自动步兵单项赛规划（雷达组）

**一、雷达站建设的整体思路**

1.1 多方位的视野

第一点是最基本功能即为云台手提供多方位的视野，使得云台手得到获知战场的整体信息，同时视野图像也能提供给雷达站程序作进一步的自动分析。该功能技术实现主要依靠机械结构设计，传感器的选择及安装以及如何通过自定义 UI 进行显示。

1.2 为战场敌方车辆定位

第二点即为战场敌方车辆定位，也为雷达站最核心的功能之一。该功能于比赛中除了能给到操作手 UI 上实时的小地图信息以外，更重要的是其为雷达站进一步的自动分析和决策提供了可能，其精度决定了分析和决策的可靠性。该功能实现主要依靠神经网络给出车辆的图像预测框以及对激光雷达点云和图像预测框信息融合来得到车辆相对于相机的位置，此外为了准确得到车辆的世界坐标，相机相对于世界坐标系的位姿估计也是一项十分重要的工作。同时，为了提高程序的鲁棒性，我们还设计了一系列后处理方式来处理误识别的信息。

1.3 雷达自动决策

第三点为雷达自动决策。该项功能决定了雷达站这个兵种的上限，本赛季我们的设计思路主要就其防御功能来考虑，即预警，该功能在赛场上能提供给操作手视野盲区的信息，助其及时反应来避免被袭击。其分为两大块，飞镖预警及车辆预警。飞镖预警我们主要采用了传统方法对发射飞镖头进行检测，车辆预警我们主要根据第二部分得到的位姿及车辆位置信息，采取了图像反投影检测和直接位置检测两种方式来检测进入感兴趣区域的敌方车辆，并通过自定义UI 和车间通信来反馈预警信息。

1. **雷达站的赛程规划（说明：雷达站任务的完成进度需要购买一定的硬件支持）**

2.1 搭建雷达站（时间：目前至明年比赛前）

需要机械组的同学设计图纸，搭建雷达站机械结构

2.2 目标定位实现（时间：目前至寒假）

在高性能GPU电脑上搭建神经网络模型，并不断训练，训练出一个鲁棒性较好的模型

2.3 UI设计（寒假期间）

使用Python的QT库在裁判系统提供的接口上进行设计简单的界面

2.4设计算法（目前至寒假）

设计算法把相机检测的第一视的图片转换成上帝视角的地图

2.5 雷达部署（寒假期间）

使用一个激光雷达，故而雷达传感器类可以基于全局类来设计，但我们同时要对雷达和相机信息进行融合，这就出现了雷达对多个相机要提供接口的情况，故而雷达传感器类设计是雷达点云信息全局处理加反投影深度图对象化处理。由于采用了 Livox 雷达提供的基于 ROS 的 demo 驱动，我们雷达传感器类实际上是订阅Livox 雷达的 ROS 驱动点云节点，通过一个线程不断接收雷达点云信息。由于 livox 雷达是基于非重复式扫描设计，其点云需要一定的积分时间，设计一个队列来接收，该队列存储的实际上是投影到相机平面上点云（在投影相机平面外的点云则去除）的像素平面坐标数组，每个队列对应一张深度图，实际上便为各个投影点在相机坐标系的 z 坐标矩阵，深度图初始全部位置值为 ，代表全部位置无信息。队列长度上限设置为 200，结合 ros 点云发送频率 10Hz，实际上我们深度图上保留了 20s 的点云，由于我们主要需要深度信息，物体运动产生的点云结构畸变对于赛场整体的深度信息影响不大，而更长的积分时间能够使得深度图更加稠密，故而我们保留了较长时间段内的全部点云。

2.6与裁判系统通信（寒假期间）

通过 USB 转 TTL 进行和裁判系统主控模型的串口通信，以读取裁判系统信息和发送信息到裁判系统。小地图通信基于官方协议。当前面算法中得到小地图信息后，对算法得到的小地图位置执行变换，设原来位置为（x,y)，变换为（x，15+y），这是由于世界坐标系原点定义到赛场左上方，而小地图坐标原点定义在赛场左下方。预警信息发送基于车间通信协议，经由 numpy 库将位置转化为 flfloat32 类型，然后发送。由于协议规定每轮发送一辆车信息，故而用全局变量控制车辆发送顺序，从 1 到 5 轮换。其中，算法处理时，对于未预测到位置的车辆 id，会将其位置置 0，发送上则对置 0 的 id 执行不发送处理。