### 电计2003 邓人玮 20201071080 CG第一次作业报告

# 开发和运行环境说明

语言：ISO C++14

IDE：Visual Studio 2022

环境：Eigon:x64-windows 3.4.0；OpenCV:x64-windows 4.5.5

\*使用vcpkg打包集成到项目中

·Eigon方便管理2维数据和三维数据

·OpenCV用于把图像显示出来

# 算法设计思想和实现技术

先是新建一个光栅化器（Rasterizer）类，该类可以生成一个帧缓存frame\_buf用来创建cv::Mat对象image，并最终通过cv::imshow来输出图像。在该类中通过对frame\_buf的值作修改即可达到对对应像素修改颜色的效果。

主要设计思想是先求出所有需要着色的**点的列表**，然后再通过**着色函数**

void shade(const std::vector<Vector2i>& points, const Vector3f& color);

对传入的列表中的对应的所有像素点进行着色。

如果需要的是画出直线的点的列表，则使用

std::vector<Vector2i> line\_points(Vector2f begin, Vector2f end);

函数（传入起点和终点）。该函数采用的是**一般整数bresenham直线光栅化**算法；

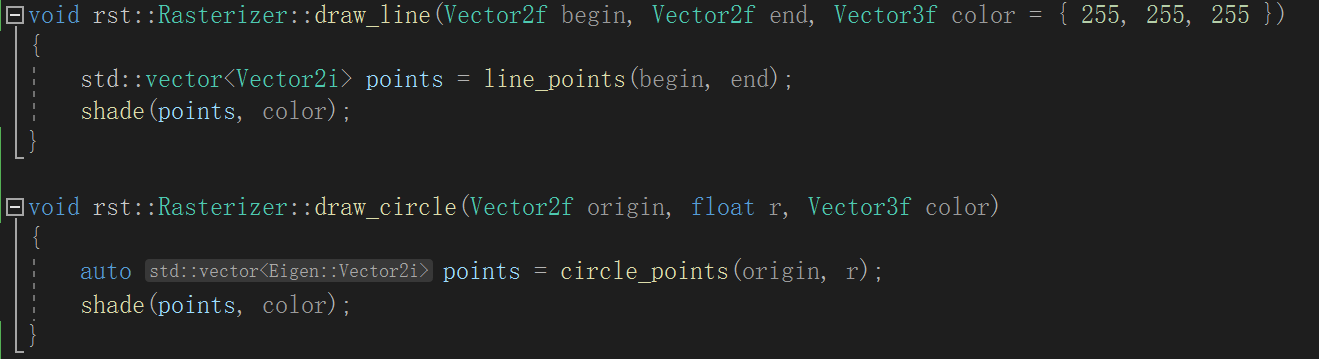
如果需要的是画圆的点的列表，则使用

std::vector<Vector2i> circle\_points(Vector2f origin, float r);

函数（传入原点和半径）。该函数采用的是**Bresenham画圆算法**（利用八方对称性）。

\*具体实现参见rasterizer.cpp

于是要画圆draw\_circle或直线draw\_line，只需先获取到所需的点的列表，再shade即可。

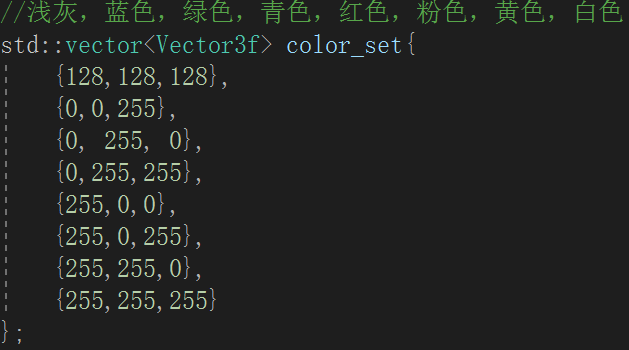


### 任务1实现：

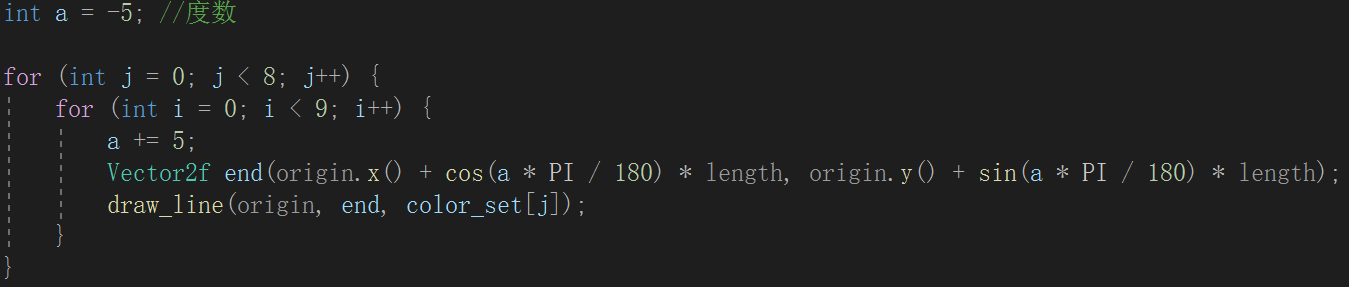
任务1的实现在void rst::Rasterizer::mission1(const Vector2f& origin, const int length)函数中。(rasterizer.cpp)

首先是数清楚样例图片中的每个颜色占多大角度，一共多少条。然后得出递增的角度是5°，并且每种颜色9根，每种颜色占45°。

然后查清楚每种颜色对应的RGB分别是多少，按顺序记录在color\_set中。



起点已知，终点利用角度和线段长度算出来。用一个for语句迭代角度a，每次+5°即可。



### 任务2实现：

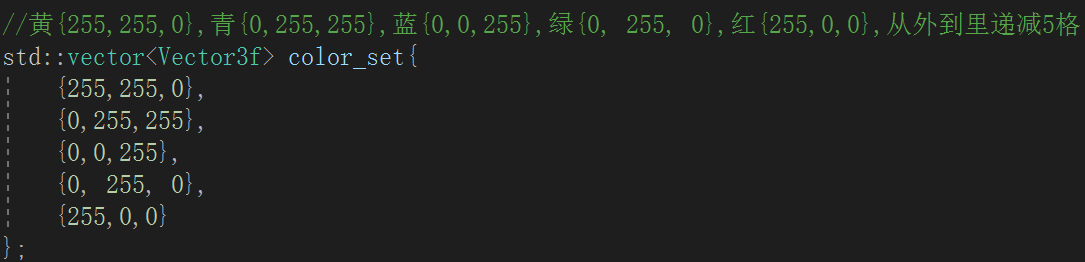
任务2的实现在void rst::Rasterizer::mission2(const int a, const int interval)函数中。（rasterizer.cpp）

其中interval是网格间距，以网格中心点为原点(0,0)，网格坐标坐标的最大正值为a。

\*当a=25时，x，y轴的网格坐标范围是-25~25，横纵各有51根线。

首先是画网格，使用了函数std::vector<Vector2i> grid\_square\_points(int line\_num, int interval);实现见rasterizer.cpp代码部分，主要思想是从中心开始每隔一个间隔就画上下左右四根线。

然后确定画的每个圈的颜色列表color\_set：



由于有5个大圆，于是for循环5组，每组的任务有：

1. 画大圆，用draw\_circle函数；
2. 把网格坐标原点（0，0）还有随着循环递减的a传入函数circle\_points，得出大圆所在网格的逼近点--每个小圆中心点的网格坐标列表。
3. 把每个小圆的中心点网格坐标列表中的点逐个转换为真实坐标作为圆心传入画圆函数draw\_circle画出来，颜色与大圆颜色相同，其中小圆半径为0.25个网格单位长度。

·要求栅格大小可以调整：只需要调节该任务的函数

void rst::Rasterizer::mission2(const int a, const int interval)中的interval即可。

# 程序的运行结果与分析

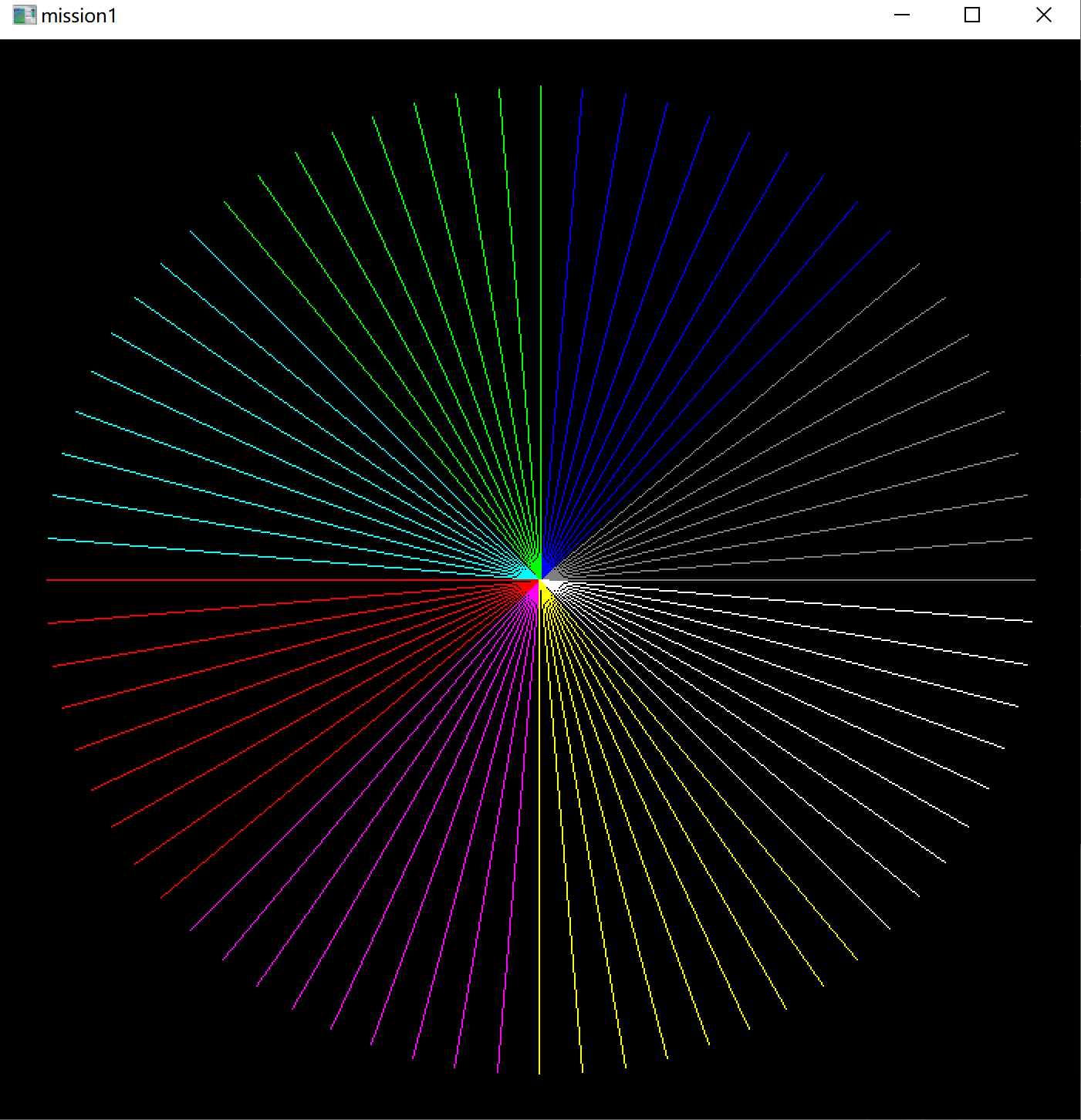
参数：画布：宽700，高700。任务一线长320。

任务二坐标范围-25~25，正方形网格间距12.

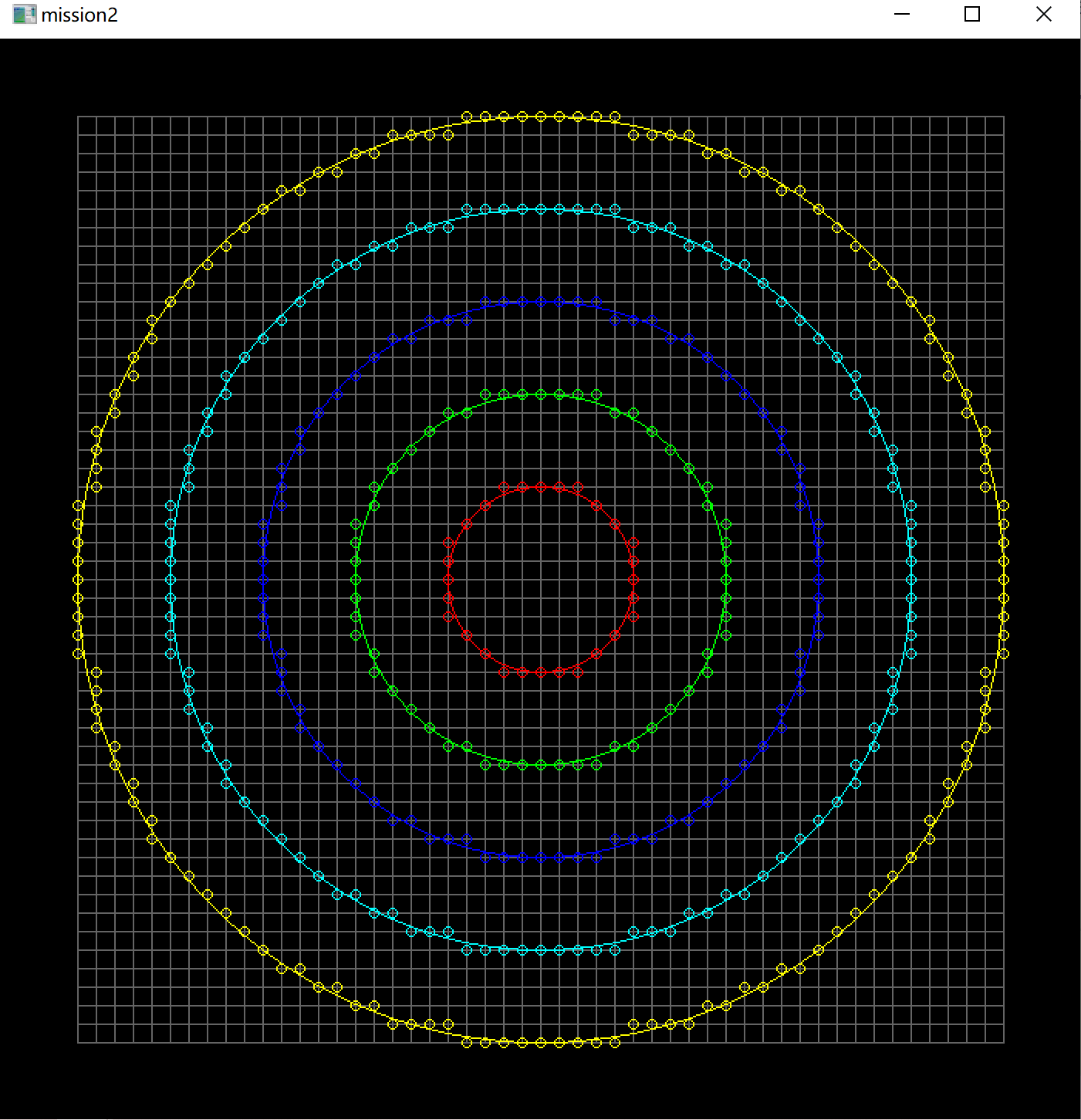
主函数代码参见main.cpp

结果如下：

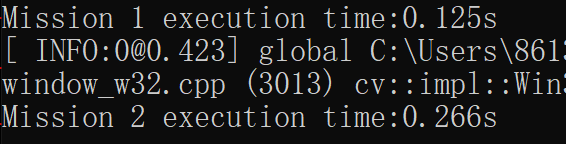
任务1：



任务2：



性能：用时如下：



问题：最开始画圆时大圆与网格对不上，有一格的误差

改进：通过查看公式发现在计算d的时候应该把x-1带进去计算，而不是当前的x。

分析：最终结果与理想状态基本上完全一致，很好地完成了此次任务。