The 8th ACM/EG Expressive Symposium
C. Kaplan, A. Forbes, and S. DiVerdi (Editors)

Sketch-Based Modeling of Parametric Shapes

Bastien Wailly and Adrien Bousseau

Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur

1. Introduction

We demonstrate a sketch-based modeling system running on a multi-touch pen tablet. Our system takes inspiration from the work of Nishida et al. [NGDGA*16], who proposed to use deep convolutional networks to interpret sketches of parametric shapes. While Nishida et al. applied this approach to the creation of procedural buildings, we focus on the creation of simple shapes (cuboids, cylinders, cones, spheres, pyramids) that users can assemble to create more complex objects. In this poster we describe the main components of our system – the deep convolutional networks used for sketch interpretation, the training data, the user interface, and the overall software architecture that combines these components. We will allow conference attendees to test our system on a pen tablet.

2. Sketch Interpretation Networks

Following Nishida et al. [NGDGA*16], we rely on two types of deep convolutional networks for sketch interpretation. First, a *classification* network recognizes what shape is drawn. Then, a *parameter estimation* network predicts the position and dimensions of the recognized shape. There is one such network per shape category since different categories have different parameters. We use the same network architectures as Nishida et al. in our implementation.

3. Training Data

We train the deep networks with synthetic line drawings generated from random variations of the shapes supported by our system (Figure 1). We use the script-based 3D modeller *OpenSCAD*† to create a large number of random shapes, from which we extract silhouette, ridge and valley lines. We export these lines as SVG vector files, which facilitates their subsequent stylization. In particular, we add random noise, displacements and rotations to the lines to achieve a hand-drawn style similar to what users of our interface typically draw. Finally, we render the shadow that each shape casts on the ground plane when lit from above. We found that providing these shadows to the deep network allows it to disambiguate the elevation of the shape with respect to the ground. A similar use of cast shadows for sketch-based modeling has been proposed by Cohen et al. [CMZ*99], although we are the first to leverage this depth cue in a deep learning context.

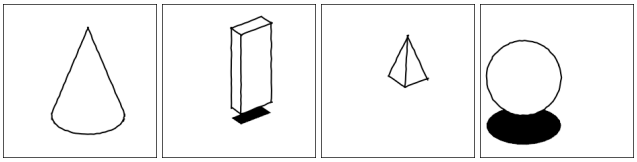

Figure 1: *A few samples of our synthetic training data.*

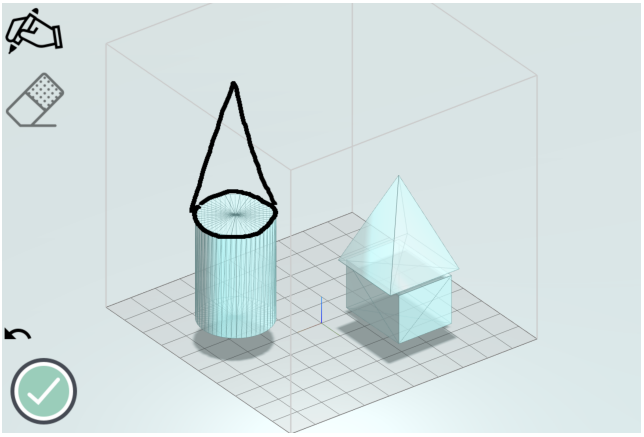

Figure 2: *Screenshot of our user interface.*

We render all drawings from a 3/4 bird's eye viewpoint relative to axis-aligned shapes. Non axis-aligned objects can be created by rotating the scene with respect to this canonical viewpoint.

4. User Interface

We implemented our system on a multi-touch pen tablet to exploit both finger-based and pen-based interaction. Fingers control camera motion (rotate, zoom, pan), while the pen is used to sketch. Finger taps are also used to click on the buttons of our interface to

† http://www.openscad.org/

select the eraser, undo the last sketch, or call the sketch interpretation engine on the current sketch. Finally, we also use a finger tap to select an existing object, which automatically elevates the ground plane to the highest point of this object, such that new shapes can be drawn on top of it. We facilitate perspective drawing of axis-aligned shapes by displaying a grid on this ground plane.

5. Software Architecture

We implemented our system according to a client-server architecture. The client runs in a web browser using HTML5, and communicates with a remote Node.js server using HTTP posts. The client sends sketches as bitmap images, while the server sends back shape parameters, which are converted to 3D meshes by the client for display. The server calls deep networks implemented using the Caffe Python library [JSD*14]. This architecture allows easy deployment and update of the system on pen tablets.

6. Conclusion

This poster and the accompanying demo will give the Expressive audience the opportunity to test a simple yet effective sketch-based modeling system, and learn about the design decisions that informed its implementation.

Acknowledgements

This work was supported by the ERC starting grant D3 (ERC-2016-STG 714221) and research and software donations from Adobe.

References

[CMZ*99] COHEN J. M., MARKOSIAN L., ZELEZNIK R. C., HUGHES J. F., BARZEL R.: An interface for sketching 3d curves. In *Proc. Symposium on Interactive 3D Graphics (I3D)* (1999). 1

[JSD*14] JIA Y., SHELHAMER E., DONAHUE J., KARAYEV S., LONG J., GIRSHICK R., GUADARRAMA S., DARRELL T.: Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), ACM. 2

[NGDGA*16] NISHIDA G., GARCIA-DORADO I., G. ALIAGA D., BENES B., BOUSSEAU A.: Interactive sketching of urban procedural models. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2016). 1