



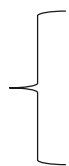
# 基于图像方法的代码查重方案

报告人：郑程耀

# Contents

01

构想提出



图像领域有众多优秀的理论和方法

帮助同学们深入理解课堂知识

02

方案实现



代码转图像

提取sift特征点

匹配与过滤

封装、功能增强

03

效果展示



视频演示

不知道这里写啥了为了对称美观随便写点

# 构想提出



图2. 图像算法发展至今，不乏优秀的理论和算法


# 方案实现

## 01 Code to Image

- 删除注释和非关键部分 (#include, import, define)
- 删除空格
- 查询ascii码表, 字符转ascii码 (0-126)
- 归一化到 [0, 255], 以CV\_8UC1格式创建cv::Mat

# 方案实现

## 01 Code to Image



```
File Edit Selection Find View Goto Tools Project Preferences Help
getIntrinsicParameter.cpp x
1 #include "getIntrinsicParameter.h"
2
3 void getIntrinsicParameter(double focal_length, cv::Poi
4 {
5     K = cv::Mat_<double>(3,3);
6     K.at<double>(0,0) = focal_length;
7     K.at<double>(0,1) = 0;
8     K.at<double>(0,2) = principal_point.x;
9     K.at<double>(1,0) = 0;
10    K.at<double>(1,1) = focal_length;
11    K.at<double>(1,2) = principal_point.y;
12    K.at<double>(2,0) = 0;
13    K.at<double>(2,1) = 0;
14    K.at<double>(2,2) = 1;
15 }
16
```

图3. 图是一份代码正在失去梦想

# 方案实现

## 02 Sift 特征提取

- 建立DOG金字塔
- 极值搜索
- 极值清洗
- 生成1\*128维的特征描述子

# 方案实现

02

Sift 特征提取

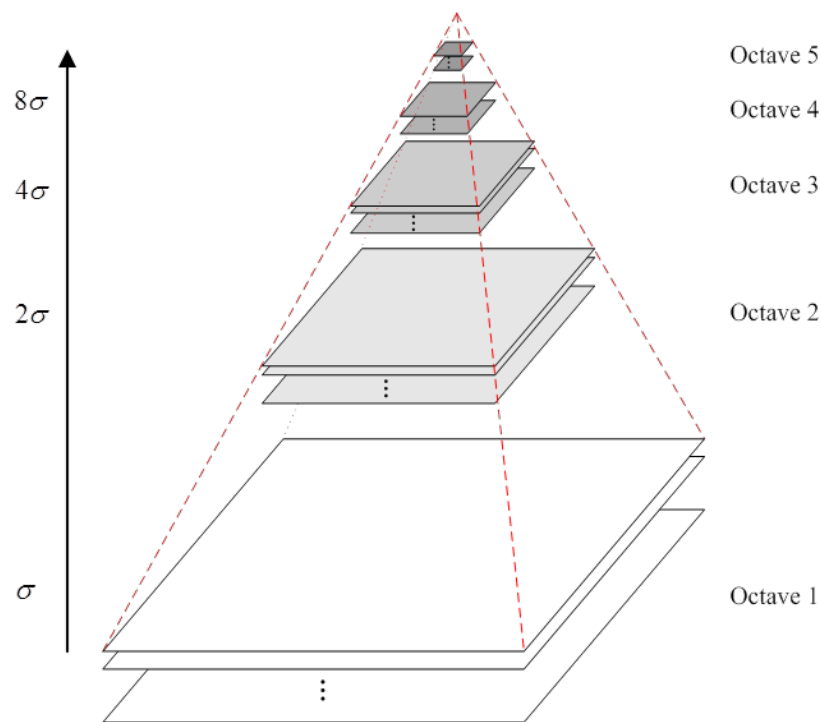


图4. 不同尺寸下的图像尺度

## 方案实现

## 02

## Sift 特征提取

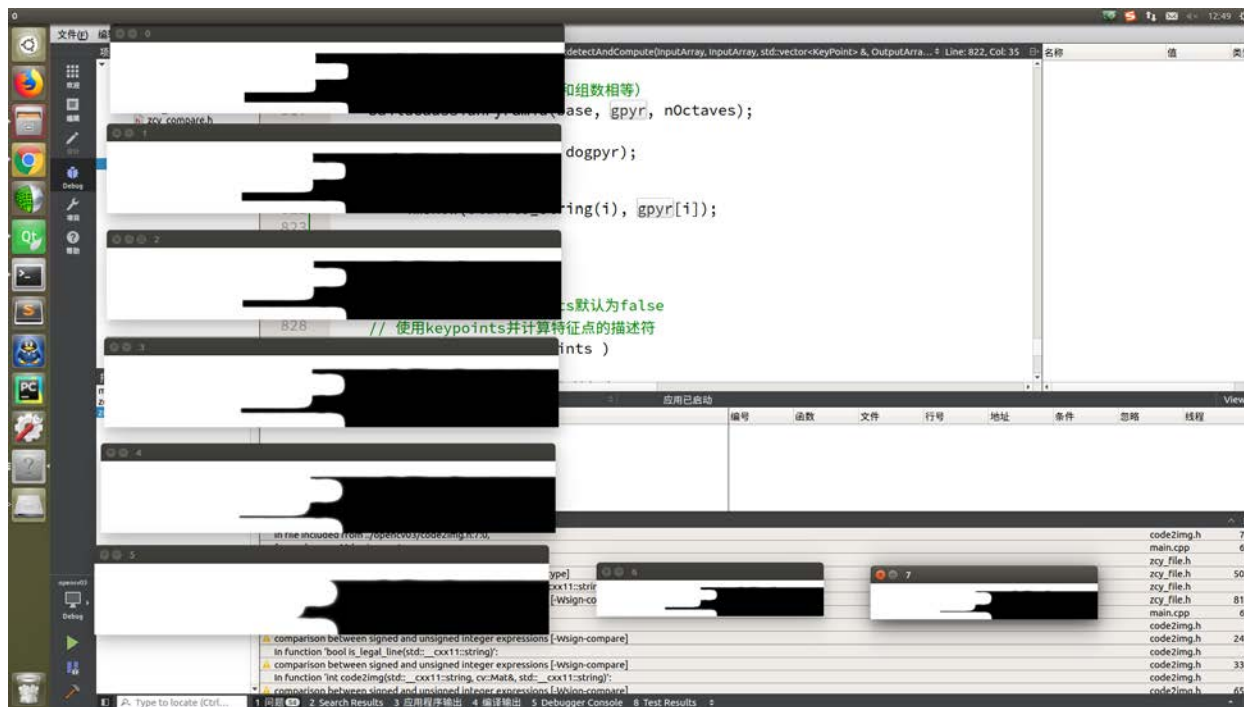


图5. 不同尺度的图像



# 方案实现

02

Sift 特征提取

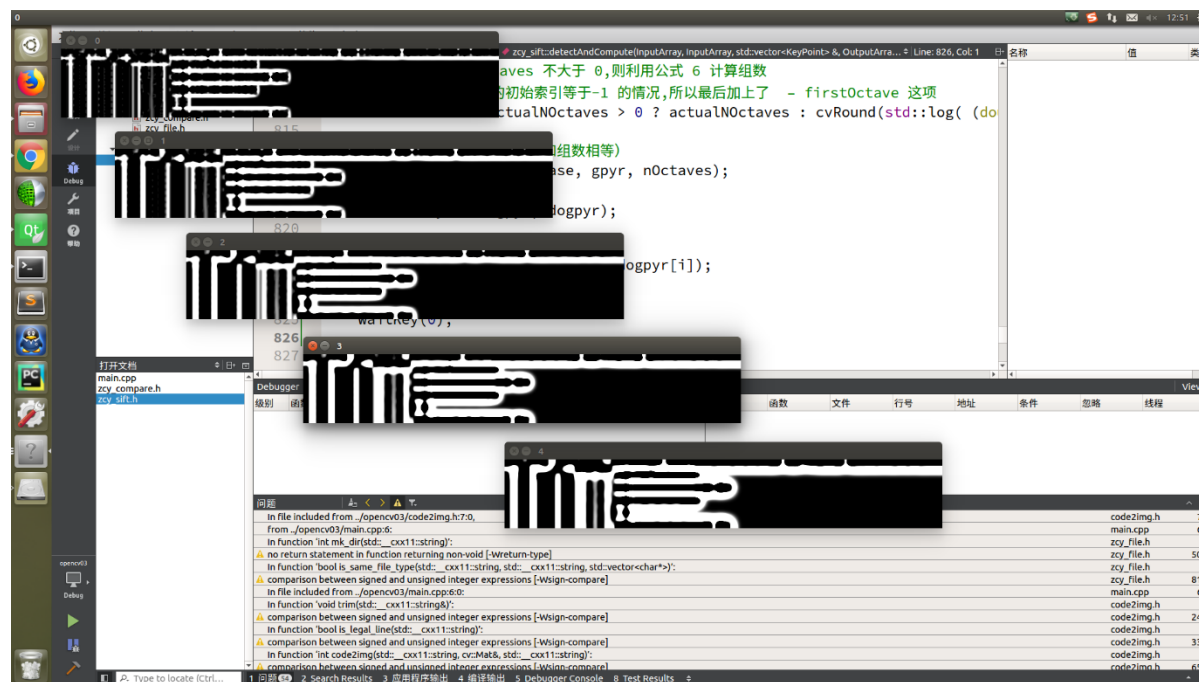


图6. 不同尺度的图像 (DOG金字塔)

# 方案实现

03

匹配与过滤

- 暴力匹配
- Ransac筛选
- 登峰造极孤独求败——吕氏平行滑窗筛选法则

# 方案实现

03

匹配与过滤

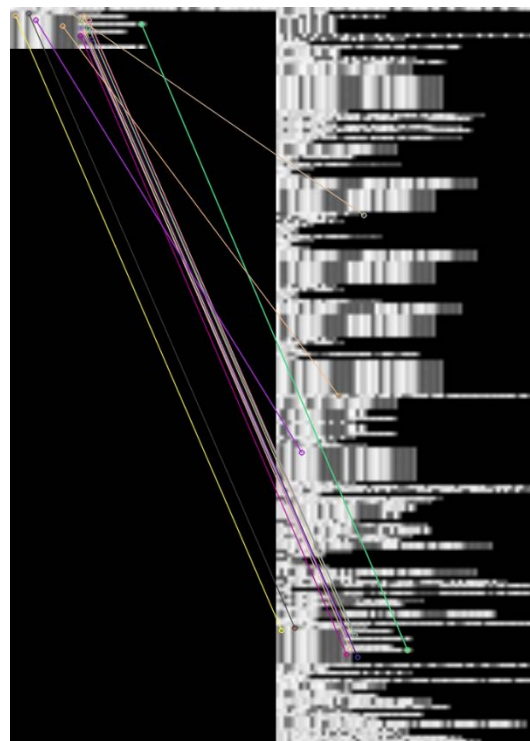
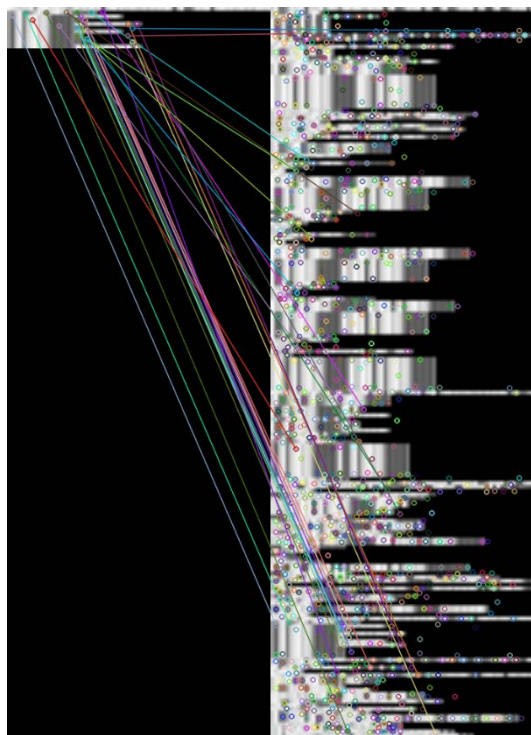
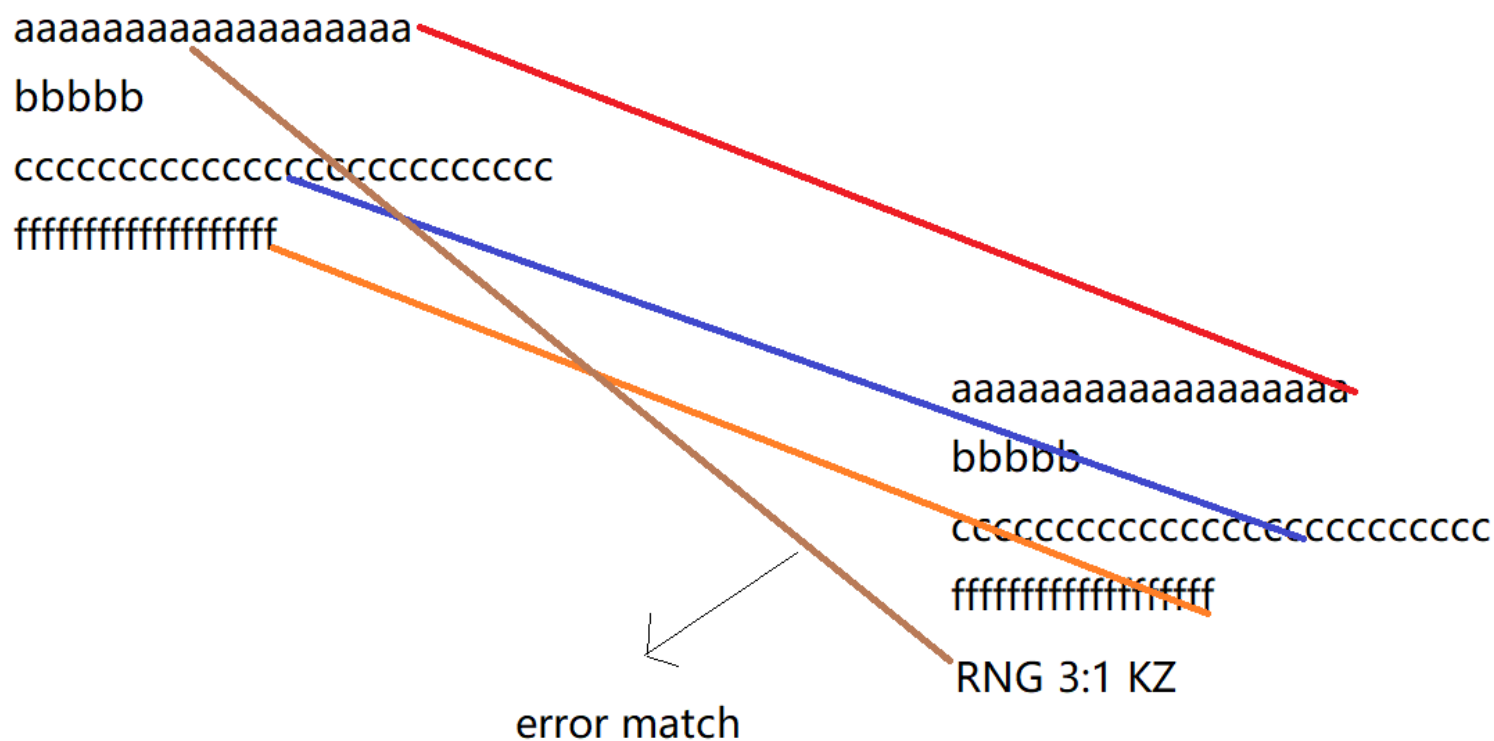
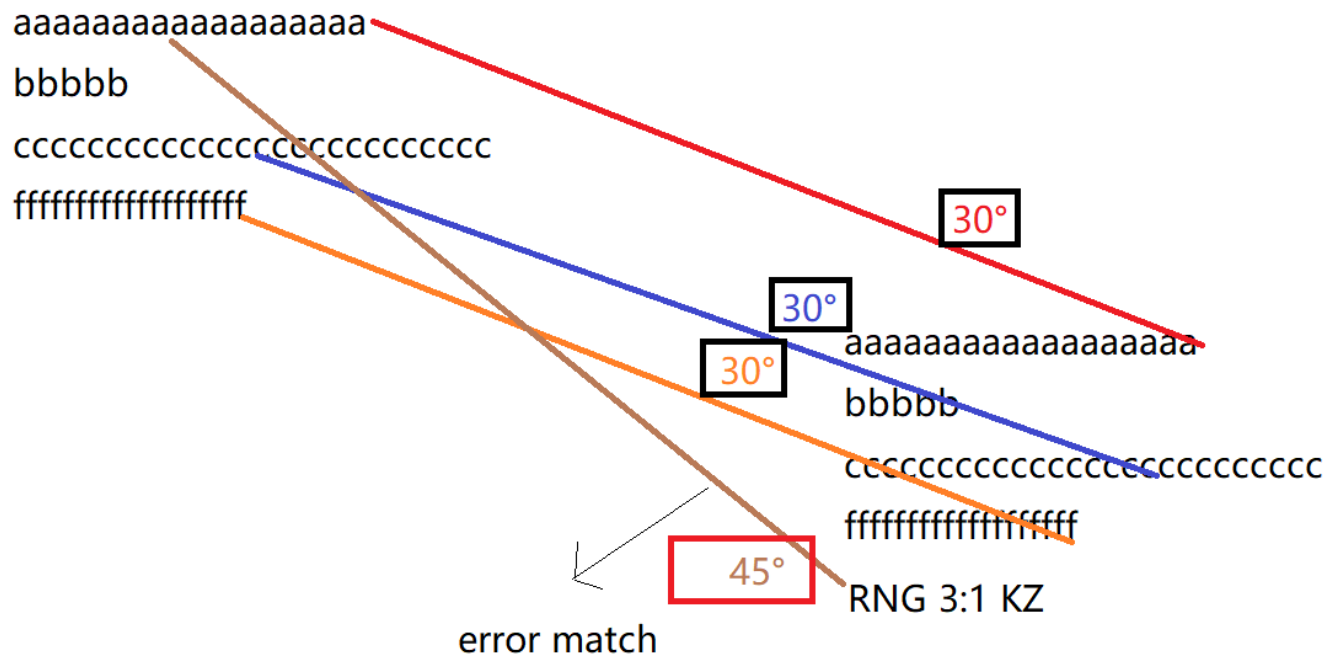


图7. 暴力匹配与ransac筛选

# 依然存在错匹配对



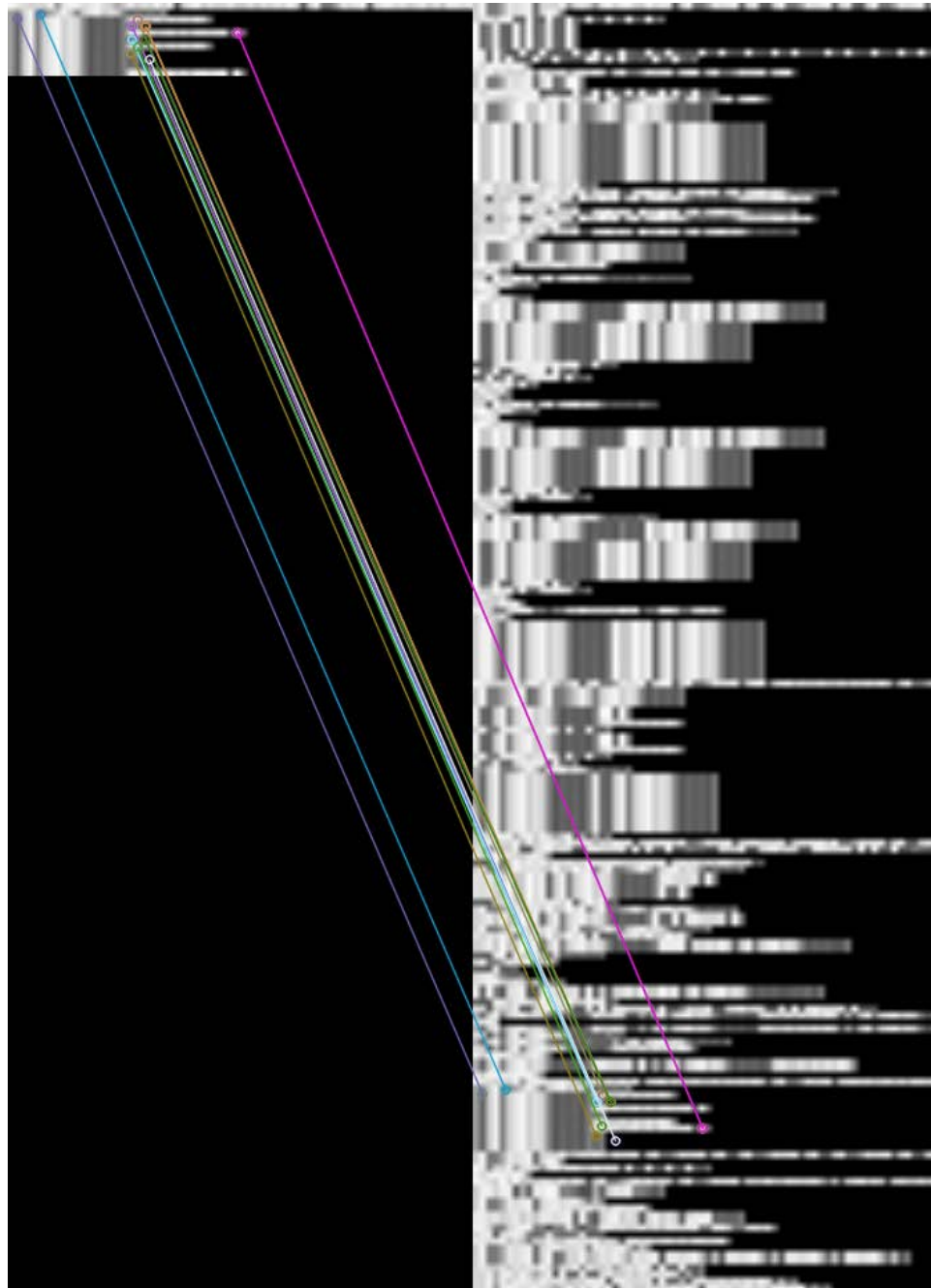
# 同一代码段匹配线段平行



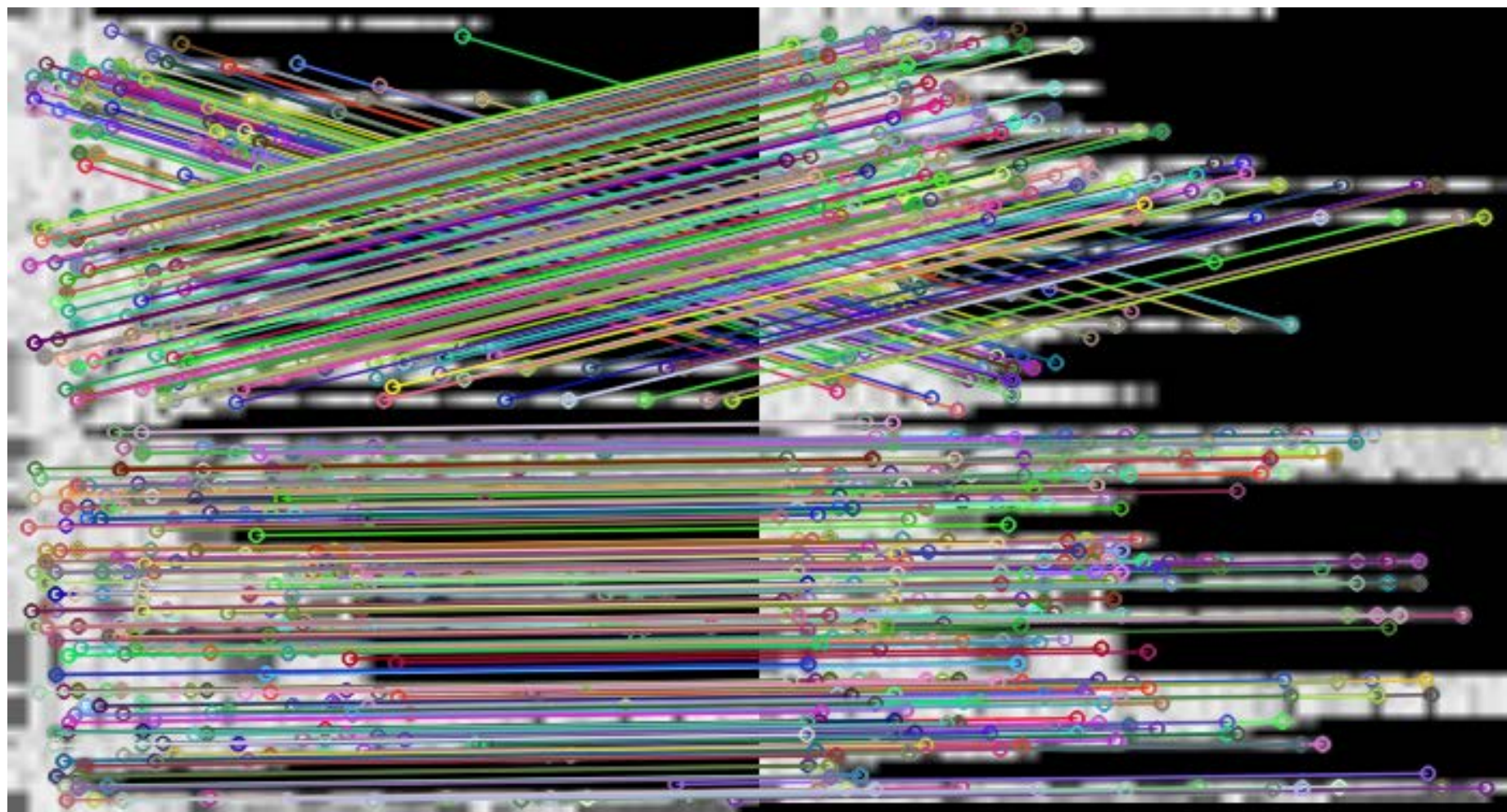
# 最快步骤

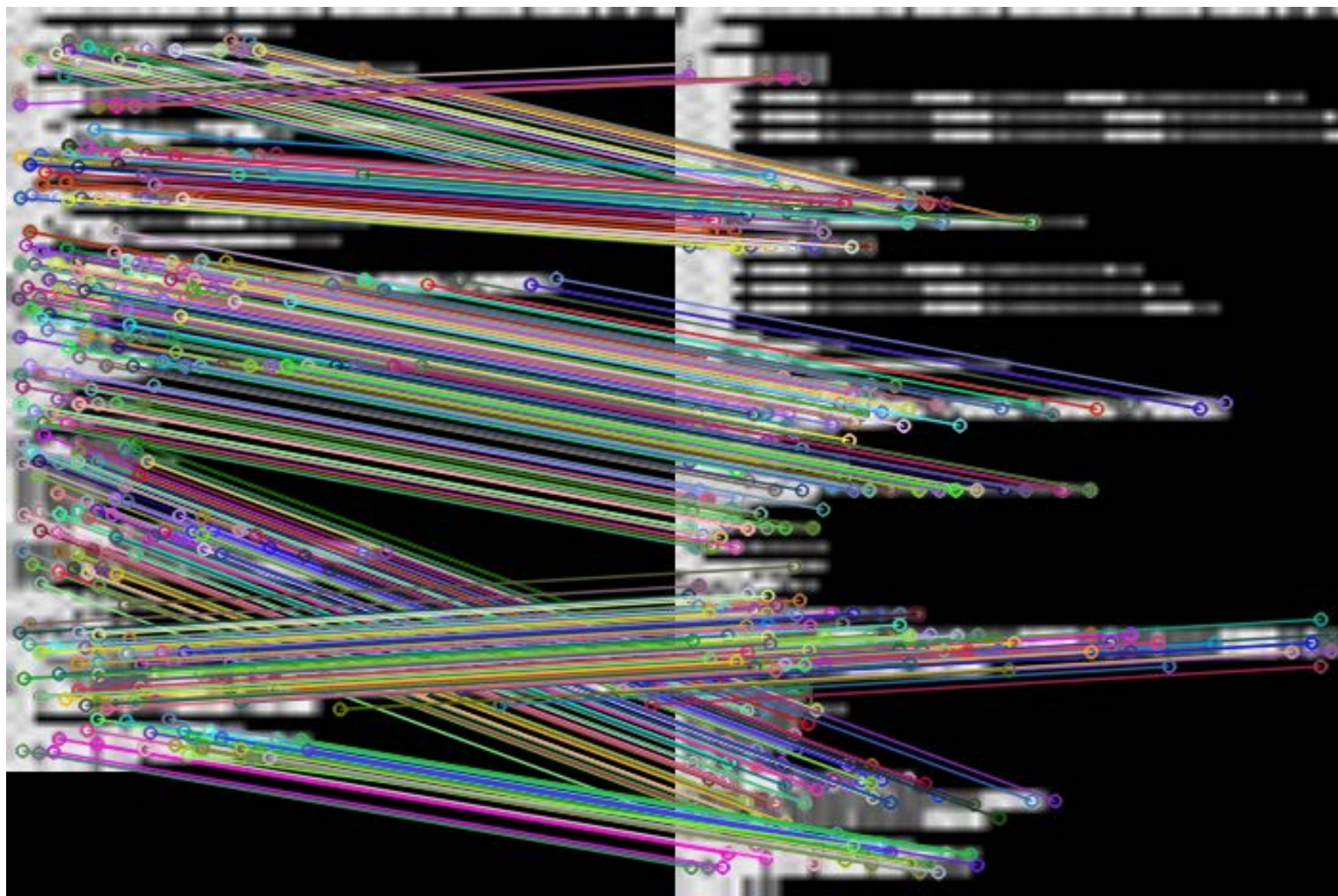
- 所有匹配点按y轴坐标排序（快排序，最少时间空间复杂度）
- 检查每一个匹配对前后n个匹配对的角度
- 如果角度相似的超过了5个，说明是正确的

# 丧心病狂的视觉盛宴









# 效果展示

03

封装

- 模式一：比较两份代码相似度  
-c main.cpp test.cpp
- 模式二：比较目录下两两文件夹内所有代码相似度  
-codes2imgs /root/src /root/tgt  
-CheckCurDir /root/src /root//tgt



# 效果展示

04

问题

- 变了变量名呢?
- 改了代码块位置呢?

请各位老师批评指正，谢谢！