

三维模型表面特征的水墨效果渲染算法

景 昊 周秉锋
(北京大学计算机科学技术研究所 北京 100871)
(jinghao@icst.pku.edu.cn)

摘 要 提出一种对三维模型实现水墨效果渲染的两步二维纹理映射方法. 该方法首先通过模型顶点的视线向量和法线向量的点乘积计算其夹角的余弦值, 作为第一步的纹理坐标进行映射; 再通过欧拉公式计算径向曲率及其方向导数值, 作为第二步映射的纹理坐标. 为达到实时的交互浏览, 在第二步映射时还可采用基于视点无关的曲率特征映射策略 (如平均曲率、高斯曲率、主曲率等). 在快速生成水墨画效果的同时也显示出更多的特征细节.

关键词 非真实感绘制; 三维模型特征; 径向曲率; 水墨效果
中图法分类号 TP391

Chinese Painting Style Rendering for Surface Features on 3D Models

Jin gHao Zhou Bin gfeng
(Institute of Computer Science and Technology, Peking University, Beijing 100871)

Abstract We present a two-pass texture mapping procedure to produce Chinese painting style images for 3D models. By the procedure, we first use the cosine values of the angle between view vector and normal vector for each vertex as the texture coordinates, through calculation of the dot product of the two vectors. The derivative of radial curvature on the model surface is used as the second pass texture mapping coordinates, by calculation using Euler formula. To improve rendering speed, we explore other rendering strategies in the second pass mapping based on view-independent curvature values such as mean curvature, Gaussian curvature, or principle curvature. Our algorithm cannot only render the model efficiently in Chinese painting style, but also often shows additional features of the 3D model.

Keywords non-photorealistic rendering; 3D model feature; radial curvature; Chinese painting style

与真实感绘制技术关注于模拟真实世界的光照环境进行渲染不同,非真实感绘制技术更加强调艺术表现力.在非真实感的渲染风格中,卡通渲染技术^[1]已经被大多数人所认识.卡通渲染通过为对象添加明暗部和颜色使得对象更具有层次感且更加生动.与用平滑线性插值方法计算顶点颜色的高洛得渲染不同,在卡通渲染技术中通常满足 2 个条件:有明显的轮廓边和明显的明暗颜色分界.卡通渲染方法可以表现三维模型的部分特征,但细节仍不够完

善.研究者们又提出了通过提取特征线建模再绘制笔画的方法,水墨画效果生成算法即是其中的一类典型的方法.已有的水墨画效果生成算法通常是首先提取出模型的特征线^[2-4],再根据特征线生成风格化的笔画效果^[5-6],或者采用多个参数模拟毛笔的绘制过程^[7-9],模拟过程较为复杂,绘制的表面特征也不够丰富.

本文提出了一种基于三维模型表面特征的水墨效果渲染算法,其与已有的方法的区别在于:无需

收稿日期:2007-03-16; 修回日期:2007-08-08. 基金项目:国家自然科学基金(60573149). 景 昊,男,1980 年生,博士研究生,主要研究方向为非真实感绘制技术、基于物理的流体模拟技术. 周秉锋,男,1963 年生,博士,教授,博士生导师,主要研究方向为几何造型、CAD/CAM、彩色图像处理、数字图像半色调、非真实感绘制技术、基于图像的绘制与建模、虚拟现实等(cczbf@pku.edu.cn).

进行特征线提取,直接通过对三维模型表面特征的定义计算顶点的渲染颜色值,从而生成近似水墨画的渲染效果.在实现渲染算法时,采用二维纹理映射的方法.采用本文算法进行水墨效果渲染,在处理三角形数量 100K 以上的大型三维模型时可以达到交互浏览的效果.

1 背景工作

在高洛得渲染等真实感渲染算法中,对高光的处理是很重要的步骤,而卡通渲染风格的生成无需考虑高光部分,可通过

$$C_i = a_g \times a_m + a_l \times a_m + (\max\{L \cdot n, 0\}) \times d_l \times d_m \quad (1)$$

来计算决定当前顶点的颜色.其中, C_i 是所求的顶点的颜色值, a_g 是全局的环境光照系数, a_l 和 d_l 是点光源的环境光和漫射光系数, a_m 和 d_m 是三维模型材质的环境光和漫射光系数, L 是点光源到模型上的顶点的向量, n 是顶点的法线向量. $L \cdot n$ 是 L 和 n 2 个向量夹角的余弦值.

卡通渲染存在着明显的明暗分界,能够表现部分模型特征,如图 1 所示.在实际进行卡通渲染时,通常采用一维纹理映射的方法.在图 1 的例子中,采用了 3 种颜色纹理进行卡通着色,一种表示模型亮的部分,另外 2 种表示模型暗的部分.模型上颜色的交界是由光线和顶点法向量夹角的余弦值 $L \cdot n$ 控制的.通过选择不同的映射纹理,我们可以通过对上述卡通渲染方法的变化绘制出各种非真实感渲染的效果,如铅笔草稿图的效果^[1].

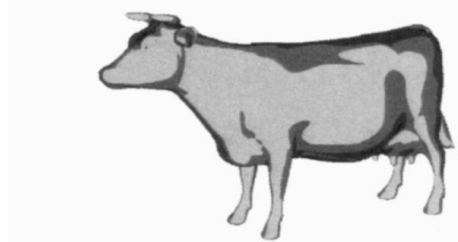


图 1 卡通渲染示例

通过提取特征线建模再绘制笔画生成风格化的非真实感绘制效果是另一种常用的风格化绘制技术^[5-11].该类算法首先提取三维模型表面的特征线段作为笔画的基本路径,通过多个参数对笔画的路径建模后进行风格化绘制.在绘制毛笔画时,根据毛笔的部分特征设置参数控制水墨笔画的生成,如力度、笔和纸的浸润程度等,或者以仿真建模的方式进行模拟,如石永鑫等提出的基于粒子系统的模

拟^[12]和张海江等提出的应用分形方法模拟笔触效果的仿真算法^[13]等,但这些算法过程均较为复杂.已有的算法处理的三维模型顶点数量较小,并且主要关注的是模型的轮廓线,因为轮廓线提取^[14-15]较为简单,并且数量不多,计算量较小.

内部特征线是三维模型的一个重要特征,包括与视点无关的脊线(或谷线)^[4]和与视点相关的启发性轮廓线^[2-3].脊线是通过模型表面主曲率(与视点无关)的计算提取的,而启发性轮廓线是通过计算模型表面的径向曲率(与视点相关)提取的,这些线段能够给予观察者更多的模型细节,尤其是对于三角形数量较大的三维模型.已有的水墨效果生成算法均未考虑对内部特征线的处理,本文提出的渲染算法可以表现模型的内部特征,通过二维纹理映射的方法简单、快速地实现水墨画绘制的效果.

2 基于径向曲率特征的水墨渲染静态图像算法

在本文的水墨画风格渲染算法中,将光源放在视点处对模型进行渲染,因此主要考虑 2 种特征对着色的影响:一种是轮廓线,通过法线向量和光线向量的点乘积获取;另一种是内部特征线.为生成水墨渲染的效果,本文算法中采用的所有颜色均为灰度颜色.

和卡通渲染一样,我们同样采用纹理映射的方法来实现水墨画风格的静态图像渲染.本文算法分为两步绘制过程.

2.1 第一步绘制

第一步通过法线向量和光线向量的点乘积绘制轮廓线的渲染效果,具体的实现过程如下:

Step 1. 定义如图 4 所示的二维纹理.将纹理环境模式设置为 GL_LINEAR .OpenGL 提供了 2 种纹理模式 $GL_NEAREST$ 和 GL_LINEAR ,采用 $GL_NEAREST$ 模式会形成如图 1 所示的明显的明暗分界,而 GL_LINEAR 模式则会形成较为平滑的明暗变化,如图 2,3 所示.本文采用的纹理中黑色部分的宽度表示了最后形成的笔画宽度.

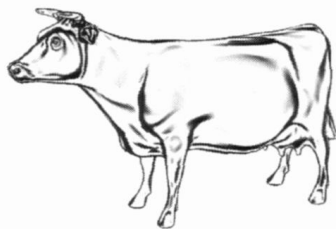


图 2 本文算法示例 1

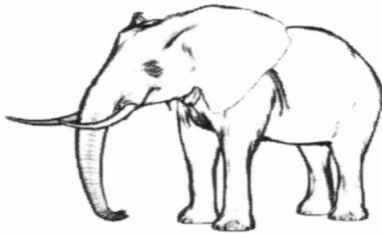


图3 本文算法示例2

Step2. 计算每个顶点的 $L \cdot n$ 值,即视线向量和法线向量夹角的余弦值. 用该值控制图4所示纹理的垂直方向纹理坐标值.

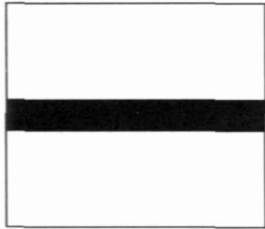


图4 本文采用的二维纹理

Step3. 关闭光照模式,打开纹理模式 (GL_TEXTURE 2D) 绘制三维模型.

第一步绘制表现了模型的轮廓边特征,如图5所示. 相比于图2,还缺乏模型表面的一些细节,这些细节将会在第二步绘制模型的时候进行渲染

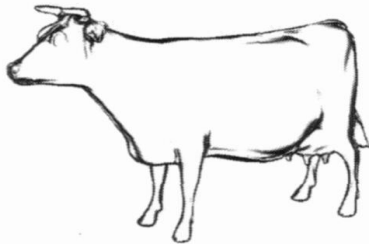


图5 第一步绘制结果

2.2 第二步绘制

顶点颜色计算. 首先定义如图4所示的二维纹理并将纹理环境模式设置为 GL-LINEAR ;然后计算径向曲率的方向导数;最后确定顶点的颜色值,以表现模型的内部特征细节.

曲面的某顶点 P 的径向曲率定义如图6所示^[2],将光线向量 v 投影到顶点的切平面上,得到向量 w , w 和顶点的法线 n 组成的平面和曲面的交线为径向曲线,该曲线在点 P 处的曲率即为曲面上 P 点的径向曲率. 径向曲率的值由欧拉公式

$$K_r(p) = K_1(p) \cos^2 \phi + K_2(p) \sin^2 \phi \quad (2)$$

得到. 其中 ϕ 是图6中的向量 w 和 P 点最大主曲率的夹角.

在第二步绘制中,计算模型上顶点 i 的颜色公式为 $C_i = a_g \times a_m + a_l \times a_m + D_w K_r(i) \times d_l \times d_m$;其

中 $D_w K_r(i)$ 是第 i 个顶点的径向曲率在 w 方向的导数,其余项的含义与式(1)相同.

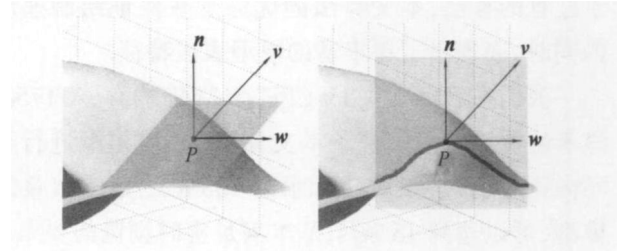


图6 径向曲率的定义

第二步绘制的具体过程如下:

Step1. 定义如图4所示的二维纹理. 将纹理环境模式设置为 GL-LINEAR .

Step2. 根据式(2)计算径向曲率 $D_w K_r$,用 $D_w K_r$ 的值控制图4纹理的垂直方向纹理坐标值.

Step3. 关闭光照模式,打开纹理模式 (GL_TEXTURE 2D) 绘制三维模型.

在第二步绘制完成之后,可以得到如图2,3和图7所示的水墨渲染效果. 本文算法在提取模型表面特征的同时,也实现了水墨笔画的渲染效果,其提取的特征应与图8所示的启发性轮廓线近似,将图3与图8相比,模型表面的特征细节均较好地表现出来.

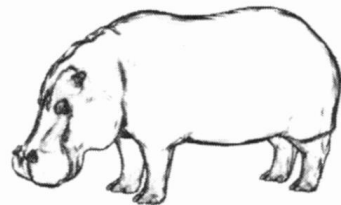


图7 本文算法示例3

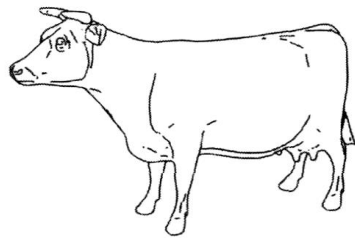


图8 启发性轮廓线提取效果

已有的基于模型表面特征的非真实感渲染算法通常分为两步:首先提取出模型的表面特征线,再对特征线进行参数化建模生成笔画路径,控制笔画的宽度、端点、方向等参数渲染生成风格化的特征笔画,以此来绘制三维模型. 其参数控制过程较为复杂,并且对于三角形数量较多、表面细节较多的三维模型,其计算速度慢. 基于计算速度的考虑,已有算法通常只考虑对三维模型的轮廓、折痕和边界等

简单特征信息进行风格化绘制,其绘制结果与图 5 所示的本文算法的第一步绘制结果近似^[5-6]. 相比于已有的算法,本文算法的优势是在保证绘制速度的同时,表现出了更丰富的模型表面特征.

我们在 P 2.0GHzCPU,512MB 内存,ATI7500 显卡的计算机上实现了本文算法. 在对模型进行交互浏览时,对于如图 2 所示的 100K 三角形数量的模型,可以达到 15 帧/s,基本满足实时浏览的要求.

3 基于视点无关的曲率特征的水墨渲染

在基于径向曲率的渲染算法实现中,计算量的主要瓶颈在于径向曲率的计算. 在绘制每一静态帧时,径向曲率及其导数的计算时间占整个绘制时间的 50% ~ 60% 左右. 由于径向曲率是与视点相关的,因此在浏览模型时,每次视点变换必须要重新计算径向曲率,导致浏览时的帧速率降低;基于径向曲率的渲染虽然基本满足实时浏览的要求,但对于三角形数量更大的模型则比较勉强. 因此,本文将基于径向曲率的渲染算法扩展到基于视点无关的曲率特征(包括平均曲率,高斯曲率和主曲率)的渲染算法. 在预处理步骤计算各种曲率、对模型进行浏览时无需重新估算曲率,以提高浏览时的绘制速度,这同样能表现模型的表面特征,绘制结果表现的特征与模型上的脊线(或谷线)近似. 图 9 所示为基于视点无关的曲率特征进行渲染的结果.

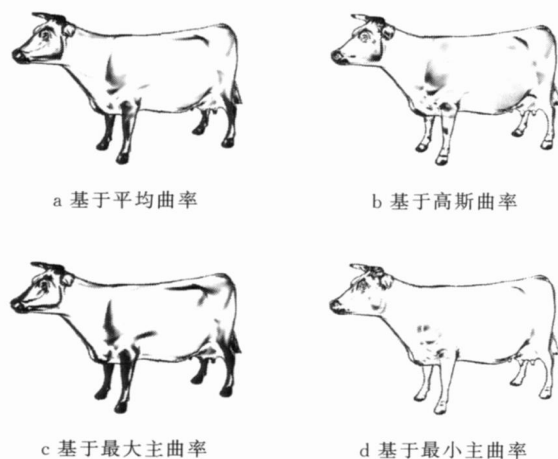


图 9 基于视点无关曲率的渲染效果图 1

3.1 基于平均曲率特征的水墨效果渲染

基于平均曲率特征的渲染算法同样分为 2 步渲染. 第一步渲染和基于径向曲率特征的渲染算法相同. 在第二步中,计算模型上顶点 i 的颜色公式为

$$C_i = a_g \times a_m + a_l \times a_m + K_{\text{mean}}(i) \times d_l \times d_m;$$

其中 $K_{\text{mean}}(i)$ 是第 i 个顶点的平均曲率,其余项的含义与式(1)相同. 在实现时采用如图 4 所示的二维纹理,采用 $K_{\text{mean}}(i)$ 控制纹理的垂直方向坐标.

3.2 基于高斯曲率特征的水墨效果渲染

基于高斯曲率特征的渲染算法第一步同样和基于径向曲率特征的渲染算法相同. 在第二步中,计算模型上顶点 i 的颜色公式为 $C_i = a_g \times a_m + a_l \times a_m + K_g(i) \times d_l \times d_m$; 其中 $K_g(i)$ 是第 i 个顶点的高斯曲率,其余项的含义与式(1)相同. 绘制时采用 $K_g(i)$ 控制图 4 中纹理的垂直方向坐标.

3.3 基于主曲率特征的水墨效果渲染

与基于平均曲率和基于高斯曲率的渲染算法相同,在基于主曲率的渲染算法中,基于最大主曲率和最小主曲率的顶点颜色分别由 $C_i = a_g \times a_m + a_l \times a_m + K_{\text{max}}(i) \times d_l \times d_m$ 和 $C_i = a_g \times a_m + a_l \times a_m + K_{\text{min}}(i) \times d_l \times d_m$ 得到. 其中 $K_{\text{max}}(i)$ 和 $K_{\text{min}}(i)$ 分别代表第 i 点的最大主曲率和最小主曲率,绘制时分别采用 $K_{\text{max}}(i)$ 和 $K_{\text{min}}(i)$ 控制图 4 中纹理的垂直方向坐标.

如图 9 所示,采用基于视点无关曲率的渲染算法时,对于 100K 三角形数量的三维模型的交互浏览速度可以达到 25 帧/s. 而对于如图 10 所示的 150K 三角形的三维模型,交互浏览速度可以达到 20 帧/s.

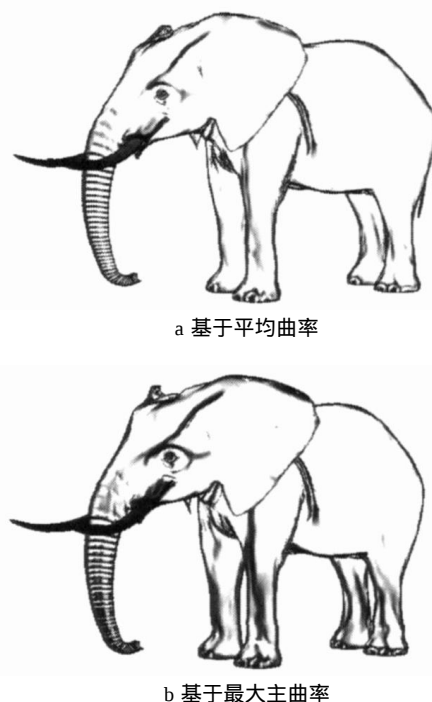


图 10 基于视点无关曲率的渲染效果图 2

通过与基于径向曲率的绘制结果比较发现,基于径向曲率的渲染算法绘制的效果能更好地表现模型的表面特征,但绘制速度较慢;而采用基于视点无

关的曲率特征的渲染算法绘制的效果虽然不如前者,但绘制速度较快。通过比较,在本文提出的基于4种不同的视点无关的曲率特征(平均曲率,高斯曲率,最大主曲率和最小主曲率)的算法中,基于平均曲率和最大主曲率的绘制效果要优于另外2种,基于这2种曲率特征的绘制表现了模型表面更多的特征,且噪声较少。图10所示为大象的三维模型采用基于平均曲率和最大主曲率的渲染效果。因此,在需要生成静态渲染图像或者对实时性要求不高的情况下,可以采用第2节中的基于径向曲率的水墨渲染算法,而在对实时性要求很高的情况下绘制大型的三维模型,则可以采用基于平均曲率或者最大主曲率的渲染算法。

4 结 论

本文在传统的卡通渲染技术的基础上,提出了一种基于模型表面特征的水墨渲染算法,通过光线与法线夹角的余弦值和径向曲率的导数计算出渲染颜色,生成水墨画风格的静态渲染效果;通过对平均曲率等与视点无关的曲率特征的计算进行渲染,以满足对大型模型的实时性绘制要求。与已有的方法相比,本文算法不需要很多的控制参数来控制笔画的生成,并且能够表达更多的模型特征细节。本文算法通过二维纹理实现,较为简单且绘制快速,对较大三角形数量的模型可以达到实时浏览的效果。

本文算法的计算瓶颈主要来自于顶点曲率的计算,特别是在基于径向曲率的渲染算法中。在未来的工作里,我们将在GPU中采用可编程的顶点处理器进行曲率和曲率导数的计算,以加快计算速度。

参 考 文 献

- [1] Lake A, Marshall C, Harris M, et al. Stylized rendering techniques for scalable real-time 3D animation[C] // Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, Annecy, 2000: 13-20
- [2] DeCarlo D, Finkelstein A, Rusinkiewicz S, et al. Suggestive contours for conveying shape[J]. ACM Transactions on Graphics, 2003, 22 (3): 848-855
- [3] Sousa MC, Prusinkiewicz P. A few good lines: suggestive drawing of 3D models[J]. Computer Graphics Forum, 2003, 22 (3): 381-390
- [4] Ohtake Y, Beliaeva A, Seidel HP. Ridge-valley lines on meshes via implicit surface fitting[J]. ACM Transactions on Graphics, 2004, 23 (3): 609-612
- [5] Yeh JW, Ouh Young M. Non-photorealistic rendering in Chinese painting of animals[J]. Journal of System Simulation, 2002, 14 (9): 1220-1224+1262
- [6] Kalnins RD, Markosian L, Meier BJ, et al. WYSIWYG NPR: drawing strokes directly on 3D models[J]. ACM Transactions on Graphics, 2002, 21 (3): 755-762
- [7] Xu Songhua, Xu Yingqing, Kang Sin g Bin g, et al. Animating Chinese paintings through stroke-based decomposition[J]. ACM Transactions on Graphics, 2006, 25 (2): 239-267
- [8] Xu S, Lau FC, Tan g F, et al. Advanced design for a realistic virtual brush[J]. Computer Graphics Forum, 2003, 22 (3): 533-542
- [9] Chu NSH, Tai CL. An efficient brush model for physically-based 3D painting[C] // Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, 2002: 413-421
- [10] Kalnins RD, Davidson PL, Lee Markosian, et al. Coherent stylized silhouettes[J]. ACM Transactions on Graphics, 2003, 22 (3): 856-861
- [11] Isenberg T, Halper N, Strothotte T. Stylizing silhouettes at interactive rates: from silhouette to gestural strokes[J]. Computer Graphics Forum, 2002, 21 (3): 249-258
- [12] Shi Yongxin, Sun Jizhou, Zhang Hai jiang, et al. Graphical simulation algorithm for Chinese ink wash drawing by particle system[J]. Journal of Computer-Aided Design & Computer Graphics, 2003, 15 (6): 667-672 (in Chinese)
(石永鑫, 孙济洲, 张海江, 等. 基于粒子系统的中国水墨画仿真算法[J]. 计算机辅助设计与图形学学报, 2003, 15 (6): 667-672)
- [13] Zhang Hai jiang, Wang Xiu jin, Sun Jizhou, et al. Fractal-based simulation of the diffusion effect of Chinese ink wash drawing[J]. Journal of Computer-Aided Design & Computer Graphics, 2004, 16 (4): 555-558 (in Chinese)
(张海江, 王秀锦, 孙济洲, 等. 应用分形仿真水墨扩散轮廓[J]. 计算机辅助设计与图形学学报, 2004, 16 (4): 555-558)
- [14] Raskar Ramesh, Cohen Michael. Image precision silhouette edges[C] // Proceedings of Symposium on Interactive 3D Graphics, Atlanta, 1999: 135-140
- [15] Markosian L, Kowalski MA, Trachtenberg SJ, et al. Real-time non-photorealistic rendering[C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, 1997: 415-420