

# 田一丁

📞 +1 267-249-1202 📩 tonytg@seas.upenn.edu 🏠 <https://tonyxtian.com>

## 🎓 教育经历

|  |                   |
|--|-------------------|
| University of Pennsylvania 宾夕法尼亚大学                         | 2023.08 – 2027.05 |
| Computer Engineering, BSE 计算机工程, 本科                        |                   |
| Computer Graphics, MSE 计算机图形, 加速硕士                         |                   |
| • GPA: 3.78  |                   |
| • 相关课程: GPU编程, 3D建模, 计算机动画, UI/UX设计, 机器学习, 操作系统实现, 数据结构与算法 |                   |

## 🔧 项目经历

|   |                   |
|---|-------------------|
| MatForge - 高级材质渲染系统   Vulkan, Slang, C++  | 2025.11 – 2025.12 |
| • 设计生产级GPU路径追踪器, 在统一Vulkan光线追踪管线中实现四篇SIGGRAPH论文(2023-2024), 贡献2400+行C++和Slang代码用于采样和几何系统  |                   |
| • 实现SIGGRAPH 2024的四元优化低差异序列(QOLDS), 设计47维Owen扰动的Base-3 Sobol生成器, 在512 SPP下相比传统PCG采样提升+2.57 dB PSNR, MSE降低44.7%, 性能开销<1%   |                   |
| • 开发RMIP (矩形最小最大图像金字塔) 相交着色器, 通过分层纹理空间光线遍历和自定义Vulkan相交着色器实现无需细分的位移贴图  |                   |
| • 将技术集成到完整的蒙特卡洛路径追踪管线, 支持glTF 2.0场景、HDR环境贴图和KHR材料扩展   |                   |
| • 项目仓库: <a href="https://github.com/MatForge/MatForge">https://github.com/MatForge/MatForge</a>   |                   |
| Vulkan草地渲染器 - 基于物理的草地模拟   Vulkan, GLSL, C++   | 2025.10 – 2025.11 |
| • 在Vulkan中实现实时草地渲染系统, 使用计算和曲面细分管线以交互帧率模拟多达100万根草叶的物理动画  |                   |
| • 开发GPU计算着色器进行物理模拟, 对二次贝塞尔曲线草叶应用重力、胡克定律恢复力和带空间湍流的风力, 每帧处理100万+草叶状态更新  |                   |
| • 设计三级GPU剔除系统 (朝向、视锥、基于距离的概率剔除), 在100万草叶时通过高效原子操作和间接绘制调用实现4.31倍性能提升   |                   |
| • 构建硬件曲面细分管线, 基于相机距离动态LOD, 从10到2细分的指数衰减生成平滑草叶几何体  |                   |
| • 项目仓库: <a href="https://github.com/tonytgrt/Vulkan-Grass-Renderer">https://github.com/tonytgrt/Vulkan-Grass-Renderer</a>   |                   |
| CUDA路径追踪器 - 3D PBR渲染器   CUDA, GLSL, C++   | 2025.09 – 2025.10 |
| • 蒙特卡洛路径追踪器, 能够渲染具有自定义3D模型和环境贴图的复杂3D场景  |                   |
| • 实现支持全局光照、多重重要性采样、抗锯齿、次表面散射的着色BSDF内核, 能够渲染各种带反照率和纹理贴图的PBR材质类型  |                   |
| • 集成tinyGLTF第三方库支持glTF 2.0网格加载, 以及Nvidia OptiX用于降噪  |                   |
| • 使用多种技术提升性能: 材质排序(+5%)、俄罗斯轮盘(+6% - 24%)、流压缩(+24% - 67%)、层次包围体(复杂场景3倍 - 160倍帧率)   |                   |
| • 项目仓库: <a href="https://github.com/tonytgrt/CUDA-Path-Tracer">https://github.com/tonytgrt/CUDA-Path-Tracer</a>   |                   |
| WebGPU渲染器 - Web实时渲染器   WebGPU, TypeScript, WGS  | 2025.09 – 2025.10 |
| • 使用WebGPU计算和图形管线实现三种高级渲染技术用于5000+动态点光源的实时光照: 朴素前向、Forward+和聚簇延迟渲染  |                   |
| • 使用计算着色器设计屏幕空间光源聚簇系统, 将视锥体细分为16×9×24网格, 使用指数深度切片和球体-AABB相交测试进行高效光源剔除   |                   |
| • 开发具有3个渲染目标 (位置、法线、反照率) 的G-buffer架构, 实现将几何复杂度与光照计算解耦的两遍延迟渲染  |                   |
| • 相比传统渲染实现53倍性能提升 (5000光源下从497ms降至9.3ms), 通过消除过度绘制和优化内存访问模式比Forward+快3.5倍   |                   |
| • 在线演示: <a href="https://webgpu.tonyxtian.com">https://webgpu.tonyxtian.com</a> , 项目仓库: <a href="https://github.com/tonytgrt/Project4-WebGPU-Forward">https://github.com/tonytgrt/Project4-WebGPU-Forward</a> |                   |

## ⚙️ 相关技能

- 编程语言: C++, CUDA, Python, WGSL, GLSL, 并行算法, 内存管理, 渲染管线
- 图形/渲染: NSight性能分析, 路径追踪, 延迟渲染, 光栅化, 动画系统, PBR
- 工具/API: Unreal, Unity, Maya, Nvidia NSight, WebGPU, Vulkan, Visual Studio, Qt, OpenGL, Git, CMake
- 语言: 英语 - 熟练 (TOEFL 110), 粤语 - 入门, 日语 - 入门