**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **计算机视觉** |
| **学生姓名：** |  |
| **学生学号：** |  |
| **学生专业：** |  |
| **开课学期：** | **2020-2021学年下学期** |

实验一、相机标定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地 点：** |  |  | **评 分：** |  |
| **实验日期与时间：** |  | | **实验教师：** |  |

# 一、实验目的

本次实验课程的目的是了解相机标定的原理，通过编译、运行相机标定，加深对算法的理解。

## 二、实验内容

环境配置：

相机：XiaoMi 12 Pro 广角镜头，分辨率 8192\*6144

编程环境 Python 3.9.12 + Opencv 4.5.5.64 + numpy 1.19.5

使用A4纸大小的7\*10棋盘格，边长3cm。

算法详细描述：

本实验算法分为几部分：标定棋盘角点、计算相机参数、计算标定偏差、原图去畸变。

1. 标定棋盘角点

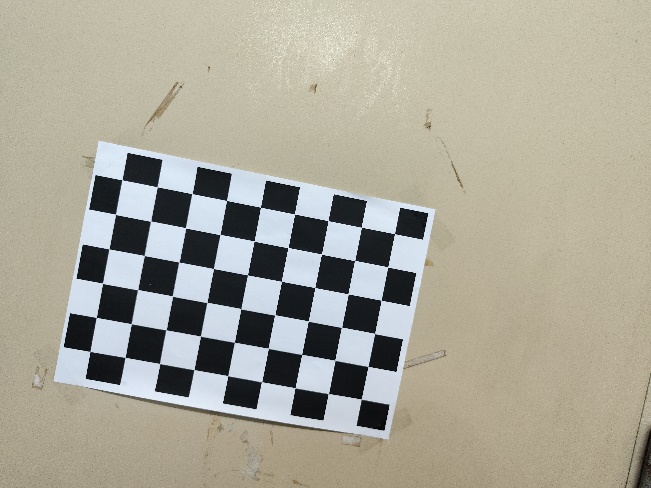


Figure 1标注图片

首先计算角点数目，此图像为 (6, 9) ，然后使用 findChessboardCorners 找到每个角点的位置。虽然 findChessboardCorners 返回的已经是亚像素坐标，但还需再使用cornerSubPix获取角点的亚像素位置。然后可以使用 drawChessboardCorners 画出角点位置，判断是否标定正确。

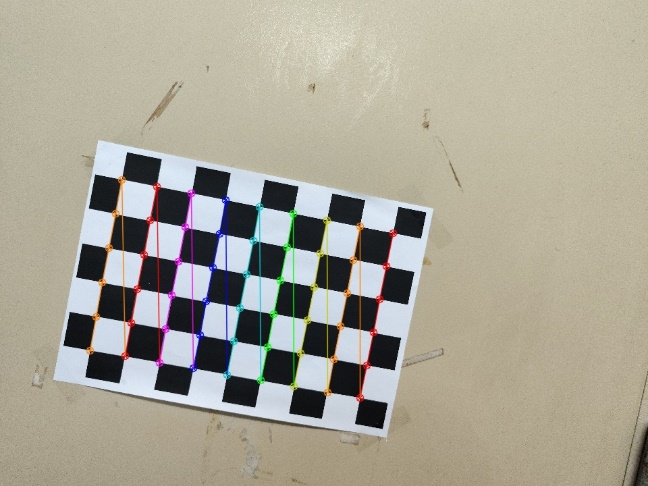


Figure 绘出角点

1. 计算相机参数

OpentCV提供了calibrateCamera计算相机参数，依据文档，需要提供objectPoints和imagePoints。objectPoints每张图片中角点在世界坐标系下的位置，imagePoints为每张图片中角点在像素坐标系下的位置。返回值cameraMatrix为相机内参矩阵，distCoeffs为相机畸变矩阵，rvecs和tvecs分别是标定图像的旋转矩阵和平移矩阵。

1. 计算标定偏差

OpentCV提供了projectPoints将世界坐标系下的点转换为像素坐标系下的点。然后计算模拟的坐标和实际坐标的平均偏差像素，得到标定偏差的一种评价量。

1. 原图去畸变

使用 undistort 将原图转化为去畸变之后的图像。

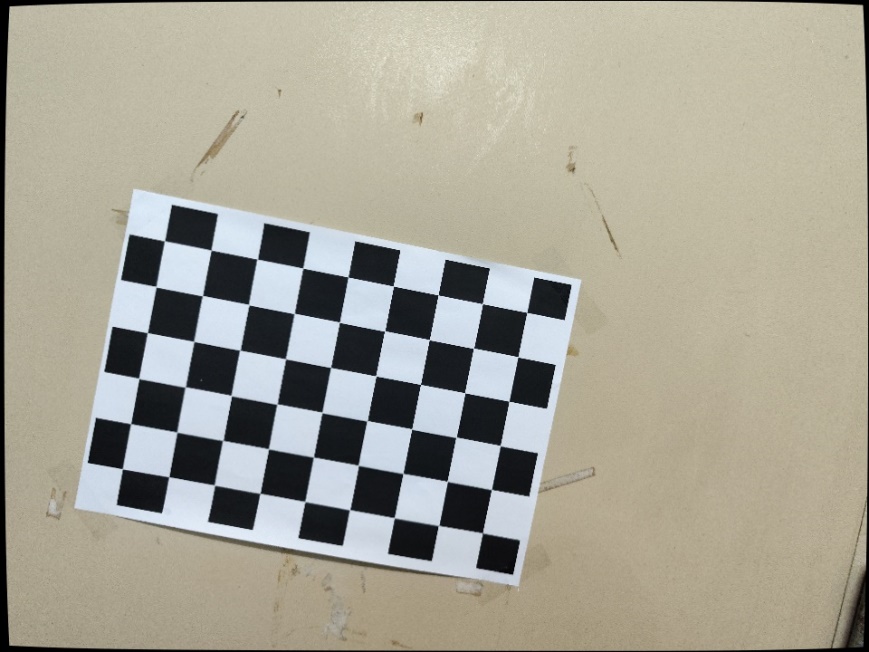


Figure 去畸变

关键代码讲解：

物体坐标：

objp = np.array([[j, i, 0] for i in range(CORNERS\_SIZE[1]) for j in range(CORNERS\_SIZE[0])], dtype=np.float32,)

这个表达式生成每个角点的世界坐标，供calibrateCamera使用

亚像素求解：

corners = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1), (cv2.TermCriteria\_EPS | cv2.TermCriteria\_EPS, 30, 0.001))

最后一个参数为迭代终止条件

计算标定偏差：

error = cv2.norm(imgpoints[i], imgpoints2, cv2.NORM\_L2) / len(imgpoints2)

此函数会计算每点之间的平面距离的平均数，即为我们所求的标定偏差

### 三、实验结果

相机内部参数：[

[5.87281130e+03, 0.00000000e+00, 4.10157887e+03],

[0.00000000e+00, 5.90670442e+03, 3.11356120e+03],

[0.00000000e+00, 0.00000000e+00, 1.00000000e+00]

]

相机畸变参数：[[ 0.01464611, 0.02222708, 0.00041811, -0.00168721, -0.05723866]]

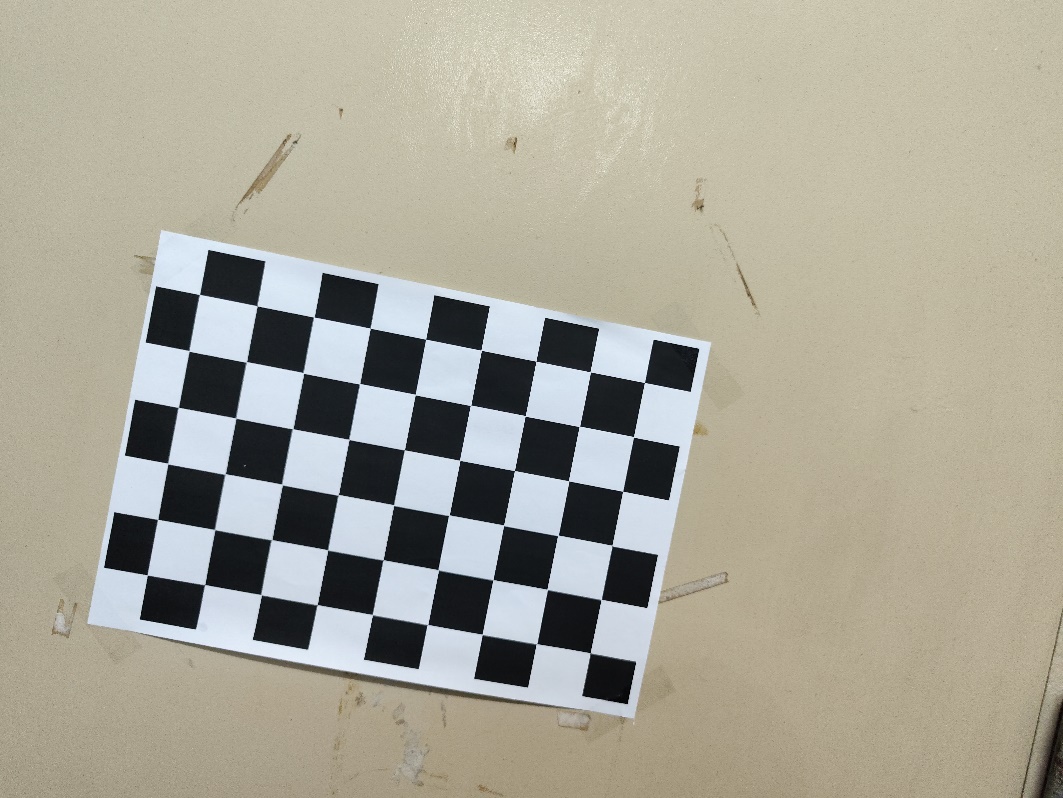


Figure 原始图片

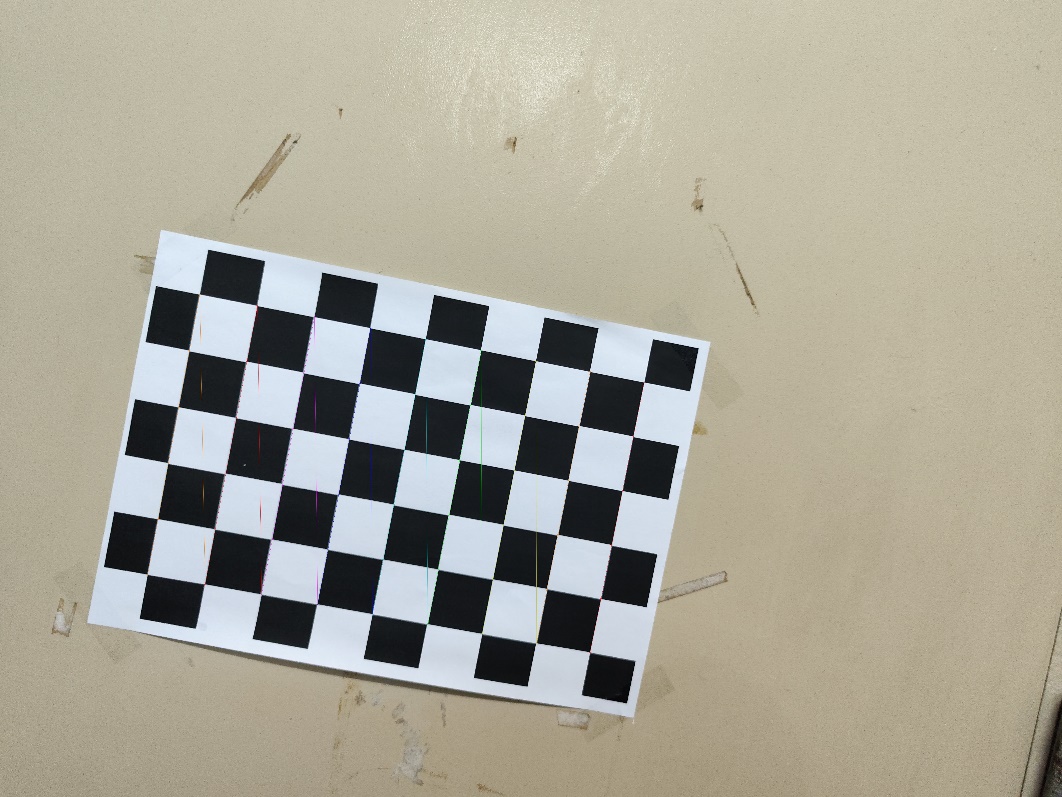


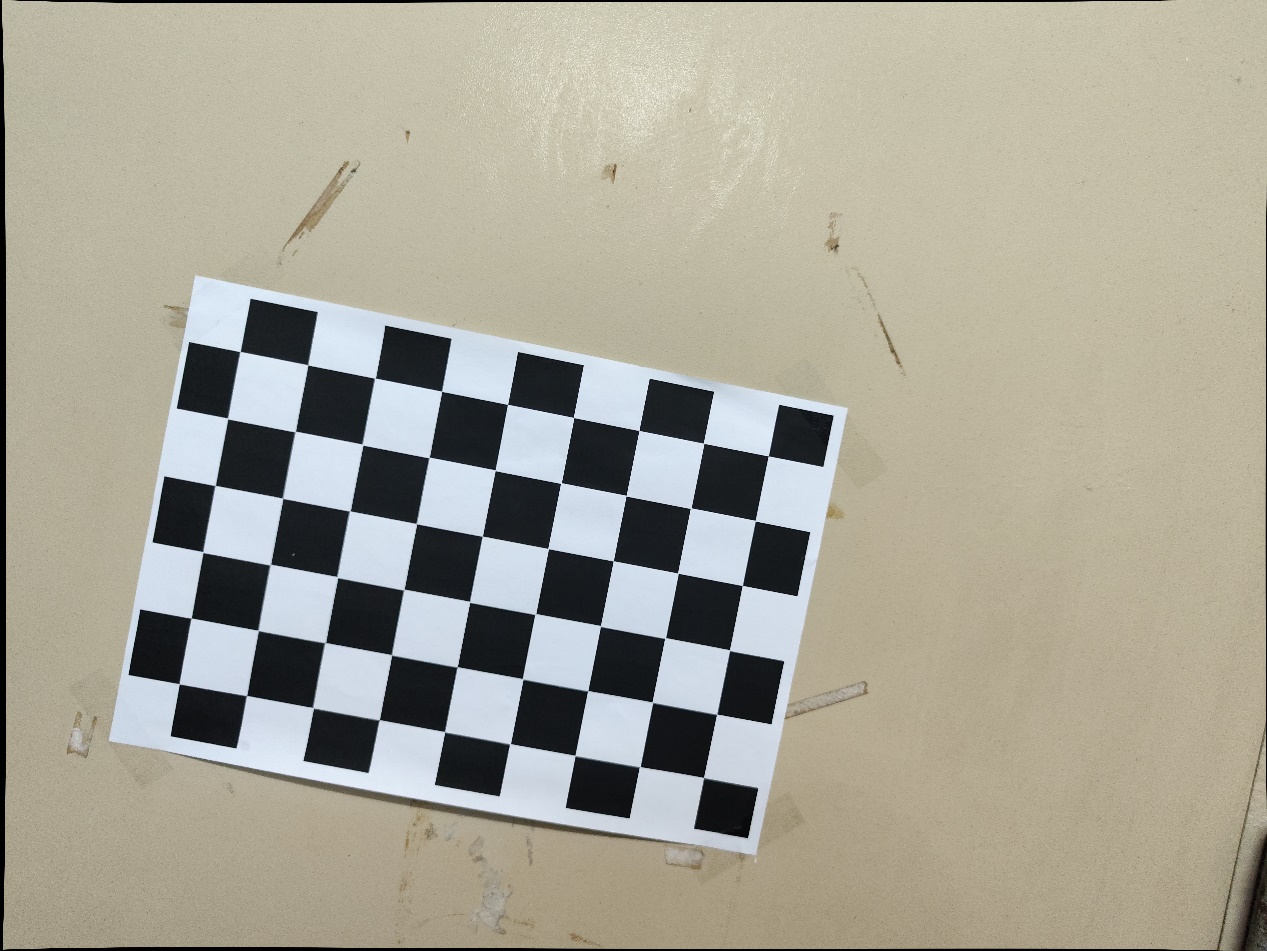
Figure 标注角点

Figure 校正后图像

所有图像平均偏差：0.2847407060700459

最高偏差：0.3587383827190461

#### 四、总结

20张图片的角点标注全部都以相同顺序、正确地标注出了所有角点。观察矫正前后的图像可以发现，图像周围存在极小黑边，所以相机确实存在轻微形变，但人眼观察极难发现形变。

所有图片的平均偏差小于0.3，最高偏差0.36，所以标定结果正常。

实验二、 AR实验

## 一、实验目的

本次实验课程的目的是了解增强现实技术（AR）的原理，通过编译、运行深度图像计算，加深对算法的理解。

## 二、实验内容

环境配置：

相机：XiaoMi 12 Pro 广角镜头，分辨率 8192\*6144

编程环境 Python 3.9.12 + Opencv 4.5.5.64 + numpy 1.19.5 + Open3d 0.15.1

算法详细描述：

本实验算法分为几部分：

1. 相机标定获得相机内参数矩阵
2. 特征点匹配

Opencv内置了多种算法计算关键点及描述子，这里可以使用ORB算法求解关键点及描述子，再使用FlannBasedMatcher得到特征点的匹配，然后再用最近邻比次近邻比过滤匹配。

1. 计算基础矩阵F和本征矩阵E

本征矩阵E，是单几何意义上的，与成像仪无关，它将左摄像机观测到的点P的物理坐标和右摄像机观测到的相同的点的位置关联起来。

基础矩阵F，则是将一台摄像机的像平面上的点在图像坐标（像素）上的 坐标和另一台摄像机的像平面上的点关联起来。

Opencv 有 findFundamentalMat 与 findEssentialMat 计算基础矩阵和本征矩阵。

1. 计算旋转矩阵R和平移向量T

可以使用SVD分解，也可以使用cv:: decomposeEssentialMat求解。注意R和T有多达四种的组合，需要结合实际相机的位置选取正确的R和T。

1. 有效特征点三角化实现重建

已知（u,v）为空间点在物理成像平面中的坐标，则（u,v,1）为空间点在物理成像平面中的齐次坐标。根据两个不同的归一化图像平面中的坐标u和u1，以及投影矩阵P0和P1，可以得到该点的三维空间坐标。

Opencv提供了triangulatePoints批量计算坐标点，但返回值为齐次坐标。

关键代码讲解：

* FlannBasedMatcher：

依据Opencv的文档，如果兴趣点是使用的ORB算法，需要以一下方式调用：

FLANN\_INDEX\_LSH = 6

index\_params = dict(

algorithm=FLANN\_INDEX\_LSH,

table\_number=6, # 12

key\_size=12, # 20

multi\_probe\_level=1,

) #2

search\_params = dict(checks=50)

matcher = cv2.FlannBasedMatcher(index\_params, search\_params)

* matches的选取：

不是所有matches都是正确的匹配。一种推荐的过滤方式是最近邻比次近邻比过滤匹配，即过滤最佳匹配距离与次佳匹配距离大于某个比值的，这个比值推荐为0.7~0.8;

all\_matches = matcher.knnMatch(des1, des2, 2)

matches = np.array([m[0] for m in all\_matches if m[0].distance < DIS\_RATE \* m[1].distance])

* triangulatePoints的参数：

第一个参数为第一个相机的投影矩阵，为 [[1,0,0,0],[0,1,0,0],[0,1,0,0]]

第二个参数为第二个相机的投影矩阵，为 R与T拼接起来

第三、四个参数为匹配点坐标，注意这里的匹配点要过滤求解基础矩阵和本征矩阵时舍弃的匹配，以达到较高的正确率

### 三、实验结果

Figure 图片1 - 左侧



Figure 图片2 - 右侧



Figure 匹配结果



Figure 匹配结果 - 局部

#### 四、深度图像验证

通过前面求得的匹配结果，可以计算得到旋转矩阵R和平移向量T，从而可以进行3D图像的重建，一下是重建结果。

旋转矩阵R：[

[ 0.98811874 -0.0285606 0.15101542]

[ 0.02792949 0.99959005 0.00629891]

[-0.15113341 -0.00200629 0.98851134]

]

平移矩阵 T：[[ 0.95407056] [ 0.14615068] [-0.26151356]]

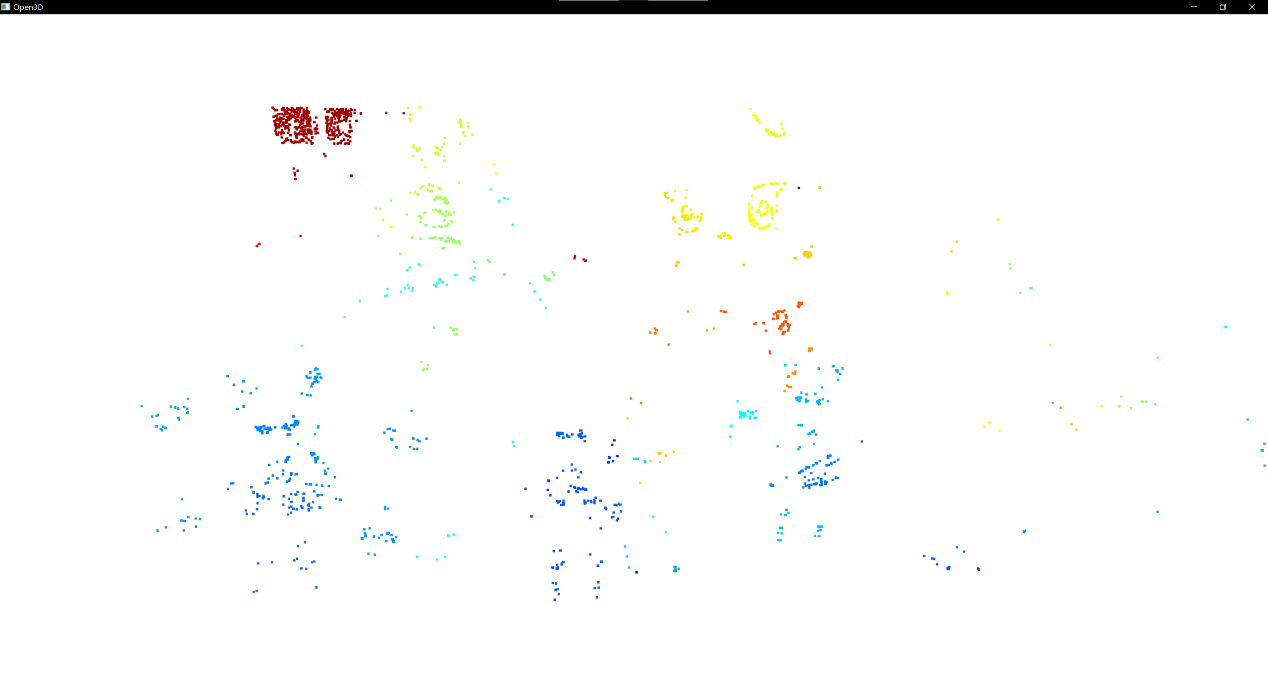


Figure 点云正面视角

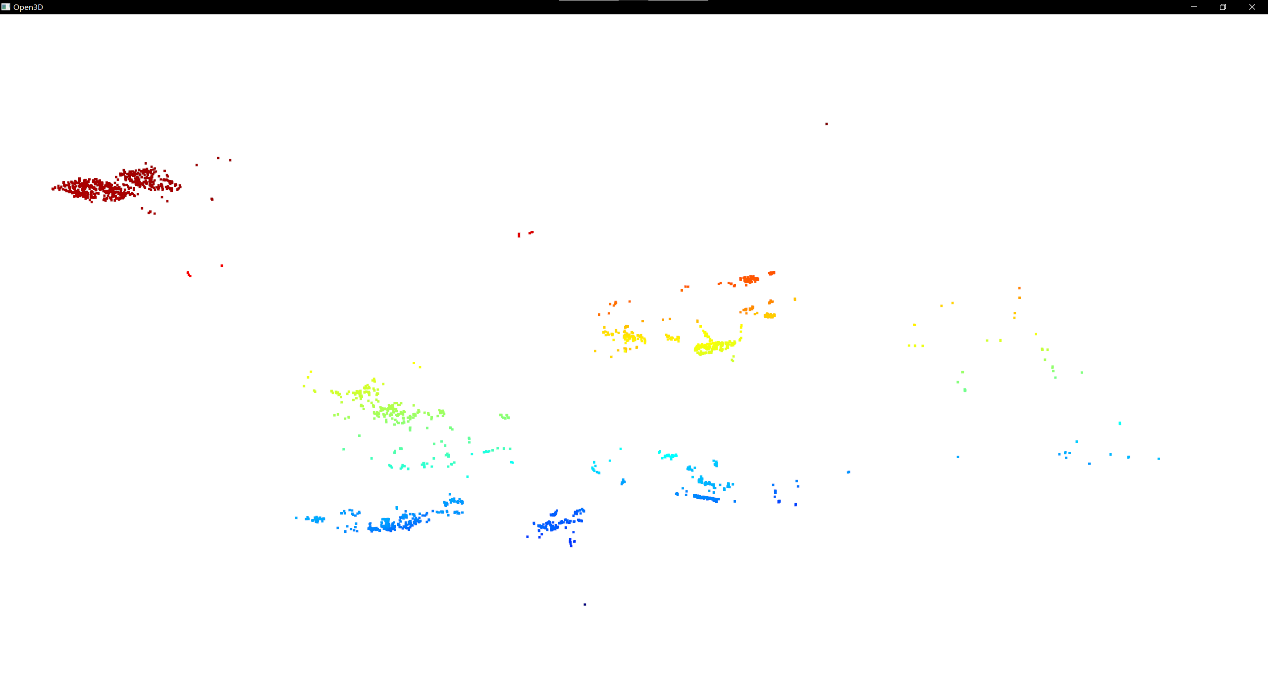


Figure 点云上方视角

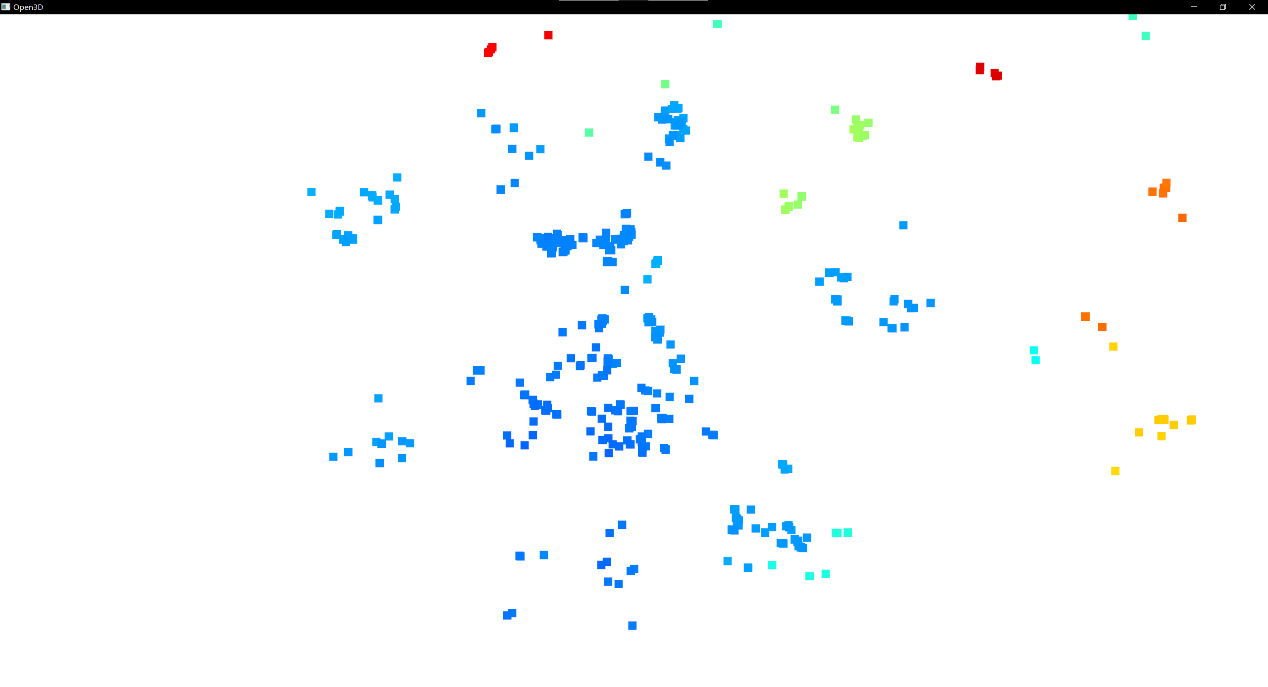


Figure 点云 - 局部

可以发现，通过特征点正确识别了前方的三个小模型，同时也识别到了后面的一个大模型及一个玩偶还有左上角的二维码。特别是前方的模型，捕获到的特征点较多，遍布了整个模型，且较好的还原了模型的三维结构。

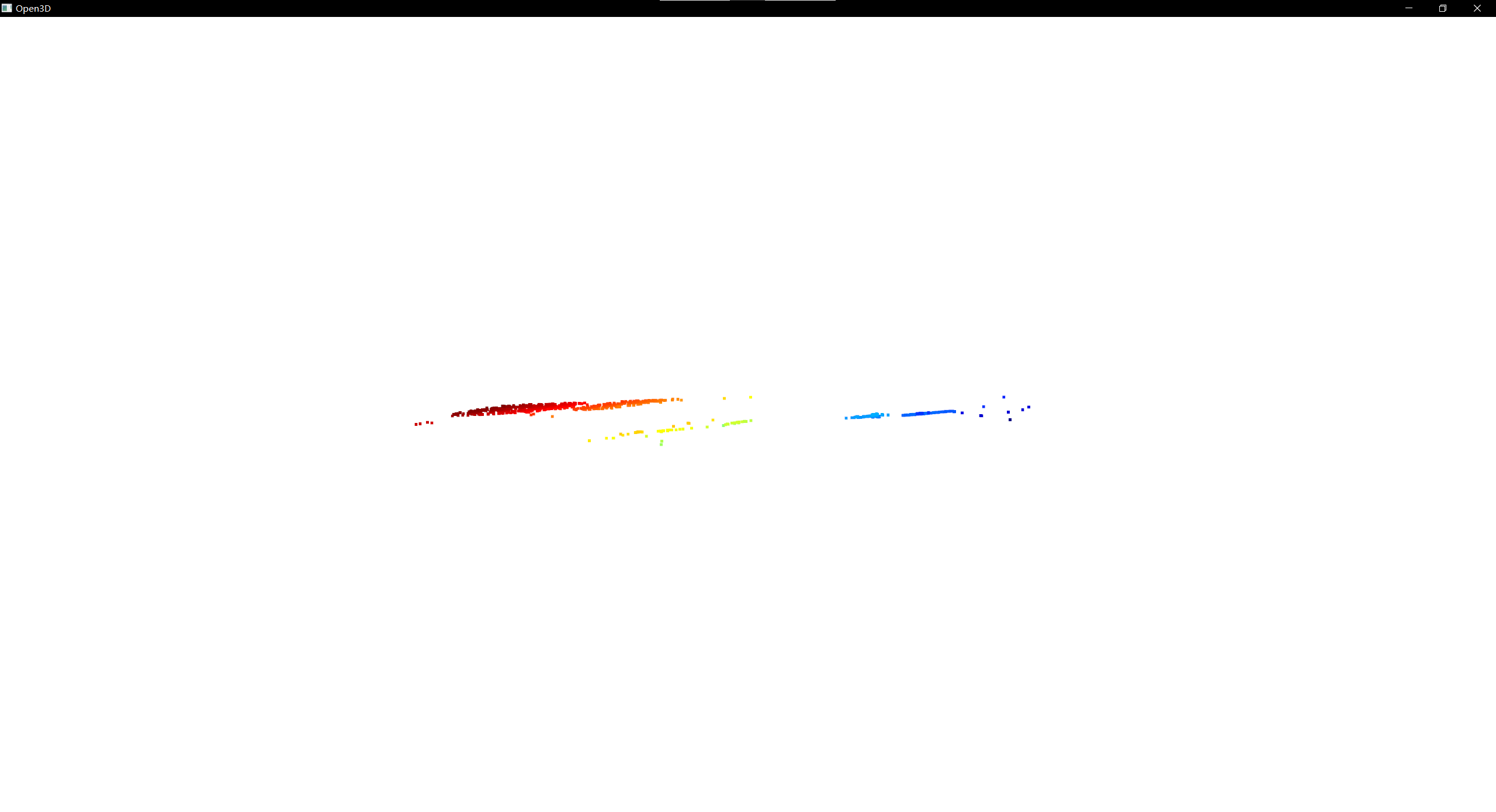


Figure 不矫正相机的重建结果

观察可以发现，无法看出任何物体。所以不矫正相机无法进行重建。