Slices: A Shape-proxy Based on Planar Sections

James McCrae   Karan Singh
University of Toronto

Niloy J. Mitra
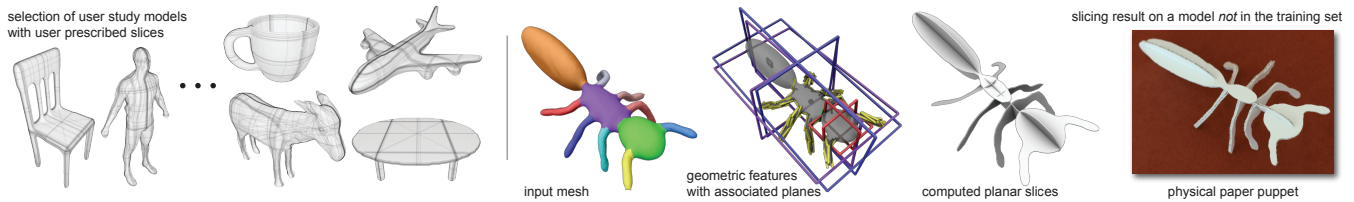KAUST/UCL

selection of user study models
with user prescribed slices

input mesh

geometric features
with associated planes

computed planar slices

slicing result on a model *not* in the training set

physical paper puppet

Figure 1: *Based on a user study, where participants abstracted meshes of common objects using a small collection of planar slices, we develop an automatic algorithm to create planar slice based abstractions of (untrained) models. Starting from a set of planar slices, approximating the object's geometric features, the algorithm picks a subset of planes based on relative feature importance learned from the user study.*

Abstract

Minimalist object representations or *shape-proxies* that spark and inspire human perception of shape remain an incompletely understood, yet powerful aspect of visual communication. We explore the use of planar sections, i.e., the contours of intersection of planes with a 3D object, for creating shape abstractions, motivated by their popularity in art and engineering. We first perform a user study to show that humans do define consistent and similar planar section proxies for common objects. Interestingly, we observe a strong correlation between user-defined planes and geometric features of objects. Further we show that the problem of finding the minimum set of planes that capture a set of 3D geometric shape features is both NP-hard and not always the proxy a user would pick. Guided by the principles inferred from our user study, we present an algorithm that progressively selects planes to maximize feature coverage, which in turn influence the selection of subsequent planes. The algorithmic framework easily incorporates various shape features, while their relative importance values are computed and validated from the user study data. We use our algorithm to compute planar slices for various objects, validate their utility towards object abstraction using a second user study, and conclude showing the potential applications of the extracted planar slice shape proxies.

Keywords: abstraction, shape proxy, shape perception

Links: ⬦DL 🅿PDF ⬛WEB 📁DATA ⬇CODE

1 Introduction

Over the last few decades, great strides have been made in the area of acquisition and modeling of 3D geometry. The underlying shape representation is typically a collection of polygons, popular for their generality, rendering efficiency, and amenability to geometry processing algorithms. Unfortunately, such a representation can be ex-

pensive and, in itself, neither conveys the essence of the depicted object nor aids in our understanding of the represented shape.

Artists and sculptors have long explored minimalist shape proxies to highlight defining aspects of familiar objects (see Figure 2). As humans, we effortlessly perceive the underlying shapes even from such sectional representations, which greatly differ from their surface representations. In fact, the sparse nature of these representations allows us to see otherwise occluded details and the absence of unnecessary detail makes such artforms attractive, fascinating, and sometimes mysterious. Evidence suggests that symbolic abstractions [Edwards 2002] dominate our mental model of objects. Thus across cultures, we both recognize shape proxies quickly and tend to communicate objects by drawing them as symbolic abstractions.

Planar section proxies are also motivated by medical and engineering visualization where section planes are used to illustrate the interior details of complex shapes (see Figure 2). These planar-sections often pass through anatomic landmarks or engineering features such as channels or bosses, reaffirming our use of high-level shape features like segments, symmetries, ridges and valleys to define planar sections in Section 4.

In geometry processing, model simplification [Cignoni et al. 1997] and variational shape approximation [Cohen-Steiner et al. 2004] remain the dominant modes of shape abstraction. Recently, there has been work on shape abstraction using curve networks [Mehra et al. 2009; de Goes et al. 2011]. These methods and their variants evaluate the quality of an approximation by the geometric deviation of the proxy from the original shape. In contrast, we aim to explore shape abstractions that are based on human perception of form. Given a shape S, our goal is to produce a proxy $\mathscr{S}$ using planar sections of the shape, such that perceptually S and $\mathscr{S}$ are comparable, while representationally $|\mathscr{S}| \ll |S|$. Note that $\mathscr{S}$ are S likely to be quite different from a purely geometric or topological standpoint. The fundamental difficulty in proposing or evaluating

Figure 2: *Planar sections in art (©Alexander Calder), sculpture (wood puppet), design (section chair, ©Hebert Franco) and scientific visualization.*

algorithmic solutions to such a problem is the lack of a computational model of our perceptual response system.

Overview. We conduct a user study to gain insight as to how humans define planar section representations of various 3D objects. We observe that humans share a consistent notion of abstraction using planar sections, and further the chosen planes are correlated to geometric shape features (Section 3). Based on this study, one possibility is to select a minimal configuration of planes that captures a given set of geometric features of the 3D shape, rather than approximating the object surface (Section 4). We show this problem is NP-Hard (Appendix), discuss various design possibilities, and propose a solution where planes are progressively selected to maximally capture shape features weighted by their importance. Each selected plane reduces the importance of the features it captures (so that subsequent planes cover different features) and favor new features on the basis of orthogonality and symmetry relationships among planes in the shape proxy. We then discuss geometric features used in our realization (Section 5) and learn their relative importance from the user-study data. Finally, we evaluate the results of our algorithm on both user-study and novel objects using a second user-study to show that planar proxies are indeed an easily recognizable shape abstraction (Section 6), before discussing potential applications and future directions (Section 7).

Contributions. The key contribution of this paper is the introduction and exploration of perceptually motivated shape proxies using planar sections. We accomplish this in many novel steps: a user study to show the problem is perceptually well-defined, a mathematical formulation of the problem we prove to be NP-Hard, an efficient approximate solution whose results are comparable to user-defined proxies, a second user study to show that the generated planar section proxies are easily recognizable shape abstractions, and explore potential applications.

2 Related Work

Perception and vision research strongly motivate our desire to represent 3D shape using planar sections. It has been shown that when viewing 3D curves, a planarity assumption allows us to better resolve a mental 3D curve from its projected 2D image [Stevens 1981; Todd and Reichel 1990]. This allows us to better understand curves that are planar sections of a shape, and prevents a common misinterpretation of non-planar curves as planar curves with the same view projection. Curvature along surface contours [Koenderink 1990] is known to capture important shape information [Feldman and Singh 2005] and influence our segmentation of shape into salient parts, suggesting that in reverse, both part segmentation and curvature extrema are important cues in the creation of planar section proxies by humans. We indeed observe this to be true in Section 3. Parallel and perpendicular planar sections of a shape have been shown to provide a better understanding of the position and orientation of a surface compared to the shaded surface itself [Sweet and Ware 2004] and might explain the aesthetic appeal in planar section based design (see Figure 2, Figure 5a). In the context of stylized sketching from 3D models, Cole et al. [2008; 2009] study aspects of human perception relating to depicting shapes using line drawings and consistency among line strokes used by humans.

Geometric features range from low-level attributes like curvature [Kalogerakis et al. 2007] that capture a local shape context to higher-level global attributes such as symmetry planes [Mitra et al. 2006]. The computation of these features remain an important research topic. Recent work in geometry processing [Kalogerakis et al. 2010], like ours, attempts to take a black-box approach to features that allows them to leverage existing algorithms and incorporate new and improved shape features. Most high-level shape

features have a natural family of planes that capture the feature as discussed in Section 5.

Curve shape proxies have been used as a shape abstraction in traditional art for centuries. They are popular in graphics and non-photorealistic rendering as expressive and compact visual proxies [Strothotte and Schlechtweg 2002], and used as intuitive handles for shape manipulation [Singh and Fiume 1998; Gal et al. 2009]. Collections of curves, often planar, are also the output of a number of sketch-based modeling systems [Schmidt et al. 2009]. Recently, however, there have been attempts to automatically create curve-based shape abstractions from 3D shapes [Mehra et al. 2009], model and manipulate shapes using 1D arterial snakes [Li et al. 2010a], as well as to determine geometric relationships (such as co-planarity or orthogonality) between these curves [Gal et al. 2009]. Our approach is complementary to these approaches as we focus on understanding and creating good planar section abstractions.

Planar surface approximation addresses the problem of finding planes as good proxies for a 3D shape. Such proxies have been used for extreme simplification leading to billboard clouds [Décoret et al. 2003], error bounded surface simplification [Cohen-Steiner et al. 2004], and as impostors [Kavan et al. 2008] for lightweight rendering. Our goal is to produce a very small number of planes like billboard clouds, but instead of approximating the surface shape most of our planes cut through the interior like polygon impostors. In this regard, our approach is directly applicable to the automatic generation of lightweight impostors (see Figure 13).

Manufacturing and acquisition or product fabrication often require the surface patches to be (nearly) developable [Kilian et al. 2008]. This process is greatly simplified when the patches are simply planar. Pop-up paper architectural models [Li et al. 2010b] and furniture made from planar sheets [Willis et al. 2010] are examples of constructing piecewise planar representations of 3D shapes for easy fabrication (see Figures 1e, 2). In reverse, real-world 3D data is often acquired in the form of 3D planar cross-sectional slices from which the 3D shape must be reconstructed [Ecker et al. 2007; Liu et al. 2008] (see Figure 14). Also, planar shape proxies and scaffolds provide a direct mapping to common 2D devices, making the manipulation, perception and annotation of shape via these proxies easier [Schmidt et al. 2009].

3 Planar Section Proxies Created by Humans

We conducted a user study to answer the following questions regarding how humans define planar section proxies:

1. Can humans, to their perceptual satisfaction, represent common 3D shapes using a small number of planar sections?

2. Is there a correlation among the planar sections prescribed by different humans, for the same 3D shape?

3. How well are the user prescribed planar sections correlated with the geometric features of the 3D shapes?

3D shapes can vary in complexity from a simple sphere, captured by three orthogonal planes (see Figure 8-bottom-left), to an intricately complex engine block (see Figure 2) that may require a large number of planar sections even for a minimal representation. A user study aiming to answer the questions above thus requires a representative set of 3D shapes of manageable and comparable complexity. These models should span various shape categories, such as airplanes, cups or tables, and also have examples capturing shape variability within each category. We found the models from the Princeton 3D segmentation benchmark (PSB) [Chen et al. 2009]

suitable as each model is roughly axis aligned, well sampled, watertight, and creates well-defined planar sections either on the entire object or per segment (see Figure 1). All objects were centered to origin and normalized to unit box.

User study design. In a pilot study, we asked 5 artists and 13 amateurs to interactively explore planar section proxies for presented 3D shapes. Without instruction, some tried to capture shape detail by spending over half an hour to place upwards of 20 planes as regularly placed sections (see Figure 5). We empirically found that most shapes in the PSB could be captured well using 5-10 planes, requiring about 5-10 minutes/model. For the actual user study we thus asked each participant to select planar sections with a soft limit of 10 planes/model over a selected set of 19 models consisting of 5 cups, 5 airplanes, 5 tables, a chair, a machine part (bearing), a biped (human) and a quadruped (donkey) (see Figure 4).

A single plane intersecting a closed manifold can produce a number of disconnected contours. In some cases all the disjoint contours of a planar section provide a useful abstraction of the shape, such as the cross-section of the legs of the table or human feet in Figure 4. Some users, however, specifically marked contours or partial contours of interest for each plane, when given appropriate tools (as in our pilot study). While the resulting proxies can be visually more pleasing, the process is more time-consuming. Hence, for the actual user study, we provided a conceptually simpler and more efficient user interface where users simply picked a unique set of planes and all the intersecting contours formed the planar section proxy.

User interface. The 19 models were displayed to 18 users in random order so that participants were less influenced by prior models when sectioning similar shapes. The users could tumble and zoom the models using a conventional 3D view manipulation interface. Planes were viewed as a plane widget (see supplementary demo) with the section curves overlaid on the model or as a filled planar section without the model. Users often toggled between the views to see the model while placing and evaluating new proxy planes.

Planes could be added, removed, or edited at any time. We provided three means of selecting planes: (i) snapping to axis aligned planes; (ii) starting from one of the axis aligned planes, rotating the plane to refine the azimuthal and polar angles, and translating the plane along its normal direction; (iii) enabling a novel widget where a plane is specified by 3 points than can be interactively moved directly on the surface of the model. The second option was typically used when a *near* axis aligned plane was to be selected, while third option was employed when participants wanted to pick a plane passing through visible features or landmarks. Users reported completing the task in two or three sessions of around 30-45 minutes each and felt the interface provided adequate control for the given task (see supplementary demo) and were satisfied with the abstractions.

3.1 User study findings

SMALL NUMBER OF PLANES SUFFICE? Users remarked that they did not feel constrained by the soft maximum, also evidenced by the user study where only 5 planar-section proxies had more than 10 planes, and a maximum of 14 planes in one case (see Figure 5). The mean and standard deviation of the number of planes used across all users and models were 4.77 and 2.16, respectively (see Figure 3-top blue columns). Thus, we observe that common 3D shapes can be satisfactorily abstracted using a small number of planar sections (question #1). Note that user #15 tended to use more planes than others (mean 8.421, standard error 2.194), but exceeded the soft maximum for only 2 models.

USERS CREATE CONSISTENT PLANAR SECTIONS? Figure 4-top summarizes the results of our study. Each planar-section chosen
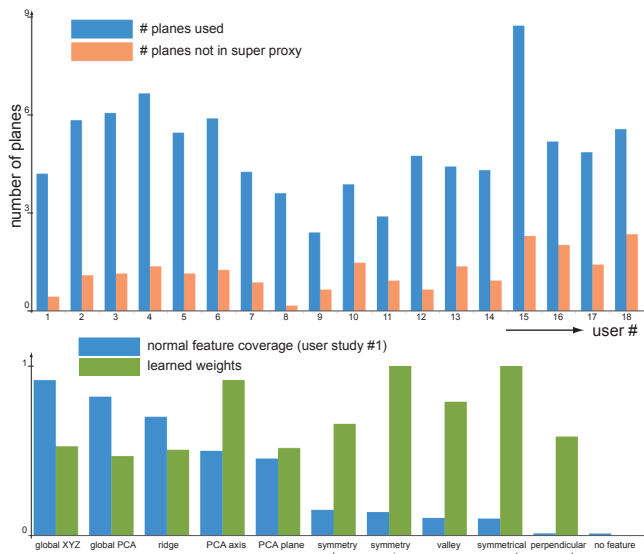

Figure 3: *(Top) For each of the models, there was a clear consistency among the slicing planes prescribed by the users. (Bottom) We observed a strong correlation among the selected planes with geometric features of the models (blue columns), while the corresponding learned weights (see Sections 4 and 5) are in green.*

by a user is overlaid on the model as a faint curve. Thus, lines of greater thickness and opacity indicate agreement among the planar sections selected by different users to represent the same shape (question #2). Next, we quantify this observation.

First, for each model, we group the planes prescribed by the different users into *equivalent* groups based on their mutual distance (in plane-space), and select a representative from each group. Specifically, for a given model, we define a *plane-likelihood* function as the sum of radial sigmoid functions defined around each plane belonging to the user defined proxies for the model. The fall-off radius is defined by the minimum of a feature size 0.1 and the minimum distance between any two planes belonging to the same proxy over all users (0.059). Groups of visually equivalent planes are then represented by the local maxima of this plane-likelihood function, which we evaluate by discretizing plane-space (see Section 4).

Next, we define the *super proxy* simply by picking points in the discrete plane-space whose plane-likelihood is above a given threshold (4.5 in our experiments). Figure 4-middle shows the super proxy on our input data indeed results in a visual proxy similar to many of the user defined proxies. The color of the super proxy indicates their plane-likelihood from hot (red) to cold (blue).

Finally, for each user, we measure the deviation from the general collective as the number of planes in their proxy that do not contribute to the super proxy (see Figure 3-top orange columns). Overall, we observe that a large number of the selected proxy planes are consistent across users.

USERS CAPTURE GEOMETRIC FEATURES? We hypothesize that users prescribe sectioning planes aligned to geometric features of the shapes (question #3). This was verbally reaffirmed post-study by two of our artist users. We also compute representative planes for various geometric features (see Section 5) for the user study objects, and test how well such planes align with the user prescribed planes. Figure 3-bottom indicates that about 1% of the user prescribed planes (17 of 1630) remain unexplained by the chosen geometric features (note that a sectioning plane can be explained by
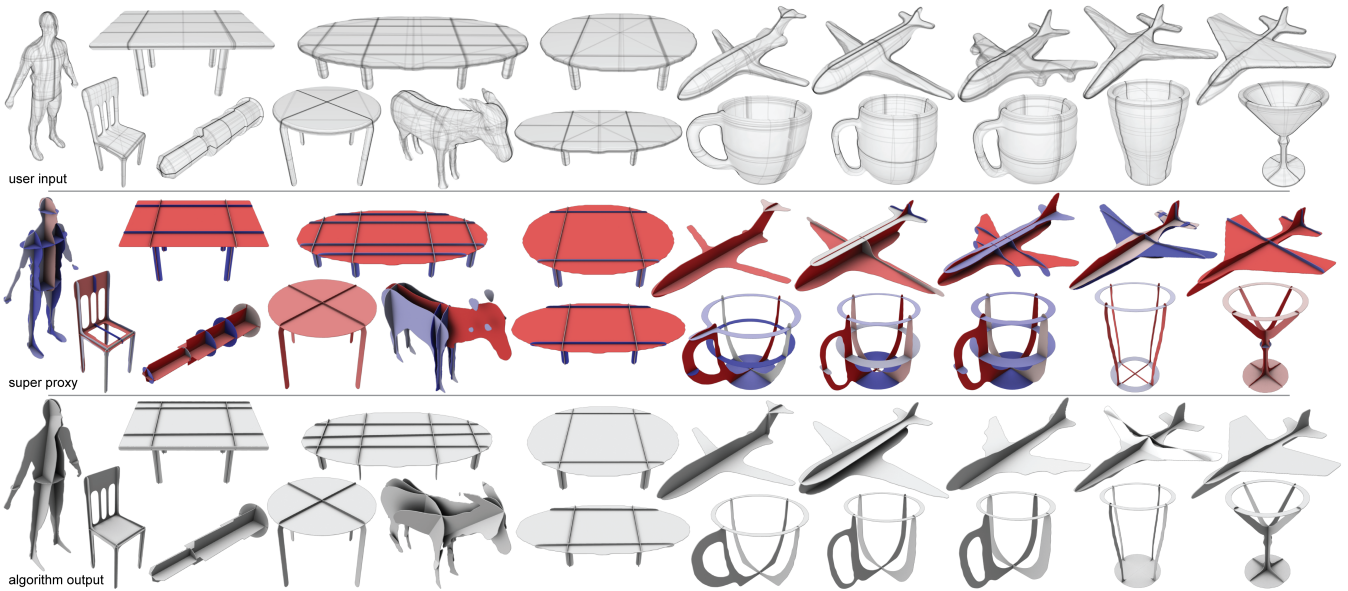
Figure 4: *User study planar section proxies as faint lines aggregated on the models (top) from which a super proxy is computed (middle). The hot (15 users) to cold (4 users) gradient indicates user agreement on the depicted planes. Note the correlation between the proxy created by our algorithm (bottom) and the user agreement on depicted planes (middle).*

multiple features, and hence the column sizes add up to more than unity). The heights of the (blue) columns give a qualitative importance of the features. However, such a reasoning ignores that often the choice of planes influences subsequent planes. In Section 5, we describe a principled approach to learn the importance of the features, while accounting for the inter-dependencies.

3.2 Discussion

Inherent shape knowledge.   In certain cases, the participants use their inherent knowledge of the shapes to select the sectioning planes, without any apparent correlation to the geometry of the models. For example, for the chair shown in Figure 5, all users simply represented the seat with a single planar section despite the geometric concavity at the bottom of the seat (7 of 16 human-authored planes intersecting the seat had a hole, we believe these users did not realize they captured a concavity as this would require inspecting the chair from below). None of the users marked two planes to indicate both the chair seat and the presence of a concavity below, indicating the use of semantic knowledge of a chair in ignoring the concavity.

Of the 17 human-authored planes in the entire study which capture no geometric features we consider, 8 of these planes are for the donkey model (see Figure 5). A number of these user planes roughly
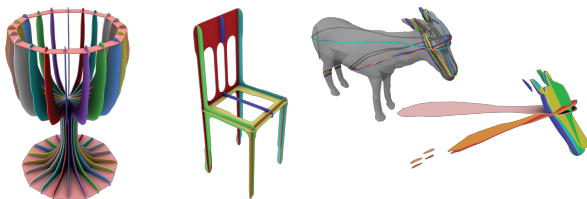

Figure 5: *User proxy problems: (left) regular sections in our pilot study; (middle) inadvertently capturing concave under side of chair; (right) planes covering no geometric feature on the donkey.*

pass through the ears, eyes or where one might imagine the donkey's mouth. The variability of these user planes further indicates the use of model semantics and the lack of a clear visual landmark in selection of the planes. Such cases being sparse (see Figure 3), we focus only on the geometric aspects of the planar proxies, without requiring additional semantic knowledge.

Design choices.   Our study takes about 2-3 hours/user, and so we only enlisted dedicated users in a controlled experimental setup, instead of devising mechanisms to deal with noisy data with high variance. We did consider using Mechanical Turks similar to Cole et al. [2009], but were unable to devise a fair experiment that can be completed in a short time. For the PSB dataset, we observed that from around 12-15 users the super-proxy set, along with the corresponding learned weights (see Section 6), stabilized. Hence, we capped our user study at 18 users.

4 Algorithm

Our goal is to produce a similar proxy $\mathscr{S}$ of shape S using a small number of planar sections, such that S and $\mathscr{S}$ are perceptually equivalent. This is an ambitious goal given the lack of a suitable computational model of perception. Several seemingly natural formulations turn out to be insufficient (see Figure 6): (i) An inefficient but conceptually simple solution is to consider the $\binom{n}{3}$ planes defined by the n mesh vertices (for a poly-mesh model) and choose those maximizing a combination of geometric functions such as, sectional area, length of the section perimeter and total curvature along the section perimeter. While maximizing these functions can locally optimize planar sections, globally they often have no perceptual significance. (ii) Another simple solution is to select axis aligned planes, or principal component (PCA) planes, globally or locally for all segments [Chen et al. 2009]. The results are unsatisfactory on two accounts. First, such axes and planes do not explicitly capture shape symmetries or geometric landmarks, and second, a simple aggregate of these planes ignores inter-plane relationships, which are especially critical when choosing a minimal set of planes. (iii) Another approach is to compute ridge and valley features for
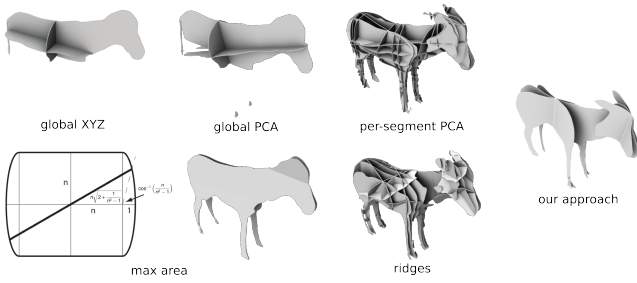
Figure 6: *Simple alternate algorithms give rise to non-intuitive planar slices. (Right) Results from our approach consist of few slices which capture many important mesh features. (Bottom-left inset) A planar section of maximum area seen within a 2D cross-section can cut a shape at an unintuitive angle.*

the mesh, making cuts in the direction of the surface normal. Where ridges and valleys are defined some geometric landmarks will be captured, however, once again the set of planes is neither minimal nor accounts for inter-plane relationships.

Instead, we design our algorithm based on the observed consistency between shape features and planes selected in our user study. We hypothesize that choosing a small number of planes to maximally *cover* the characteristic object features produces a good planar proxy. We show the problem of finding the minimum set of planes that covers a given feature set is NP Hard (see Appendix). Further, the minimum solution is not always perceptually preferable, since humans also factor in geometric relationships between selected planes when defining a proxy. We thus propose a framework where an extendable set of geometric features F (see Section 5) are used to populate a parameterized plane-space. We then iteratively select planes corresponding to the most densely populated regions of plane-space similar to the computation of the super proxy. Selected planes also *re-adjust* the plane-space with respect to covered features, introducing new features to capture geometric relationships between the selected planes and those that will subsequently be selected. Our algorithm runs in two phases: (i) initialization involving discretizing and populating the plane-space using feature set F, and (ii) iteratively selecting planes to cover any uncovered and populated regions of the plane-space.

i) Initialization. We parameterize any plane P in the normal-intercept form using its normal $\mathbf{n}$ and a scalar d (≥ 0) such that any point $\mathbf{p} \in P$ satisfies $\mathbf{n}^T\mathbf{p} + d = 0$. Representing $\mathbf{n}$ in polar-azimuthal form using (θ, ϕ) with $\theta \in [0, 2\pi]$ and $\phi \in [-\pi/2, \pi/2]$, we represent each P in a plane-space spanned by (θ, ϕ, d). We discretize the space of all possible planes by partitioning it into a collection of bins. In our experiments, we sample θ and ϕ every 2 degrees, and d every 0.02 units (all models are centered and normalized to a unit sized box). Initially, all the bins are empty.

We populate the plane-space bins using candidate feature planes. Our framework handles a collection of geometric features $F := \{f_1, f_2, \ldots\}$ extracted from an input mesh. Each feature $f_i \in F$ has an importance weight w_i, which is learned from the user study data set (see Section 5).

Suppose feature f_i is such that it can be defined by the single plane $\mathbf{n}_f^T\mathbf{p} + d_f = 0$. Each point in plane-space (θ_b, ϕ_b, d_b) has a corresponding plane $\mathbf{n}_b^T\mathbf{p} + d_b = 0$ (note the mapping from plane-space to plane is surjective, as at singularities $\phi = -\pi/2, \pi/2$ multiple bins map to the same plane). We define the distance between feature f_i's plane and bin b's plane as

$$dist(f_i, b) := \arccos(\mathbf{n}_{f_i}^T\mathbf{n}_b) + |d_{f_i} - d_b|. \qquad (1)$$

Discretization of a candidate plane to its corresponding bin in plane-space produces aliasing artifacts. Addressing this, we distribute the effect of a feature $f_i \in F$ to a local collection of bins within an influence radius r_i (set to 0.1 in our implementation). We define f_i's coverage of bin b as

$$c_i^b := 1 - dist(f_i, b)/r_i, \qquad (2)$$

for those bins b where $dist(f_i, b) < r_i$, and set $c_i^b = 0$ otherwise. Note that since the model is normalized, we treat the contributions from angular and distance deviations equally. The final weight for each bin b is then taken as the accumulated contribution over all the features as $w_b := \sum_{f_i \in F}(w_i \cdot c_i^b)$. Figure 7-top shows the plane-space after initialization with features from the *human* model with the bins colored according to their respective feature types.

Features can take different forms in the plane-space, based on whether they are one of three types: *point*, *capsule* or *spindle* (see Figure 7-top right). *Point* forms are used to represent feature types described by a single unique plane, such as a symmetry plane. For other types of features, such as the global PCA planes, the plane's normal direction is important but the particular d value is not. To capture this acceptable variation in d (i.e. ensure intersection with mesh segments relevant to the feature), we use the *capsule* form in plane-space. To create this form, we use a slightly modified form of Equation 1, choosing a d_f value in the acceptable range closest to d_b to minimize the distance. Finally, feature types with a dominant axis such as segment PCA axis features and symmetry axis features we capture using the *spindle* form. For this form we allow variation in the plane's normal (varying d accordingly so the entire axis intersects the plane). We again rely on a modified form of Equation 1 where we choose a normal n_f that is both orthogonal to the axis and minimizes distance to bin b's plane. Our system can be extended to support new feature types with minimal effort by encapsulating them with one of these three forms.

ii) Covering the plane-space. Next, we select planes to cover the populated bins $\mathscr{C}$. We maintain the current set of selected bins in $\mathscr{P}$ starting with an empty set $\mathscr{P} \leftarrow \emptyset$. In each iteration, we choose the bin $b^* \in \mathscr{C}$ with maximum coverage value, provided w_{b^*} is above a threshold weight, set to 1 in our experiments. The chosen bin is added to the current set $\mathscr{P} \leftarrow \mathscr{P} \cup b^*$.

We then *adjust* the weights of the remaining bins to account for the selection of bin b^*. For each feature $f_j \in b^*$, we adjust the weights of each of the *other* bins containing f_j. Specifically, we identify bins $b_i \in \mathscr{C}$ that are touched by feature f_j, and for each we remove its contribution to weight w_{b_i}: $w_{b_i} \leftarrow (w_{b_i} - w_j \cdot c_j^{b_i})$.

5 Shape Features

We describe our user-study guided feature set selection, while noting that other features can easily be incorporated if they take one of our plane-space forms. We learn the relative feature weights using user study data to capture their relative perceptual importance.

Principal planes. The PCA axes of the shape and of the individual segments define the oriented bounding box of the shape and its segments, respectively. One or more of these planes captures the principal directions. Based on the relative strength of the principal directions (eigenvalues of the covariance matrix), we add features as follows: (i) for a model with a single dominant principal axis, we use a single spindle form for the axis, (ii) for a model with two dominant directions, we use a single plane feature (capsule) with normal pointing in the direction of the least significant eigenvector, otherwise (iii) we use 3 planes (capsules), each having a normal in the direction of each eigenvector. In the case of segment PCA
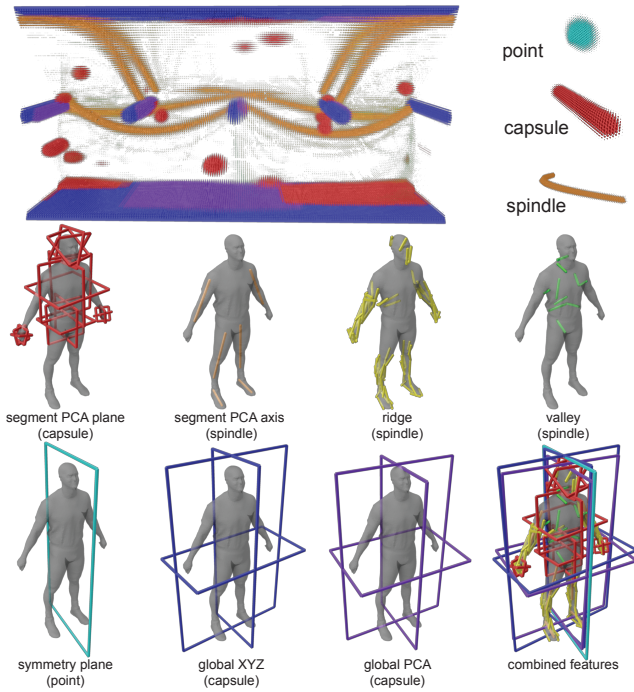
Figure 7: *Plane-space (top) after initialization, populated with the set of features (bottom) detected for the human model, has three coverage forms: point, capsule and spindle. Learned weights scale the importance of each feature type, and we visualize this by scaling the size of each plane-space bin.*

planes which take capsule forms, we constrain the range of d value so that the segment will be intersected.

Global planes. Since the initial database models are mostly oriented, the global X, Y, Z coordinate axes help to ground the shape and fix its orientation with respect to the environment. Their representative planes, like PCA planes are represented using capsule forms and can thus vary in d value.

Ridge and valley curves. These curves represent extrema of surface curvature on the shape, can be efficiently computed for detecting multi-scale features, and most importantly are believed to capture geometrically and perceptually salient surface features. We detect ridge and valley curves using [Ohtake et al. 2004]. A greedy algorithm splits these curves into line segments, treating each as a spindle feature and giving each a weight based on segment length. Since each segment is a spindle feature, a coplanar group common to a larger ridge or valley feature (e.g. a ridge along the rim of a cup) will intersect in plane-space (with maximal coverage of these ridge segments). In our implementation we chose to use a strict threshold for detecting ridges and valleys, instead of weighing each by a measure of strength (e.g. average principal curvature along the curve). By filtering out many potential ridges and valleys whose average curvature is low, we avoid the processing associated with adding these relatively insignificant features to the plane-space.

Symmetry planes and axes. Symmetry is strongly connected to shape abstraction and perception, and often is persistent across variations in object collections relating to part hierarchies [Simari et al. 2006]. We consider global reflective and rotational symmetries in our framework. We detect global symmetries [Mitra et al. 2006] for the entire shape. Reflective symmetries are captured well by a planar section defined by the symmetry plane itself. The symmetry is also visually clear in the shape contour of a planar section that is

perpendicular to symmetry plane. We add symmetry planes to the plane-space as point forms, and symmetry axes as spindle forms.

Perpendicular supports. We observe from the user-study that it is desirable to capture *spindle* features like a PCA axis or global symmetry axis of a model (see Figure 7) using a pair of orthogonal planes, with the spindle being their intersection line.

If the maximal bin b^* covers a spindle feature with axis direction **a** and intersection point **p**, we add a perpendicular support feature f_{perp}. Feature f_{perp} defines a plane feature that also intersects the axis, but in a direction perpendicular to the plane defined by b^*. The parameters for our supporting feature $\mathbf{n}_{perp}$ and d_{perp} are given by:

$$\mathbf{n}_{perp} = \mathbf{n}_{b^*} \times \mathbf{a}_f \quad \text{and} \quad d_{perp} = -\mathbf{p}^T \mathbf{n}_{perp}.$$

If d_{perp} is negative we flip the signs of both $\mathbf{n}_{perp}$ and d_{perp} to keep all d values non-negative. Perpendicular supports are point forms in plane-space. Note that each spindle feature is allowed to generate a maximum of one perpendicular support.

Symmetric supports. In order to retain planar symmetries of the model, if we select a plane that passes through a symmetric segment that is not almost perpendicular to the symmetry plane ($< 80°$ in our implementation), we add its reflected plane, if not already selected, as a point-form feature to plane-space to increase the likelihood that it will be selected.

Learning weights

In absence of a computational model for evaluating planar abstractions, we use the user study findings to learn relative weights for the various geometric features. We do this in two parts: (i) we obtain a representative set of planes called the *super proxy*, and (ii) we derive a large number of linear inequalities from this set, and apply a constrained optimization solver to learn the feature weights.

First, for each model, we form a set of planes called the super proxy using the planes chosen by the 18 users (see Section 3).

Next, we form a linear system to apply the solver by mimicking our algorithm, i.e., the selection process is the same – users add planes one by one, in order of maximal feature coverage, while the value of each plane is sufficiently high. For the super proxy this importance ordering of planes is captured by their user agreement value (see Section 3). We derive two types of inequalities. The first type are for each plane in the super proxy. For the 10 feature types in our system, using a plane selection threshold of 1, super proxy plane inequalities take the form:

$$\sum_{i=1}^{10} w_i C_i \geq 1. \tag{3}$$

Note that these inequalities are created *in order* of plane selection since plane selection influences subsequent inequalities (i.e. perpendicular and symmetric support features may be introduced). The left-hand side is the value of each super proxy plane, and is the sum of the total coverage C of each of the 10 types of features multiplied by the unknown weights. Each C_i is calculated by summing the coverages across each feature of type i in the model: $C_i = \sum_j c_j f_j$, where each j indexes a feature of type i, c_j is the coverage of feature j by the plane (Equation 2), and f_j is a feature-specific scalar (e.g. for ridges, f_j scales by the length of ridge j).

Generally, not every feature is captured by the super proxy. After processing each plane in the super proxy set and covering each of the mesh features in the plane space, there may be (and usually always are) a number of bins containing one or more features. These bins correspond to all planes that were not selected, and so their

Figure 8: *Algorithmic results on a variety of models from the PSB (see supplementary for full set). Note the consistent behavior on cups, tables, and other models similar to those in the user-study (see Figure 4) as well as completely novel model examples like heads, vases and birds.*

total value in terms of the features they cover must be below our threshold. For each of these bins, which map to a plane in model space, we have the second type of inequality:

$$\sum_{i=1}^{10} w_i C_i < 1. \tag{4}$$

We now choose weights for the features such that they satisfy all the inequalities across all models in the study. Our study consisted of 18 users and 19 models, and a total of 1630 planes were authored by all users across all models. Of these, 101 planes were selected as the representative super proxy set across all models. This set of planes formed 101 inequalities of the first type. Most of the 1630 user authored planes are captured by the super proxy, indicating consistent plane selection (see Figure 4). Our system contained 107992 inequalities of the second type, one for every bin not selected by the super proxy planes for each model.

A simple solution can involve a least squares approach to best meet the inequality constraints. However, in our case we are interested in satisfying the inequality constraints, rather than forcing equality relations, e.g., we do not differentiate between relations once they cross the threshold value. The least squares approach resulted in many zeroed out coefficients as the inequalities were driven towards equalities. Such an overfitting does not help as the set of weights do not generalize in producing useful results on new models, especially in the case where new models contain features that may be under-represented in the training set.

Simultaneously satisfying all of the inequalities is infeasible. We found numerous cases where planes in our super proxy did not have strong feature coverage. These planes corresponded to inequalities of the first type where the sum of total coverages $\sum_{i=1}^{10} C_i$ was low. Intuitively, as these planes do not cover features well, they are not very well suited for deriving feature weights. To address this, we progressively removed inequalities derived from the super proxy with low total coverage, until a solution was feasible. When $\sum_{i=1}^{10} C_i > 1.5$, the system becomes feasible and we use 74 inequalities from the super proxy set. For many planes in the super proxy, we observe a strong correlation between feature coverage and user agreement (see Figure 3). Interestingly, our algorithm

uses the learned weights to consistently pick planes of high user agreement, while leaving out planes of low user agreement (see Figure 4).

To regularize the solution, we use an objective function for the solver that minimizes the difference of each weight from 1 in order to be well balanced across each type of feature, specifically, $\sum_{i=1}^{10} (w_i - 1)^2$. To determine the weights (see Figure 3) we used MATLAB's 'fmincon' function, a constrained optimization solver which relies upon the active set method [Powell 1978].

6 Evaluation

In order to evaluate our algorithm, first we used the learned weights to recreate the planar slices for the 19 models from the training set (see Figure 4-bottom). Note the general correlation between the planes chosen by our algorithm and the level of user agreement for planes belonging to the super proxy.

In a more demanding test, we used the learned weights to create planar slices for all the models from the PSB. Figure 8 shows models similar to those in the user study dataset and is a validation of the learned weights, as well as showing visually compelling results of our algorithm on many new models (see supplementary material). As our goal is to produce perceptually sound planar slice abstractions, we performed another user study to judge the quality of the produced results, as described next.

Shape recognition user study. We conducted a user study involving 66 users, with ages ranging from 16–50, where each participant was shown one of 5 representations randomly selected from 11 models (6 from initial user study, 5 new without human annotation) which were shown in randomized order (see supplementary material). Users were asked to identify the model, and their response time was recorded (up to 5 seconds). Each model representation was rendered from the same pre-selected viewpoint and was either: a mesh rendering, algorithm generated planar slices, human annotated planar slices, global PCA planar slices, or random planar slices. Our results for response times and recognition rates are summarized in Figure 9.

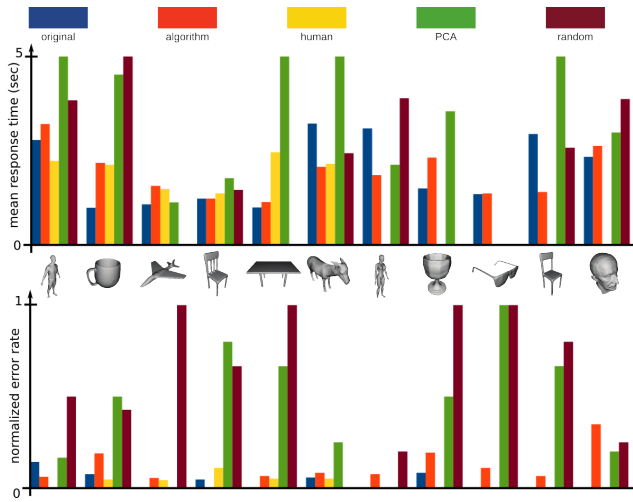For the mesh rendering, algorithm generated planar slices and hu-

Figure 9: *(Top) Average recognition times of objects and planar section proxies, for each of the representations. (Bottom) Normalized error rates in identifying objects across each representation.*

Figure 10: *Persistence of model perturbation: model (center) under increasing Gaussian noise moving left* 0.005, 0.01 *standard deviation and increasing decimation moving right* 20%, 40%.

man annotated planar slices, recognition rates were consistently high ($>90\%$) and reaction times consistently low (less than 2 seconds), averaged over all 11 models and all users. This confirms two of our predictions: planar slices are a recognizable abstraction of common objects, and that our algorithm produces comparable results to human annotations. An interesting exception was the Max Planck model (model #317 in the PSB, rightmost model in Figure 9), where recognition rates were between 66–80% for all planar slice abstractions but 100% for the mesh rendering. In other occasional failure cases, users marked the 'donkey' as a 'pig' or 'sheep', or a 'vase' as an 'ashtray'. Absence of a scale and no prior information about what to expect partially explains such erroneous entries. The global PCA planar slice and random planar slice representations were notably much worse, with average recognition rates of 57% and 38%, and average reaction times exceeding 4 seconds (twice that of the algorithm generated and human annotated abstractions).

Persistence of abstraction under model perturbation. Persistence and resilience to shape resolution and perturbation are important properties of shape abstractions. In our framework, this resilience is largely handled by the algorithms that extract our set of geometric features and some degradation in the quality of their performance is expected. Figure 10 shows that our algorithm produces perceptually persistent proxies when applied to decimated or noisy versions of the original mesh. As expected, planes of lesser importance are perturbed or excluded, since these features are not dominant. Also, the creation of many spurious features (e.g., ridges emerging from noise applied to vertex positions) is unlikely to result in new planes, as their weight are scaled by their lengths.

Planar-section aesthetics. Viewers generally found our algorithmic results in Figures 4 and 8 to be of high quality and recognizable as shape abstractions. Despite this, neither our result nor the human-derived super proxy is as magical as some of our motivational images (see Figure 2). We believe the major reasons are: (i) an intangible artistic eye for composition that we did not set out to capture; (ii) our results echo the PSB, which was designed not for aesthetic reasons but to capture a range of shapes with consistent mesh topology, resolution and size; (iii) sections as seen in the motivational images are partial planar sections of the mesh, (iv) artistically created planar section contours sometimes deviate from the planar section to partially conform to the shape silhouette.
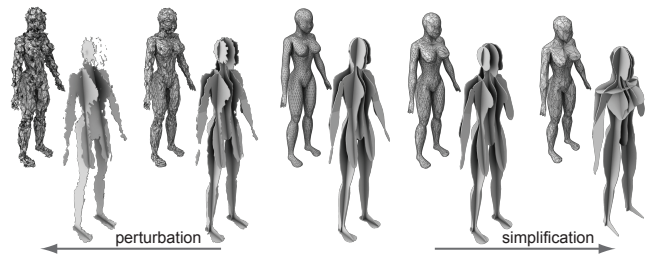
Our program (supplementary material) can be used as an interactive tool enabling artists to compose their own shape abstractions (see Figure 13-top left). We chose not to require our users to define partial planar sections to keep our user interface simple and the user study manageable. For fair comparison, our results in Figures 4 and 8 are shown as complete and unprocessed planar sections. We do, however, address the automatic computation of partial sections in three ways: (i) We note that segmentations in objects often indicate rigid parts that can articulate with respect to each other. We capture this property within our abstraction by cutting the planar section contours at segmentation boundaries and capping them with tangent-continuous cubic Hermite splines into multiple overlapping but topologically disjoint planar sections. (ii) We only retain portions of the contours that pass through segments whose features are covered by the given plane, and cap them smoothly into closed contours. (iii) We observe that planar sections provide the strongest shape cue along the section contour where the surface normal of the shape is near perpendicular to the plane normal. We verified this by examining the distribution of the angle between the surface normal along the section contour and the plane normal for the super proxy planes (e.g., $>70\%$ of curve length has angle $>70°$, see supplementary material). We can thus retain and cap portions of the section contour where the angle between the surface normal and the plane normal exceed a specified range. While it is straightforward to allow users to interactively edit parts of section contours to conform to model silhouettes, we leave the automatic inference of such hybrid planar sections to future work.

Optimizing selected planes. We propose three simple approaches to optimize the generated proxies (illustrated in Figure 11): (i) Plane refinement: The discretization of plane-space can cause selected planes to minimally deviate from the features they capture. A simple bounded linear search through plane-space in the vicinity of a selected plane to maximize section area or perimeter improves the visual quality of the planar section. In our im-
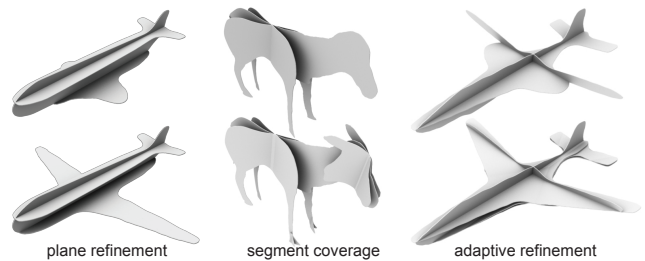

Figure 11: *Planar sections (top) can be optionally optimized to improve abstraction quality (bottom). From left to right: locally maximizing section area, using additional planes for uncovered segments, reducing weight threshold to choose additional planes.*

plementation, this optimization is always performed. (ii) Segment coverage: Our algorithm does not guarantee that all segments get covered. A user may optionally enable algorithmic plane selection to continue beyond the threshold only for planes that section uncovered segments. (iii) Adaptive thresholding: The user may wish to include a desirable plane below the default threshold that the algorithm did not include, or have a specific number of planes in mind. Tuning the threshold allows users to refine the planar proxies created by our algorithm to select more (or fewer) planes.

Performance. We include a demo (code will be made public) of our OpenGL/Qt application. The running time of our algorithm is dominated by the number of features detected for a given model as populating plane-space for each feature is expensive and scales linearly (typically 50-200 features were detected on models from the PSB). Our algorithm typically takes 5-10 seconds for most models on a 2.8GHz AMD Phenom II processor.

Limitations. While our approach produces reasonable results and is extendable, it has some fundamental limitations, which we illustrate in Figure 12. Our algorithm is only as good as the quality of features it is able to extract. Excessive and noisy features (the armadillo), or the absence of features (e.g., bust) can result in too many or too few planes being selected. Articulated figures can be problematic since features do not line up well along planes, and produce disjoint abstractions (e.g., right arm of running woman). Excessively curved structures like the chair with a curved backrest are not ideal for representation with a minimal set of planar sections. Despite these failures, we believe that planar section proxies are a conceptually useful abstraction and that our algorithm provides a worthy first attempt at their construction.
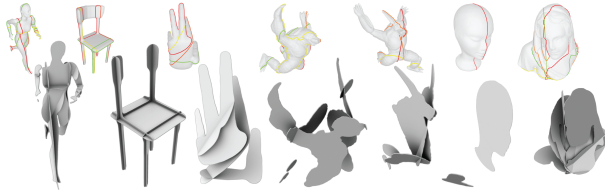

Figure 12: *Limitations: Imperfect results for articulated poses, models with highly curved characteristics components, or models that are overly smooth lacking clear geometric features.*

7 Discussion

Potential applications. Planar section proxies have numerous applications, each of significant scope, a few of which we briefly show to illustrate the practical potential of this shape abstraction.

i) Paper statues and puppets are appealing figure abstractions that can be physically constructed from planar section proxies (see Figure 13). A paper statue is made by physically printing, cutting and assembling the 2D contours of a planar section proxy. Our program automatically marks the intersection line between two planar sections on each section to be cut from opposite sides so they can be interlocked as shown in Figure 13. Interactive control on the meeting point and direction of cut on the planar sections makes assembling multiple intersecting planes easier. All cuts are color coded to aid assembly. We are also able to create articulated puppets by automatically cutting section contours at segment boundaries and creating Hermite curve capped and overlapping partial contours. These contours are physically pinned in the middle of their overlap creating a hinge joint. The car in Figure 13-top left, additionally connects section contours for the wheels with an axle and floating section contours in the front of the car to the car body with struts.
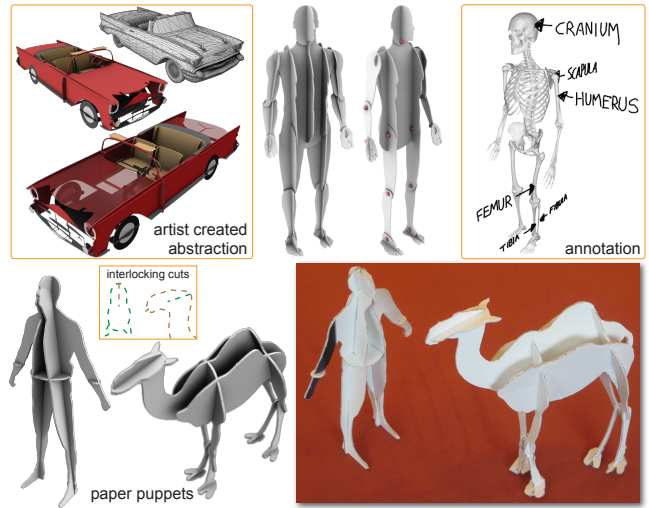

Figure 13: *Planar slices produced by our algorithm can be used for artist created abstractions, annotations, or for paper puppets.*

Partial sections and the various connectors for the car were interactively authored in about 15 minutes from the default planar section proxy. A comprehensive approach to creating complex multi degree of freedom functional models is subject to future work.

ii) Poly-postors. Planar section proxies are also invaluable as polypostors for lightweight rendering of crowds [Kavan et al. 2008]. A texture is created for the front and back sides of each plane by rendering the model orthographically for each direction (see Figure 13 top-left). Our algorithm can create effective poly-postors by selecting planes below the default threshold until the difference between the silhouette of the poly-postor and the input model for a set of views is acceptable.

iii) 3D shape annotation is an important problem in scientific visualization and education. While it is easy to project 2D strokes onto the underlying 3D shape in a given view, often annotation occurs in the space proximal to the 3D shape to indicate dimensions or to avoid obscuring the region being annotated. In such scenarios [Schmidt et al. 2009], the annotation has no 3D representation and becomes meaningless when the current view is changed. An invisible planar section proxy, however, provides a natural set of 3D planes onto which annotations can be projected based on the alignment of the plane to the current view, and the proximity of the projected annotation to the section contour (see Figure 13-top right). While the figure illustrates the 3D-ness of the annotations with excessive foreshortening, in practice past an oblique angle of 30°, text annotations billboard in place or rotate about spindle feature planes to improve readability.

iv) Compact representation. Planar section proxies, being comprised of a small set of planes and 2D contours or image masks in those planes, are computationally compact. The surface reconstruction algorithm of [Liu et al. 2008] naively applied to a few planar section proxies shows both the potential to reconstruct the original 3D shape and issues of topology that must be addressed by such a surface reconstruction algorithm (see Figure 14).

Conclusion and future work. Inspired by the consistency of abstractions produced by human subjects during our initial user study, we developed a computational approach to create geometric feature-guided abstractions of man-made shapes using only a handful of planar sections. The formulation is flexible and directly incorporates a variety of shape features. In the absence of
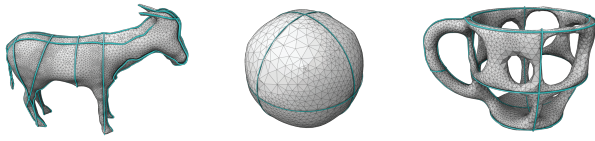
Figure 14: *3D shape reconstructed from planar section proxies (overlaid in blue) using [Liu et al. 2008].*

any suitable computational model of human perception, we learned the relative preference weights for various features using our user study data. We tested our algorithm on a variety of input models, and presented robustness results with respect to noise and varying mesh resolution. Finally, in the course of a conducted survey, we observed that both recognition quality and efficiency of abstracted models are comparable to those of the source models.

Although recent efforts in computer graphics have investigated relations and importance of 3D geometry to perception and recognition, a lot remains to be done before we develop a good understanding of the complex workings of our recognition system. Our work using planar section abstractions is a small step in this direction. In the future, we plan to explore possibilities of directly including additional requirements, e.g., view coverage, importance weighted abstraction, as well as other feature types such as curve skeletons, into our framework. In terms of applications, it will be interesting to explore the usage of planar proxies to shape recognition, particularly in authoring systems for retrieving model parts from the web, where predominantly, keyword based searches are currently employed. Our abstractions, though stable under small changes, usually change geometrically and topologically under larger shape variations, hinting at the need for an appropriate perceptual distance measure for comparing different planar abstractions. Finally, we plan to explore planar proxies towards interactive 3D sculpturing.

Appendix: Minimum Planar Section is NP-Hard

We show that finding a minimum planar section (MPS) that covers a set of shape features is NP-Hard by reducing the classic NP complete minimum vertex cover (MVC) problem to MPS in polynomial time. Formally, given a graph (V, E) a k-cover is a set of k vertices in V such that every edge in E has at least one incident vertex in the set. The MVC problem is to find a k-cover with the smallest k.

We construct an input shape to the MPS problem as a collection of $|E|$ disjoint cylinders. The shape features are thus the $|E|$ cylindrical axes (the heights and radii being irrelevant). We will construct every cylindrical axis C_e to correspond to an edge $e \in E$ such that:
Property 1: Axes C_e and C_f are co-planar *iff* e and f are adjacent. These axes will be defined as the intersection lines of $|V|$ planes representing the vertices of the graph. We first define the plane normals such that:
Property 2: Any three normal vectors n_i, of planes $i \in 1..|V|$, are linearly independent. The set of vectors $n_1..n_{|V|}$ arranged in a cone around the Z axis, where $n_i = < cos(2\pi i/|V|), sin(2\pi i/|V|), 1 >$, satisfies this property. Thus any axis $C_{(i,j)} = n_i X n_j$ is non-zero. Given distances to the origin d_i and d_j for planes i and j we can

fix a point $p_{(i,j)}$ on $C_{(i,j)}$ as $p_{(i,j)} = \frac{(d_j * (n_i.n_j) - d_i)n_i + (d_i * (n_i.n_j) - d_j)n_j}{(n_i.n_j)^2 - 1}$. Any point $p(t)$ on $C_{(i,j)}$ is thus $p(t) = p_{(i,j)} + (n_i X n_j)t$. The intersection of axis $C_{(i,j)}$ and $C_{(j,k)}$ is now defined by $d_i = n_i.(p_{(j,k)} + t * (n_j X n_k))$. We define *intersect*$(i,j,k)$ to be the parameter t along axis $C_{(j,k)}$ where it intersects $C_{(i,j)}$. Rearranging *equation 1*: *intersect*$(i,j,k) = (d_i - n_i.p_{(j,k)})/n_i.(n_j X n_k)$. Note that $n_i.(n_j X n_k) \neq 0$ by *Prop. 1*. The distances to origin d_i for the planes $i \in 1..|V|$ is fixed such that:
Property 3: The intersection axes $n_i X n_j$ and $n_k X n_l$ of any four distinct planes i,j,k,l do not intersect. We perform this assignment iteratively through the planes i from $1..|V|$ setting distance d_i as: Enumerate every permutation of planes j,k,l whose d values have been set (i.e. $j,k,l < i$). For each permutation j,k,l we compute the d for plane i at which $C_{(i,j)}$ and $C_{(k,l)}$ would intersect. We add this value of $d = n_i.p_{(j,k)} + intersect(l,j,k) * (n_i.(n_j X n_k))$ (from *equation 1*) to a set D_i of forbidden d values for plane i. We then set d_i to be any arbitrary value not in D_i.

It is easy to see *Property 1* holds for the above construction: If axes C_e and C_f are adjacent at a vertex v_i, they lie in plane i and are thus co-planar. Conversely, any co-planar axes must either be parallel or intersect. By *Property 2* no two cylindrical axes are parallel and by *Property 3* two axes can only intersect if they are adjacent.

Lemma 1: Any set of k planes that cover the cylindrical feature axes of this shape is equivalent to a k-cover of the graph. *Proof:* Given a k-cover its easy to see that if we pick the planes corresponding to the vertices in the cover, we will cover all feature axes corresponding to the graph edges. Given a set of k planes that cover the $|E|$ symmetry axes, we note by *Property 1* that for any plane to contain two or more axes, the edges corresponding to the axes must be adjacent and the plane must correspond to their common incident vertex. We add this vertex to our vertex cover. Finally, any plane that contains only one symmetry axis $C_{(i,j)}$ can be rotated around $C_{(i,j)}$ to match the vertex plane i or j and we can add either v_i or v_j to our vertex cover. As a consequence of *Lemma 1* and our $O(n^4)$ reduction, MPS=MVC.

References

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segment. In *ACM SIGGRAPH*, 1–12.

CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1997. A comparison of mesh simplification algorithms. *Computers and Graphics 22*, 37–54.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approx. In *In Proc. SIGGRAPH*, 905–914.

COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where do people draw lines? *ACM SIGGRAPH 27*, 3.

COLE, F., SANIK, K., DECARLO, D., FINKELSTEIN, A., FUNKHOUSER, T., RUSINKIEWICZ, S., AND SINGH, M. 2009. How well do line drawings depict shape? *SIGGRAPH 28*, 3.

DE GOES, F., GOLDENSTEIN, S., DESBRUN, M., AND VELHO, L. 2011. Exoskeleton: Curve network abstraction for 3d shapes. *Computer and Graphics 35*, 1, 112–121.

DÉCORET, X., DURAND, F., SILLION, F. X., AND DORSEY, J. 2003. Billboard clouds for extreme model simplification. In *In Proc. SIGGRAPH*, 689–696.

ECKER, A., KUTULAKOS, K., AND JEPSON, A. 2007. Shape from planar curves: A linear escape from flatland. In *Computer Vision and Pattern Recognition*, 1–8.

EDWARDS, B. 2002. *The New Drawing on the Right Side of the Brain*. Tarcher.

FELDMAN, J., AND SINGH, M. 2005. Information along contours and object boundaries. In *Psych. Review*, vol. 112, 243–252.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iwires: an analyze-and-edit approach to shape manipulation. *ACM SIGGRAPH 29*, 3, 1–10.

KALOGERAKIS, E., SIMARI, P., NOWROUZEZAHRAI, D., AND SINGH, K. 2007. Robust statistical estimation of curvature on discretized surfaces. In *EUROGRAPHICS*, 13–22.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3d mesh segmentation and labeling. In *ACM SIGGRAPH*, 102:1–102:12.

KAVAN, L., DOBBYN, S., COLLINS, S., ŽÁRA, J., AND O'SULLIVAN, C. 2008. Polypostors: 2d polygonal impostors for 3d crowds. In *In Proc. I3D*, 149–155.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. *ACM SIGGRAPH 27*, 3, #75, 1–9.

KOENDERINK, J. J. 1990. *Solid shape*. MIT Press, Cambridge, MA, USA.

LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. *Analysis, Reconstruction and Manipulation using Arterial Snakes 29*, 6, 152:1–152:10.

LI, X., SHEN, C., HUANG, S., JU, T., AND HU, S. 2010. Popup: Automatic paper architectures from 3d models. *ACM SIGGRAPH 29*, 4.

LIU, L., BAJAJ, C., DEASY, J., LOW, D. A., AND JU, T. 2008. Surface reconstruction from non-parallel curve networks. *CGF EUROGRAPHICS 27*, 2, 155–163.

MEHRA, R., ZHOU, Q., LONG, J., SHEFFER, A., GOOCH, A., AND MITRA, N. J. 2009. Abstraction of man-made shapes. *ACM SIGGRAPH Asia 28*, 5, 1–10.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM SIGGRAPH 25*, 3, 560–568.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. In *In Proc. SIGGRAPH*, 609–612.

POWELL, M. 1978. *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, vol. 630. Springer Verlag.

SCHMIDT, R., KHAN, A., SINGH, K., AND KURTENBACH, G. 2009. Analytic drawing of 3d scaffolds. In *ACM SIGGRAPH Asia*, 149:1–149:10.

SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Symp. on Geo. Proc.*, 111–119.

SINGH, K., AND FIUME, E. 1998. Wires: a geometric deformation technique. In *In Proc. SIGGRAPH*, 405–414.

STEVENS, K. 1981. The visual interpretation of surface contours. In *AI*, vol. 17, 47–73.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann Publishers Inc.

SWEET, G., AND WARE, C. 2004. View direction, surface orientation and texture orientation for perception of surface shape. In *Graphics Interface*, 97–106.

TODD, J., AND REICHEL, F. 1990. Visual perception of smoothly curved surfaces from double-projected contour patterns. In *J. of Exp. Psych.: Human Percep.and Performance*, vol. 16, 665–674.

WILLIS, K. D., LIN, J., MITANI, J., AND IGARASHI, T. 2010. Spatial sketch: bridging between movement and fabrication. In *In Proc. Tangible, embedded, and embodied interaction*, 5–12.