Pacific Graphics 2013
B. Levy, X. Tong, and K. Yin
(Guest Editors)

Volume 32 (2013), Number 7

Modeling by Drawing with Shadow Guidance

Lubin Fan[1] Ruimin Wang[2] Linlin Xu[2] Jiansong Deng[2] Ligang Liu[2†]

[1]Department of Mathematics, Zhejiang University, China
[2]School of Mathematical Sciences, University of Science and Technology of China, China

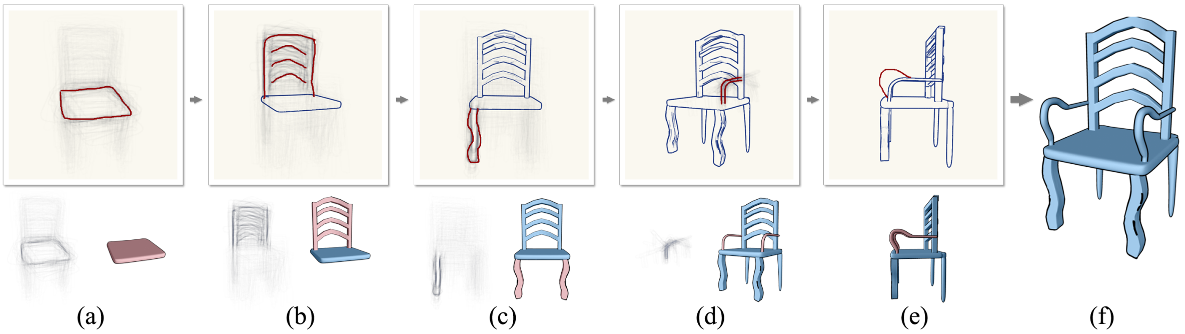


Figure 1: *An example of creating a chair model using our system. The user successively draws freeform strokes (red) to express his design concept on the screen. Our system automatically returns the corresponding 3D parts and dynamically provides* **shadow guidance** *to the user for further drawing. (a)-(e) shows strokes specified by the user, shadow provided by our system (the upper row) and the corresponding 3D models generated in our system (the lower row). The user can also edit the part by drawing near the silhouette of the model (e). (f) shows the final model generated by our system.*

Abstract

Modeling 3D objects is difficult, especially for the user who lacks the knowledge on 3D geometry or even on 2D sketching. In this paper, we present a novel sketch-based modeling system which allows novice users to create 3D custom models by assembling parts based on a database of pre-segmented 3D models. Different from previous systems, our system supports the user with visualized and meaningful **shadow guidance** *under his strokes dynamically to guide the user to convey his design concept easily and quickly. Our system interprets the user's strokes as similarity queries into database to generate the shadow image for guiding the user's further drawing and returns the 3D candidate parts for modeling simultaneously. Moreover, our system preserves the high-level structure in generated models based on prior knowledge pre-analyzed from the database, and allows the user to create custom parts with geometric variations. We demonstrate the applicability and effectiveness of our modeling system with human subjects and present various models designed using our system.*

1. Introduction

The study on creation of 3D content, which is one of the most fundamental tasks in computer graphics, aims to allow a user to turn a design concept in his mind into a 3D model. Users are always aspiring to create 3D models intuitively and creatively. In addition to one's pure imagination, the user is often expecting some inspiration from other resources, e.g., existing models [FKS*04, SI07, LF08] or photos [XZZ*11], during the design process. Moreover, the user also expects further refinement and manipulation of generated model to be readily usable.

[†] Corresponding: lgliu@ustc.edu.cn

Professional modeling packages, such as Maya, 3ds Max, and ZBrush, are difficult to master for everyday users with little knowledge on graphics. A large amount of work on modeling-by-examples [FKS*04, VK07] allows the user to choose existing 3D models or their compositions from a database. However, the user might have a hard time choosing the most suitable model from a large number of candidates. Often an perfectly suitable model does not exist when the user is seeking for a custom one.

Sketch-based user interface provides a more natural and intuitive way for users to express their intention freely. Various sketch-based modeling systems have been proposed during the last two decades [OSSJ09]. But modeling 3D shapes from scratch is still difficult since early concepts are often fuzzy and ambiguous. Moreover, the user may lack drawing skills and the modeling process can be frustrating as the strokes do not convey his intention.

In this paper, we present a novel sketch-based modeling system, which allows the user to create custom models in an intuitive and easy manner (see Figure 1). Inspired by *ShadowDraw* [LZC11], our system supports the user with visualized and meaningful *shadow guidance* under his strokes dynamically to guide the user to convey his concept. During the process, the user can freely navigate the generated 3D model at any views and draw strokes at the view he prefers. We demonstrate the applicability and effectiveness of our modeling system with human subjects and present various models designed using our system.

To the best of our knowledge, our system is the first to adopt the shadow guidance in the 3D modeling system and facilitate drawing strokes and creating 3D models in a natural and intuitive manner. The extension of *ShadowDraw* to sketch-based 3D modeling is non-trivial but our system tackles the challenges and makes the 3D modeling process just like free drawing on 2D canvas. The advantages of our system over previous sketch-based modeling systems are three-fold. First, the user does not need to have much knowledge on geometry but can easily express his design intention with shadow guidance during the modeling process. Unlike some existing systems which require the user to have good drawing skills, our system is more suitable for novices. Second, we present an efficient sketch-based shape retrieval scheme which returns not only the retrieved shape parts but also the correspondence between the query strokes and line drawing images of parts. Third, we preserve the high-level geometry structure in the modeling process such that the generated models are readily usable, and our system can also provide geometric variations to bring about more creative models.

2. Related Work

3D Modeling and sketch-based user interface are popular research topics with a lot of effort in recent years. In this section, we mainly review previous papers closely related to our work.

Sketch-based shape retrieval. The pioneering Princeton Search Engine by Funkhouser et al. [FMK*03] proposes a shape retrieval approach, which compares the descriptors between the user's drawing and the projections of the models. After that, a lot of content based 3D shape retrieval methods have been developed (see [TV08, SXY*11] and references therein). Recently, Eitzs et al. [ERB*12] propose a bag-of-words based method to encode view-dependent line drawings of 3D shapes. Similarly, our system also turns the shape parts searching problem into the comparison between a query image and line drawing images of parts. For our system which allows user to sketch at arbitrary views and provides shadow guidance in interactive rate, we adapt the sketch-based image retrieval method in [LZC11] to obtain candidate parts and compose shadow image simultaneously.

Sketch-based modeling. Various sketch-based modeling systems have been proposed during the last two decades (see [OSSJ09] for a survey). Many systems interpret the input 2D strokes as shape hints of 3D models. The systems [IMT99,KH06,NISA07] interpret user's strokes as feature curves to infer the shape of a closed 3D surface. [ZHH96, YSvdP05, BBS08] interpret strokes as parameters for construction of geometric primitives. Others use strokes to infer the shape's silhouette from different views [RDI10] or to edit the mesh [NSACO05, ZNA07]. Unlike the system with a suggestive interface for 3D drawing in [IH01], we generate the model by assembling the existing parts from database with guidance.

As the number of model collections grows rapidly, it is popular to take the strokes as queries and placement guidelines for assembly-based modeling in recent years. Magic Canvas [SI07] queries models by using the strokes as contours and composes a 3D scene. Lee and Funkhouser [LF08] extend this idea to create novel models from parts, and the user's strokes indicate both shape and placement of them. Recently, Xu et al. [XCF*13] propose the *Sketch2Scene* system which performs co-retrieval and co-placement of 3D relevant models to compose 3D scene. Our work is closely related to *Sketch-to-Design* system [XXM*12] which uses pre-analyzed context-information to retrieve parts and assembles them into a new model. In our system, we also analyze the structure information of the models in preprocessing stage, and use them as the cues in online modeling process. Moreover, we improve user experience by providing the shadow image to guide user's further drawing, which is very suitable for novices.

Drawing with guidance. Researches on drawing with guidance have attracted attention recently. The iCanDraw system [DPH10] provides an interface which guides the user through step-by-step instructions to draw a human face exactly same as the reference image. *ShadowDraw* [LZC11] helps the user in freeform 2D drawing with visual reference, *shadows*, displayed under user's strokes. Our system gets inspiration from this work. However, it is a non-trivial task

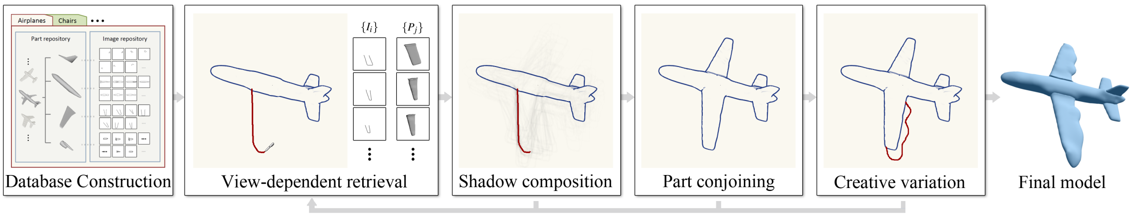


Figure 2: *System pipeline. The user can construct a 3D model in a part-by-part fashion. As the user draws on the canvas, our system interprets user's strokes as similarity queries into database to compose the shadow image and return the corresponding candidate 3D parts. Then the system assembles the selected part by taking user's strokes and prior knowledge as hints. The user can draw strokes near the added parts to make variations.*

to extend it into 3D modeling. More than the single view shadow, we investigate the view-dependent shadow such that the shadow guidance is able to guide user's drawing at any views.

3. Overview

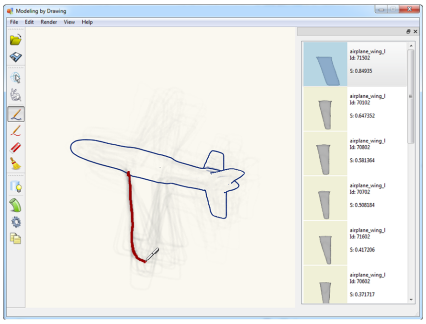
Our system consists of an offline preprocessing to construct a database from a 3D shape collection (Section 4) and an online shadow-guided interactive modeling system (Section 5).

Database construction. We build the database from a collection of pre-segmented 3D shapes which are grouped into classes. In this paper, we adopt five classes of models, i.e., airplanes, chairs, tables, vases, and bicycles, to illustrate our system. The database consists of two repositories. First, the part repository is comprised of parts consistently segmented from shapes in each class. All parts are grouped by their semantic labels, and the structure information is learnt from the original shape. Second, the image repository consists of the line drawing images of each part at 114 sampled views. They are encoded by a binary descriptor. The search engine based on the descriptor and the min-hash algorithm [Chu08] enables efficient retrieval in the modeling process.

User interface. Our system provides a virtual 2D interface to make the user feel like drawing on a canvas with the resolution of 600 × 600 pixel (see Figure 3). While the user drawing, an evolving shadow image will be presented under user's strokes to provide guidance for further drawing. In addition, there is a suggestion panel displaying a list of candidate parts retrieved from the database according to user's drawing. The generated model is displayed in NPR (line drawing) fashion on the canvas. Note that, in the whole modeling session, the user can freely draw strokes and navigate the model at any views.

Modeling by drawing. The user can construct a 3D model in a part-by-part fashion (see the accompanying video). The user starts drawing on a blank canvas at the default view. Then the shadow guidance evolves as the user draws. Modeling proceeds in the following steps (see Figure 2):

1. *Shadow-guided drawing.* The user draws further strokes on the canvas guided or inspired by the shadow image. The strokes provide not only geometric hints for the parts but also the size, position, etc.
2. *View-dependent retrieval.* The system returns a sorted list of line drawing images and the corresponding candidate parts both in descending order of matching scores (Section 5.1).
3. *Shadow image composition.* A shadow image is composed of the retrieved images with the weights obtained by matching and spatially varying information, and it will be aligned to user's sketch and presented under it (Section 5.2).
4. *Part conjoining.* The part suggested by the system or selected by the user will be fitted and placed into the current model by taking the strokes and pre-analyzed structure information as hints, and the whole model will be enforced (Section 5.3).
5. *Creative variation.* To create a custom model, the user can draw strokes near the added parts to make variations by sketch-based deformation and editing (Section 5.4).



airplane_wing_l
Id: 71502
S: 0.84935

airplane_wing_l
Id: 70102
S: 0.647352

airplane_wing_l
Id: 70802
S: 0.581364

airplane_wing_l
Id: 70702
S: 0.508184

airplane_wing_l
Id: 71602
S: 0.417206

airplane_wing_l
Id: 70602
S: 0.371717

Figure 3: *A screen shot of our system. The canvas for drawing is in the middle, and an evolving shadow image is presented under user's strokes. The suggestion panel displaying the sorted list of candidate parts is shown on the right. Common tools are displayed on the left.*

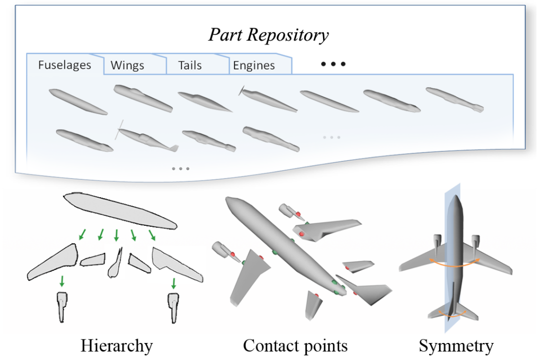


Figure 4: *Part repository in our database. The collection of models are pre-segmented to make up our part repository (top). After consistent segmentation of models, we analyze the hierarchy, contact points, and symmetry of each model. The contact points are highlighted, "mortises" are shown in green and "tenons" are in red.*

4. Database Construction

Our preprocessing consists of three steps. In the first step, the part repository is built from the collection of 3D shapes. Second, we extract the line drawing images of each part at sampled views to construct image repository. At last, we encode the patches of sampled line drawing images.

4.1. Part Repository

To build the part repository, all the 3D shapes in the same class are normalized and aligned with common orientation first. Then, we perform consistent segmentation within each class to decompose the shapes into semantic parts [WAvK*12]. For example, an airplane is decomposed into four parts: fuselage, wings, tails, and engines. To support the part conjoining in the modeling session, we pre-analyze each shape and its parts to learn the structure information, i.e., hierarchy, contact points, and symmetry (see Figure 4).

Hierarchy. We generate a hierarchy for each model. We set the root node as the part, nearest to the center of the model's oriented bounding box (OBB). Then we analyze the connection between segmented parts and recursively set child nodes as the parts connected to the nodes in upper level. It should be noted that this hierarchy obeys the coarse-to-fine principle in sketching.

Contact points. For any pair of parts that are connected in the original model, we record the contact points of them, which are used for composing different parts into a new model. We define the contact points of each part as virtual points at the center of each cutting edge loops generated by segmentation . To simplify the part conjoining problem in the modeling process, we classify the contact points into two types: *mortise* and *tenon*. For a pair of contact points belongs to two connected parts respectively, we define the point on the part that lies in the coarse level of the hierarchy to be *mortise* and the other one to be *tenon*. In modeling process, only a pair of *mortise* and *tenon* contact points can be connected.

Symmetry. We also apply symmetry analysis [MGP06] to detect the global reflection symmetry as well as inter-part symmetry within each model. We record the axes of reflection and align them in modeling process to preserve the global symmetry of the model.

4.2. Image Repository

In order to support sketch-based part retrieval and shadow guidance, we build the image repository by extracting the line drawing images of each part from sampled views.

View sampling. We sample 114 view positions on the view sphere including 100 uniformly sampled views, 6 canonical side views and 8 corner views. We also assign the camera up-direction for each view position. We set the default viewpoint $v_0 = (0.0, 0.0, 1.0)$ and the corresponding up-direction $u_0 = (0.0, 1.0, 0.0)$. We define the *consistent* camera up-direction u^* of any point v on the view sphere as: u^* is rotated from up-direction u_0 of the initial viewpoint v_0 by the 3D transformation matrix $T(v_0, v)$, which is computed based on the shortest arc quaternion rotating v_0 to v, as shown in Figure 5 (bottom left). We set the camera up-direction u_i at each sampled view position v_i equal to the consistent camera up-direction of v_i. With this method, we obtain the sampled views $\{(v_i, u_i)\}_{i=0}^{113}$.

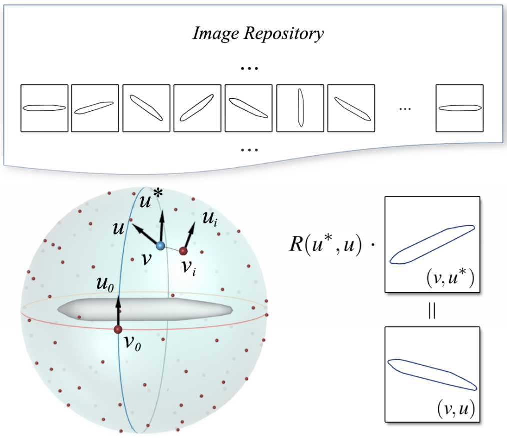


Figure 5: *Image repository in our database. The projection images of all 3D parts from the sampled views on the unit sphere (bottom left) make up our image repository (top). For any viewpoint v (blue), u is current up-direction and u^* is the consistent up-direction. The line drawing image of view (v, u) can be obtained by applying the rotation matrix R to the line drawing image of view (v, u^*) (bottom right).*

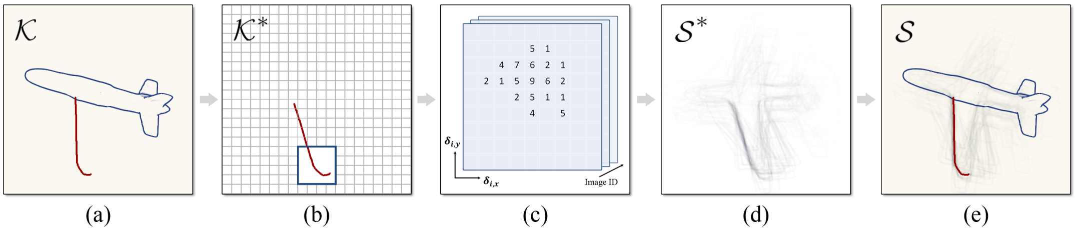


Figure 6: *View-dependent retrieval and shadow composition. First, we align user's drawing image $\mathcal{K}$ (a) to the consistent up-direction u^* at user's view position v to obtain corrected drawing image $\mathcal{K}^*$ (b); then we retrieve candidate line drawing images using 3D voting histogram (c) and compose the shadow image $\mathcal{S}^*$ at u^* of v (d); lastly, we obtain shadow image at user's up-direction $\mathcal{S}$ and display it under user's strokes (e).*

Line drawing images. For each part we extract the *suggestive contour* [DFRS03] as an image with the resolution of 300×300 from each view of $\{(v_i, u_i)\}_{i=0}^{113}$. The image repository consists of the line drawing images of all the parts in part repository.

In this way, for any view position v on the view sphere and its up-direction u on the tangent plane, the line drawing image of (v, u) can be approximated by rotating the line image at the nearest sampled view v_i by 2D rotation $R(u_i, u)$ which rotates the consistent up-direction u_i at v_i to u. Thanks to the sampled view positions and the corresponding camera up-directions, we reduce the storage as well as the search space, and provide the efficient view-dependent retrieval.

4.3. Search Engine

Our search engine is based on encoding the patches of all the images in the repository.

Descriptor. For each image, we set the grid of $9 \times 9 = 81$ patches with 50% overlap between neighboring patches. The resolution of each patch is 60×60 pixel. Then we compute the low-dimensional binary descriptor BiCE [Zit10], which encodes the histogram of edge positions and orientations, on each patch. The descriptor has 432 binary bits by using a three dimensional histogram with 18 positions perpendicular to the edge, 6 positions tangent to edge, and 4 discrete edge orientations.

Min-hash. We employ the min-hash algorithm [Chu08] with inverted file structure as our search engine. To increase both precision and recall, we compute the min-hash signature with 64 *phrases* (for disambiguating we use *phrases* here instead of *sketches* in [Chu08]) for descriptors of each patch of each image. For each new *phrase*, we store it as a new entry in the inverted list. We also build indices for the sampled 114 view positions. Consequently, the final search space for the query is formed by the intersection of the *phrases* and view positions.

5. Modeling by Drawing

The modeling process of a new shape is performed online at an interactive rate based on the established database and search engine.

5.1. View-dependent Retrieval

Given a user's drawing image $\mathcal{K}$, its corresponding view position v and up-direction u are obtained from the system, and our system dynamically returns top N matching images $\{I_i\}_{i=1}^{N}$ and the corresponding sorted candidate parts $\{P_j\}$.

Up-direction alignment. Since the line drawing images of each part are extracted at sampled view positions with consistent camera up-directions. First we align user's drawing image $\mathcal{K}$ to the consistent up-direction u^* at view position v as described in Section 4.2. We obtain the corrected user's drawing image $\mathcal{K}^* = R(u, u^*)\mathcal{K}$ (see Figure 6 (b)).

Candidate images retrieval. Then we query the corrected drawing image $\mathcal{K}^*$ with its view position v in image repository to retrieve a list of candidate images $\{I_i\}$ sorted in descending order based on the similarity to $\mathcal{K}^*$. Since we have to compose and display the shadow guidance image $\mathcal{S}$ based on $\{I_i\}$ later, we adopt the following method to get a list of tuples $\{(I_i, s_i, \delta_i)\}_{i=1}^{N}$, where I_i is candidate image, s_i is the corresponding matching score, and δ_i is the offset between $\mathcal{K}^*$ and I_i.

Given $\mathcal{K}^*$, we compute the BiCE descriptors of its patches first. To increase translation invariance in the images, we use a higher resolution grid of 17×17 with 75% overlap between neighboring patches. So we compute the descriptors and *phrases* on the patches (60×60 pixel) as described in Section 4.3 on reduced $\mathcal{K}^*$ with the resolution of 300×300 pixel. Given the user view position v, we select the K nearest sampled view positions $\{v_j\}_{j=1}^{j=K} (\subset \{v_i\}_{i=1}^{i=N})$ in spherical distance as query view positions.

Using the inverse lookup table, we match each *phrase* of $\mathcal{K}^*$ and $\{v_j\}_{j=1}^{j=K}$ in image repository. We apply a 3D vote histogram to aggregate the matches and approximate the offset between $\mathcal{K}^*$ and each I_i simultaneously (see Figure 6 (c)).

The size of the histogram is $m \times n \times n$, where m denotes the number of images at $\{v_j\}$ in the whole image repository, n means the system allows $\mathcal{K}^*$ shifting less than $\lfloor (n-1)/2 \rfloor$ grids from the database images in both directions of $x-$ and $y-$axis. We compute the matching score s_i of image I_i as the maximum value of histogram of I_i. We also get δ_i from the offset between the bin with maximum value and the center bin. With this method, we obtain $\{I_i\}$ efficiently. Since the user's stroke changes gradually, at each time step, the system only updates the votes resulting from *phrases* derived from patches that have changed. In practice, we return top $N = 100$ candidate images from $K = 6$ nearest sampled views, and set $n = 9$ in 3D histogram.

Candidate parts sorting. We obtain the candidate parts and the corresponding ranks $\{(P_j, r_j)\}$ based on $\{(I_i, h_i, \delta_i)\}$. We collect the corresponding parts of $\{I_i\}$ as $\{P_j\}$. Since several candidate images may correspond to one part, the number of candidate parts is smaller than N. And we define the rank of each candidate part P_j as

$$r_j = \frac{\sum_i \kappa(I_i, P_j) h_i w_i}{\sum_i h_i w_i} \qquad (1)$$

where $\kappa(I_i, P_j)$ denotes the relationship between I_i and P_j. If P_j is the corresponding part of I_i, $\kappa(I_i, P_j) = 1$, otherwise 0. We define w_i as the average value of non-zero pixels of weight image W_i which is used to compose the shadow image in Section 5.2. W_i measures the global matching between the candidate images I_i and $\mathcal{K}^*$. The candidate parts $\{P_j\}$ are displayed in suggestion panel in descending order of rank.

5.2. Shadow Guidance

We compose the shadow guidance image $\mathcal{S}$ based on image retrieval results $\{(I_i, s_i, \delta_i)\}_{i=1}^{N}$ (see Figure 6 (d-e)).

Shadow composition. First, we refine the alignments between the candidate images $\{I_i\}$ and the corrected user's drawing image $\mathcal{K}^*$ based on the approximate offsets $\{\delta_i\}$. Similar to [LZC11], we refine the $\{\delta_i\}$ using Generated Hough transform for 2D translation and get a new set of aligned images $\{\tilde{I}_i\}$ associated with refined offsets $\{\tilde{\delta}_i\}$.

To display the shadow image more clearly where $\tilde{I}_i$ matched to $\mathcal{K}^*$ better, we define the normalized weight image W_i for each $\tilde{I}_i$ as

$$W_i = \frac{s_i V_i}{\sum_j s_j V_j} \qquad (2)$$

where s_i is the candidate image matching score of I_i, and V_i stands for the spatially varying match term in [LZC11] (Equation 8).

Then the shadow image $\mathcal{S}^*$ in consistent up-direction is composed as the blending of $\{\tilde{I}_i\}$ with corresponding $\{W_i\}$:

$$\mathcal{S}^* = \alpha \sum_{i=1}^{N} W_i \tilde{I}_i \qquad (3)$$

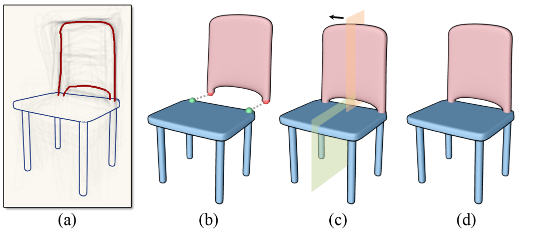


Figure 7: *Part conjoining. (a) User's drawing image; (b) the connection is built between the new part (red) and current model (blue) by matching the contact points and transform the new part to get the initial placement; (c) the placement is refined by aligning the symmetrical planes of the new part and current model; and (d) the final model.*

where α is used to adjust the visibility of shadow image. Since $\mathcal{S}^*$ is at the consistent up-direction u^* of view position v, we compute the final shadow image $\mathcal{S} = R(u^*, u)\mathcal{S}^*$, which rotates $\mathcal{S}^*$ from the consistent up-direction u^* to user's up-direction u.

Hierarchy-based guidance. To augment the shadow suggestion of the part at level less than three in hierarchy, we predict the possible appearance of the context in the original shape collection. We take advantage of pre-analyzed symmetry and hierarchy of each shape to generate the shadows of relevant parts, which are the symmetric counterpart and the adjacent parts in both finer and coarser level (see Figure 1(a-c)). We modify Equation 3 to add the shadows of them,

$$\mathcal{S}^* = \alpha \sum_{i=1}^{N} W_i \tilde{I}_i + \mathcal{S}^*_{sym} + \mathcal{S}^*_{adj} \qquad (4)$$

where $\mathcal{S}^*_{sym} = \beta \sum_j \tilde{I}^j_{sym}$ and $\mathcal{S}^*_{adj} = \gamma \sum_k \tilde{I}^k_{adj}$ denote the shadows composed from aligned images of the symmetric counterpart and the adjacent parts respectively. Here $\{\tilde{I}^j_{sym}\}$ are the list of line drawing images of the symmetric part at the same views of $\{I_i\}$ and are aligned by the same refined offsets $\{\tilde{\delta}_i\}$ of $\{\tilde{I}_i\}$, and so do $\{\tilde{I}^k_{adj}\}$. In practice, we set $\beta = \frac{\alpha}{2}$, $\gamma = \frac{\alpha}{4}$, and $\alpha = 1.0$.

5.3. Part Conjoining

Once the user wants to add the retrieved part P_{new} (the top one of candidate parts or selected by the user in suggestion panel) into the current model $\{P_s\}$, the system automatically fits P_{new} to user's drawing $\mathcal{K}$ by fitting the size of 2D bounding box of P_{new} projected from the current view to that of $\mathcal{K}$'s. Then we consider three problems in placement: obtaining the parts $\{Q_t\}(\subset \{P_s\})$ which are connected to P_{new}, connecting them appropriately, and preserving the symmetry based on prior knowledge (see Figure 7).

First, we obtain the $\{Q_t\}$ by checking whether the parts

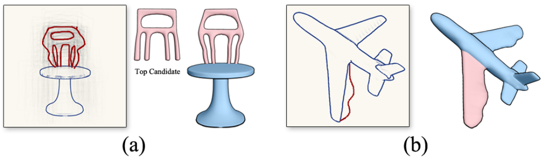


Figure 8: *Creative variations generated using our system. (a) The user draws the chair back and the silhouette of the top candidate part is fitted to user's drawing. (b) The left wing of the airplane is deformed according to user's stroke.*

in $\{P_s\}$ are covered by $\mathcal{K}$. In addition, we obtain the *mortise* contact points $\{m_\alpha\}$ on $\{Q_t\}$ nearby $\mathcal{K}$. Then we build the connection between P_{new} and $\{Q_t\}$ by associating the *tenon* contact points $\{t_\beta\}$ on P_{new} with the mortises $\{m_\alpha\}$. We match them using a greedy strategy. We pick up the closest contact point pair (m_α, t_β) sequentially until there exists no pairs. We compute the transformation of each pair and blend them to a final transformation. After that, we employ mass-spring system used in [JTRS12], which enforces the contact constraints between contact points. To preserve the symmetry of the new model, we consider both aligning symmetrical planes and preserving inter-part symmetry. Some adjacent parts (e.g., the back and seat of chair) are self-symmetric and they share the symmetrical planes. We place P_{new} by aligning their symmetrical planes. For P_{new} with inter-symmetric counterpart (e.g., the wings of airplane), once P_{new} is placed, we obtain the symmetric counterpart from part repository and place it according to the symmetry.

5.4. Creative Variations

The user can make creative variations to generate parts dissimilar to those in database using our system.

Global fitting. When no satisfying part is obtained from user's drawing, our system fits the silhouette of the selected part to user's drawing $\mathcal{K}$ by deformation using mean value coordinates [JSW05] (see Figure 8 (a)). First, the user selects one of the most similar parts to $\mathcal{K}$ from suggestion panel and adds it into canvas. Then our system embeds it into a cage constructed by its oriented bounding box. We build the correspondence between cage vertices and stroke points automatically by uniformly sampling points on edges of the cage and finding the nearest points on the strokes.

Local editing. The system also provides a sketch-based local editing tool for the user to modify the detail of the part by drawing another stroke near it (see Figure 8 (b)). We extract the line segments used for line drawing rendering in real time and amend them to be a continuous curve using shortest path. Then we clip the curve by the ends of stroke, and parameterize both the stroke and clipped curve to $[0,1]$ based on arc lengths on the screen. This induces a mapping between the user's stroke and mesh vertices. Finally, we adopt a modified Laplacian mesh editing method [NSACO05] to deform the part referring to user's stroke.

6. Results

We have implemented our system on a PC with dual-core Intel i5 CPU and 4GB RAM.

Dataset. To illustrate our system, we create a database consisting of 249 3D models with five categories collected from the Princeton Shape Benchmark (PSB) [SMKF04], COSEG [WAvK*12], and [vKXZ*13]. The five categories are vases and teapots (52 models), airplanes and birds (62 models), tables (40 models), chairs (75 models), and bicycles (20 models). Each category contains semantically similar models. We preprocess these model categories to construct the part repository and image repository automatically. The models from PSB in the same category are decomposed consistently using the method in [WAvK*12], and we reuse the co-segmentation results of the models collected from [WAvK*12, vKXZ*13]. For those segmentation results which are unsatisfactory (less than 8%), we manually correct them. It takes about four hours to generate the part repository and half an hour for the corresponding image repository.

Experimental results. Figure 1 shows an example of modeling session using our system. As the user draws at different views, our system provides dynamic shadow guidance under user's strokes to guide furthering drawing. By considering the user's current stroke as well as the prior knowledge of the model structure (see Figure 1 (a-c)), our system provides hierarchical shadow guidance. For example, when the user draws the chair seat, the shadows of chair back and legs are also displayed. When user draws the detail part, e.g., the armrests, the shadow concentrates in current strokes. It helps the user to realize the global structure of the model at the beginning of modeling, rather than to focus on the details. In addition, the user can draw strokes near the silhouette of the added armrests to make a custom one.

Figure 9 bottom shows bicycles created by users using our system. The modeling session of the red bicycle is shown in

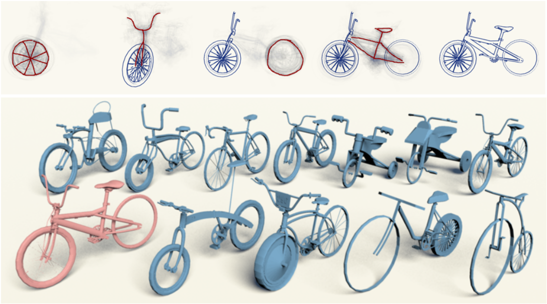


Figure 9: *Bottom: bicycle models created by users using our system. Top: modeling session of the red bicycle.*

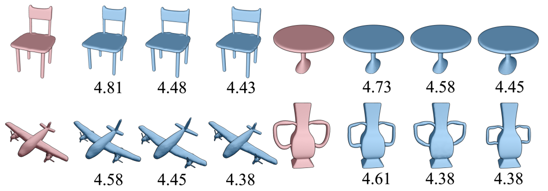


Figure 10: *The top three models of each category in* reference-oriented modeling *task. The reference models are shown in red. The generated models (blue) are sorted by cross-rated average similarity scores (number below each model, 1 as "totally different" and 5 as "exactly the same").*

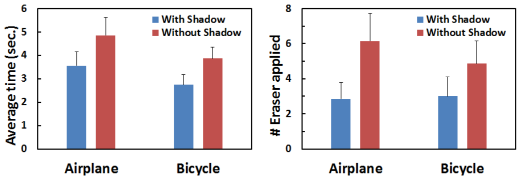


Figure 12: *Statistics of User study III. Left: average time spent on each task. Right: average number of times the eraser operation applied.*

the top. Figure 11 shows more designed models. In each category, the users were asked to freely create 3D models according to their own intentions. It can be seen that the generated models have many variations and can reflect users' design concepts well. See more 3D models created by our system in the accompanying video.

6.1. User Studies

To evaluate the availability and effectiveness of our system, we conducted four user studies: *Reference-oriented modeling*, *free modeling*, *modeling with/without shadow guidance*, and *comparison with other system*. We recruited 19 college students (8 women and 11 men) for the studies. Eight of them have experience on 3D modeling with commercial packages. 11 participants have computer graphics background, and the rest are novices who are interested in 3D modeling. Before the experiments, the participants were encouraged to spend 15 minutes getting familiar with the system under our instructions. At the end of studies, they were asked to fill in a short questionnaire. During the studies, we recorded the sequence of strokes (both drawing and erasing) and time spent on each task. Details of user studies can be found in the supplementary material.

User study I: Reference-oriented modeling. In the first study, we aimed to evaluate the basic availability of our system. The participants were asked to design models as similar as possible to reference models (i.e., vase, airplane, table, and chair), which were created by a professional using our system. After they finished the task, the modeling results were cross-rated among the participants. Figure 10 shows the top-ranked models of each category. It seems that participants can recreate models using our system with minor differences. The differences among *Vase* models are larger than others', since the handles of the reference model were modified, and there is no exactly the same one in the dataset.

User study II: Free modeling. In this study, we aimed to investigate effectiveness of our system for conceptual design. The participants were asked to freely create models given the name of category. The models in different categories generated by the users are shown in Figure 11. We note that the participants created various models by combining parts from different model categories and editing them.

User study III: Modeling with/without shadow guidance. Given two reference models (airplane and bicycle), we invited eight participants to reconstruct the models with and without shadow guidance. To eliminate the learning effect, the participants were asked to finish the four sub-tasks randomly. Figure 12 left shows the average time spent on each task, and the right chart shows average number of times the eraser operations applied. We find that shadow guidance helps the participants to present their design intentions efficiently and accurately. This is also confirmed in questionnaire, participants gave 3.7 on average for the effectiveness of shadow guidance. They said the shadow suggestion guided them to draw the shape in right scale and position which was also demonstrated in *ShadowDraw* system [LZC11].

User study IV: Comparison with *Sketch-to-design*. We compared our system with *Sketch-to-design* [XXM*12] which was a 3D modeling system allowing the user to draw strokes on a reference model and providing suggestion by context-information. We conducted a subjective comparison between the two systems as the dataset of chair models provided by *Sketch-to-design* is relatively small. In this study, we asked each participant to create chair models using both systems separately. Then they were asked three questions:

Q1. Which system is easier to use?
Q2. Which kind of suggestion is more intuitive?
Q3. Which system do you prefer?

Database	#Models	#Parts	T_R	T_S	T_C
Chairs	75	485	110	86	22
Tables	40	185	139	93	33
Airplanes+Birds	62	271	76	105	38
Vases+Teapots	52	170	129	62	15
Bicycles	20	86	83	83	17

Table 1: *Response time (in milliseconds) of each stage of our system on various categories. T_R, T_S, and T_C denote average response time of view-dependent retrieval, shadow composition and part conjoining, respectively.*

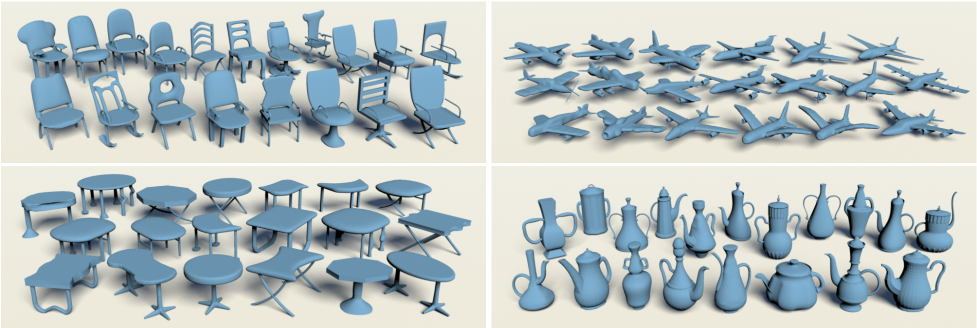


Figure 11: *3D models created by users using our system.*

We find that our system is comparable to *Sketch-to-design* which provides a simple user interface for users to sketch and replace the 3D parts of reference model, but our system provides more intuitive suggestion and allows the user to make variation of the model easily. These are confirmed by the above questions. For Q1, about 42% of participants thought our system was easier to use than *Sketch-to-design*, and 57% of them thought shadow guidance was more intuitive. For Q3, 52% of the participants preferred our system.

Statistics. Table 1 shows the average response time of our system on each category. Since we employ the min-hash technique with inverted index structure as our search engine, the response time of retrieval does not depend on the size of database, but rather on the geometric variations of the 3D models within the database. The fewer variations in the part repository, the more shared *phrases* in the patches of line drawing images, and the search space will be denser. The response time of shadow composition depends on the initial offsets approximated in retrieval stage. Since the positions of wings and tails are various, it takes more time to compose shadow images on *Airplane* category.

On average the participants took 3 minutes to model an object in *User study I* and 4.5 minutes to create a new model in the second study, which demonstrate that our system is easy to use. This is confirmed by the participants in questionnaire, about 89% of them thought our system was easy to use, and 94% of them thought our user interface was intuitive. In general, all participants agreed that the shadow guidance was useful and they did appreciate it. They gave a score of 3.6 on average for the quality of shadow (with 1 as "bad" and 5 as "great"). 89% of the participants said they like our system, the rest thought that the shadow guidance was useful but they preferred to create models from scratch, not by combining the existing parts.

6.2. Limitations

Our system suffers some limitations. First, our current system provides limited variations of the models. For augmenting the variations, other sketch-based editing tools can be adopted. Second, our system focuses on conveying the user's design intention intuitively and quickly. We simplify the part placement problem to match the corresponding contact points, but it reduces the degree of freedom of the placement. Third, our system relies on the database built in the offline stage. Our preprocessing step is designed to be automatic. But for the unsatisfactory results, e.g., inconsistent segmentation, we need to correct them manually. For significantly large model repositories or models with large variations, more manual efforts are needed in preprocessing stage. Thus, more sophisticated automatic shape analysis techniques can be employed, e.g., [vKXZ*13].

7. Conclusion

In this paper, we present a novel sketch-based modeling system which allows novices to easily and quickly create 3D custom models. Our system supports the user with visualized and meaningful shadow suggestions to guide the user's drawing dynamically. The shadow guidance is achieved by the view-dependent shape retrieval and shadow composition techniques which are based on pre-analyzed 3D model parts and the corresponding line drawing images. In addition, our system preserves the high-level geometry structure in generated models, and provides geometric variations to bring about more creative models. We demonstrate the applicability and effectiveness of our system with user studies, which show that our system produces readily usable 3D models for users with some drawing and modeling skills but lack of expertise.

For future work, we are interested in seeking a more efficient way to support more intelligent guidance for modeling. Although the current shadow suggestion dynamically provides guidance for user during modeling process, it relies on the retrieved line drawing images only. An interesting way is to compose the shadow guidance with other information, for example, more prior knowledge of the dataset and functionality of the added parts. Another interesting way is to explore more intuitive and helpful guidance for both novices and experts during the modeling process.

Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments. This work is supported by the National Natural Science Foundation of China (61222206, 61073108, 11031007) and the One Hundred Talent Project of the Chinese Academy of Sciences.

References

[BBS08] Bae S.-H., Balakrishnan R., Singh K.: Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models. In *Proc. User interface software and technology* (2008), pp. 151–160. 2

[Chu08] Chum O.: Near duplcate image detection: min-hash and tf-idf weighting. In *BMVC* (2008). 3, 5

[DFRS03] DeCarlo D., Finkelstein A., Rusinkiewicz S., Santella A.: Suggestive contours for conveying shape. *ACM Trans. on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 848–855. 5

[DPH10] Dixon D., Prasad M., Hammond T.: icandraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems* (2010), pp. 897–906. 2

[ERB*12] Eitz M., Richter R., Boubekeur T., Hildebrand K., Alexa M.: Sketch-based shape retrieval. *ACM Trans. on Graphics (Proc. SIGGRAPH) 31*, 4 (2012), 31:1–10. 2

[FKS*04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by example. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 23:652–663. 1, 2

[FMK*03] Funkhouser T., Min P., Kazhdan M., Chen J., Halderman A., Dobkin D., Jacobs D.: A search engine for 3d models. *ACM Trans. on Graphics 23*, 1 (2003), 83–105. 2

[IH01] Igarashi T., Hughes J. F.: A suggestive interface for 3d drawing. In *Proc. ACM symposium on User interface software and technology* (2001), pp. 173–181. 2

[IMT99] Igarashi T., Matsuoka S., Tanaka H.: Teddy: a sketching interface for 3d freeform design. *ACM Trans. on Graphics (Proc. SIGGRAPH)* (1999), 409–416. 2

[JSW05] Ju T., Schaefer S., Warren J.: Mean value coordinates for closed triangular meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 561–566. 7

[JTRS12] Jain A., Thormhlen T., Ritschel T., Seidel H.-P.: Exploring shape variations by 3d-model decomposition and part-based recombination. *Computer Graphics Forum (Proc. Eurographics) 31*, 2 (2012), 631–640. 7

[KH06] Karpenko O. A., Hughes J. F.: Smoothsketch: 3d free-form shapes from complex sketches. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006), 589–598. 2

[LF08] Lee J., Funkhouser T.: Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling* (2008), pp. 97–104. 1, 2

[LZC11] Lee Y. J., Zitnick L., Cohen M.: Shadowdraw: Real-time user guidance for freehand drawing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 30*, 4 (2011), 27:1–10. 2, 6, 8

[MGP06] Mitra N. J., Guibas L., Pauly M.: Partial and approximate symmetry detection for 3d geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006), 560–568. 4

[NISA07] Nealen A., Igarashi T., Sorkine O., Alexa M.: Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. on Graphics (Proc. SIGGRAPH) 26*, 3 (2007). 2

[NSACO05] Nealen A., Sorkine O., Alexa M., Cohen-Or D.: A sketch-based interface for detail-preserving mesh editing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 1142–1147. 2, 7

[OSSJ09] Olsen L., Samavati F., Sousa M. C., Jorge J.: A taxonomy of modeling techniques using sketch-based interfaces. *Computers and Graphics 33*, 1 (2009), 85–103. 2

[RDI10] Rivers A., Durand F., Igrarshi T.: 3d modeling with silhouettes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 4 (2010). 2

[SI07] Shin H., Igarashi T.: Magic canvas: Interactive design of a 3-d scene prototype from freehand sketches. In *GI '07 Proceedings of Graphics Interface 2007* (2007), pp. 63–70. 1, 2

[SMKF04] Shilane P., Min P., Kazhdan M., Funkhouser T.: The princeton shape benchmark. In *Proc. Shape Modeling International* (2004), pp. 167–178. 7

[SXY*11] Shao T., Xu W., Yin K., Wang J., Zhou K., Guo B.: Discriminative sketch-based 3d model retrieval via robust shape matching. *Computer Graphics Forum (Proc. Pacific Graphics)* (2011), 2011–2020. 2

[TV08] Tangelder J. W., Veltkamp R. C.: A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications 39*, 3 (2008), 441–471. 2

[VK07] Vladislav Kreavoy Dan Julius A. S.: Model composition from interchangeable components. In *Proc. Pacific Conference on Computer Graphics and Applications* (2007), pp. 129–138. 2

[vKXZ*13] van Kaick O., Xu K., Zhang H., Wang Y., Sun S., Shamir A., Cohen-Or D.: Co-hierarchical analysis of shape structures. *ACM Trans. on Graphics (Proc. SIGGRAPH) 32*, 4 (2013). 7, 9

[WAvK*12] Wang Y., Asafi S., van Kaick O., Zhang H., Cohen-Or D., Chen B.: Active co-analysis of a set of shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 31* (2012). 4, 7

[XCF*13] Xu K., Chen K., Fu H., Sun W.-L., Hu S.-M.: Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Trans. on Graphics 32*, 4 (2013). 2

[XXM*12] Xie X., Xu K., Mitra N. J., Cohen-Or D., Chen B.: Sketch-to-design: Context-based part assembly. *ArXiv e-prints* (2012). arXiv:1212.4490. 2, 8

[XZZ*11] Xu K., Zheng H., Zhang H., Cohen-Or D., Liu L., Xiong Y.: Photo-inspired model-driven 3d object modeling. *ACM Trans. on Graphics (Proc. SIGGRAPH) 30*, 4 (2011), 80:1–10. 1

[YSvdP05] Yang C., Sharon D., van de Panne M.: Sketch-based modeling of parameterized objects. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling* (2005). 2

[ZHH96] Zeleznik R. C., Herndon K. P., Hughes J. F.: Sketch: an interface for sketching 3d scenes. *ACM Trans. on Graphics (Proc. SIGGRAPH)* (1996), 163–170. 2

[Zit10] Zitnick C. L.: Binary coherent edge descriptors. In *ECCV* (2010), pp. 170–182. 5

[ZNA07] Zimmermann J., Nealen A., Alexa M.: Silsketch: automated sketch-based editing of surface meshes. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling* (2007), pp. 23–30. 2