Sketch2Pose: Estimating a 3D Character Pose from a Bitmap Sketch

KIRILL BRODT and MIKHAIL BESSMELTSEV, Université de Montréal, Canada

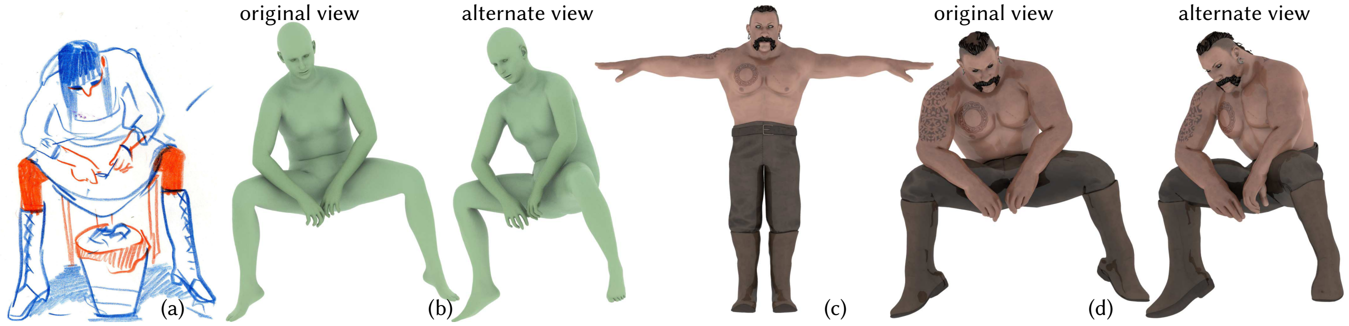

Fig. 1. Given a single natural **bitmap** sketch of a character (a), our learning-based approach allows to automatically, with no additional input, recover the 3D pose consistent with the viewer expectation (b). This pose can be then automatically copied a custom rigged and skinned 3D character (c) using standard retargeting tools (d). Input image © Olga Posukh.

Artists frequently capture character poses via raster sketches, then use these drawings as a reference while posing a 3D character in a specialized 3D software — a time-consuming process, requiring specialized 3D training and mental effort. We tackle this challenge by proposing the first system for automatically inferring a 3D character pose from a single bitmap sketch, producing poses consistent with viewer expectations. Algorithmically interpreting bitmap sketches is challenging, as they contain significantly distorted proportions and foreshortening. We address this by predicting three key elements of a drawing, necessary to disambiguate the drawn poses: 2D bone tangents, self-contacts, and bone foreshortening. These elements are then leveraged in an optimization inferring the 3D character pose consistent with the artist's intent. Our optimization balances cues derived from artistic literature and perception research to compensate for distorted character proportions. We demonstrate a gallery of results on sketches of numerous styles. We validate our method via numerical evaluations, user studies, and comparisons to manually posed characters and previous work.

Code and data for our paper are available at http://www-labs.iro.umontreal.ca/ bmpix/sketch2pose/.

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**.

Additional Key Words and Phrases: character posing, rigged and skinned characters, sketch-based posing, character sketches

ACM Reference Format:
Kirill Brodt and Mikhail Bessmeltsev. 2022. Sketch2Pose: Estimating a 3D Character Pose from a Bitmap Sketch. *ACM Trans. Graph.* 41, 4, Article 85 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530106

Authors' address: Kirill Brodt, kirill.brodt@umontreal.ca; Mikhail Bessmeltsev, bmpix@iro.umontreal.ca, Université de Montréal, 2900 Edouard Montpetit Blvd, Montréal, QC, H3T 1J4, Canada.

0730-0301/2022/7-ART85 $15.00
https://doi.org/10.1145/3528223.3530106

1 INTRODUCTION

“Art is not what you see, but what you make others see.“

— Edgar Degas

Artists routinely capture human poses via diverse drawings, from quick gestures to detailed character sketches. Sketching character poses is one of the core elements in artist training. In modern digital media production, artists often draw sketches of characters at the early *storyboarding* stage. Often drawn within tens of seconds each, those sketches serve as efficient and direct means to capture ideas and convey the poses to other team members. These natural, freely drawn sketches are then often used as a reference while manually posing a 3D character in an animation software.

The manual posing step, however, requires special training, and is tedious, time-consuming, taking up several minutes per pose even for a rough draft. For professionals, it is a frustrating task that may distract from the creative process and slow down the production pace; for classically trained artists with little knowledge of 3D animation software, it requires specialized training and thus may become an obstacle to implementing their ideas.

A system enabling artists to directly use a sketch as the only input to automatically pose a rigged and skinned 3D character thus would significantly simplify and democratize posing 3D characters, benefiting both novice and professional users. Creating such a system, unfortunately, faces significant challenges: First, contrary to the assumptions of previous work that targeted clean vector drawings of a single style [Bessmeltsev et al. 2016], sketches are often drawn on traditional media, such as pen and paper, or in a raster drawing software, are stylistically various, and are full of construction lines and extra strokes. Converting those images into clean vectorized drawings remains an open problem [Stanko et al. 2020]. More importantly, sketches are imprecise, incomplete, often contain occlusions, and may substantially distort character's proportions, whether due to errors [Schmidt et al. 2009] or artistic license: As the Degas's quote can be interpreted, drawings are not created to be a perfect depiction of reality, but rather a means to convey an idea to a human observer.

We propose the first framework addressing all those challenges. We introduce a system that algorithmically computes a complete 3D character pose given a single *raster* sketch of a character. Our system supports a variety of sketch styles, including gesture drawings (see e.g., Fig. 5, Fig. 7), contour drawings (e.g., Fig. 12, top), and complex detailed sketches (e.g., Fig. 1). Our framework successfully handles complex incomplete sketches with distorted body proportions and occlusions (e.g., Fig. 1, Fig. 4). We thus enable artists to pose a 3D character directly and automatically from the sketches they draw, enhancing and possibly simplifying the current media production pipeline. Our system allows artists without specialized 3D training to pose 3D characters using only their drawings.

The key challenge in the problem of posing 3D digital characters via a sketch is inferring the *artist-intended* 3D pose from the 2D drawing. While human observers generally have no problem imagining a consistent 3D pose from a drawing, mathematically the task is highly ambiguous. Similar to the well-known computer vision problem of inferring a human pose from a photograph, we also face the fundamental ambiguity of reconstructing 3D content from 2D. For photographs, this ambiguity is typically resolved by assuming that the image is a *projection* of a human pose onto the screen. Unlike photographs, however, sketches cannot be interpreted as projections of the 3D character onto the view plane (Fig. 5): drawn with a goal to capture and convey the essence of the pose, they are only approximate depictions of the character.

More specifically, drawings often significantly distort body proportions or depict characters with unrealistic body shapes (Fig. 4). While proportions are often distorted even for sketches of static poses, such distortion is a core and intentional component of gestures of dynamic poses [Kwon and Lee 2012]. Furthermore, artists use nonlinear [Singh 2002], grossly exaggerated or otherwise inaccurate perspective [Schmidt et al. 2009; Sudarsanam et al. 2008; Zhong et al. 2020], and incorrectly depict foreshortening (Fig. 6) [Wnuczko et al. 2016]. Naively ignoring these issues and applying standard optimizations relying on 2D joint positions only, may lead to wrong or imprecise results (Fig. 4).

We center our analysis on three elements of a character drawing that we believe are essential to resolving these issues: bone tangents, self-contacts, and foreshortening (Fig. 2). We observe that while, as outlined above, 2D joint positions themselves are unreliable, 2D bone tangents can be strong indicators of intended 3D bone direction. We further observe that depicted self-contacts (e.g., forearms and thighs in Fig. 1) are critical for understanding of a drawn pose and can be strong cues to disambiguate the unknown body part depth. Finally, we explicitly model and undo the distortions of the bone foreshortening using statistical analysis.

Overview. We introduce a novel system for inferring a 3D character from a single bitmap based a combination of optimization, deep learning, statistical analysis, and observations from perceptual and artistic literature. Our optimization is guided by three main subsystems predicting 2D bone tangents, self-contacts, and bone foreshortening. Equipped with these three predictions, we use a state-of-the-art optimization framework with a novel loss designed specifically to compensate for the inaccuracies in the natural drawings. Our optimization balances the pose realism against the image cues, while allowing body shape to change. This optimization enables our system to infer complex 3D poses with significantly distorted body lengths and proportions (Sec. 7). We infer the pose of a parameterized human model SMPL-X [Pavlakos et al. 2019], which can be automatically transferred to a custom character via standard animation software or modern retargeting systems [Aberman et al. 2020].

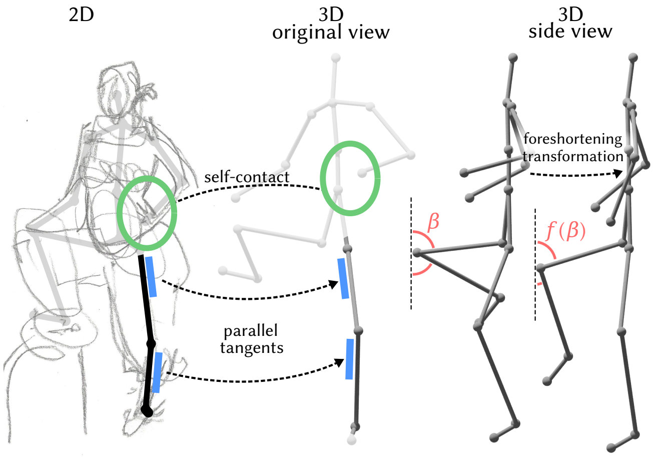

Fig. 2. Our analysis is centered around three elements: bone tangents, self-contacts, and foreshortening. In particular, we aim to preserve 2D bone tangents from the original view when computing the 3D pose and keep the relative positions of joints participating in a self-contact. Finally, we model and undo the distortions of the depicted bone foreshortening, adaptively reducing angles between the bones and the screen, as compared to a naive reconstruction. Input image © Olga Posukh.

Contribution. Our contribution is two-fold:

- We present the first large-scale dataset of 2D pose annotations on character sketches. Our dataset includes 14,462 skeletons, each consisting of 18 manually annotated 2D labels with locations of joints. Around 1000 images also contain 2D locations of self-contacts. In addition, we have collected and annotated a smaller dataset containing 310 high-resolution raster character sketches, collected from a variety of artists highlighting different styles and characters, that we share with the most permissive usage license (CC-BY).
- More importantly, we present the first framework that algorithmically reconstructs a 3D character pose directly from a single natural sketch. Our framework supports many sketch styles, including gesture drawings.

We validate our system in a number of ways (Sec. 7). First, we present a gallery of 3D character poses computed automatically and without any additional input from natural sketches. Second, we perform user studies demonstrating the efficacy of our method. Finally, we quantitatively and qualitatively compare our algorithmic results to manually posed characters and previous work.

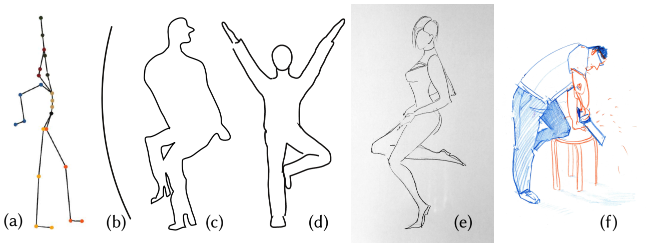

Fig. 3. The previous posing approaches were constrained to working with either ambiguous inputs, such as stick figures [Davis et al. 2003; Hecker and Perlin 1992; Lin et al. 2010; Mao et al. 2005] (a), lines of actions [Guay et al. 2013, 2015] (b), or silhouettes [Won and Lee 2016] (c); or unambiguous, but clean vector curve drawings [Bessmeltsev et al. 2016] (d), which are hard if possible to obtain automatically from the raster drawings artists create. Our framework allows to pose 3D characters *directly* via natural bitmap character sketches of different styles, including rough gesture drawings (e) and detailed character sketches (f), containing inaccuracies, ambiguities, extra strokes, and rudimentary shading. We furthermore show that our method works for rasterized clean vector drawings (d) explored in previous works. Input image (f) © Olga Posukh.

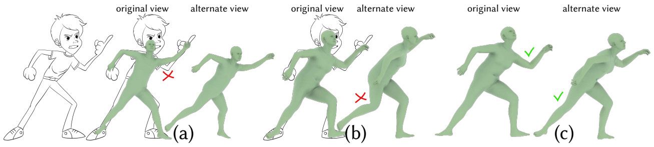

Fig. 4. Distinguishing body part foreshortening, i.e. intentional out-of-plane rotation, from atypical or inaccurate body proportions when no out-of-plane rotation is intended, is ambiguous. An underestimation of body part length, when an exact solution satisfying the 2D keypoints does not exist, may lead to an inexpressive least-squares solution (a, see e.g. the character's left elbow); an overestimation leads to an unintended foreshortening (b, see e.g. the character's right knee). Our method addresses this ambiguity (c).

2 RELATED WORK

To our knowledge, no algorithm is capable of inferring a 3D character pose from a natural raster character sketch. The two closest areas of relevant previous work are character posing interfaces and human pose inference from an RGB photograph. We focus on the most relevant works.

2.1 Character Posing Interfaces

A common technique to pose a 3D character is via time-consuming direct manipulation of joint angles (Forward Kinematics, FK) or joint positions (Inverse Kinematics, IK) [Zhao and Badler 1994]. The problem of IK is inherently underdetermined. In their pioneer work, Grochow et al. [2004] address this issue via a Gaussian process–based model trained on a motion capture dataset, requiring exact and feasible positions of terminal joints.

Previous research explored a various range of alternative inputs to a posing system, including stick figures [Davis et al. 2003; Hecker and Perlin 1992; Lin et al. 2010; Mao et al. 2005], static or dynamic lines of action [Guay et al. 2013, 2015], silhouettes [Won and Lee 2016], custom sketch strokes [Hahn et al. 2015], tangible devices [Glauser et al. 2016], and clean vector drawings [Bessmeltsev et al. 2016].

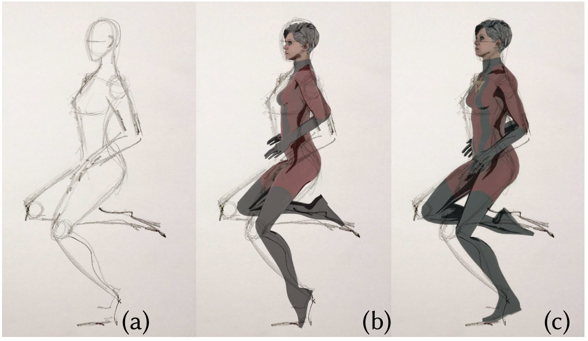

Fig. 5. Unlike photographs, character sketches cannot be interpreted as projections of a 3D character onto the screen. Left: input sketch, middle: a character posed manually by an expert, right: our automatic result.

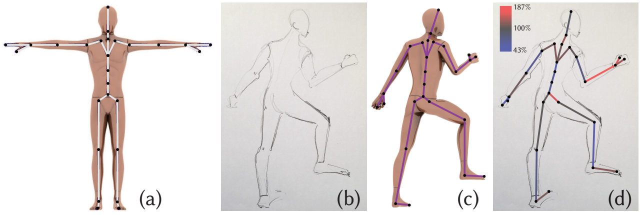

Fig. 6. One of the main sources of distortions in character sketches is unreliably depicted body part lengths. In contrast to a perfect projection (c, manually posed by an expert given image (b) as a reference), artist routinely draw bones longer (d, red), often exceeding full their 3D length, or shorter (d, blue) than their correct projection. In (d) we color the bones based on the ratio of their depicted length and their correct 2D projection length in (c), from blue to red.

Stick figures (Fig. 3a) and silhouettes (Fig. 3b) [Won and Lee 2016] are inherently ambiguous even for human observers [Bessmeltsev et al. 2016]. This ambiguity is often resolved via manual annotation [Davis et al. 2003; Hecker and Perlin 1992; Mao et al. 2005], physical constraints [Lin et al. 2010], or by putting user in the loop [Davis et al. 2003]. Some works restrict output [Jain et al. 2012] or measure proximity to human pose datasets [Choi et al. 2012; Wei and Chai 2011]. These methods are sensitive to inaccurate positioning of 2D stick figures [Davis et al. 2003]. Character sketches are imprecise yet drawn to unambiguously convey a pose to a human observer and thus contain the necessary cues for a reconstruction, which we analyze and incorporate in our optimization, allowing us to overcome the drawing distortions and inaccuracies.

In contrast to alternative posing interfaces, such as tangible devices [Glauser et al. 2016], multi-view incremental approaches [Guay et al. 2013], or sketch abstractions [Hahn et al. 2015], we infer a 3D pose directly from a single natural character sketch. Our system thus allows artists to convert existing drawings, for instance, the storyboards typically created during the ideation and planning stage, into 3D poses without spending extra effort on posing.

Choi et al. [2016] introduce a motion editing system via sketch strokes, requiring an existing motion of a character as an input. Their system is complementary to ours, as we focus on reconstructing a static pose from a single natural sketch.

Gesture3D [Bessmeltsev et al. 2016] reconstructs a character pose from a clean vector drawing (Fig. 3d), assuming no extra strokes and precise connectivity (e.g. T-junctions). Sketches found in the wild are often rich with extra strokes, shading elements, and imprecise connections (Fig. 3e, f) and often cannot be automatically vectorized to that precision [Stanko et al. 2020]. Our system directly accepts such sketches as input. Furthermore, they minimize foreshortening, assuming "flat" 3D poses where each body part is nearly parallel to the screen; we explicitly predict foreshortening, lifting that assumption. We compare to Gesture3D in Section 7.

2.2 Human Pose Estimation from a Single Photograph

The problem of inferring a 3D human pose from a single RGB photograph, or monocular pose estimation, is an extensively studied topic in computer vision. Traditional approaches [Chen et al. 2011; Gall et al. 2010; Ionescu et al. 2014; Ramanan 2011; Sapp et al. 2010; Yang and Ramanan 2013] relied on custom image-based features and often used traditional machine learning techniques. Those approaches have been largely superseded by the deep learning-based approaches. Here we only outline the most relevant works. For a survey, see, e.g. Chen et al. [2020].

3D Pose Estimation. Many learning-based approaches predict 3D human poses relying on large 3D datasets with the corresponding images [Pavlakos et al. 2017; Rogez et al. 2017; Tekin et al. 2016; Tomè et al. 2017; Toshev and Szegedy 2014; Zhou et al. 2016] or combining those with 2D in-the-wild pose datasets [Mehta et al. 2017a; Tekin et al. 2017; Yang et al. 2018; Zhou et al. 2017]. This line of work has been extended by enforcing skeletal consistency [Mehta et al. 2017b; Shi et al. 2020; Sun et al. 2017], joint constraints [Akhter and Black 2015; Mehta et al. 2020], or bone lengths [Dabral et al. 2017; Ramakrishna et al. 2012; Wang et al. 2019a]. All these methods require a significant amount of 3D labeled data. For our task, we would need thousands sketches and their corresponding 3D poses; no such dataset exists.

Some works sidestep this dependency on full 2D image – 3D skeleton annotations by using unpaired 2D-3D data [Tung et al. 2017] or by relying on well-established methods in 2D pose estimation [Cao et al. 2019; Carreira et al. 2016; Chen et al. 2018; Newell et al. 2016; Papandreou et al. 2017] and focusing on the 3D lifting in a supervised [Martinez et al. 2017] or self-supervised manner [Novotný et al. 2019]. trained in a fully supervised manner, Supervised learning is infeasible in our context; unsupervised learning would require a model of how 3D joints get projected onto 2D labels. As discussed in Sec. 1, character sketches cannot be interpreted as perfect projections of 3D characters, but rather are artistic depictions of those. It is thus unclear how to define such projection models that support incorrectly drawn perspective and distortions of body proportions, typical for character sketches. Our system targets to infer the pose consistent with the *artist intent*, despite the distortions and ambiguities (Fig. 4c).

3D Shape Estimation. We are inspired by an alternative line of work that predicts pose and shape of a human simultaneously [Bogo et al. 2016; Dwivedi et al. 2021; Kanazawa et al. 2018; Madadi et al. 2018; Pavlakos et al. 2019, 2018; Xiang et al. 2019; Xu et al. 2019]. They represent the shape and pose via a parametric model of human body shapes, such as Skinned Multi-Person Linear Model (SMPL) [Loper et al. 2015] or Adam [Joo et al. 2018]. Madadi et al. [2018] predict SMPL parameters via first inferring 3D heatmaps, which are then processed via a denoising autoencoder and fed into an MLP. Kanazawa et al. [2018] regress SMPL parameters using weak supervision: they match the 2D keypoints locations and capitalize on the known structure of SMPL space that allows for efficient compact natural pose discriminators. Kolotouros et al. [2019] directly regress SMPL parameters on labels obtained with optimization-based approach of Bogo et al. [2016]. Joo et al. [2021] further improve results via using a fine-tuning framework based on a regression model. Müller et al. [2021] focus on predicting 3D shape and poses with self-contacts, using either SPIN [Kolotouros et al. 2019] or EFT [Joo et al. 2021] as the underlying framework. A recent continuation of this line of work by Dwivedi et al. [2021] focuses on clothed people. Specifically, they introduce a differentiable semantic rendering loss that distinguishes between clothed and mininally-clothed regions. Our self-contact detection uses the loss from Müller et al. [2021]. In contrast to their work that relies on 3D pose estimation to predict self-contacts, we predict the depicted self-contacts directly from the input image and map those onto the 3D model (Sec. 4.3). Its contemporaneous work [Fieraru et al. 2021] similarly relies on a dataset containing mesh regions in contact. Our dataset only contains image-space locations of self-contacts — more ambiguous yet easier to collect. We use the framework of Joo et al. [2021] with [Kolotouros et al. 2019]. Our work, however, differs in two important aspects. First, [Joo et al. 2021] target reconstructing the 3D pose such that predicted 2D labels are projections of the 3D joints — a natural requirement for photographs. Our goal, however, is different: we aim to predict the artist-intended 3D pose whose projection may deviate significantly from the drawn sketch – this is reflected in our novel loss formulation (Sec. 5) that includes explicitly predicting bone foreshortening (Sec. 4.4). We compare with those reprojection-based approaches in Figs. 4, 14, Sec. 7, and Supplementary Materials.

3 KEY PRINCIPLES AND OVERVIEW

Even a simple task of reconstructing a 3D skeleton given 2D *projections* of its joints is ill-posed, as formally there is an infinite number of solutions. For character sketches, however, the problem of reconstructing 3D pose is more ambiguous due to the distorted proportions, perspective, and foreshortening. In order to infer the artist-intended pose, we distill the knowledge in drawing literature, as well as perception and modeling research, to formulate the observations true across a wide variety of sketch styles. These observations guide our algorithmic choices.

3.1 Key Principles

Foreshortening. Artistic depiction of foreshortening is often far from accurate (Fig. 6). While drawing, artists do not use precise

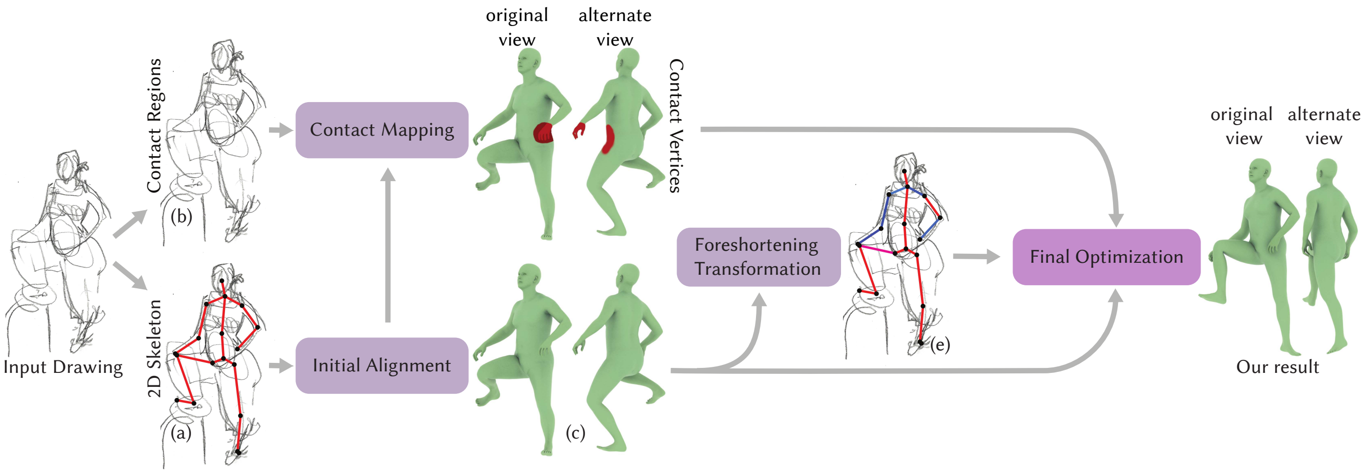

Fig. 7. Starting with an input drawing, we first predict 2D joint positions, or 2D skeleton, which is used in the initial rough alignment of a 3D human model. We then predict screen-space contact regions, which we map onto the roughly aligned 3D model, resulting in a set of contact vertices (in red). We compensate for inaccuracies in depicting bone lengths in Foreshortening Transformation stage. Finally, we leverage the bone tangents of the 2D skeleton, the roughly aligned 3D pose, as well as the transformed foreshortening in an optimization framework yielding the final result. Input image © Olga Posukh.

mathematical measurements for orthographic or perspective projection, instead relying on their experience and rules of thumb [Hogarth 1996; Walt Stanchfield 2020]. Naively reconstructing a 3D pose with the assumption that the drawn foreshortening is exact, i.e. that 2D bones are a projection of 3D bones, often leads to a grossly inaccurate prediction of the angles the bones make with the screen (Fig. 10a). Previous sketch-based modeling literature relied on the assumption of *minimal* foreshortening, i.e. that the characters are drawn from a viewpoint where all body parts are nearly parallel to the screen [Bessmeltsev et al. 2016]. Both of these assumptions, however, are generally incorrect for a character drawing: Artists do depict shorter limbs as an indicator of bone foreshortening, but often exaggerate the effect [Walt Stanchfield 2020].

Predicting true angles the body parts make with the screen is further impeded by two main factors: First, since cartoon characters often have unrealistic or heavily distorted proportions, there is a fundamental ambiguity whether the shorter depicted length indicates different character proportions or foreshortening. Second, artists often use non-linear perspective [Singh 2002], so inferring camera parameters from an input image is ill-posed. We propose a statistics-based solution that predicts bone foreshortening under orthographic projection, thus compensating for artist inaccuracies (Fig. 9(c)).

Tangent Significance. For orthographic or perspective cameras, the length of a bone's 2D *projection* would never exceed its full length. For sketches, however, this is not so. Character bones often extend beyond their normal length on the drawings due to drawing inaccuracies or artistic license [Hogarth 1996; Thomas and Johnston 1981; Walt Stanchfield 2020]. In those cases, even when character's body proportions are known, an exact 3D configuration for the given projection does not exist, and the least-squares solution fails to match the expressiveness of the drawing (Fig. 4a).

While unreliable depiction of bone lengths invalidates the direct use of absolute joint positions, artistic literature repeatedly stresses the importance of correct depiction of joint angles. Angles are considered one of the key elements of the drawing, creating pose expressiveness and dynamism [Walt Stanchfield 2020]. We speculate therefore that in interpreting a drawn pose, human observers resolve the inaccuracies in absolute positions by relying on correctly drawn angles: both joint angles and bone tangents, i.e. angles they form with the coordinate axes. We therefore aim to preserve bone tangents, i.e. their angles with the coordinate axes. In other words, we expect the 3D bone projection to be parallel to the depicted bones in 2D, subject to regularity cues. Clearly, this also guides the reconstructed 3D joint angles to have similar projections to the depicted 2D joint angles.

Perceived self-contacts. Self-contacts, or contacts between different body parts, are key elements of many poses [Hogarth 1996]. Depending on a drawing, self-contacts may be explicitly drawn (Fig. 11, left) or somewhat ambiguously suggested (e.g. Fig. 7). We speculate that human observers use perceived self-contacts as one of the cues to resolve depth ambiguity, associating similar depths to touching body parts. Clearly, for a 3D character pose to be similar to the drawing in the original view, the depicted self-contacts must be preserved, regardless of the difference in character's proportions. We therefore predict perceived contacts between different body parts using a neural network and enforce those during optimization. For each predicted contact in the drawing, we consider the participating body parts, and both preserve their 2D relative positions at the point of contact, as well as enforce true 3D contacts between them.

Pose Naturalness and Regularity. Finally, we observe, consistently with the previous work [Bessmeltsev et al. 2016; Xu et al. 2014], that human observers rely on Gestalt simplicity cues [Koffka 1955] in interpreting drawings. We speculate that given the approximate nature of character sketches, viewers use *regularity* cues such as symmetry and parallelism, as well as expect the pose to be close to natural. We leverage regularity as one of the cues in our optimization and use an existing framework biasing the result towards natural poses [Joo et al. 2021].

3.2 Algorithm Overview

Given a single bitmap sketch of a character in a target pose (Fig. 7, left), our algorithm automatically infers a parametric human model SMPL [Loper et al. 2015] in the depicted pose (Fig. 7, right). The pose can then be transferred automatically onto a custom 3D character via standard animation software, such as Autodesk Maya or Blender, or via more advanced modern retargeting methods [Aberman et al. 2020].

We first predict three key elements of a character drawing: 2D bone tangents, body part contacts, and bone foreshortening. We use convolutional networks to predict 2D locations of main joints and image-space body part contacts; we then map the latter onto the 3D mesh (Sec. 4). We then use a nonlinear optimization using standard position-based L_2 reprojection loss to get a rough estimate of the pose, which we use to estimate foreshortening factor for each bone (Sec. 4.4) and contact vertices (Sec. 4.3). Finally, we use the three key elements in the nonlinear optimization with a novel loss that balances the perceptual cues, pose naturalness, and similarity to the input drawing, producing the final result (Sec. 5).

4 INFERRING KEY ELEMENTS OF A DRAWING

In the first stage of our algorithm, we infer the three elements of a drawing that we believe are key to its interpretation: 2D bone tangents, body part contacts, and bone foreshortening.

4.1 2D Joint Positions

We predict the 2D positions of the most important skeletal joints and rely on our final optimization (Sec. 5) to reconstruct the full 3D pose. In total, we predict 2D positions of $K = 18$ main joints (4 joints for each leg, 3 for each arm, 3 joints for the torso, 1 joint for the head).

To this end, we train a top performing deep convolutional 2D pose estimation network [Sun et al. 2019; Wang et al. 2019b; Xiao et al. 2018] on the dataset of sketches with their 2D skeletal annotations we collected (Sec. 6). We resize the input drawing preserving the aspect ratio and pad them to the resolution of 384x288 pixels. The network outputs a 96x72 pixel heatmap for each of the K joints, showing the per-pixel confidence score of the chosen joint location. The joint position is then taken as the maximum point on the heatmap. For the details on architecture please refer to the original paper [Sun et al. 2019].

The output of this stage of our algorithm is the 2D positions $\hat{x}_j^{2D}$ of K skeletal joints in the image coordinate frame (Fig. 7a). 2D bone tangents are then defined as differences of those 2D positions.

4.2 Initial Alignment of the 3D Model

We leverage these 2D positions in an initial optimization that produces a SMPL model roughly aligned with the drawing (Fig. 7c) via a state-of-the-art framework Exemplar Fine-Tuning (EFT) [Joo et al. 2021]. Starting with a pre-trained network performing regression from the input image into the space of the SMPL parameters [Kolotouros et al. 2019], EFT optimizes the weights of the network to minimize the L_2-reprojection distance between 2D joint positions $\hat{x}_j^{2D}$ and 2D orthographic projections x_j^{2D} of the 3D SMPL joints:

$$E_{2D} = \sum_j w_j \|\hat{x}_j^{2D} - x_j^{2D}\|^2. \tag{1}$$

We train the regression network on the poses produced by Pavlakos et al. [2019], which uses a pose naturalness prior. We thus inherit naturalness of poses as an implicit prior. For all details, please refer to the original paper [Joo et al. 2021]. We run their method for 150 iterations, with default parameters. The output of this stage is a set of 85 parameters encoding a human in a pose roughly similar to the drawing, including 24 × 3 parameters for body pose, 10 for body shape, 2 for camera 2D translation and 1 for uniform scale. The input RGB images are 224 × 224px.

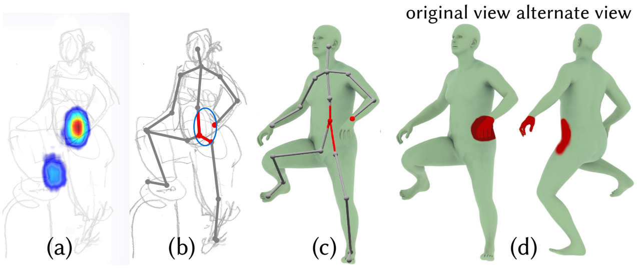

Fig. 8. To detect self-contacts, we first predict a self-contact heatmap (a), which we first threshold and split into connected components, or *contact regions* (b). We then overlap each contact region with the 2D skeleton and mark the corresponding 3D SMPL skeleton segments (c). Finally, we use SMPL skinning to find all the mesh vertices corresponding to these bone segments (d).

4.3 Detecting Self-Contacts

We then detect depicted self-contacts in the image space and map those onto the vertices of the roughly aligned mesh (Fig. 8). The vertices will be then used in the final optimization that enforces contacts between some of them (Sec. 5).

Our sketch dataset contains 2D positions of perceived self-contacts. We use it to train a 2D contact prediction network, outputting 2D contact heatmaps. The network has the same architecture as in Sec. 4.1.

The self-contact heatmap predicts areas of potential contacts in the image space. We first need to filter out noise and separate different *contact regions*, which we map to separate groups of SMPL mesh vertices that each should have at least one pair of vertices touching. To this end, we first threshold the self-contact heatmap with a conservative threshold of 0.5, and compute the connected components over the thresholded heatmap, forming contact regions.

Our next step is mapping each contact region onto the vertices of the roughly aligned SMPL mesh. Note that a straightforward approach of simply projecting each connected component onto the

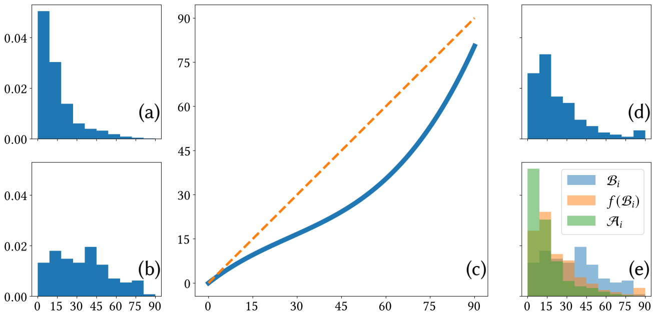

Fig. 9. In the foreshortening transformation stage, we design a function (c) that transforms the bone-screen angles after the initial optimization closer to the ground-truth angles. To this end, we find the transformation bringing the distribution of such angles after the initial optimization $\mathcal{B}$ (b) closer to the distribution of ground-truth angles $\mathcal{A}$ (a), yielding the transformed angle distribution $f(\mathcal{B})$ (d). The final angles are then used to compute the target foreshortening of each body part used in the final optimization. Here we display the histograms for the left thigh bone.

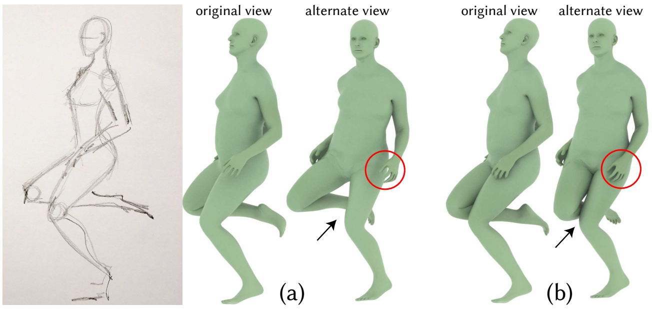

Fig. 10. Without the foreshortening transformation (a), distortions in the depicted body part lengths often lead to unexpected angles with the screen. We correct these distortions (b).

mesh may lead to suboptimal results, since the mesh often deviates significantly from the drawing (Fig. 8c). Instead, we use the predicted 2D skeleton as a proxy to find this mapping. We first compute convex hull of each contact region for robustness, then intersect each hull with the 2D skeleton, forming a set of 2D bone segments (Fig. 8b). We then use the linear parameterization of each bone to transfer the segments in contact onto the 3D SMPL skeleton (Fig. 8c), and finally mark all mesh vertices skinned to these 3D segments as contact vertices for this contact region (Fig. 8d).

4.4 Foreshortening Transformation

In this step, our goal is to compensate for the distortions in the depicted foreshortening, introduced by artist inaccuracies, exaggerated perspective, and proportions mismatch (Fig. 10a). The output of this step, bone foreshortening under orthographic projection, will inform the angles between the 3D bones and the screen (Fig. 10b).

A straightforward solution would be to use a ground-truth dataset with correspondences between depicted 2D poses and intended 3D poses; such dataset, unfortunately, does not exist. Instead, our insight is that while the correspondences are unknown, the reconstructed and intended angles the bones make with the screen should be equally distributed.

As a proxy for the unknown distribution of the intended angles, we take the distribution $\mathcal{A}_i$ of angles each bone i in a motion capture dataset [Mahmood et al. 2019] makes with an appropriate view plane (Fig. 9a). Ideally, the choice of the view planes should capture the drawing angles the artists choose for a given pose; as an approximation, we use the dataset's default camera plane. Note that while this computation can easily be extended to multiple view planes, we did not find it necessary, since the dataset already provides enough pose variety even from the default viewpoint. We then compute the distribution $\mathcal{B}_i$ of angles each bone i makes with the screen after performing rough optimization (Sec. 4.2) for all the images in our drawing dataset (Fig. 9b).

Our goal is now to find a function that, for each bone i, transforms the reconstructed angles $\beta_i \sim \mathcal{B}_i$ such that their distribution matches $\mathcal{A}_i$ as closely as possible. As noted in Wnuczko et al. [2016], the accuracy in observer's perception of 3D directions seems to vary with the foreshortening angle; we conjecture the same is true for artists depicting foreshortened 3D bones. We furthermore observe that (1) artists seem to exaggerate perspective for foreshortened lines; (2) even when intending no foreshortening, artists often draw slightly shorter bones due to inaccuracies or weak but inaccurate perspective (e.g. Fig. 6). Guided by these observations, we first represent the distributions of angles $\mathcal{A}_i$ and $\mathcal{B}_i$ via histograms with 10 equal bins, from 0° to 90° (Fig. 9a, b). We then model the transformation as a cubic polynomial:

$$f(\beta) = a\beta^3 + b\beta^2 + c\beta + d, \tag{2}$$

with initially unknown parameters a, b, c, d, and $\beta \in [0, \pi/2]$. To make the distributions of $f(\mathcal{B}_i)$ and $\mathcal{A}_i$ similar, we find the unknown parameter values by minimizing a sum of Earth Mover's Distances [Rubner et al. 2000] between the two angle distributions for each bone, discretized as histograms:

$$\min_{a,b,c,d \in \mathbb{R}} \sum_i \inf_{\zeta \in \Pi(\mathcal{A}_i, \mathcal{B}_i)} \mathbb{E}_{(\alpha,\beta)\sim\zeta} |\alpha - f(\beta)|, \tag{3}$$

where $\Pi(\mathcal{A}_i, \mathcal{B}_i)$ is the set of all joint distributions whose marginals are $\mathcal{A}_i$and $\mathcal{B}_i$. As we observed above, foreshortening is typically exaggerated, so we add constraints $0 \leq f(\beta) \leq \beta$. We would like to preserve the bones parallel to the screen to be still parallel after the transformation, i.e. $\beta = 0$, so we set $d = 0$. Minimizing the energy in Eq. 3, we get $a = 0.312, b = -0.448, c = 0.503$.

After this transformation, the angle distributions are better aligned (Fig. 9). This optimization is done only once for the dataset. We have additionally tested other classes of functions (cubic splines and piecewise linear functions). They result in similar, somewhat more complex functions that have little effect on the results, so we chose the cubic function as the simplest option.

At test time, for a given image, after the roughly aligned 3D skeleton is computed (Sec. 4.2), we calculate the angles β_i between each bone and the screen. We note that since SMPL is limited to human proportions, atypical proportions of the depicted character likely cause discrepancies in body shape estimation, and, as a result, in angles β_i. For many characters, we observe that those mismatches can be explained by an atypical scale of upper body relative to the lower body. To alleviate this issue and avoid incorrectly foreshortened bones, we subtract the minimal angle with the screen, computed separately over the upper body and lower body:

$$\beta_i' = \beta_i - \min_{j \in B} \beta_j,$$

where B is the set of upper or lower body bones.

Finally, the predicted foreshortening is represented by $\varphi_i = \cos f(\beta_i')$, the target foreshortening for bone i used in the final optimization.

5 3D POSE OPTIMIZATION

The second stage of our system is the optimization that starts with the roughly aligned pose (Sec. 4.2) and finds the artist-intended 3D pose of the given character. To this end, the optimization leverages the 2D bone tangents, body part contacts, and bone foreshortening computed in the previous stage. As a framework, we use EFT [Joo et al. 2021] which optimizes over the weights w of the regression neural network, initialized by the rough alignment stage (Sec. 4.2). We follow their optimization process (Adam algorithm, default PyTorch parameters, learning rate of 10^{-6}). We perform a fixed number of 60 iterations.

Within that framework, instead of the traditional position-based L_2 reprojection loss (Eq. 1), we propose the following novel loss for our task, based on our principles (Sec. 3.1):

$$\min_{w} E_{\text{parallel}} + \lambda_{\text{f}} E_{\text{f}} + E_{\text{contacts}} + E_{\text{reg}}. \tag{4}$$

We denote $x_j^{3D} \in \mathbb{R}^3, j = 1, \ldots, K$ the 3D joint positions of the SMPL model. Note that these positions are, for a fixed input image, functions of the neural network weights w. For each bone i connecting joints j_1 and j_2, we denote its 3D vector as $b_i^{3D} = x_{j_2}^{3D} - x_{j_1}^{3D}$, and its orthographic projection onto the screen as b_i^{2D}. Details about the individual terms in Eq. 4 are below, in the order they appear:

- Parallelism. Guided by our principle of tangent significance, we favor parallelism between the projected 3D bones and their 2D depictions:

$$E_{\text{parallel}} = \sum_i \left(\frac{b_i^{2D}}{\|b_i^{2D}\|} \cdot n \right)^2,$$

  where n is a normal to the depicted bone $\hat{x}_{j_2}^{2D} - \hat{x}_{j_1}^{2D}$.
- Foreshortening. We use the transformed bone foreshortening calculated in Sec. 4.4 to guide the target length of each bone's projection:

$$E_{\text{f}} = \sum_i (\|b_i^{2D}\| - L_i \varphi_i)^2,$$

  where L_i is the length of the bone i, as estimated by the rough alignment stage (Sec. 4.2). Note that here we use a fixed bone length L_i, as opposed to $\|b_i^{3D}\|$ that can vary during the optimization. In our experiments, we found that otherwise the optimization often exploits that dependency to minimize the energy, adjusting the bone lengths instead of the angles between screen and bones.
- Contacts. For each set of contact vertices computed in Sec. 4.3 (Fig. 8), we enforce physical contact between at least one pair of vertices . We define E_{cont3D} as the sum of four energy terms from [Müller et al. 2021], aimed at minimizing Euclidean distances between contact vertices, aligning their normals, and avoiding self-collisions. Please see Appendix A for details. Furthermore, as outlined in Sec. 3.1, we aim to preserve the relative positions of the bones and joints in each contact region. To this end, for each contact region we find points on the 2D skeleton that are the closest to the contact, compute their 2D positions relative to each other, and aim to preserve those 2D positions between the same points on the 3D SMPL skeleton. Precisely, to determine those points, we select the local maxima of the heatmap over each 2D bone. We then connect each such point with all others within the same contact region, forming vectors $\hat{c}_i^{2D}$, which capture the relative position of such point with respect to another one. We aim to preserve these vectors exactly for the 3D pose, when projected onto the original view. For each point on the 2D skeleton, we find its corresponding point on the 3D skeleton by using the linear (arclength) parameterization of each bone and simply taking the 3D point with the same parameter value along the same bone. Denoting the vectors connecting the corresponding 3D skeleton points after projection as c_i^{2D}, we set:

$$E_{\text{cont2D}} = \sum_i \|c_i^{2D} - \hat{c}_i^{2D}\|^2.$$

  The final term is $E_{\text{contacts}} = \lambda_{\text{cont3D}} E_{\text{cont3D}} + \lambda_{\text{cont2D}} E_{\text{cont2D}}$.
- Regularity. As suggested by perception studies and previous work [Bessmeltsev et al. 2016; Xu et al. 2014], we speculate that human observers leverage regularity cues when interpreting sketches. In particular, the viewers expect nearly parallel 2D bones to stay parallel in 3D; feet nearly parallel to the floor to be standing on the floor. In enforcing this, we utilize perception research–indicated angle threshold [Hess and Field 1999] of 17°, below which we consider 2D bones to be parallel to each other or the floor. E_{reg} thus is a simple sum of squared differences between the corresponding normalized bone directions b_i and their target directions.

The naturalness of the poses is enforced implicitly by the EFT framework itself, as discussed in 4.2. For all the results presented in the paper, we use $\lambda_{\text{f}} = 10, \lambda_{\text{cont3D}} = 0.06, \lambda_{\text{cont2D}} = 10^{-3}$.

6 DATASET

We propose two novel datasets: the first large-scale dataset of 2D pose annotations for character sketches (D1), and a smaller dataset of high-quality character sketches (D2). Dataset D1 contains more than 3K images containing one or more sketched characters in articulated poses with 2D position annotated for each key joint (up to 18 per skeleton), in total containing 14,462 skeletons. Each image was annotated by a single annotator only; human annotations of sketches, however, are largely consistent, as shown by previous work [Bessmeltsev et al. 2016]. Dataset D2 contains 310 high-quality character sketches, with a very permissive usage license (CC BY 2.5). We use D1 to train and validate our 2D keypoint prediction network. We show a few examples from D1 in the Supplementary Materials; all the input images of the results in the paper, unless indicated otherwise, are from D2, and thus are not used in training.

Data Collection and Annotation. For D1, as the first step of data collection, we manually query search images via engines such as Google, Bing, and Baidu for character sketches and filter out irrelevant images. We additionally collect images with similar queries from Flickr and Pinterest and removed duplicates. We then hired an annotation service that marked up to 18 joint locations for each drawn character, with particular instructions to annotate occluded or not explicitly drawn joints if their position is clear from context, but skip the ones that are ambiguous. Naturally, this dataset contains sketches of numerous styles and complexities, greyscale and color, digitally drawn and scanned pen-and-paper drawings.

For D2, we collect high-resolution scans or photographs of character sketches from artists of different backgrounds. The sketches contain gestures, contour drawings, and detailed character sketches of humans in various, often highly articulated poses. The sketches are done in a variety of techniques on paper (pencil, pens, watercolor).

7 RESULTS AND VALIDATION

So far we have shown many examples of 3D characters, algorithmically posed via a single bitmap sketch (Fig. 1, 4, 5, 7). Our learning-based solution allows to pose 3D characters via natural, noisy, incomplete, and inaccurate character sketches, inaccessible to previous work. Our novel optimization allows us to successfully resolve ambiguities, inaccuracies, and distortions typical for character sketches, see e.g. Fig. 11, 12 for additional results. Our method robustly handles occlusions (e.g. the left arm in Fig. 12, second row) and altogether missing body parts (Fig. 12, top), typical for incomplete quick sketches or gestures. For all the examples, our method convincingly recreates the drawn poses in 3D.

Note that we only target estimating body pose, not its shape. Therefore, after our optimization we set the shape to the SMPL default for all our results.

We validate the key aspects of our method in a number of ways. The questionnaires used in the evaluations and detailed results are included in our supplementary.

Ablation Study. We perform an ablation study of our method (Fig. 13). We demonstrate results on a challenging example, each time skipping one component of our algorithm by disabling the corresponding loss term. For reference, we show a reprojection-based method [Müller et al. 2021] (Fig. 13a), highly sensitive to the depicted bone lengths, introducing strong unexpected foreshortening. Disabling the foreshortening transformation Fig. 13b also leads to a foreshortened pose, albeit slightly less (right shin) due to focusing on parallelism instead of 2D joint positions. Optimization without contacts results in an incorrect depth prediction of the left hand (Fig. 13c). Disabling the regularity term results in a left knee with a bent, invisible from the front (Fig. 13d). Please see the supplementary materials for the ablation study on the other inputs.

Foreshortening Transformation Function. We evaluate the robustness of Eq. 3 by fitting the function to smaller random subsets $\mathcal{A}$ (10%, 1%, 0.1%, and 0.05% of full training dataset). We get the same of similar coefficients of a, b and c, within the tolerance of 0.2, as for full training dataset. We further test robustness by fitting our foreshortening transformation function on a subset of standing poses only, obtaining $a' = 0.05, b' = 0.12, c' = 0.23$, as opposed to the original values in Sec. 4.4 ($a = 0.312, b = -0.448, c = 0.503$). Despite the difference in coefficients, the two polynomials $f(\beta)$ (Eq. 2) with those coefficients are virtually identical over the interval we are interested in, i.e. $\beta \in [0, \pi/2]$.

Qualitative Evaluation. We asked 2 artists and 7 non-professionals to comment on the results of our algorithm. We showed them each pair of input and our algorithmic result and asked to comment on the following statement, separately for each pair, "This 3D character pose captures the artist intended drawn pose.", with 5 Likert-type reply options: "Strongly disagree" (-2), "Disagree" (-1), "Neither disagree nor agree" (0), "Agree" (1), "Strongly Agree" (2). On average, the respondents agreed with the statement ($avg = 1.06, std = 0.38$). The layout of the study and the results are presented in the Supplementary.

Comparison to Prior Art. In Figure 15, we compare our method to Gesture3D [Bessmeltsev et al. 2016]. Their method relies on having a clean vector drawing, including inferring joint depth order from clean vector T-junctions, and assuming all the terminal joints are clearly visible and outlined (a). Our method handles a much wider variety of inputs, including natural bitmap sketches found in the wild. Automatically vectorizing such sketches to get a similar quality vectorization is an open problem (Figure 15c). Even supplied with correct 2D labels, Gesture3D does not capture the notion of pose naturalness, often resulting in unnatural poses (bottom middle). Furthermore, Gesture3D is designed for drawings with minimal foreshortening, producing flat, static poses (Figure 15, top middle). Our method successfully captures complex poses with significant foreshortening (Figure 15, top right).

We compare with reprojection-based methods in Fig. 4, 14, and Supplementary Materials. We use the implementations provided by the authors. Whenever a method accepts 2D labels as input, we supply the 2D labels predicted by our 2D network for a fair comparison. We first train SPIN [Kolotouros et al. 2019] on the results of Pavlakos et al. [2019], then improve those with the system of Joo et al. [2021], and fine-tune the SPIN regression network on those poses for better quality. For the methods directly predicting a 3D pose from an image, we retrained them using our dataset, following their training protocol. We run our method directly on the input sketch.

These methods do not aim to capture the artist-intended pose, rather focusing on the task of finding a natural pose with projections of 3D joints close to the 2D joint positions. In the presence of distortions, typical for character sketches, such as incorrect depiction of perspective and bone lengths, such approach often leads to exaggerated foreshortening (Fig. 4, 14), irregularities or unnatural poses (Fig. 12). Our method successfully reconstructs poses close to natural in the presence of such distortions and inaccuracies for both standard proportions (Fig. 14, top) and non-humanoid or characters with unrealistic or non-human proportions (Fig. 14, bottom).

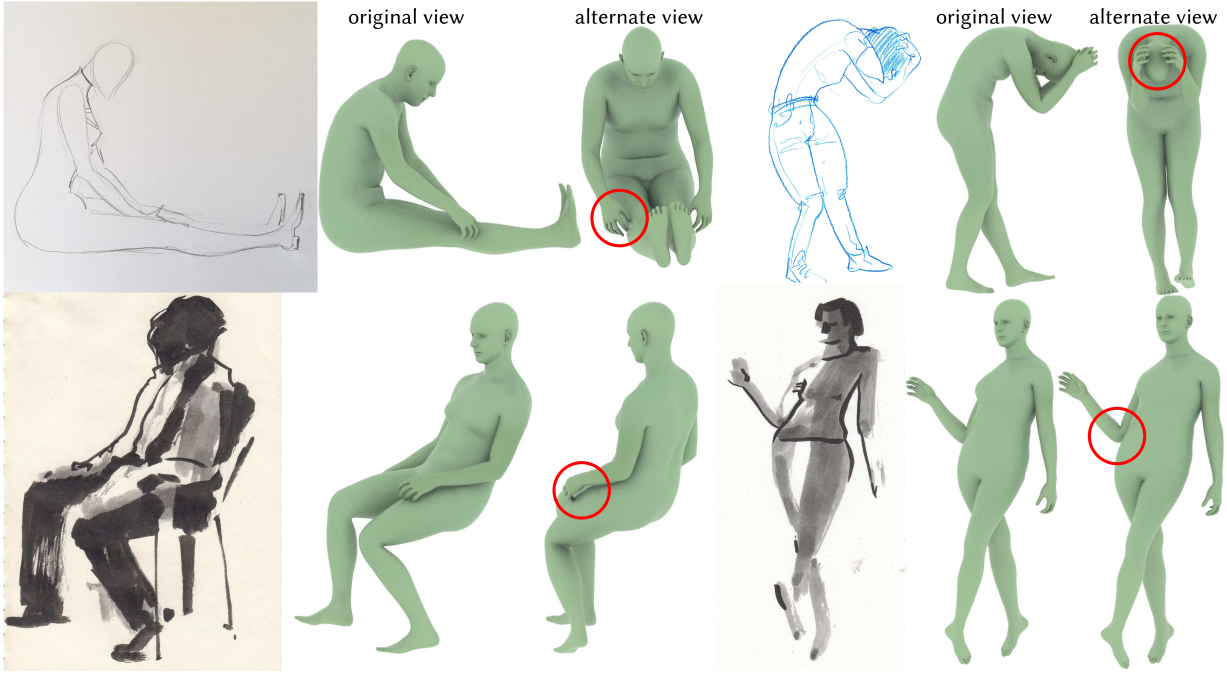

Fig. 11. A few examples of poses with self-contacts. Input images (except top-left) © Olga Posukh.

Qualitative Comparison. We validate the quality of our results by comparing them to the state of the art alternative [Müller et al. 2021] via a comparative perceptual study. Study participants were shown input sketches, together with our algorithmic posing result and an alternative posing result. The layout of the study is presented in the Appendix. The input sketch was shown on the top, marked as A, and the two posing results were placed at the bottom in random order and marked as "B" and "C". Participants were then asked "Which of the poses below, B or C, more accurately captures the drawn pose A on top? If both are equally acceptable, choose 'Both'. If neither, select 'Neither' ". We included 12 questions. We collected answers for each query from 14 different participants, including 5 males and 9 females, age ranging from 21 to 32 years; 3 were artists. The study data is presented in the supplementary.

To avoid the influence of the body shape on the study results, we reset body parameters to the average body shape for both methods. Similarly, since neither hands nor turn of the head are guided by the input sketch and are only controlled by their respective priors, we reset these parameters to their default values.

Fig. 16 summarizes the results. Participants preferred our results over the one of 64% of the time, ranked our methods on par 10% of the time, and preferred the alternative only 8% of the time. This study convincingly demonstrates that the 3D poses we produce are more consistent with viewer expectation than the ones produced by previous approaches.

Hand poses and head turn. Our system does not capture hand poses and the turn of the head; inferring those features from incomplete drawings proved to be a challenge. As a follow-up to our study (Fig. 16), we have asked users who selected 'Neither' for their comments, and most of the comments addressed hand poses and the turn of the head. Instead of relying on heuristics, we allow for a simple user interaction: the user is able to choose one of the predefined hand poses (fist, flat palm, palmar flection) for each hand and turn the head around its axis (Fig. 17). For this figure, a user adjusted the hand poses and the head turn within a few seconds. All the other results were processed in a fully automatic way; comparison with the previous work was done with automatically computed results.

Comparison with manually posed characters. We provided six of our input sketches and the SMPL model in a neutral pose to two 3D modeling experts and asked them to manually pose the characters into the poses drawn on the sketches. The artists took roughly from 5 to 15 minutes to pose the character for each drawing, while our algorithm inferred each pose in 1.5 minutes on average (Sec. 7).

We have furthermore performed a qualitative comparison user study with the same layout as for the comparison with the previous work, each time presenting our results and manually posed results in a random order. We asked 6 participants. Participants preferred our results 27% of the time, ranked both our and manual results as equally good 18% of the time, and preferred the manually posed characters 44% of the time. The participants chose "Neither" 11% of the time, disagreeing with both manually posed and our results.

Finally, we have quantitatively compared our algorithmic results with the manually posed 3D characters, as shown in Table 1. With respect to the standard MPJPE and PA-MPJPE metrics, we show that our results have smaller or equal errors than the previous work, similar to the natural variation between different experts. Those standard metrics, however, are not perception-based and thus are not necessarily indicative of user preferences.

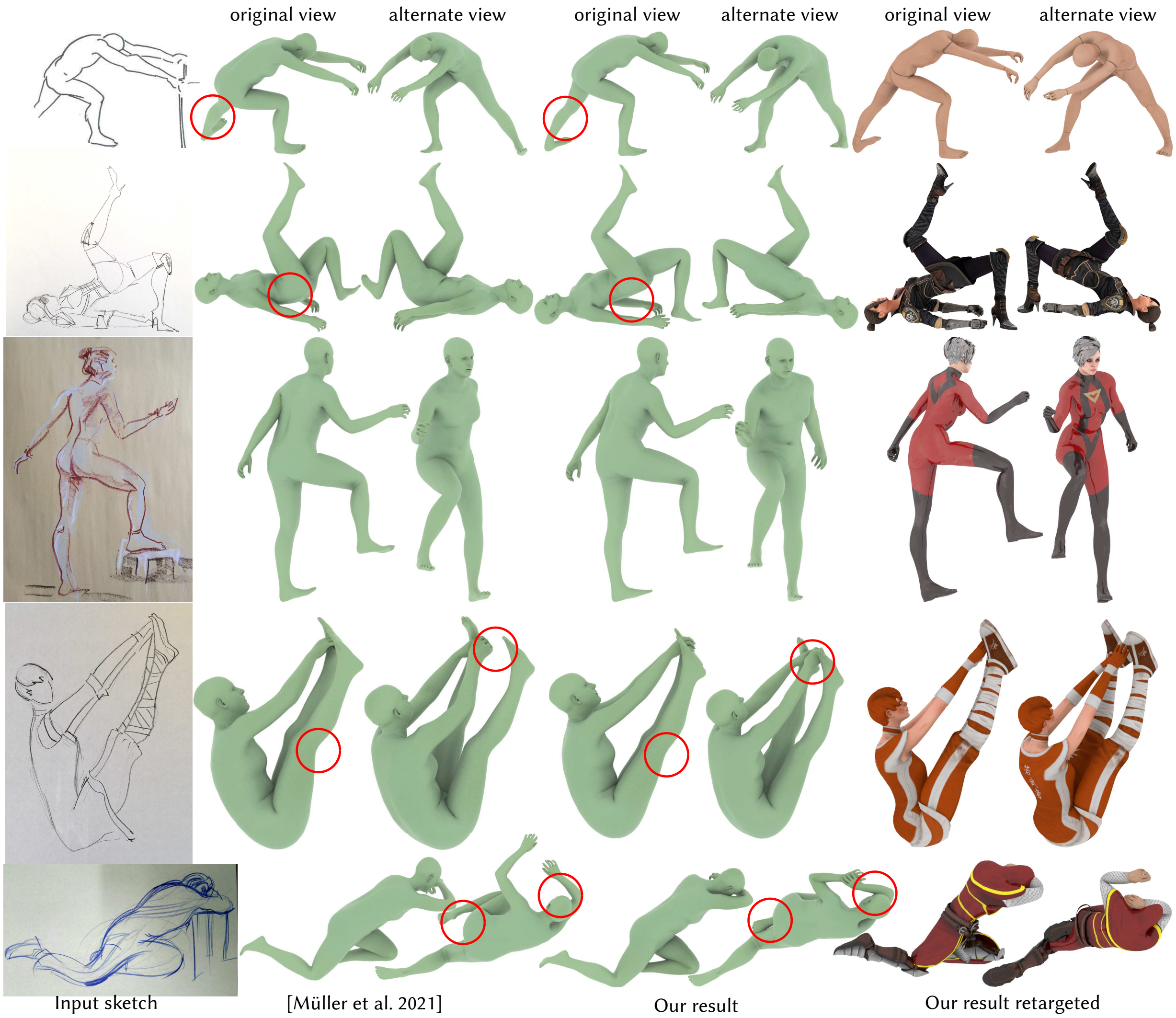

Fig. 12. Our algorithmic results can be automatically transferred onto a given custom character using standard tools (we used Blender's Animation Retargeting feature). We show the results of Müller et al. [2021] for comparison. Input image (top) © Rafianimates, (third row) © Brad Regier, (bottom) © Zoska Leutina.

Table 1. Error metrics on the manually posed characters.

	MPJPE	PA-MPJPE
[Pavlakos et al. 2019] vs Expert 1	266	232
[Pavlakos et al. 2019] vs Expert 2	282	237
[Kolotouros et al. 2019] vs Expert 1	189	152
[Kolotouros et al. 2019] vs Expert 2	223	153
[Joo et al. 2021] vs Expert 1	107	93
[Joo et al. 2021] vs Expert 2	136	100
[Müller et al. 2021] vs Expert 1	126	105
[Müller et al. 2021] vs Expert 2	150	116
Ours vs Expert 1	**103**	**78**
Ours vs Expert 2	**126**	**88**
Expert 1 vs Expert 2	116	79

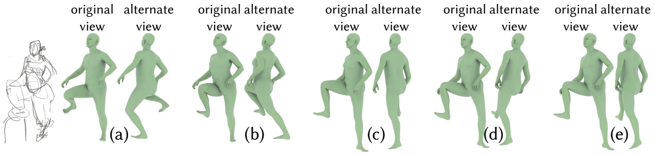

Fig. 13. An ablation study of our algorithm. (a) Result of a reprojection-based method of Müller et al. [2021]. (b) Our optimization without foreshortening transformation (b), without contacts (c), without the regularity energy (d), and our final result (e). For each pose, the original view is on the left, the alternate view is on the right.

Parameter Sensitivity. We show (Fig. 19) that our method produces plausible results for a range of parameters. Naturally, changing λ_f provides a way to balance trusting the depicted foreshortening.

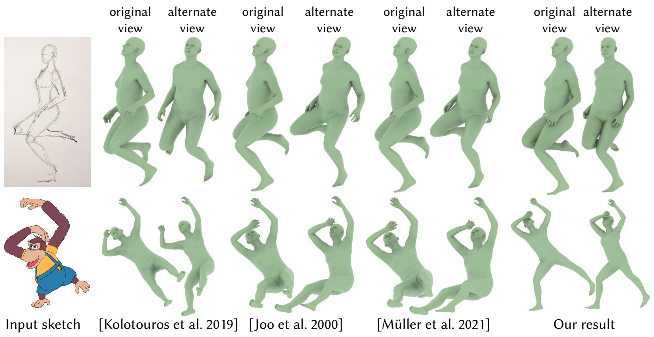

Fig. 14. Additional comparisons with reprojection-based approaches [Joo et al. 2021; Kolotouros et al. 2019; Müller et al. 2021], which fail to reconstruct plausible poses due to the typical inaccuracies and distortions of a character sketch. Our method correctly recovers the intended 3D poses (right).

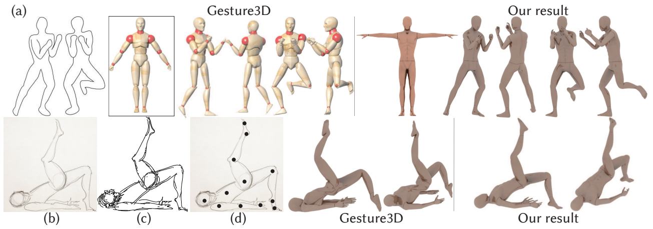

Fig. 15. Gesture3D [Bessmeltsev et al. 2016] only accepts clean vector drawings (a), unable to find 2D joint locations otherwise. Given a noisy sketch (b), modern vectorization methods often produce noisy vectorizations (c), incompatible with Gesture3D. Even provided with 2D labels (d), Gesture3D may produce implausible or static and flat poses (middle). Our method directly successfully infers 3D poses from a variety of sketches, including rasterized clean vector drawings like (a) and noisy raster drawings (b), producing realistic, expressive, and dynamic poses (right).

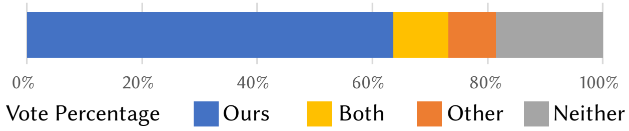

Fig. 16. Summary of comparative preferences in our perceptual study. Participants strongly preferred our results over the state-of-the-art alternative [Müller et al. 2021].

Similarly, increasing λ_{cont2D}, λ_{cont3D} prioritizes self-contacts in the final pose.

2D Keypoint Detection Validation. We evaluate the performance of 2D keypoint detection on our validation dataset, consisting of 882 drawings each containing a single character (roughly 6% of our dataset). We use a standard Percentage of Correct Keypoints (PCK@0.5) metric on this dataset, as well as mean Average Precision (mAP) metric on a range of Object Keypoint Similarity (OKS) thresholds. Overall, the 2D keypoint detection network reaches 0.891 of PCK@0.5 and 0.854 of mAP, which is substantial considering the

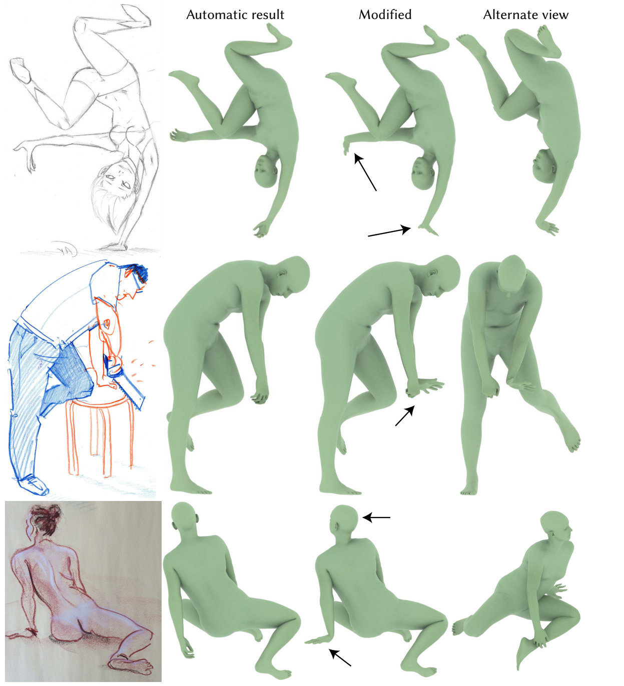

Fig. 17. We allow for a simple user interaction to edit the pose features our system does not infer: selecting from a small set of predefined hand poses and adjusting the turn of the head. This interaction typically takes a few seconds. Input image (top) © Achonan, (middle) © Olga Posukh, (bottom) © Brad Regier.

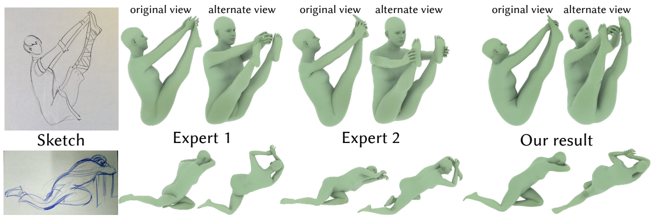

Fig. 18. Our algorithmic results (right) are often visually comparable with the characters manually posed by experts (middle). Our computations are roughly 3-10 times faster than manual posing.

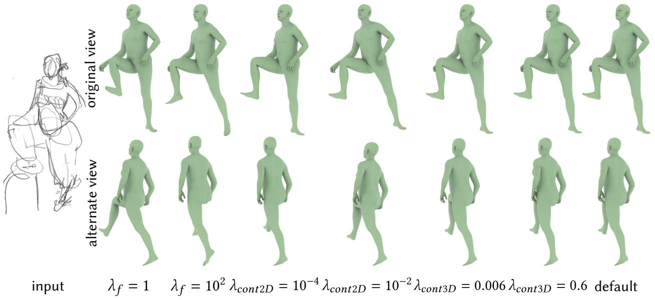

Fig. 19. Our method is robust to significant changes in the parameters. Our result with the default parameter values is on the right.

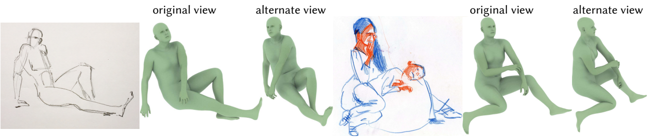

Fig. 20. Our algorithm may incorrectly resolve depth order (left, where the left arm should be behind) and cannot handle multiple characters (right). Input image (right) © Olga Posukh.

complexity of the task of inferring 2D joints for often incomplete sketches with occlusions and sparsely drawn curves. We observe that we reach a higher mean average precision score compared to the standard COCO keypoint detection benchmark in computer vision (0.795 of mAP [Liu et al. 2021]). We speculate that this may be to either the smaller size of our validation set or perhaps lower variability of line styles and textures in sketches compared to the influence of lighting effects in photographs.

Without training on our dataset, the pre-trained 2D keypoint detector of Sun et al. [2019] performs worse on our data (0.54 mAP, computed over the 13 joints we have in common). The pretrained OpenPose detector [Cao et al. 2019] fails on our data (0.002 mAP).

Input Quality, Style Independence, and Robustness. We demonstrate that due to the variety of our 2D keypoint annotated dataset, our system is robust to different drawing resolutions and quality, including high-quality scans (e.g. Fig. 17) and low-resolution or low-quality photographs of sketches (Fig. 12, bottom). Many of the drawings contain extra strokes, elements of shading, or simple noise, which would be an issue for previous methods assuming clean input; our method successfully handles those. Similarly, our system supports drawings of many styles, including gesture drawings (e.g. Fig. 12 top), detailed character sketches (e.g. Fig. 17), and more abstracted painterly drawings (Fig. 11, center and right).

Parameters and Performance. We have implemented the system in Python using PyTorch library. All the results presented in the paper were computed with the default parameters presented in the text. On our desktop machine (single Intel® Core™ i7-9700K CPU @ 3.60GHz with NVIDIA® GeForce® RTX 2080Ti), each of our results takes roughly 90 seconds. Most of the time is spent in the 3D optimization, where the bulk of time (90%) is taken by the generalized winding numbers computation for the self-contacts loss. The rest of the pipeline is almost immediate.

Limitations. Our system reconstructs the depth order of body parts based solely on the 2D information and pose naturalness prior, so it can occasionally misinterpret which body part is close to the viewer. Furthermore, one system can only pose a single character from a sketch, leaving the task of multiple character posing to future work (Fig. 20).

8 CONCLUSIONS AND FUTURE WORK

We have presented and validated the first method to infer a 3D humanoid character pose from a single bitmap sketch and introduced the first large-scale dataset of 2D skeletal joint annotations for bitmap sketches. Our system combines a modern deep learning framework with an optimization, guided by observations on the nature of sketches. Our method can process drawings of many different styles with occlusions, distorted proportions, and extra strokes or elements of shading, allowing to directly use natural drawings without any preprocessing or cleanup. We confirm that the poses our framework produces agree with the observers' expectations, by a significant margin more than the previous work.

Our work raises many directions for future research. First of all, we hope that the introduction of the 2D joint labels dataset will inspire follow-up research in 2D character inbetweening, segmentation, or consolidation of character sketches, among other possibilities. An interesting extension of our work would be to generalize it to arbitrary non-humanoid skeletons, where pose datasets are unavailable, via physics-based animation systems. Finally, an important line of research would generalize our method to non-skeletal rigs, supporting facial animation and nonlinear deformations.

ACKNOWLEDGMENTS

We would like to thank Oleg Muranov and Anastasia Seretkina for insightful discussions and data. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No.: RGPIN-2019-05097 ("Creating Virtual Shapes via Intuitive Input"), the Fonds de recherche du Québec - Nature et technologies (FRQNT) under Grant No.: 2020-NC-270087, and the NSERC-FRQNT NOVA Grant No. 314090. We also thank Maria Myronova and Elena Uretskaya for sharing their sketches.

REFERENCES

Kfir Aberman, Peizhuo Li, Sorkine-Hornung Olga, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62.

Ijaz Akhter and Michael J. Black. 2015. Pose-conditioned joint angle limits for 3D human pose reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June-2015 (2015), 1446–1455. https://doi.org/10.1109/CVPR.2015.7298751

Mikhail Bessmeltsev, Nicholas Vining, and Alla Sheffer. 2016. Gesture3D: Posing 3D Characters via Gesture Drawings. *ACM Trans. Graph.* 35, 6, Article 165 (Nov. 2016), 13 pages. https://doi.org/10.1145/2980179.2980240

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. arXiv:1607.08128 [cs.CV]

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv:1812.08008 [cs.CV]

João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. 2016. Human Pose Estimation with Iterative Error Feedback. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4733–4742. https://doi.org/10.1109/CVPR.2016.512

Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. 2011. Silhouette-based object phenotype recognition using 3D shape priors. In *IEEE International Conference on Computer Vision, ICCV*. 25–32.

Yucheng Chen, Yingli Tian, and Mingyi He. 2020. Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding* 192 (Mar 2020), 102897. https://doi.org/10.1016/j.cviu.2019.102897

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded Pyramid Network for Multi-Person Pose Estimation. (2018).

Byungkuk Choi, Roger Blanco i Ribera, J. P. Lewis, Yeongho Seol, Seokpyo Hong, Haegwang Eom, Sunjin Jung, and Junyong Noh. 2016. SketchiMo: Sketch-Based Motion Editing for Articulated Characters. *ACM Trans. Graph.* 35, 4, Article 146 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925970

M. G. Choi, K. Yang, T. Igarashi, J. Mitani, and J. Lee. 2012. Retrieval and visualization of human motion data via stick figures. *Computer Graphics Forum* 31 (2012), 2057–2065.

Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, and Arjun Jain. 2017. Structure-Aware and Temporally Coherent 3D Human Pose Estimation. *CoRR* abs/1711.09250 (2017). arXiv:1711.09250 http://arxiv.org/abs/1711.09250

James Davis, Maneesh Agrawala, Erika Chuang, Zoran Popović, and David Salesin. 2003. A Sketching Interface for Articulated Figure Animation. *Proc. Symposium on Computer Animation* (2003), 320–328.

Sai Kumar Dwivedi, Nikos Athanasiou, Muhammed Kocabas, and Michael J. Black. 2021. Learning To Regress Bodies From Images Using Differentiable Semantic Rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11250–11259.

Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. 2021. Learning Complex 3D Human Self-Contact. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 2 (May 2021), 1343–1351. https://ojs.aaai.org/index.php/AAAI/article/view/16223

Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans Peter Seidel. 2010. Optimization and filtering for human motion capture : AAA multi-layer framework. *International Journal of Computer Vision* 87, 1-2 (2010), 75–92.

Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig Animation with a Tangible and Modular Input Device. *ACM Trans. Graph.* 35, 4, Article 144 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925909

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. 2004. Style-Based Inverse Kinematics. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 522–531. https://doi.org/10.1145/1015706.1015755

Martin Guay, Marie-Paule Cani, and Rémi Ronfard. 2013. The Line of Action: An Intuitive Interface for Expressive Character Posing. *ACM Trans. Graph.* 32, 6, Article 205 (Nov. 2013), 8 pages. https://doi.org/10.1145/2508363.2508397

Martin Guay, Rémi Ronfard, Michael Gleicher, and Marie-Paule Cani. 2015. Space-Time Sketching of Character Animation. *ACM Trans. Graph.* 34, 4, Article 118 (July 2015), 10 pages. https://doi.org/10.1145/2766893

Fabian Hahn, Frederik Mutzel, Stelian Coros, Bernhard Thomaszewski, Maurizio Nitti, Markus Gross, and Robert W. Sumner. 2015. Sketch Abstractions for Character Posing. In *Proc. Symp. Computer Animation*. 185–191.

Ronie Hecker and Kenneth Perlin. 1992. Controlling 3D objects by sketching 2D views. *Proc. SPIE* 1828 (1992), 46–48.

Robert Hess and David Field. 1999. Integration of contours: new insights. *Trends in Cognitive Sciences* 3, 12 (1999), 480–486.

Burne Hogarth. 1996. *Dynamic Figure Drawing*. Watson-Guptill.

Clara Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Analysis & Machine Intelligence* 36, 7 (2014), 1325–1339.

Alec Jacobson, Ladislav Kavan, and Olga Sorkine. 2013. Robust Inside-Outside Segmentation using Generalized Winding Numbers. *ACM Trans. Graph.* 32, 4 (2013).

Eakta Jain, Yaser Sheikh, Moshe Mahler, and Jessica Hodgins. 2012. Three-dimensional proxies for hand-drawn characters. *ACM Trans. on Graphics* 31, 1 (2012), 1–16.

Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. 2021. Exemplar Fine-Tuning for 3D Human Model Fitting Towards In-the-Wild 3D Human Pose Estimation. *2021 International Conference on 3D Vision (3DV)* (2021), 42–52.

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8320–8329. https://doi.org/10.1109/CVPR.2018.00868

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-End Recovery of Human Shape and Pose. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 7122–7131. https://doi.org/10.1109/CVPR.2018.00744

K. Koffka. 1955. *Principles of Gestalt Psychology*. Routledge & K. Paul.

Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. 2019. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *Proceedings of the IEEE International Conference on Computer Vision*.

Ji-yong Kwon and In-Kwon Lee. 2012. The Squash-and-Stretch Stylization for Character Motions. *IEEE Trans. Vis. Comput. Graph.* 18, 3 (2012), 488–500. https://doi.org/10.1109/TVCG.2011.48

Juncong Lin, Takeo Igarashi, Jun Mitani, and Greg Saul. 2010. A Sketching Interface for Sitting-pose Design. In *Proc. Sketch-Based Interfaces and Modeling Symposium*. 111–118.

Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. 2021. Polarized Self-Attention: Towards High-quality Pixel-wise Regression. *Arxiv Pre-Print arXiv:2107.00782* (2021).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graph.* 34, 6, Article 248 (Oct. 2015), 16 pages. https://doi.org/10.1145/2816795.2818013

Meysam Madadi, Hugo Bertiche, and Sergio Escalera. 2018. SMPLR: Deep SMPL reverse for 3D human pose and shape recovery. *CoRR* abs/1812.10766 (2018). arXiv:1812.10766 http://arxiv.org/abs/1812.10766

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

C Mao, S F Qin, and D K Wright. 2005. A sketch-based gesture interface for rough 3D stick figure animation. *Proc. Sketch Based Interfaces and Modeling* (2005).

Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *ICCV*.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017a. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE. https://doi.org/10.1109/3dv.2017.00064

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera. *ACM Transactions on Graphics* 39, 4 (2020), 1–24. https://doi.org/10.1145/3386569.3392410 arXiv:1907.00837

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017b. VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics* 36, 4 (2017), 1–13. https://doi.org/10.1145/3072959.3073596 arXiv:1705.01583

Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. 2021. On Self-Contact and Human Pose. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recogßnition (CVPR)*.

Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. *CoRR* abs/1603.06937 (2016). arXiv:1603.06937 http://arxiv.org/abs/1603.06937

David Novotný, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. 2019. C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion. *CoRR* abs/1909.02533 (2019). arXiv:1909.02533 http://arxiv.org/abs/1909.02533

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin P. Murphy. 2017. Towards Accurate Multi-person Pose Estimation in the Wild. *CoRR* abs/1701.01779 (2017). arXiv:1701.01779 http://arxiv.org/abs/1701.01779

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1263–1272. https://doi.org/10.1109/CVPR.2017.139

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. 2018. Learning to Estimate 3D Human Pose and Shape From a Single Color Image. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 459–468. https://doi.org/10.1109/CVPR.2018.00055

Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2012. Reconstructing 3D Human Pose from 2D Image Landmarks. In *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 573–586.

D. Ramanan. 2011. Part-Based Models for Finding People and Estimating Their Pose. *Springer* (2011), 199–223.

G. Rogez, P. Weinzaepfel, and C. Schmid. 2017. LCR-Net: Localization-Classification-Regression for Human Pose. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1216–1224. https://doi.org/10.1109/CVPR.2017.134

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

Benjamin Sapp, Alexander Toshev, and Ben Taskar. 2010. Cascaded models for articulated pose estimation. *Lecture Notes in Computer Science* 6312 (2010), 406–420.

Ryan Schmidt, Azam Khan, Gord Kurtenbach, and Karan Singh. 2009. On expert performance in 3D curve-drawing tasks. *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling - SBIM '09* 1 (2009), 133. https://doi.org/10.1145/1572741.1572765

Mingyi Shi, Kfir Aberman, Andreas Aristidou, Taku Komura, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020. MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency. *ACM Trans. Graph.* 40, 1, Article 1 (Sept. 2020), 15 pages. https://doi.org/10.1145/3407659

Karan Singh. 2002. A Fresh Perspective. In *Proceedings of the Graphics Interface 2002 Conference, May 27-29, 2002, Calgary, Alberta, Canada* (Calgary, Alberta). 17–24. http://graphicsinterface.org/wp-content/uploads/gi2002-3.pdf

Tibor Stanko, Mikhail Bessmeltsev, David Bommes, and Adrien Bousseau. 2020. Integer-Grid Sketch Simplification and Vectorization. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Geometry Processing)* 39, 5 (jul 2020), 149–161. http://www-sop.inria.fr/reves/Basilic/2020/SBBB20

Nisha Sudarsanam, Cindy Grimm, and Karan Singh. 2008. Non-Linear Perspective Widgets for Creating Multiple-View Images. In *Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR*

'08). Association for Computing Machinery, New York, NY, USA, 69–77. https://doi.org/10.1145/1377980.1377995

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR.*

Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. 2017. Compositional Human Pose Regression. In *2017 IEEE International Conference on Computer Vision (ICCV).* 2621–2630. https://doi.org/10.1109/ICCV.2017.284

Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. 2017. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. arXiv:1611.05708 [cs.CV]

Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2016. Direct Prediction of 3D Body Poses From Motion Compensated Sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Frank Thomas and Ollie Johnston. 1981. *The Illusion of Life: Disney Animation* (1st hyperi ed.). Disney Editions, New York, N.Y.

Denis Tomè, Chris Russell, and L. Agapito. 2017. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5689–5698.

Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition.* 1653–1660. https://doi.org/10.1109/CVPR.2014.214

Hsiao-Yu Fish Tung, Adam W. Harley, William Seto, and Katerina Fragkiadaki. 2017. Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision. arXiv:1705.11166 [cs.CV]

Leo Brodie Walt Stanchfield. 2020. *Gesture Drawing for Animation* (1 ed.). Independently published.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. 2019b. Deep High-Resolution Representation Learning for Visual Recognition. *TPAMI* (2019).

Keze Wang, Liang Lin, Chenhan Jiang, Chen Qian, and Pengxu Wei. 2019a. 3D Human Pose Machines with Self-supervised Learning. *IEEE transactions on pattern analysis and machine intelligence* (2019).

Xiaolin Wei and Jinxiang Chai. 2011. Intuitive Interactive Human-Character Posing with Millions of Example Poses. *IEEE Comput. Graph. Appl.* 31, 4 (2011), 78–88.

Marta Wnuczko, Karan Singh, and John M Kennedy. 2016. Foreshortening produces errors in the perception of angles pictured as on the ground. *Attention, Perception, & Psychophysics* 78, 1 (2016), 309–316.

Jungdam Won and Jehee Lee. 2016. Shadow Theatre: Discovering Human Motion from a Sequence of Silhouettes. *ACM Trans. Graph.* 35, 4, Article 147 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925869

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. 2019. Monocular Total Capture: Posing Face, Body, and Hands in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple Baselines for Human Pose Estimation and Tracking. In *European Conference on Computer Vision (ECCV).*

Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. *Transactions on Graphics (Proc. SIGGRAPH 2014)* 33, 4 (2014). https://doi.org/2601097.2601128

Yuanlu Xu, Song Chun Zhu, and Tony Tung. 2019. DenseRaC: Joint 3D pose and shape estimation by dense render-And-compare. *arXiv* (2019), 7760–7770.

Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy S. J. Ren, Hongsheng Li, and Xiaogang Wang. 2018. 3D Human Pose Estimation in the Wild by Adversarial Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018.* IEEE Computer Society, 5255–5264. https://doi.org/10.1109/CVPR.2018.00551

Y. Yang and D. Ramanan. 2013. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 12 (2013), 2878–2890. https://doi.org/10.1109/TPAMI.2012.261

Jianmin Zhao and Norman I. Badler. 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics* 13, 4 (1994), 313–336.

Yue Zhong, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. 2020. Deep Sketch-Based Modeling: Tips and Tricks. arXiv:2011.06133 [cs.CV]

Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. 2017. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach. In *The IEEE International Conference on Computer Vision (ICCV).*

Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. 2016. Deep Kinematic Pose Regression. In *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 9915),* Gang Hua and Hervé Jégou (Eds.). 186–201. https://doi.org/10.1007/978-3-319-49409-8_17

A DETAILS ON THE CONTACT TERM E_{cont3D}

In this section, we precisely follow Müller et al. [2021]; this is given only for completeness, all the missing details can be found in the original paper.

Definition A.1. Given a mesh M, we define two vertices v_i and $v_j \in M$ to be in *self-contact*, if $||v_i - v_j|| < t_{\text{eucl}}$ and $geo(v_i, v_j) > t_{\text{geo}}$, where $geo(v_i, v_j)$ is the geodesic distance between v_i and v_j. We use $t_{\text{geo}} = 30cm$ and $t_{\text{eucl}} = 2cm$.

We denote all vertex pairs in M satisfying this definition as M_C. We further define an operator $\mathcal{U}(\cdot)$ that returns a set of unique vertices in M_C, and an operator $f_g(\cdot)$ that takes v_i as input and returns the Euclidean distance to the nearest v_j that is far enough in the geodesic sense.

We set

$$E_{\text{cont3D}} = \mathcal{L}_{\tilde{C}} + \mathcal{L}_{\text{C}} + \mathcal{L}_{\text{P}} + \mathcal{L}_{\text{A}},$$

where

$$\mathcal{L}_{\tilde{C}} = \frac{1}{|\mathcal{U}(\tilde{M}_C)|} \sum_{v_i \in \mathcal{U}(\tilde{M}_C)} \tanh f_g(v_i)$$

encourages every vertex in the predicted set of contact vertices $\tilde{M}_C$ to be in contact. Vertices in contact are pulled together via a contact term $\mathcal{L}_{\text{C}}$. To prevent self-intersections, vertices inside the mesh are pushed to the surface via a pushing term $\mathcal{L}_{\text{P}}$. Finally, $\mathcal{L}_{\text{A}}$ aligns the surface normals of two vertices in contact.

To compute self-contact terms, we first find which vertices are inside, $M_I \subset M$ via generalized winding numbers [Jacobson et al. 2013]. SMPL-X is not a closed mesh; this complicates the self-intersection tests. We close it by adding a vertex at the back of the mouth. In addition, SMPL and SMPL-X often self-intersect by default, e.g. torso and upper arms. We identify such common self-intersections and filter them out from M_I. To capture fine-grained contact, we map the union of inside and contact vertices onto the HD SMPL-X surface, i.e. $M_D = HD(M_I \cap M_C)$, which is further segmented into an inside M_{D_I} and outside $M_{D_I^\complement}$ subsets via intersection tests. The objectives are defined as

$$\mathcal{L}_{\text{C}} = \alpha_1 \sum_{p_i \in M_{D_I^\complement}} \tanh^2 \frac{f_g(p_i)}{\alpha_2},$$

$$\mathcal{L}_{\text{P}} = \beta_1 \sum_{p_i \in M_{D_I}} \tanh^2 \frac{f_g(p_i)}{\beta_2},$$

$$\mathcal{L}_{\text{A}} = \sum_{(p_i, p_j) \in M_{D_C}} 1 + \langle N(p_i), N(p_j) \rangle.$$

Here M_{D_C} is the subset of vertices in contact in M_D. We use the same parameter values as in the original paper, $\alpha_1 = \alpha_2 = 0.005$, $\beta_1 = 1$, and $\beta_2 = 0.04$.