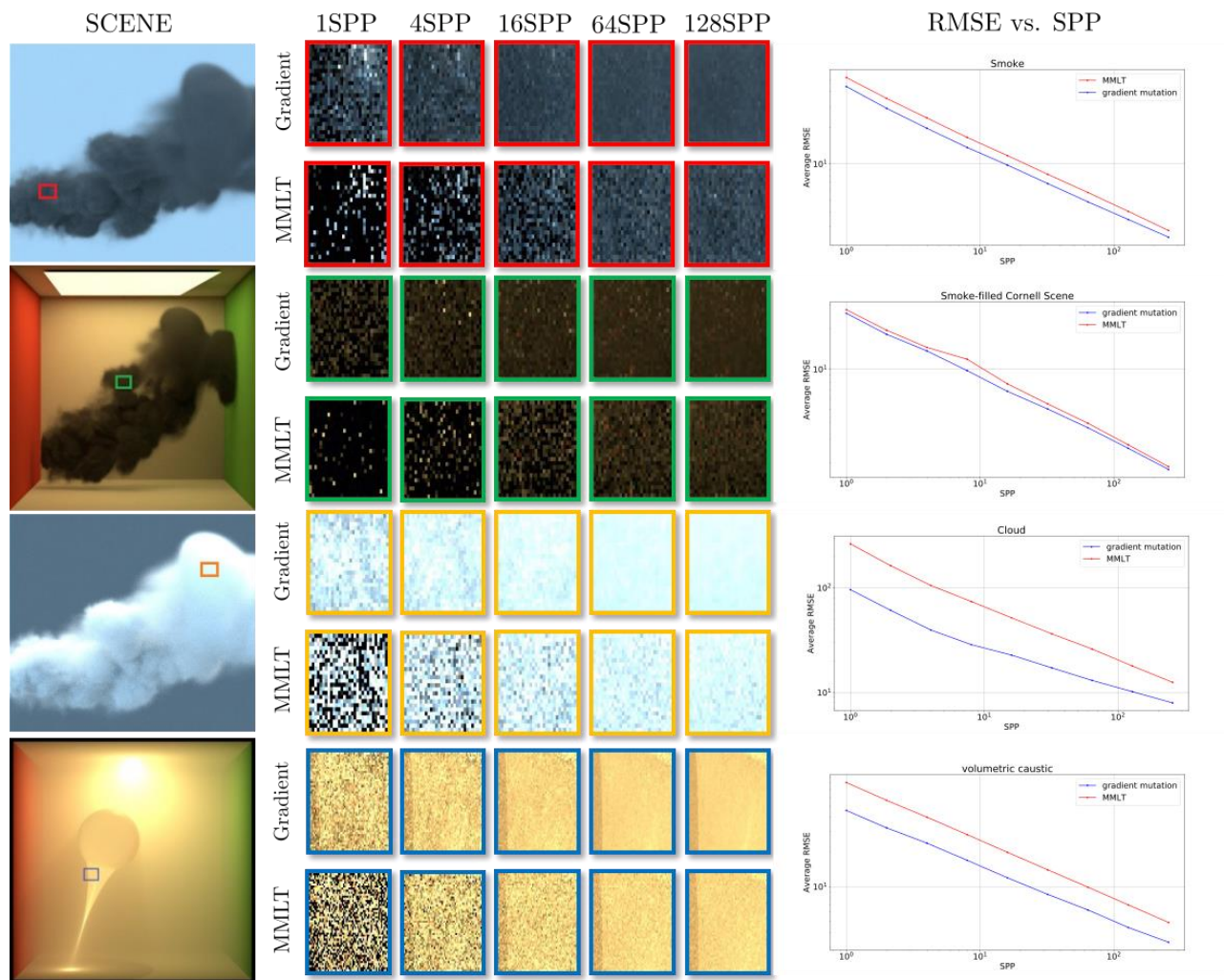


陆国伟

项目集

硕士毕业论文

Gradient-Domain Volume Rendering



我的硕士论文 'Gradient-Domain Volume Rendering' 提供一种思路，可以计算一条光路相对Volume中的一个点的梯度，并在MLT中利用该梯度快速估算临近光路的贡献值。这样省去了重新构建光路并计算光路贡献值的计算量，提高了渲染的效率。目前筹备投稿Computer Graphics Forum。

Small project

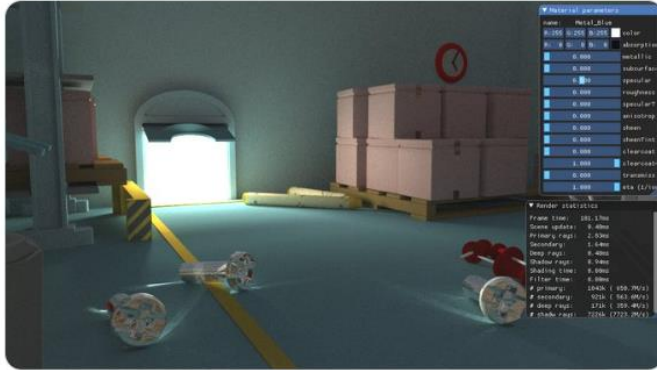
Streaming BDPT based on Light House 2



Jacco Bikker
@j_bikker

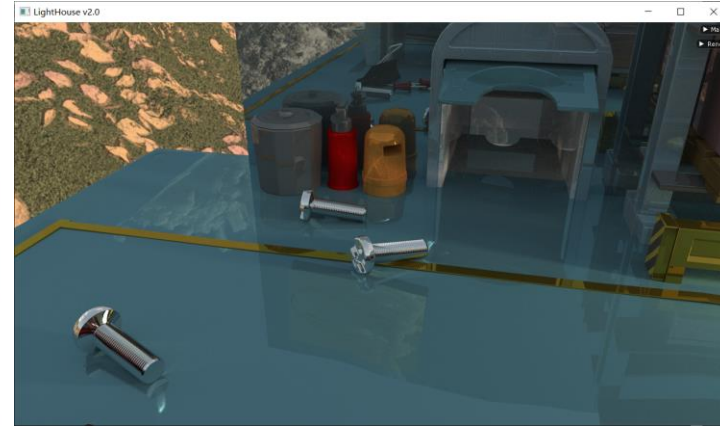
Lighthouse 2 now features a BDPT core, built by Guowei (Peter) Lu github.com/pasu. The new core is now available on Github:

github.com/jbikker/lighth... . Also in the latest version: improvements to the SVGF filter, the reference core and the reference (Lambert) material.



10:39 AM · Nov 14, 2019 · Twitter Web App

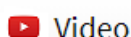
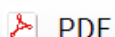
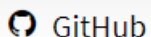
12 Retweets 47 Likes



Light House 2 是一个实时光线追踪渲染框架。我实现了基于GPU的双向路径追踪, 该渲染引擎基于wavefront管线, 使用OptiX框架, 支持能量守恒和光蚀等光学规则。

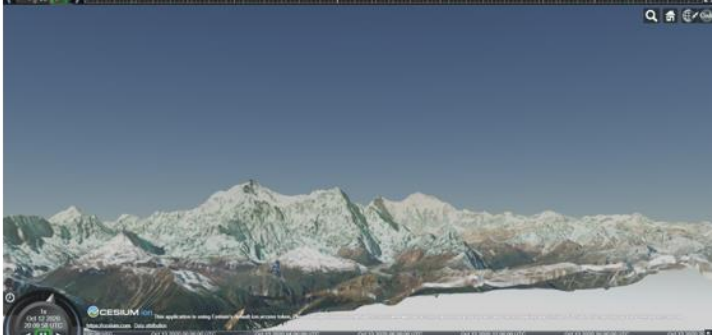
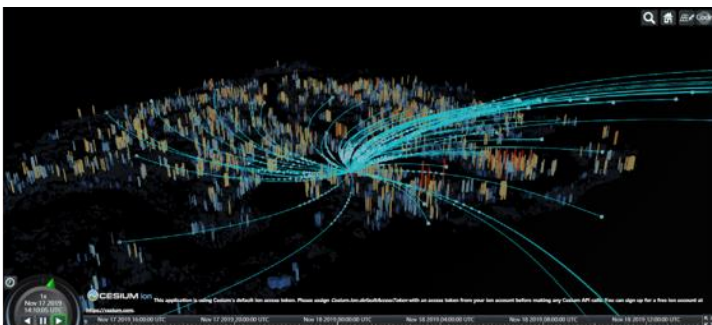
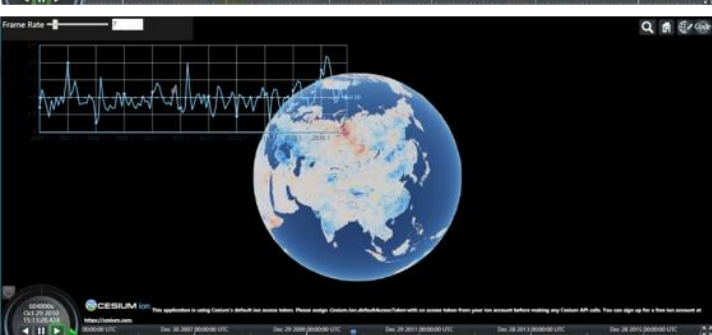
主要技术点:

- C++
- CUDA
- OptiX
- Wavefront
- 批次射线的可见性判断



Cesium教程&示例

JS/WebGL



个人业余时间完成了Cesium教程系列，并实现了一些Cesium功能的扩展（支持ArcGIS地形数据，MapBox矢量数据等）和可视化范例（全球AQI实时展示，动态流动线，近十年全球气温变化等）。很多Cesium开发者阅读过该系列教程和范例。

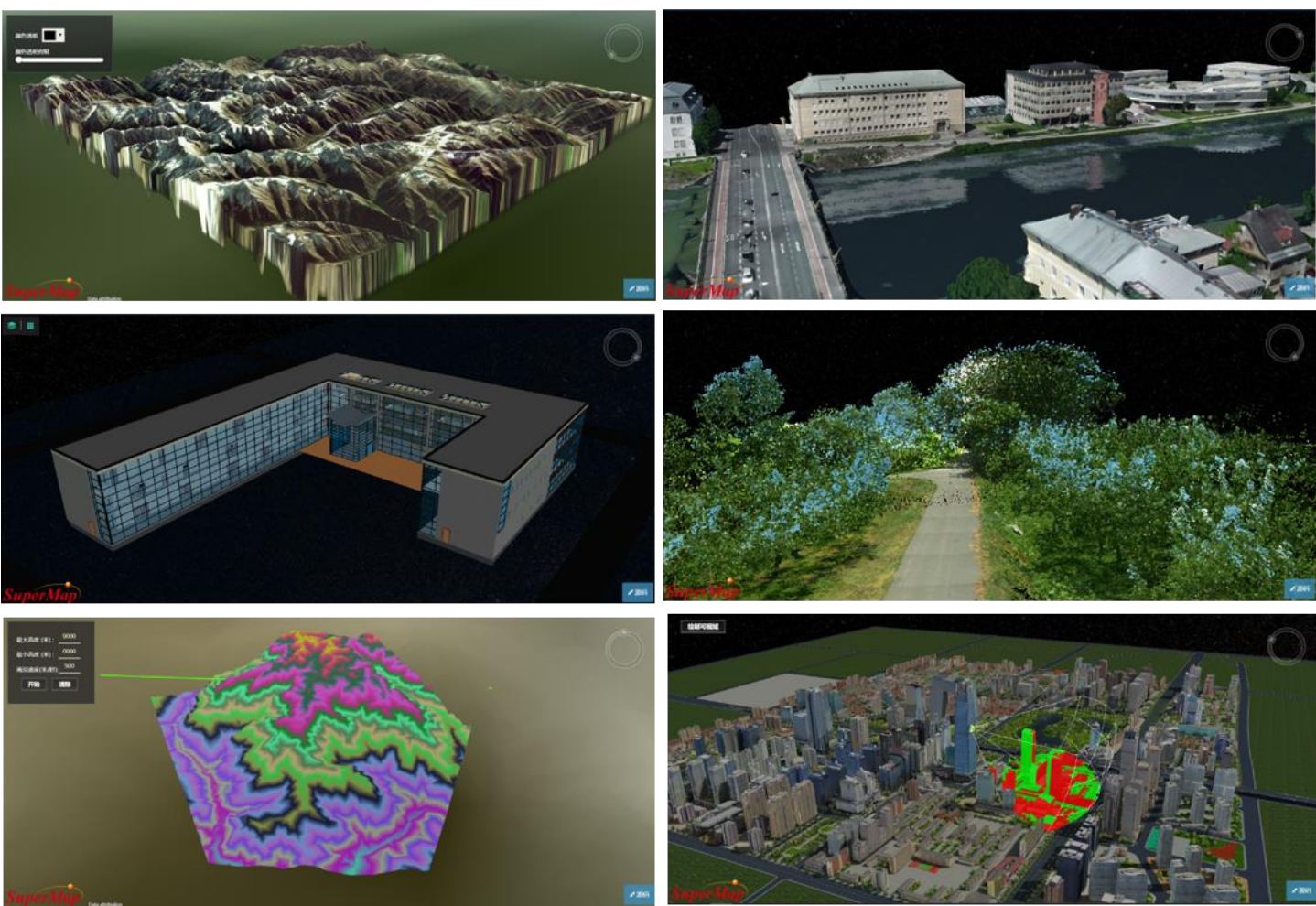
教程

GitHub

Video

S3M

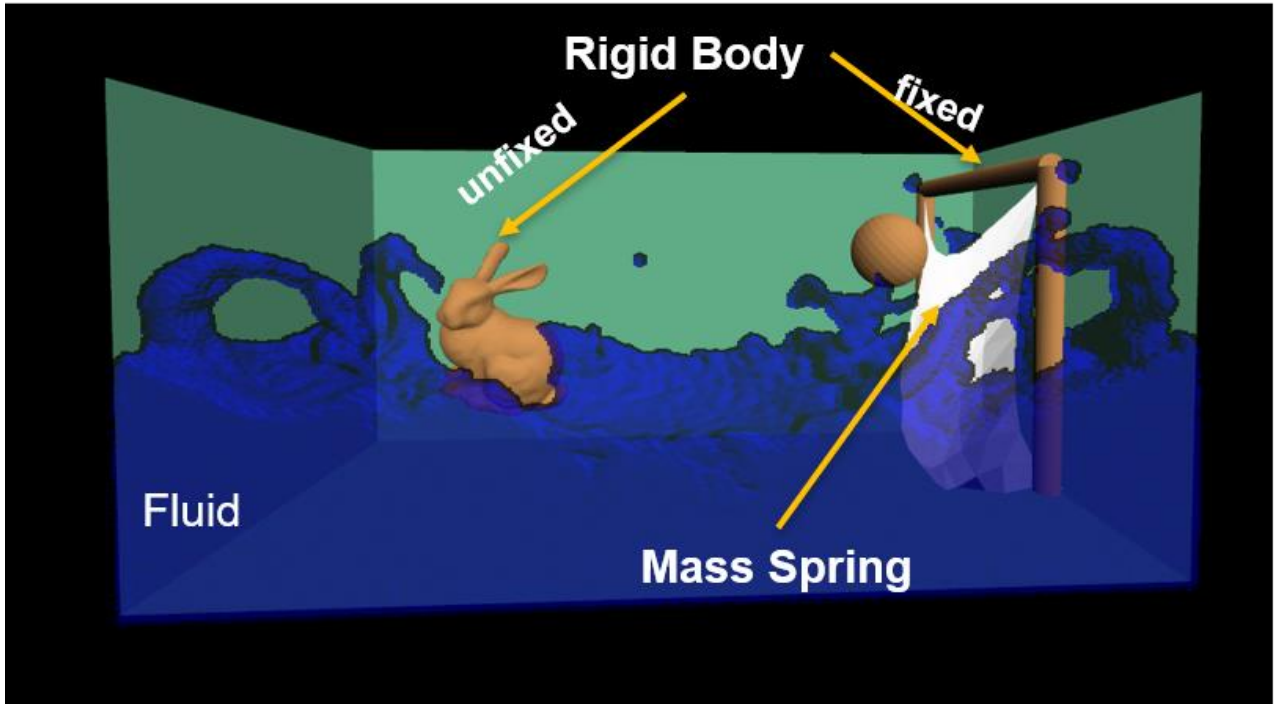
C++/JS/WebGL



S3M (Spatial 3D Model)是一种开放式、可扩展的三维地理空间数据格式，为海量多源异构空间三维模型数据（倾斜摄影，BIM，点云等）在不同终端之间的传输、交换与共享提供数据格式的具体规范，是中国地理信息产业协会发布的第一个团体标准。本人在职期间是该格式的主要设计者之一，并承担了数据生成，加载和渲染等功能的开发。

Position Based Fluid Simulation

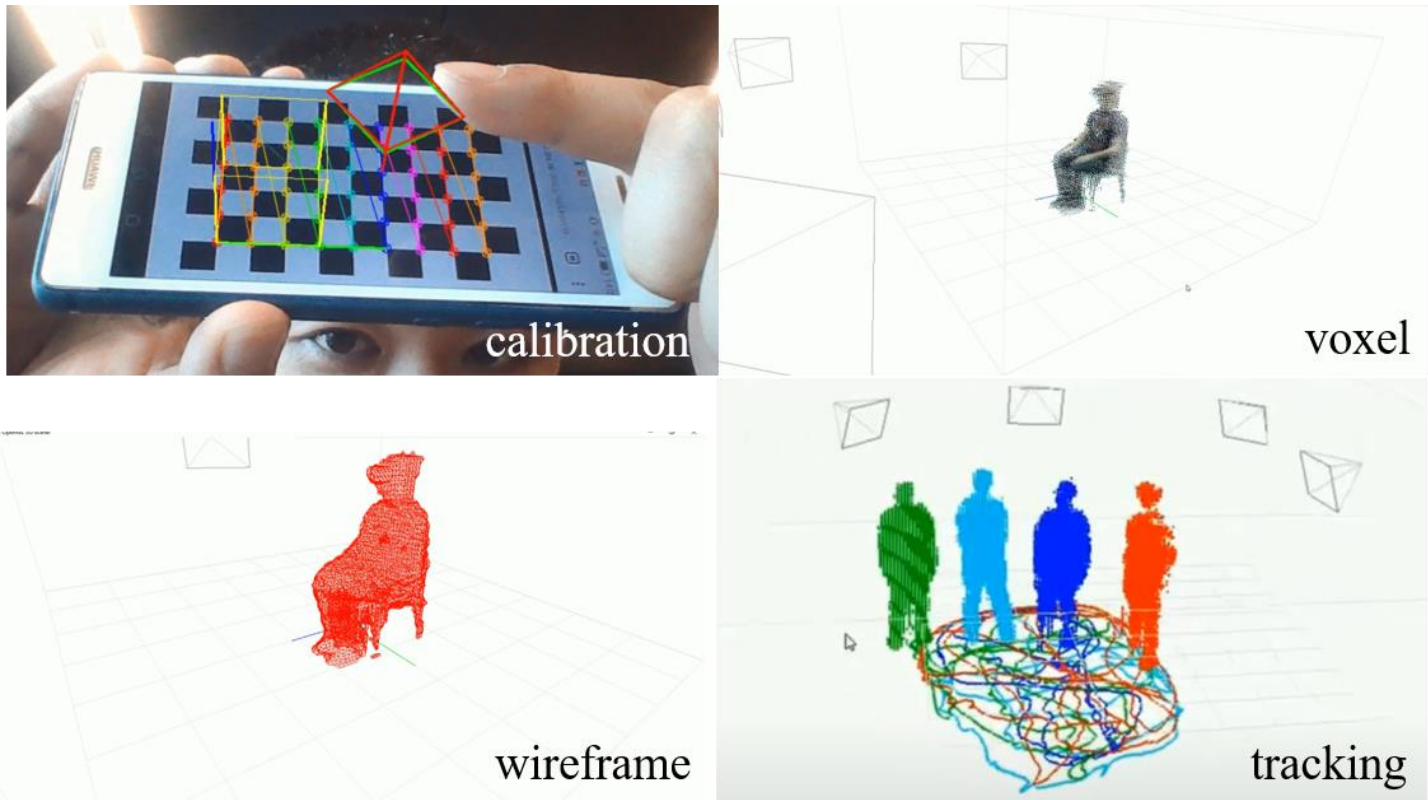
C++/Compute shader/OpenGL



Particals	Time Per Iteration	Frame Rate
64K	1.3ms	110fps
128K	2.0ms	70fps


这是'Game Physics'课程的迷你项目，并入选该课程的名人堂 (第二名)。主要思路是基于流体的密度不变特性。它包括流体和刚体，流体和布料以及流体表面重构（将流体粒子渲染成水面）。

Voxel-based 3D Reconstruction and Tracking C++/OpenCV



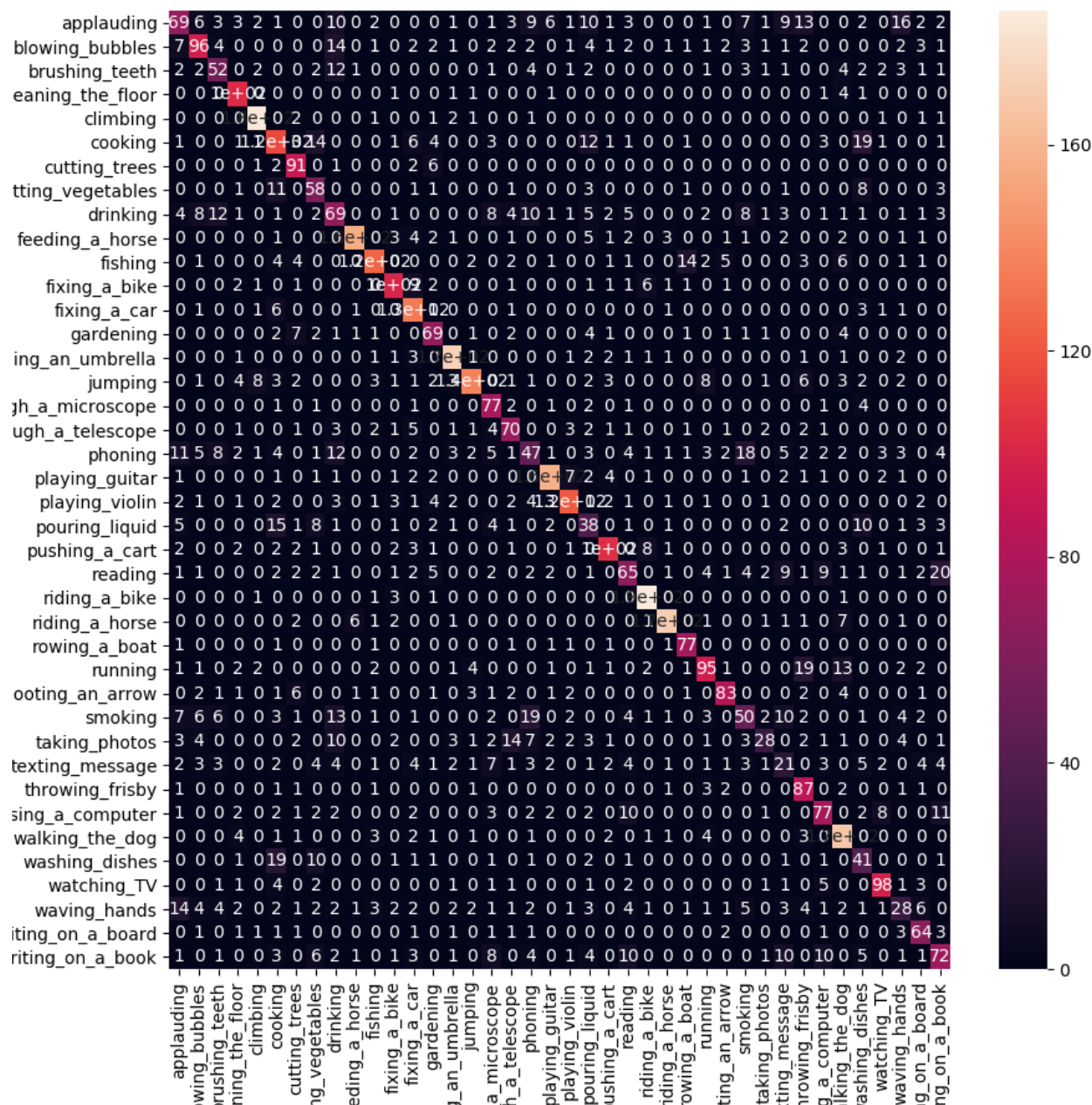
‘Computer Vision’的作业，包括相机纠偏，基于体素的三维重构和对象跟踪。主要技术包括：

- Marching cube
- K-means

 Video

Action Recognition with Automatic Model Search

Python/Keras/Tensorflow



‘Computer Vision’的课程项目。通过创建一个CNN系统，可以识别Stanford-40图像数据库中的人类行为（钓鱼，骑车等），该神经网络使用了迁移学习，数据增强，weight decay和自定义学习效率。

路径追踪(GPU&WebGPU)

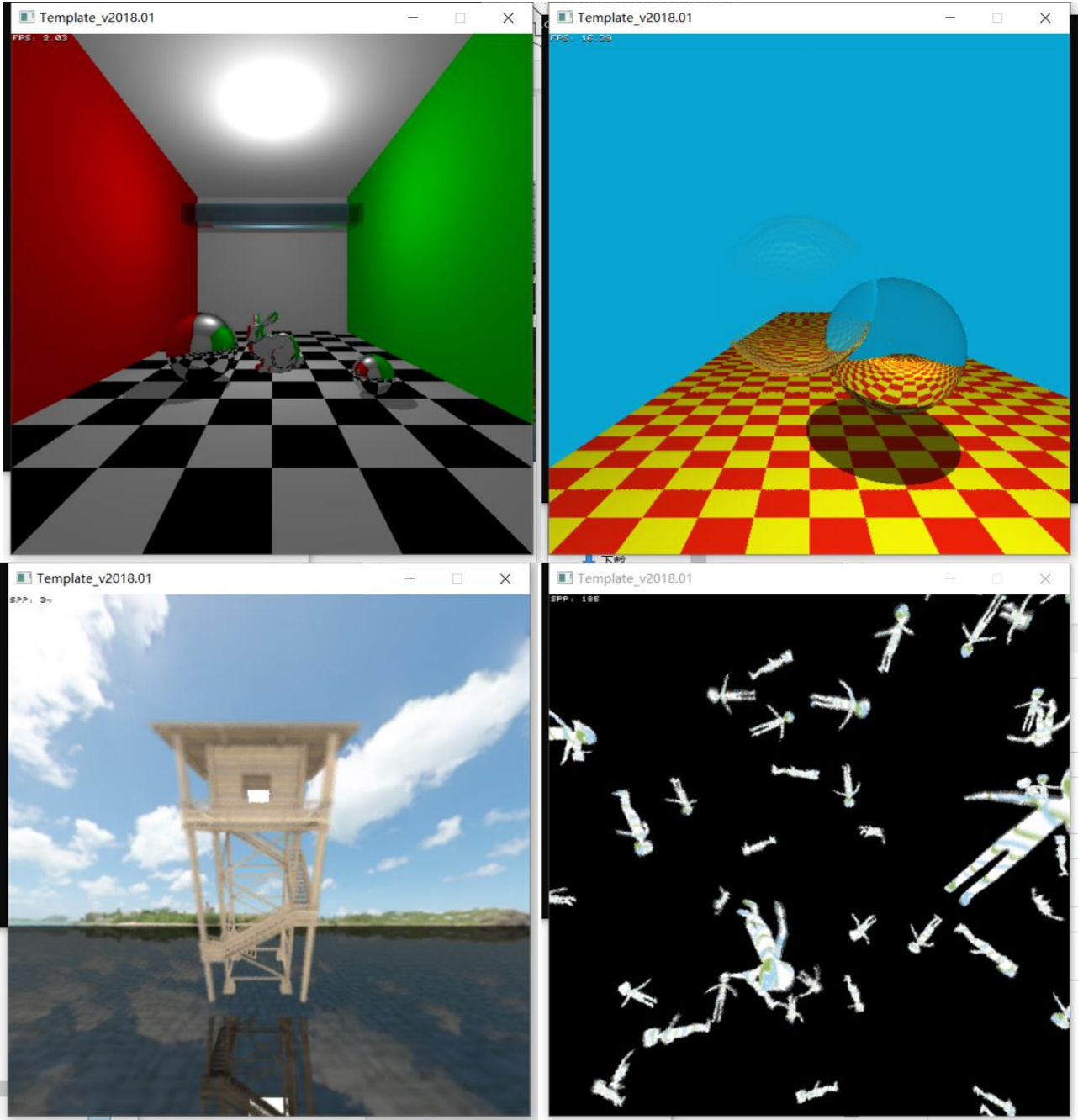
C++/JS/Compute Shader



个人实现的路径追踪框架，主要语言是 C++，JS和compute shaders. 用户可以在桌面端构建BVH并实时交互式浏览，并且可以在浏览器端加载该BVH并浏览（浏览器需支持compute shader）。它支持wavefront和megakernel。

Path Tracer (CPU)

C++



该路径追踪是‘Advanced Graphics’课程的作业。主要功能是：

- BVH (SAH + Top-Level + SIMD Intersection)
- NEE+MIS, Photon Mapping (Simple), Ray Packets
- Depth Field, Motion Blur
- Multithreading, Filter

刚体和软体碰撞模拟

C++



这是'Game Physics'课程作业，它包括以下功能：

- 刚体碰撞
 - 多刚体之间的碰撞和穿插
 - 基于动量守恒的碰撞算法
- 有约束的碰撞
- 基于有限元的软体变形

刚体碰撞：

 [GitHub](#)

有约束的碰撞：

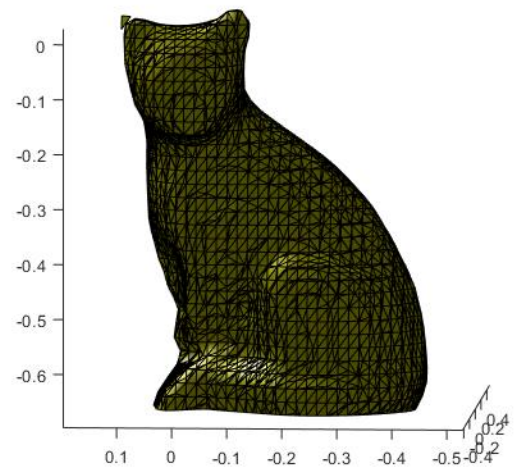
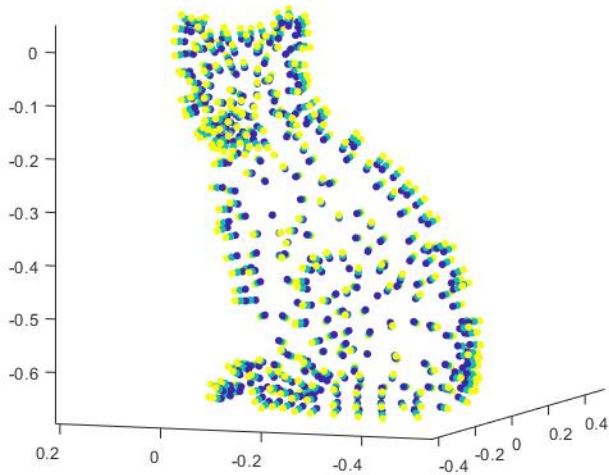
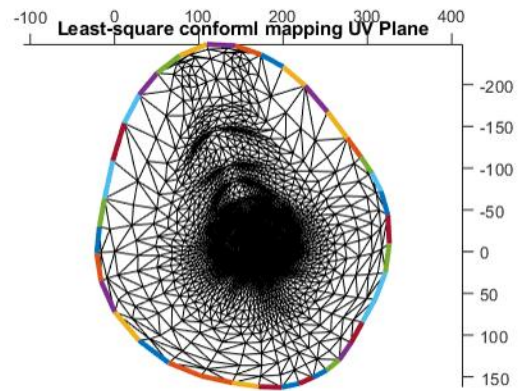
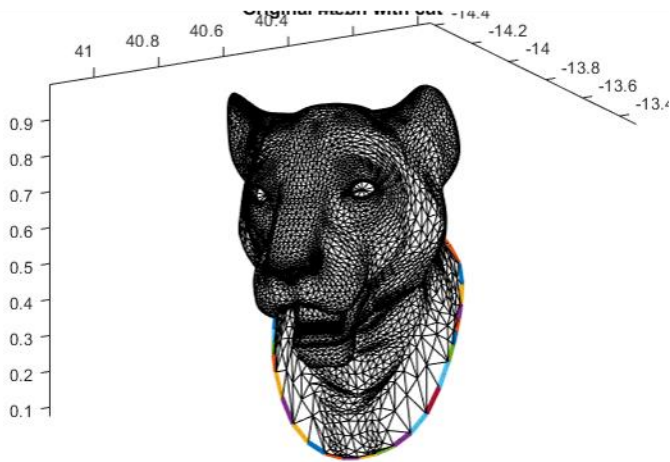
 [GitHub](#)

软体形变：

 [GitHub](#)

 [Video](#)

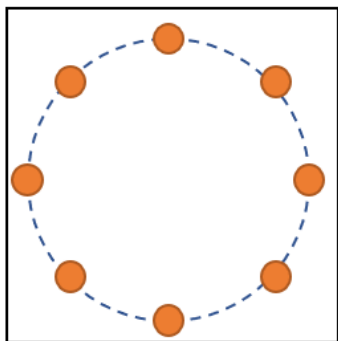
3D Model matlab



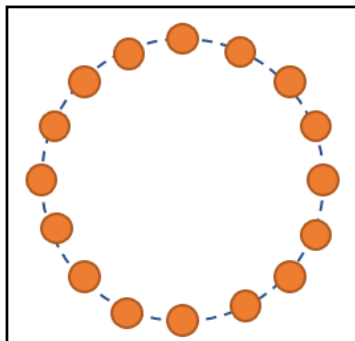
我自学了'3D model'课程并完成了课程作业，包括通过稀疏矩阵的linear least-squares系统，简易的moving least squares和Least-Squares Conformal Mapping算法。**学术诚信：**在完成Least-Squares Conformal Mapping项目时我参考了相关代码，因此严格讲，这部分的代码不是本人独立完成。

A Comparative Study of Collision Avoidance Algorithms

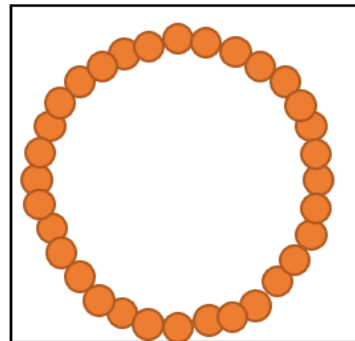
C++



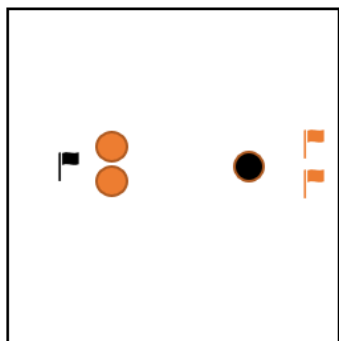
8-way-confusion



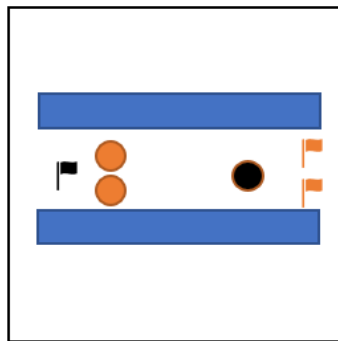
16-way-confusion



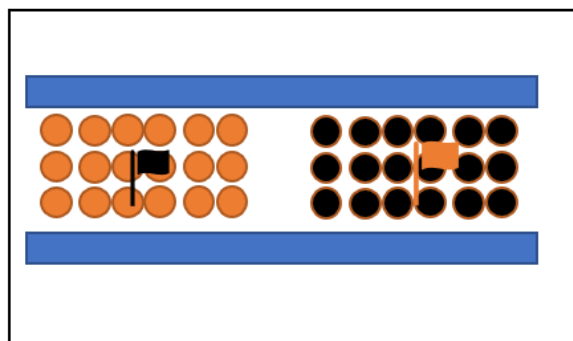
32-way-confusion



crossing



Squeeze crossing



Group intersection



Large scale scenario

‘Crowd Simulation’课程项目，我在 UUCS(Utrecht University Crowd Simulation)框架中添加了*implicit crowd*算法。UUCS是一个闭源项目。我在几个测试场景下对比了该算法和其他算法的效果。



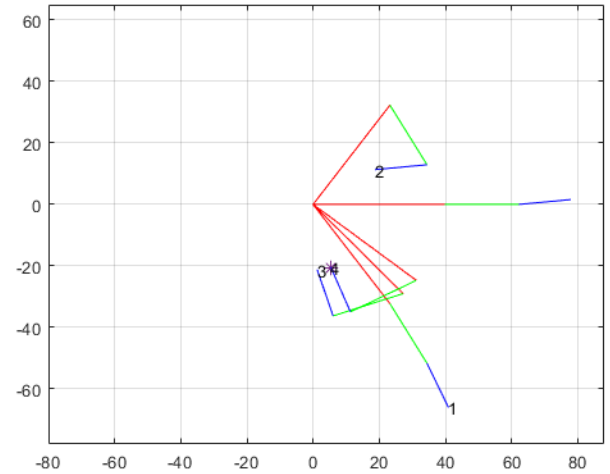
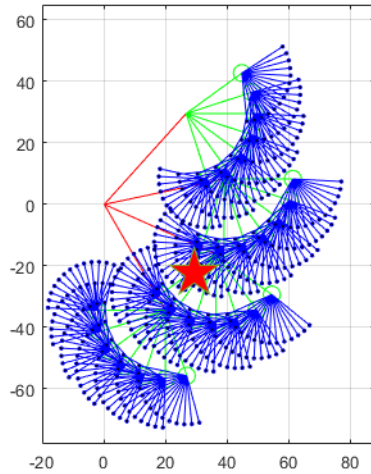
PDF



Video

Inverse Kinematics for Human Fingers

matlab

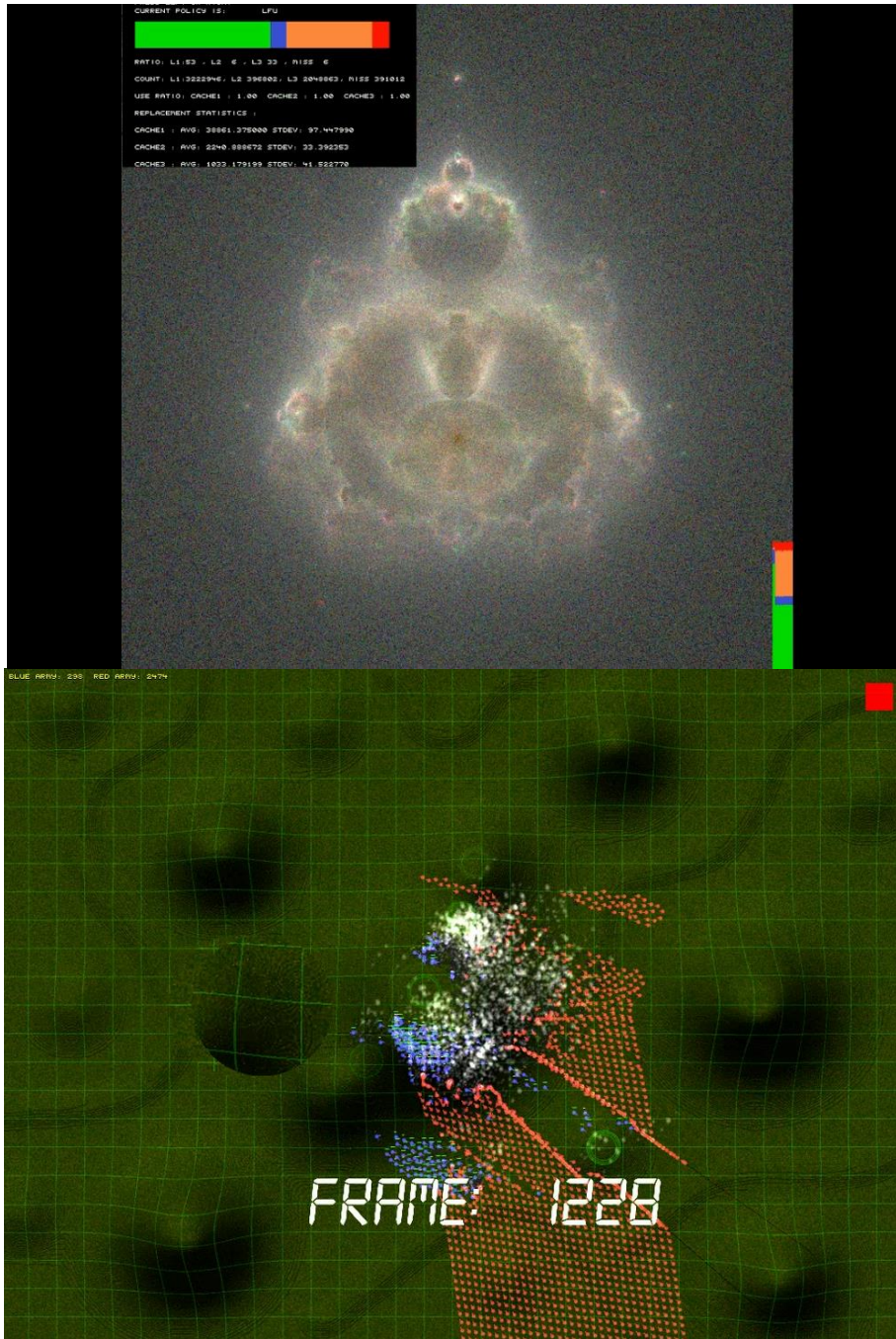


‘Motion and Manipulation’项目。首先，我模拟了2D场景下手指关节的位置以及指尖可以抵达的范围。然后考虑关节之间的约束条件，反算三个关节之间的角度，从而能够让指尖抵达指定位置。我们提供了三个算法：

- Pseudo Inverse
- Pseudo Inverse with Optimization Derivation
- Extended Jacobian Method

CPU Caching Simulator and Optimization

C++



‘Optimization and Vectorization’课程作业。CPU缓存模拟器，支持三级缓存以及退出策略，并实时检测缓存命中率。最后，利用课程所学知识，优化一个坦克游戏，其性能提高了25倍。