Gesture3D: Posing 3D Characters via Gesture Drawings

Mikhail Bessmeltsev, Nicholas Vining, and Alla Sheffer
University of British Columbia

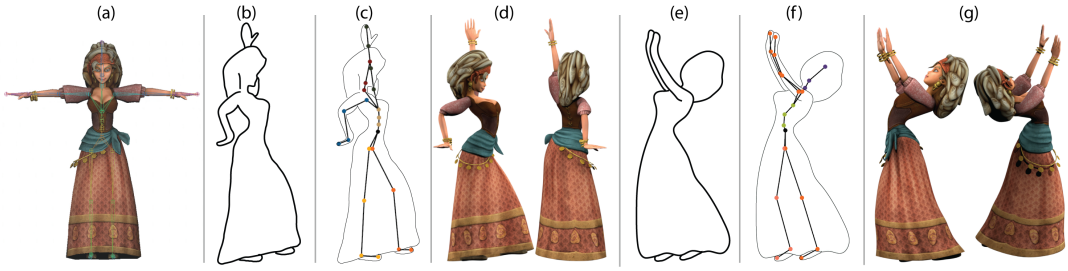


Figure 1: *Gesture3D: gesture drawings (b,e) of an input character model (a); estimated 2D skeleton projections (c,f) and new poses automatically computed from the drawings (d,g).*

Abstract

Artists routinely use *gesture drawings* to communicate ideated character poses for storyboarding and other digital media. During subsequent posing of the 3D character models, they use these drawing as a reference, and perform the posing itself using 3D interfaces which require time and expert 3D knowledge to operate. We propose the first method for automatically posing 3D characters directly using gesture drawings as an input, sidestepping the manual 3D posing step. We observe that artists are skilled at quickly and effectively conveying poses using such drawings, and design them to facilitate a single perceptually consistent pose interpretation by viewers. Our algorithm leverages perceptual cues to parse the drawings and recover the artist-intended poses. It takes as input a vector-format rough gesture drawing and a rigged 3D character model, and plausibly poses the character to conform to the depicted pose. No other input is required. Our contribution is two-fold: we first analyze and formulate the pose cues encoded in gesture drawings; we then employ these cues to compute a plausible image space projection of the conveyed pose and to imbue it with depth. Our framework is designed to robustly overcome errors and inaccuracies frequent in typical gesture drawings. We exhibit a wide variety of character models posed by our method created from gesture drawings of complex poses, including poses with occlusions and foreshortening. We validate our approach via result comparisons to artist-posed models generated from the same reference drawings, via studies that confirm that our results agree with viewer perception, and via comparison to algorithmic alternatives.

Keywords: character posing, sketch-based modeling, gesture drawing

Concepts: •**Computing methodologies** → **Mesh geometry models;**

SA '16 Technical Papers, December 05-08, 2016, , Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2980240

ACM Reference Format
Bessmeltsev, M., Vining, N., Sheffer, A. 2016. Gesture3D: Posing 3D Characters via Gesture Drawings. ACM Trans. Graph. 35, 6, Article 165 (November 2016), 13 pages. DOI = 10.1145/2980179.2980240 http://doi.acm.org/10.1145/2980179.2980240.

1 Introduction

Gesture drawings - rough, yet expressive, contour drawings of posed characters (Figure 1b,e) - are routinely used by artists to quickly convey the action, form, and pose of a character figure [Blair 1994; Hampton 2009; Nicolades 1975]. Artists are trained to create *descriptive* gesture drawings which unambiguously convey a character's pose in just a few minutes [Leland 2006], and use them ubiquitously when conceiving character poses and motion key-frames for storyboarding. In digital media production, artists subsequently apply these envisioned poses to 3D character models. In current practice, posing is performed separately, using the drawings as a reference only, and requires additional, often significant, user interaction (Section 2). We seamlessly connect the ideation and modeling steps by introducing the first method for 3D character posing which poses the characters algorithmically using gesture drawings as input, allowing artists to directly communicate their ideas using drawings and sidestepping the mental overhead of interacting with a complex software interface. As demonstrated, our method plausibly poses 3D characters using quickly generated, rough, vectorized gesture drawings and rigged character models, provided in a neutral bind pose, as the only input; no manual input or additional markup is required. It successfully handles complex poses with varying and significant part foreshortening, occlusions, and drawing inaccuracies (Figure 1).

The advantage of gesture drawings over other types of 2D inputs explored by previous posing approaches (Section 2) is the lack of perceptual ambiguity. Unlike stick-figures, lines of action, and outer silhouettes (Figure 2), gesture drawings allow artists to *unambiguously* convey poses to human observers. By identifying and leveraging the perceptual pose cues used by artists when creating these drawings, we are able to automatically recover character poses that are consistent with artist intent.

Our framework centers around analysis of the stroke curves forming the gesture drawings (Section 4). Like many other line drawings, gesture drawings are dominated by contour curves, conveying the occlusion contours of the depicted characters. However, since gesture drawings focus on conveying pose rather than shape, they typically only depict approximate, abstracted, character anatomy. In particular, artists typically use simple low-curvature stroke segments to outline body parts and use higher-curvature sections to depict their connecting joints [Hampton 2009]. These high-curvature *anatomical landmarks* assist observers in parsing the drawings. The abstracted contour strokes of a gesture drawing are designed to con-

vey largely smooth 3D character geometry. As observed by [Bessmeltsev et al. 2015] in such scenarios the contours of both individual body parts and part chains are usually *continuous*; thus adjacent contour stroke segments always outline adjacent body parts, and adjacent body part outlines are typically depicted using one shared stroke, or multiple Gestalt continuous [Koffka 1955] strokes. We also observe that body part contours are consistently *oriented* with respect to the parts' skeletal bone and rarely *cross* the bone's 2D projection. Combined together, these three *contour consistency* cues allows observers to identify poses with globally consistent joint and bone locations. Generic projected contours allow multiple depth interpretations, thus artists are trained to use drawing cues to reduce ambiguity. When estimating depth from 2D drawings, viewers prefer *less foreshortened* interpretations of the observed shapes, thus to best convey the intended poses artists seek to select viewpoints with smaller foreshortening [Hogarth 1996]. We also observe that in gesture drawings artists prominently use local, suggestive, occlusions to convey changes in depth and to specify depth order between adjacent joints. Gestalt psychology [Koffka 1955; Attneave and Frost 1969] points to a persistent viewer preference for *simple* drawing interpretations. In the context of gesture drawings, we believe that viewers use two types of simplicity cues: in cases where drawings are ambiguous, viewers prefer more *natural* poses, or ones with angles closer to those in the input bind pose; viewers also visually complete hidden body parts and correct drawing inaccuracies by using *regularity* cues, such as pose symmetry [Bessmeltsev et al. 2015]. Finally, we note that while human observers can clearly parse professional gesture drawings, reliance on the drawing to accurately depict character proportions and projected joint locations must be qualified: drawings are typically *inexact*, as even experts depict foreshortened objects inaccurately [Schmidt et al. 2009] and fail to correctly account for perspective [Xu et al. 2014].

Overview. We use these observations to pose the input 3D rigged character model into the artist intended pose conveyed by the input gesture drawing. We first match skeletal elements against corresponding contour stroke segments, placing joints next to their matching contours. We formulate joint placement as a discrete 2D embedding that matches joints to corresponding contour samples and is dominated by contour consistency and anatomical landmark matching considerations. We then compute the desired embedding by casting it as a variation of the tree-structured Markov Random Field (MRF) problem (Figure 9b, Section 5). We extend our solution to 3D by leveraging the depth order implied by occlusion contours, and the observations about viewer preference for simple and less foreshortened poses. To overcome drawing inaccuracy, we formulate 3D embedding as an energy minimization problem which balances landmark-implied 2D joint placement against the simplicity and foreshortening cues (Figure 9d, Section 7).

Contribution. Our contribution is two-fold: we formulate the properties of effective gesture drawings, bringing together insights from multiple sources in the areas of psychology, art, and computer graphics, highlighting key perceptual cues which enable viewers to perceive the artist intended character poses; we then use these observations to introduce the first gesture drawing based algorithm for posing 3D characters. Our method enables artists to directly convert their ideated posed character drawings into 3D character poses, and supports complex drawings with occlusions, variable body part foreshortening, and drawing inaccuracies.

Validation. We exhibit a gallery of character poses obtained automatically from gesture drawings of a range of 3D characters (Section 9) and validate our algorithm in a number of ways (Section 8). We evaluate our results against ground truth data, by first rendering projected contours of posed character models, then using these contours as input to our method and comparing our results against original poses; we compare our algorithm's results with characters posed by artists given the same drawings as input; we compare the character-contour correspondences computed by our method against manual annotation by human observers; and we collect qualitative result evaluations by experts and non-experts alike. Finally, we compare our method against prior work and algorithmic alternatives. These validations confirm that the poses we compute are consistent with viewer perception and artist intent.

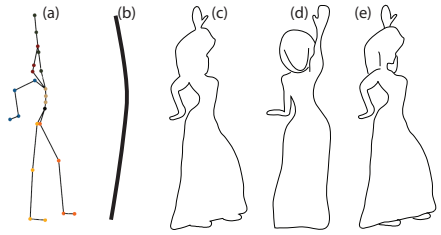


Figure 2: *Stick figure drawings (a), lines of action (b), and outer silhouettes (c) allow for multiple perceptually valid pose interpretations. (d) Poor view selection results in highly foreshortened contours leading to loss of pose information (e.g bends on the left arm or the curved spine). Gesture drawings, consciously drawn from descriptive views (e) effectively convey the intended pose.*

Limitations. Our method assumes vectorized input drawings, consistent with sketch-based modeling papers such as [Bessmeltsev et al. 2015; Karpenko and Hughes 2006; Shtof et al. 2013]. We do not attempt to address the problem of generating vectorized inputs in this paper, and assume that they are either manually generated or automatically produced using a state-of-the-art algorithm such as [Favreau et al. 2016]. Our current implementation requires between 3 and 4 minutes to produce a pose from a sketch; this means that it is most suitable for applying to existing assets such as storyboard drawings, and that significant optimization work would be needed for an interactive application.

2 Related Work

We build on prior work in several domains: sketch based character modeling, pose reconstruction from images, and character posing.

Sketch Based Modeling Posing from gesture drawings faces some of the same challenges as single-view sketch-based character modeling. As pointed out by Bessmeltsev et al. [2015], modeling a posed character given a single contour sketch is inherently ambiguous, as the contours alone allow multiple perceptually plausible interpretations. To generate the user-desired models, existing methods either rely on incremental contour drawing order [Cherlin et al. 2005], employ primitive-based annotation [Cherlin et al. 2005; Gingold et al. 2009; Shtof et al. 2013], require users to provide a matching posed 3D skeleton [Bessmeltsev et al. 2015], or enforce a range of strong simplifying assumptions [Cordier et al. 2011; Buchanan et al. 2013; Karpenko and Hughes 2006] - for instance, Cordier et al. [2011] only handle mirror-symmetric poses, while other methods assume that the 3D generators of the drawn contours are planar and near-parallel to the view plane [Buchanan et al. 2013; Karpenko and Hughes 2006; Bai et al. 2016], an assumption that does not hold for most character poses [Bessmeltsev et al. 2015]. Our work addresses generic gesture drawings with no such limitations (Figure 1), solving an essentially inverse problem to that of [Bessmeltsev et al. 2015]: instead of computing a character geometry given a sketch and a 3D pose, we compute a character pose given a gesture drawing and a character model.

Rather than creating a model from scratch, methods such as [Kraevoy et al. 2009] deform a 3D character template to fit a contour drawing. They either expect the template and drawn poses to be aligned, or expect users to manually specify coarse template-drawing correspondences. They then use local shape compatibility between the input outlines and the corresponding 3D geometry to obtain dense correspondences. Since contours in gesture drawings are approximate and highly abstracted, local shape compatibility cannot be used as a reliable criterion in our setup. Despite this extra challenge, our method does not require manual correspondences nor expects the drawn pose to resemble the input bind one.

Pose Reconstruction From Images Gesture posing is related to the classical computer vision problem of pose reconstruction from monocular images and video; see [Tekin et al. 2015] and the references therein. Reconstruction from video aims to capture a continuous motion, where the pose in each frame is very close to a previously reconstructed pose in the preceding frame, and heavily relies both on this existing previous pose and on fine image-level correspondences between frames (e.g. [de Aguiar et al. 2008; Gall et al. 2009; Tekin et al. 2015].) Gesture posing has more in common with pose estimation from a single frame, where no such priors are available (e.g. [Gall et al. 2010; Sapp et al. 2010; Chen et al. 2011; Ionescu et al. 2014]). Both outlines and incidental-view occlusion contours (Figure 2c,d) are insufficient to deduce a pose; single-frame pose estimation methods therefore frequently combine this information with textural and shading cues which are unavailable in our setup. Recent posing approaches (e.g. [Ionescu et al. 2014]) predict the most likely 3D pose by learning from large databases of real and synthetic human motion data. Such databases bias the results toward more frequent poses and can be difficult to obtain for non-humanoid or non-realistic characters, or for extreme/non-physical poses. Our framework overcomes the lack of extensive anatomic pose priors, and allows recovery of atypical poses by leveraging the descriptive cues artists provide when creating gesture drawings.

Character Posing Interfaces In most industry setups, characters are posed via 3D skeleton manipulation. Users either manually adjust joint angles, or use Inverse Kinematics (IK) based tools to place bone end-points at specific locations [Zhao and Badler 1994]. While IK-based frameworks relieve some of the tedium of adjusting individual joints, they still require experience with 3D modeling systems and non-trivial posing time. As Grochow et al.[2004] point out, posing a character using only handle constraints is inherently ambiguous as specified handle locations often allow multiple internal joint configurations. They resolve this ambiguity by building an IK system based on a learned model of human poses. Their method requires training data from motion capture that is similar to the pose envisioned by the artist. When artists ideate their desired poses they prefer to use pen and paper, avoiding this mentally cumbersome 3D machinery. Using these ideation drawings as-is to create 3D poses saves artists time and effort.

Hahn et al. [2015] and Guay et al. [2013] propose incremental, multi-view, sketch-based posing interfaces. Lines of action, imaginary lines running down a character's spine or other major bone chains (Figure 2b) are used by artists for coarse pose communication [Guay et al. 2013]. Guay et al. use line-of-action strokes to pose characters by placing user-specified corresponding bone-chains along these strokes. This input allows multiple pose interpretations for body parts not directly present on the line of action or its continuation, and requires an incremental multi-view interface to pose non-coplanar bone-chains. Hahn et al. [2015] propose an interface where a user poses characters one limb at a time, by first drawing a stroke along a limb in the current pose and then drawing a corresponding stroke depicting its new pose. The system then poses the limbs by aligning them to the strokes. It assumes uniform foreshortening along the posed limbs, and requires multiple stroke pairs and view changes to generate complex poses. Our work complements these approaches by providing a single-view drawing-based posing mechanism, allowing artists to directly use their gesture and keyframe drawings for character posing.

A number of recent methods use stick-figures [Hecker and Perlin 1992; Davis et al. 2003; Mao et al. 2005; Lin et al. 2010] - 2D projections of the desired 3D skeleton of the posed character (Figure 2a) - to compute a corresponding 3D skeletal pose As the authors acknowledge, stick-figures are inherently ambiguous and allow for multiple geometrically valid and perceptually plausible 3D interpretations. Hecker and Perlin [1992] and Mao et al. [2005] propose users to encode the relative depth of bones and joints via pen pressure or stroke width. Such interfaces become unwieldy for typical characters (e.g. Figure 1) which have dozens of bones. Davis et al. [2003] resolve ambiguities through user annotation, followed by users selecting the desired character pose from multiple plausible solutions. Lin et al. [2010] use stick-figures to pose characters sitting in a chair, and reduce ambiguities by using specific priors relevant only for sitting characters. Jain et al. [2012] manually place stick figure joint locations and bounding boxes on multiple frames of an input animation, and then manually select motion capture data similar to the input animation frames to resolve depth ambiguities. Wei et al. [2011] and Choi et al. [2012] use drawn stick-figures to query a database of human poses. Such databases can be difficult to obtain for custom skeletons, especially of non-humanoid or non-realistic characters. Reliance on databases inherently biases the reconstructed poses toward more frequent database instances. In contrast to stick figures, gesture drawings are unambiguous to human observers, motivating our approach. At the same time while matching 2D stick figures to 3D skeletons is straightforward up to inherent ambiguity between symmetric limbs, matching characters to gesture drawings is an open and challenging problems we successfully address for the first time.

Small inaccuracies in 2D stick-figures can lead to large changes in the recovered 3D pose [Davis et al. 2003]. To improve accuracy Davis et al. [2003] advise artists to first draw a gesture or bubble sketch of the target posed character, and then use it to assist in positioning the stick-figure (Figure 21). Our work operates directly on gesture drawings and robustly overcomes artist inaccuracies by balancing image conformity against other perceptual cues (Figure 15).

3 Parsing Gesture Drawings

Gesture drawings are ubiquitously used by artists to clearly convey complex 3D poses.To understand and formulate the properties that make them effective, we combine observations from drawing tutorials, modeling research, and perception studies.

Figure 3: *Portion of a gesture drawing with annotated joint (blue) and part (red) contours.*

Anatomical Landmarks In a typical character drawing, most strokes depict projected contours, i.e. curves along which the normal to the posed character's body lies in the image plane. Unlike detailed drawings of geometric shapes, gesture drawings focus on depicting pose and motion; hence their contour strokes are often highly abstracted and only approximate the shape of the actual 3D contours. We note that gesture drawings employ idealized character anatomy, well described by a union of approximately cylindrical body parts connected by spherical joints [Hampton 2009; Blair 1994; Hogarth 1996] (see inset). Conse-

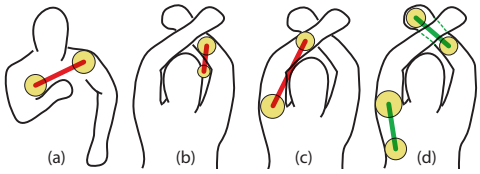


Figure 5: *Implausible bone locations that violate (a) adjacency, (b) orientation, or (c) crossing cues; consistent placements (d).*

quently, contours of body-parts surrounding skeletal bones are typically dominated by low-curvature lines. In contrast, joint contours in all views are well approximated by circular arcs whose radii are roughly equal to the body radius around the joints. These higher curvature joint contour arcs are most prominent next to bent or terminal (single bone) joints. As a consequence of this curvature difference, we speculate that humans can easily discern the likely locations of such prominent joints, or anatomical landmarks, in a gesture drawing, and use those to anchor the overall character pose. Since artists seek to communicate their target pose, they typically select views where multiple anatomical landmarks are visible and clearly depicted [Hale and Coyle 1991]. Clearly not all high-curvature contour segments correspond to joints (see the skirt "corners" in Figure 1); many drawn joints are not bent and therefore not easy to pinpoint; and multiple joints may have the same radii, making them hard to distinguish. Our algorithmic challenge is to discern the relevant markers on the drawing and to associate them with their corresponding joints.

Figure 4: *Contour-skeleton correspondences, with Gestalt continuous contours connected by dashed lines.*

Contour Consistency. As noted by Bessmeltsev et al. [2015], absent occlusions a typical character's contour is a single closed curve; each body part around a terminal bone (bone with a terminal joint) is outlined by a single contour segment, while parts around interior, or non-terminal, bones define two outline segments, one on each side of the bone; and adjacent segments along the contour correspond to adjacent skeletal bones (see inset). In the presence of occlusions, the *Gestalt continuation* principle [Koffka 1955] indicates that viewers complete the drawing by mentally connecting pairs of end-points of partially occluded curves (T-junction stems) by invisible contour sections if they can be smoothly connected (Figure 4). In this scenario, the properties above continue to hold once these invisible contour sections are taken into account. In this general case, terminal bones correspond to a single sequence of (one or more) Gestalt continuous curves, and interior bones correspond to two such sequences - one on the left and one on the right. Adjacent segments along the same contour stroke still correspond to adjacent bones, while bones joined by a valence two joint correspond to either immediately adjacent, or Gestalt-continuous, left and right contour segments. In addition to reflecting skeletal *adjacencies*, body part contours are consistently *oriented* with respect to their corresponding skeletal bones - a body's surface and consequently its contours clearly separate inside from outside (Figure 5b). Since body mass typically surrounds the bones, contours rarely *cross* 2D bone projections (Figure 5c). Viewers are known to rely on domain priors when deciphering drawings, and therefore we expect them to indirectly leverage this set of contour-bone *consistency* expectations when parsing gesture drawing and matching joints to landmarks.

Simplicity Previous graphics research (e.g. [Xu et al. 2014]) had heavily relied on insights from Gestalt psychology [Koffka 1955] which points to a viewer preference for *simple* or regular drawing interpretations. While some of these works (e.g. [Xu et al. 2014]) focus on generic *regularities* such as symmetry or parallelism, others (e.g. [Bessmeltsev et al. 2015]) highlight domain-specific simplicity priors. We speculate that viewers leverage both regularity and naturality when interpreting gesture drawings: they choose more likely or *natural* character poses among those consistent with the drawn contours (Figure 8), and use *regularity* cues, particularly symmetry, when presented with different ambiguous inputs (for instance when mentally completing partially occluded poses, such as the hands of the character in Figure 8, or the fetal pose in Figure 14, top row).

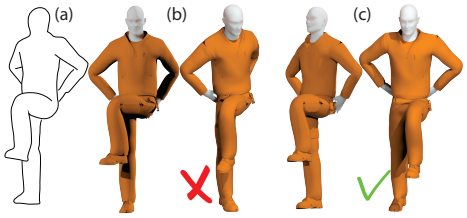


Figure 8: *Less natural (b) and more natural (c) interpretations of a drawn pose (a) (leg bent sideways vs forward).*

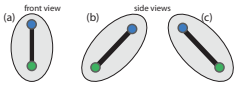


Figure 6: *Depth ambiguity*

Depth In general, an infinite number of 3D geometries have the same 2D projection. However, for each individual bone of a known length, if the 2D positions of its end-joints are known, the z-difference between the bone end-points is fully determined; what needs to be determined is their depth order (see inset).

Figure 7: *Occlusion types.*

While the simplicity priors discussed above often help viewers to resolve order ambiguities, contours of posed characters taken from a poor viewpoint (Figure 2d) remain ambiguous. Consequently, artists are consistently advised to strategically select descriptive views [Eissen and Steur 2011], and specifically to avoid views with large uneven foreshortening. Our observation of artist-generated gesture drawings suggests that in selecting views they also strategically use occlusions to clarify depth ordering, and add suggestive local, intra-part, occlusion contours (see inset) to further clarify local depth order.

Inaccuracy Experiments [Schmidt et al. 2009] show that even trained artists fail to correctly draw foreshortened shapes and frequently exaggerate perspective scaling effects. As indicated by prior work on interpreting design sketches [Xu et al. 2014], viewers are adept at mentally correcting such errors by biasing the envisioned solutions toward more simple and less *foreshortened* interpretations. In the context of gesture drawings, we observe that while viewers use landmarks to anchor the envisioned pose, they mentally tweak the locations of these landmarks in favor of such simpler pose interpretations.

4 Framework Overview

The input to our method is a rigged and skinned character model, in a bind pose, and a roughly same scale vectorized gesture drawing. As artists typically create the gesture drawings using the character as a reference, scale similarity is easy for them to satisfy; al-

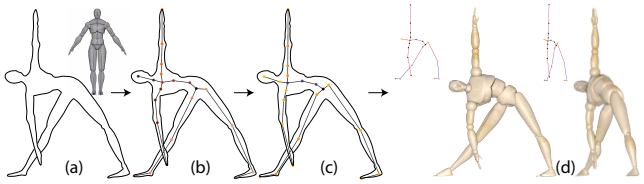


Figure 9: *Overview: (a) algorithm input; (b) discrete 2D joint embedding; (c) optimized 2D embedding; (d) 3D skeleton (color visualizes depth) and posed model.*

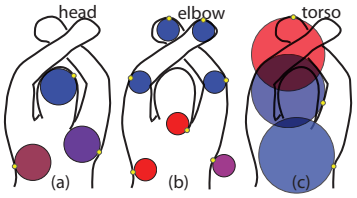


Figure 10: *Joint cost visualization. Here the color shows the matching cost on a scale from red (poor match) to blue (good).*

ternately, manually scaling drawings generated independently from the character model takes seconds for both experts and amateurs. As is typical of skeletal posing systems, the pose of a rigged character is fully determined by the positions of its joints. We compute the joint positions that best reflect the depicted pose using the following steps (Figure 9).

Joint-Contour Matching We first match drawn contours against the body parts they describe, and place a projected character skeleton in the image plane so that its surrounding body contours roughly align with their matched drawn ones. We formulate the matching as a computation of optimal joint locations along the contours. As the continuous solution space of all possible joint locations is too large to operate on efficiently, we discretize the problem by considering only a finite set of *potential joint locations* on the 2D drawing. We associate each possible joint location with an unary assignment probability derived from our anatomical landmark prior (Section 5.1), and associate binary and ternary probabilities for assignments of adjacent pairs and triplets of joints based on consistency, simplicity, and low foreshortening priors (Section 5.3). The resulting discrete optimization problem can be cast as a High-Order Tree-Structured Markov Random Field (MRF) problem [Koller and Friedman 2009]. We then minimize this combined cost function subject to additional global constraints imposed by the drawing (Section 5.4). Adding these constraints makes the general assignment problem NP-hard; however, as we demonstrate, our greedy solution framework works well in practice (Section 5.5).

2D Pose Optimization Our discrete solution considers only a finite set of possible joint locations; accordingly while it provides a good estimate of the joint locations and joint contour correspondences, the final joint placement may be locally sub-optimal. We consequently use continuous location optimization to further improve this solution and compute joint locations that best capture the artist intent (Figure 9b-c).

Full Pose Estimation. We proceed to fully pose the character by assigning 3D positions to its joints, further adjusting the joint 2D positions when necessary. We note that exact 2D joint locations are more sensitive to artist errors than bone directions and lengths, and consequently rely on the latter when recovering the full pose. We seek poses that satisfy the ordering cues provided by occluding contours in the gesture drawing, and which balance preservation of the bone directions and 2D lengths, estimated from the drawing, against our expectations of simplicity and foreshortening minimization.

We formulate joint positioning as a constrained energy minimization problem, then obtain the minimum by recasting the energy in term of twist variables [Bregler et al. 2004] and using a Newton-type solution method that follows the approach of [Gall et al. 2010].

5 Character-Contour Correspondence

Initialization To evaluate anatomical landmark correspondences, we need to associate a likely contour arc radius for each character joint. To compute the radius we use a variation on the method of Thierry et al. [2013] to fit a sphere to the region on the character mesh surrounding the joint. While many joints are well approximated by spheres, some parts of a character, such as the palm of the hand, are more elliptical and consequently have a range of plausible contour arc radii. Given the extracted mesh region around each joint we therefore use PCA to obtain the maximum and minimum radii by computing a bone-aligned bounding box of the mesh region next to the joint, and using its non-bone-aligned dimensions as minimum and maximum radii values, and use a discrete set of joint radii with a step of ε within this range in subsequent computations. We set ε to 2% of the drawing bounding box and use it as the default discretization density throughout the discrete solution.

To facilitate the computation and evaluation of contour consistency in the presence of occlusions, we preprocess the contours to detect Gestalt continuous strokes. We use the continuity test described in [Bessmeltsev et al. 2015]: given each pair of strokes, we connect their end-points with a straight line and measure the angles between this line and the stroke tangents. A pair of strokes is classified as Gestalt Continuous if both angles are below the 18° threshold identified in perception literature [Hess and Field 1999]. For each pair of drawing strokes we test all four end-point configurations. When strokes are deemed continuous we retain the connecting line as a *Gestalt bridge* between them. We consider each pair of strokes connected by a bridge as a single *bridged* contour.

5.1 Solution Space

As previously noted, artists approximate the contours surrounding joints as circular arcs centered at the joints whose radius reflects the distance from the character joint to the surrounding surface. We therefore expect joints with visible contours to be located approximately a radius distance away from these contours along the contour normal (Figure 10). We use this observation to generate potential locations for joints with visible contours. We uniformly sample the input drawing contours at ε-intervals, and treat the samples as potential tangential contact points for joint circle placement. For each sample point we consider the options of placing the circle on either side of its contour, conceptually duplicating all samples into left and right instances. We compute *potential* joint *locations* by placing each joint along the normal to the contour at the sample at an offset equivalent to it's circle radius (Figure 10).

Character joints may be entirely occluded (e.g the man's palms in Figure 8). To be able to plausibly place such joints, we sample the bounding box of the drawing using a regular grid with density equal to ε and add these samples to the discrete solution space.

5.2 Unary Assignment Cost

We compute, for each joint, the likelihood that it is placed at each potential location. The grid-based locations are assigned the maximal assignment cost of 1 since, absent information to the contrary, we expect contours associated with joints to be visible. For tangential locations, we aim to match appropriate joints to corresponding anatomical landmarks, and hence prioritize placements where sections of the contours are well aligned (in terms of both location and normal) with the joint's circle. Since non-terminal joints are often adjacent to multiple contour segments on different sides of the circle (Figure 10), our evaluation looks at all contour samples close to the circle and not just those immediately next to the *originating* tangent sample. Since humans rarely draw perfect circular arcs, we do not expect perfect alignment; to evaluate fit between a joint i and a potential location P_a^i we therefore measure the portion of a circle with radius r_i centered around P_a^i that approximately aligns with the contours using simple distance and normal thresholds. Specifically, we uniformly sample the circle and count the percentage of circle sample points s_c that have nearby contour samples s with contour normals n_s close to the circle sample point normals:

$$T(P_a^i) = \{s_c : |\|s_c - s\| < \min(\varepsilon, \frac{r_i}{2}) \text{ and } \angle(s_c - P_a^i, n_s) < \alpha\}$$
$$C(i, P_a^i) = 1 - \|T(P_a^i)\|/N \quad (1)$$

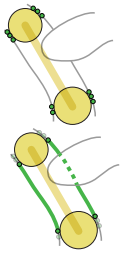
Here N is the number of samples on the circle. The angle threshold α is set empirically to 15°. When a contour matching a terminal joint is visible in the drawing, we expect a non-negligible portion of the contour to closely align with the joint's circle. We found this threshold based solution to work better than using a falloff function that depends on how close the contours are to the circle. We consider terminal joint locations to be reliable if at least 15% of their osculating cycle is matched by the contours, and assign the maximal cost of 1 to locations that do not pass this threshold. For each joint i and a potential assigned location P_a^i , in addition to the cost we store the originating contour sample s_a^i and the set of all contour samples S_a^i that satisfy the alignment threshold.

Position Consolidation. Near high-curvature regions on the contours, we typically encounter several potential low cost joint locations for a given joint which have nearly identical sets of well-aligned contour points. To reduce the solution space during computation we consolidate these potential joint locations into one, selecting the location whose originating sample lies closest to the stroke's curvature extremum.

5.3 Assignment Compatibility

Our compatibility term is designed to promote contour consistency, and to weakly encourage less foreshortened and more natural solutions.

Bone Contours. Each pair of position assignments for the end-joints of a bone indirectly defines the contour segments corresponding to this bone (Figure 4). Given a pair of such assignments, we compute the potential bone contour segments defined by these assignments as follows (see inset).

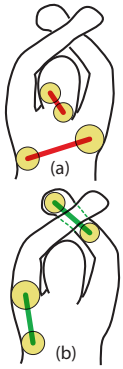


We consider all pairs of well aligned samples, where each sample is associated with a different end-joint. If the two samples lie on the same contour, or on contours connected via bridges, we associate the contour segment or segment chain between them with the bone. We trim the segments by selecting the two samples, one in each joint's set, that are closest to one another along this shared contour as segment end points. We use the computed bone segments to assess the compatibility of the bone's end-joint assignments. Note that occlusions or poor assignments may lead to bones with no corresponding contours.

Consistency. We explicitly prohibit inconsistent assignments where a bone's end-joints lie on opposite sides of the bone's contour, violating our orientation prior. Since a bone is expected to be inside the body part it anchors, it typically should not cross its associated contours. We use a consistency penalty cost C_c, which is set to 1 if a bone's 2D projection intersects any of its associated contour segments, and is 0 otherwise. We use a penalty instead of a hard constraint to account for drawing inaccuracies and sampling artifacts.

We prefer assignments where bones are associated with at least one, either simple or bridged, contour segment. Moreover, we aim for adjacent bones to be associated with the same continuous contour. We encode both preferences by focusing on the contour associated with the originating samples of the end-joint assigned locations: we leave the consistency cost C_c unchanged if a pair of end-joints of a bone are assigned locations with the same originating single or bridged contour, and set it to 1 otherwise.

Bone Contour Conformity We expect the contour segments associated with bones to have relatively low-curvature (see inset). To evaluate contour conformity, we measure the ratio between the length of each bone segment and the Euclidean distance between its endpoints:



$$C_{cf}(i,j) = 1 - e^{-(1-L_c/L)/2\sigma^2},$$

where L_c is the length of the contour segment and L is the Euclidean distance between its end-points. We empirically set $\sigma = 4\%$ of the bounding box diagonal. If a bone has multiple associated contour segments, we repeat the cost computation and, to be conservative, use the lower of the two costs as the conformity cost. If the joints have no shared bone contours, we set the cost to 1.

Pose Preferences. We assign a per-bone cost term for each assignment of its end-joints to a pair of potential positions, based on the difference between the bone length and the image-space distance between the two positions. We expect the artist to select views where the drawn body parts, and consequently bone projections, undergo relatively small foreshortening; we therefore weakly penalize foreshortening when it occurs. While real character bones do not stretch, artist drawings can contain errors in character proportion description. We therefore tolerate assignments where the image-space distance is larger than the respective bone length, but penalize such assignments with a large penalty cost. The combined cost is:

$$C_l(i,j) = \begin{cases} 1 - e^{-(l'_{ij} - l_{ij})^2/2\sigma^2}, & \text{if } l'_{ij} > l_{ij} \\ 1 - e^{-(l'_{ij} - l_{ij})^2/2(\sigma/3)^2}, & \text{otherwise .} \end{cases} \quad (2)$$

where $l'_{ji} = \|P_a^i - P_a^j\|$ and l_{ij} is the bone length. We use the same σ as for bone conformity. We evaluate the difference between the two lengths rather than their ratio, since ratio-based costs are extremely sensitive to artist errors on short bones.

We encode our expectation for simpler, more natural character poses, depicted from a descriptive view, as a preference for 2D joint angles in the output pose that are close to their bind pose counter-

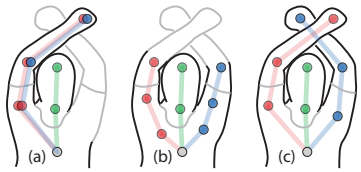


Figure 11: *Full solutions: (a) contains overlaps; (b) poor coverage; (c) preferred.*

parts:

$$C_n(i,j,k) = 1 - e^{-(\gamma-\gamma')^2/2\sigma_a^2}$$

Here γ and γ' are the current and bind pose angles respectively between pairs of emanating bones (i,j) and (j,k) at a joint j. We set σ_a to $\pi/3$ if the 3 involved joints share an originating contour, and $\pi/6$ otherwise, enforcing a stronger preference for the bind pose angle when there is no clear contour suggesting 2D bone directions, and a weaker preference for bind pose angles when the adjacent bones follow one continuous contour and the 2D bone direction is well-suggested. These costs are measured for each triplet of adjacent joints. This term can be replaced by more advanced anatomical machinery used in prior work for predicting plausible angles: for instance, if multiple reference poses are provided, one can look at the smallest angle difference across these poses.

Combined Local Cost Function. Combining the different terms above, the cost for assigning a pair of bone end-points i and j to a pair of locations is measured as

$$E(i,j) = 1-(1-C_l(i,j))(1-C_{cf}(i,j))(1-W_cC_c(i,j)). \quad (3)$$

We empirically set the consistency penalty weight to $W_c = 0.9$. The combined energy function encoding all local preferences for a given assignment of joints to point locations is

$$E_{match} = \sum_i C(i,P_a^i) + \sum_{i,j} E(i,j) + \sum_{ijk} C_n(i,j,k) \quad (4)$$

where the first term sums the per-joint assignment costs, the second sums the per-bone costs and the third considers the joint triplet costs. All terms have equal weight.

5.4 Global Consistency

In addition to the local criteria above, when evaluating the plausibility of a skeleton embedding we need to evaluate the likelihood of the overall contour-to-skeleton assignments it imposes (Figure 11). In addition to the bone-segment correspondence computed earlier, this task requires a joint-contour correspondence. We compute segments associated with joints as follows. For terminal joints we consider the longest segment delineated by its aligned samples which does not overlap the segments associated with its bone. For interior joints we consider each pair of bones emanating from the joint. If the bones are associated with segments on the same contour, we associate the contour segment in-between them with the joint (Figure 4).

In real life, projected visible contours of different character body parts can *overlap* only if the two parts are in contact (i.e. on opposite sides of the contour), or if one is both perfectly parallel to and occluding the other (Figure 14, top row). We therefore test whether any pair of same-side contour segments associated with disjoint bones or joints overlap and, if they do, this configuration is assigned a high penalty score, empirically set to 10 (Figure 11a).

In a drawing that contains only contours of body parts surrounding skeletal bones, a valid solution must associate all contours with some bone or joint. In practice our drawings can and do occasionally contain extra curves, e.g the cat and horse ears in Figure 14. Thus instead of full coverage, we seek for a sufficient one, requiring coverage of over 85% percent of the contours (Figure 11b-c). We note that when the soft non-overlap constraint is satisfied, our local energy terms implicitly encourage coverage maximization, since we penalize joints not being matched to contours and discourage undesirable foreshortening. We incorporate coverage constraints into our framework as discussed in Section 5.5.

Our local energy does not clearly distinguish between fully or partially symmetric solutions. While hard to penalize locally, partial symmetries (e.g. left arm and right leg mapped to the same side of the spine) are easily detected on a complete solution by evaluating the degree of *twist* the spine must undergo to accommodate them. While twist can be intentional, we expect it to be clearly indicated by the contours, with the undesired "untwisted" solution in these cases violating consistency constraints. We differentiate between fully symmetric solutions by observing that, all things being equal, artists strongly prefer views where the face of the character is clearly visible. We similarly use this *frontal* preference in our global pose evaluation.

5.5 Solver Mechanism

Optimizing E_{match} alone without addressing global preferences can be cast as a classical tree-structured high-order Markov Random Field (MRF) problem by translating our cost terms into probabilities, and optimized efficiently using standard techniques [Koller and Friedman 2009]. Unfortunately, we are not aware of any standard mechanism that allows us to incorporate the coverage constraints into such frameworks; the general problem of maximal *a posteriori* estimation over a Markov Random field is a classical NP-hard problem [Shimony 1994]. Instead we develop a simple domain-specific method that works well on all our inputs. We note that, on typical gesture drawings, for terminal joints our unary cost computation produces only about a dozen possible assignments with less than maximal cost; furthermore, our desired assignment is expected to match most terminals, with the exception of occluded ones, to one of these below maximum cost placements. Because of our stringent contour consistency constraints, given the correct assignment of terminals, using the basic E_{match} optimization for assigning other joints results in the desired global solution. Clearly we do not *a priori* know what this correct terminal assignment is; however, given the small number of terminals (typically six or less) and the small number of placement choices for them, an exhaustive search of all possible alternatives is a practical option.

This search can be further sped up by traversing the different alternatives in a strategic order. Specifically, we order all possible terminal assignments based on the sum of their unary costs, and then process them in increasing cost order, penalizing assignment combinations where terminal assignments violate the non-overlap constraints and placing them at the end of the queue. For each terminal assignment we then optimize E_{match} on a reduced set of joints and with a reduced solution space. Specifically, when a terminal joint has a below maximum cost assignment, we remove this node from the solved-for joint set and update the unary and binary costs of its neighboring vertex to reflect the selected assignment. We let the optimization determine the best assignment for terminal joints associated with the maximal cost, but remove all assignments with below maximum cost from their solution space. If the located solution satisfies all our constraints, and in particular if it produces over 85% coverage, we stop the iterations.

The same coverage can sometimes be produced by a permutation of the desired terminal placements; however different permutations lead to different minima of matching energy E_{match} which may better satisfy our preference for more front facing and less twisted solutions. We thus process all partially and fully symmetric permutations of the obtained solution, and select the least twisted and most front facing one from among those solutions that satisfy all our constraints.

6 2D Pose Optimization

While our discrete solver correctly captures the overall contour-joint correspondences, it operates on a finite set of potential positions and thus may end up generating imperfect joint placements (Figure 9b). Moreover, to enable an efficient solutions, our discrete formulation assumes all joints are fully flexible. In real models, many joints have a reduced set of degrees of freedom (DOFs), with pelvic and shoulder joints typically supporting only rigid transformations. To address both issues we iterate over the joints to further optimize their positions and enforce the allowable degrees of freedom. For each joint we use a local random walk to find a new location that improves the overall matching energy (Equation 4) while constraining the joint to remain on the same side with respect to all nearby contours, and disallowing moves that violate consistency or introduce overlaps. For joints with a reduced DOF set, we then recompute the positions of the joint and its immediate neighbors which satisfy the DOF constraints and are maximally close to the current ones, using an ICP variant. Specifically, given the current 2D locations of a joint and its neighbors, we search for a 3D transformation of these joints in the bind pose that satisfies the DOF constraints while maximally aligning the 2D coordinates of each joint and its current location. We repeat the two steps until convergence.

7 Full Pose Optimization

Once we have generated a 2D skeletal embedding, we associate a depth value with each joint by leveraging viewer expectations of simplicity and weak foreshortening. In this process we also refine image plane joint positions to correct drawing and 2D estimation inaccuracies. In our computations we assume an orthographic projection since, as noted by [Xu et al. 2014], estimates of artist intended perspective are highly unreliable. Our solution is based on three key observations. First, we note that even small inaccuracies in depicting body proportions, due to inexact foreshortening, inaccurate perspective, and other artifacts, accumulate to form large errors in 2D joint placement. Therefore, rather than minimizing absolute 2D solution displacement compared to the 2D embedding, we encode conformity with this embedding in terms of slopes and lengths of projected bones. Second we note that human observers are know to underestimate foreshortening in drawings [Schmidt et al. 2009], a fact that often causes artists to exaggerate it [Hogarth 1996]. Consequently, foreshortening predictions based directly on drawn body-part lengths may be inaccurate. In our observations, viewers rely on relative rather than absolute foreshortening when predicting a character's pose from a drawing - even when presented with a reference model. Consequently, when predicting the degree of foreshortening per bone, we similarly take into account relative foreshortening as compared to other bones. Our last observation is that while we seek for natural poses, i.e. those closer to the input bind pose, minimizing this difference directly is problematic as many drawn poses are quite far from the input one by design. For this reason, we do not explicitly consider the distance to the bind pose in our optimization. Instead we use the bind pose as an initial guess for the solution and limit the step size in each iteration so that our final pose gradually evolves from the bind pose. In doing so, we indirectly guide our final solution towards a more natural pose by searching for a smooth motion path from the bind pose to the final one.

Conformity We encode conformity to the estimated 2D skeletal pose as preservation of 2D bone slopes and lengths:

$$E_c = \sum_{(i,j)\in S} w_c(i,j)((P_i^y - P_j^y) - d_{ij}^y)^2 + ((P_i^x - P_j^x) - d_{ij}^x)^2 \quad (5)$$

where P_k are joint positions, S is the set of all skeletal bones, and d_{ij}^x, d_{ij}^y are the x and y differences between joint positions in the 2D embedding. To focus on relative rather than absolute bone projection preservation we set $w_c(i,j) = 1/l_{ij}^2$ where l_{ij} is the length of the bone (i,j).

Foreshortening When the 2D projected bone lengths l'_{ij} are fixed, the depth along each bone is fully determined by the difference between the 3D and 2D projected bone lengths: ($d_{ij}^z = \sqrt{l_{ij}^2 - l'^2_{ij}}$). However image space lengths are sensitive to artist errors, as well as scale mismatches between the character model and the drawing. Leveraging our previous observations about human preference for foreshortened interpretations, we consequently combine conformity with a foreshortening minimization term which, together with the regularity constraints below, aims to mitigate drawing inaccuracies:

$$E_v = \sum_{(i,j)\in S} w_v(i,j) \cdot (P_i^z - P_j^z)^2. \quad (6)$$

The weights $w_v(i,j)$ are determined by the anticipated foreshortening of the bone (i,j):

$$w_v(i,j) = \begin{cases} e^{-\frac{(f_{ij}-f_{avg})^2}{2\sigma_f^2}}, & \text{if } f_{i,j} < f_{avg} \\ 1.0, & \text{otherwise} \end{cases} \quad (7)$$

Here $f_{ij} = l'_{ij}/l_{ij}$ is bone foreshortening and f_{avg} is the average bone foreshortening for the entire character in the 2D solution. This weight is a monotonically decreasing function of the 2D-3D length ratio and is maximized when this ratio is equal to or larger than the average across the drawing. We view a ratio below 0.6 of the average as intentional foreshortening and consequently force the weight of the foreshortening minimization term drop to zero for such ratios by setting $\sigma_f = 0.2$ using the three-sigma rule.

Regularity Previous work on the interpretation of drawings (e.g. [Xu et al. 2014; Bessmeltsev et al. 2015]) has discussed numerous domain-specific regularity criteria. In our work we found four key regularity cues which viewers expect to hold when envisioning drawn poses: parallelism, symmetry, contact, and smoothness. We use the 2D embedding to detect near-regular relationships and then strictly enforce them in 3D. For each pair of bones (i,j) and (m,n) with roughly parallel 2D projections (within 10°), we enforce their 3D bone directions to be the same: $P_i - P_j = l_{ij}/l_{mn}(P_m - P_n)$. Similarly, if two symmetrical limb bones are nearly symmetric around the spine plane, we force exact symmetry - since symmetry is detected in 3D, we enforce this constraint in a post-process step. We also note that human observers expect close 2D adjacencies, specifically contacts observed in 2D, to be preserved in 3D. We therefore detect pairs of adjacent 2D joint contour segments and constrain the distance between their corresponding 3D joints. Lastly, we note that gesture drawings typically aim to convey aesthetic poses [Guay et al. 2013]. Motivated by Guay et al., we fit a quadratic polynomial spline to each skeletal limb in the 2D embedding; if all joints along the limb are deemed to be close enough to this spline, i.e. within half each joint's radius from it, we

add soft constraints attracting them toward corresponding spline locations.

Joint Ordering. The drawing contours define two types of occlusions, iter- and intra- part (Figure 7). Inter-part occlusions, such as an arm in front of a body, indicate that at a particular point along one bone, the body part surrounding this bone is in front of a particular location on the body part around another bone. We encode these using relative locations on the participating bones:

$$P_i^z t_{ij} + P_j^z (1 - t_{ij}) + R_{ij}(t_{ij}) < P_k^z t_{kl} + P_l^z (1 - t_{kl}) - R_{kl}(t_{kl})$$

Here the two participating bones are (i, j) and (k, l), t_{ij} and t_{kl} are the linear parameters of the occluded and occluder points and R_{ij} and R_{kl} are the corresponding body part radii at these points.

Intra-part occlusions, depicted via local contour T-junctions, encode pairwise joint ordering between end-joints i and j of individual bones. The joint associated with the stem of the "T" is expected to be farther away than the one associated with its top. To enforce these relationships we add the inequality constraint:

$$P_i^z < P_j^z.$$

Solver We minimize $E_c + E_f$ subject to the simplicity and order constraints detailed above. While our posing criteria are for convenience expressed via positions, using positions as optimization variables is problematic, since preserving fixed bone lengths using a position based formulation requires quadratic constraints, which are known to be hard to operate on [Gall et al. 2010] . Instead we follow the standard approach used in kinematics and robotics and represent our 3D pose in terms of *twist coordinates* θ_{ij} [Bregler et al. 2004]. We then use a solution method advocated by Gall et al. [2010], who represent vertex positions via twists, and use a Taylor expansion to linearize the resulting expressions. Using such linearizion we formulate the optimization of E as a sequence of constrained quadratic optimizations. We augment the quadratic function being minimized at each iteration with a stabilization term aimed at keeping the new solution close to the previous one:

$$\alpha \sum_{ij} (\theta_{ij}^n - \theta_{ij}^{n-1})^2 \tag{8}$$

Here the sum is evaluated over all twist variables θ_{ij} in the current n and previous $n-1$ iterations. We use a large $\alpha = 200$ to avoid introducing unnecessary and unnatural deviations from the bind pose. Note that since the stabilizer is computed with respect to the previous solution, this process allows for slow, but arbitrarily far, deviation from this pose. The resulting quadratic optimization with ordering constraints is solved at each iteration using the Gurobi optimizer (www.gurobi.com). Since we have just a few dozen variables the entire process takes on average 30 seconds.

8 Validation

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

Ground Truth and Perception Comparison. We validate our method on Ground Truth (GT) data, by posing two models into complex poses (Figure 12) and using retraced projected occlusion contours as inputs to our method together with the same models in neutral bind pose. Our results closely resemble the original.

Our method aims to recover the viewer-perceived pose from the drawings; therefore a more interesting test is to compare our poses to viewer perceived ones. We performed this test using the same data, by providing our inputs to two 3D modeling experts and asking them to pose the models into poses depicted by the drawings.

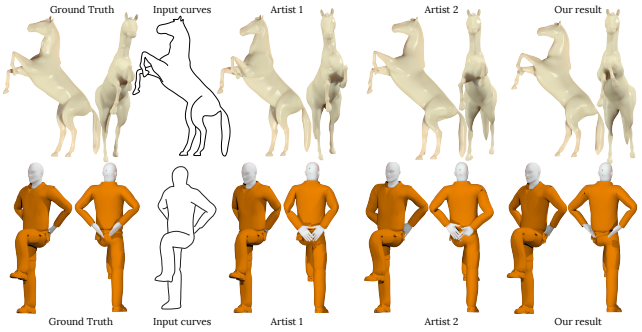


Figure 12: *Comparing our results to GT data and artist modeled poses. We use as input the projected contours of the posed GT models combined with their bind posed originals (Figure 14) to automatically create poses qualitatively similar to both GT and artist results.*

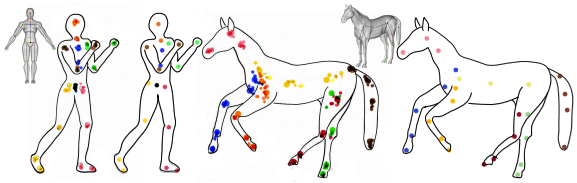


Figure 13: *Overlays of viewer created skeleton embeddings (lines removed for clarity) and our results on same inputs.*

The result (Figure 12) are visually even more similar to ours than ground truth. We showed each artists the ground truth models, our results and the result produced by the other artist, without identifying which output was produced by which method, and asked "How well do these poses capture the artist intended pose?". Both assessed all the shown 3D poses as reflective of the drawn one, and one commented that our result was "the most natural". The artists required roughly 15 minutes to pose each model, 5 to 10 times more than our automatic posing times of 1.5 and 3 minutes.

Perceived 2D Skeletal Embedding. To evaluate consistency across viewers and to compare our algorithm with viewer perception, we asked 10 viewers to manually embed skeletons to match 4 gesture drawings. We provide viewers with 2D images of the models and skeletons in the bind pose, with joints clearly marked, and with bone chains numbered and colored with different colors to facilitate distinction between symmetric limbs. While viewers found the task conceptually easy, marking locations for all joints and connecting them took participants 5 to 10 minutes per drawing. Figure 13 summarizes the resulting embeddings on two complex inputs, with various user embeddings overlaid to visualize correlations across viewers. The full set of viewer results is included in the supplementary material. Viewer embeddings are largely consistent and agree very well with our algorithmic results, confirming that our method is built on solid perceptual foundations.

Qualitative Evaluation. We asked 3 artists and 6 non-experts to comment on our results. We showed them each pair of input and result separately and asked "How well does this 3D character pose capture the artist intended drawn pose?". All respondents agreed that our results successfully capture the drawn poses. Minor differences noted by two participants included: variation in geometric details beyond the control of a skeletal rig, such as extended vs contracted character belly in the yoga pose, Figure 14, top; and insufficient tightness of the cross-armed pose in Figure 14,bottom. The latter example is particularly challenging since the artist did not draw the actual character palms.

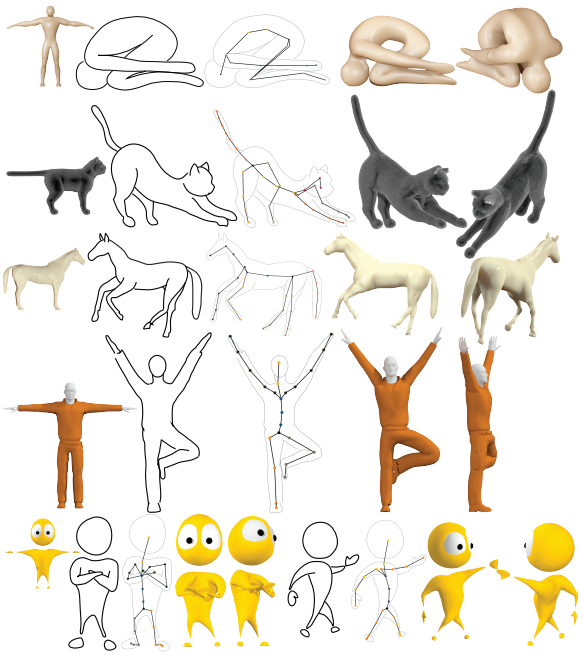


Figure 14: *Typical two-stage processing results. Left to right: input model, drawing, 2D skeleton fitting, output model.*

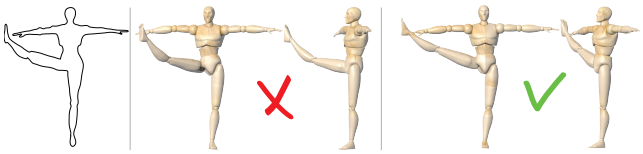


Figure 15: *(center) 3D posing using only drawing conformity, (right) full 3D solution.*

9 Results

Throughout the paper we have shown numerous examples of gesture posing using our method. These examples range from relatively simple occlusion-free and relatively flat ones, e.g. Figure 15, to the karate, cat, and dance poses which exhibit large foreshortening and complex occlusions (Figures 1, 14, 21). Our results extend beyond typical humanoid models attempted by previous 2D posing methods (e.g. [Davis et al. 2003]), to whimsical characters and animals (Figure 14). Across all examples our method believably reproduces the drawn poses. It seamlessly overcomes drawing inaccuracies, clearly visible in inputs such as the gymnastics poses in Figures 9, 15, 16 where the drawn limbs are consistently longer and skinnier in proportion to its torso than those of the character model.

Workflow. Most of our inputs were created using a traditional keyframing workflow, where the artists had a model in front of them and drew the poses with this character in mind (Figures 1, 14). Artists were given generic instructions to draw a gesture drawing of a given 3D model, in a pose performing a given action. They created their inputs using standard 2D editing tools and received no information about our system. We also evaluated an inverse workflow inspired by legacy drawings - tracing the strokes on existing gesture drawings and adjusting the character dimensions to roughly fit those (e.g. the karate sequence in Figure 21). This workflow can enable non-artists to create compelling poses and animations by re-using existing material and assets, but is likely to be more challenging as the character proportions are more likely to differ.

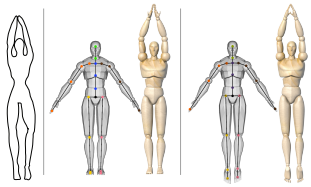


Figure 16: *Impact of different bind poses.*

Figure 17: *We capture poses consistent with artist intent even when the gesture drawings contain large errors in character proportions.*

Impact of Design Choices. Figures 5, 8, 11, and 15 demonstrate the importance of our algorithmic choices, highlighting what can happen if we omit one or more of the perceptual cues we employ. Figure 15 demonstrates the effect of our foreshortening and regularity terms on 3D pose reconstruction. Absent these terms, the posed character better conforms to the input contours, but the 3D pose becomes less predictable or natural. Figure 8 further highlights the distinction between more and less natural interpretations.

Figure 16 shows the impact on our results of using different bind poses. As demonstrated the bind pose impacts part orientation for cases where the drawing does not provide clear pose information, e.g. the feet of the character, or when the skeletal resolution is not sufficient to capture orientation details, e.g. the character's palm orientation.

Input Quality and Robustness. As noted previously, gesture drawings frequently depict inaccurate character proportions. Our inputs are no exception, yet without a ground-truth reference artist error is essentially impossible to measure. However, if we assume that our result is consistent with artist intent (as confirmed by our studies), then the mismatch between the contours of our models and the drawn strokes provides a good proxy for this error (Figure 17). We note that drawing errors are most evident for minimally foreshortened poses where 2D proportions should match the 3D ones, yet even for such inputs inspection points to significant artist errors. For example, while the pose on the left in Figure 17 was drawn by a professional, the proportions between the arms and legs are significantly off when compared to the 3D model. Following the assumption that our result is consistent with artist intent, a good numerical proxy for artist error is provided by comparing the ratios of limb skeletal-chain lengths to the lengths of the torso skeletal-chain projected to the image plane before, and after, full-pose optimization. For perfect inputs, such as the ground truth results in Figure 12, the before and after ratios (and lengths) are essentially identical. For the input freehand drawings, these ratios change by an average of 15%, indicating significant measurable inaccuracy. We also note that, for a typical model, the average difference between the fitted radius of a joint and the actual measured radius of its contour is 15%, and the worst difference is 40%. These evaluations confirm that our method produces good results even when the input drawings are inaccurate.

To further test sensitivity to proportion inaccuracy, we modified the lengths of the dancer's arms in the drawing in Figure 1,e and reposed the model using these modified drawings (Figure 18). We

Figure 18: *Altering the length of the arms in the dancer example (a) produces similar results to the original when the arms are 30% longer (b), and foreshortened results (as expected) when the arms are 30% shorter (c).*

Figure 19: *Occlusion test: Even with legs entirely hidden our method is able to plausibly recover the dancer pose.*

produced nearly-identical results when the drawn arms were 30% longer, and the expected foreshortened results (where the arms are at the same image space angle, but point inward) when the drawn arms were 30% shorter. To validate that our algorithm remains robust in the presence of significant occlusions, we removed the visible toes of the dancer model, making both legs fully hidden; our method produced a valid solution where the feet of the dancer touched the hem of the skirt (Figure 19).

Most of our inputs were generated directly in vector format, using tools such as Adobe Illustrator, either from scratch or by tracing over existing raster drawings, similar to the inputs used by sketch-based modeling frameworks (e.g. [Karpenko and Hughes 2006; Shtof et al. 2013]). We also tested our method on automatically vectorized raster input, generated using Adobe Illustrator (Fig. 20); even though the output of this method is not as clean as a manually vectorized input drawing, our method successfully creates a pose that reflects the artist's intent.

Comparison to Prior Art. Figure 21 compares our results against [Davis et al. 2003], the closest prior work in terms of 2D posing ability. While both methods recover qualitatively similar poses, we compute the pose fully automatically, and use only the drawings and the model in a bind pose as inputs. In contrast Davis et al. use a much more elaborate and time consuming process - users first draw a stick figure on top of the drawing, marking all 2D joint locations, then add extra annotations and select between multiple solutions to resolve input ambiguities. As our user study shows, while drawing a stick figure is not difficult it is nevertheless time consuming.

Parameters and Runtimes. All our results were computed with the default parameters listed in the text. For the multi-component model 'wynky' (Figure 14, bottom row) we disabled the crossing cost as on this model bones must intersect contours. Our method takes between 1 to 3 minutes to compute the output pose; roughly 60% of this time is spent on the 2D discrete embedding computation.

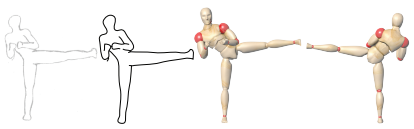


Figure 20: *Processing an automatically vectorized raster gesture drawing from a commercial software package. Left to right: input bitmap, auto-vectorized input, generated poses.*

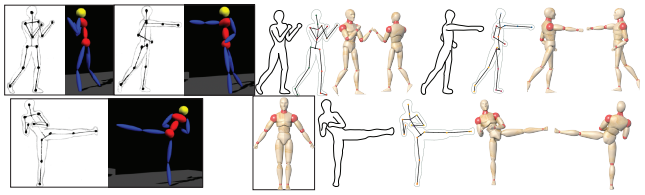


Figure 21: *(right) Davis et al.[2003] trace stick figures over gesture drawings and then pose characters semi-automatically. (left), We use the original drawings to automatically pose characters.*

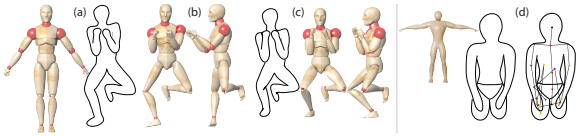


Figure 22: *Extreme mismatch in proportions between model and drawing (a) can lead to poor depth reconstruction (b); correcting the proportions in the drawing (c) corrects the reconstruction. (d) Ambiguous drawings using highly oblique views can cause our 2D pose estimation to fail.*

Limitations. Our method is inherently limited by the descriptiveness of the drawing (Figure 22). We rely on a combination of drawing and model's proportions to predict foreshortening. When the proportions of the drawn and posed characters are drastically different (in Figure 22a-b the drawn arms are much shorter and the drawn legs much longer than their model counterparts), our framework will by necessity misestimate the degree of output foreshortening. Once the drawing proportions are adjusted we correctly recover the intended pose (Figure 22c). Our pose estimation can fail when a gesture is not evident from the drawing itself, due to e.g. oblique views (Figures 2c, 22d), but can typically correctly recover the pose given a more descriptive view (Figure 1).

10 Conclusions

We have presented and validated the first method for character posing using gesture drawings. Our method leverages a set of observations about the key properties of gesture drawings that make them effective at conveying character pose. Using these observations we are able to first recover a 2D projection of the character's pose that matches the drawing, and then imbue it with depth. We are able to reconstruct convincing 3D poses, confirmed to agree with viewer expectations, from a single gesture drawing while robustly correcting for drawing inaccuracy.

Our work raises many directions for future research. It is empirically known that in artist drawings "errors of intent are inherent and unavoidable, and furthermore can be of significant magnitude" [Schmidt et al. 2009]. An interesting perceptual question would therefore be to explore when and where artist intent and viewer perception diverge, and at which point human observers are no longer able to correct for artist inaccuracies. The algorithmic impact of this exploration would provide more strict definitions of when and how pose recovery should deviate from conformity constrains. Our framework focuses on drawing cues, and it would also be interesting to explore how we can combine those cues with stronger anatomical priors on plausible character poses and other domain cues.

11 Acknowledgements.

We would like to thank Elena Uretskaya, Alina Zhuk, and Minchen Li for sketches, and Ksenia Popova, Svetlana Zhuk, and Alexander Pyatkov for enlightening conversations. The authors were supported by NSERC.

References

ATTNEAVE, F., AND FROST, R. 1969. The determination of perceived tridimensional orientation by minimum criteria. *Perception & Psychophysics 6*, 6, 391–396.

BAI, Y., KAUFMAN, M. D., LIU, C. K., AND POPOVIC, J. 2016. Artistic-dynamics for 2d animation. *ACM Trans. Graph. 35*, 4.

BESSMELTSEV, M., CHANG, W., VINING, N., SHEFFER, A., AND SINGH, K. 2015. Modeling character canvases from cartoon drawings. *Transactions on Graphics 34*, 5.

BLAIR, P. 1994. *Cartoon Animation*. Walter Foster Publishing.

BREGLER, C., MALIK, J., AND PULLEN, K. 2004. Twist based acquisition and tracking of animal and human kinematics. *Int. J. Comput. Vision 56*, 3, 179–194.

BUCHANAN, P., MUKUNDAN, R., AND DOGGETT, M. 2013. Automatic single-view character model reconstruction. In *Proc. Symp. Sketch-Based Interfaces and Modeling*, 5–14.

CHEN, Y., KIM, T., AND CIPOLLA, R. 2011. Silhouette-based object phenotype recognition using 3d shape priors. In *IEEE International Conference on Computer Vision, ICCV*, 25–32.

CHERLIN, J. J., SAMAVATI, F., SOUSA, M. C., AND JORGE, J. A. 2005. Sketch-based modeling with few strokes. *Proc. 21st spring conference on Computer graphics 1*, 212, 137.

CHOI, M. G., YANG, K., IGARASHI, T., MITANI, J., AND LEE, J. 2012. Retrieval and visualization of human motion data via stick figures. *Computer Graphics Forum 31*, 2057–2065.

CORDIER, F., SEO, H., PARK, J., AND NOH, J. Y. 2011. Sketching of mirror-symmetric shapes. *IEEE Trans. Visualization and Computer Graphics 17*, 11.

DAVIS, J., AGRAWALA, M., CHUANG, E., POPOVIĆ, Z., AND SALESIN, D. 2003. A Sketching Interface for Articulated Figure Animation. *Proc. Symposium on Computer Animation*, 320–328.

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Transactions on Graphics 27*, 1.

EISSEN, K., AND STEUR, R. 2011. *Sketching: The Basics*. Bis Publishers.

FAVREAU, J.-D., LAFARGE, F., AND BOUSSEAU, A. 2016. Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM Trans. Graph (Proc. SIGGRAPH 2016)*.

GALL, J., STOLL, C., DE AGUIAR, E., THEOBALT, C., ROSENHAHN, B., AND SEIDEL, H. P. 2009. Motion capture using joint skeleton tracking and surface estimation. *IEEE Computer Vision and Pattern Recognition Workshops*, 1746–1753.

GALL, J., ROSENHAHN, B., BROX, T., AND SEIDEL, H. P. 2010. Optimization and filtering for human motion capture : AAA multi-layer framework. *International Journal of Computer Vision 87*, 1-2, 75–92.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2D-to-3D modeling. *ACM Trans. Graph. 28*, 5.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graph. 23*, 3 (Aug.), 522–531.

GUAY, M., CANI, M.-P., AND RONFARD, R. 2013. The Line of Action : an Intuitive Interface for Expressive Character Posing. *ACM Trans. on Graphics*, 6, 8.

HAHN, F., MUTZEL, F., COROS, S., THOMASZEWSKI, B., NITTI, M., GROSS, M., AND SUMNER, R. W. 2015. Sketch abstractions for character posing. In *Proc. Symp. Computer Animation*, 185–191.

HALE, R., AND COYLE, T. 1991. *Master Class in Figure Drawing*. Watson-Guptill.

HAMPTON, M. 2009. *Figure Drawing: Design and Invention*. Figuredrawing.info.

HECKER, R., AND PERLIN, K. 1992. Controlling 3d objects by sketching 2d views. *Proc. SPIE 1828*, 46–48.

HESS, R., AND FIELD, D. 1999. Integration of contours: new insights. *Trends in Cognitive Sciences 3*, 12 (Dec.), 480–486.

HOGARTH, B. 1996. *Dynamic Figure Drawing*. Watson-Guptill.

IONESCU, C., PAPAVA, D., OLARU, V., AND SMINCHISESCU, C. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Analysis & Machine Intelligence 36*, 7, 1325–1339.

JAIN, E., SHEIKH, Y., MAHLER, M., AND HODGINS, J. 2012. Three-dimensional proxies for hand-drawn characters. *ACM Trans. on Graphics 31*, 1, 1–16.

KARPENKO, O. A., AND HUGHES, J. F. 2006. SmoothSketch: 3D free-form shapes from complex sketches. *ACM Trans. on Graphics 1*, 212, 589–598.

KOFFKA, K. 1955. *Principles of Gestalt Psychology*. Routledge & K. Paul.

KOLLER, D., AND FRIEDMAN, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

KRAEVOY, V., SHEFFER, A., AND VAN DE PANNE, M. 2009. Modeling from contour drawings. In *Proc. Symposium on Sketch-Based Interfaces and Modeling*, 37–44.

LELAND, N. 2006. *The New Creative Artist*. F+W Media.

LIN, J., IGARASHI, T., MITANI, J., AND SAUL, G. 2010. A sketching interface for sitting-pose design. In *Proc. Sketch-Based Interfaces and Modeling Symposium*, 111–118.

MAO, C., QIN, S. F., AND WRIGHT, D. K. 2005. A sketch-based gesture interface for rough 3D stick figure animation. *Proc. Sketch Based Interfaces and Modeling*.

NICOLADES, K. 1975. *The Natural Way to Draw*. Houghton Mifflin.

SAPP, B., TOSHEV, A., AND TASKAR, B. 2010. Cascaded models for articulated pose estimation. *Lecture Notes in Computer Science 6312*, 406–420.

SCHMIDT, R., KHAN, A., KURTENBACH, G., AND SINGH, K. 2009. On expert performance in 3D curve-drawing tasks. *Proc. Symposium on Sketch-Based Interfaces and Modeling 1*, 133.

SHIMONY, S. E. 1994. Finding maps for belief networks is np-hard. *Artificial Intelligence 68*, 2, 399 – 410.

SHTOF, A., AGATHOS, A., GINGOLD, Y., SHAMIR, A., AND COHEN-OR, D. 2013. Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum 32*, 2, 245–253.

TEKIN, B., SUN, X., WANG, LEPETIT, V., AND FUA, P. 2015. Predicting people's 3d poses from short sequences. In *arXiv preprint arXiv:1504.08200*.

THIERY, J.-M., GUY, E., AND BOUBEKEUR, T. 2013. Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Transaction on Graphics 32*, 6.

WEI, X., AND CHAI, J. 2011. Intuitive interactive human-character posing with millions of example poses. *IEEE Comput. Graph. Appl. 31*, 4, 78–88.

XU, B., CHANG, W., SHEFFER, A., BOUSSEAU, A., MCCRAE, J., AND SINGH, K. 2014. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics 33*, 4.

ZHAO, J., AND BADLER, N. I. 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics 13*, 4, 313–336.