Modeling Character Canvases from Cartoon Drawings

MIKHAIL BESSMELTSEV, WILL CHANG, NICHOLAS VINING, ALLA SHEFFER
University of British Columbia
and
KARAN SINGH
University of Toronto

We introduce a novel technique for the construction of a 3D character proxy, or *canvas*, directly from a 2D cartoon drawing and a user-provided correspondingly posed 3D skeleton. Our choice of input is motivated by the observation that traditional cartoon characters are well approximated by a union of generalized surface of revolution body parts, anchored by a skeletal structure. While typical 2D character contour drawings allow ambiguities in 3D interpretation, our use of a 3D skeleton eliminates such ambiguities and enables the construction of believable character canvases from complex drawings. Our canvases conform to the 2D contours of the input drawings, and are consistent with the perceptual principles of Gestalt continuity, simplicity, and contour persistence. We first segment the input 2D contours into individual body part outlines corresponding to 3D skeletal bones using the Gestalt continuation principle to correctly resolve inter-part occlusions in the drawings. We then use this segmentation to compute the canvas geometry, generating 3D generalized surfaces of revolution around the skeletal bones that conform to the original outlines and balance simplicity against contour persistence. The combined method generates believable canvases for characters drawn in complex poses with numerous inter-part occlusions, variable contour depth, and significant foreshortening. Our canvases serve as 3D geometric proxies for cartoon characters, enabling unconstrained 3D viewing, articulation and non-photorealistic rendering. We validate our algorithm via a range of user studies and comparisons to ground-truth 3D models, and artist drawn results. We further demonstrate a compelling gallery of 3D character canvases created from a diverse set of cartoon drawings with matching 3D skeletons.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Modeling*

Additional Key Words and Phrases: sketch-based modeling, 2D characters

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

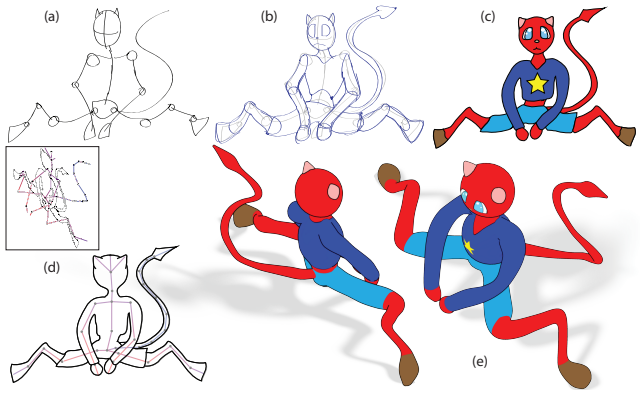


Fig. 1. Character drawings (c) are traditionally anchored around a skeleton (a), surrounded by generalized surfaces of revolution (b). We use the drawn character contours (d) and a corresponding 3D skeleton (red-to-blue coloring reflects near-to-far skeleton depth variation), to automatically compute a 3D canvas, employed to freely manipulate the character in 3D (e).

1. INTRODUCTION

Non-photorealistic 2D cartoon characters (Fig. 1(c)) are a mainstay of computer animation. Viewers appreciate the feel of hand-drawn art, while animators enjoy the flexibility and explicit control offered by this medium. Despite advances in 2D character manipulation [Jacobson and Sorkine 2011; Anime Studio 2013], this flexibility comes with the tedium of drawing numerous individual frames, and the cumbersome burden of managing view and temporal coherence. Recent research [Schmid et al. 2011] and practice [Paperman 2012] advocate the use of an underlying 3D proxy, or *canvas*, to enable easy 3D control over the view, pose, deformation and painterly rendering effects of cartoon characters. In current animation practice, such proxies are manually constructed using 2D cartoon drawings as a visual reference, and are then manually rigged to suitably designed skeletons for posing and animation [Maraffi 2003]. We introduce a novel approach for automatically constructing a rigged 3D character canvas directly from a single 2D cartoon drawing and a correspondingly posed, user-supplied, 3D skeleton, sidestepping the time-consuming manual modeling and rigging steps (Fig. 1(d,e)).

Our 3D canvases allow artists to directly articulate the drawn characters, generate convincing cartoon style character renders from alternate views (Fig. 1(e)), and provide support for various 3D effects created by drawing on and around the canvas (Fig. 24). Using a skeleton as an aid, our framework infers complex, complete character shapes from individual 2D drawings with significant contour depth variation, foreshortening, and multiple inter-part occlusions (Fig. 3 (left)) - a significant deviation from prior art, which

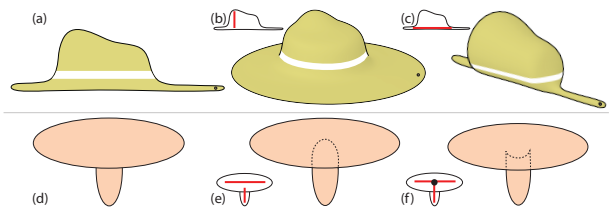


Fig. 2. Character contours alone (left) frequently do not provide sufficient information to conclusively determine 3D shape both on occlusion free (top) and partially occluded (bottom) inputs. A 3D skeleton, shown in the insets, resolves much of the ambiguity present in contours alone facilitating plausible shape interpretation.

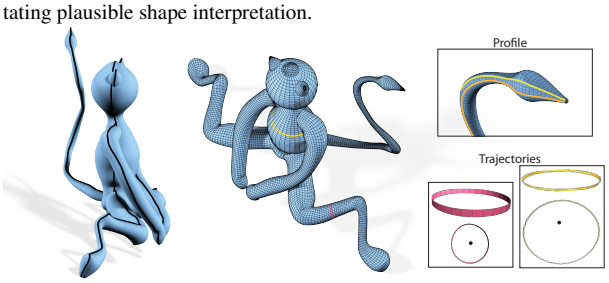


Fig. 3. The canvas (center) of the catwoman in Fig. 1: (left) thick black line shows reconstructed 3D contour curves, (right) insets visualize representative trajectories and profiles.

assumes drawn contours that are largely occlusion free, flat, and nearly perpendicular to the view direction (Section 2).

Our choice of input and subsequent construction methods are motivated by the observation [Hogarth 1996; Williams 2001] that cartoon character anatomy is well described by a union of body parts supported by a skeletal system, where each part is approximately a generalized surface of revolution (Fig. 1(a,b)). Artist-drawn character contours are inherently ambiguous (Fig. 2) and human observers frequently rely on either explicit familiarity with the drawn objects, or on semantic information encoded by additional drawing elements, such as facial features, to consistently interpret the 3D character shape. Such extra information is hard to enumerate or formalize algorithmically; our input skeleton, posed to reflect the character's structure, helps resolve these shape ambiguities.

Overview. Our guiding premise is that when artists create descriptive character drawings, they inherently rely upon and exploit the same perceptual principles that viewers use to lift drawings off paper and into 3D [Shao et al. 2012]. Following previous work, we rely on viewer preferences for *conformity* and *simplicity* (Section 3, Fig. 6) in reconstructing individual part geometry. Conformity is the unstated belief that the drawing is an accurate representation of the 3D character, and that the projected contours of the 3D characters will conform to the drawn contours in the input view and pose. Simplicity (or the law of Pragnanz [Koffka 1955]) states that viewers rely on symmetry assumptions as strong cues for image understanding [Hoffman 2000; Pizlo and Stevenson 1999]. Given viewer familiarity with character anatomy expected to resemble partwise surfaces of revolution, this principle suggests a strong viewer preference for envisioning body-parts with maximal rotational symmetry around the bone axis (Fig. 6 (b)).

We augment these two principles with observations about *Gestalt continuation* and shape *persistence* which help us parse complete, complex drawings and reconstruct coherent overall character shapes. To handle inter-part occlusions in the drawings, we exploit *Gestalt continuation* by noting that viewers resolve occlusions in line drawings by grouping together disjoint curves whose end-points can be smoothly connected [Koffka 1955] (e.g. the outlines of the tights of the catwoman in Figure 1). In reconstructing the complete character geometry from a single view drawing, we rely on the notion of shape, or contour, *persistence*. Contour persistence or the non-accidental view assumption [Nakayama and Shimojo 1992; Xu et al. 2014] indicates that viewers perceive the artist-selected view and pose as non-accidental and expect the drawn contours to be indicative of contour shape in alternate, and especially nearby, views.

We begin the modeling process by segmenting the input 2D contours into sections outlining individual body parts corresponding to the bones of the input skeleton. We resolve inter-part occlusions and group disjoint outline segments by leveraging skeletal depth and Gestalt continuation. We use the computed contour segmentation to generate the 3D canvas geometry, modeling body parts using generalized surfaces of revolution. While a canonical surface of revolution is defined by rotating a fixed planar *profile* curve along a circular *trajectory* around an axis, we account for a range of body shapes by supporting both more complex closed planar trajectory curves, and by allowing the profile shape to vary smoothly as the profile rotates around the part's bone or axis (Fig. 3). Supporting profile variation is critical for processing asymmetric part contours, such as those on the catwoman's hoofs. To balance conformity against simplicity we first refine the artist given straight-line skeleton to a geometric curve-skeleton [Cornea et al. 2005], and symmetrically locate it with respect to the artist-drawn contours. The surfaces of the different body parts are then optimized to form a unified 3D canvas centered around this curve-skeleton by enforcing conformity while balancing individual part simplicity against contour persistence across the canvas. Our final canvases are represented as quad-dominant meshes (Fig. 3 (center)) with explicit angular and axial parameterization which supports a range of texturing effects (Fig. 24).

Contribution. Our overall contribution is a framework for computing a believable 3D character canvas from two pieces of user input: a vectorized, single-view, descriptive, 2D contour drawing and a correspondingly created and posed 3D skeleton. Our key technical contributions are two algorithms derived from perception principles. First, we present a novel algorithm for correctly segmenting artist-drawn contours into body part outlines associated with individual skeletal bones, which can robustly handle multiple inter-part occlusions (Section 4). Second, we show how to use this segmentation to generate believable 3D character canvases which balance simplicity and persistence, allowing for variable contour depth and overcoming inaccuracies in skeleton posing (Section 5). Our resulting 3D character canvases are, as the name suggests, an ideal support structure for painterly strokes and cartoon rendering; however, they are not designed to capture the complex detail of realistic 3D character models.

Evaluation. We evaluate our approach in a number of ways (Sections 6, 7). We show that the task of positioning a 3D skeleton to match a 2D cartoon drawing is well-defined and intuitive, taking most artists less than ten minutes for typical cartoon drawings. We validate our segmentation algorithm via a user study, verifying that viewers consistently segment and associate character contours to skeletal bones and that this segmentation matches our algorithmic output. We reproduce ground truth 3D character shapes from a contour rendering and 3D skeleton, and compare our results to both ground truth and artist drawings created from the same in-

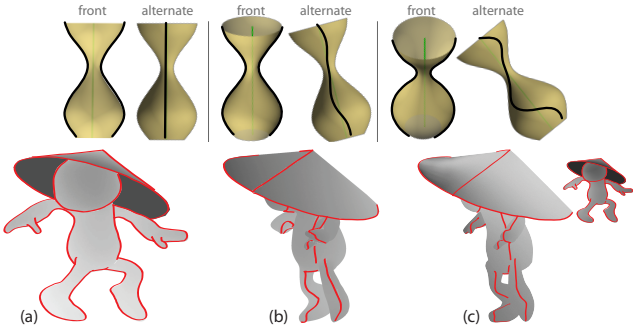


Fig. 4. Character drawings do not conform to the assumptions made in previous work. (top) The contours of a surface of revolution whose axis is not in the view plane are typically not planar. (bottom) The contours of a typical character include numerous occlusions; a single contour curve can consist of multiple part outlines (see combined left arm and torso outline in (a)) and as shown by the side view (b) the contour curves are far from planar or view aligned. Our method (c) successfully handles such inputs generating a character model largely similar to the ground truth input (b).

put and in the same views, validating that our results are visually similar to both. We show a variety of character canvases created from diversely sourced contour drawings and 3D skeletons, demonstrating our approach to be resilient to complex views, and poses with multiple occlusions and significant foreshortening. These canvases are illustrated using cartoon shading and other forms of non-photorealistic rendering, and are confirmed by artists to show plausible alternate-view renders of the drawn inputs. Finally, we compare our method to prior work, producing similar output quality with significantly less user-input.

2. RELATED WORK

Existing 2D animation tools support a limited range of 3D effects. They enable occlusions via explicit layering [Adobe Flash 2013; Anime Studio 2013; Hornung et al. 2007] and approximate out-of-plane deformation using non-uniform scaling that mimics foreshortening [Jacobson and Sorkine 2011]. These approaches use a fixed 2D contour topology and are inherently unsuitable for generic 3D manipulation which requires topological changes in character contour and reveals *a priori* occluded geometry (see Fig. 1(e)).

Recent industry-driven research (e.g. [Paperman 2012]) aims to enhance hand-drawn animation with 3D effects such as volumetric textures [Schmid et al. 2011; Bassett et al. 2013], or cloth simulation [Jain et al. 2012], by utilizing separately created 3D models or proxies in the background. Our work produces the underlying 3D proxy required by these techniques using a single 2D cartoon frame and an appropriately posed 3D skeleton as input. Our problem formulation is a novel intersection of skeleton-driven 3D modeling, sketch-based single-view modeling, and 3D character construction. We consequently position our work relative to prior art roughly classified into these three categories.

Skeleton-based 3D modeling. Organic 3D forms created using implicit functions defined around interactively manipulated skeletal primitives have existed for at least two decades [Bloomenthal and Wyvill 1997]. Recently, [Borosán et al. 2012; Bærentzen et al. 2014] proposed to simultaneously create 3D shapes and corresponding skeletons as a means to integrate skeletal deformation and interactive shape sculpting. While those methods do not require a skeleton as an input, neither framework can incorporate a complete 2D drawing into the modeling process. Since our canvas parts are modeled as generalized surfaces of revolution, they can be represented as implicit function primitives or polar annular meshes and can readily serve as input to these skeleton driven 3D modelers.

Sketch-based 3D modeling. Constructing 3D models using sketch-based approaches is a well-researched problem; see [Olsen et al. 2009] for a review. The majority of modelers are incremental, with the result strongly dependent on the order in which the artists draw strokes. Many systems employ a multi-view approach: users build models by drawing contour strokes in views where they are expected to lie in the view plane, or to project with little foreshortening onto the evolving geometry [Igarashi et al. 1999; Tai et al. 2004; Nealen et al. 2007]. Frequent view changes and incremental drawing order are critical when modeling characters using such approaches, as it is next to impossible for all contours of an articulated character to be entirely flat (Fig. 4). Our framework is independent of drawing order; it allows artists to freely sketch the characters they envision and to convert legacy sketches into 3D canvases without need for oversketching. Single-view incremental modeling approaches, e.g. [Cherlin et al. 2005; Gingold et al. 2009; Shtof et al. 2013; Chen et al. 2013] rely on drawing order and additional annotations to facilitate modeling of complex shapes. Cherlin et al. [2005] treat each new pair of contour strokes as 2D profiles defining a new generalized surface of revolution part, whose trajectory is either circular or manually defined. Gingold et al. [2009] interactively place tubular and elliptical 3D primitives to match artist drawn contours; as they note, their system does not directly use the 2D image. Olsen et al. [2011] use a combination of contour and annotation strokes and rely on the drawing order to incrementally create model parts. [Shtof et al. 2013] and [Chen et al. 2013] use predefined parametric primitive fitting and rely on user assistance, and an incremental modeling approach, to generate 3D models from annotated sketches and photographs respectively.

Instead of sketch annotation or primitive fitting, we use an input 3D skeleton, *a priori* necessary to articulate the model later on, to facilitate character modeling from a complete singe view sketch.

Sýkora et al. [2014] use user annotation to recover a bas-relief with approximate depth from a single sketch which they use to illuminate a 2D drawing. Their method makes no effort to create a 3D canvas that is plausible from all views; as they note, their proxy meshes "expose their approximate nature when rendered from side-views using a perspective camera".

A range of methods attempt to recover character models from single view sketches with no extra input [Cordier et al. 2011; Buchanan et al. 2013; Karpenko and Hughes 2006; Entem et al. 2014]. However, in doing so they, by necessity, enforce a range of strong simplifying assumptions. Buchanan et al. lift an occlusion-free 2D contour into 3D by placing circular arcs along a 2D geometric skeleton; they assume the entire contour to be planar and near-perpendicular to the view direction. Cordier et. al. lift contour drawings of reflectively symmetric and symmetrically posed characters into 3D. They expect every part contour to be planar, and expect each part to be represented as a separate curve in the drawing. Karpenko and Hughes successfully process character drawings containing partial occlusions and asymmetric poses, but assume that each contour curve is planar and perpendicular to the view direction. Lacking part structure, they cannot leverage geometric priors on individual body-part shape and use surface inflation (Fig. 5) to generate the outputs. Entem et al. model animals from a canonical side-view sketch and rely on T-junctions to segment the contours into separate part outlines. They assume all contour curves to be planar and perpendicular to the view direction, and

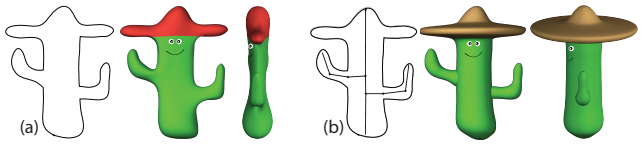


Fig. 5. (a) Lacking part information, character shape reconstruction can at best exploit overall shape smoothness, e.g [Karpenko and Hughes 2006; Nealen et al. 2007]; (b) by using a skeleton to facilitate contour partition and part modeling, we generate a more believable character shape.

only handle local T-junction type occlusions between immediately adjacent body parts. The assumptions listed above do not hold for the vast majority of articulated character drawings: these drawings frequently contain general inter-part occlusions, individual contour curves frequently extend across multiple body parts, these parts are rarely perfectly symmetric, and part contours are rarely planar (Fig. 4). By leveraging the additional information provided by the 3D skeleton, our method successfully relaxes all of these assumptions and is able to handle inputs such as 'sneaky' (Fig. 4 (bottom)) or the catwoman (Fig. 1, 3) which repeatedly violate them.

3D character construction. A range of methods reconstruct character shapes from multiple contour images. Multi-view reconstruction methods model human subjects starting from a large collection of outer contours captured from a range of views and poses [Moeslund et al. 2006]. Our approach is inspired by insights from this line of work, and especially those methods that use a skeleton to assist reconstruction [Sand et al. 2003; Ye et al. 2012], but focuses on the distinct and different problem of reconstructing a 3D shape canvas given a single descriptive character drawing that contains both occluding and outer contours and a 3D skeleton.

Finally, a number of methods [Fiore et al. 2001; Rivers et al. 2010; Jain et al. 2012; Levi and Gotsman 2013] use collections of character drawings taken from different views to create a 3D shape proxy, or to enable direct rendering from in-between views. To facilitate the process they rely on user-annotated dense correspondences, either in-between instances of the same curve drawn in different views [Fiore et al. 2001; Rivers et al. 2010] or between each curve and a corresponding bone in a matching user-positioned 2D [Jain et al. 2012] or 3D [Levi and Gotsman 2013] skeleton. These methods require at least three strategically posed drawings to achieve acceptable results. We generate reposed character renders that are qualitatively comparable to renders produced by these methods from a single, descriptive drawing and a matching skeleton with no additional annotation (Fig. 25).

3. FRAMEWORK OVERVIEW

We now describe the three key components of our canvas computation framework, and the observations that motivate them (Fig. 7).

Algorithm Input. The input to our system is a 2D vectorized cartoon drawing and a correspondingly posed 3D skeleton with no extra annotation. Like other research in articulated figure modeling [Borosán et al. 2012; Tai et al. 2004; Bærentzen et al. 2014] our approach is based on the proposition from cartoon drawing literature [Hogarth 1996; Williams 2001] that character shape is well approximated by a union of body parts represented by generalized surfaces of revolution around a skeletal structure. As the shape of a surface of revolution is driven by the choice of an axis, leveraging this observation for modeling requires a skeletal structure (Fig. 5). While curvature extrema and discontinuities in character contours hint at the underlying skeletal structure, automatic skeleton extraction [Borosán et al. 2012; Buchanan et al. 2013] may not reflect the artist-intended shape as it always aligns the skeleton with the dominant axis in elliptical regions. This is illustrated by Fig. 2 (top), where using a geometric skeleton would lead to the "snake swallowing an elephant" reconstruction on the right. This bias is confounded by ambiguous skeleton topology in the presence of occlusions (Fig. 2, bottom). Fortunately, artists can consistently and efficiently pose a 3D skeleton to match a 2D contour drawing (Section 6.1), motivating our choice of input.

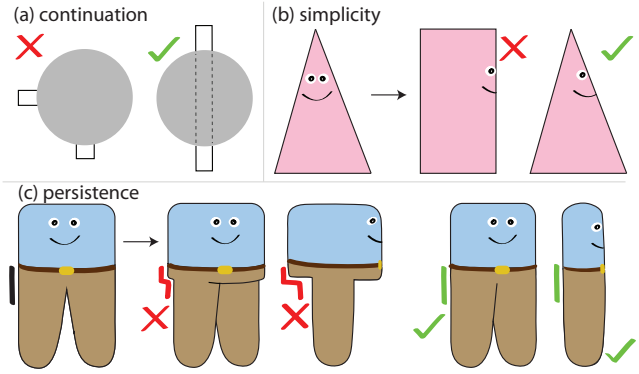


Fig. 6. (a) Perceptual studies indicate that viewers group curves that can be smoothly joined together ((a), right), seeing those as a continuation of one another; while treating those with non-smooth transition ((a), left) as distinct; viewers tend to prefer interpretations that balance part simplicity (b) against contour persistence, preferring interpretations that preserve contour shape under small view changes (c).

Skeleton-Driven Contour Segmentation. To successfully capture body parts with surfaces of revolution, we must first identify which portions of the input contour belong to the same body part (Fig. 7 (a)). Our algorithm therefore begins by segmenting the input contours into sections associated with each bone. This segmentation is guided by the principles employed in 3D skeleton-driven surface segmentation algorithms, e.g [Cornea et al. 2005; Au et al. 2008]. These methods construct surface charts whose connectivity reflects skeletal adjacencies, associating charts with proximal bones, and aligning chart boundaries with curvature extrema. We apply these principles of surface segmentation to 2D contour drawings. Since, in the presence of occlusions 2D proximity is not a reliable proxy for 3D proximity (Fig. 9), we leverage skeletal depth information to facilitate correct proximal bone-to-contour association and use Gestalt continuity [Koffka 1955] to correctly group disjoint contour segments (see Section 4, Fig. 6 (a)).

Canvas Modeling. We construct a 3D canvas from our segmentation by exploiting the perceptual cues of sketch conformity, simplicity, and contour persistence (Fig. 7 (c)). In our context, conformity requires that the contours of the created 3D canvas project onto the 2D character contours in the input drawing with reasonable accuracy, and simplicity implies a preference for maximally symmetric surface-of-revolution part geometries (Fig. 6 (b)). Maximizing symmetry when recovering 3D part geometry requires an optimal local axis of revolution. However, while the artist-posed straight-line skeletons adequately describe the character structure, they are not detailed or accurate enough to capture a geometrically centered *curve skeleton* [Cornea et al. 2005] of the target character surface. We therefore generate the desired curve skeleton by leveraging a correspondence between the straight skeleton and the seg-

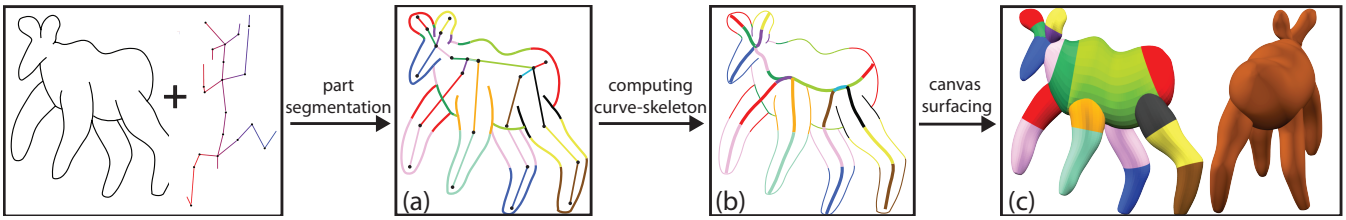


Fig. 7. Canvas construction: Given a sketch and a skeleton (shown in side view) we first segment the input contours into sections associated with skeletal bones (a), correspondences shown by matching color), correctly resolving occlusions; we use the segmentation to replace the straight-line skeleton by a curved-skeleton optimized for symmetry (b); and finally generate maximally simple body part geometries around this new skeleton while maintaining contour persistence with respect to the input drawing (c).

mented 2D contours (Fig. 7 (b)), before computing the final canvas surface. We position the curve skeleton to maximize the symmetry of body parts. (Section 5.1). Using only conformity and simplicity to compute the canvas geometry around this curve skeleton leads to plausible individual part geometries, but ignores the shape correlation between adjacent body parts outlined with a single contour. Contour persistence (Fig. 6 (c)) argues for these joint contours to retain their shape when the viewpoint changes, and especially to avoid introducing sharp discontinuities [Xu et al. 2014]. Accounting for simplicity alone can introduce such undesirable artifacts (see Fig. 6 (c)) and the accompanying video). We therefore enforce persistence across the character model by restricting the change in local profile slope with respect to its corresponding axis, allowing trajectory shape to deviate from a perfect circle to accommodate this constraint (Section 5, Fig. 3).

4. PART SEGMENTATION

Existing research on skeleton-assisted part segmentation of 3D shapes [Cornea et al. 2005; Au et al. 2008] employs a number of perception-driven segmentation criteria, variants of which apply to the segmentation of 2D contours (Fig. 8). The primary criterion is topological - in 3D each bone corresponds to a single segment, and segments are adjacent only if the corresponding bones are. The secondary criterion is bone proximity - segments are computed so as to be closest to their associated bones. Lastly, while the placement of segment boundaries is dominated by proximity to the corresponding bones, boundary locations are aligned with local curvature extrema on the surface to better match bends at skeletal joints. In describing how to apply these criteria for 2D contour segmentation we first address the simpler, occlusion-free setup, and then describe the extension to the general case.

Bisector-Based, Occlusion-Free Contour Segmentation. Absent occlusions, the contour of a drawn character is a single closed curve. In this scenario (Fig. 8) each terminal bone corresponds to a single segment and each interior bone (purple in the Figure) corresponds to two segments, one on each side. A circular "half-edge" traversal of the contour uniquely defines the connectivity between the segments (Fig. 8(b)). We can therefore generate a segmentation by appropriately positioning the boundary points between these topological segments. While we can optimize for proximity by segmenting the contours using the Voronoi diagram of the bones (Fig. 8(c)), as-is this segmentation results in a different, undesirable, segment connectivity; note in particular the green and blue segments at the bottom. However, using a subset of the diagram intersections - specifically, the first intersection between the contour and a ray emanating from each skeletal joint along its

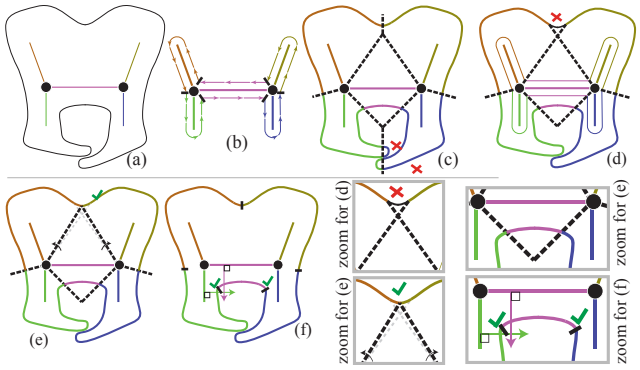


Fig. 8. Skeleton-driven segmentation of a simple contour (a) must match skeletal topology (b) and reflect bone proximity. Proximity alone does not guarantee skeleton matching segment topology (c). A more topologically consistent segmentation (d) may need to be refined by bisector rotation to avoid segment overlap (e). Boundaries are then adjusted to best align with negative curvature extrema (f).

angular bisector - to define boundaries of contour segments associated with the participating bones (Fig. 8(d)) - results in a solution largely consistent with the circular ordering. Inconsistencies show up only at locations where the contours veer far from the skeleton; at these locations bisector rays starting at adjacent joints can cross prior to intersecting the contour, resulting in ill-defined, overlapping, segments. Such interior intersections can be trivially detected and fixed by rotating the offending bisectors in opposite directions to move the intersection onto or outside the contour (Fig. 8(e)). The resulting segmentation has the desired connectivity and each segment is adjacent to its associated bone. As a last step, we adjust boundary locations to align them with bends at skeletal joints by moving them to nearby curvature extrema (Fig. 8(f)).

Contour Segmentation with Occlusions. While the algorithm above works extremely well for occlusion-free closed contours, real-world character contours contain inter-part occlusions which pose two further challenges (Fig. 9). First, in the presence of occlusions, 2D distances are not a reliable proxy for 3D distance; in Figure 9(b), for example, the contour between the pinkie and ring finger bones is closer, in 3D, to the ring finger bone despite being closer in 2D to the pinkie bone. Second, occlusions fragment the single closed contour into multiple disjoint contours, complicating the use of topological criteria for segmentation. When contours are fragmented a bone can be associated with any number of disjoint

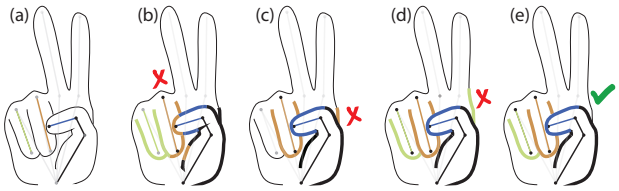


Fig. 9. A character drawing with inter-part occlusions contains multiple contour curves and the left and right outlines of a body part may now contain multiple Gestalt continuous segments (a); thus 2D proximity based segmentation is no longer adequate (b). Taking into account skeletal depth as well as 2D proximity but neglecting Gestalt continuity leads to better, but still perceptually wrong results (c,d). Our framework accounts for both considerations resulting in the desired segmentation (e).

segments; e.g. in Figure 9, the terminal bone of the partially occluded ring finger should be associated with two disjoint contour segments. Furthermore, adjacent skeletal bones may correspond to segments on different contour strokes. Nevertheless, as we discuss below, the overall bisector-based segmentation strategy remains applicable, but requires modifications that leverage the depth information provided by the input 3D skeleton to better estimate proximity, and use Gestalt continuation to analyze disjoint contours.

2D to 3D Proximity. We first note that 2D proximity is still a good proxy for 3D proximity; a bone can be associated with a farther away contour using the bisector based approach **only** if the body part associated with this bone is partially occluded and the contour in question belongs to the occluder. For typical 3D character geometry, the depth ordering between bones reflects depth ordering between their corresponding body parts, as well as their contours. Thus in general, a contour closest to a bone in 2D, should be associated with a different bone only if that bone is nearer to the viewer than the original one. While it is conceivable to create geometry and poses that violate this assumption, drawings of such shapes are inherently ambiguous even to human observers and are thus beyond the scope of this paper. Consequently for closest to the viewer bones we can still use 2D proximity as a reliable proxy for its 3D counterpart. Similarly, for bones farther away we can still continue to rely on 2D proximity as long as we ignore, or skip, contours associated with nearer to the viewer bones.

While a total depth ordering of bones may not exist, a total ordering of *mini-bones* is readily created by precisely subdividing bones that overlap in depth (*a la* the painter's algorithm) or approximately by simply subdividing all bones into mini-bones of some small maximum length (one tenth of the shortest bone in our implementation). As discussed below, the latter approach helps address the one-to-many bone to segment matching problem, as we can plausibly assume that each mini-bone has at most one visible contour segment on each side. Mini-bones resulting from subdividing a skeletal bone are seen as meeting at unarticulated valence two joints.

Topological Consistency. The bisector-based segmentation algorithm for occlusion-free inputs ensures topological consistency along the closed input contour - that is, adjacent bones are mapped to adjacent, continuous, contour segments. When occlusions are present, adjacent mini-bones can be associated with different, disjoint, contour segments (Fig. 9(c,d)) or alternatively with hidden, or imaginary, segments. Unlike the occlusion-free case, a traversal of a single input contour curve in a circular fashion does not induce a traversal of the skeleton and vice versa; at most, we can hope that,

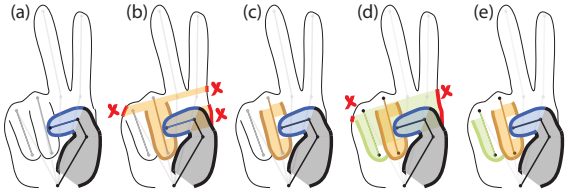


Fig. 10. Segmentation algorithm: iterating between a z-ordering based pass and consistency validation.

as we traverse along mini-bone half-edges on the skeleton using the same counter-clockwise traversal, the associated contour portions should either be continuous, or plausibly connected by an obscured contour portion. We argue that humans employ the Gestalt continuity principle to evaluate association probability in such cases, and ignore associations inconsistent with this principle (Fig. 9(c,d)).

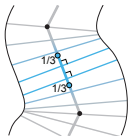
Rather than directly assigning contour segments to bones so that every assignment is Gestalt continuous, we employ a restart mechanism with a taboo list. After assigning mini-bones to contours, we evaluate all assignments of adjacent mini-bones for Gestalt continuation. When assignments are inconsistent, as is the case in Figures 9(c,d), the proximity criterion argues for keeping the correspondence for the segment closer to the bone in 2D, while disassociating the segment further away from the bone. If and when an assignment is deemed inconsistent, we restart the near-to-far processing algorithm as the disassociated segment needs to be associated with a different bone.

Final Algorithm. Our final segmentation algorithm that accounts for both proximity and topological consistency proceeds as follows (Fig. 10):

- We traverse all mini-bones in near-to-far depth order (Fig. 10(a)). The rationale for the ordering is that shallower bones, closer to the viewer, have priority over deeper bones in associating with visible contours as a consequence of the 2D to 3D proximity linkage.
- For each joint of a mini-bone we compute two joint bisector rays in 2D, one on each side of the joint as described in Section 4.0.1, and associate each ray with the first intersecting contour segment that has not yet been mapped to a shallower bone. The ray intersections (from a single joint for a terminal mini-bone, or from two joints on the same side of internal mini-bones) demarcate contour segments that are mapped to the mini-bone. Joint bisector ray intersections for deeper bones segment and associate with the closest intersecting contour segment that has not already been mapped to a shallower bone. The orange bone for example, does not associate with the tip of thumb since this tip is already mapped to the shallower blue bone in Figure 10(b).
- Once all the mini-bones for a sequence of bones connected via valence two joints have been traversed (or an individual bone if it has no valence two joints), we evaluate the contour segments associated with these mini-bones for Gestalt continuity (Section 4.0.2).
- If erroneously mapped contour segments are detected, we disassociate them from their current bones, forbid them from being associated to these bones in the future, and restart the algorithm. In Figure 10(b), once the incorrectly associated segment on the right side of the hand is found, we restart the algorithm and prohibit the ring finger from associating with that segment. In the next iteration we generate the configuration in Figure 10(c). Similarly, a new incorrect segment for the pinkie

bone is found and the algorithm is restarted (Fig. 10(d)). Finally, we finish with the correct assignment in Figure 10(e).

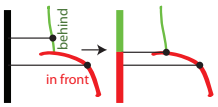
4.0.1 *Mini-joint bisector rays.* Strictly speaking the joint bisectors for internal mini-bones are simply the two opposing directions orthogonal to the bone in 2D (blue in the inset). For mini-bones close to the end-joints of an original bone, such orthogonal internal bisectors are likely to intersect the joint bisectors emanating from these end-joints before reaching the contours resulting in overlapping segments which would need to be fixed later on (Section 4.0.3).



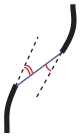
To reduce the number of subsequent fixes we preemptively rotate the internal bisectors. Specifically, we split the bone into thirds; the joint bisectors of the mini-bones in the middle third are left orthogonal to the bone, while at both ends of the bone we set the internal bisector angle to smoothly change from orthogonal to aligned with the end-joint bisector (see inset).

4.0.2 *Evaluating Segment Continuity.* For each pair of rays bounding a mini-bone, or sequence of mini-bones, we evaluate whether the mini-bone joint assignments are consistent with the Gestalt continuity principle by testing if their associated contours are perceived as a continuation of one another. We consider all the possible scenarios enumerated in Fig. 11:

– In the most common scenario where both rays intersect the same contour segment (Fig. 11(a)) this contour is clearly continuous.
– If neither ray is associated with a contour intersection (Fig. 11(b)) we similarly deem the assignments as consistent; this case suggests that the contour segment associated with the mini-bone chain between them is occluded.
– In more rare cases, the two rays intersect different contour segments immediately next to a shared T-junction (Fig. 11(c)).



This scenario is consistent with a local occlusion (see inset). To associate each mini-bone with a single contour, we move the intersection point on the occluded contour (see inset) to the T-junction. The current mini-bone is now associated only with the occluding contour.
– In the fourth scenario, the two rays may intersect different contour segments while not next to a common T-junction (Fig. 11(d)).



This is the first scenario where Gestalt continuity needs to be taken into account to decide if the assignment is topologically consistent. According to perception studies [Hess and Field 1999], more than 90% of viewers visually connect disconnected curve segments into a single contour if the angles between the segments and a straight line connecting their end-points (see inset) are less than 18°. We employ this test as-is to evaluate *Gestalt continuity* for pairs of ray-contour intersections along different contours. If the two contours are deemed discontinuous, we assume that the ray intersection, or contour assignment, that is closest to the bone in 2D is more likely to be correct, and disassociate the farther away contour segment.
– One ray intersects a contour segment and the other ray has no associated intersection (Fig. 11(e)). Here we test whether Gestalt continuity is satisfied across a sequence of mini-bones that have no associated intersections due to occlusion using the same test as above.

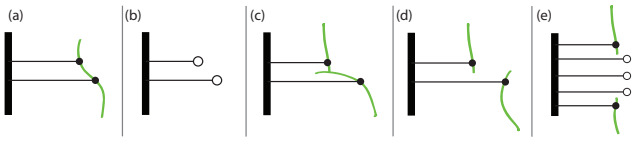


Fig. 11. Possible scenarios of contour intersections (filled circles) for rays bounding a mini-bone. Empty circle means the ray has no associated contour intersection.

4.0.3 *Rotating intersecting rays.* Similar to the occlusion free scenario, if two rays intersect prior to intersecting the same contour curve, they conceptually create overlapping segments. Thus, to preserve consistency we rotate them to flip intersection order. We apply the same rule to rays intersecting disjoint but Gestalt continuous curves, using the criterion above to determine continuity.

Once the distance and continuity driven segmentation is complete, we locally slide the boundary points on their associated curves towards local curvature extrema. Whenever a section of a contour remains unmapped, we split it between the closest adjacent mapped segments. In our experiments the resulting contour segmentations agreed with viewer intent (see Section 6.3), and we never observed an entire curve left unassigned.

5. CANVAS MODELING

A canonical surface of revolution can be computed analytically from its 3D axis of revolution and its 2D contours (Fig. 4 (top)) by first positioning the contours in 3D by leveraging rotational symmetry at all contour points, and then defining the surface by setting the radius of revolution at each point on the axis to the orthogonal distance from the axis to the 3D contours [Wong et al. 2004]. In this scenario, the part segmentation computed in Section 4 would be sufficient to precisely define a 3D canvas for contour drawings that depict canonical surfaces of revolution around corresponding bones of the input 3D skeleton. Unfortunately, character body parts are rarely perfectly symmetric. Furthermore, our input, artist-provided 3D skeletons are typically only a coarse piece-wise linear approximation of a geometrically centered exact *curve-skeleton* [Cornea et al. 2005] of the target character surface (Fig. 12).

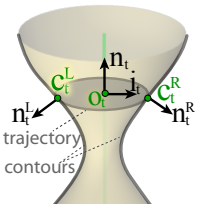
To recover a plausible canvas surface despite inexact skeleton posing and imperfect part symmetry we use a three-step process. We first compute a 3D curve-skeleton which is close to the artist defined straight-line one, but well-centered with respect to the drawn contours (Section 5.1). We then use continuity along contour curves to determine the canvas connectivity across input skeleton joints, and construct a quad dominant mesh to represent the canvas (Section 5.2). Finally, we compute the optimal 3D vertex positions across the canvas (Section 5.3), balancing rotational part symmetry with respect to the curve skeleton against contour conformity and persistence.

5.1 Computing a 3D curve-skeleton and 3D contours

We define the curve skeleton to have the same topology as its straight-line counterpart, and aim to position each branch so that it is maximally centered with respect to the contours of its corresponding body part. We initialize the curve skeleton by evenly sampling the straight-line skeleton, adding samples along the continuation of terminal bones until the point where that continuation's projection into 2D space intersects with a drawn contour, to support surface formation in these areas. Each curve skeleton vertex o_t is associated with a planar trajectory t with the vertex serving as its origin. We simultaneously compute the positions of both the curve

skeleton vertices and the right and left contour points on their trajectories, balancing contour symmetry with respect to the 3D curve-skeleton, similarity between the curve- and straight-line skeletons, and 3D contour smoothness subject to input conformity (Fig. 12).

Symmetry. In our computation we seek three-fold symmetry. First, we aim for left and right contour curves to be maximally mirror symmetric around the curve skeleton. Given a planar trajectory with center o_t and normal n_t that intersects the 3D contours at points c_t^L and c_t^R, mirror symmetry is satisfied if the contour points are symmetric around the plane (with plane normal i_t) containing the axis of revolution ((o_t, n_t) and the view direction (z-axis) as shown in the inset. We also seek local front-back symmetry at each contour point expressed as an expectation for the surface normal along the contour to be inside the plane spanned by the local axis of revolution and the contour point. Finally, to optimize the rotational symmetry of the surface profiles connecting adjacent trajectories we expect the lines connecting adjacent trajectory origins to be aligned with their respective normals. The combined symmetric energy is formulated as,



$$E_s = \sum_t \|(c_t^L - o_t)\cdot i_t + (c_t^R - o_t)\cdot i_t\|^2 + \left(n_t^L \cdot (n_t \times (c_t^L - o_t))\right)^2 + \left(n_t^R \cdot (n_t \times (c_t^R - o_t))\right)^2 + \sum_{(t',t)} \|(o_{t'} - o_t) \times (n_t + n_{t'})/2\|^2 \quad (1)$$

where $i_t = n_t \times (0,0,1)$, and the last term's summation index (t', t) represents all adjacent pairs of trajectories. The first term expresses the mirror symmetry between contours; the next two express the local front/back symmetry at each contour; and the last term encodes origin alignment. Since this term is direction invariant, we explicitly constrain the lines connecting pairs of adjacent trajectory origins to have the same orientation as the normals $(o_{t'} - o_t)\cdot(n_t + n_{t'}) > 0$ with consistently oriented n_t and $n_{t'}$. Lastly to ensure trajectory planarity we enforce

$$(c_t^L - o_t)\cdot n_t = (c_t^R - o_t)\cdot n_t = 0. \quad (2)$$

Skeleton similarity. Since we expect the artist skeleton to approximate the target curve skeleton shape, we minimize the distance between the joints j_c and j_l on the two skeletons,

$$E_c = \sum_j \|j_c - j_l\|^2. \quad (3)$$

Contour depth. Finally, we minimize depth change along contours,

$$E_z = \sum_{(c_t^s, c_{t'}^s)} (c_t^s.z - c_{t'}^s.z)^2 \quad (4)$$

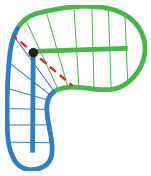
where c_t^s and $c_{t'}^s, s \in \{L, R\}$ are consecutive points on the same contour curve. This term is most important at joints, where it communicates depth information between adjacent body parts.

In the combined energy functional, symmetry and skeleton similarity are assigned unit weights, while contour depth is assigned a smaller weight of 0.1:

$$E = E_s + E_c + 0.1E_z. \quad (5)$$

Fig. 12. Curve skeleton computation: (a) user posed straight-line skeleton with the initial trajectory centers and their corresponding trajectory contour points marked; (b,c) front and side views of curve skeleton and 3D contours; (d) final surface with contours highlighted.

Trajectory Normal Computation. Simply including the trajectory normals n_t as unknowns in Equation 5 results in a highly nonlinear formula that is challenging to optimize efficiently. We therefore reduce the energy complexity to a simple quadratic formulation by independently pre-computing these normals. In general we expect trajectory normals to be close to the directions of the straight-skeleton bones that the trajectories are originally associated with. On a curved skeleton, however, we expect these directions to change smoothly at valence 2 joints. We use the segmentation to determine the best transition angle by considering whether the joint has visible segment boundaries associated with it (see inset). If so, we rotate the axis at the curve-skeleton vertices closest to the joint so that the plane will intersect the contour just next to the boundary point. If both boundaries are visible we use an average rotation to best fit both, while if no boundary is visible we rotate the axis to the relevant joint bisector. We then smoothly propagate the rotation along the bones. Note that those rotations may differ from the joint bisector, shown as a red dashed line in the inset.



Contour-Skeleton Matching. To account for input contour shape, we need to match the curve-skeleton vertices *of each* bone with densely sampled points on the input contours that we previously associated with this bone during our segmentation process. Incorporating the search for best skeleton/contour correspondences into the curve-skeleton computation is both challenging and time consuming. We therefore pre-compute the matches by leveraging the expectation that contour points on each trajectory are mirror symmetric around the local trajectory axis. This expectation implies that the line connecting such pairs of points should be orthogonal to the 2D projection of the local axis. To compute the correspondences for each initial curve-skeleton vertex, we shoot rays left and right orthogonally to local trajectory axis n_t to locate pairs of intersections on contours belonging to opposite sides of the body part. Note that in the presence of occlusions we may locate only one such intersection, or no intersections at all. These intersections are used as the image space locations of the corresponding contour points and are fixed throughout the optimization process.

We consequently solve for the 3D positions of the curve-skeleton vertices and the depth of their associated contour points using a quadratic solver that minimizes the combined energy function subject to the equality and inequality constraints above. We then compute the radii r_t of each trajectory as the average distance from its origin to its two contour points and use those in the subsequent canvas mesh computation step.

5.2 Canvas Connectivity

We represent the canvas using a set of planar, closed vertex cycles, or *trajectories* circling the skeleton, connected by a quad-dominant mesh. (Fig. 13). We place cycles around each trajectory center computed in the previous curve-skeleton computation stage; all cycles

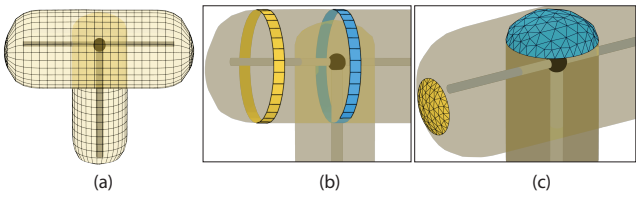


Fig. 13. Canvas connectivity (a) with close-ups of quad strips between trajectories (b) and triangulated terminal trajectories (c).

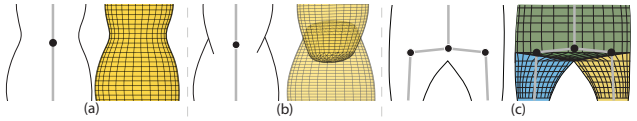


Fig. 14. Connectivity across joints: (a) visually continuous parts; (b) Discontinuous parts; (c) the top part is deemed continuous with both lower ones, while the two bottom parts are deemed discontinuous since their shared contour curve has a cusp between them.

have the same number of vertices and a consistent circular indexing facilitating explicit angular and axial parameterization of the parts. We then form quad strips between pairs of adjacent cycles along each skeleton bone placing edges between vertices with same angular index on both (yellow strip in Fig. 13(b)) and triangulate the last, terminal, cycles at each terminal joint (yellow, Fig. 13(c)). The connectivity choices at interior joints are determined based on the interaction between the drawn outlines of the participating parts. Specifically, for each pair of parts adjacent to a joint we determine if the parts are a continuation of one another or not. If two body parts are deemed continuous we fuse their canvas surfaces, placing a quad strip between the part trajectories immediately adjacent to one another across the shared joint (blue strip in Fig. 13(b)). If a part has no continuous neighbors across an interior joint, its last cycle at the joint is simply triangulated (blue, Fig. 13 (c)).

Two parts are deemed continuous if their outlines are either adjacent to one another along a single smooth contour curve or are Gestalt continuous (Fig. 14). We deem a contour curve smooth if it has no cusp at the boundary between the two outlines. This smoothness requirement is motivated by the observation that artists frequently omit drawing small T-junctions, connecting what in 3D should be separate contours into a single, albeit non-smooth one (Fig. 14(c)). Our joint processing can, by design, lead to non-manifold, as well as self intersecting canvases. If desired, the surfacing step (Section 5.3), which leverages our current canvas connectivity, can be followed by a more complex fusion process similar to [Borosán et al. 2012; Bærentzen et al. 2014] resulting in a smooth manifold mesh. However, we found this step unnecessary for the canvas applications shown in this paper.

5.3 Canvas Surfacing

The key step in computing the canvas shape is to position trajectory vertices balancing the goals of maximally symmetric body parts, contour conformity, and persistence. The remaining vertices, those in the triangulated regions next to terminal trajectories, are computed in a post-process which seeks for smooth canvas geometry overall.

We constrain each trajectory t with vertices $v_0^t, \ldots, v_n^t = v_0^t$ to be orthogonal to the previously computed normal n_t,

$$(v_i - v_{i-1}) \cdot n_t = 0 \quad i = 1, \ldots, n.$$

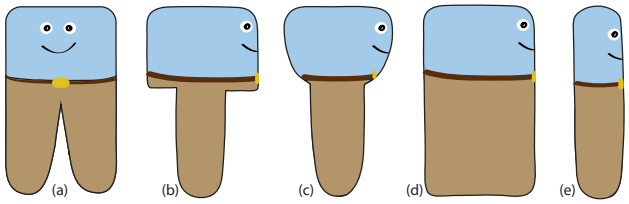


Fig. 15. Given the input sketch (a), contour persistence indicates that side view contours (b,c) significantly differing from front-view ones are undesirable. Viewers similarly do not anticipate extreme foreshortening (d). Our result (e) is persistent with the front view contours.

Part Symmetry. To maximize part symmetry we seek canvas trajectories which are as circular as possible and aim for profiles connecting consecutive trajectories along each bone to have as constant as possible angle of revolution, or slope, with respect to each trajectory's axis. We cast circularity as a quadratic energy term,

$$E_c(t) = \sum_i (v_i^t - (v_{i-1}^t + v_{i+1}^t)/2 - \delta_i^t)^2 \qquad (6)$$

where the vectors δ_i^t are the Laplacian coordinates of the i'th vertex in a planar circle whose normal and radius are the pre-computed n_t and r_t. To account for different axes of revolution assigned to different trajectories, we express profile symmetry for each trajectory t and a neighboring trajectory t' as

$$E_p(t,t') = \sum_i (v_i^t - M_{t',t} v_i^{t'} - R^n(v_{i+1}^t - M_{t',t} v_{i+1}^{t'}))^2, \quad (7)$$

where R^n is a rotation matrix of π/n around the axis (o^t, n^t), and $M_{t',t}$ is the shortest path coordinate transformation aligning the axis $(o^{t'}, n^{t'})$ with (o^t, n^t).

Conformity. We want the visible contours of the canvas to match the artist drawn ones. To achieve this, the contour vertices on the final trajectories, i.e. those whose normals are in the view plane, must coincide in 2D with the previously computed trajectory contour points c_t. While we do not know the final trajectory shape, we assume that this shape will remain close to the ideal circular one; we therefore select the left and right vertices whose normals on these ideal trajectories are most orthogonal to the view direction as the potential contour vertices. For each such vertex v_t, if a matching (left or right) trajectory contour point c_t exists we force their 2D locations to coincide,

$$v^t.x = c^t.x \text{ and } v^t.y = c^t.y.$$

Persistence. Previous work has relied upon part symmetry and contour conformity alone when attempting to recover 3D models from character drawings. This produces intuitive individual part geometries, and plausible transitions between both discontinuous parts and those deemed continuous along both side contours; however, it also generates sharp depth discontinuities, contradicting viewer perception, between parts classified as continuous along only one side contour, such as a leg and a torso (Fig. 15(b)). The reason for such discontinuities is that in these situations the trajectories adjacent across the joints typically have vastly different radii and far apart centers. Since the artist contours provide no hint of discontinuity, we believe that viewers mentally eliminate them by deforming the parts to bring them closer together. Moreover, we speculate that viewers expect the character contours to maintain their overall drawn shape in alternate views up to inevitable

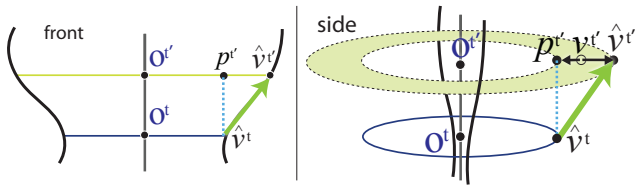


Fig. 16. We constrain the profile angle to the range between the ideal profile slope given by the two ring radii and the axis direction.

foreshortening, avoiding the behavior visualized in Figure 15(c). This observation is supported by the minimal-variation principle observed by [Xu et al. 2014]. Following these observations we incorporate persistence into our setup as follows. When two parts are continuous along only one side contour we explicitly minimize the depth variation along quad-strips connecting these parts,

$$E_d(t,t') = \sum_i (v_i^t.z - v_{j(i)}^{t'}.z)^2 \qquad (8)$$

where t is the trajectory with the smaller radius and $v_{j(i)}^{t'}$ is the closest vertex to v_i^t in image space, on the larger trajectory. We use vertex positions on perfect circular trajectories with centers o_t and $o_{t'}$ and radii r_t and $r_{t'}$ to compute these distances. Note that both values $v_i^t.z$ and $v_{j(i)}^{t'}.z$ are free, but the correspondences between their vertices $j(i)$ are fixed throughout the optimization.

To avoid creating discontinuities elsewhere, when two parts are continuous along both side contours, we minimize profile variation along the quad-strip joining them using Equation 7. This formulation leverages the slope along the two contours to optimize for depth variation consistent with viewer perception.

Lastly, to avoid undesirable derivative discontinuities (Fig. 15(c)) anywhere across the canvas surface we explicitly constrain the profile angle with respect to each axis of revolution to the range between the ideal profile slope given by the two ring radii and the axis direction (Fig. 16),

$$(v_i^{t'} - \hat{v}_i^{t'}) \cdot (\hat{v}_i^{t'} - \hat{v}_i^t) \le 0$$
$$(v_i^{t'} - p_i^{t'}) \cdot (\hat{v}_i^{t'} - \hat{v}_i^t) \ge 0$$

Here $\hat{v}_i^t$ are the positions of the corresponding cycle vertices v_i^t on an ideal circular trajectory, and $p_i^{t'} = \hat{v}_i^t + o^{t'} - o^t$. In the Figure 16, for the trajectory t' with an adjacent trajectory t, those two inequalities constrain vertex positions along t' to lie within the green ring whose boundaries are derived from the contour slopes between the pair of trajectories t and t'.

Given the terms above we proceed to optimize symmetry and persistence at joints subject to the trajectory planarity, conformity and profile slope constraints listed above:

$$E = \sum_t w(r_t) E_c(t) + \sum_{(t,t')\in B\setminus J} E_p(t,t') + \sum_{(t,t')\in J} E_d(t,t'), \quad (9)$$

Here B is the set of pairs of canvas trajectories connected by a quad strip and J is the subset of such pairs with only one-sided contour continuity across joints. To promote the preservation of smaller trajectories, where even a small absolute error introduces large deviation from the ideal circular shape (Fig. 15(d)) we introduce per-trajectory weights $w(r_t) = 25e^{-(r_t/2\sigma)^2}$ with σ set to one third of the average trajectory radius. All other terms in the functional are assigned unit weights. To avoid depth ambiguity, we fix the z coordinate of one vertex. We use a quadratic programming package [Gurobi Optimization 2013] to obtain the desired minimizer.

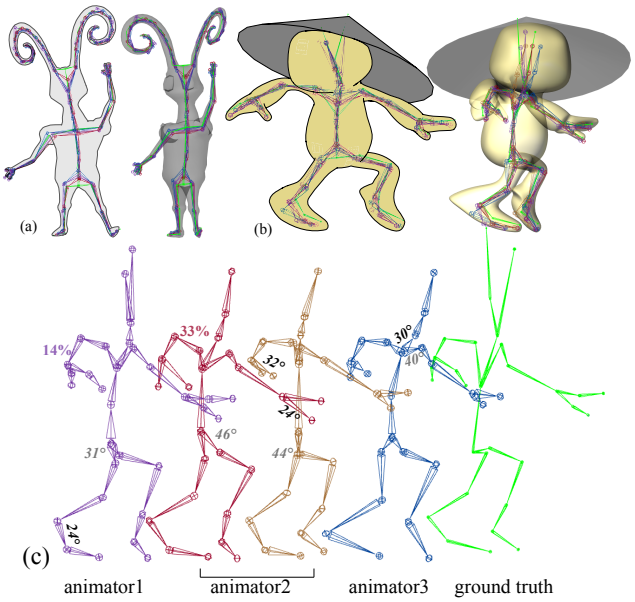


Fig. 17. Ground truth (green) and 3D skeletons created by 3 animators overlaid on two ground truth 2D character drawings (a), (b), also shown from an alternate view overlaid on the ground truth 3D canvas. The skeletons in (b) shown individually (c). The purple and maroon skeletons, created by manipulating an overlaid 2D skeleton have differences in 3D limb length between symmetric limbs. The maximum difference for each skeleton, 14% and 33%, is marked on the longer limb. The brown skeleton was created by animator #2 mimicking the workflow of animator #3. The angular deviation between the corresponding bones on the ground truth and artist skeletons is dominated by control bones (hips and shoulders) which have no impact on the result geometry. The maximal deviations without (and with) control bones are: 24° (31°) for the purple skeleton, 24° (46°) maroon, 32° (44°) brown , and 30° (40°) blue. Average angle differences are 13°, 15°, 15°, and 18° respectively.

The resulting canvas is smoothed using standard Laplacian smoothing, while weakly holding the positions of contour vertices to eliminate local artifacts that can emerge due to imperfections in the input contours and small surface discontinuities due to the use of range constraints. To position the vertices in the triangulated regions next to terminal bone tips we use a simple Laplacian formulation that enforces tangent continuity with the rest of the surface.

6. PERCEPTUAL AND DESIGN VALIDATION

We perform three-fold validation of the key aspects of our algorithm: we evaluate artist ability to provide the desired inputs, compare our results to ground truth and artist drawings, and validate our segmentation algorithm via a user study.

6.1 Creating Overlaid 3D Skeletons

Current animation practice uses 2D character drawings, such as those used as inputs to our system (e.g. Fig. 1), as a visual reference to manually author a 3D character model in a symmetric canonical pose [Maraffi 2003]. A 3D skeletal structure is then interactively created and positioned within this 3D model. Our workflow expects

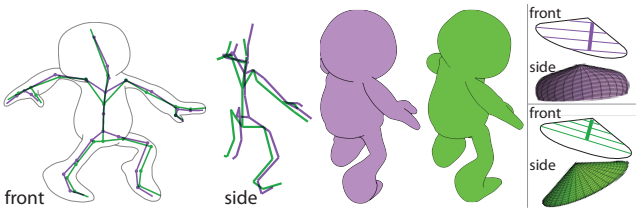


Fig. 19. Left: Given the same input sketch, small variations in skeleton posing (green and purple Figure in 17) lead to minor changes in character shape. Right: significant change in bone slope and location for a symmetric contour leads to larger shape difference.

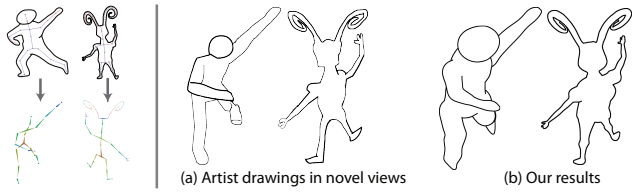


Fig. 20. Comparison of our results (b) to sketches produced by artists (a) for the same view and pose.

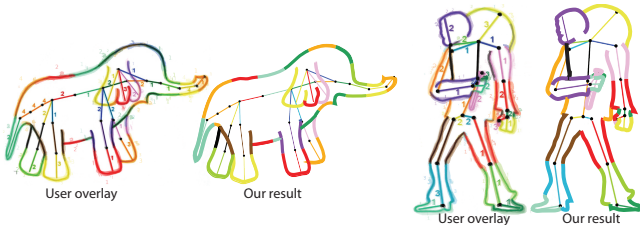


Fig. 21. Overlaid user segmentations (left) for both the elephant and the scientist are qualitatively similar to the algorithmic results (right).

animators to effectively create a 3D skeleton without an explicit 3D model, and pose it directly over a 2D character drawing.

To ensure the viability of our workflow, we asked three Maya animators to create 3D skeletons over two ground truth drawings (Fig. 17). Two animators (purple and maroon in Fig. 17) first created a 2D skeleton overlaid on the drawing and then re-positioned joints in an alternate view to get a desired 3D pose. One (purple) further used a measurement tool to compare symmetric parts and then further moved joints in 3D in an attempt to equalize the lengths of symmetric parts. These skeletons show a discrepancy in the average 3D length of symmetric parts (8% avg., 14% max. for purple and 19% avg., 33% max. for maroon) in Fig. 17, c.

The third animator (blue) first used the drawing simply as a visual reference, to create a symmetric, canonical skeleton and roughly pose it in 3D. This 3D skeleton was then moved onto the drawing and the pose refined by rotations and symmetric scaling of parts, to satisfactorily overlay the skeleton in 2D on the drawing. We described this workflow to animator #2 (maroon), who concurred that despite the natural tendency to first oversketch a 2D skeleton on the drawing, a canonical 3D skeleton allowed animators better control over symmetry and part proportion. The brown skeleton in Figure 17(c) was easily created by animator #2 using this workflow.

All animators took between 5-10 minutes to create and pose these qualitatively similar skeletons in 3D. The above exploration gives us confidence that animators imagine the 3D pose of 2D character drawings consistently and with practice can capture this pose with a 3D skeleton, overlaid directly on a 2D drawing.

Robustness to Input Variation. We also examined the impact of using different artist skeletons on the canvases created by our system (Fig. 19). As demonstrated, while the character pose predictably changes with changes in bone posing, the body part shape remains largely fixed, thanks to our robust curved-skeleton computation stage. The invariance to minor posing changes is important, since artists are unlikely to pose a skeleton perfectly. The shape change is most pronounced (Fig. 19 (right)) when a bone for a perfectly symmetric surface of revolution is significantly misaligned compared to the expected axis. Such misplacement is easy to spot and fix. Overall, as long as the depth ordering of the bones is correct, the intrinsic geometry of our results changes only marginally with changes in 3D skeleton posing. In particular angle and depth changes (in this example we have bones orientations vary by up to 30°) cause only small difference in the results. The output is more dependent on the image space skeleton positioning, and in particular on how off-center the skeleton is with respect to the drawing. Artists can easily center skeletons in 2D.

6.2 Comparison to ground truth and artist drawings

To validate our method, we compare the canvases created by our algorithm to ground-truth canvases for given contours and to alternate view drawings created by an artist given the same input as our method. To perform the ground truth comparison, we had an artist create and rig four 3D models (Fig. 18). We then used contour only renders of these models and the artist skeletons as input to our method. The resulting canvases are extremely similar to the original models, validating our design choices.

We further validated the perceptual correctness of our framework by comparing these results to artist generated drawings of the input sketched characters in alternate views (Fig. 20). We provided another artist with our input drawings with the skeleton overlaid, a 2D view of the skeleton in the desired output pose, and a 3D posed skeleton which allowed the artist to better relate the two poses. We then asked him to redraw the input character matching the skeleton pose. The results were qualitatively very similar, though the artist's characters tended to be leaner than our interpretation.

6.3 Perceived Contour Segmentation

To evaluate consistency across viewers and compare our algorithm with viewer perception, we asked 12 viewers to hand segment the contours on four simple and five complex drawings and associate each segment with bones of an overlaid 2D skeleton. We chose to distinguish joints by color as numbered labels for skeletons with dozens of bones were visually confusing. None of the users remarked that color based segment association was problematic as a task. Fig. 21 summarizes the resulting segmentations on two complex inputs, with various user segmentations overlaid to visualize correlations across viewers. The full set of user segmentations is included in the supplementary material. While viewers had less information than our algorithm (a 2D rather than 3D skeleton), their segmentations are largely consistent and match well our algorithmic segmentation. We thus believe that our 3D character canvas is built on a robust and perceptually meaningful contour segmentation algorithm.

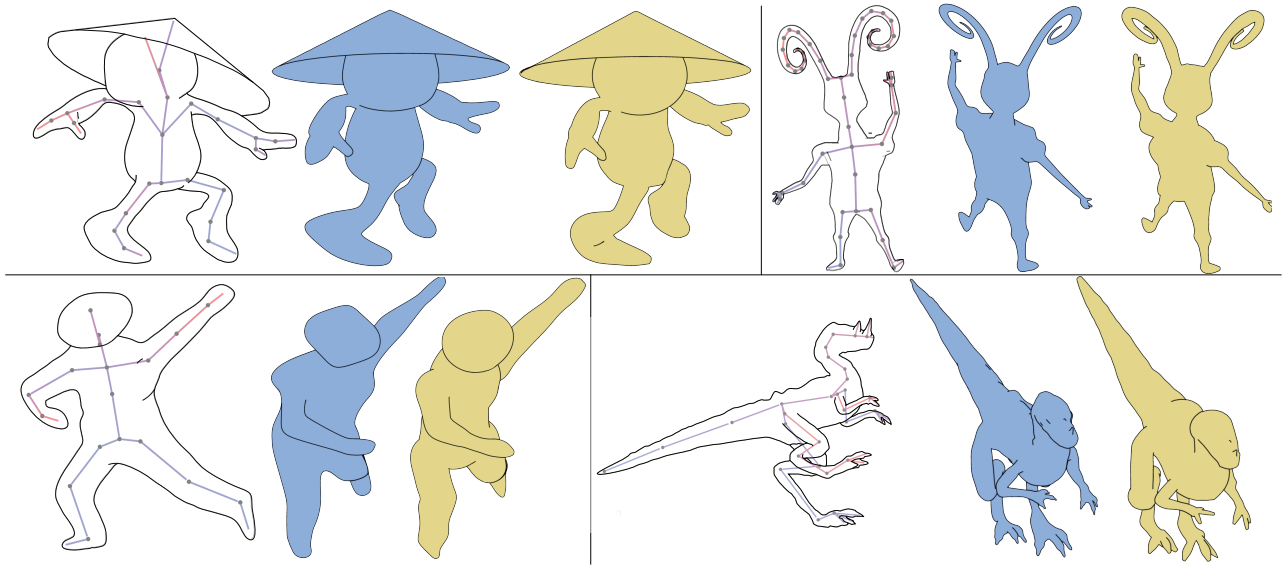


Fig. 18. Comparing our results to ground truth data: Left to right: contours and skeletons of ground truth (GT) models; GT (blue) and our (yellow) models rendered from alternate views.

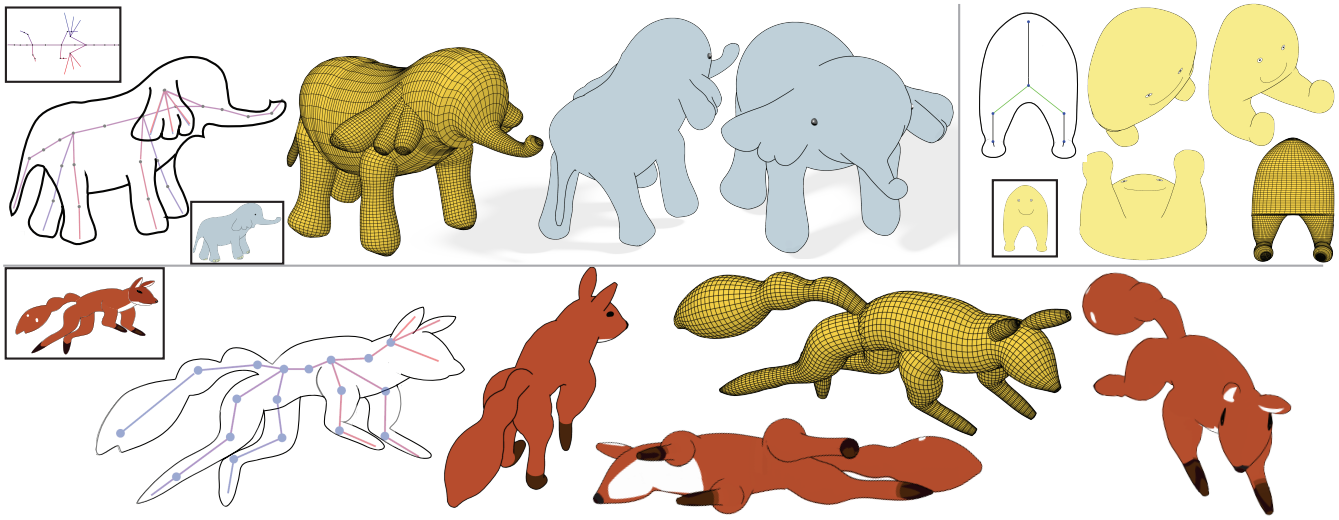


Fig. 22. Canvases and alternate view renders generated using our system from the inputs on the right.

7. RESULTS

We demonstrate the results of our character canvas modeling framework throughout this paper. We show both the actual canvases created by the method (Fig. 3, 22), as well as a range of NPR renders created using these canvases from different view directions in both the input and alternate poses (see Fig. 1, 23). The rendering examples include significant changes in contour topology compared to the input view, which cannot be achieved purely by 2D control (e.g. see back view of the catwoman Fig. 1). Using our canvases, with their built-in cylindrical parameterization, one can easily apply advanced rendering techniques such as fur or feathers simulation (Fig. 24), enabling artists to generate 3D effects without resorting to complex 3D modeling tools.

One of the main technical challenges, addressed by our method, and showcased by these examples is correct resolution of inter-part occlusions. Not only does it enable artists to draw characters in natural rather than artificial canonical poses, but it enables them to draw characters whose anatomy does not allow for such occlusion-free pose, e.g. one simply cannot draw a quadropus (Fig. 23) with both the head and all four legs fully visible. Other such examples include the fox and anteater (Fig. 22, 24).

Workflow. The inputs we evaluated our framework on were created using two workflows motivated by different target applica-

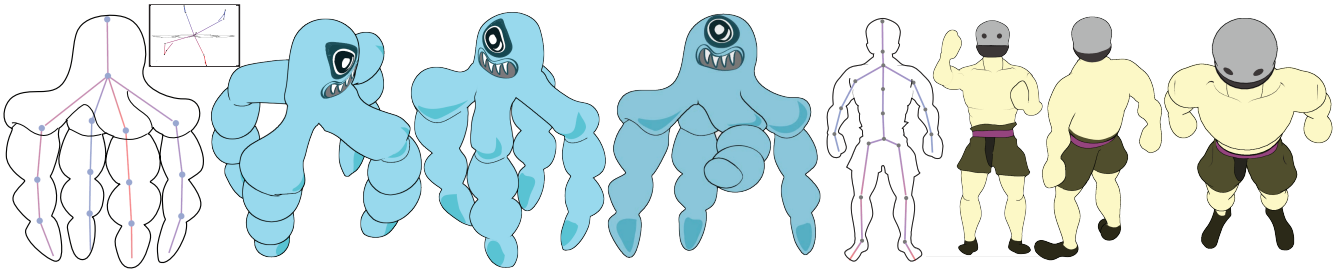


Fig. 23. A variety of re-posed renders generated automatically from the inputs on the right.

Fig. 24. The explicit cylindrical parameterization of our canvases allows for a range of advanced texturing effects, hard to achieve without a an underlying 3D structure.

tions. In the first one, an artist created a set of sketches, e.g. catwoman or elephant and then fitted a skeleton to those using Maya or other animation software (see Section 6.1). This framework is best suited for creating new cartoon art and bringing to life legacy characters, where a drawing of the character already exists.

In the second workflow, artists created and pose a 3D skeleton first, and use it as an armature over which to draw character contours from an interesting viewpoint (fox, anteater, quadropus). This approach is particularly useful in animation setups where artists already have a skeletal animation sequence they want to adapt to a new character. The accompanying video shows several animation sequences, each generated from a single frame, created using this workflow. The amount of work required to generate these animations was drastically lower than using the traditional 2D animation workflow, where key-frames describing out-of-plane motion are typically drawn by hand.

Global Symmetry. Besides local symmetry which is used throughout the algorithm, characters frequently exhibit left-right global symmetries, which viewers use to complete occluded body part geometry. We employ this principle in two examples to recover fully occluded geometry (elephant) or correct for inaccurate artist drawing (fox) by enforcing similar trajectory shape for matching trajectories on symmetric joints.

Impact of Design Choices. Figure 15 demonstrates the importance of our design choices when surfacing the canvas. Not accounting for persistence at joints (Fig. 15(b)) results in unexpected surface discontinuities. Locally minimizing depth variation (Fig. 15(c)) is similarly insufficient. Our framework (Fig. 15(e)) which constrains profile slope and minimizes foreshortening produces more natural results.

Parameters and Runtimes. Our method has no tunable parameters. For canvas modeling we use thirty vertices per trajectory and have uniform trajectory density across all bones; the density is determined so as to have at least ten trajectories along the shortest bone, and to have the distance between consecutive trajectories be no more than one percent of the character's bounding box diagonal. Our software takes between ten to sixty seconds to generate a canvas on an Intel Core i7 machine with 32GB of RAM. Roughly 25% of this time is spent in the segmentation stage and the rest is spent by the QP solver computing the canvas surface. This fast turnaround allows artists to quickly repose the skeleton or update the drawing were they to find the results inadequate.

Comparison to Prior Art. Figure 25 highlights our ability to generate models of equal complexity to those generated by multi-view approaches such as [Rivers et al. 2010; Levi and Gotsman 2013], without the need for multiple corresponding drawings. We performed this comparison by using one of the input views utilized by these prior systems, tracing 2D curves over it as our sketch input and posing corresponding skeletons. Our method employs significantly less user input than Levy et al, who require at least three

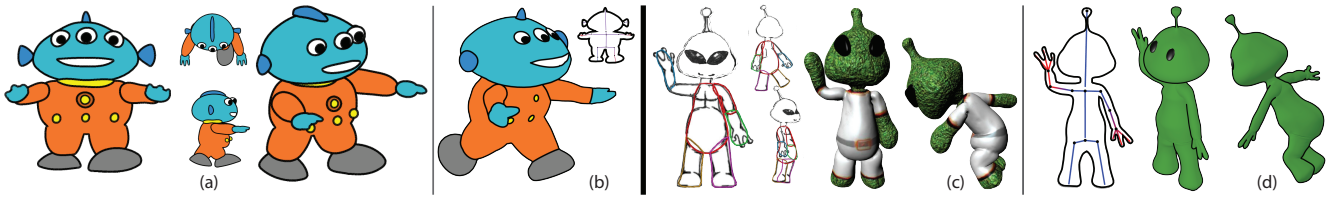


Fig. 25. Given a single drawing and a posed skeleton we generate qualitatively similar results (b,d) to those created by multi-drawing systems which employ manually specified curve correspondences between drawn curves: [Rivers et al. 2010] (a) and [Levi and Gotsman 2013] (c).

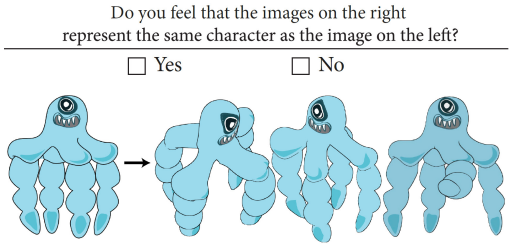


Fig. 26. An example of the qualitative evaluation questionnaire.

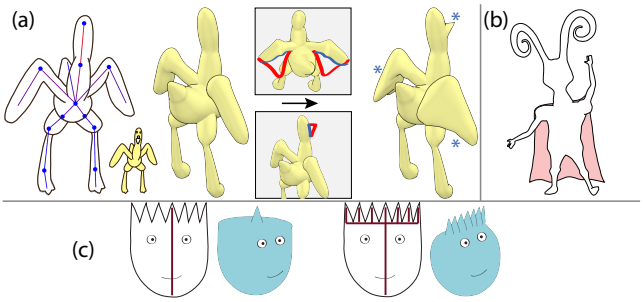


Fig. 27. Our ability to plausibly recover character shape is limited by the descriptive power of the inputs. Absent cues to the contrary we generate round bird wings, instead of anatomically correct ones (a). Since we use a standard mesh representation, the canvas can be easily edited to correct the wings or add extra features (beak) using standard tools (a, right). Geometries not-well represented by generalized surfaces of revolution, such as loose clothing (b, pink cape) must be modeled by other means. While some fine details can be captured by using skeleton refinement (c), alternate editing tools are likely to achieve this goal faster.

corresponding drawings each with an appropriately posed skeleton. While Rivers et al. do not require a skeleton, they still expect at least three drawings with correspondences and cannot articulate the results.

We successfully handle a much wider range of sketches than previous methods, most of which, e.g. [Buchanan et al. 2013] can handle only occlusion free inputs. While Cordier et al. [2011] support partial occlusions, they assume perfect rigid mirror symmetry, and expect every part silhouette to be drawn as a separate curve. Karpenko and Hughes [2006] make a similar curve planarity assumption. Our framework successfully handles complex occlusions, including scenarios deemed ambiguous by previous methods (e.g. elephant in Fig. 22, see [Karpenko and Hughes 2006]); does not require posing symmetry (e.g. see the mad scientist) nor separate part outlines (e.g. see hind side of the fox), and plausibly recovers non-planar contours (see Fig. 4). As demonstrated by Figure 5, our shape computation which aims to maximize simplicity, generates results more consistent with user expectation than inflation based frameworks such as [Karpenko and Hughes 2006; Nealen et al. 2007]. By accounting for persistence (Fig. 15) our method avoids depth discontinuities at complex joints bound to show up when parts are assumed to have perfect rotational symmetry [Buchanan et al. 2013; Cordier et al. 2011].

Qualitative Evaluation. We asked six computer artists to provide visual critique of our outputs (catwoman, elephant, quadropus) by showing them the input drawings and the output renders (see Figure 26 and supplementary material), and asking them if our results represent the same character as the input drawing. All six agreed that our results faithfully capture the original input in new poses and views and expressed strong interest in using our system in their work.

Limitations and Improvements. Like human observers, our method's ability to predict the shape of a character is inherently limited by the descriptive power of the input drawings, and our algorithm can be misled by badly posed or obfuscated drawings. For example, faced with an oblique view of a bird's wings, neither viewers nor our method can guess their depth without resorting to prior knowledge of bird anatomy (Fig. 27(a)). Since selecting a single view where all character body parts are well described can sometimes be challenging, we provide users with an incremental, overdraw interface. In this interface, users can first generate a character model from a single view, and then update the canvas from another view using contour overdrawing framework that follows [Nealen et al. 2005] (Fig. 27(a)).

While our method is robust against minor inaccuracies in the input skeleton, major errors in skeleton depth placement may clearly cause undesirable artifacts such as intersections between body parts. We did not encounter such situations on the tested inputs. We believe that the simpler solution would be for the artist to adjust the skeleton, if and when they find the result unsatisfactory, and rerun the algorithm. However if desired, one can incorporate additional non-intersection constraints into the optimization in Equation 9, or fix the self-intersections as a post-process step once the canvas is generated. Regardless, we are still dependent on the ability of the artist to pose a skeleton with respect to a cartoon drawing in a manner that avoids intersection between body parts.

A fundamental premise of our work is that the 3D canvas is a collection of generalized surfaces of revolution parts, each part being defined by a bone of the input 3D skeleton. Surface detail for a 3D canvas that strongly deviates from this premise, like cloth folds with internal occluding contours (Fig. 27(b)) are thus not captured by our approach. While the hair spikes of (Fig. 27(c)) can be constructed using surface of revolution parts, it is unlikely that artists would provide the necessary definition for each hair spike with a bone on the input 3D skeleton. Thus while our system is well suited for canvas creation, artists should combine it with other

mesh-editing tools to generate detailed, dressed, characters. Some cartoon characters may have elements which are designed to consistently face towards the camera regardless of the viewer position (cartoon eyes, or the ears of a cartoon mouse); we do not attempt to recover these features from the input sketch. In a production environment such features are best implemented using billboard vector elements. In general, realistic cartoon drawings combine a mix of strokes that define a 3D canvas, view-dependent 3D geometry, and 3D detail drawn on and around the surface of the 3D canvas [Schmid et al. 2011]. We have focused on simplified cartoon drawings where the strokes strictly comprise a character canvas. The classification of strokes of arbitrary cartoon drawings as described, and their 3D reconstruction, is subject to future work.

8. CONCLUSIONS

We presented the first, to our knowledge, system for 3D character canvas modeling from a single naturally-posed character drawing and overlaid 3D skeleton. We can process input with complex inter-part occlusions and large variations in contour depth. As demonstrated, our output 3D geometry is appropriate for use as an animation canvas: facilitating non-trivial reposing and large viewpoint changes of complex characters, that remain consistent with the input drawing, and enabling non-photorealistic animation using painterly strokes on and around the canvas.

Our work is aligned with a recent trend to simultaneously model 3D character geometry and its corresponding skeleton [Borosán et al. 2012; Bærentzen et al. 2014]. While we have focused on 3D proxy geometry creation from minimal input in the form of drawn contours, our coupling of 3D skeleton and input drawings using a perceptual framework is extensible. In the future we expect that our algorithmic approach, adapted to richer input drawings, embellished with internal contours, construction lines and shading, will result in fully detailed and complex 3D character models.

REFERENCES

ADOBE FLASH. 2013. Professional. Adobe, Inc.

ANIME STUDIO. 2013. Smith Micro Software.

AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. *ACM Trans. Graph. 27,* 3, 44:1–44:10.

BÆRENTZEN, J. A., ABDRASHITOV, R., AND SINGH, K. 2014. Interactive shape modeling using a skeleton-mesh co-representation. *ACM Trans. Graph. 33,* 4 (July), 132:1–132:10.

BASSETT, K., BARAN, I., SCHMID, J., GROSS, M., AND SUMNER, R. W. 2013. Authoring and animating painterly characters. *ACM Trans. Graph. 32,* 5 (Oct.), 156:1–156:12.

BLOOMENTHAL, J. AND WYVILL, B., Eds. 1997. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

BOROSÁN, P., JIN, M., DECARLO, D., GINGOLD, Y., AND NEALEN, A. 2012. Rigmesh: Automatic rigging for part-based shape modeling and deformation. *ACM Trans. Graph. 31,* 6, 198:1–198:9.

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, T., ZHU, Z., SHAMIR, A., HU, S.-M., AND COHEN-OR, D. 2013. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 32,* 6.

CHERLIN, J. J., SAMAVATI, F., SOUSA, M. C., AND JORGE, J. A. 2005. Sketch-based modeling with few strokes. *Proceedings of the 21st spring conference on Computer graphics - SCCG '05 1,* 212, 137.

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

CORNEA, N. D., SILVER, D., YUAN, X., AND BALASUBRAMANIAN, R. 2005. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer 21,* 11, 945–955.

ENTEM, E., BARTHE, L., CANI, M.-P., CORDIER, F., AND VAN DE PANNE, M. 2014. Modeling 3D animals from a side-view sketch. *Computer and Graphics* 38.

FIORE, F. D., SCHAEKEN, P., AND ELENS, K. 2001. Automatic inbetweening in computer assisted animation by exploiting 2.5 D modelling techniques. *Proc. Conference on Computer Animation*, 192–200.

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

GUROBI OPTIMIZATION. 2013. http://www.gurobi.com/.

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

HOFFMAN, D. D. 2000. *Visual intelligence: how to create what we see*. Norton, New York, NY.

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

HORNUNG, A., DEKKERS, E., AND KOBBELT, L. 2007. Character animation from 2D pictures and 3D motion data. *ACM Transactions on Graphics 26,* 1, 1–9.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3d freeform design. In *Proc. SIGGRAPH*. 409–416.

JACOBSON, A. AND SORKINE, O. 2011. Stretchable and Twistable Bones for Skeletal Shape Deformation. *ACM Trans. Graph. 30,* 6, 165:1–8.

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

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

KOFFKA, K. 1955. *Principles of Gestalt Psychology*. International library of psychology, philosophy, and scientific method. Routledge & K. Paul.

LEVI, Z. AND GOTSMAN, C. 2013. ArtiSketch: A System for Articulated Sketch Modeling. *Computer Graphics Forum 32,* 2, 235–244.

MARAFFI, C. 2003. *Maya Character Creation: Modeling and Animation Controls*. Voices that matter. Pearson Education.

MOESLUND, T. B., HILTON, A., AND KRGER, V. 2006. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding 104,* 23, 90 – 126.

NAKAYAMA, K. AND SHIMOJO, S. 1992. Experiencing and Perceiving Visual Surfaces. *Science 257*, 1357–1363.

NEALEN, A., SORKINE, O., AND ALEXA, M. 2007. FiberMesh: Designing Freeform Surfaces with 3D Curves. *ACM Trans. Graph. ACM SIGGRAPH 2007 Papers 26,* 3, 1–8.

NEALEN, A., SORKINE, O., ALEXA, M., AND COHEN-OR, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph. 24,* 3, 1142–1147.

OLSEN, L., SAMAVATI, F., AND JORGE, J. A. 2011. Natura sketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31,* 6, 24–34.

OLSEN, L., SAMAVATI, F., SOUSA, M., AND JORGE, J. 2009. Sketch-based modeling: A survey. *Computers & Graphics 33*.

PAPERMAN. 2012. Walt Disney Animation Studios.

PIZLO, Z. AND STEVENSON, A. 1999. Shape constancy from novel views. *Perception & Psychophysics 61,* 7, 1299–1307.

RIVERS, A., DURAND, F., AND IGARASHI, T. 2010. 2.5D cartoon models. *ACM Transactions on Graphics*.

SAND, P., MCMILLAN, L., AND POPOVIC, J. 2003. Continuous capture of skin deformation. *ACM Transactions on Graphics (TOG)*, 578–586.

SCHMID, J., SENN, M. S., GROSS, M., AND SUMNER, R. W. 2011. Overcoat: an implicit canvas for 3d painting. *ACM Trans. Graph. 30*, 28:1–28:10.

SHAO, C., BOUSSEAU, A., SHEFFER, A., AND SINGH, K. 2012. Crossshade: Shading concept sketches using cross-section curves. *ACM Trans. Graphics 31,* 4.

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

SÝKORA, D., KAVAN, L., ČADÍK, M., JAMRIŠKA, O., JACOBSON, A., WHITED, B., SIMMONS, M., AND SORKINE-HORNUNG, O. 2014. Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transaction on Graphics 33,* 2, 16.

TAI, C.-L., ZHANG, H., AND FONG, J. C.-K. 2004. Prototype Modeling from Sketched Silhouettes based on Convolution Surfaces. *Computer Graphics Forum 23,* 1, 71–83.

WILLIAMS, R. 2001. *The Animator's Survival Kit*. Faber and Faber.

WONG, K.-Y. K., MENDONA, P. R., AND CIPOLLA, R. 2004. Reconstruction of surfaces of revolution from single uncalibrated views. *Image and Vision Computing 22,* 10, 829 – 836.

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

YE, G., LIU, Y., HASLER, N., JI, X., DAI, Q., AND THEOBALT, C. 2012. Performance capture of interacting characters with handheld kinects. In *Proc. European Conference on Computer Vision*. 828–841.