Monster Mash: A Single-View Approach to Casual 3D Modeling and Animation (Supplementary Material)

1 INTRODUCTION

In this supplementary document, we show additional comparisons with the current state-of-the-art in sketch-based modeling (Section 2), we also demonstrate how our framework handles more complex input sketches and how the user can control the amount of inflation locally (Section 3).

2 COMPARISON

Fig. 1c demonstrates unnatural bulges produced by Laplacian smoothing at the boundaries of individual components. A similar drawback is visible also in the results of Entem et al. [2015], where individual model parts are connected using integral surfaces (see Fig. 2c, upper row). Although in their framework strokes do not need to be added one by one, the resulting model cannot be directly animated. An option would be to apply a shape-preserving deformation like ARAP [Sorkine and Alexa 2007] on their mesh; nevertheless, in this case the intrinsic shape of the model would stay fixed during the deformation and would not adapt to a new pose; also unwanted penetrations can occur. In the method of Bessmeltsev et al. [2015], as well as in Entem et al. [2019], no explicit component-merging phase is used and thus all model parts fully penetrate by default, with no smooth blending between them (see Fig. 4e–f and Fig. 2c, middle & bottom row). Although, using Bessmeltsev et al.'s method, the model can be directly animated using the provided skeleton, it needs to be carefully planned and prepared in advance by the user using multiple viewpoints. Also, during animation, unwanted penetrations need to be avoided manually. In contrast to the method of Li at al. [2018], our approach provides better control over the resulting 3D structure without needing to leave the 2D domain and while also providing support for animation (see Fig. 5).

We compared the running times of our method with the bas-relief method of Dvorožňák et al. [2018] for the model of elephant and unicorn. We created the models with similar triangulation as in [Dvorožňák et al. 2018], i.e., approximately 22 k vertices and 36 k faces for the unicorn and 39 k vertices and 68 k faces for the elephant model. For the comparison we used a computer with a comparable performance. Our prototype implementation requires in average 8.6 seconds (unicorn) and 22.3 seconds (elephant) to converge to the satisfaction of all depth-ordering constraints while the method of Dvorožňák et al. [2018] converged in 3.8 seconds (unicorn) and 12.4 seconds (elephant) respectively. In our solution, the user can immediately start to animate the mesh even during the optimization process and thus does not need to wait for the convergence. For these high-quality triangulations, the full-fledged per-frame computation of the ARAP-L objective runs at 2.2 fps (unicorn) and 1.1 fps (elephant), and the interleaved processing runs at 8.8 fps (unicorn) and 4.6 fps (elephant).

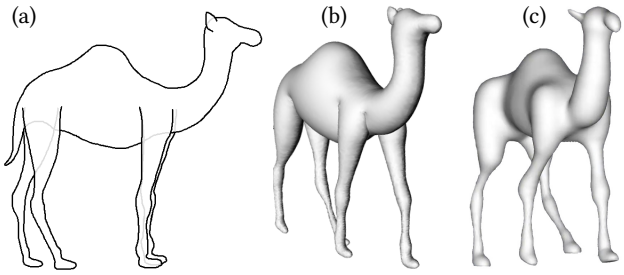

Fig. 1. Comparison with RigMesh [Borosán et al. 2012]: (a) input hand-drawn sketch; (b) side view of slightly deformed 3D model inflated using our approach; (c) 3D model in a similar pose produced by RigMesh. Note how our approach models the intended shape better, without any unnatural bulges caused by Laplacian smoothing on transitions between individual parts, as seen in the model produced by RigMesh. See also our supplementary video to compare the relative ease of interactive modeling using our approach versus RigMesh. Result (c) courtesy of © Borosán et al. (used with permission).

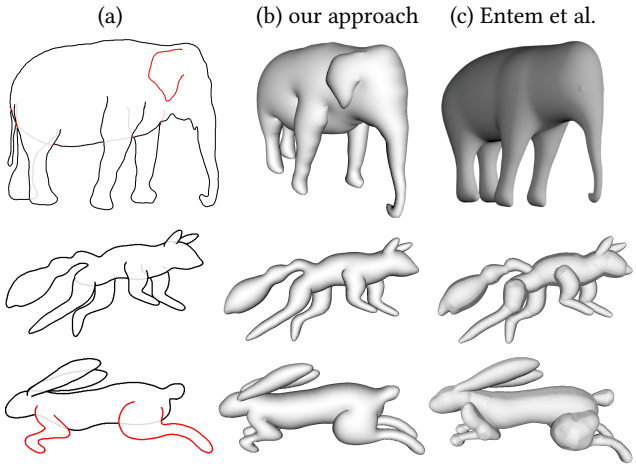

Fig. 2. Comparison with Entem et al. [2015; 2019]: the input sketch with draw occluded curves and relative depth ordering of individual segments (a) is used to produce the final mesh using our method (b). Although in the method of Entem et al. segmentation and relative depth ordering can be computed automatically, in the resulting mesh (c) transitions between individual components look unnatural and the model cannot be directly animated. Result (c) in the top row courtesy of © Elsevier (used with permission).

3 ADVANCED INPUT AND CONTROL

To make the interactive experience with our system more intuitive for novice users, we deliberately disabled advanced widgets in the

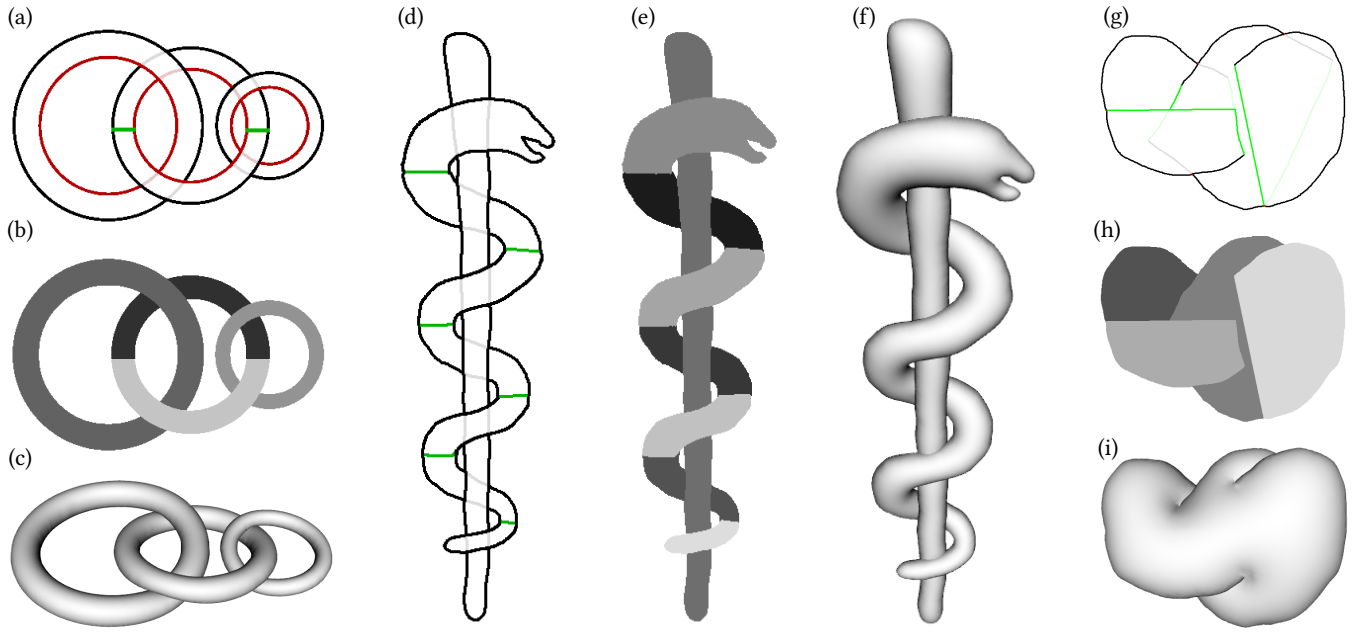

Fig. 3. Examples of meshes (c, f, i) with more complex topologies created using stroke labelling widget: red stroke in the input sketch (a) denotes a hole in a segment and green stroke (a, d, g) represents a connection boundary between two segments that can possibly have a different absolute depth order (b, e, h). The input sketch (g) depicting an object with cusps is based on an a similar example from Karpenko and Hughes [2006].

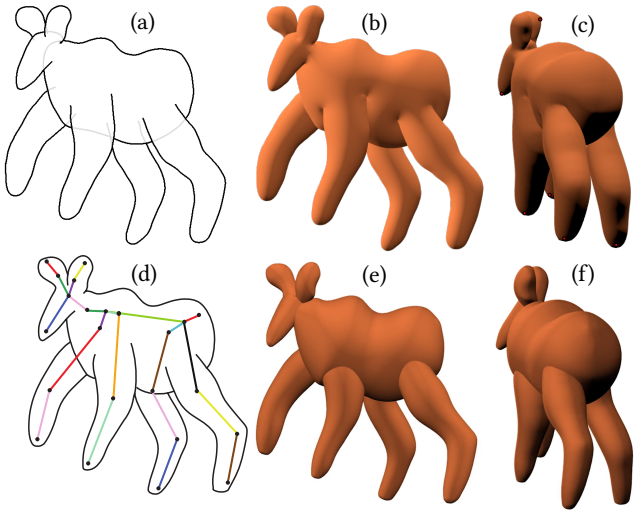

Fig. 4. Comparison with Bessmeltsev et al. [2015]: The original drawing together with the information about relative depth ordering of individual components (a) was used to perform inflation (b). The method of Bessmeltsev et al. requires the drawing to be augmented with a 3D skeleton (d) to perform inflation (e). By manipulating the skeleton's joint rotations the head can be turned left (c). In our approach, the head can be rotated as well using a single control point (f). Note how our approach produces smooth transitions between individual components whereas Bessmeltsev et al.'s method resorts to penetrations without blends. Images (d, e, f) courtesy of © Bessmeltsev et al. (used with permission).

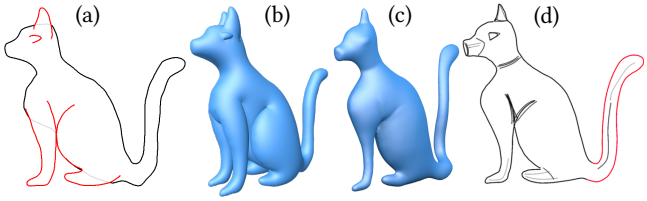

Fig. 5. Comparison with Li et al. [2018]: a hand-drawn sketch with an additional boundary-shifting constraint (red curve) (d) was used to as an input to the neural method of Li et al. to produce the final 3D shape (c). In our case, relative depth-ordering constraints as well as information about symmetries were provided as input (a) to create the final inflated shape (b). Note how our approach provides better flexibility in controlling the structure of the final 3D shape from a single viewpoint. To achieve comparable results multiple viewpoints would be necessary for the neural approach of Li et al. Images (c, d) courtesy of © Li et al. (used with permission).

UI, requiring a deeper understanding of the underlying algorithm. An experienced user can use those, e.g., to create more complex mesh topologies or locally control the amount of inflation of the final 3D model. One of those advanced controls is the *stroke labeling* widget that enables the creation of holes, interlocked segments, and cusps. Since our formulation is agnostic to the complexity of underlying mesh topology, our system can deliver meaningful output even in those more complex cases (see Fig. 3). Another widget we (mentioned in the main paper in Section 3.1) enables the user to control the *amount of inflation* (denoted as parameter c). This parameter can be controlled locally for each mesh component or

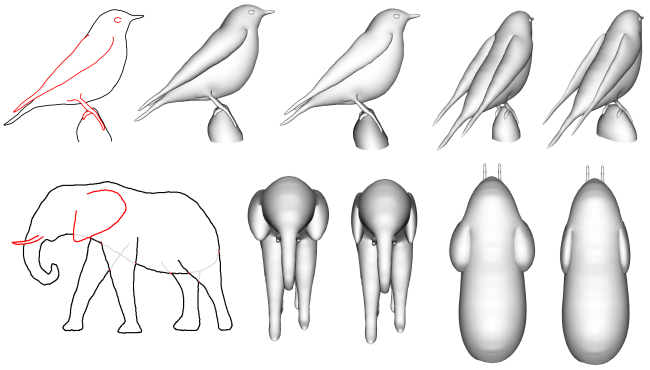

Fig. 6. Two examples of meshes where a user specified a different amount of inflation c to selected regions. The default inflation amount $c = 2$ was altered for the wings of a bird ($c = 0.5$) and pedestal ($c = 6$) (upper row), and the ears of an elephant ($c = 0.5$, bottom row).

even on a vertex level. In Fig. 6, we present two examples of meshes having a variable amount of inflation per segment. Note how this additional local control allows the user to reduce the bulkiness of parts and give them a more natural look.

REFERENCES

Mikhail Bessmeltsev, Will Chang, Nicholas Vining, Alla Sheffer, and Karan Singh. 2015. Modeling Character Canvases from Cartoon Drawings. *ACM Transactions on Graphics* 34, 5 (2015), 162.

Peter Borosán, Ming Jin, Doug DeCarlo, Yotam Gingold, and Andrew Nealen. 2012. RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation. *ACM Transactions on Graphics* 31, 6 (2012), 198.

Marek Dvorožňák, Saman Sepehri Nejad, Ondřej Jamriška, Alec Jacobson, Ladislav Kavan, and Daniel Sýkora. 2018. Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. 5.

Even Entem, Loïc Barthe, Marie-Paule Cani, Frederic Cordier, and Michiel van de Panne. 2015. Modeling 3D animals from a side-view sketch. *Computers & Graphics* 46 (2015), 221–230.

Even Entem, Amal dev Parakkat, Loïc Barthe, Ramanathan Muthuganapathy, and Marie-Paule Cani. 2019. Automatic Structuring of Organic Shapes from a Single Drawing. *Computers & Graphics* 81 (2019), 125–139.

Olga A. Karpenko and John F. Hughes. 2006. SmoothSketch: 3D free-form shapes from complex sketches. *ACM Transactions on Graphics* 25, 3 (2006), 589–598.

Changjian Li, Hao Pan, Yang Liu, Alla Sheffer, and Wenping Wang. 2018. Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling. *ACM Transactions on Graphics* 37, 6 (2018), 238.

Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. 109–116.