

屏幕空间流体场景的真实感绘制*

王章野¹⁺, 张海涛¹, 施肖菁², 王丰金¹, 张 昆¹, 彭群生¹

¹(浙江大学 CAD & CG 国家重点实验室, 浙江 杭州 310027)

²(温州大学 物理与电子信息工程学院, 浙江 温州 325035)

Realistic Rendering of Fluid Scenes in Screen Space

WANG Zhang-Ye¹⁺, ZHANG Hai-Tao¹, SHI Xiao-Jing², WANG Feng-Jin¹, ZHANG Kun¹,
PENG Qun-Sheng¹

¹(State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China)

²(College of Physics & Electronic Information Engineering, Wenzhou University, Wenzhou 325035, China)

+ Corresponding author: E-mail: zywang@cad.zju.edu.cn

Wang ZY, Zhang HT, Shi XJ, Wang FJ, Zhang K, Peng QS. Realistic rendering of fluid scenes in screen space. *Journal of Software*, 2012, 23(Suppl. (2)): 138-148 (in Chinese). <http://www.jos.org.cn/1000-9825/12033.htm>

Abstract: The study presents a new realistic rendering algorithm for a fluid scene with SPH (Smoothed Particle Hydrodynamics) in a screen space. First, the paper proposes an anisotropic based surface smoothing model: The study adopts the Laplacian Smoothing to acquire relative smooth particle position distribution in 3D space. Next an anisotropic modal is used to produce initial fluid surface. Last, a bilateral filtering depth map, we got the final desired smoothing fluid surface. Next, the study present a finer fluid surface illumination model to improve fluid scenes rendering, also using curvature based model to add realistic details of fluid. The study realizes that all these are in GPU. Compared with other methods, the approach can greatly improve the realism of fluid scenes rendering in screen space without losing the real-time performance, and could be found great potential applications in the domains such as real-time simulation and game, etc.

Key words: realistic rendering; fluid scenes; screen space real-time rendering; anisotropic modeling; SPH

摘 要: 现有屏幕空间流体场景的实时绘制方法存在着表面提取不光滑、光照不自然及缺乏真实感细节等缺陷。提出了一种屏幕空间流体场景真实感绘制算法。该算法基于 SPH(光滑粒子动力学)框架。首先提出了一种基于各向异性的表面光滑模型:采用 Laplacian 平滑方法对粒子空间位置进行初步平滑处理;通过各向异性的流体建模进行进一步地平滑处理;再通过双边滤波方法对深度图进行后期平滑,得到了较为理想的平滑流体表面。为了改进流体场景的光照效果,提出了一种精细的流体表面光照模型,该模型不仅考虑环境光的影响,还考虑了光源的漫反射及高光分量的定量影响。为了增加流体场景的真实感细节,提出了一个基于曲率的高效简单的浪花生成模型。最后采用 GPU 加速技术,实现了卷浪及海啸巨浪冲击海滩等复杂流体景象的实时绘制。相比现有算法,该方法能在不失实时绘制性能的前提下,极大地改进屏幕空间流体场景绘制的真实感,在实时仿真及游戏设计等领域有着重大的应用潜力。

关键词: 真实感绘制;流体场景;屏幕空间实时绘制;各向异性建模;光滑粒子动力学

* 基金项目: 国家自然科学基金(60833007, 60970075, 61272302); 国家重点基础研究发展计划(973)(2009CB320802)

收稿时间: 2012-05-20; 定稿时间: 2012-09-29

基于物理的流体模拟方法在自然景象的真实感模拟、工程仿真、影视特技制作及计算机动画等领域都有着重大的应用价值.但基于物理的流体模拟方法一般都存在着计算模型复杂,难以实时实施等不足,这就极大地限制了它在实时仿真及游戏设计等领域的应用.为克服这种缺陷,近年来一种基于屏幕空间的流体场景实时绘制新技术^[1,2]得到了较快的发展.与传统基于物理的流体模拟方法不同,该方法的优势在于无需构建表面网格模型,因而能对流体表面进行直接的实时绘制.但该方法目前还存在着表面提取不光滑、光照效果不自然及缺乏浪花泡沫这些真实感细节等不足之处.为克服这些不足,本文提出了一种新的屏幕空间流体场景真实感绘制算法.该算法能够在不失实时绘制性能的前提下,改进屏幕空间流体场景绘制效果的真实感.本文的贡献是:

提出了一种基于屏幕空间绘制的表面平滑模型,该模型采用各向异性的椭球模型取代球体粒子模型来获取平滑的原始深度图,从根本上解决了目前方法难以获取的平滑表面问题.

提出了一种基于菲涅尔模型的流体表面定量光照模型.该光照模型不仅考虑了环境光的影响,而且考虑了光源漫反射及高光反射分量的定量影响,从而改进了目前光照模型中的光照的不真实感.

提出了一个基于曲率的简单但高效的浪花生成模型,并在 GPU 构架下设计加速算法,实现了屏幕空间流体场景的真实感实时绘制.

本文第 1 节介绍相关工作.第 2 节阐述基于各向异性的表面平滑建模方法.第 3 节描述了屏幕空间流体场景真实感绘制的实现,包括改进的流体场景精细光照模型、基于曲率的浪花生成及 GPU 加速绘制框架等.第 4 节呈现了算法的绘制结果及分析讨论最后是本文工作的总结与展望.

1 相关工作

在计算机图形学领域,基于物理的流体模拟方法是自然景象真实感模拟一种有效的方法.目前,基于物理的流体模拟方法主要有基于网格的欧拉(Euler)方法和基于粒子的拉格朗日(Lagrangian)方法这两类.这二者目前都已经通过离线绘制方式成功地用于高清电影特技的制作和其他商业软件.若要将基于物理的流体模拟方法满足在线实时需求(例如:游戏),一种可能的方法是降低网格分辨率(欧拉法)或是降低粒子数量(拉格朗日法),但往往需要降低几个数量级才能达到实时要求,这样不可避免地会导致画面质量变得很差.

欧拉方法采用网格来离散流体所在的空间,能很好地描述流体的速度、压强等属性,具有差分运算方便、数值精度高等优点,但当场景网格分辨率增长率大时,其计算量将急剧上升.欧拉法的另一个不足是在三维空间中,难以很好地表现丰富的流体细节.为此,越来越多的计算机图形学研究者将研究兴趣转向基于粒子的拉格朗日法的流体模拟,尤其是 SPH(smoothed particle hydrodynamics,即光滑粒子动力学)方法上,因其可以模拟更多的流体细节(泡沫、水花等)和处理更复杂的流体表面.另一原因是基于粒子的模拟方法的成果更易集成到现有基于物理的仿真系统中.

1995 年,Stare 等人首次采用 SPH 方法成功模拟了火焰及气体的扩散现象^[3],2003 年,Muller 等人采用 SPH 方法实现了交互速度级的流体模拟^[4].此后有不少研究利用 SPH 流体模拟方法来模拟水、烟、气泡、水花飞溅等效果^[5,6].但这些工作都没有涉及到算法加速及简化问题.当粒子数量急增时,这些方法都无法解决呈几何倍数增长的计算量问题.1999 年 Desbrun 等人首次引入了自适应采样的粒子系统框架,开始着手于解决粒子规模和丰富画面细节的平衡问题^[7].Adams 等人通过引入局部粒子特性函数将这个自适应框架用于水体的模拟^[8],但该算法缺乏并行性,难以在 GPU 中并行运算以达到实时模拟运算.2009 年,延诃等人提出非均匀粒子采样模型来实现流体的实时模拟^[9],该模型在非均匀粒子系统中引入了广义的距离场函数,不清同时考虑了场景几何、物理及邻域粒子密度等因素的影响,能够有效地减少流体模拟时的粒子数,同时不失场景的真实感.在此基础上,何猷等人考虑了自适应 SPH 粒子系统流体模拟时与复杂边界的实时交互作用^[10].

虽然上述工作促进了基于物理 SPH 流体模拟研究的进展,但对于这类方法,仍存在一个亟待解决的问题,那就是粒子表面的提取.Marching Cube 方法^[11]是目前用于提取标量场等值面的最流行的方法,在流体模拟中的应用是用于提取气体-液体交界面.在基于粒子的拉格朗日方法中,通常利用对称的各向同性的核函数,从大量粒子中生成质量场,然后提取等值面,最终利用 Marching Cube 方法生成三角网格.

Marching Cube 方法尽管被广泛应用,但它仍然存在不少缺点:首先,该方法是视点无关的,因此整个流体场景都会被提取成三角网格,而其中很多三角面片对视点来说是不可见的,因此带来了运算上的浪费;其次,Marching Cube 是基于固定网格的,这并不适用于大范围自由流动的流体场景;最后,Marching Cube 运算代价相对高昂,为 GPU 带来额外的负担,因此不能充分利用 GPU 的图形硬件加速功能来实现较高分辨率网格实时流体绘制.为改进 Marching Cube 方法,Rosenberg 等人在粒子等值面提取中对 Marching Cube 进行了优化,在 3000 个粒子规模的情形下达到了实时拟和绘制^[12],但由于缺乏足够多的粒子数,其表面绘制结果凹凸不平,质量较差. Williams 等人^[13]提出了一种称为 Marching Tiles 的新的表面提取方法,它允许网格的光滑处理以及设置约束条件,可以生成光滑的表面,但其速度离实时流体绘制仍有较远的距离.

为此,近几年开始有些学者将目光渐渐转向其他的流体绘制方法,比如屏幕空间绘制技术.屏幕空间绘制技术不是新近几年才出现的新兴技术,它在过去的几十年中已经得到了广泛的研究,其中最著名的当属光线跟踪(Raytracing)算法^[14].目前的流体模拟中,较为流行的作法是基于 SPH 框架得到模拟的结果,再将场景导入基于 Raytracing 算法的 POV-Ray 绘制引擎^[15],经 Marching Cube 提取流体表面,由此得到一帧较高绘制质量的画面,将序列画帧合到一起,就得到了动态的流体场景.但由于 Raytracing 算法的耗时性,阻碍了其实时应用.

2004 年, Johanson 等人^[16]采用投影网格(Projected Grid)将高度场的水体映射到,并只对视点内的场景进行模拟和绘制,从而实现了无限范围的海洋场景的实时绘制.但该方法不适用于基于物理的流体模拟.2007 年, MÜLLER 等人^[1]提出了一种屏幕空间网格的新方法.该方法首先将粒子深度记录到深度图中,再对深度图用二项分布滤波进行平滑处理,从深度图中构建起屏幕空间的网格模型.再转换到三维空间进行遮挡剔除、反射击、折射击及其他着色效应的计算.该方法的优势在于只产生可见、与视点相关的流体表面.但该方法在构建网格模型时易丢失如流体厚度等信息,故最终绘制结果难令人满意;且该方法采用简单的二项式分布滤波,易造成流体边界的扩散.2009 年, Laan 等人^[2]提出了一种不需构造网格模型,而直接绘制流体的方法.该方法通过使深度图的曲率最小化,部分解决了直接绘制粒子球体带来的液体表面凹凸起伏的问题,并通过获取流体的厚度,使绘制的结果能反映出流体底部光照的贡献,并达到实时绘制的结果.但该算法的流体表面整体绘制的结果仍显得不够平滑,且光照效果不够真实,缺乏如浪花泡沫等真实感细节.

综上所述,虽然目前基于屏幕空间的方法能将基于物理的流体场景绘制达到了实时的效果,但仍存在场景绘制明显真实感不足的缺陷.究其原因,是由于现有模型的全向异性机制,即将粒子以球体形式直接绘制到深度图中,导致原始数据(凹凸的深度图)与最终目标(平整的水面)偏差过大而造成不平整的表面效果;而在光照模型中主要只考虑了环境光的影响,而未细致地考虑光源漫反射及高光反射分量的定量建模.

要真实感地实时绘制出屏幕空间的流体场景,必须充分考虑粒子密度分布全向异性的特点;而光照模型亦不能过于简化;流体的细节特点也要考虑.为此,本文提出了一种屏幕空间流体场景的真实感绘制新算法,该算法基于 SPH 框架,引入全向异性机制对流体表面进行平滑建模;基于菲涅尔模型对流体光照模型进行细致定量地建模,并建立了一种简易高效的基于曲率的浪花生成模型.从而使屏幕空间的流体场景绘制在不失实时性的前提下,极大地提高了其绘制的真实感.下面我们将对此进行详细讨论.

2 基于全向异性的表面平滑建模

本节将讨论基于全向异性的表面平滑建模的方法.首先阐述基于粒子位置分布的初步平滑建模;接着介绍全向异性的流体表面构建,最后描述采用双边滤波法对深度图的最终平滑实现.

2.1 基于粒子位置分布的初步平滑建模

现有屏幕空间的流体绘制方法,将球体来代表流体粒子,导致了凹凸不平的表面绘制结果.其主要有两个成因.其一是粒子的位置参差不齐,导致很难构建一个完全平滑的表面;其二是流体表面周围粒子的密度分布不适合用简单的球体来表现.本节针对第 1 个成因,将采用拉普拉斯平滑方法(Laplacian smoothing)^[17]进行改进.第 2 个成因的改进将在下一节讨论.

为了解决流体粒子位置不规则问题,我们对粒子位置进行一次初步平滑处理,该过程实质上是对三维位置

数据的降噪处理,即在三维空间中采用拉普拉斯平滑处理,以有效地降低了粒子位置上的噪声:

$$\bar{x}_i = (1 - \lambda)x_i + \lambda \sum_j w_{ij} x_j / \sum_j w_{ij} \quad (1)$$

式中 \bar{x}_i 为原粒子位置 x_i 经平滑处理后的新位置; w_{ij} 是一个关于粒子 i 和粒子 j 的各向同性权重函数,其定义为

$$w_{ij} = \begin{cases} 1 - (\|x_i - x_j\|/r_i)^3, & \|x_i - x_j\| < r_i \\ 0, & \|x_i - x_j\| \geq r_i \end{cases} \quad (2)$$

式(1)、式(2)表明,在粒子 i 的半径 r_i 范围以内的所有邻域粒子都对粒子 i 的新位置 \bar{x}_i 产生贡献,而 r_i 半径之外则不受影响.式(1)中的 λ ,在通常拉普拉斯平滑中是一个满足 $0 < \lambda < 1$ 的常数.但我们考虑到邻域粒子密度大小的影响,故将 λ 值定义为

$$\lambda = \begin{cases} c \times \sum_j w_{ij}, & c \times \sum_j w_{ij} < 1 \\ 0, & c \times \sum_j w_{ij} \geq 1 \end{cases} \quad (3)$$

式(3)中 c 是可调的参数.绘制时通过调节 c 值大小,我们可控制拉普拉斯平滑处理的程度.图 1 给出了采用拉普拉斯平滑处理前后粒子分布的对比图.从中可看到较为明显的初步平滑改进效果.

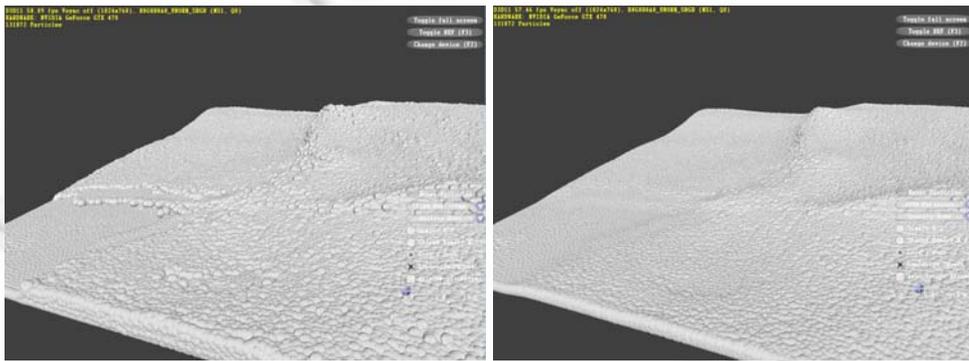


Fig.1 Comparison before(left) and after(right) Laplace smoothing

图 1 粒子位置分布经拉普拉斯平滑初步处理前(左图)、后(右图)的效果对比

经过以上初步平滑处理之后,我们再来获取流体表面的深度图.其步骤为:先将绘制目标初始化为背景深度值;然后开启深度测试下绘制每个粒子,只有当新的深度值比当前像素深度值更小时,才进行深度值覆盖,并记录下视点空间的像素深度值.在所有粒子历遍后,深度图上就保留了每个像素上最接近相机视点的粒子深度值.这就为之后的场景绘制打好了基础.

2.2 各向异性的流体表面构建

上一节我们已解决了导致屏幕空间的流体表面绘制不平滑的第一个成因.对于第 2 个成因,即用球体难以有效地表现流体表面周围粒子的密度分布问题,我们将采用各向异性建模的思想加以解决.由于流体(本文指的是液体,下同)表面具有张力,液滴边界会趋向于融合,并向周围液面扩散.因此从物理角度看,粒子对周围的密度贡献不是各向同性,而应该是各向异性的.当粒子处于液体表面时,粒子的密度贡献应该呈椭球形分布,即:其长轴和次长轴平行于液体表面,短轴垂直于液体表面.传统 SPH 流体模拟中普遍采用各向同性核函数,其缺点在于忽视了流体表面张力的各向异性等固有属性.因此,在 SPH 模拟中引入各向异性核函数,将能使流体模拟和绘制结果更加符合物理规律.

Liu 等人^[18]在 SPH 框架下采用各向异性核函数来模拟材料的形变,使得结果更加真实.Dinh 等人^[19]应用各

向异性核函数对体素数据进行表面重建.Yu 等人^[20]采用各向异性核函数定义粒子密度分布,以提取等值面.他们以主成分分析 PCA 方法获取各向异性核的轴方向和长度来构建密度场并提取等值面,获得了较为满意的平滑结果.但该文以 Marching Cube 提取密度场的等值面并构建网格,在粒子的密度分布从各向同性核变换到各向异性时,只简单设置周围粒子数作为阈值条件,这样就导致粒子密度分布在阈值上下有着视觉突变的问题.本节提出了一种改进的各向异性建模方法,以构建更加平滑的流体表面效果.

我们认为在流体(液体)模拟中,每个液体微团应根据其所处位置和状态的不同而具有不同的密度分布.比如液体内部的微团,在各个方向上应具有相同的密度分布;而处于液体表面的微团,其密度分布应该在液面的法线方向上比切线方向衰减更快,形成一个类似椭球状的密度分布.因此在屏幕空间流体绘制中,我们虽然也以球体代替粒子用于绘制,但为描述粒子的密度变化,我们为每个粒子 i 生成一个变换矩阵 G_i ,用于旋转和缩放代表粒子的球体.这样,经旋转和缩放变换后的球体(即椭球体)就能较正确描述各种情况液体微团的密度分布了.

为了确定变换矩阵 G ,我们对每个粒子的邻域粒子应用 WPCA 加权主成分分析法(weighted principal component analysis)^[21],该方法可克服一般 PCA 方法在采样个数较少或采样位置有噪声时易出错的缺点.其具体步骤如下:首先应用 WPCA 计算邻域粒子位置点的加权均值,接着构造一个加权的零和协方差矩阵 C ,并对 C 使用特征分解,解得的特征向量给出了 3 个主轴的方向,而特征值表示邻域粒子在这 3 个方向上的方差;最后以特征值和特征向量构造出各向异性矩阵 G .下面给出其定量描述:

$$x_i^w = \sum_j w_{ij} x_j / \sum_j w_{ij} \quad (4)$$

$$C_i = \sum_j w_{ij} (x_j - x_i^w)(x_j - x_i^w)^T / \sum_j w_{ij} \quad (5)$$

式中 x_i^w 是加权均值, C_i 是协方差, w_{ij} 是各向同性的权重函数,同式(2).

计算时,需要对粒子 x_i 邻域半径 r_i 内的所有粒子 x_j 进行计算.较大的邻域半径 r_i 有助于获取更多的采样数据.经过特征值分解之后,协方差矩阵 C 的形式如下:

$$C = R \Sigma R^T \quad (7)$$

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \quad (8)$$

式中 R 是 3 个特征列向量 v_1, v_2, v_3 组成的旋转矩阵,我们使用 R 来旋转球体的主轴, Σ 是含有特征值 σ_1, σ_2 和 σ_3 ($\sigma_1 > \sigma_2 > \sigma_3$) 的对角矩阵,我们使用 σ_1, σ_2 和 σ_3 来分别影响椭球的长轴、次长轴和短轴的长度,分别对应 v_1, v_2, v_3 这 3 个主轴方向.但在特征值分解之后,有可能产生比 σ_1 小得多的 σ_2 和 σ_3 ,为避免绘制时出现过扁的椭球体及在两个椭球之间出现空隙的现象,为此我们对此进行改进:

$$\tilde{\sigma}_k = \max(\tilde{\sigma}_k, \sigma_1 / k_r) \quad (9)$$

这样,当 σ_2 或 σ_3 过小时,我们用上式中的 σ_1 / k_r 来代替过小的轴长,其中 k_r 是极限缩放比例,本文中我们取值为 $k_r = 5$.由于 $\tilde{\sigma}_k$ 是方差项,我们对球体各轴进行以下比例的缩放:

$$w_k = \frac{c}{r} \sqrt{\tilde{\sigma}_k} \quad (10)$$

式中 w_k 是球体上对应轴的缩放比例, c 是我们定义的全局缩放系数, r 是原球体半径.这里我们取 $1.2 < c < 2.0$,以免使两个椭球部分重叠在一起而出现空隙.另外,当邻域粒子数量过少时,椭球体易出现形变,同时为避免文献[20]中出现的因粒子密度分布在阈值上下变动时出现突变的现象,我们将 $\tilde{\Sigma}$ 设置为单位对角矩阵和生成的矩阵 Σ 的线性插值:

$$\tilde{\Sigma} = \begin{cases} \text{diag}(\omega_1, \omega_2, \omega_3), & N > N_i \\ (1 - \lambda)I + \lambda \text{diag}(\omega_1, \omega_2, \omega_3), & N_0 < N < N_i \\ 1, & N < N_0 \end{cases} \quad (11)$$

而其中

$$\lambda = \frac{N - N_0}{N_i - N_0} \quad (12)$$

式中 λ 是插值系数,式(11)中 $\tilde{\Sigma}$ 是最终用于球体的缩放系数;式(12)中的 N_0, N_r 分别为确定线性插值的使用条件.经实验取值 $N_0=8, N_r=30$ 可达较为满意的效果.这样最终各向异性变换矩阵 G 就被定义为

$$G = R\tilde{\Sigma}R^T \tag{13}$$

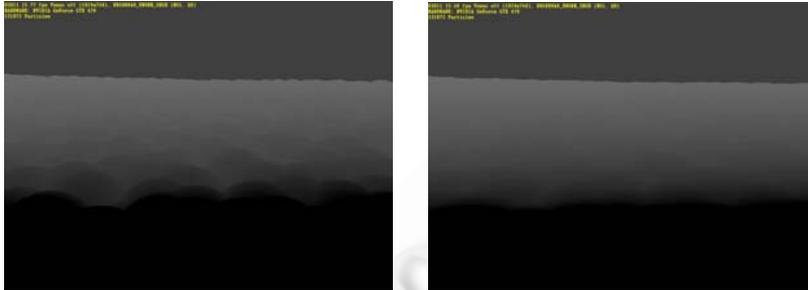


Fig.2 Isotropic(left) and anisotropic(right) depth comparison
图 2 基于各向同性(左)和各向异性(右)绘制的深度图对比

在得到各向异性变换矩阵 G 之后,我们将其应用在绘制粒子球体深度图的步骤中.我们可以在 GPU 的着色程序(vertex shader)中将球体的每个顶点应用矩阵 G 变换得到椭球体,再将椭球体深度值绘制到目标纹理中,就得到了深度图.图 2 为采用各向异性机制前后的深度度绘制对比;图 3 为平静水面的各向同性及各向异性的初步绘制结果对比.从中可看出,经我们的各向异性构建处理,流体表面得到了进一步的平滑.

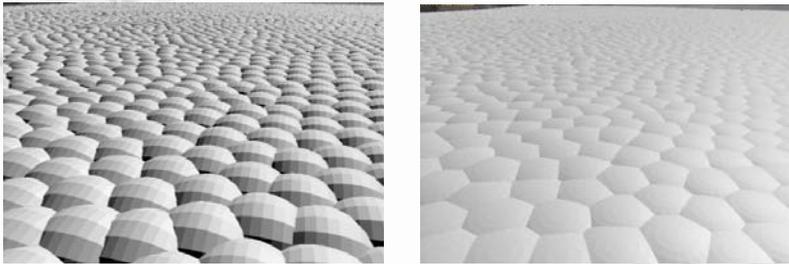


Fig.3 Isotropic(left) and anisotropic(right) primary rendering comparison
图 3 平静水面的各向同性(左)和各向异性(右)初步绘制结果对比

2.3 双边滤波对深度图的平滑实现

上面我们已得到经进一步平滑的流体场景的深度图,但仍没有彻底解决凹凸球体无法表现平滑的流体表面这一问题.平滑深度图实际上是一个二维图像上的降噪问题,常用的高斯滤波(或其变种如双边高斯滤波等)法对高频细节即边界信息的保护效果不明显,会导致边界扩散;且当使用较大的核半径时,易导致大范围的深度值过于平整,减少了表面细节.而双边滤波法^[22]具有能够较好地保持高频细节,使边界清晰等优点,故本文采用双边滤波法对深度图进行最终的平滑实现.

$$I(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\varepsilon) c(\varepsilon, x) s(f(\varepsilon), f(x)) d\varepsilon \tag{14}$$

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\varepsilon, x) s(f(\varepsilon), f(x)) d\varepsilon \tag{15}$$

式中 $f(x)$ 为输入图像, $I(x)$ 为双边滤波后的输出图像,函数 $c(\varepsilon, x)$ 表示当前像素 x 与其临近点 ε 的图像空间临近度, $s(f(\varepsilon), f(x))$ 表示当前像素 x 与它的临近点 ε 的像素值,即其深度值相似度.双边滤波图像空间邻近度函数 $c(\varepsilon, x)$ 和像素值相似度函数 $s(f(\varepsilon), f(x))$ 都可取为高斯函数形式,即:

$$c(\varepsilon, x) = e^{-0.5 \left(\frac{d(\varepsilon, x)}{\sigma_d} \right)^2} \tag{16}$$

$$s(f(\varepsilon), (x)) = e^{-0.5 \left(\frac{\delta(f(\varepsilon), f(x))}{\sigma_f} \right)^2} \quad (17)$$

其中,

$$d(\varepsilon, x) = d(\varepsilon - x) = \|\varepsilon - x\| \quad (18)$$

$$\delta(f(\varepsilon), f(x)) = \delta(f(\varepsilon) - f(x)) = \|f(\varepsilon) - f(x)\| \quad (19)$$

图 4 为应用双边滤波前后的流体深度图效果对比,从中可看出经双边滤波后,深度图更加平滑了.

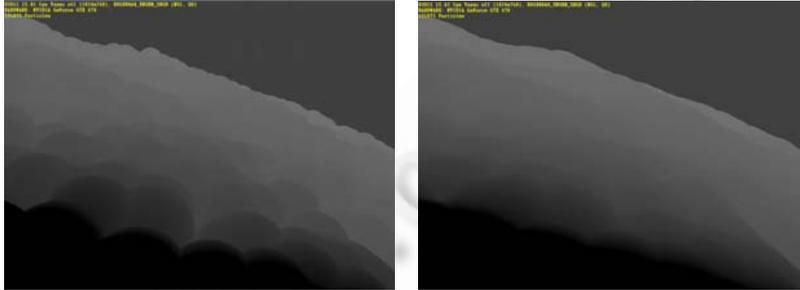


Fig.4 Before(left) and after(right) bilateral filtering depth map comparison

图 4 深度图应用双边滤波前(左)后(右)对比

3 屏幕空间流体场景的真实感绘制

现有屏幕空间流体场景的光照明模型为减少计算起见,往往只采用了简化的模型.这些模型主要考虑了环境的光照影响及较为粗略的高光反射贡献,这样就不可避免地导致场景光照效果的不自然感;另外,现有绘制方法缺乏流体场景的细节效果,如浪花泡沫等,这就极大地降低了场景的真实感.为此,我们提出了一种基于菲涅尔原理改进的精细光照明模型及一种基于曲率的浪花生成模型,并利用 GPU 加速技术,实现了空间流体场景的真实感实时绘制.下面分别加以讨论.

3.1 流体场景的精细光照明模型

Laan 等人^[2]的屏幕空间流体场景光照明模型,主要考虑了环境光的影响及光源高光反射的影响,其光照明模型可用下式来表示:

$$C_{out} = a(1 - F(\bar{n} \cdot \bar{v})) + bF(\bar{n} \cdot \bar{v}) + k_s(\bar{n} \cdot \bar{h})^\alpha \quad (20)$$

式中 C_{out} 为最终的光强度(文献[2]中称其为颜色);为 F 由菲涅尔公式计算得到的光反射率; \bar{n} 为表面法向单位矢量; \bar{v} 为相机方向单位矢量; \bar{h} 为入射光方向与相机方向夹角的半角方向单位矢量; a 为折射的流体颜色; b 为反射的场景颜色;式右边第 3 项为高光项, k_s 和 α 为高光常数.从式(20)可以看出,这是个较为简单的模型,并没有考虑入射光的漫反射贡献,而对高光项反射系数的随意设定更是导致场景光照不真实的主要因素之一.为此,我们提出了一个更为精细的屏幕空间流体场景光照明模型对此加以改进:

$$C = aR_{refrac} + bR_{reflec} + I_p R_{reflec} \cos i + I_p R_{reflec} \cos^\alpha \theta \quad (21)$$

式中 C 为最终的光强度; a, b 和 α 的含义与式(20)同; i 为光源入射角; θ 为镜面反射方向和视线方向的夹角; I_p 为入射光的强度; R_{reflec} 和 R_{refrac} 分别为根据菲涅尔公式(Fresnel equation)定量计算出的光的反射和折射系数:

$$R_{reflec} = \left(\frac{\sin(\theta_i - \theta_r)}{\sin(\theta_i + \theta_r)} \right)^2 + \left(\frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} \right)^2 \quad (22)$$

$$R_{refrac} = (1 - R_{reflec}) \quad (23)$$

式中 θ_i 和 θ_r 分别为光源的入射角和反射角.式(21)的改进光照模型相比原先的简单光照模型式(20),增加了入射光的漫反射项的贡献.而对于高光反射项,则采用菲涅尔公式定量地确定其反射系数.这样就避免了原模型因

随意设定高光反射系数而产生耀目的不自然光照效果.另外,在式(21)计算流体折射部分 a 的贡献时,我们已考虑了其衰减散射的机制,在此就不详细讨论了.因为流体表面经过平滑步骤,而为了计算的准确性及效率,对深度图进行屏幕到空间的变化,得到所有流体表面的像素在空间的位置,再生成相应像素的法向值,保存成一张法向图.光照绘制时的法向都由此图读取.



Fig.5 Fluid scene rendering based on our illumination modal

图 5 由本文光照模型绘制生成的流体场景效果图

图 5 为由本光照模型最终合成的流体场景效果图.图 6 为 Laan 光照模型和本文光照模型绘制效果的对比图.从图中可看出,本文模型不仅考虑了环境、流体折射的影响,同时还考虑了入射光的漫反射及高光反射的定量影响,因而场景的光照效果更加真实自然.



Fig.6 Laan illumination model^[2](left) and ours(right) comparison

图 6 Laan 光照模型^[2](左)和本文光照模型(右)流体场景绘制效果的对比

3.2 基于曲率的浪花生成模型

在流体(液体)动态场景绘制中,浪花及泡沫这些细节效果对于场景的真实感生成起着至关重要的影响.但现有流体模拟方法常采用基于物理的粒子系统模型由于其计算的复杂性,必定会影响系统运行的实时性.Thurney 等人^[23]采用在高度场生成的动态海面上加粒子的方法来模拟浪花,但其视觉融合效果相当差;Laan 等人^[2]采用基于噪声大小在表面上添加浅灰颜色的方式以模拟泡沫,但其视觉效果尚有待提高.

众所周知,平静的水面是不会产生浪花泡沫的.只有当动态流体产生一定程度卷曲时,浪花及泡沫才会产生.为此,我们提出了一个简单而又高效的浪花生成模型.该模型是基于粒子所处流体表面的曲率而定的.一般来说,曲率越大处,其产生浪花泡沫的可能性就越大.流体表面三维空间点 $P(x, y, z)$ 的平均曲率 H 可由下式来确定:

$$H = \frac{K_x + K_y}{2} \quad (24)$$

式中 K_x 和 K_y 分别为沿 X 轴和 Y 轴的主曲率.实际运行时,我们设定一个阈值,当 H 值大于此阈值时,白色浪花就产生;否则就不产生.

为了加快计算,对流体表面的某个像素点的主曲率,我们采用了如下的方法来模拟计算,避免了数学上求解的主曲率方程的复杂性,又能提供很好的视觉效果:

$$K_x = \sin(\arccos(\text{dot}(n, n_x))) \tag{25}$$

$$K_y = \sin(\arccos(\text{dot}(n, n_y))) \tag{26}$$

其中 n 为当前像素法向, n_x 为当前像素相邻 u 方向上相邻的像素法向, n_y 为当前像素相邻 v 方向上相邻的像素法向, 法向是由第 3.1 中的流体表面法向图获得.

为充分利用现有图形硬件的并行功能,我们在 GPU 中设计并实现了上述模拟和绘制的所有步骤,在不失实时性能的前提下,实现了屏幕空间流体场景的真实感绘制.

4 结果

基于上述模型,我们在一台配置为 Intel Core(TM) i5-2500,3.30GHz CPU,4GB 内存,NVIDIA GTX 680 显卡的高档微机上实现了我们的算法.所有程序都是在 Direct 11 上实现的.图 7 为卷浪场景实时绘制的视频序列,图 8 为海啸引起的巨浪淹没海滩附近物体的动态过程的实时绘制序列图片.

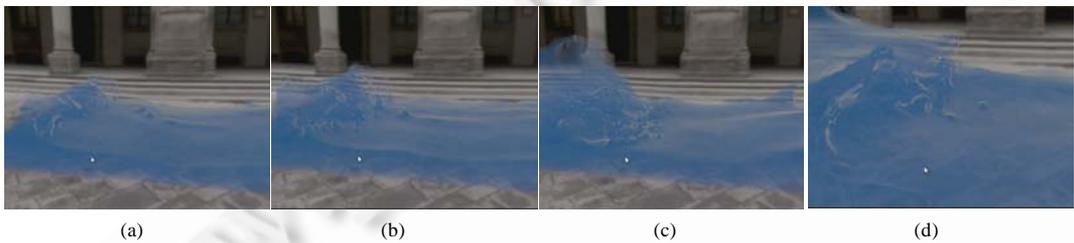


Fig.7 Breaking wave moving sequence

图 7 卷浪动态场景的实时绘制序列

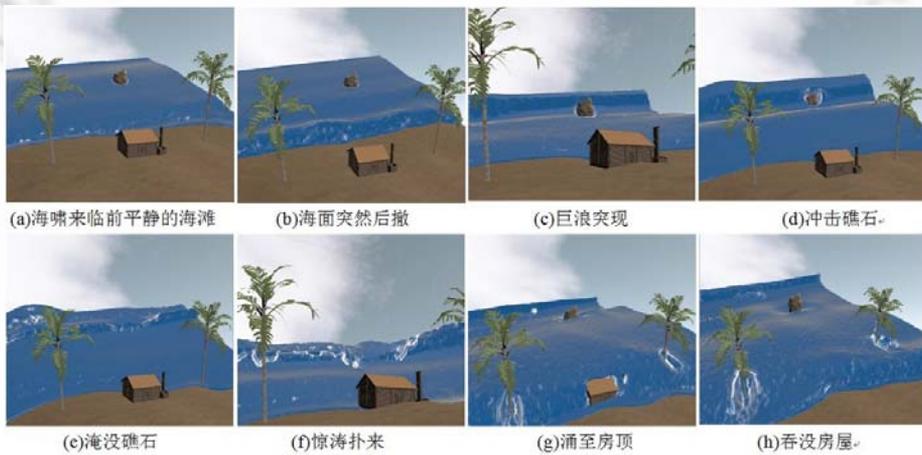


Fig.8 Tsunami destroying beach sequence

图 8 海啸巨浪冲击海滩过程实时绘制的图像序列

Table 1 Computation and rendering performance analysis of break wave case

表 1 卷浪实例的计算与绘制性能分析

粒子数(k)	WS (ms)	LS (ms)	AS (ms)	DS (ms)	总计(ms)	帧率(fps)
32	2.53	0.17	1.51	15.84	20.05	49.88
64	5.06	0.41	4.56	15.80	25.83	38.72
128	9.85	0.88	11.24	15.82	37.79	26.46

WS 表示没经平滑处理时每帧绘制所需的时间(ms);LS,AS 和 DS 分别表示经拉普拉斯平滑(laplacian

smoothing)、各向异性表面构建平滑(anisotropic smoothing)和双向滤波深度平滑(depth smoothing)时每帧绘制所需的时间(ms)。

Table 2 Computation and rendering performance analysis of Tsunami case
表 2 海啸实例的计算与绘制性能分析

粒子数(k)	WS (ms)	LS (ms)	AS (ms)	DS (ms)	总计(ms)	帧率(fps)
32	5.50	0.22	2.98	15.26	23.96	41.73
64	9.89	0.75	7.84	16.56	35.04	28.54
128	20.54	4.10	20.10	16.53	61.27	16.32

注:表 2 中各量含义同表 1。

表 1 和表 2 分别对上面卷浪及海啸场景实时绘制的性能进行了分析。从表中可以看出,虽然这两个动态场景较为复杂,而双向滤波深度平滑(depth smoothing)这一步骤花费了较多的计算时间。因为在对深度图进行的双边滤波,我们并未将他进行并行化,而只是在传统的渲染管线的像素着色器阶段,对每个像素进行滤波,程序中要去采样计算周围的 18×18 个像素,只有一个核去处理,所以非常耗时,也可以从表 1、表 2 中看出,不管粒子数多少,需要的时间都是一样的,因为渲染图像的像素尺寸是一样的。今后将对双边滤波改进,使其可以并行化,将会大大提高效率。

但总体上说,还是达到了实时绘制。而场景的真实感比起现有其他算法(如:文献[2])则有了较大的改进。欲了解更多详情,请观看本文所附的视频文件(ScreenSpace.wmv,其最新演示版本可访问网址下载获取:<http://www.cad.zju.edu.cn/home/zywang/ScreenSpace.wmv>),从中可以看出,依据本文算法生成的屏幕空间流体场景实时绘制的效果有较强的真实感。

5 结 论

本文提出了一种新的屏幕空间流体场景真实感绘制算法,该算法以 SPH(光滑粒子动力学)为框架。本算法的优势在于保证屏幕空间实时性能的前提下,大大改进了流体场景的绘制质量。我们提出了一种基于各向异性流体表面平滑模型。本模型中,我们将粒子看作是动态变化的椭球体而非各向同性的球体,经过粒子空间位置平滑、各向异性表面建模平滑及双边滤波深度图平滑这 3 个步骤的处理,使屏幕空间的流体表面的抽取效果大为改善;提出的精细流体表面光照模型和基于曲率的浪花生成模型使得流体场景的绘制更具真实感。并在 GPU 上设计并实现了加速绘制的框架以保障复杂场景绘制的实时性。相比 Laan 等人^[2]的算法,本文算法的绘制质量有了相当大的提高。

在将来的工作中,我们将进一步完善本文的各向异性屏幕空间流体场景真实感绘制算法,并采取适当的简化措施进一步提高算法效率,并改进浪花飞溅的真实感效果,以求在更为复杂的流体场景中实现更高帧率的真实感实时绘制,以使该技术在实时仿真及游戏等领域得到应用推广。

References:

- [1] Müller M, Schirm S, Duthaler S. Screen space meshes. In: Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation (SCA 2007). Eurographics Association, 2007. 9–15.
- [2] van der Laan WJ, Green S, Sainz M. Screen space fluid rendering with curvature flow. In: Proc. of the 2009 Symp. on Interactive 3D Graphics and Games, 2009. 91–99.
- [3] Stare J, Flume E. Depicting fire and other gaseous phenomena using diffusion processes. In: Proc. of the SIGGRAPH. 1995. 129–136.
- [4] Müller M, Charypar D, GROSS M. Particle-Based fluid simulation for interactive applications. In: Proc. of the ACM SIGGRAPH Symp. on Computer Animation. 2003. 154–159.
- [5] Keiser R, Adams B, Gasser D, Bazzi P, Dutre P, Gross M. A unified lagrangian approach to solid-fluid animation. In: Eurographics Symposium on Point-Based Graph. 2005. 125–148.
- [6] Müller M, Solenthaler B, Keiser R, Gross M. Particle-Based fluid interaction. In: Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2005. 237–244.
- [7] Desbrun M, Cani MP. Space-Time adaptive simulation of highly deformable substances. Technical Report, INRIA No.3829, 1999.
- [8] Adams B, Pauly M, Keiser R, Guibas LJ. Adaptively sampled particle fluids. ACM Trans. on Graphics, 2007,26(3):481–487.

- [9] Yan H, Wang ZY, He J, Chen X, Wang CB, Peng QS. Real-Time fluid simulation with adaptive SPH. *Computer Animation and Virtual Worlds*, 2009,20(5):417-426.
- [10] He J, Chen X, Wang ZY, Yan H, Cao C, Peng QS. A new adaptive model for real-time fluid simulation with complex boundaries. *The Visual Computer*, 2010,26(4):243-252.
- [11] Lorensen WE, Cline HE. Marching cubes. A high resolution 3D surface construction algorithm. In: *Proc. of the 14th Annual Conf. on Computer Graphics and Interactive Techniques*. ACM Press, 1987. 163-169.
- [12] Rosenberg ID, Birdwell K. Real-Time particle isosurface extraction. In: *Proc. of the 2008 Symp. on Interactive 3D Graphics and Games*. New York: ACM, 2008. 35-43.
- [13] Williams BW. Fluid surface reconstruction from particles [MS. Thesis]. The University of British Columbia, 2008.
- [14] Whitted T. An improved illumination model for shaded display. *Communications of the ACM*, 1980,23(6):343-349.
- [15] Johanson C. Real-Time water rendering—Introducing the projected grid concept [MS. Thesis]. Lund University, 2004.
- [16] Taubin G. Geometric signal processing on polygonal meshes. *Eurographics State of the Art Reports*, 2000. 158-164.
- [17] Liu MB, Liu GR, Lam KY. Adaptive smoothed particle hydrodynamics for high strain hydrodynamics with material strength. *Shock Waves*, 2006,15:21-29.
- [18] Dinh HQ, Turk G, Slabaugh G. Reconstructing surfaces using anisotropic basis functions. In: *Proc. of the Int'l Conf. on Computer Vision (ICCV)*. 2001. 606-613.
- [19] Yu JH, Turk G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In: *ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 2010. 217-225.
- [20] Kalaiah A, Varshney A. Statistical point geometry. In: *Proc. of the 2003 Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*. 2003. 107-115.
- [21] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: *Proc. of the IEEE Int'l Conf. on Computer Vision*. 1998. 836-846.
- [22] Thurey N, Muller-Fischer M, Schirm S, Gross M. Real-Time breaking waves for shallow water simulations. In: *Proc. of the 15th Pacific Conf. on Computer Graphics and Applications*. 2007. 39-46.



王章野(1965—),男,博士,副教授,主要研究领域为计算机真实感图形,虚拟现实,信息可视化,红外成像仿真.



王丰金(1988—),男,硕士,主要研究领域为计算机图形学,虚拟现实.



张海涛(1988—),男,硕士生,主要研究领域为计算机真实感图形,虚拟现实.



张昆(1987—),男,硕士,主要研究领域为计算机图形学,游戏设计.



施肖菁(1970—),女,博士,讲师,主要研究领域为计算机图形学,图像处理.



彭群生(1947—),男,博士,教授,博士生导师,主要研究领域为计算机真实感图形,虚拟现实,视觉处理,生物计算,可视分析.