Photo-Inspired Model-Driven 3D Object Modeling

Kai Xu[*†] Hanlin Zheng[‡] Hao Zhang[†] Daniel Cohen-Or[§] Ligang Liu[‡] Yueshan Xiong[†]

[*]National University of Defense Technology [†]Simon Fraser University [‡]Zhejiang University [§]Tel-Aviv University

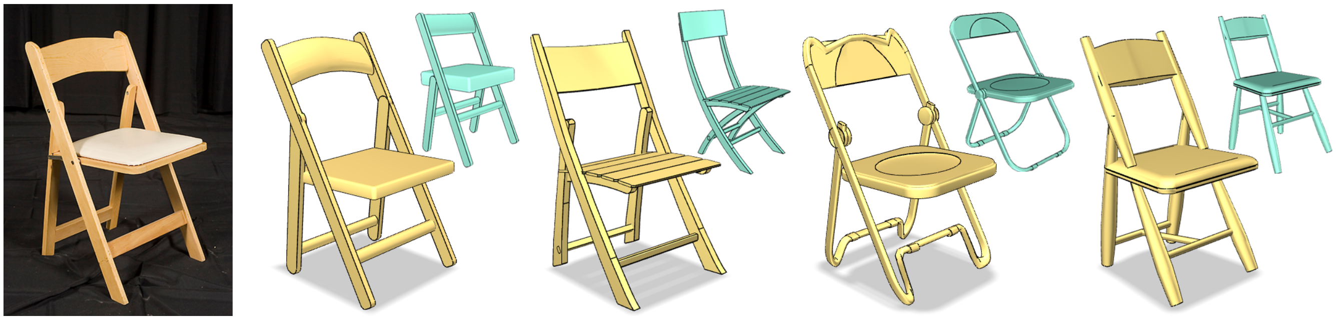

Figure 1: *Photo-inspired 3D modeling of a chair from four different 3D candidates (cyan). The new models (yellow) are created as geometric variations of the candidates to fit the target object in the photo while preserving the 3D structure of the candidates.*

Abstract

We introduce an algorithm for 3D object modeling where the user draws creative inspiration from an object captured in a single photograph. Our method leverages the rich source of photographs for creative 3D modeling. However, with only a photo as a guide, creating a 3D model from scratch is a daunting task. We support the modeling process by utilizing an available set of 3D candidate models. Specifically, the user creates a digital 3D model as a geometric variation from a 3D candidate. Our modeling technique consists of two major steps. The first step is a user-guided image-space object segmentation to reveal the structure of the photographed object. The core step is the second one, in which a 3D candidate is automatically deformed to fit the photographed target under the guidance of silhouette correspondence. The set of candidate models have been pre-analyzed to possess useful high-level structural information, which is heavily utilized in both steps to compensate for the ill-posedness of the analysis and modeling problems based only on content in a single image. Equally important, the structural information is preserved by the geometric variation so that the final product is coherent with its inherited structural information readily usable for subsequent model refinement or processing.

Links: DL PDF WEB VIDEO

1 Introduction

Content creation in 3D is one of the most fundamental tasks in computer graphics. The ultimate goal is to allow artists and even novice users to quickly turn a design concept into a digital 3D model. Creativity is often called upon during design and modeling and as such the user needs to be inspired [Chaudhuri and Koltun 2010]. The inspiration may arise from pure imagination, but more often than not, it can trace its origins to one or more existing concepts with the end product being a variation or composition from one or more existing models [Funkhouser et al. 2004; Kraevoy et al. 2007; Lee and Funkhouser 2008; Chaudhuri and Koltun 2010]. Moreover, the modeling process does not end with an initial creation — the created 3D model is meant to be subsequently refined and manipulated. It is therefore highly desirable for the created model to be *readily usable* for such further processing.

In this paper, we introduce an algorithm for creative 3D modeling where the user is inspired by a single photograph and the creation process is driven by an available set of 3D candidate models. Specifically, the user creates a realistic and readily-usable digital 3D model as a geometric variation from one of the 3D candidates. We focus on the modeling of man-made objects. Even within the same object class, man-made objects often exhibit immensely rich shape variability (e.g., consider all the chairs, tables, or lamps we encounter) which provides the modeling challenge.

Photographs provide perhaps the richest source of creative inspiration. They are easy to find and acquire and the captured objects are shown in their natural appearance and surroundings to provide the most inspiring modeling context for the user. Requiring only a single photograph instead of captures from multiple views allows direct use of the vast source of images that are already available on-line or elsewhere. However, with only a photo as a guide, creating a 3D model from scratch is a daunting task. We support the modeling by utilizing a set of 3D candidates. The created model is a geometric variation from the set, obtained by deforming a candidate model so that its *silhouette* in the appropriate view matches that of the target object in the photograph; see Figure 1.

In our setting, each candidate model has been pre-analyzed so that its geometry representation is endowed by high-level structural information to drive the object creation process. On one hand, the geometry and structure of the 3D candidates can effectively guide object analysis within the photograph. More importantly, the deformation applied to the chosen candidate is *structure-preserving* — it retains the structural information in the candidate so that the produced variation remains coherent and readily usable. At the same time, the high-level structure of the candidate is exploited to provide extra constraints including symmetry to alleviate the ill-posedness

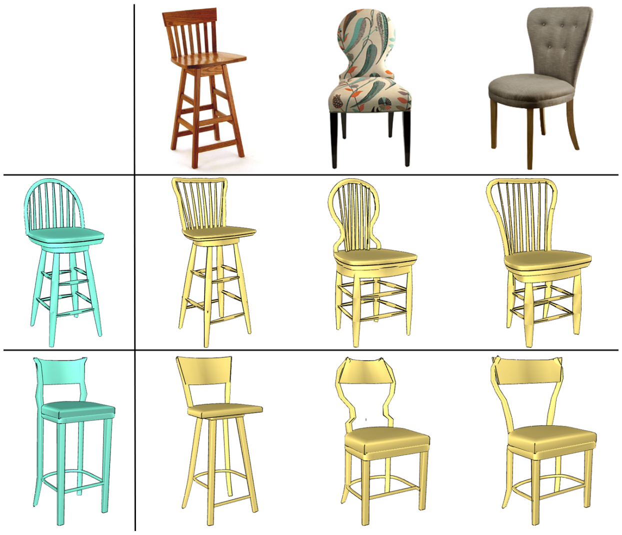

Figure 2: *A matrix of photo-inspired creations (rows: 3D model candidates; columns: photographic inspirations).*

of the modeling problem based only on a single photograph.

Creative modeling from a shape repository is not new. Modeling by example [Funkhouser et al. 2004] and recent works on sketch-based modeling, e.g., [Shin and Igarashi 2007; Lee and Funkhouser 2008], rely on part retrieval from large 3D shape databases followed by cutting and gluing to compose new models. The key point to make is that assembling the retrieved parts into an integral and readily usable model is difficult. Regardless of any stored structure attributes in the source models, any high-level structure of the assembled object must be re-discovered through post-analysis.

Overview. The input to our algorithm consists of a single photograph containing the target object O and a pre-analyzed set $\mathcal{S}$ of 3D candidates. $\mathcal{S}$ only contains models belonging to the same class as O. During pre-processing, the candidate models are corresponded with each other at the part level. The parts are scale-normalized, fitted with a set of primitives, and geometrically and structurally analyzed with the results stored. The model creation process is executed in two major steps (see Figure 4):

1. **Model-driven object analysis:** The goal of this step is to obtain a labeled segmentation of O, where the labels imply part correspondence between O and the candidates to facilitate subsequent processing. The image-space segmentation is driven by a roughly retrieved representative model R based on global shape similarity between silhouette images of O and those of the candidates. This step involves some user interaction to select R and to align it with O so as to bootstrap a multi-kernel graph cut [Delong et al. 2010].

2. **Silhouette-guided structure-preserving deformation:** At the core of our algorithm is a *silhouette-guided deform-to-fit*. Specifically, we deform a 3D candidate model C to fit its silhouette to that of the photographed object O while preserving the structure of C — this step is *automatic*. The 3D candidate model can be automatically retrieved from the set based on the labeled segmentation available for O so that it resembles O. The candidate can also be manually chosen by the user (last three chairs in Figure 1) or randomly chosen from the set. Using a randomly chosen 3D candidate can sometimes bring a "modeling surprise" and lead to an interesting creation.

We adapt the component-wise controller setup developed by Zheng et al. [2011] to compute structure-preserving deformation. However our problem setting and theirs have several critical differences. Their work is developed for *interactive* shape editing where a single user interaction is propagated, one controller at a time, throughout the model. We compute the deformed model *automatically* under *silhouette constraints* which simultaneously influence all the controllers. Moreover, we are faced with a 2D-to-3D analysis problem. Thus the stored high-level structural information in C is not merely preserved but heavily utilized to compensate for the ill-posedness of the deform-to-fit with respect to content in a single image.

Contributions. The main contributions of our work are:

- A new approach to creative modeling of 3D objects that is photo-inspired, model-driven, and structure-preserving: The produced variation inherits the high-level structure of a pre-analyzed candidate so that it is readily usable without post-analysis, in contrast to methods based on part composition.
- An automatic method for silhouette-driven shape modeling from a single image: Most silhouette-based 3D modeling techniques require multi-view silhouettes [Rivers et al. 2010] and are appearance-driven [Kraevoy et al. 2009] rather than structure-driven. Our method preserves the structures of a given model and utilizes such structures to maximally exploit silhouette constraints from a single image.
- An interactive model-driven method for object extraction and segmentation in a single image: High-level structural information in pre-analyzed models is utilized to obtain a labeled segmentation via multi-kernel graph cuts [Delong et al. 2010].

We demonstrate the effectiveness of our modeling technique on many examples of man-made objects. While our deformation is guided by view-dependent cues provided through silhouettes, the manipulation of the 3D candidate model is conducted in a view-independent way. We believe that the automation of silhouette-guided modeling that is structure-preserving will open new opportunities for future work. A unique strength of our approach is that it automatically fits a 3D model to a single-view photo and modifies the model while maintaining its consistency and coherence from multiple views; see Figure 3. This approach stands out with respect to other silhouette-guided methods, e.g., [Nealen et al. 2005; Kraevoy et al. 2009; Tan et al. 2010], which rely on smoothness prior only and are not designed to work on man-made models.

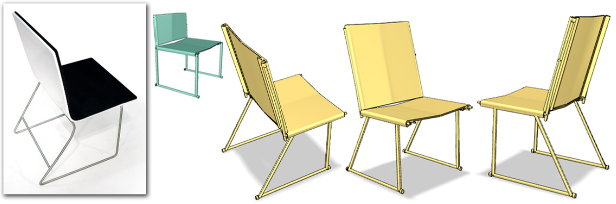

Figure 3: *A photo-inspired model creation (yellow) shown in different views. Although our modeling technique is based on a single view, the result is a full 3D model that is coherent from all views.*

2 Related work

Enabling a novice user to easily create digital 3D models has been a long-standing problem in computer graphics. The literature is rich with modeling systems that assist the user in interactive free-form shape deformation or shape editing that matches a sketch or contents in an image. In the following, we review mainly previous works most closely related to our modeling approach.

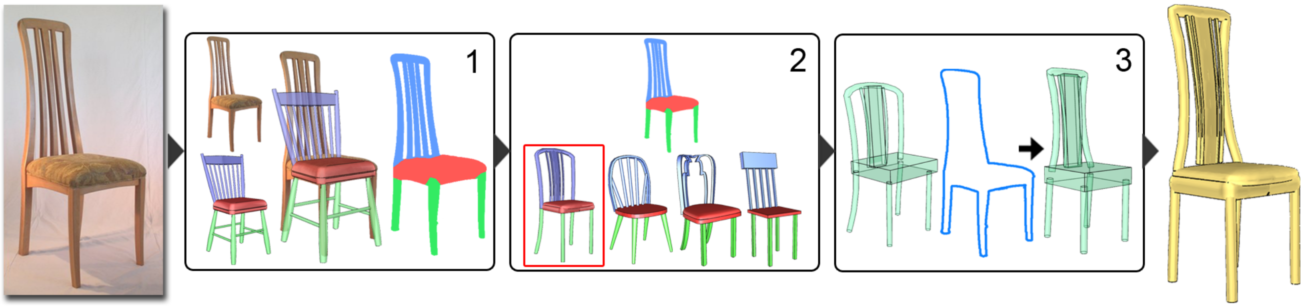

Figure 4: *Overview of photo-inspired 3D modeling. Given a single input photograph, a model-driven approach is applied to obtain a labeled segmentation of the photographed object (step 1). The candidate to be deformed can be randomly chosen or retrieved from the candidate set (step 2). The chosen candidate is deformed to achieve a fit in the silhouettes while preserving its structure (step 3).*

Sketch-based modeling. Freehand drawings provide a rich and natural source of inspiration for 3D creation. In well-known modeling systems such as Teddy [Igarashi et al. 1999] and SmoothSketch [Karpenko and Hughes 2006], the user sketches representative feature curves such as silhouettes [Rivers et al. 2010] or crease lines, in 2D or 3D, and from single view or multiple views. The main challenge lies in how to meaningfully infer the underlying 3D geometry from just a few casual strokes. In our work, we also use silhouette contours in the input photo to guide the structure-preserving deformation. However, by driving the analysis and modeling using a pre-analyzed 3D model, we bypass the difficult task of interpreting a 3D model from only a set of 2D curves.

Modeling from single images. Producing sketches which sufficiently and accurately convey a shape is challenging for most everyday users. Photographs are much easier to acquire using consumer cameras or from online resources. 3D shape reconstruction from 2D images has been an extensively studied problem in computer vision. When the input is a single image, the problem can be severely ill-posed and necessitates additional cues such as symmetry [Shimshoni et al. 2000] or user hints [Lau et al. 2010]. This latter work allows the user to sketch a customized object over a single photo. The photo provides references for the perspective and relative dimensions of the object and also helps the user define various constraints on the sketches. Then the 3D model is recovered via an optimization. Instead of reconstructing the exact shape from a photo, our method uses the photo as a reference to inspire the user without committing to its fine details or appearance.

Data-driven object modeling. Existing 3D shape repositories have provided the base for model creation. The recent work of Xu et al. [2010] performs co-analysis of a set of man-made 3D models at the part level and allows synthesizing new objects with different styles; the creative inspiration in their work comes from a set of 3D models. Perhaps the most classic approach is to compose parts from existing models, tracing back to the work of Funkhouser et al. [2004] on modeling by example. In their work, a new model can be created by replacing some of its parts by other similar parts retrieved from a large database. The Shuffler system of Kraevoy et al. [2007] allows the user to create new models by shuffling interchangeable components from a set of compatible 3D models. Creative inspiration for the parts can come from user sketches [Shin and Igarashi 2007; Lee and Funkhouser 2008] with the modeling process supported by sketch-driven shape retrieval. The exploratory tool of Talton et al. [2009] allows model creation based on parametric design spaces. Chaudhuri and Koltun [2010] introduce an approach for generating data-driven suggestions for creativity support in 3D modeling. All works involving part composition require assembling parts from different sources into a single model. Stitching parts with varying scales and orientations into a coherent model can be a highly non-trivial task, especially for man-made objects. Simple smooth blending often produces unnatural results that break engineering principles. If there are topological variations between the part boundaries, the problem is not even well-defined.

Data-driven methods have also been used in the modeling of human faces [Blanz and Vetter 1999] and full bodies [Allen et al. 2003; Anguelov et al. 2005]; these works are based on morphable models learned from the given databases. The work of [Xi et al. 2007] is driven by single photos. Specifically, they model the variation of body parts by segmenting a collection of human bodies and then building a human body model from a single photo by learning a relationship between the 3D meshes and their 2D silhouettes. These modeling techniques on organic shapes rely on the construction of continuous parametric or statistical models. Our work focuses on man-made shapes where the large variability between corresponding parts makes it difficult to compute correspondence and to build morphable models leading to meaningful shape variations.

Interactive shape editing. Rather than creating a new model, the user can create a variation by interactively editing an existing shape. There have been numerous such methods designed for organic shapes [Sorkine and Botsch 2009], which are carried out by manipulating different editing handles such as anchor vertices, skeletons, cages, and silhouettes. The works of [Kraevoy et al. 2009] and [Tan et al. 2010] make use of user sketches and images, respectively, to manipulate 3D organic shapes, where a template 3D model is aligned and then deformed to match the target. Recently, 1D feature wires [Gal et al. 2009] and 3D bounding primitives called controllers [Zheng et al. 2011] are used as editing handles in an analyze-and-edit framework for man-made models, where the detected structures in the models are preserved during editing. We adopt the controller-based deformation setup from [Zheng et al. 2011], not for interactive editing, but in an automatic setting, to modify a given 3D model under silhouette constraints.

3 Model-driven image-space object analysis

Given an input photograph containing the object O to be modeled, the goal is to obtain a labeled segmentation of O. This is accomplished using a multi-kernel graph cut driven by a representative model R roughly retrieved from $\mathcal{S}$. Figure 5 illustrates the process in some more detail. The labeled segmentation can be used to query $\mathcal{S}$ to retrieve a candidate C that resembles O.

Candidate model set. The candidate set $\mathcal{S}$ is pre-processed to possess part correspondence throughout. Due to possible variations in the part scales of the models, we adopt the recent method of Xu

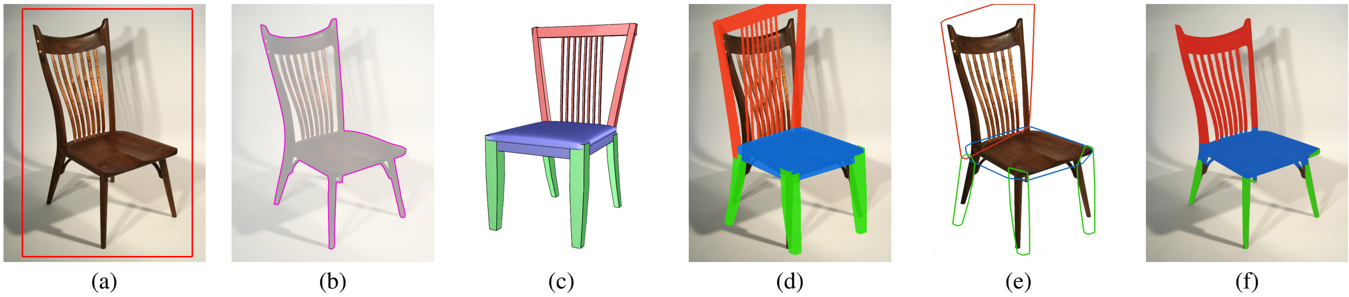

Figure 5: *Steps of model-driven object segmentation. Given an input photograph, the user draws a box (a) to seed a graph cut algorithm and produce a rough object silhouette (b). A 3D model (c) is retrieved based on the silhouette, then overlaid onto the photo (d). Convex hulls of the model parts (e) provide seeds to a multi-kernel graph cut to produce the final segmentation (f); color labels reflect object parts.*

et al. [2010] to perform part correspondence and subsequently normalize the anisotropic part scales of each model. With their method, each part of a candidate model is associated with a tight oriented bounding box (OBB). To enable silhouette-driven model variation (see Section 4 for details), each model is also fitted with a refined set of controller primitives with their geometric interrelations, e.g., symmetry and proximity, detected and stored. The vast majority of the candidate models were those used in the prior work of Xu et al. [2010]. Additional models from http://archive3d.net/ were also added to enrich the candidate sets by number.

Retrieval of representative model. The representative model R is a retrieved candidate that serves to guide the labeled segmentation of O in the photo. As we allow user interaction in aligning R to O before executing the graph cut, R only needs to roughly match O. To this end, we extract a rough silhouette of O using GrabCut [Rother et al. 2004], where the user only needs to place a bounding box around O in the photo to seed the process. The segmented shape of O is then compared to all the 2D projections of each model in $\mathcal{S}$ captured from different views.

Each 3D candidate is projected from 100 viewpoints uniformly distributed over a bounding sphere. A projection contains both silhouette and region information of the shape from a certain view. We adopt the integrated 2D shape descriptor of Zhang et al. [2002] to measure the similarity between O and a projection; the descriptor combines contour-based Fourier descriptors and region-based Zernike moments. We retrieve a small number (10 in our experiments) of candidates that are most similar to O from their best views. The user selects one of them as the representative R. The integrated shape descriptors are fairly compact; they are computed for all the views in preprocessing and stored with the set $\mathcal{S}$.

Model-driven labeled segmentation. To effectively guide the graph-cut segmentation, the representative R should be properly overlaid onto the photo to align with the object O. We rely on user interaction to achieve this, where the user takes R in its retrieved view to overlay it onto O and then adjusts the view and part scales. View adjustment is done by rotating R; since the retrieved view of R already roughly matches that of O in the photo, the rotational adjustment is typically minor. Since R has been pre-segmented with each segment bounded by an OBB, the user is allowed to scale the parts via their OBBs to obtain a better alignment; this is designed to handle discrepancies between part scales in R and O. With R properly aligned with O, we use it to bootstrap a graph cut algorithm to obtain a labeled segmentation of O. Note that the labels are not semantic, they merely serve to establish correspondence between the segments of O with those of the candidate models.

Let $\{R_1, \ldots, R_n\}$ be the set of *connected* parts from R and let H_i be the convex hull of R_i in the projected view. We define the background H_{n+1} as the complement of $H = \bigcup_{i=1}^{L} H_i$ in the image plane. The segmentation of O is expressed as the problem of labeling each pixel with one of the $n+1$ labels corresponding to the n parts and background. We use the multi-kernel graph cut algorithm of Delong et al. [2010], which executes an energy minimization on a weighted sum of three terms: the data term, the smoothness term, and the label term; see the referenced paper for details. We define the data term using $H_i \cap H(O)$ for each label $1 \leq i \leq n+1$, where $H(O)$ denotes the region enclosed by the rough silhouette of O obtained from the previous step. The weights for the three terms above are set at 1, 1, and 20, respectively, throughout. Note that a precise segmentation from this step is not necessary as it only serves to guide the refined, part-level retrieval in the next step.

Candidate retrieval. The 3D candidate C for photo-inspired modeling can be chosen randomly or manually. We can also retrieve C that best resembles O from $\mathcal{S}$. With a labeled segmentation of O available, such a retrieval can be done at the part level. The key advantage of part-level retrieval is that we are able to handle a target object O whose part scales differ significantly from those of the 3D candidates. The retrieval bears some similarity with the rough retrieval of R before, but with two differences. First, we measure similarity between the image of each segment of O and the images of the corresponding model parts in $\mathcal{S}$. The similarity scores are summed up over all the segments in O. Second, instead of enumerating views, we enumerate the scalings in three axis directions of the OBB of each model part. Along each direction, we sample 5 values, resulting in 125 total images per part. The similarity measure between the projected images again uses the Zernike moments. We retrieve 10 models based on the combined similarity score and the user selects among them the best candidate C. Similar to view-based global retrieval, Zernike moments for all the scales can be pre-computed and stored with the candidate set $\mathcal{S}$.

User interaction. In this step of the modeling process, user interaction is involved in several situations. To assist the labeled segmentation, the user draws a bounding box and then performs interactive view alignment and part scaling to seed the graph cut algorithm. View and pose estimation, as well as semantic segmentation, are difficult computer vision problems that are particularly challenging with only a single view given. We have opted to rely on light user interaction rather than resorting to automatic analysis. The user also chooses among a small set of retrieved 3D models the representative R and then the candidate C to deform. Such user involvement is not so critical since even randomly chosen 3D candidates would lead to inspired model creations. However, the obtained results would vary in their degree of conformation to the given photographs, e.g., see Figure 11 for an example.

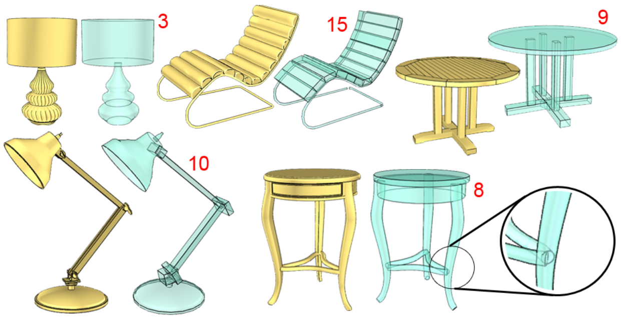

Figure 6: *Fitting controllers to 3D models. Cuboids and generalized cylinders allow close approximations of many man-made shapes with relatively small controller counts (numbers in red).*

4 Model-driven deformation

In this core modeling step, we deform a 3D candidate C automatically to fit the photographed target object O. The deformation is guided by silhouette correspondence between C and O, where the user-specified view for the photograph, obtained in the previous step, is reused. Different from previous silhouette-driven deformation techniques [Nealen et al. 2005; Kraevoy et al. 2009; Tan et al. 2010], we utilize the pre-analyzed structure of C and develop a novel iterative procedure which allows significant geometric variations at the part level of C while preserving its structure. The final model is thus coherent and readily usable.

Component-wise controller setup. Our deformation algorithm adopts the component-wise controller setup of Zheng et al. [2011]. The controllers are simple primitives which enclose the components of a model segmentation and the deformation operates entirely on an organization of the controllers. Our method differs from [Zheng et al. 2011] in several critical aspects. First, their algorithm involves interactive shape manipulation which propagates an edit one controller at a time. Manipulating individual controllers to respect silhouette constraints can be a tedious task. In particular, the user must operate in a single view (that of the photo) with a constrained object pose; this not only introduces occlusions but also obscures operations such as controller rotation. From a computational view, optimizing for one controller at a time can easily lead to suboptimal and instable solutions that are counterintuitive and counterproductive to the user. Our deformation is fully automatic and the silhouette constraints simultaneously affect all controllers in the subsequent optimization. Secondly, our problem is not merely one that involves deformation; we are also faced with an ill-posed 2D-to-3D modeling problem whose solution benefits from a model-driven approach. In particular, we exploit object symmetry in combination with silhouette constraints to first reconstruct the shape and pose of controllers contributing to the model's silhouettes before using them as handles to deform and fit the whole model.

We use two types of primitives as component-wise controllers only: cuboids and generalized cylinders (GCs); their simplicity allows straightforward transforms and in our particular setting, it also facilitates silhouette detection and feature extraction. The controllers are built by recursive primitive fitting [Zheng et al. 2011] on the geometry of C; see Figure 6 for a few results. The silhouette S_C of C's controllers can be traced based on the user-specified view direction from the previous step. The controllers that contribute to S_C are called *external controllers* and the other controllers the *internal controllers*. S_C is segmented into a sequence of curve segments, according to the external controllers. The inter- and intra-controller symmetries along with proximity are automatically detected and stored to constrain the deformation. Controller construction generally takes few minutes (on average 2 minutes) per model. Structural analysis of the controllers takes about 20 to 500 ms per model. Note that these and other preparations for the candidate models in $\mathcal{S}$ are performed offline prior to the modeling process.

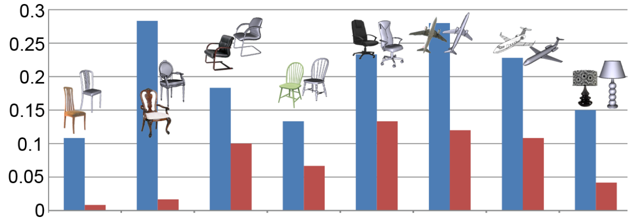

Figure 7: *Improvements by using label constraints (red) vs. without (blue) in silhouette correspondence. Plotted are error rates measured against ground truths produced by manually and consistently labeling over-segmentions of the silhouette contours.*

Overview. Deformation of the controllers is performed in two steps. First, we reconstruct the shape, position, and orientation of the external controllers (Section 4.1) under the guidance of silhouette correspondence between C and O. Then we iteratively optimize the structure of the entire set of controllers guided by the external controllers and other constraints (Section 4.2). Finally, the underlying geometry of C is deformed according to the final configuration of the controllers to obtain the final model.

4.1 Silhouette-guided external controller construction

We wish to reconstruct the external controllers which would reproduce the silhouette S_O of the target object O. First, we establish correspondence between curve segments of S_C, the silhouette of the external controllers, to segments along S_O. To this end, we exploit the part correspondence between O and C implied by the labeled segmentation of O. However, given the segment correspondence, the controller reconstruction problem is ill-posed. We show how object symmetry can be exploited to alleviate the problem; this is particularly useful when modeling man-made objects.

Silhouette correspondence. We first compute a point correspondence between S_C and S_O. The problem is not new and there are many existing solutions. We adopt the recent graph matching method of [Duchenne et al. 2009] for the task as it utilizes constraints on high-order local structures. With part correspondence between C and O, we enforce label constraints in their graph matching formulation. Specifically, we modify the similarity function to assign a large value to two local structures with inconsistent labels, thus penalizing invalid correspondences. Figure 7 demonstrates improvements made by incorporating label constraints.

For each curve segment belonging to an external controller, its corresponding segment on S_O is naturally implied by the point correspondence. Note that segment correspondences are not one-to-one; not all segments on S_O have counterparts in S_C. This raises no issues since all we need is for each external controller to have a target on S_O to guide its reconstruction.

2D-to-3D reconstruction by reflectional symmetry. Based on the segment correspondences, we wish to reconstruct the transformation (rigid or non-rigid) for each external controller to align its silhouette segment to its counterpart on the target object. The core problem is how to recover the 3D position of a curve segment along the 2D silhouette. The 2D-to-3D problem is clearly ill-posed. Additional cues must be exploited to resolve the ambiguities.

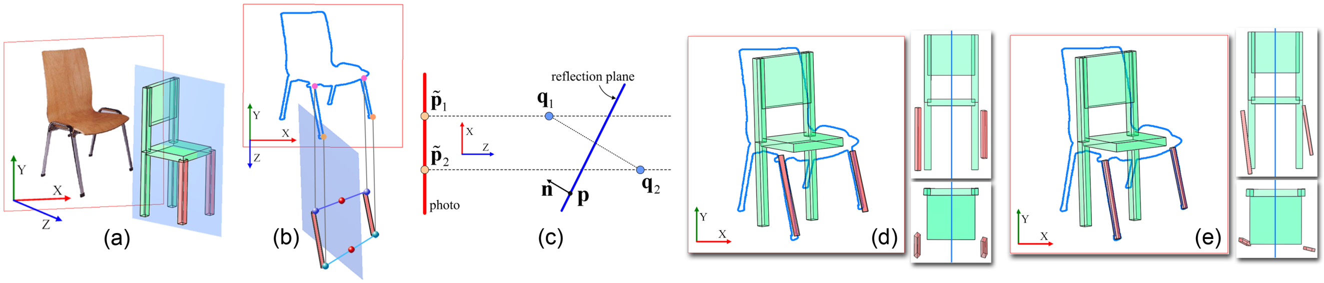

Figure 8: *Illustration of 3D position recovery by reflectional symmetry. To reconstruct the two front legs (a), two pairs of symmetric points along the edges of the two cuboid controllers are used (b). The edges are reconstructed by recovering the 3D positions of the corresponding symmetric point pairs in the photo (c). With the symmetry constraint, the reconstructed controllers approximately preserve the global reflectional symmetry (blue plane) of the candidate model; see top and front view in (d). Simply fitting the cuboids in the XY plane breaks the symmetry (e). Note that the recovered symmetric legs differ in lengths due to (weak) perspective projection in the photo; this is fixed by the structure optimization scheme described in Section 4.2.*

We show how object symmetry can be exploited in our setting. Symmetry is quite common in man-made objects and has been extensively used as an important clue for reconstruction from a single image, e.g., [Shimshoni et al. 2000]. Our setting is a bit different from the general reconstruction problem as we have a 3D silhouette whose correspondence with the image is already established. We develop a scheme to recover the 3D positions of a pair of points on the controller silhouettes that are *reflectionally symmetric* to each other in object space. The scheme operates under the assumption that the input photo was captured under weak perspective, i.e., an approximately orthogonal projection. If the effect of perspective projection is strong, the geometry recovery can be less faithful, e.g., a trapezoidal box, albeit with reflectional symmetry, may result from a perspective photo of a rectangular box.

Suppose that two points $\mathbf{p}_1 = (u_1, v_1, w_1)$ and $\mathbf{p}_2 = (u_2, v_2, w_2)$ on S_C are reflectionally symmetric with respect to the plane $\pi_0 = (\mathbf{p}_0, \mathbf{n}_0)$ defined by point $\mathbf{p}_0$ and normal $\mathbf{n}_0$. Let $\tilde{\mathbf{p}}_1 = (\tilde{x}_1, \tilde{y}_1)$ and $\tilde{\mathbf{p}}_2 = (\tilde{x}_2, \tilde{y}_2)$ be the corresponding points on S_O, respectively; see Figure 8(a). Let $\mathbf{q}_1 = (x_1, y_1, z_1)$ and $\mathbf{q}_2 = (x_2, y_2, z_2)$ be the 3D positions of $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$, respectively, that are to be recovered. Suppose that the view direction is along the z-axis. Then the view transformation $(\mathbf{R}, \mathbf{t})$ can be computed from the user specified view direction. The reflectional plane $\pi = (\mathbf{p}, \mathbf{n})$ between $\mathbf{q}_1$ and $\mathbf{q}_2$ is then calculated as $\mathbf{p} = \mathbf{R}\mathbf{p}_0 + \mathbf{t}, \mathbf{n} = \mathbf{R}\mathbf{n}_0$. The 3D positions of $\mathbf{q}_1$ and $\mathbf{q}_2$ must satisfy the following two constraints:

$$(\mathbf{q}_1 - \mathbf{q}_2)\cdot\mathbf{n} = ||\mathbf{q}_1 - \mathbf{q}_2||,\ (\mathbf{q}_1 - \mathbf{p})\cdot\mathbf{n} = -(\mathbf{q}_2 - \mathbf{p})\cdot\mathbf{n};\quad (1)$$

$$(x_i, y_i) = (\tilde{x}_i, \tilde{y}_i)(i = 1, 2),\ z_1 + z_2 = w_1 + w_2;\qquad (2)$$

where (1) ensures that $\mathbf{q}_1$ and $\mathbf{q}_2$ are symmetric about π, as shown in Figure 8(b), and (2) is used to constrain the projections and the depths of $\mathbf{q}_1$ and $\mathbf{q}_2$. Finally, $\mathbf{q}_1$ and $\mathbf{q}_2$ can be obtained by solving a nonlinear least-squares problem composed by the two constraints above. See Figures 8(c) and (d) for a result and comparison with an alternative without relying on reflectional symmetry.

External controller reconstruction. While reflectional symmetries are ubiquitous in man-made objects, in the absence of such symmetries, other assumptions have to be made. We have opted for a simple remedy. Given two corresponding silhouette segments γ_C and γ_O, we "recover" the 3D position of γ_O by orthogonally (with respect to the image plane) projecting it onto a *supporting plane* for γ_C in 3D space. The supporting plane is determined as the best fitting plane to the segment γ_C and its corresponding segment along the medial axis of the external controller containing γ_C. This scheme is applied to any γ_O whose corresponding γ_C is not part of a reflectional symmetry — most generalize cylinder (GC) controllers fall into this category.

Given the recovered target segment $\hat{\gamma}_C$ for silhouette segment γ_C belonging to an external controller c_i, we now describe how the new external controller $\hat{c}_i$ is constructed. If c_i is a cuboid, we perform re-fitting using the mesh-less deformation method of [Müller et al. 2005] based on shape matching, which computes an optimal transformation to match the initial configuration c_i to the target $\hat{c}_i$ under the soft constraints provided by γ_C and $\hat{\gamma}_C$. As for allowed transformations, we consider rigid transformation and non-uniform frustum scaling along its three principal axes.

If c_i is a GC controller, we create a new GC that interpolates $\hat{\gamma}_C$. If only one side of the silhouette of c_i is recovered, we compute the new medial axis of $\hat{c}_i$ by offsetting $\hat{\gamma}_C$ with the original radius distribution along the medial axis segment corresponding to γ_C in c_i. If both sides of the silhouettes are recovered, we compute the new medial axis, as well as the radius, by tracing along the locus of midpoints between the two silhouette segments. The new GC is reconstructed from the new medial axis based on the obtained radius distribution.

4.2 Structure-preserving controller optimization

We have so far only reconstructed the external controllers. In this section, we optimize the structure of all the controllers with the help of structural information pre-analyzed for the candidate model. The optimization problem is difficult to formulate and solve globally since there are multiple, not just one [Zheng et al. 2011], controllers that constrain each other. Therefore, we opt to greedily and iteratively optimize the structure of the controllers. The process is described in Algorithm 1.

Respected constraints. First, the reconstructed external controllers pass on the silhouette constraints, which may be altered during the iteration. Intrinsic to the controllers themselves, we maintain their self-symmetries, inter-symmetries, proximities and other mutual relations (co-planarity, parallelism, and orthogonality). While symmetry constraints are enforced at the controller level, other constraints are modeled by feature curves. The definition and enforcement of these constraints (carried out by `StructOptController` and `StructOptCurve` in Algorithm 1) are as in [Zheng et al. 2011].

Before starting the iteration, the algorithm first computes the reference shape for each controller, which is used in `Refit` after the

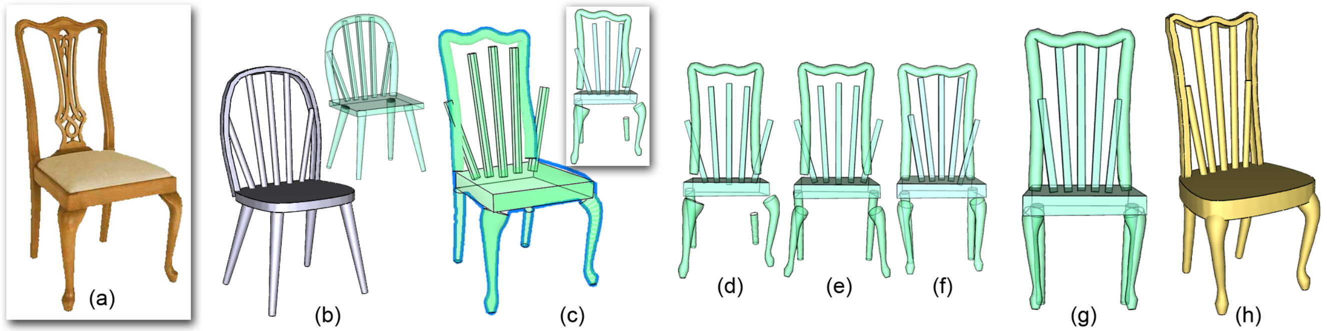

Figure 9: *One iteration step of controller optimization when deforming a 3D candidate (b) to fit a photo (a). The result of reconstructing external controllers (c), though fitting well to the photo silhouette, violates the inherent structure of the candidate, e.g., proximity and symmetry (see insert) between the controllers. We first symmetrize the individual controllers (d) and then optimize the structure using symmetry (e) and proximity constraints (f). The final controllers are well structured (g) and the underlying geometry is deformed accordingly (h).*

Algorithm 1: Controller structure optimization

```
input  : Original controllers $\Omega^O = \{c_i^O\}_{i\in\Omega}$; Reconstructed
         controllers $\Omega^R = \{c_i^R\}_{i\in\Omega}$;
output: Deformed controllers $\Omega^D = \{c_i^D\}_{i\in\Omega}$

foreach controller $c_i^R \in \Omega^R$ do
    if $c_i^R$ is a GC controller then
        $c_i^R \leftarrow$ SymmetrizeGC ($c_i^R, c_i^O$);
    $c_i^D \leftarrow c_i^R$;
while Termination criteria not met do
    $\Omega^D \leftarrow$ StructOptController ($\Omega^D$);
    $\Omega^D \leftarrow$ StructOptCurve ($\Omega^D$);
    foreach controller $c_i^D \in \Omega^D$ do
        $c_i^D \leftarrow$ Refit ($c_i^D, c_i^R$);
```

shapes of the controllers are altered by structure optimization. For a GC controller, we symmetrize its shape (`SymmetrizeGC`) if it is symmetric according to its original configuration. Cuboid controllers already have regular shapes since they have been refitted in the reconstruction stage (Section 4.1).

Iterative optimization. The algorithm iteratively alters all the input controllers, starting from the external controllers, based on symmetry and proximity constraints. Figure 9 shows one iteration of the process of structure optimization. For each group of symmetric controllers, we take the transformation of the controller which is closest to the viewpoint and apply the transformation to all other members in the group (adjusted according to symmetry). In dealing with proximity constraints, the algorithm operates on all feature curves of the controllers, including the occlusion silhouettes of the external controllers whose type is GC. For each controller c, we update its individual feature curves with respect to the feature curves of the other controllers in Ω^D, and then restore the shape of c via refitting. The influence over a curve of c from its proximate curves is found by local frame encoding [Gal et al. 2009], weighted according to the distance between the centers of the two curves.

Since the resulting curves may violate the original shape characteristics of a controller, we perform refitting (`Refit`) to restore the shape characteristics using the method of [Müller et al. 2005] that is based on shape matching, as in Section 4.1. Since the target shapes of the controllers were initialized with the reconstructed configuration, the refitting respects the reconstruction results while ensuring valid shapes and structures of the individual controllers. To respect the silhouette constraint during the optimization, we simply assign higher weights to feature curves at the silhouettes in the setting of a least-squares refitting [Müller et al. 2005].

Termination and geometry update. The iteration terminates when either the maximum movement of all controllers is lower than a threshold (1% of the length of bounding box diagonal of C) or a maximum number of iteration step is reached. Finally, we transfer the deformation applied to the controllers to the underlying geometry. For a cuboid controller, we simply apply the same transformation of the controller onto the geometry it encloses. For a GC controller, we perform skeleton driven deformation using dual quaternion [Kavan et al. 2007].

5 Modeling results

The results presented in this section mainly demonstrate the power and versatility of photo-inspired 3D object modeling. This is highlighted by a small stress test which we call the “Google chair challenge”. Chairs exhibit amazing varieties and are constantly the subject of creative design. We performed a Google image search on “chair” and tested our technique on the returned images one by one, using the 3D chair models created by the prior work of Xu et al. [2010]. Figure 10 shows the models produced based on the first 11 images. We are unable to produce reasonable results on three of the photos due to lack of reasonable candidates from $\mathcal{S}$.

Figure 11 shows a gallery of model creations for different object classes, where one 3D candidate is deformed to create variations based on three photographic inspirations. We emphasize again that although our modeling is based on a single view, the result is a full 3D model that is coherent from all views and structure-preserving with respect to the 3D characteristics of the shape, e.g., symmetry. This is demonstrated in Figure 3.

Structure preservation and conformation to silhouette constraints are clearly evident. However, these characteristics of our method may lead to certain unnatural results. One may observe that the model created out of the third photo in the Google chair challenge has an odd deformation; this is caused by the extra piece under the seat which the graph cut was unable to segment out. For the chair created from the marked photo (red box) in Figure 11, one may expect the back to be deformed to better fit the image silhouette. However, the deformation allowed is dictated by the controllers and

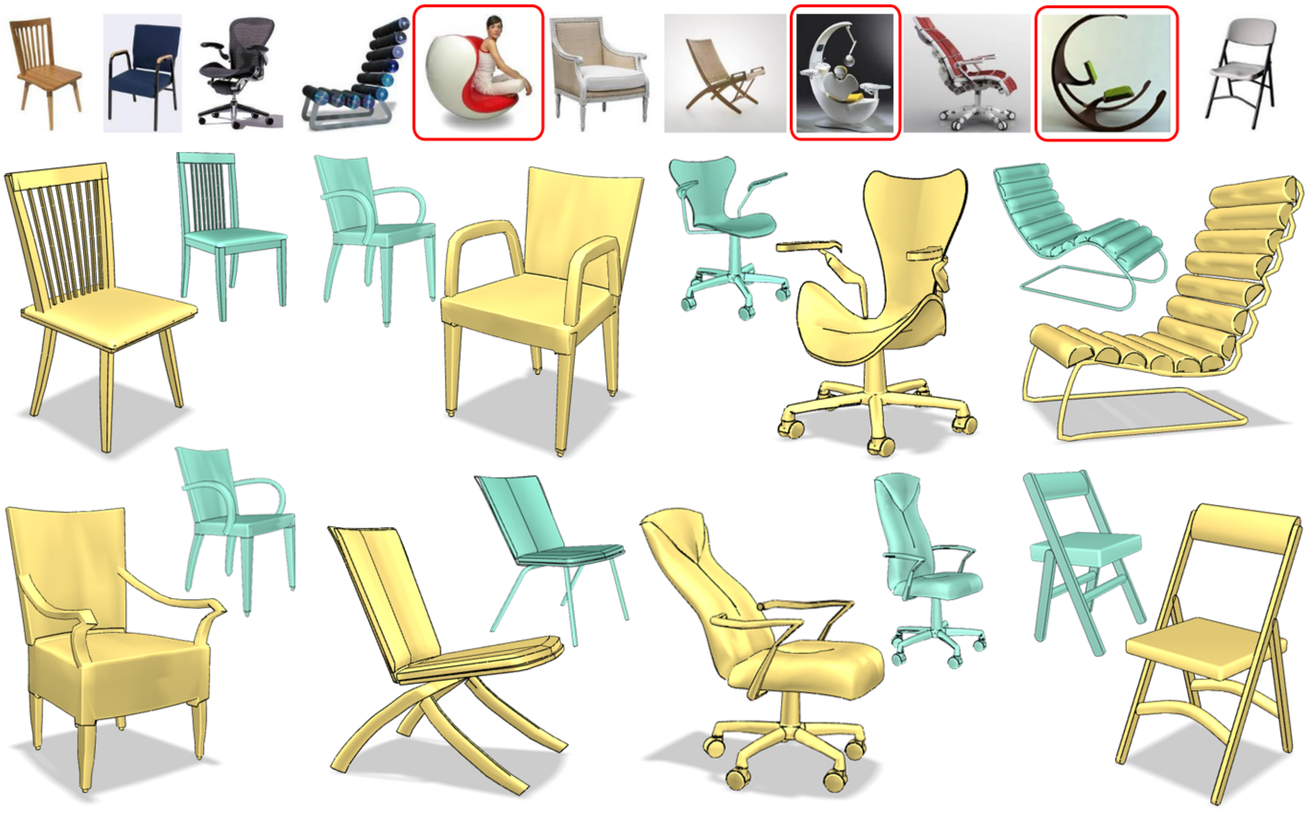

Figure 10: *The "Google chair challenge": modeling chairs after Google image search returns on "chair". Top row lists the top 11 returned images from the search. We are unable to model objects contained in the images marked by red boxes. 3D models created from other photos (yellow) are shown below where the candidates were retrieved from the database for best resemblance to the photographed targets.*

in that case, the back is modeled by cuboids which would not allow a free-form bend. We indeed consider preserving the characteristics of the controllers as part of the structure preservation.

Given a pre-analyzed candidate set, the creation of a model out of a given photo is interactive. The user mainly needs to adjust the view and part scales to align a 3D candidate to the target object in the photo. These interactions are fairly quick and the silhouette-driven deformation is performed in real time.

6 Discussion, limitation, and future work

Despite much effort invested in developing a large variety of 3D modeling tools, it still remains a challenge for a novice user to quickly create a realistic and readily usable digital realization of a design concept or physical 3D object. Creating a digital 3D model of any complexity from scratch is difficult, whether the creative inspiration comes from sketches or photographs. It has been a common practice to utilize the growing collection of 3D models so that the created model is a variation or composition from them. In our approach, the variation is a geometric one which allows us to preserve the structural properties of a candidate model.

Creative inspiration. We believe photographs and sketches truly provide the richest source of creative inspiration for 3D object modeling, far beyond what existing 3D shape repositories can offer at this point. One of the main motivations of this work is to effectively leverage that source. However, 3D inference from a single photo or sparse sketches is highly non-trivial; we alleviate that difficulty by a model-driven approach. While our current work is photo-inspired, to adapt it to work with user sketches is not too far-fetched. The user mainly needs to sketch out the silhouette of the target object to guide the candidate retrieval and structure-preserving deformation. A photo and sketch can even be combined allowing the user to sketch over the photo, perhaps to tune the fine-scale appearance of a model already created by our current technique.

Reusability. 3D modeling is a time-consuming and delicate process, and as such any effective modeling system must encourage reusability. Modeling via part composition encourages reusability at the part level while our approach reuses higher-level structures, which we believe to be more expensive to acquire. Unlike image synthesis, where the end goal is typically a satisfactory visual result, object modeling is a continuous process where any created model is meant to be further refined and manipulated. Hence it is highly desirable for the model to be integrated and coherent rather than merely being a disorganized assembly of parts. In this context, the structure, rather than the appearance, of the model plays a more useful role. We envision that as more effort is geared towards the modeling of man-made objects and as the complexity of the objects increases, structure reuse becomes even more crucial. As a result, the library of models which serve to support creative 3D modeling will have to become more structurally organized beyond possessing only their low-level segmentations.

Geometry vs. structure variation. Our core modeling approach is intrinsically limited by the available structures in the candidate set; it creates new geometric variations but not new structures. This choice is intentional as we aim for structure reuse. At the same

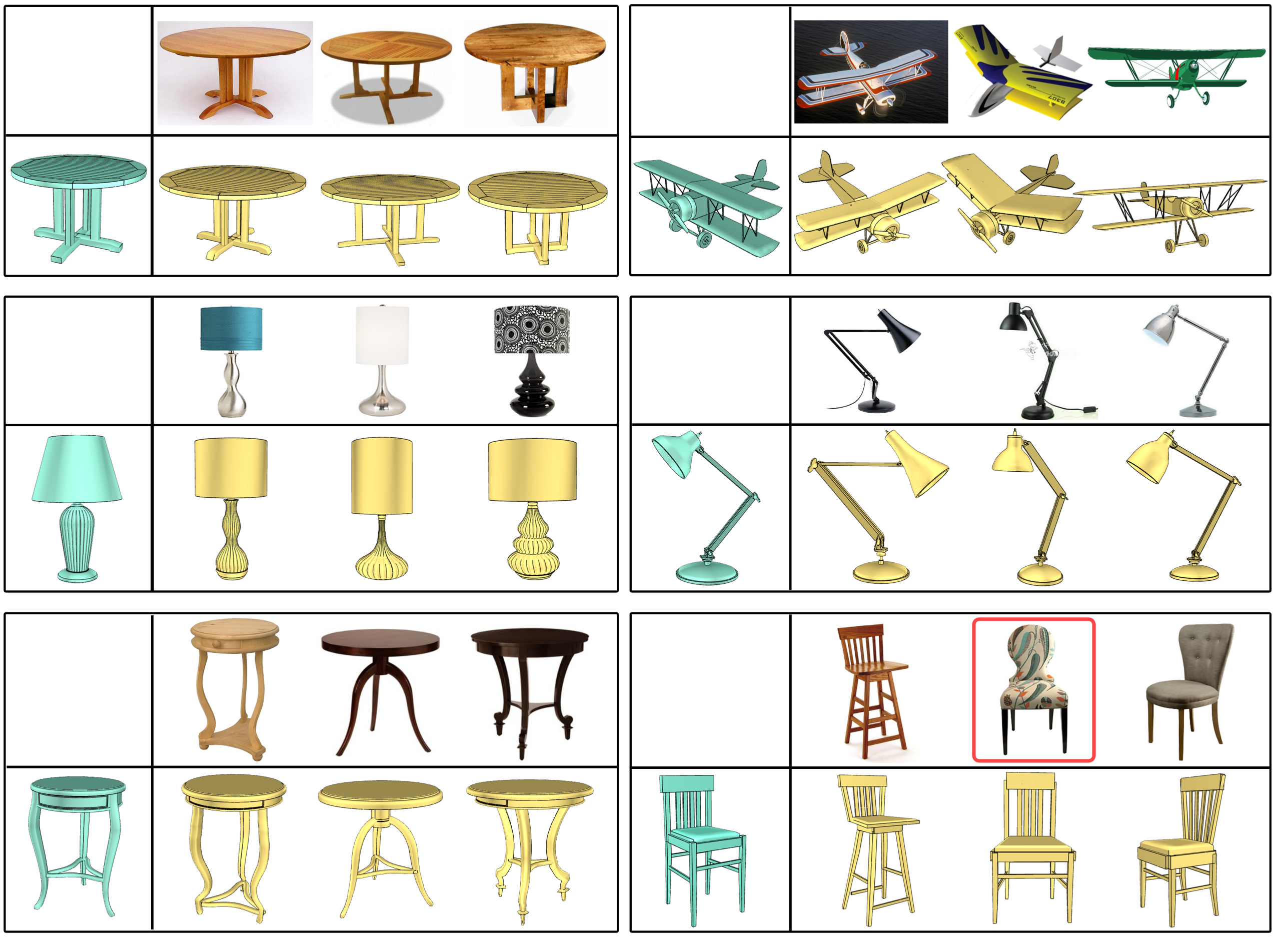

Figure 11: *A gallery of 3D model creations for different object classes and from varying photographic inspirations. In each case, one 3D candidate is deformed to fit three photographed objects. Note that the model created out of the marked chair photo does not match well the image silhouette since its cuboid controllers are not allowed to bend. The 3D candidate was chosen randomly from the chair set.*

time, we argue that the generation of new structures is facilitated by having an existing one and only having to modify it. Building a new structure from scratch or inferring the high-level structure from a low-level representation, such as one obtained via part composition, is more difficult [Xu et al. 2010; Zheng et al. 2011].

Structure vs. appearance. Our modeling approach is structure-oriented and is not meant to reproduce the fine-scale appearance of the photographed object. Man-made objects with highly detailed silhouettes, for example, may not be well handled by our technique; methods aimed at reconstruction are more suitable. With the high-level structures modeled using our method, reconstructing fine-scale details of the shape, e.g., guided by sketches, is possible.

Other limitations. Results from the previous section show that the meaningfulness of the models created using our technique depends on the extraction of meaningful silhouettes and the availability of the appropriate controllers. The major cue employed in our external controller construction step is reflectional symmetry; we do not consider rotational or other forms of symmetries which may strengthen the analysis. The remedy provided to handle cases in the absence of symmetries is quite basic. Figure 12 shows how our method performed on an asymmetric target object. Since the 3D candidate is symmetric, structure preservation leads to a model not in close resemblance to the input. Giving more weights to conformation with the image silhouette leads to an incoherent model. More advanced means to handle asymmetry are needed.

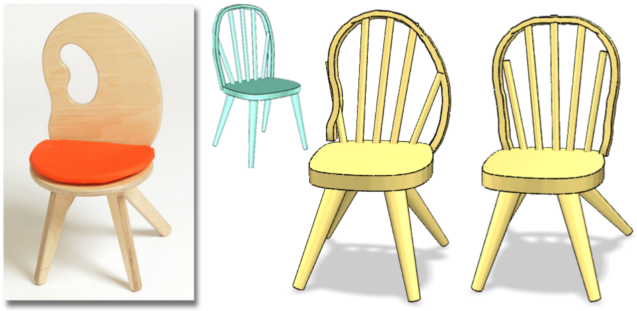

Figure 12: *Modeling results from an asymmetric target object reveals a conflict between silhouette conformation (left result) and structure preservation (right result); neither is really satisfactory.*

Future work. We feel that it is promising to treat the result from our modeling technique as a starting point for further model refinement. We plan to investigate effective means of structure modification and editing of fine-detailed features. The interesting question is how we can develop a model-driven approach to allow their reusability. We would also like to incorporate additional constraints such as crease curves to better steer the structure-preserving deformation. Of general interest is the interesting problem of part stitching, for which we believe a data-driven approach is worth investigating. Finally, it will be our continuing pursuit to find more means to inspire the user in creative 3D modeling.

Acknowledgements. We first thank the anonymous reviewers for their valuable comments and suggestions. We are grateful to the authors of [Zheng et al. 2011] for sharing an early manuscript of their work. Thanks also go to Aiping Wang from NUDT for fruitful discussion. This work is supported in part by grants from NSERC (No. 611370), a research fund for the Doctoral Program of Higher Education of China (No. 20104307110003), the Israeli Ministry of Science, the Israel Science Foundation, the National Natural Science Foundation of China (61070071), and the 973 National Key Basic Research Foundation of China (No. 2009CB320801).

References

Allen, B., Curless, B., and Popović, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. on Graphics (Proc. SIGGRAPH) 22*, 3, 587–594.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. 2005. SCAPE: shape completion and animation of people. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3, 408–416.

Blanz, V., and Vetter, T. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, 187–194.

Chaudhuri, S., and Koltun, V. 2010. Data-driven suggestions for creativity support in 3D modeling. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 29*, 6, 183:1–9.

Delong, A., Osokin, A., Isack, H., and Boykov, Y. 2010. Fast approximate energy minimization with label costs. In *Proc. IEEE Conf. on CVPR*, 2173–2180.

Duchenne, O., Bach, F., Kweon, I., and Ponce, J. 2009. A tensor-based algorithm for high-order graph matching. In *Proc. IEEE Conf. on CVPR*, 1980–1987.

Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., and Dobkin, D. 2004. Modeling by example. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3, 652–663.

Gal, R., Sorkine, O., Mitra, N. J., and Cohen-Or, D. 2009. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3, 33:1–10.

Igarashi, T., Matsuoka, S., and Tanaka, H. 1999. Teddy: A sketching interface for 3D freeform design. In *Proc. SIGGRAPH*, 409–416.

Karpenko, O. A., and Hughes, J. F. 2006. SmoothSketch: 3D free-form shapes from complex sketches. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3, 589–598.

Kavan, L., Collins, S., Žára, J., and O'Sullivan, C. 2007. Skinning with dual quaternions. In *Proc. of Symposium on Interactive 3D Graphics and Games*, 39–46.

Kraevoy, V., Julius, D., and Sheffer, A. 2007. Model composition from interchangeable components. In *Proc. Pacific Graphics*, 129–138.

Kraevoy, V., van de Panne, M., and Sheffer, A. 2009. Modeling from contour drawings. In *Proc. EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling*, 37–44.

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

Lee, J., and Funkhouser, T. 2008. Sketch-based search and composition of 3D models. In *Proc. of EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*.

Müller, M., Heidelberger, B., Teschner, M., and Gross, M. 2005. Meshless deformations based on shape matching. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3, 471–478.

Nealen, A., Sorkine, O., Alexa, M., and Cohen-Or, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3, 1142–1147.

Rivers, A., Durand, F., and Igarashi., T. 2010. 3D modeling with silhouettes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3, 109:1–8.

Rother, C., Kolmogorov, V., and Blake, A. 2004. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3, 309–314.

Shimshoni, I., Moses, Y., and Lindenbaum, M. 2000. Shape reconstruction of 3D bilaterally symmetric surfaces. *Int. J. of Comp. Vis. (IJCV) 39*, 2, 97–110.

Shin, H., and Igarashi, T. 2007. Magic canvas: interactive design of a 3D scene prototype from freehand sketches. In *Proc. of Graphics Interface*, 63–70.

Sorkine, O., and Botsch, M. 2009. Interactive shape modeling and deformation. In *Proc. Eurographics Tutorial*.

Talton, J. O., Gibson, D., Yang, L., Hanrahan, P., and Koltun, V. 2009. Exploratory modeling with collaborative design spaces. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 28*, 5, 167:1–10.

Tan, G., Chen, W., and Liu, L. 2010. Image driven shape deformation using styles. *Journal of Zhejiang University (SCIENCE C) 11*, 1, 27–35.

Xi, P., Lee, W.-S., and Shu, C. 2007. A data-driven approach to human-body cloning using a segmented body database. In *Proc. Pacific Graphics*, 139–147.

Xu, K., Li, H., Zhang, H., Cohen-Or, D., Xiong, Y., and Cheng, Z. 2010. Style-content separation by anisotropic part scales. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 29*, 6, 184:1–9.

Zhang, D., and Lu, G. 2002. An integrated approach to shape based image retrieval. In *Proc. Asian Conference on Computer Vision*, 652–657.

Zheng, Y., Fu, H., Cohen-Or, D., Au, O. K.-C., and Tai, C.-L. 2011. Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum (Proc. EUROGRAPHICS) 30*, 2, 563–572.