**公安视频图像信息应用平台车辆研判功能分析**



1. 总体业务流程
2. 优先完成
   1. 套牌分析
      1. 功能描述

根据系统中保存的过车记录，分析出有套牌和嫌疑的过车记录。

* + 1. 功能分析

套牌**分析的流程**是**先查询**需要分析的车牌，**再分析**是否存在套牌现象。查询模式有两种：普通查询和高级查询。普通查询先输入车牌号，再分析该车牌，目的是对一个号码进行分析。



高级查询事先限定一些条件，然后再搜索满足条件的车牌，目的是对满足条件的一些号码进行分析。条件有分析时段、车辆类型、颜色、结果状态（是否被处理过？）和车牌号码。



据此推测内部实现应该分为两部分：查询和分析。查询有直接利用车牌作为关键字索引（select \* from database where plates=”xxxxxx”）和给出限定条件间接索引(select \* from database where type=”xxx” and color=”xxxx” )；分析流程是根据查询操作找到的车牌，再在数据库中找出与这些车牌有关的过车记录，通过过车记录来判这些车是否存在套牌现象（将每次记录作为点，按照时间次序依次连接起来作为边，我们把点的直线距离和时间的商作为边的权重，若存在一条边的权重大于阈值，则判断为套牌车）

* + 1. 流程实现
       1. 搜索一个车牌

1. 直接根据车牌从数据库中查询出来该车的全部记录(select \* from database where plates=”xxx” order by time)
2. 根据记录的时间前后为基础，从早到晚排序，将记录经纬度作为点，顺序连接起来作为边，将权重定义为“两点之间的直线距离/两点时间” 若存在某一边的权重大于阈值，则断定此车存在套牌现象。
   * + 1. 根据条件搜索多个车牌

在搜索一个车牌的基础上，再加入一个步骤。根据限定条件从数据库中查询出车牌(select plates from database where type=”xxx” and color=”xxx” and (time between tc1 and tc2))，并存入数组中，然后循环执行搜索一个车牌的算法。

最后，将有套牌现象的车的所有信息输出到数据库(insert into result\_db (plates,plates\_color,plate\_img,location,plates\_categary,car\_categary,car\_color,pass\_time,final\_state) values (...))和屏幕。

* + 1. 流程图



1. 套牌车检测算法

a)研判一辆车 b)研判多辆车

* 1. 关联分析
     1. 功能描述

分析某车辆的关联车辆信息

* + 1. 功能分析

该功能要求输入车牌号码、分析时段、分析区域、跟车时间、跟车次数、车牌归属和过滤车牌；输出分析任务列表。



分析该模块实现为：查询区域内所有摄像头在分析时段内的记录（

select cameras,time from database where plates=”target” ，

select \* from database where camera=cameras and time between tc1 and tc2）

tc1是目标车在第c个摄像头处的时间并回溯10s,t2后推10s，然后根据跟车时间，跟车次数，车牌归属和过滤车牌这些条件来筛选满足条件的车辆。从输出为任务列表这一点来判断，估计分析过程会比较慢，可以开多线程来分析多个任务。

//Apriori关联分析算法

* + 1. 流程实现

1. 查询区域内所有的摄像头在分析时段内的记录,找出目标车牌所有记录(select \* from database where plates=”xxx” and time between t1 and t2)
2. 根据记录中所查询到的目标车辆出现的位置和时间，搜索除该车外所有车在时间前后的记录(select \* from database where location in plates\_location and time between tc1 and tc2)(tc1 tc2是目标车辆在plates\_location位置的时间往前后推30s)并以车牌记录位置为索引，放入map数据结构中。
3. 遍历map，并找到出现次数最多的几辆车。
4. 将找的车辆信息输出到数据库以及屏幕上，表头如“套牌分析”
   * 1. 流程图



1. 关联分析
   1. 轨迹碰撞
      1. 功能描述

对嫌疑车辆的过车记录进行分析与提取，并通过地图分析出该嫌疑车的行车轨迹。

* + 1. 功能分析

**输入：车牌号码 结果条数 碰撞区域；其中碰撞区域有区域（摄像头）和分析日期，并且可以添加项目**



分析其实现应该为通过车牌号码为索引，搜索区域内所有摄像头在分析日期内的记录，并对记录基于日期从早到晚进行排序，所得的结果就是路径，因为要通过地图进行分析的缘故，故要结合GIS通过最短路径算法实现。因为可以新增碰撞区域，故有互斥，相交和包含三种状况。则可以先通过集合运算计算出摄像头，再作为一个新的区域分析；或是直接假设各区域互斥，开启多线程进行查找。

* + 1. 流程实现

1. 通过车牌号码为索引，搜索区域内所有摄像头在分析日期内的记录，并对记录基于日期从早到晚进行排序(select \* from database where plates=”xxx” and cameras in (range) order by time)
2. 搜索数据库得到的记录中的时间和位置就是数据库的输出结果
3. 根据最短路径算法和GIS，计算出实际可能的路径。
4. 输出到屏幕
   * 1. 流程图



1. 关联分析
   1. 首次进城
      1. 功能描述

用于查询某地区或若干卡口点在某时间段内首次出现哪些车辆。

* + 1. 功能分析

输入：分析时段 回溯天数 车辆归属地

输出：卡口名称 车牌号码 车牌颜色 车身颜色 车辆类型 通过卡口的时间 布控？

通过查询这些卡口的摄像头和地区边缘摄像头在（分析时段-回溯天数）的记录，来确定是否有该记录

* + 1. 流程实现

1. 查询这些卡口在分析时段t时回溯天数内r的记录(select \* from database where location in (xx,xx,xx,...) and time between t-r and t )
2. 按照车牌为索引统计并检查这些记录。
3. 若记录仅有一次，则为首次出现，并将该车的所有信息输出到屏幕
   * 1. 流程图



1. 首次入城
   1. 昼伏夜出
      1. 功能描述

用于查询在某地区后若干卡口点经常在夜晚出没，而白天极少出没的车辆

* + 1. 功能分析

输入：和高危时段输入基本相同，但要求输入两个时间段 以及在该时间段内出现的次数

输出：卡口名称 车牌号码 白天通过次数 晚上通过次数 布控？

应该存在多个拍摄地点，查询这些摄像头在分析日期的时间段内的所有捕捉记录(Select \* from database where camera in (range) order by plates)，再以车牌号为索引，通过次数为限制条件，逐个检查车辆是否昼伏夜出。

* + 1. 流程实现

1. 查询抓拍地点范围内的摄像头在分析日期内拍到的数据(Select \* from database where camera in (range) and time between t1 and t2 order by plates)
2. 根据白天出现次数 晚上出现次数和白天时间 晚上时间作为限制条件，计算出白天出现次数和晚上出现次数的比例，由这个比例的大小来判断出是否昼伏夜出。
3. 将有嫌疑的车辆信息输出到数据库和屏幕
   * 1. 流程图



1. 昼伏夜出
   1. 频繁夜出
      1. 功能描述

用于查询某段时期内在某地区或若干卡口点频繁夜间出行的车辆

* + 1. 功能分析

同昼伏夜出

* + 1. 流程实现

1. 查询部分同昼伏夜出
2. 利用夜出时间和通过次数来判断车辆是否有嫌疑
3. 将嫌疑车辆输出到屏幕
   * 1. 流程图



1. 频繁夜出
   1. 隐匿车辆
      1. 功能描述

用于查询案发前后出现次数发生显著差异的车辆

* + 1. 功能分析



和频度分析雷同，检查两次，然后比对前后次数

* + 1. 流程实现

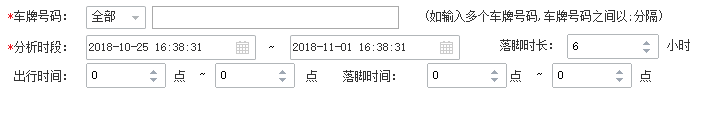
1. 查询部分同昼伏夜出
2. 统计案发前出现的次数和案发后出现的次数，并得到他们的比值，根据比值来判断是否为隐匿车辆
3. 将最后的结果输出到屏幕
   * 1. 流程图



1. 隐匿车辆
   1. 落脚点分析
      1. 功能描述

用于分析车辆可能的落脚点

* + 1. 功能分析



没有要求输入区域或地点，以车牌号码为索引对整个数据库进行搜索，得到在分析时段范围内的记录(select \* from database where plates in (ranges) and time between s and e)；将出行时间、落脚时间和落脚时长作为限制条件，来筛选满足条件的车辆。

* + 1. 流程实现

1. 知道车牌号码，分析时段，对范围内的摄像头拍到的数据进行搜索(select \* from database where plates in (p) and location in (ranges) time between s and e)
2. 将出行时间、落脚时间作为限制条件，落脚时长作为阈值，来筛选满足条件的车辆。
3. 输出满足条件的车辆的所有信息。
   * 1. 流程图



1. 落脚点分析
2. 次要目标
   1. 多进不出
      1. 功能描述

用于查询某段时间内有哪些车辆多次进城不出城，可用于查询不同车牌号进出城的的车辆

* + 1. 功能分析

输入：分析时段 进城次数 出城次数

输出： 车牌号码 车牌颜色 进城次数 出城次数 详情 布控

和首次入城一样，都是查询区域边缘的摄像头记录，需要数据库数据包含进出的标志（进出口摄像头）

* + 1. 流程实现

1. 查询边缘摄像头在分析时段的所有记录(select plates from database where time between t1 and t2)，并根据车牌作为索引来建立map数据结构
2. 根据摄像机的位置的不同来判断是出城还是入城，并将相应的结果统计到map中
3. 根据每个元素的出城入城比例来判断是否是嫌疑车辆。
4. 找到车牌号并从数据库中搜索所有的信息(select \* from database where plates in (plates we search))
5. 输出所有的数据
   1. 异常行为
      1. 功能描述

用于查询在某段时间内，按特定轨迹行驶的车辆

* + 1. 功能分析

输入：除了标准输入外，还有正常轨迹的添加（应该是摄像头的位置）





从功能描述上来看，应该是监控特定的车辆。

比对路径即可

* + 1. 流程实现

1. 取出起点所拍到的所有车牌(select carplates from database where location=”start” time between t1 and t2)
2. 检查我们添加的所有卡口监控记录(select \* from database where plates in (carplates) location in (range) and time between t1 and t2 order by plates)并以车牌为索引，建立map数据结构
3. 对每个map元素以时间为次序来排序
4. 依次检查每一个车辆的location是否与我们规定的途径点（location）相符。若不相符（没有该卡口的记录），则根据上一个有记录的卡口的时间和下一个记录的时间之差，计算出异常时间；若这个异常时间超过了我们规定的时间，则将该路径（两个卡口的位置）记录为异常路径（记录在另一个以车牌为索引的map数据结构中）。
5. 输出结果
   1. 单日过城
      1. 功能描述

**用于查询某日直接通过城市的车辆**

* + 1. 功能分析

又普通查询和高级查询之分，普通查询就是通过车牌号直接索引城市卡口数据库。高级查询就是套牌分析的高级查询



* + 1. 流程实现
       1. 搜索一个车牌

1. 直接根据车牌从数据库中查询出来该车出城入城的全部记录(select \* from database where plates=”xxx” and location in (edge of city) and time between t1 and t2 order by time)
2. 通过统计车辆在单日内在出入口的记录数量，来确定车辆是否单日过城。
   * + 1. 根据条件搜索多个车牌

在搜索一个车牌的基础上，再加入一个步骤。根据限定条件从数据库中查询出车牌(select plates from database where location in (edge of city) and type=”xxx” and color=”xxx” and (time between tc1 and tc2))，并存入数组中，然后循环执行搜索一个车牌的算法。

最后，将有套牌现象的车的所有信息输出到数据库(insert into result\_db (plates,plates\_color,plate\_img,location,plates\_categary,car\_categary,car\_color,pass\_time,final\_state) values (...))和屏幕。

* 1. 频度分析
     1. 功能描述

分析出在特定时间段内，通过特定地点一定次数范围内的车辆。

* + 1. 功能分析

频度分析要求输入： 分析时段 通过次数 分析区域（应该是摄像头组成的环）实现的功能应该是分析出在限制时间内，出入特定区域一定次数的满足条件的车辆。

输出： 车牌号码 通过地点 通过次数 详情 布控？

分析该功能通过调取范围内边缘摄像头(边缘检测)在分析时段所拍下所有车的记录（以车的车牌号码作为索引，范围是该区域所有的摄像头在分析时段所拍下的记录），并以拍摄时间为基准进行排序，并记录有几对结果来实现判断。

* + 1. 流程实现

1. 进行边缘检测，基本原理（第一象限）：存在一个点a,使得点集中其余任意点的横纵坐标均小于这个点。以此类推可以得到其余象限的情况。所以，我们可以规定一个中心点（经纬度），并根据这个中心点，以范围作为半径，构建坐标系，并计算出摄像机在这个坐标系中的相对位置，然后就可以利用这个算法来求出边缘摄像机。
2. 查询这些摄像机头记录(select \* from database where location in (edge) and time between t1 and t2 order by time)
3. 根据摄像头记录的时间先后顺序排序，并除以2，计算出出入次数。
4. 搜索范围内其余所有摄像头(select \* from database where location not in (edge) and location in (range) and time between t1 and t2 order by time)
5. 比较该搜索结果第一项的时间ts和先前查询这些摄像头记录的初始时间t，若ts<t，则表示第一次边缘拍到的是驶离该区域；否则，为驶入该区域。
6. 根据摄像头记录的时间先后顺序排序，对2求余，结合汽车的初始状况判断是否还在范围内。
   1. 高危时段
      1. 功能描述

用于查询某高危时段内在某地区后若干卡口点出没的车辆

* + 1. 功能分析

输入：分析时段 时间段 抓拍地点

输出：同上

应该存在多个拍摄地点，查询这些摄像头在分析日期的时间段内的所有捕捉记录

* + 1. 流程实现

查询抓拍地点，在分析时段在时间段的记录(select \* from database where location in (range) and time in (section))