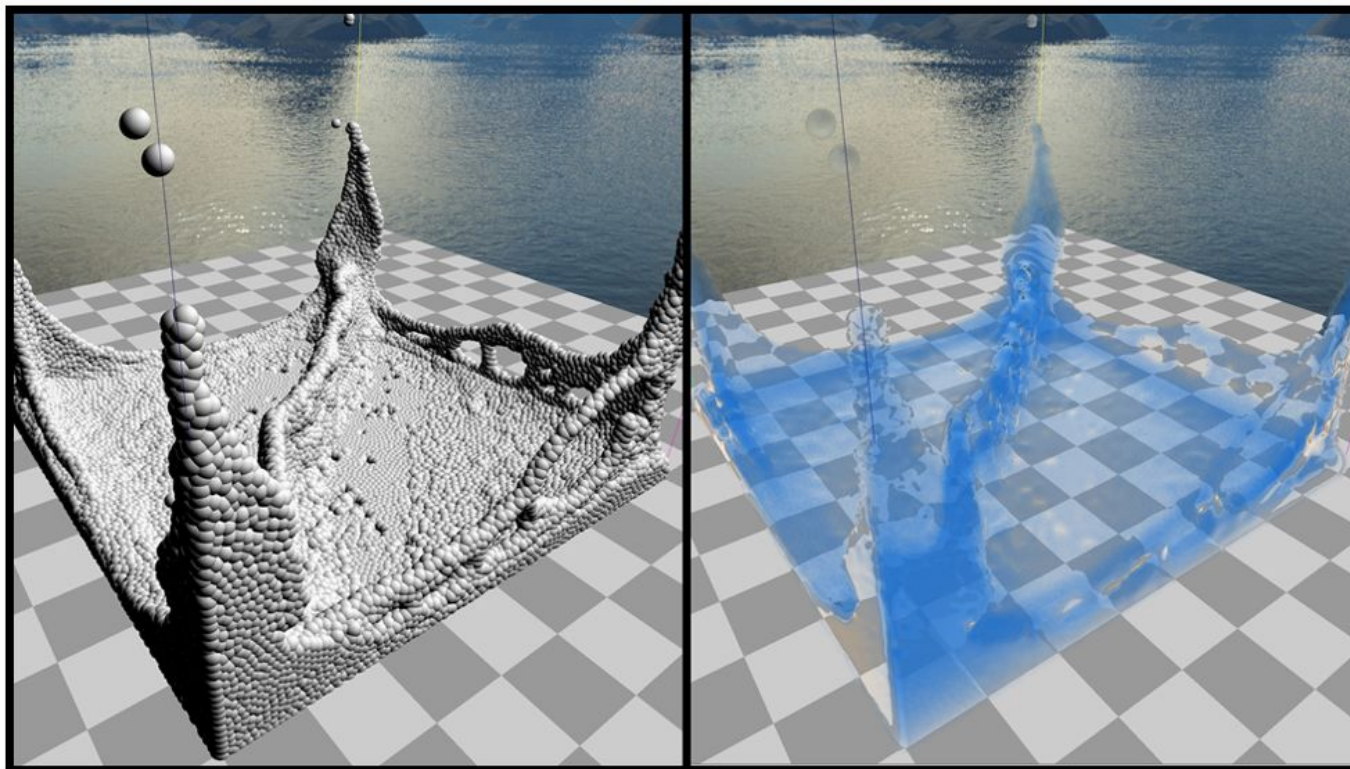


知乎

首发于
探索SPH

赞同 710

分享

液体渲染：一种屏幕空间方法



Nae Zhu

已关注

YiQiuuu、jeffzhu、陆柒、Gfans、Ubp.a 等 710 人赞同了该文章

本文介绍一种液体粒子模拟的屏幕空间渲染（Screen Space Fluid Rendering, 简称SSF）[1]方法。这种方法没有M
广泛的应用前景。

▲ 赞同 710 ▼

● 35 条评论

➤ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

..

文是对粒子模拟渲染部分的补充。



赞同 710



分享

传统方法

计算机图形学中，欧式网格和拉氏粒子是两种主要的流体离散方法。无论是哪一种方法，都需要先重建液体表面，再进行渲染着色，才能得到最终图像。液体外的其他流体各有渲染方法，如渲染烟

雾可用体渲染。在

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

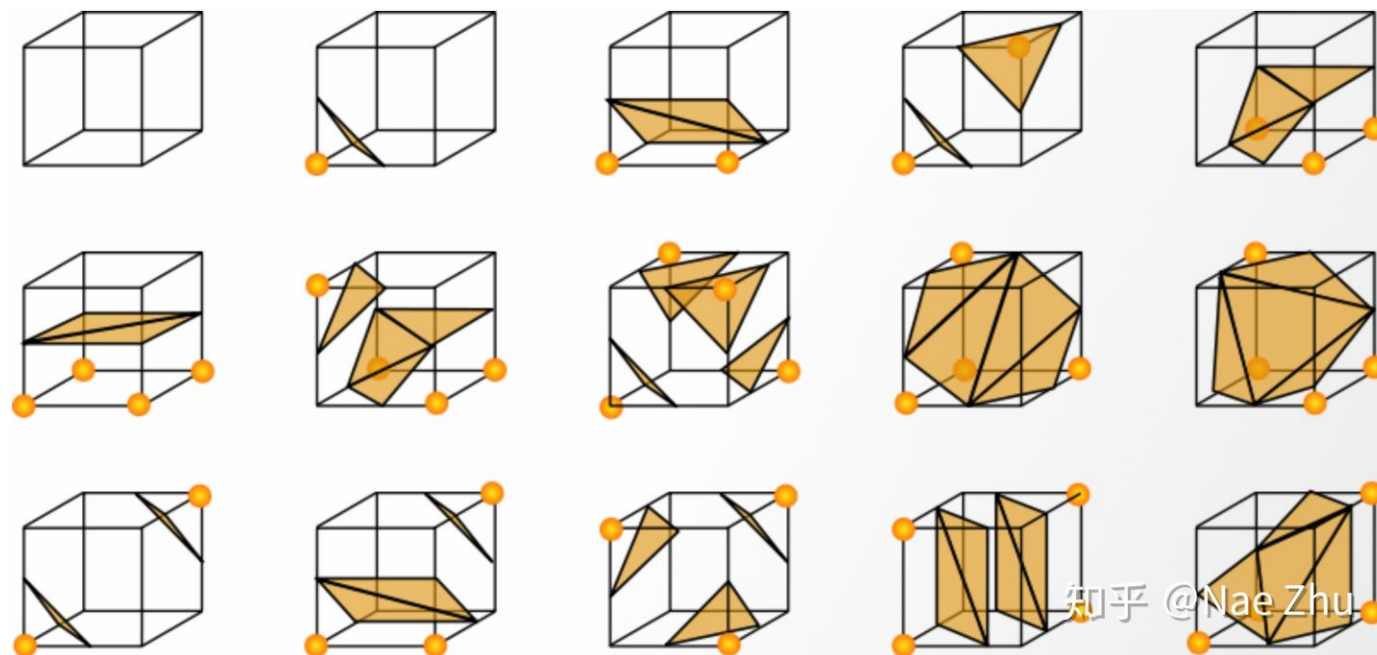
高维上的等高线)，将液体表面定义为液体密度等于某个小常数 C 的点集。Marching Cubes方法将空间离散为等大的立方体网格，检查每个网格上8个顶点的液体密度情况。



赞同 710



分享



论文[3]中的15种顶点和重建形态

若网格上一条边的两个端点密度值跨过阈值 C ，则算法认为这条边上存在一个表面Mesh的顶点。上图中，被标记的顶点密度大于 C ，否则小于 C 。将每个网格构造的多边形拼接，即可得到整个液体表面的Mesh。油管上有一个很棒的演示动画，把构造表面的过程可视化为一个立方体在空间中扫描的过程。可以理解算法名字中“移动（Marching）”一词的来历。（视频版权受限，无法搬运，抱歉！）

欧式流体天生就是网格化的，一种方法是直接对流体进行网格化，另一种方法是使用SPH方法。SPH方法的密度是通过周围粒子的密度来计算的。

赞同 710

35 条评论

分享

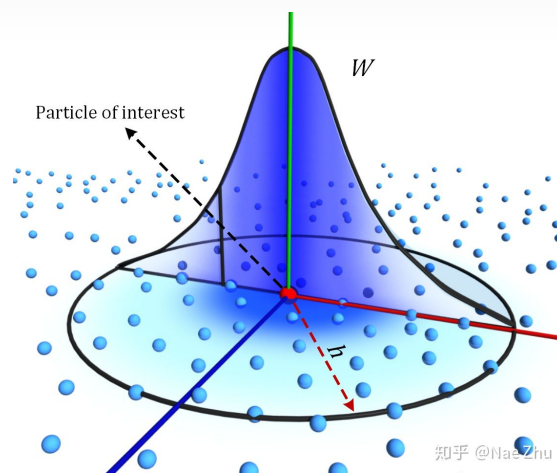
取消喜欢

收藏

申请转载



知乎

首发于
探索SPH

图片来源Neutrino-SPH Dynamic Simulation System[4]

直接使用SPH计算密度重建得到的液体表面会有很强的粒子感。Adam等人提出使用流体的局部区域信息重构粒子数量和大小，Yu等人提出使用在液体不同区域使用椭球形的粒子替代球形粒子，都取得了很好的重建效果。可以看到，液体平静的表面基本不再有粒子状的伪像（artifact），浪尖等细节处也被还原得更加精细。

赞同 710

分享

▲ 赞同 710 ▼

35 条评论

分享

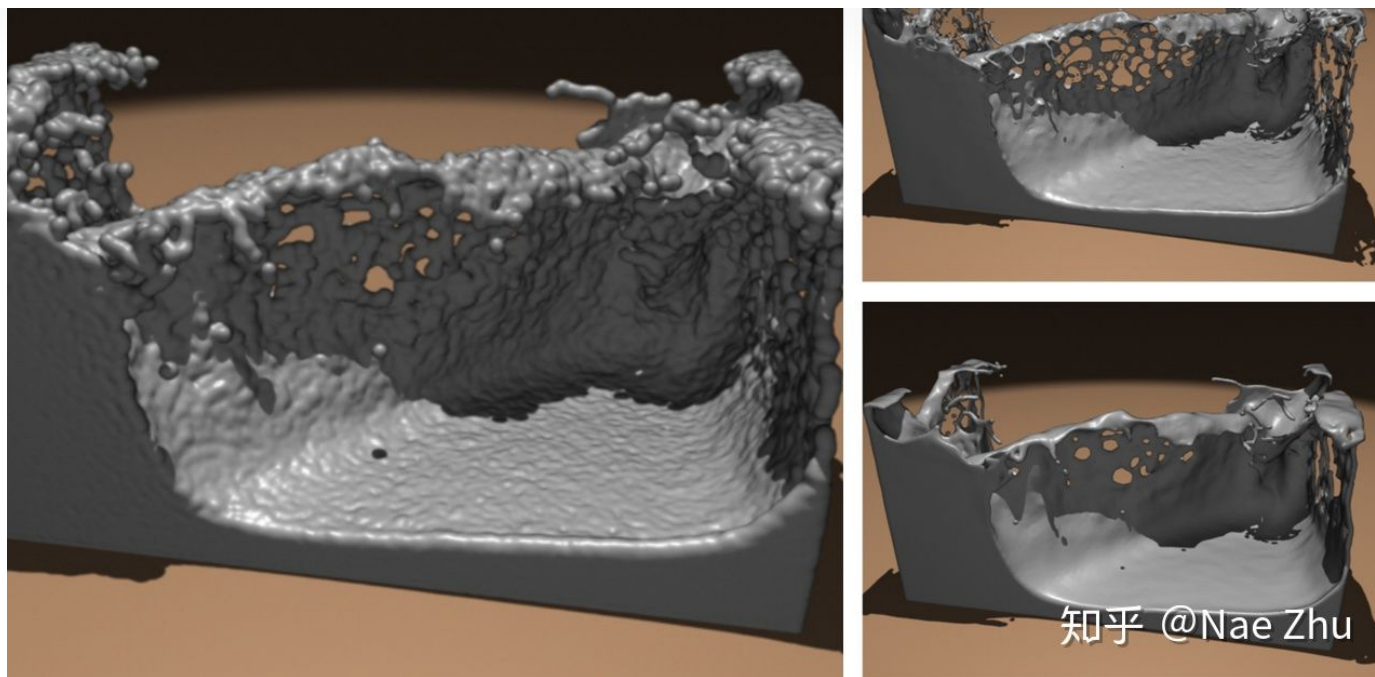
取消喜欢

★ 收藏

申请转载

...

知乎

首发于
探索SPH

左：球形粒子，右上：Adam，右下：Yu

赞同 710

分享

屏幕空间渲染

传统方法的效果优良，但是实现麻烦，复杂度高。例如Yu的方法，不仅使用了PCA处理粒子形状的变换矩阵，还需要为渲染重建一次邻居搜索哈希网格，代价比较高昂。

屏幕空间渲染（Screen Space Fluid Rendering, 简称SSF）则给出了粒子渲染的一种新思路。这个算法不涉及液体网格表面的匹配和重建，实现相对简单得多。在不追求全局光照的实时管线渲染应用中，它的效果还比较令人满意。这个算法也是Position Based Fluids原论文中使用的渲染算法。

▲ 赞同 710 ▼

● 35 条评论

➤ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

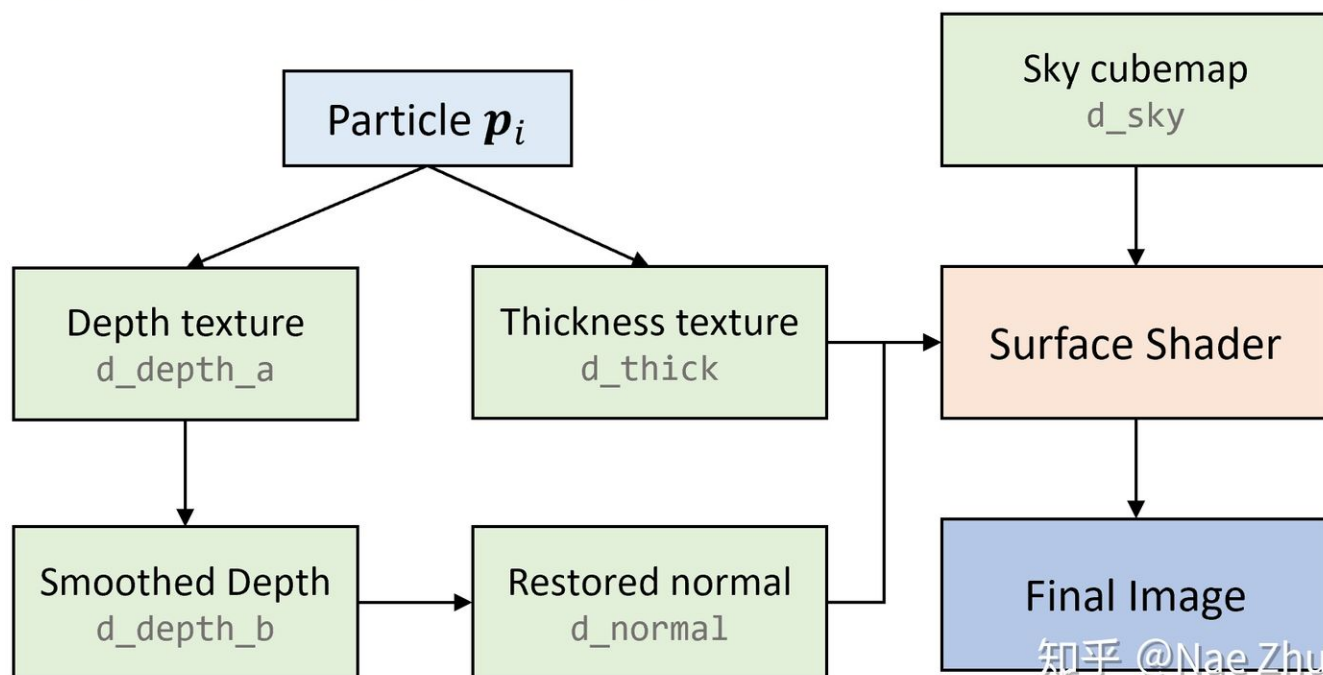
首发于
探索SPH

下液体的厚度。然后对深度纹理进行平滑，并从平滑后的深度纹理中还原表面法线。得到法线后便可以计算反射/折射，对表面进行最终的着色了。

赞同 710

分享

■ Vertex Buffer ■ Texture ■ Shader



渲染用纹理和流程

深度纹理

深度纹理是整个算法的核心。深度指的是液体表面到摄像机的距离。我们使用的是Y轴向上的摄像机，则深度可以认为是被渲染物理的Z坐标。渲染时，开启深度测试，这样我们就能保证处于表面的粒子覆盖内部的*

▲ 赞同 710 ▼

35 条评论

➤ 分享

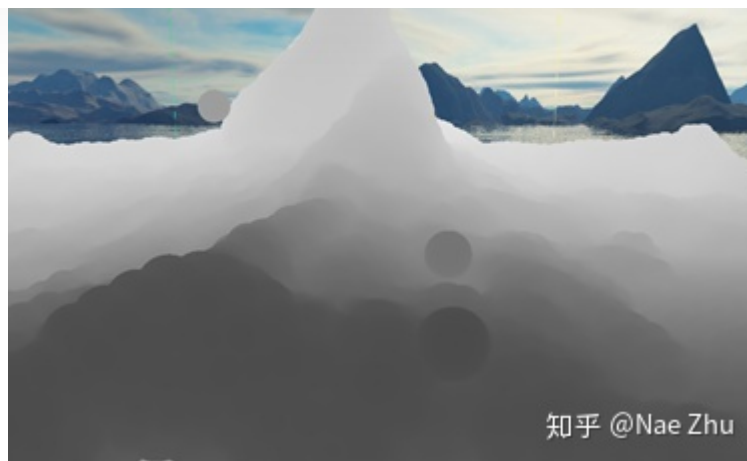
♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

深度纹理。越白距离摄像机越远

赞同 710

分享

深度纹理是使用Point Sprite渲染的。所谓Point Sprite，就是将粒子以Point形式渲染，这样会将粒子渲染成一个正方形。为了渲染成圆形，在Fragment Shader中读取PointCoord，discard圆以外的像素即可。同时要注意以下两点

1. 要在Vertex Shader中根据粒子深度正确地计算并设置Point大小，以符合透视规律。
2. 要反应球形粒子不同位置的深度差异，因此需要结合粒子位置和PointCoord计算深度值。

厚度纹理

厚度纹理记录了表面下液体的厚度。着色时考虑液体厚度，能够增加液体的层次感。根据Beer-Lambert定律，约厚的液体透光率越低。

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

厚度纹理。越白越厚

赞同 710



分享

厚度纹理也是利用Point Sprite渲染的。与深度纹理一样，也需要根据粒子深度调整渲染Sprite的大小。不同的是，渲染时关闭深度测试，开启加法混合，从而使得重叠的粒子厚度能够叠加。同时，要根据PointCoord正确地计算每个像素的厚度值，反应球形粒子四周薄，中间厚的特点。

法线还原

得到深度纹理就可以还原表面法线了。法线储存在一张RGB三通道24bits的二维纹理上。

三维空间中的参数曲面 $\mathbf{P}(s, t) = (x, y, z)$ 上一点的法向量可由公式 $\mathbf{n} = \frac{\partial \mathbf{P}}{\partial s} \times \frac{\partial \mathbf{P}}{\partial t}$ 给出。实际上，两个偏导给出了切平面上的两个不平行的向量，因此叉乘可以得到这个平面的法向量。

纹理坐标就是曲面参

纹理中记录了Z坐标

赞同 710



35 条评论

分享

取消喜欢

收藏

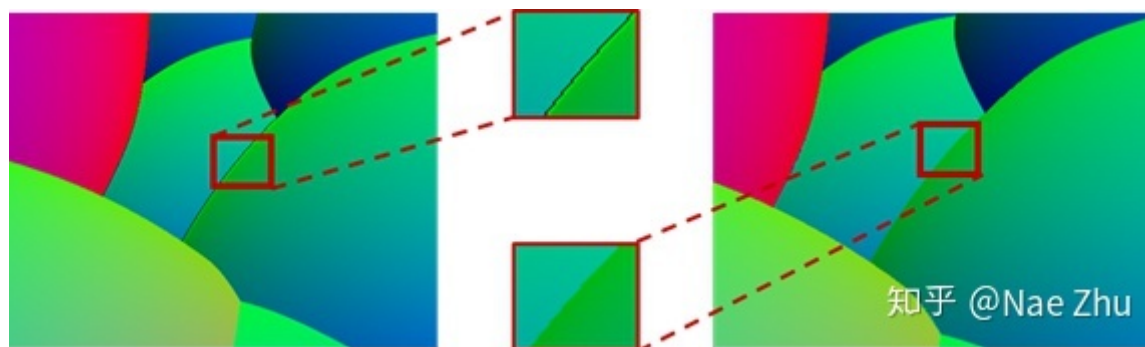
申请转载



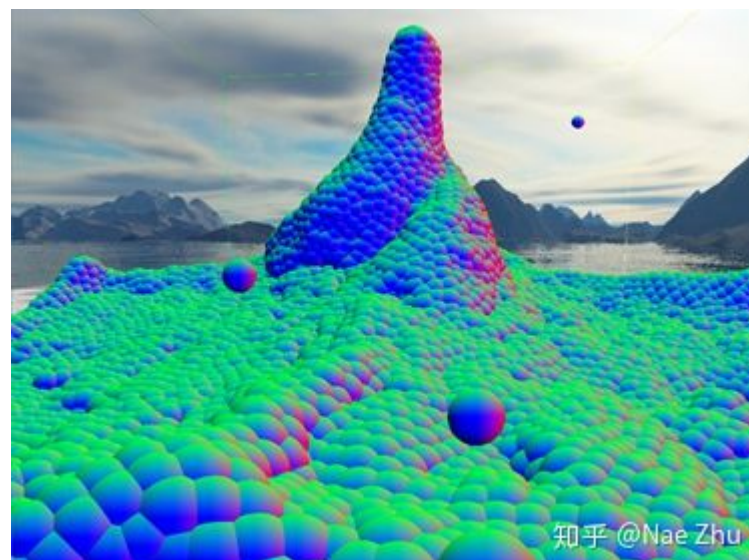
知乎

首发于
探索SPH

上面的近似取了前向差分和后向差分的较小值。这是因为深度纹理上有许多深度不连续的地方。使用单独使用前向或后向差分都会导致不正确的结果。



深度不连续处的差分处理。左：错误的法向量还原，右：正确的法向量还原



▲ 赞同 710 ▼

● 35 条评论

➤ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...

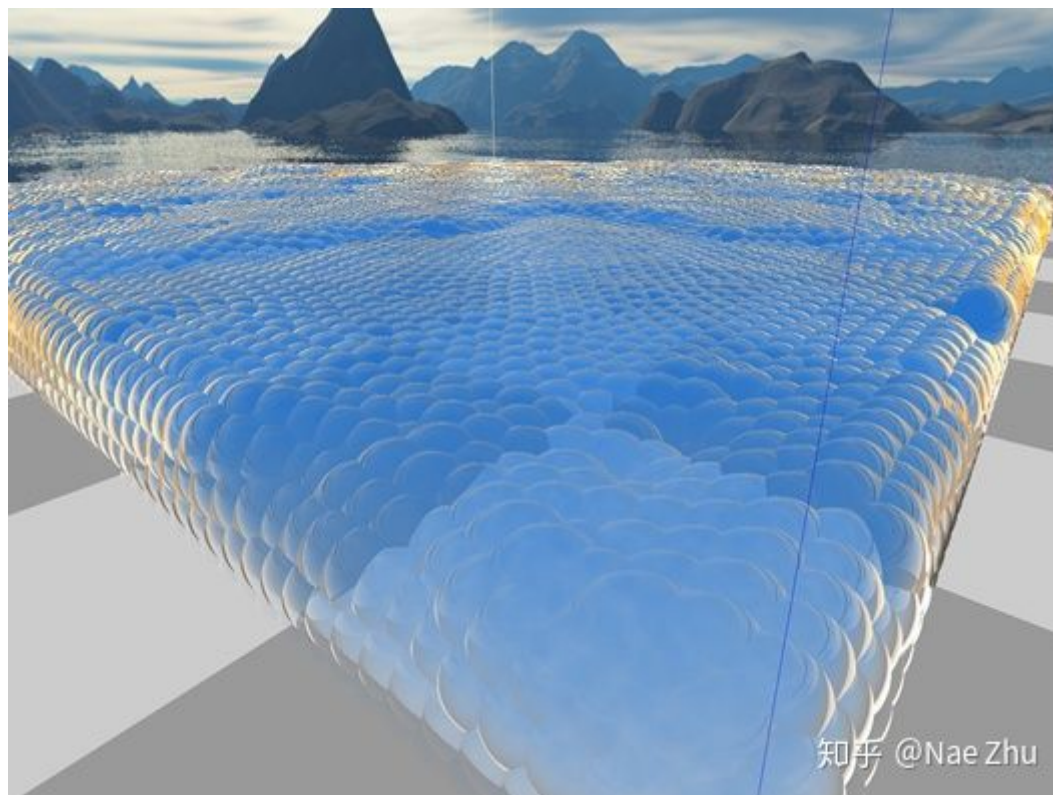
知乎

首发于
探索SPH

直接平滑法向量也是一种可行的做法。但因为粒子的深度信息未变，而计算着色时，需要使用粒子位置做光线追踪操作，因此会损失一些精确性。另外，因为法向量是三维的，而深度是标量，平滑法向量的计算量也比较大。

深度平滑

没平滑的直接渲染得到的画面会有很强的粒子感。



赞同 710



分享

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

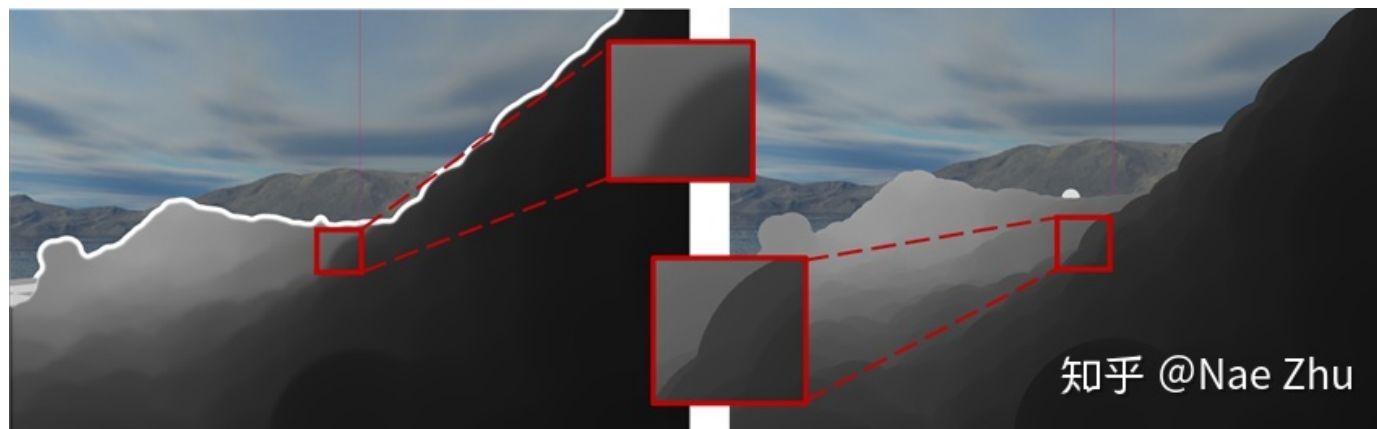
📄 申请转载

...

知乎

首发于
探索SPH

一块。



平滑深度纹理。左：高斯模糊，右：双边滤波

双边滤波是一个复杂度为 $O(r^2)$ 的算法， r 是平滑半径。测试中发现，当 r 大于15个像素的时候，就会对实时性产生显著影响了。然而，粒子大小常常不止15像素。一种方法是，使用双边滤波算法的一次近似，[2]中有描述。但这种方法会带来一些Artifact。因此我采用的方法是使用小平滑半径平滑多次，从而以较小的复杂度近似达到大平滑半径的效果。

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

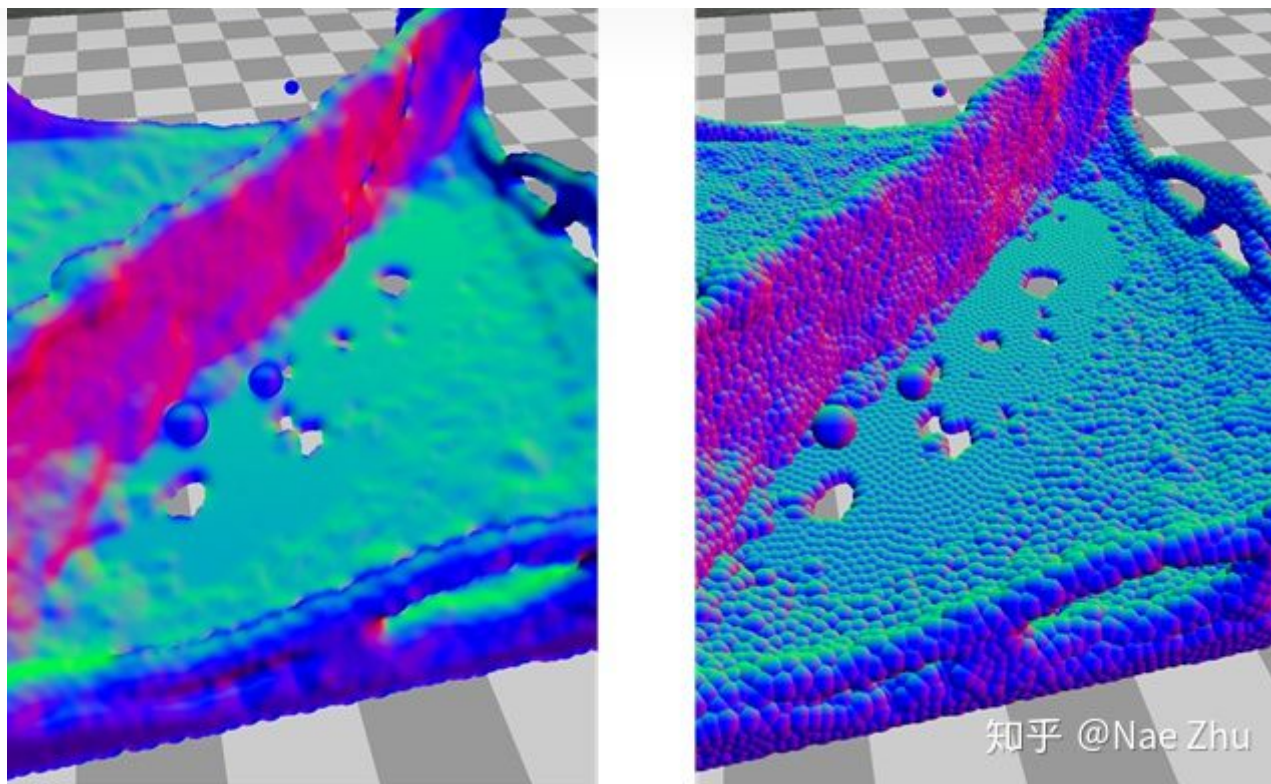
📄 申请转载

...

知乎

首发于
探索SPH

..



平滑后法线。左：平滑后，右：平滑前

表面着色

有了深度纹理，可以还原每个像素在摄像机空间的坐标。有了法向量纹理，可以计算着色。我们综合考虑液体表面的折射和反射，计算每个像素最终的颜色。

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

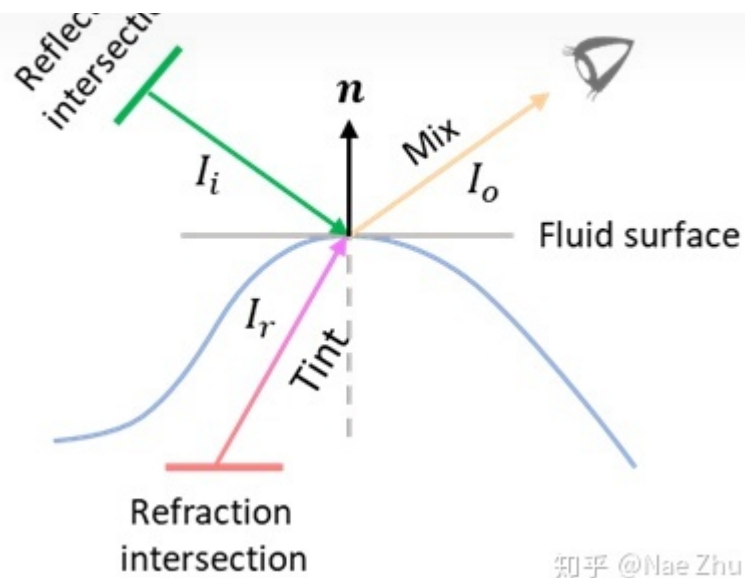
♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

光线在液体表面的传播情况

首先我们可以计算表面的折射方向和反射方向，并通过反向追踪光线，得到折射光和反射光的颜色。下面的公式中，折射方向没有严格地按照折射定律计算，而是把出射光向法线靠近得到。

反射方向 $I_i = -I_o + 2n$

折射方向 $I_r = -I_o - 0.2 \cdot n$

计算出射光颜色，不仅要考虑光线追踪得到的颜色，还要考虑折射光线所穿过的液体的颜色。

Beer-Lambert定律解释了为什么一杯水是透明的，游泳池泛着蓝光，而大海是深沉的碧蓝。根据Beer-Lambert定律，液体的透光量随液体的厚度呈指数衰减。我们使用修改的Beer-Lambert定律计算折射光的颜色

▲ 赞同 710 ▼

35 条评论

分享

取消喜欢

★ 收藏

申请转载

...

知乎

首发于
探索SPH

..

出射光的颜色是反射颜色和折射颜色的混合，混合的比例由菲涅尔定律给出。菲涅尔定律的计算量较大，计算机图形学中常常使用Schlick公式代替。Schlick公式根据出射光与法线的夹角给出了反射率

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos \theta)^5, \quad R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2} \right)^2$$

其中 n_1 和 n_2 分别是水和空气的折射率。

于是我们终于得出最终的渲染结果。

▲ 赞同 710 ▼

● 35 条评论

➦ 分享

♥ 取消喜欢

★ 收藏

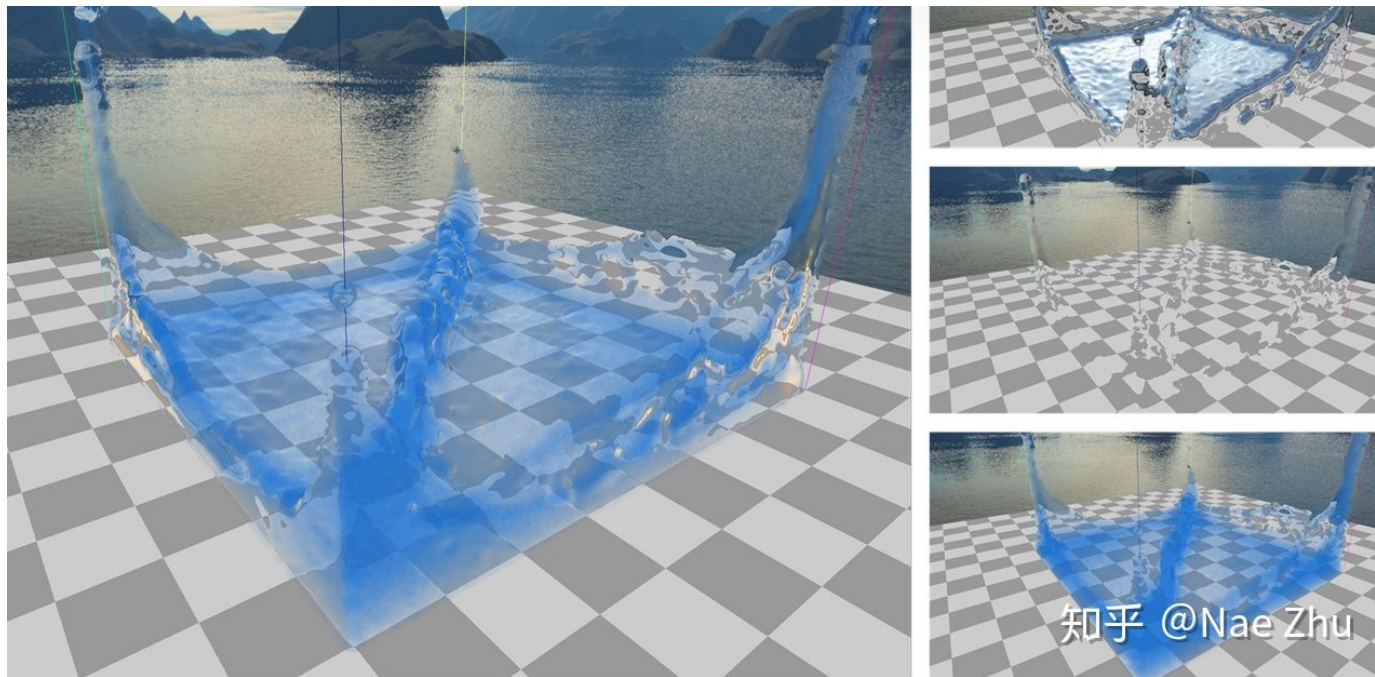
📄 申请转载

...

知乎

首发于
探索SPH

..



左：最终渲染，右：反射颜色，折射颜色，厚度染色后折射颜色

可能的改进

还有许多地方能够改进屏幕空间渲染的效果。下面列出几条我能想到的

1. 实现阴影和焦散。焦散的实现可参考[6]
2. 改进平滑算法。双边滤波算法保留深度的不连续性的同时，也一定程度上保留了粒子边界。
3. 借鉴[5]的思想，渲染到深度纹理时，将粒子渲染为椭球形，而不是球形。

脚注

▲ 赞同 710 ▼

● 35 条评论

➤ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

..

这是我第一次做管线渲染，也是第一次做物理模拟。项目的选题很大程度上受到了 @Raymond Fei [7] 的影响。有任何描述错误，或需要补充说明的部分，请不吝指出！

[1] W. J. van der Laan, S. Green, and M. Sainz, "Screen space fluid rendering with curvature flow," in Proceedings of the 2009 symposium on Interactive 3D graphics and games - I3D ' 09, 2009, p. 91.

[2] Screen Space Fluid Rendering for Games - Nvidia

[3] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH ' 87, 1987, pp. 163–169.

[4] Neutrino-SPH Dynamic Simulation System

[5] J. Yu and G. Turk, "Reconstructing surfaces of particle-based fluids using anisotropic kernels," ACM Trans. Graph., vol. 32, no. 1, pp. 1–12, Jan. 2013.

[6] C. Wyman and S. Davis, "Interactive Image-Space Techniques for Approximating Caustics," in Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, 2006, pp. 14–17.

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

..

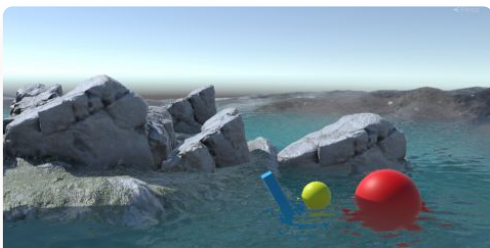
编辑于 2018-06-23 20:52

计算机图形学 计算机科学 渲染

文章被以下专栏收录

探索SPH
SPH in Computer Graphics

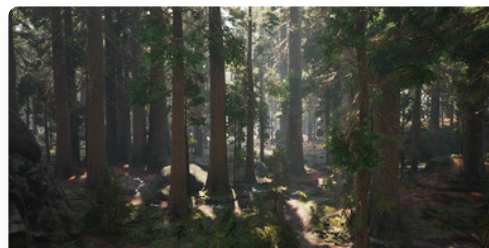
推荐阅读

Unity URP真实感水体渲染工
具（一）水的形态，折射 差

alean...

发

▲ 赞同 710 ▼

影视和引擎当中的PBR材质和线
性色彩空间渲染概念：2.RenderPass-渲
染管道

最简单的2D云

这篇笔记要生成的云纹理。代码
Github，对应SimplestCloud，
论背景Gardner'明过程序化的方

取消喜欢 ★ 收藏 申请转载 ...

35 条评论

 切换为时间排序

写下你的评论...



Milo Yip

2018-06-24

印象中还有一篇 Müller, Matthias, Simon Schirm, and Stephan Duthaler. "Screen space meshes." *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007.

 6

Nae Zhu (作者) 回复 Milo Yip

2018-06-24

嗯，09年I3D这篇Screen Space Fluids正是受到了Muller的启发，Related Works里有提到

 赞

猪小弟

2018-06-23

先膜一波，腿腿牛匹，学习了

 3

Nae Zhu (作者) 回复 猪小弟

2018-06-23

嘿嘿

 赞

林修

 赞同 710  35 条评论 分享 取消喜欢 收藏 申请转载

...



Nae Zhu (作者) 回复 林修

2019-09-14

像这样简单的算法可以直接从OpenGL开始做。其他一些可能需要在成熟的框架上实现，例如渲染算法很多会在mitsuba的基础上开发

2



林修 回复 Nae Zhu (作者)

2019-09-14

好的谢谢

赞

展开其他 3 条回复



Nae Zhu (作者)

2018-07-09

不需要真的绘制椭球体，而是在屏幕空间点精灵+fragment shader discard椭球以外的部分

1



k-ye

2018-06-24

帅炸！回头抄一波...

1



douysu

2020-12-01

很棒，赞赞赞

赞



芒果Uki

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

📄 申请转载

...

知乎

首发于
探索SPH

..



晓曦

2020-07-18

这是什么软件做出来的呢

赞



演奇

2019-09-12

现在的好多算法都是重构吗？就是重新写一套类似maya,3dmax的引擎。渲染特效模拟等等。

赞



Nae Zhu (作者) 回复 演奇

2019-09-14

如果只是重构就发不了文章了

赞



flashiyi

2019-04-08

其实这是个很直观的想法，我想有不少人最初的方案都是这个（因为2d的就和这个差不多）。总之，最后还是用了这么不物理的法子.....

赞



墨以白

2018-07-14

视频好像加载不出来😅

赞



长毛

2018-07-11

依然很贵....

赞同 710

35 条评论

分享

取消喜欢

收藏

申请转载

...


生生

知乎

首发于
探索SPH

是的。平滑之外可以2 pass实现。平滑又需要额外几个pass，每个pass代价也比较大

👍 赞

 小木船

2018-07-09

你好，我也研究了一段时间PBF的SSF渲染，看到你的文章很有感触，你做的效果很棒，给了我很大的启发，首先说声感谢。我也一直致力于各向异性粒子的渲染，而不是点精灵，点精灵虽然简单，但是无法改变圆球的本质，所以各种滤波方法虽然能平滑一些，但是还是不能从根本上解决问题。关于各向异性我研究了一些，但是限于OpenGL只是一知半解，低版本的OpenGL中glSloidSphere函数可以通过传递不同的轴参数绘制任意形状和姿态的椭球体。但是高版本的OpenGL中，至今未找到好的解决方式，想请教下楼主。期待回复，感谢感谢！

👍 赞



小龙虾 回复 小木船

2020-10-22

用yu的PCA方法自己生成矩阵去算吧

👍 赞



小龙虾 回复 小木船

2020-10-22

我也觉得yu的方法更符合平滑，不过速率真的不太行

👍 赞



ZUHXS

2018-07-03

这个天空盒真实印象深刻233

👍 赞



ExboCooop

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

♥ 取消喜欢

★ 收藏

📄 申请转载

...



Nae Zhu (作者) 回复 ExboCooope

2018-06-28

可以新增1bit的粒子边界分量，关键是怎么利用这个信息。一种思路是，对粒子边界的像素使用不同的平滑算法，但我觉得进考虑边界本身是不够的，因为你需要保持平滑后边界上和边界周边深度的连续性。另一种思路是对高斯模糊补充一个采样条件：采样点与中心点深度超过一个阈值，则跳过采样点的加权平均。这种做法能得到比双边滤波更平滑的结果，但其实不需要记录边界分量。实际上，对于这种特殊形势的图像平滑问题，用marching cubes训练CNN也许性能更好。。

赞

CarefreeQ

2018-06-26

6666 行动派，

赞



zz楠

2018-06-25

6666

赞



王天飞

2018-06-25

先摸一波，腿腿牛匹，学习了

赞



0x2F8E64

2018-06-24

感谢，对新手帮助很大

赞

赞同 710

35 条评论

分享

取消喜欢

收藏

申请转载

...

知乎

首发于
探索SPH



👍 赞

1 2 下一页

▲ 赞同 710 ▼

💬 35 条评论

➦ 分享

❤️ 取消喜欢

★ 收藏

📄 申请转载

