Supplemental Document for “Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling”

1 Training data generation

Our training dataset is automatically generated through non-photo-realistic rendering (NPR) of 3D models [Rusinkiewicz and DeCarlo, 2013], which makes it possible to build a very large and diverse dataset with ground truth signals.

Each sample of the training data contains a pair of input data and ground truth output data. The input consists of the line sketch rendered by NPR, silhouette mask, optional depth samples and curvature hints. The output data contains the projected surface curvature directions for training DFNet, and the depth and normal maps of the rendered surface for training GeomNet.

The overall guidelines for data generation are:

1. The input sketch and output geometry should have matching complexity. This reduces the amount of ambiguity from sparse 2D sketches to 3D shapes.

2. The observed surface patch and rendered sketch should be in forms human users are likely to model. This avoids extreme surfaces and sketches that are simply unreasonable for sketch-based modeling, and enhances the usability of trained CNN network.

Below we give details about what 3D models we use, how we process them for NPR rendering, how curvature directions are built, what viewpoints are selected, how the sketch is generated with proper level-of-detail, and how optional depth sample points and curvature hints are generated.

1.1 3D Models

We collected 70 training and 20 testing 3D models (Fig. 1) from commonly used databases, including Aim@Shape [aim,], Princeton Segmentation Benchmark [Chen et al., 2009] and Stanford 3D Scanning Repository [sta,]. The models are mostly freeform smooth shapes with moderate complexity which we aim to model, instead of objects with many details, or highly structured and symmetric CAD models.

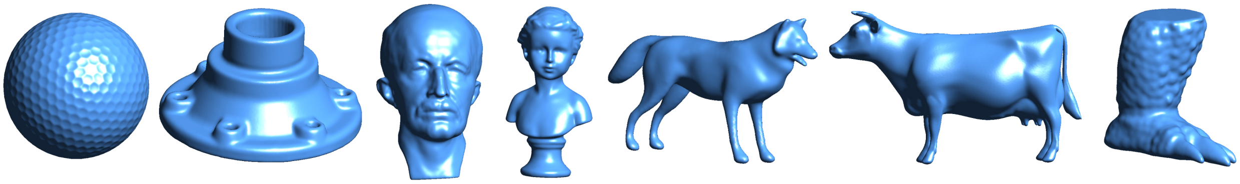

Figure 1: Sample 3D models used for training data generation.

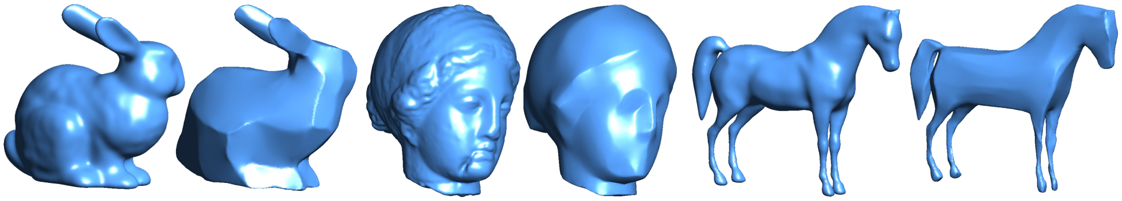

Figure 2: Shapes with sharp creases are generated from smooth models by repetitively applying geometry texture removal filter.

The 3D models are remeshed and smoothed to have regular and high-quality surfaces, which is critical for generating plausible line sketches with NPR rendering that relies heavily on the estimation of high order differential quantities. Shapes with sharp features (Fig. 2) are conveniently created by iteratively applying normal filtering [Wang et al., 2015] on smooth 3D objects.

1.2 Curvature direction field

To train the DFNet which regresses curvature direction field from sketches, we must generate the projected curvature direction field of rendered surfaces. To do this, we first compute a smooth curvature aligned 4-direction field on the surface mesh of a 3D model [Vaxman et al., 2016], which is regard as the regularized curvature direction field. We then evaluate the projected directions on each pixel of the rendered image, which is done efficiently by a customized shader program. The projected four directions are no longer orthogonal, and thus are encoded by two complex numbers (see Sec. 3.2 in text).

1.3 Viewpoint selection

Each 3D model is normalized to have unit bounding sphere radius, and translated to the origin. Then we uniformly sample 50 viewpoints over the sphere of radius 1.5 at origin; for each viewpoint 4 up-directions with 90 degrees angle apart are generated for rotating the camera. For all these viewpoints, we compute the PCA axis of the observed surface patch. Sorted in increasing order according to the angle of PCA minor axis with the camera viewing axis, the top 10 view points for the model are selected. In addition, we zoom into the models with three different scales, 0.8, 1.0, 1.3. After a sketch is rendered, it is checked for coverage of whole image: if it is too big ($>95\%$ of whole image) or too small ($<5\%$ of whole image), we skip the sketch. This way, we make sure the surface is observed in a reasonable angle through which a user would likely draw.

1.4 Sketch rendering

Sketches are rendered with the program by [Rusinkiewicz and DeCarlo, 2013]. There are several thresholds controlling whether a nontrivial point on the surface is part of ridges/valleys, suggestive contours, apparent ridges, etc. We tune the parameters according to this rule: *the sketch rendered should allow a human observer to easily reconstruct the important shape variations mentally.*

To augment the sketches with different level-of-details, we first process the 3D models by smoothing them to generate two additional level-of-details, and then render sketches for the processed models. See Fig. 3 for example.

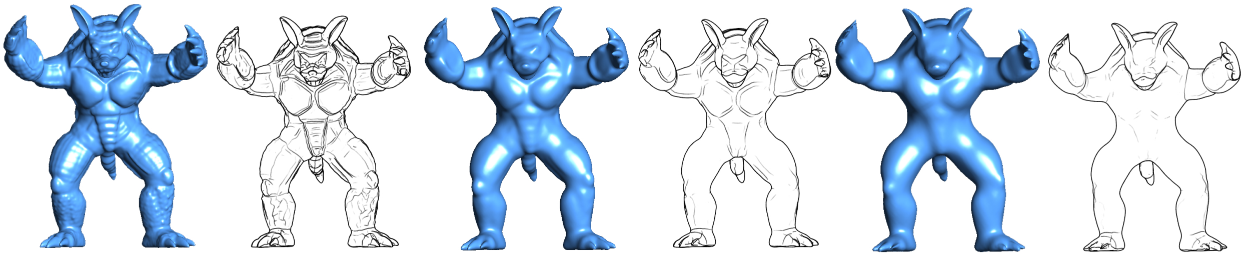

Figure 3: A smooth 3D model is smoothed into 2 additional level-of-details and augmented sketches are rendered from them.

1.5 Depth samples and curvature hints

Depth samples are primarily placed along portions of the silhouette curves to anchor the surfaces, which complies with the general modeling scenario where the shape contour is usually initialized to the drawing plane and tuned later. We sample random but continuous proportions of the contour curves from 6 levels, [1/4, 1/3, 1/2, 2/3, 3/4, 1] of whole length. Additional inner depth samples are generated inside the contour. They are placed over significant extreme depth values of the surface, to better bound the local variation. The number of inner depth samples is random but less than 5.

Curvature hints include specifying sharp features and target curvature values for strokes. To generate curvature hints data, we first detect ridge/valley curves and sharp features of the 3D models. We then compute the surface curvature values (Sec. 3.1) across these spatial curves. Input data, including sharp feature mask, mask and target values of curves with curvature hints, are then generated for a random subset of these detected curves.

Again, these pixel-wise data are efficiently computed by customized shader program.

2 User evaluation

We designed a small user study to evaluate our new sketch-based modeling tool. In this study the participants are asked to model shapes as similar as possible to given targets by interactively sketching lines. Besides archiving the user modeled shapes, we also collect feedbacks and comments by asking how the users assess the modelling tool.

2.1 Participants

We invite 5 users (3 male and 2 female) to take this study. The users have different backgrounds: (P1, graduate, Graphics), (P2, undergraduate, Data Mining), (P3, graduate, Graphics), (P4, graduate, Graphics), and (P5, researcher, Computer Vison). Among them P3 and P4 have some modeling experience, and none of them have sketch drawing training and experience. All the participants preformed the sketching tasks on a DELL 21.5" touch screen with a drawing pen.

2.2 Procedure

We divide the study into three parts: training, modeling and discussion. On average the whole process takes about 1 hour for one user.

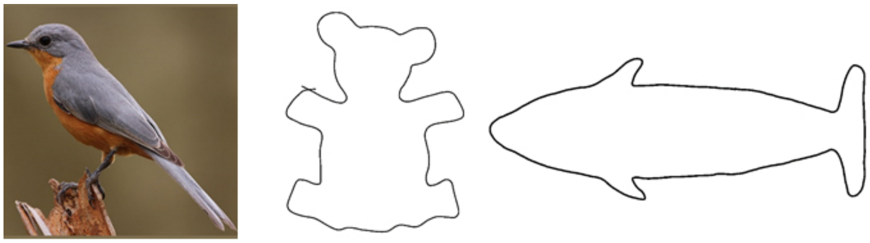

Figure 4: Reference images and contours for the target shapes.

Training session. The training session is designed to make users get familiar with the necessary background and our program interface. One of the authors shows them the style of lines used for sketching, some example sketches and correspondingly modeled shapes, and the usage of the UI. Users are encouraged to first try and draw some casual shapes to learn and explore the system. The average duration for this session is about 20 minutes.

Modeling session. In this session, the participants are asked to model three specific shapes, i.e. bird, teddy and dolphin. For bird model, we provide the user an RGB image as shown Fig. 4, and they are required to model a shape as similar as possible. For teddy and dolphin models, we use MeshLab to show user the target shape in another screen. We also capture the contour lines of teddy and dolphin for user, shown in Fig. 4. These images can be loaded into our interface for reference.

Discussion session. In this session the users provide comments and assessments about our system after finishing the modeling tasks. We designed three questions to encourage the assessment:

1. Do you think the strokes are intuitive?
2. When the first try does not match the target well, do you think the modifications are easy to conceive?
3. Is the multi-view modeling easy to use?

We collected and summarized their comments in the following section.

2.3 Results

Modeling session duration. We recorded the time used by each user for every modeling task. The average modeling time for bird is about 5 minutes, teddy 15 minutes and dolphin 20 minutes. Modeling shapes with more details are naturally more time consuming.

Results. All user input sketches and result shapes are shown in Figs. 5,6,7.

2.4 Summary and analysis

Based on the answers to the questions asked in the discussion session and the result shapes, we see that:

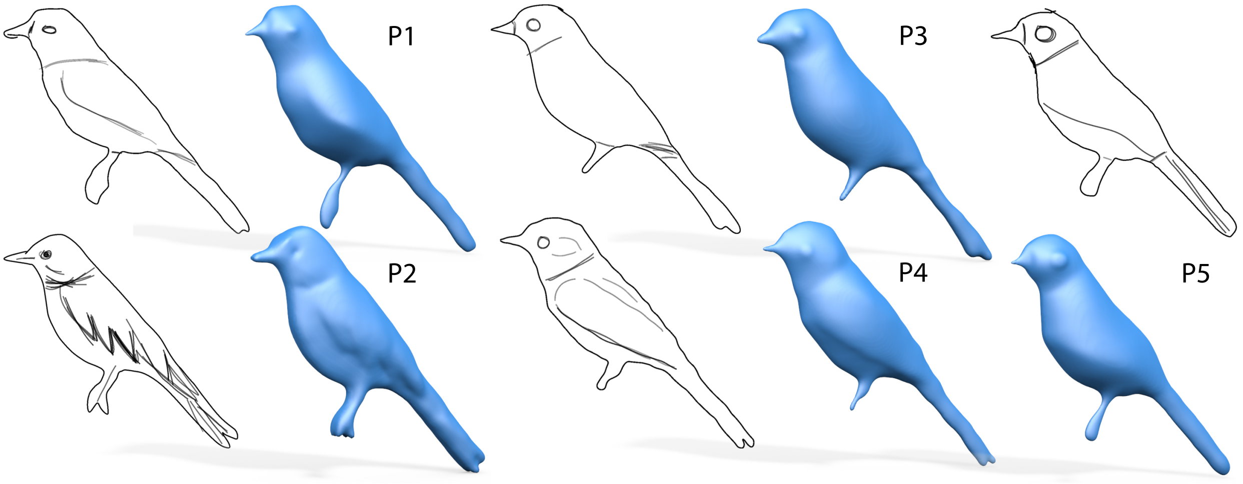

Figure 5: Result bird shapes created by all users. Users use one sketch for the front patch and make a symmetric back patch to model the complete shape.

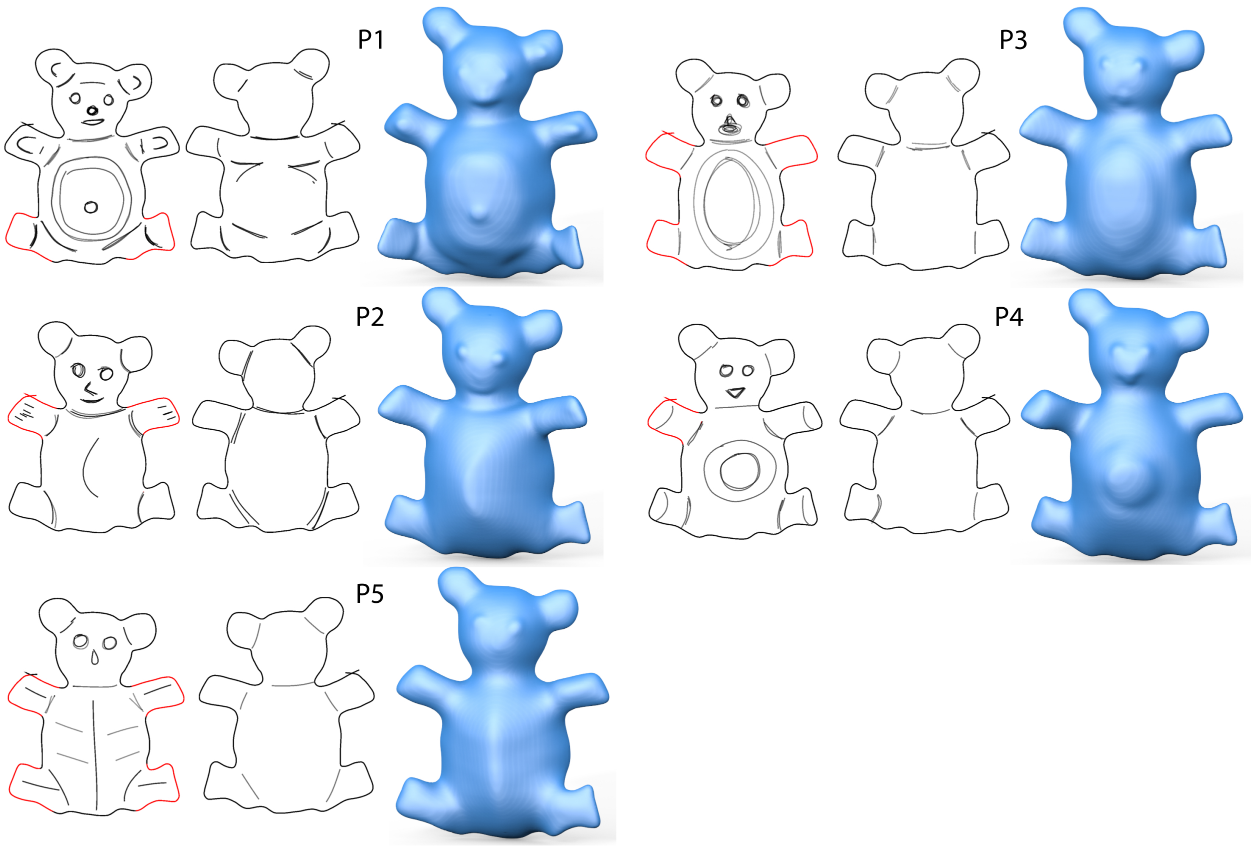

Figure 6: Result teddy shapes created by all users. Users use two sketches to model the complete shape. Part of front patch contours marked red are re-sketched by the users mostly to make limbs lifted from plane. Back patches copy contours of front ones.

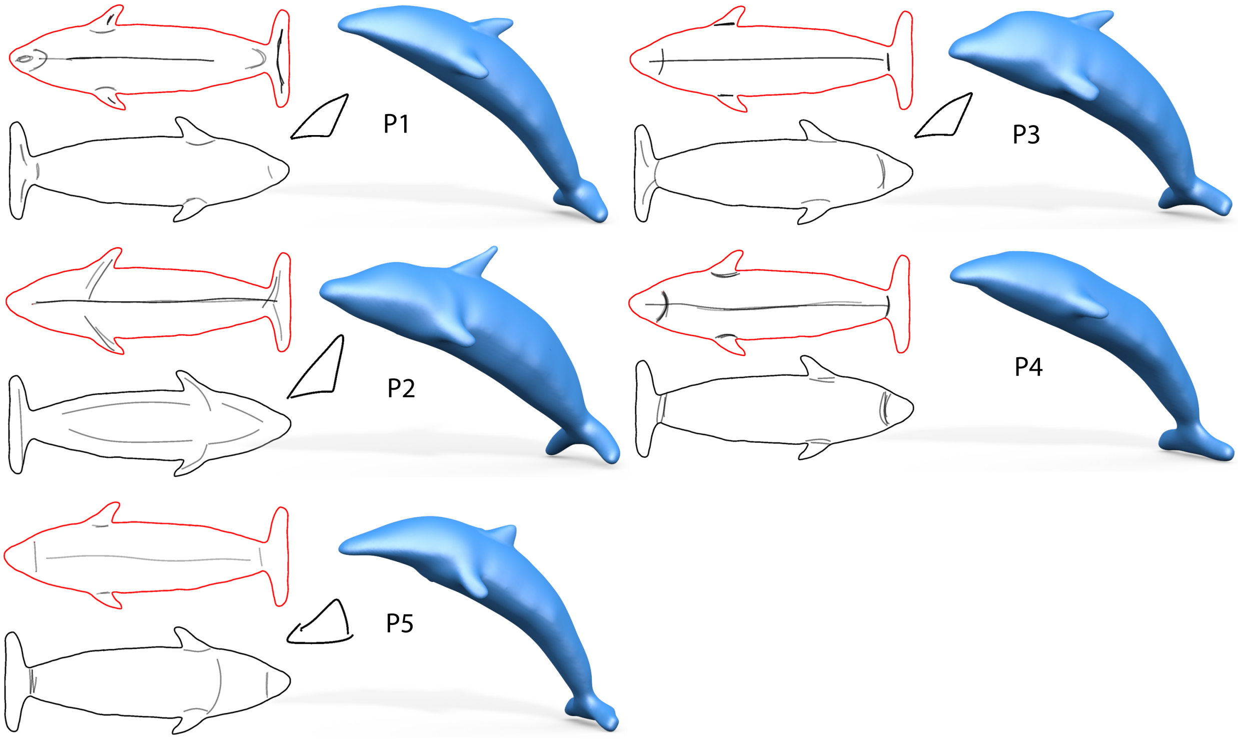

Figure 7: Result dolphin shapes created by all users. Most users use three sketches to model the complete shape. Top patch contours marked red are re-sketched by the users to bend the surface. Bottom patches copy contours of top ones. The triangular patches model the top fins.

1. Our input lines are intuitive to users without any drawing experience. The sketch lines are consistent with how users perceive 3D objects.

2. Regardless of prior modeling or drawing experience, after about 20 minutes training to get familiar with our modeling tool, any participant can sketch each of the testing target shapes. When the user wants more details longer modeling time is used, and users usually want to explore and add more details.

3. Our algorithm is robust in handling different input strokes and can generate similar results.

References

[aim,] AIM@SHAPE. http://visionair.ge.imati.cnr.it/ontologies/shapes/.

[sta,] The stanford 3d scanning repository. http://graphics.stanford.edu/data/3Dscanrep/.

[Chen et al., 2009] Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3d mesh segmentation. *ACM Trans. Graph. (SIGGRAPH)*, 28(3):73:1–73:12.

[Rusinkiewicz and DeCarlo, 2013] Rusinkiewicz, S. and DeCarlo, D. (2013). Real-time suggestive contours.

[Vaxman et al., 2016] Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommes, D., Hildebrandt, K., and Ben-Chen, M. (2016). Directional Field Synthesis, Design, and Processing. *Computer Graphics Forum.*

[Wang et al., 2015] Wang, P.-S., Fu, X.-M., Liu, Y., Tong, X., Liu, S.-L., and Guo, B. (2015). Rolling guidance normal filter for geometric processing. *ACM Trans. Graph. (SIGGRAPH ASIA)*, 34(6):173:1–173:9.