

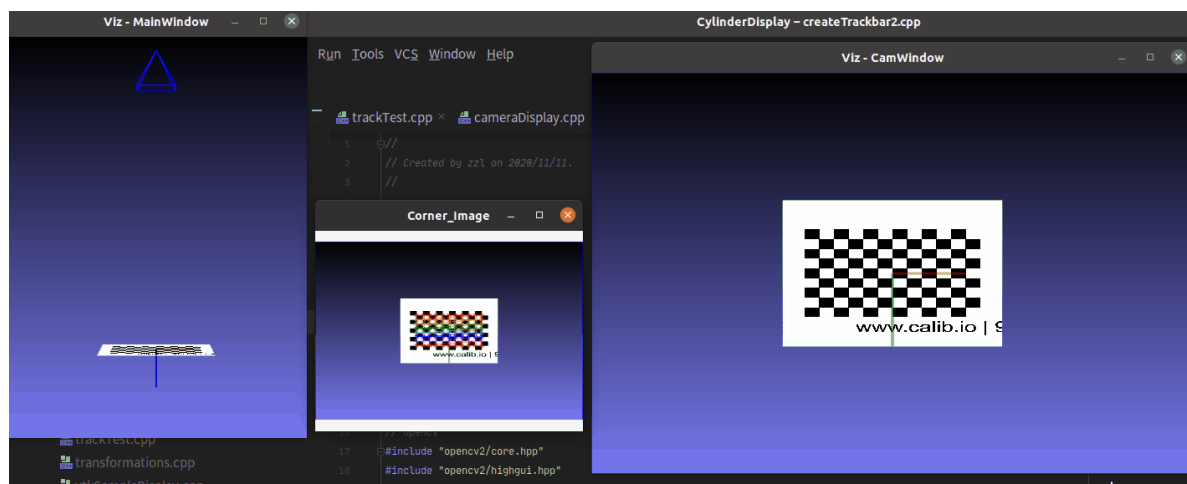
单目相机标定模拟

基于OpenCV中的Viz模块，虚拟一个相机，设置相机的内参数。然后在相机视野下放置标定板，通过相机标定算法，最终再获取相机内参数。

当然最终相机标定还是存在误差，我猜测主要原因是标定板旋转在变换的过程中，仿射变换导致图像质量下降，角点提取的不准确。

当然，这个项目的主要意图还是示意吧，重点在于自己创造数据，虚拟数据。

效果图：



创建窗口

第一步就是创建窗口。一个窗口为主窗口（从上帝视角看标定板的位置），还有一个窗口为相机窗口（相机所获取的图像）。

```
viz::viz3d mainWindow("Mainwindow"); // 创建主窗口
viz::viz3d camwindow("Camwindow"); // 创建相机窗口
mainWindow.spinOnce(); // 初始化
camwindow.spinOnce(); // 初始化
```

值得一提的是，在每个窗口都执行了一次spinOnce()的函数。若不更新这个窗口，会莫名报错误：

```
/home/zzl/Blog/CylinderDisplay/cmake-build-debug/Trackbar2

Process finished with exit code 139 (interrupted by signal 11: SIGSEGV)
```

Process finished with exit code 139 (interrupted by signal 11: SIGSEGV)

初始化相机

初始化相机主要包括：

1. 相机的内参数
2. 相机的朝向 (**WCameraPosition**)
3. 相机的位置 (**Affine3f**)，相对于世界坐标系的位置

```

// Init Cam
Matx33f Intrisic(700,0,360,0,700,240,0,0,1); // 相机内参数
viz::Camera Cam(Intrisic,Size(720,480)); // 设置相机
viz::WCameraPosition camOrient_main(Cam.getFov(),1.0,viz::Color::blue()); //
相机位置（主窗口）
viz::WCameraPosition camOrient_cam(Cam.getFov(),1.0,viz::Color::blue()); //
相机视角
Affine3f camPosition(Mat::eye(3,3,CV_32F),Vec3f(0,0,-8)); // 相机窗口中的观测位
姿(相机位置也就是观测位姿)

// Observe Position
Eigen::AngleAxisf Rotation_x(CV_PI / 2 ,Eigen::Vector3f(1,0,0)); // 主窗口中的
观测位姿
Mat RotationX;
eigen2cv(Rotation_x.matrix(),RotationX);
Affine3f ObsePosition(RotationX,Vec3f(0,20,-3));

mainwindow.setWindowSize(Size(Cam.getWindowSize().width/2,Cam.getWindowSize().h
eight)); // 设置主窗口大小

// Display camera in window
mainwindow.showwidget("Cam",camOrient_main,camPosition); // 显示相机
camWindow.setCamera(Cam); // 设置相机窗口的观测相机的内参数
camWindow.showwidget("Camera",camOrient_cam,camPosition);

```

其中 `setCamera` 为设置窗口观测相机的内参数。初始相机位置在距离世界坐标系Z轴的-8位置处（距离尺度），由 `camPosition` 定义。观测位置（主窗口上帝视角）由 `ObsePosition` 定义。

设置标定板图像

```

// Display an Image
Mat image = imread("/home/zzl/Blog/CylinderDisplay/data/chessboard.png",0);
// 读入图像
viz::WImage3D Image_main(image,Size(3,2));viz::WImage3D
Image_cam(image,Size(3,2)); // 设置图像
mainwindow.showwidget("Image_image",Image_main,Affine3f::Identity());
camWindow.showwidget("Image_cam",Image_cam,Affine3f::Identity());

camWindow.setViewerPose(camPosition); // 设置观测位置
mainwindow.setViewerPose(ObsePosition);

namedwindow("Corner_Image",WINDOW_NORMAL); // 角点可视化窗口名称

```

读入图像，并在两个窗口中显示图像。设置初始的观测位姿为“初始化相机”中定义中的位置。这里需要设置图像的大小（不是像素大小，应该是实际大小，有待验证），我并没有严格缩放标定板，这也可能是标定误差的来源。

设置一个窗口“Corner_Image”作为角点检测的显示窗口

可视化标定板位置

其原理就是：在相机窗口（camWindow）下获取相机的观测位姿（getViewerPose），并设置为当前的相机的位姿。这样就可以获得相机的实时图像。通过在主窗口（mainwindow）下，通过对观测位姿取反（固定相机不动），得到标定板的位姿，并实时更新。

这里的问题就在于如果对位姿直接取反，会因为平移向量取了负值，导致在相机的镜像位置。所以需要抵消相机平移距离的影响。

```
while(!mainwindow.wasStopped()){

    mainwindow.showWidget("Coordinate_main",viz::WCoordinateSystem(),Affine3f::Identity()); // 显示坐标系

    camwindow.showWidget("Coordinate_cam",viz::WCoordinateSystem(),Affine3f::Identity()); // 显示坐标系
    camwindow.setRenderingProperty("Coordinate_cam",viz::OPACITY,0.4); // 设置相机窗口下的坐标系的显示参数(透明度)

    camwindow.setWidgetPose("Camera",camwindow.getViewerPose()); // Refresh Pose in camwindow 相机位置与观测位姿重合，这样屏幕上的图像就是相机实际获取的图像
    // 抵消相机平移距离的影响
    Vec3f tmpTrans = camwindow.getViewerPose().inv().translation(); // 位姿直接取反，平移向量会变成负值，需要单独计算
    tmpTrans += camPosition.translation();

    // Detect Corner and draw 检测并绘制角点
    Mat tmpCornerImage = camwindow.getScreenshot();
    Mat DrawCorner = tmpCornerImage.clone();
    vector<Point2f> corner;
    bool find =
    findChessboardCorners(tmpCornerImage,Size(11,8),corner,CALIB_CB_ADAPTIVE_THRESH+
    CALIB_CB_NORMALIZE_IMAGE);
    if(find){
        drawChessboardCorners(DrawCorner,Size(11,8),corner,true);
        imshow("Corner_Image",DrawCorner);
        waitKey(50);
    }else{
        putText(DrawCorner,"Couldn't Find Corner",Point(DrawCorner.rows/2,DrawCorner.cols/2),1,2,Scalar(0,0,255),4,LINE_8,false);
        imshow("Corner_Image",DrawCorner);
        waitKey(50);
    }

    mainwindow.setWidgetPose("Image_image",Affine3f(camwindow.getViewerPose().rotation().inv(),tmpTrans)); // 更新主窗口下标定板的位姿

    mainwindow.spinOnce(1,true); // 循环刷新窗口
    camwindow.spinOnce(1,true);

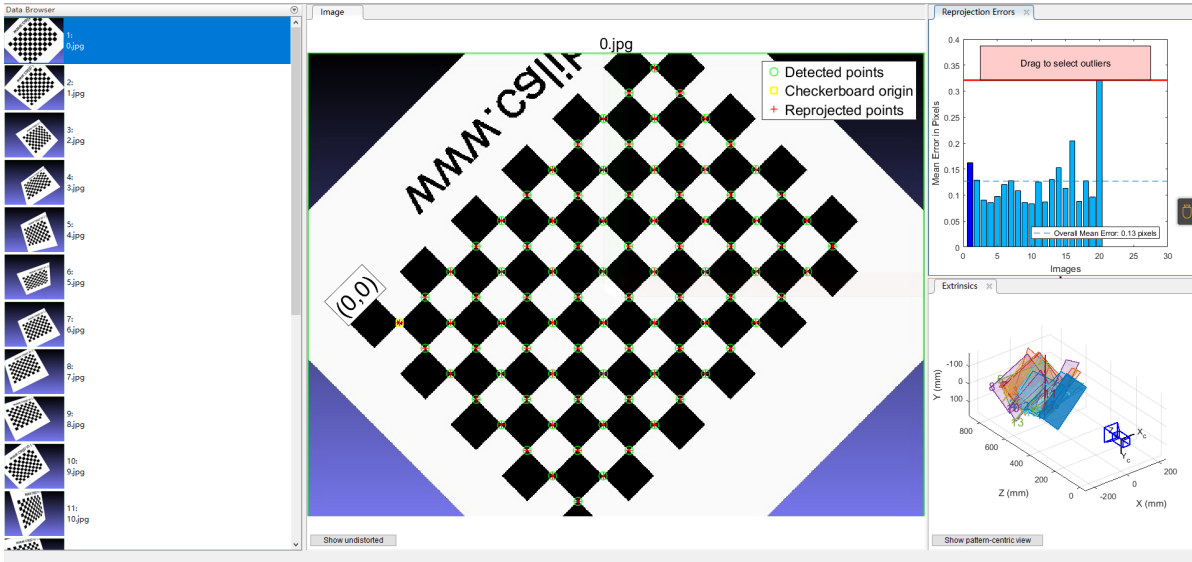
}
```

最后的最后，检测角点，然后执行相机标定就好了。

相机标定

提取了相机图像，导入Matlab、OpenCV中进行标定，最后结果为：

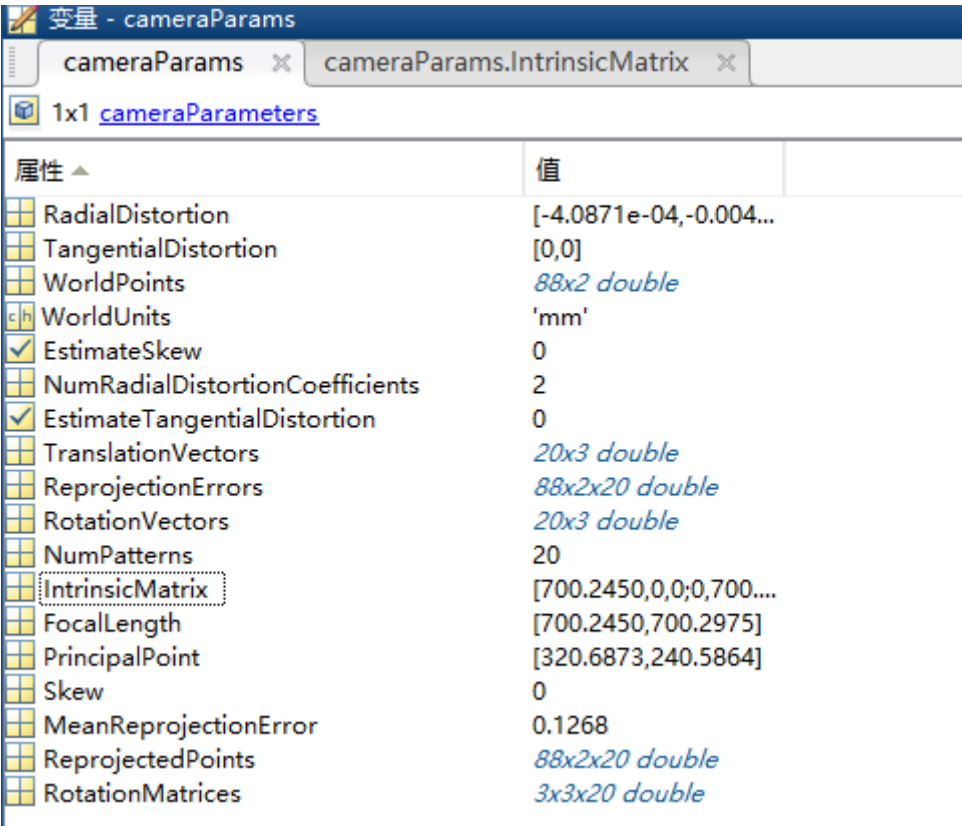
Matlab



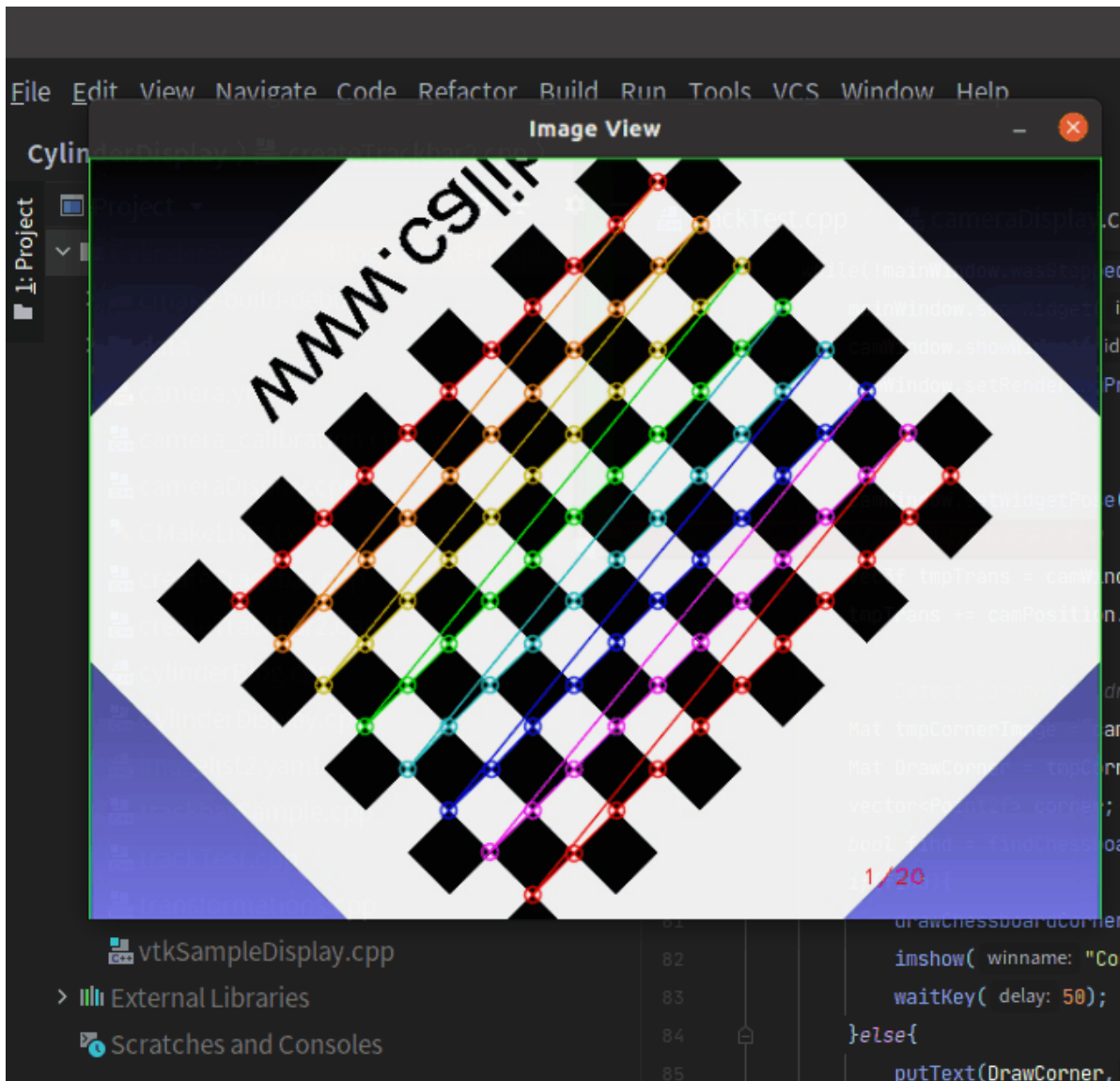
内参数：

700.245041295754	0	320.687252173318
0	700.297465024913	240.586408260201
0	0	1

重投影误差为：MeanReprojectionError = 0.126787073893052



OpenCV



重投影误差:

```
/home/z2l/Blog/CylinderDisplay/cmake-build-debug/calibration
RMS error reported by calibrateCamera: 0.192629
Calibration succeeded. avg reprojection error = 0.1926287

Process finished with exit code 0
```

内参数:

```
camera_matrix: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [ 7.0032084773881923e+02, 0., 3.1904012287367055e+02, 0.,
|       7.0032523785915430e+02, 2.3977747260855980e+02, 0., 0., 1. ]
distortion_coefficients: !!opencv-matrix
  rows: 5
  cols: 1
  dt: d
  data: [ -2.4806931071062219e-03, 1.3023210689646942e-02,
|       1.7170418911115857e-04, -3.0869009135838838e-04, 0. ]
avg_reprojection_error: 1.9262865827049289e-01
```