



# (12)发明专利申请

(10)申请公布号 CN 109858421 A

(43)申请公布日 2019.06.07

(21)申请号 201910069844.4

(22)申请日 2019.01.24

(71)申请人 广东第二师范学院

地址 510303 广东省广州市海珠区新港中路351号

(72)发明人 梁嘉慧 卢燕萍 陈怡航 张淑莲  
谢敏峰 肖选良 朱雄泳 陈忆群  
张渝荣

(74)专利代理机构 广州粤高专利商标代理有限公司 44102

代理人 林丽明

(51)Int.Cl.

G06K 9/00(2006.01)

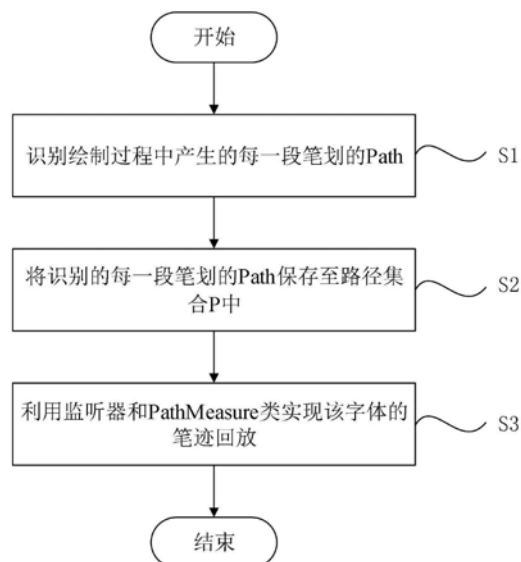
权利要求书2页 说明书5页 附图7页

## (54)发明名称

一种基于可缩放矢量图形的安卓平台笔迹回放方法及装置

## (57)摘要

本发明公开了一种基于可缩放矢量图形的安卓平台笔迹回放方法,包括以下步骤:S1:识别绘制过程中产生的每一段笔划的Path;S2:将识别的每一段笔划的Path保存至路径集合P中;S3:利用监听器和PathMeasure类实现该字体的笔迹回放。本发明将笔迹用Path类保存为可缩放矢量图形文件,大大减少存储空间,占用内存少且便于往后实现回放功能直接进行代码层面的操作;能够有效降低回放过程的计算量,简化回放代码,并且具备良好的动画效果;也不需要存储,符合实时性,能更灵活地实现实时回放书写过程。本发明能极大减少存储空间的同时,也能极大简化回放代码,提高代码易读性。



1. 一种基于可缩放矢量图形的安卓平台笔迹回放方法,其特征在于,包括以下步骤:

S1: 识别绘制过程中产生的每一段笔划的Path;

S2: 将识别的每一段笔划的Path保存至路径集合P中;

S3: 利用监听器和PathMeasure类实现该字体的笔迹回放。

2. 根据权利要求1所述的基于可缩放矢量图形的安卓平台笔迹回放方法,其特征在于,所述每一段笔划的Path保存为可缩放矢量图形文件。

3. 根据权利要求2所述的基于可缩放矢量图形的安卓平台笔迹回放方法,其特征在于,所述步骤S1的识别绘制过程中产生的每一段笔划的Path,其中识别一段笔划的Path的具体步骤中用一字体集合处理模块对Path进行处理操作,具体为:

S1.1: 定义 $L_0$ 为当前Path长度, $L_t$ 为前一Path长度, $L_0=L_t=0$ ;

S1.2: 定义 $P_0$ 为当前处理Path,绘制过程中,每生成一段Path,则把该段Path存入 $P_0$ ,测量一次 $P_0$ 的长度 $l$ ,令 $L_0=l$ ;

S1.3: 若 $L_0$ 大于 $L_t$ ,则执行S1.4;若 $L_0$ 不大于 $L_t$ ,执行S1.5;

S1.4: 令 $L_t=L_0$ ,定义 $L_H$ 为历史Path长度,令 $T_p$ 表示操作类型, $T_p \in (0,1)$ ,其中 $T_p=0$ 代表替换模式, $T_p=1$ 代表添加模式;

若 $L_0$ 大于0,则 $L_H=L_0$ , $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=0$ ;

跳过步骤S1.5;

S1.5: 若 $L_0$ 不等于0或 $L_H$ 等于0,返回步骤S1.2;若 $L_0$ 等于0且 $L_H$ 不等于0, $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=1$ ;

S1.6: 在字体集合处理模块中判断 $T_p$ 的值:

若 $T_p=0$ ,把 $P_0$ 与路径集合P中最后加入的Path替换;

若 $T_p=1$ ,把 $P_0$ 加入路径集合P中。

4. 根据权利要求3所述的基于可缩放矢量图形的安卓平台笔迹回放方法,其特征在于,所述步骤S2的具体步骤为:

S2.1: 判断路径集合P是否已创建,若未创建则创建一个路径集合P;

S2.2: 定义 $P_s$ 为目标Path变量, $L_s$ 为 $P_s$ 的长度,传入的Path存入 $P_s$ ,并测量 $P_s$ 长度 $l_s$ , $L_s=l_s$ ;

S2.3: 若路径集合P为空,把 $P_s$ 直接加入P;

若路径集合P不为空,且 $T_p=1$ 则将 $P_s$ 加入至P中;

若路径集合P不为空,且 $T_p=0$ 则替换P中最后加入的Path;

S2.4: 重复步骤S2.1至S2.3直至将所有笔划的Path存入P。

5. 根据权利要求4所述的基于可缩放矢量图形的安卓平台笔迹回放方法,其特征在于,所述步骤S3的具体步骤为:

S3.1: 遍历路径集合P,定义 $P_i$ 为获取的第 $i$ 个Path;

定义 $S_i$ 为动画状态, $S_i=Start$ , $S_i \in \{Start, None\}$ ;

定义监听器A和监听器B,其中监听器A为AnimatorListener类型的监听器,监听器B为AnimatorUpdateListener类型的监听器;

S3.2: 通过监听器A改变动画状态,具体为:

重写方法onAnimationStart,设置动画状态为Start,表示动画开始的状态为Start;重

写方法onAnimationEnd,设置动画状态为None,表示动画结束的状态为None;

S3.3:通过监听器B的getAnimatedValue方法获取自动更新的动画属性值 $m \in [0,1]$ ,当获取并更新一次动画属性值 $m$ ,则:

强制当前视图绘制,若 $S_i = \text{Start}$ ,截取该 $P_i$ 进行绘制,定义 $L_x$ 为 $P_i$ 的部分长度, $L_M$ 为 $P_i$ 的完整长度,则 $L_x = L_M * m$ ,多次迭代,实现回放动画效果;

若 $S_i = \text{None}$ ,则将完整笔划重绘并显示在画板上。

6.一种基于可缩放矢量图形的安卓平台笔迹回放装置,其特征在于,包括:

识别模块,用于识别绘制过程中产生的每一段笔划的Path,所述识别模块包括字体集合处理模块;

封装模块,所述封装模块用于将识别的每一段笔划的Path保存至路径集合P中;

动画回放模块,所述动画回放模块利用监听器和PathMeasure类实现该字体的笔迹回放。

## 一种基于可缩放矢量图形的安卓平台笔迹回放方法及装置

### 技术领域

[0001] 本发明涉及笔迹显示领域,更具体地,涉及一种基于可缩放矢量图形的安卓平台笔迹回放方法及装置。

### 背景技术

[0002] 智能移动设备平台上的回放技术已遍布许多安卓(Android)软件中,比如:字帖类软件,画图类软件等。在市面上,大多数软件采取的回放笔迹技术,目前比较流行的两种算法:一是历史坐标回溯法,此方法需根据事先存储字迹的历史定位坐标与回放的设定参数,从数据库中检索得到后在将轨迹再在画板上重新绘制,绘制完成代码繁琐且需要较大的内存,不符合移动设备轻应用的需求;二是通过录制屏幕生成媒体文件再重放,也有存储量大的问题。Android自带的Picture类,其使用过程需要存储记录绘图操作,需要存储的媒体文件造成无益的空间占用并且不能很好体现实时性

### 发明内容

[0003] 本发明为克服上述现有技术所述的至少一种缺陷,提供一种基于可缩放矢量图形的安卓平台笔迹回放方法及装置。

[0004] 本发明旨在至少在一定程度上解决上述技术问题。

[0005] 本发明的首要目的是提供一种基于可缩放矢量图形的笔迹回放方法,解决现有技术中存储量大且代码繁琐的问题

[0006] 本发明的进一步目的是提供一种基于可缩放矢量图形的笔迹回放装置。

[0007] 为解决上述技术问题,本发明的技术方案如下:

[0008] 一种基于可缩放矢量图形的安卓平台笔迹回放方法,包括以下步骤:

[0009] S1:识别绘制过程中产生的每一段笔划的Path;

[0010] S2:将识别的每一段笔划的Path保存至路径集合P中;

[0011] S3:利用监听器和PathMeasure类实现该字体的笔迹回放;

[0012] 上述方法中以相对成本较低的方式解决难题,即使用集合存储可缩放矢量图形(SVG)格式的每笔笔划,同时结合能够直接改变属性值的动画类ValueAnimator类,通过属性动画让任何对象的属性都跟着时间的改变进行变化,同时通过PathMeasure类追踪Path的坐标进行绘制定位,最终实现书写过程回放的功能,从而有效简化了代码,避免存储过多的参数以及产生媒体文件占用过大内存,也大大减少了存储空间。

[0013] 优选地,每一段笔划的Path保存为SVG文件,SVG是可伸缩矢量图形(Scalable Vector Graphics),Path类是SVG基本形状中最强大的一个,绘制过程中的笔迹均用Path类保存为SVG文件,由于SVG是一个纯粹的XML文件,占用内存小,能够大大地减少存储空间,也便于往后实现回放功能直接进行代码层面的操作。

[0014] 优选地,步骤S1的识别绘制过程中产生的每一段笔划的Path,其中识别一段笔划的Path的具体步骤中用一字体集合处理模块对Path进行处理操作,具体为:

- [0015] S1.1: 定义 $L_0$ 为当前Path长度, $L_t$ 为前一Path长度, $L_0=L_t=0$ ;
- [0016] S1.2: 定义 $P_0$ 为当前处理Path,绘制过程中,每生成一段Path,则把该段Path存入 $P_0$ ,测量一次 $P_0$ 的长度 $l$ ,令 $L_0=1$ ;
- [0017] S1.3: 若 $L_0$ 大于 $L_t$ ,则执行S1.4;若 $L_0$ 不大于 $L_t$ ,执行S1.5;
- [0018] S1.4: 令 $L_t=L_0$ ,定义 $L_h$ 为历史Path长度,令 $T_p$ 表示操作类型, $T_p \in (0,1)$ ,其中 $T_p=0$ 代表替换模式, $T_p=1$ 代表添加模式;
- [0019] 若 $L_0$ 大于0,则 $L_h=L_0$ , $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=0$ ;
- [0020] 跳过步骤S1.5;
- [0021] S1.5: 若 $L_0$ 不等于0或 $L_h$ 等于0,返回步骤S1.2;若 $L_0$ 等于0且 $L_h$ 不等于0, $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=1$ ;
- [0022] S1.6: 在字体集合处理模块中判断 $T_p$ 的值:
- [0023] 若 $T_p=0$ ,把 $P_0$ 与路径集合P中最后加入的Path替换;
- [0024] 若 $T_p=1$ ,把 $P_0$ 加入路径集合P中。
- [0025] Path类封装了由线段、二次曲线与三次曲线组成的复合(多轮廓)几何路径。它可以用Canvas类的drawPath方法进行绘制、填充,或者用于裁剪或以路径的方式绘制文本。由于在利用Canvas类实现绘制功能的绘画过程中,一段连续线条将产生多段Path,本步骤中只要迭代识别出该字的每段连续线条完整的一笔。
- [0026] 优选地,步骤S2的具体步骤为:
- [0027] S2.1: 判断路径集合P是否已创建,若未创建则创建一个路径集合P;
- [0028] S2.2: 定义 $P_s$ 为目标Path变量, $L_s$ 为 $P_s$ 的长度,传入的Path存入 $P_s$ ,并测量 $P_s$ 长度 $l_s$ , $L_s=l_s$ ;
- [0029] S2.3: 若路径集合P为空,把 $P_s$ 直接加入P;
- [0030] 若路径集合P不为空,且 $T_p=1$ 则将 $P_s$ 加入至P中;
- [0031] 若路径集合P不为空,且 $T_p=0$ 则替换P中最后加入的Path;
- [0032] S2.4: 重复步骤S2.1至S2.3直至将所有笔划的Path存入P。
- [0033] 优选地,步骤S3的具体步骤为:
- [0034] S3.1: 遍历路径集合P,定义 $P_i$ 为获取的第 $i$ 个Path;
- [0035] 定义 $S_i$ 为动画状态, $S_i=Start$ , $S_i \in \{Start, None\}$ ;
- [0036] 定义监听器A和监听器B,其中监听器A为AnimatorListener类型的监听器,监听器B为AnimatorUpdateListener类型的监听器;
- [0037] S3.2: 通过监听器A改变动画状态,具体为:
- [0038] 重写方法onAnimationStart,设置动画状态为Start,表示动画开始的状态为Start;重写方法onAnimationEnd,设置动画状态为None,表示动画结束的状态为None;
- [0039] S3.3: 通过监听器B的getAnimatedValue方法获取自动更新的动画属性值 $m \in [0, 1]$ ,当获取并更新一次动画属性值 $m$ ,则:
- [0040] 强制当前视图绘制,若 $S_i=Start$ ,截取该 $P_i$ 进行绘制,定义 $L_x$ 为 $P_i$ 的部分长度, $L_M$ 为 $P_i$ 的完整长度,则 $L_x=L_M*m$ ,多次迭代,实现回放动画效果;
- [0041] 若 $S_i=None$ ,则将完整笔划重绘并显示在画板上;
- [0042] Android的AnimatorListener类可以从动画接收通知,通知指示动画的相关事件。

而Android的另一个监听类——AnimatorUpdateListener类可以通过调用getAnimatedValue函数接收每个动画帧通过ValueAnimator计算的回调值。

[0043] 一种基于可缩放矢量图形的安卓平台笔迹回放装置,包括:

[0044] 识别模块,所述识别模块用于识别绘制过程中产生的每一段笔划的Path,所述识别模块包括字体集合处理模块;

[0045] 封装模块,所述封装模块用于将识别的每一段笔划的Path保存至路径集合P中;

[0046] 动画回放模块,所述动画回放模块利用监听器和PathMeasure类实现该字体的笔迹回放。

[0047] 与现有技术相比,本发明技术方案的有益效果是:

[0048] 本发明将笔迹用Path类保存为SVG文件,大大减少存储空间,占用内存少且便于往后实现回放功能直接进行代码层面的操作;基于属性值动画原理以及ValueAnimator类实现对笔迹的回放,只需要对只需要对动画的监听去做一些操作,在监听中将这个值设置给Path的长度属性,不断更新这个属性值便可实现动画效果。能够有效降低回放过程的计算量,简化回放代码,并且具备良好的动画效果;也不需要存储,符合实时性,能更灵活地实现实时回放书写过程。本发明能极大减少存储空间的同时,也能极大简化回放代码,提高代码易读性。

## 附图说明

[0049] 图1为一种基于可缩放矢量图形的安卓平台笔迹回放方法的流程图;

[0050] 图2为识别绘制过程中产生的每一段笔划的Path的流程图;

[0051] 图3为将识别的每一段笔划的Path保存至路径集合P中的流程图;

[0052] 图4为利用监听器和PathMeasure类实现该字体的笔迹回放的流程图;

[0053] 图5为笔迹原图展示图;

[0054] 图6为笔迹回放展示图;

[0055] 图7为一种基于可缩放矢量图形的安卓平台笔迹回放装置示意图。

## 具体实施方式

[0056] 附图仅用于示例性说明,不能理解为对本专利的限制;

[0057] 为了更好说明本实施例,附图某些部件会有省略、放大或缩小,并不代表实际产品的尺寸;

[0058] 对于本领域技术人员来说,附图中某些公知结构及其说明可能省略是可以理解的。

[0059] 下面结合附图和实施例对本发明的技术方案做进一步的说明。

[0060] 实施例1

[0061] 本实施例提供一种基于可缩放矢量图形的安卓平台笔迹回放方法,如图1,包括以下步骤:

[0062] S1:识别绘制过程中产生的每一段笔划的Path;

[0063] S2:将识别的每一段笔划的Path保存至路径集合P中;

[0064] S3:利用监听器和PathMeasure类实现该字体的笔迹回放;

- [0065] 每一段笔划的Path保存为SVG文件;
- [0066] 步骤S1的识别绘制过程中产生的每一段笔划的Path,其中识别一段笔划的Path的具体步骤中用一字体集合处理模块对Path进行处理操作,如图2,具体为:
- [0067] S1.1:定义 $L_0$ 为当前Path长度, $L_t$ 为前一Path长度, $L_0=L_t=0$ ;
- [0068] S1.2:定义 $P_0$ 为当前处理Path,绘制过程中,每生成一段Path,则把该段Path存入 $P_0$ ,测量一次 $P_0$ 的长度 $l$ ,令 $L_0=1$ ;
- [0069] S1.3:若 $L_0$ 大于 $L_t$ ,则执行S1.4;若 $L_0$ 不大于 $L_t$ ,执行S1.5;
- [0070] S1.4:令 $L_t=L_0$ ,定义 $L_h$ 为历史Path长度,令 $T_p$ 表示操作类型, $T_p \in (0,1)$ ,其中 $T_p=0$ 代表替换模式, $T_p=1$ 代表添加模式;
- [0071] 若 $L_0$ 大于0,则 $L_h=L_0$ , $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=0$ ;
- [0072] 跳过步骤S1.5;
- [0073] S1.5:若 $L_0$ 不等于0或 $L_h$ 等于0,返回步骤S1.2;若 $L_0$ 等于0且 $L_h$ 不等于0, $P_0$ 传入处理 $P_0$ 的字体集合处理模块, $T_p=1$ ;
- [0074] S1.6:在字体集合处理模块中判断 $T_p$ 的值:
- [0075] 若 $T_p=0$ ,把 $P_0$ 与路径集合P中最后加入的Path替换;
- [0076] 若 $T_p=1$ ,把 $P_0$ 加入路径集合P中。
- [0077] 步骤S2的具体步骤如图3,具体为:
- [0078] S2.1:判断路径集合P是否已创建,若未创建则创建一个路径集合P;
- [0079] S2.2:定义 $P_s$ 为目标Path变量, $L_s$ 为 $P_s$ 的长度,传入的Path存入 $P_s$ ,并测量 $P_s$ 长度 $l_s$ , $L_s=l_s$ ;
- [0080] S2.3:若路径集合P为空,把 $P_s$ 直接加入P;
- [0081] 若路径集合P不为空,且 $T_p=1$ 则将 $P_s$ 加入至P中;
- [0082] 若路径集合P不为空,且 $T_p=0$ 则替换P中最后加入的Path;
- [0083] S2.4:重复步骤S2.1至S2.3直至将所有笔划的Path存入P。
- [0084] 步骤S3的具体步骤如图4,具体为:
- [0085] S3.1:遍历路径集合P,定义 $P_i$ 为获取的第i个Path;
- [0086] 定义 $S_i$ 为动画状态, $S_i=Start$ , $S_i \in \{Start, None\}$ ;
- [0087] 定义监听器A和监听器B,其中监听器A为AnimatorListener类型的监听器,监听器B为AnimatorUpdateListener类型的监听器;
- [0088] S3.2:通过监听器A改变动画状态,具体为:
- [0089] 重写方法onAnimationStart,设置动画状态为Start,表示动画开始的状态为Start;重写方法onAnimationEnd,设置动画状态为None,表示动画结束的状态为None;
- [0090] S3.3:通过监听器B的getAnimatedValue方法获取自动更新的动画属性值 $m \in [0,1]$ ,当获取并更新一次动画属性值 $m$ ,则:
- [0091] 强制当前视图绘制,若 $S_i=Start$ ,截取该 $P_i$ 进行绘制,定义 $L_x$ 为 $P_i$ 的部分长度, $L_M$ 为 $P_i$ 的完整长度,则 $L_x=L_M*m$ ,多次迭代,实现回放动画效果;
- [0092] 若 $S_i=None$ ,则将完整笔划重绘并显示在画板上。
- [0093] 在具体实施过程中,如图5和图6,图5表示笔迹原图,精油本实施例提供的方法进行笔迹还原,具体过程展示如图6所示,可以很好的还原笔迹。

[0094] 实施例2

[0095] 本实施例提供一种基于可缩放矢量图形的安卓平台回放装置,如图7,包括:

[0096] 识别模块,所述识别模块用于识别绘制过程中产生的每一段笔划的Path,所述识别模块包括字体集合处理模块;

[0097] 封装模块,所述封装模块用于将识别的每一段笔划的Path保存至路径集合P中;

[0098] 动画回放模块,所述动画回放模块利用监听器和PathMeasure类实现该字体的笔迹回放。

[0099] 相同或相似的标号对应相同或相似的部件;

[0100] 附图中描述位置关系的用语仅用于示例性说明,不能理解为对本专利的限制;

[0101] 显然,本发明的上述实施例仅仅是为清楚地说明本发明所作的举例,而并非是对本发明的实施方式的限定。对于所属领域的普通技术人员来说,在上述说明的基础上还可以做出其它不同形式的变化或变动。这里无需也无法对所有的实施方式予以穷举。凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明权利要求的保护范围之内。



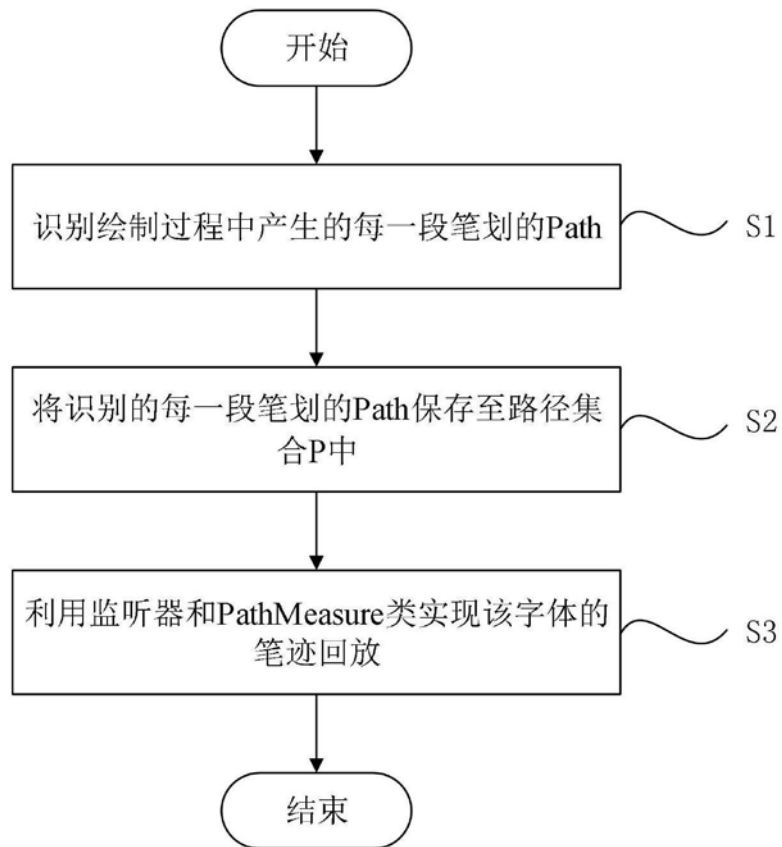


图1

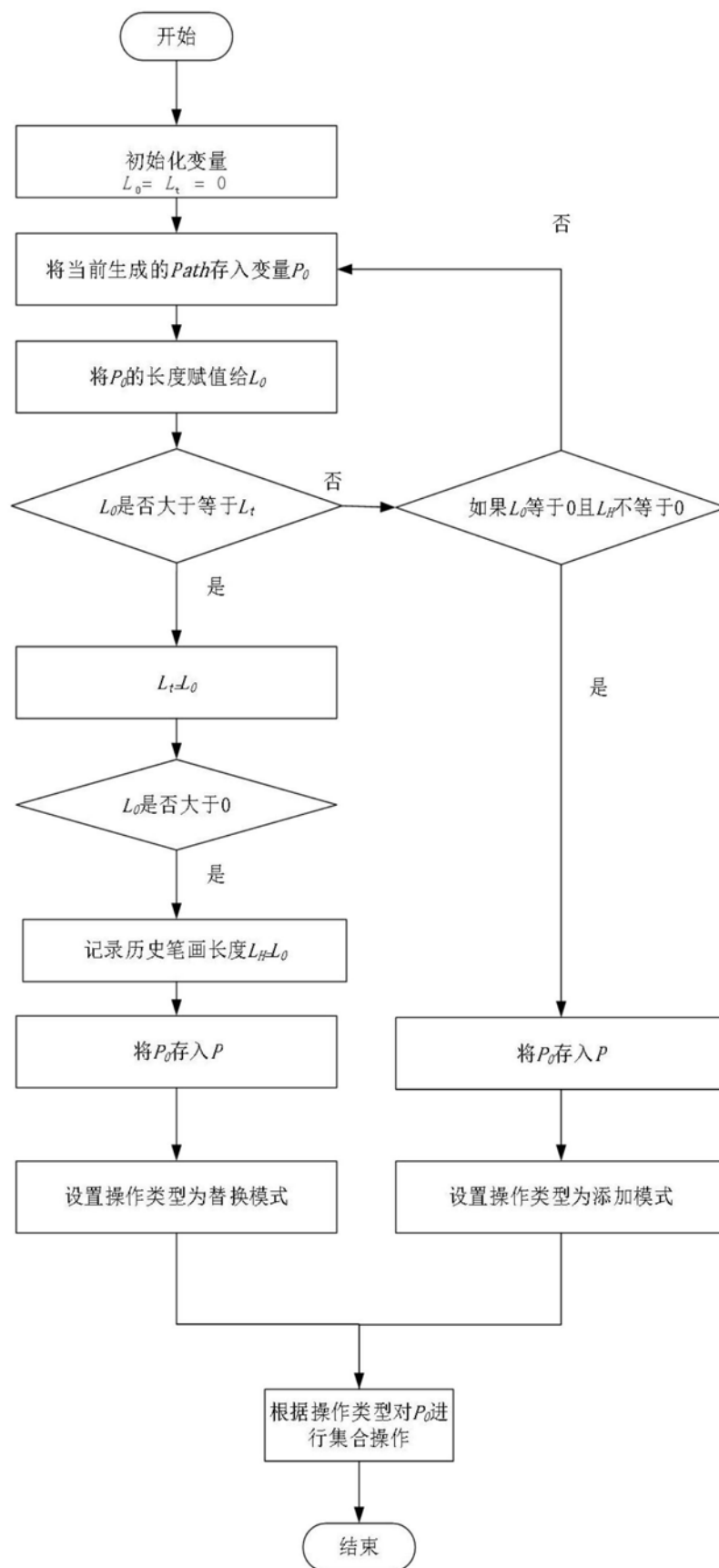


图2

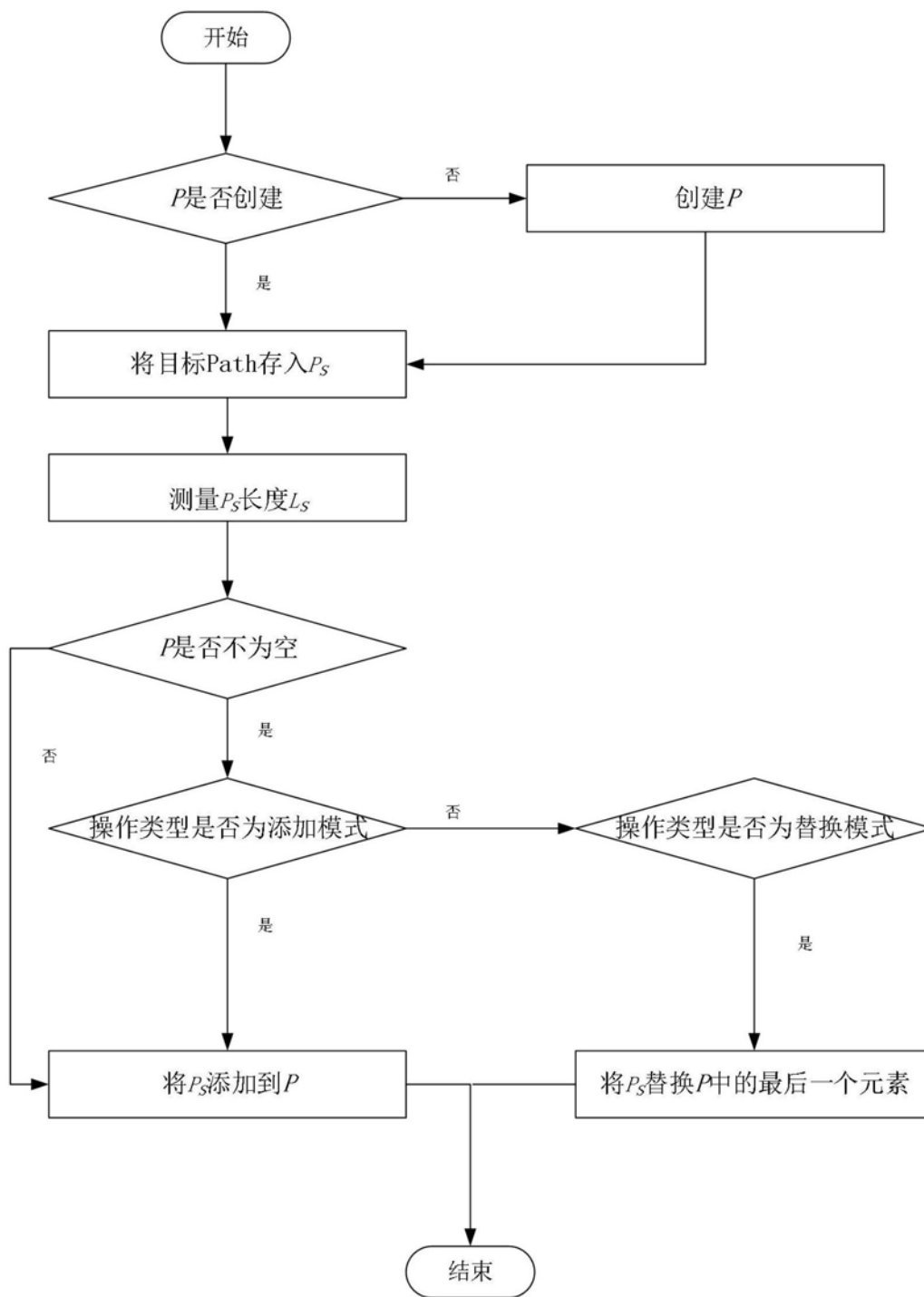


图3

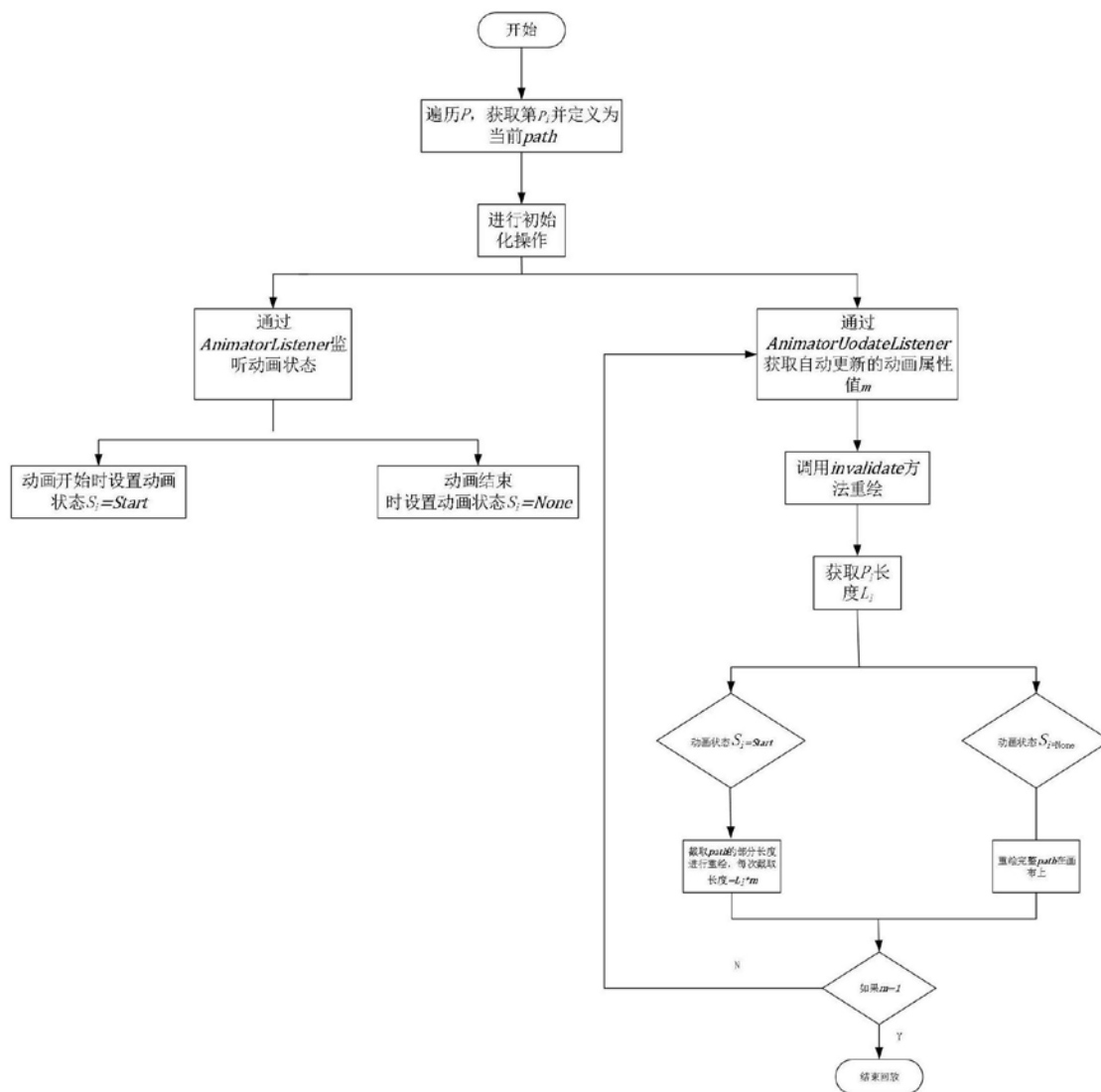


图4

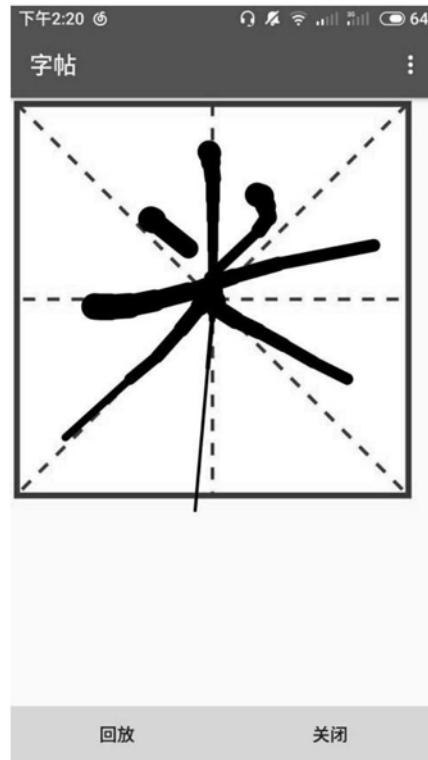


图5

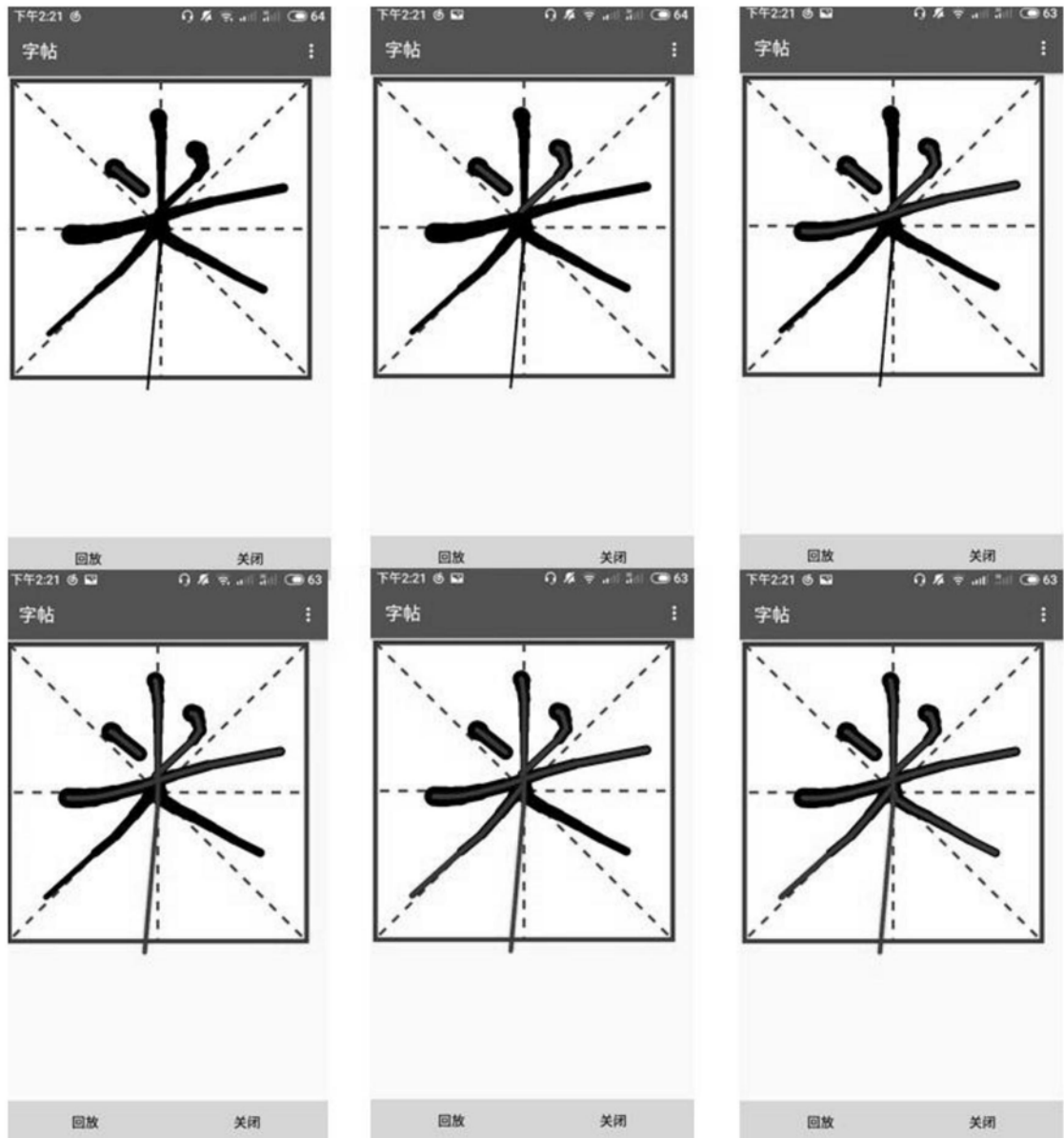


图6

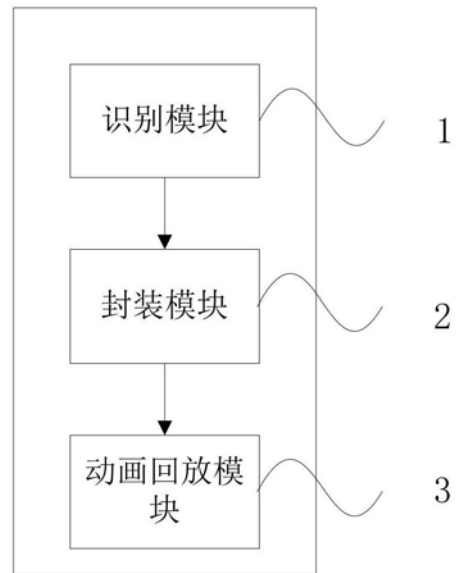


图7