思路介绍

1.整体框架

整个框架示意图如图1所示，不同视角的视频同步送入模型中（要求帧同步），通过yolov7检测器进行目标检测，然后不同视角的视频分别进行单相机内部的跟踪，为每个目标形成局部id，紧接着进行跨相机的匹配。下面会详细介绍各部分的细节与方法。



图1 跨相机多目标跟踪框架

2.目标检测

目标检测直接采用yolov7检测器即可，不过需要注意的是，三个视频需要先对齐，然后同步送入模型中，此外，在左侧视频中由于弯道的影响导致人员遮挡，因此在图片送入模型前的预处理阶段需要对左侧视频进行掩码处理。

3.单相机多目标跟踪

直接采用bytetrack进行跟踪，没有什么需要更改的，可能仅仅是一些阈值需要更改一下。

4.跨相机匹配

在进行跨相机匹配之前需要注意的是，比赛中包含领骑员，而领骑员是不需要进行跟踪的，因此根据领骑员首先出现在所有人的最前面这一先验来提取其特征，然后后面每一帧需要进行跨相机匹配的时候都需要将单相机的跟踪结果与领骑员的reid特征进行匹配，匹配上了则把领骑员去除。

此外由于右侧相机角度问题，150m线所在位置处于严重遮挡，因此设置了一个范围，在该范围内的轨迹不进行全局匹配，也需要去除掉。

剩余的轨迹在进行跨相机匹配，其具体方案如图2所示，首先单相机多目标跟踪输出了各自的局部轨迹，此时各个局部轨迹互相独立，即不同相机中的同一ID对应的目标可能不是同一个目标。为了统一各个局部轨迹的ID，构建了一个全局轨迹池，该全局轨迹池可以看作是存储了所有全局轨迹信息的模板池，而且会不断更新自己的信息（位置、外观特征），用于对局部轨迹的ID重分配。各个相机所生成的局部轨迹都要与全局轨迹池中的轨迹计算相似度（外观相似度、IOU），根据相似度使用匈牙利算法进行轨迹的匹配，这种情况下会产生三种结果，如图2所示：

1. 成功匹配：成功匹配意味着局部轨迹与全局轨迹池中的轨迹相似度很高，很大概率为同一目标，因此将局部轨迹的ID修改为与之匹配的全局轨迹池中的轨迹的ID，进行轨迹的统一，除此之外，将匹配到的全局轨迹池中的轨迹进行外观特征和位置的更新，在这里需要严格控制匹配，因此需要将匈牙利匹配的阈值调小一点；
2. 全局轨迹池未满且存在未匹配的局部轨迹：没有与全局轨迹池中的轨迹成功匹配的局部轨迹被认为是没有全局化的轨迹，也就是全局轨迹池中并未出现该目标的全局轨迹，更进一步来说就是该目标在相机中第一次出现，由于在单个相机中新出现的轨迹不具备全局信息，因此需要对其ID进行修改成为全局下的ID，即将其ID修改成为全局轨迹池中包含的所有轨迹中最大的ID，并将其加入全局轨迹池，新加入全局轨迹池的轨迹需要初始化其全局下的位置信息。
3. 全局轨迹池已满且存在未匹配的局部轨迹：全局轨迹池已满说明所有的目标的全局轨迹都以创建，没有匹配上可能是由于切换了相机导致第一次根据IOU计算相似度使得相似度很低，因此再次只根据外观特征进行计算相似性，然后把匈牙利匹配阈值调到很大，进行强制匹配。

其中对于全局轨迹的外观特征更新，采用指数移动平均的方式，将各个相机中的局部轨迹更新至全局轨迹。对于位置的更新则分别更新全局轨迹在不同相机中的位置，即一个全局轨迹包含有在不同相机中的位置，当与局部轨迹成功匹配时，将该全局轨迹的位置（在匹配到的局部轨迹出现的相机中的位置）更新为该局部轨迹当前的位置。对于位置的初始化则直接把该全局轨迹在各个相机中的位置初始化为其局部轨迹出现在的相机中的当前位置，如若存在在某个相机中未曾出现该目标，则把在该相机下的位置初始化为0。

图2 跨相机匹配方法

5.输出信息计算规则



准备工作：1.将0m，50m，150m线手动记录；2.判断点在线左方还是右方；3.保存前一帧的位置；4.判断点是否跨越了线；5.在三相机中，目标框的点都是取的左下角点。

1.视频帧：记录当前为第几帧，利用视频帧率计算画面时间；

2.圈数：如果目标在第1个相机中，且目标在相邻两帧中跨越了线，圈数加1

3.关键点：根据目标出现在的相机和目标与线的位置判断目标在哪个区间段

4.排名：先根据圈数和关键点来进行粗略排名，如果有在同一圈同个区间段的目标，则根据点的位置来进行排名（受相机角度、弯道、检测效果较大影响）；

5.比赛计时：记录圈数中首次出现1的帧数，然后利用视频帧率计算比赛时间，直到圈数信息中全部为比赛的最大圈数+1

6.冲刺时间：当目标只剩下最后三圈时，记录开始冲刺的帧数，计算冲刺时间，直到圈数信息中全部为比赛的最大圈数+1

7.区间速度：当目标跨越某条线时，记录下当前帧，并利用在前一条线记录下的帧数，计算区间速度

注意：由于右侧相机150m线处处于遮挡区域未进行跨相机匹配形成全局轨迹，因此在用到150m线时，实际是根据视频上方虚拟构造出来的一条线来进行判断的。具体来说将实际的150m旋转得到一条虚拟的线，把所有框过真实线的时间记录下来（因为没有轨迹信息，只有当前帧的位置，所有判断过线是设置了一个距离，小于该距离的都视为过线，所以一个框在真实线这里会记录连续的几帧），然后将记录的帧从小到大排序并计算相邻的差值，选出五个最大额差值，把记录的帧分成与人数相同的分数，然后取每一份的平均值作为过线的帧，这样处理可以保证速度范围不会太离谱，但是可能精确性不是特别高，会有几帧的差距，然后在过虚拟线的时候将所有的时间换算成在实际线的时间。

6.可能会修改的部分

(1)一些阈值，包括检测的nms中的阈值，还有单相机跟踪的阈值，当然还有跨相机匹配部分的一些阈值，不仅是一些阈值，可能还会有一些定死的参数，比如轨迹长度、离视频四周的距离这些，一般不需要更改，只要阈值部分可能会改的多一点，这些主要影响着能不能检测跟踪到以及获取的reid特征是不是可靠的。

(2)输出信息部分，尤其是根据虚拟的150m线来获取的一些参数，这些参数可能精度不是很准，需要提高精度，还有就是如果存在未跟踪到的情况如何进行保证会有参数输出。

7.近期更改记录

（1）iou阈值调高了一点确保没有漏检

（2）\_convert\_frame150\_time函数，将过虚拟线时的时间转化成过真实线的时间。

（3）\_record\_frameon150和\_get\_value获取在真实的150m线处的帧，以及在过虚拟线时获取对应目标的过实际150m线的帧

（4）\_count\_turns 正常在中间视频而且过了0m线及判断圈数+1，但是0m线处有遮挡时，圈数没有+1，导致后续排名都是错的，因此添加了如果目标出现在左侧视频中时圈数与前一个id的人的圈数不一致，即将其圈数+1。