# 实验所完成的主要研究工作

## 主要研究工作概述

### 概述

近年来,计算机视觉技术领域取得了长足的进展,在目标检测、语义分割等方向都有大量出彩的技术出现,同时计算机视觉技术的落地与应用也愈发重要.本项目聚焦我国的视觉障碍人群,利用语义分割,光流估计等计算机视觉技术,构建基于双目相机的盲人户外智能语音辅助系统,帮助盲人更加安全便捷地进行户外行走.

首先,项目研究了室外语音辅助系统的搭建与实现,第一步构建手机终端程序,作用是采集定位信息与方位朝向信息.第二步构建服务端,运行盲道分割与偏移判断,障碍物检测与避障等核心代码,最后通过蓝牙耳机对用户进行偏离警示.

之后,对感知与规划过程进行了重点研究.利用自建盲道人行道数据集,通过轻量级实时语义分割网络bisenetv2进行训练,实现了对盲道与人行道的像素级语义分割.

### 系统核心功能

通过利用本系统,视障人士出行时得以选择安全且正确的行走路线,能够在语音的引导下走上盲道同时避开途中的障碍物,当遇到需穿过马路情景时能够引导用户在斑马线上通过马路,最终到达正确目的地。

本系统的核心技术包括感知,规划,决策,控制四个方面,其中重点与难点在于感知与规划.

## 基于神经网络的室外目标语义分割方法与后处理研究

图像语义分割任务是计算机视觉领域重要的基础研究任务，其任务是给定一张图像，要求算法为图像中的每个像素预测其所属的语义类别.而针对盲道与人行道进行语义分割是智能导盲系统的重要组成部分, 在兼顾实时性的同时提高盲道分割的效率和质量具有重大意义.

盲道与人行道作为特征较为明显的两类道路标识线,在输入图像较为清晰,背景环境简单,样本类型单一的情况下,利用二值化,canny边缘检测等传统图像处理算法能够取得较为良好的分割效果[1],且分割速度极快,但由于现实环境复杂多变,相机防抖的缺失,盲道种类繁多而传统图像处理方法依赖人工设计特征等原因, 而深度学习泛化能力较强,利用深度学习技术进行语义分割更加适宜本项目的需求.

### 系统感知需求

本系统的核心宗旨是让盲人尽可能的走在安全的地方,而盲道与人行道作为相对更加安全的区域是我们所提倡前往的,因此对于盲道与人行道的感知不能够仅仅停留在检测的程度,还需要进行进一步的语义分割,提取延伸方向,与抽象化区域等工作.

最终需要感知到的信息包括如下:

1. 盲道与人行道的走向
2. 盲道与人行道的可行走区域
3. 盲道起点与用户当前所处位置的角度与距离

此处主要使用相机获取的RGB图像信息.  
 进行规范化描述即为:

input:predicted\_img
  
   
 output:
  
 blind\_road\_find=1//是否发现盲道
  
 blind\_road\_distance=0.5m//盲道与使用者的距离
  
 blind\_road\_angle=-30°//盲道在使用者的左边30度角位置
  
 on\_blind\_road=1//使用者是否位于盲道上
  
 blind\_road\_departure//使用者是否正在偏离盲道
  
   
 sideways\_find=1//是否发现人行道
  
 sideways\_distance=3.5m//人行道与使用者的距离
  
 sideways\_angle=30°//人行道在使用者的右边30度角位置
  
 on\_sideways=1//使用者是否位于人行道上
  
 sideways\_departure//使用者是否正在偏离人行道

### 基于 BiSeNet V2 网络模型的盲道与人行道分割

BiSeNet是一种用于实时语义分割的网络结构,于2018年由Yu[2]等人在ECCV2018上提出,其使用用于保留原图像信息的Spatial Path和用于获取较大感受野的Context Path,构成了一个小型且高效的双边分割网络.2020年Yu等人再次提出了BiSeNet V2[3]

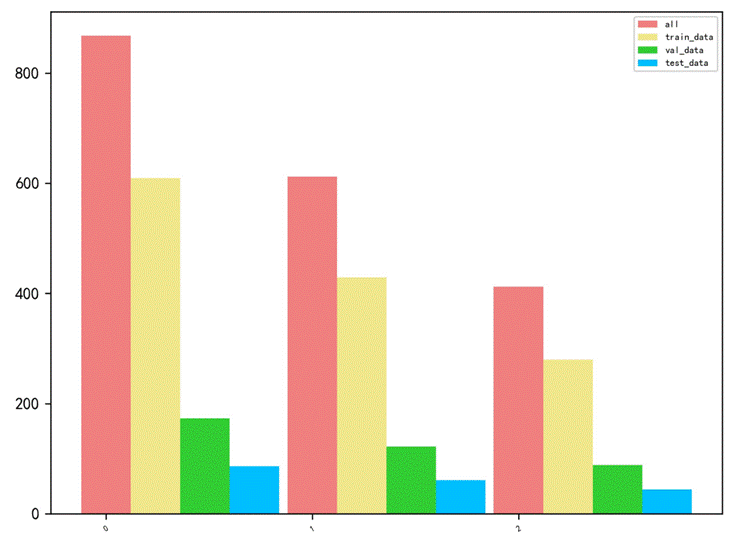
我们针对盲道与人行道自身的特征及实际环境下采集的图像特点进行预处理以及数据增强，建立了盲道数据集,并利用该数据集基于BiSeNet V2进行训练,得到最终模型,其可以解决在满足实时性的基础上解决盲道与人行道的语义分割问题.

#### 实验数据集的构建

由于目前没有专门的大型盲道分割数据集, 因此我们采取了人工拍摄了盲道人行道图像与网络中盲道人行道图像数据集[4]相结合的方式,为了保证模型的泛化能力,我们采用了不同时间,不同天气,不同场景的盲道与人行道图像,利用Labelme进行标注,得到标注后的json文件, 通过编写python脚本批量将 json 文件转换为包含标签掩码的图片.



最终我们的数据集包括868张打标签的 512 × 512的图像,其中包含盲道标签的图像612张,包含人行道标签的图像412张,训练集:验证集:测试集=7:2:1,



以下是标注数据集的一些实例

|  |
| --- |
| fig: |
| fig: |

#### 训练细节

在正式训练前,采用了数据增强技术,对数据进行随机垂直与水平翻转,随机图像截取,随机图像模糊,随机亮度,随机对比度,随机色调等处理.

损失函数是对系统在不同参数值下损失的描述，衡量的是模型的预测值与真实值的不一致程度，是衡量一个模型好坏的重要标准,我们选取交叉熵损失函数作为损失函数. 训练过程中选用的优化方法是随机梯度下降法(Stochastic Gradient Descent, SGD), 学习率动态调整方式为polynomial.

训练时的将Batch Size设置为4,学习率设置为0.005,迭代70轮.

#### 实验结果分析

总体来说,loss收敛良好,在验证集上miou达到0.9182,Kappa一致性检验值为0.936,一致性较好.

|  |
| --- |
| fig: |
|  |

在验证集上miou达到0.9182

以下是测试集上的效果与标注效果比对,可见除较远处人行道难以识别外,模型效果良好. 在实际测试的推理过程中,每张图片平均预测用时约253ms,能够满足系统实际需求.

|  |
| --- |
| fig: |
|  |

### 基于语义分割结果获取盲道与人行道走向

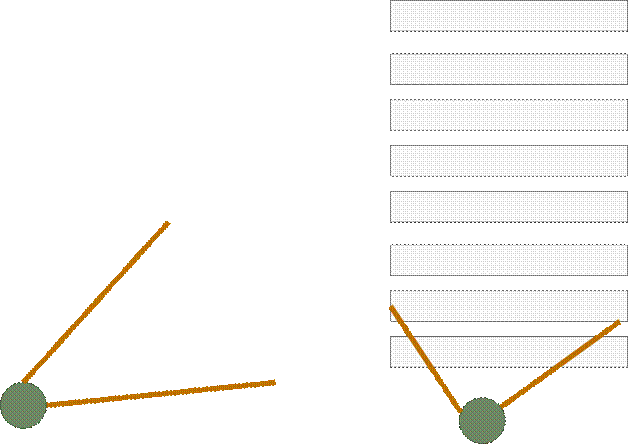
由于要平衡"走安全的路"与"走便捷的路"两种决策,为避免走安全的路却走在错误的方向的情况出现,需要获取盲道与人行道的走向,以实现行走大方向的不偏移.

由于语义分割后的结果是像素级别的分割,需要进行进一步分析出更多抽象化特征

对于盲道而言,由于是一整块区域,无需进攻更多处理,只需使用最常用的最小二乘法获取最优拟合直线即可.而对于人行道则需要进行更多处理.

#### 人行道走向获取

人行道检测存在以下典型应用场景:



考虑到盲人智能辅助系统的全局规划需求,我们前文进行语义分割得到的只是像素级别的类别判定, 未对人行道的走向分析与区域抽象化进行研究,因而只能够判定某区域是否存在人行道,当面临更加复杂的需求如对比人行道走向与全局路径时,便难以进行.

我们采用如下算法进行人行道区域与走向的提取方式:

Input:
  
 语义分割预测结果图像
  
   
 Algorithm:
  
 初始化阈值bw\_height,bw\_width;
  
 处理图像,过滤出只有人行道的二值图;
  
 对图像进行侵蚀处理;
  
 对此二值图像提取轮廓列表(道路候选点)contours;
  
 for contour in contours:
  
 获取轮廓contour最小外接矩形box;
  
 如果 box.高>bw\_height 或 box.宽>bw\_width:
  
 box[0]加入left\_line列表;
  
 box[2]加入right\_line列表
  
 利用RANSAC算法过滤与拟合道路边界点;

##### 道路候选点提取,过滤与拟合

语义分割后能够得到像素级别的类别判定, 由于像素颜色代表着分类标签,可以容易地过滤出只有人行道的二值图,通过侵蚀处理,对当前的二值图像通过光栅扫描进行轮廓提取[5],并求取每个轮廓的最小外接矩形的四个顶点坐标,分别作为左边界点与右边界点.此时便获取了人行道的候选点.

随机一致采样性(RANSAC)算法假设数据中包含正确数据和异常数据(或称为噪声),常中被用来直接过滤提取道路边界点后的噪点, 我们将以上获得的道路边界点用RANSAC曲线模型进行拟合,从而得到边界线与交点.

如下是对一个人行道区域进行抽象的完整过程

|  |
| --- |
| fig: |
| fig: 侵蚀 |
| fig: 计算最小外接矩形 |

## 基于光流估计的盲道偏移检测研究

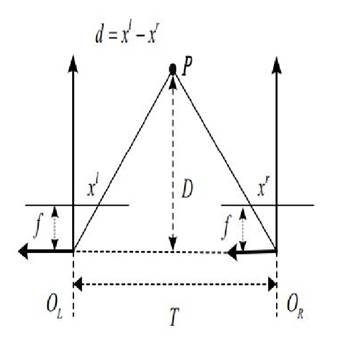
### 测角与测距

#### 双目立体视觉测距原理

在计算机视觉(computer vision)相关理论中，双目立体相机在同一时刻拍摄左右两幅图像对，通过提取图像特征点并进行立体匹配，利用三角关系计算目标点在左右图像中的视差，由此计算出环境的立体深度图，可以测量计算出环境物体的立体信息。

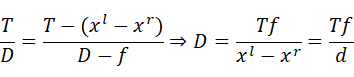
双目视觉计算深度首先要标定双目相机的内参外参，通过标定推导图像坐标系到摄像机坐标系转换关系和摄像机坐标系到世界坐标系的转换关系, 使用张有正法计算相机焦距 fig: 和成像原点 fig:, 通过 Brown提出的算法解算图像畸变系数。主要获得左右摄像机的内外参数、相对几何关系, 包含平移向量和旋转矩阵两个参数。  
 然后进行左右图像畸变校正, 畸变校正后的左右图像对是精确对应的, 使得双目视图的对极线在相同水平面。通过上面方法获得相机的内参和外参, 双目相机畸变校正实验通过openCV 对应函数编程实现。

第三步是进行左右视图的立体匹配，通过立体匹配在图像对中计算对应匹配点，算法第一步是构建以能量为基础的评估函数，根据最优化理论最小化评估函数来计算配对特征点的视差值。在数学原理上双目立体匹配算法是一个凸优化目标函数代价最小化问题，能量评估函数的选择对立体匹配的效果起到至关重要的作用。



图

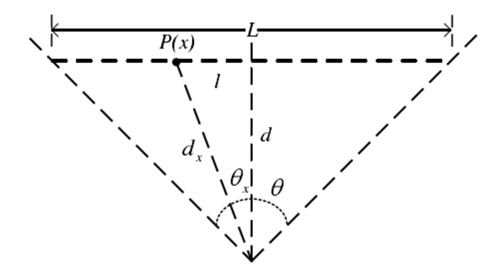
最后一步是根据三角关系计算深度图，如图 2-3所示, 点 fig: 在规定区域内双目相机中成像的水平坐标为 fig: 和 fig: 为摄像机的焦距, fig: 表示 点与相机平面之间的垂直长度, 双目相机的两个光分坐标别用 fig: 来表示, fig: 表示基线长度是双 目像机光心间的连线长度。对于任一像素点以d来表示两相机间的视差值, 结合三角形相似定理完成深度公式推导可得下式。



根据相机内外参和视差d计算出深度D。

#### 测定物体与盲人的相对角度

测距与测角的原理如下图:



本系统采用的是双目相机类型是Realsense D415,其视场角fig:为69.4度,采集图像的像素为1280\*720,令X=1280,p的位置为(x,y)，已知p点的深度距离为fig:

\1. 计算距离d米远处的相机的真实可测水平长度 fig:

\2. 计算实际中点 p 到相机视角正前方方向的垂直距离 fig:

\3. 根据fig:与fig:计算目标的角度fig:。

### 静态判断盲人与盲道的相对位置

为了解决这个问题,我们将图像坐标系映射到现实,利用**盲道边界线的深度信息、角度信息与图像坐标系中坐标**得到,以盲人为中心的现实坐标系中盲人与两边界的距离情况.具体转换如图所示:

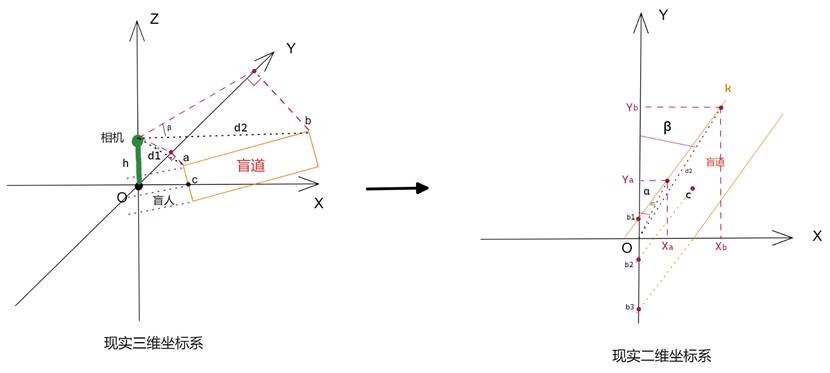


图1 转换坐标系

#### 坐标系定义

成像变换涉及到不同坐标系统之间的变换。考虑到图像采集最终结果是要得

到计算机里的数字图像，在对3D空间景物成像时涉及到的坐标系统主要有以下

几个(如图2)：

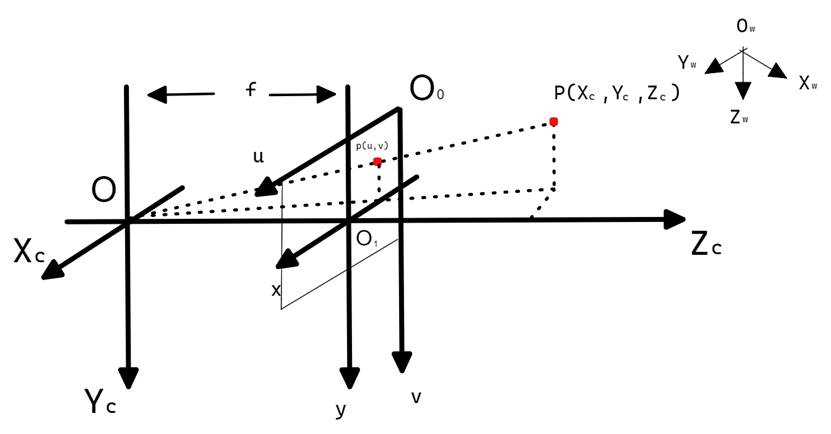


图2 摄像机成像模型及坐标系定义

世界坐标系 fig: : 也称真实或现实世界坐标系, 它是客观世界的绝 对坐标系。一般的 3D 场景都是用这个坐标系统来表示的。  
 摄像机坐标系 fig: ：以摄像机为中心制定的三维直角坐标系统。原 点位于摄像机的光心(Optical center), 其 fig: 平面位于焦平面(focal plane)上, 并 且 fig: 轴同光轴(optical axis)重合。  
 成像平面坐标系 fig: : 定义在摄像机成像平面上以物理单位表示的二维坐 标系。原点是摄像机光轴和成像平面(image plane or retinal plane)的交点, 且 fig: 轴 和 fig: 轴分别与摄像机坐标系的 fig: 轴和 fig: 轴重合。  
 像素坐标系 fig: : 计算机内数字图像所用的坐标系统, 原点位于数字图像 的左上角, fig: 轴向右, fig: 轴向下, 分别表示某个像素在图像数组中的列数与行数。

像素坐标系:计算机内数字图像所用的坐标系统，原点位于数字图像的左上角，u轴向右，v轴向下，分别表示某个像素在图像数组中的列数与行数。

#### 盲人坐标系

由于人是处于三维空间内的，所以可以很自然地构建出一个盲人的三维视角，同时以盲人的脚为坐标系原点、以盲人朝向为Y轴正方向，得到一个盲人与盲道相对位置的三维坐标系（如图3）。

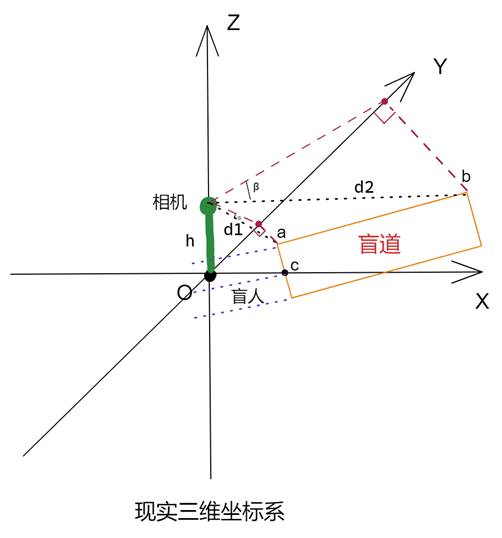


图3 现实三维坐标系

#### 平面二维坐标系建立

盲人是否处在盲道上是通过盲人的脚是否处在两盲道边界线内来判定的，也就是说需要得到一个以盲人的脚为坐标系原点，盲人朝向为坐标系Y轴正方向的二维平面坐标系。这个坐标系可以近似看做是盲人的俯瞰图（如图4）。

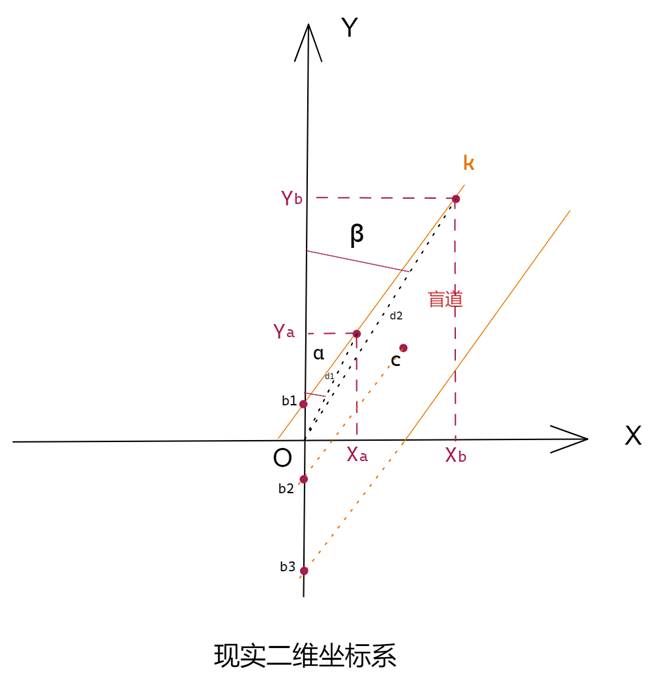


图4 现实2维坐标系

为此，需要获得盲道延伸至盲人脚下的两条边界线，选取通过语义分割得到的盲道轮廓边界的上fig:的一个点fig:和下的一个点fig:来求取盲道的斜率fig:。

已知fig:点的深度fig:和fig:点的深度fig:、点的水平角度fig:和点的水平角度fig:，可以得到点的坐标fig:，以及点的坐标fig:。通过这两点的坐标就能求出盲道的斜率k。

fig: h为相机高度

fig:

fig:

fig: h为相机高度

fig:

fig:

fig:

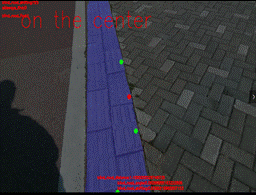
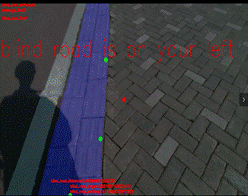
由于盲道大多数情况下都是直的，所以盲道内的斜率都为fig:，至此就能求出盲道在fig:轴上的截距fig:和fig:，接着通过计算原点距离和的距离以及是否在和内来判定是否在盲道内。由于能得到盲道的底部中心位置，所以盲人距离底部中心的偏移程度fig:也能求出来。



据此能判断盲人的偏移程度。如果盲人处于中心位置，那原点应该和fig:重合，也就是fig:，如果盲人稍有偏移，那会在原点的上下方偏移一点的位置，fig:，如果盲人完全偏移，那会在原点的上下方偏移很多的位置，fig:。

#### 实时检测实验

在户外盲道上进行测试，判断算法的适用性。图5是人与盲道的相对位置的检测结果与数据

(a)位置1(b)位置2(c)位置3

图5 基于坐标系的盲道偏移检测

|  |
| --- |
|  |
| 位置判定 |
| 盲道位置判定 |
| 偏移指数 |
| 真实情况 |

表1 定位结果

### 基于视频判断盲人的行走偏移趋势

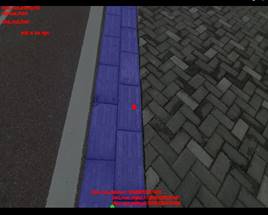
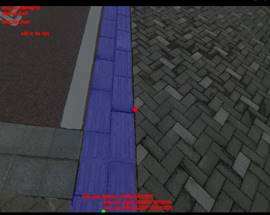
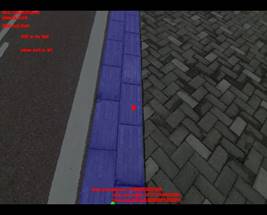
#### 连续点偏移策略

为了更好地矫正盲人的行走路线，会抽取连续三帧对盲人的行走趋势进行判定。如果盲人连续三帧向盲道的一侧走去，就判定盲人有偏离盲道的趋势，并语音播报矫正盲人。

如果盲人未在盲道内，就判断盲道在盲人的哪一侧，并语音播报指示给盲人听。

\1. 获取输入的连续三帧数据

盲人的行为是动态的，不能通过单单的一帧去判断盲人的动向，因为正常人走路是摇晃的，并不会呈直线走路，所以只能通过连续的三帧图像去粗略判断盲人的大致走向。

(a)位置1(b)位置2(c)位置3

图6 偏移趋势测定

\2. 给出偏移趋势

行人在这三帧中是不断向右偏移的，所以判定为行人有偏离盲道的趋势，并向盲人提出矫正。

|  |
| --- |
|  |
| 偏移方向 |
| 偏移指数 |
| 真实情况 |

表2 偏移数据

#### 基于稀疏光流的盲人行走偏移趋势估计

光流[1]法是利用图像序列中像素在时间域上的变化以及相邻帧之间的相关性来找到上一帧跟当前帧之间存在的对应关系，从而计算出相邻帧之间物体的运动信息的一种方法.

其基于视频连续图像中像素亮度一致性和相关性以及邻域元素结构相似性等原则,估计相邻帧中像素的对应关系,求取物体在成像平面中的运动矢量,而稀疏光流则是对图像稀疏特征点的运动跟踪。

考虑到算法的实时性,相较于稠密光流,稀疏光流能够更加满足实时性的需求,我们采用Lucas-Kanade算法进行计算稀疏特征集的光流.

##### 准备工作

考虑到本系统所要求的实时性,我们希望光流判断用时要少于0.15s,且不易受移动物体的影响.

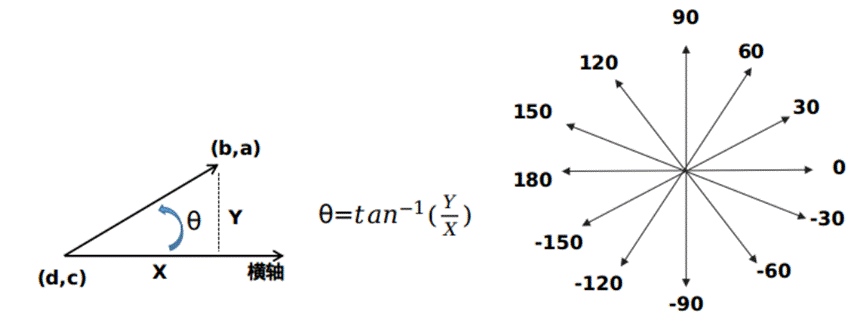
为了方便统计光流向量,我们生成HOF光流直方图进行可视化特征.即计算光流向量与横轴的夹角,计算的夹角结果分配到各自的区间（这里面每**30度**定义一个区间）,即如下图所示,将0-30度作为0类,30-60度作为1类向量,自从得到12类特征向量.

考虑到在盲道行走场景中,图像是运动的,图像中的像素点的向后移动(我们将本分类向量中即**2类与3类向量**近似看作像素点向后移动)即代表着相机(盲人)的向前移动.

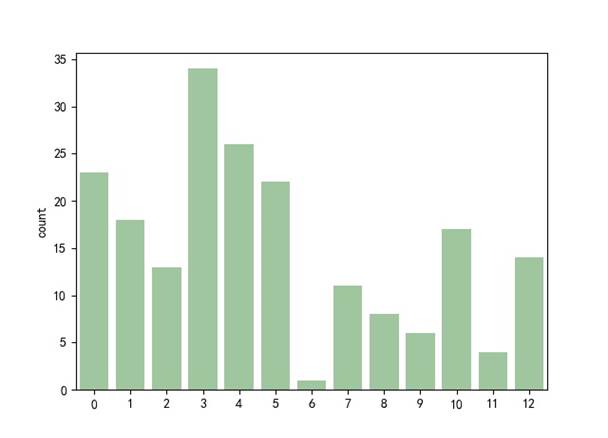
此外,我们假设正常情况下盲人不会倒退行走,因此对于像素点向前偏移的矢量,我们更加关注其水平分量.

因此我们将类别为0,1,10,11,12的特征向量代表像素点向左偏移,即人向右偏移.

同理我们将类别为4,5,6,7,8,9的特征向量代表像素点向左偏移,即人向右偏移.



得到类如下图的光流直方图



##### Lucas-Kanade算法

Bruce D. Lucas 和 Takeo Kanade在1981年提出的Lucas Kanade（LK）算法成为了稀疏光流算法的重要技术，它仅依赖于围绕某个兴趣点的一些小窗口导出的局部信息.

LK算法是一种两帧差分的光流估计算法，其基本思想基于以下三个假设。

**亮度恒定**：场景中目标图像的像素看起来在帧到帧移动是不发生改变。对于灰度图像（对于彩色图像同适用）这意味着像素的灰度值不会随着帧的跟踪改变。

**时间持续性**：图像上相机的移动随时间变化缓慢。实际上，这意味着时间的变化不会引起像素位置的剧烈变化，这样像素的灰度值才能对位置求对应的偏导数。

**空间一致性**：场景中相同表面的相邻点具有相似的运动，并且其投影到图像平面上的距离也比较近。

后人提出用图像金字塔的方法来解决大运动的跟踪的问题, 结合Lucas-Kanade方法,通过建立金字塔,在多尺度下计算光流, 金字塔光流的思想是对每个图像帧向下采样,分别建立多级金字塔,当采样到足够小以后,相邻图像帧之间的运动将变得很小,以至于可以看成是物体的运动随时间变化很缓慢的情况,这时候就可以用Luacs-Kanade方法计算目标的光流,再将计算出来的光流向底层投影,计算下一层的光流,直到估算出原图像帧的光流。

##### 特征点选取

为了推断出更多规律,我们首先选取**完全直线行走**场景进行测试

为了寻找能够更好描述场景运动的特征点选取方法,我们采用如下特征点选取策略进行测试与对比:

\3. Shi-Tomasi角点检测[2]并选取最多100个强角点

\4. 遮挡除盲道外的区域,角点检测选取最多100个强角点

\5. 盲道正中心的+-200像素,间隔20

\6. 盲道区域内随机选取100个特征点

\7. 1920*1280分割成16个480*320的小块,考虑到行走过程中像素点的移动可能会偏离图像区域,只选取中间9个的区域进行光流判断

\8. 1920*1280分割成128个120*160的小块,选取中心进行判定

此外为了尽量减少光流计算错误的情况,我们去除横向或纵向移动太大的特征向量.正如前文所讨论的我们将**2\***\*类或3类向量为占比最高量看做盲人向前直线行走的标志.\*\*

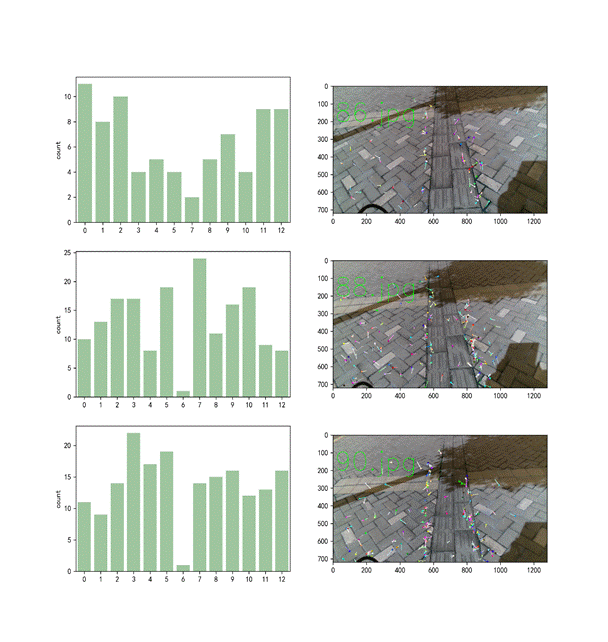
我们选取直线行走的三帧并统计其判断直线准确率与运行速度,正确率表格如下

|  |
| --- |
| **选取策略** |
| 角点检测选取最多100个强角点 |
| 遮挡除盲道外的区域,角点检测选取最多100个强角点 |
| 盲道正中心的+-100像素,间隔20 |
| 盲道分为3块,3块中心的+-50像素,间隔20 |
| 取盲道区域10\*25 |
| 取盲道区域20\*50 |
| 全局分块为20\*50 |
| 全局分块为20\*100块 |

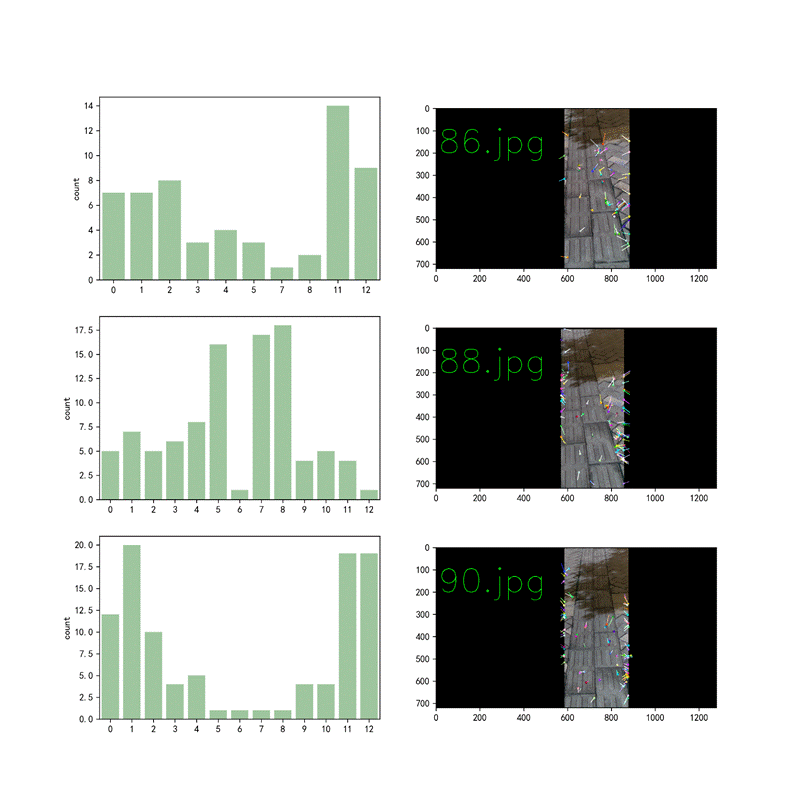
考虑到本系统所要求的实时性,我们希望光流判断用时要少于0.15s,另外考虑到摄像机拍摄距离有限,盲道具有更高的辨识度,同时不易受移动物体的影响,我们对**取盲道区域特征点**的策略进行进一步研究.

部分选取策略详细过程记录如下:

###### Shi-Tomasi角点检测并选取最多100个强角点

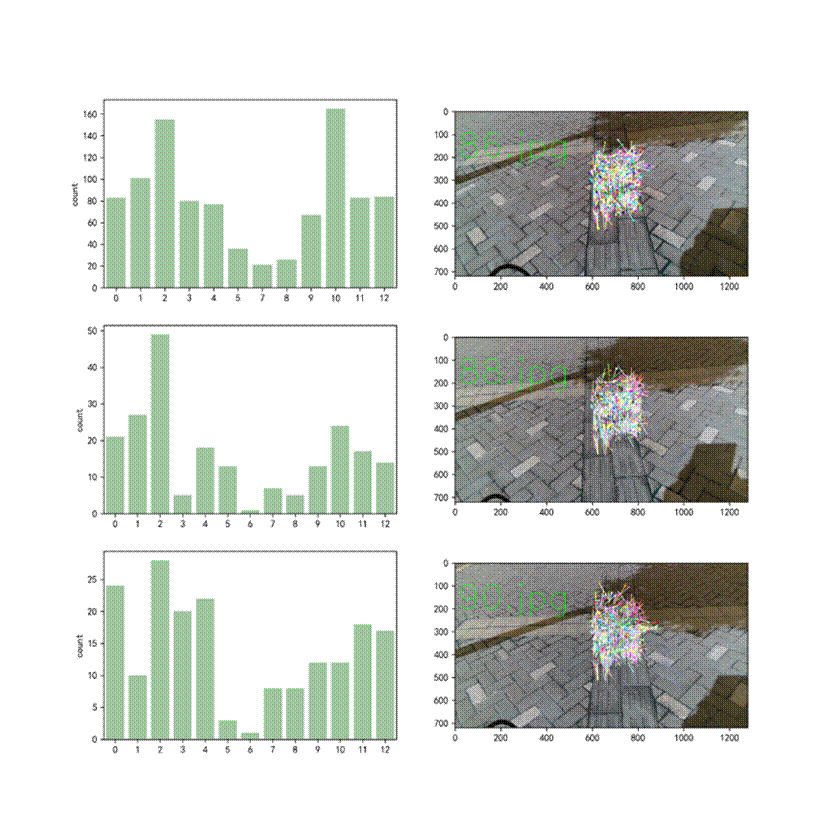


###### 遮挡除盲道外的区域,角点检测选取最多100个强角点



###### 取盲道区域20\*50

平均用时0.1s



###### 全局分块为20\*50

去除较远处点与边界处点  
 平均用时约0.1s

##### fig:

##### 选取盲道区域特征点策略的研究

选取更大数据量, 对取盲道区域20*50的策略进行进一步研究,发现正确率略有下降,而****进一步细化****为40*50进行取样后效果良好,平均速度0.15s亦可接收.

此外随着选取区域一再靠近使用者,直行的特征向量占比更加突出,平均用时0.1s,运行效率良好.

因此我们选取取盲道区域40\*50的特征点选取方法进行特征点选取,并进行进一步的光流估计,从而得到当前盲人的行进方向,将其与盲道走向对比,即可获取实际偏移情况.

|  |
| --- |
| fig: 选取区域靠近使用者前 |
|

## 基于深度图的障碍物感知与避障策略研究

### 基于深度图的障碍物感知研究

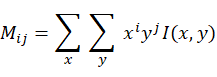
#### 算法步骤

1. 设定阈值
2. 图像获取与深度补全。深度补全的任务是从稀疏的深度图中恢复完整的深度图。【论文引用：[[1802.00036] 捍卫经典图像处理：CPU上的快速深度完成 (arxiv.org)](https://arxiv.org/abs/1802.00036)】【参考：[【论文详解】- Depth Completion - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/41074764#:~:text=论文《In Defense of Classical Image Processing%3A,Fast Depth Completion on the CPU》提出了一种用经典的图像处理算法进行深度补全的算法，不依赖于任何训练数据，没有使用深度学习模型。)】该深度补齐算法算法采用了一种贪婪的策略，有效深度周围的空值可能具有相似的值，所以采用了先填充小hole在填充大hole的顺序，最后减少输出噪声和平滑局部平面。膨胀操作的关键在于kernel形状和大小的设计。size太大，填充的深度值大于实际其作用区域；太小，膨胀不足以使边缘通过后期的闭运算连接起来。



图4-1 深度补全算法步骤

1. 形态学腐蚀膨胀。通过膨胀和腐蚀图片结果，肉眼可见的对于噪点水平得到降低，保证障碍目标的最大连通区域。从直观上来看膨胀操作能够补齐空洞，腐蚀操作可以削弱突起。在主流应用中腐蚀常用来消除斑点类噪声，并保证图像中的目标完整保留。膨胀操作一般用在寻找连通分支时候，连通分支是颜色近似或强度相同的独立块像素点区域。根据以上原理在系统中形态学运算采用先膨胀然后腐蚀的操作，能够消除目标障碍周围噪声和其他噪声影响。
2. 边缘检测(canny算子):首先通过高斯平滑滤波来消除图像噪点。通过求导完成梯度计算之后已基本实现对图像的平滑处理，为细化梯度幅度矩阵后做“非极大值抑制”，搜索图像中所有的备选轮廓边缘点；最后使用双阈值递归嵌套的方式搜索目标的边缘像素点，来完成对障碍目标的边缘轮廓检测。
3. 寻找轮廓列表。在实现边缘检测步骤之后，将得到的结果给到Opencv的轮廓搜索方法，在二值化的图像中搜索障碍物轮廓位置坐标和提取到的轮廓数目。
4. 聚类:获取所有轮廓的最小外接矩形,距离过远或没有检测数据则忽略,在设定阈值下合并矩形框。
5. 质心坐标获取。障碍物多是规则矩形或圆形，计算图像中的质心首先需要计算图像的矩。原始图像矩Mij计算式为：



由图像矩Mij计算公式可得到质心坐标fig:

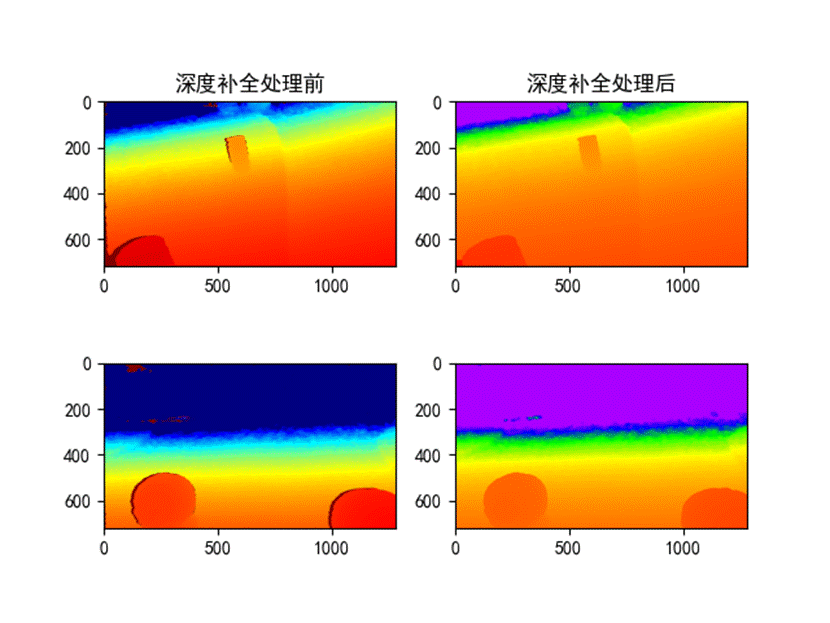


图4-2 深度补全前后效果图

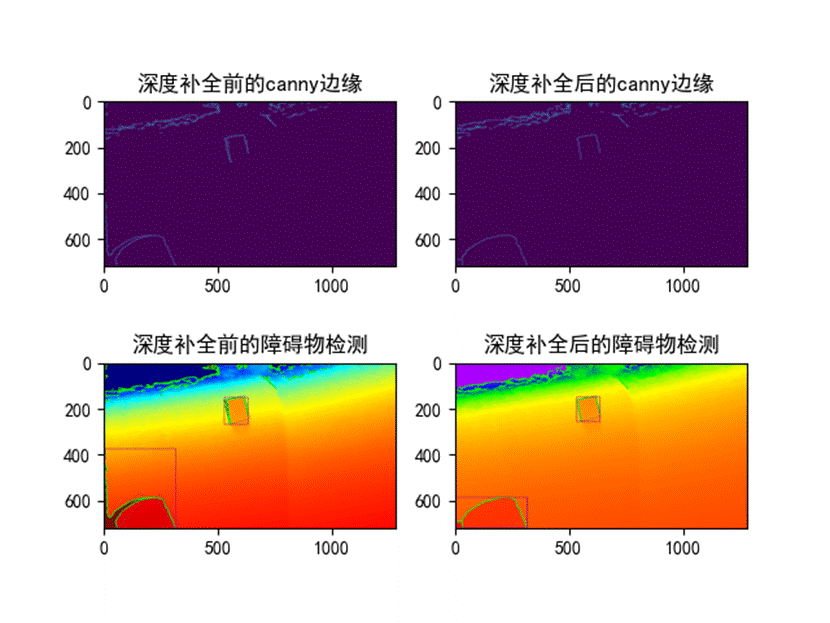


图4-3 深度补全前后障碍物检测效果图

### 基于RRT的避障策略与路径规划研究

1.通过障碍物与盲道人行道信息生成当前局部地图

语义分割盲道和人行道可以得到盲道和人行道在一帧图像中的位置，通过边缘检测可以找出图像中的障碍物的位置，双目深度相机可以返回该处障碍物的深度（距离），通过障碍物距盲人的距离和角度可以计算得到在以盲人为起点的二维图中的二维坐标；将障碍物直线后方的盲道或者人行道作为局部路径规划的终点，生成当前局部地图。

2.避障路线规划

局部路径规划算法使用基于采样的运动规划算法，为了简化问题，避障算法的障碍物取为较为简单规则的正方形（宽度等于深度图中障碍物的宽度）。使用RRT算法进行路径规划分为任务空间内的随机树生长和可行路径的反向搜索两个阶段。RRT算法通过逐步迭代的增量方式进行随机树的构造，首先在任务区域内选定起始节点qinit作为树的根节点，通过从根节点不断的扩展出叶节点的方式构建随机树。首先以概率pg选择目标位置qgoal作为随机目标点qrand，或者以概率1-pg在任务空间内随机选择一个随机目标点qrand：从随机树当前所有的叶节点中，选择出一个离qrand最近的，称之为临近节点qnear；然后从qnearr向qrand的方向延伸一个步长的距离ε，得到一个新的节点qnew，在延伸过程中，判断是否与已知的威胁区域有冲突，若无冲突接受该新节点qnew，并将其添加为随机数的结点；若qnew与威胁区域有冲突，说明该次扩展出的新节点不符合安全要求，则舍弃该新节点，重新进行随机目标点的选取。通过这样不断的延伸扩展，当随机树中的叶节点与目标位置足够近的时候，则认为随机数的构造工作完成，此时以距离目标位置最近的叶节点为起始，依次向上搜索父结点，则可以获得一条从起始位置到目标位置的可行路径。

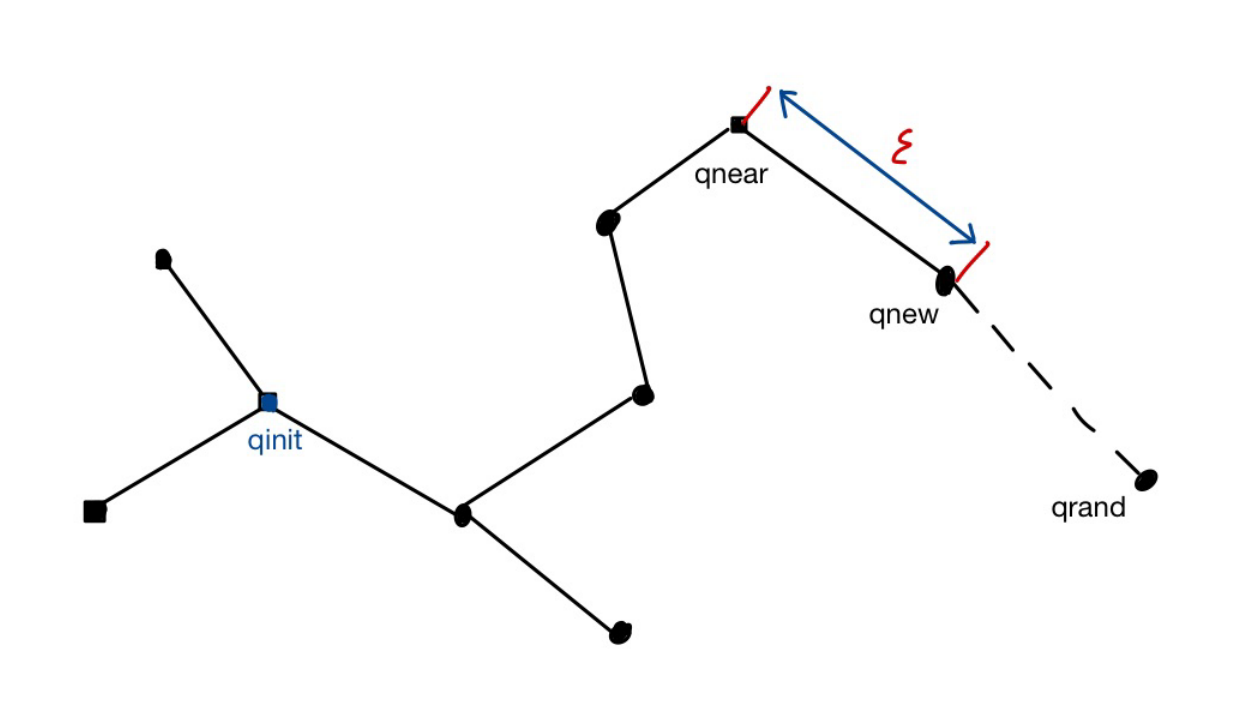


图4-4 RRT的算法节点的扩展

在盲人行进过程中，当障碍物出现在规划的盲道上时，程序语音提示前方有障碍物并开始构建当前路径规划的二维地图；RRT算法从起点（盲人当前位置）开始扩展节点，节点扩展时避开可能有威胁的障碍物区域，在不断的扩展之后，当终点离随机树的叶节点足够近时，此时以距离目标位置最近的叶节点为起始，依次向上搜索父结点，则可以获得一条从起始位置到目标位置的可行路径；

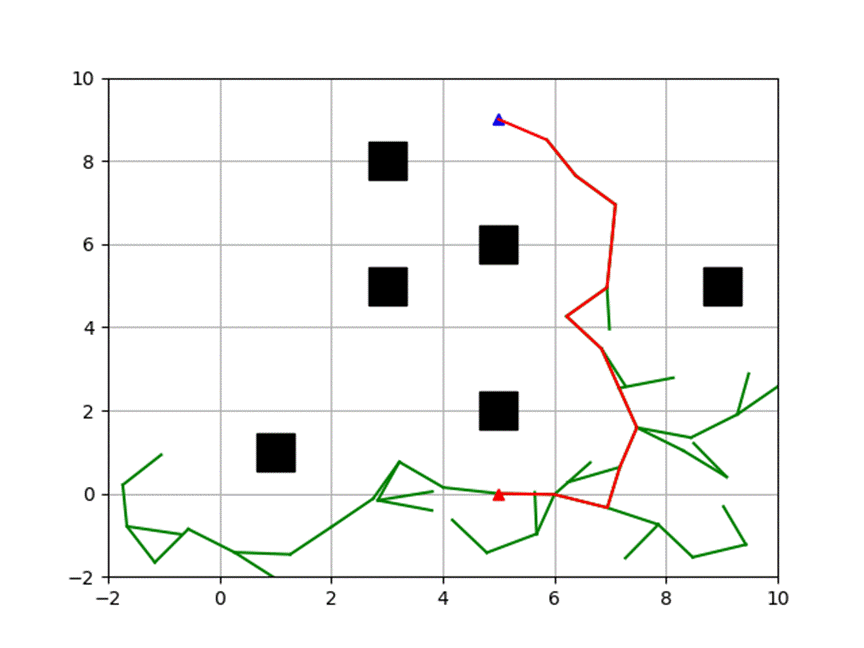


图4-5 仿真地图中RRT 避障路径规划

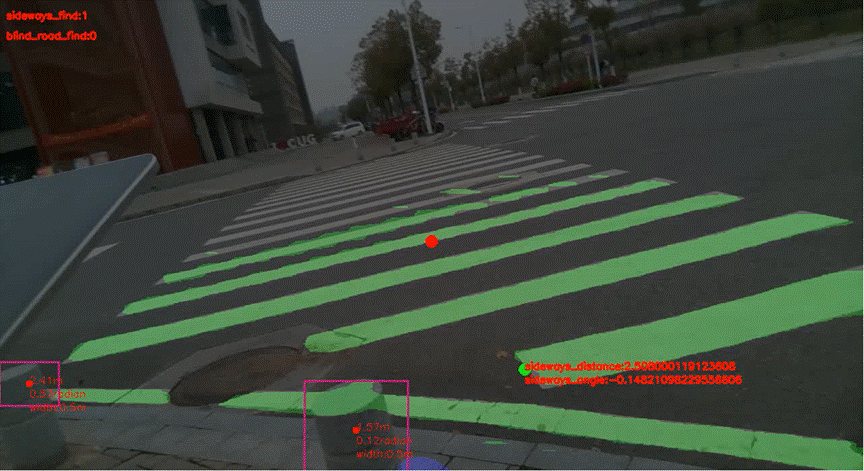


图4-6 识别出来的障碍物信息

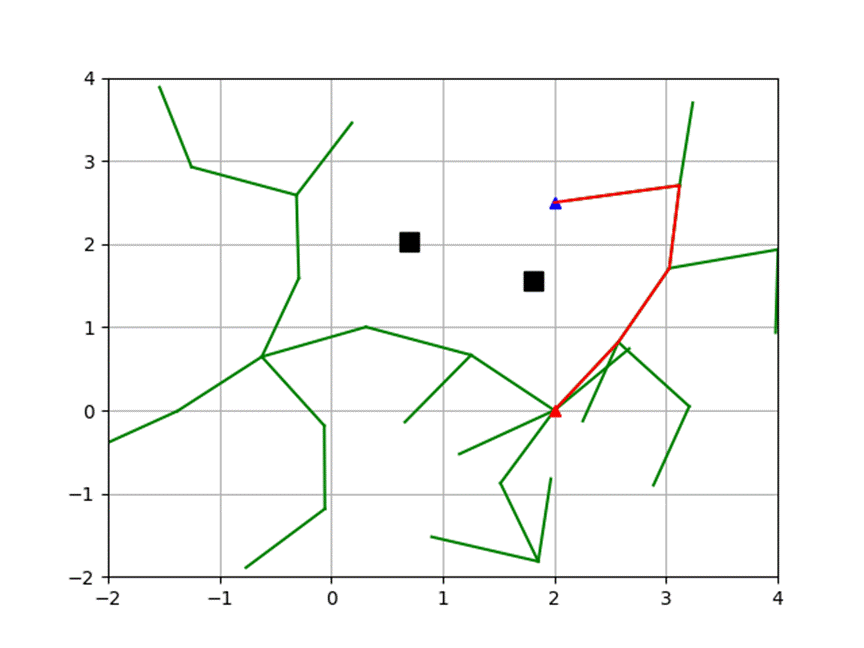


图4-7 仿真地图中RRT 避障路径规划