

针对浏览式草图造型的按需部件建议技术

Xuekun Guo

2016.04.25

1 快速匹配数据结构

1.1 超面片图表示

类似 [XJK⁺16]，我们将三维模型表示成粗粒度形式—超面片图。图上顶点代表超面片。超面片通过对模型做基于边界特征的分割得到。图上边描述超面片间的相邻关系。边的权重表示相连的超面片组成同一部件的概率（如图1所示）。

超面片图包含了相邻超面片合并为更大超面片的先验概率。我们在离线阶段得到每个数据库模型的超面片图表示。

为了给数据库模型建立超面片图表示，我们将模型表面分割成200小块 [HKG11]。一小块视为一个超面片（对应超面片图内一个顶点）。每一对相邻超面片通过超面片图上一条边相连。

为了计算边的权重，我们对三维模型做900次Randomized Segmentation [GF08]。分割的目标部件数目由2变化到10。给一个分割结果，我们为每一个超面片建立一个直方图，描述该超面片属于各分割部件的概率。直方图的一个直方条针对一个分割部件，其值是该超面片位于该分割部件内的面积比例（如图2）。数学上讲，超面片 s 针对分割部件 g 的直方条 h_s^g 定义如下：

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

其中， $Face(x)$ 是属于 x 的面片集合， $Aera(f)$ 指面片 f 的面积。

这些直方图做为计算超面片位于同一部件内概率的特征向量。我们定义 $P(s, s')$ 为相邻超面片 s 与 s' 位于同一部件上的概率（称为超面片间一致性）。 $P(s, s')$ 为两超面片对应的直方图 H 与 H' 的 χ^2 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Randomized Segmentation得到的部件的集合， g_i 是第*i*个分割部件， N 是分割部件个数。

为了得到超面片图上边的权重，我们首先将两超面片在不同Randomized Segmentation结果上的一致性累加，然后通过分割次数归一化。

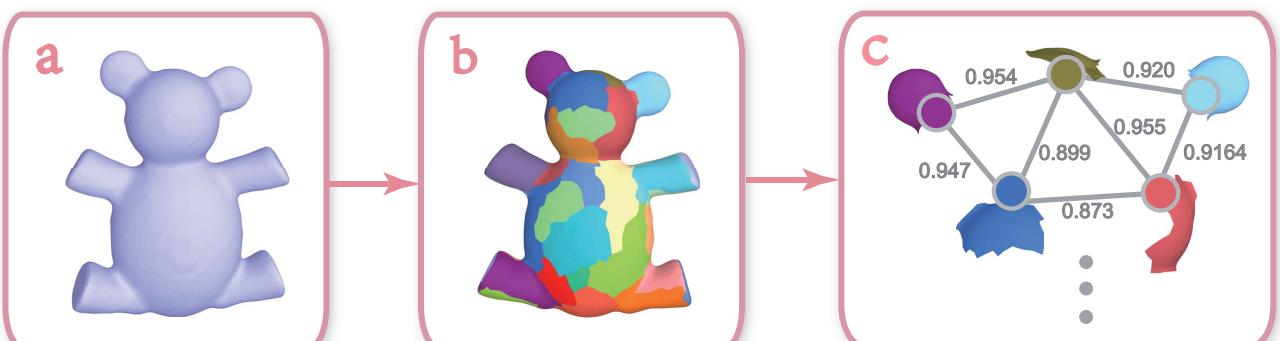
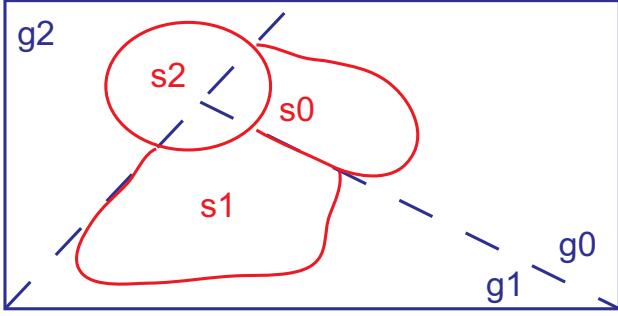


Figure 1: 三维模型超面片图建立过程。给一个Teddy模型 (a)，我们将它分割成50个超面片 (b)。一个超面片是超面片图的一个顶点。相邻超面片由带权重的边连接 (c)。



	g0	g1	g2
Histogram(s0)	= 1.0	0.0	0.0
Histogram(s1)	= 0.0	1.0	0.0
Histogram(s2)	= 0.2	0.15	0.65

Figure 2: 超面片在分割部件上的分布示意图。 s_0 , s_1 和 s_2 是超面片。 g_0 , g_1 与 g_2 是在模型上通过Randomized Segmentation得到的分割部件。

1.2 轮廓层次索引

我们将三维模型局部匹配问题转化为多视角下的二维轮廓局部匹配问题。二维轮廓局部匹配通过比较代理轮廓与数据库模型轮廓求解。我们为数据库模型提取 $T_M = 14$ 个视角下的轮廓(boundary contours) [DFRS03], 包括6个规范视角, 8个角落视角。通过这种方式, 我们得到三维模型的二维轮廓 $\Gamma_M = \{S_i | 1 \leq i \leq T_M\}$, S_i 表示第*i*个视角下的轮廓。

轮廓描述符 查询轮廓(代理模型轮廓)整体做一个轮廓段。模型轮廓在每个尺度下都由若干轮廓段构成(见??节)。模型轮廓的轮廓段与查询轮廓相匹配, 从而得到查询轮廓与模型轮廓的局部匹配。为了比较两个轮廓段, 我们为它们计算轮廓描述符。类似[XJK⁺16], 我们采用Riemenschneider等人[RDB10]提出的角度矩阵做为轮廓描述符。该描述符描述了轮廓段上采样点间的空间分布。我们为轮廓提取描述符, 在不同的采样距离下 $\Theta = \{3, 6, 9, 12, 15\}$ 提取轮廓描述符。一个轮廓段内有*m*个采样点。给一对采样点(b_i, b_j), 我们计算角度值 α_{ij} :

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

其中, $\delta = 2$ 是偏移量, $\langle l_1, l_2 \rangle$ 代表直线 l_1 与 l_2 间的夹角。轮廓描述符即为该角度值构成的矩阵 A ($A_{ij} = \alpha_{ij}$)。两个轮廓段 S 与 T 间的距离是两轮廓段的角度矩阵 A^S 与 A^T 的欧式距离:

$$D(S, T) = \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m (A_{ij}^S - A_{ij}^T)^2$$

建立轮廓层次索引 在我们的方法中一个最大的挑战是快速地在数据库模型内找到与代理部件轮廓相匹配的轮廓。直接一个接一个地比较代理部件轮廓与各模型的轮廓明显不可行, 因为匹配时间会随数据库模型数量的增大而急剧增大。因此, 我们需要设计一个具有可扩展性的方法。我们开发了一个分层的索引数据库结构, 为了数据库内模型轮廓, 来加速模型匹配过程。

对数据库模型的每一个轮廓, 我们按不同的采样距离等距采样, 得到一组采样轮廓, 每个采样轮廓称为一个点轮廓。我们首先建立一个kNN图 [SK02], 为了数据库内所有模型轮廓, 基于它们的匹配程度。图的顶点表示模型轮廓, 对每一个顶点, 我们将与之最相似的 $k = 30$ 个顶点连接在一起。匹配分数及两轮廓内采样点间的对应关系存储在对应边上。

接着, 我们在图上采样出一组种子点。我们随机地选择一个顶点做为种子, 然后采用最好优化(best-first search)方法遍历该图。当前顶点与种子点的匹配分数高于一个阈值 $\delta_c = 0.7$ 时, 当前顶点做为种子点。我们以当前顶点为起点开始遍历过程。该遍历过程不断重复, 直至所有顶点被访问到。通过这种方式, 我们从kNN图内采样出一组种子点。

为了加速建立轮廓层次索引数据结构, 我们首先过滤掉不相似的三维模型, 通过比较它们的三维模型分布(3D shape distribution) [FMK⁺03]。接着, 索引在最相似的80个模型的轮廓上建立。

轮廓层次索引辅助轮廓局部匹配 在运行中, 给一个代理模型轮廓 E_C , 我们首先将它与所有的种子轮廓相匹配。匹配结果存在优先队列内, 并按匹配分数升序排列。我们接着遍历kNN图通过最好优先方式。优先队列队顶出队, 其邻居与 E_C 相比较, 并入队。此时, 我们利用预存储在图内边上的对应信息, 可以快速地确定数据库轮廓上与 E_C 相匹配的初始轮廓段。为了降低误差积累, 我们进一步优化初始轮廓段, 通过一个简单的初始轮廓段优化方法。具体地讲, 我们首先将初始轮廓段延伸至 $0.3N_{frag}$ (N_{frag} 是初始轮廓段的长度)。接着, 我们在延伸轮廓段上找到与 E_C 相匹配的部分, 即为最终匹配轮廓段。为了进一步加速匹配过程, 我们将匹配分数大于0.9的顶点去掉。

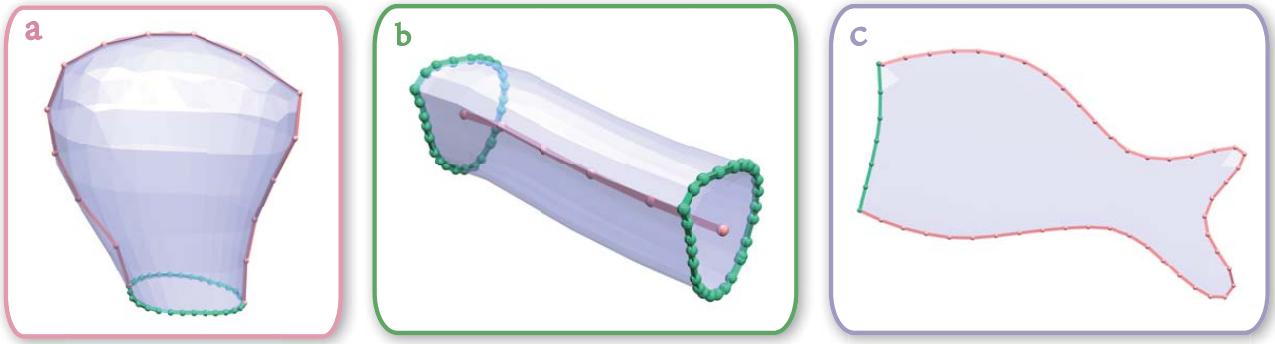


Figure 3: 三种基础代理模型。 (a) 广义球体。 (b) 广义柱体。 (c) 薄板模型。

2 候选部件搜索

本章中，我们介绍RC- k NNG如何从未做预分割的模型数据库内快速地精确地搜索出与代理模型相匹配候选部件。

2.1 代理模型创建

与 [FKS⁺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，从而使用户更容易表达造型意图。当然，用户输入的草图也可直接用于从数据库内搜索部件。

我们提供的三种基本代理模型是：广义球体，广义柱体，薄板模型（如图3）。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：在当前模型上画一个圆环表示柱体底座，然后用一条曲线指示中心线。中心线位于用户视线方向与圆球起点定义的空间平面上。广义柱体的一端可以放缩，得到圆锥或圆台。要得生成薄板型代理模型，用户需要首先在当前模型上画一条曲线，再绘制出其轮廓。该轮廓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代理模型施加各种操作，包括放缩，切割，拖拽顶点等。通过如上方法生成的各种代理模型见图3及本文视频。

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的体积变化。代理模型建成后，我们提取代理模型在体积变化最大的视角下的二维轮廓，用于描述用户的建模意图（期望的部件）。这里的体积变化指画草图前后代理模型的体积之差。

2.2 候选部件提取

给三维代理部件与数据库模型，我们的目标是在数据库模型上找到候选部件，该候选部件与三维代理部件最相似，通过基于超面片图的多视角模型匹配方法。我们首先找到候选三维部件的二维轮廓，通过将代理轮廓与数据库轮廓做匹配。三维候选部件的超面片然后被指示出来，通过graph-cut技术，在超面片图上。

轮廓匹配 轮廓匹配通过轮廓对应的轮廓描述符匹配实现。给一个通过三维代理轮廓 C ；及数据库模型轮廓 S ；我们的目标是在 S 上找到与 C 最匹配的部分 F ，使其与 C 最相似。该问题形式化如下：其中， \mathbf{A} 是 S 的轮廓描述符，大小为 $M \times N$ ， \mathbf{A}' 是 C 的轮廓描述符，大小为 $N \times N$ ($N \leq M$)。 $A_{[.]}$ 是矩阵的元素。 $V(\mathbf{A}, \mathbf{A}')$ 是匹配分数，指出 C 与 F 的差异程度。

候选轮廓段搜索 一个数据库模型的所有轮廓与三维代理部件轮廓的匹配结果组织为 $\|\Theta_M\|$ 个匹配结果组内，根据采样距离。为了找到最佳匹配结果，我们首先找到最佳匹配结果组。给定代理轮廓 C 及其采样距离 θ_B ，数据库模型轮廓 Γ_M 及其采样距离 Θ_M ，最佳匹配结果组选择问题形式化为：

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

其中 $j_{\theta_i}^j$ 表示第 j 个匹配结果的匹配分数，在采样距离 θ_i 下得到的。我们选择出得到最低的3个匹配结果，从最佳匹配结果组内，并得到候选轮廓段。得到候选轮廓段的视角是候选视角。

候选部件提取 给一个候选轮廓及其数据库模型，该数据库模型表示为超面片图表示形式。我们的目标是找到候选三维部件的超面片图，该候选部件在候选视角下与候选轮廓最相似（如图 4）。我们将三维候选部件的指认问题形式化为子图搜索问

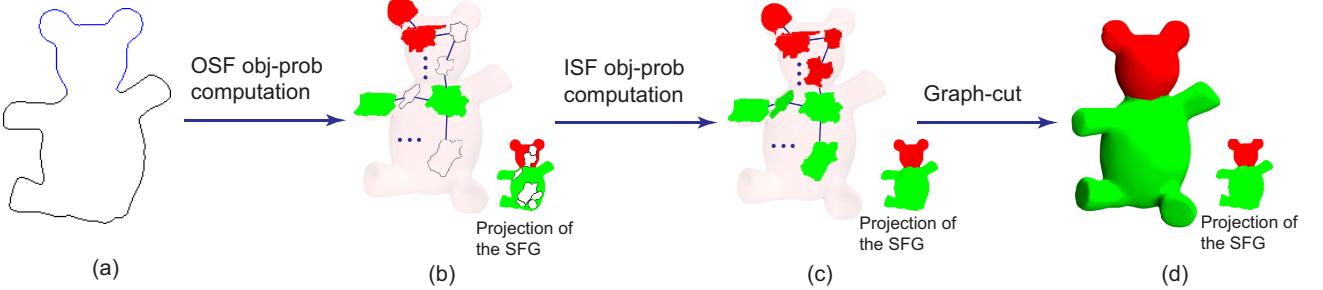


Figure 4: 三维候选部件指认技术流程。给一个候选轮廓（蓝色（a）），我们首先计算OSF obj-prob（b）。obj-prob的值映射至不同颜色。红色表示高值。绿色表示低值。然后，ISF obj-prob计算出来（c）。最后，三维候选部件的超面片通过graph-cut技术指认出来（d）。

题，目标函数为：

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

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，计算将标志 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，鼓励相邻顶点 v_i 与 v_j 的空间一致性。

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 属于三维候选部件的概率。obj-prob的计算方法在接下来介绍。

对outer SF (OSF)，其轮廓与模型轮廓有重合，其obj-prob通过其与候选三维部件的重合程度决定。对一个超面片图来讲，除了Outer SF就是Inner SF (ISF)。我们认为，在整张图内，OSF与ISF的obj-prob相一致。因此，我们让OSF的obj-prob流入ISF，根据SF间的一致性。obj-prob流问题形式化为最小二乘问题：

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

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。

2.3 候选部件排序

我们基于候选部件与三维代理部件及当前模型间的兼容性对候选部件排序。

在计算兼容性的时候，我们考虑如下因素：1) 候选部件与代理部件间的二维投影的相似性；2) 候选部件与三维代理部件间三维相似性；3) 候选部件与三维代理部件间的背景一致性。给一个候选部件 p （ p 提取自数据库模型 M ），三维代理部件 B ，两者的兼容性定义如下：

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

其中， Σ 指 p 的背景区域， X 指 B 的背景区域， $\omega = 0.8$ ， $\xi = 0.15$ ， $\psi = 0.05$ 。 $T_p(\cdot, \cdot)$ 计算三维模型的二维投影的相似性。 $T_g(\cdot, \cdot)$ 计算三维模型的三维相似性。 $T_c(\cdot, \cdot)$ 计算三维模型间的一致性。

二维投影相似性 $T_p(p, B)$ 定义如下：

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

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。 $V(\cdot, \cdot)$ 的定义见方程 (2)。 $\omega_m = 0.9$ ， $\omega_a = 0.1$ 。

三维模型相似性 $T_g(p, B)$ 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]。 $\alpha = 0.9$, $\beta = 0.1$ 。

背景一致性 $T_c(\Sigma, X)$ 定义如下:

$$T_c(\Sigma, X) = T_g(X, \Sigma).$$

兼容性分数值低于 θ_j 的候选部件被丢弃。留下的候选部件排序后建议给用户。如果需要多样性的部件建议, 候选部件用Maximal Marginal Relevance criterion [CK10]技术重新排序。

3 候选部件优化

用户点击鼠标即可执行部件边界优化。该步骤是可选的, 它并不会对最终造型结果有太大影响。初始候选部件提示给用户, 做为部件建议。如果用户选中该部件, 系统会自动地对该部件细化。候选部件细化是可选项。

我们首先得到ROI区域 Ω_{obj} 与 Ω'_{obj} 在数据库模型上, 通过扩展初始切割线, 在初始候选部件的内部与外部, 扩展长度为 $0.1l_{dia}$ 。 l_{dia} 是初始候选部件的包围盒的对角线长度。代理部件的ROI区域 Ω_{proxy} 通过扩展其边界, 类似数据库模型的方法得到。接着我们大致地将 Ω_{obj} 与 Ω_{proxy} 对齐。面片层级的部件提取问题形式化为能量最优化问题, 目标函数定义如下:

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

其中, $G^f = (\nu^f, \varepsilon^f)$ 是ROI区域 $\Omega_{obj} \cup \Omega'_{obj}$ 的对偶图。 ε^f 表示面片, ν^f 表示面片间相邻关系。 $X = \{x_{v_i} | v_i \in \nu\}$ ($x_{v_i} = \{0, 1\}$) 是 G^f 的一个标记。 $E_d(x_{v_i})$ 是数值项, 计算将标记 x_{v_i} 给 v_i 的能量耗费, 其定义如下:

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

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 与代理部件ROI Ω_{proxy} 间距, $g_c(v_i)$ 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OBB的体积。

$E_s(x_{v_i}, x_{v_j})$ 是光滑项, 鼓励 v_i 与 v_j 间的空间一致性。其定义如下:

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

其中 $\theta \in [0, \frac{\pi}{2}]$ 是相邻面片 v_i 与 v_j 的共边与初始切割线的拟合平面间的夹角。

4 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??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5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。 kNN 图辅助的轮廓匹配得到的性能提升如表1。

Algorithms	Database Size	114	205	513
Brute-force	59.58	106.99	240.07	
<i>k</i> NNG	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*NNG表示我们的轮廓索引数据结构。

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::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

4.1 *k*NN图

4.2 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
- **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，躯干，胳膊，腿。2) 为每一个部件提取各视角下的边界轮廓线。3) 将所有部件的轮廓线组织在*k*NN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6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7。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。分数成为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8内，我们给出一组代表性的建议部件及其得分。

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。

Type \ STAT.	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 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 \ 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

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：

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

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9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10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造型过程的概念设计阶段具有很大助益。

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。

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

与现有的要求预分割及标记的部件建议技术 [XXM⁺13, FWX⁺13] 相对，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，这类技术也可以提供来自不同类模型的部件。然而，他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长而急剧增长。从某种程度上讲，我们的RC-*k*NNG技术就是针对组合搜索问题的。如果现有的部件建议技术 [CK10, XXM⁺13, FWX⁺13] 采用我们的RC-*k*NNG技术，他们也可以达到类似我们技术的效果。

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.
- [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.
- [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.
- [FKS⁺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⁺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⁺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.
- [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.
- [RDB10] Hayko Riemenschneider, Michael Donoser, and Horst Bischof. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part V*, chapter Using Partial Edge Contour Matches for Efficient Object Category Localization, pages 29–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [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.
- [XJK⁺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⁺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.



Figure 5: 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。

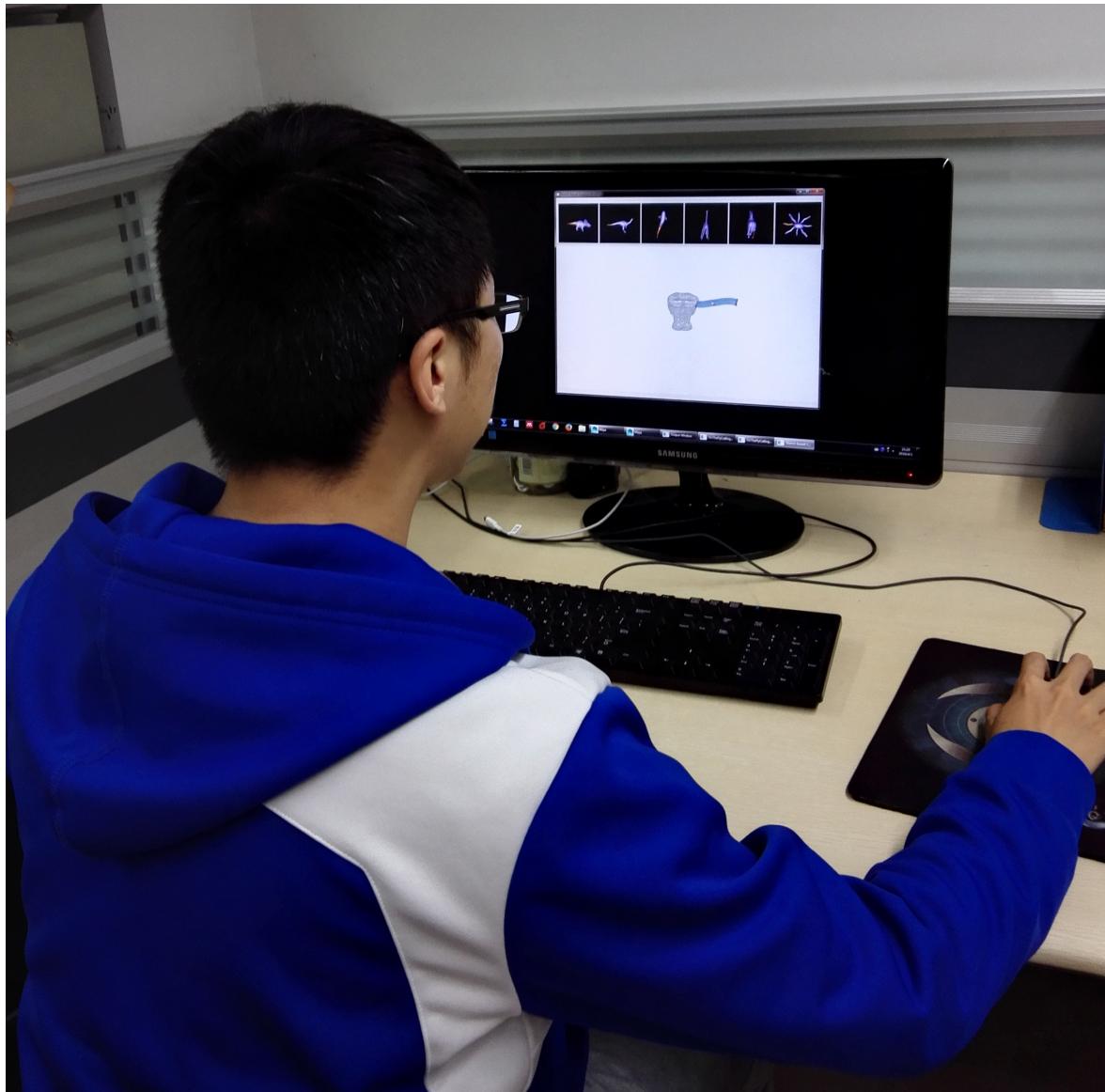


Figure 6: 一位测试者正在操作造型系统完成造型任务。

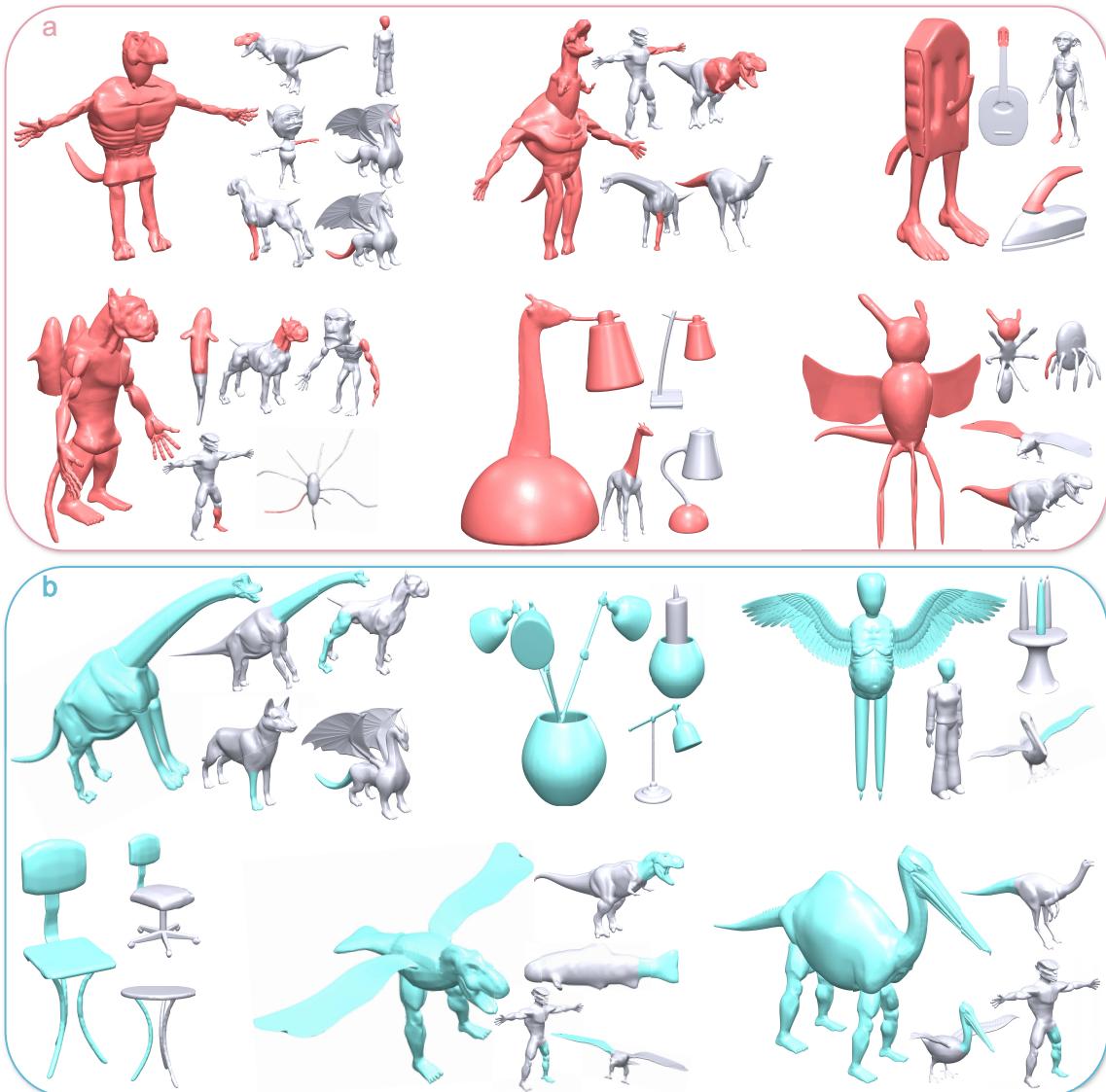


Figure 7: Ours系统 (a) 与PreSeg系统 (b) 造型结果。每个造型结果，其构成部件在右侧。

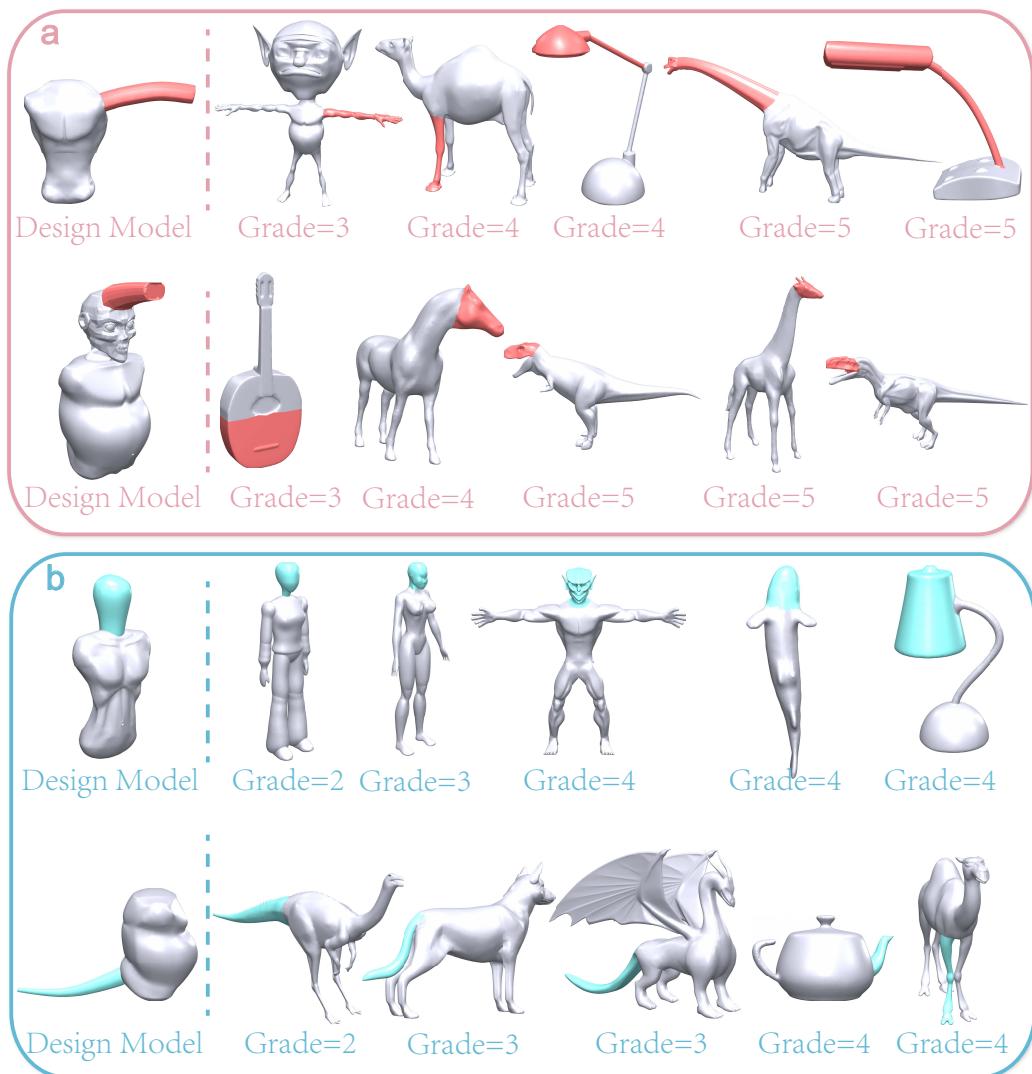


Figure 8: 具有代表性的建议部件及其得分。 (a) 图给出Ours系统建议的部件（红色）。 (b) 图给出PreSeg系统建议的部件（蓝色）。

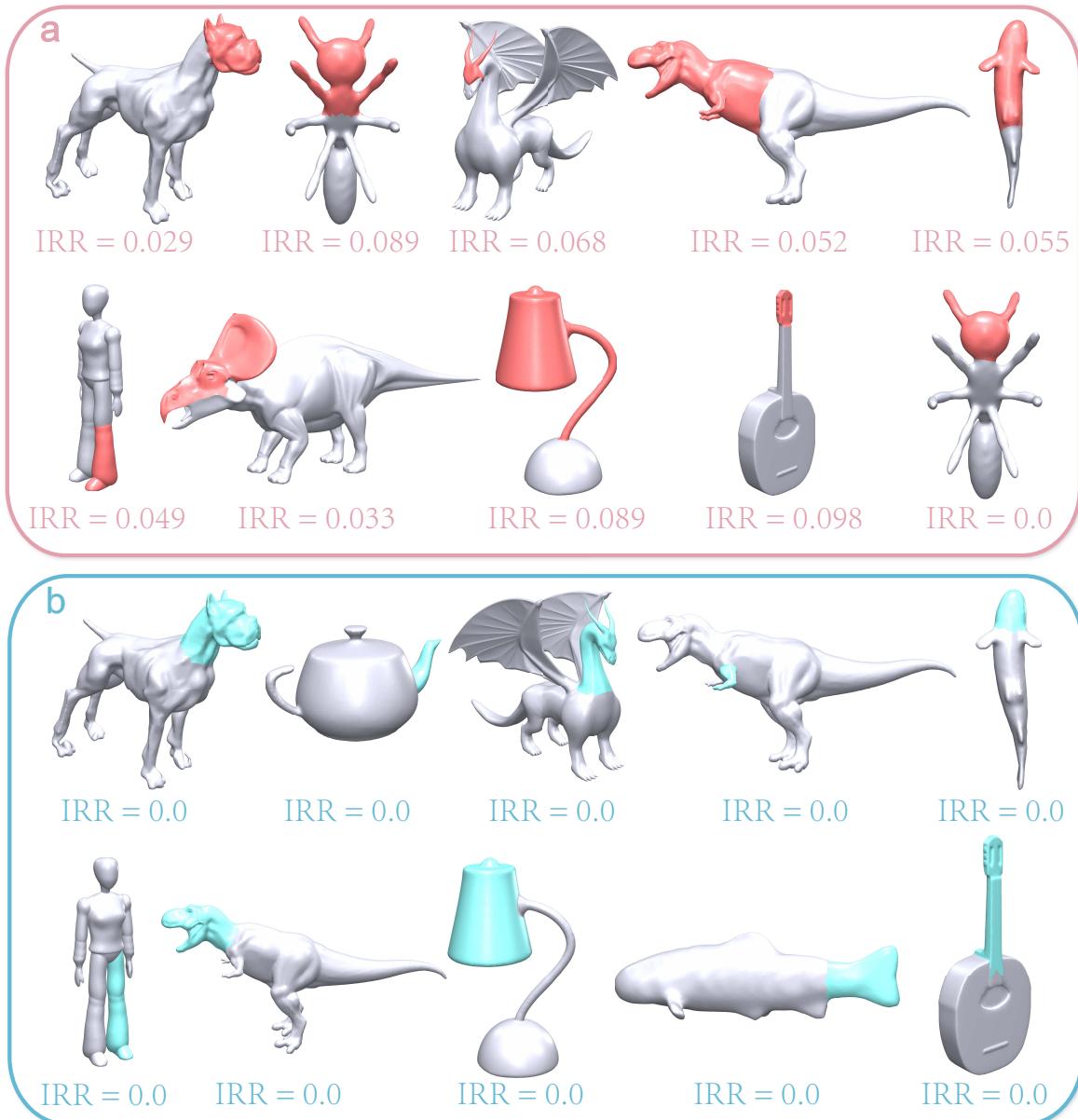


Figure 9: 具有代表性的建议部件及其非规则度量值。 (a) 图给出Ours系统建议的部件（红色）。 (b) 图给出PreSeg系统建议的部件（蓝色）。

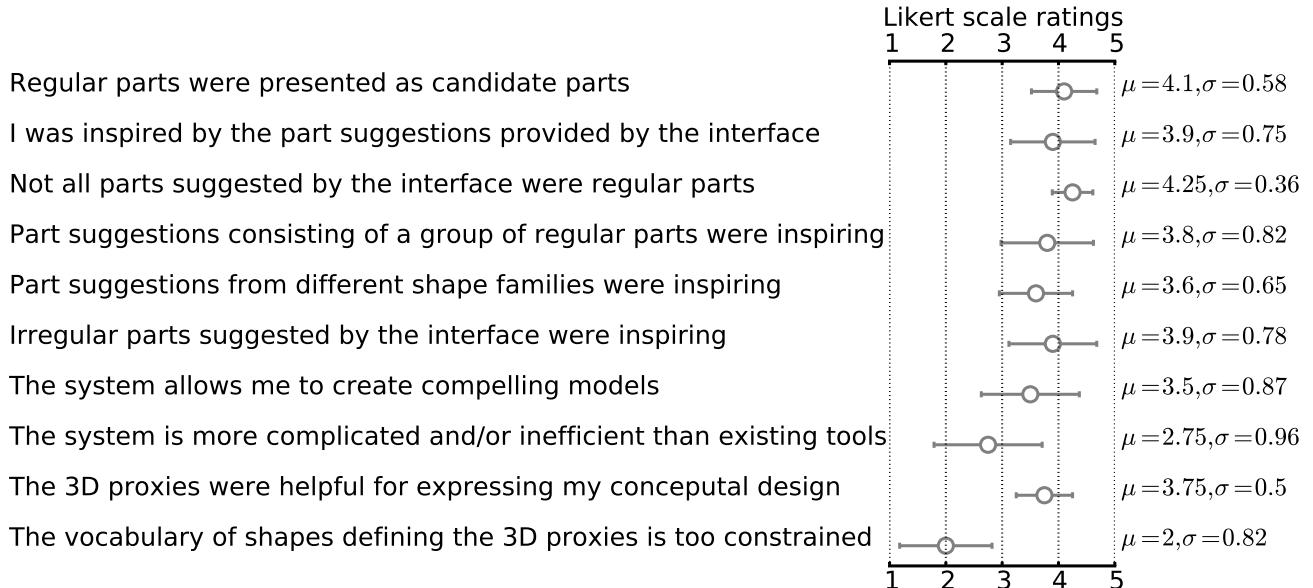


Figure 10: 调查问卷统计结果。“1”表示“强烈同意”。“2”表示“同意”。“3”表示“中立”。“4”表示“不同意”。“5”表示“强烈不同意”。