**Grasp Pose Detector使用指南**

1. GPD 环境配置

1.1包含目录及头文件

#include "../gpd\_1.0.0/include/grasp\_pose.h"

**1.2库目录及lib文件**

#pragma comment (lib, "../gpd\_1.0.0/lib/gpd\_grasp\_pose.lib")

**1.3动态链接库文件**

gpd\_1.0.0\bin目录中下列文件复制到程序可执行模块即.exe所在的目录。

gpd\_grasp\_pose.dll

gpd\_camera\_data.dll

gpd\_pose\_estimation.dll

gpd\_recognition.dll

gpd\_registration\_3d.dll

gpd\_segmentation.dll

**1.4 第三方依赖库文件**

gpd\_1.0.0\3rdparty目录中文件为视觉算法第三方依赖库，如果没有配置第三方依赖库则需要复制对应动态链接库文件到程序可执行模块即.exe所在的目录。第三方依赖库版本如下：

PCL 1.9.1

OpenCV 3.4.5

Glog 0.4.0。

注：1、gpd\_1.0.0为视觉算法初始版本号，此处根据项目实际应用版本号进行配置。

2、调试设置：项目->属性->调试->工作目录 ->$(SolutionDir)$(Platform)\$(Configuration)\。

**2. GPD项目应用**

**2.1 建立项目文件目录**

以比亚迪项目为例：定义项目名称“GPD\_BYD”，电控壳类别编号“1”，托盘类别编号“2”。

复制项目模板文件夹GPD到程序可执行模块即.exe所在目录，根据项目名称修改文件夹名称为GPD\_BYD。根据抓取目标类别数量复制object\_x文件夹，修改文件夹名中“x”为对应抓取目标类别编号。

**2.2 确定视觉算法方案**

根据项目需求和项目总体方案，更新文档project\_record.docx相关内容并确定视觉算法方案。

**2.3 制作抓取目标3D模型**

模型制作方法参考“附录1”，通常情况下一个类别抓取目标只需要制作一个模型，几种特殊情况说明如下：

* 某些情况下需要对一个类别抓取目标制作多个3D模型，例如复杂形状物体的不同视角模型
* 某些情况下需要对一个类别抓取目标同时制作3D模型和refine模型，例如高精度抓取场景

**2.4 视觉算法调用方式**

以比亚迪项目编号 x = 1 电控壳为例：

std::string config\_object\_1 = " GPD\_BYD\\object\_1\\config\\grasp\_pose.json"; //定义抓取目标1 配置文件

std::shared\_ptr<gpd::GraspPose> p\_object\_1\_(GetModelBasedPtr(config\_object\_1));//初始化抓取目标1 实例指针

…… //3D相机数据获取、机器人控制等代码

p\_object\_1\_->SetInputPointCloud(object\_points);//输入点云

p\_object\_1\_->GetGraspPose(&object\_pose); //获取位姿（根据项目流程调用）

**2.5 视觉算法参数设置与性能优化**

参数说明参考“附录 2”，离线测试和性能优化需要保存PLY格式现场点云文件scene.ply，放置于\项目目录\object\_x\pointcloud\。

附录

1. 抓取目标3D模型制作

1.1 3D建模

1.1.1 3D相机建模

a) 根据项目需求确定3D相机工作距离，调节参数使得采集点云数据完整、边缘清晰、噪点尽量少；

b) 模拟抓取目标实际摆放情况扫描一帧点云作为3D模型。

注：相机工作距离、参数设置或者摆放位置发生较大变化的情况下，需要重新制作模型。

1.1.2 3D机械模型/3D扫描仪建模

利用抓取目标3D机械模型转换格式或者3D扫描仪建模。

1.2 3D模型坐标系建立

根据项目需求确定抓取位姿，参考抓取位姿建立3D模型坐标系，坐标系精度尽量高（平移误差小于0.1毫米，旋转误差小于0.1度）。

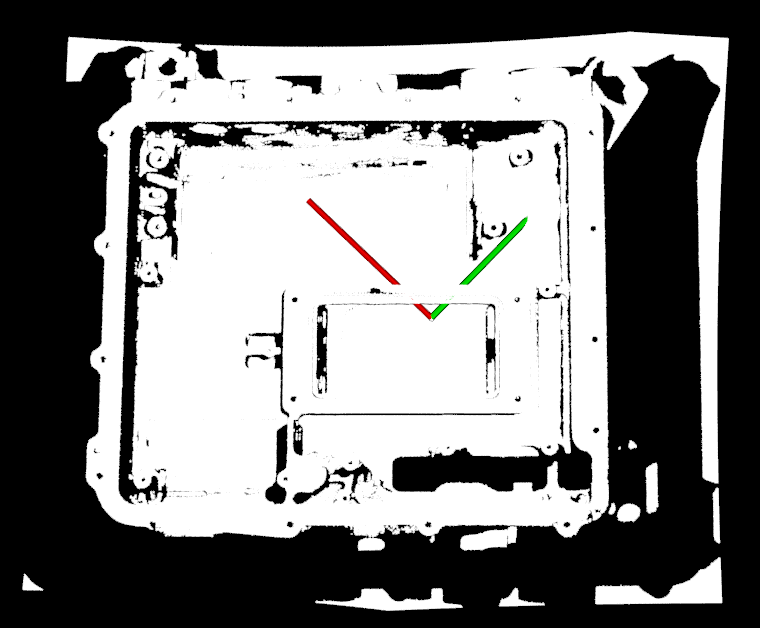
 

图1. 机器人抓取位姿 图2. 3D模型坐标系

1.3 3D模型后处理

根据视觉方案需求，选择保留3D模型中所有点或者进行特定分割，输出PLY格式模型文件"model.ply"。

根据视觉方案需求，选择是否需要制作refine模型，输出PLY格式模型文件 "model\_refine.ply"。

注：a)3D模型放置位置为\项目目录\object\_x\model\_3d\；

b)单个类别对应多个refine模型， 按照model\_refine\_(a.b.……)方式命名；

c)采用3D相机建模方式在保存PLY模型文件时如果有格式转换，需要保证模型点间距与采集的原始点云点间距一致。



图3.refine模型制作

2. 视觉算法参数说明

2.1 配置文件 grasp\_pose.json

{

/\*Porject Name: GPD\_BYD\*/

/\*Object Number: object\_1\*/

//Pose Estimation

"Visualization": false, //可视化开关

"CameraOffline": false, //相机离线测试开关

"InstanceMethod": "CG", //实例提取方法

"RefineModelNum": 2, //优化模型个数

"RefineRegistrarion": "ICP", //配准方法 IA\_ICP | ICP

"ObjectModelPath": "model\_3D//model.ply", //3D模型文件名

"PointCloudPath": "pointCloud//scene.ply", //离线测试点云文件名

//Grasp Pose of Model

"UseModelPose": true,

"X": -0.138, //抓取位姿 X (毫米)

"Y": -0.031, //抓取位姿 Y

"Z": 0.476, //抓取位姿 Z

"RX": 0.39, //抓取位姿 RX （度）

"RY": 0.04, //抓取位姿 RY

"RZ": 132.47 //抓取位姿 RZ

}