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

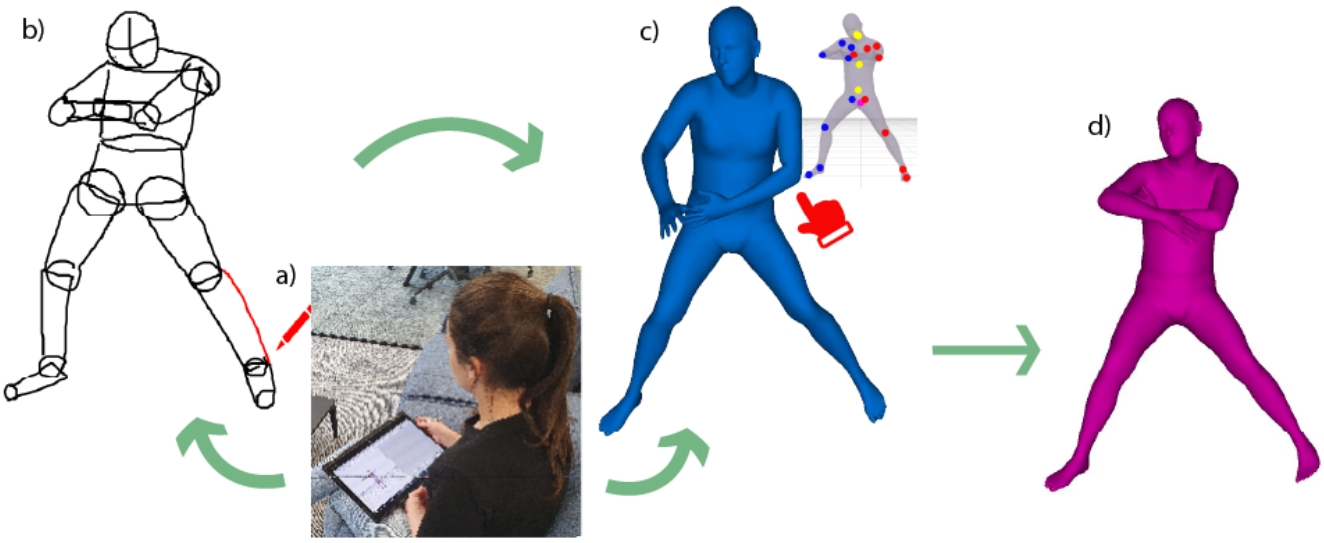

Figure 1: Overview of our interactive sketch-based 3D figure posing. From figure sketches (b) we infer an initial 3D prediction (c). The user (a) then iterates on the sketch, or on the 3D pose using Forward and Inverse Kinematics handles to achieve a final refined pose (d).

Abstract

It can be easy and even fun to sketch humans in different poses. In contrast, creating those same poses on a 3D graphics "mannequin" is comparatively tedious. Yet 3D body poses are necessary for various downstream applications. We seek to preserve the convenience of 2D sketching while giving users of different skill levels the flexibility to accurately and more quickly pose/refine a 3D mannequin.

At the core of the interactive system, we propose a machine-learning model for inferring the 3D pose of a CG mannequin from sketches of humans drawn in a cylinder-person style. Training such a model is challenging because of artist variability, a lack of sketch training data with corresponding ground truth 3D poses, and the high dimensionality of human pose-space. Our unique approach to synthesizing vector graphics training data underpins our integrated ML-and-kinematics system. We validate the system by tightly coupling it with a user interface, and by performing a user study, in addition to quantitative comparisons.

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

1. Introduction

Sketching people's body poses is challenging but fun. The artist's aim could be just creative pleasure, *e.g.* there are hundreds of YouTube videos showcasing body-pose drawing. Or the artist may have some specific downstream task such as story-boarding an action film or drafting a comic book. Sketching humans, as in [26], is an often-recommended strategy for artists to get started. Assemblies of primitives help the artist to figure out framing and pose, before embellishing with clothes and facial features.

We seek to give artists who focus on people's poses the fun and convenience of sketching, while iterating to end up with usable and editable 3D human mannequin geometry. On one hand, we are inspired by sites like Figurosity that stretch skilled artists so they can hand-draw realistically proportioned people in interesting poses. On the other hand, we wish to emulate the accessibility and 2D-to-3D functionality of Teddy [21] and MonsterMash [15].

To make progress and lower the barrier to entry, we propose a hybrid machine learning (ML) and kinematics sys-

tem. We want the user to manipulate the sketch or the pose of a human mannequin, while keeping the body shape fixed.

Overview Through our web interface, a user sketches on a blank canvas as pictured in Fig 1.b. The sketch should depict one person in the desired pose. Their body should be made up of cylinders, an ellipsoid for the head, and with circles at the joints. The sketch can be imperfect, with some parts missing, with moderately messy lines, and out of proportion limbs. Then our neural-network based model interprets the sketch as a posed 3D mannequin, as shown in Fig 1.c. The user can explore other poses by redrawing sections of the sketch. Equally, our generated mannequin has basic rigging and joint limits that support the user in obtaining a refined pose (Fig 1.d) through both forward kinematics (FK) and inverse kinematics (IK) interaction handles.

To our knowledge, this is the first human-in-the-loop system for making and editing mannequin poses based on such sketch inputs. Our main technical contribution is a vector-graphics data synthesis and augmentation algorithm designed specifically to a) overcome the absence of real paired training data for sketches with 3D, and to b) cope with the highly variable and sometimes unrealistic sketching styles of beginners.

2. Related Work

Here, we recap examples of sketch-based figure modeling, ranging from general-purpose sketching to our nearest neighbors in articulated creature sketching.

Classic 3D from Sketches: Starting with Teddy [21], a user could learn to slightly adjust their drawing style to hint to the system what 3D shape was desired. FiberMesh [40] builds on Teddy's interactive sketch modeling and blobby-surface optimization, to enable more controlled modeling. Some sketch lines serve as control curves and remain on the constructed 3D surface, where the user can push and pull them. This ability to refine the output is important to us.

In contrast to Teddy and FiberMesh, ILoveSketch [3] and EverybodyLovesSketch [2] allow the artist to determine construction surfaces by sketching and drawing 2D curves on planes in multiple views. This works well for fairly skilled artists and man-made shapes. Between Teddy and ILoveSketch, [45] is multi-view, silhouette-based, and includes Boolean operations, where the 2D sketches must capture the shape from two orthographic views, *e.g*. the top and side.

In this space, [25, 17] and [41] sit closest to our use-cases. [25] expects a template 3D model *e.g*. of a dog. Their template is deformed to align to the sketch contour via HMM feature correspondences between sketch strokes and template vertices. Similarly, [17] presents an easy-to-use single view method, but requires descriptive stroke annotations *e.g*. normals and length information, and object part annotations of primitive shapes. In contrast, we ask our users to draw primitives without needing annotations, and infer the SMPL parameters of a human-specific learned feature representation [35]. NaturaSketch [41] uses the contour inflation technique in Teddy. Unlike ours, their interface does not allow the user to modify or refine the produced 3D shape. Another approach focuses on sketches of symmetric shapes such as animals in a relatively symmetric pose[16]. The structurally symmetric parts are determined from the strokes in a contour image, and the depth hierarchy of these parts is determined automatically. Only contour sketches from a side-view can be constructed. Similarly, [14] is better at capturing fine details from the sketch, but requires more input than [16], namely a depth hierarchy of the semantic parts in the sketch.

Data-Driven Learning for Reconstruction: Machine learning (ML) models are emerging that seek to capture correlations between sketches and their corresponding 3D shapes. Sketch2Pose [9] is concurrent work to our own, and despite being non-interactive, is our closest-neighbor, having an ML approach to finding at least the 2D joints, and using the same SMPL body representation [35]. We discuss [9] further in the supplementary material. However, since realistic sketch-to-3D paired data is difficult to obtain in large enough quantities, many approaches exploit the ShapeNet [10] 3D repository, and seek to render its models in sketch-like Non-Photorealistic (NPR) styles, *e.g*. Suggestive [12] or Neural Contours [33].

As with photo-based 3D inference of shapes [49], convolutional neural networks (CNNs) are being explored to handle sketches [36, 53, 34, 48, 28]. Given multiple sketches from different views, the network in [36] infers depth and normal maps to construct 3D point clouds.While the results on these are remarkable, the network requires carefully rendered orthographic drawings from a side or frontal view. It is not suitable for amateur sketches.

In contrast, [53] takes on the task of 3D point cloud reconstruction from a single sketch. Their model is composed of an off-the-shelf sketch synthesis network [34], a sketch standardization module, and a reconstruction network. For generalization, augmentation includes deformations such as distortion, dilation, and erosion. They compare to previous works that use edge maps or NPR algorithms for sketch dataset generation, and show that training with their sketches generalizes better to real human drawings. That approach is scrutinized among many others in the large-scale study of Yue *et al*. [55]. Yue *et al*. determine the key challenges of working with sketch data. The most prominent differences between sketch and RGB image inputs are the former's sparsity and variation in sketching styles, and the imperfect nature of free-hand sketches in terms of perspective. They conduct experiments with three objects classes from ShapeNet. They also show that training with multiple categories vs. a single category hurts perfor-

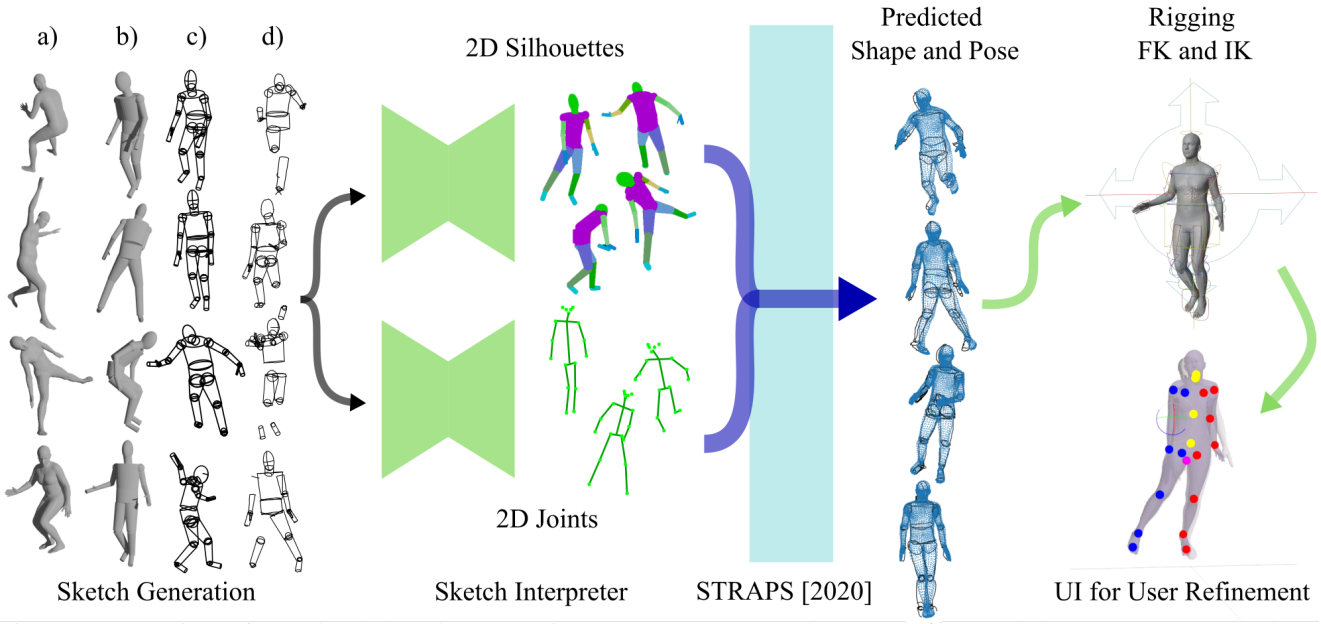

Figure 2: Overview of our sketch-based mannequin poser. (a) We sample SMPL poses and (b) generate our novel 3D primitive human model by placing 3D primitives on each sampled pose. (c) To generate sketches, we render the 3D primitive human to a 'perfect' clean sketch. (d) Our augmentation scheme alters the clean sketches to mimic human-made sketches during network training. Given a figure sketch, our *Sketch Interpreter* predicts 2D labeled silhouettes and joints from which 3D body pose and shape are inferred. The user can refine predicted poses or finetune their sketch interactively using our easy web-based user interface. Please note the poses in the figure are randomly selected samples and are unrelated through (a-d).

mance. On the basis of their evaluations, we compare to SynDraw [52] as a data-augmentation baseline.

A leading work for sketch-based modeling is SketchCNN [31]. They have a two-stage approach instead of directly inferring the 3D geometry from sketches. First, an input sketch is mapped to an intermediate flow field. This representation contains local curvature information from the sketched object. Then a second network predicts the depth and normal maps corresponding to the input sketch and inferred flow-field. The user needs to distinguish contour strokes from other strokes. The interactive user interface allows for sketching from multiple views and 3D refinement. Also, the system can handle additional hints about curvature and depth by stroke annotation. This is still less input than counterpart BendSketch[30] requires, *i.e*. to separately annotate each stroke *e.g*. ridge/valley, curvature line, depth discontinuity, and boundary.

Delanoy *et al*. [13] present a modeling tool for predicting volumetric occupancy grids from sketches. Their pipeline has an initial single-view volume prediction step utilizing a user-drawn sketch from a viewpoint. The user can continue refining the shape from different viewpoints, and updated volumes are obtained iteratively. Their approach is based on two-volume predicting networks: a single-view prediction CNN and an updater CNN. Initially, the single view CNN is trained on groundtruth sketch-3D model pairs. Then, the updater network uses the output of the first network from a random view and compares its own construction. Note that their method is aimed at professionals who are experienced in perspective drawing, while ours is for amateurs too. Similar to [13], Sketch2CAD[29] presents a data-driven modeling system aimed at users experienced in sketching and product design, but inexperienced in 3D modeling. The authors draw parallels between the steps in a CAD modeling session and those an industrial designer follows when sketching in 2D. Motivated by these similarities, they create a tool where the user sketches the shape edits incrementally. The system automatically processes each increment into an appropriate CAD operation. Overall, their tool could be attractive to product designers.

Sketch-Based Articulated 3D Figure Modeling: Some earlier work in sketch-based articulated figure modeling focused on stick figures. Davis *et al*. [11] provide a medium for artists to create 3D animations from a sequence of stick figure sketches, with user annotated skeletal keypoints. The system is not fully automated, but gives the artist a choice to select among possible 3D poses. In addition to 3D pose lifting, [37] infers the sketched character's body proportions and transfers it to a morphable 3D model.

Motion Doodles [51] explores the task of sketching motion. The user can author a jump or somersault by drawing a path. The system supports both 2D and 3D animation for sketched characters. [22] infers the 3D motion from hand-drawn sketch animations, but requires the labeling of body

landmarks on the sketched body. In contrast to these methods, Akman *et al*. [1] propose a deep learning approach to directly predict 3D point clouds from 2D stick figures. By interpolating the latent features of two sketches, the reconstructed 3D point clouds can be post-processed into articulated mesh models. However, their user interface does not allow for inputting start and end sketches or correcting erroneous reconstructions.

Apart from stick figures, methods for modeling and animating more complex shapes were proposed [27, 6, 7]. To accurately lift the 3D pose from input sketches, these methods require the artist to explicitly define a 3D skeleton. ArtiSketch [27] requires the user to create multi-view sketches of the character. Along with multiple sketches, the artist must also provide 3D skeletal pose models for each view, using an external 3D tool. In [6], 3D articulated figure modeling can be done from single contour drawings of cartoon characters and corresponding 3D skeletons. [6]'s need for explicit pose information is alleviated in Gesture3D [7] but the system still requires a template 3D mesh instead of a predefined posed skeletal structure for each input sketch. The recent MonsterMash [15] achieves great modeling successes using only sketches, and is therefore one of our baselines. The need for a model template or 3D pose information is eliminated by the user separately annotating meaningful parts of the sketch and indicating if a part is positioned in front of neighboring parts. In contrast to these methods, RigMesh [8] combines the rigging and animation steps by automatically constructing the skeleton from contour sketches. As a variant of Teddy, which uses the Chordal Axis Transform [44] to inflate the contour sketch, RigMesh creates the skeleton of the 3D figure from the chordal axis. However, inferring the skeleton from simple contours suffers from unnatural and ill-positioned joints [6].

3D Human Pose and Shape Estimation: There is a rich history of human 3D pose and shape estimation algorithms for a single RGB image. Recently, classical methods [47, 18, 4] have been replaced by deep learning based approaches, *e.g*. SPIN[24], HMR[23], [32], and [42]. Please see [54] for a more in-depth survey in this field. For this work, we employ the architecture in [46] for estimating 3D parameters from human sketches.

3. Method

Our system enables a user with a digital stylus to quickly position the limbs of an articulated 3D human "mannequin," mostly by means of figure sketching, as illustrated in Fig 1. An overview of our method is in Fig 2. In Sec 3.1, we describe our underlying generalized-cylinder-like representation [38] called the 3D Primitive Human Body Model. To overcome a lack of real and varied training data for sketch-based 3D pose estimation, we introduce our core synthetic data augmentation strategy for generalizing this model to artist drawn sketches in Sec 3.2. In Sec 3.3 we describe the model we use for predicting human pose from 2D sketches.

3.1. Figure Sketching

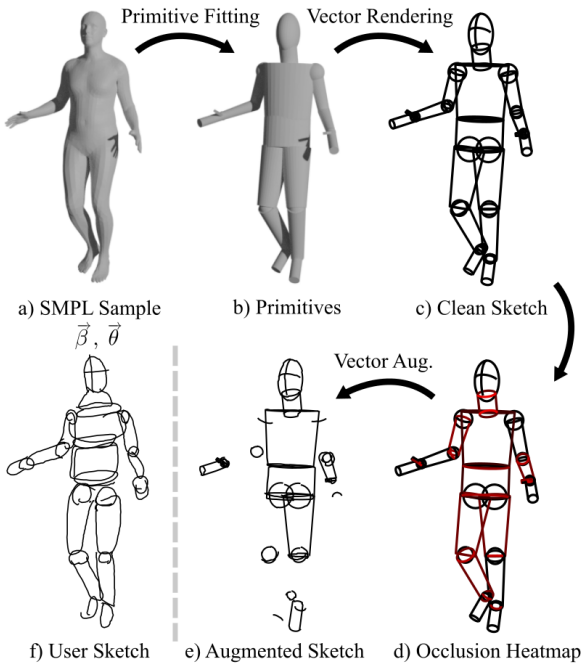

Figure 3: Overview of our synthetic sketch generation. a) we generate a sample SMPL body with pose $\vec{\theta}$ and shape $\vec{\beta}$. In b) we fit geometric primitives to the generated body. We then render these primitives down to clean vector sketch lines in c). d) While all joints and vector strokes are susceptible to being deleted as part of augmentation, strokes with more hidden nodes are given a higher probability of being deleted. e) Our augmentations include local node translational jitter, deleting strokes and joints, and global stroke translations. f) An example *user* sketch from our user study.

For human figure sketching, artists commonly use either their imagination or reference images as inspiration. To quickly convey the reference pose of a human, regardless of the details in the inspiration, one method is to use simple primitive shapes for body parts, as advocated by Lee & Buscema [26]. Inspired by this, we created the *3D Primitive Human Body Model* (**3DPHB**) in Fig 3.b.

3DPHB has correspondences to the SMPL parametric body model [35]. Figs. 3a-b show an example of how our Primitive Human relates to SMPL bodies: we sample a set of body shape and pose parameters ($\vec{\beta}$, $\vec{\theta}$) and place the following primitives:

- a 3D ellipsoid for the head,
- tapered cylinders with varying radii at each base for limbs,
- spheres for joints,
- and two tapered cylinders for upper and lower torso.

Note that we draw samples with very distinct poses, but with little variation in the shape, so the generated mannequins will differ mostly just in pose. Informally, we observed that allowing shape to vary significantly would mean that beginner artists got less predictable results. The lengths of body part primitives are directly aligned with the original body. The upper torso and lower torso are aligned to the width of the shoulders-waist and waist-hip segments, respectively. The head is sized w.r.t. the top of the head *vs.* neck, and to fit between the ears.

3.2. Part-Aware Augmentations for Figure Sketches

A major obstacle for training CNNs for sketch-based tasks is the lack of sketch datasets with corresponding 3D labels. Paired data for sketch-based 3D reconstruction would ideally span various poses and drawing styles, though we view "style" as mostly meaning artist thoroughness and precision. Thus, for 3D pose prediction from sketches, we generate a synthetic sketch and 3D human pose dataset coupled with a vector graphics augmentation scheme to generalize to human-made sketches.

Using our 3DPHB model, we produce sketch-like renderings using a set of different line types: silhouettes, contours, creases, and borders, as can be seen in Fig. 3.c. To convey the head orientation, a vertical and horizontal line are used for the eyes and nose.

Renderings are stored into a vector graphics format for our synthetically generated sketches. This format is extremely flexible and useful for storing extra information such as per-stroke body part labels and occlusion. We associate each stroke with a corresponding body part label. Such labeling of the 2D renderings allows us to control the types of augmentations we can apply to each body part during CNN training. More specifically, for a given camera pose, we use ray casting operations to determine whether a stroke is visible or occluded. Since each body part could consist of several strokes, we assign each part with an average occlusion rating based on this information. Our body part-aware sketch augmentations include global translation of body parts, local stroke jitter, part-based hiding tied to occlusions, and random part-based hiding. The latter two are needed to simulate two artist behaviors we observed in a pilot study. First, people differ in which primitive lines they decide to put on the canvas. Depending on preference, the artist could decide to keep or discard strokes for occluded body parts. Also, people tend to forget to draw the sketch lines for some body parts, especially at already-crowded joints.

More formally, our part-aware figure sketch augmentations can be defined as a set of transformations applied to each body part. We denote an undisturbed figure sketch rendering as set S and an augmented sketch as S^*. We denote all augmentation operations as $aug(*, A)$ where A is the set of part augmentations that affect different body parts. S and S^* have the relationship that $S^* = aug(S, A)$, where S consists of a set of body parts $B = \{j, l, t, h\}$. Here, j, l, t, and h denote joints, limbs, the two-piece torso, and the head with neck. Each body part $b \in B$ could consist of several strokes s_b and different augmentations could affect either a complete body part or a single stroke. Specifically, we use the following set of augmentations: $A = \{translate, jitter, hide_{random}, hide_{occluded}\}$. A body part could be translated by sampling translation offsets $b^* = b+t$, where $t \sim \mathcal{N}(\mu, \sigma^2)$ is a randomly sampled from a Gaussian Distribution with mean μ and variance σ^2. Similarly, we add jitter to strokes, so $s_b^* = s_b + t, t \sim \mathcal{N}(\mu, \sigma^2)$. Another type of augmentation is hiding of body parts. We simulate missing body parts by turning off sketch lines randomly. Further, to implement occlusion-based hiding, we give an occlusion rating to each stroke: given a stroke s consisting of subnodes $n_s = \{n_1, n_2, n_3, ..., n_n\}$ with total number of occluded nodes $v \leq n$, the occlusion rating $o_s = \frac{v}{n}$. Thus, a high o_s implies a higher probability of the stroke s being hidden.

3.3. 3D Pose Estimation from Sketches

To predict the 3D body pose from a sketch, we use a two-step pipeline, depicted in Fig. 2. Given an input figure sketch, our Sketch Interpreter infers 2D joint locations and silhouettes. We use these intermediate representations between sketch input and 3D output due to the nature of the sketch medium in general [55]. Inferring 3D information from 2D input alone is ambiguous. Sketches are sparse in nature and compared to RGB images, they contain fewer 3D cues, lacking textures and shadows. Those cues are available for 3D lifting tasks that start with real photographs. Previous work in both image-based[50] 3D reconstruction shows that using silhouettes as an intermediate representation does help.

The 3D lifting stage is used to predict 3D pose information using the intermediate representations inferred from the input sketches. As a means of lifting, we incorporate the sketching-unaware pre-trained 3D network from STRAPS [46], which takes silhouettes and 2D joint locations as input, to infer the 3D shape and pose parameters that we seek, to reflect the upstream input sketch.

Implementation Details Our Sketch Interpreter is built on top of the codebases of DensePose [19] and Keypoint-RCNN [20]. 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

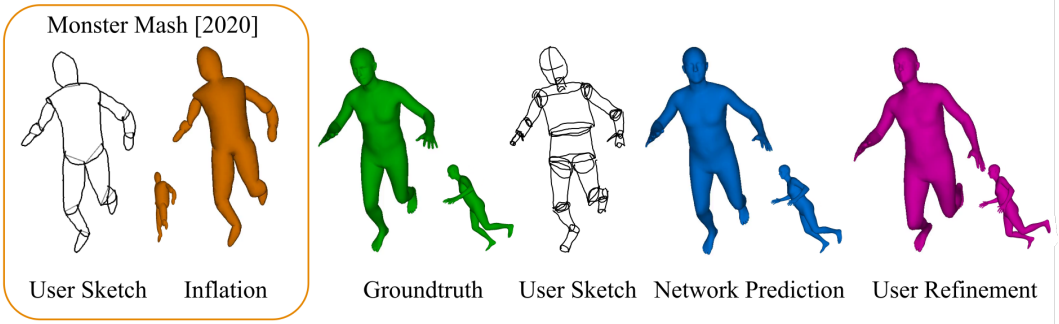

Figure 4: Example sketch, intermediate output, and refinement from our user study and MonsterMash experiment. For each model, we show both a front view (large) and a side view (small). The green model shows the frontal pose that was given to users of both systems as a reference. Highlighted with orange is a user's sketch input to MonsterMash [15] and the resulting 3D shape. Note that the grey lines in the MonsterMash sketch are auto-completed as an intermediate step in their pipeline. Our initial network prediction (blue) is already faithful to the input sketch, and the groundtruth (green); the user chose to refine further via classic 3D controls (purple). MonsterMash does more than posing, but since its inflated models start in a planar world, much of the information on depth embedded in the sketch is lost.

are 25K sketches in our dataset. We train all models using [19] and [20]'s 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 [46] network. All models were implemented using PyTorch [43]. The code will be made ready on GitHub.

4. Experiments

We validate our system and its components in three ways. First, a user study in Sec. 4.1 compares our approach in the context of posing 3D mannequins to match a reference image. The evaluation metric is in Sec. 4.2. Second, we perform ablations of our sketch generation pipeline, and comparisons to [9] and against a state-of-the-art sketch augmentation baseline [52] in Sec. 4.3. Third, we demonstrate qualitative results of *interactive* user sketching and mannequin posing here and in the supplemental video.

4.1. User Study

We evaluated our system with a collection of novice users. This means they had a variety of backgrounds, but had never done sketch-based modeling or similar tasks. We define two modes, so users were asked to either 1) adjust a human body from a canonical pose (T-pose) using 3D handles, or 2) users were asked to first sketch out a human pose, allowing our model to predict a mannequin pose to replace the T-pose, and then continuing to refine the pose as in the first mode. To help even the playing field between amateur artists and to make the results more measurable, users were given a reference render of a known ground truth posed human mesh that they must aim to closely mimic. The reference renders were generated using the same 3D body shape that users refined in both modes.

User Interface We built our UI to allow for both 2D and 3D manipulation of sketches, and direct manipulation of a 3D posed human. Users can draw strokes using their fingers or a stylus on a touch screen, or a mouse, though all our users opted to use a stylus for their sketches. We used a combination of three.js, Blender, JavaScript, and our Python backend. Users were allowed access to the UI via a browser, making it readily accessible on most consumer electronic devices. For 3D refinement of rigged meshes, we adapted the Rigify extension in Blender to post-process network predictions. This allowed us to add joint limits and expose interactive 3D models amenable to control via Forward Kinematics (FK) and Inverse Kinematics (IK). These controls are available to the user in both study modes.

Study Details We invited 12 users to participate in our user study. We asked users to go through a four-step process: a tutorial video, a practice period, and both a session in the manual T-pose refinement mode and a sketch-to-3D-session. The tutorial video introduced users to our UI and gave them an overview of the generic task they needed to accomplish along with illustrations of how to sketch figures and how to use FK and IK. The practice period was 10 minutes long and allowed users to become familiar with the UI, figure sketching, and 3D controls. After the practice session, users were asked to finish both tasks sequentially, with 10 minutes for each. We randomized the order of tasks and the reference render for each user. We had arbitrarily chosen the parameters controlling the amount of data augmentation in the training of our model. Later, we will refer to this parameter setting as Augmentation 2.

Data Collected All user actions, including sketch strokes and timing, were recorded for evaluation. To measure how much each user spent for each task, we compute how much time was spent on sketching and separately on using 3D FK/IK controls to the point where the user was satisfied with the result. We also collected feedback from users via a questionnaire at the end of the user study.

a) Free-flow Sketch b) Front View c) Side View

Figure 5: Some users (separate from the evaluated user-study) were allowed to sketch on a blank canvas. Some sketches are in a). Users (one for each row) then refined the network's predictions (blue) to get the pose that they were really after in 3D (highlighted in purple), potentially deviating from the pose they had in mind initially.

4.2. Evaluation Pipeline

We perform evaluation using two 3D metrics: Chamfer Distance on sampled point clouds and Mean Per Joint 3D Position Error (MPJPE) [39] on the underlying joints. For both modes of the user study, we report metrics comparing the refined 3D models to the ground truth, averaged across all user sessions. For Sketch Refinement, we also evaluate the initial prediction from our system from the initial user sketch, before any further refinement. We've found that users may often misjudge the orientation of meshes along depth given the reference 2D image. To account for this, all 3D metrics are computed after meshes have been aligned, as a rigid body, to the groundtruth using Iterative Closest Point (ICP) [5].

4.3. Ablation Study

We subsequently ran an ablation study to quantify the effect of our sketch augmentations on sketch-based 3D pose estimation. We trained our Sketch Interpreter (Fig 2) with three levels of sketch augmentation severity, so based on Ours(Default) plus and minus 10%. More specifically, we experiment with the severity of our part-aware augmentation strategy. We train two more baseline models without vector-based augmentations: our sketches and our sketches but with a single-piece torso in-place of two. We also compare to the network and optimization of Sketch2Pose[9].

4.4. Baseline Systems

Sketch2Pose [9] was developed in parallel to our approach, and represents the current SoTA in pose-from-sketch inference. We compare against it in Table 2, though the comparison is somewhat unfair to us, evaluating their offline computer vision system to our interactive real-time system for artists to sketch and iterate. Their supervised training lets them interpret sketches as 2D joint locations, but then requires a costly 90 sec. optimization to produce a mesh. Our method performs comparably to Sketch2Pose on real sketches, in a fraction of the time, as seen in Fig. 6 (Right). We compare against this one-shot image-to-model converter because of the partial overlap in their initial stage, despite ours being a real-time system without reliance on labeled sketches.

To evaluate our body part and occlusion aware vector augmentations, we train our Interpreter with the vector augmentation baseline scheme from SynDraw [52].

MonsterMash [15] requires a very specific set of input strokes for sketches to be valid, so our user study sketches do not produce meaningful results in their tool. We do however compare qualitatively on an adapted reference sketch in Fig. 4. With 30 minutes of practice, a user was able to sketch meaningful input that would produce the best available inflated mesh. This comparison too is somewhat unfair, because the authors of that system could likely teach our users more about the intended drawing rules.

5. Results

We present qualitative and quantitative results for our interactive sketch-based system. All quantitative results are reported on real sketches collected during our user study, using *Ours(Default)* settings, as shown in Tab 2.

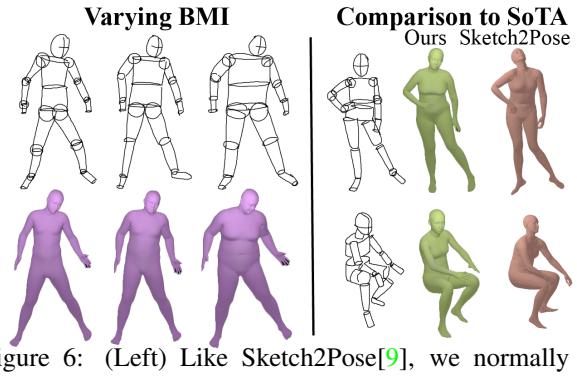

Figure 6: (Left) Like Sketch2Pose[9], we normally fix shape, and infer only pose. Here exceptionally, are our pose *and shape* estimation results for sketches with varying BMI. (Right) Generalizing to unusual poses may prove challenging, but our current results are on par with Sketch2Pose, pictured. Also, ours is 560x faster and allows the artist to refine the result further.

User Study We report quantitative results from our user study in Table 1. Sketch+Refine performs similarly to classic manual refinement on chamfer distance while achieving better scores on 3D joints. On average, sketching alone was almost four times as fast as manual refinement and achieves competitive accuracy. When adding the extra time needed by some users to refine our prediction, the total time remains shorter compared to manual refinement. Crucially 70% of our users preferred using sketching and partial refinement to just manual refinement.

Ablation We ablate our system in Table 2. We use all 48 real human-drawn sketches from our user study (includes practice sessions) for comparing our ablated model and competitors. All three of our augmentation schemes perform better than or similarly to baseline methods. Among those, Ours(Heavy), which is the scheme with the most severe augmentations, outperformed all baselines and competitors. We observed that users tend to forget to put down strokes for body parts; thus, training with missing strokes and body parts better helped to generalize to real sketches.

Qualitative Results Please refer to Figure 5 for free-flow sketches and to Figure 4 for a sample of a user study sketch. Please also see the supplemental material. We limit the scope of this paper to human pose estimation from sketches. However, our system can interpret simple shape variations, as illustrated in Fig. 6 (Left).

6. Conclusion

We demonstrated a highly interactive system that allows users to sketch the desired 3D pose of their mannequin, in "The Marvel Way" [26]. The system provides multiple methods for refining the estimated pose, which is important to users for the creative process to be more than just a curiosity. Surprisingly, manipulating even IK handles is slow and cumbersome enough, that users starting from a T-Pose had a disadvantage compared to starting by sketching out the body using primitives. Sketching required very little training and was also reported to be more fun!

The system has limitations (shown in Fig 7) that are obvious once the artist draws body shapes that are very different from the training data. This is a problem for very unusual poses, such as drawing upside-down people. Future work will explore the trade-offs when training a model for a variety of poses and body shapes. Until then, the resulting SMPL-based mannequin can be manipulated using sliders (not in our UI) to achieve other shapes. Scaling to other creatures is achievable using our synthetic vector-graphics renderer.

Acknowledgements We would like to thank Prof. Iasonas Kokkinos for valuable guidance and PhD funding from Niantic and Microsoft.

	Chamfer↓	Joint3D↓	Time↓
Canonical Pose	0.02931	0.2647	-
Manual Refine.	0.00730	0.1208	402.35s
Sketch Prediction	0.01006	0.1224	135.498s
Sketch + Refine.	**0.00652**	**0.0933**	**313.877s**

Table 1: Quantitative results of the user study with 18 participants. Distances are in meters. Our prediction from user sketches alone scores competitively compared to manual refinement, while taking a quarter of the time on average. Further interactive manual refinement (Sketch + Refine) improves mesh and joint metrics, while still taking little time.

	Chamf.↓	Joint3D↓	Joint2D↓	MPVPE↓
Sing.Torso	0.02097	0.2595	69.47	0.3126
Doub.Torso	0.0143	0.1825	53.08	0.2260
SynDraw [52]	0.0245	0.2537	92.65	0.3007
Ske2Pose [9]	0.0070	0.1682	21.09	0.1607
Ours(Def.)	0.0086	0.1149	19.54	0.1391
Ours(Light)	**0.0065**	0.1031	18.62	0.1300
Ours(Heavy)	0.0069	**0.0953**	**18.20**	**0.1230**

Table 2: Quantitative comparison of our augmentation method and its ablations against baselines. We used Default augmentation (so untuned for this test set) for our user study, which is already competitive against [9] (which requires an extra 90 second optimization). We also show tests of our model with ±10% augmentation, which indicates novices could benefit from heavier augmentation.

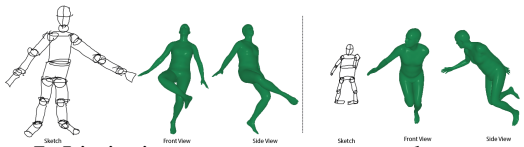

Figure 7: Limitations: our system can produce unexpected mannequin poses when provided with limb lengths that are implausible (left) or out-of-distribution sketches, *e.g.* input that is too small.

References

[1] Alican Akman, Yusuf Sahillioglu, and T Metin Sezgin. Generation of 3d human models and animations using simple sketches. *arxiv*, 2020. 4324

[2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Everybodylovessketch: 3d sketching for a broader audience. In Andrew D. Wilson and François Guimbretière, editors, *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, Victoria, BC, Canada, October 4-7, 2009*, pages 59–68. ACM, 2009. 4322

[3] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160, 2008. 4322

[4] Alexandru O. Balan, Leonid Sigal, Michael J. Black, James E. Davis, and Horst W. Haussecker. Detailed human shape and pose from images. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 4324

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 4327

[6] Mikhail Bessmeltsev, Will Chang, Nicholas Vining, Alla Sheffer, and Karan Singh. Modeling character canvases from cartoon drawings. *ACM Trans. Graph.*, 34(5), Nov. 2015. 4324

[7] Mikhail Bessmeltsev, Nicholas Vining, and Alla Sheffer. Gesture3d: Posing 3d characters via gesture drawings. *ACM Trans. Graph.*, 35(6), Nov. 2016. 4324

[8] Péter Borosán, Ming Jin, Doug DeCarlo, Yotam Gingold, and Andrew Nealen. Rigmesh: Automatic rigging for part-based shape modeling and deformation. *ACM Trans. Graph.*, 31(6), Nov. 2012. 4324

[9] Kirill Brodt and Mikhail Bessmeltsev. Sketch2pose: Estimating a 3d character pose from a bitmap sketch. *ACM Transactions on Graphics*, 41(4), 7 2022. 4322, 4326, 4327, 4328

[10] Angel X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Qixing Huang, Zimo Li, S. Savarese, M. Savva, Shuran Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015. 4322

[11] James Davis, Maneesh Agrawala, Erika Chuang, Zoran Popović, and David Salesin. A sketching interface for articulated figure animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, page 320–328, Goslar, DEU, 2003. Eurographics Association. 4323

[12] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, page 848–855, New York, NY, USA, 2003. Association for Computing Machinery. 4322

[13] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A. Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(1), July 2018. 4323

[14] Marek Dvorožňák, Saman Sepehri Nejad, Ondřej Jamriška, Alec Jacobson, Ladislav Kavan, and Daniel Sýkora. 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*, Expressive '18, New York, NY, USA, 2018. Association for Computing Machinery. 4322

[15] Marek Dvorožňák, Daniel Sýkora, Cassidy Curtis, Brian Curless, Olga Sorkine-Hornung, and David Salesin. Monster mash: A single-view approach to casual 3d modeling and animation. *ACM Trans. Graph.*, 39(6), Nov. 2020. 4321, 4324, 4326, 4327

[16] Even Entem, Loic Barthe, Marie-Paule Cani, Frederic Cordier, and Michiel van de Panne. Modeling 3d animals from a side-view sketch. *Comput. Graph.*, 46(C):221–230, Feb. 2015. 4322

[17] Yotam Gingold, Takeo Igarashi, and Denis Zorin. Structured annotations for 2d-to-3d modeling. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, New York, NY, USA, 2009. Association for Computing Machinery. 4322

[18] Peng Guan, Alexander Weiss, Alexandru Balan, and Michael Black. Estimating human shape and pose from a single image. pages 1381–1388, 09 2009. 4324

[19] 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. 4325, 4326

[20] 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. 4325, 4326

[21] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A sketching interface for 3d freeform design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 409–416, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 4321, 4322

[22] Eakta Jain, Yaser Sheikh, and Jessica Hodgins. Leveraging the talent of hand animators to create three-dimensional animation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 93–102, 2009. 4323

[23] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 4324

[24] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019. 4324

[25] Vladislav Kraevoy, Alla Sheffer, and Michiel van de Panne. Modeling from contour drawings. In *Proceedings of the 6th*

Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '09, page 37–44, New York, NY, USA, 2009. Association for Computing Machinery. 4322

[26] Stan Lee and John Buscema. *How to draw comics the Marvel way*. Simon and Schuster, 1984. 4321, 4324, 4328

[27] Zohar Levi and Craig Gotsman. Artisketch: A system for articulated sketch modeling. *Computer Graphics Forum*, 32(2pt2):235–244, 2013. 4324

[28] Bo Li, Yijuan Lu, Afzal Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose Saavedra, and Shoki Tashiro. Shrec'13 track: Large scale sketch-based 3d shape retrieval. *EG 3DOR 2013*, 2013, 01 2013. 4322

[29] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. Sketch2cad: Sequential cad modeling by sketching in context. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 4323

[30] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: Modeling freeform surfaces through 2d sketching. *ACM Trans. Graph.*, 36(4), July 2017. 4323

[31] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 4323

[32] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 4324

[33] D. Liu, M. Nabail, A. Hertzmann, and E. Kalogerakis. Neural contours: Learning to draw lines from 3d shapes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5427–5435, 2020. 4322

[34] Runtao Liu, Qian Yu, and S. Yu. An unpaired sketch-to-photo translation model. *ArXiv*, abs/1909.08313, 2019. 4322

[35] 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. 4322, 4324

[36] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pages 67–77. IEEE, 2017. 4322

[37] Chen Mao, Sheng Feng Qin, and David K Wright. Sketching-out virtual humans: from 2d storyboarding to immediate 3d character animation. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, pages 76–es, 2006. 4323

[38] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978. 4324

[39] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. 4327

[40] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: Designing freeform surfaces with 3d curves. *ACM Trans. Graph.*, 26(3):41–es, jul 2007. 4322

[41] L. Olsen, F. Samavati, and J. Jorge. Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications*, 31:24–34, 2011. 4322

[42] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. *2018 International Conference on 3D Vision (3DV)*, pages 484–494, 2018. 4324

[43] 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. 4326

[44] Lakshman Prasad. Morphological analysis of shapes. *CNLS newsletter*, 139(1):1997–07, 1997. 4324

[45] Alec Rivers, Frédo Durand, and Takeo Igarashi. 3d modeling with silhouettes. *ACM Trans. Graph.*, 29(4), jul 2010. 4322

[46] 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. 4324, 4325, 4326

[47] Leonid Sigal, Alexandru Balan, and Michael J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1337–1344, Red Hook, NY, USA, 2007. Curran Associates Inc. 4324

[48] Dmitriy Smirnov, Mikhail Bessmeltsev, and Justin Solomon. Learning manifold patch-based representations of man-made shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4322

[49] Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, and Kui Jia. Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 4322

[50] Anh Thai, Stefan Stojanov, Vijay Upadhya, and James M Rehg. 3d reconstruction of novel object shapes from single images. *arXiv preprint arXiv:2006.07752*, 2020. 4325

[51] Matthew Thorne, David Burke, and Michiel van de Panne. Motion doodles: An interface for sketching character motion. *ACM Trans. Graph.*, 23(3):424–431, Aug. 2004. 4323

[52] Bastien Wailly and Adrien Bousseau. Line rendering of 3d meshes for data-driven sketch-based modeling. In *Journées Francaises d'Informatique Graphique et de Réalité virtuelle*, 2019. 4323, 4326, 4327, 4328

[53] Jiayun Wang, Jierui Lin, Qian Yu, Runtao Liu, Yubei Chen, and Stella X Yu. 3d shape reconstruction from free-hand sketches. *arXiv preprint arXiv:2006.09694*, 2020. 4322

[54] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225, 2021. 4324

[55] Zhong Yue, Gryaditskaya Yulia, Zhang Honggang, and Song Yi-Zhe. Deep sketch-based modeling: Tips and tricks. In *Proceedings of International Conference on 3D Vision (3DV)*, 2020. 4322, 4325