

## 实时数字浮雕建模

周世哲<sup>1,2)</sup>, 刘利刚<sup>1,2)\*</sup>

<sup>1)</sup> (浙江大学数学系计算机图形图像研究所 杭州 310027)

<sup>2)</sup> (浙江大学 CAD & CG 国家重点实验室 杭州 310027)

(shizhezhou@zju.edu.cn)

**摘 要:** 针对现有的数字浮雕生成方法速度较慢且浮雕细节不够丰富的缺点, 提出了新的高细节实时可交互地生成数字浮雕的方法. 首先提出了一种高细节动态范围压缩函数, 并在理论上证明了它比目前的压缩函数能更好地保持细节; 然后基于该函数建立了一个实时数字浮雕建模系统, 其将建模、交互和渲染流程放到 GPU 上实现, 并利用多目标渲染提高帧缓存的利用率, 获得了实时建模速度. 系统中还设计了一个交互控制界面, 使用户能实时地调整刻画浮雕的各种参数. 实验结果证明, 该方法能快速、便捷地制作出比现有方法具有更高细节特征的数字浮雕.

**关键词:** 数字浮雕建模; 动态范围压缩; 图形处理单元; 多目标渲染

**中图法分类号:** TP391.41

## Realtime Digital Bas-Relief Modeling

Zhou Shizhe<sup>1,2)</sup> and Liu Ligang<sup>1,2)\*</sup>

<sup>1)</sup> (Institute of Computer Graphics and Image Processing, Department of Mathematics, Zhejiang University, Hangzhou 310027)

<sup>2)</sup> (State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027)

**Abstract:** Current methods for generating digital bas-relief are pretty slow and can not be used to preserve complex details in the result. This paper proposes an approach for interactively generating digital bas-relief in real time. By the method, first we present a novel dynamic range compression function which is proved to be able to better preserve details than previous method. Then a modeling system for generating bas-relief is developed based on the compression function. The system integrates all the processing including modeling, interaction and rendering into GPU and use texture computation technique of multi-targets rendering to gain a high rate of utilization of frame buffers. Therefore, the system can be used to generate bas-relief models in a real-time rate. The user can adjust parameters to obtain various bas-relief models in the system by an easy-to-use user interface. Experimental results show that the approach can be applied to efficiently generate more detailed bas-relief than previous approaches.

**Key words:** digital bas-relief modeling; dynamic range compression; GPU; multi-targets rendering

作为一门古老而成熟的艺术形式, 浮雕造型近年来被引入了计算机图形学领域.

按照厚度的不同, 浮雕可分为 2 种基本形态: 高浮雕和浅浮雕. 本文专注于浅浮雕造型, 传统制作工艺完全依赖艺术家的空间想象, 制作过程烦琐且漫

长, 而现在借助计算机和数控雕刻机床, 一般人也可轻易地进行创作. Weyrich 等<sup>[1]</sup>提出了一种基于三维场景生成数字浮雕的方法: 即对一个网格模型, 提取其观察视图的高度场, 并对这个高度场进行梯度域压缩以获取浅浮雕. 由于该算法最终归结为对

线性系统的求解,而目前的数字浮雕算法普遍需要一个初始化的过程,即在生成浮雕前必须先分解一个大型稀疏线性矩阵;在用户每一次旋转视角或是改变场景的尺寸和形状后,必须再对其进行一次求解,以获取新指定的视点的浮雕.因此在这样的框架下,创作人员无法实时地以任意角度进行浮雕建模,同时也不能方便地通过调整参数来控制浮雕的建模效果.

为了解决这个问题,本文提出了利用 GPU 加速实时获取数字浮雕的方法.该方法以获取最大的并行度为理念,将整个运算过程放在 GPU 和其附带的帧缓存内进行,以获得实时速度.通过设计一个高细节的动态范围压缩函数,获得了较以前方法细节更明显的浮雕;在实时交互上,本文采用了 Drop\_to\_Best(DtB)的技术,使用户能更流畅地调整建模角度.但在某些角度,模型生成的浮雕可能带有瑕疵.为此,我们设计了多个参数调整界面,使得用户能对这些不满意的局部进行修改;整个修改过程也是实时地进行.本文系统在较大规模,如  $1000 \times 1000$  网格顶点的浮雕尺寸,也能进行实时建模,从而能方便地获取更为细腻的浮雕效果.

## 1 相关工作

浮雕建模的过程应该使得浮雕尽量保持原有模型的细节和外观,模型高度场的动态压缩方法是其中关键的技术;因此浮雕建模的研究受到了高动态范围(high dynamic range, HDR)图像的压缩技术的启发. Fattal 等<sup>[2]</sup>提出了在图像梯度域利用对数衰减函数来进行色彩强度缩减的思想,对数压缩的特点是对越大的自变量,其函数值相对于自变量的衰减越快,而较小的值则相对得到保护<sup>[3]</sup>. 同样地,将这样的函数用于网格高度场压缩,则能在光滑地压缩外形的同时维持细节,文献[1]采用了这种对数函数族来压缩高度场的梯度. Song 等<sup>[4]</sup>的算法也使用了梯度域形状表示,不同的是他们结合了网格显著特征<sup>[5]</sup>定义和用于处理网格表面信号的非尖锐遮罩<sup>[6]</sup>技术来度量网格的视觉特征量,然后重建出保持原网格视觉特点和几何细节的浅浮雕模型.

近年来 GPU 通用计算的研究得到开展,一些基于 GPU 的线性代数和偏微分方程的数值解法被实现. 如 Krüger 等<sup>[7]</sup>实现了基于 GPU 求解大型稀疏线性方程组的高斯-赛德耳迭代法和共轭梯度法; Cuntz 等<sup>[8]</sup>在 GPU 上利用粒子水平集方法可视化

三维流场; Bolz 等<sup>[9]</sup>则实现了稀疏非结构化矩阵的共轭梯度法和多重网格法. Zhou 等<sup>[10]</sup>在 GPU 上实现了 W 型多重网格方法,并将它嵌入实时流体模拟,提高了模拟速度. 在纹理数据管理方面,目前由 NVIDIA 公司推出的计算设施统一架构(compute unified device architecture, CUDA)技术能自动地管理显存内数据,如 Zhou 等<sup>[11]</sup>利用 CUDA 来生成  $kD$ -Tree 并将之用于实时光照. 而本文系统并没有采用 CUDA,而是直接使用了 OpenGL 的帧缓存扩展来完成纹理数据管理.

## 2 浅浮雕建模原理

在数字化浅浮雕建模方面,目前较为成熟的方法是使用梯度域非线性压缩重建法<sup>[1]</sup>. 其基本思想是动态压缩原网格模型不同级别的细节来生成依附于一平面上的浮雕模型,其结果如图 1 所示. 现将其算法做如下描述.

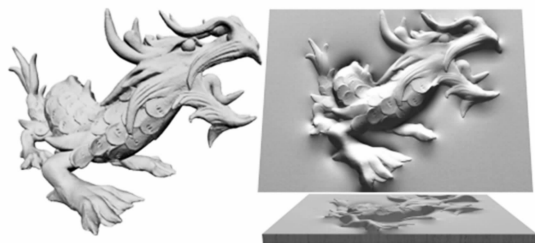


图1 从三维模型生成浅浮雕模型

给定一三维场景(可以是单个模型或若干个的组合),可得到带有深度信息的二维观察视图,该深度信息可定义了一张坐落在屏幕坐标域  $\pi$ (即浮雕附着的零高度平面)上的高度场曲面,记为  $h(x, y)$ . 计算其梯度向量  $\mathbf{g}(x, y) = \nabla h(x, y)$ . 注意在每一点都有梯度方向和模长,即  $\mathbf{g}(x, y) = \|\mathbf{g}\| \mathbf{d}(x, y)$ .

保持梯度方向  $\mathbf{d}(x, y)$  不变,在模长  $\|\mathbf{g}\|$  上压缩,  $\mathbf{g}'(x, y) = \|\mathbf{g}'\| \mathbf{d}(x, y) = C(\|\mathbf{g}\|) \mathbf{d}(x, y)$ .

压缩函数  $C$  是至关重要的,它直接影响浮雕的轮廓线、局部细节和浮雕的厚度. 较好的一种压缩函数是采用对数压缩<sup>[1-2]</sup>,文献[1]中采用的函数为

$$C(x; c) = \frac{1}{c} \ln(1 + cx) \quad (1)$$

其中参数  $c$  可用来控制压缩程度,  $c$  越小,压缩率越小,浮雕的细节也越突出. 但通过实验发现,当减小至  $c=0.2$  后,细节增强已不明显;故本文以此参数下的结果作为比照对象.

得到了压缩后的梯度场之后,就可以对其积分

来重建新的高度场. 具体方法导出为一个 Poisson 方程

$$\Delta r(x,y) = \nabla \cdot \mathbf{g}' \tag{2}$$

其中  $r(x,y)$  即新生成的浮雕高度场. 该方程的解等价于使  $r(x,y)$  的梯度和压缩后梯度  $\mathbf{g}'$  相等的变分最小问题

$$r(x,y) = \arg \min_h \iint \|\nabla h(x,y) - \mathbf{g}'(x,y)\|^2 dx dy \tag{3}$$

的解, 所以其解能最大程度地保持压缩后模型曲面的几何特点. 如果选取保持细节的压缩函数, 则生成的浮雕也能很好地维持原模型视觉特征和局部细节.

3 高细节的动态范围压缩函数

梯度压缩函数对浮雕的厚度和细节起着决定作用. 研究发现, 一个好的压缩函数  $f(x)$  应满足条件

$$\begin{cases} f(x) \geq 0, f'(x) > 0, f''(x) < 0, \forall x \geq 0 \\ \frac{f(x_1)}{x_1} > \frac{f(x_2)}{x_2}, \text{ if } x_2 > x_1. \end{cases} \tag{4}$$

即在非负区间,  $f(x)$  是非负的单调递增的上凸函数; 同时  $f(x)$  还应动态压缩各个区间段. 本文将满足式(4)的函数  $f(x)$  定义一个压缩率为

$$R_{\text{cpr}}(f(x)) = \frac{f(x)}{x}, \forall x > 0.$$

经过仔细分析, 本文选择分段函数

$$a(x; \alpha, \beta) = \begin{cases} \alpha \arctan(\beta x), & 0 \leq x \leq \phi_{\text{thred}} \\ 0, & \text{else.} \end{cases} \tag{5}$$

作为压缩函数. 现对式(1)(5)表示的两族函数的压缩特性进行分析和比较.

可以证明(详见附录), 当参数  $\alpha\beta > 1$  时, 两族函数在正实数区间存在唯一的交点(如图 2 中的黑点所示, 其横坐标标记为  $x=i$ ); 且交点左(右)侧  $a(x)$  的压缩率小(大)于对数压缩, 即

$$\begin{cases} R_{\text{cpr}}(a(x)) \leq R_{\text{cpr}}(C(x)), & x \in (0, i] \\ R_{\text{cpr}}(a(x)) > R_{\text{cpr}}(C(x)), & \text{else.} \end{cases} \tag{6}$$

注意该结论对这两族函数是普遍成立的, 并不受各自参数选取的影响. 该结论保证: 无论  $C(x; c)$  中的参数  $c$  如何取值, 式(5)通过调节  $\alpha, \beta$  都可以到达浮雕细节比式(1)更强的效果. 图 2 给出了式(5)在  $\alpha=1.5, \beta=0.8$  时的图像(图中较细的线), 以及  $\alpha=0.75, \beta=1.8$  时图像(图中较粗的线)与函数  $\ln(1+x)$ (图中虚线)的图像对比. 为图示清晰, 这里

我们仅以  $C(x) = \ln(1+x)$  为函数族  $C(x; c)$  的代表.

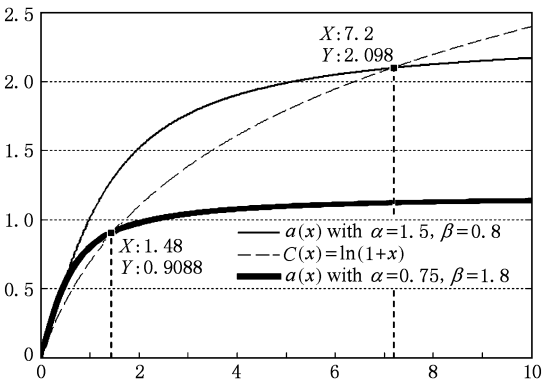


图 2 函数  $a(x)$  与  $\ln(1+x)$  的函数曲线

由图 2 可清楚地看到, 交点(图 2 中黑点)将我们的函数划分为低压缩率段(交点至原点部分)和高压缩率段(交点以右部分). 这说明, 函数  $a(x)$  在有效压缩梯度的同时能更好地保存并强调细节. 采用不同压缩函数所建浮雕结果对比如图 3 所示.

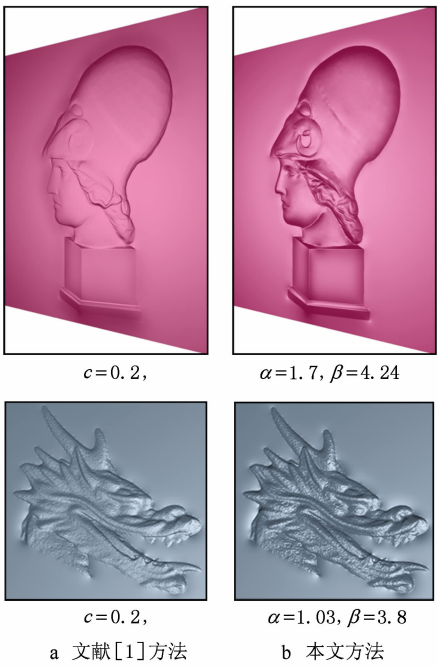


图 3 浮雕结果比较

4 基于 GPU 实时建模

以上提到的算法核心是 Poisson 方程, 若以差分格式离散偏微分算子, 将得到一个大型稀疏线性系统; 传统方法采用矩阵分解法求解, 则需要一个初始化的过程. 该过程耗时与  $\pi$  的尺寸相关, 对尺寸为

100×100~800×800 像素的浮雕建模耗时一般在 10 s 到 1 min. 若改变  $\pi$  的形状或者缩放其尺寸,都将改变该系统,需要再进行一次初始化.

为了达到实时建模,我们在 GPU 上进行 Jacobi 迭代来求解式(2),也可使用加权的 Jacobi 迭代<sup>[7]</sup>或多重网格法<sup>[10]</sup>. 在 GPU 上每一次迭代归结为运行五次纹理采样函数,效率非常高,可在短时间内运行许多次. 迭代至收敛的次数与  $\pi$  的尺寸相关,如表 1 所示. 使用 Jacobi 方法对迭代初值(iterative starting value, ISV)的选取十分敏感,实验发现,若以原始高度场  $h$  为 ISV,则无法收敛;而采用  $0.01h$  可迅速收敛. 其原因是浅浮雕的厚度较小,因此厚度与该估计值是最接近的.

表 1 实时浮雕建模速度

浮雕尺寸 /像素	平均迭代次数	平均帧率/帧/s	
		不使用 DtB	使用 DtB
1000×1000	650	10	18
500×500	180	17	32

本文系统采用 GPU 帧缓存扩展的多纹理计算技术,在后缓存 BackFB 中后台计算浮雕并在前缓存 FrontFB 中实时显示,如图 4 所示. 在从 OpenGL 绘制场景中提取深度后,本文采用多目标渲染(multi-targets rendering, MTR)的技术,利用一个像素程序同时计算出高度场和 ISV,将其保存到 2 幅纹理中;然后继续利用 MTR 技术和多个像素程序依次计算压缩梯度和压缩散度准备迭代(虚线箭头). 为了进一步加速,我们还用一个像素程序计算新生成的浮雕的法向,这样能再缩短每一帧浮雕的绘制用时. 注意,本文将后台数据纹理全存放在 BackFB 中,以最大程度地避免帧缓存切换耗时.

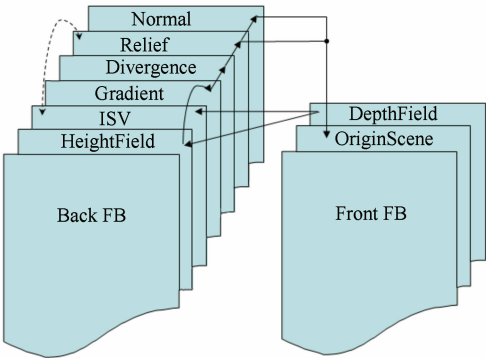


图 4 算法流程与数据纹理布局

5 交互方式与细节控制

本文系统为用户提供 2 类实时交互:场景视角旋转和浮雕细节参数调整. 在用户实时交互阶段,采用 DtB 的技术来进一步提高系统运行的流畅程度,如表 1 所示. 具体做法是在用户改变场景视角时,系统自动降低迭代次数;用户松开鼠标交互确定之后,立刻增加迭代次数以获取最精确解. 这样,用户在一次交互期内能流畅作业,并随时获得最佳浮雕结果.

本文系统支持实时调整各种参数. 用户可以通过调节函数式(5)中的 2 个实参数  $\alpha, \beta$  来控制浮雕的厚薄与细节的强弱;其中  $\alpha$  为前置系数. 将式(5)代入式(2)可得

$$\Delta r(x,y) = \nabla \cdot \mathbf{g}' = \alpha(\nabla \cdot \arctan(\beta \| \mathbf{g} \|) \mathbf{d}(x,y)).$$

由此可知  $\alpha$  可以线性地缩放式(2)的解,因此  $\alpha$  可以对浮雕的整体厚度进行线性控制. 在实验中发现,若生成边长为 25 cm 的正方形浮雕,取  $\alpha \in [1.0, 3.8]$  可以获得最大处厚度在 0.26~1.0 cm 之间的浅浮雕. 图 5 所示为取不同  $\alpha$  时的浮雕厚度效果,其中上图  $\alpha=1.0$ ,下图  $\alpha=2.5$ .

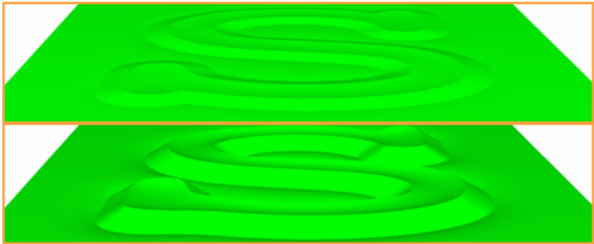


图 5 通过  $\alpha$  调节浮雕厚度

在实验中还发现,增大  $\alpha$  和  $\beta$  都能增强浮雕上细节效果的凸显程度. 但是如前所述,  $\alpha$  的增大会使浮雕厚度增加,选取太大的  $\alpha$  会使得生成的浮雕厚度过大,有违浅浮雕的特点;而通过调节  $\beta$  来控制细节的凸显程度则能避免该问题. 图 6 所示为调节  $\beta$  对浮雕细节程度的控制效果( $\alpha$  维持不变),其中左图中  $\beta=1.24$ ,右图中  $\beta=6.82$ . 因此,在本文建模系统中,鼓励用户以调节  $\beta$  为主、调节  $\alpha$  为辅的方式来控制浮雕细节.

此外,在模型高度场中存在某些高度落差较大、但小于  $\phi_{\text{thred}}$  的区域,在浮雕中会形成不连续的竖直突出效果(文献[1]结果中也有此现象),这是由于选取了不恰当的  $\phi_{\text{thred}}$  而无法正确识别浮雕边界的

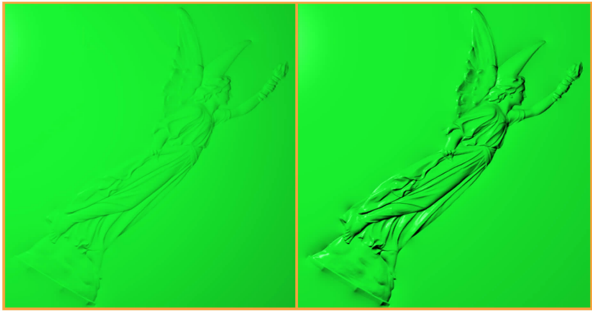


图 6 通过  $\beta$  调节浮雕细节

缘故. 本系统中用户可以通过实时调节  $\phi_{\text{thred}}$  消除这种现象, 而仅轻微地影响其余的部分, 如图 7 所示. 注意, 图中原本不自然的“高墙”部分(红色标识区域)在调节了  $\phi_{\text{thred}}$  后得到了平滑.

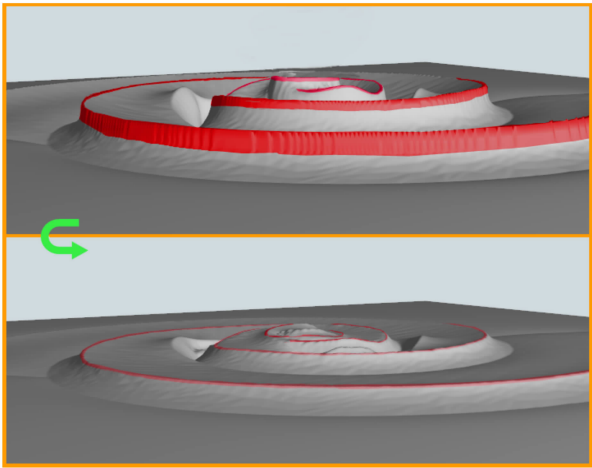


图 7 调整阈值  $\phi_{\text{thred}}$  以消除瑕疵(红色部分)

6 结论与展望

本文介绍了采用 GPU 加速技术构建的实时数字浮雕建模系统; 还研究了影响建模效果的关键因素, 进而提出一种新的高细节的动态范围压缩函数以提高浮雕细节度. 实验证明, 该系统不仅能使艺术创作者从任意三维模型场景中实时快速地制作大尺寸浅浮雕, 且建模效果好于传统的方法. 图 8 中显示了该系统从 2 个模型中制作尺寸为  $1000 \times 1000$  像素的浅浮雕(上), 及它们在该角度建模效果的侧视图(下), 可以看到所形成的浮雕具有较薄的厚度却能提供较好的视觉效果. 图 9 显示了系统对同一模型从不同视点进行建模的效果.

实时浮雕的创建过程实际上是一种浮雕风格的

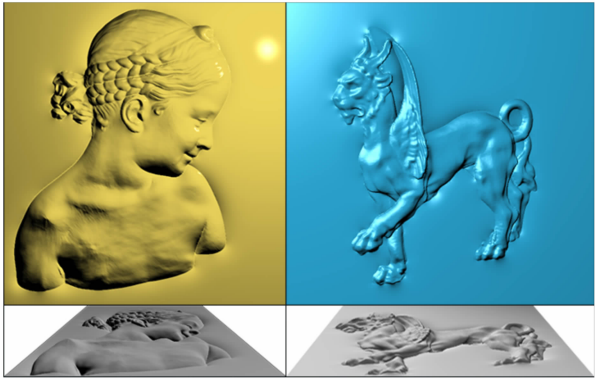


图 8 更多的浮雕结果

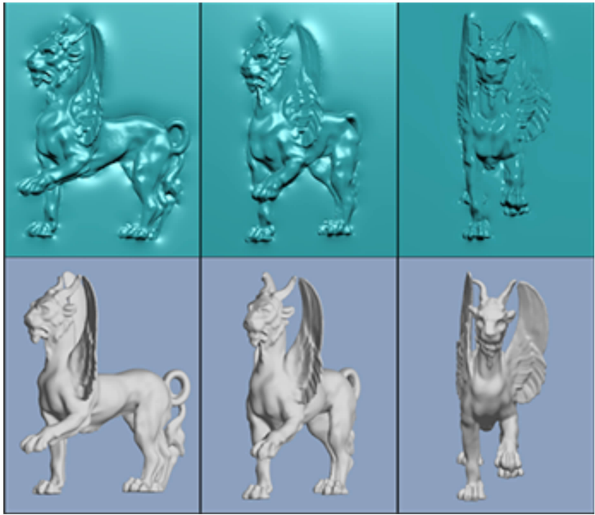


图 9 从不同视点的建模效果

三维动画, 这种动画形式也是以前没有出现过的. 未来我们将继续该方面的研究及其在浮雕动画和网格绘制方面的应用.

参考文献 (References):

[1] Weyrich T, Deng J, Barnes C, *et al.* Digital bas-relief from 3D scenes [J]. *ACM Transactions on Graphics*, 2007, 26 (3): 32-38

[2] Fattal R, Lischinski D, Werman M. Gradient domain high dynamic range compression [J]. *ACM Transactions on Graphics*, 2002, 21(3): 249-256

[3] Michael A. A tone mapping algorithm for high contrast images [C] // *Proceedings of the 13th Eurographics Workshop on Rendering*, Pisa, 2002: 145-156

[4] Song W H, Belyaev A, Seidel H. Automatic generation of bas-relief from 3D shapes [C] // *Proceedings of the IEEE International Conference on Shape Modeling and Applications*, Washington D C, 2007: 211-214

- [5] Lee C H, Varshney A, Jacobs D W. Mesh saliency [J]. ACM Transactions on Graphics, 2005, 24(3): 659-666
- [6] Luft T, Colditz C, Deussen O. Image enhancement by unsharp masking the depth buffer [J]. ACM Transactions on Graphics, 2006, 25(3): 1206-1213
- [7] Krüger J, Westermann R. Linear algebra operators for GPU implementation of numerical algorithms [J]. ACM Transactions on Graphics, 2003, 22(3): 908-916
- [8] Cuntz N, Kolb A, Strzodka R, *et al.* Particle level set advection for the interactive visualization of unsteady 3D flow [C] //Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization, Eindhoven, 2008: 719-726
- [9] Bolz J, Farmer I, Grinspun E, *et al.* Sparse matrix solvers on the GPU: Conjugate gradients and multigrid [J]. ACM Transactions on Graphics, 2003, 22(3): 917-924
- [10] Zhou Shizhe, Man Jiaju. Realtime fluid simulation based on multigrid method [J]. Journal of Computer-Aided Design & Computer Graphics, 2007, 19(7): 935-941 (in Chinese)  
(周世哲, 满家巨. 基于多重网格方法的实时流体模拟[J]. 计算机辅助设计与图形学学报, 2007, 19(7): 935-941)
- [11] Zhou K, Hou Q M, Wang R, *et al.* Real-time *k*D-tree construction on graphics hardware [J]. ACM Transactions on Graphics, 2008, 21(5): 231-241

附录

证明. 当参数  $\alpha\beta>1$  时,2 族函数式(1)(5)在正实数区间有且仅有一个的交点.

式(1)和(5)的导函数分别为

$$a'(x)=\frac{\alpha\beta}{(1+\beta^2x^2)}$$
$$C'(x)=\frac{1}{(1+cx)}$$

有

$$a'(x)-C'(x)=\frac{-(\beta^2x^2-c\alpha\beta x-\alpha\beta+1)}{(1+\beta^2x^2)(1+cx)}$$

记

$$d(x;\alpha,\beta,c)=\beta^2x^2-c\alpha\beta x-\alpha\beta+1\tag{7}$$

式(6)是  $x$  的二次函数,由  $\beta^2>0$  及  $c\alpha\beta/\beta^2>0$  知方程  $d(x)=0$  必有正实根  $x_d$ ;再由假定  $\alpha\beta>1$ ,有  $d(0)=1-\alpha\beta<0$ ,所以当

$$x\in(0,x_d)\text{时},d(x)>0,\text{故}$$
$$a'(x)-C'(x)>0,x\in(0,x_d);$$
$$\text{又 }a(0)=C(0)=0,a'(x)>0,C'(x)>0,\text{故}$$
$$\exists x_1\in(0,x_d),$$
$$\text{使得}$$
$$a(x_1)>C(x_1)\tag{8}$$

$$\text{考虑到 }\lim_{x\rightarrow+\infty}a(x)=\frac{\alpha\pi}{2},\lim_{x\rightarrow+\infty}C(x)=+\infty,\text{故定能取到一个}$$
$$\text{充分大的 }x_2>x_d>x_1,\text{使得}$$
$$a(x_2)<C(x_2)\tag{9}$$

由式(8)(9),推出 2 族函数在区间  $(x_1,x_2)$  上必上有交点,且显然地,交点两侧 2 族函数的关系如式(6)所述. 由  $a'(x)>0,C'(x)>0$  知它们在区间  $x\in(0,+\infty)$  上都单调递增且都无拐点,因此两族函数在正数区间上有且仅有一个交点.

证毕.