基于二维码技术和无人机拍摄的露天停车场快速部署智能停车系统

研究报告

胡峰源

中国人民大学附属中学

**2018.10.10**

**摘要**

针对如何充分利用有限的停车资源来最大限度满足车辆的停泊需求这一难问题，本研究报告首次提出基于二维码技术和无人机拍摄技术的联用来设计可以快速部署露天停车场的智能停车系统，并初步验证了其实现过程。新设计的智能停车系统结构上由车位二维码模块、无人机拍摄模块、数据处理模块和信息反馈模块四部分组成，采用无人机和二维码技术实现智能停车，并采用了自创的车位智能推荐系统，实现自动找车位、推荐车位、停车后标记车位防忘等功能。相较于传统停车场更高效，与其他智能找车位技术相比具有成本低、易维护以及功能集成、对用户友好的特点，具有一定的实用价值。

研究报告首先介绍了系统的组成结构。系统的二维码模块自制，无人机拍摄模块采用大疆Spark(晓)型无人机，数据处理模块与信息反馈模块均使用python编程，实现了对无人机拍摄的影像数据的处理、无人机调动的管理、车位的智能推荐以及图形化的人机交互系统。

然后研究报告介绍了系统的四个子系统及其功能的实现过程。

①无人机工作子系统。它实现了三个功能：空中自动巡航、对地拍摄和多台无人机全自动交替使用。该系统保障了无人机拍摄模块的正常工作，同时实现了全自动循环运行。

②系统识别二维码子系统。它将无人机拍摄的影像数据进行处理，提取其中的二维码信息。

③系统更新车位状态子系统。它根据识别出的二维码信息实时更新车位状态。

④人机互动子系统。它包括两个功能：车位显示和车位智能推荐。该系统读取车位状态信息并通过图形交互系统显示车位实时状态，并根据算法将车位智能推荐给用户。

本文最后部分的工作是编写相关小程序，对系统功能进行了模型化模拟，达到了预期目标。

目 录

1. **引言………………………………………………………………………………1**

1.1研究背景……………………………………………………………………1

1.2研究目的与意义……………………………………………………………1

1.2.1研究目的………………………………………………………………1

1.2.2研究意义………………………………………………………………2

1. **基于二维码技术和无人机拍摄的露天停车场智能停车系统结构组成……2**

2.1设计理念……………………………………………………………………2

2.2车位二维码模块……………………………………………………………4

2.3无人机拍摄模块……………………………………………………………4

2.4数据处理模块………………………………………………………………5

2.5信息反馈模块………………………………………………………………6

1. **样板软件结构……………………………………………………………………7**

3.1无人机的控制与调动………………………………………………………7

3.2二维码的抓取与识别………………………………………………………8

3.2.1 二维码的识别与信息提取……………………………………………9

3.2.2 车位状态的存储………………………………………………………9

3.2.3 长时间未被扫描到的空闲车位状态的更新………………………10

3.3系统状态的更新……………………………………………………………10

3.4图形交互系统设计…………………………………………………………11

1. **总结与展望……………………………………………………………………11**
2. **参考文献………………………………………………………………………11**
3. **附件……………………………………………………………………………13**
4. 引言

**1.1研究背景**

我国现在停车场车位数量远远低于停车需求，且集成化、智能化程度不高、仅仅停留在对有效车位的信息统计与发布上，并未重视如何让车主有效率地找到一个停车位。虽然有一些可以识别车位的技术，但这些技术都不适用于快速部署，所以临时停车场车位管理混乱，找车位效率低下等问题十分突出。同时停车过程中存在车位竞争、空游等现象，也这使得停车场管理效率低下，耗费大量人力物力。如何充分利用有限的停车资源来最大限度满足车辆的停泊需求成为了当前急需解决的问题。基于无人机视觉识别二维码的智能停车场管理系统具有可快速部署性、高效性、便捷性、可靠性和安全性等优点，可以很好解决我国停车难的问题，具有非常大的理论意义和实用价值。

目前关于车位识别的成熟技术主要有微波、超声波、红外、地感及图像识别等，其中红外和超声波识别易受天气影响，微波发射功率大、成本高，图像识别需要架设较高的摄像机立柱、成本高。地感线圈安装复杂，且维修困难，易损坏。同时以上技术都不具备可以快速部署的能力，建设周期长，耗资大。所以廉价、对天气适应性强且可以快速部署的车位识别技术具有很大的研究价值。

无人机是一种有动力、可控制、可执行多种任务的无人驾驶航空器。其体积小、造价低、使用方便、适应性强，且具有优越的性能，在遥感航拍、抗震救灾、农林业、环境监测、通信中继和快递服务等许多方面得到了广泛的应用。无人机使用成本低，不存在人员疲劳和安全问题。同时随着智能无人机技术的不断提高，无人机的安全性和自主控制性将会得到较大的改善，运营成本和风险将会进一步降低，运营效率也将得到极大的提高。

**1.2研究目的与意义**

**1.2.1研究目的**

目前很多停车场对如何让车主有效率地找到车位的问题重视不够，且无法准确把握车位状态，导致车主进入停车场后无法快速找到车位，损失了车主的时间。特别在一些规划临时停车场的大型活动中，由于临时停车场功能较为单一，不智能化，导致车位管理混乱，停车效率低下等问题，对活动产生了负面影响。而且在汽车寻找车位时，由于低速运转的汽车发动机对燃料的利用效率不高，污染比普通转速时更严重，从而造成了大量污染。为解决目前停车场管理混乱，停车效率低下等问题，本报告针对露天停车场设计了一套结合车位寻找、车位智能推荐及车位防忘等功能于一体，方便快速部署的智能停车系统，它具有如下特点：

1. 所设计的车位二维码面积大、内容简单，易于无人机进行识别，且二维码识别技术及印刷技术非常成熟，价格低廉，适合普及。小二维码的设计方便了车主停完车后记住车位以及结束定位，使系统操作更加简便可靠。
2. 无人机系统较普通二维码找车位技术成本低，普通二维码找车位技术需要安装大量高清摄像头以拍摄地面二维码，此技术成本高，需要铺设大量电缆，维护较难，同时存在监控死角。而无人机在天空中巡航拍摄很好地解决了这些问题，降低了成本和维护难易程度，减少了监控死角。无人机自动飞行技术也较为成熟，可以实现无人操作自动巡航，自动避障，降低人力成本。
3. 信息反馈系统在将空余车位信息反映给车主的同时，根据其他车主的位置信息推荐最佳车位，有效避免了多辆车争同一个车位的事件，节省车主时间，且空车位信息及其他车辆的位置信息也有助于车主自行判断行车路线。出停车场的车主由于在停车后扫过车位右上角的小二维码，其位置停止上传，所以不会出现在小程序中与正在寻找车位的车辆混淆，不会对车主判断造成干扰。
4. 本系统结构简单，可以快速部署。其中车位二维为印刷车位二维码的大型贴纸等价格低廉，生产快，改造停车场周期短，方便清除的一次性物品。该智能停车系统具有搭建快速的特点，也具有帮助引导参加活动的车辆的功能，虽然使用寿命短，但方便，成本低，极大方便了人流量较大的活动的车位管理。

**1.2.2研究意义**

本系统采用无人机和二维码技术实现智能停车，并采用了自创的车位智能推荐系统，可以实现自动找车位、推荐车位、停车后标记车位防忘等功能。相较于传统停车场更高效，与其他智能找车位技术相比具有成本低、易维护以及功能集成、对用户友好的特点，具有一定的实用价值。且本系统最为突出的特点是具有部署方便快速的特点，适合部署于临时停车场以方便管理。

1. 基于二维码技术和无人机拍摄的露天停车场智能停车系统结构组成

本系统结构上由四大模块组成：车位二维码模块、无人机拍摄模块、数据处理模块和信息反馈模块。系统整体结构如图1所示。

**图1 系统结构示意图**

**2.1设计理念**

本系统通过无人机巡航拍摄地面二维码识别车位状态。当车位被占用时，停在车位上的车会遮挡住印刷在车位上的二维码，无人机空中巡航时无法拍摄到。当车位未被占用时，印刷在车位上的二维码不会被遮挡，可以被无人机拍摄到。数据处理系统识别无人机拍摄的影像数据，在描述车位状态的文件中将识别出的二维码代表的车位状态更新为“空闲”，若在一段时间内未拍摄到代表某一车位的二维码，则将该车位的状态更新为“被占用”。信息反馈系统读取车位状态文件，并以图形交互方式将车位状态呈现给用户。无人机车位识别流程如图2所示。

**图2 无人机车位识别流程**

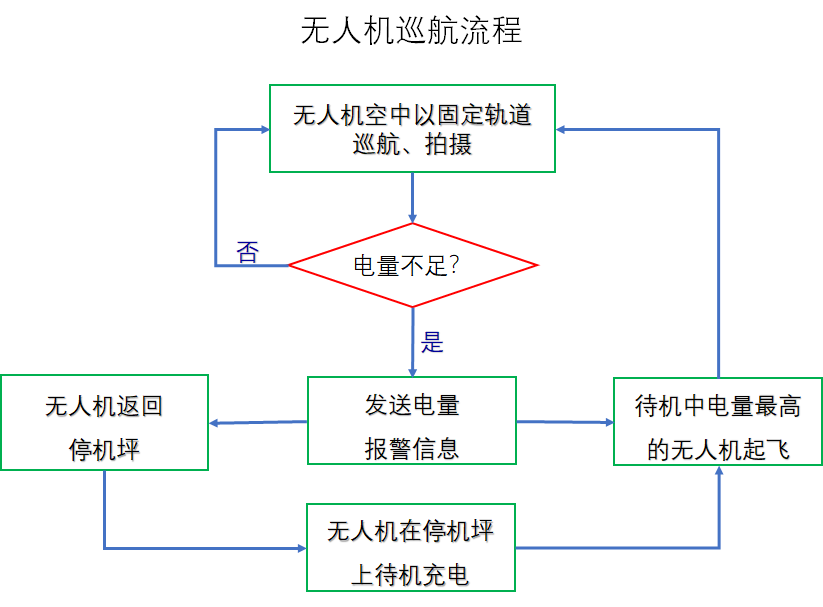
**2.2车位二维码模块**

每一个车位二维码模块由一大一小两个二维码组成，两个二维码的内容都是车位编号。其中大的二维码印刷在车位正中央，边长约为车位宽度的0.8倍，供无人机拍摄。小二维码印刷在车位一角，供车主停车后使用信息反馈模块扫描，起到为车主更新车位，以及更新车位状态的功能。印刷二维码的涂料采用自发性发光涂料，可以在夜晚发光，保障系统在夜晚也能正常工作。二维码的载体是大型贴纸等等价格低廉，生产快，改造停车场周期短，方便清除的一次性物品，在使用前快速粘贴到指定车位上。

**2.3无人机拍摄模块**

无人机拍摄模块由两台无人机及多块电池组成。无人机拍摄模块工作时由一台无人机在空中巡航，另外一台无人机在换上充满电的电池后待机，空闲的其他电池充电。空中巡航的无人机以固定轨道绕停车场飞行，该飞行模式由无人机内置芯片控制。其配备的摄像头俯拍地面，通过高清实时图传将影像数据实时传输到数据处理模块。

空中巡航的无人机低电量时，将自动向信息处理模块发送电量报警信息，并自动返航。信息处理系统收到电量报警信息后将命令待机中电量剩余最多的无人机起飞进行巡航。整体过程如图3所示。



**图3 无人机巡航流程**

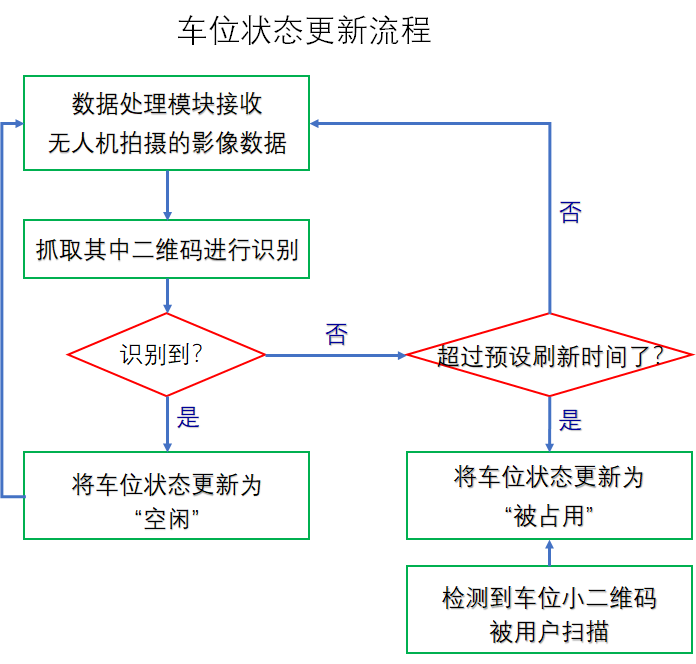
**2.4数据处理模块**

数据处理模块接收到无人机拍摄模块传输的影像数据后，进行二维码抓取与识别。二维码抓取与识别由计算机软件完成，该软件可使用python语言，调用zbar和OpenCV库进行编程与调试。

二维码信息识别后，数据处理模块通过软件算法，编辑车位状态文件，将识别出的二维码代表的车位更新为“空闲”。若在一段时间内未识别到代表某一车位的二维码，则数据处理模块将判定该车位被占用，并编辑车位状态文件，将该车位状态更新为“被占用”。

当数据处理模块接收到无人机电量报警信息后，将识别待机中的无人机的电量，选取电量最高的无人机，命令其起飞进行巡航。

当车位小二维码被信息反馈模块扫描时，信息反馈模块将识别二维码信息，向数据处理模块发送车位状态更新信息。数据处理模块接收到车位状态更新信息后将该车位状态更新为“被占用”。整体过程如图4所示。



**图4 车位状态更新流程**

**2.5信息反馈模块**

信息反馈模块读取车位状态信息，通过图形界面显示车位状态。图形界面为停车场的车位位置简化图，以不同颜色的圆角矩形代表车位，其中不同车位状态由不同颜色表示。

同时信息反馈模块会收集车主的位置信息，舍去离其他车主较近的车位，并根据附近设施推荐车位。

当车位小二维码被信息反馈模块扫描时，信息反馈模块将识别二维码信息，向数据处理模块发送车位状态更新信息。同时信息反馈模块将在图形界面用特殊颜色标记该车位。车主在取车时可以根据图形界面的颜色标记快速找到停车车位。车主使用无人机车位识别系统的流程如图5所示。

**图5 车主使用无人机车位识别系统的流程**

车主使用该系统明具体包括以下步骤，参见图6。

**图6 智能停车系统简单示例**

步骤一：若车主1是新用户，车主1扫描停车场入口处粘贴的小程序码进入信息反馈系统的小程序，通过微信账号登陆，停车场入口处摄像头扫描车主1车牌并上传到云存储器上，小程序向车主1确认车牌号，无误后将车牌号与车主1的微信绑定。小程序启动后将开启手机的定位功能，将车主1的位置实时上传到云存储器上。

若车主1使用过本系统，车主1无需扫描小程序码，停车场入口处摄像头扫描车主1车牌并提取云存储器上车主1的数据，以推送的形式将小程序打开方式发送到车主微信上。车主1启动小程序后小程序将开启手机的定位功能，将车主1的位置实时上传到云存储器上。

步骤二：信息反馈系统的小程序通过数据处理系统中各个车位的实时状态用不同颜色显示在小程序的车位图上，如图6中①、②、⑧所示。

步骤三：信息反馈系统会向云存储器上传车主的位置数据并接收其他车主的位置信息，将其余车主位置（如图6中车主2、3、4）和车主1的位置显示在小程序上以帮助车主1判断，如图6中⑤、⑥所示；用红色标记并舍去距离其他车主最近的车位，如图6中③所示；再通过行车距离测算得到离车主1最近且附近无其余停车车主的车位，用蓝色标记该车位，如图6中④所示；同时结合地图，用黄色标记离车主期望去的附近功能区最近的空车位，如图6中⑨、⑩所示，请车主1选择。车主1选择车位后用导航算法将最佳路线推荐给车主1。线路显示如图6中⑦所示。

步骤四：车主根据信息反馈系统停车完毕后扫描印刷在车位边角上的小二维码，信息反馈系统将小二维码信息传输给数据识别系统并结束对车主1的定位，保持车主1的车位显示为蓝色，将云存储器上该车位的状态更新该车位为“被占用”。

步骤五：若车主1为新用户，第一次停车结束后小程序将请车主进行实名注册，并建议车主完善个人车辆型号等相关信息。这些信息将与车牌号一起与车主1的微信绑定。

1. 样板软件结构

**3.1无人机的控制与调动**

无人机使用大疆Spark(晓)型无人机，由于大疆无人机比较封闭，于是使用大疆的官方软件进行无人机的控制。无人机的路线规划使用大疆“指点飞行”功能，通过预先规划好的路径定点，使无人机向指定点飞行。无人机飞行及二维码拍照如图7所示。

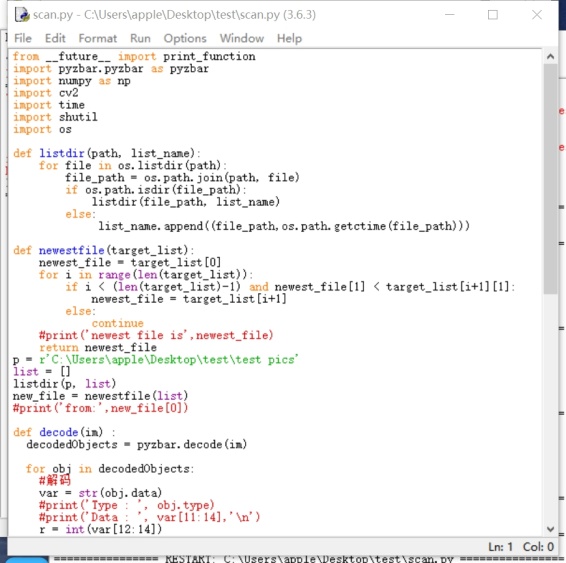
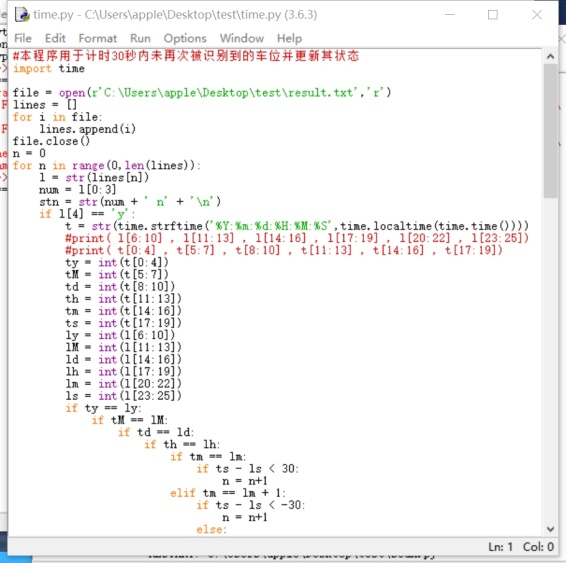
大疆无人机拥有低电量自动返航功能，在低电量时可以自主返航，同时由于大疆app无法向其他设备发出电量报警信号，所以多台无人机循环暂时无法实现。

**图7 无人机飞行过程及所拍二维码图**

**3.2二维码的抓取与识别**

本部分编写了三个程序以实现二维码的识别与信息提取、车位状态的存储及长时间未被扫描到的空闲车位状态的更新的功能。其中scan.py实现了识别与信息提取以及车位状态的存储，time.py实现了对长时间未被扫描到的空闲车位状态进行更新。通过运行run.py实现scan.py和time.py同时周期性运行，间隔约为五秒。所编写的Scan.py和time.py程序运行示意如图8所示。完整的run.py、scan.py和time.py程序如附图1-附图3所示。

**图8 scan.py和time.py程序运行示意图**

**3.2.1 二维码的识别与信息提取**

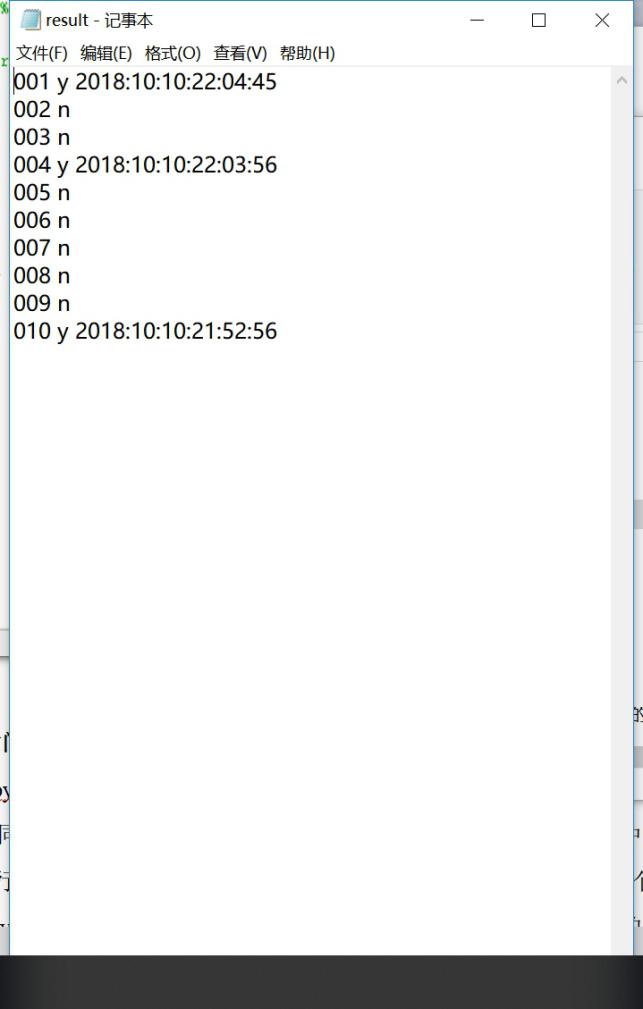
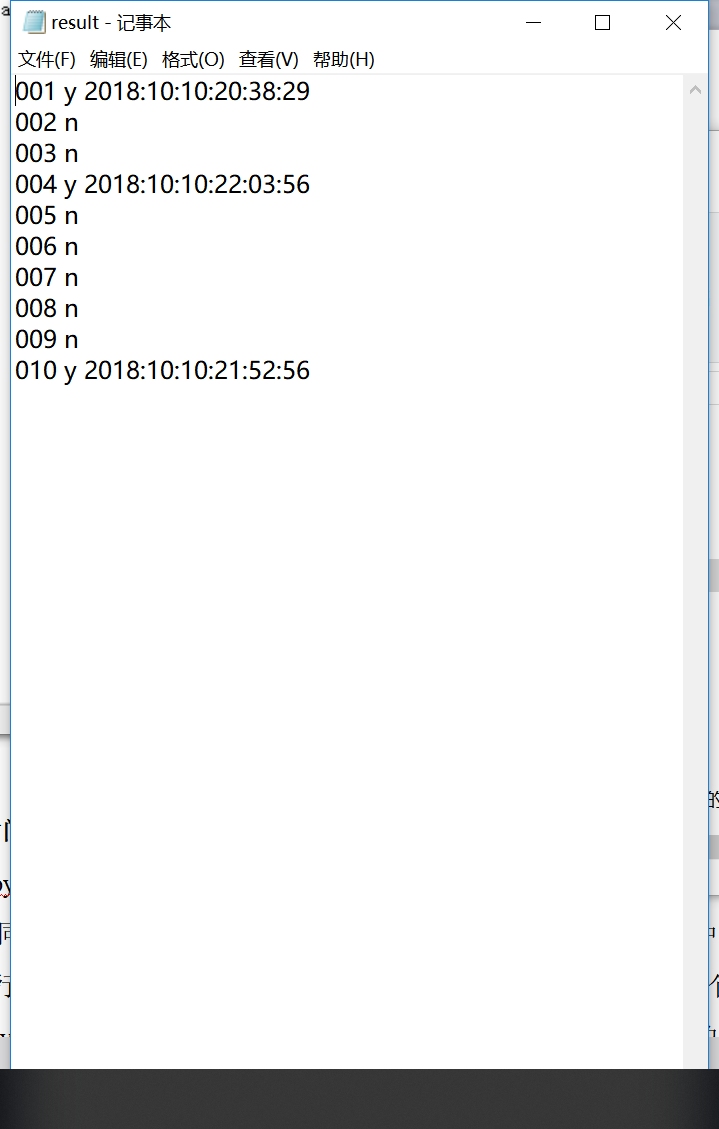
无人机航拍时可以拍摄视频和图片，供数据处理系统识别二维码。由于无人机航拍的图片具有分辨率高、清晰度高、易于识别等特点，同时本系统对车位状态刷新频率要求较低，所以实验时采用识别航拍图片的方法确认车位状态。

软件环境为Windows 10专业版，Python3.6.3。通过命令行使用pip安装zbar和OpenCV后，使用这两个库编写scan.py。首先使用zbar识别图像中的二维码信息，其中包括车位编号、印刷日期等。然后用OpenCV配合Python自带的Numpy框选并显示二维码图像。OpenCV与Numpy的功能与识别二维码无关，仅供调试使用，实际解码过程可以只使用zbar进行二维码的信息识别。识别出的二维码如图9所示，蓝色边框标识出了识别出的二维码边界。

****

**图9 运行scan.py识别框选出的测试用二维码图像**

**3.2.2 车位状态的存储**

scan.py使用zbar识别二维码后，先提取出识别到的车位编号，之后打开并读取历史车位状态文档result.txt，该文档每行形式为“编号 n”。引号内的空格与实际文件中的空格相同。其中编号为车位编号，由小到大依次排列且与本身行数相对应。scan.py在读取文档之后自动创建一个列表，其中每个元素代表文档的对应行。程序使用提取的车位编号从列表中找出对应元素，并更改该元素的内容为“编号 y 识别时间”。其中识别时间通过Python自带的time库进行时间的读取，包括年、月、日、时、分、秒的信息，且使用二十四小时制。scan.py未识别到的车位编号代表的元素内容不变，之后程序根据新的列表创建一个新的result.txt文档替换旧版本，文档内每行形式为“编号 y 识别时间”或“编号 n”。****编号的顺序与旧result.txt相同。旧result.txt和识别出的result.txt如图10所示(未启用time.py)。

**图10 旧result.txt和识别出的result.txt**

**3.2.3 长时间未被扫描到的空闲车位状态的更新**

time.py读取result.txt文件并自动创建一个列表，创造方式和元素类型与scan.py相同。time.py读取列表中元素代表车位状态的字符位(样板程序中为第五位)，并进行判定，若识别到“n”则保持该元素内容不变，并开始判定下一个元素。若识别到“y”，则继续进行判定，通过time库得到当前时间，并把时间的信息拆分为年、月、日、时、分、秒，一一与元素的时间信息进行比对判定。若最终比较并发现当前时间与元素的时间信息的时间差不超过30秒，则保持该元素内容不变，并开始判定下一个元素。若比较发现当前时间与元素的时间信息的时间差超过了30秒，则保持元素“y”字符前的部分不变，将“y”字符及其以后的字符替换成“n”，并进行下一个元素的判定。

**3.3系统状态的更新**

本系统采用基于阿里云的云端服务器进行信息存储与更新。在scan.py识别出二维码并生成results.txt文件之后，upload.py会自动联网向服务器上传该文件。upload.py使用python进行编程。使用urllib、requests、json库，且周期性运行，间隔与scan.py相同，大约为五秒。

**3.4图形交互系统设计**

图形交互系统位于用户客户端，通过读取云服务器上的车位状态文件，提取信息以更新车位状态。图形交互系统根据车位状态文件中的信息判断车位是否被占用，并通过不同颜色标记车位，显示车位状态。同时图形交互系统会根据停车场周围环境，进行智能推荐。智能推荐车位预先存储在停车场车位图中，以特别颜色标记。

1. 总结与展望

本系统实现了高精度且智能化的停车场车位管理。采用无人机和二维码技术实现智能停车，并采用了自创的车位智能推荐系统，实现自动找车位、推荐车位、停车后标记车位防忘等功能。相较于传统停车场更高效，与其他智能找车位技术相比，具有价格低廉、结构简单、维护方便等特点。

此外，通过定制无人机，可以实现无人机的1080p实时图传以及电量低时自动返航并命令待机中的无人机自动起飞等功能，使系统更加一体化、智能化、自动化。同时无人机还可以当作监控摄像头使用，具有视野广、死角少等特点，更加方便了对停车场的管理。

总而言之，本技术具有明显的创新性和较大的应用潜力，适合深入研究。

1. 参考文献
2. 赵军，李佳，刘泽俊，董志阳。智能停车场管理系统设计[J]。软件工程，2017，20（2）：44-46。
3. 王亿书，吕亚俊，魏博。基于计算机视觉的港区停车场泊位自动监测系统的设计与实现[J]。计算机应用与软件，2009，26（12）：75-78。
4. 黄泽满，刘勇，周星，陈天伟，梁毅东。民用无人机应用发展概述[J]。赤峰学院学报（自然科学版），2014，30（12）：30-32。
5. 王国辉。无线充电技术及其特殊应用前景[J]。电子产品世界，2014，（7）：21-24。
6. 吕跟来。停车场智能停车引导系统[J]。电脑知识与技术，2017，13 (25)：167-169。
7. 毕烨，陈峤鹰。基于计算机的停车场监控系统研究[J]。现代电子技术，2016，39（10）：84-87。
8. 潘冬阳，吴官朴，屈小琳，丁同强。基于 Pixhawk 的停车场无人机导航系统研究[J]。科教导刊，2018，（12）：271-272。
9. 任传祥，张梁，尹唱唱，高静，管德永，卢晓。基于太阳能无人机群的大型停车场车位检测系统[P]。实用新型专利，2016，授权公告号 CN2053543374U。
10. 苏金河，李秀维，付保保。一种结合拍摄无人机寻找车位装置[P]。发明专利申请，2017，申请公告号 CN 106781660 A。
11. 高英杰，叶全意。一种基于无人机的停车场监控方法[P]。发明专利申请，2018，申请公告号 CN 107886761 A。

附件：

**附件1 run.py程序：**

import os

import time

s = 1

while s == 1:

os.system(r'C:\Users\apple\Desktop\test\scan.py')

os.system(r'C:\Users\apple\Desktop\test\time.py')

time.sleep(5)

**附件2 scan.py程序：**

from \_\_future\_\_ import print\_function

import pyzbar.pyzbar as pyzbar

import numpy as np

import cv2

import time

import shutil

import os

def listdir(path, list\_name):

for file in os.listdir(path):

file\_path = os.path.join(path, file)

if os.path.isdir(file\_path):

listdir(file\_path, list\_name)

else:

list\_name.append((file\_path,os.path.getctime(file\_path)))

def newestfile(target\_list):

newest\_file = target\_list[0]

for i in range(len(target\_list)):

if i < (len(target\_list)-1) and newest\_file[1] < target\_list[i+1][1]:

newest\_file = target\_list[i+1]

else:

continue

#print('newest file is',newest\_file)

return newest\_file

p = r'C:\Users\apple\Desktop\test\test pics'

list = []

listdir(p, list)

new\_file = newestfile(list)

#print('from:',new\_file[0])

def decode(im) :

decodedObjects = pyzbar.decode(im)

for obj in decodedObjects:

#解码

var = str(obj.data)

#print('Type : ', obj.type)

#print('Data : ', var[11:14],'\n')

r = int(var[12:14])

#保存为文件

file = open(r'C:\Users\apple\Desktop\test\result.txt','r')

lines = []

for i in file:

lines.append(i)

file.close()

n=str(var[11:14] + ' y ' + str(time.strftime('%Y:%m:%d:%H:%M:%S',time.localtime(time.time()))) +'\n')

lines.pop(r-1)

lines.insert(r-1,n)

file\_write\_obj = open(r'C:\Users\apple\Desktop\test\result.txt', 'w')

for var in lines:

file\_write\_obj.writelines(var)

file\_write\_obj.close()

return decodedObjects

def display(im, decodedObjects):

#显示图片

for decodedObject in decodedObjects:

points = decodedObject.polygon

if len(points) > 4 :

hull = cv2.convexHull(np.array([point for point in points], dtype=np.float32))

hull = list(map(tuple, np.squeeze(hull)))

else :

hull = points;

n = len(hull)

for j in range(0,n):

cv2.line(im, hull[j], hull[ (j+1) % n], (255,0,0), 3)

cv2.namedWindow("Results",0);

cv2.resizeWindow("Results", 1280, 960);

cv2.imshow("Results", im);

cv2.waitKey(0);

#主程序

if \_\_name\_\_ == '\_\_main\_\_':

pic = new\_file[0]

im = cv2.imread(str(pic))

decodedObjects = decode(im)

display(im, decodedObjects)

**附件3 time.py程序：**

#本程序用于计时30秒内未再次被识别到的车位并更新其状态

import time

file = open(r'C:\Users\apple\Desktop\test\result.txt','r')

lines = []

for i in file:

lines.append(i)

file.close()

n = 0

for n in range(0,len(lines)):

l = str(lines[n])

num = l[0:3]

stn = str(num + ' n' + '\n')

if l[4] == 'y':

t = str(time.strftime('%Y:%m:%d:%H:%M:%S',time.localtime(time.time())))

#print( l[6:10] , l[11:13] , l[14:16] , l[17:19] , l[20:22] , l[23:25])

#print( t[0:4] , t[5:7] , t[8:10] , t[11:13] , t[14:16] , t[17:19])

ty = int(t[0:4])

tM = int(t[5:7])

td = int(t[8:10])

th = int(t[11:13])

tm = int(t[14:16])

ts = int(t[17:19])

ly = int(l[6:10])

lM = int(l[11:13])

ld = int(l[14:16])

lh = int(l[17:19])

lm = int(l[20:22])

ls = int(l[23:25])

if ty == ly:

if tM == lM:

if td == ld:

if th == lh:

if tm == lm:

if ts - ls < 30:

n = n+1

elif tm == lm + 1:

if ts - ls < -30:

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

elif th == lh + 1 or th == lh - 23:

if tm == lm - 59:

if ts - ls < -30:

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

elif td == ld + 1:

if th == lh - 23:

if tm == lm - 59:

if ts - ls < -30:

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

elif tM == lM + 1:

if td == ld - 30 or td == ld - 29 or td == ld - 28 or td == ld - 27:

if th == lh - 23:

if tm == lm - 59:

if ts - ls < -30:

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

elif ty == ly + 1:

if tM == lM - 11:

if td == ld - 30:

if th == lh - 23:

if tm == lm - 59:

if ts - ls < -30:

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

lines.pop(n)

lines.insert(n,stn)

n = n+1

else:

n = n+1

file\_write\_obj = open(r'C:\Users\apple\Desktop\test\result.txt', 'w')

for var in lines:

file\_write\_obj.writelines(var)

file\_write\_obj.close()