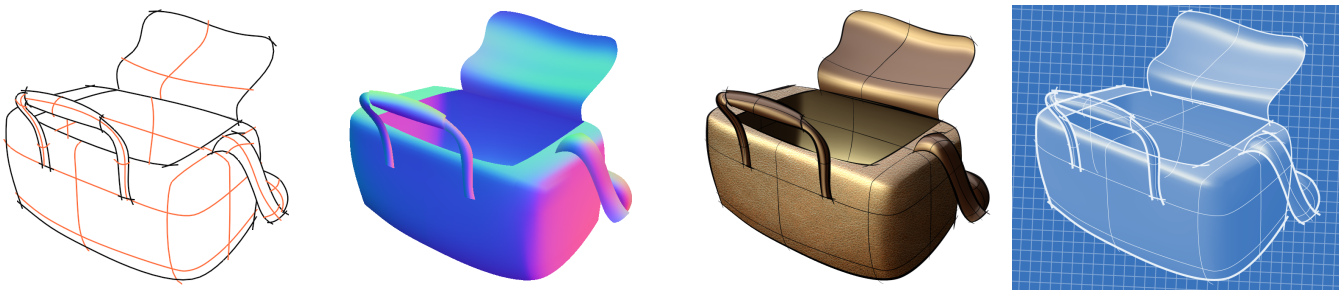
CrossShade: Shading Concept Sketches Using Cross-Section Curves

Cloud Shao, Adrien Bousseau, Alla Sheffer, Karan Singh

CrossShade: Shading Concept Sketches Using Cross-Section Curves

Cloud Shao[1*] Adrien Bousseau[2*] Alla Sheffer[3] Karan Singh[1]
[1] University of Toronto [2] REVES - INRIA Sophia Antipolis [3] University of British Columbia



(a) Input curves (b) Estimated normals (c) Shading

Figure 1: *Concept sketches (a) frequently use cross-sections (drawn in orange) to successfully convey 3D shape with just a handful of lines. We derive the mathematical properties of cross-section curves and leverage them to automatically estimate surface normals across the drawn objects (b). The resulting normal field allow users to shade the objects using a variety of shading styles and setups (c).*

Abstract

We facilitate the creation of 3D-looking shaded production drawings from concept sketches. The key to our approach is a class of commonly used construction curves known as *cross-sections*, that function as an aid to both sketch creation and viewer understanding of the depicted 3D shape. In particular, intersections of these curves, or *cross-hairs*, convey valuable 3D information, that viewers compose into a mental model of the overall sketch. We use the artist-drawn cross-sections to automatically infer the 3D normals across the sketch, enabling 3D-like rendering.

The technical contribution of our work is twofold. First, we distill artistic guidelines for drawing cross-sections and insights from perception literature to introduce an explicit mathematical formulation of the relationships between cross-section curves and the geometry they aim to convey. We then use these relationships to develop an algorithm for estimating a normal field from cross-section curve networks and other curves present in concept sketches. We validate our formulation and algorithm through a user study and a ground truth normal comparison. As demonstrated by the examples throughout the paper, these contributions enable us to shade a wide range of concept sketches with a variety of rendering styles.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—;

Keywords: sketching, product design, sketch-based modeling, non-photorealistic rendering, cross-sections

Links: DL PDF

*Cloud Shao and Adrien Bousseau are joint first authors

1 Introduction

Designers use line drawings, or *concept sketches*, to quickly convey a mental 3D model during the early phases of the design process [Pipes 2007]. As illustrated in Figures 1(a) and 2, effective drawings frequently rely on *cross-sections* and other construction lines as an aid to both the sketching process and the mental reconstruction of a 3D form [Robertson 2003; Eissen and Steur 2008]. While concept sketches facilitate viewer understanding of 3D shapes, they require a mental leap to imagine the appearance of the drawn object, imbued with specific material properties and lighting. Designers usually convey a more comprehensive *3D-look* by artistically shading these sketches (Figure 2 d). This shaded imagery, often referred to as *production drawings*, is the traditional mode of communicating 3D concepts between designers and their patrons. However, production drawings require significantly more time, effort and expertise to create than the concept sketches on which they are overlaid [Pipes 2007; Eissen and Steur 2008]. They are also often laboriously redrawn for different shading parameters, such as changes in colors, materials or lighting conditions. While modeling the objects in 3D can substitute the manual shading process, for a variety of reasons artists avoid 3D tools at early stages of the design process [Pipes 2007; Cook 2008; Bae et al. 2008], as explained in Section 2.

Our work bridges the gap between concept sketches and production drawings by facilitating 3D-like shading with user-desired rendering styles and parameters (Figure 1). We target sketches based on cross-sections and use the drawn curves to successfully estimate the surface normals necessary to convey a 3D-look. Designers can use our system to draw curves from scratch or over existing sketches. This approach integrates seamlessly with the traditional sketching workflow since cross-sections are a standard artistic tool for communicating shape and can be added at will to a sketch.

Cross-section curves depict intersections of the imagined 3D surfaces with 3D planes [Eissen and Steur 2008]. These curves, and especially their intersections or *cross-hairs*, carry important perceptual shape information [Stevens 1981; Knill 1992; Mamassian and Landy 1998]. Consequently, designers draw cross-sections at places where they maximize sketch clarity (Figure 2). Despite their ubiquity there is no consensus in the design or perception literature on where cross-sections are drawn or what exactly they aim to convey. To answer these questions we compile insights from design

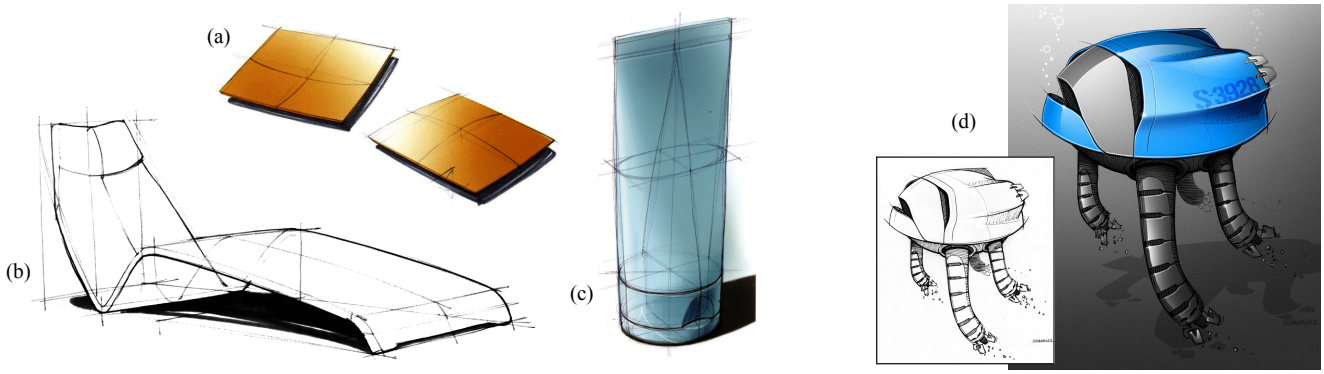


Figure 2: *Cross-section curves are commonly used by designers to depict shape variations. Cross-sections emphasize curvature (a) and symmetry (b), © [Eissen and Steur 2008]. Designers explore ideas quickly with concept sketches, some of which are refined and shaded for presentation (d), © Mike Serafin www.memikeserafin.com.*

books, sketching tutorials, and perception literature and use those to formulate the connections between the drawn 2D cross-sections and the 3D geometry they convey (Section 3). Specifically, we note that cross-hairs typically represent curves that lie in orthogonal planes and that individual cross-sections simultaneously suggest lines of principal curvature and geodesics. The interconnectivity of cross-section curves enables viewers to resolve conflicts between local cues to arrive at a consistent global interpretation.

We restate these connections in mathematical terms to infer the 3D shape of the cross-section curves from the sketch (Section 4). The extracted 3D cross-sections define 3D surface normals at cross-hairs, which we then propagate along and in-between curves using a hybrid lofting approach which aims to align the surface curvature lines with the cross-sections. We validate our observations and algorithm by comparison of surface normals to those perceived by humans, and to ground truth data (Section 5).

The extracted normal field provides for new rendering capabilities, allowing users to dynamically visualize the effects of different materials, lighting conditions and rendering styles over the depicted shape (Section 6).

In summary, we present the first approach to facilitate 3D-like production shading of 2D cross-section based concept sketches. From a technical standpoint, this paper introduces two contributions:

- An explicit mathematical formulation of the relationship between sketched cross-section curves and the 3D geometry they aim to convey.
- An algorithm for extracting a normal field from cross-section curve networks based on the formulation above.

While our goal is not to reconstruct an actual 3D model, our estimated surface normals have the potential to accelerate the 3D modeling phase of the design process. We also note that our work is most relevant in contexts where cross-sections are traditionally used, such as production design of man-made objects, but may also apply in other setups such as cartoon rendering.

2 Related Work

Production Design Workflow. Concept sketches are widely used in the early stages of product design to quickly illustrate concepts and are then refined and shaded for presentation to the client [Pipes 2007; Eissen and Steur 2008]. In current practice, designers have access to the alternate workflow of transforming concept sketches into a complete 3D model, which can subsequently be shaded. While an approved and finalized concept will indeed undergo this transformation for use in downstream manufacturing processes, this is undesirable for early conceptual design for various reasons. Direct sketching allows designers to express ideas faster, with far fewer preliminary steps and distractions than 3D modeling tools [Pipes 2007; Bae et al. 2008]. In addition, 3D modeling from sketches is time-consuming, regardless of whether it is undertaken manually using sketches as a reference, or using 3D curve sketching tools and then manually surfacing the collection of curves. Much of this 3D modeling effort is also wasted, given the frequency of iteration in the early design stages [Cook 2008].

Colorizing and Shading Sketches. Designers commonly use painting tools such as Alias SketchBook Pro© and Adobe Photoshop© to shade their sketches. More advanced colorization tools propagate scribbles to accelerate the definition of constant color regions in the drawing [Sýkora et al. 2009]. Recent methods support smooth color variations using a few vector primitives [Orzan et al. 2008; Finch et al. 2011]. Despite these advances, painting convincing 3D-like shading and highlights still requires a great amount of artistic skill and time. Our method goes one step further by deriving surface normals directly from the sketch, allowing users to edit high-level scene properties such as material appearance and lighting instead of color regions.

Sketching 3D models. Although our goal is not to reconstruct a complete 3D model, our work is related to sketch-based modeling methods, reviewed by [Olsen et al. 2009]. Most such methods, e.g. [Igarashi et al. 1999; Nealen et al. 2007; Bae et al. 2008], require users to sketch from multiple views to create an unambiguous 3D shape. Our approach is more related to methods that extract 3D information from a single sketch. Schmidt et al. [2009b] use *scaffolds* to support single and multi-view sketching of 3D curves. Our approach builds on similar ideas but exploits a different family of construction lines. In addition, we complement the 3D curves with normals across the drawn objects. Gingold et al. [2009] use interactive annotations over a pre-defined set of geometric primitives to facilitate 3D modeling from a single sketch. Our method is not restricted to specific shapes and requires little interaction beyond sketching and labelling cross-sections. Andre et al. [2007] use a grid of orthogonal geodesics to generate relief-like 3D models, while Andre and Saito [2011] allow users to model objects composed of ellipsoids by drawing their two major orthogonal cross-sections. We draw inspiration from both papers, incorporating some of their observations on perception of intersecting contours into our formulation of cross-section properties (Section 3). We further refine them to account for a much more general set of shapes.

Sketching normal and height fields. Our approach is motivated by the work of Johnston [2002] and Winnemöller et al. [2009] that

generate normal maps to shade cartoon images by diffusing silhouette normals inside the drawing. In a related work, Joshi and Carr [2008] propose a method to inflate a thin-plate spline surface from boundary curves. The underlying assumption of these methods is that the normal or height map in the interior is a convex combination of the contour and silhouette normals or height values, which allows for shading a variety of blobby shapes but is rarely true for product sketches. Our method is especially well suited for this task, allowing for unlimited local normal control through cross-section specification.

Researchers proposed several indirect interfaces for specifying normal maps [Zhang et al. 2002; Okabe et al. 2006; Wu et al. 2007]. Instead our method is embedded in the traditional sketching workflow using curves that are commonly drawn by designers, without requiring additional indications apart from curve labeling. All of these methods propagate the known normals using a distance-based convex combination approach. We propose instead a lofting scheme based on Coons interpolation that better preserves the curvature directions provided by the cross-section curves (Figure 5).

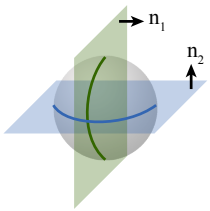
Sketch and Contour Interpretation. Automatic line drawing interpretation aim to reconstruct shapes from silhouette and other feature lines [Cooper 2008], a challenging and often ill-posed task [Malik 1987; Lipson and Shpitalni 1996]. Ecker et al. [2007] address a related problem of depth estimation from planar curves formed by projecting arbitrary lines over a scene. Instead we address the potentially easier and better defined task of deriving 3D normals from artist-drawn cross-section curves, which *a priori* are positioned to best convey shape. Since there are no clear guidelines on where to draw cross-sections, we attempt to answer this question ourselves. Our work therefore shares many similarities with [Cole et al. 2009], where the authors investigate the location of feature lines in line drawings.

There is also a significant body of perception research on how viewers interpret drawn lines, and especially intersecting curves. We discuss these works in detail in the next section.

3 Understanding Sketched Cross-Sections

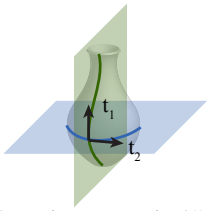
While a sparse set of arbitrary cross-sections is too ambiguous to represent 3D shape [Ecker et al. 2007], strategically placed cross-sections are widely used by designers to successfully convey 3D surfaces. Unfortunately, design literature provides no single, definitive answer as to where cross-sections should be drawn, what they should represent, and how they are perceived by viewers.

To address these questions we investigated design practices via conversations with designers, video-taped sketching sessions, and reviewed a collection of imagery and design texts [Eissen and Steur 2008; Robertson 2003]. Viewer perception must, consciously or not, influence designers who ultimately aim to maximize the sketch clarity. Consequently, we combine our observations with findings from perception literature. We use all of these to propose five mathematical properties of designer-drawn cross-sections.



1. Orthogonal Cross-Hair Planes: Recommendations in sketching literature [Eissen and Steur 2008], plane-based 3D shape abstraction [McCrae et al. 2011], and perception studies [Stevens 1981], indicate that designers consistently use orthogonal planes for intersecting cross-sections. Mathematically, this relationship can be stated as requiring the plane normals $\mathbf{n}_1$ and $\mathbf{n}_2$ to satisfy

$$\mathbf{n}_1 \cdot \mathbf{n}_2 = 0. \tag{1}$$



2. Cross-Sections as Curvature Lines: Designer conversations, imagery and literature consistently communicate curvature and local reflective symmetries at cross-hairs: "Cross-sections on a surface explain or emphasize its curvature", "...bend or transform the object's surface.", "...are added to clarify the intended shape and keep it symmetrical." [Eissen and Steur 2008]. This is verified by perception studies which indicate that observers interpret intersecting curves as aligned with the principal lines of curvature [Stevens 1981; Mamassian and Landy 1998]. Locally this implies that the tangents to the cross-section curves at cross-hairs are orthogonal:

$$\mathbf{t}_1 \cdot \mathbf{t}_2 = 0. \tag{2}$$

The more global implication of this observation is that the curvature tensor on the mentally reconstructed 3D geometry around the cross-hairs is aligned with the cross-section curves. Note that, since curvature lines on surfaces are not necessarily planar [do Carmo 1976], cross-section curves may deviate from curvature lines away from cross-hairs. While one can draw cross-sections that do not satisfy plane or tangent orthogonality, see Figure 3, viewers are likely to interpret those as representing a different geometry that conforms to the two criteria.

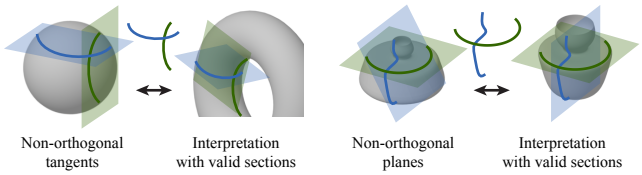
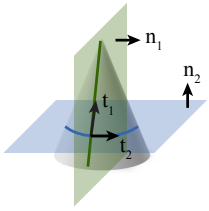


Figure 3: *Cross-sections that violate tangent or plane orthogonality yield ambiguous drawings which viewers tend to interpret as having an alternative interpretation where both are orthogonal.*

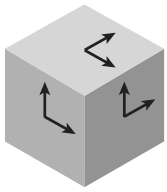
3. Cross-sections as Local Geodesics: A surface curve is a geodesic if it defines the shortest distance between two points. If a planar curve is a geodesic then the surface normal along it lies in the 3D plane of that curve [do Carmo 1976]. Perceptual studies [Knill 1992] indicate that, without cues to the contrary, humans perceive intersecting cross-section curves as geodesics. To express this property we first prove that given the plane and tangent orthogonality constraints, the normal to the supporting plane of a geodesic cross-section curve is co-linear with the tangent of the intersecting curve at a cross-hair. The surface normal at a cross-hair $\mathbf{n}$ is orthogonal to $\mathbf{t}_1$ and $\mathbf{t}_2$. If cross-section i is a geodesic then $\mathbf{n}$ is also orthogonal to $\mathbf{n}_i$. However we cannot have four mutually orthogonal vectors in R^3, thus $\mathbf{n}_i$ must be collinear with $\mathbf{t}_j$. When both curves are geodesics we have:

$$\mathbf{t}_1 \times \mathbf{n}_2 = 0, \mathbf{t}_2 \times \mathbf{n}_1 = 0. \tag{3}$$

Under this assumption, the degrees of freedom for both tangents and normals diminish radically [Stevens 1981; Andre et al. 2007].



While cross-hair plane and tangent orthogonality constraints together force one of the curves to be a surface geodesic (see Appendix), one can easily find cross-section sketches where the second curve is not a geodesic, such as the circular cross-section of a cone (see inset). Despite this, we found a clear tendency by viewers to perceive normals which were as consistent as possible with the geodesic interpretation subject to other visual cues (see Section 5).

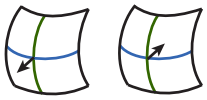


4. Minimal Foreshortening: When drawing a shape, designers favor *informative viewpoints* that convey most visible surfaces with minimal foreshortening [Eissen and Steur 2011]. Such viewpoints were shown to improve *sketchability* by Bae et al. [2008]. Isometric and dimetric projections are also commonly used in engineering drawings to avoid excessive foreshortening (see inset).

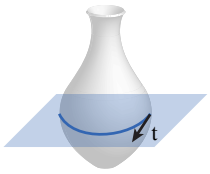
A single cross-hair is minimaly forshortened when both tangents lie in the viewplane. Using a coordinate frame where z is orthogonal to the view plane, one can capture foreshortening minimization over a cross-hair network by minimizing

$$\sum (\mathbf{t}_1^z)^2 + (\mathbf{t}_2^z)^2 \tag{4}$$

over all intersections. The Appendix shows that for a single cross-hair this condition is equivalent to the normal alignment preference observed by Stevens [1981]. Thus even though his study does not explicitly address foreshortening, it reinforces the minimal foreshortening hypothesis. Like the geodesic criterion, this is a weak cue, more important when the local surface is near perpendicular to the viewplane, but one that viewers correctly relax in the presence of contradictory cues (see the ellipsoid and inset (c) in Figure 9).



5. Orientation: The perception of shape from intersecting cross-sections is subject to an inherent ambiguity between opposite normal orientations. The inset Figure illustrates how the same cross-hair can be interpreted as a convex surface with normal pointing down or as a concave surface with normal pointing up. Studies indicate that on average viewers prefer interpretations consistent with a *view from above*, and with convex rather than concave shape [Mamassian and Landy 1998]. Still, since individual interpretations vary, orientation choice cannot be fully automated in general.



However, we observe that orientation can be determined uniquely when cross-sections extend to a *smooth silhouette* of the drawn object. A smooth silhouette in our setup is defined as a line where the normal to a smooth surface is perpendicular to the view direction. By definition, the surface in the immediate vicinity of a smooth silhouette must be closer to the viewer than the silhouette itself (see inset). For a cross-section touching a silhouette, the inward pointing tangent at ϵ-offset from the intersection must have a positive depth, uniquely determining the orientation. In most concept sketches, the drawn silhouette cross-section intersections are sufficient for both viewers and our algorithm to completely resolve surface normal orientation.

Summary: Our analysis reveals five properties of cross-section curves relating 3D cross-section planes and tangents at cross-hairs, as well as a way to disambiguate the orientation of cross-sections that intersect smooth silhouettes. We validate these properties through a user study in Section 5.

4 Geometry Estimation Algorithm

4.1 Overview

We leverage the properties described above to estimate the normal field across objects in concept sketches based on cross-sections. We represent each sketch as a network of 2D annotated smooth curves, corresponding to cross-sections, smooth silhouettes, and other curves treated as patch boundaries. We use standard vector drawing tools to draw these curves. Our algorithm consists of two main steps illustrated in Figure 4.

Cross-Section Plane Estimation: For each network of cross-sections in the sketch, we estimate the support plane of each curve by casting the different cross-section properties as components in a constrained optimization formulation that solves for the plane normals and relative offsets (Section 4.2, Figure 4 (b)). The plane equations determine the 3D positions of the cross-section curves in a local frame.

Normal Propagation: We use the 3D curves to estimate normals across the drawn object. The surface normals at the cross-hairs are defined by the cross-product of the intersecting 3D tangents. We propagate these normals along and in between the curves in a manner consistent with the perception of cross-sections as principal curvature directions, by relying on a rotation minimizing construction inspired by [Biard et al. 2010] (Section 4.3, Figures 4 (c) and (d)).

An alternative workflow would be to use the 3D cross-sections to extract a depth field and use it to obtain normals. However, this alternative is not suitable for our needs. While normals are only affected by the object shape in the immediate vicinity, computing depth values across the entire object requires a global solution. Moreover such a solution would need to be robust to depth inconsistencies which often show up in artist drawn sketches [Schmidt et al. 2009a]. Our approach which operates directly on normals sidesteps these difficulties. While depth can help with complex shading effects such as shadows, normals are sufficient to generate the aspired 3D look, for production drawings.

In the discussion below we use the view space coordinate frame where x and y are the horizontal and vertical axes of the image plane, and z is the vector pointing out of the plane. Recovering 3D information from 2D data typically requires inverting the projection transformation, a non-trivial task for free-hand sketches where even expert designers don't draw perspective accurately [Schmidt et al. 2009a]. Therefore, the formulation below implicitly assumes orthographic projection, accounting for weak perspective as well as other drawing imperfections through the use of tolerance-based, rather than exact, constraints.

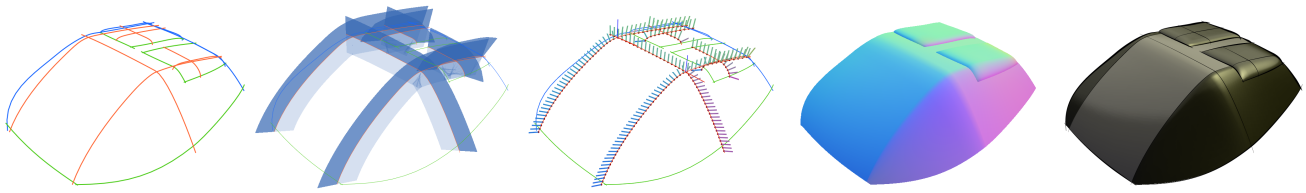
4.2 Estimating Cross-Section Curve Planes

Given a network of curves, let $\mathbf{n}_i$ be the normal to the plane of the ith cross-section and $\mathbf{t}_{ij}$ be its 3D tangent at the intersection with cross-section j. The 2D curves define the x and y coordinates for each tangent, such that only $\mathbf{t}_{ij}^z$ is unknown. To obtain the plane normals we combine the five properties above into a constrained minimization formulation. We enforce tangent and plane orthogonality, which are persistently satisfied by designer sketches, using constraints. However to allow for drawing inaccuracies we reformulate those as inequalities:

$$\begin{aligned} -\epsilon \leq \mathbf{n}_i \cdot \mathbf{n}_j \leq \epsilon \\ -\epsilon \leq \mathbf{t}_{ij} \cdot \mathbf{t}_{ji} \leq \epsilon \end{aligned} \tag{5}$$

where $\epsilon = 0.1 = \arccos(85^\circ)$ for artist drawn sketches. We use a tighter threshold of $\epsilon = 0.05$ for the ground truth comparisons (Section 5) where the 2D tangents are known to be nearly exact.

For a single cross hair combining both constraints with either geodesic alignment (Equation 3) or foreshortening minimization (Equation 4), leads to an under-constrained problem, with a continuous set of solutions. However, if both are enforced, the solution



(a) Input curves (b) Estimated curve planes (c) Normals along curves (d) Normals over the sketch (e) Resulting shading

Figure 4: *Our algorithm takes as input an annotated sketch (a). Orange curves denote cross-sections, blue curves represent smooth silhouettes, and green curves correspond to other object boundaries. We first optimize for the supporting plane of each cross-section and compute the 3C cross-sections based on those (b). We use the resulting 3D curves to compute 3D normals at each intersection and interpolate normals along the curves (c). We finally generate a normal field in between the curves using Coons' interpolation (d).*

is unique up to orientation. At the same time these two criteria serve as weaker cues which are not always satisfied. We therefore cast the search for the planes as a constrained minimization where the objective function aims to satisfy both properties, subject to the inequalities:

$$\min_{\mathbf{n}_i} \sum_{ij} (\|\mathbf{t}_{ji} \times \mathbf{n}_i\|^2 + \|\mathbf{t}_{ij} \times \mathbf{n}_j\|^2) + (\mathbf{t}_{ij}^z)^2 + (\mathbf{t}_{ji}^z)^2. \quad (6)$$

The sum goes over all the intersections in a cross-hair network. We found that the equal weight assigned to both terms leads to results consistent with both ground truth comparisons and human perception (Section 5).

We also add equality constraints describing several obvious geometric relationships. Since each tangent is orthogonal to the corresponding normal we have

$$\mathbf{t}_{ij} \cdot \mathbf{n}_i = 0. \quad (7)$$

Rather than enforcing the normals $\mathbf{n}_i$ to be unit length, we found it more convenient to assume that $\mathbf{n}_i^z = 1$ for all i. This equality reduces the degree of several of the terms, in particular making the optimized functional (Equation 6) quadratic and thus easier to minimize. This assumption works in the general position setup, where no cross-sections are perpendicular to the view plane, which is the typical case for concept sketches.

We also recognize that at each cross-hair, the intersecting 3D curves share the same depth z. Using this observation we associate an offset value $\mathbf{c}_i$ with each plane that represents its location in 3D and express the offset using one equation for each of the planes at each cross-hair position $\mathbf{x}$:

$$\mathbf{x} \cdot \mathbf{n}_i + \mathbf{c}_i = 0, \;\; \mathbf{x} \cdot \mathbf{n}_j + \mathbf{c}_j = 0. \quad (8)$$

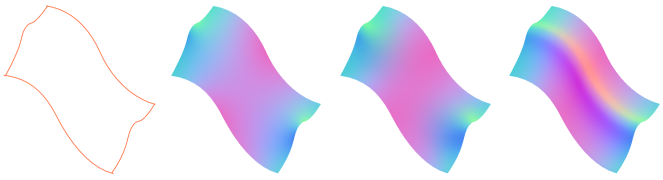
Since $\mathbf{n}_i^z = \mathbf{n}_j^z = 1$, subtracting the two equations eliminates the depth variable and gives the offsets in terms of the normals and the x, y components of the cross-hair position which are known:

$$\mathbf{x}_x(\mathbf{n}_i^x - \mathbf{n}_j^x) + \mathbf{x}_y(\mathbf{n}_i^y - \mathbf{n}_j^y) + (\mathbf{c}_i - \mathbf{c}_j) = 0. \quad (9)$$

The offset is relative to one arbitrarily chosen origin cross-hair where we set the position to be (0,0,0) and the two crossing planes to have offset 0.

Solution: We use the interior point method, as implemented in MATLAB, to solve for the normals and offsets. Since the number of variables is typically very small, we achieve near interactive results even though this is not the most efficient implementation.

A simple counting argument shows that for any loop of n cross-sections with n intersections in general position, i.e. with no parallel tangents, the constraints alone fully define the normals up to an orientation choice. In other cases the optimization of the geodesic and foreshortening terms together similarly leads to a sparse set of solutions, defined up to local orientation choice at each cross-hair.



(a) Input curves (b) Laplace (c) Thin-Plate Splines (d) Coons

Figure 5: *Solving the Laplace or the Thin-Plate-Spline equations (b,c) propagates normals isotropically. Coons interpolation (d) better preserve the directionality of the curve network.*

To obtain a consistent set of cross-hair orientations, we specify an initial guess for the optimization which reflects the orientation at two cross-hairs. We observe that for a single cross-hair one can compute an analytic solution which minimizes foreshortening, while enforcing the three other properties.The initial guess for the global solve consists of two such local solutions and zero for all other values.

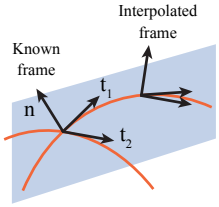
If the network touches a silhouette curve, we select for the initial guess cross-hairs closest to the silhouette and make an orientation choice consistent with the orientation at the silhouette (Section 3). If not, we select far apart cross-hairs and perform the optimization four times using the four possible orientation pairs as initial guesses. We then select the solution most consistent with the *view from above* prior. Users can enforce an alternative solution by fixing the orientation of one or more cross-hairs.

Once the curves' planes are obtained we have an approximation of the curve network in 3D space. We compute the 3D tangents at the intersections such that they lie in their respective planes. The 3D normal at each intersection is the cross product of the two tangents.

4.3 Normal Propagation

We use the computed 3D cross-section curves and the normals at the intersections to compute the normals across the object surface. To align the principal curvature directions with the cross-section curves, consistent with viewer perception, we need the normal gradients to align whenever possible with the curves. Existing normal propagation methods such as Lumo [Johnston 2002] and ShapePalettes [Wu et al. 2007] do not achieve such alignment, even when the input normals are specified along the boundaries of a rectangular, parameterized domain, see Figure 5 (b,c), let alone when the normals are only specified at a sparse set of intersections or along open curves.

Instead we use a method inspired by the work of Biard et al.[2010]. They construct patches bounded by curvature lines from quadruplets of points with associated principal curvature frames, by first generating suitable 3D bounding curves with associated principal curvature frames and then use Coons interpolation to generate patches on which these curves serve as curvature lines. In our case the 3D curves we aim to align with curvature lines are already given, eliminating a significant degree of freedom. To approximate their construction we use a twist-minimizing propagation of principal curvature frames along the 3D cross-section curves and then use a variation of the Coons interpolation to obtain normals across the surface.



Propagating Curvature Frames Along Cross-Sections. At each cross-hair the curvature frame consists of the tangent $\mathbf{t}_1$ to the current curve, the surface normal $\mathbf{n}$, and the tangent $\mathbf{t}_2$ of the intersecting curve. We propagate those along the curves, such that the principal curvature frame at any point contains the tangent $\mathbf{t}$ to the curve at that point. To minimally rotate the rest of the frame we extract the angle of rotation between $\mathbf{t}_1$ and $\mathbf{t}$ around the axis of rotation $\mathbf{t}_1 \times \mathbf{t}$ and apply the same rotation to the other frame components. For a portion of a cross-section curve bounded by only one intersection, this process is sufficient. For a segment bounded by two intersections we smoothly blend the rotated frames, obtained from the two intersections independently, using an arc-length based weighting. We use the normals defined by these frames, shown in Figure 4(c), to compute the normal field across the rest of the shape.

Interpolating Normals in-between Curves. To compute the normal field inside a four sided region bounded by cross-section curves with specified normals, we apply Coons interpolation directly to the curve normals, using discrete bi-linearly blended Coons patches [Farin and Hansford 1999]. The obtained results visually align the lines of curvature with the cross-sections capturing the viewer perceived shape, as shown in Figure 4 (d). To apply this step however, we need to pre-process the curve networks to extract the four-sided regions as well as to process regions with different topology.

To obtain a well defined set of manifold regions, we extend "hanging" cross section end-points in the tangent direction to the nearest boundary or silhouette curve. We then extract the region connectivity from the intersecting curves by constructing a planar map [Gangnet et al. 1989] which generates a list of regions bounded by curve segments. We distinguish four possible configurations:

1. The region is bounded by 4 cross-section curves. We directly apply Coons interpolation on the quad.

2. The region is bounded by $n \neq 4$ cross-section curves. We apply mid-point subdivision to generate n quads from the n-sided face (Figure 6(a)). The faces resulting from this subdivision share normals along the artificial segments. To position the midpoint we use the templates described by Nasri et al. [2009]. To assign normals to each boundary between a pair of adjacent quadrilateral faces we use the discrete Coons formulation [Farin and Hansford 1999] treating the two faces as a single quadrilateral domain. We then apply Coons interpolation in the interior of each face.

3. The region is bounded by more than 3 consecutive cross-section curves and by a boundary curve. We generate artificial normals along the boundary and we subdivide the resulting n-sided face (Figure 6(b)). To assign normals to the boundary curve we copy normals from the opposite cross-section segments and rotate the set of normals so that they agree with the normals of the adjacent segments at the extremities of the boundary (Figure 7).

4. The region is bounded by 2 or 3 consecutive cross-section curves and by boundary curves. We generate a quad that covers all the curves, and trim it to remove extraneous parts (Figure 6(c)). To do so, we first extend the cross-section segments in their tangential directions. We then close the quad with artificial sides that we generate by sweeping the existing cross-section segments against their adjacent segments. We apply Coons interpolation on the quad before trimming.

While other configurations may occur, we didn't observe them in design sketches. If such a configuration shows up users can easily resolve it by introducing additional cross-sections.

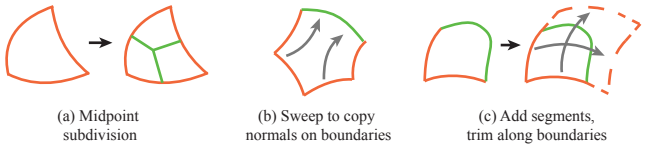


Figure 6: *Illustration of our strategies to handle non-quad faces. If a face is bounded by $n \neq 4$ segments, we subdivide it (a). If a face is bounded by more than 3 cross-sections and by boundaries, we copy the normals of the cross-sections onto the boundaries via sweeping (b). If a face is bounded by 2 or 3 cross-sections, we create a quad by sweeping each cross-section against its adjacent segment, and trim the quad along the boundaries (c).*

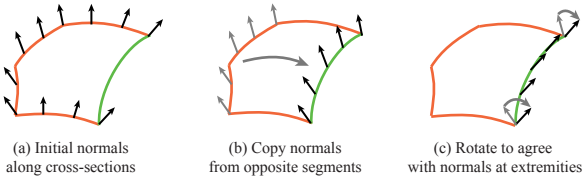


Figure 7: *Sweeping normals from cross-sections onto boundaries. We first copy normals from the segments opposite to the boundary. We then rotate the set of artificial normals so that they agree with the normals available at the extremities of the segment.*

5 Validation

We validate the use of designer cross-sections for shading concept sketches in general and using our algorithm in particular via a combination of a user study and a ground-truth (GT) comparison. The first part of the study aims at the general question "Do design cross-sections effectively depict 3D shape?" by posing the following queries:
Q1. Are viewers persistent, and consistent with each other in their perception of design cross-hairs?
Q2. Are viewers accurate in their perception of cross-hairs?
Q3. Is the persistence/consistency/accuracy of the perceived cross-hairs correlated to a global context, and the geodesic, minimal foreshortening and up-orientation properties of cross-sections?

The study also serves to validate our algorithmic results by measuring similarity between them and viewer perceived cross-hair normals. Lastly we use the example corpus in the study to compare these results to GT normals.

5.1 User Study Design

Sketch data-set. We choose a corpus of 11 models (see Figures 8) of varying shape complexity and recognizability, depicted using silhouettes and cross-sections of varying interconnectivity. The corpus comprises 4 artist sketches and 7 ground-truth 3D shapes. While the former can be used to answer questions about persistence and consistency, they cannot be used to evaluate accuracy. Thus, to address accuracy we use ground truth shapes, depicted using cross-sections with orthogonal planes and tangents at cross-hairs. We deliberately select some cross-hairs that violate the geodesic and minimal foreshortening property to address **Q3**. We address context in **Q3** by presenting the sketches in three forms: as *complete* sketches with cross-sections and silhouettes; as *partial* cross-sections clipped around cross-hairs at roughly a third of their arc-length; and as *isolated* crosshairs, randomly translated in the viewplane to mitigate any memory bias (see Figure 9).

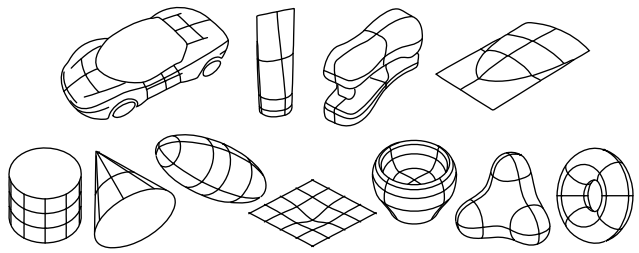
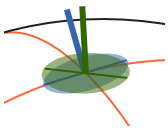


Figure 8: *User study corpus comprising 4 artist sketches: car, tube, stapler and bump (top row), and 7 ground truth surfaces: cylinder, cone, ellipsoid, ashtray, bowl, trebol, torus (bottom row).*



User Interface. For each cross hair we ask users to interactively specify the surface normal using a line connected to the cross-hair center, providing a 2D projection of the perceived normal. While gauges [Koenderink et al. 1992; Cole et al. 2009] are the more typical interface for querying normals, our pilot study, conducted using them revealed a user tendency to orient the gauge by attempting to turn the base into an ellipse whose axes were aligned with the cross-hair in the viewplane, introducing a bias error. Our interface avoids such bias since the cross-hair itself perceptually functions as a base.

User Study Protocol. Participants (25 designers and 30 non-designers) were shown through a web interface, in random order, 50 to 80 sketches on which a single cross-hair normal was queried. Each cross-hair normal was queried twice at different points in the study. The overal task took 8-15 minutes.

5.2 User Study Findings

Overall 2706 normals were specified on 40 complete, 28 partial, and 14 isolated cross-hairs images. We use 2D angles to measure all differences. We found less than a 2^o difference in favor of designers on every measured statistic between designers and non-designers, indicating similar perceptual acuity, and thus report combined statistics. We account for alternate interpretations when measuring persistence and consistency by reporting the acute angle between normal samples, but also the percentage of samples that are flipped (the angle between the normals is obtuse) (see Table 1). To avoid bias we do not discard outliers, as a result the median value is typically more representative than the average.

Q1: Design cross-hairs are *persistent*, in that the difference between normals set twice by the same viewer at different points in the study is small, with the median standing at 6^o. They are also *consistent*, measured as a pairwise difference between normals across viewers, with a median of 10.6^o.

Q2: Viewer perception of design cross-hairs normals is *accurate* when compared to GT, with a median of 8^o, and a mean of 14^o validating the orthogonal plane and tangent properties we used to construct our GT data. The deviation of user perceived normals is also comparable to that reported for other line drawing styles [Cole et al. 2009].

Q3: The impact of *global context* on cross-hair perception is evident from the increase in the number of flipped orientations, in the computation of each statistic and general deterioration of consistency and accuracy from *complete* to *partial* to *isolated* cross-hairs. User normals on the cone (Figure 9) highlight the importance of context. Without context, (Figure 9 (a,b)) these normals are far from GT and largely consistent with both curves being geodesics, while still minimizing foreshortening (Figure 9 (a,b)). In context, the accuracy increases dramatically. Comparison of user normals on the ellipsoid, with and without context (inset (c)), shows similar effect with respect to foreshortening minimization. In the full context, in the presence of silhouettes, we did not observe flipped orientations. The results for the ashtray (concave) and bump (convex) (Figure 9, (e,d)) both reinforce viewer tendency to perceive the surface orientation as vertical and viewed from above, even when no silhouettes are present. However as demonstrated by the ashtray, the flipped orientation is preferred by a small set of users. For isolated cross-hairs, inputs with no clear up direction (Figure 9 (a)), users are split almost evenly between the two choices.

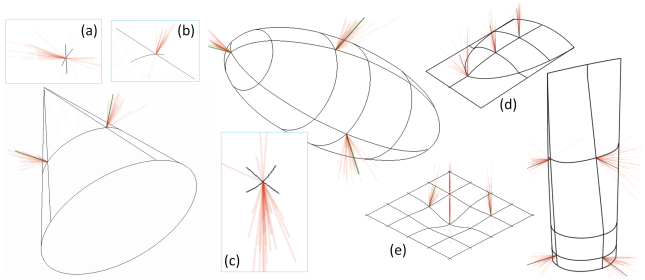


Figure 9: *Representative user (red), algorithm (black), and GT (green) normals on concept sketches and isolated cross-hairs. Insets (a) and (b) correspond to isolated cross-hairs on the cone and (c) on the ellipsoid. (a) highlights how absent the context viewers perceive the local surface as cylindrical, consistent with both cross-hairs being geodesics; In (c) the same effect is evident with respect to foreshortening. With no context viewers select a normal which points more forward rather than sideways.*

5.3 Algorithm Validation

Table 1 shows our algorithmic cross-hair normals to be consistent with viewer perceived normals, on both GT and artist drawn inputs, with median difference of 8.5^o. It also highlights the accuracy of our method, showing that the 3D difference between the algorithmic normals and ground-truth data is 2.3^o, a near perfect precision. Figure 10 compares the normals estimated by our method with ground truth normals and normals diffused from silhouettes [Johnston 2002]. While the midpoint subdivision introduces small errors in non-quad faces, our normal fields match ground truth closely and produce convincing shading. In contrast, normals from silhouette do not capture the internal structures of the shapes.

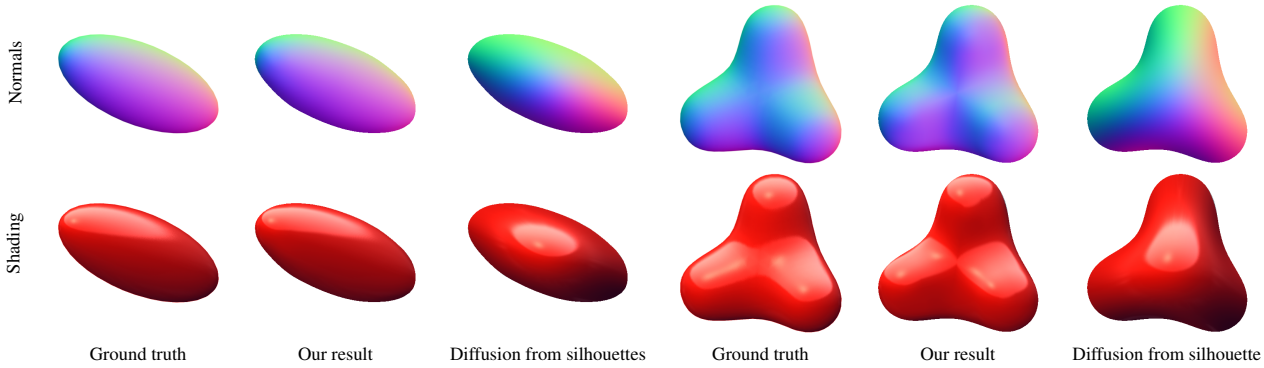


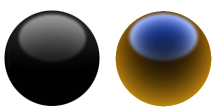
Figure 10: *Comparison with ground truth normals and normals from silhouettes [Johnston 2002]. Our normal fields and shading match ground truth closely. Small deviations from ground truth appear in the center of the shape where midpoint subdivision is applied.*

	Pers. intra user	Cons. inter user	Acc. user/GT	Acc. user/GT - outliers	Cons. user/algo	Acc. algo/GT
Complete						
md.	5.8	10.6	8.3	7.7	8.5	2.3
mn.	10.1	15.7	13.7	10.3	12.3	2.6
std.	12.3	16	20.4	9.5	11.9	2.1
flip	1.1	2.2		5	1.2	
sampl.	655	21126	853	811	1303	26
Partial						
md.	6.7	10.7	8.6	8	9.7	
mn.	12.1	17.4	18.1	12	14.5	
std.	16.8	19.2	30.9	12.4	16.3	
flip	6.2	9.6		5	5.9	
sampl.	463	15149	467	444	934	
Isolated						
md.	7.5	11.5				
mn.	11.6	16				
std.	13.3	16.1				
flip	10.4	19				
sampl.	235	7805				

Table 1: *Summary of our validation data. (Left to right) persistence same user, consistency between users, accuracy wrt. to GT, accuracy with worst 5% of errors removed (notice the significant drop in the standard deviation), consistency wrt. to our solution, accuracy of our solution wrt. to GT.*

6 Results

We tested our method on a variety of inputs, ranging from simple models such as the toothpaste or the sofa (Figure 11), to elaborate products like the car (Figure 13), the coffee-maker (Figure 13), and the camera (Figure 14). In all cases our method reconstructs a believable normal field which lends itself to the subsequent shading.



Rendering. While automatic shading quality can never match the expressiveness of manual artwork, our results capture many of the features commonly used in production drawings. We use the lighting model by Gooch et al. [1998] to mimic illustrative shading. Users can additionally indicate the base color of the object using a standard painting tool. Following designers' practice, we apply the abstract environment maps shown as inset to suggest studio lighting over glossy plastic (Figure 13, coffee-maker) and outdoor lighting over metal and car paint (Figure 14, robot). Finally we keep the cross-sections visible to convey shape, as suggested by Eissen and Steur [2008]. Figure 12 provides a visual comparison between sketches manually shaded with SketchBook Pro© and the result of our method.

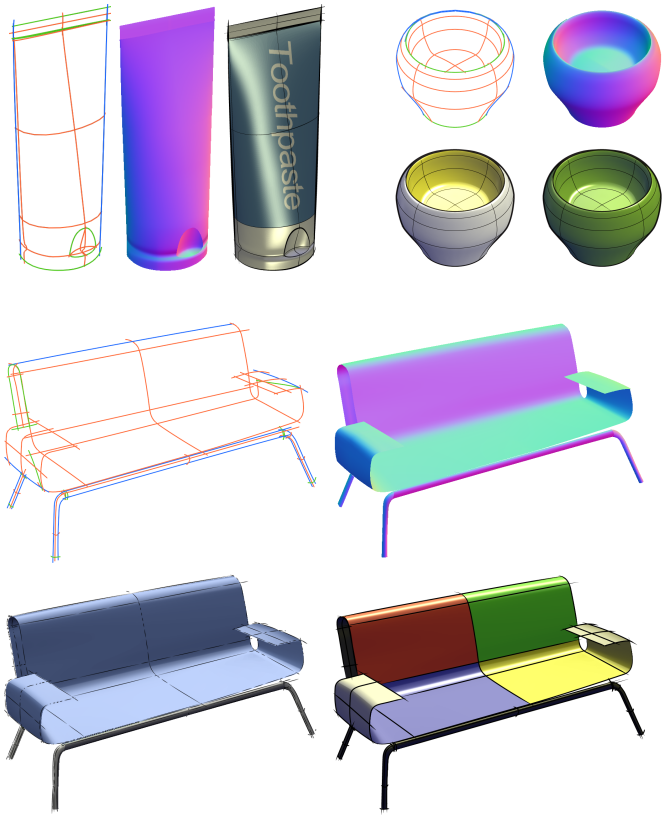
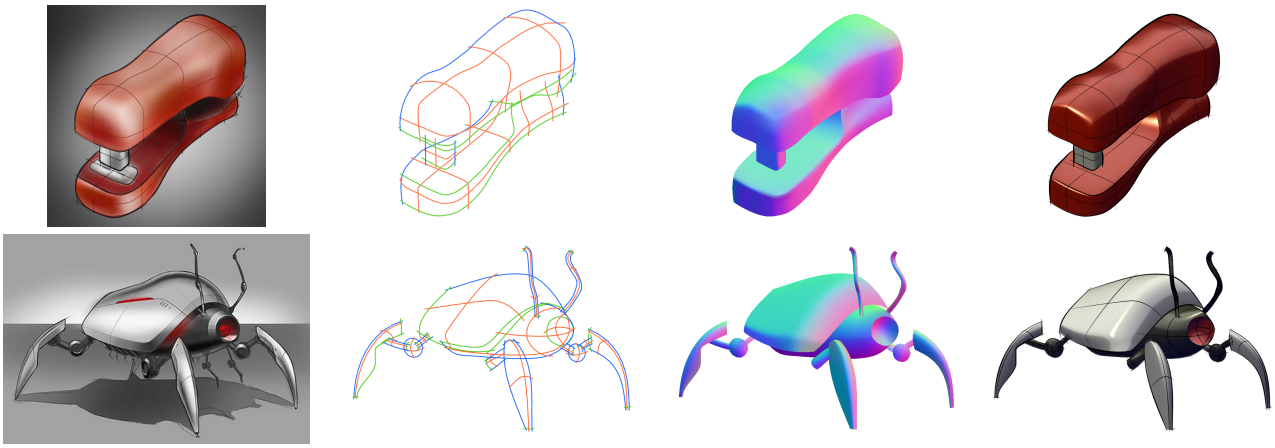


Figure 11: *Even on simple inputs the shading enabled by our method provides for a much richer, complex 3D look.*

User Interface. Our system is based on an interactive interface where users sketch annotated lines either from scratch or on top of a pre-existing sketch. The accompanying video shows a typical drawing session. We provide a layer based interface to draw surfaces in different depths and support occlusion between overlapping layers. Our implementation of the normal estimation and propagation allows users to create normal fields interactively, with total runtime ranging from less than a second for simple models like the mouse (Figure 4, 2 layers and 14 curves) to around 30 seconds for multi-layer sketches like the car (Figure 13, 17 layers, 119 curves). Most users draw layers independently in an iterative process. Users can then choose from a variety of rendering styles.



(a) Manual shading (b) Input curves (c) Normals (d) Our shading

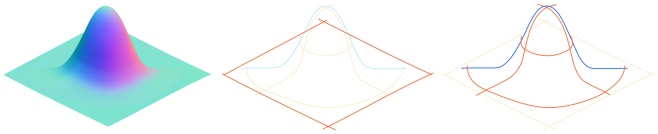
Figure 12: *Comparison between our shading and manually shaded sketches which we used as inspiration. Stapler by Che Yan, robot by Spencer Nugent© on http://www.sketch-a-day.com/.*

7 Conclusions

We presented a method for shading cross-section based concept sketches, enabling designers to easily convert such basic sketches into elaborate production drawings. The technical contribution of of our work is twofold. We identify and formulate five key properties of designer-drawn cross-sections and cross-hairs: orthogonality, alignment with curvature lines, alignment with geodesics, minimal foreshortening, and silhouette-consistent orientation. Then, we use these properties to automatically compute normal fields from sketched cross-section curve networks. Both the properties and the algorithm are validated through a combination of a user study with a ground truth comparison. In this comparison the median difference between our normals and ground-truth at cross-hairs was under **three degrees**, i.e. we obtained near perfect reconstruction.

Limitations. We make two assumptions about the input sketches. We assume orthographic or near-orthographic projection (our inequality formulation can account for some perspective deformation, but not all), and general position - i.e. we expect the cross-section planes to not be orthogonal to the view plane, as the alternative is inherently ambiguous.

Our patch-based normal interpolation does not support open regions. As a result, users sometimes need to draw open contours in separate layers and close the regions with additional curves. Figure 15 illustrates how two layers are used to draw the open contour of a bump over a flat surface. Normal continuity across the additional curves could be enforced with a smoothing post-process around the curves.



(a) Normals (b) First layer (c) Second layer

Figure 15: *Limitation of our normal interpolation. The open contour of the bump needs to be closed and drawn in a separate layer to form valid regions for the Coons interpolation.*

Future Work. We foresee a number of areas for future work. We would like to combine cross-sections with other construction lines as well as other priors for processing different types of sketches. As an example, in addition to disambiguating orientation, smooth silhouettes could be used as an additional term in the optimization to enforce normals to be perpendicular to the view plane along them. A more ambitious research direction is to extend our method to perform accurate geometry reconstruction, potentially relying on intrinsic symmetries to reconstruct occluded regions.

Acknowledgments

Many thanks to Mike Boers for help with output rendering, Che Yan for sharing her design expertise, Maneesh Agrawala for fruitful discussions on shading in design sketches, and George Drettakis, Olga Sorkine, Pascal Barla and Romain Vergne for their comments on early versions of the paper. We also thank Koos Eissen, Roselien Steur, Spencer Nugent and Mike Serafin for allowing us to use their inspiring sketches. INRIA acknowledges generous support from Adobe and Autodesk (software and research donation) and NVIDIA (Professor Partnership Program). This research was supported by NSERC, GRAND NCE and MITACS NCE.

References

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

ANDRE, A., SAITO, S., AND NAKAJIMA, M. 2007. Crosssketch: freeform surface modeling with details. In *Proc. Symp. on Sketch-Based Interfaces and Modeling*, 45–52.

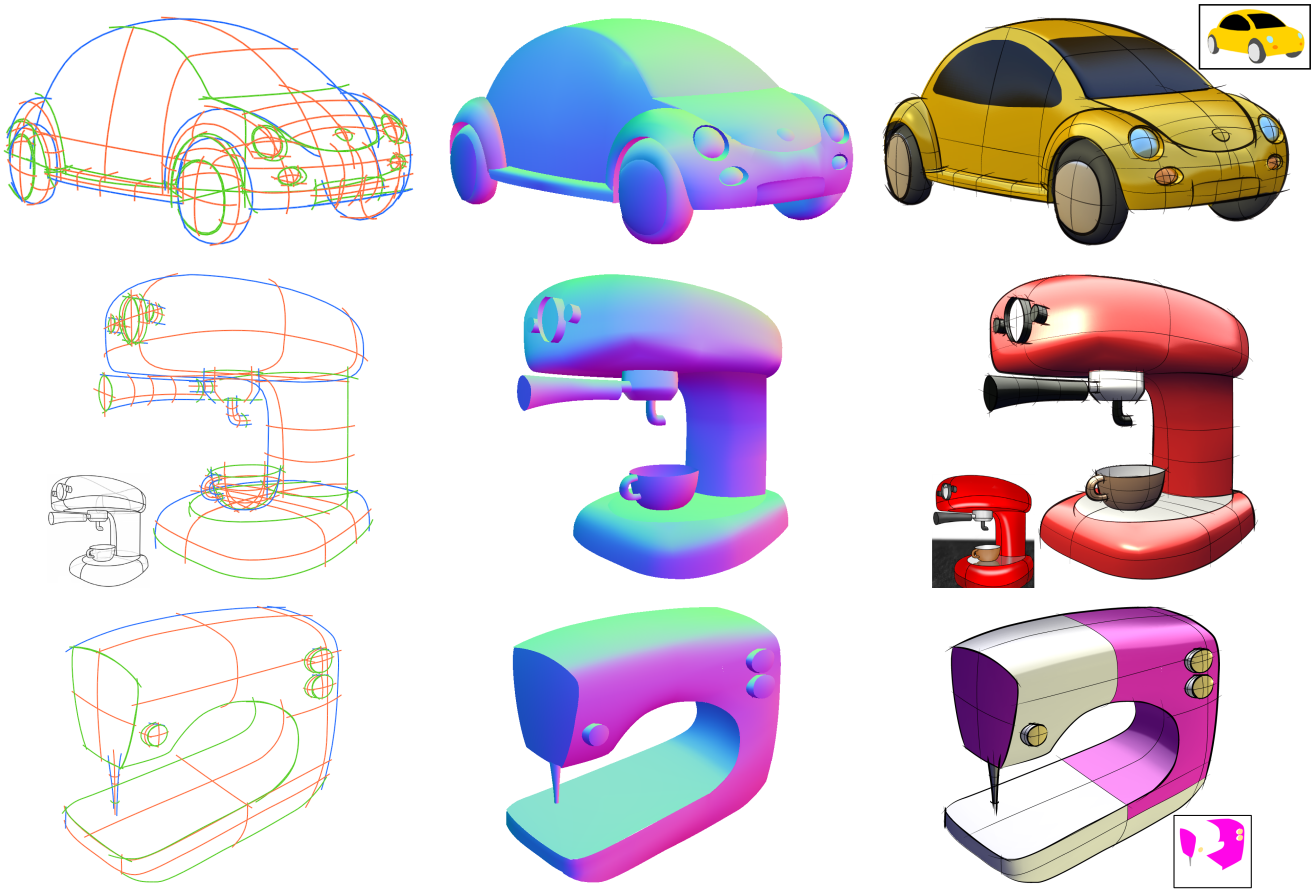
BAE, S., BALAKRISHNAN, R., AND SINGH, K. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proc. User Interface Software and Technology*.

BIARD, L., FAROUKI, R. T., AND SZAFRAN, N. 2010. Construction of rational surface patches bounded by lines of curvature. *Comput. Aided Geom. Des. 27*, 359–371.

COLE, F., SANIK, K., DECARLO, D., FINKELSTEIN, A., FUNKHOUSER, T., RUSINKIEWICZ, S., AND SINGH, M. 2009. How well do line drawings depict shape? *ACM Trans. on Graph. (Proc. of SIGGRAPH) 28*, 3.

COOK, R. 2008. Behind the scenes at pixar. In *Keynote talk, Siggraph Asia*.

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



(a) Input curves (b) Normals (c) Shading

Figure 13: *Results of our method on complex inputs. Insets show the original curves and manual shading for the coffee machine by Schmidt et al.[2009b] , and the user-provided base color for the car and sewing machine.*

(a) Traced curves (b) Normals (c) Composite shading (d) Alternative shading

Figure 14: *Variations on a theme: (top) our method makes it easy to apply different shading styles to models. (bottom) Users can trace over an existing image (a, shown as inset) and composite the resulting shading with the hand drawn sketch (c) or create other shading styles (d) through base color selection. Original sketches from Spencer Nugent© on http://www.sketch-a-day.com/.*

DO CARMO, M. P. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ.

ECKER, A., KUTULAKOS, K., AND JEPSON, A. 2007. Shape from planar curves: A linear escape from flatland. In *IEEE Computer Vision and Pattern Recognition*.

EISSEN, K., AND STEUR, R. 2008. *Sketching: Drawing Techniques for Product Designers*. Bis Publishers.

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

FARIN, G., AND HANSFORD, D. 1999. Discrete coons patches. *Computer Aided Geometric Design 16* (August), 691–700.

FINCH, M., SNYDER, J., AND HOPPE, H. 2011. Freeform vector graphics with controlled thin-plate splines. *ACM Trans. on Graph. (Proc. SIGGRAPH Asia) 30*.

GANGNET, M., HERVÉ, J.-C., PUDET, T., AND VAN THONG, J.-M. 1989. Incremental computation of planar maps. *SIGGRAPH 23* (July), 345–354.

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

GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. *SIGGRAPH*, 447–452.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. *SIGGRAPH*.

JOHNSTON, S. F. 2002. Lumo: illumination for cel animation. In *Proc. Symp. on Non-Photorealistic Animation and Rendering*.

JOSHI, P., AND CARR, N. A. 2008. Repoussé: Automatic inflation of 2d artwork. In *Proc. of Sketch Based Interfaces and Modeling*.

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.

KOENDERINK, J. J., DOORN, A. J. V., AND KAPPERS, A. M. L. 1992. Surface perception in pictures. *Perception & Psychophysics*, 487–496.

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.

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.

MCCRAE, J., SINGH, K., AND MITRA, N. J. 2011. Slices: A shape-proxy based on planar sections. *ACM Trans. on Graph. (Proc. SIGGRAPH Asia) 30*, 6.

NASRI, A., SABIN, M., AND YASSEEN, Z. 2009. Filling N -Sided Regions by Quad Meshes for Subdivision Surfaces. *Computer Graphics Forum 28*, 6 (Sept.), 1644–1658.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. on Graph. (Proc. SIGGRAPH) 26* (July).

OKABE, M., ZENG, G., MATSUSHITA, Y., IGARASHI, T., QUAN, L., AND YEUNG SHUM, H. 2006. Single-view relighting with normal map painting. In *Proc. Pacific Graphics*, 27–34.

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

ORZAN, A., BOUSSEAU, A., WINNEMÖLLER, H., BARLA, P., THOLLOT, J., AND SALESIN, D. 2008. Diffusion curves: A vector representation for smooth-shaded images. *ACM Trans. on Graph. (Proc. SIGGRAPH) 27*.

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

ROBERTSON, S. 2003. *How to Draw Cars the Hot Wheels Way*. MBI.

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

SCHMIDT, R., KHAN, A., SINGH, K., AND KURTENBACH, G. 2009. Analytic drawing of 3d scaffolds. *ACM Trans. on Graph. (Proc. SIGGRAPH Asia) 28*, 5.

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

SÝKORA, D., DINGLIANA, J., AND COLLINS, S. 2009. LazyBrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum (Proc. EUROGRAPHICS) 28*, 2.

WINNEMÖLLER, H., ORZAN, A., BOISSIEUX, L., AND THOLLOT, J. 2009. Texture design and draping in 2d images. *Computer Graphics Forum (Proc. Symp. on Rendering) 28*, 4.

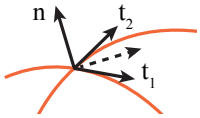
WU, T.-P., TANG, C.-K., BROWN, M. S., AND SHUM, H.-Y. 2007. Shapepalettes: interactive normal transfer via sketching. *ACM Trans. on Graph. (Proc. of SIGGRAPH) 26*.

ZHANG, L., DUGAS-PHOCION, G., SAMSON, J.-S., AND SEITZ, S. M. 2002. Single view modeling of free-form scenes. In *IEEE Computer Vision and Pattern Recognition*, 990–997.

Appendix

Given the tangent and plane orthogonality properties, we prove that one of the plane normals must be collinear with the tangent of the other plane (i.e. one of the curves is a surface geodesic).

Proof. We have $\mathbf{t}_1 \cdot \mathbf{t}_2 = 0$, $\mathbf{n}_1 \cdot \mathbf{n}_2 = 0$, $\mathbf{t}_1 \cdot \mathbf{n}_1 = 0$, and $\mathbf{t}_2 \cdot \mathbf{n}_2 = 0$. Assuming, $\mathbf{t}_1$ is not collinear with $\mathbf{n}_2$, then, these two vectors define a plane P, s.t. both $\mathbf{t}_2$ and $\mathbf{n}_1$ are orthogonal to it (as both are orthogonal to the two spanning vectors). Since a plane allows only one normal (up to sign), then $\mathbf{t}_2$ and $\mathbf{n}_1$ are co-linear.



Stevens suggests that users perceive the normal to a cross-hair as being orthogonal to the tangent bisector (see inset for notations) [Stevens 1981]. We prove that this is equivalent to locally minimizing foreshortening $(\mathbf{t}_1^z)^2 + (\mathbf{t}_1^z)^2$.

Proof. Given tangent orthogonality, the local optimum of this term is obtained, given the tangent orientation in the inset, when $\mathbf{t}_1^z = -\mathbf{t}_2^z$. Let the outward normal n to the surface be defined as $\mathbf{t}_1 \times \mathbf{t}_2$. Hence $n^x = \mathbf{t}_1^y\mathbf{t}_2^z - \mathbf{t}_2^y\mathbf{t}_1^z = \mathbf{t}_2^z * (\mathbf{t}_1^y + \mathbf{t}_2^y)$ and $n^y = \mathbf{t}_1^z\mathbf{t}_2^x - \mathbf{t}_2^z\mathbf{t}_1^x = -\mathbf{t}_2^z(\mathbf{t}_1^x + \mathbf{t}_2^z)$. This 2D normal is consequently orthogonal to the 2D bisector of $t\mathbf{t}_1$ and $\mathbf{t}_2$, $b = ((\mathbf{t}_1^x + \mathbf{t}_2^x)/2, (\mathbf{t}_1^y + \mathbf{t}_2^y)/2)$, which is equivalent to the bisector condition.