WIR3D: Visually-Informed and Geometry-Aware 3D Shape Abstraction

Supplementary Material

A. Data Preprocessing

We leverage the input surface not just in our SDF regularization, but also in generating supervision data specialized for our task. Specifically, we generate stylized Freestyle renders which isolate the key geometric features of a shape for our stage I optimization. In the case where a user does not supply keypoints, we leverage the priors of 2D foundation models to automatically detect keypoints which correspond to salient shape features.

Freestyle Rendering Standard opaque surface renders are not ideal for our curve representation, which are non-occlusive by construction. Optimizing with these surface renders can result in under-detailed abstractions or particular Janusing artifacts [54], where curves positioned on the opposite side of the viewed surface end up being optimized for the wrong side. Furthermore, when the shape is untextured, surface renders may be poor at exhibiting key geometric structures.

To resolve this, we render the shapes in a stylized fashion to allow for each view to isolate the shape geometric structure and take into account the occluded shape features. Specifically, we render the shapes using the Freestyle rendering engine [11] in Blender and render the shape in terms of its view-dependent contours, *without* accounting for occlusions. These Freestyle renders are purely based on the shape geometry and do not take into account any textures. Thus, these renders are appropriate for the first stage of our optimization (Sec. 3.3) where we focus on capturing the shape geometry.

Keypoint Detection. When keypoints are not included in the input, we automatically identify keypoints of interest on the shape's surface using the 3D feature extraction method developed in Backto3D [66]. This method back-projects 2D image features to a 3D shape using a simple averaging scheme. Our assumption, following Backto3D, is that these backprojected features contain meaningful information of the shape's salient visual features, and thus can be leveraged for identifying keypoints relevant to those features.

Specifically, we render views of the 3D model and encode them using CLIP RN50x16, back-project the pixel-level latent features to shape vertices, and average the features among duplicate vertices captured in different views.

Once we have 3D features on the shape, we apply KMeans clustering over these features [35] and obtain k latent clusters, where k is the number of keypoints we wish to obtain. We interpret these clusters as aggregating surface points with similar visual content. We identify the vertex whose features are closest to the cluster centers as keypoints, since these vertices are most likely to represent the key visual feature associated with the cluster. We make k the number of curves we initialize in stage 2 of our optimization, though this number can be adjusted depending the number of salient elements on the shape.

Figure 12. **NEF qualitative comparison.** We show NEF results on the same models we compare to 3Doodle in the main paper. NEF is specialized for simple manufactured CAD shapes, so it struggles to fit edges to more complex surfaces. This limitation was similarly observed in 3Doodle.

B. Neural Edge Field Comparison

We show a qualitative comparison to NEF in Fig. 12, using the same models we show in the main paper for 3Doodle, except for the models which NEF fails to produce meaningful point clouds for. Though NEF can capture the rough silhouette of the target shape, the method is specialized for simple manufactured surfaces with sharp corners, so it struggles to place curves meaningfully on more complex surfaces. This results in a messier and harder-to-identify abstraction.

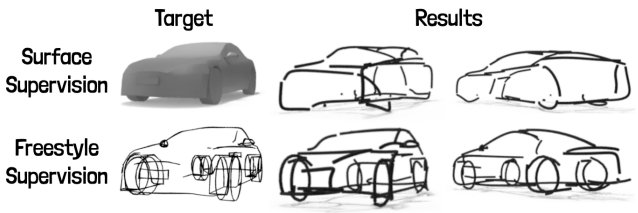

Figure 13. **Freestyle render ablation.** Running our method without freestyle renders still produces a reasonable abstraction, but key geometric features, such as the wheels of the car, may be missed due to the lack of visual signal from the surface renders.

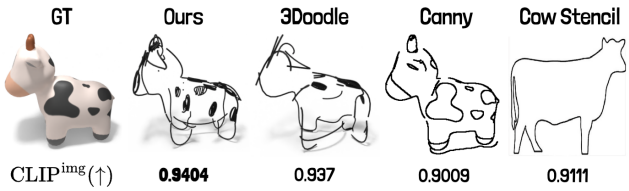

Figure 14. **Perceptual metrics reliability.** We show the unreliability of CLIP$^{\text{img}}$ in evaluating semantic similarity of curve abstraction to a target. We show for a given view, our stroke abstraction, 3Doodle's, the edge map for the view extracted using Canny edge detection [3], and a random image of a cow stencil obtained from Google. Note that though the Canny edge map captures the entire geometric structure and textures of the shape, its CLIP$^{\text{img}}$ score is shockingly lower than that of the stencil image. The vast difference in the two images also demonstrates how small differences in score can indicate major differences in quality.

C. Perceptual Metric Details.

The LPIPS metric is based on an AlexNet architecture trained for image classification fine-tuned with a linear layer on an annotated perceptual similarity dataset. Notably, we use the VGG variant of LPIPS for optimization, which is a commonly performed split between optimization and evaluation, and is similarly done in 3Doodle.

CLIP$^{\text{img}}$ is computed by encoding both the stroke renders and shape renders through a CLIP ViT/B-32 model, computing the cosine similarity, and scaling the score to [0-1]. Note that we only use the ResNet variants of CLIP for our optimization.

D. Ablations

No intermediate CLIP layers. As established in [61], the intermediate CLIP layers are essential for capturing the geometric structure of the target. Optimizing on only the fully-connected CLIP output results in abstractions that have some semantic correspondence with the target but the specific geometric features are noisy (Fig. 18).

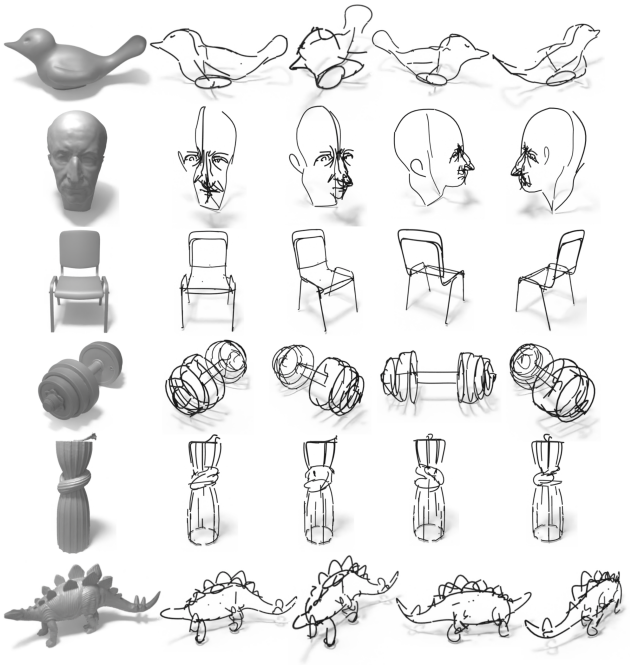

Figure 15. **Qualitative results for untextured shapes.** We show the result of our method on a collection of untextured meshes. Our method is effective and robust on a wide collection of different geometrie, such as the spiral column band (Row 6), the parallel rows of spines on the stegosaurus (Row 5).

No Stage 1. Stage 1 is essential for obtaining abstractions with visual volume. Without it the abstractions are flattened, so look reasonable from certain angles but not in others (Fig. 17).

Freestyle renders. We ablate on Freestyle render supervision in Fig. 13, instead running our method using opaque surface renders. The resulting abstraction is reasonable, but misses important geometric feature detail in the wheels and side mirrors of the car.

SDF loss. We ablate on the SDF loss in Fig. 19. The SDF loss prevents texture features from floating off the surface implied by the rest of the strokes, such as the spots on Bob circled in red.

E. Additional Applications

Detail Refinement. We show an additional example of keypoint-based abstraction refinement in Fig. 20. For our refinement application, we freeze the existing curve set and optimize 6 new curves randomly initialized in a local Gaussian around each keypoint. We use the same losses as the main method and only sample views where the keypoints are visible, and optimize for 100 iterations.

Figure 16. **Multi-view fidelity.** WIR3D adheres to the abstracted object in a 3D-consistent manner such that its properties can be perceived from every viewing angle.

Curve-Based Shape Deformation. Our deformation application exploits the close correspondence between the optimized curves and key visual features on the input surface, thanks to the SDF and keypoint localization losses. We develop a simple skinning system for the surface where each vertex is assigned a set of skinning weights to points sampled on all the curves in the scene. These skinning weights are based on the L2 distance between each vertex and sampled point, and a softmax is applied to ensure they sum to 1. Transformations to each curve can then be automatically mapped to the surface through these skinning weights, and

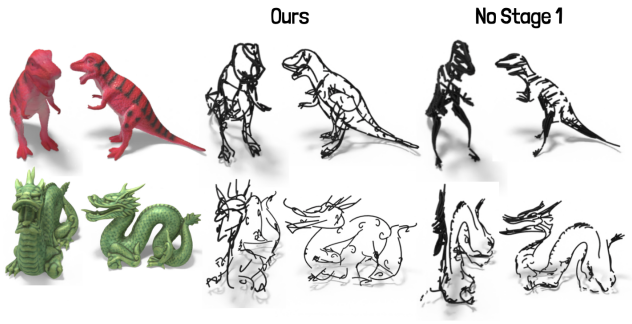

Figure 17. **Stage 1 ablation.** Stage 1 optimization is essential for capturing the full extent of the input geometry. Without it, the optimization tends to bias towards certain views, while the overall abstraction experiences a "flattening" effect.

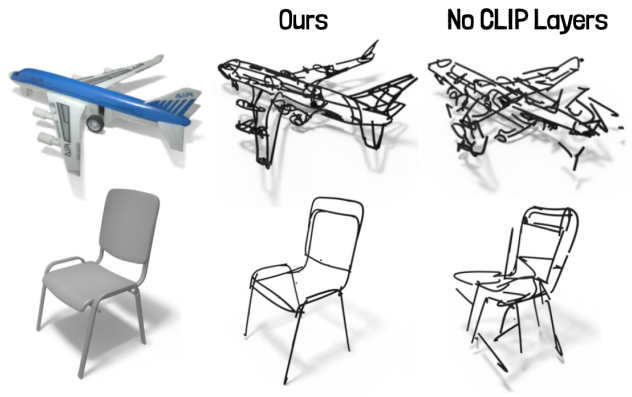

Figure 18. **CLIP layers ablation.** Supervising with the intermediate activations of CLIP is critical for maintaining coherent geometry. Using only the fully-connected CLIP output results in rough semantic abstraction, but the input shape geometric features are not well-preserved.

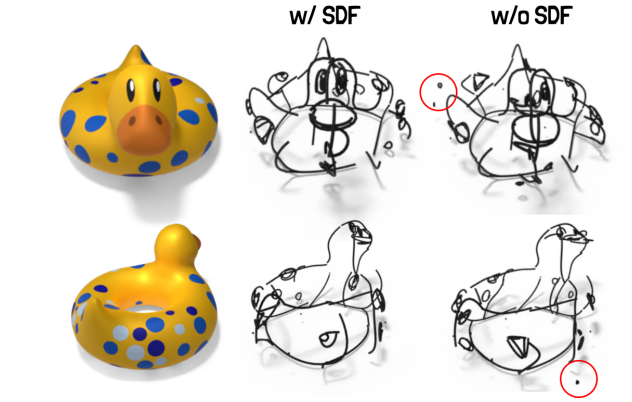

Figure 19. **SDF ablation.** The SDF loss helps to ensure abstracted visual features will stay anchored to the surface implied by the strokes. Without it, some features may hover outside the surface, such as the smaller spots on Bob.

Figure 20. **Texture keypoint control.** We expand on the keypoint control example shown in the main paper with a textured example. We show how by selecting keypoints on the texture on the plane, we are able to refine the abstraction by incorporating those texture elements.

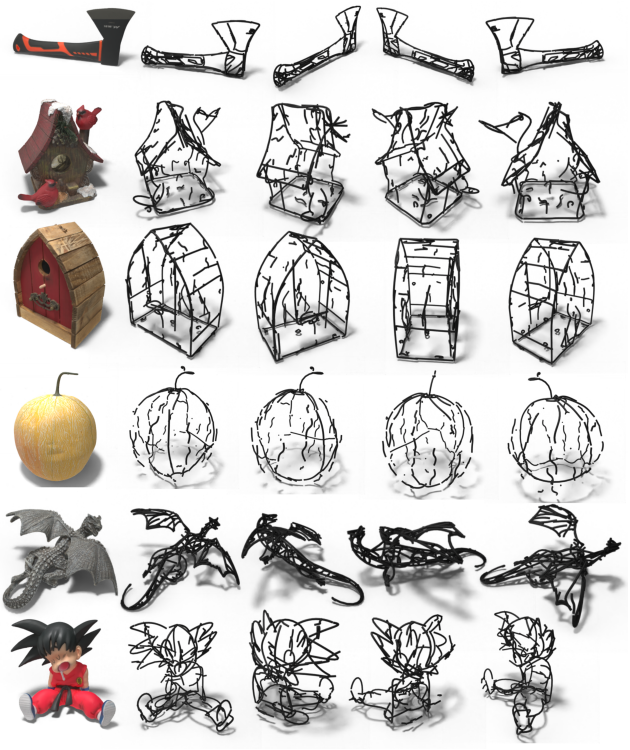

Figure 21. **Additional textured results.** We show additional textured results from the Meta DTC dataset [41].

the procedure can be performed at interactive speeds. We implement this deformation system as a proof-of-concept script, and show videos of the working system in the supplemental material, with screenshots displayed in Fig. 10. Note that *no smoothing postprocesses* are applied to the mapped transformations, and the smoothness of the deformations are a result of the effectiveness of the curves in interpolating the quantities along the surface.

F. Additional Abstraction Results

Texture Abstractions. Additional results on textured shapes are shown in Fig. 21. Our method is robust to many different types of models ranging from manufactured shapes with sharp edges to organic shapes with complex curvature.

Scene Abstraction. We show an example of our method run on a large scene in Fig. 22. Our method is able to reproduce the global scene layout, and successfully abstracts objects at different scales in the scene (e.g. house, trees, animals).

Multi-View Fidelity. Our curves are defined in 3D, so our abstraction is view-consistent by construction. However, this does not guarantee the curves plausibly represent the shape from arbitrary views. Fig. 16 shows that our abstraction faithfully represents the shape for densely sampled views in a 360 range.

G. Optimization Details.

For both stages, we optimize for 20000 iterations, sample 1 view per iteration, and use an ADAM optimizer with a learning rate of 1e-3. For CLIP supervision we sample 4 augmentations per view. In stage 1 of the optimization, we use the RN101 CLIP architecture, with $\lambda_{\text{fc}} = 0.1$. In stage 2, we use the RN50x16 architecture, $\lambda_{\text{lpips}} = 0.1$, $\lambda_{\text{fc}} = 75$, and $\sigma = 0.1$.

H. User Study Screenshots

We show screenshots from our user study in Fig. 23. The question order and the order of Wir3D versus 3Doodle assignment to "Sketch1/Sketch2" are randomized for each respondent. All results shown are rotating gifs, so that users can evaluate based on the full 360 views of the abstraction. At the beginning of the study, we present three examples of abstractions of different quality and explain the factors that determine their quality, so that users can make more precise judgments in their visual evaluation.

Figure 22. **Scene abstraction.** Our method extends to scene abstraction. Note how our method reproduces the global scene layout and captures all the objects in the scene despite the large scale differences.

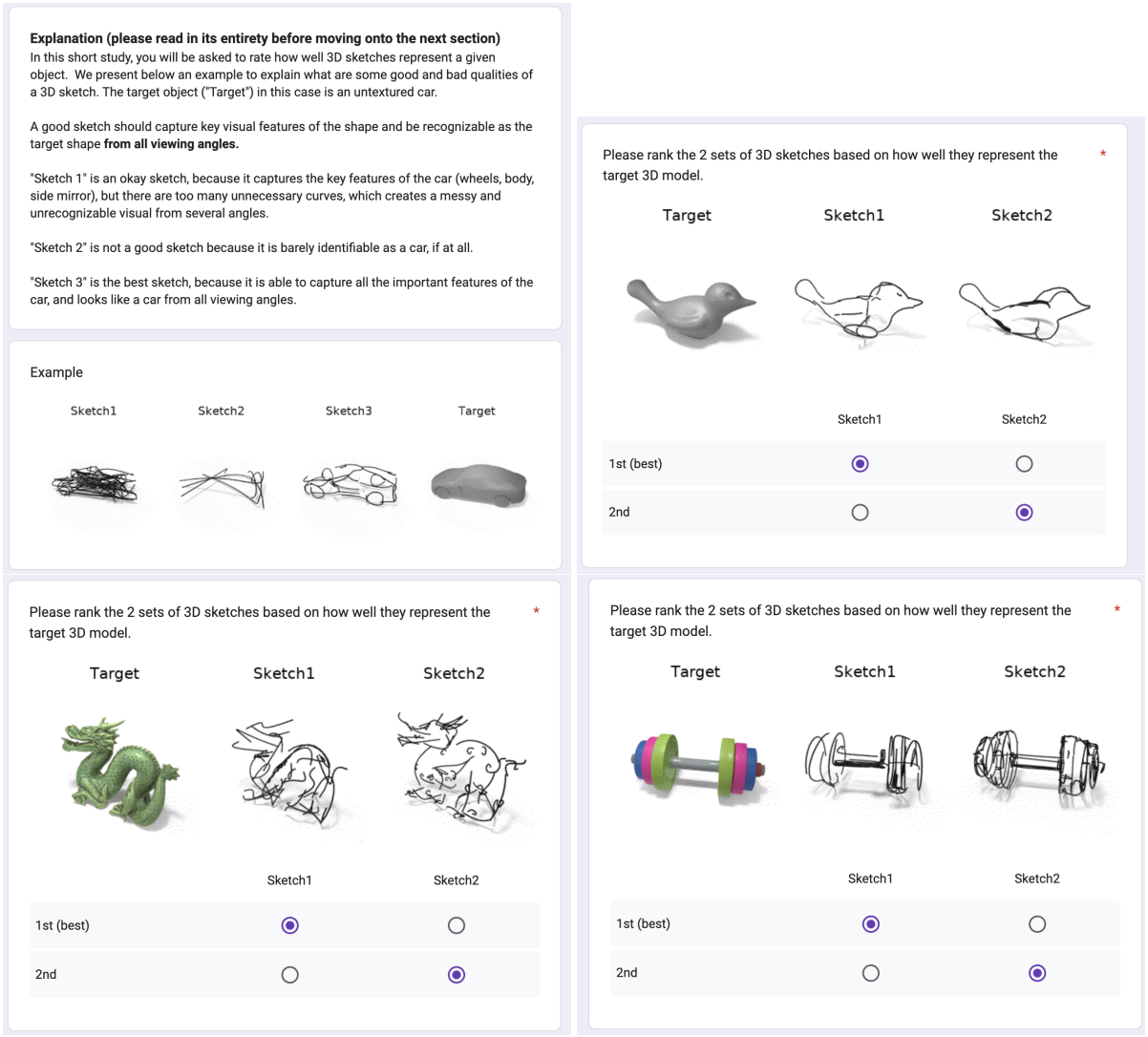

Figure 23. **Perceptual Study Screenshots.** Screenshots from our perceptual study. The question order and the order of Wir3D versus 3Doodle assignment to "Sketch1/Sketch2" are randomized for each respondent.

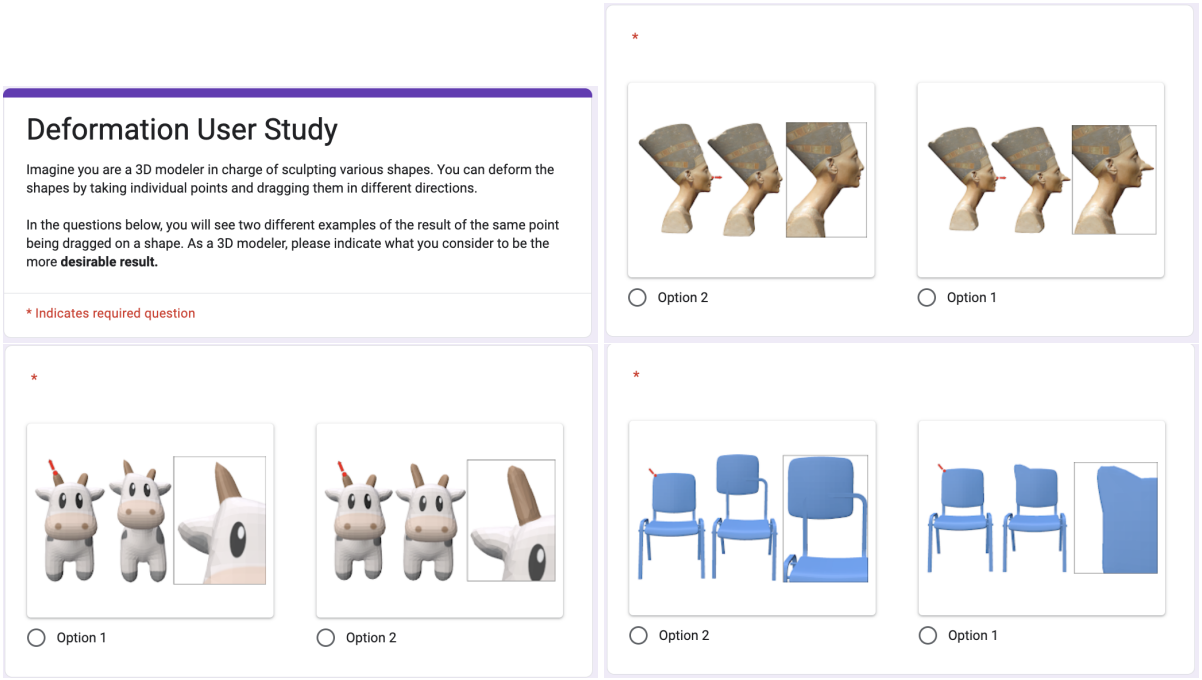

Figure 24. **Deformation Application User Study.** Screenshots from our user study comparing our deformation application using WIR3D curves as handles against ARAP [53] The question order and the order of Wir3D versus 3Doodle assignment to "Option 1/Option 2" are randomized for each respondent.

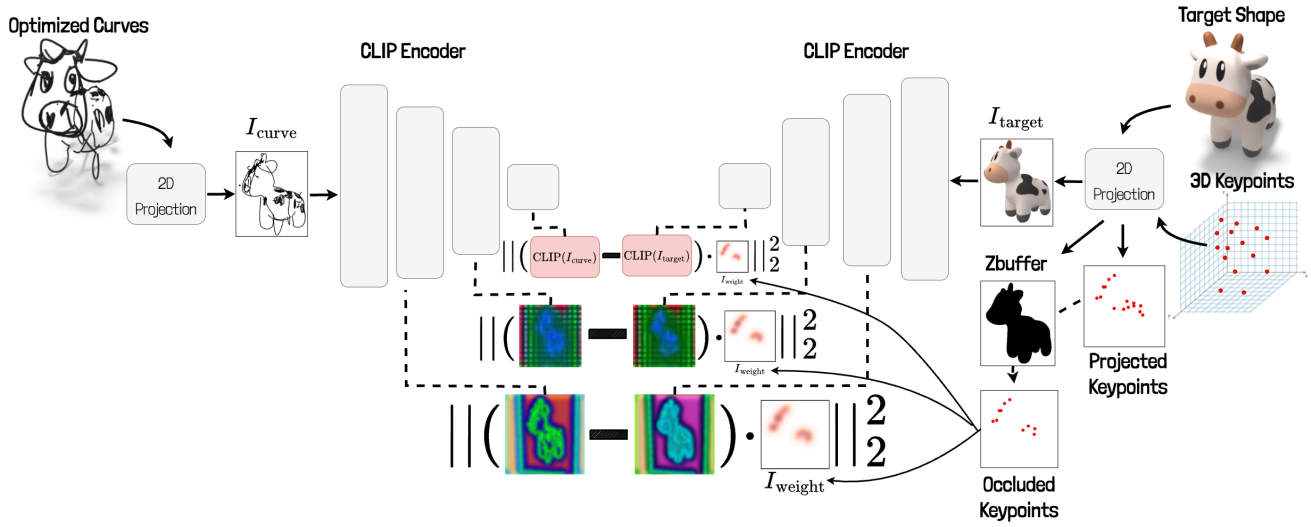

Figure 25. **Comprehensive Localized Keypoint Loss.** We show a comprehensive visualization of the localized keypoint loss.