True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization

Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James Mccrae, Karan Singh

▶ To cite this version:

Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James Mccrae, et al.. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. ACM Transactions on Graphics, 2014, Proceedings of SIGGRAPH, 33 (4), 10.1145/2601097.2601128 . hal-01060850

HAL Id: hal-01060850
https://inria.hal.science/hal-01060850

Submitted on 4 Sep 2014

True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization

Baoxuan Xu[1] William Chang[1] Alla Sheffer[1] Adrien Bousseau[2] James McCrae[3] Karan Singh[3]
[1] University of British Columbia [2] Inria [3] University of Toronto

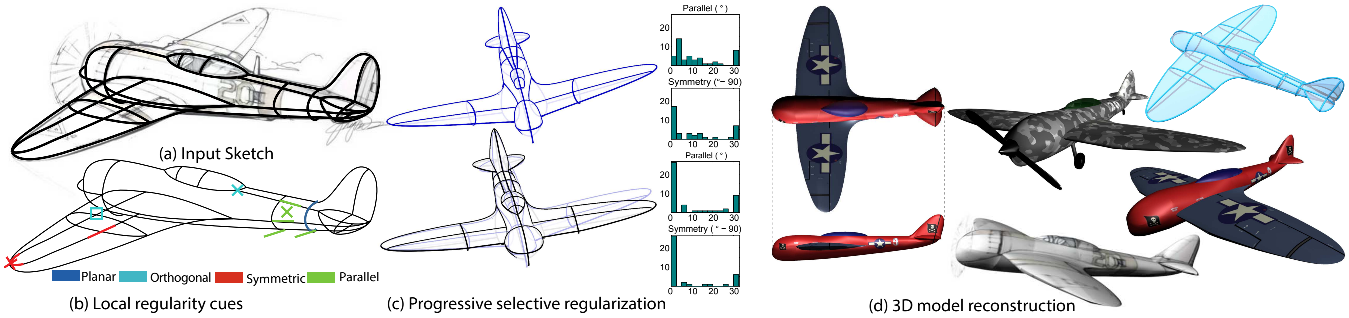


Figure 1: *True2Form takes as input a 2D vector sketch, here traced on a design drawing (a). We formulate a set of local 3D regularity properties that our algorithm detects and applies selectively to lift the curves off the page into 3D (b). A baseline inaccurate 3D result (c, top), is gradually corrected by the selective regularization process (c, bottom). The accompanying histograms show how angles between all potentially parallel and locally symmetric curves are randomly spread initially but converge to a globally consistent state where these properties are either ignored or precisely enforced. For example, the curve tangents on the tapered fuselage are not treated as parallel in global context, despite being near-parallel in 2D (b, green cross). Our 3D curve network can be easily surfaced to bring the sketch to life or integrate the airplane design with its drive-train (d). Airplane sketch by Spencer Nugent.*

Abstract

True2Form is a sketch-based modeling system that reconstructs 3D curves from typical design sketches. Our approach to infer 3D form from 2D drawings is a novel mathematical framework of insights derived from perception and design literature. We note that designers favor viewpoints that maximally reveal 3D shape information, and strategically sketch descriptive curves that convey intrinsic shape properties, such as curvature, symmetry, or parallelism. Studies indicate that viewers apply these properties selectively to envision a globally consistent 3D shape. We mimic this selective regularization algorithmically, by progressively detecting and enforcing applicable properties, accounting for their global impact on an evolving 3D curve network. Balancing regularity enforcement against sketch fidelity at each step allows us to correct for inaccuracy inherent in free-hand sketching. We perceptually validate our approach by showing agreement between our algorithm and viewers in selecting applicable regularities. We further evaluate our solution by: reconstructing a range of 3D models from diversely sourced sketches; comparisons to prior art; and visual comparison to both ground-truth and 3D reconstructions by designers.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: sketch-based modeling, descriptive lines, regularity

Links: DL PDF

1 Introduction

Product design, from the inception of an idea to its realization as a 3D concept, is extensively guided by free-hand sketches [Pipes 2007; Eissen and Steur 2011]. Sketches form an appropriate projective surface for our mental vision, allowing designers to quickly explore the essence of a 3D shape. Sketches also form an effective tool for visual communication, leveraging the human ability to imagine a 3D shape from a compact set of descriptive, semantically diverse, though often inexact, 2D curves. We combine design and perceptual principles to propose *True2Form*, a sketch-based modeling tool that infers 3D shapes from a single design sketch. We focus on reconstructing free-form, piecewise-smooth models of man-made objects. In this domain, we exploit the tendency of designers to strategically sketch curves that are *descriptive* of the geometry. These curves trigger our perception of *geometric regularities* that aid the inference of depth in line drawings. The phrase "true to form" meaning "exactly as expected", signifies our attempt to reproduce the 3D "form" viewers expect from a 2D sketch.

Techniques supporting digital design exist in a continuum between traditional 2D sketching and 3D CAD modeling. 3D sketch-based *multi-view* workflows like Fibermesh [Nealen et al. 2007] and ILoveSketch [Bae et al. 2008] combine 2D drawing and 3D navigation to interactively bridge sketching and CAD modeling. We, in contrast, leverage the descriptive power of 2D design sketches [Eissen and Steur 2011; Pipes 2007] to directly estimate 3D shape. Designers using *True2Form* simply draw in 2D or trace over existing sketches (Figure 1 (a)) and disambiguate accidental occlusion from curves that intersect in 3D, as they commonly do in real drawings. Our system is then able to infer a network of general 3D curves, without requiring the 3D shape or its scaffold to be dominated by straight lines [Lipson and Shpitalni 1996; Schmidt et al. 2009b]. Despite its *single-view* focus, *True2Form* can incorporate and benefit from occasional view changes while drawing (Figure 13).

We base our approach on the premise that the drawing principles guiding designers in their choice of descriptive curves are the same principles that aid viewers in lifting the sketch off paper into a 3D shape. We draw inspiration from design, vision and perception literature in our formulation of two principles, *sketch fidelity* and *shape regularity*. Fidelity expresses a two-way relationship between the sketch and the shape it represents - the sketch should be a faithful projection of the 3D shape and the 3D geometry should be maximally captured by the sketch with only minor shape variation between the 2D curves and their 3D counterparts (Figure 2). Fidelity itself is perfectly satisfied by a flat 3D shape coincident with the sketch, regardless of how geometrically irregular the shape might seem. Designers encourage 3D, out of plane shape interpretations by drawing curves that emphasize shape regularity and intrinsic properties like curvature and symmetry. For example, we observe that smoothly crossing curves are universally perceived as orthogonal in 3D and that proximal curves intersecting a common curve often convey parallelism in 3D (Figure 1 (b)). The perceptual motivation for these two principles comes from viewer preference for regular shapes [Pizlo and Stevenson 1999] seen from non-accidental viewpoints [Nakayama and Shimojo 1992].

Enforcing regularity cues lifts the relevant portions of the curve network into 3D, off the paper. Our algorithmic challenge is for each sketch to detect the same set of globally-compatible regularity cues that humans would employ in their interpretation of a consistent 3D shape. Enforcing a set of cues that is either too restrictive or too lax, leads to distorted results (Figures 4, 5). Our framework builds upon observations specific to design sketches, to detect *applicable* regularity cues consistent with those perceived by viewers. We first generate an initial 3D reconstruction by enforcing orthogonality of smoothly crossing curves while satisfying sketch fidelity. We next note that the degree to which a regularity cue is satisfied in this reconstruction is strongly indicative of its applicability. Thresholding all plausible regularity cues at once however, results in inconsistent reconstructions as it ignores interconnections over the global network of curves (Figure 5). Instead, we cast regularity detection as a rounding problem, that progressively drives the continuous applicability likelihoods of individual cues defined over the $[0, 1]$ range to binary values. We express regularities as soft constraints weighted by their likelihood and progressively round likelihoods within an ϵ of 1 or 0, determining the regularities to be applicable or not. Each rounding yields a new 3D curve network that optimizes fidelity subject to hard applicable regularity constraints and soft likelihood-weighted unresolved ones, updating the likelihoods of the unresolved cues. The process is repeated until all regularity cues are resolved. Our detection and strict enforcement of regularities consistent with human perception, is key to correcting the inevitable inaccuracy in sketches [Schmidt et al. 2009a].

Our core technical contribution is thus both a formulation of the geometric properties of *descriptive* sketch curves, detailed in Section 3, and a framework to reconstruct 3D models from sketches by optimizing fidelity subject to automatically detected applicable regularity cues, described in Section 4. We validate this contribution in Section 6 by comparing the regularities selected by our algorithm with the ones perceived by human observers. Our algorithm largely makes the same choices as humans on a test set with 50 questions.

We also evaluate our algorithmic implementation (Section 5) in a variety of ways (Section 7): we present a large compelling set of complex 3D models created from diverse design sketches; we show our reconstructed 3D curve networks to be plausible by conducting a qualitative comparison to artist-estimated models combined with visual validation by designers; we analyze the impact of varying both curve input and solution strategies on the output 3D curves; and lastly compare our results to prior art.

2 Related Work

Sketch-based modeling has matured over the past two decades; we refer readers to [Olsen et al. 2009] for a survey of existing methods. Sketching interfaces can be roughly described as based on a single-view or a multi-view metaphor. Using multi-view tools, artists sketch strokes from different viewpoints onto existing 3D geometry [Igarashi et al. 1999; Nealen et al. 2007; Kara and Shimada 2007; Orbay and Kara 2012] or use strokes to define transient construction surfaces on which 3D curves are drawn [Bae et al. 2008]. Such tools combine the fluidity of sketching with a typical 3D CAD workflow based on frequent view changes. In contrast, we adopt a single-view approach that minimizes view changes to mimic traditional pen-on-paper sketching and allow 3D recovery from existing drawings. Figure 19 demonstrates our method's ability to reconstruct 3D models similar to those sketched with ILoveSketch [Bae et al. 2008] from their 2D projection alone, while Figure 13 shows how our algorithm also generalizes to multi-view interaction.

Recent single-view approaches rely on user indications or construction lines to model smooth objects. Schmidt et al. [2009b] require users to specify polyhedral scaffolds as a support for 3D recovery from input sketches. Figure 18 demonstrates that our algorithm is capable of reconstructing similar 3D models without the need for 3D scaffolds. Olsen et al. [2011] and Sýkora et al. [2014] combine user indications and shape inflation to model smooth shapes from existing drawings and photographs, while Gindgold et al. [2009] let users position parameterized primitives on an existing sketch, using various annotations to enforce alignment, equal length and symmetry. Shtof et. al [2013] and Chen et al [2013] combine user indications and optimization to snap parameterized primitives to contours in an image. Instead, our approach builds directly upon the descriptive power of artist-drawn curves to recover piecewise-smooth design curve networks with minimal annotation.

Past sketch based modeling methods [Andre and Saito 2011; Shao et al. 2012] have used a subset of descriptive curves called cross-sections. Cross-sections are planar curves, assumed to have orthogonal planes and tangents at intersections. The space of shapes described strictly by these curves is somewhat narrow (Figure 20, first row). This space is further narrowed by [Andre and Saito 2011], who make additional regularity assumptions on the curve networks. Shao et al. [2012] infer cross-sections as a step towards estimating a 3D normal field, rather than a 3D curve network. Consequently their 3D cross-section networks are often disconnected (Figure 20, third row) and instead of correcting for drawing inaccuracies they absorb them into an approximate but smooth normal field (Figure 20, second row). Both artifacts are inappropriate in the context of 3D shape reconstruction.

Automatic line-drawing interpretation aims to reconstruct shapes from silhouette and other feature-lines in a single image [Cooper 2008], a challenging and often ill-posed task [Malik 1987]. Inspired by research in computer vision, e.g. [Lowe 1987], Lipson and Shpitalni [1996] and subsequent work [Lee et al. 2008; Lau et al. 2010; Tian et al. 2009] estimate 3D models from engineering drawings dominated by straight lines and orthogonal corners, by detecting and enforcing regularity constraints such as straight-line parallelism and corner orthogonality. Wang et al. [2009] combine these cues with face orthogonality and curve parallelism, extending the set of reconstructed models. While we draw inspiration from this body of work, our algorithm is aimed at a more ambitious set of input sketches, in particular piecewise-smooth curve networks that dominate modern product design. These curve networks have a much greater degree of freedom and few if any of the regularities listed above, and thus cannot be handled by existing methods.

Properties of artist-drawn descriptive curves have also been used to

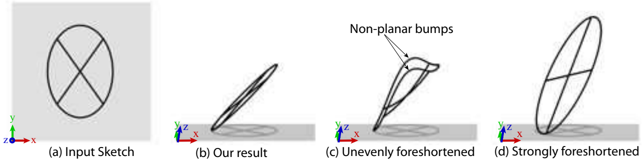


Figure 2: *The input sketch (a) is a precise projection of all three shapes (b..d), that also show the sketch plane faintly from a near horizontal view. We lift the sketch to a near circular 3D reconstruction in (b), with small and uniform foreshortening. The shape in (c) is similar to (b) barring two non-planar bumps, missing in the accidental view of the sketch. The elliptical shape in (d) has uniform but unexpectedly large foreshortening.*

design algorithms for cycle detection [Zhuang et al. 2013] and surfacing [Bessmeltsev et al. 2012] of 3D curve networks. Our method complements these algorithms by performing the necessary, earlier step of converting a 2D sketch into a 3D curve network.

3 Understanding Sketched Curve Networks

We combine observations from the design and perceptual literature to formulate properties of design sketches that explain their effectiveness in conveying complex, smooth 3D shapes. Existing modeling research has validated subsets of our properties in different contexts such as engineering drawings dominated by straight orthogonal lines [Lipson and Shpitalni 1996; Tian et al. 2009] and drawings composed of orthogonal planar cross-section curves [Shao et al. 2012]. We complement these findings to propose a more comprehensive set of properties representative of general design sketches of piecewise-smooth shapes. Our observations point to two contributing factors guiding sketch interpretation: *sketch fidelity* and *regularity* of the 3D interpretation of a sketch.

3.1 Sketch Fidelity

Projection accuracy. While designers often draw with approximate perspective [Schmidt et al. 2009a], they aim at producing a faithful projection of the imagined 3D shape. This observation implies that the projection of our 3D reconstructions to the view-plane should align as much as possible to the input curves while granting leeway to correct for inaccuracy inherent to free-hand sketching.

Minimal variation. Design books recommend using viewpoints that "optimize shape information" and instruct designers to *minimize foreshortening* over most faces of the object [Eissen and Steur 2011]. This recommendation is consistent with the perceptual notion of *general* or *non-accidental viewpoints* [Nakayama and Shimojo 1992; Mather 2008], which suggests that observers interpret 2D geometric properties as strongly correlated with 3D geometry rather than being caused by a particular choice of viewpoint.

Prior work builds on this principle by predicting that straight lines in 2D correspond to straight lines in 3D [Lowe 1987; Lipson and Shpitalni 1996; Tian et al. 2009]. While we adopt a similar observation for straight lines, it is not sufficient to relate the 2D and 3D geometry of curved lines in sketches of smooth shapes. We introduce a more general principle of *minimal 2D-to-3D variation* that states that the shape of the 2D curves should closely reflect their shape in 3D, up to small and evenly distributed foreshortening (Figure 2). In other words, we expect the 2D and 3D curves to be locally affine invariant. Affine invariance, together with projection accuracy suggests a designer preference for locally planar curves, an observation supported by perceptual studies that report better viewer understanding of shapes represented using planar contours [Pizlo and Stevenson 1999].

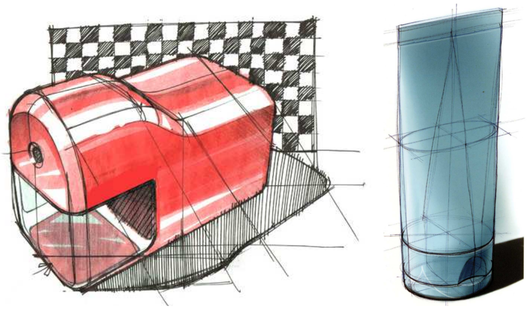


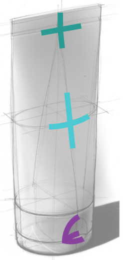
Figure 3: *Descriptive curves convey 3D shape in design sketches, with occluded curves often drawn faintly. See insets in the text for highlighted properties. Pencil sharpener by Spencer Nugent©, toothpaste by Koos Eissen and Roselien Steur©.*

Ambiguity. Because sketches form 2D projections of 3D shapes they exhibit two inherent types of ambiguities. First, in the presence of hidden lines the curve geometry alone does not provide enough information to distinguish occlusions from intersections. Artists employ different conventions to convey occlusions, such as drawing hidden parts with faint or dashed lines (Figure 3). Our user interface supports similar disambiguation (Section 5.1). Second, sketches suffer from an inherent convex/concave ambiguity allowing for alternative global interpretations. We adopt the approach of Shao et al. [2012] to resolve the ambiguity by favoring more convex shapes viewed from above. We also allow users to select a different interpretation, if the solution is not what they envisioned.

3.2 Shape Regularity

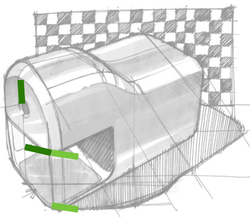
Fidelity, taken in isolation, is best satisfied by the least foreshortened interpretation: the 2D curve network itself. Perceptual studies and modeling research suggest that viewers imbue line-drawings with depth variation by imposing *regularity* constraints on the possible solutions [Lipson and Shpitalni 1996; Pizlo and Stevenson 1999], consistent with the *minimum principle* or *Law of Prägnanz* from Gestalt theory [Koffka 1955]. These constraints however, are inherently context-based and are, we believe, only applied when consistent with each other and with sketch fidelity.

Orthogonality. Perceptual studies indicate that observers interpret intersecting smooth curves, or *smooth-crossings*, as aligned with the lines of curvature of an imaginary surface [Stevens 1981; Mamassian and Landy 1998], and thus having orthogonal tangents at these intersections. Designers leverage this perceptual bias to depict smooth shapes effectively using cross-sections [Shao et al. 2012] and other curves [Bordegoni and Rizzi 2011; Bessmeltsev et al. 2012] aligned with lines of curvature.



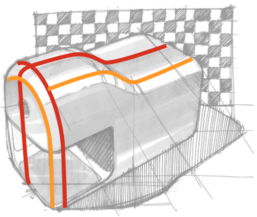
Our observations, confirmed by the study in Section 6, indicate that smooth-crossings are always perceived as orthogonal by viewers and we speculate that such intersections are specifically drawn to convey orthogonality (inset, light cyan). In contrast other intersections like sharp corners or U-turns, can indicate geometric features such as sharp edges or silhouettes, where the intersecting curves are not necessarily orthogonal. Still, these intersections also tend to be interpreted as orthogonality cues, when context allows for it (inset, dark cyan and purple), as studied by Perkins [1971] for triplets of lines depicting cubic corners.

Parallelism. The preference for regular shapes as well as the expectation of minimal 2D to 3D variation suggest that curve parallelism observed in 2D extends to 3D. Parallelism of straight lines have been successfully used in 3D reconstruction from both images [Lowe 1987] and engineering drawings [Lipson and Shpitalni 1996; Tian et al. 2009]. Unfortunately design sketches contain few straight lines and rarely contain complete parallel curves.



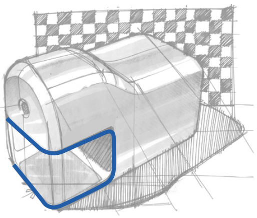
Following an extensive review of design sketches we deduce a local parallelism cue applicable to sketches composed of curved lines. We observe that artists tend to strategically place intersecting curves along a given curve such that the tangents at adjacent intersections are frequently parallel (inset, light green). Our study indicates that viewers tend to interpret such adjacent 2D tangents as parallel in 3D, when context allows for it.

Symmetry. Designers also position curves to emphasize intrinsic shape properties like local symmetries (inset, red) [Eissen and Steur 2011], consistent with viewer tendency to interpret intersecting curves as geodesics over smooth surfaces [Knill 1992].



Shao et al. [2012] account for this property by encouraging cross-section curves to align with geodesics at all intersections. Most shapes however, cannot be fully described by geodesics alone and sketches often contain other curves (inset, orange).

Curve planarity. Pizlo and Stevenson [1999] have shown the importance of planar contours on our ability to recognize the same shape under different viewpoints.



Designers exploit this perceptual effect by drawing globally planar cross-section curves over smooth surfaces [Eissen and Steur 2011; Shao et al. 2012]. Artistic preference for planar curves also relates to our *minimal variation* principle since planar curves are affine invariant under near-orthographic projections. Many 3D objects however, cannot be represented only with planar curves, requiring non-planar descriptive lines to effectively communicate their shape (inset, dark blue).

Planar curves when present facilitate more global regularities, such as curve parallelism and orthogonality. A design tutorial comments "usually the sections are perpendicular" [Eissen and Steur 2011] and our observation of design sketches confirms frequent use of curves lying in orthogonal or parallel planes. Similar grouping strategies have been observed in plane-based 3D shape abstraction [McCrae et al. 2011].

Applicability. The regularity cues discussed have two common properties. First, with the exception of smooth-crossing orthogonality, they all are context-based and may or may not apply in any particular instance. Second, as is the nature with regularizers, they are strict – a curve is either a geodesic or not, two intersecting curves are either orthogonal or not. In other words, artists and viewers do not make an effort to optimize toward regularity (Figure 4 (b)), they either locally expect full regularity or none. We use these observations to detect applicable regularizers and reconstruct 3D curve networks from artist sketches.

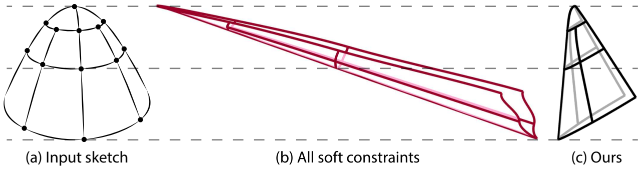


Figure 4: *Treating all possible regularity cues as soft constraints generates an unnatural 3D output (b). Detection and strict enforcement of only the applicable cues produces a plausible 3D shape (c).*

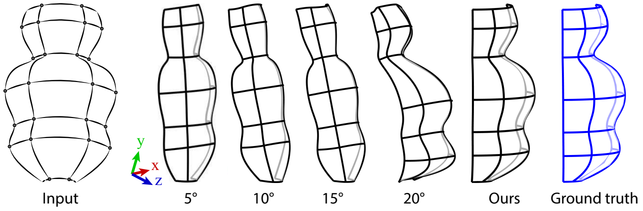


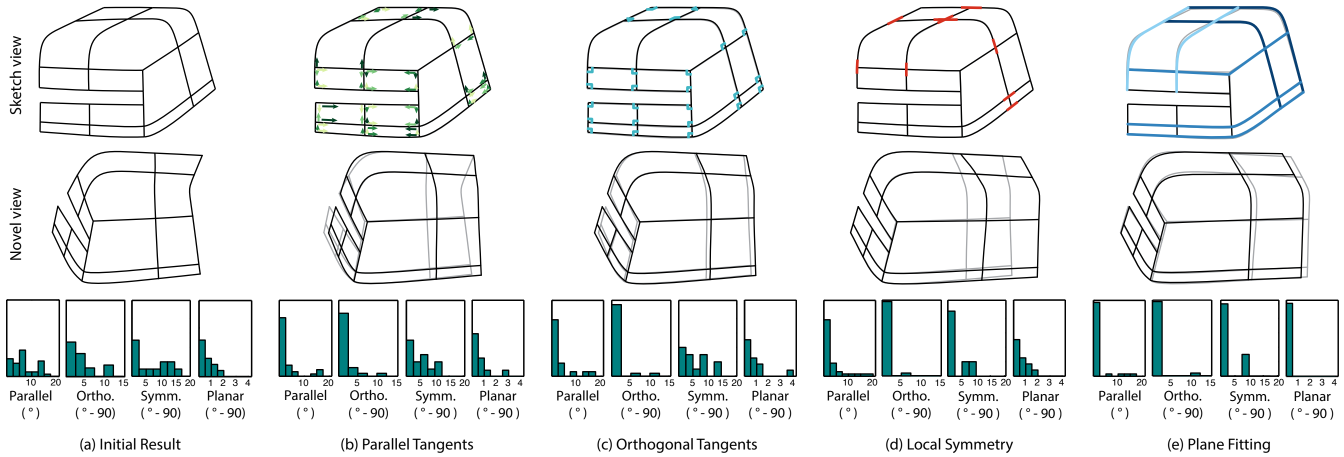
Figure 5: *Simple thresholding of the regularizers by the listed hard angle threshold to determine applicability does not account for interconnectedness between the chosen regularities, resulting in awkwardly shaped 3D reconstructions. Our method (right) correctly detects the applicable set of regularities, leading to a near perfect reconstruction of the sketched ground truth model.*

4 Reconstruction Framework

Given a set of applicable regularities, we can formulate 3D reconstruction as a constrained optimization problem, optimizing fidelity subject to regularity constraints (Section 5). However, to obtain the desired reconstruction we must first determine which regularities are applicable before applying them strictly and selectively. An incorrect set of regularity cues, either too large or too lax (Figures 5 and 6 (a..d)) leads to imperfect reconstruction.

To start the process we recall that orthogonality at smooth crossings is one regularity that always applies. Optimizing fidelity subject to smooth-crossing orthogonality lifts the curve network into a baseline reconstruction that we use to bootstrap our search for additional applicable regularities. Specifically, we observe that the degree to which each regularity is satisfied in this baseline solution provides a local predictor, or likelihood, of its applicability. Unfortunately, directly enforcing all regularities that fall within some likelihood threshold can have significant undesirable global effects as it ignores interconnections within the network. Figure 5 shows the distorted reconstructions produced by such a naive thresholding approach.

Instead, we propose a method for determining applicability inspired by recent integerization and classification methods developed for mesh processing tasks [Bommes et al. 2009; Sharf et al. 2008; Li et al. 2011] as well as regularization strategies used in line drawing beautification [**?**]. The common denominator of these techniques is the initial use of continuous variables as approximation of the target discrete ones, followed by a progressive rounding process that pushes these values toward their discrete counterparts. We propose a similar approach tailored to the specifics of our setup. Instead of directly classifying regularities as applicable or not, we associate each with a likelihood score in the $[0, 1]$ range based on the degree to which it is satisfied. A likelihood score of one indicates an applicable regularity, and a score of zero indicates an inapplicable one. We then add these regularities into the optimized energy functional as soft optimization terms weighed proportionally to their likelihood. We optimize the augmented functional subject to previously detected applicable regularities and re-compute the likelihood scores on the evolving 3D curve network. When a likelihood score



(a) Initial Result (b) Parallel Tangents (c) Orthogonal Tangents (d) Local Symmetry (e) Plane Fitting

Figure 6: *Stages of solving the card reader model. Each step has an overlay with the previous step and indicates the newly found regularizers. Histograms show the distribution of angles of each regularizer at each step of the process.*

gets close to one, we classify the corresponding regularity as applicable, and when it falls below ϵ we classify it as inapplicable. We repeat the optimization process until all regularities are classified, or until likelihood scores no longer change between iterations. In the latter case we use the conservative solution of classifying the remaining regularities as inapplicable.

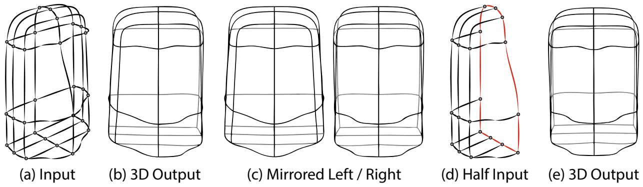
We note that it is beneficial to process the set of potential regularity cues one class of cues at a time, rather than all at once. Some cues like tangent parallelism are easier to predict from the 2D sketch and can be processed first. Other regularity types are correlated and resolving one class of cues can help detect the applicability of another: local symmetry is only meaningful at orthogonal intersections; and curve level cues become more evident after local cues are resolved. Since enforcing curve-level regularities requires per-curve plane fitting it can introduce bias if done sequentially. To avoid biasing planarity and inter-plane relationships should be detected and enforced all at once. We thus resolve regularities in a logical class order of parallelism, orthogonality, and symmetry, followed by curve level and inter-curve regularities. Processing in order inflicts small changes to the output at every stage, further adding numerical stability and efficiency. Overall, we thus process each class of regularity cues in sequence, and progressively resolve all possible cues in that class as applicable or not, before moving to the next class.

5 Implementation

We first describe our sketching interface followed by our numerical formulation of the curve properties from Section 3, and details of the regularity detection algorithm proposed in Section 4.

5.1 User Interface

Our algorithm is integrated with a vector-based single-view sketching system that provides basic curve drawing and editing functionality (see accompanying video) and can import/export sketches from popular vector graphics software like Adobe Illustrator, Autodesk SketchBook Designer and Maya. We support the use of layers to represent independent parts of an object (see the mixer sketch in Figure 16 where different layers are assigned different shades of gray). We stack the reconstructions of independent parts from back to front to generate a complete 3D model, using a z-buffer like algorithm to prevent interpenetration.



(a) Input (b) 3D Output (c) Mirrored Left / Right (d) Half Input (e) 3D Output

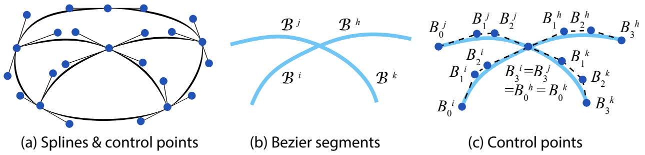
Figure 7: *Even well drawn sketches (a) are unable to perfectly capture global symmetry, resulting in asymmetric reconstruction (b). Instead of symmetrizing the output, such as by mirroring one or the other half (c), we let artists optionally draw half the shape and indicate global symmetry in the drawing (d, red curve), from which we are able to reconstruct a perfectly symmetric 3D output (e).*

Annotations. To distinguish 2D curve crossings which correspond to 3D intersections from those caused by occlusions, we request users to annotate the latter. In our figures we represent valid curve intersections with small dots. Users can also optionally flip between the convex and concave interpretation at any intersection if the default setting does not correspond to the intended shape.

Modeling globally symmetric shapes, common in product design, can be eased by allowing users to draw half the shape and annotate its global symmetry curve (Figure 7). This option significantly reduces user effort in both drawing and occlusion annotation of the input. We handle global symmetry by enforcing planarity, orthogonality, and symmetry along the marked curve, and mirror the final 3D reconstruction. Providing these annotations took less than five minutes for the most complex results in this paper (Figure 21).

5.2 Numerical Formulation of Curve Properties

We design our curve representation and the property formulations to be as simple as possible to optimize numerically, while adequately capturing the complexity of design sketches. We represent both input and output curves by piecewise cubic (4-point) Bézier splines. We place Bézier segment end-points at network intersections and introduce segments as necessary between intersections to accurately capture the sketched 2D geometry (Figure 8). The Bezier representation is transparent to users, who draw strokes via Illustrator, Maya, or our internal polyline-based UI. We fit Bezier curves to strokes adaptively using iterative least-squares. Complex curve sections can and do consist of multiple segments (Figure 8



(a) Splines & control points (b) Bezier segments (c) Control points

Figure 8: *Cubic Bézier network representing an input sketch (a). Notations: Bézier segments (light blue), control points (dark blue) and polygons (dashed lines) (b,c).*

(top curve), wing of the fighter in Figure 18). The use of an internal Bezier representation is motivated by the observations of [Bae et al. 2008] that designers draw smooth inflection-free strokes well captured by cubic Bezier segments. Our ”minimal variation” principle suggests that the complexity of the 2D curves is reflective of their 3D form. Representing curves with Bezier segments simplifies our numerical formulation without compromising the expressiveness of our 3D networks.

Following the *minimal variation* principle which suggests that the complexity of a 2D curve reflects its 3D complexity, we expect the number and distribution of control points sufficient to represent the curve shape in 2D to be sufficient to reconstruct its 3D counterpart. Using one to one 2D to 3D control point correspondence simplifies the formulation and reduces the number of variables.

We denote the i^{th} Bézier segment as $\mathcal{B}^i$, its 3D control points as $B^i_{0..3}$ and their original locations in the 2D sketch as $\bar{B}^i_{0..3}$. We use a coordinate system where x and y are the horizontal and vertical axes of the image plane, and z is the depth axis. For each pair of consecutive G^1 continuous segments $\mathcal{B}^i$ and $\mathcal{B}^h$, we constrain the control points B^i_2, $B^i_3 = B^h_0$, B^h_1 to be collinear, maintaining G^1 continuity (Figure 8), as dictated by the minimal variation principle. Straight line Bézier segments are similarly constrained.

Sketch Fidelity We encode fidelity to the sketch by relating the 2D and 3D Bézier control polygons, instead of the curves themselves. This approach drastically simplifies the formulation with no discernible negative impact on the results.

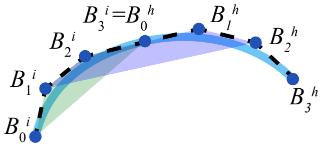
Projection Accuracy. We account for projection accuracy using two terms, one accounting for absolute positions of the control points and one accounting for the slope of control polygon edges

$$E_{accuracy} = \sum_{i,k} \left\| B^i_k.(x,y) - \bar{B}^i_k \right\|^2 + \quad (1)$$
$$\sum_{i,k} \omega_d(d) \left\| \left(B^i_{k+1}.(x,y) - B^i_k.(x,y) \right) - \left(\bar{B}^i_{k+1} - \bar{B}^i_k \right) \right\|^2.$$

Our control point spacing reflects both intersection frequency and local shape variation and is thus intentionally uneven. To account for uneven length of polygon edges we weight the individual slope terms, to allow for greater deviation as the spacing increases using:

$$\omega_d(d) = e^{-d^2/2\sigma^2} + \epsilon. \quad (2)$$

Here, $d = \left\| \bar{B}^i_{k+1} - \bar{B}^i_k \right\|$, σ is set using the standard three sigma rule to 1/3 of the sketch bounding box diagonal, and $\epsilon = 0.01$. Our formulation implicitly assumes orthographic projection, since Schmidt et al. [2009a] and our experiments indicate that the perspective in free-hand sketches is too inexact to meaningfully invert.



Minimal variation. To encode the expectation that 2D and 3D curves are affine invariant, we leverage affine combinations of adjacent 2D control points. For affine invariant curves, subject to orthographic projection, such combinations are expected to be invariant. In this formulation we separately account for quadruplets of adjacent control points in general position (i.e. where no three points are collinear) and triplets of adjacent collinear control points. In the general case, illustrated in the inset, we consider both the four control points defining each Bezier segment $\mathcal{B}^i$ (green quadruplet, with points $B^i_0, B^i_1, B^i_2, B^i_3$) and adjacent control points across consecutive segments $\mathcal{B}^i$ and $\mathcal{B}^h$ (blue quadruplet, with points $B^i_1, B^i_2, B^h_1, B^h_2$). For all such quadruplets we denote the four control point positions as $Q_0 \ldots Q_3$ and compute the affine weights q_0, q_1, q_2 that satisfy

$$\bar{Q}_3 = q_0\bar{Q}_0 + q_1\bar{Q}_1 + q_2\bar{Q}_2, \ \textbf{s.t.}\ q_0 + q_1 + q_2 = 1. \quad (3)$$

over the input 2D curves. Note that the weights can often be negative. For affine invariant curves under orthographic projection this relationship will hold for the z coordinate as well. When three control points considered are near collinear in 2D the computation of affine weights in Equation 3 will either fail or produce one or more zero weights. Instead, we encode minimal variation for any triplet T_0, T_1, T_2 of collinear control points using linear interpolation, which similarly holds under orthographic projection. We thus compute t_0 in 2D to satisfy

$$\bar{T}_1 = t_0\bar{T}_0 + (1 - t_0)\bar{T}_2 \quad (4)$$

Minimal variation is then expressed as

$$E_{variation} = \sum_Q \omega_d(d) \left\| q_0Q_0 + q_1Q_1 + q_2Q_2 - Q_3 \right\|^2$$
$$+ \sum_T \omega_d(d) \left\| t_0T_0 + (1 - t_0)T_2 - T_1 \right\|^2, \quad (5)$$

where the sums go over all the quadruplets and triplets of interest The weights $\omega_d(d)$ are computed using Equation 2 with d as the length of the longest line between the participating points.

Foreshortening. We encode minimal foreshortening as

$$E_{foreshortening} = \sum_{i,k} \omega_d(d) \left(B^i_{k+1}.z - B^i_k.z \right)^2. \quad (6)$$

where the sum goes over all successive control points of all curves and $d = \left\| \bar{B}^i_{k+1} - \bar{B}^i_k \right\|$.

The *fidelity energy* we optimize combines these three terms

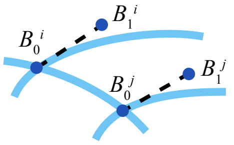
$$E_{fidelity} = w_a E_{accuracy} + w_v E_{variation} + w_f E_{foreshortening} \quad (7)$$

using projection accuracy weight of $w_a = 1$, half as important as shape variation $w_v = 2$, and a weak foreshortening weight $w_f = 0.001$. When minimizing this energy we constrain one control point to have $z = 0$, since the energy is invariant under z translation.

Shape Regularity We complement fidelity energy with regularity constraints to bring depth to the sketch. While these regularity cues are inherently angular, expressing angles as trigonometric functions of control points, is numerically problematic. We thus recast them as linear or quadratic functions of control point positions.

Tangent orthogonality. The tangents at the end points B_0^i or B_3^i of a Bézier curve are simply the vectors to adjacent control points B_1^i or B_2^i respectively. Given two Bézier segments $\mathcal{B}^i$, $\mathcal{B}^j$ that intersect at a shared control point $B_0^i = B_0^j$, tangent orthogonality at the intersection is expressed as

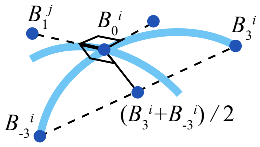
$$C_{orthogonal} = \left(B_1^i - B_0^i\right) \cdot \left(B_1^j - B_0^j\right) = 0.$$



Parallelism. Given two adjacent intersections along a curve (see inset), we consider the two adjacent segments $\mathcal{B}^i$ and $\mathcal{B}^j$ forming these intersections and express parallelism as similarity between their tangent directions, normalized using 2D lengths.

$$C_{parallel} = (B_1^i - B_0^i)/\left\|\bar{B}_1^i - \bar{B}_0^i\right\| - (B_1^j - B_0^j)/\left\|\bar{B}_1^j - \bar{B}_0^j\right\| = 0.$$

Local symmetry. If a curve defines a local symmetry plane, then any curve orthogonal to this curve, is symmetric around this plane, and consequently orthogonal to the plane (as well as the curve). Recall that a vector is orthogonal to a plane, if it is orthogonal to two independent vectors in it. Since we only evaluate symmetry of $\mathcal{B}^i$ at the intersection with $\mathcal{B}^j$ if $\mathcal{B}^j$ is *a priori* orthogonal to $\mathcal{B}^i$ we only need to consider one more vector not collinear with the tangent to $\mathcal{B}^i$ at the intersection. We use the construction shown in the inset to obtain such an in-plane vector which deviates far from the tangent, resulting in



$$C_{symmetry} = \left(\frac{B_3^i + B_{-3}^i}{2} - B_0^i\right) \cdot (B_1^j - B_0^j) = 0. \qquad (8)$$

We do not evaluate symmetry when the considered curve forms a straight line, and hence does not define a unique plane, at the relevant intersection, i.e. when $\frac{B_3^i + B_{-3}^i}{2} - B_0^i \approx 0$.

Curve-level regularity. To evaluate curve planarity as well as interaction between curve planes we introduce a per curve normal variable n. We express planarity as

$$C_{planar} = \sum_{(i,k)\neq(j,l)} \left((B_k^i - B_l^j) \cdot n\right)^2 = 0. \qquad (9)$$

The sum includes all pairs of control points on a given curve. The formulation implicitly grants more weight to pairs of farther apart points, capturing scenarios such as the spiral (Figure 12) where curves are planar locally but not globally. We express parallelism using the same formula, this time applied to the set of curves we wish to make parallel, using a shared normal variable. Orthogonality between curve planes is expressed as the the dot product between their normals. To evaluate curve linearity, or straightness, we measure how well the interior control points can be described as a linear combination of the end points $C_{linear} = \sum_{i,k}(B_0^0 t_{i,k} + B_3^n(1 - t_{i,k}) - B_k^i)^2$, where $t_{i,k}$ are computed by minimizing this formula on the 2D curves.

5.3 Optimization Details

Out goal is to minimize the fidelity energy $E_{fidelity}$ subject to all *applicable* regularity constraints, i.e. those that can be enforced with only minor increase in this energy.

Baseline reconstruction. Given a 2D sketch we obtain the baseline reconstruction by minimizing fidelity subject to smooth-crossing orthogonality constraints. Since minimal variation indicates that tangent parallelism in 2D is a good predictor of parallelism in 3D, we augment the initial optimization with soft parallelism constraints weighted by the 2D angles between each pair of adjacent curves. Specifically, $w_p = 0.1e^{-\alpha^2/\sigma^2}$ where α is the 2D angle and $\sigma = 15°$. At this stage, since few regularities are enforced, we aim to minimize 2D deviation from the sketch, and use a higher accuracy weight $w_a = 10$ in the optimization.

Since orthogonality constraints are quadratic, our constrained optimization is non-convex and needs a good initial guess to converge to the global minimum. A naive guess of setting x and y coordinates of the control points to their positions in the sketch and setting $z = 0$ everywhere, is not sufficient for this purpose. To lift the guess out of the plane, we first lift one arbitrary smooth-crossing into 3D, by solving our optimization problem on a sub-network of four Bezier curves surrounding this crossing, using the view from above preference or user indicated orientation to eliminate ambiguity. We then propagate this solution across the sketch by optimizing our energy functional while over-emphasizing tangent parallelism using $w_p = 5$. The resulting non-flat curve network provides the desired initial guess for our baseline reconstruction.

Likelihood score. All the regularities we enforce can be encoded in terms of angles. We thus use angle based functions to evaluate the likelihood that a specific regularity cue is applicable, given an intermediate 3D curve network. We empirically map angles within a tight inner bound I_b of $5°$ to a likelihood of 1 and those outside a loose outer bound O_b of $15°$ to 0. We also define a sharp likelihood fall-off beyond I_b, using a monomial likelihood function

$$L(\alpha) = \min(1, a\alpha^b). \qquad (10)$$

where α is the angle, and a and b are computed to produce the values of 1 and 0.005 at the two respective bounds.

Progressive optimization. To recover the applicable regularity constraints we iteratively minimize an augmented energy functional

$$E_{fidelity} + w_c \sum L(\alpha_i) C_k^2,$$

where the sum goes over all currently considered regularity cues weighted by their likelihood computed on the evolving 3D curve network, subject to all previously detected applicable regularities. We set the regularity weight to $w_c = 50$, which combined with the sharp fall-off in the likelihood function, ensures that shape regularity does not come at the expense of a large decrease in fidelity. E.g. when the angle deviation is $10°$ the weight product $w_c L(\alpha)$ is only 2. After each solver iteration we classify regularities with a likelihood of 1 as applicable constraints, and those with likelihood 0 or those whose likelihood did not increase following the optimization step as inapplicable. Classified regularity cues are removed from the sum above (applicable cues are added to a set of hard constraints) and the solver iterates until all regularity cues in a class are classified. Four iterations typically suffice for each class.

We use the order of resolution described in Section 4, first optimizing tangent parallelism, the intersection orthogonality, followed by local symmetry cues, only considered for curve intersections already detected as orthogonal.

Curve and network level constraints. We optimize for curve planarity, linearity and inter-plane relations last, processing them together, as sequential processing can, in our experience, bias the solution. Solving together for both control point positions and plane normals is numerically problematic as it requires either minimizing or enforcing a quartic term (Equation 9). We break the process into a local-global solve, first computing best-fitting normals, keeping positions constant and then solving for positions while fixing the normals. Thus, for each curve we first compute an individual best-fit plane normal by minimizing Equation 9. We then compute a weighted average of the angular deviations between this normal and the vectors connecting all pairs of control points along the curve

$$\alpha = \sum_{(i,k)\neq(j,l)} \angle(B_k^i - B_l^j, n) w_n / \sum_{(i,k)\neq(j,l)} w_n$$

where $w_n = (90 - \angle(B_k^i - B_l^j, n))^2$. These angles are used to determine the likelihood $L(\alpha)$ of each curve being planar. Linearity is similarly measured as an angular deviation, this time using angles between pairs of lines connecting control points $\angle(B_k^i - B_l^j, B_m^m - B_p^o)$. To evaluate inter-plane parallelism, we perform single-linkage clustering of plane normals (greedily merging the closest clusters as long as their joint angular deviation with respect to a joint plane normal is within the outer bound O_b). The joint normal for a cluster is the minimizer of Equation 9 over all clustered curves. To evaluate and enforce inter-plane orthogonality we pick pairs of clusters whose plane normals are within O_b of orthogonality, and solve for plane normals jointly by minimizing Equation 9 for each cluster subject to normal orthogonality. We then use Equation 9 with these normals as orthogonality proxy during optimization, and use the average deviation of the planes form the computed normals as orthogonality likelihood.

We use the same iterative resolution process as before augmenting the functional with all the curve-level terms at once each weighed by its likelihood. After each iteration we recompute the per-plane and per cluster normals. If a cluster is deemed non-planar, we generate smaller sub-clusters, by re-clustering its members as before, stopping short of the final merge.

Our goal in detecting applicable regularities is to be conservative, thus when uncertain we aim to delay ultimate decision making until all context is considered. Specifically, since parallelism is determined first, the decision process lacks larger context and can sometime generate false positives. To enable subsequent optimization to relax false regularities, we use soft rather than hard tangent parallelism constraints, with weight w_c. This lazy approach to uncertainty, allows some regularities to be detected indirectly by enforcing others. Curve-level regularities, in particular, may enforce local cues for which there was no definitive evidence a priori (see the local symmetry histogram in Figure 6 (d, e)).

Solver. We use Lagrange multipliers to enforce constraints and minimize the augmented functional using a Newton solver, with bounded step size. Specifically, we bound mutiplier increments to have norm less than one, by normalizing them by the biggest increment size if it is above one. This normalization stabilizes convergence, without significant impact on solver speed. The solver converges within seconds on all the inputs we tested.

6 Perceptual and Design Validation

Given the perceptual complexity of our problem, we chose to validate our hypotheses and design choices in two ways. First, while some of our regularity cues have been anecdotally reported on in *Gestalt* and vision literature, we performed a study to formally test consistency between humans and our algorithm in perceiving these cues in example sketches. Second, to evaluate the plausibility of our 3D outputs, we asked designers to draw and model in 3D their impression of a set of ortho-projected curve renders of 3D models (*ground-truth* data) and compare their outputs with our algorithmic results.

6.1 Sketch Regularity Perception

We aim to answer two questions: **Q1:** Do humans consistently perceive 3D parallelism, orthogonality, symmetry, linearity and planarity cues in 2D sketches? **Q2:** Does human perception, when consistent, match our algorithmic output for the above cues?

Study Design: Our test data-set comprised 10 drawings, representative of the sketch inputs to our algorithm (Figure 9). The drawings comprise a mix of 2 ground-truth sketches and 8 artist sketches, capturing a range of sketch inaccuracy and curve complexity. We included 5 half-drawings of symmetric objects, since our algorithm is capable of using half a sketch to successfully reconstruct the 3D shape. As regularity properties are Boolean, we formulated our study as a set of 50 (Y/N) questions (5 per sketch). The number of sketches and questions was chosen so the typical time to complete the study was 10-15 mins. The questions asked users to imagine the sketch as a 3D model and then answer queries relating to the 5 regularity cues, for example *curve1 imagined in 3D defines a plane of local symmetry for curve3 (Y/N)*. Queried curves were unobtrusively numbered avoiding perceptual bias possible when using color or visual markings. The number of questions pertaining to each of the five cues were: parallelism 9, orthogonality 14, symmetry 5, linearity 12, planarity 10,. The study (supplementary material) was performed by 33 participants (19 electronically, 14 on print-outs), 24 of whom had some computer graphics background.

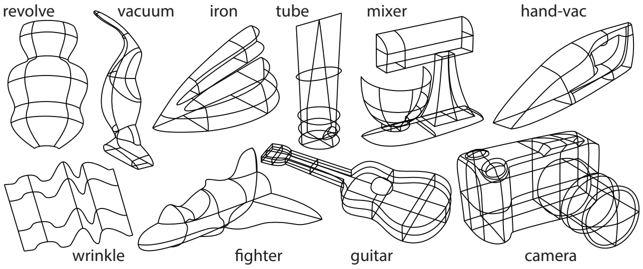


Figure 9: *User study corpus comprising 10 sketches.*

Study Results: We removed two incomplete forms and report the results of 31 participants.

Q1: For a sample size of 31, agreement of 22 or more participants ($> 70\%, p = 0.0294$) is statistically significant (i.e. $p > 0.05$). For the 50 questions the histogram of agreement between participants was $[90 - 100\% : 36, 70 - 90\% : 8, 50 - 70\% : 6]$, indicating statistically significant human consistency in perceiving 44/50 questions. 27 questions had 100% agreement. As expected, the precision and simplicity of the sketch impacted consistency of perception: the revolve, wrinkle, tube and mixer drawings had near perfect agreement on all questions, whereas the iron, vacuum, and hand-vac were the most inconsistent (Figure 10). Broken down by regularity cue the number of significantly consistent answers was (linearity 12/12, planarity 8/10, perpendicularity 13/14, symmetry 3/5, parallelism 8/9). These statistics clearly indicate a consistency in human perception of geometric regularity in sketches.

Q2: Our algorithm agreed with the majority of participants on 46/50 questions. None of the 4 anomalous questions had significant human consistency (see iron and hand-vac for anomalous examples in Figure 10). We believe this is sufficient validation that our algorithm is able to reliably predict geometric regularities within the global context of a design sketch.

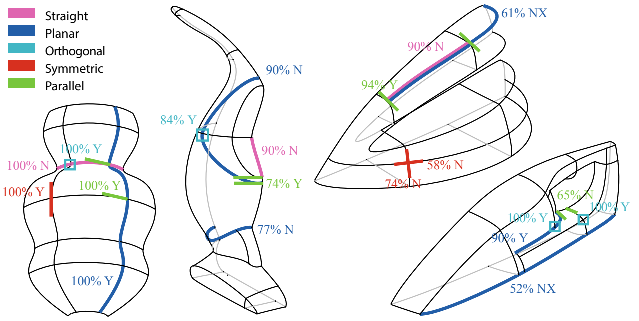


Figure 10: *Queries and results (%age human consistency and majority response) illustrated on 4 models. Questions where the algorithm disagrees with the majority are further labeled X. Red lines mark the curve participants were queried about as being a plane of local symmetry for the intersecting curve.*

6.2 Algorithm Evaluation

To evaluate our end goal of creating plausible 3D shape from 2D sketches, we designed a set of 3D curve networks as ground truth data (Figure 11, left column). These models capture the variety in the curve geometry of real-world design sketches with curves ranging from straight lines (spiral and wrinkle), circular arcs (revolve), inflections (wrinkle and cross) and non-planar curves (spiral and saddle). We then picked typical generic views for these models as input sketches for our algorithm, which successfully constructed a 3D curve network for each input.

We validated our input sketches as ground truth data by testing that they were meaningfully and consistently perceived by artists. We did this by asking two artists to draw orthogonal views to the given sketch as they imagined it using any construction lines or other drawing aids. While one artist commented that drawing orthogonal to a non-canonical viewpoint was somewhat cumbersome, both produced meaningful sketches that qualitatively matched the input 3D data in orthogonal views (Figure 11, last two columns). We also asked a trained modeler to recreate 3D models of the perceived shapes from the input sketches, using them as a visual reference in Autodesk Maya (middle column).

We note that qualitatively, ground truth, 3D modeler output, user sketches and our algorithmic output, are visually similar in the views orthogonal to the input sketch (Figure 11). Our results for the spiral and revolve are near identical to ground truth and the wrinkle, saddle and cross have marginally less foreshortening than ground truth, which aligns with the human tendency to overcompensate for foreshortening in drawings [Schmidt et al. 2009a]. Finally, we asked the 3D modeler to view the ground truth data and our algorithmic result alongside his creations interactively in 3D and comment on them. Summarily, he felt that all three sets of output captured the essence of the shape intended by the input sketch and would make worthy 3D mock-ups for conceptual design.

7 Results

We evaluated our method on a variety of sketches, ranging from simple abstract shapes (Figure 12) to elaborate drawings of complex design objects such as airplanes, cars or vacuum cleaners (Figures 1 and 21). We include both fully drawn models such as the guitar and camera (Figure 16), and models where users exploit symmetry to draw only half the shape (Figure 1). Several inputs (gamepad, pencil-holder, tape-holder, video demo) were sketched on blank canvas by Xch'e' Hernandez Simper, an industrial design student. While he experienced a stiff learning curve to adopt the system - largely due to lack of UI features such as undo, scale/translate/rotate, and other shortcuts, he was able to create a range of models once these were overcome. Based on the experience he believes that following some UI standardization the system will be widely and successfully used by professionals. Many inputs were traced over real-world freeform sketches by the authors from design books and websites (mixer, camera, iron, airplane, racecar, sewing machine, toothpaste, pencil-sharpener, vacuum cleaners) with minor edits. See pencil-sharpener and toothpaste in Figures 3, 14, 21 as well as Figure 16. These inputs are representative of the irregularity and inaccuracy encountered in real sketches. One input (guitar) was traced over a photograph, augmented by a few descriptive curves. While the feature curves are accurate projections of 3D geometry with unknown perspective, the added curves are imagined and approximate. Several inputs were traced over the projection of ILoveSketch and 3DScaffolds curve networks (car, fighter, coffee machine). These curves were designed by multiview systems anterior to ours, and were not aimed to specifically follow our conventions. Many of the sketches were traced with Illustrator in a few minutes (toothpaste) to an hour (camera). The only inputs free from sketch and projection inaccuracies are the 5 ground-truth 3D models in Figure 11.

We surfaced the reconstructed curve networks using the methods of [Bessmeltsev et al. 2012; Zhuang et al. 2013], to aid 3D visualization. All our outputs visually conform to user expectations. Our

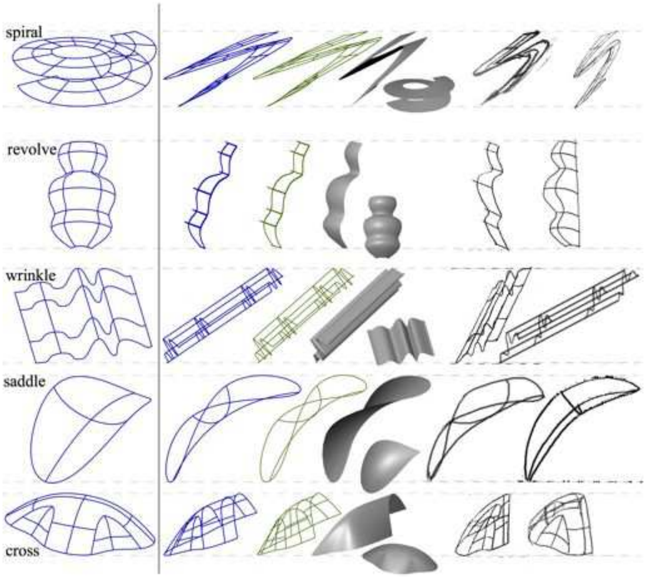


Figure 11: *The curves in the extreme left column (ground truth) are shown to artists as the input sketch. The other columns show a view orthogonal to the sketch to illustrate perceived depth. From left to right we show ground truth curves, our algorithmic output (green), artist modeled 3D surfaces and alternate view curves sketched by two artists.*

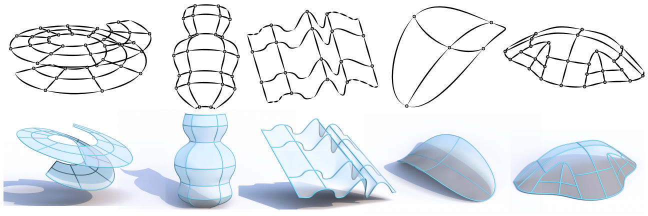


Figure 12: *Alternate view visualization of our results.*

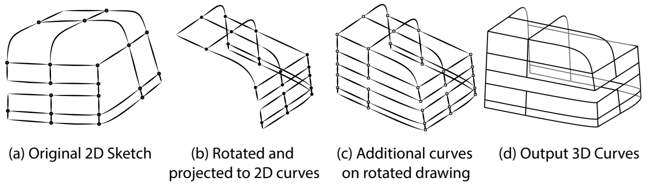

(a) Original 2D Sketch
(b) Rotated and projected to 2D curves
(c) Additional curves on rotated drawing
(d) Output 3D Curves

Figure 13: *Multi-view modeling: an initial sketch (a) results in a partial 3D model (b), which can be rotated to draw previously occluded parts(c) , and recompute the final 3D model (d).*

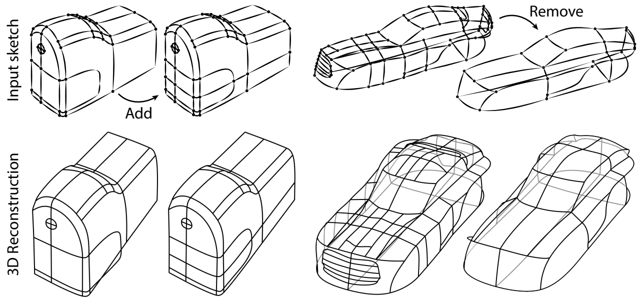


Figure 14: *Sketch refinement/coarsening: The curves traced over the pencil-sharpener sketch from Figure 3 lack sufficient connective context to correctly reconstruct the complex non-planar curve. We can fix this by adding an intersecting curve (a). An ILoveSketch curve network can be dramatically simplified by removing curves, while preserving the overall reconstructed shape (b).*

algorithm provides rapid feedback, with reconstruction times ranging from around 3 sec. for small models like the toothpaste to just under 30 sec. for the most complex ones like the camera.

Most of our results were produced using a single-view interface, mimicking a traditional pen-on-paper workflow. Users can thus ideate by drawing directly within our system or simply trace over napkin sketches or inspirational photographs (Figures 1, 16). Figure 13 shows our reconstruction algorithm used within a multi-view interface, where users can first draw part of the model in one view, rotate the reconstructed result and continue to draw additional details. Users can pick either drawing workflow to meet their needs.

Robustness to Input Edits. Figure 14 shows the impact of adding and removing sketched curves on our 3D output. Adding a single curve to the pencil sharpener constrains the non-planar curve to better align with the rest of the network. Removing many input curves from a car traced over a model from [Bae et al. 2008] reduces sketch clutter but retains the overall structure of the sketched vehicle.

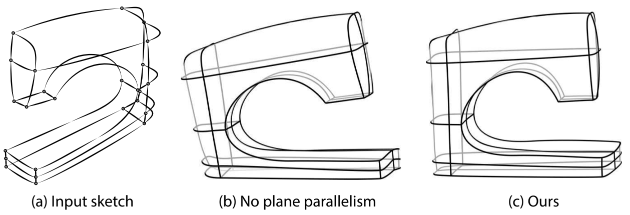


Figure 17: *Enforcing plane orthogonality and parallelism effectively undoes inexact perspective distortions in sketch input, restoring the parallel horizontal planes on this sewing machine.*

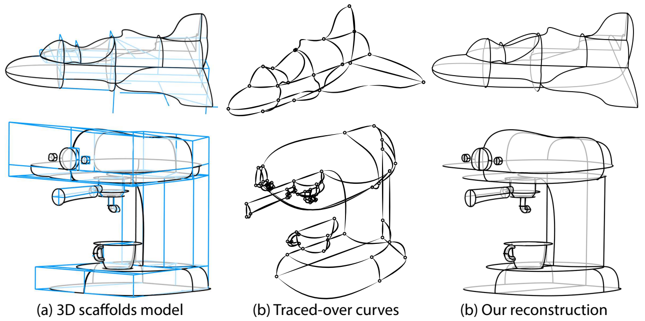


Figure 18: *Comparison with 3D scaffolds [Schmidt et al. 2009b] (a). Input curves traced over a projection of a curve network produced with the scaffold (b). Our qualitatively similar results generated without the need for scaffolding (c).*

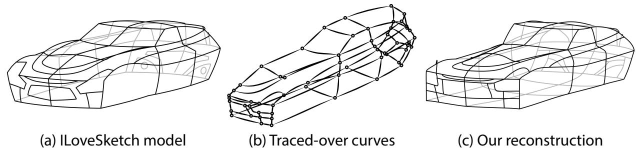


Figure 19: *Comparison with multi-view framework [Bae et al. 2008] (a). Input curves traced over a projection of a curve network produced with ILoveSketch (b). We achieve qualitatively similar results from a single-view sketch (c).*

Impact of Design Choices. Figure 15 demonstrates the importance of our design choices in producing plausible 3D reconstructions. Minimizing foreshortening while ignoring 2D-to-3D variation (Figure 15 (b, g)) leads to disastrous results. Minimizing variation but ignoring foreshortening produces a more plausible but foreshortened solution (Figure 15(d)) while appropriately accounting for both produces a result well aligned with viewer perception (Figure 15 (e, h)). Figure 15 (i, j) shows the importance of tangent parallelism in sketch interpretation. Figure 17 illustrates the impact of inter-plane orthogonality and parallelism in undoing distortions in sketches with large and inexact perspective.

Figure 5 and the angle histograms in Figures 1, 21, show the importance of progressive regularity detection and enforcement in propagating sketch regularity across the curve network (often regularities with small absolute error in the baseline reconstruction end up as inapplicable, while others with higher error are eventually enforced).

Comparison to Prior Art. Figures 18 and 19 highlight our ability to generate models of equal complexity to those generated by multi-view, e.g. [Bae et al. 2008], and scaffold-based [Schmidt et al. 2009b] methods, without the need for view rotations or meticulous incremental scaffolding. We performed this comparison by picking informative views for 3D outputs of these prior systems, and tracing 2D curves over them as our sketch input.

Methods such as [Tian et al. 2009; Wang et al. 2009] which target reconstruction of boxy CAD shapes dominated by straight-lines and orthogonal features, as the authors acknowledge, cannot handle design sketches of smooth free-form shapes (e.g. plane, vacuums, game-pad) addressed by True2Form.

Our method is closer in spirit to CrossShade [Shao et al. 2012] but is a significant advance, as illustrated by Figure 20. Our framework is more generic, as Shao et al. recover 3D geometry only for planar and orthogonal cross-section curves. On the vacuum example, only the highlighted (red) subset of curves satisfies these requirements; it

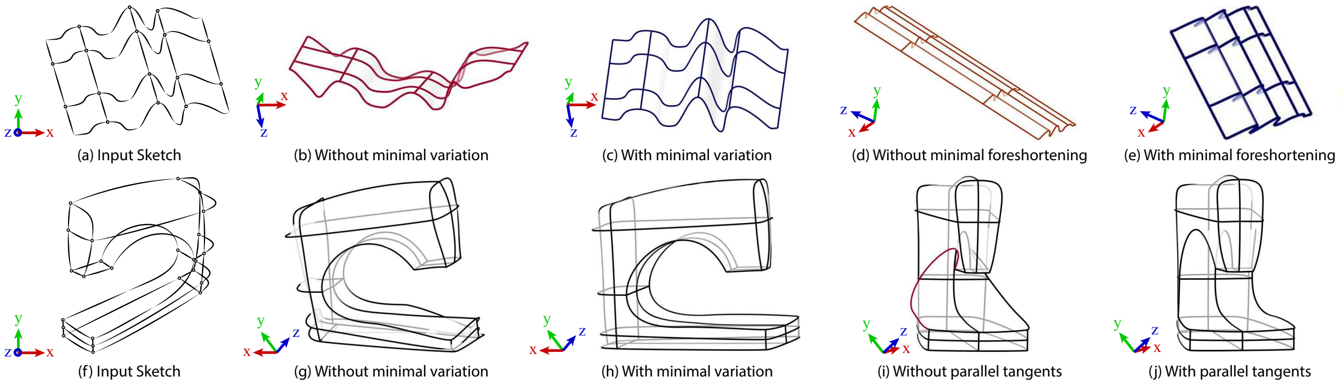

(a) Input Sketch
(b) Without minimal variation
(c) With minimal variation
(d) Without minimal foreshortening
(e) With minimal foreshortening
(f) Input Sketch
(g) Without minimal variation
(h) With minimal variation
(i) Without parallel tangents
(j) With parallel tangents

Figure 15: *Evaluating design choices. (Top) reconstruction without and with variation minimization and foreshortening minimization on a simple input. (Bottom) results without and with variation minimization and tangent parallelism.*

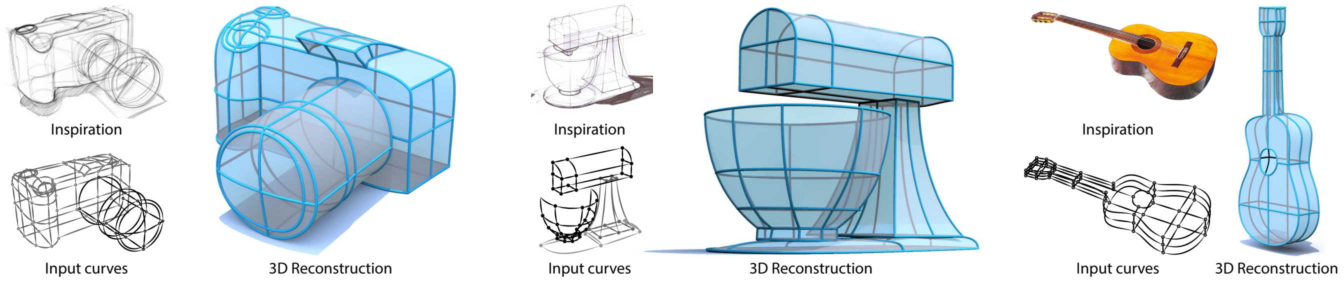


Figure 16: *Our single-view modeling system allows us to reconstruct 3D models by tracing curves over existing sketches and photographs. Camera sketch by Spencer Nugent©, Mixer sketch by Koos Eissen and Roselien Steur©, curves traced by the authors.*

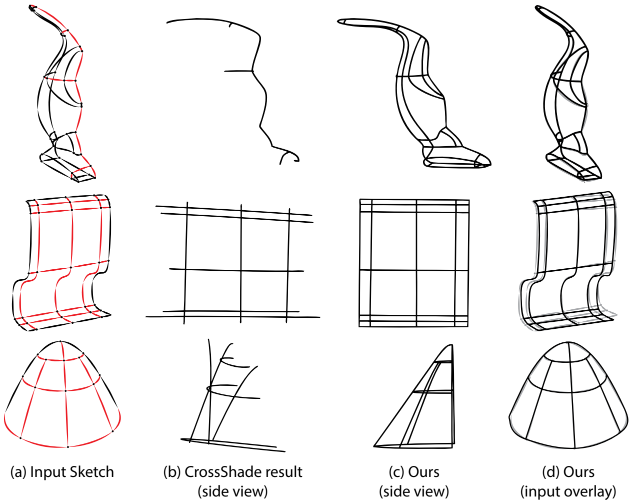

(a) Input Sketch
(b) CrossShade result (side view)
(c) Ours (side view)
(d) Ours (input overlay)

Figure 20: *Comparison with CrossShade [Shao et al. 2012]: CrossShade only reconstructs planar cross-section curves, the subset of sketch curves in red. Curves are forced to align perfectly with the sketch, which is undesirable in the presence of innacuracy and perspective distortion (middle row). Finally, it encourages local symmetry at all intersections, disconnecting the network in the presence of non-geodesic curves such as the horizontal sections of the rounded cone (bottom row).*

alone does not provide enough information to intelligently recover the full vacuum shape. In general, their method is unable to handle models like the vacuum or spiral (Figure 12), which cannot be described via planar curves alone. The curved surface patch (Figure 20, middle) shows the importance of allowing the projection of the shape to deviate from the input sketch in order to correct for artist inaccuracy. This correction is undesirable in the context of CrossShade who reconstruct normal maps but is important for believable 3D reconstruction. Lastly, the rounded cone (Figure 20 (bottom)) shows the drawbacks of using the approach of Shao et al, even when all curves satisfy their constraints, as their approach of softly enforcing symmetry everywhere at the expense of connectivity can break the model apart. Enforcing connectivity while softly minimizing regularity everywhere is similarly undesirable (Figure 4). Our selective enforcement produces the desired 3D output.

8 Conclusions

Designers leverage descriptive curves to effectively convey 3D information in 2D drawings. The implicit perceptual cues in these sketched curve networks aid viewers in inferring free-form piecewise-smooth shapes from line-drawings. We mimic the results of this 3D shape inference by: the formulation of a set of regularity properties; the progressive detection of regularity cues in an input sketch, that is consistent with human perception; and the enforcement of these applicable cues, to lift a sketch curve network into 3D, while preserving fidelity to the input sketch. We are able to reconstruct convincing 3D shapes from a single sketch, correcting for inaccuracy and achieving geometric complexity well beyond prior art.

Despite its benefits, our method has a few limitations. One, our approach relies on drawing from informative viewpoints. It is possible to trick our algorithm into misinterpreting a sketch, just as it is possible to trick humans with *trompe- l'oeil* drawings. Sometimes for complex models finding a single descriptive view that is comfortable for drawing is challenging. Using an incremental multi-view

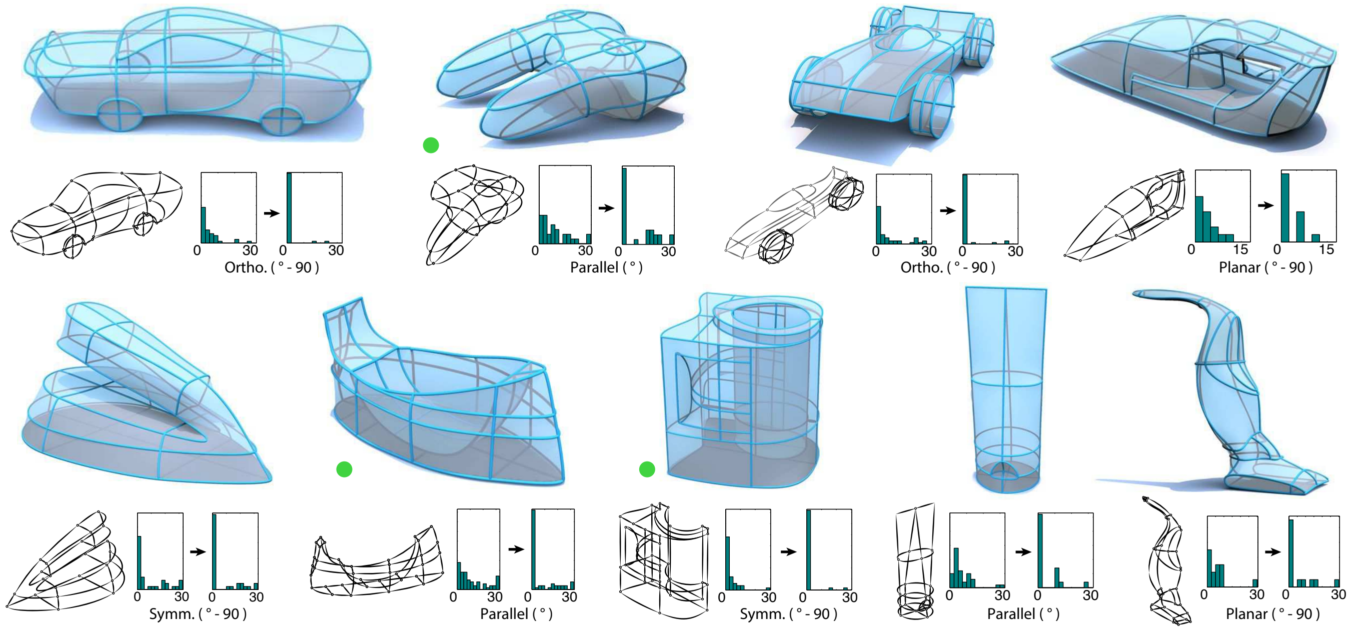


Figure 21: *Additional results with representative regularizer value histograms on baseline and final models, highlighting the importance of the the iterative labeling in avoiding false positives and negatives. Gamepad, Tape Holder, Pencil Holder by Xch'e' Hernandez Simper©(green dot), via our interface. Other models were traced by the authors.*

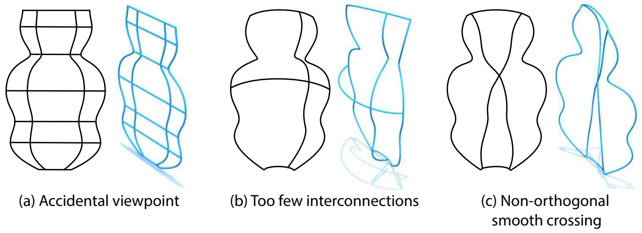

(a) Accidental viewpoint
(b) Too few interconnections
(c) Non-orthogonal smooth crossing

Figure 22: *Our algorithm fails to reconstruct meaningful curve networks when curves are drawn from accidental viewpoints (a), when the network does not contain enough connectivity (b), or when the curves violate principles of design sketching (c).*

approach (Figure 13) or exploiting symmetry planes (Figure 1), can help with this problem. Two, as our approach is purely geometric, we require sufficiently descriptive and interconnected sketches to produce desirable 3D output. Shape inference from sparse sketches in particular, can benefit from prior knowledge of the 3D objects, just as human interpretation of sketches draws upon their visual experience. Three, our set of sketch properties in Section 3 is not exhaustive and can be extended to incorporate properties like equal length curves or geometric primitives like circles.

Figure 22 spans the range of failure cases we have observed. Drawing many curves from strongly foreshortened or accidental viewpoints (Figure 22 (a)) results in poor reconstruction. Fixing this failure case requires redrawing the model from a better view. Fortunately, designers are taught to pick what we consider good views [Eissen and Steur 2011]. Drawing too few interconnected curves (Figure 22 (b)) prevents our algorithm from utilizing context well. Such cases are easily fixed by adding curves, particularly smoothly-crossing curves. Lastly, our algorithm is based on principles of sketch understanding which if violated can produce undesirable results (Figure 22 (c)). Such cases require that the inconsistent curves be removed or redrawn.

In summary, True2Form is arguably the first approach able to infer plausible piecewise-smooth 3D shapes from general design sketches. We believe that such approaches in the future, will play an important role in the early stages of conceptual design.

Acknowledgments

Many thanks to George Drettakis, David Bommes and Christian Richardt for feedback on drafts of the paper. We thank Koos Eissen, Roselien Steur and Spencer Nugent for allowing us to use their concept sketches. Many thanks to Xch'e' Hernandez Simper, and Cloud Shao for generating input drawings. Inria acknowledges generous support from Adobe. The work of the Canadian authors was supported by NSERC and GRAND NCE. This work was also partially funded by ANR-12-JS02-003-01 DRAO.

References

ANDRE, A., AND SAITO, S. 2011. Single-view sketch based modeling. In *Proc. Sketch-Based Interfaces and Modeling*.

BAE, S., BALAKRISHNAN, R., AND SINGH, K. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proc. User Interface Software and Technology*.

BESSMELTSEV, M., WANG, C., SHEFFER, A., AND SINGH, K. 2012. Design-driven quadrangulation of closed 3d curves. *ACM Trans. Graph. 31*, 5.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3, 77:1–77:10.

BORDEGONI, M., AND RIZZI, C. 2011. *Innovation in Product Design: From CAD to Virtual Prototyping*. Springer.

Chen, T., Zhu, Z., Shamir, A., Hu, S.-M., and Cohen-Or, D. 2013. 3-sweep: Extracting editable objects from a single photo. *ACM Trans. Graphics 32*, 6.

Cooper, M. 2008. *Line Drawing Interpretation*. Springer.

Eissen, K., and Steur, R. 2011. *Sketching: The Basics*. Bis Publishers.

Gingold, Y., Igarashi, T., and Zorin, D. 2009. Structured annotations for 2D-to-3D modeling. *ACM Trans. Graph. 28*, 5.

Igarashi, T., Matsuoka, S., and Tanaka, H. 1999. Teddy: a sketching interface for 3D freeform design. *Proc. SIGGRAPH*.

Kara, L. B., and Shimada, K. 2007. Sketch-based 3d-shape creation for industrial styling design. *IEEE Comput. Graph. Appl. 27*, 1, 60–71.

Knill, D. C. 1992. Perception of surface contours and surface shape: from computation to psychophysics. *Journal of Optical Society of America 9*, 9, 1449–1464.

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

Lau, M., Saul, G., Mitani, J., and Igarashi, T. 2010. Modeling-in-context: user design of complementary objects with a single photo. In *Proc. Sketch-Based Interfaces and Modeling*, 17–24.

Lee, S., Feng, D., and Gooch, B. 2008. Automatic construction of 3d models from architectural line drawings. In *Proc. Interactive 3D graphics & games*, 123–130.

Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J. 2011. GlobFit: Consistently fitting primitives by discovering global relations. *ACM Trans. Graph. 30*, 4.

Lipson, H., and Shpitalni, M. 1996. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design 28*, 651–663.

Lowe, D. G. 1987. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell. 31*, 3, 355–395.

Malik, J. 1987. Interpreting line drawings of curved objects. *International Journal of Computer Vision 1*, 1, 73–103.

Mamassian, P., and Landy, M. S. 1998. Observer biases in the 3D interpretation of line drawings. *Vision research 38*, 18, 2817–2832.

Mather, G. 2008. *Foundations of sensation and perception*. Taylor and Francis.

McCrae, J., Singh, K., and Mitra, N. 2011. Slices:a shape-proxy based on planar sections. *ACM Trans. Graph. 30*, 6.

Nakayama, K., and Shimojo, S. 1992. Experiencing and Perceiving Visual Surfaces. *Science 257*, 1357–1363.

Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. 2007. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. 26*.

Olsen, L., Samavati, F., Sousa, M., and Jorge, J. 2009. Sketch-based modeling: A survey. *Computers & Graphics 33*.

Olsen, L., Samavati, F., and Jorge, J. A. 2011. Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31*, 6, 24–34.

Orbay, G., and Kara, L. B. 2012. Sketch-based surface design using malleable curve networks. *Computers & Graphics 36*, 8, 916–929.

Perkins, D. 1971. Cubic corners, oblique views of pictures, the perception of line drawings of simple space forms. geometry and the perception of pictures: Three studies. Tech. rep., Harvard Univ., Cambridge, MA. Graduate School of Education.

Pipes, A. 2007. *Drawing for Designers*. Laurence King.

Pizlo, Z., and Stevenson, A. 1999. Shape constancy from novel views. *Perception & Psychophysics 61*, 7, 1299–1307.

Schmidt, R., Khan, A., Kurtenbach, G., and Singh, K. 2009. On expert performance in 3D curve-drawing tasks. In *Proc. Sketch-Based Interfaces and Modeling*.

Schmidt, R., Khan, A., Singh, K., and Kurtenbach, G. 2009. Analytic drawing of 3d scaffolds. *ACM Trans. Graph. 28*, 5.

Shao, C., Bousseau, A., Sheffer, A., and Singh, K. 2012. Crossshade: Shading concept sketches using cross-section curves. *ACM Trans. Graphics 31*, 4.

Sharf, A., Alcantara, D. A., Lewiner, T., Greif, C., Sheffer, A., Amenta, N., and Cohen-Or, D. 2008. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph. 27*, 5, 110:1–110:10.

Shtof, A., Agathos, A., Gingold, Y., Shamir, A., and Cohen-Or, D. 2013. Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum 32*, 2, 245–253.

Stevens, K. A. 1981. The visual interpretation of surface contours. *Artificial Intelligence 17*.

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 Trans. Graphics 33*.

Tian, C., Masry, M., and Lipson, H. 2009. Physical sketching: Reconstruction and analysis of 3D objects from freehand sketches. *Computer Aided Design 41*, 3, 147–158.

Wang, Y., Chen, Y., Liu, J., and Tang, X. 2009. 3D reconstruction of curved objects from single 2D line drawings. *IEEE Computer Vision and Pattern Recognition 0*, 1834–1841.

Zhuang, Y., Zou, M., Carr, N., and Ju, T. 2013. A general and efficient method for finding cycles in 3d curve networks. *ACM Trans. Graph. 32*, 6.