

位移纹理形状创新应用

宣江华, 金连甫

(浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

摘 要: 三维模型的形状合成是计算机图形学和辅助设计的重要研究课题。借鉴三维模型的位移映射, 提出一种形状创新方法, 能够生成新颖的三维造型。包括四个部分: 首先从特征形状源模型提取三维图案作为位移纹理的生成元素; 其次根据宿主网格模型定义三维构图知识, 包括构图坐标系类型和其他纹理映射参数; 第三是宿主模型进行自适应剖分; 最后根据三维图案和映射参数生成位移纹理并与宿主模型融合后生成新的三维造型。

关键词: 位移映射; 自适应剖分; 三维图案; 形状创新

中图法分类号: TP391 文献标识码: A 文章编号: 1001-3695(2006)04-0172-03

Shape Innovation Based on Displacement Mapping

XUAN Jiang-hua, JIN Lian-fu

(College of Computer Science & Technology, Zhejiang University, Hangzhou Zhejiang 310027, China)

Abstract: Shape synthesis on 3D models is an important research field of computer graphics and CAD. The paper presents a shape innovation approach which is inspired by displacement mapping and can generate innovative shape. The approach includes four parts. The first is extracting 3D pattern from an interesting model which is taken as primitive of displacement texture. The second is defining the 3D design knowledge of host polygonal model, including mapping coordinate type and other parameters. The third is subdividing adaptively polygons of the host according to the design knowledge. At last the new model is generated by fusing the 3D patterns into the host model.

Key words: Displacement Mapping; Adaptive Subdivision; 3D Pattern; Shape Innovation

形状合成是计算机图形学的一个研究课题, 常用的方法包括形状过渡(Morphing) 和融合(Fusing)。形状过渡是指从一个模型 A 的形状逐渐变化到另外一个模型 B。常见的形状过渡可以分为基于网格模型和体模型的方法。基于网格模型的形状过渡通常包括建立顶点对应关系、拓扑合并和插值三个步骤^[1,2], 但是模型需要满足亏格为零甚至星型的拓扑要求。基于体模型的形状过渡包括特征对应^[3]与边界跟踪^[4]两种。前者需要用户指定两个模型间的对应特征点, 而后者只要把两个模型的方位统一即可。体模型的形状过渡的优势是其没有拓扑限制。形状融合是指把模型 A 的部分形状融合到宿主模型 B 之上生成新的形状, 目前较为常见的技术包括基于调和映射的网格融合^[5]、基于泊松方程的梯度编辑法^[6]。形状融合相对于形状过渡, 其形状创新能力更强。在计算机辅助设计中, 尤其是以形状创新为主要目的的领域, 这些技术由于速度和操作上的不足, 不利于设计者的发散思维, 同时缺乏自动形状推理能力, 所以本文提出了一种基于位移映射的形状创新方法。位移映射是 Cook 首先提出的^[7], 常用于图形的真实感绘制, 通常有微多边形^[8]、图像绘制^[9]和面片镶嵌^[10]三种方式。其中面片镶嵌类型的位移映射方式真实地改变了几何形状, 能够产生新的形状, 而且速度比微多边形类型算法更为有效。借鉴三维模型位移映射方法, 并考虑位移纹理提取、图案生成和融合控制, 本文提出一种形状创新方法, 以三角网格模型为计算对象, 能够自动生成新颖的三维造型。

1 三维图案的提取

三维图案拓扑结构类似于二维图像, 与图像不同之处在于其每个像素包含第三维的高度值和法矢量, 而不仅仅是像素的颜色灰度等。三维图案可以通过对二维图像上的像素附加高度值获取, 如通过对轮廓图上的像素计算与轮廓线的距离的方式形成三维图案。更方便和实用的方式是从感兴趣的三维模型提取, 即通过选择三维模型单侧视图的一定区域, 并计算该区域内可见点到参考面的距离, 提取三维图案, 这是本文采用的方法。像素法矢量通过高度值的中心差分法求取。

(1) 选择特征形状源模型 M_s , 而后根据需要调整至其最佳正交投影视图 V_s , V_s 满足以下两个条件: 能够充分表现特征形状 SH , 也就是 V_s 中尽量避免 SH 内的三维元素形成遮挡; 适合定义参考面 S_r , 便于计算特征形状内像素点到 S_r 的距离。

(2) 需要用户在 V_s 上选择包含了特征形状的封闭区域 A , A 内的每个像素将作为三维图案 PTN 的像素, 像素到 S_r 的距离是其高度值。

(3) 定义参考面 S_r , S_r 是形成三维图案的关键, 包括四类: 平面、柱面、球面和用户自定义曲面。

(4) 提取 A 内的像素, 转换为三维空间坐标后计算与参考面的距离, 作为像素的高度值, 最后形成一个矩形作为三维图案, 并对三维图案进行后期处理, 包括归一化、平滑、锐化和法矢量计算等。图 1 是运用平面参考面提取的一个三维图案。

参考面 S_r 的定义和像素高度值计算根据 S_r 类型不同需要不同的参数和计算方式。其中用户自定义曲面需要附加定

义距离计算矢量 VCT_d , 其距离计算方式如下: 设像素转换后三维坐标点为 PNT_1 , 通过 PNT_1 , 方向矢量为 VCT_d 的直线为 LN , LN 与 S_r 的交点为 PNT_2 , 矢量 $VCT_{21} = PNT_2 - PNT_1$, $|VCT_{21}|$ 为该像素的高度绝对值, 正负由 VCT_d 和 VCT_{21} 的内积符号确定。三维图案获取后, 后期处理的第一步是归一化, 使所有像素高度值 $h \in [0, 1]$, 为形状融合的计算提供统一规范。为了减少形状融合后的失真度, 本文还提供了两类函数用于边缘过渡, 即线性过渡和可变指数函数。最后因为三维图案和二维图像的表一致性, 为二维图像处理技术在三维形状上的应用提供了方便。

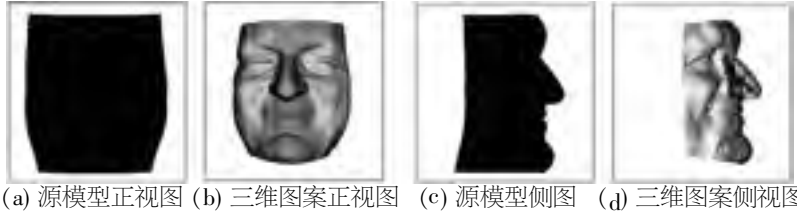


图 1 人脸三维图案

2 构图知识定义

三维图案 PTN 与宿主模型 M_h 的形状融合借鉴了纹理映射方法, 但是由于其三维特性, 需要控制的参数要比纹理映射复杂得多。本文给出了四种类似与纹理坐标系的 ST 坐标系类型及其融合控制参数, 提供方便灵活的形状融合手段。四种 ST 坐标系包括平面坐标系、柱形坐标系、球形坐标系和宿主模型曲面坐标系。下面以三维空间任意点 PNT 为例, 阐述不同坐标系下 ST 的计算过程。

平面坐标系 CDS_{pln} 定义包括四个参数: 原点 PNT_o , 单位法矢量 VCT_n , S 值方向矢量 VCT_s , T 值方向矢量 VCT_t 。 PNT 的 S 和 T 值的计算过程如下: 记 PNT 在 CDS_{pln} 上投影点为 PNT_p , 矢量 $VCT_{op} = PNT_p - PNT_o$ 。

$$s = VCT_s \cdot VCT_{op} / |VCT_s|^2 \quad t = VCT_t \cdot VCT_{op} / |VCT_t|^2$$

柱形坐标系 CDS_{cln} 定义包括三个参数: 起始点 PNT_b , 终点 PNT_e , S 值起始方向矢量 VCT_s 和 S 值第一半圆标志矢量 VCT_{sfo} 。 PNT 的 U 和 V 值的计算过程如下: 记矢量 $VCT_t = PNT_e - PNT_b$, PNT_b 和 PNT_e 所在直线 LN_{axis} , VCT_s 、 VCT_t 和 VCT_{sf} 两两相互垂直, PNT 在 LN_{axis} 上的投影为 PNT_p , 矢量 $VCT_{bp} = PNT_p - PNT_b$, 矢量 $VCT_p = PNT - PNT_{po}$ 。

$$s_o = \begin{cases} 0, VCT_p \cdot VCT_{sf} \geq 0 \\ \pi, VCT_p \cdot VCT_{sf} < 0 \end{cases}$$

$$s = \frac{\arccos(VCT_p \cdot VCT_s / |VCT_s| \cdot |VCT_p|) + s_o}{2\pi}$$

$$t = VCT_t \cdot VCT_{bp} / |VCT_t|^2$$

球形坐标系 CDS_{spr} 的定义包括三个参数: 球心 PNT_c , S 值起始矢量 VCT_s , S 值第一半圆标志矢量 VCT_{sf} , T 值起始矢量 VCT_t 。 PNT 的 S 和 T 值计算过程如下: 记矢量 $VCT_c = PNT - PNT_c$ 。

$$s_o = \begin{cases} 0, VCT_p \cdot VCT_{sf} \geq 0 \\ \pi, VCT_p \cdot VCT_{sf} < 0 \end{cases}$$

$$s = [\arccos(VCT_c \cdot VCT_s / |VCT_s| \cdot |VCT_c|) + s_o] / 2\pi$$

$$t = VCT_c \cdot VCT_t / |VCT_c| \cdot |VCT_t|$$

宿主模型曲面坐标系 CDS_{surf} 的定义方式与平面坐标系类似, 但是其距离计算不是矢量的模, 而是测地线的长度。其定

义包括三个参数: 原点 PNT_o , S 向终点 PNT_s 和 T 向终点 PNT_t , 这三个点都必须在模型曲面上。 PNT 的 S 和 T 值的计算过程如下: 记三个点确定投影平面 PLN_p , 矢量 $VCT_s = PNT_s - PNT_o$, 矢量 $VCT_t = PNT_t - PNT_o$, PLN_p 的单位法矢量为 VCT_n , PNT_o 到 PNT_s 和 PNT_t 的测地线长度 len_s 和 len_t , PNT 在 PLN_p 上的投影为 PNT_p , 通过 PNT 和 PNT_p 的直线与宿主模型曲面的交点为 PNT_i , 矢量 $VCT_{oi} = PNT_i - PNT_o$ 。

$$VCT_n = \text{normalize}(VCT_s \cdot VCT_t)$$

$$len_s = \text{geodesic}(PNT_o, PNT_s)$$

$$len_t = \text{geodesic}(PNT_o, PNT_t)$$

$$s = \frac{\text{geodesic}(PNT_o, PNT_i) \cdot VCT_s \cdot VCT_{oi}}{|VCT_s| \cdot |VCT_{oi}| \cdot len_s}$$

$$t = \frac{\text{geodesic}(PNT_o, PNT_i) \cdot VCT_t \cdot VCT_{oi}}{|VCT_t| \cdot |VCT_{oi}| \cdot len_t}$$

其中 normalize 和 geodesic 是矢量归一化和测地线长度计算函数。

在形状融合过程中, 除了 S 和 T 值外, 本文中还定义了一些其他一些参数满足计算三维特性和灵活控制的需要。 PM_{offset} : 高度偏移值, 即图像所有像素点高度的统一偏移量。 PM_{scale} : 高度缩放因子, 即图像所有像素点高度值的缩放比例。 PM_{vt} : 用于选择三维模板上像素点的二维单位矢量, $(1, 1)$ 表示左下角作为模板原点, $(-1, -1)$ 则是右上交作为模板原点, 按长宽比例选取像素点。 PM_{rs} , PM_{rt} : 三维图案在 S 和 T 方向重复次数。 PM_{brs} , PM_{brt} : 在 S 和 T 方向上图案间的间隔比率。 PM_{mins} , PM_{maxs} , PM_{mint} , PM_{maxt} : 参与形状融合的点 S 和 T 值界限。

PM_{nc} , PM_{fc} : 参与形状融合的点距离对应参考点最近和最远界限。 PM_{ne} : 参与形状融合的点的法矢量偏离界限。最后三类界限参数限制了参与形状融合计算的空间范围, 只有在此范围内的点才会进行坐标变换。设 PNT 在不同 ST 坐标系下对应参考点为 PNT_r (PNT_r 的计算在第五节中详细说明), 矢量 $VCT_r = PNT_r - PNT$, PM_{nc} 和 PM_{fc} 限制对象是 $|VCT_r|$, PM_{ne} 的限制对象是点 PNT 的法矢量和 VCT_r 间的最大夹角。

3 多分辨率自适应剖分

根据上一节定义的参数, 可以从确定宿主三维模型中各个定点的 S 和 T 值, 进而知道需要参与计算的顶点。但是由于宿主模型的三角面片分辨率往往不能满足形状融合的需要, 所以有必要把宿主模型的三角面片进行剖分, 同时考虑到运算效率, 要尽量减小网格剖分的范围和次数, 所以本文采用了一种仅对形状融合范围内三角面片的有限剖分算法。记 VTX 为宿主模型是所有顶点集合, VTX_o 为在计算范围外的顶点集合, VTX_i 为计算范围内的顶点集合, TRI 为所有三角面片的集合。根据第二节介绍三类界限参数, 把每个顶点归入 VTX_o 和 VTX_i 。三角面片可以分为 TRI_o , TRI_i , TRI_1 和 TRI_2 四类, 定义如下:

$$VTX = VTX_o \cup VTX_i$$

$$VTX_o \cap VTX_i = \emptyset$$

$$TRI_o = \{ T(P_1, P_2, P_3) \mid P_1, P_2, P_3 \in VTX_o \}$$

$$TRI_i = \{ T(P_1, P_2, P_3) \mid P_1, P_2, P_3 \in VTX_i \}$$

$$TRI_1 = \{ T(P_1, P_2, P_3) \mid P_l \in VTX_i, P_m, P_n \in VTX_o \} \quad l, m, n \in \{1, 2, 3\}, l \neq m, m \neq n$$

$$TRI_2 = \{ T(P_1, P_2, P_3) \mid P_l \in VTX_o, P_m, P_n \in VTX_i \}$$

$$TRI = TRI_o \cup TRI_i \cup TRI_1 \cup TRI_2, TRI_m \cap TRI_n = \emptyset \quad m \neq n, TRI_m, TRI_n \in \{TRI_o, TRI_i, TRI_1, TRI_2\}$$

首先要把 TRI_1 和 TRI_2 内的三角面片进行剖分, 剖分后的三角面片分别属于 TRI_o 或 TRI_i , 根据界限参数可以把这些三

角面片做如图 2 所示的剖分。接着是对 TRI_i 内的三角面片 $T(P_1, P_2, P_3)$ 进行二分递归剖分, 计算三个顶点在 ST 平面上的二维距离, 距离最大的两个顶点组成的边作平均剖分, 生成两个子三角面片。递归的退出根据两个条件, 即递归深度, 这是为了保证计算效率; 顶点间最大 ST 距离是否已经小于三维图案像素点 S 和 T 方向跨度中的最小值。像素点 S 和 T 方向跨度的计算在下一节介绍。

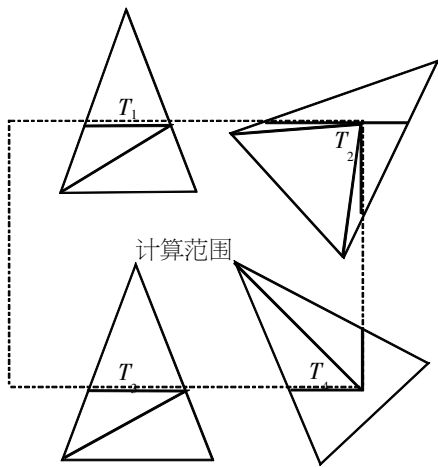


图 2 边界三角面片分割示意图

4 形状的融合

通过前面的参数定义和宿主模型的多分辨率自适应剖分, 已经能够为形状融合提供形状源, 最后需要把计算范围内的顶点与三维图案进行融合。为顶点 VTX_h 计算其 S 和 T 值, 这已经在第三节进行了详细介绍; 要找到三维图案 PTN 中与顶点相对应的像素 PXL , PXL 的高度值 h_{pxl} 作为顶点的偏移距离计算依据, 偏移方向则根据矢量 $VCT_r = VTX_h - PNT_r$, 其中 PNT_r 是 VTX_h 在 ST 坐标系下的参考点。 PNT_r 在平面坐标系下是 VTX_h 在 CDS_{pln} 上的投影点 PNT_p ; 在柱形坐标系下是 VTX_h 在中心轴直线 LN_{axis} 上的投影点 PNT_p ; 在球形坐标系下是球心 PNT_c ; 在宿主模型曲面坐标下是 VTX_h 在曲面上的投影点 PNT_i 。记 PTN 的高度为 h_{ptn} , P 的宽度为 w_{ptn} , VTX_h 的 S 和 T 值为 s 和 t , PXL 的行坐标为 r_{pxl} , 列坐标为 c_{pxl} , 每个像素在 S 和 T 方向的跨度 $span_s$ 和 $span_t$ 。

$$span_s = \frac{PM_{maxs} - PM_{mins}}{(1 + PM_{brs}) \cdot w_{ptn} \cdot PM_{rs}}$$
$$off_s = \begin{cases} s - PM_{mins}, PM_{vct}[0] \geq 0 \\ PM_{maxs} - s, PM_{vct}[0] < 0 \end{cases}$$
$$r_{pxl} = (off_s / span_s) \% ((1 + PM_{brt}) \cdot h_{ptn})$$
$$c_{pxl} = (off_t / span_t) \% ((1 + PM_{brs}) \cdot w_{ptn})$$

$$span_t = \frac{PM_{maxt} - PM_{mint}}{(1 + PM_{brt}) \cdot h_{ptn} \cdot PM_{rt}}$$
$$off_t = \begin{cases} t - PM_{mint}, PM_{vct}[1] \geq 0 \\ PM_{maxt} - t, PM_{vct}[1] < 0 \end{cases}$$

这样得到顶点对应的 PXL 及其高度值 h_{pxl} 和法矢量 NML_{pxl} 。 VTX_h 的偏移方向是 VCT_r , 偏移值为:

$$dst = \begin{cases} PM_{offset} + h \cdot PM_{scale}, & r_{pxl} < h_{ptn}, c_{pxl} < w_{ptn} \\ PM_{offset} \end{cases}$$

VTX_h 的法向 NML_{vx} 是 NML_{pxl} 以 VCT_r 为坐标轴方向进行旋转变换后得到的。计算范围内所有顶点经过这样的坐标变换后, 形成新的融合三维造型。

5 实现与实例

在实现过程中, 本文主要设计了几个类: CPolygon, CPattern, CCoordinates, CMap, CMaps。CPolygon 提供三维图案的提取和网格剖分功能, CPattern 提供三维模板自身的设置和数据

访问, CCoordinates 用力定义 ST 坐标系及 S 和 T 值的计算, CMap 把 ST 坐标系及其参数与三维模板关联, CMaps 是 CMap 的集合, 宿主模型与根据 CMaps 中的 CMap 对象分别进行形状融合。另外实现中本文采用与 XYZ 轴方向平行的宿主模型最小包围盒的最大半边长 $len_{halfedge}$ 作为三维模板高度值的缩放比例进行预处理, 即三维模板在与宿主模型融合时像素高度值 $h \in [0, len_{halfedge}]$ 。图 3 是一个利用人脸和龙的三维图案, 以苹果和钟作为宿主模型的形状融合实例。

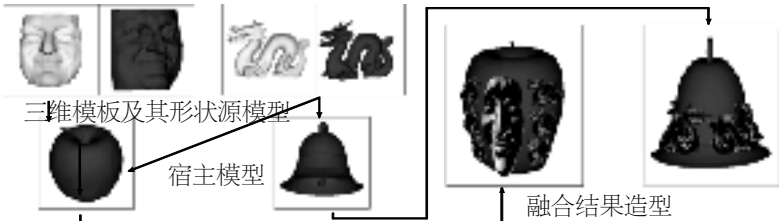


图 3 形状融合实例

6 小结

本文的方法能够直观方便地实现形状融合, 并且通过提供多种 ST 坐标系和三维构图知识参数, 能够比较灵活地控制形状融合的方式, 最后生成新颖的三维造型。在设计和试验过程中也发现一些有待改进的方面: 存在遮挡关系的模型或其视图, 被遮挡的形状无法体现于三维模板, 这个问题需要采取其他处理算法, 可以考虑三维模板的立体选择并结合调和映射。

宿主模型和三维模板交接的地方存在走样, 本文提出了三维模板边缘过渡改善这个问题, 另外还可以从宿主模型的剖分方向和形状过渡考虑提高效果。完全自适应剖分后模型过大, 本文采用递归深度控制解决, 但会损失一定的精度。综上所述, 本文提出的方法虽然不能实现任意模型的形状完全融合, 但是已经能够为形状设计者提供一种有效的形状创新途径。

参考文献:

[1] ent J R, Carlson W E, Parent R E. Shape Transformation for Polyhedral Objects[J]. Computer Graphics, 1992, 26(2): 47-54.

[2] Kanai T, Suzuki H, et al. 3D Geometric Metamorphosis Based on Harmonic Map[J]. The Visual Computer, 1998, 14(4): 166-176.

[3] Leros A, Garfinkle C D, Levoy M. Feature-based Volume Metamorphosis[C]. SIGGRAPH, 1995. 449-464.

[4] Breen D E, Whitaker R T. A Level-Set Approach for the Metamorphosis of Solid Models[J]. IEEE Transactions on Visualization and Computer Graphics, 2001, 7(2): 173-192.

[5] 刘刚, 金小刚, 彭群生. 蒙太奇网格融合[J]. 软件学报, 2002, 14(8): 1425-1432.

[6] Yu Y, Zhou K, Xu D, et al. Mesh Editing with Poisson-based Gradient Field Manipulation[C]. SIGGRAPH, 2004. 644-650.

[7] Cook R L. Shade Trees[C]. SIGGRAPH, 1984. 223-231.

[8] Cook R L, Carpenter L, Catmull E. The Reyes Image Rendering Architecture[J]. Computer Graphics, 1987, 21,(4): 95-102.

[9] Wang L, Wang X, Tong X, et al. View-dependent Displacement Mapping[C]. SIGGRAPH, 2003. 334-339.

[10] Doggett M, Hirche J. Adaptive View Dependent Tessellation of Displacement Maps[C]. Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware, 2000. 59-66.

作者简介:

宣江华(1978-), 男, 浙江上虞人, 硕士研究生, 主要研究方向为计算机网络与通信、计算机图形学等; 金连甫(1955-), 男, 浙江台州人, 副教授, 硕士生导师, 主要研究方向为计算机网络应用、计算机网络安全等。