光栅化作业报告

2015011308 计 53 唐适之

概要

本作业实现了直线画线算法，及其三种反走样算法——SSAA（提高分辨率后使用 OpenCV 缩小）、Sampling（区域采样）、Kernel（加权区域采样）。程序输出不同颜色不同方向的直线，以进行全面的比较。为了处理直线相交的情况，本程序绘制一定灰度的像素时，实际上是以该灰度为比例，将当前绘制的颜色与已绘制的背景，即以前绘制的直线，进行混合。

效果

输出效果如下（从左至右依次为朴素算法、SSAA、Sampling、Kernel）：

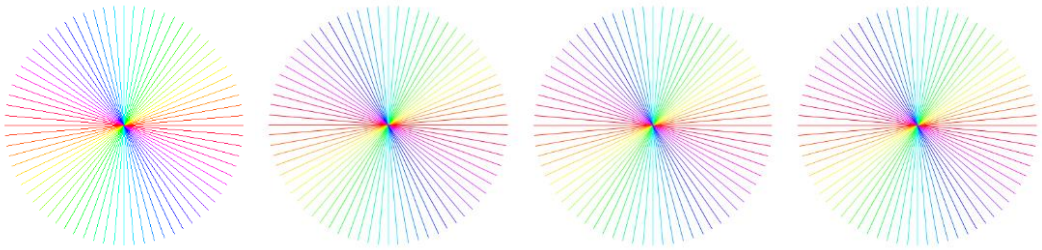
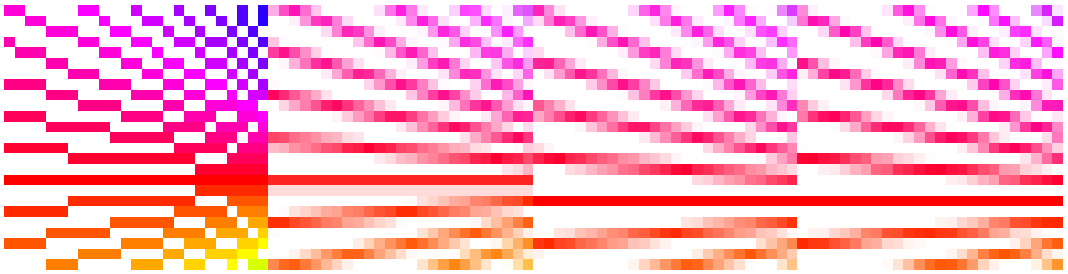


图 1: 完整效果

为展示反走样效果，将原图局部放大后效果如下（顺序同上）：



可以看出，三种反走样算法均实现了反走样效果。其中，SSAA 与 Sampling 表现相似，这是因为使用 OpenCV 在缩小图片时实际上是采用的 Sampling 算法。Kernel 算法绘制的直线的间断感比 Sampling 算法绘制的小，体现了加权的优势。

另外从效率上看，若记直线长度为 n，采样精度为 d（每个像素划分成 d^2 个小格），通过预处理权值矩阵的局部和，程序实现了 O(n·d)的 Sampling 和 Kernel 算法，而 SSAA 算法的复杂度为 $O(n \cdot d^2)$，效率不如 Sampling 和 Kernel 算法。

原始输出图像请见 output 文件夹。

具体实现

本程序首先实现了 Draw 类作为朴素算法，其中 draw 函数将直线镜像翻转为斜率∈[0,1]的情况，交给 drawImpl 执行具体算法。DrawWidth、DrawSampling、DrawKernel 分别直接或间接继承 Draw 类，覆盖其 drawImpl 函数以执行 SSAA、Sampling 和 Kernel 算法。最后 Canvas 类负责与 OpenCV 交互以输出。

具体代码见 src 文件夹，此外还有 Doxygen 导出的各函数和类的说明，见 doc/doxygen/index.html。