DCCVT: Differentiable Clipped Centroidal Voronoi Tessellation

Supplementary Material

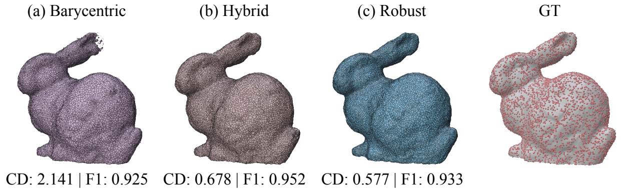

Figure 1. Visual comparison of barycentric (a), hybrid (b), and robust (c) projection methods against the ground truth (GT).

This supplemental document provides additional details on:

- our robust plane fitting method and its impact on the results (Sec. 1);
- the detailed derivation of **??** (Sec. 2);
- an analysis of how denser point clouds affect our optimization (Sec. 3);
- an ablation study on the effect of SDF regularization over the Thingi32 dataset (Sec. 4);
- additional information and results (Sec. 5);

1. Robust plane fitting

?? defines our minimization problem for robust plane fitting, which is used in the vertex projection step. In this section, we provide a more detailed description of how this optimization is carried out. We also present an ablation study demonstrating the effectiveness of our robust projection.

Solution to Eq. (8). The solution to this problem corresponds to the eigenvector with smallest eigenvalue of the covariance matrix

$$\mathbf{C}_j = \frac{1}{4} \sum_{i \in \mathcal{T}_j} \Delta\mathbf{s}'_i \, {\Delta\mathbf{s}'}^{\top}_i . \tag{1}$$

and define the centered coordinates $\Delta\mathbf{s}'_i = \mathbf{s}'_i - \bar{\mathbf{s}}'_j$. From these, we compute the covariance matrix $\mathbf{C}_j$ of the projected sites:

$$\mathbf{C}_j = \frac{1}{4} \sum_{i \in \mathcal{T}_j} \Delta\mathbf{s}'_i \, {\Delta\mathbf{s}'}^{\top}_i . \tag{2}$$

The plane normal $\mathbf{n}_j$ is given by the eigenvector corresponding to the smallest eigenvalue of $\mathbf{C}_j$, normalized as $\mathbf{n}_j = \mathbf{e}_{\min} / \|\mathbf{e}_{\min}\|$, where $\mathbf{e}_{\min}$ is the eigenvector of $\mathbf{C}_j$ with the smallest eigenvalue.

Proof. To avoid the explicit normalization of $\mathbf{n}_j$ we first reformulate the problem as

$$\arg\min_{\mathbf{n}_j} \frac{1}{4} \sum_{i \in \mathcal{T}_j} \left(\mathbf{n}_j \cdot \Delta\mathbf{s}'_i\right)^2, \text{ subject to } \|\mathbf{n}_j\|^2 = 1. \tag{3}$$

We then model the added constraint with a Lagrangian formulation:

$$\begin{aligned} \mathcal{L}(\mathbf{n}_j, \lambda) &= \frac{1}{4} \sum_{i \in \mathcal{T}_j} \left(\mathbf{n}_j \cdot \Delta\mathbf{s}'_i\right)^2 - \lambda\left(\|\mathbf{n}_j\|^2 - 1\right) \\ &= \frac{1}{4} \sum_{i \in \mathcal{T}_j} \left(\mathbf{n}_j^{\top} \Delta\mathbf{s}'_i\right)\left({\Delta\mathbf{s}'}^{\top}_i \mathbf{n}_j\right) - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j - 1\right) \\ &= \mathbf{n}_j^{\top}\left(\frac{1}{4} \sum_{i \in \mathcal{T}_j} \Delta\mathbf{s}'_i \, {\Delta\mathbf{s}'}^{\top}_i\right)\mathbf{n}_j - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j - 1\right) \\ &= \mathbf{n}_j^{\top}\mathbf{C}_j\mathbf{n}_j - \lambda\left(\mathbf{n}_j^{\top}\mathbf{n}_j - 1\right). \end{aligned} \tag{4}$$

Deriving this equation by $\mathbf{n}_j$ and setting the result to 0 yields

$$\mathbf{C}_j\mathbf{n}_j = \lambda\mathbf{n}_j, \tag{5}$$

hence $\mathbf{n}_j$ is an eigenvector of $\mathbf{C}_j$ associated to eigenvalue λ. The average distance to this plane corresponds to $\mathbf{n}_j^{\top}\mathbf{C}_j\mathbf{n}_j = \lambda(\mathbf{n}_j^{\top}\mathbf{n}_j) = \lambda$. Therefore we minimize the problem by finding the eigenvector with smallest eigenvalue. ☐

Method	CD ↓	F1 ↑	NC ↑
Barycentric	3.524	0.675	0.920
Hybrid	2.418	0.686	0.926
Robust (Ours)	2.326	0.684	0.930

Table 1. Ablation study comparing our robust projection, hybrid and barycentric interpolation. Reported values are average errors on the Thingi32 dataset using our near sampling/upsampling approach with an unconverged SDF representation.

Ablation study of our robust projection. Figure 1 and Table 1 show the performance of our robust projection (c) compared to the barycentric (a) and hybrid (b) approaches, applied within our near sampling/upsampling setting. The barycentric method corresponds to projecting vertices using interpolated SDF and gradient values obtained from barycentric coordinates, followed by projection onto the zero-level set with **??**. The hybrid method applies this barycentric projection only when the vertex lies inside its tetrahedron; otherwise, it falls back to the robust projection. We observe that the barycentric method generates the largest errors, as it is not robust to instabilities arising from extrapolation. The hybrid method alleviates this issue and significantly reduces the error, but abnormal spatial gradients can still occur, leading to slight instabilities during optimization. Applying our robust projection consistently in all cases provides the most stable and accurate results.

2. Detailed derivation of ??

Proof. Define the centroid and mean SDF value of tetrahedron $\mathcal{T}_j$ as

$$\bar{\mathbf{s}}_j = \tfrac{1}{4}\sum_{i\in\mathcal{T}_j}\mathbf{s}_i, \quad \bar{\phi}_j = \tfrac{1}{4}\sum_{i\in\mathcal{T}_j}\phi_i. \tag{6}$$

We minimize the following problem

$$\min_{\nabla\phi^{(j)}} \frac{1}{4}\sum_{i\in\mathcal{T}_j}\left((\bar{\phi}_j + (\mathbf{s}_i-\bar{\mathbf{s}}_j)^\top\nabla\phi^{(j)}) - \phi_i\right)^2. \tag{7}$$

To formulate this problem in a more compact manner, we define the vectors of centered coordinates and SDF values relative to the centroid and mean:

$$\Delta\mathbf{S}_j = \begin{bmatrix}\mathbf{s}_A-\bar{\mathbf{s}}_j\\ \mathbf{s}_B-\bar{\mathbf{s}}_j\\ \mathbf{s}_C-\bar{\mathbf{s}}_j\\ \mathbf{s}_D-\bar{\mathbf{s}}_j\end{bmatrix}, \quad \Delta\Phi_j = \begin{bmatrix}\phi_A-\bar{\phi}_j\\ \phi_B-\bar{\phi}_j\\ \phi_C-\bar{\phi}_j\\ \phi_D-\bar{\phi}_j\end{bmatrix}. \tag{8}$$

The problem of Eq. (7) then becomes

$$\begin{aligned}&\arg\min_{\nabla\phi^{(j)}} \frac{1}{4}\left\|\Delta\mathbf{S}_j\nabla\phi^{(j)} - \Delta\Phi_j\right\|^2 = \\ &\arg\min_{\nabla\phi^{(j)}} \frac{1}{4}\left((\nabla\phi^{(j)})^\top\mathbf{G}_j\nabla\phi^{(j)} - 2(\nabla\phi^{(j)})^\top\Delta\mathbf{S}_j^\top\Delta\Phi_j\right)\end{aligned} \tag{9}$$

where $\mathbf{G}_j = \Delta\mathbf{S}_j^\top\Delta\mathbf{S}_j$ is the local Gram matrix. Deriving the above function with respect to $\nabla\phi^{(j)}$ and setting the result to 0 finally yields

$$\nabla\phi^{(j)} = \mathbf{G}_j^{-1}\Delta\mathbf{S}_j^\top\Delta\Phi_j = \mathbf{W}_j\Delta\Phi_j. \tag{10}$$

□

3. Higher target point cloud density

Figure 4 compares MTet [?] and our method under different target point cloud densities with an unconverged SDF. Both methods benefit from increased point density in the target point cloud. However, our method with near sampling/upsampling consistently achieves the best results compared to the MTet baseline.

4. Ablation of SDF regularization

As shown in **??**, SDF regularization is critical for removing small floating zero-crossing regions around the optimized mesh. While these artifacts are often small, they represent topological noise that degrades the cleanliness of the final reconstruction.

Quantitatively, this trade-off is captured in Tab. 2. We observe that applying mild SDF regularization ($\lambda_{\text{Eik}} = 0.02$) leads to a significant improvement in the Hausdorff Distance (HD), confirming the suppression of these spurious outliers. While this smoothing constraint results in a marginal decrease in surface precision—reflected by a slight rise in Chamfer Distance and lower F1 score compared to the unregularized baseline, it provides a necessary balance. It ensures the resulting SDF remains topologically consistent without the severe degradation seen at high regularization levels ($\lambda_{\text{Eik}} = 2.00$).

Method	CD ↓	F1 ↑	NC ↑	HD ↓
$\lambda_{\text{Eik}} = 0.00,\ \lambda_{\text{H}} = 0.00$	2.241	0.690	0.929	464.3
$\lambda_{\text{Eik}} = 0.02,\ \lambda_{\text{H}} = 0.10$	2.276	0.680	0.930	369.8
$\lambda_{\text{Eik}} = 2.00,\ \lambda_{\text{H}} = 10.0$	5.940	0.634	0.904	499.7

Table 2. Ablation study on SDF regularization. We report average performance on the Thingi32 dataset for different values of the regularization weight λ_{sdf}.

5. Additional information & results

To ensure a rigorous and fair comparison, both MTet and DCCVT were initialized with the same initial sites and optimized using the Adam optimizer with the same hyperparameters ($LR = 5\times10^{-4}$, $\beta = (0.8, 0.99)$) for 1000 iterations. A key distinction lies in the optimization process: while MTet is driven purely by Chamfer distance of extracted vertices, DCCVT incorporates CVT regularization to maintain discretization stability. Our experiments show that although both methods yield comparable results on regular grids, DCCVT exhibits superior robustness when initialized with non-uniform site distributions, such as near-sampling, where MTet tends to struggle. This robustness indicates that DCCVT's performance gains are intrinsic to its mathematical formulation rather than specific initialization advantages.

Figure 5 shows additional results of our method with a converged SDF. For the Voromesh results, we applied a very small CVT regularization term to prevent co-planarity during mesh extraction. This increases the runtime of Voromesh, since a Delaunay triangulation must be recomputed at every iteration of the optimization process. We found this term to be necessary, particularly with unconverged SDF representations, where Voromesh frequently produces co-planar configurations in the final site distribution that are not supported by gDel3D [?]. As an alternative, CGAL could be used to perform the mesh extraction for Voromesh.

Fig. 6 presents additional comparisons between our method and MTet under both converged and unconverged SDFs in the best-case of near sampling and upsampling setting. The results highlight the adaptivity and surface regularization of our approach, which generates significantly improved surface discretization.

Figure 3 shows that while reconstruction quality is typically stable across SDF states, Mesh ID 252119 (Fig. 2) suffers a catastrophic failure in the converged state. This mirrors the issue in Mesh ID 398259 (**??**): the fully converged SDF provides a poor initialization that traps the optimization in a local minimum. The coarser, unconverged SDF avoids this pitfall, demonstrating that a less refined initialization can sometimes offer better regularization for the optimization loop.

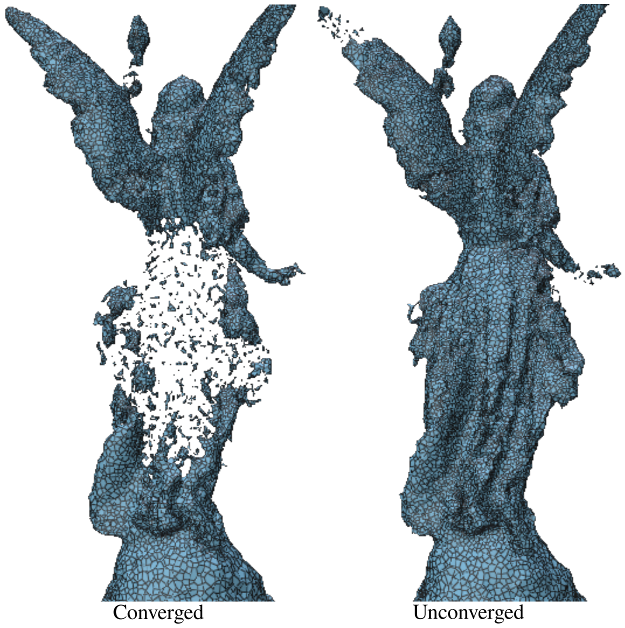

Figure 2. Hotspot converged produces a SDF that is very hard to recover from similarly to Mesh ID 398259 featured in **??**

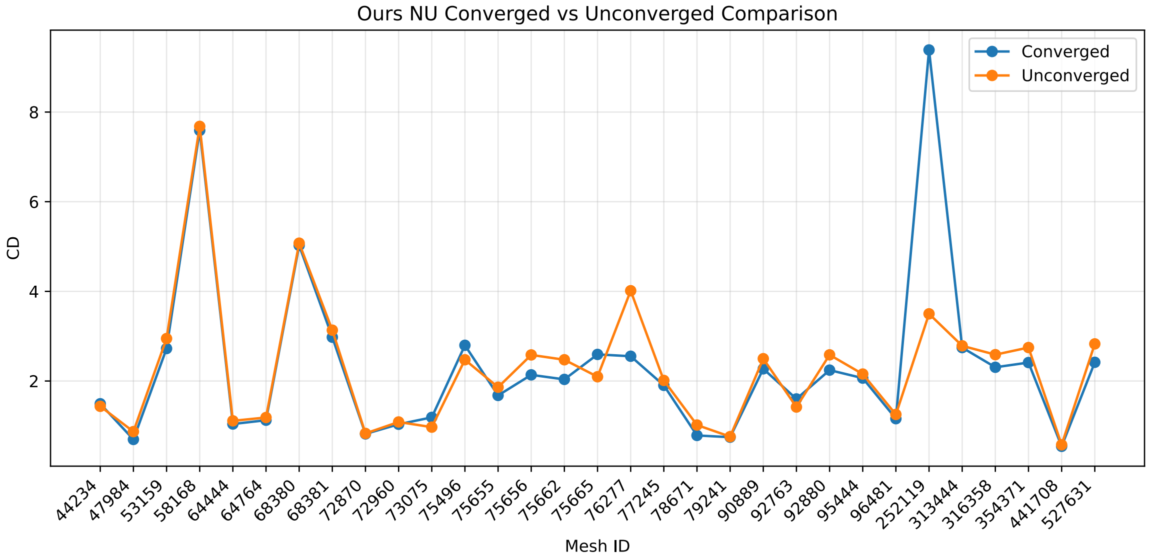

Figure 3. Per mesh Chamfer distance metric comparison between converged and unconverged Hotspot SDF state

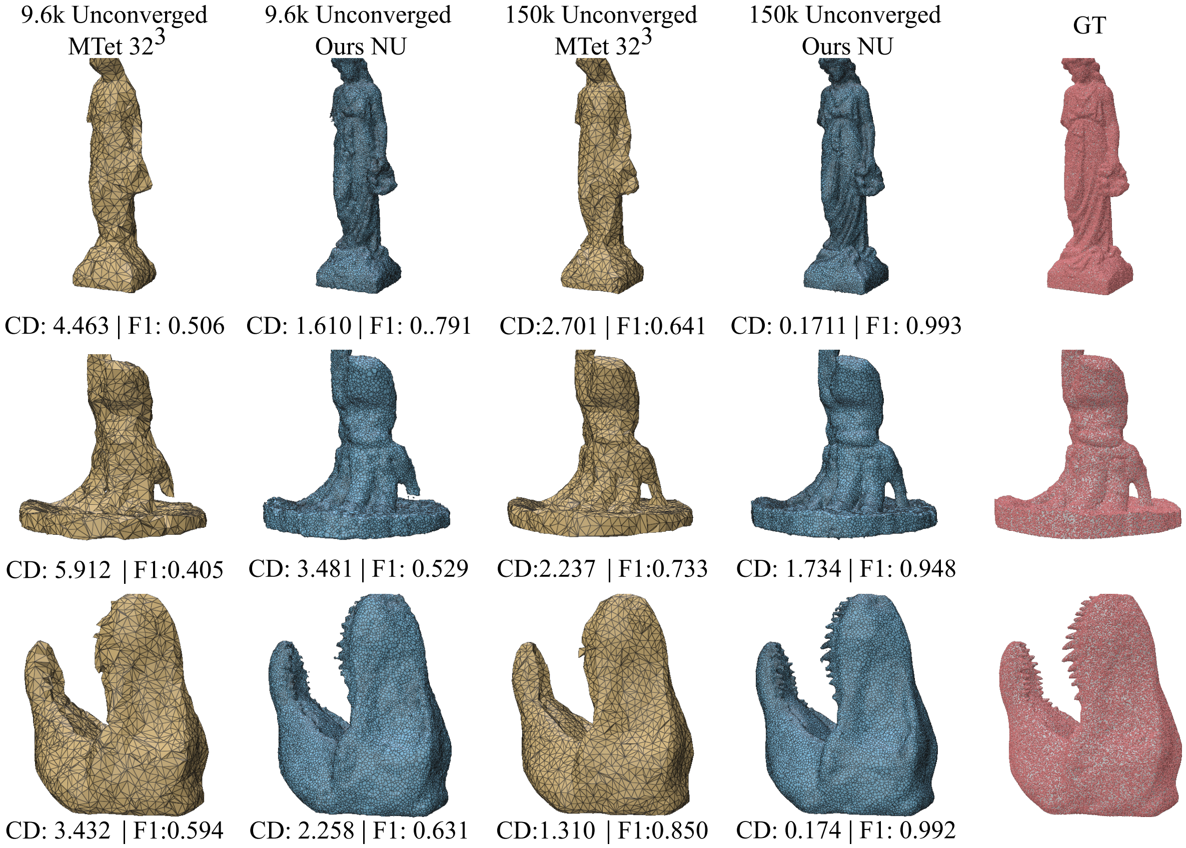

Figure 4. Comparison of MTet and our method (with near-sampling and upsampling, NU) under different target point cloud densities. Results are shown for $9.6k$ and $150k$ target points with an unconverged SDF. The ground truth (GT) is shown with $150k$ sampled points.

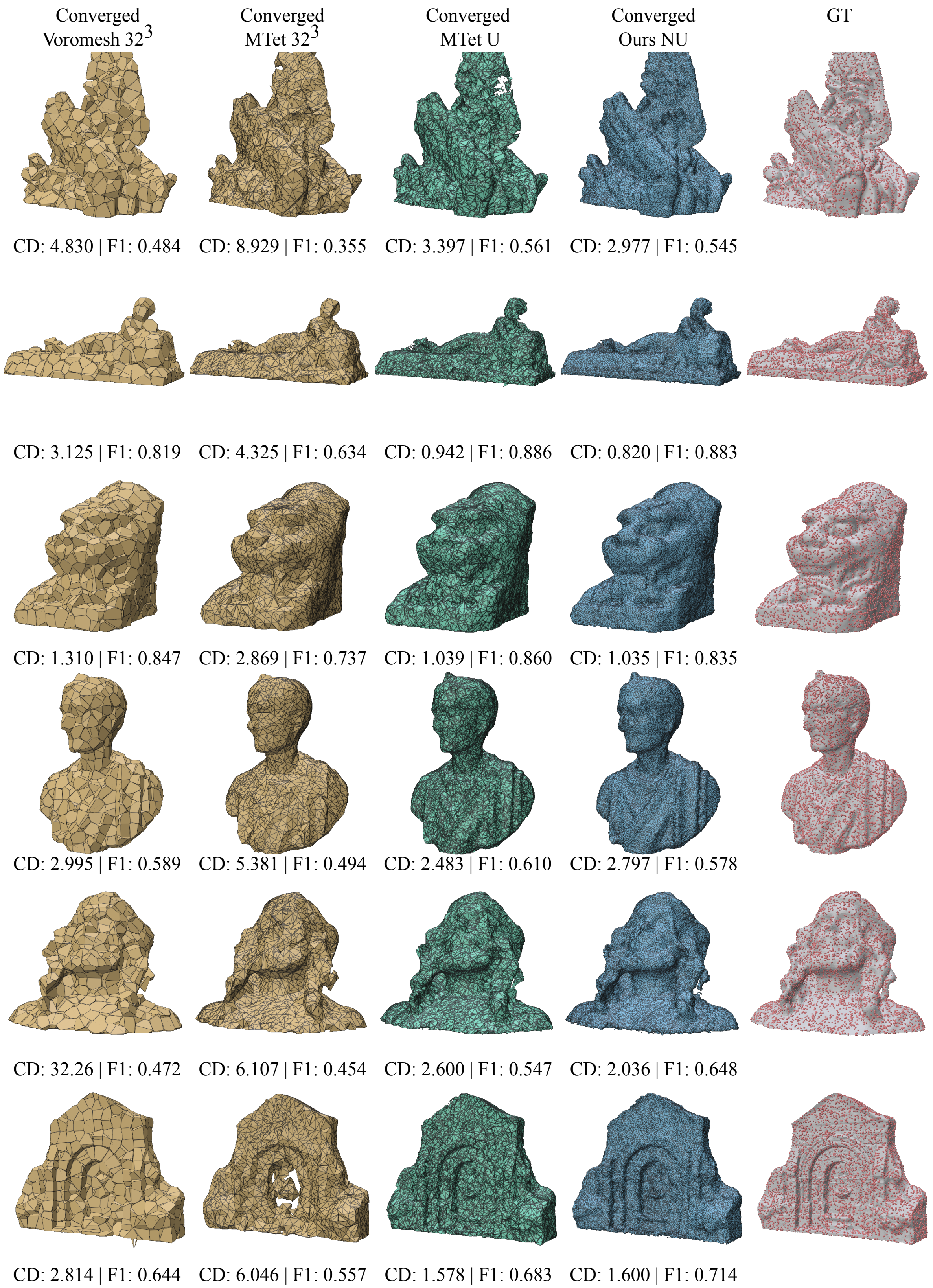

Figure 5. Comparison of Voromesh, MTet (with and without our upsampling), and our method with our near-sampling/upsampling) on converged SDFs. The ground truth (GT) is shown in the last column with 9600 sampled points.

Figure 6. Comparison of MTet (with our upsampling) against ours (with our near-sampling/upsampling) in both converged and unconverged SDF case.