



(12)发明专利申请

(10)申请公布号 CN 111290955 A

(43)申请公布日 2020.06.16

(21)申请号 202010100224.5

(22)申请日 2020.02.18

(71)申请人 卡斯柯信号有限公司

地址 200070 上海市静安区天目中路428号  
凯旋门大厦27层C/D座

(72)发明人 陈晓轩 向王涛 张致恺 孙军峰  
金博 王祥丰 兰青辉

(74)专利代理机构 上海科盛知识产权代理有限公司 31225

代理人 应小波

(51)Int.Cl.

G06F 11/36(2006.01)

G06K 9/34(2006.01)

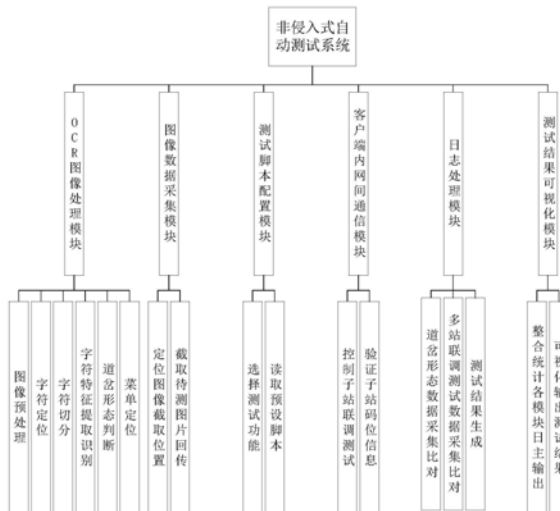
权利要求书1页 说明书5页 附图2页

(54)发明名称

用于地铁信号系统测试软件的非侵入式自动化测试系统

(57)摘要

本发明涉及一种用于地铁信号系统测试软件的非侵入式自动化测试系统,包括:OCR图像处理模块,用于识别处理地铁信号系统测试软件界面中的信息;图像数据采集模块,用于实时监控和采集地铁信号系统测试软件图形化界面中带有测试感兴趣信息的待处理图片;测试脚本配置模块,用于设定测试流程;客户端内网间通信模块,用于多机联动测试需求;日志处理模块,用于进行测试逻辑比对;测试结果可视化模块,用于对测试逻辑的输出结果进行可视化。与现有技术相比,本发明解决地铁信号系统测试软件中对测试人员依赖过重、不支持自动化测试的问题,有利于提高地铁信号系统测试的全自动化,有利于推动地铁的自动化进程,降低了开发、测试成本等优点。



CN 111290955 A

1. 一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,该系统基于OCR与形态学图像处理技术,所述的系统包括:

OCR图像处理模块,用于识别处理地铁信号系统测试软件界面中的信息;

图像数据采集模块,用于实时监控和采集地铁信号系统测试软件图形化界面中带有测试感兴趣信息的待处理图片;

测试脚本配置模块,用于设定测试流程;

客户端内网间通信模块,用于多机联动测试需求;

日志处理模块,用于进行测试逻辑比对;

测试结果可视化模块,用于对测试逻辑的输出结果进行可视化,便于测试人员复查。

2. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的OCR图像处理模块包括图像预处理单元、字符定位单元、字符分割单元、字符识别单元、目标菜单定位单元、地铁路段形态识别单元、和图像逻辑比对单元。

3. 根据权利要求2所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的OCR图像处理模块基于OCR图像识别技术对地铁信号系统测试软件图形化界面读取图像,并根据其图形化特征定位需要识别的信息,

首先通过学习的过程为生产过程参数建立对应的字符特征库,并构造分类器,然后对所识别的子图像进行二值化处理得到只有前景色和背景色的二值图片,再次对二值图像进行行、列字符切分获得单个字符,最后对分割出的单个字符进行特征提取,将提取到的特征与字符特征库进行匹配,匹配成功则识别出字符。

4. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的图像数据采集模块通过测试脚本的预设方式,由此根据测试需求编写测试脚本后,可采集到用户界面中的界面信息和操作点。

5. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的图像数据采集模块通过按键的中断方式,实时截取界面图片,用于作为测试样例和构造字符特征库。

6. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的测试脚本配置模块提供可编写的测试脚本,通过排布函数级的子模块,根据不同的测试需求组合成不同的测试流程。

7. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的客户端内网间通信模块,服务于地铁信号测试系统软件间的日志通信和路段状态逻辑判断,为更多多机联动测试的测试需求提供了测试基础。

8. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的日志处理模块用于记录当前时间、操作日志、运行输出日志。

9. 根据权利要求1所述的一种用于地铁信号系统测试软件的非侵入式自动化测试系统,其特征在于,所述的测试结果可视化模块在综合OCR图像处理模块、日志处理模块的输出后,统计测试结果并且通过统计方法,进而将测试结果可视化地输出,包括站点状态、逐条测试结果成功率、测试异常点信息,作为输出文件,提供给测试人员作为复查和核对的依据。

## 用于地铁信号系统测试软件的非侵入式自动化测试系统

### 技术领域

[0001] 本发明涉及软件图形用户界面测试技术领域,尤其是涉及一种基于OCR与形态学图像处理技术的用于地铁信号系统测试软件的非侵入式自动化测试系统。

### 背景技术

[0002] 自动化测试技术按照机制可以分为侵入式和非侵入式。侵入式测试技术采取某种方式修改测试目标软件的内部代码或者控制其运行环境;而非侵入式测试技术采取监控测试目标软件的用户界面,不修改软件内部结构或者代码,而是模拟测试人员完成测试流程。由于地铁信号系统的涉密性高,机制复杂,侵入式测试技术会更改测试目标软件的内部代码或者控制其运行环境,导致不可控的安全问题。所以在地铁信号系统测试软件这个场景下适合采用非侵入式自动测试机制。

[0003] 现阶段对于地铁信号系统的测试基本上还是停留在测试人员根据需求文档,先用自然语言写好测试需求,然后将地铁站内信息制作成测试表,由测试人员逐条阅读,通过键盘或者鼠标输入测试条件,用眼睛观察用户界面的对于地铁线路的图形化信息和对应的站点信息,整个测试过程非常缓慢而且由于人为疏忽容易导致错误。

[0004] 具有图形用户接口的地铁信号系统测试软件,对图形界面的输入操作必须由手工操作完成,输出判断也必须由人脑对照着测试需求表进行。由于测试人员的操作差异,被测试系统的性能不能被完全利用,系统的测试速度受限于测试人员的速度,系统的测试准确度也受限于测试人员的判断准确度,而且没有可视的量化的测试效果。而且地铁信号系统测试需求多样复杂,不同版本的地铁信号系统测试软件的操作和界面有差异,无法使用固定的流程和操作方式覆盖所有的测试需求。

### 发明内容

[0005] 本发明的目的就是为了解决上述现有技术存在的缺陷而提供一种用于地铁信号系统测试软件的非侵入式自动化测试系统。

[0006] 本发明的目的可以通过以下技术方案来实现:

[0007] 一种用于地铁信号系统测试软件的非侵入式自动化测试系统,该系统基于OCR与形态学图像处理技术,所述的系统包括:

[0008] OCR图像处理模块,用于识别处理地铁信号系统测试软件界面中的信息;

[0009] 图像数据采集模块,用于实时监控和采集地铁信号系统测试软件图形化界面中带有测试感兴趣信息的待处理图片;

[0010] 测试脚本配置模块,用于设定测试流程;

[0011] 客户端内网间通信模块,用于多机联动测试需求;

[0012] 日志处理模块,用于进行测试逻辑比对;

[0013] 测试结果可视化模块,用于对测试逻辑的输出结果进行可视化,便于测试人员复查。

[0014] 优选的,所述的OCR图像处理模块包括图像预处理单元、字符定位单元、字符分割单元、字符识别单元、目标菜单定位单元、地铁路段形态识别单元、和图像逻辑比对单元。

[0015] 优选的,所述的OCR图像处理模块基于OCR图像识别技术对地铁信号系统测试软件图形化界面读取图像,并根据其图形化特征定位需要识别的信息,

[0016] 首先通过学习的过程为生产过程参数建立对应的字符特征库,并构造分类器,然后对所识别的子图像进行二值化处理得到只有前景色和背景色的二值图片,再次对二值图像进行行、列字符切分获得单个字符,最后对分割出的单个字符进行特征提取,将提取到的特征与字符特征库进行匹配,匹配成功则识别出字符。

[0017] 优选的,所述的图像数据采集模块通过测试脚本的预设方式,由此根据测试需求编写测试脚本后,可采集到用户界面中的界面信息和操作点。

[0018] 优选的,所述的图像数据采集模块通过按键的中断方式,实时截取界面图片,用于作为测试样例和构造字符特征库。

[0019] 优选的,所述的测试脚本配置模块提供可编写的测试脚本,通过排布函数级的子模块,根据不同的测试需求组合成不同的测试流程。

[0020] 优选的,所述的客户端内网间通信模块,服务于地铁信号测试系统软件间的日志通信和路段状态逻辑判断,为更多多机联动测试的测试需求提供了测试基础。

[0021] 优选的,所述的日志处理模块用于记录当前时间、操作日志、运行输出日志。

[0022] 优选的,所述的测试结果可视化模块在综合OCR图像处理模块、日志处理模块的输出后,统计测试结果并且通过统计方法,进而将测试结果可视化地输出,包括站点状态、逐条测试结果成功率、测试异常点信息,作为输出文件,提供给测试人员作为复查和核对的依据。

[0023] 与现有技术相比,本发明具有以下优点:

[0024] (1) 本发明采用非侵入式的自动测试机制,通过基于OCR的图像处理技术,并且模拟测试人员的操作,输出结果日志,对测试流程中具有重复性的操作进行了自动化实现,在未对系统性能产生影响、未改变系统稳定性的前提下实现了一种减少测试人员数量的自动测试方法;而在传统的针对地铁信号系统测试软件的使用和测试中,要求大量的测试人员,对测试文档有深入了解后耗费大量时间逐条手动测试;

[0025] (2) 本发明实现的多机联合测试功能,通过内网间通信实现了地铁信号系统中测试的基本需求,极大地简化了多机联合的测试流程,节约了人力成本;而在传统的地铁信号系统测试过程中,多机联合测试需要多个测试人员协同测试,同步查看操作是否实现预期要求,测试流程复杂,测试人工成本高;

[0026] (3) 本发明可支持多个测试软件版本、多个不同地铁站点的测试,不需要根据不同的测试软件版本、不同的地铁站点去开发不同的客户端,省去了大量的研发与测试成本;而传统的地铁信号系统测试软件每次软件版本或者站点变动需要对测试人员重新测试,测试成本高,版本切换成本高,站点变动成本高;

[0027] (4) 本发明不仅支持现有测试需求,对于地铁信号测试中还未提出的需求,也可通过组合需求的操作,编辑测试脚本,从而实现自动化测试,大大降低了研发和测试的成本;而传统测试流程需要对新出现的测试需求,转换成自然语言,训练测试人员,形成测试流程,测试流程周期长,人工成本高;

[0028] (5) 本发明支持新老机型不同测试环境的自动化测试,对于地铁信号测试场景中的大量旧式台式机同样支持,解决了传统测试方案实施的局限性;而传统测试中,由于新旧设备差距大,导致不同测试人员测试速度不一致,测试需求间的无法协调统一同步测试,测试成本高。

### 附图说明

[0029] 图1为本发明非侵入式自动化测试系统的功能模块图。

[0030] 图2为本发明中所述的测试脚本配置模块的工作流程图。

[0031] 图3为本发明中所述的客户端内网间通信模块的流程图。

[0032] 图4为本发明中各个模块在实际使用中的协同工作流程图。

[0033] 图5为本发明中所述的日志处理模块中涉及的状态机的验证方式示意图。

### 具体实施方式

[0034] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明的一部分实施例,而不是全部实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都应属于本发明保护的范围。

[0035] 如图1所示,本发明是一种基于OCR图像处理技术的针对地铁信号系统测试软件的非侵入式自动化测试系统,包括OCR图像处理模块1、图像数据采集模块2、测试脚本配置模块3、客户端内网间通信模块4、日志处理模块5、测试结果可视化模块6。本发明的OCR图像处理模块1用于识别处理地铁信号测试系统中的屏幕信息,包括图像预处理、字符定位、字符切分、字符特征提取识别、道岔形态判断、菜单定位;该OCR图像处理模块1是基于OCR图像识别技术,对地铁信号测试系统运行过程中产生的参数建立对应的字符特征库,并构造分类器。(1) 在站点识别定位的子任务中,首先对所识别图像先通过预处理去除站点、站点菜单、地铁线路、路线预警等噪声,再通过图形图像处理操作定位字符位置,对字符位置进行二值化处理得到只有前景色和背景色的二值图像,然后对二值图像进行行、列字符切分获得单个字符,最后对单个字符进行特征提取,将提取到的特征与字符特征库进行匹配,匹配成功则识别出字符;(2) 在菜单定位的子任务中,首先对所识别图像先通过预处理去除站点字符、地铁线路、路线预警等噪声,再通过图形图像处理操作拟合出菜单位置,对菜单边界做平滑处理后,识别出菜单的位置;(3) 在道岔形态判断的子任务中,首先对所识别图像先通过预处理去除站点字符、菜单、路线预警等噪声,再通过图形图像处理操作,依据道岔特征,拟合判断出道岔形态,识别出道岔形态。

[0036] 本发明的图像数据采集模块2用于实时采集地铁信号系统测试软件运作过程中需要进行识别定位的图像。由于地铁信号测试环境中,新老设备差距大,测试过程无动态连续图像变化,故不采用视频流图像数据采集,而采用对设备性能依赖度低的瞬时图像采集的方式。又由于地铁信号测试软件需求表中提供的每条测试任务都是随机初始状态的,前后测试任务的所在位置和图像空间顺序不可预测且无逻辑关联,故图像数据采集模块2采用了初始标定测试空间和测试前多次初始化位置的鲁棒性高的策略,具体指在地铁信号系统测试软件开始测试前,截取多屏的站点初始信息调用OCR图像处理模块1识别完毕作为测试

标尺和坐标,在测试进行中,由预先定位的标尺和坐标辅助定位,精准定位到测试指令所在位置,再次截取待测图像,并将待测图像回传给OCR图像处理模块1进行对于目标字段或者路段的识别。

[0037] 本发明的测试脚本配置模块3用于实现不同测试任务的子操作的逻辑组合。如图2测试脚本配置模块流程图所示,测试人员先预先根据测试需求按照预定义的格式编写测试脚本,在实际测试时,本发明的测试脚本配置模块3中的脚本解释器会根据测试任务读取所需的脚本文件,然后循环执行脚本内容,将本发明中的其他模块按照测试要求的逻辑顺序串联成与测试人员实际操作相一致的测试流程,达到模拟测试人员实操的测试效果。并且针对地铁信号测试系统测试需求的特点,单条测试任务要求对不同的站点、路段进行相同的重复操作,解释器通过读取测试列表,循环执行任务,以此减少对于同一场景的计算机空间资源需求,并且加快整体的测试流程。针对特定的测试需求,该测试脚本配置模块3不仅可以模拟测试人员的操作流程,包括左键、右键、选择站点、选取菜单、滚动滚动条位置等操作,还可以对于屏幕位置做初始化,增强测试过程的鲁棒性,使单个测试脚本可以运用于不同的车站,大大降低了开发和测试成本。开发测试人员只需要排列组合可实现的子操作,按照测试人员的操作逻辑顺序,即可编写出直观的测试脚本,并且切实可行。可选用不同的测试脚本针对不同的测试需求,也可将多个测试脚本做逻辑组合,对新需求新功能进行测试,大大提升了本发明的适用范围和扩展性。

[0038] 本发明的客户端内网间通信模块4用于实现多个站点间的联调测试需求。对于地铁信号系统测试软件中多站联调的测试需求,该模块通过TCP通信协议发送开始测试信号,通过输出的日志文件回传作为测试的验证信息,完整覆盖了针对办理连锁进路的多机联调测试需求。如图3客户端内网间通信的流程图所示,中控工作站首先与子工作站建立TCP连接,在测试过程中,本发明的客户端内网间通信模块在中控工作站将测试需求内容以约定的编码形式编写成协议内容,并向子工作站发送TCP协议包;子工作站监听TCP协议包,接受TCP协议包后,根据约定的解码形式解析协议并做出相关动作。在具体测试任务中,中控工作站向子工作站发送办理连锁进路测试请求后,子工作站会根据系统内逻辑办理固定连锁路段的进路,以此模拟列车通过该路段的动态过程,并且子工作站生成此过程的日志信息,通过TCP通信协议回传至中心工作站,中心工作站将子站回传日志与测试需求中的预期路段日志对比统计,输出保存多机联调办理连锁进路的测试日志,供测试人员排查。客户端内网间通信模块3作为多机联调测试需求的基础工具之一,为地铁信号系统测试软件的自动化测试中更多的多机联调测试需求打下了理论和测试基础。

[0039] 本发明的日志处理模块5用于处理各个不同需求的测试日志,根据地铁信号系统测试的测试需求和逻辑关系,比对测试过程中生成的各种日志与测试需求预期日志,并生成测试结果。具体的,针对道岔状态判断的测试需求,日志处理模块5对比测试过程中不同站点道岔在不同操作下的图形学形状,并且用01标志位表示,实现与测试需求中结果的布尔判断相一直,比对后生成道岔状态的测试结果。针对多站联调的测试需求,日志处理模块5对比子站通过UDP协议回传的日志与测试需求中的预期路段日志,统计后输出多站联调的测试结果。

[0040] 本发明的测试结果可视化模块6用于将日志处理模块5的结果整合统计,并做可视化输出,便于测试人员排错与查看。

[0041] 本发明在模仿测试人员进行地铁信号系统的测试时,各个模块协同工作的逻辑流程如图4所示。首先根据预先配置好的测试脚本配置模块3调用Windows API操作,模拟测试人员在测试实际流程中的键盘鼠标操作;依据脚本配置,在需要定位界面中站点信息时,调用图像数据采集模块2,截取OCR图像处理模块3所需的界面截图,并将包含需要识别信息的截图传给OCR图像处理模块;OCR图像处理模块3根据脚本配置中不同的任务,采用不同的图像处理策略,经过图片预处理、中英文字符检测识别、轨道岔路形态识别等流程后,将当前任务中感兴趣的图像信息在整张图中定位并记录,构建带有信息位置坐标的整体测试图像;对OCR图像处理模块1中所识别到的关键信息,日志处理模块5通过状态机验证的方式,判别当前测试任务是否成功;测试结果可视化模块6根据日志处理模块5的判别结果,整合多次测试的结果,并且输出可视化的测试日志供测试人员查看。

[0042] 上文所述的日志处理模块5中的状态机的验证方式,以实际中测试需求中测试道岔的逻辑为例,如图5所示。该状态机针对测试道岔的需求,首先通过图像数据采集模块2采集测试过程中对道岔操作前后的界面截图,然后通过OCR图像处理模块1经过图像预处理、字符定位、道岔特征判断等流程,判断出道岔前后形态的变化。由此得到当前测试界面在状态机中的当前状态,对当前状态道岔形态进行判断,若道岔不为水平:验证道岔的状态,若当前状态符合预期,则验证经过轨道路线;若当前状态不符合预期,则判断该条测试失败。若道岔为水平:验证结果轨道路线,若经过轨道路线不符合预期,则判断该条测试失败;若经过轨道路线符合预期,则判断该条测试成功。状态机的判断结果均通过日志处理模块5记录,传交给可视化模块6最终进行结果整合以及可视化输出。

[0043] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

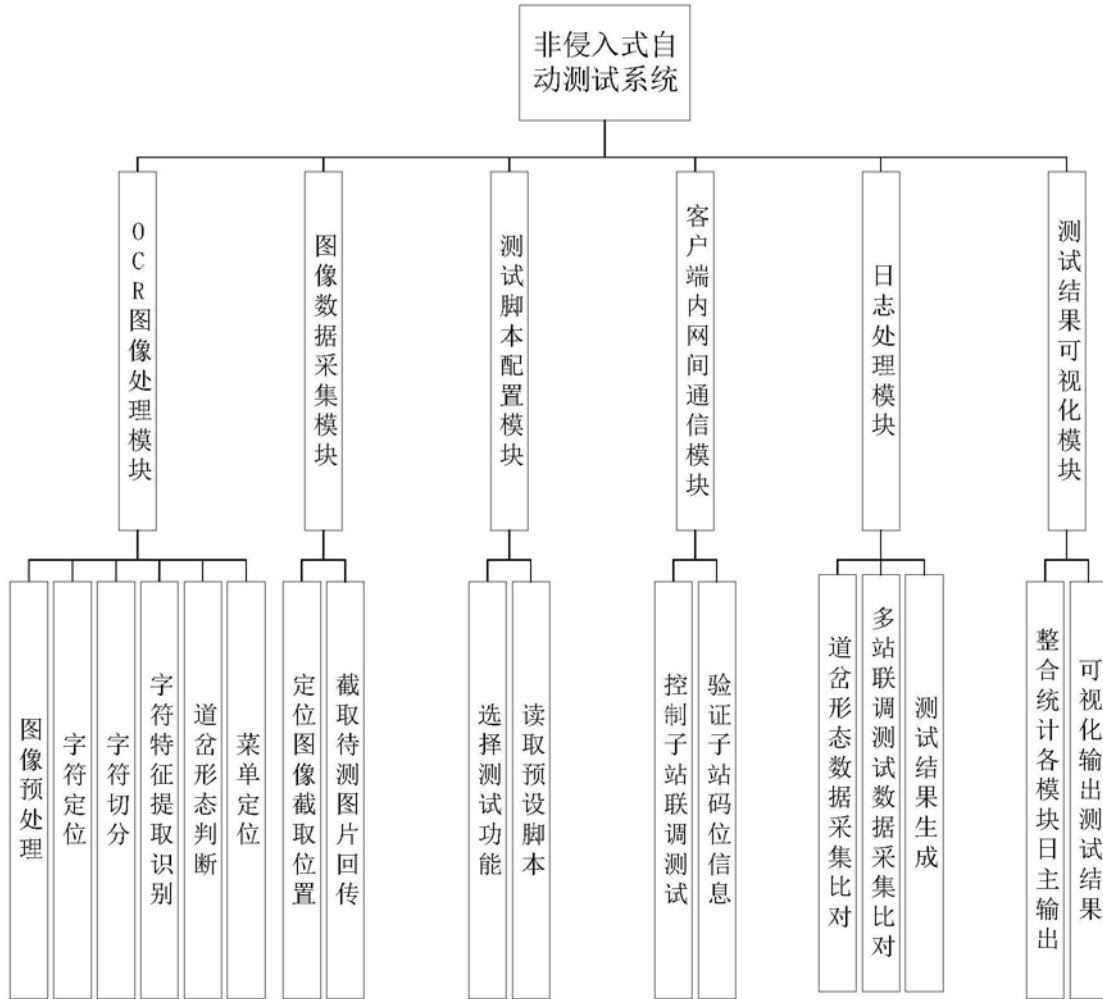


图1

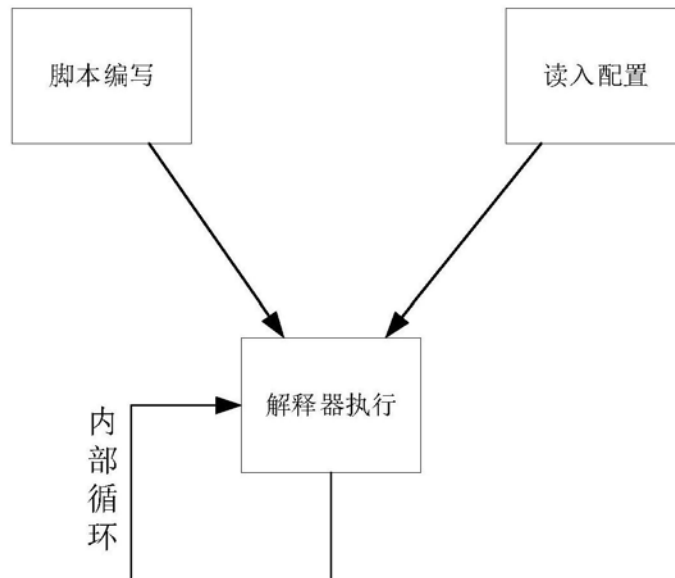


图2





图3

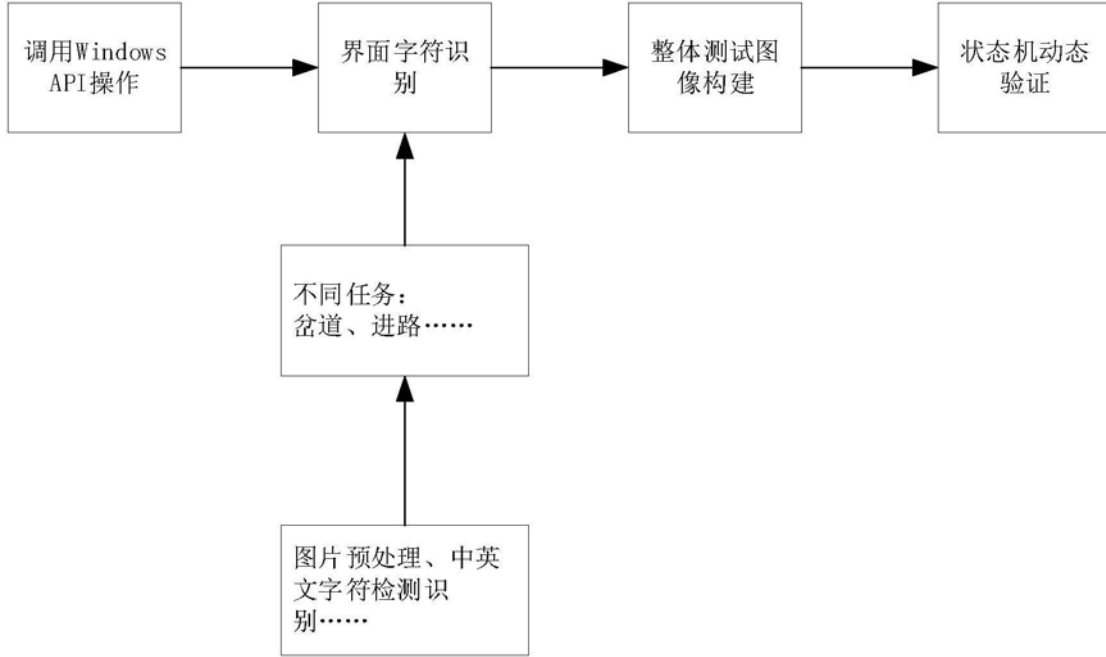


图4

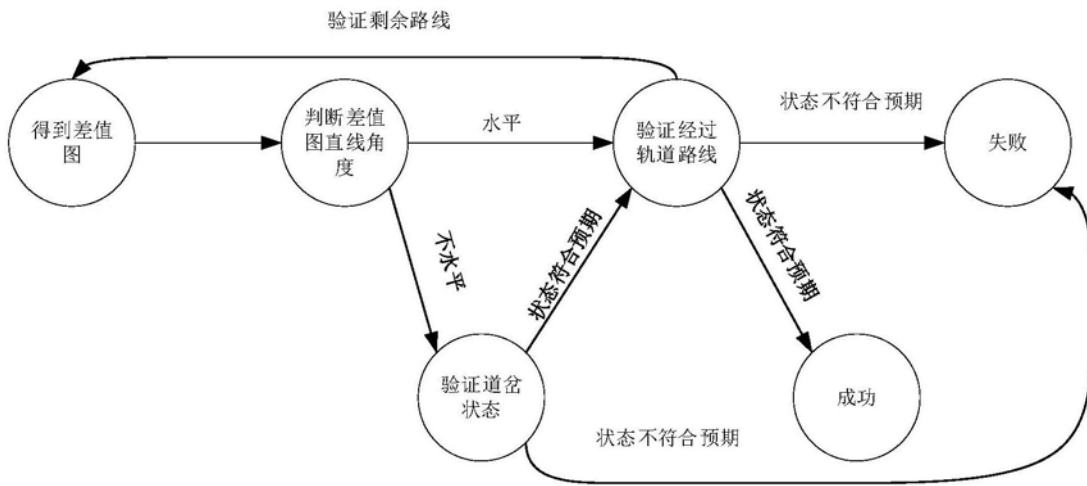


图5