视频组----事故判断部分

# 一、前提

只允许四种情况的ID消失：1、碰撞；2、遮挡；3、驶离画面（左右边界）；4、驶远（远方）

视频大小:W\*H=1920\*1080

# 二、判断是否出现id消失

　该ID消失是在卡尔曼滤波以后的消失,也就是说,当视频中出现某一帧后面该ID消失,输出的txt中该ID仍存在并有一个预测位置,若X帧以后该ID仍不出现,输出文档中该ID才消失。X帧的值由跟踪算法确定。

## 2.1 有id消失

1. 返回消失对象的ID、类别和对应视频帧数，并记录消失对象的位置和体积（x,y,s,h）
2. 获得消失前一帧的消失对象类别和体积，判断是否小于对应体积阈值；消失前一帧消失对象的位置是否在图片边缘且行驶方向为远离图片行驶
3. 若出现驶离画面和驶远，无事故。未出现，进行4）
4. 寻找出在消失前一帧以ID中心点（x，y）为中心 r 距离区域内的所有ID，对每个ID寻找其当量直线，计算每个当量直线和消失ID的当量直线之间的最小距离,记录最小距离的对象carshone的信息;
5. 比较最小距离和设定的阈值，若小于阈值，疑似碰撞;若大于阈值，无事故;

PS：a、可以判断出是否驶离画面和驶远；

b、体积阈值包括后续阈值均由经验确定；

c、当量直线为中心点和速度方向的直线与消失对象框的交点AB

## 2.2 无id消失

寻找各物体是否有速度突变（用加速度判断）。若有，将加速度突变值与加速度阈值比较。若大于，认为发生出事故; 若小于，无事故;

# 三、疑似碰撞确认

# **输入:like\_accidentboxes**

# **输出:miss\_to\_accidentboxes**

　上述仅为确认发生疑似碰撞，之后需进一步确认。对于疑似碰撞的对象，在后11帧中做检测，检查是否出现，若出现，出现坐标和原坐标是否有较大的变动;

　　若变动超过疑似对象的0.7\*wideth,noacci\_count++;因为like\_accidnet数组不清空,所以每一帧出现的疑似对象会一直累计在里面;若noacci\_count>=1,认为正常

　　若不超过阈值,like\_check++;若11帧检测结束以后,其位置变动仍然小于0.7\*wideth,则认为可能出事故;(防止速度慢的对象的遮挡情况被当作事故)

若10帧都不出现,则认为可能出事故。

**四.速度滤波**

**输入:miss\_to\_accidentboxes**

**输出:vfilter\_fmiss , vfilter\_for\_stay**

**1.速度消畸变:** 对可能出事故的对象的速度进行矫正(畸变存在的缘故);

**2.速度滤波:**

**vfilter\_fmiss--------满足(1)(2)或(1)(3),出事故**

(1) 消失物体miss\_box或相撞对象crashone的速度大于T;

(2) 两者速度均大于L的

(3)miss为y向移动,crashone大于L,miss的vy\_filter大于vy\_T

**vfilter\_for\_stay--------满足(1)(2)或(1)(3)**

(1)miss和crashone的速度小于T ,但均大于t ;

(2)两者速度均大于l

(3)miss为y向移动,crashone大于l ,miss的vy\_filter大于vy\_T

where:

T----------上限阈值 vfilter\_thre

L----------下限阈值 vfilter\_lowthre

vy\_T------vy消畸变后的上限阈值 vy\_filter\_thre

t ----------次上限阈值 vsumfilter\_forstay\_thre

l –---------次下限阈值 vfstay\_lowstay

**五.静止检测**

当横向移动物体未达到300时,因为纵向物体vum\_filter小,因此会导致这样的事故会被速度滤波滤掉.

使用条件:速度滤波过后的事故数目为0

**输入:vfilter\_fmiss , vfilter\_for\_stay**

**输出:stay\_to\_accidentboxes,出事故**

1. crashone,满足以下条件:

(1)在确定为vfilter\_for\_stay后F内

(2)出现并且速度小于VS

(3)与前一帧的坐标之差小于DS

where:

F-------检查帧数范围 stay\_checkthre

VS-----认为静止的速度阈值 vsum\_staythre

DS-----认为静止的相邻帧坐标差值 slide\_staythre

2. miss的筛选同上

**六. 事故筛选**

通过速度滤波(vfilter\_fmiss)和静止检测(stay\_to\_accident)两个分支找出的事故数目并不能作为最终的事故输出.因为由于事故的特殊性,往往一个事故只会满足其中一种判定,当然也可能都不满足.但是,这两种方法找出的事故是可以作为互补的.

但是,有时候速度滤波找出1个对象,但是静止检测可以找出好多个对象(10+),这样的情况下,假如两者合并输出,错误率太高.因此,设立事故筛选机制,使事故输出数目尽量少且精准.

**机制:**

1. 倘若count1和count2一者为0,直接全部输出另一者.

2. 倘若count1+count2<=6,全部输出;否则,只输出速度滤波的结果.

Where:

count1------速度滤波后的事故数目 vfilterfimss\_count

count2------静止检测后的事故数目 stayaccident\_count

achievement:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 视频 | 速度滤波事故数目 | 静止检测事故数目 | 输出事故数目 | 能否检测出事故 |
| 04 | 2(1+1’) | 1(0+1) | 3 | Y |
| 05 | 0 | 1(1+0) | 1 | Y |
| 10 | 0 | 3(0+3) | 3 | N |
| 13 | 0 | 2(1+1’) | 2 | Y |
| 23 | 2(1+1’) | 0 | 2 | Y |
| 24 | 1(0+1) | 2(1+1’) | 3 | Y |
| 1min\_num1 | 1(1+0) | 11(0+11) | 1 | Y |
| 005\_1 | 3(1+1’+1’’) | 1(0+1) | 4 | Y |
| 005\_2 | 0 | 1(1+0) | 1 | Y |
| 11 | 1(1+0) | 0 | 1 | Y |
| 13(truck-truck) | 1(1+0) | 0 | 1 | Y |
| carbicycle | 1(1+0) | 4(1+1’+1’’+1’’’) | 5 | Y |
| carcar1 | 2(1+1) | 0 | 2 | Y |
| carcar2 | 1(1+0) | 0 | 1 | Y |
| Heavytruck-car | 1(0+1) | 0 | 1 | N |

Ps: 3(1+1’+1’’) , 3 为事故总数, 1为事故, 1’和1’’均为同一事故,这是由于撞击后人,车轮什么的会甩出来,认为是事故对象

**七. 当前问题**

1. **需要新的想法,再设立新规则,**提高识别率和正确率.

2. 因为目前的视频情况都是撞击后有id消失的情况, 所以尚未对未消失id的情况设立规则