

# 针对浏览式草图造型的按需部件建议技术

Xuekun Guo

2016.04.25

## 1 引言

开发易用且功能强大的三维造型技术一直是计算机图形学的重要目标之一。专业工具，如Autodesk Maya<sup>®</sup>，具有强大的三维造型功能，但操作复杂，需要长时间的学习才能掌握。那些易用的工具，如Spore Creature Creator<sup>®</sup>，将造型过程简化为拼接预定义好的部件，然而造型能力有限。弥合三维造型工具简单易用与功能强大之间的鸿沟是图形学界最具挑战性的问题之一。

针对该问题，研究人员提出基于草图的造型技术 [LIAL07]。这种造型方式简单易用，得到很多关注 [TSH99]。但大多数用户的绘画技术不好，完全按草图很难恢复出具有丰富几何细节的三维模型。最近几年，研究人员尝试通过草图从三维模型数据库内搜索模型[ERB<sup>+</sup>12]，或者通过重新组合现有模型的部件产生新模型[LF08]。这些技术使得普通用户可以快速造型高质量的模型。然而，这些技术仅允许组合预定义的部件或对部件做简单变形，造型结果单一、功能简单。

在本文内，我们提出一种基于数据驱动的草图式造型方法。该方法不再局限于使用预定义的部件组合模型，而是实时地从模型数据库内提取出符合造型意图的部件。我们的方法不需要预分割或标记的三维模型数据库。我们的技术使用户通过草图简单方便地创建新模型（如图1）。用户仅仅需要具备使用Teddy系统的草图绘制技术，就可以创建出多样的具有丰富几何细节的模型。

实现这样的系统需要解决如下两个技术难题：1) 部件搜索空间非常大。因为模型数据库未预分割，我们不得不实时地找到并切割与用户意图匹配的部件。为了将搜索空间限制在可解范围内，我们设计了一种平衡策略，使提示的部件在保持可用的前提下尽可能地多样化。2) 我们的方法必须足够快，满足用户交互。我们发现，如果三维模型的二维投影相似，两三维模型也相似。因此，我们将三维模型匹配问题转化为多视角下的二维轮廓匹配问题。本文的贡献有三点：

- 提出一个基于草图的探索式造型方法。使用该方法，用户可以从未预分割的三维模型数据库内，实时提取符合用户意图的个性化部件，创建高质量的模型。
- 提出一个快速的基于草图的三维模型局部匹配算法。该算法将三维模型的局部匹配问题转化为多视角下的二维轮廓局部匹配问题。这些轮廓组织在一个kNN图内。
- 提出一种新的草图驱动的模型分割算法。该算法基于超面片轮廓与候选轮廓间的匹配程度，在模型的超面片图上实时分割出与草图相匹配的部件。此外，还有一种新的部件优化算法，进一步优化候选部件。

## 2 相关工作

**探索式造型技术** 自从研究人员意识到三维造型是一种开放式的创新过程后 [CK10]，探索式造型技术越来越受重视。很重要的一类探索式造型技术专注于向用户建议可用部件。Funkhouser等人提出Modeling by Example [FKS<sup>+</sup>04]系统，允许用户基于三维代理模型搜索数据库，组合部件得到新模型。Chaudhuri等人 [CKGK11]提出一个概率推理模型，基于几何与语义兼容性提示可用部件。

与现有的部件建议技术不同，我们不依赖预分割与标记的模型数据库，而是实时地从数据库内提取出候选部件。因此，我们的方法允许用户高效地访问更广阔的造型空间。

**模型语义分割技术** 模型语义分割技术在探索式造型技术中占有重要地位，因为几乎所有的探索式造型技术都需要预分割或标记的模型数据库。针对模型语义分割，研究人员提出各种各样的解决方案。。。A研究人员提出了各种各样不同的技术，包括自动化的或手动的 [Sha08, CGF09]。最近，研究人员又提出针对一组模型做共分割（co-segmentation）的技术 [KHS10, HFL12, HKG11, SvKK<sup>+</sup>11]。共分割技术将一组模型视为一个整体，分割的过程中，可以利用该组模型的共有结构，因此得到更加一致性的结果。这些工作可用于自动化地预分割模型数据库，利于基于部件组合的探索式建模技术的发展。A

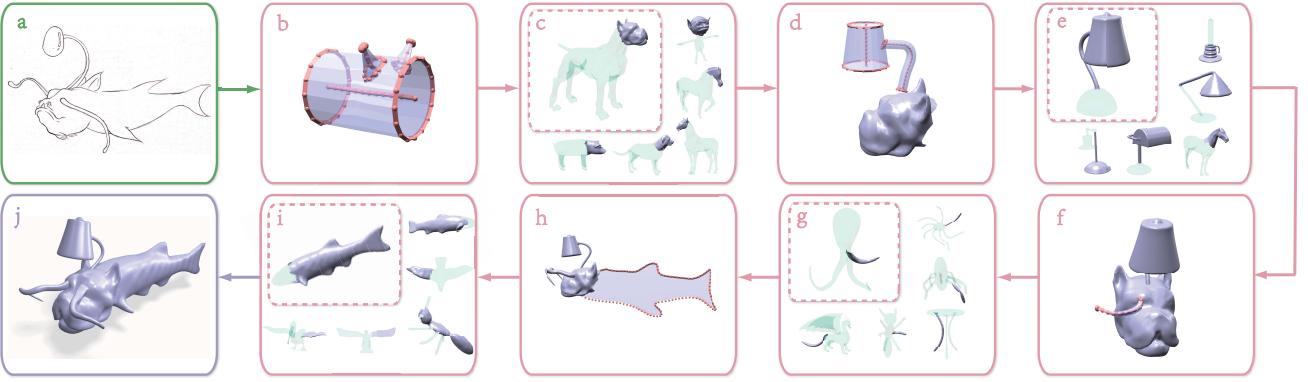


Figure 1: 给一张概念设计图 (a) 做为参考，用户使用我们的方法非常容易地创建具有丰富几何细节的模型 (j)。首先，系统基于草图生成三维代理模型 (b, d, f, h)。接着，代理模型用于实时地从数据库提取个性化的候选部件。我们的方法创建的模型可以由非典型部件（如 (c) 内动物头部分与 (e) 内台灯上半部分）与来自不同类模型的部件（如 (c) 内部件来自四肢动物类，(e) 内部件来自台灯，(g) 内部件来自章鱼）构成。

我们的技术不需要预分割与标记的模型数据库，而是基于用户草图实时分割部件。因此，我们的分割算法不仅速度快，满足实时交互的要求，还需要确保分割部件与用户草图相匹配。这就需要我们提出一种实时的轮廓感知模型分割算法。

**基于草图的模型搜索技术** 我们的工作与基于草图的模型搜索技术 [FMK<sup>+</sup>03, ERB<sup>+</sup>12] 密切相关。这类技术要求用户绘制模型轮廓，通过“二维-三维”的全局匹配，搜索得到三维模型。正是由于草图简单易用，Lee等人 [LF08] 将草图式模型搜索与组合式造型融合到统一的框架。这样的技术既集成了草图简便易用的特点，又具有组合式造型功能强大的优点。Xie等人 [XXM<sup>+</sup>13] 提出一种基于上下文的组合式造型技术，用户通过草图勾勒期望部件的轮廓，系统基于当前模型的已有部件提示可用部件。针对大多数用户缺少绘画技术的情况，Fan等人 [FWX<sup>+</sup>13] 提出 shadow guidance 的思想，在草图绘制面板上动态提示可用部件的轮廓，以此辅助用户绘制草图。

总得来看，这些技术都需要预分割或标记的模型数据库，从而将部件搜索简化为“二维-三维”全局匹配问题。同时，它们仅允许用户使用预先分割出的部件，极大地限制了造型空间。而我们的方法使用未做预分割的数据库，实时提取符合用户意图的部件，扩展了造型空间。

### 3 方法概述

**用户交互** 用户可以先从数据库内选择一个基础模型（某部件或者某模型）。接着，用户绘制草图创建代理模型，以表示造型意图（第5.1节）。系统实时地从数据库内提取一系列与草图相匹配的候选部件。与现有方法不同，我们提示的部件可以是非常规部件，而现有方法仅可提示预分割的部件（如图2）。

用户从建议列表内选择满意的部件。该部件自动与当前模型组合到一起。如果没有满意的部件，用户可以编辑代理部件重新搜索，得到更多建议部件。该过程不断重复，直至用户满意。

**离线阶段** 离线阶段，我们建立快速匹配数据结构（第4.2节）与数据库模型的超面片图表示（第5.2.1节），用于实时提取候选部件。

为了建立快速匹配数据结构，我们首先提取数据库模型在不同视角下的轮廓（boundary contour） [DFRS03]。然后，将所有数据库模型轮廓组织到  $k$ NN 图内，用于轮廓快速匹配。

为了建立数据库模型的超面片图表示形式，我们将模型分割成200个片断，每个片断做为一个超面片，并为相似。将数据库模型表示成超面片图形式是为了候选部件的快速提取。

**在线阶段** 在线阶段，用户创建代理模型后，我们基于轮廓匹配从数据库内提取出一组与之相匹配的候选部件（第5.2节）。用户可以对候选部件进一步优化（第6节）。

### 4 快速匹配数据结构

我们通过二维轮廓局部匹配实现代理模型与数据库模型的局部匹配。该方法的最大挑战是二维轮廓快速匹配。将代理模型轮廓逐一地与数据库模型轮廓匹配明显不可行，因为匹配时间会随模型数据库的增大而急剧增加。因此，我们提出一个轮廓索

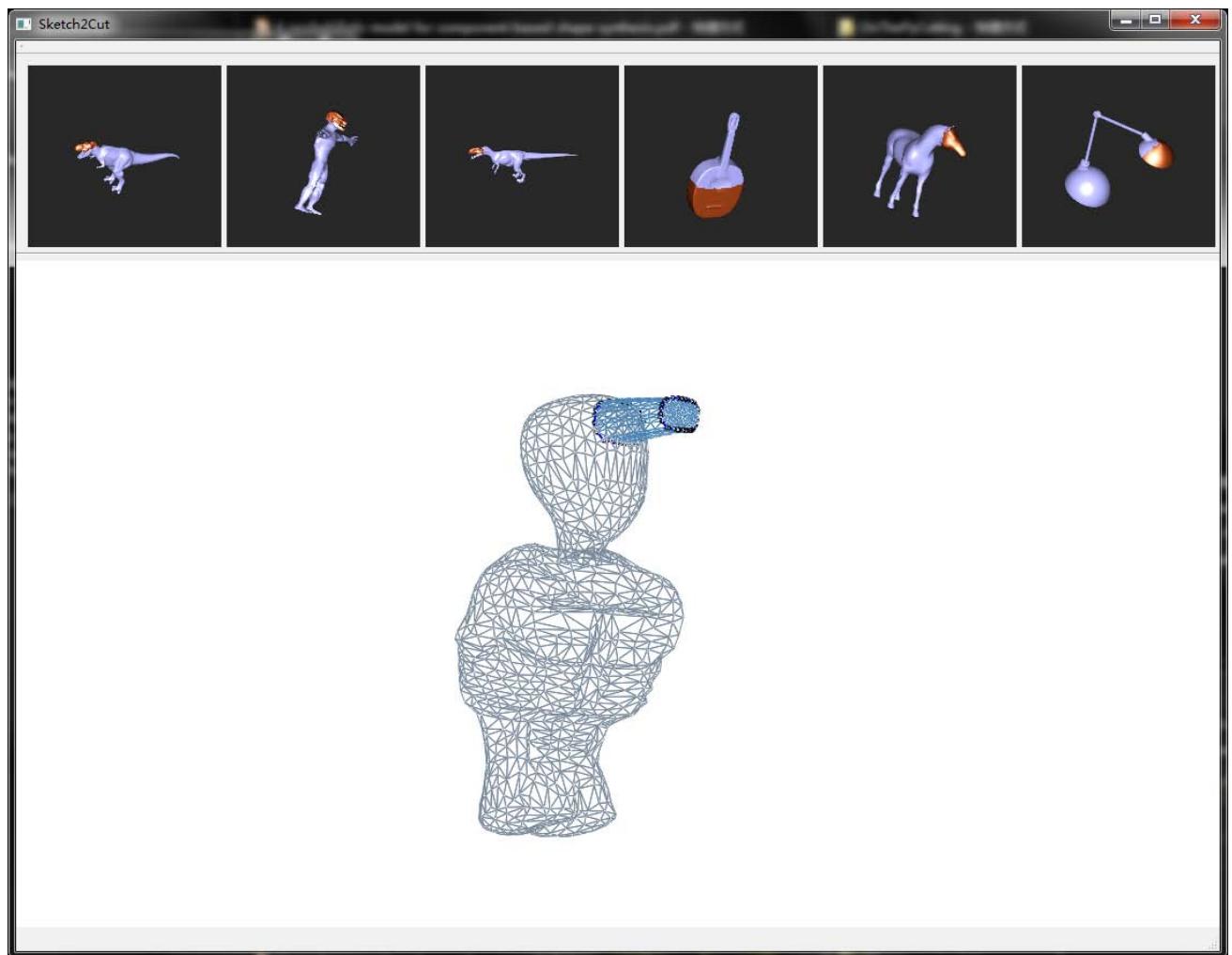


Figure 2: 交互界面截图。界面内展示了当前模型，用户草图，代理模型，系统建议的非常规部件。

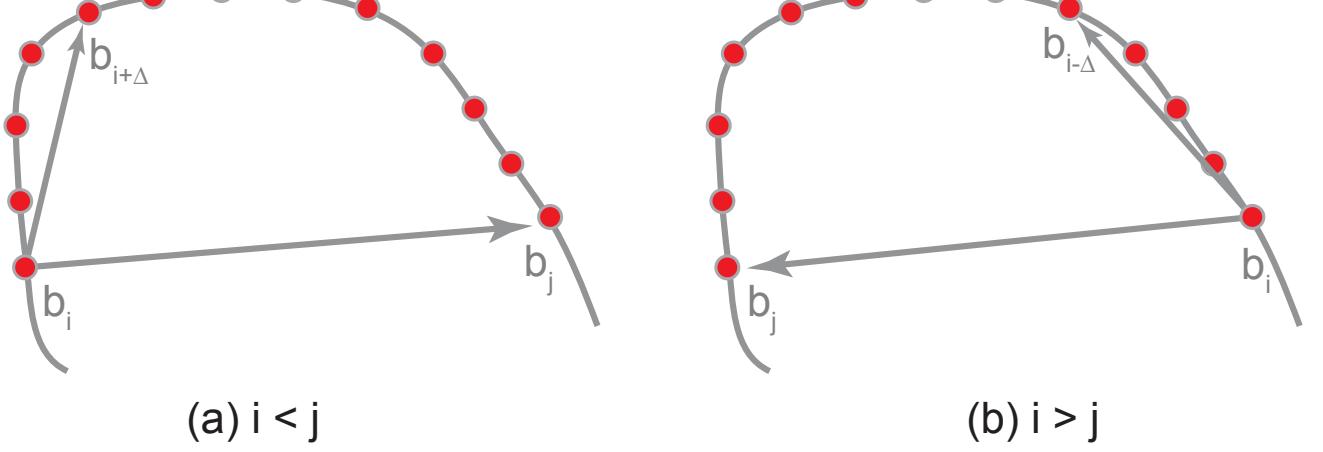


Figure 3: 角度描述符示意图。

引数据结构来加速轮廓匹配过程。

#### 4.1 轮廓描述符

为了匹配轮廓，我们首先提取轮廓描述符。类似 [XJK<sup>+</sup>16]，我们使用角度矩阵做为轮廓描述符。该描述符描述了轮廓上采样点的分布情况。为了提取轮廓描述符，我们首先在轮廓上等距采样；然后为任意一对采样点( $b_i, b_j$ )计算角度描述符 $\alpha_{ij}$ （如图 3）：

$$\alpha_{ij} = \begin{cases} \frac{\langle \overrightarrow{b_i b_j}, \overrightarrow{b_i b_{i+\Delta}} \rangle}{\langle \overrightarrow{b_i b_j}, \overrightarrow{b_i b_{i-\Delta}} \rangle} & \text{if } i < j, \\ \frac{\langle \overrightarrow{b_i b_j}, \overrightarrow{b_i b_{i-\Delta}} \rangle}{\langle \overrightarrow{b_i b_j}, \overrightarrow{b_i b_{i+\Delta}} \rangle} & \text{if } i > j, \\ 0 & \text{if } \|i - j\| \leq \Delta \end{cases}$$

其中 $\Delta = 4$ 。 $\langle l_1, l_2 \rangle$ 代表直线 $l_1$ 与 $l_2$ 间的夹角。轮廓上所有采样点对的角度描述符构成的矩阵即为轮廓描述符 $\mathbf{A}$  ( $A_{ij} = \alpha_{ij}$ )。轮廓 $S$ 与 $S'$ 做匹配定义为：

$$J(\mathbf{A}, \mathbf{A}') = \min_k \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\mathbf{A}_{[k+i, k+j]} - \mathbf{A}'_{[i, j]})^2, \quad (1)$$

其中 $\mathbf{A}$ 是轮廓 $S$ 的描述符，大小为 $M \times M$ ； $\mathbf{A}'$ 是轮廓 $S'$ 的描述符，大小为 $N \times N$  ( $N \leq M$ )。 $\mathbf{A}_{[i, j]}$ 与 $\mathbf{A}'_{[i, j]}$ 表示矩阵元素。 $S$ 内起点为 $k + i$ 终点为 $k + j$ 的轮廓段是 $S$ 上与 $S'$ 相匹配的轮廓段。 $J(\mathbf{A}, \mathbf{A}')$ 是 $S$ 与 $S'$ 的匹配分数；分数越小，轮廓越相似。

#### 4.2 建立轮廓索引数据结构

在预处理阶段，我们为数据库模型提取 $T_M = 14$ 个视角下的轮廓，包括6个规范视角（Canonical side views），8个角视角（Corner views）。通过这种方式，我们得到数据库模型的二维轮廓 $\Gamma_M = \{S_i | 1 \leq i \leq T_M\}$  ( $S_i$ 表示第*i*个视角下的轮廓)。

对每一个数据库模型轮廓，我们按采样距离 $\Theta = \{3, 6, 9, 12, 15\}$ 得到一组轮廓描述符。然后，我们将所有数据库模型轮廓组织到kNN图 [SK02]内。图的顶点表示模型轮廓。对每一个顶点，我们找到与之最相似的30个顶点并建立边。边上存储两顶点的匹配分数及对应关系。

接着，我们在kNN图上采样出一组种子点。首先，我们随机选择一个顶点做为种子点，然后采用最好优先（best-first search）方法遍历全图。如果当前顶点与种子点的差异值高于0.7，我们令当前顶点为新的种子点并重新开始遍历过程。如上过程不断重复，直至图上所有顶点被访问到。通过这种方式，我们从kNN图内采样出一组种子点。

为了加速建立kNN图，在寻找与当前轮廓最相似的轮廓时，我们仅考虑与当前轮廓对应的模型最相似的前80个模型的轮廓。三维模型相似度通过3D shape distribution [FMK<sup>+</sup>03]衡量。



Figure 4: 轮廓索引建立过程图示。

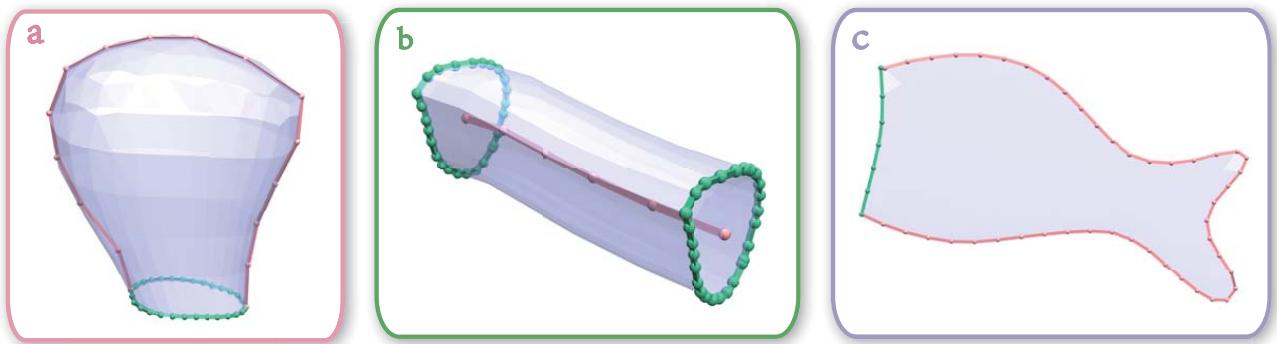


Figure 5: 三种基础代理模型。 (a) 广义球体。 (b) 广义柱体。 (c) 薄板模型。

### 4.3 轮廓索引辅助的轮廓匹配

在搜索阶段，给一个代理模型轮廓 $C$ ，我们首先将它与所有的种子轮廓做匹配。匹配结果存储在优先队列内，匹配分数最低的位于队首。接着，我们以最好优先方式遍历整个 $k$ NN图：优先队列队顶出队，将其邻居与 $C$ 做匹配，匹配结果入队。该过程中，我们可以利用边上存储的对应信息，快速确定当前轮廓与 $C$ 相匹配的轮廓段（称为初始轮廓段）。为了降低积累误差，初始轮廓段还需要进一步优化。优化方法如下：1) 将初始轮廓段的长度延伸至 $0.3N_{frag}$  ( $N_{frag}$ 是初始轮廓段的长度)；2) 在延伸轮廓段上找到与 $C$ 相匹配的部分，即为最终的匹配轮廓段。为了进一步加速匹配过程，不考虑匹配分数大于0.9的顶点。

## 5 候选部件搜索

本章中，我们介绍如何从未做预分割的模型数据库内快速、精确地搜索出与代理模型相匹配的候选部件。

### 5.1 代理模型创建

与 [FKS<sup>+</sup>04]类似，我们使用三维代理模型从数据库内搜索部件并辅助部件组合。我们提供三种基本代理模型，使用户更容易表达造型意图。用户输入的草图也可直接用于部件搜索。

我们提供三种基本代理模型（如图5）：广义球体、广义柱体、薄板模型。广义球体通过Nealen等人 [ATOM07]的方法生成。广义柱体通过两笔生成：首先，在当前模型上画一个圆圈表示柱体底座；然后，画一条曲线指示柱体中心线。中心线位于用户视线方向与圆圈起点定义的空间平面上。广义柱体的两端可以放缩，得到圆锥或圆台。创建薄板模型，用户首先在当前模型上画一条曲线，再画出薄板模型轮廓线。该轮廓线位于用户视线方向与曲线起点定义的空间平面上。我们的系统允许用户对三种基础代理模型施加放缩、切割、变形等操作。更复杂的代理模型可以通过基础代理组合得到。

用户创建代理模型过程中，系统记录每次操作的视角及操作前后代理模型体积的变化。代理模型建成后，我们提取代理模型

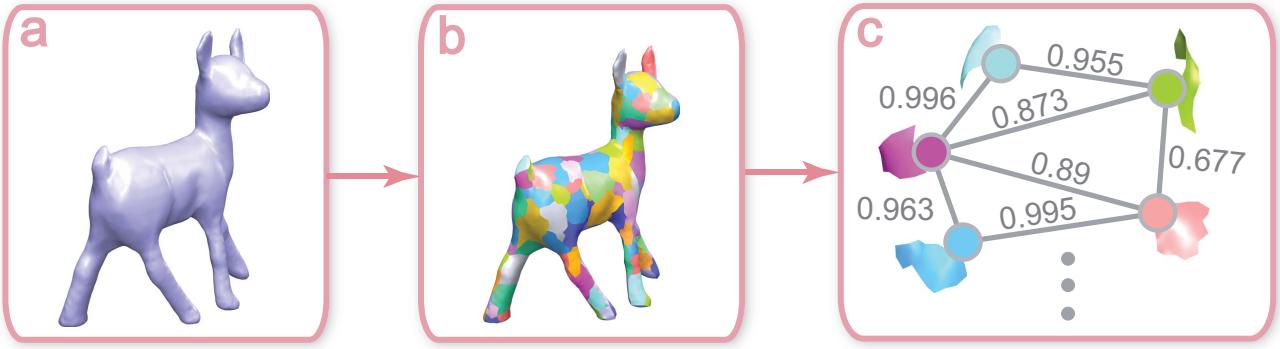


Figure 6: 三维模型超面片图建立过程示意图。我们将三维模型（a）分割成200个片断（c）。一个片断视为一个超面片，做为超面片图上一个顶点。在超面片图上，相邻超面片间用带权重的边相连。

在体积变化最大的视角下的二维轮廓（该轮廓称为主轮廓），用于描述用户的建模意图。体积变化次之的视角下提取的轮廓是副轮廓，用于候选部件排序。最后，我们采用第4.1节提出的方法提取主、副轮廓的描述符（采样距离为9）。

## 5.2 候选部件搜索

给一个三维代理模型与一个数据库模型，我们的目标是在数据库模型上找到与代理模型最相似的部分（称为候选部件）。首先，我们通过代理模型轮廓与数据库模型轮廓做匹配得到候选轮廓段。然后，基于候选轮廓段与数据库模型的超面片图表示提取候选部件。

**候选轮廓段搜索** 依据采样距离，我们将一个数据库模型的所有轮廓与代理模型主轮廓的匹配结果划分为 $\|\Theta_M\|$ 个匹配结果组。为了选择出候选轮廓段，我们首先选择出最佳匹配结果组。给定代理轮廓 $C$ ，数据库模型轮廓 $\Gamma_M$ 及其采样距离 $\Theta_M$ ，最佳匹配结果组选择问题定义为：

$$\min_{\theta_i \in \Theta_M} \sum J_{\theta_i}^j,$$

其中 $J_{\theta_i}^j$ 表示在采样距离 $\theta_i$ 下最好的第 $j$ 个匹配结果的匹配分数（见公式（1））。我们从最佳匹配结果组内选择出最好的前3个匹配结果，并由此得到候选轮廓段。

### 5.2.1 三维模型的超面片图表示

我们在预处理阶段为数据库模型建立超面片图表示形式 [XJK<sup>+</sup>16]。图上顶点表示超面片。图上边描述超面片间相邻关系。边上权重表示相邻超面片合并成更大超面片的先验概率（如图 6）。

为了建立超面片图，我们首先将模型分割成200个片断 [HKG11]。一个片断视一个超面片。相邻超面片用一条边相连。

为了计算边权重，我们对三维模型做多次Randomized Segmentation [GF08]。对每一次分割结果，我们为每个超面片建立一个直方图，描述该超面片属于各分割片断的概率（如图7）。超面片 $s$ 针对分割部件 $g$ 的直方条 $h_s^g$ 定义为：

$$h_s^g = \frac{\sum_{f \in F(g) \cap F(s)} A(f)}{\sum_{f \in F(s)} A(f)},$$

其中 $F(x)$ 是属于 $x$ 的面片集合， $A(f)$ 是面片 $f$ 的面积。相邻面片 $s$ 与 $s'$ 位于同一部件的概率 $P(s, s')$ 为两超面片的直方图 $H$ 与 $H'$ 的 $\chi^2$ 的距离：

$$P(s, s') = 1 - \chi^2(H, H') = 1 - \frac{1}{2} \sum_{g_i \in G} \frac{(h_s^{g_i} - h_{s'}^{g_i})^2}{(h_s^{g_i} + h_{s'}^{g_i})^2},$$

其中 $G = \{g_i | 1 \leq i \leq N\}$ 是一次分割结果的所有片断。为了得到超面片图上边的权重，我们首先将相关的两超面片在不同分割结果上的概率累加，再通过分割次数归一化。

### 5.2.2 候选部件搜索

给一个候选轮廓及数据库模型，我们的目标是在数据库模型上找到与候选轮廓最匹配的候选部件。我们将候选部件搜索问题

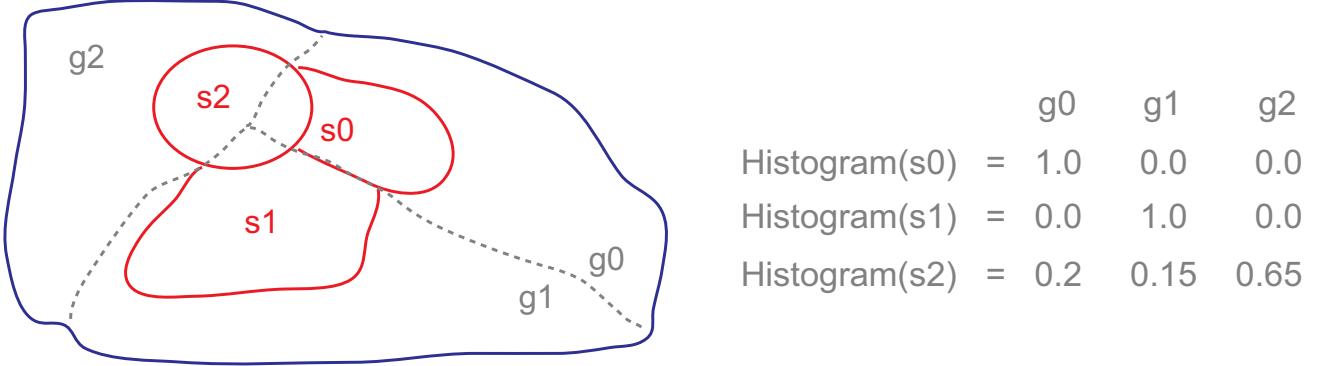


Figure 7: 超面片在分割片断上的分布示意图。 $s_0$ ,  $s_1$ 和 $s_2$ 是超面片。 $g_0$ ,  $g_1$ 与 $g_2$ 是通过Randomized Segmentation [GF08]得到的分割片断。这个图左侧[s0,s2]改样子，右侧改数据。

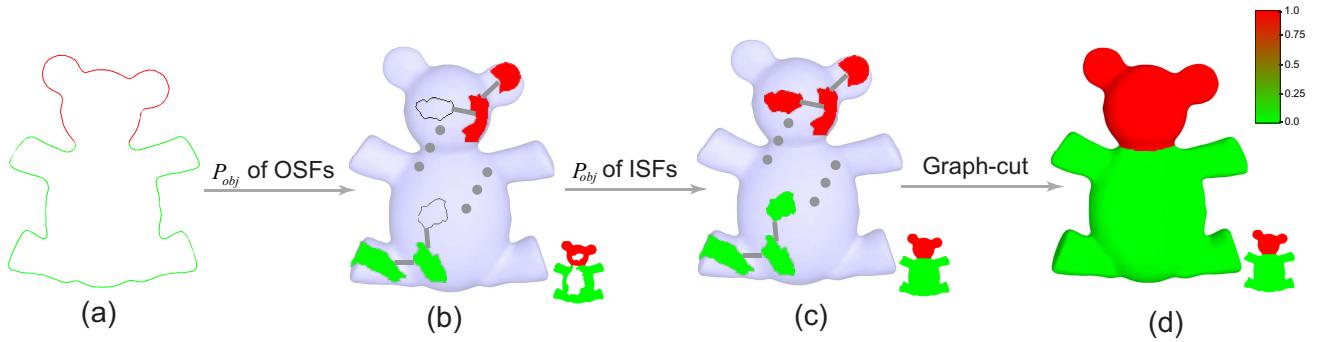


Figure 8: 候选部件搜索算法流程。给一个候选轮廓 ((a) 内红色部分)，我们首先计算外超面片obj-prob (b)；然后计算内超面片obj-prob (c)；最后通过graph-cut技术得到整个候选部件 (d)。优化该图；颜色条字再大些。inner super-face, outer super-face

形式化为超面片图上的子图搜索问题（如图 9），其目标函数如下：

$$\min_L \left( \sum_{v_i \in V} D_{v_i}(L_{v_i}) + \sum_{(v_i, v_j) \in E} S_{v_i, v_j}(L_{v_i}, L_{v_j}) \right),$$

其中超面片图 $G = (V, E)$ ， $V$ 表示顶点集合， $E$ 表示边集合。 $L = \{L_{v_i} | v_i \in V\}$ 是超面片图 $G$ 的一个标记。 $D_{v_i}(L_{v_i})$ 是数值项，表示将标记 $L_{v_i}$ 置给 $v_i$ 的能量耗费。 $S_{v_i, v_j}(L_{v_i}, L_{v_j})$ 是光滑项，表示相邻顶点 $v_i$ 与 $v_j$ 具有相同标记的能量耗费。

该目标函数采用min-cut/max-flow技术求解 [BK04]。n-links的容量是图内边的权重。t-links的容量定义为：

$$D_{v_i}(1) = -\ln \Pr(I_{v_i} | obj'),$$

$$D_{v_i}(0) = -\ln(1 - \Pr(I_{v_i} | obj')),$$

其中 $\Pr(I_{v_i} | obj')$  (obj-prob) 表示 $v_i$ 属于候选部件的概率。

对外部超面片（自身轮廓与模型轮廓有重叠的超面片），obj-prob通过其自身轮廓与候选轮廓的重合程度得到。超面片图内，除了外部超面片之外的超面片称为内部超面片。我们认为，在超面片图内外部超面片与内部超面片的obj-prob相一致。因此，整张图内的obj-prob可以形化为：

$$\mathbf{AX} = \mathbf{b},$$

条件为：

$$\mathbf{X} \geq 0,$$

其中 $\mathbf{A} = \mathbf{I} - \mathbf{A}'$ ， $\mathbf{A}'$ 是带权重的相邻矩阵，在ISFs之间。 $\mathbf{X}$ 是ISF的概率。 $\mathbf{b} = \mathbf{BY}$ ，其中 $\mathbf{B}$ 是ISF与OSF间的带权相邻矩阵， $\mathbf{Y}$ 是OSF的概率。

### 5.3 候选部件排序

我们基于候选部件与代理部件的兼容性对候选部件排序。在计算兼容性时，我们考虑如下因素：1) 候选部件的二维投影与代理部件的相似性；2) 候选部件与代理部件的几何相似性；3) 候选部件的背景与代理部件的相似性。给一个候选部



Figure 9: 类似sketch-to-design Fig 4, 来张图, 展示三个量



Figure 10: 图内要proxy与model align到一起; 三个ROI区域不同颜色显示出; 左右两张子图, 左侧子图给出预计算出来的各种量, 右侧给出优化结果; 反正这张图类似Fig 8, 展示这部分算法流程。

件 $p$  ( $p$ 取自数据库模型 $M$ ) 与代理部件 $B$ , 两者的兼容性定义如下:

$$J(p, B) = \omega T_p(p, B) + \xi T_g(p, B) + \psi T_c(p, B),$$

其中 $\omega = 0.8$ ,  $\xi = 0.15$ ,  $\psi = 0.05$ 。 $T_p(\cdot, \cdot)$ 计算模型二维投影的相似性。 $T_g(\cdot, \cdot)$ 计算模型的几何相似性。 $T_c(\cdot, \cdot)$ 计算模型背景的相似性。

**二维投影相似性**  $T_p(p, B)$ 定义如下:

$$T_p(p, B) = \omega_m \frac{1}{\min_i J(\mathbf{A}_m, \mathbf{A}_i)} + \omega_a \frac{1}{\min_i J(\mathbf{A}_a, \mathbf{A}_i)},$$

其中 $\mathbf{A}_m$ 与 $\mathbf{A}_a$ 是 $p$ 的主轮廓与副轮廓的描述符, 在最佳采样距离下。 $\mathbf{A}_i$ 是 $B$ 在最佳采样距离下第 $i$  ( $1 \leq i \leq T_M$ ) 个视角下的轮廓的描述符。 $J(\cdot, \cdot)$ 的定义见公式 (1)。 $\omega_m = 0.9$ ,  $\omega_a = 0.1$ 。

**几何相似性**  $T_g(p, B)$ 定义如下:

$$T_g(p, B) = \alpha D(p, B) + \beta D(B, p),$$

其中 $D(\cdot, \cdot)$ 是largest common pointset measure [AMCO08]。 $\alpha = 0.9$ ,  $\beta = 0.1$ 。

**背景相似性**  $T_c(p, B)$ 定义如下:

$$T_c(p, B) = T_g(X, \Sigma),$$

其中 $\Sigma$ 指 $p$ 的背景 ( $M$ 上与 $p$ 相邻的部件的集合)。 $X$ 指 $B$ 的背景 (当前模型上与 $B$ 相邻的部件的集合)。

候选部件基于兼容性分数由高低排列。如果用户需要多样性的部件建议, 我们采用技术Maximal Marginal Relevance criterion [CK10]排列候选部件。

## 6 候选部件优化

首先, 我们在候选部件上取得与切割线距离为 $l_{dia}$ 的面片 (记为 $\Omega_{obj}$ ,  $l_{dia}$ 表示候选部件包围盒对角线长)。然后, 用类似方法, 在候选部件的补部件上得到 $\Omega'_{obj}$ , 在代理部件上得到 $\Omega_{proxy}$ 。接着, 我们大致地对齐 $\Omega_{obj}$ 与 $\Omega_{proxy}$ 。候选部件优化问题

Algorithms	Database Size	114	205	513
Brute-force	59.58	106.99	240.07	
<i>k</i> NNNG	6.94	12.66	28.93	
CPU and GPU Acceleration	0.36	1.03	1.48	

Table 1: 时间以秒计算。Brute-force版本处理数据库模型与模型轮廓一个接一个地。可以观察到采用加速结构后，我们的算法效率不断提高。*k*NNNG表示我们的轮廓索引数据结构。

被形式化成能量优化问题，其目标函数如下：

$$E(X) = \sum_{v_i \in \nu^f} E_d(x_{v_i}) + \sum_{(v_i, v_j) \in \varepsilon^f} E_s(x_{v_i}, x_{v_j}),$$

其中 $G^f = (\nu^f, \varepsilon^f)$ 是 $\Omega_{obj} \cup \Omega'_{obj}$ 的对偶图。 $\nu^f$ 是顶点集合。 $\varepsilon^f$ 是边集合。 $X = \{x_{v_i} | v_i \in \nu\}$  ( $x_{v_i} = \{0, 1\}$ ) 是 $G^f$ 的一个标记。 $E_d(x_{v_i})$ 是数值项，表示将标记 $x_{v_i}$ 置给 $v_i$ 的能量耗费，定义如下：

$$E_d(x_{v_i}) = \begin{cases} -\ln D_{obj}(v_i), & x_{v_i} = 1, \\ -\ln(1 - D_{obj}(v_i)), & x_{v_i} = 0, \end{cases}$$

其中 $D_{obj}(v_i)$ 表示将标记1置给 $v_i$ 的概率，定义如下：

$$D_{obj}(v_i) = \omega_c g_c(v_i) + \omega_s g_s(v_i),$$

其中 $\omega_c = 0.5 + 0.2d$  ( $d$ 表示 $v_i$ 与初始切割线间距)。 $\omega_s = 1 - \omega_c$  ( $\omega_s \geq 0$ )。 $g_s(v_i)$ 是 $v_i$ 与 $\Omega_{proxy}$ 间的距离，定义如下：

$$g_c(v_i) = \begin{cases} \frac{\text{vol}(\Omega_{obj})}{\text{vol}(\Omega_{proxy})}, & v_i \in \Omega_{obj}, \\ \frac{\text{vol}(\Omega'_{obj})}{\text{vol}(\Omega_{proxy})}, & v_i \in \Omega'_{obj}, \end{cases}$$

$V(\cdot)$ 计算模型体积。 $E_s(x_{v_i}, x_{v_j})$ 是光滑项，计算相邻顶点 $v_i$ 与 $v_j$ 具有相同标记的能量耗费，定义如下：

$$E_s(x_{v_i}, x_{v_j}) = \begin{cases} 0, & x_{v_i} = x_{v_j}, \\ \exp\left(\frac{2\theta}{\pi}\right) & x_{v_i} \neq x_{v_j}, \end{cases}$$

其中 $\theta \in [0, \frac{\pi}{2}]$ 是面片 $v_i$ 与 $v_j$ 的公共边与分割线拟合平面间的夹角。

候选部件优化是可选项，不会对最终造型结果有太大影响。

## 7 实验结果

我们用C++语言实现系统原型，并在标准台式机 (3.5 GHz Intel Core I7-3770K processor, 8GB memory, and an Nvidia GeForce GTX 660 GPU video card) 上做实验。图1给出一个例子的造型流程。给一个概念设计，用户通过草图表达其造型意图。系统基于草图生成三维代理模型并实时地从数据库内提取建议部件。这些建议部件有非典型部件 ((c) 内的头部部件)，由一组常规部件构成的部件 ((e) 内的台灯部件) 及来自不同类模型的部件 (例如，(c) 内来自狗模型的部件，(g) 内来自章鱼模型的部件)。图11给出更多的造型结果。可见，用户可以使用我们的技术生成很多极具创意的结果。

**时间统计** 大概需要平均针8秒为一个模型建立其超面片图表示。对200个模型的数据库，建立轮廓索引需要约2小时 (CUDA与OpenMP®加速)。用户平均画3笔建造一个代理部件，平均需要22秒。轮廓匹配平均需要0.078毫秒。给一个代理轮廓及模型在各视角下的轮廓，平均需要0.337秒找到其候选轮廓 (包括轮廓匹配步骤)。需要0.15毫秒找到初始候选部件。部件优化平均需要0.84秒。*k*NN图辅助的轮廓匹配得到的性能提升如表1。

### 7.1 *k*NN图

### 7.2 对比

为了定性与定量地评估我们方法的效果，我们对比了如方法：

- **Ours.** 本文提出的方法



Figure 11: 系统造型结果的例子。在每张子图内，我们给出用户绘制的草图（紫色），生成的代理模型（浅蓝色），对应的数据库模型及提取出的部件（数据库模型上的蓝色部件）。注意结果模型都是由非常规部件或来自不同类的部件构成。

STAT. Type	Ave.	SD.	Max.	Med.	Min.
Ours::Atypical	0.26	0.24	0.67	0.25	0.0
Ours::Regular	0.74	0.24	1.0	0.75	0.33
PreSeg::Atypical	0.0	0.0	0.0	0.0	0.0
PreSeg::Regular	1.0	0.0	1.0	1.0	1.0

Table 2: 非典型部件与典型部件在模型内所占比例统计结果。“Ours::Atypical”与“Ours::Regular”表示Ours系统造型结果内非典型部件与典型部件的比例。“PreSeg::Atypical”与“PreSeg::Regular”分别表示PreSeg系统造型结果内非典型部件与典型部件的比例。

STAT. Type	Ave.	SD.	Max.	Med.	Min.
Ours::Total	3.59	1.03	5	3	1
PreSeg::Total	3.06	0.74	5	3	1
Ours::Atypical	4.47	0.72	5	5	2
Ours::Regular	3.02	0.78	5	3	1
PreSeg::Atypical	-	-	-	-	-
PreSeg::Regular	3.06	0.74	5	3	1

Table 3: 建议部件得分统计结果。“Ours::Total”与“PreSeg::Total”分别表示Ours系统与PreSeg系统建议部件的得分。“Ours::Atypical”与“Ours::Regular”分别表示Ours系统建议的非典型部件与典型部件的得分。“PreSeg::Atypical”与“PreSeg::Regular”分别表示PreSeg系统建议的非典型部件与典型部件的得分。

- **PreSeg.** 基于本文方法实现的基于预分割的模型数据库的方法。该方法实现细节如下：1) 每个数据库模型都预分割成常规部件（典型语义部件）。比如，人模型分成4部分：头部分，躯干，胳膊，腿。2) 为每一个部件提取各视角下的轮廓线。3) 将所有部件的轮廓线组织在kNN图内。图内顶点是部件轮廓。通过部件轮廓全局匹配为顶点建立边。3) 在运行阶段，输入三维代理部件，系统采用类似Ours的方法通过全局匹配返回一组候选部件。

我们邀请4位三维造型艺术家与16位学生参与用户测试。4位艺术家中2位具有6年以上三维造型经验，另外2位有4年经验。所有的参与者位于22到31岁之间。我们将参与者分成A, B两组。每一组都由2位艺术家与8位学生构成。在每一组内男女人数相同。我们设计了两个开放式造型任务：

- **生物造型** 假设你是Valve®公司的一名三维造型师。项目组计划为电子游戏Dota 2®加入一个新英雄。请使用我们的造型系统设计该角色的原型。
- **日常用品造型** 假设你是创意生活公司的一名设计师。要求你为客户设计一个个性化的日常用品（例如，个性化的台灯，创意服饰等）。请使用我们的系统设计该日常用品的原型。

**用户测试第一阶段** 第一阶段的目标是验证非典型部件（非常规部件，由一组常规部件构成的部件）对用户造型是否有用。我们让所有的测试者完成两个造型任务。A组用户首先使用Ours系统，然后是PreSeg系统。B组用户首先使用PreSeg系统，然后是Ours系统。在用户使用系统之前，我们先用20分钟时间向其介绍系统操作方法。之后，用户花30分钟操作系统，包括三维代理部件造型，浏览建议部件与组合部件。图12展示了一位测试者正在做用户测试。该阶段的一部分造型结果见图13。我们统计了非典型部件与典型部件在各模型内的比例（表2）。很明显，Ours系统的造型结果可以由非典型部件构成。然而，PreSeg系统的造型结果仅由典型部件构成。Ours系统提供的非典型部件有利于用户的造型。

**用户测试第二阶段** 在第二阶段，我们的目标是验证Ours系统建议的部件是否比PreSeg系统建议的更能激发造型灵感。A组测试者先使用PreSeg系统，然后是Ours系统。B组用户相反。我们要求所有测试者为系统提供的所有建议打分。分数分为5等：5分表示非常具有启发性，1分表示非常平庸。

表3给出了Ours系统与PreSeg系统建议的部件的得分情况。由表我们可以很明显地发现Ours系统建议部件的得分高于PreSeg系统的。在图14内，我们给出一组代表性的建议部件及其得分。

为了定量地评估两个系统建议部件的非规则性，我们设计了非规则度量标准。直观地讲，不规则度量描述了当前部件与同一模型上相关典型部件间的差异程度。给一个部件 $p$  ( $p$ 来自三维模型 $M$ )，其非规则程度值 $IRR(p)$ 定义如下：

$$IRR(p) = \max \{d(p, q_i) | \{q_i\} = r(p)\},$$

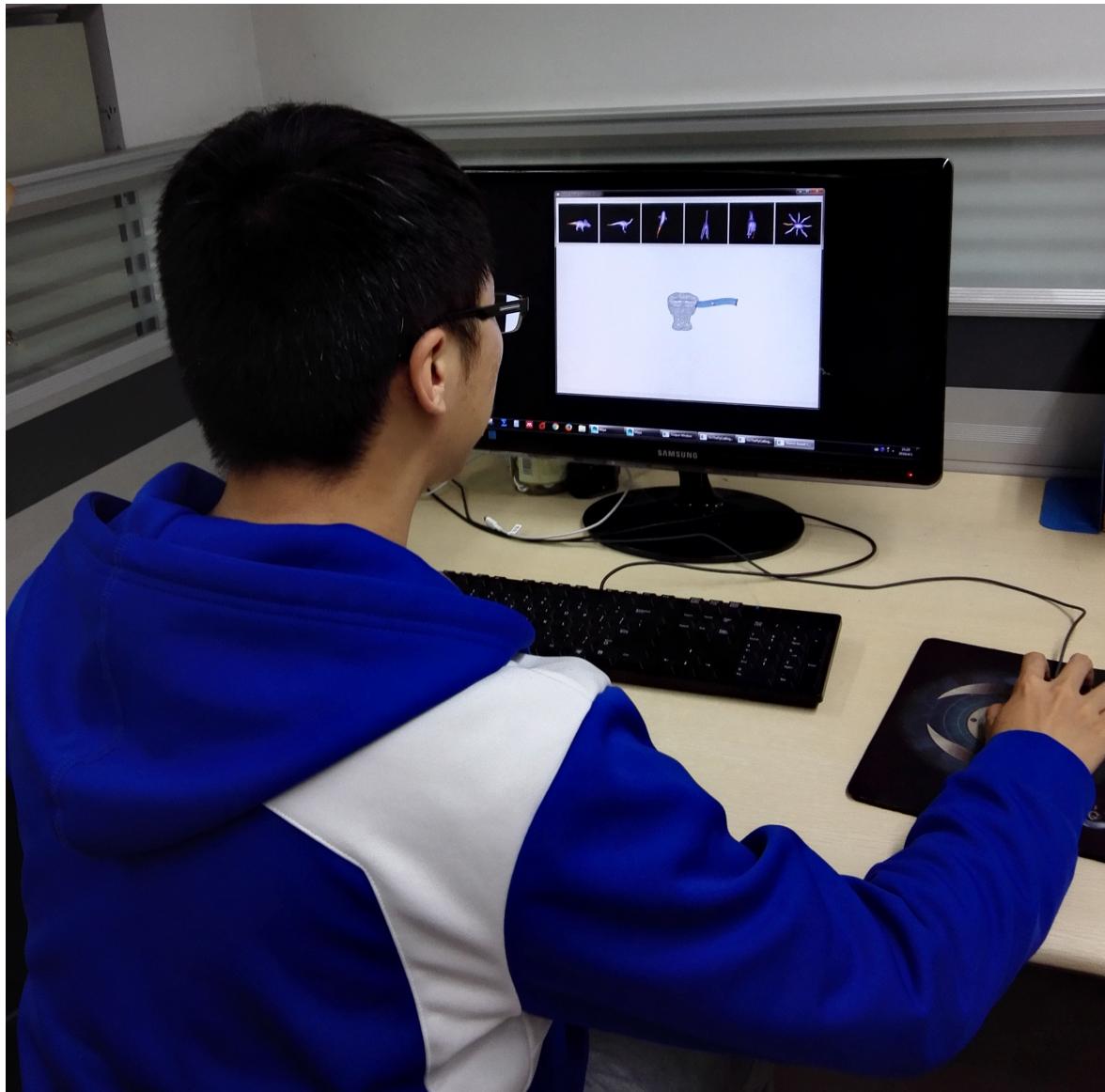


Figure 12: 一位测试者正在操作造型系统完成造型任务。

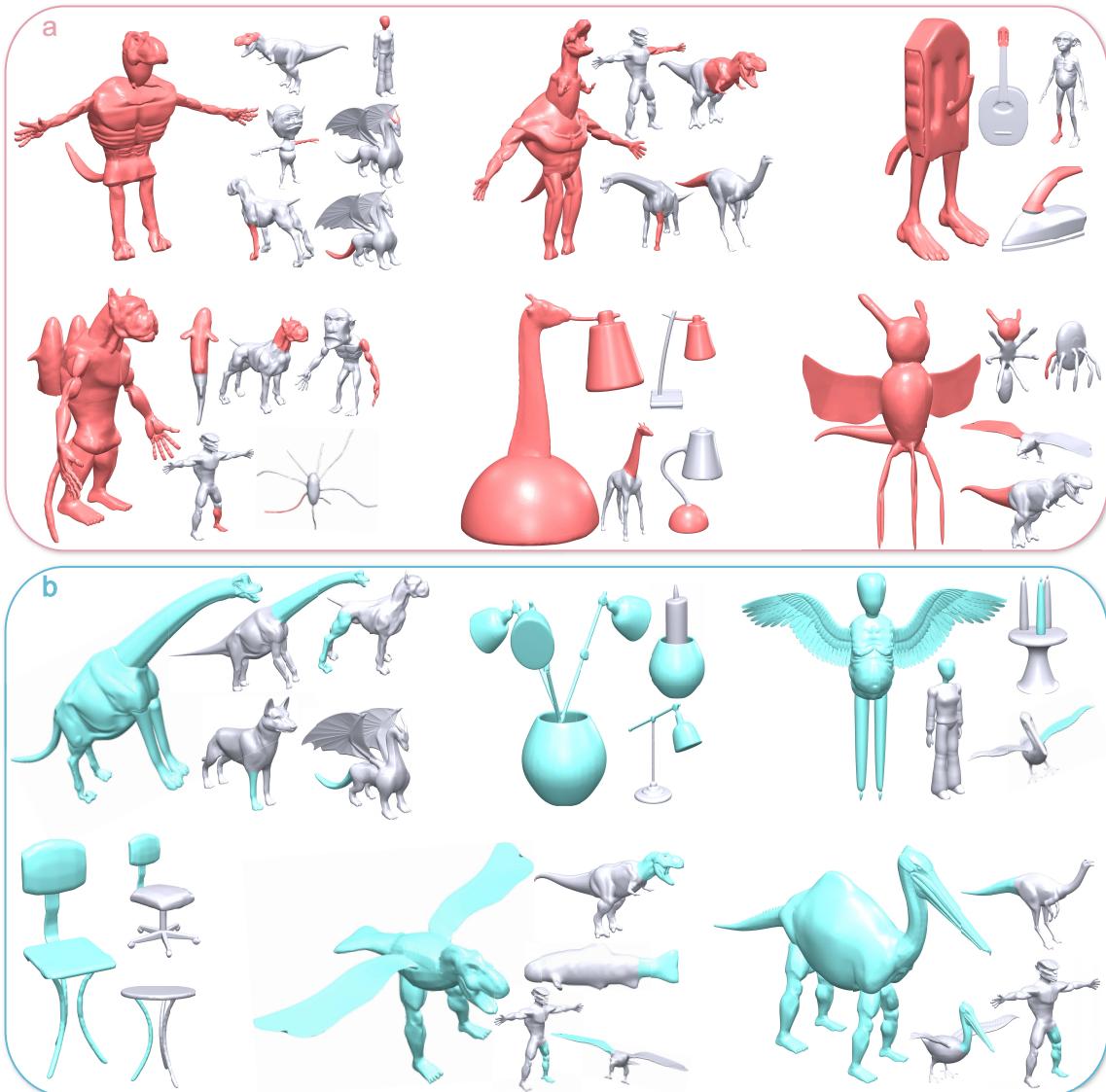


Figure 13: Ours系统 (a) 与PreSeg系统 (b) 造型结果。每个造型结果，其构成部件在右侧。

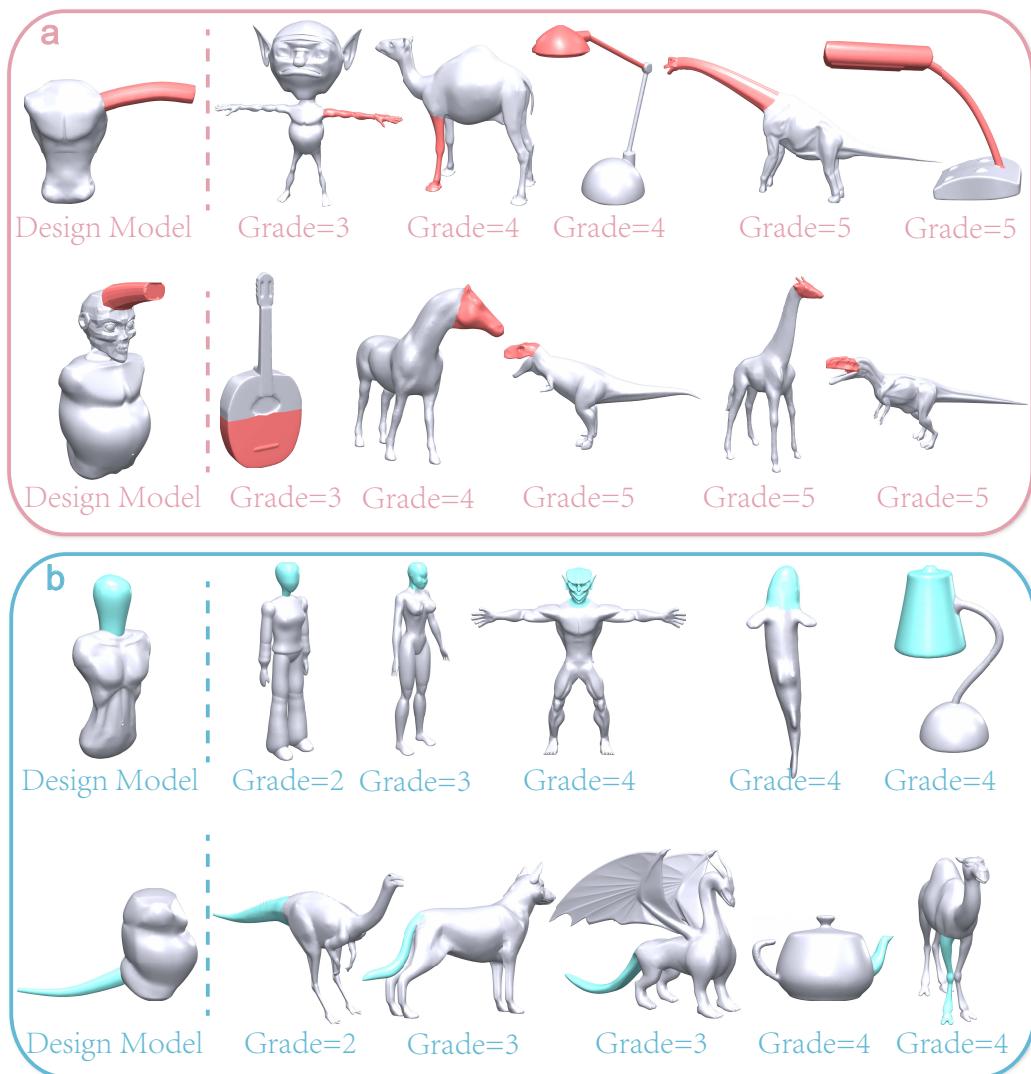


Figure 14: 具有代表性的建议部件及其得分。 (a) 图给出Ours系统建议的部件（红色）。 (b) 图给出PreSeg系统建议的部件（蓝色）。

Type \ STAT.	Ave.	SD.	Max.	Med.	Min.
Ours::Total	0.024	0.033	0.098	0	0
PreSeg::Total	0	0	0	0	0
Ours::Atypical	0.062	0.022	0.098	0.053	0.025
Ours::Regular	0	0	0	0	0
PreSeg::Atypical	-	-	-	-	-
PreSeg::Regular	0	0	0	0	0

Table 4: 非规则度量值统计结果。“Ours::Total”与“PreSeg::Total”分别代表Ours系统与PreSeg系统建议部件的非规则度量值。“Ours::Atypical”与“Ours::Regular”分别代表由Ours系统建议的非典型部件与典型部件的非规则度量值。“PreSeg::Atypical”与“PreSeg::Regular”分别代表PreSeg系统建议的非典型部件与典型部件的非规则度量值。

Type \ Grade	1	2	3	4	5
Ours::Atypical	0.0	0.08	0.03	0.6	0.91
Ours::Regular	1.0	0.92	0.97	0.4	0.09
PreSeg::Atypical	0.0	0.0	0.0	0.0	0.0
PreSeg::Regular	1.0	1.0	1.0	1.0	1.0

Table 5: 各分值内非典型部件与典型部件所占比例统计结果。“Ours::Atypical”与“Ours::Regular”分别表示Ours系统的造型结果内非典型部件与典型部件在各分值内所占比例。“PreSeg::Atypical”与“PreSeg::Regular”分别表示PreSeg系统的造型结果内非典型部件与典型部件在各分值内所占比例。

其中 $r(p)$ 是 $M$ 上与 $p$ 相交的典型部件的集合。 $d(p, q_i)$ 是 $p$ 与 $q_i$ 的Light Field Descriptor [CTSO03]之差的归一化值。

表4给出了两个造型系统建议部件的非规则度量值。由表可以很明显地看出，Ours系统建议的部件相对PreSeg系统的更加不规则。非典型部件导致了非规则性。图15给出了一组代表性的建议部件及其非规则度量值。

我们统计了Ours系统与PreSeg系统提供的非典型部件与典型部件的得分（表3）。可以很明显地发现，非典型部件的平均得分高于典型部件的。表5给出了每一个分值内非典型部件与典型部件所占比例。很显然，非典型部件得分高于典型部件。由此，可以得出结论Ours系统建议的部件相对PreSeg系统的更具启发性；相对典型部件，非典型部件对用户的启发更多。

**用户测试第三阶段** 在第三阶段，测试者被要求完成一份调查问卷。该调查问卷包括10个陈述句，测试者需要在标准5分李克特量表上为每个陈述打分。图16给出调查问题统计结果。总体来看，测试者对Ours系统给予较高评价。他们觉得Ours系统建议的部件更具启发性，尤其是那些非典型部件。他们认为我们的系统对概念设计阶段有很大助益。

与现有的要求预分割及标记的部件建议技术 [XXM<sup>+</sup>13, FWX<sup>+</sup>13]不同，我们的技术不仅可以提供非典型部件，还能提供来自不同类模型的部件。如果允许从不同类模型内搜索部件，现有技术也可以提供来自不同类模型的部件。然而，它们仍然不太适合提供非典型部件。

建议由一组预分割部件构成的部件看起来容易，实际非常困难。这本质是个组合搜索问题。该问题的复杂度随预分割部件的增加而急剧增大。

## 8 结论

在本文中，我们提出针对浏览式草图造型的按需部件建议技术。在用户输入草图表示其造型意图后，我们的系统基于草图得到三维代理部件，并以之为查询模型到三维模型数据库内搜索候选部件。这些候选部件是实时地从数据库模型上匹配并分割下来的。与以前的方法不同，我们的方法不需要预分割的数据库，实时地提取出与用户草图相匹配的部件，因此极大地扩展了建模空间。我们的系统为艺术家提供了完成开放式造型任务的新方法。同时，我们的方法也推动了草图式造型技术的发展。

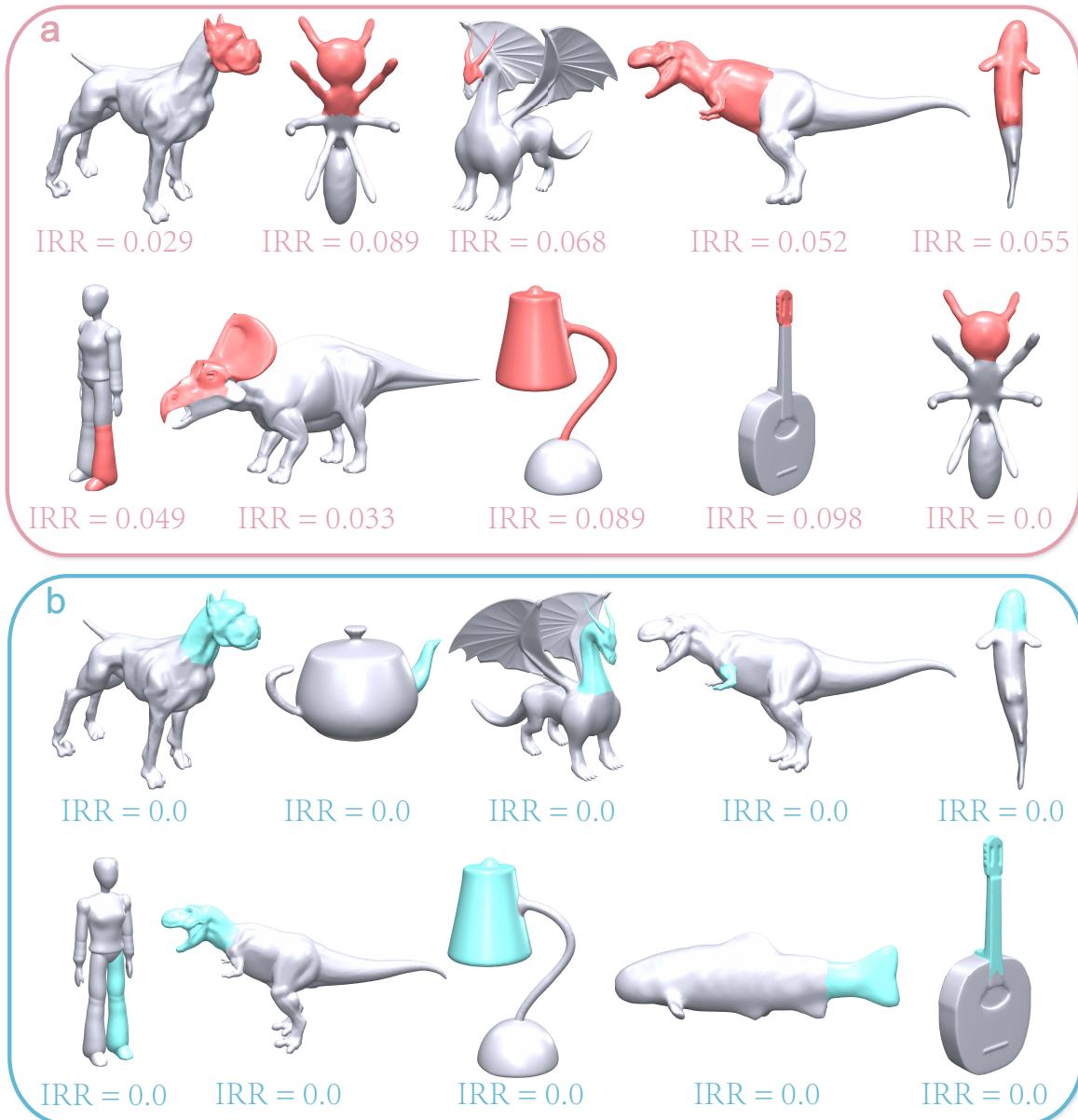


Figure 15: 具有代表性的建议部件及其非规则度量值。 (a) 图给出Ours系统建议的部件（红色）。 (b) 图给出PreSeg系统建议的部件（蓝色）。

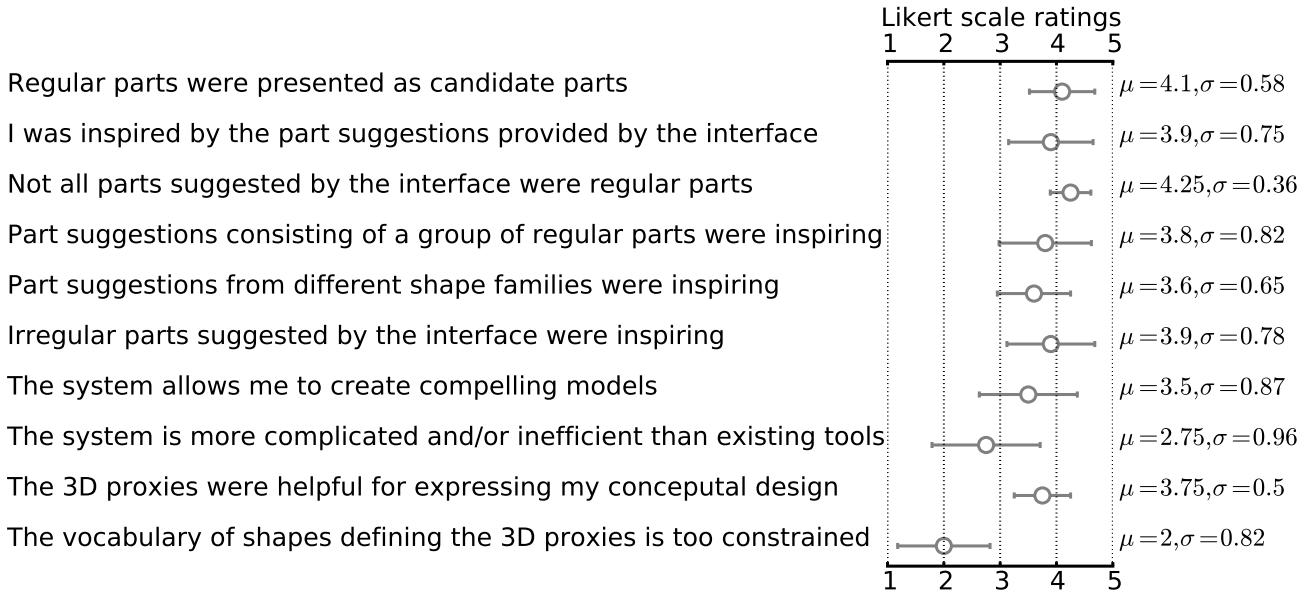


Figure 16: 调查问卷统计结果。“1”表示“强烈同意”。“2”表示“同意”。“3”表示“中立”。“4”表示“不同意”。“5”表示“强烈不同意”。

## References

- [AMCO08] Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*, 27(3):85:1–85:10, August 2008.
- [ATOM07] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. FiberMesh: designing freeform surfaces with 3D curves. *ACM Transactions on Graphics*, 26(3):Article No. 41, 2007.
- [BK04] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [CGF09] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics*, 28(3):73:1–73:12, July 2009.
- [CK10] Siddhartha Chaudhuri and Vladlen Koltun. Data-driven suggestions for creativity support in 3d modeling. *ACM Transacitons on Graphics*, 29(6):Article No. 183, 2010.
- [CKGK11] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. *ACM Transacitons on Graphics*, 30(4):Article No. 35, 2011.
- [CTSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.
- [DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3):848–855, 2003.
- [ERB<sup>+</sup>12] Mathias Eitz, Ronald Richter, Tammy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *ACM Transactions on Graphics*, 31(4), 2012.
- [FKS<sup>+</sup>04] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM Transactions on Graphics*, 23(3):652–663, 2004.
- [FMK<sup>+</sup>03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.
- [FWX<sup>+</sup>13] Lubin Fan, Ruimin Wang, Linlin Xu, Jiansong Deng, and Ligang Liu. Modeling by drawing with shadow guidance. *Computer Graphics Forum*, 32(7):157–166, 2013.
- [GF08] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics*, 27(5), 2008.

- [HFL12] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. *Computer Graphics Forum*, 31(5):1703–1713, August 2012.
- [HKG11] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. Joint shape segmentation with linear programming. *ACM Transactions on Graphics*, 30(6):125:1–125:12, December 2011.
- [KHS10] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics*, 29(4):102:1–102:12, July 2010.
- [LF08] Jeehyung Lee and Thomas Funkhouser. Sketch-based search and composition of 3d models. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM’08, pages 97–104, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.
- [LIAL07] Joseph J. LaViola, Takeo Igarashi, Christine Alvarado, and Hod Lipson. Sketch-based interfaces: techniques and applications. In *SIGGRAPH Course Notes*, 2007.
- [Sha08] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
- [SK02] Thomas B. Sebastian and Benjamin B. Kimia. Metric-based shape retrieval in large databases. In *Proceedings of the 16 th International Conference on Pattern Recognition*, ICPR ’02, pages 30291–, Washington, DC, USA, 2002. IEEE Computer Society.
- [SvKK<sup>+</sup>11] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Transactions on Graphics*, 30(6), 2011.
- [TSH99] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3D freeform design. In *Proceedings of ACM SIGGRAPH*, pages 409–416, 1999.
- [XJK<sup>+</sup>16] Xuekun Guo, Juncong Lin, Kai Xu, Siddhartha Chaudhuri, and Xiaogang Jin. CustomCut: On-demand Extraction of Customized 3D Parts with 2D Sketches. *Submitted to SGP’2016*, 2016.
- [XXM<sup>+</sup>13] Xiaohua Xie, Kai Xu, Niloy J. Mitra, Daniel Cohen-Or, Wenyong Gong, Qi Su, and Baoquan Chen. Sketch-to-design: Context-based part assembly. *Computer Graphics Forum*, 32(8):233–245, 2013.