基于图像化几何的三维模型彩绘*

张新宇[1+], 张三元[1], 叶修梓[1,2,3]

[1](浙江大学 计算机科学与技术学院,浙江 杭州 310027)

[2](浙江大学 CAD&CG 国家重点实验室,浙江 杭州 310027)

[3](SolidWorks 公司,美国)

Painting Based on Imaged Geometry for 3D Models

ZHANG Xin-Yu[1+], ZHANG San-Yuan[1], YE Xiu-Zi[1,2,3]

[1](College of Computer Science, Zhejiang University, Hangzhou 310027, China)

[2](State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

[3](SolidWorks Corporation, USA)

+ Corresponding author: Phn: +86-571-87953039, E-mail: zhangxinyu@zju.edu.cn, http://www.zju.edu.cn

Received 2003-04-03; Accepted 2003-10-08

Zhang XY, Zhang SY, Ye XZ. Painting based on imaged geometry for 3D models. *Journal of Software*, 2004,15(3):461~467.

http://www.jos.org.cn/1000-9825/15/461.htm

Abstract: Painting system based on texture mapping is often limited by the model's parameterization in a 2D texture space. For models with complex topologies or complicated distributions of the structural details, finding the parameterization can be very difficult and usually must be performed manually. Here a novel data representation and a system for direct painting on 3D surface are presented. By creating 3D colored points for each triangle, the system can support a great variety of painting operations similar to the conventional 2D pixels editor. One key ingredient of this method is a novel adaptive data representation including geometry, topology and color. This technique is called the imaged geometry, which allows users to treat each triangle as a triangle image, and to paint the points on 3D triangle surface without any parameterization. This system can take any triangle model as an input and produces an output with colored points on triangles of the model surface. Because this method is adaptive, details are created in the triangles required by the texture painting, which reduces memory usage. Based on the imaged geometry, the models with complex topologies can be painted in an easy way.

Key words: 3D painting; painting system; 3D content creation; data structure

* Supported by the National Natural Science Foundation of China under Grant Nos.60273060, 60073026 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2002AA411010 (国家高技术研究发展计划(863)); the Software Special Project of the National Ministry of Science and Technology of China under Grant No.2003AA4Z1020 (国家科技部软件重大专项)

作者简介: 张新宇(1975－),男,新疆轮台人,博士生,主要研究领域为计算机辅助设计和图形学,数字几何处理;张三元(1963－),男,博士,教授,主要研究领域为计算机辅助设计和图形学,曲面造型,虚拟现实和图像处理;叶修梓(1966－),男,博士,教授,博士生导师,主要研究领域为计算机辅助设计和图形学,图像处理,生物信息技术,网络和信息安全技术.

摘 要: 基于纹理映射的彩绘系统通常受到从三维模型空间到二维纹理空间参数化的限制.如果一个模型具有复杂的拓扑结构或高度复杂的表面细节,寻找一个好的参数化方法通常是非常困难的,即使可以实现,通常也不能自动完成,需要用户手工交互进行.针对这种问题,提出了一种用于在三维模型表面直接进行彩绘的数据表示方法和一个原型系统,通过在三维模型中的每一个三角形上生成带有颜色信息的几何点,该系统可以支持许多与二维图像编辑类似的操作.其中一个关键的内容是提出了一种包含几何、拓扑和色彩信息在内的自适应的数据表示方法.这种方法被称为图像化的几何,它允许用户将每一个三角形视为一个三角形图像,并可以在不需要任何参数化的前提下在三维模型表面进行彩绘.系统的输入为一个常见的三角形网格模型,通过系统处理后变成一个在三角形上分布着带有颜色信息的几何点的三维表面.因为创建过程是自适应的,因此只有那些需要进行绘画纹理的三角形才需要创建几何点,这样可以节省存储消耗.将这种数据表示用于三维模型彩绘可以很好地处理具有复杂拓扑结构的模型.

关键词: 三维模型彩绘;彩绘系统;三维内容创建;数据结构

中图法分类号: TP391 **文献标识码**: A

传统的纹理映射将二维图像应用到三维模型上,在不增加几何数据的情况下,用以表示三维模型表面特征.该技术被广泛地应用到许多表面属性的表示上,如表面颜色、法向量、透明度、光照、表面位移等.大多数彩绘系统使用纹理映射将三维几何和拓扑关系映射到二维纹理空间,因此这种三维彩绘系统受三维模型到二维纹理空间参数化过程的限制,其根本原因是:所有的参数化方法都不可避免地产生不连续性、变形和其他缺陷.

我们的彩绘系统将二维图像技术扩展到三维模型中的每一个三角形中(在此我们假设给定的模型为三角形网格).为达到这个目的,我们在每一个三角形内生成带有颜色信息的几何点,这类似于二维图像中的像素.只要这些几何点足够密,整个模型将被这些带有颜色信息的几何点所覆盖(如图 1 所示).我们可以根据每一个三角形的面积和周长自适应地为其生成足够多的几何点.

(a) Two imaged triangles (b) An imaged bunny triangle model

(a) 两个图像化的三角形 (b) 一个图像化的兔子模型

Fig.1 Imaged geometry

图 1 图像化的几何

将该方法应用到三维彩绘系统中,有以下优势:

首先,这是一种新颖的三维模型数据表示,包含了几何和颜色信息,可以方便地应用到三维彩绘系统中;其次,彩绘系统不需要参数化过程;另外,如果用户需要,传统的纹理映射技术依然可以应用到该模型上,而且可以实现从纹理映射方法到我们提出的图像化的几何之间的数据转换.

然而这种数据表示也有缺点.数据量随着生成的三角形图像阶数的增加而增加,因此,为了运行系统和存储模型,我们需要更多的存储空间.

本文第 1 节介绍相关工作.第 2 节描述图像化几何的数据表示.第 3 节讨论基于图像化几何进行三维彩绘的方法.第 4 节给出几个彩绘结果实例.第 5 节对今后的研究内容进行展望.第 6 节是结论.

1 相关工作

利用三维彩绘系统,设计人员可以直接在三维模型表面生成纹理细节.Hanrahan 等人[1]首次提出并实现了在三维表面进行直接彩绘的设想.现在该方法已被许多商业三维系统所采用.在这些系统中,设计人员首先要准备一个 UV 映射,然后系统将屏幕空间上的笔画先投影到三维模型表面上,再投影到二维纹理空间.设计人员可以在这样的系统中通过编辑二维纹理图像直接对三维模型进行彩绘.但是准备 UV 映射即使对专业人员也是一件非常困难和繁琐的工作,更何况许多设计人员并不熟悉这一数学过程,因此一些人提出自动映射的方法[2~6].然而,我们知道自动映射同样不能解决参数化本身造成的变形和缺陷,而且这种自动化过程并不都能自动完成.

最近几年,一些学者开始关注无须参数化的彩绘方法.Debry 等人[7]提出用八叉树来存储纹理数据,进而实现无须参数化的纹理映射.Benson 等人[8]几乎同时也提出用自适应的八叉树存储纹理数据.利用八叉树数据结构的三维彩绘还有待进一步研究.

另外,Zwiker 等人[9]提出的针对点集曲面的相关工作与我们的工作有相似之处.在他们的系统中,也是利用三维几何点,并将其视为三维图像基元.但一个明显的区别是,他们处理的模型是三维点集曲面,输出也是点集曲面.点集曲面不存储连接信息,因此不涉及参数化过程.在过去的几年中,其他彩绘系统针对多边形模型、三角网格模型[10]和隐式曲面[11,12]作了相应的研究.与之相比,我们的方法完全不同.

2 数据表示

为了实现无参数化的三维彩绘,我们提出了一种灵活的而且能够自适应生成的数据表示.正如前面提到的,我们的目的是为了免去繁琐的参数化过程,但是如果没有二维图像空间,那些原本存储在二位图像中的纹理信息就必须转而存储在模型中.

假设给定一个三角网格模型 M,我们需要为其中的每一个三角形生成附着于其上的带有颜色信息的几何点,并且称这样的三角形为三角形图像.如果 M 中的一个三角为 t_i,为了在三角形 t_i 表面生成均匀的几何点,我们采用如下方法:首先在三角形 t_i 的三条边上分别均匀地生成 n_i+1 个几何点,我们将 n_i 称为三角形图像的阶数;然后连接相邻两边上的点,并保留那些平行于 3 条边的线段,这些线段在三角形内部的交点加上三条边上的等分点就是那些待创建并且带有颜色信息的几何点的位置.对每一个三角形 t_i 而言,将生成$(n_i+1)(n_i+2)/2$ 个几何点,其中包括三角形的 3 个顶点的位置(如图 2 所示).每一个三角形的 n_i 值是不同的,这也正是自适应和多分辨率的思想所在.如果我们用一个索引值表示一个几何点在三角形中的位置,三角形 t_i 共有$(n_i+1)(n_i+2)/2$ 个索引值,在这些位置上将存储颜色信息(r,g,b,a).我们将由这种方法生成的数据称为图像化的几何.利用这种方法,可以由原模型 M 生成一个新模型 M',新模型包含了那些图像化的几何信息.

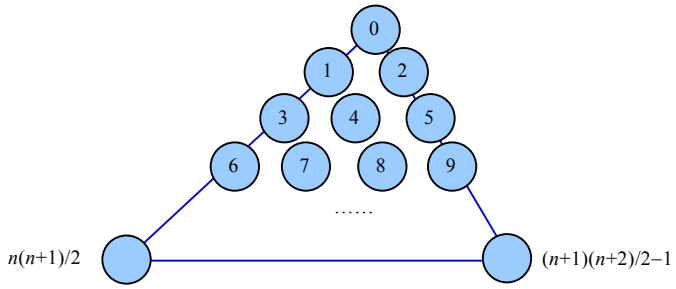

Fig.2 Creation of points on a triangle

图 2 一个三角形内几何点的创建

在我们的系统中,用一个整型来表示一个几何点的位置索引,用 4 个字节表示一个颜色信息.实际上,位置信息隐含在颜色信息数组的序列中,因此我们不需要存储它.这样,每一个三角形增加的数据量为 $2(n_i+1)(n_i+2)$.下表 1 列出了 Stanford 兔子模型的数据增加量(其中 n_i=7).原始模型的顶点数为 1 494,三角形数为 2 915,需要的存储空间为 46 932 字节.在生成图像化的几何后,新模型需要 513 332 字节.M'和 M 的数据量之比

为 10.94,这个数值随着 n_i 的增加而增加.

Table 1 Data increase towards the Stanford bunny triangle model
表 1 Stanford 兔子模型的数据增加量

Triangles	Original data (Byte)	Result data (Byte)	Ratio
2 915	46 932	513 332	10.94

2.1 多分辨率

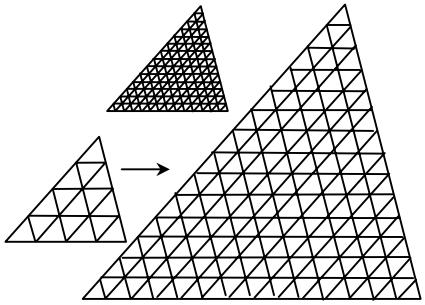

Fig.3 Multiresolution representation
图 3 多分辨率表示

在过去的几年中,多分辨率的思想在图形学的许多领域(如几何造型和网格处理)备受青睐.在我们的系统中,为了降低存储消耗,我们也采用多分辨率的技术来表示三角形图像,即不同的三角形使用不同的阶数,在不同的缩放情况下,分辨率也不同.

正如前文所述,多分辨率图像化的几何根据三角形面积和周长来确定其分辨率.如图 3 所示,左边面积较小的三角形的阶数也应较小,如果它具有与中间的三角形同样的阶数,看上去就像左上角的三角形,所有的几何点重叠在一起,会造成存储上的浪费.

我们在图 4 中用更直观的方式模拟并说明了这种情况.图 4(a)比图 4(b)和图 4(c)的分辨率更合理.但是,从另一个角度说,如果追求更多的细节和更高的平滑度,我们可能需要相邻的点稍微重叠在一起,这时图 4(c)就更合适了.

(a) (b) (c)

Fig.4 The colored points would lap over if the resolution is higher than the original. Middle and right triangle have over-lapped points
图 4 如果分辨率过高,带有颜色信息的几何点将重叠在一起.中间和右边的三角形有重叠的几何点

另一方面,我们知道,图形库(如 OpenGL)可以指定几何点光栅化的大小.在我们的机器上,在反走样状态下,OpenGL 所能支持的光栅化点的大小为 1.0~10.0.因此,利用三角形的面积、周长和光栅化点的大小,就可以决定合适的分辨率和在此分辨率下带有颜色信息的几何点的大小.图 5 左边的三角形根据这几个参数得到比较合适的分辨率.当通过变焦希望得到模型的更多细节时,我们可以提高分辨率,以消除由于低分辨率造成的相邻点之间的缝隙.图 5 右边的三角形说明了由于变焦或不合理初始化造成的不合适的分辨率.

Fig.5 Left triangle (high resolution) and right triangle (low resolution)
图 5 左三角形(高分辨率)和右三角形(低分辨率)

2.2 反走样

为了消除生成的几何点尖锐的边和角对显示效果的影响(如图 6 所示),图形库中的反走样参数应该设置为启动状态,这一技术对我们的系统非常重要.另一个反走样的问题是:如果两个相邻的三角形的分辨率不同,在相邻边上,由不同三角形生成的几何点将出现错位.图 7 是这种情况的模拟结果.我们的系统解决这种反走样的一个简单方法是:限制相邻三角形图像阶数和几何点光栅化大小的差别,这个差值一般限定为 1 或 2.用户也可

以根据实际情况手动调整这个参数.左边的三角形图像的阶数为 20,几何点光栅化大小 10.0;右边的三角形图像的阶数为 25,几何点光栅化大小 7.0.

Fig.6 Rasterized points without antialiasing

图 6 无反走样情况下光栅化的点

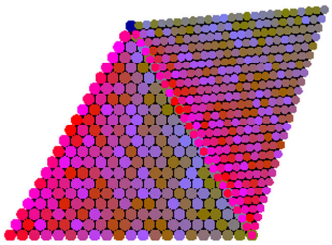

Fig.7 Aliasing result because of the difference of the order and point size between two adjacent triangles

图 7 由于相邻的三角形的阶数和光栅化点的大小不同造成的走样

3 彩绘方法

本节论述如何在一个给定的三角形网格模型上进行彩绘.首先,我们要为整个模型打一个底色.随后,我们在屏幕上移动鼠标,在模型表面生成相应的色彩.图 8 提前展示了两个图像化的三角形及其上的彩绘效果.

(a) An imaged triangle with blue base color and painting a lovely footprint and the abbreviate word of our lab (CGIM)

(a) 一个图像化的三角形,打了蓝色的底色,在上面画了一个脚丫和我们实验室的英文缩写(CGIM)

(b) An imaged triangle with mixed base texture and a monster head

(b) 一个图像化的三角形,打了自动生成的混合底色,在上面画了个怪物头像

Fig.8

图 8

我们的系统采用传统的用户界面,而不是一些系统所采用的接触式交互工具.在一般的二维绘画系统中,鼠标光标指定了笔刷的位置,然而在三维系统中,笔刷首先要根据屏幕上的坐标在三维模型表面定位.

3.1 底 色

通常,我们先用一个单一颜色或一幅图像对三维模型表面纹理进行初始化,这相当于在绘画之前先打一个底色.对单一颜色,我们还不需要对三角形图像化,只需要设置三角形的 3 个顶点的颜色即可,图形硬件将通过插值计算三角形内部点的颜色.只有当用户进入彩绘的第 2 步时才需要生成相应的三角形图像.当然,我们也可以用一幅图像作为底色,即我们的数据表示可以利用传统的纹理映射,并且可以将给定的纹理图像转换成图像化的几何.实现起来也很简单,描述如下:首先,根据前述的方法生成图像化的几何;然后,利用 UV 映射为每一个几何点寻找图像空间的相应像素位置,并将像素值读取到几何点的颜色值中.三角形内部点对应图像空间的点可以由所属三角形的 3 个顶点的两次线性插值得到.

3.2 二维屏幕空间到三维模型空间

我们可以根据鼠标在屏幕上的坐标,找到预先设定的笔刷在三维模型表面的位置.在 OpenGL 中可以利用

选择和反馈机制来实现这种交互操作.利用选择和反馈机制,用户可以通过鼠标点击和移动在屏幕上设定一个窗口,通过投影变换决定在模型空间中哪些数据落在这个窗口中.基于该技术,我们可以获得鼠标指定坐标下正在处理的三角形,如果这个三角形没有被图像化,并且当前颜色与笔刷颜色不同,我们就立即将这个三角形图像化,并再次利用选择机制,为那些落在笔刷下的几何点用笔刷颜色进行着色.在整个过程中,系统自动实现前面提到的多分辨率和反走样问题.其他更复杂的笔刷可以考虑用更多的参数加以设计,如法向量、笔刷形状等.

4 结 果

通过一些模型,验证了我们的系统.上一节我们已经预先展示了两个图像化三角形和彩绘的结果(如图 8 所示).用户可以在几分钟内完成这些作品.图 9 左边是一个小鹿模型,其中字母 r 被单独提取出来放大,用以显示模型表面带有颜色信息的几何点.图 9 右边是一个轮胎模型,上面打了许多孔.我们特意在 SolidWorks 中生成这种规格非零的拓扑复杂的模型,用以说明对这种无法参数化的模型,我们的方法依然可以轻松处理.我们在轮胎上画了两个可爱的卡通蜗牛,操作起来就像用画图板进行二维图像创作一样,用户可以在十几分钟内完成这样的彩绘过程.正如在讨论多分辨率情况时所述,彩绘质量受图像化几何的分辨率和光栅化几何点大小的影响.一般情况下,分辨率越高,光栅化几何点越小,彩绘模型的显示质量越高;反之则越差.

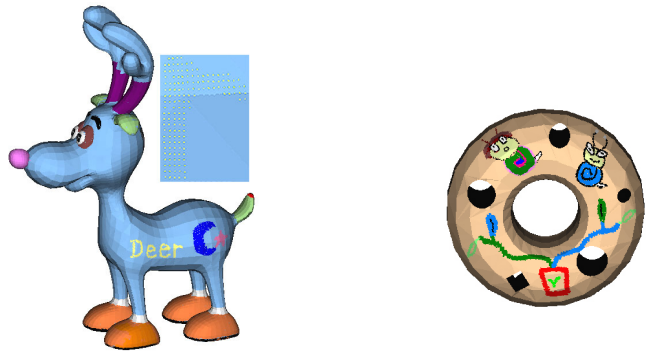

Fig.9 Result examples: The left shows a painted deer, the right is a broken torus

图 9 结果实例:左边是一个小鹿模型,右边是一个打了孔的轮胎模型

5 今后的工作

现在,我们的系统只是一个原型系统,因此有许多可以扩展和改进的地方:

• 这种新颖的三维模型数据表示不仅可以应用到三维彩绘上,而且可以实现网格模型和点集曲面混合造型.

• 为了能够将该方法应用到大型数据集上,并且达到彩绘和交互操作的实时性,需要进一步降低时间和存储消耗.

• 当模型中的三角形不满足 Delaunay 三角化条件时,如何更好地实现反走样还有待进一步的研究.

6 结 论

三维彩绘将传统的二维绘画扩展到三维.很长一段时间,人们利用纹理映射和参数化对三维模型进行彩绘.利用这种技术的三维彩绘系统已经可以实现,但是参数化限制了其应用.因此,我们提出一种新颖的数据表示以实现在任意三角网格模型上直接进行三维彩绘,这是一种更为一般的数据表示,其思想与用八叉树实现无参数化的纹理映射的方法完全不同.这种图像化几何的方法将每一个三角形视为一个三角形图像,可以直接对三角形内的几何点进行着色,而无须任何参数化和映射处理.系统的输出是原始网格加上附着于其上的带有颜色信息的几何点.将这种数据表示用于三维模型彩绘,可以很好地处理具有复杂拓扑结构的模型.

References:

[1] Hanrahan P, Haeberli P. Direct WYSIWYG painting and texturing on 3D shapes. In: Forest B, ed. Proc. of the SIGGRAPH 1990. New York: ACM Press, 1990. 215~224.

[2] Haker S, Angenet S, Tannenbaum A, Kikinis R, Sapprio MH. Conformal surface parameterization for texture mapping. IEEE Trans. of Visualization and Computer Graphics, 2000,6(2):181~189.

[3] Igarashi T, Cosgrove D. Adaptive unwrapping for interactive texture painting. In: John H, ed. Proc. of the 2001 ACM Symp. on Interactive 3D Graphics. New York: ACM Press, 2001. 209~216.

[4] Lévy B. Constrained texture mapping for polygonal meshes. In: Fiume E, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 417~424.

[5] Maillot J, Yahia H, Verrobust A. Interactive texture mapping. In: James TK, ed. Proc. of the SIGGRAPH 1993. New York: ACM Press, 1993. 27~34.

[6] Sander PV, Snyder J, Gortler SJ, Hoppe H. Texture mapping progressive meshes. In: Fiume E, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 409~416.

[7] Debry D, Gibbs J, Petty DD, Robins N. Painting and rendering textures on unparameterized models. In: John H, ed. Proc. of the SIGGRAPH 2002. New York: ACM Press, 2002. 763~768.

[8] Benson D, Davis J. Octree textures. In: John H, ed. Proc. of the SIGGRAPH 2002. New York: ACM Press, 2002. 785~790.

[9] Zwicher M, Pauly M, Knoll O, Gross M. Pointshop 3D: An interactive system for point-based surface editing. In: John H, ed. Proc. of the SIGGRAPH 2002. New York: ACM Press, 2002. 322~329.

[10] Agrawala M, Beers AC, Levoy M. 3D painting on scanned surfaces. In: Robert K, ed. Proc. of the SIGGRAPH 1995. New York: ACM Press, 1995. 145~150.

[11] Pedersen HK. Decorating implicit surfaces. In: Robert K, ed. Proc. of the SIGGRAPH 1995. New York: ACM Press, 1995. 291~300.

[12] Perry RN, Frisken SF. Kizamu: A system for sculpting digital characters. In: Fiume E, ed. Proc. of the SIGGRAPH 2001. New York: ACM Press, 2001. 47~56.