根据本次夏令营的比赛规则，视觉方面主要任务包括引导无人机进行自身定位，以及目标物体的识别。在比赛期间，我们主要实现了两套不同的方案，即全场定位与识别方案，以及利用guidance进行初步定点，然后利用视觉进行辅助精确定位。第一套方案相对比较复杂，设计难度较大，我们花了将近两个星期的时间完成设计及编码，但在最后与ROS等融合时，遇到了一些问题，难以在剩余的时间内完成，因此切换到第二套方案，下面我将主要介绍第一套方案，第二套方案为第一套方案的子集，将稍作介绍。

全场定位与识别方案。考虑到guidance在长期内进行定位存在漂移，以及在低空时数据可能存在较大失真，我们设计了利用场上固定坐标物体进行定位的方案，这些固定物体包括一三区的盒子，二区的九宫格，四区LED屏等，以及遍布场内的二维码，这些固定物体以下统称为观测对象，其中二维码观测对象场地坐标不固定，但可以确认为整型的量级，一三四区的盒子拥有全场唯一的位置信息。

无人机在空中飞行时，系统内会维护一个观测对象集合，集合内的元素即为上述各种类型的观测对象。每个观测对象拥有包括场地坐标，图像中的位置坐标，图像中的坐标系（其中X轴始终平行于一四区方向，且由一区指向四区， Y轴始终平行于一二区方向，且由一区指向二区），观测对象类型，观测对象生命值信息。由于每次起飞的位置坐标固定，摄像头看到的第一个二维码也是固定的，因此可以确定第一个二维码在场地上的坐标，并且由于二维码的长宽信息固定，根据图像中二维码每条边所占据的像素点的数量，可以确定此时地面与无人机图像的尺寸的比例信息，同时，根据电子罗盘返回的yawl角信息，对二维码从中心到四条边中心点的四组向量求取cos数值，可以得到此时图像中X轴和Y轴的方向，基于上述方向与比例信息，即可求解图像中任意一点相对于X轴和Y轴的信息，将其映射到实际空间就可以得到图像中任意一点的场地坐标。此外，所有对象都存在一个寿命数值，每次观测到对象时，都会将该寿命值置成10，每处理完一帧图像，就会对所有观测对象寿命值减1，若观测对象连续10次无法观测到，即生命值减到0，就将其清除出观测对象集合。若能够连续观测到，则该数值一直都大于0，可以保证其一直在观测集合内，此举主要考虑到了无人机在连续飞行时，有些观测对象可能在某几帧图像中无法观测到，但由于飞行器飞行速度相对比较慢，根据先验估计可知此时这个对象应该存在，不应该被清除出观测对象集合，只有连续多次无法观测到时，才认为其已经脱离视野，可以清除。

获得第一个观测对象后，飞行器起飞到一定高度，由于我们使用了一个视场角为100度左右的相机，因此，在这个高度在X方向上可以看到多于两个的二维码。因此当观测到一个新的对象出现时，将其图像坐标输入，利用此时系统内存在的观测对象去估计该对象数值，若为二维码，则将估计出的数值进行量化，即可得到精确的X，Y坐标， 同理若为其他对象，也会对估计出来的数值进行量化，确保其收敛到精确的场地坐标，不会形成累计误差。如1区盒子位置是固定的，当观测到一个对象，其估计出来的X，Y坐标和该盒子实际坐标误差小于一定范围，则可直接将该对象的实际坐标写入，而不使用观测数值。对于二维码而言，若估计到一个二维码坐标与一个整形的坐标误差在一定范围，则可直接写入该整形坐标。

因此，无人机在飞行时，将不断的利用已有的观测对象去估计新的观测对象，并将新的观测对象赋予精确的坐标，误差不会累计。此外，由于一二三四区的的盒子均在观测对象集合内部，而场地中的车无法同时挡住四个二维码，因此，在整个飞行过程中，系统内一直都存在观测对象，利用观测对象就可以得知此时摄像头中心位置所对应的实际坐标，同时该系统前提包括一二三四区域的目标物体识别，因此实际上已经包含了物体识别的流程，因此实现了全场的定位与目标的识别。

以上流程考虑到的是理想状态下，实际飞行中无人机由于需要一些起降，投弹等操作，在飞行高度较低时，系统内所有的观测对象寿命值都会变成0，此时将无法实现定位。为了解决该问题，我们做了以下操作，第一无论飞行器飞至任何区域，系统内的最后一个观测对象都不会被清除，但其生命值依然每次会减1，计算该观测对象最后一次被观测到（即生命值置为10的那一次）到当前生命值共经过了多少帧图像，统计这段时间内guidance给出的位置偏移量，加到最后一次观测到对象时自身的位置估计，即可得到当前状态的位置估计，这种策略主要考虑到guidance在短期内漂移不会太大，当无人机再次观测到观测对象时，可以修正该误差，从而实现无论飞行器飞行高度变化，系统都能无累计偏差的估计自身位置。

此外，在飞行过程中，观测对象内部可能会存在多个对象，如何区选择可靠性最高的那个对象用来对新的对象进行估计。在这里，我选择的策略是首先考虑生命值，生命值越大说明其是最新观测到的，拥有更高的准确度。当存在生命值相同时，利用类型进行排序，由于一三四区域的盒子，LED等全场坐标是唯一的，能够修正任何误差，因此具有较高的优先级，二维码只有相对的数值，优先级较低，二区的九宫格实际坐标并不像其他区域那么准确，因此优先级最低。当类型也是一致时，由于图像中心的对象畸变数值较小，估计相对更为准确，因此应该具备更高的优先级。理想的描述该集合应该时一个最大堆，但由于一般观测对象数量较小，只有四到五个，排序的成本较低，直接使用数组表示该集合，对集合进行排序取第一个数值即可。

上述方案的实现都依赖于一个基本的条件，即二维码的精确识别，以及一二三四区目标物体的识别。识别的流程如下：

1.提取场地中尺寸较大的正方形物体。在这里我们采用了AprilTags的代码，AprilTags的机制是首先计算图像中各个点的梯度数值与方向，对梯度数值和方向均比较接近的点进行聚类，形成一条条小的直线，在对这些直线进行合并，合并的条件主要是直线的方向和截距接近，并且前一段直线末端与后一段直线的前端距离在一定范围内，对搜索到的直线进行深度搜索，寻找能够由四条直线组成首尾相接的四边形。因此，AprilTags考虑到了各种各样的变形情况，搜索的范围很大。但是，实际上我们场地上的观测对象尺寸均比较大，并且均为正方形，每条边之间的夹角基本上接近90度，因此在深度优先搜索的过程中，加入这些限制条件，一方面降低了搜索的规模，同时排出了很多的干扰噪声，确保最后搜索出来的结果是一个略大于观测对象的集合，同时为了提高系统的适应性，各种限制因素是动态变化的，比如限定正方形的尺寸阈值与高度有关，两者存在一个反比的关系，高度越低时，要求该正方形边在图像中的长度越大。

2.对第一步搜索出来的对象进行分类。对于二维码我们将对其进行解码操作，只有解码出来的结果与实际编码数值误差小于一定范围时，才接受该正方形为二维码，实际操作时，每个tags有25个bit，只有错误的位小于2个时，才会接受该对象。

3. 对于被第二步拒绝的物体，进行判定。估计该正方形中心的实际坐标，将该坐标与场地中固定的观测对象实际坐标相比较，寻找到该正方形最有可能的对应物体，若两者实际距离数值在一定范围内，则再计算该正方形与实际物体的特征匹配程度，如对于一区盒子，其内部应该时一片黑色，因为所有娃娃在一开始就会被飞机都抓走。如果特征可以进行匹配，则接受该物体。

4.对2和3得到的对象，求取图像坐标轴。由于预先知道了各个观测对象物理空间对应的大小，以及实际坐标与图像坐标，因此只要获取其图像中X轴和Y轴的方向即可。在这里有两种方案可以选择，一是使用二维码的解码结果可以得知方向，二是使用电子罗盘获取一个粗略的数值，计算与该方向最为接近的一条边，该边即为直线北方向的Y轴，实际上并不要求方向朝北，只要将飞机yawl角减去起飞时朝着一二区方向的角度，即可得到相对于一二区方向的角度，而实际正方形的边必然是平行于一二区方向或者一四区方向，因此，求取四条边与上述的角度的符合成都，就可以拿到X轴，Y轴的方向。基于上述信息就可以得到基于该观测对象的坐标系。

5. 将上述新增的观测对象加入集合。

6. 维护上述观测对象，将生命数值为0的对象清除出观测对象集合，利用可靠性最高的观测对象估计当前飞机的位置。

整个飞行流程将循环执行以上流程。

然而，实际上当我们把上述代码写完了以后，并没有将其使用到实际无人机上。主要限制因素包括两点，一是guidance数据频率为20Hz左右，与图像采集识别频率基本接近一致，并且无法确保guidance给的数据是与当前图片的估计是同步的。因此，可能存在图像中采用的guidance估计数值并不是这段图像时间内的偏移量，从而估计出错误的位置。第二，上述代码占用CPU资源太大，在笔记本上测试时可以比较流畅，但在Manifold上速度明显偏慢，且代码中存在不少的逻辑判断，使用GPU加速后效果并不明显。当图像中存在的特征比较多的时候，识别速度只有10Hz，甚至有可能更低。此外，最后让我们决定放弃该代码还是由于策略的变化，飞行器只执行一段区域内的来回操作，guidance可以确保在短时间内飞行中不会有太大便宜，到了目标区域再使用视觉定位就可以，所以，并不需要进行全场定位。

因此，第二套方案直接使用了第一套方案的目标识别代码，将其封装成接口就完成了，只用了两个小时左右。

全场定位与目标识别的代码见附件，由于版本更迭原因，在一些细节问题处理上可能与上述流程不完全一致。