

Project report for the CG 100433 course

Project Title

基于神经网络的水彩风格渲染

Team member

胡行健 1752896

魏鹤达 1752762

Abstract

Motivation

传统的 CG 技术无法较好地实现水彩渲染效果。

The goal

我们希望通过神经网络风格化重建图像，实现水彩风格渲染。

The content of our project

1. OpenGL 部分：预渲染，得到线稿和颜色信息。
2. 神经网络部分：水彩风格化重建图像。

Involved techniques

1. 现代 OpenGL 的编程。
2. 基础图形学。
3. 基础神经网络。

Motivation

本来目标是实现“基于 GAN 的卡通渲染”，但实施过程中，发现：

1. 没有必要使用 GAN；
2. 传统 CG 技术能较好地完成卡通渲染任务；
3. 神经网络无法得到比传统卡通渲染更好的效果；
4. 神经网络的效果有水彩风格；
5. 传统 CG 技术无法较好地实现水彩风格渲染；

因此转而研究“基于神经网络的水彩风格渲染”。

The Goal of the project

1. 实现不依赖 CV 技术、高效、高质量的实时描边。
2. 实现基于神经网络的水彩风格渲染。

The Scope of the project

图形学部分：

1. 3D 模型的加载
2. 3D 模型的制作及修改（纹理贴图、曲面细分等）
3. 冯氏光照模型的应用及改进
4. 基础的空间变换
5. 背部扩张法描边

其他：

1. 深度学习-图像处理
2. Git 项目的多人合作实践

未涉及到的图形学知识

1. 光线追踪
2. 阴影
3. 动画、骨骼绑定
4. 物理引擎

Involved CG techniques

3D 模型的加载 & 3D 模型的制作及修改

我们采用 `Assimp` 库实现 3D 模型加载。

但由于我们的 3D 模型来自互联网，普遍存在以下问题：

1. 无法加载顶点法向量
2. 无法正确读取纹理路径
3. 无法正确处理 UV 映射关系

针对问题1/2，我们首先研究了 OBJ 文件格式的存储结构；然后将网络上下载模型通过 3ds MAX 软件设置顶点法向量，之后导出为 OBJ 格式；最后修改 `Assimp` 库纹理读取相关部分，由“从 `mtl` 文件中获取纹理路径”改为“由用户传入指定的纹理路径”。

针对问题3，我们分析问题原因为，原模型的 UV 映射过于复杂；于是我们通过 3ds MAX、Photoshop 等软件，重新制作了简化版的 UV 贴图。

冯氏光照模型的应用及改进

1. 卡通渲染实验（不是水彩渲染）
设置漫反射阈值，实现了硬边阴影，具有卡通效果。

2. 阴影描边

同样通过设置漫反射阈值，实现了阴影描边。

基础的空间变换

演示中，实现了光源随摄像机一同转动的效果。

背部扩张法描边

通过将模型背面往法线方向扩展一定距离，实现轮廓描边。
优化了扩张距离的计算，使得描边均匀，具有更好的效果。

Project contents

AnimeGI 目录下

```
光照 shader:
  1.lamp.fs
  1.lamp.vs
模型正常加载 shader:
  1.model_loading.fs
  1.model_loading.vs
背部扩张法 shader:
  model_backface.fs
  model_backface.vs
阴影描边 shader / 卡通渲染 shader:
  model_shadowLine.fs
  model_shadowLine.vs
各类头文件:
  camera.h      相机变换用
  mesh.h        模型加载用
  model.h       模型加载用
  shader.h      编译 shader 用
  std_image.h   加载纹理用
主函数:
  main.cpp
```

AnimeGI_py 目录下

```
./dataset
  blur.py      用于将训练集图片缩小至 1/8 分辨率
  Datasets.py  用于将训练集相关格式载入 Tensorflow
./drawing
  AnimeGIDraw.py  加载训练好的模型，完成相关处理后返回重建图片
  SketchKeras.py  用于线稿（边缘）提取。最终方案未使用此文件
  waifu2x.py     用于卡通图像清晰化。最终方案未使用此文件
./img_dataset  用于存放训练集和测试集图片
./net
  AnimeGInet.py  用于向 Tensorflow 描述神经网络结构
  SubNet.py     用于向 AnimeGInet.py 提供底层的网络结构描述
./Training
  AnimeGIDraftModelTraining.py  用于 GAN 的训练。最终方案未使用此文件
```

<code>AnimeG1Training.py</code>	用于 <code>pix2pix</code> 网络的训练。最终方案使用此文件
<code>./Drawing.py</code>	处理从 C 获取的信息，调用 <code>AnimeG1Draw.py</code> ，并展示重建后的图片
<code>./hyperparameter.py</code>	用于存放各类超参数
<code>./train_all.py</code>	用于同时训练 GAN 和 <code>pix2pix</code> 网络。最终方案未使用此文件
<code>./train_draft.py</code>	用于训练 GAN。最终方案未使用此文件
<code>./train_draw.py</code>	用于训练 <code>pix2pix</code> 网络
<code>./utils.py</code>	封装了一些工具函数

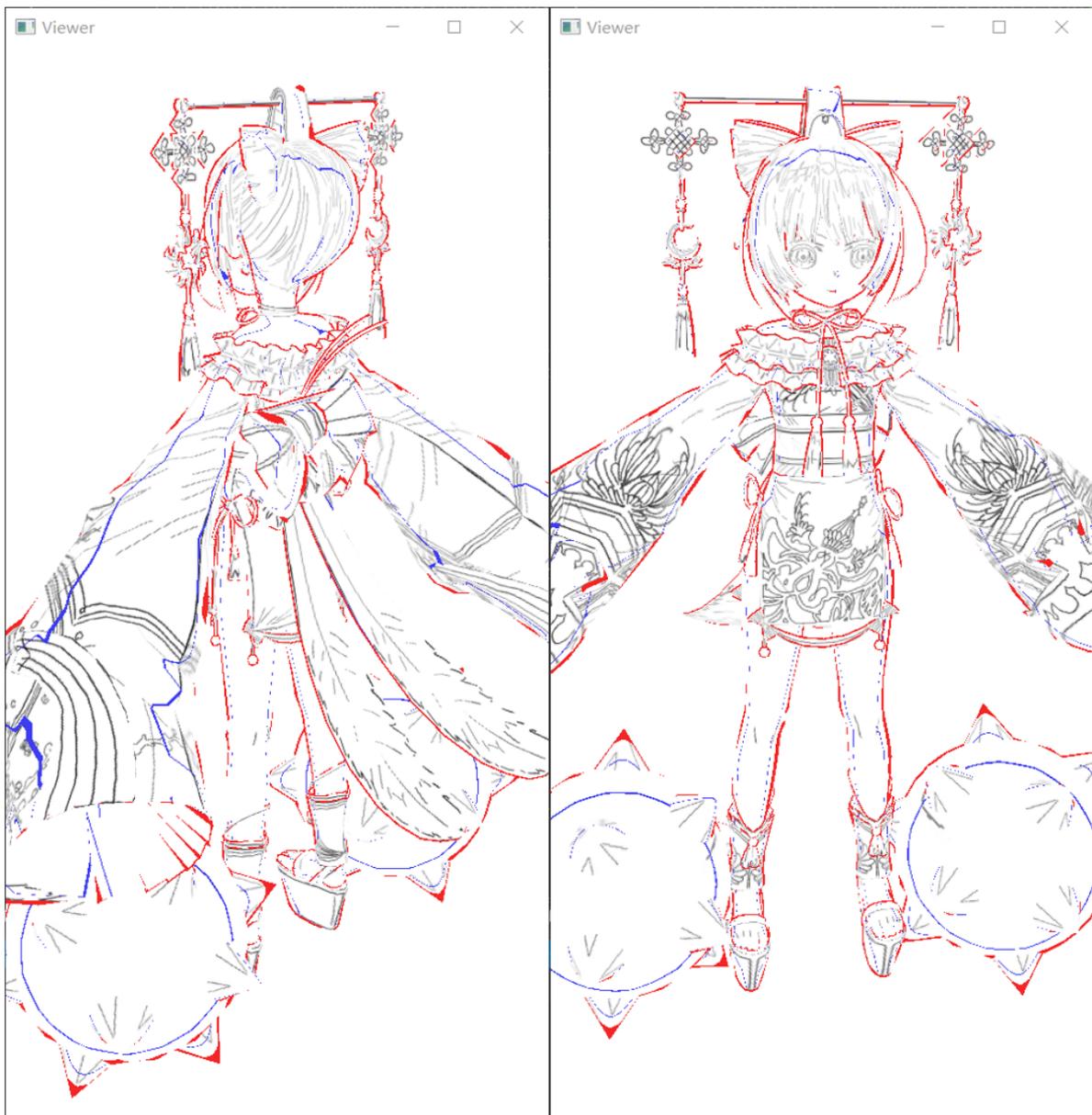
Implementation

背部扩张法描边 + 阴影描边 + 纹理描边 -> 描边预渲染

1/8 分辨率预渲染 + 描边预渲染 -> 神经网络风格化重建图像 -> 输出

Results

描边预渲染

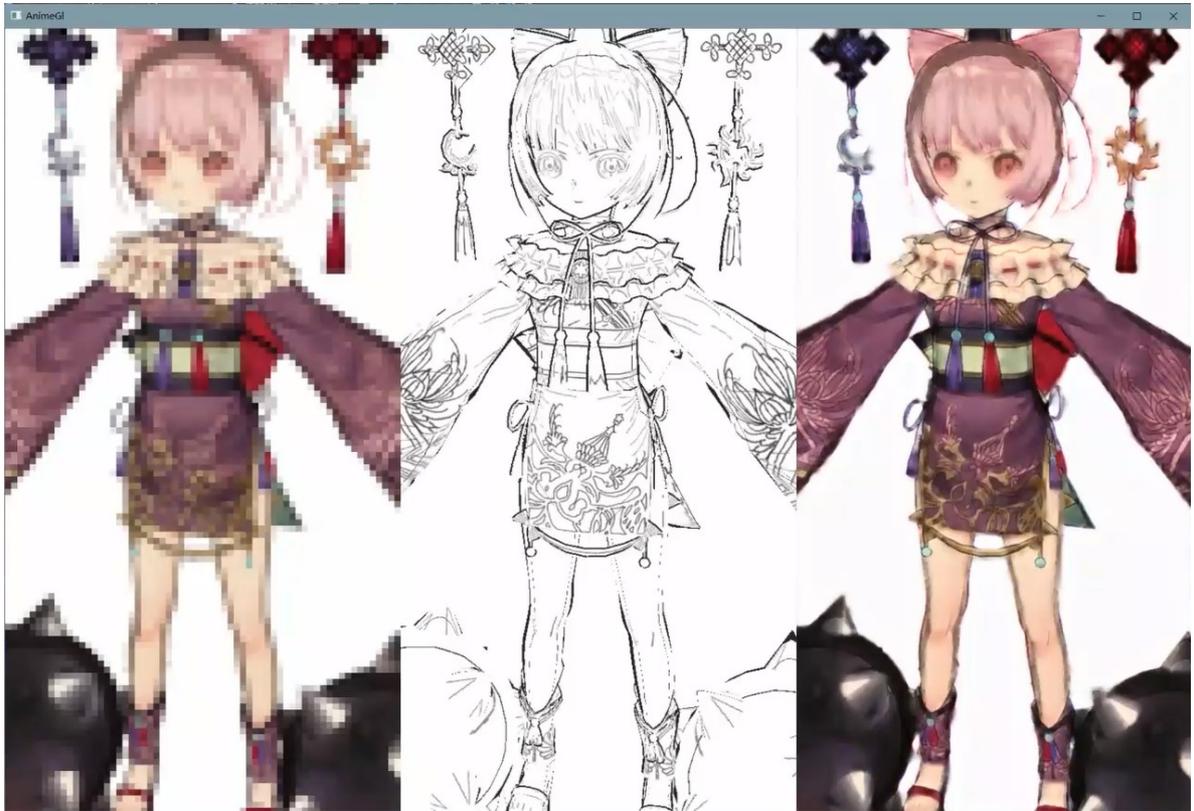


1.模型边缘 (红) 2.阴影边缘 (蓝) 3.纹理边缘 (黑)

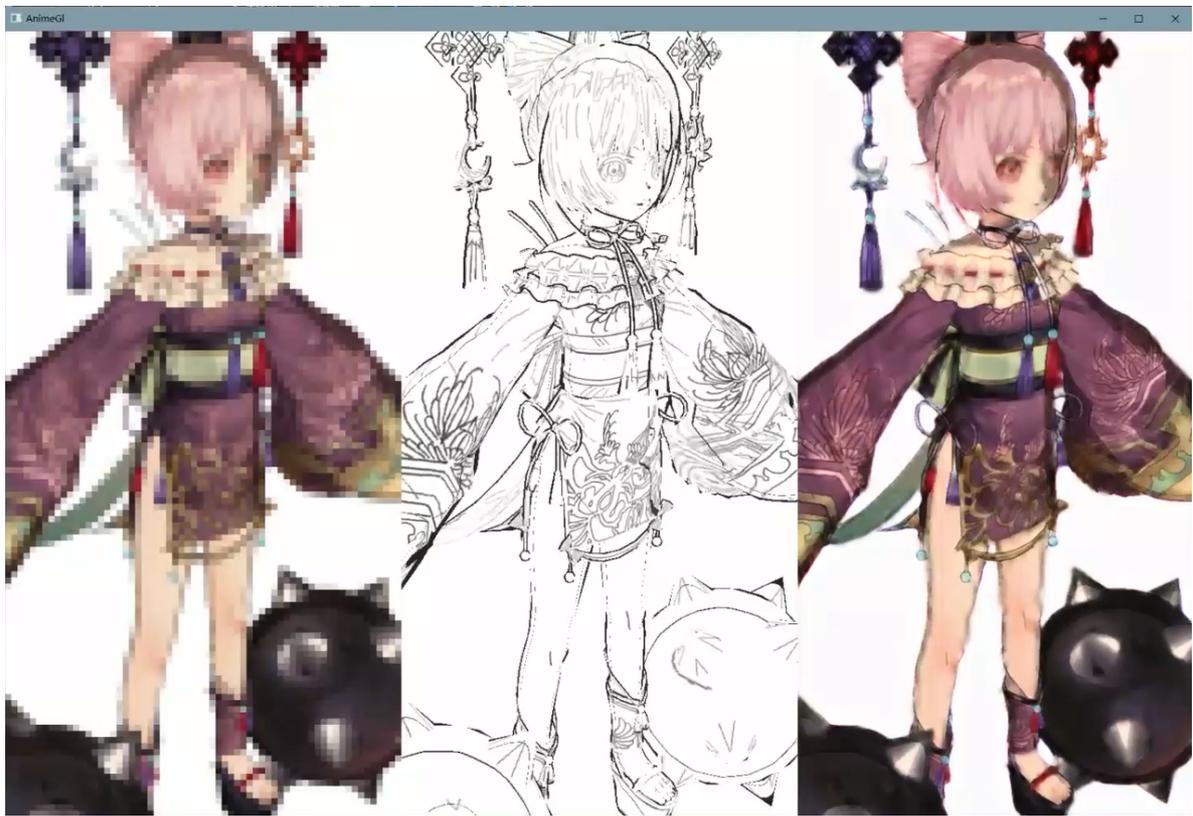
神经网络风格化重建图像



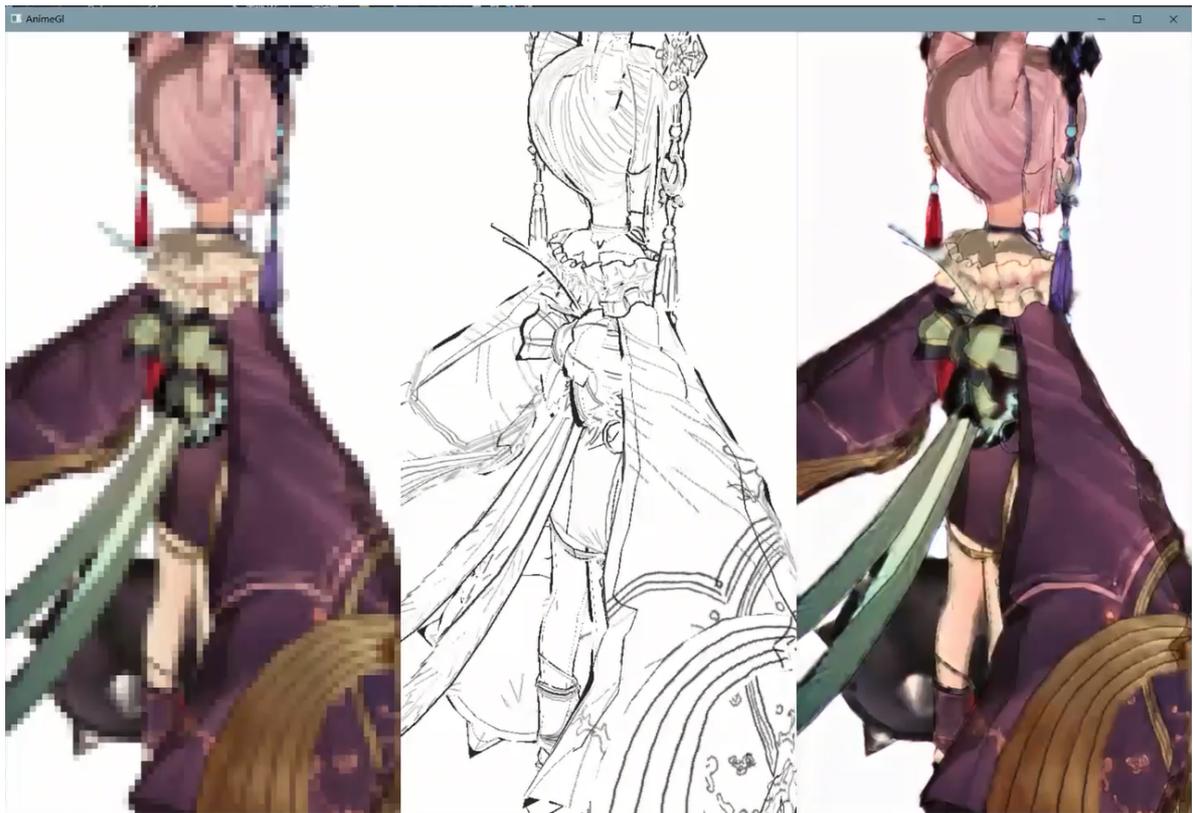
边缘 ||| 1/8分辨率预渲染 ||| 输出



1/8分辨率预渲染 ||| 边缘 ||| 输出



1/8分辨率预渲染|||边缘|||输出



1/8分辨率预渲染|||边缘|||输出

卡通渲染 (基于冯氏光照模型改进)



Roles in group

1752896 胡行健 0.6

负责:

总体方案设计;

全部神经网络部分:

数据集制作;

神经网络代码编写;

神经网络训练;

将 C-OpenGL 部分与 python-NN 部分结合;

C-OpenGL 部分:

协助完成轮廓扩张法 (此方案后被抛弃);

优化背部扩张法, 使描边宽度一致;

完成阴影描边、卡通渲染的 shader 编写;

绘制纹理描边贴图;

1752762 魏鹤达 0.4

负责:

C-OpenGL 部分:

使用saba项目以及阅读理解saba源码框架并尝试修改完成ModelViewer方面的程序框架和结构。

使用Assimp读取模型

使用3dsmax、maya修改导出模型

给模型加入坐标变化、光照(实现于FragShader)并渲染

将背部扩张法实现于vertex shader中, 并且提出优化方向。

在项目中引入OpenCV, 导出模型数据集

References

Sketchkeras	用于数据集边缘提取
PaintTensorFlow	模仿了它的项目架构 修改了它的 SubNet/Datasets/Training 等文件, 用于本项目中
Assimp	修改并使用它加载模型
LearnOpenGL	修改并使用了它的 shader.h/camera.h, 用于本项目中
stb_image	使用它加载纹理