, and robustly *optimizable*. As existing families often fall short, we introduce SuperFrusta, a unified analytic primitive designed to meet all three desiderata.

A SuperFrustum is the zero-level set of a signed distance function

$$SF(\mathbf{p}) \;=\; f(\mathbf{p};\theta), \qquad \theta \;=\; (\mathbf{s}, r, d, t, b, o), \tag{3}$$

with parameters $\theta = (\mathbf{s}, r, d, t, b, o)$. These 8 scalars intuitively control anisotropic scale ($\mathbf{s}$), profile rounding (r), dilation (d), taper (t), bulge (b), and onion/shell thickness (o), as shown in Fig. 2 (further implementation details and the reference code are provided in the supplementary). Its continuous, piecewise-C^1 formulation spans a wide range of shapes including cuboids, cylinders, cones, and tori, and is differentiable almost everywhere, enabling stable gradient-based fitting.

The complete primitive assembly $z = E(z)$ is formed by composing transformed SuperFrusta. Each instance i has a pose (R_i, t_i) and shape parameters θ_i, yielding a signed distance $g_i(\mathbf{p}) = f(R_i^\top(\mathbf{p} - t_i); \theta_i)$. The final implicit field $\mathcal{F}$ is obtained by recursively applying a smooth union operator U:

$$\begin{aligned} \mathcal{F}_1(\mathbf{p}) &= g_1(\mathbf{p}), \\ \mathcal{F}_{k+1}(\mathbf{p}) &= U\big(\mathcal{F}_k(\mathbf{p}),\, g_{k+1}(\mathbf{p});\, \beta_k\big), \end{aligned} \tag{4}$$

where β_k controls blend sharpness. The final surface is the zero level set of $\mathcal{F}$.

3.3. Shape Decomposition for SuperFrusta

ResFit initializes primitives from the volumetric regions produced by a shape decomposition method, and its performance improves when the chosen decomposition strategy aligns with the primitive family's expressiveness. While recent work adapt Approximate Convex Decomposition (ACD) [39] for initializing primitives, we find an adapted variant of Morphological Shape Decomposition (MSD) [28] is more suitable for initializing SuperFrusta.

MSD is an iterative "peel the thickest part first" technique. At each step, it finds the largest connected region of roughly uniform thickness, extracts it, removes it from the shape, and repeats on the residual. This process yields a thickness-ordered set of volumetric regions for primitive initialization, as shown in Figure 4.

Formally, given a signed distance field $f(\mathbf{p})$, each iteration k identifies the thickest interior region Γ_k by finding the connected component (cc) that survives erosion up to a radius $|\tau|$:

$$\Gamma_k \subseteq \{\mathbf{p} \in \Omega \mid f(\mathbf{p}) \le \tau\}, \qquad \Gamma_k \text{ is a cc.} \tag{5}$$

The threshold $\tau \le 0$ is the minimum value such that $\mathrm{Vol}(\Gamma_k)$ meets a volume fraction κ. To recover its full spatial extent, we dilate Γ_k back by the same radius, $R_k = \Gamma_k \oplus B_{|\tau|}$. This part R_k is recorded and subtracted from the shape by updating the residual field:

$$f_{k+1}(\mathbf{p}) = f_k(\mathbf{p}) \setminus R_k, \quad f_1 \equiv f. \tag{6}$$

Repeating this process produces a sequence of candidate regions $\{R_k\}$ ordered by decreasing thickness.

MSD offers two key advantages over ACD [14, 40] for this task. First, ACD's convexity constraint over-partitions non-convex structures that a single SuperFrustum can model, such as the bent and hollow forms shown in Figure 4 (bottom). Second, MSD is substantially more robust to the noisy surface artifacts present in the residual volumes generated during our iterative fitting loop, making it better suited for ResFit.

For each decomposed part volume, we instantiate a SuperFrustum. We initialize its parameters by using PCA on points sampled within the volume. Cylindricity score along

Figure 4. Morphological Shape Decomposition (MSD) iteratively extracts connected regions of similar thickness. Top: successive MSD partitions of a input mesh. Bottom: MSD yields regions that form suitable initialization seeds for SuperFrusta —capturing non-convex structures such as bicycle tires (left), a cat's curved tail (center), and bowl rims (right). In contrast, CoACD over-partitions these regions into many convex fragments, often using axis-aligned cuts that produce semantically misaligned parts.

the different PCA axis is used to select a canonical direction. Pose (R, t) and Scale is then inferred w.r.t the canonical axis. Refer to the supplementary for further details.

3.4. Decomposition-Aware Optimization

We optimize the assembly parameters to maximize the objective $\mathcal{O}$ (Eq. 1) in two stages. First, a differentiable phase minimizes a corresponding loss via gradient descent. Second, a discrete pruning phase removes primitives that degrade $\mathcal{O}$. The differentiable loss comprises three components addressing reconstruction fidelity, program parsimony, and program quality.

Reconstruction. The reconstruction loss is a differentiable surrogate for $\mathcal{R}$ in Eq. 2. We supervise the predicted occupancy field $\hat{o}(\mathbf{p}) = \sigma(-\beta\,\mathcal{F}(\mathbf{p}))$ of the current assembly against the ground-truth occupancy $o(\mathbf{p})$. Samples $\mathbf{p}$ are drawn uniformly from the shape's volume and densely near its surface. To better reconstruct thin, high-curvature structures, each point is weighted by the principal curvature $\kappa(\mathbf{p})$ of the target mesh. The loss is evaluated only within a spatial mask $\mathcal{M} = \{\mathbf{p} \mid \mathcal{F}(\mathbf{p}) < \tau\}$ to focus optimization on signals from the assembly's vicinity.

$$
\begin{aligned}
w(\mathbf{p}) &= 1 + \sigma(\kappa(\mathbf{p})), \\
\mathcal{L}_{\text{rec}} &= \frac{1}{|\mathcal{M}|} \sum_{\mathbf{p}\in\mathcal{M}} w(\mathbf{p})\big(\hat{o}(\mathbf{p}) - o(\mathbf{p})\big)^2.
\end{aligned} \tag{7}
$$

Parsimony. To encourage compact assemblies, each primitive i is assigned a stochastic existence variable $q_i \in (0, 1)$ sampled via a Gumbel-Softmax distribution. Its signed distance field is then modulated as $f_i^*(\mathbf{p}) = q_i\, f_i(\mathbf{p}) + (1 - q_i)$, which smoothly erodes primitives with low existence probability. The parsimony loss penalizes the expected number of active primitives: $\mathcal{L}_{\text{count}} = \sum_i q_i$.

Quality. To improve editability and prevent geometrically entangled or overly blended assemblies, we add a structural regularizer that combines overlap and smooth-union consistency losses:

$$
\mathcal{L}_{\text{qual}} = \underbrace{\max(1, \sum\nolimits_i \hat{o}_i(\mathbf{p}))}_{\mathcal{L}_{\text{overlap}}} + \underbrace{\hat{o}(\mathbf{p}) - \min(\sum\nolimits_i \hat{o}_i(\mathbf{p}), 1)}_{\mathcal{L}_{\text{union}}},
$$

where $\hat{o}_i$ is the occupancy of primitive i. The $\mathcal{L}_{\text{overlap}}$ term penalizes regions where multiple primitives are simultaneously active, discouraging redundant coverage. The $\mathcal{L}_{\text{union}}$ term penalizes regions that are occupied by the smooth union assembly but not by any of the independent primitives, discouraging excessive blending.

The total differentiable loss is the weighted sum of these components:

$$
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{count}}\mathcal{L}_{\text{count}} + \lambda_{\text{qual}}\mathcal{L}_{\text{qual}}. \tag{8}
$$

Pruning. After the differentiable optimization converges, a discrete pruning step further simplifies the assembly. Primitives with negligible volume or contribution are tested for removal, and deletions are greedily accepted if they improve the primary objective $\mathcal{O}$.

4. Experiments

Datasets. We evaluate on two datasets capturing generated and real-world assets. As the *3DGen-Prim* dataset [39] is not public, we recreate it using 510 prompts from 3DGen-Bench [47] with the Hunyuan3D-2.1 [33] generator. Our second dataset contains 500 geometrically diverse shapes from Toys4K [30], selected via farthest-point sampling.

Metrics. We evaluate *reconstruction accuracy* and *program quality*. For accuracy, we use standard metrics: voxel IoU (128^3), Chamfer Distance (CD), and Earth Mover's Distance (EMD) over 2048 randomly sampled surface points. We also report a *Bidirectional Surface IoU (BiSurfIoU)* to better capture surface fidelity, computed as the mean of IoU scores from near-surface points sampled on both the target and reconstructed shapes, with the latter surface extracted via dual contouring of the SDF. For program quality, we introduce four metrics. Program length ($|z|$) and

Method	Reconstruction Quality				Program Quality			
	IOU (↑)	BiSurfIOU (↑)	CD (↓)	EMD (↓)	#Prims (↓)	Overlap (↓)	IntraPrim (↓)	InterPrim (↑)
3DGen-Prim Dataset								
PA [45]	34.62	30.56	7.588	0.119	19.81	0.402	0.359	0.238
PA (TTO)	53.73	37.90	0.742	0.131	64.87	0.493	0.244	0.156
MPS [18]	82.67	71.53	0.884	0.093	42.96	0.684	0.250	0.157
Ours	88.74	80.19	0.168	0.073	23.98	0.210	0.244	0.207
Toy4K Dataset								
PA [45]	36.76	31.63	5.771	0.142	19.14	0.341	0.322	0.255
PA (TTO)	49.08	37.78	1.220	0.146	44.78	0.408	0.237	0.179
MPS [18]	80.60	72.75	1.147	0.086	30.62	0.588	0.245	0.201
Ours	89.92	85.66	0.154	0.067	23.67	0.208	0.221	0.202

Table 1. Evaluation on 3DGen-Prim [47] and Toys4K [30] datasets. Our method achieves the best reconstruction and program quality scores simultaneously—improving IOU by 6–9 points while using roughly half as many primitives. Gold = best, Silver = second best.

Overlap Ratio (the volumetric percentage of the shape covered by multiple primitives) measure parsimony and redundancy. To quantify semantic coherence, we use two metrics based on PartField features [16]. After associating surface points on the target mesh to their nearest primitive, we compute: *IntraPrim*, the mean feature variance within each primitive (lower is better), and *InterPrim*, the average nearest-neighbor distance between primitive feature centroids (higher is better). These scores are aggregated using a size-weighted average.

Baselines. We compare our method against two state-of-the-art approaches. *Primitive Anything (PA)*[45] is a learning-based method trained on a large dataset of manually annotated shapes to predict assemblies of cuboids, cylinders, and ellipsoids from point cloud inputs. Following the original work, we also report its test-time optimization variant, *PA (TTO)*, which refines its predictions using Chamfer Distance. *Marching Primitives (MPS)*[18] serves as a strong optimization-based baseline that directly optimizes a superquadric-based assembly from an SDF grid to achieve state-of-the-art reconstruction fidelity. We run MPS at 128 voxel resolution to match our input. We omit comparisons to methods outperformed by MPS [7, 17] or those without public code [37, 39].

Implementation Details. All experiments use a fixed set of hyperparameters unless stated otherwise. ResFit runs for a maximum of 10 fitting rounds or until convergence, with each round applying 7 iterations of MSD. The high-level objective $\mathcal{O}$ (Eq. 1) combines curvature-weighted surface IoU with a program-length penalty ($\alpha = 10^{-3}$). During optimization, the loss weights are set to $\lambda_{\text{count}} = 10^{-3}$ and $\lambda_{\text{qual}} = 10^{-2}$ (Eq. 8). Additional optimization details and ablations are provided in the supplementary material.

4.1. Parsimonious High fidelity Assemblies

Table 1 summarizes the reconstruction and program quality metrics on both datasets. Across all reconstruction measures, our method improves IoU scores by $+6.1$ points on 3DGen-Prim and $+9.3$ points on Toys4K over prior work. We attribute this performance to the expressivity of the SuperFrustum primitive and the iterative analysis-optimization loop of ResFit.

These reconstruction gains are accompanied by improved program quality. Our assemblies use approximately half as many primitives as Marching Primitives while reducing volumetric overlap by over $3\times$. The inferred primitives also demonstrate high semantic coherence: our method achieves the lowest *IntraPrim* scores, indicating high semantic purity within primitives, and among the highest *InterPrim* scores, reflecting meaningful distinctions between parts. These results demonstrate that ResFit produces assemblies that are simultaneously more *accurate*, *compact*, and *semantically interpretable*.

Qualitative comparisons in Figure 5 corroborate these findings. Our assemblies exhibit higher geometric fidelity and are more interpretable, using a compact set of non-overlapping, semantically aligned primitives. In contrast, baseline reconstructions can show lower fidelity on complex structures and tend to produce assemblies with greater primitive overlap.

4.2. Ablative Analysis

Primitive Assembly Design. Table 2 compares different primitive families and composition operators. Our full SuperFrustum formulation achieves the highest reconstruction fidelity. Disabling smooth unions reduces accuracy and increases overlap, as continuous volumes must then be formed by intersecting primitives rather than by smooth blending. Superprimitive [22], a variant of our primitive

Figure 5. Our method reconstructs target shapes with high geometric fidelity and produces more interpretable assemblies, using compact, minimally-overlapping primitives. In contrast, Primitive Anything [45] (PA) and Marching Primitives [18] (MPS) often lose fine structure and generate assemblies with substantial primitive overlap.

	S. Union	IOU	CD	#Prims	Overlap
Cuboid	✗	82.33	0.129	20.49	0.298
S Q	✗	76.50	0.523	18.17	0.286
S P	✗	86.66	0.134	21.36	0.291
S F	✗	87.15	0.173	18.68	0.257
S F	✓	88.37	0.147	21.46	0.199

Table 2. Primitive representation ablation: SuperFrustum delivers superior performance over Cuboids, Superquadrics (SQ) and SuperPrimitive(SP) (ref. Section 4.2). Combining SuperFrustum with smooth union further improves performance.

Step	**Method**	IOU	CD	#Prims	Overlap
Single	CoACD	86.56	0.368	26.67	0.226
Single	MSD	87.95	0.337	28.17	0.236
ResFit	CoACD	88.02	0.290	24.18	0.214
ResFit	MSD	89.86	0.157	23.46	0.207

Table 3. Fitting and decomposition ablation: ResFit, which interleaves analysis and optimization, outperforms a single-shot fitting approach (cf. Section 4.2). Moreover, pairing ResFit with MSD yields better results than using CoACD [40].

without tapering or bending also lowers accuracy, confirming these degrees of freedom are important for capturing curved and non-uniform structures. Substituting our primitive with cuboids or superquadrics (SQs) further degrades performance. SQs are particularly susceptible to poor local minima when their axes misalign with the target geometry—an issue that methods like MPS mitigate via non-differentiable heuristics such as periodic axis-flipping, which are excluded from our controlled comparison.

Decomposition and Fitting Strategy. Table 3 compares our iterative ResFit procedure against a single-shot fitting baseline, using both MSD and CoACD for initialization. The single-shot approach optimizes all primitives simultaneously after the initial decomposition. This makes it sensitive to the initial partition, as it has no mechanism to reallocate capacity to unexplained regions, resulting in less accurate and less compact assemblies.

In contrast, ResFit uses multiple refinement rounds to progressively reallocate primitives toward residual errors and prune where unnecessary. This iterative process achieves higher fidelity with fewer primitives and lower overlap. When comparing decomposition strategies, MSD consistently outperforms CoACD. MSD's ability to produce non-convex partitions provides better initializations for our primitives, especially on the curved, hollow, and branching geometries as shown in Figure 4.

Timing. On the Toys4K test set, the full ten-round version of ResFit takes 652.6 s per shape on average. However, even a two-round variant offers a strong quality–time trade-off: it runs in 184.1 s while achieving 86.54 IOU with only 15.54 primitives. This matches—and slightly exceeds—the reconstruction accuracy of MPS at 256^3 resolution (86.30 IOU) while using *over* $5\times$ *fewer* primitives and a comparable runtime (194.6 s). PA (58.3 s) and MPS (37.9 s at 128^3) are faster but produce lower-quality assemblies. We note that ResFit is not yet optimized for speed; dedicated CUDA

Figure 6. Assigning per-primitive spherical 2D textures & optimizing it against a textured mesh, begets *Edtiable* & *Deployable* assets.

Method	IOU	CD	#Prims	Overlap
CAPRI-Net [46]	84.46	0.100	23.0	N/A
Ours (Solid)	82.64	0.192	14.18	0.342
Ours	88.30	0.127	13.16	0.186

Table 4. CSG inference on the ABC dataset [12]: Our method achieves comparable reconstruction accuracy to CAPRI-Net [46] while using significantly fewer primitives.

kernels for SuperFrustum may reduce runtime.

5. Applications

Our representation enables several downstream uses that combine visual quality, editability, and analytic structure. We highlight four such applications and provide implementation details in the supplementary.

Editable and Deployable Asset Generation. Our primitives are simultaneously compact, editable, and capable of high-fidelity reconstruction, allowing them to serve directly as deployable 3D assets. To produce textured assemblies, we associate each primitive with a local 2D spherical texture map that we optimize against the target textured mesh. The resulting textured assemblies can be directly deployed in real-time sphere traced scenes, while remaining editable (see Fig. 6).

Inferring Canonical CSG Programs. Although our framework is designed for smooth, soft-union assemblies, it can also infer discrete Constructive Solid Geometry (CSG) programs composed of canonical solids. We achieve this by constraining the parameters of SuperFrustum to be a barycentric interpolation of parameters to fetch canonical shapes—cuboid, cylinder, cone, and sphere—within the SuperFrustum formulation. Despite the lack of subtraction as a compositional operator, our primitive space natively contains "subtracted" shapes via the onion operator, which we include in the list of canonical shapes. Fitting under these constraints yields solid CSG programs that remain compact and interpretable. In Table 4, we compare this constrained version of our method, named "solid" to CAPRI-Net [46] on a randomly sampled subset of the ABC dataset [12] containing 100 samples. Our approach achieves nearly the same CSG reconstruction accuracy as CAPRI-Net while using roughly two-third as many primitives. As shown in Fig. 6, our inferred programs produce cleaner, less entangled assemblies—owing to the analytic expressivity of our primitives and the structured refinement in ResFit.

Figure 7. Our method can infer CSG programs using canonical solids (e.g., cylinders, cuboids; cf. Sec. 5), producing far more parsable and structured trees than CAPRI-Net [46].

Figure 8. Our method can infer primitive assemblies from images by leveraging Text-2-3D models (Hunyuan3D-2.1 [33]).

Image to primitive. Our method can seamlessly be combined with 3D generative models to achieve image to primitive assembly. In Figure 8, we use Hunyuan3D-2.1 [33] with ResFit on samples from the 3DGen-Bench suite [47].

Semantic Segmentation Enrichment. Manually annotating parts can be quite expensive - as a result datasets often provide coarse grained annotations. ResFit can help to annotate finer parts. In Figure 9, we intersect coarse semantic labels from the PartObjVerse dataset [42] with our primi-

Figure 9. ResFit enables finer, semantically consistent part segmentation. Row 1 shows the coarse semantic regions provided in PartObjVerse. Row 2 shows the primitive assemblies inferred by ResFit. Intersecting each coarse region with its corresponding assembly (Row 3) yields meaningful sub-parts—capturing functional structure while remaining strictly within the original semantic boundaries.

tive assemblies to enhance segmentation granularity. As a result, we subdivide large parts into functionally meaningful subcomponents without drifting outside their semantic boundaries. This suggests a promising direction for integrating analytic decomposition as a prior for open-world part segmentation tasks.

6. Conclusion

We introduced a framework for converting 3D shapes into compact and editable assemblies of analytic primitives. Our method combines two contributions: SuperFrustum, an expressive, compact, and optimizable analytic primitive; and Residual Primitive Fitting, an iterative inference algorithm that couples shape analysis with primitive optimization to recover parsimonious yet accurate assemblies. Together, they shift the reconstruction–parsimony Pareto frontier, achieving state-of-the-art performance across benchmarks while producing high-fidelity, editable shape programs. Despite its expressiveness, ResFit is still restricted by its purely additive composition; shapes requiring subtractive operations remain challenging. Future work includes extending our decomposition strategies (e.g., tree-of-shapes) and developing richer applications in CSG modeling, interactive editing, and structured scene understanding.

References

[1] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 1

[2] Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1981. 2, 3

[3] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2): 115, 1987. 1

[4] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 45–54, 2020. 1, 2, 3

[5] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 31–44, 2020.

[6] Boyang Deng, Sumith Kulal, Zhengyang Dong, Congyue Deng, Yonglong Tian, and Jiajun Wu. Unsupervised learning of shape programs with repeatable implicit parts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[7] Elisabetta Fedele, Boyang Sun, Leonidas Guibas, Marc Pollefeys, and Francis Engelmann. SuperDec: 3D Scene Decomposition with Superquadric Primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 3, 6

[8] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4857–4866, 2020. 1, 3

[9] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Spaghetti: editing implicit shapes through part aware generation. *ACM Trans. Graph.*, 41(4), 2022. 3

[10] R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. Shapeassembly: learning to generate programs for 3d shape structure synthesis. *ACM Trans. Graph.*, 39(6), 2020. 3

[11] R. Kenny Jones, Renhao Zhang, Aditya Ganeshan, and Daniel Ritchie. Learning to edit visual programs with self-supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3

[12] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[13] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: generative recursive autoencoders for shape structures. *ACM Trans. Graph.*, 36 (4), 2017. 3

[14] Jyh-Ming Lien and Nancy M Amato. Approximate convex decomposition of polyhedra. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 121–131, 2007. 4

[15] Jian Liu, Xiaoshui Huang, Tianyu Huang, Lu Chen, Yuenan Hou, Shixiang Tang, Ziwei Liu, Wanli Ouyang, Wangmeng Zuo, Junjun Jiang, et al. A comprehensive survey on 3d content generation. *arXiv preprint arXiv:2402.01166*, 2024. 1

[16] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 6

[17] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and accurate superquadric recovery: A probabilistic approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2676–2685, 2022. 6

[18] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory Chirikjian. Marching-primitives: Shape abstraction from signed distance function. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 6, 7

[19] Khaled Mammou et al. Voxelized hierarchical approximate convex decomposition – v-hacd version 4. https://github.com/kmammou/v-hacd, 2022. GitHub repository, accessed 2025-11-13. 2, 3

[20] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6):Article 242, 2019. 3

[21] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, and Mathieu Aubry. Differentiable Blocks World: Qualitative 3D Decomposition by Rendering Primitives. In *NeurIPS*, 2023. 3

[22] paniq. sdsuperprim. Shadertoy shader, . https://www.shadertoy.com/view/Xdy3Rm. 2, 3, 6

[23] paniq. sdsuperprim. Shadertoy shader, . https://www.shadertoy.com/view/MsVGWG.

[24] paniq. sdsuperpill. Shadertoy shader, . https://www.shadertoy.com/view/MdK3RW. 2, 3

[25] Chanhyeok Park and Minhyuk Sung. Split, merge, and refine: Fitting tight bounding boxes via over-segmentation and iterative search. In *Proceedings - 2024 International Conference on 3D Vision, 3DV 2024*, pages 1468–1477. Institute of Electrical and Electronics Engineers Inc., 2024. Publisher Copyright: © 2024 IEEE.; 11th International Conference on 3D Vision, 3DV 2024 ; Conference date: 18-03-2024 Through 21-03-2024. 3

[26] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 4

[27] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021. 1, 3

[28] I. Pitas and A.N. Venetsanopoulos. Morphological shape decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):38–45, 1990. 2, 3, 4

[29] Inigo Quilez. 3d sdfs — distance functions. https://iquilezles.org/articles/distfunctions/, n.d. 2, 3

[30] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, 2021. 5, 6

[31] Chunyi Sun, Junlin Han, Runjia Li, Weijian Deng, Dylan Campbell, and Stephen Gould. Unsupervised decomposition of 3d shapes into expressive and editable extruded profile primitives. In *SIGGRAPH*, 2025. 3

[32] Chun-Yu Sun, Qian-Fang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Trans. Graph.*, 38(6), 2019. 3

[33] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 5, 8

[34] Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Emiris, Yannis Avrithis, and Leonidas Guibas. Generating part-aware editable 3d shapes without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4466–4478, 2023. 1

[35] Jean-Marc Thiery, Emilie Guy, and Tamy Boubekeur. Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia 2013)*, 32(6):Art. No. 178, 2013. 2, 3

[36] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2, 3

[37] Cheng Wang, Xinzhu Ma, Bin Wang, Shixiang Tang, Yuan Meng, and Ping Jiang. Point2primitive: Cad reconstruction from point cloud by direct primitive prediction. *arXiv preprint arXiv:2505.02043*, 2025. 6

[38] Chen Wang, Hao-Yang Peng, Ying-Tian Liu, Jiatao Gu, and Shi-Min Hu. Diffusion models for 3d generation: A survey. *Computational Visual Media*, 11(1):1–28, 2025. 1

[39] Yuhan Wang, Weikai Chen, Zeyu Hu, Runze Zhang, Yingda Yin, Ruoyu Wu, Keyang Luo, Shengju Qian, Yiyan Ma, Hongyi Li, Yuan Gao, Yuhuan Zhou, Hao Luo, Wan Wang, Xiaobin Shen, Zhaowei Li, Kuixin Zhu, Chuanlang Hong, Yueyue Wang, Lijie Feng, Xin Wang, and Chen Change Loy. Light-sq: Structure-aware shape abstraction with superquadrics for generated meshes. In *SIGGRAPH Asia*, 2025. 2, 3, 4, 5, 6

[40] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022. 2, 3, 4, 7

[41] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Trans. Graph.*, 40(4), 2021. 3

[42] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024. 8

[43] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: learning articulated shapes from sparse image ensemble via 3d part discovery. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 3

[44] Mohsen Yavartanoo, Jaeyoung Chung, Reyhaneh Neshatavar, and Kyoung Mu Lee. 3dias: 3d shape reconstruction with implicit algebraic surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12446–12455, 2021. 3, 4

[45] Jingwen Ye, Yuze He, Yanning Zhou, Yiqin Zhu, Kaiwen Xiao, Yong-Jin Liu, Wei Yang, and Xiao Han. Primitiveanything: Human-crafted 3d primitive assembly generation with auto-regressive transformer, 2025. 1, 3, 4, 6, 7

[46] Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. Capri-net: Learning compact cad shapes with adaptive primitive assembly. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11768–11778, 2022. 8

[47] Yuhan Zhang, Mengchen Zhang, Tong Wu, Tengfei Wang, Gordon Wetzstein, Dahua Lin, and Ziwei Liu. 3dgen-bench: Comprehensive benchmark suite for 3d generative models, 2025. 5, 6, 8

[48] Mingrui Zhao, Yizhi Wang, Fenggen Yu, Changqing Zou, and Ali Mahdavi-Amiri. Sweepnet: Unsupervised learning shape abstraction via neural sweepers. In *European Conference on Computer Vision*. Springer, 2024. 3

[49] Xiangyu Zhu, Zhiqin Chen, Ruizhen Hu, and Xiaoguang Han. Controllable shape modeling with neural generalized cylinder. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[50] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 900–909, 2017. 3