文章编号：1000-8608(2017)03-0241-06

# 组件匹配与融合的三维形状建模研究

白茂东， 华顺刚*， 苏铁明

（大连理工大学 机械工程学院，辽宁 大连 116024）

**摘要：** 提出一种基于组件匹配、融合的三维形状生成算法. 根据 Hausdorff 距离对输入形状的各组件进行匹配，形成匹配组件对，进行球面参数化映射，合并组件对的球面网格模型；通过反映射建立组件间的顶点对应关系，采用不同的融合系数进行插值融合，生成一系列连续变化的组件；最后根据输入形状的连接关系进行重新连接，生成完整的变形形状. 实验表明，采用所提算法可以生成合理、相似的变形形状，同时保留了输入形状的功能和表面细节.

**关键词：** 组件匹配；球面参数化；网格融合；重连接

**中图分类号：** TP391 **文献标识码：** A **doi：** 10.7511/dllgxb201703004

## 0 引 言

基于实例的三维形状建模技术通过对已有形状的编辑修改生成新的、功能相似的三维形状，该技术可广泛应用于 CAD、工业造型设计、创新设计以及计算机动画和虚拟现实等领域[1-2]，近年来已成为计算机图形学界的研究热点.

目前已有多种三维形状变形生成算法涌现，其中具有代表性的方法是对现有形状的组件进行替换、重组得到具有创意的三维形状[3]，这类方法包括：基于形状的约束推导分析生成新形状[4]、根据样式转移方法生成融合形状[5]、利用概率形状结构指导组件的替换或重组[6]，以及基于集合演变方法进行三维形状的生成[7]等. 该类方法依赖输入形状结构，当替换含有不兼容的组件结构时，会导致不合理的结果. 另一类方法通过拓扑变换、融合及重建来生成具有新的拓扑结构的三维形状[8]. 这类方法可以生成拓扑结构不同的形状，但因需泊松重建，所以难以保证形状的细节；且在变形过程中因拓扑发生了改变，会生成较多不合理的新形状，需进行人工干预排除. 此外，Kanai 等[9]提出了一种针对等拓扑 genus-0 模型的变形框架，这种变形方法需要模型特征的严格对齐.

本文提出一种基于实例的三维网格形状变形生成算法. 三维网格形状由点、线和面片组成，通过组件匹配、融合、重连接生成新的连续变化的形状，这里的组件是在预处理过程中形成且具有模型部分功能意义的封闭网格. 算法输入的是经规则化及组件分割等预处理后的形状，算法步骤综述如下：首先，基于 Hausdorff 距离[10]进行输入形状中各组件的匹配，形成匹配组件对；其次，对各组件进行球面参数化映射，合并两组件的球面网格模型，该模型中包括两个输入组件的所有顶点，通过反映射使组件对间建立顶点对应关系，进行插值融合；最后，根据输入形状的接触及连接关系信息，进行重连接生成完整的变形形状. 其中插值融合形成新的网格模型包含组件的表面细节特征，采用不同的融合系数，可产生一系列连续变化的形状.

## 1 组件匹配

根据设计者意图选取两个三维形状作为输入，其中一个作为原始形状，另一个作为参考形状，为便于组件的匹配和融合，需要对输入的形状进行预处理操作，包括形状规则化、方向归一化以及组件分割等. 通过调整参考形状顶点可以对形状的外形尺寸进行规则化处理，根据主成分分析法[11]可以对模型方向进行归一化. 此外，许多现有算法[3,12]可以对形状进行组件分割，使所输入

**收稿日期：** 2016-08-15； **修回日期：** 2017-03-28.
**基金项目：** 国家自然科学基金资助项目(61300085).
**作者简介：** 白茂东(1989-)，男，博士生，E-mail：baimaodong@mail.dlut.edu.cn；华顺刚*(1964-)，男，教授，E-mail：hsgang02@dlut.edu.cn；苏铁明(1972-)，男，讲师，E-mail：tiemings@dlut.edu.cn.

形状由若干组件组成. 由于本文所述重点是组件的匹配和融合,在此,对预处理操作不做详细叙述.

组件匹配的目的是建立两个给定形状所有组件之间最佳的对应关系. 对于来源于不同形状的两组件,可以计算它们点云数据之间的 Hausdorff 距离来测量组件距离.

Hausdorff 距离表示的是一个度量空间内两个非空子集之间的相互距离. 它可以反映两个集合之间的差距,因此,用它表示两个组件之间的不相似度.

对于一个度量空间内的两个非空集合 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 和 $\boldsymbol{Y}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n\}$,$\boldsymbol{X}$ 到 $\boldsymbol{Y}$ 的 Hausdorff 距离定义为

$$h(\boldsymbol{X},\boldsymbol{Y})=\max_{\boldsymbol{x}\in\boldsymbol{X}}\{\min_{\boldsymbol{y}\in\boldsymbol{Y}}\{\|\boldsymbol{x}-\boldsymbol{y}\|\}\} \tag{1}$$

其中 $\|\boldsymbol{x}-\boldsymbol{y}\|$ 是集合内任意元素之间的距离度量. 这里使用 Hausdorff 距离更普通的定义形式:

$$H(\boldsymbol{X},\boldsymbol{Y})=\max\{h(\boldsymbol{X},\boldsymbol{Y}),h(\boldsymbol{Y},\boldsymbol{X})\} \tag{2}$$

该式反映了两个集合之间的相似程度[13-14].

组件距离为通过式(2)计算的组件网格点云数据之间的 Hausdorff 距离.

本文使用一种贪婪算法对两个形状中的组件进行匹配. 其具体过程如下:首先,对于原始形状中的每一个组件,它到参考形状中每个组件的距离,可以通过式(2)计算,并保存在距离集 $H$ 中;其次,找出其最小的 Hausdorff 距离 $H_{\min}$,如果 $H_{\min}$ 小于相似度阈值 $\theta$,那么与 $H_{\min}$ 距离对应的两个组件被认为是匹配的并作为一个匹配组件对;然后,删除集合 $H$ 中与此匹配组件对相关的其他元素,重复这一过程直到出现 $H_{\min}$ 大于相似度阈值 $\theta$ 或者一个形状中的所有组件都完成配对,如果仍存在不可匹配的组件,则被视为没有对应关系. 图 1 列举了两个形状匹配的一些具体实例,其中匹配的组件对用相同的颜色表示,无匹配关系的组件用浅棕色表示.

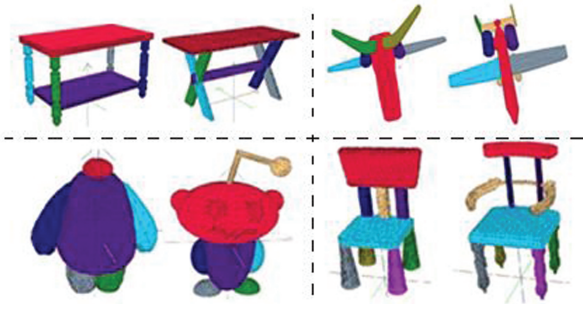


图 1 两个形状的组件匹配

Fig. 1 The components matching for two shapes

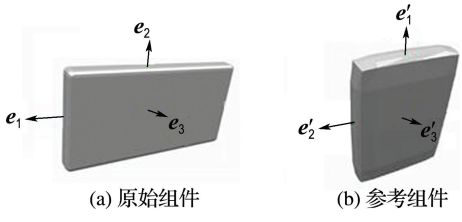
## 2 基于球面参数化的组件融合

为充分保持组件表面特征,对于匹配组件对,采用基于组件融合的球面参数化[15]方法进行融合. 该方法分为以下 3 步:第 1 步对匹配组件对的方向和大小进行标准化;第 2 步将匹配的组件分别参数化到一个单位球面上生成拓扑球 $M_o$ 和 $M_r$,并融合生成新的拓扑球 $M_c$;第 3 步利用 $M_c$ 顶点对匹配组件对进行反映射,获得顶点对应关系,并进行顶点插值生成新的融合组件.

### 2.1 组件标准化

组件标准化的实质是对匹配组件对的顶点集 $\boldsymbol{V}_o$ 和 $\boldsymbol{V}_r$ 进行标准化. 首先使用主成分分析法(principal component analysis,PCA)[11,16]对组件点云信息进行分析,然后计算出组件 3 个主成分,确定组件的主轴.

通常把组件的第一主成分作为主轴,但对于不同的组件,其参考组件的第一主成分可能会存在很大差异,因此,将原始组件的第一主成分作为主轴,对于参考组件,将最接近原始组件主轴的主成分作为匹配组件的主轴. 如图 2 所示,原始组件的第一主成分 $\boldsymbol{e}_1$ 作为主轴 $\boldsymbol{e}_{\text{orig}}$. 显然,对于参考组件,其第一主成分 $\boldsymbol{e}_1'$ 与 $\boldsymbol{e}_1$(即 $\boldsymbol{e}_{\text{orig}}$)存在很大差异,因此,将最接近 $\boldsymbol{e}_{\text{orig}}$ 的主成分 $\boldsymbol{e}_2'$ 作为参考组件的主轴 $\boldsymbol{e}_{\text{ref}}$.




(a) 原始组件 (b) 参考组件

图 2 原始和参考组件的主轴选择

Fig. 2 Principal axes selection for original and reference components

接着将匹配组件对的主轴 $\boldsymbol{e}_{\text{orig}}$ 和 $\boldsymbol{e}_{\text{ref}}$ 进行对齐. 在由 $\boldsymbol{e}_{\text{orig}}$ 和 $\boldsymbol{e}_{\text{ref}}$ 确定的平面中,通过旋转向量 $\boldsymbol{e}_{\text{orig}}$,使之与向量 $\boldsymbol{e}_{\text{ref}}$ 一致,在三维坐标系中,可以通过旋转矩阵 $\boldsymbol{M}$ 实现向量的旋转:

$$\begin{pmatrix} \boldsymbol{e}_{\text{orig}} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}}{|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}|} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{orig}}}{|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{orig}}|} \end{pmatrix}\boldsymbol{M}=\begin{pmatrix} \boldsymbol{e}_{\text{ref}} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}}{|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}|} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{ref}}}{|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{ref}}|} \end{pmatrix} \tag{3}$$

由此可以得到旋转矩阵

$$\boldsymbol{M}=\begin{pmatrix} \boldsymbol{e}_{\text{orig}} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}}{\left|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\right|} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{orig}}}{\left|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{orig}}\right|} \end{pmatrix}^{-1}\begin{pmatrix} \boldsymbol{e}_{\text{ref}} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}}{\left|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\right|} \\ \dfrac{\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{ref}}}{\left|\boldsymbol{e}_{\text{orig}}\times\boldsymbol{e}_{\text{ref}}\times\boldsymbol{e}_{\text{ref}}\right|} \end{pmatrix} \tag{4}$$

同时得到主轴规范后原始组件和参考组件的顶点集 $\boldsymbol{V}'_{\text{o}}(\boldsymbol{V}'_{\text{o}}=\boldsymbol{V}_{\text{o}}\boldsymbol{M})$ 和 $\boldsymbol{V}'_{\text{r}}(\boldsymbol{V}'_{\text{r}}=\boldsymbol{V}_{\text{r}})$. 为了将参考组件转变成与原始组件具有相同包围盒大小的结构，利用一个 $3\times3$ 对角矩阵 $\boldsymbol{S}$，$\boldsymbol{S}=\mathbf{diag}\{s_x,s_y,s_z\}$，对组件进行缩放，其中 $s_x$、$s_y$ 和 $s_z$ 分别为原始组件和参考组件的包围盒 3 个主轴方向边的长度比.

## 2.2　球面参数化

组件标准化后，根据球面中心投影法将组件映射到一个单位球上，组件的质心即为球心. 为了消除映射产生的叠影，本文使用拉普拉斯迭代方法[17]进行过滤. 拉普拉斯算子通常被定义为一种低通滤波器，它可以过滤高频，而叠影常发生在球体表面顶点的高频区域，所以使用拉普拉斯迭代来消除叠影. 其数学表达式为

$$L(x_i)=\Delta x_i=\sum_{j\in i^*}w_{i,j}(x_j-x_i);\ i\in\{1,2,\cdots,n\} \tag{5}$$

其中集合 $i^*$ 是顶点 $x_i$ 邻域顶点的集合，对于每个 $i$，权数 $w_{i,j}$ 是正数且总和为 1，即 $\sum_{j\in i^*}w_{i,j}=1$. 本文把三维顶点坐标 $\boldsymbol{v}_i$ 作为一个向量信号，那么使用拉普拉斯算子进行一次迭代可以表述为

$$\boldsymbol{V}'=\boldsymbol{V}+\Delta\boldsymbol{V}=\{\boldsymbol{v}'_i\,|\,\boldsymbol{v}'_i=\boldsymbol{v}_i+\Delta\boldsymbol{v}_i,\Delta\boldsymbol{v}_i=L(\boldsymbol{v}_i)\} \tag{6}$$

迭代的过程即为消除叠影的过程，迭代完成后，即可得到原始组件和参考组件的球面参数化网格模型 $M_{\text{o}}$ 和 $M_{\text{r}}$.

接下来对网格模型 $M_{\text{o}}$ 和 $M_{\text{r}}$ 进行拓扑融合生成拓扑球 $M_{\text{c}}$，拓扑球 $M_{\text{c}}$ 包含 $M_{\text{o}}$ 和 $M_{\text{r}}$ 中所有顶点及网格相交生成的交点. 如图 3 所示，实心点属于参数化到 $M_{\text{o}}$ 上的顶点，空心点属于参数化到 $M_{\text{r}}$ 上的顶点. 在 $M_{\text{o}}$ 和 $M_{\text{r}}$ 的网格模型上，其任意一条边与球心 $O$ 的连线可以形成新的面，在 $M_{\text{c}}$ 中当两者相交时会产生一条通过球心的直线（如图 3 中的红线所示），该直线与单位球表面的交点，即为新生成的点. 然后遍历每一条边，并按照顺时针方向进行边循环创建面，如图 4 所示，新形成的面可能含有 3 条边、4 条边等，最后基于锐角三角形原则进行三角化形成最终的网格模型 $M_{\text{c}}$.

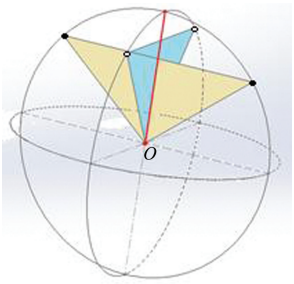



图 3　$M_{\text{o}}$ 和 $M_{\text{r}}$ 之间的交点

Fig. 3　The intersection between $M_{\text{o}}$ and $M_{\text{r}}$

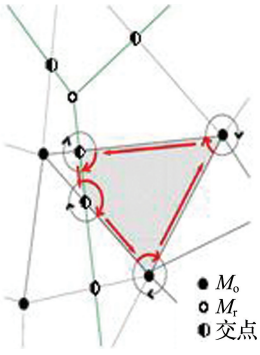



图 4　在 $M_{\text{c}}$ 中沿着边循环创建面

Fig. 4　Facet created along edge cycles in $M_{\text{c}}$

## 2.3　反映射与组件生成

融合生成的网格模型 $M_{\text{c}}$ 包含了原始组件和参考组件的所有顶点信息，反映射的过程实际就是利用 $M_{\text{c}}$ 的顶点坐标分别表示原始组件及参考组件顶点的过程. 根据上述 $M_{\text{c}}$ 的顶点分类，反映射过程中，它的每一个顶点都必须被考虑和确定，这里用计算反映射原始组件 $\boldsymbol{V}_{\text{c-o}}$ 的变换进行说明. 对于 $M_{\text{o}}$ 的每一个顶点，它必须被包含在 $M_{\text{c}}$ 上一个确定的三角形面片中，考虑到三角形顶点的相对位置，利用 $M_{\text{c}}$ 包含的三角形去计算 $M_{\text{o}}$ 的顶点，并用 $M_{\text{c}}$ 上的顶点坐标进行表示. 这样 $M_{\text{o}}$ 的顶点就得到了重新计算，也就完成了 $\boldsymbol{V}_{\text{c-o}}$ 的变换. 同理可得到反映射参考组件的 $\boldsymbol{V}_{\text{c-r}}$ 的变换，具体的组件融合与反映射过程如图 5 所示.

如上所述，利用 $M_{\text{c}}$ 的网格结构表示原始组件以及参考组件的变换，即 $\boldsymbol{V}_{\text{c-o}}$ 和 $\boldsymbol{V}_{\text{c-r}}$，可以获取原始组件及参考组件上所有顶点的对应关系. 如图 6 所示，通过在 $\boldsymbol{V}_{\text{c-o}}$ 和 $\boldsymbol{V}_{\text{c-r}}$ 进行顶点插值，生成组件由原始组件向参考组件进行转变. 融合顶点可以计算如下：

$$\boldsymbol{V}_{\text{c}}(k)=k\boldsymbol{V}_{\text{c-o}}+(1-k)\boldsymbol{V}_{\text{c-r}} \tag{7}$$

其中 $k(0<k<1)$ 为融合系数，反映了几何学上与参考组件的接近程度.

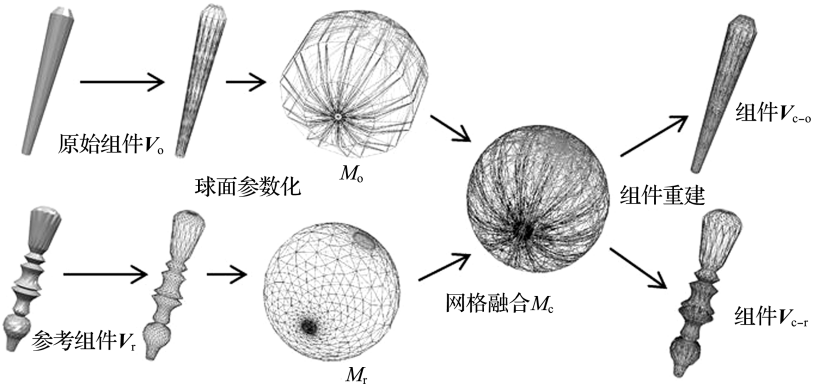



图 5 组件球面参数化、网格融合及重建过程

Fig. 5 The process of component spherical parameterization, mesh merging and reconstruction

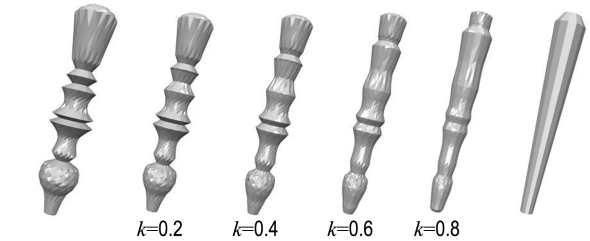



图 6 组件融合

Fig. 6 Component merging

## 3 组件重连接

虽然组件融合保留了原始顶点的拓扑关系，但融合生成的组件在几何形状、维度以及位置上会产生变化. 如果直接将生成的组件进行连接，则可能会发生如图 7(a)所示的组件分离等现象. 为确保连接和拓扑变化的合理性，本文选择几何尺寸最大或者位于原始形状正中心的组件作为基础组件，然后获取相对于该组件其他所有组件的位置，从而进行组件重连接. 如果原始形状的两个组件具有连接关系 $c_1=(\boldsymbol{S}_1,\boldsymbol{S}_2)$，$\boldsymbol{S}_1$ 和 $\boldsymbol{S}_2$ 是通过两个组件最近的顶点集，那么，可以类似地获得它们在参考形状中分别匹配的组件对应的连接关系 $c_2=(\boldsymbol{S}_1',\boldsymbol{S}_2')$. 对于融合系数 $k$，融合后的组件对应连接关系 $c_b(k)$可以通过公式 $c_b(k)=kc_1+(1-k)c_2$ 计算得到. 如图 7(b)所示，按照其计算后的连接关系调整融合后的组件，将形成完整合理的形状.



(a) 融合后的组件分离现象



(b) 重连接后的合理形状

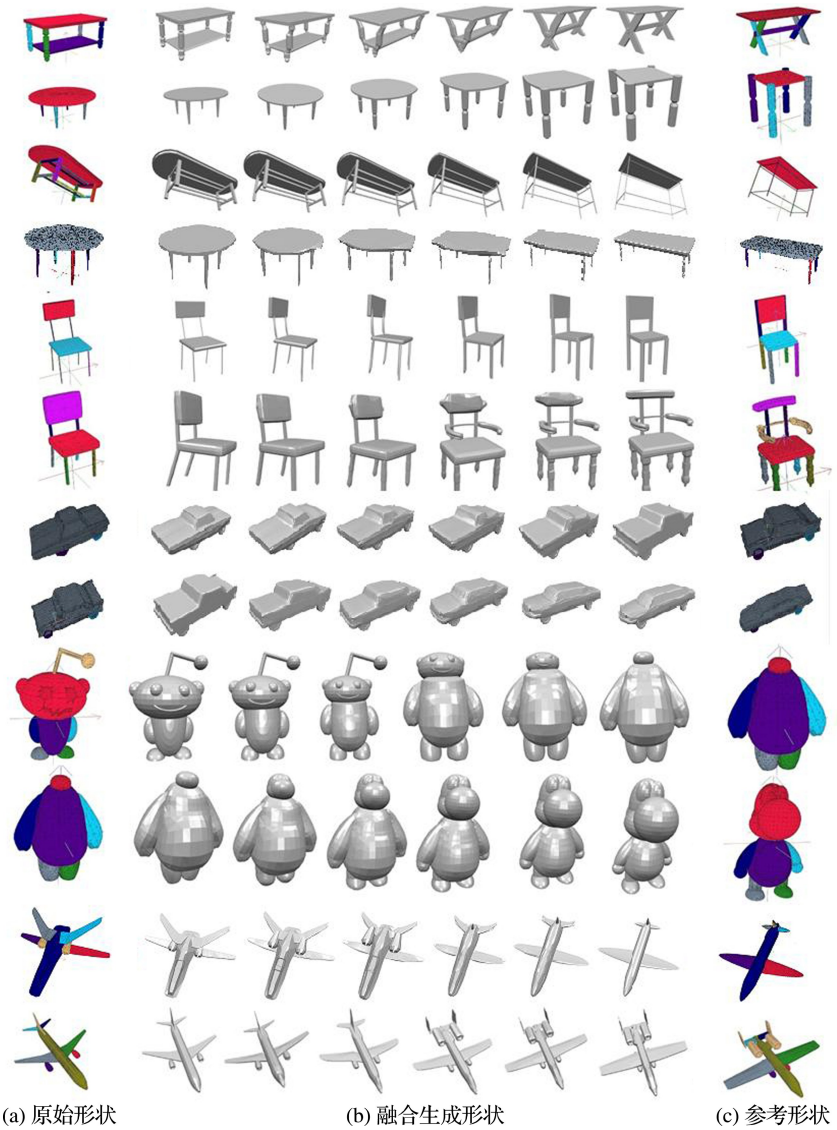
图 7 组件重连接

Fig. 7 Component relinking

## 4 实验验证

在 Intel i5 处理器、3.30 GHz 主频、8 GB 内存台式机上对本文算法进行实验验证，使用 C++语言在 VS2010 开发平台对 off 网格模型进行处理. 为验证本文算法的有效性和实用性，从互联网和一些文献中[8,18]收集若干三维网格模型，如桌子、椅子、飞机等，实验结果如图 8 所示.

每一行从左到右融合系数 $k$ 依次取值为 0、0.2、0.4、0.6、0.8、1.0. 对于输入形状，如果所有组件都具有一对一匹配关系，那么无论选择谁作为原始组件，其变形结果都相同；对于没有匹配关系的组件，只作简单保留. 相较于组件替换及拓扑变化的形状生成方法[8]，本文算法能够生成合理的表面变形，同时保留输入模型功能合理性和表面细节. 在图 8 中，每行模型都能基于输入产生完整的过渡变化效果且保持了输入的功能合理性，第 1、2 和 4 行原始桌腿的表面纹理在变化中被很好地继承下来，第 9 和 10 行娃娃头部特征也被很好地继承下来.



(a) 原始形状　　(b) 融合生成形状　　(c) 参考形状

图 8　形状生成实例

Fig. 8　The examples of shape creation

根据输入形状的多样性，虽然所提出的算法可以生成大量的变形形状，但每一个三角形面片和顶点都将基于输入的网格模型被计算处理成新的网格，因此，需要输入具有良好网格结构的形状模型.

## 5　结　语

基于组件匹配和融合，实现了三维形状的生成. 算法首先基于 Hausdorff 距离对输入形状的组件进行匹配，获得匹配组件对；为避免表面凹凸遮挡现象，解决顶点对应的准确性，便于获得形状表面细节，将匹配组件对球面参数化，融合生成新的包含两组件全部顶点的网格模型；然后用融合的网格对组件进行反映射计算，新生成的网格模型包含输入组件的所有顶点对应关系. 顶点插值时根据融合系数的不同取值，获得不同的组件形状；最后基于输入形状的连接关系将生成的组件进行重连接，产生连续变化的形状. 本文提出的匹配融合算法生成的形状具有合理的几何外观和功能，并保留了输入形状的细节.

## 参考文献：

[1] WATSON I, PERERA S. Case-based design: A review and analysis of building design applications [J]. **Artificial Intelligence for Engineering Design Analysis and Manufacturing**, 1997, **11**(1):59-87.

[2] LIU Qiaosheng, XI Juntong. Case-based parametric design system for test turntable [J]. **Expert Systems with Applications**, 2011, **38**(6):6508-6516.

[3] KALOGERAKIS E, CHAUDHURI S, KOLLER D A. A probabilistic model for component-based shape synthesis [J]. **ACM Transactions on Graphics**, 2012, **31**(4):55.

[4] JAIN A, THORMÄHLEN T, RITSCHEL T, *et al*. Exploring shape variations by 3D-model decomposition and part-based recombination [J]. **Computer Graphics Forum**, 2012, **31**(2):631-640.

[5] HAN Zhizhong, LIU Zhenbao, HAN Junwei, *et al*. 3D shape creation by style transfer [J]. **The Visual Computer**, 2015, **31**(9):1147-1161.

[6] CHAUDHURI S, KALOGERAKIS E, GUIBAS L, *et al*. Probabilistic reasoning for assembly-based 3D modeling [J]. **ACM Transactions on Graphics**, 2011, **30**(4):35.

[7] XU Kai, ZHANG Hao, COHEN-OR D, *et al*. Fit and diverse: Set evolution for inspiring 3D shape galleries [J]. **ACM Transactions on Graphics**, 2012, **31**(4):57.

[8] ALHASHIM I, LI Honghua, XU Kai, *et al*. Topology-varying 3D shape creation via structural blending [J]. **ACM Transactions on Graphics**, 2014, **33**(4):158.

[9] KANAI T, SUZUKI H, KIMURA F. Metamorphosis of arbitrary triangular meshes [J]. **IEEE Computer Graphics and Applications**, 2000, **20**(2):62-75.

[10] Wikipedia. Hausdorff distance [Z/OL]. [2016-07-28]. https://en.wikipedia.org/wiki/Hausdorff_distance.

[11] ABDI H, WILLIAMS L J. Principal component analysis [J]. **Wiley Interdisciplinary Reviews: Computational Statistics**, 2010, **2**(4):433-459.

[12] SHAMIR A. A survey on mesh segmentation techniques [J]. **Computer Graphics Forum**, 2008, **27**(6):1539-1556.

[13] ROTE G. Computing the minimum Hausdorff distance between 2 point sets on a line under translation [J]. **Information Processing Letters**, 1991, **38**(3):123-127.

[14] RUCKLIDGE W. Efficient computation of the minimum Hausdorff distance for visual recognition [D]. Ithaca: Cornell University, 1995.

[15] ALEXA M. Recent advances in mesh morphing [J]. **Computer Graphics Forum**, 2002, **21**(2):173-196.

[16] PAPADAKIS P, PRATIKAKIS I, PERANTONIS S, *et al*. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation [J]. **Pattern Recognition**, 2007, **40**(9):2437-2452.

[17] TAUBIN G. Signal processing approach to fair surface design [C] // **Proceedings of the ACM SIGGRAPH Conference on Computer Graphics**. New York: ACM, 1995:351-358.

[18] ALHASHIM I, XU Kai, ZHUANG Yixin, *et al*. Deformation-driven topology-varying 3D shape correspondence [J]. **ACM Transactions on Graphics**, 2015, **34**(6):236.

# Research on 3D shape modeling using component matching and merging

BAI Maodong, HUA Shungang*, SU Tieming

(School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China)

**Abstract**: A 3D shape creation algorithm based on component matching and merging is proposed. Components are matched between two input shapes according to the Hausdorff distance metric to generate matched component-pairs. The components are mapped onto the spherical surface to carry out the spherical parameterization. The spherical meshes of two matched components are merged, and the vertices correspondence between matched component-pairs are built through reverse mapping. The interpolation is concluded to create a series of in-betweens in terms of the merging coefficient. Finally, the novel variations are obtained by relinking components with reference to the connection of input shapes. Experiments show that the proposed algorithm can generate reasonable and plausible variations, while retaining the function and adequate surface details of the input shapes.

**Key words**: components matching; spherical parameterization; mesh merging; relinking