

基于 CUDA 的并行快速铅笔画生成算法研究

汇报人：刘朝洋

指导老师：刘斌

信息工程学院

西北农林科技大学
NORTHWEST A&F UNIVERSITY

中国 · 杨凌



主要内容

| 目录

1. 研究背景

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望

2. 研究内容

3. 实验结果与分析

4. 总结与展望



研究背景

2

- ▶ 铅笔画生成算法被广泛应用在图像处理、游戏、动画等领域；
- ▶ 目前大部分铅笔画生成算法是基于 CPU 实现的，其面临的主要问题是很难实现铅笔画风格的实时渲染；
- ▶ 基于 GPU 的并行计算技术的不断发展，使得铅笔画风格的实时渲染成为可能。



OpenCL

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

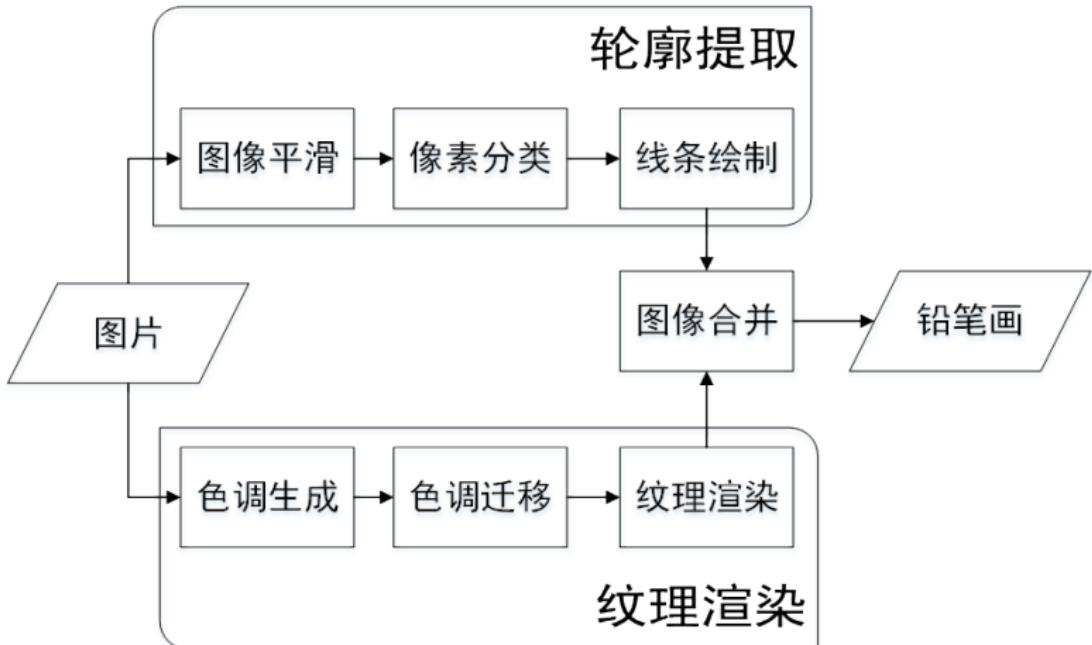
算法效果

定量分析

运行时间与加速比

总结与展望

3

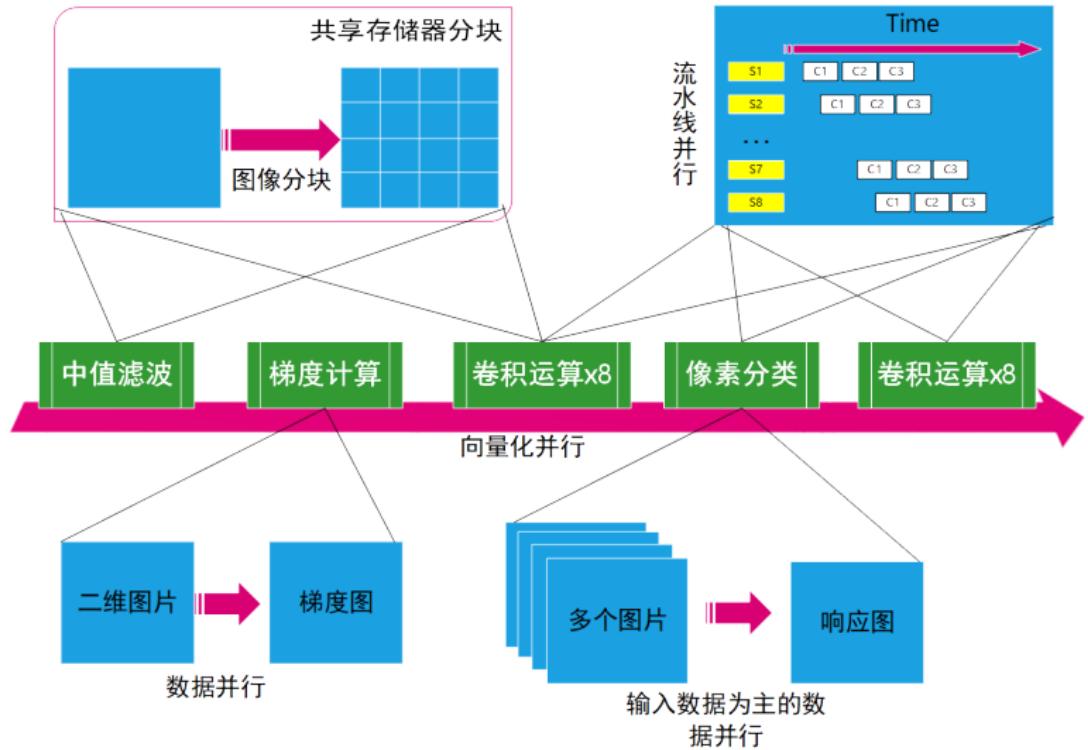


15

研究背景
研究内容
轮廓提取
纹理渲染
任务级并行
并行实现

实验结果与分析
算法效果
定量分析
运行时间与加速比
总结与展望

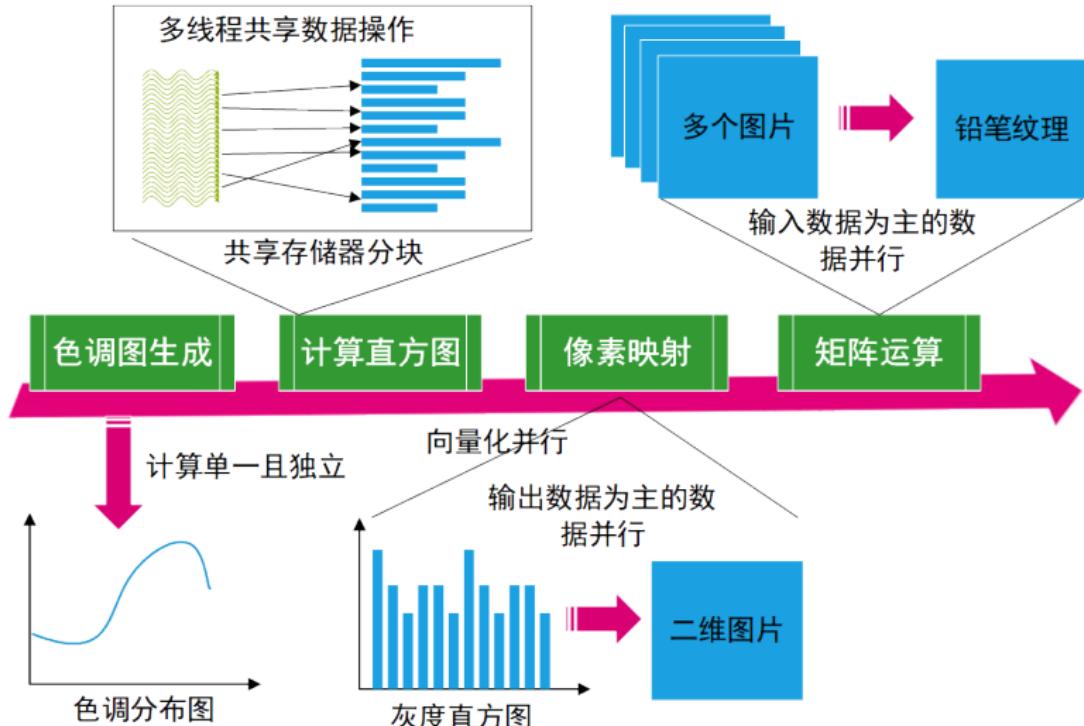
4



研究背景
研究内容
轮廓提取
纹理渲染
任务级并行
并行实现

实验结果与分析
算法效果
定量分析
运行时间与加速比
总结与展望

5



研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望

6

常量存储器

1	0	0
0	1	0
0	0	1

⊗ 加权求和

1	2	3
1	2	3
1	2	3



CUDA设备

Grid 1



Block (1,1,0)

Thread (0,0,0)	Thread (1,0,0)	Thread (2,0,0)	Thread (3,0,0)	Thread (4,0,0)
Thread (0,1,0)	Thread (1,1,0)	Thread (2,1,0)	Thread (3,1,0)	Thread (4,1,0)

共享存储器

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

↑ 图像分块

二维图片

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

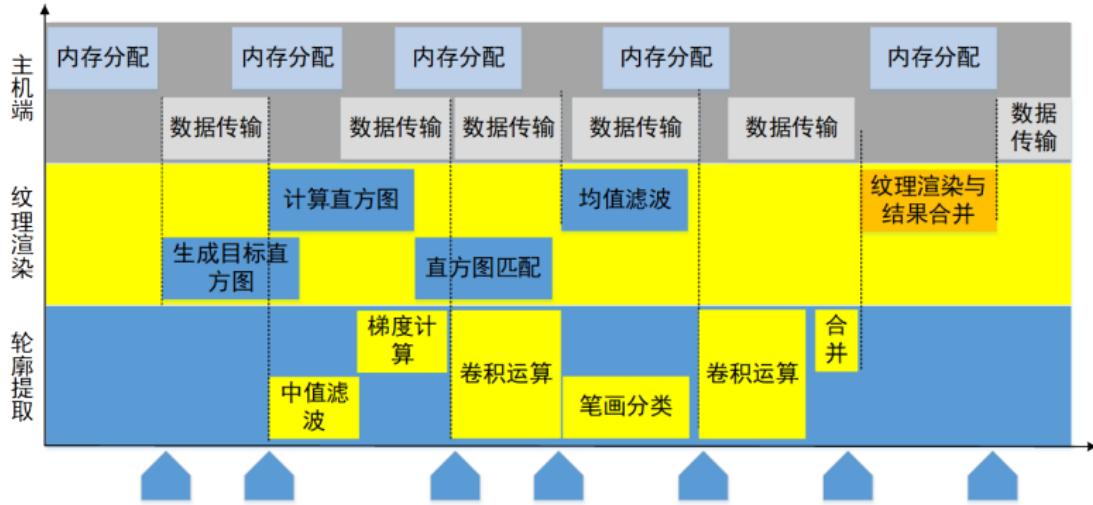
算法效果

定量分析

运行时间与加速比

总结与展望

7



说明

使用 CUDA 流技术，对无依赖性的数据计算和数据传输任务重新调度，以实现任务的重叠执行。



代码仓库

| 并行实现

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

8

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望

Fibird / cuda_pencildraw

Watch 0

Star 0

Fork 0

Code

Issues 2

Pull requests 0

Projects 0

Wiki

Insights

Settings

No description, website, or topics provided.

Edit

Add topics

Cuda 81.2%

C++ 8.8%

CMake 5.6%

C 4.4%

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 Fibird add test files

Latest commit b5978c3 a day ago

inputs

add test files

a day ago

pencil

add test files

a day ago

src

add test files

a day ago

test

add test files

a day ago

.gitignore

do a little change

a month ago

CMakeLists.txt

refactor pencil_drawing project

2 months ago

README.md

First commit

5 months ago

url

https://github.com/Fibird/cuda_pencildraw



研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

9

算法效果

定量分析

运行时间与加速比

总结与展望



图: 水波潋滟



图: 枯藤老树

注

图片来自香港中文大学铅笔画生成算法测试数据集，由于篇幅原因这里只展示两张图片的实验结果。

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

10



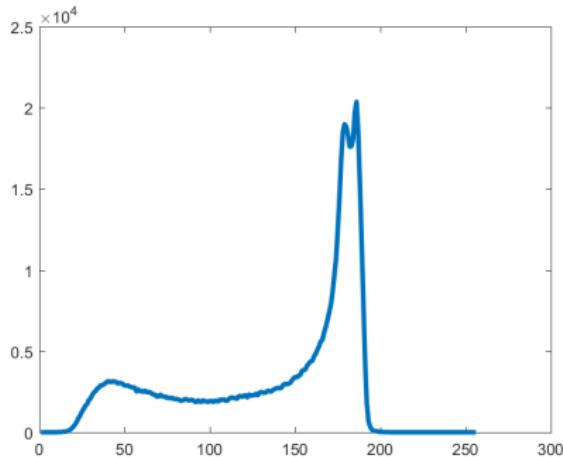
图：“水波潋滟”铅笔画结果



图：“枯藤老树”铅笔画结果

结论

处理结果的线条富于变化，具有线条交叉的效果，而且纹理细腻，表现出了铅笔画的光线与阴影效果，比较接近真实的铅笔画。



图：实际铅笔画色调分布图

Li Cewu 等人通过对实际铅笔画的大量分析，建立了实际铅笔画的色调分布模型，如上图所示为该模型的色调分布图。

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

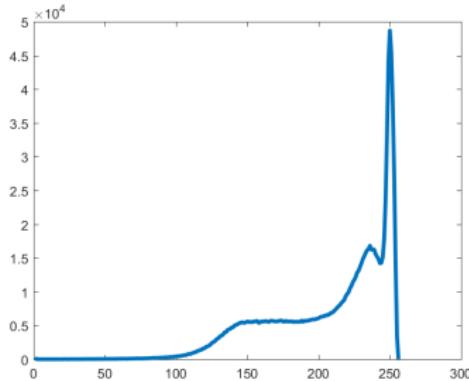
算法效果

定量分析

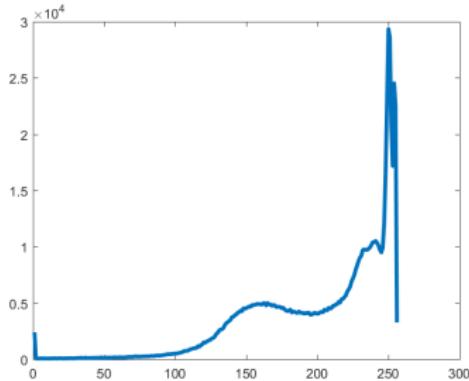
运行时间与加速比

总结与展望

12



图：“水波潋滟”铅笔画色调分布图



图：“枯藤老树”铅笔画色调分布图

结论

本算法生成的铅笔画的色调分布图和实际铅笔画色调分布图的大致相同，因此可以得出本算法生成的铅笔画与实际铅笔画十分接近。

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望

13

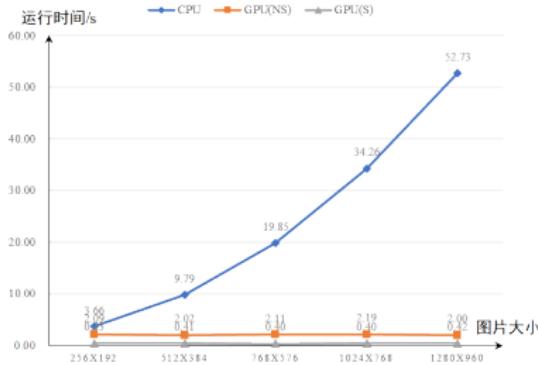


图: 图“水波潋滟”实验结果

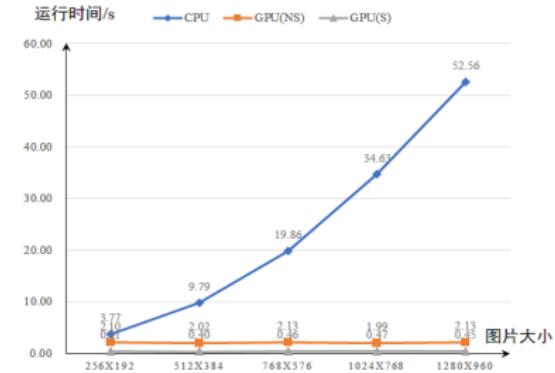


图: 图“枯藤老树”实验结果

结论

GPU 并行程序的处理速度远快于 CPU 串行程序，加速比达 10 以上，而且运行时间稳定，不受图片大小的影响，满足实时渲染的要求。



研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望

14

表：两种 GPU 算法的平均加速比

图片大小	GPU(No-Stream)	GPU(Stream)
256x192	1.78	9.23
512x384	4.68	22.53
768x576	9.32	44.50
1024x768	16.22	80.59
1280x960	25.15	120.32

结论

使用 CUDA 流技术的 GPU 算法的加速比大于未使用 CUDA 流的算法，前者大概为后者的 5 倍左右，因此使用 CUDA 流技术可以大大加快程序运行速度。





总结

本文针对目前铅笔画生成算法实时效果差的问题，提出了一种基于 CUDA 的并行快速铅笔画生成算法，这种算法不仅可以获得逼真的铅笔画效果，而且处理速度快，满足实时渲染的要求，在游戏、动画等领域具有广泛的应用前景。

展望

目前本算法只能生成单色调铅笔画，可以将算法适当改进以实现彩色铅笔画的生成。

15

研究背景

研究内容

轮廓提取

纹理渲染

任务级并行

并行实现

实验结果与分析

算法效果

定量分析

运行时间与加速比

总结与展望



感谢您的聆听！
欢迎多提宝贵意见和建议！

西北农林科技大学
NORTHWEST A&F UNIVERSITY

中国 · 杨凌