

组号:

(按汇报顺序) 第三组

论文简介:

本文介绍了一种基于 GPU 加速的矢量笔画渲染技术,并提出了一种高效的渲染方法,结合了光栅纹理的表现力和矢量图形的实时可编辑性,支持普通笔画、印章笔画和喷枪笔画等多种笔画类型。通过 GPU 加速,该技术能够在实时交互中实现复杂的笔画效果,同时保持高效的渲染性能。为了验证该技术的实际应用价值,作者开发了一个名为 Ciallo 的开源原型绘画系统,该系统集成了矢量填充功能,支持艺术家在绘制过程中自由编辑笔画形状,并在动画制作中高效地重用颜色标记。

任务内容:

研读论文,了解论文中描述的算法和项目架构以及实现方案。复现成果,通过论文作者提供的源代码获取可获取同样结果的可执行文件,并借此测试论文中描述的功能和效果。撰写汇报文案,将论文中的算法、解决问题的思路、成果效果用更通俗的文字描述出来,并融合进自己的思考,以便于论文的汇报。制作汇报 PPT,用于辅助论文汇报和成果展示。

关于基础要求:复现成果,通过论文作者提供的源代码获取可获取同样结果的可执行文件,并借此测试论文中描述的功能和效果。

关于核心要求 1:以我们对于该论文描述的计算方式和架构的理解,我们无法提出更快的算法,但是源代码中有很多从工程的角度上可以改进的地方(这篇论文的成果是想要投入生产使用的,这种要求应该合理),源代码缺少异常情况处理,这导致在我们的实际测试过程中软件出现了崩溃的情况,因此是需要加入异常处理机制的,实际生产环境要是程序异常崩溃是有可能造成大量损失的;源代码没有做多线程优化,尽管用 GPU 加速了图形可并行计算的部分,但是软件本题是没有做多线程的,这使软件核心 workflow 依然是串行的,并不能完全优化,且尤其是对于多条曲线的渲染部分也是串行的,如果做成多线程,分别去给 GPU 派发任务,理论让更能发挥 GPU 的性能,加快速度;源代码用 OpenGL 来渲染窗口,做法比较粗糙,做出的软件的操作逻辑也比较糟糕,可以使用一些框架重写交互逻辑,比如 Qt,也可以直接做成前后端分离的模式,这样也更能保证用户操作时的流畅度;还有一条,作为绘制 2D 动画的软件,没有引入图层机制和撤销/反撤销,(和对图层混合模式的修改),这是最需要优化的部分。

关于核心要求 2:我们使用控制变量法,调整算法中可调整的参数分别为 2、4、...、512 这些 2 的幂数(软件设计最高 512),以此来绘制不同的曲线来测试绘制结果,并比对在 2K 分辨率下从哪个值开始继续增大参数(提高计算精度)也不会增强效果(结论是在 250 左右);我们通过多种极端情况实验,测试软件对于曲线的识别、编辑边界,以及修改难易度(结论是识别效果并不好,仅对近似标准的贝塞尔曲线效果较好,只要曲线稍微有一点随机就会存在识别不了曲线端点的情况,且如果对曲线进行编辑,因为没有图层机制如果掩盖了其他曲线可能会导致信息缺失,另外两个曲线重合起来的话是有可能认为其中一条曲线被截断的,修改的难易度并不低,体感上对于无法熟练使用 Photoshop 的钢笔工具的人来说(比如我们两个)操作逻辑是难以适应的);我们还通过多次随机实验(随机画线)测试其填充功能的稳定性,(结论是并不稳定,尤其是曲线有重合时,有时会将两条曲线分别计算,有时会将重叠的两个曲线的两侧部分合为一条曲线计算)

团队分工:

由张宗卓和宋浩宇一起完成论文研读

由张宗卓完成论文汇报文案撰写

由宋浩宇完成论文代码成果复现
由张宗卓和宋浩宇一同完成汇报 PPT 制作