Supplementary Material
Interactive Sketching of Mannequin Poses

Gizem Unlu*
University College London
g.unlu@cs.ucl.ac.uk

Mohamed Sayed
University College London
m.sayed@cs.ucl.ac.uk

Gabriel Brostow
University College London
g.brostow@cs.ucl.ac.uk

1. Implementation Details

We will release our code on acceptance. Our Sketch Interpreter is built on top of the codebases of DensePose [2] and Keypoint-RCNN [3] for predicting silhouettes and 2D keypoints from human sketches respectively. Both networks are trained on our synthetic sketch dataset from scratch. DensePose predicts a mapping between images of humans and the surface of a template 3D model and is normally trained using a manually annotated dataset of humans. For our task, we generate a synthetic dataset with dense surface correspondences between synthetic sketches and the 3D body template, along with body part segmentation maps and 2D joint locations. There are 25K sketches in our dataset. We train all models using the default hyperparameters for 100K epochs with a learning rate of 0.002 on a single NVIDIA Titan Xp 12GB.

We lift 2D joints and silhouettes using a pretrained STRAPS [7] network. We used the pretrained weights for predicting 3D shape and pose parameters for the SMPL body model [5].

All models were implemented using Python and PyTorch [6]. The code will be made available on Github.

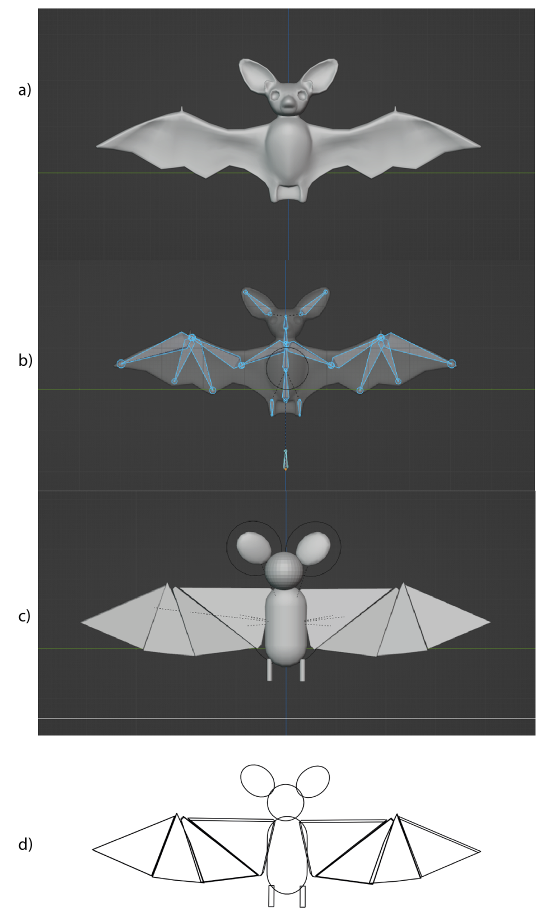

Figure 1. Sketch generation from a 3D bat model. Our synthetic sketch rendering system can be adapted to other creatures. a) 3D bat model. b) underlying bat skeleton. c) paramterized body part primitives. d) vector sketch.

2. Sketch Formulation

2.1. Qualitative Ablation Study on Sketch Augmentations

Our augmentation scheme is robust to missing strokes and even whole limbs and body parts, as can be in Fig 2. Fig 2.a and Fig 2.c show examples where the user forgot to place strokes for the right arm joint and the whole torso, respectively. The inferred 3D pose is often robust to such cases, as our body-part aware augmentation scheme is formulated to hide whole body parts during training. Fig 2.b illustrates a missing arm; the model successfully hallucinates a plausible pose for the arm (Fig 2.b Green) while keeping the rest of the body pose mostly unaffected.

Our 3D Primitive Human Body Model can represent front and back facing body poses. Fig 2.d illustrates this case: the addition of facial cross strokes (for the eyes and nose) to the left sketch changes the characters pose from

*http://visual.cs.ucl.ac.uk/pubs/sketch2mannequin

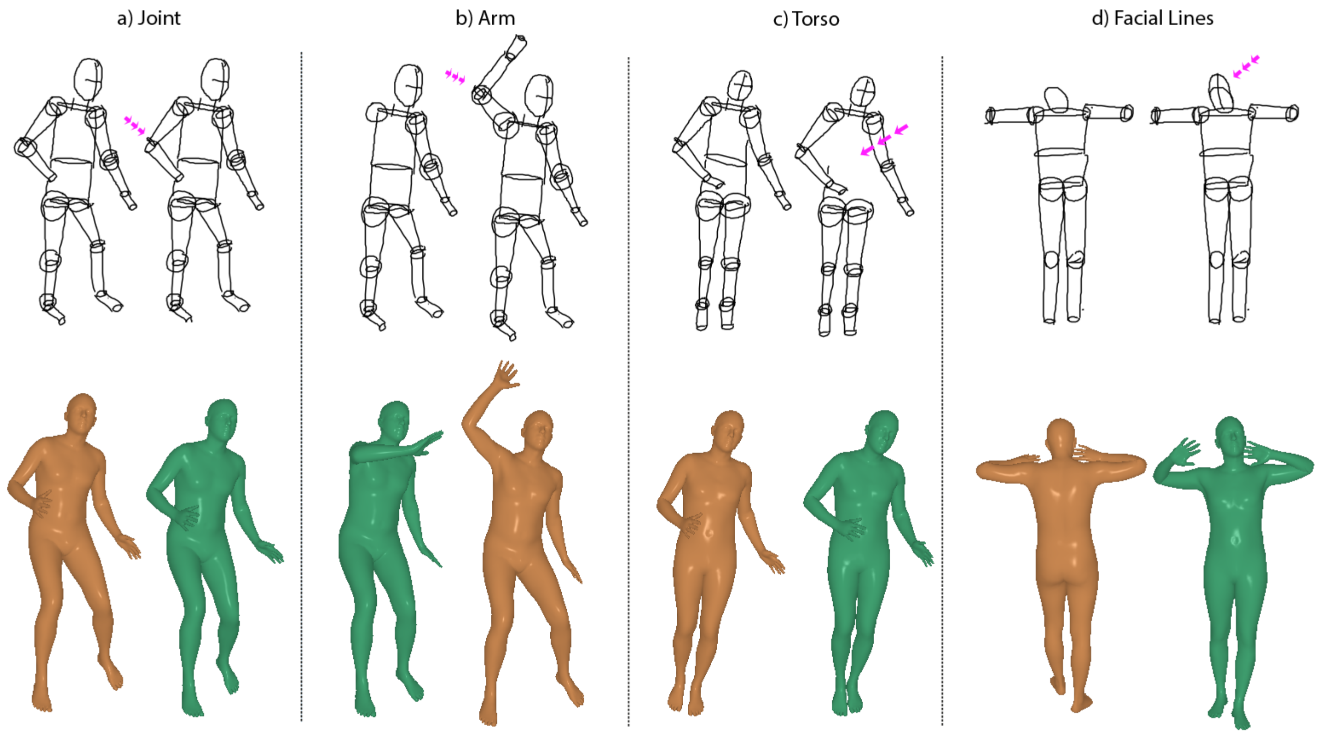

Figure 2. Effect of missing sketch lines for joints and limbs. a-c) Our model is robust to missing limbs, joints, and sketch lines. d) Face strokes define head orientation and by extension - through our model's prior on realistic joint limitations - body orientation.

facing away from the camera to facing the camera.

2.2. Other Sketchable Characters

Our approach could be expanded to characters beyond humans, with a little effort but without resorting to costly data collection and labeling. In Figure 1 we illustrate a pipeline for generating sketches of bats but we can extend the approach to any rigged and skinned 3D creature. Following our human model, we represent a primitive bat model composed of 3D primitives such as triangular prisms, 3D ellipses and cylinders, The same figure also shows a sketch rendered from this primitive model in Blender. A training dataset of bat sketches could be easily made by posing the bat and rendering sketches coupled with our vector-graphics augmentation scheme.

3. User Study

During our user study, we recorded how much time each user spent for the two tasks they were given: posing by sketching and 3D refining versus 3D refining from a T-Pose. In Fig. 3, we present these results. We can see that users consistently spent less time sketching than using the 3D controls. This further indicates that our sketch-based tool is a fast and easy way of posing 3D characters compared to manual refinement. For the two modes, a single user was given two different target poses, and poses were different across users.

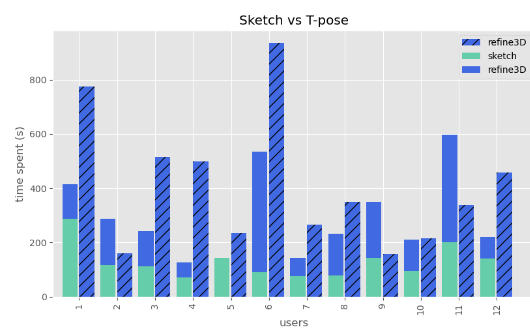

Figure 3. Time spent on both manual refinement and sketch + refinement per user in our user study. Sketching along is consistently very quick compared to manual refinement, and even with the added time penalty of further manual refinement is still faster when compared to pure manual refinement.

4. Sketch2Pose [1]

In preparing this manuscript, we discovered a preprint of Brodt & Bessmeltsev's Sketch2Pose on their website in May 2022, with the paper set to officially appear in ToG in July 2022 [1]. They will obviously go to press ahead of us,

	Sketch2Pose [1]	Ours
Input	Rasterized sketch	(initial) Rasterized sketch (later) Edits on sketch and direct manipulation of skeleton
Output	Default body shape and target pose in SMPL space	Default body shape and target pose in SMPL space AND kinematic skeleton rig
Speed (ave)	To sketch: ?? To process: 90 sec To refine/edit: -	To sketch: 130 sec To process: 0.16 sec To refine/edit: 162 sec
Target user scenarios	User with access to 2D sketches, who wants to convert them to 3D posed meshes	Artists who wish to actively sketch and iterate in 2D and 3D
Compatible sketch styles	General and varied	Figure sketch more or less based on cylinders
Training Data	15k Paired real-sketches and 2D joint annotations	Synthetic renders of mocap-based poses
Effort needed for other creatures	New SMPL-like parameterization Re-creating manual sketches 15k annotations of 2D skeletons New heuristics for self-contacts, bone foreshortening, and bone tangents	New SMPL-like parameterization Converter for "SMPL" to synthetic cylinders; simulator or pose-library
Date of publication	July 2022	TBD (!)

Table 1. Key factors in common and differentiating Sketch2Pose [1] (middle column) versus our approach (right). The artist-training for our system consists of watching a 2:40min tutorial video, while training for Sketch2Pose is either N/A or zero, since the user hopefully has pre-existing artistic skills and does not interact with that system.

but the two research projects have been developed around the same time and completely independently, with our initial efforts starting in 2020. Before realizing that our artists were vehement about using a system that allowed iteration in 2D and editing of the 3D result, our system was initially even called "Sketch2Rig."

We appreciate that reviewers will notice similarities and differences between the two approaches, and for convenience, aim to summarize both for the interested reader in Table 1. In brief, we both set out to break the organic-human-sketch-to-3D barrier. Sketch2Pose got there first, and we respect that they can also accommodate a larger variety of sketching styles. We both focus on one body shape in SMPL space, under different poses. But the methods are different, with big implications for scalability to other creatures, and the outcomes are very different because we target different users. Ours is an interactive tool that gets used by artists hands-on. Sketch2Pose is for an operator who wants to process pre-existing artist sketches as if they were photos going through a monocular pose estimation vision system, but finds the unrealistic style and proportions prohibitive.

Details on Quantitative Evaluation In Table 2 of the main paper, we provided a quantitative comparison of the accuracy of our model versus Sketch2Pose, despite the mismatch in speed and use-cases. Here, we provide further details for that experiment. Following the COCO 2D human pose estimation convention [4], we predict 17 keypoints: 5 facial and 12 body keypoints (3 keypoints for arms and legs). Sketch2Pose uses 18 joints on the body (3 keypoints for limbs, 3 keypoints along the spine, and 1 for the head.) We map 12 of their keypoints to ours, using the keypoints on the limbs, to calculate an error metric for 2D pose. For 3D pose, we follow the SMPL convention and use 21 joints for calculating MPJPE.

References

[1] Kirill Brodt and Mikhail Bessmeltsev. Sketch2pose: Estimating a 3d character pose from a bitmap sketch. *ACM Transactions on Graphics*, 41(4), 7 2022. 2, 3

[2] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 1

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Zitnick. Microsoft coco: Common objects in context. 05 2014. 3

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-

person linear model. *ACM Trans. Graph.*, 34(6), oct 2015. 1

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[7] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Synthetic training for accurate 3d human pose and shape estimation in the wild. In *British Machine Vision Conference (BMVC)*, September 2020. 1