著 录 项 目

发明名称：一种基于界面图标特征的安卓软件重打包检测方法

申请人：

地址：

发明人（包括第一发明人身份证号）：赵士轩(230103199803125711) 王明 曾子涵 马骏

总联系人：

联系电话：

联系电邮：

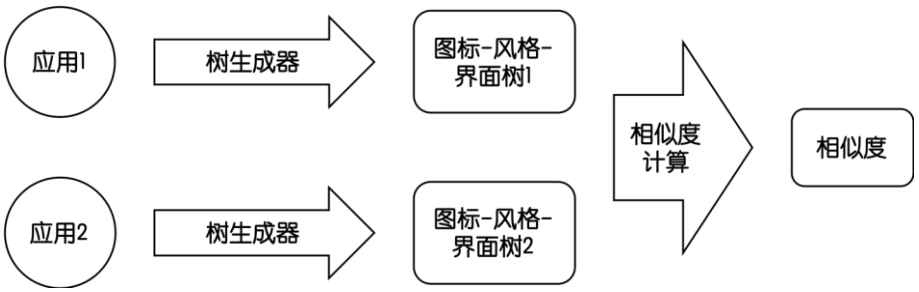
撰写人：赵士轩，17600598771，shixuan.zhao@hotmail.com

代理人：

说 明 书 摘 要

本发明提供了一种基于界面图标特征的安卓应用重打包检测方法，其特征在于，包括以下步骤：步骤 1、动态执行安卓应用，对每个界面筛选并转储界面的全部图标；步骤 2、对转储的图标进行根据风格进行分类，并将转储的图标-风格-界面树作为应用软件胎记；步骤 3、在多个应用间比较转储图标的风格和图标图像本身的相似程度得出相似度，从而判定出是否重打包。本发明的安卓软件相似度比对方法使用图标这一视觉特征结合动态执行安卓应用的策略，提高了对代码混淆和加密这两类反检测方式的抵抗性，提高了安卓应用重打包检测的准确性。

摘 要 附 图



权 利 要 求 书

1．一种基于界面图标特征的安卓应用重打包检测方法，其特征在于，包括以下步骤：

步骤 1、动态执行安卓应用，对每个界面筛选并转储界面的全部图标；

步骤 2、对转储的图标进行根据风格进行分类，并将转储的图标-风格-界面树作为应用软件胎记；

步骤 3、在多个应用间逐对比较转储图标的风格和图标图像本身的相似程度得出应用对之间的相似度，从而判定待检测应用对是否是重打包应用对。

2．根据权利要求 1 步骤 2 所述的重打包检测方法，其特征在于，图标-风格-界面树的特征在于：

安卓应用的图标是指安卓应用界面中抽象化的、用于标示某个功能而使用的小型图片。图标是应用开发者为了提示用户操作当前界面的特定功能而设计的，因此随着界面的跳转会发生改变。

数据模型图标-风格-界面树是一个有向图，$G=(V,E)$。其中，V表示界面图标及特征的集合，E表示界面的转移；

G的节点$v=(I,S,l)\in V$，表示一个特定界面的图标及特征的集合。其中I是图标的集合；S为一个映射$I\to\mathbb{N}$，表示图标的风格；l标示界面的层计数，即从初始界面转移到当前界面最少经过的操作数，也是在图标-风格-界面树中界面所在的层数。这个界面视为树的一个节点。而G的边$e\in E\subseteq V\times V$，$e=(v_1,v_2)$意味着存在从界面$v_1$到界面$v_2$的转移。

3．根据权利要求 1 步骤 2 所述的重打包检测方法，其特征在于，采用有限循环动态执行安装应用并构造界面的图标与图标对应风格的集合，包括以下步骤：

3.1 筛选并转储当前界面的全部图标，并对每一个图标进行风格分类，将转储的图标与图标对应的风格添加到输出集合中；

3.2 在布局中随机选择一个筛选过的可交互控件，然后随机选择一个该控件所支持的行为；

3.3 在所选控件上执行所选行为，转移至一个新的界面，循环返回 2.1 所述步骤；

3.4 当循环计数达到设定的阈值，跳出循环，输出。

4．根据权利要求 3 所述的重打包检测方法，其特征在于，有限循环动态执行具体包括：

批注 [jm1]: 还是直接定义为树？

批注 [jm2]: 没有定义， 具体指代什么？

批注 [jm3]:

在循环开始前，层计数器l归零，当前选中界面$u=(C,I,a)$置为软件的初始化界面u_i，其中C为所有控件的集合，a为已经在当前界面执行的操作数；

进入主循环后：如果l小于一个给定阈值δ_l，则返回上一级界面，l减去 1。否则首先令a为零，筛选出当前界面u中控件集合C中的有效控件集合$C_{valid}\in C$，根据有效控件集合C_v选择出有效图标集合$I_{valid}\in I$；如果在现有的图标-风格-界面树中找到了相似的界面，则认定重复，返回上一层界面，l减去 1；否则将有效图标分类出风格后与界面和层数一同加入输出的图标-风格-界面树。

接下来如果a小于给定阈值δ_a，则在C_{valid}中找到任意一个没有选择过的可交互控件c，随机选择一个该控件所支持的行为在控件c上执行操作，a增加 1，并认为转移到了下一层界面$u'=(C',I',a')$，l增加 1，重复执行主循环。如果a大于等于给定阈值δ_a，若当前界面为初始化界面u_i，则退出循环，算法结束；否则返回上一级界面，l减去 1。

5．根据权利要求 4 所述的重打包检测方法，其特征在于：构建图标-风格-界面树时，一组界面的相似度计算方法如下：

5.1 首先将两个界面转储出的图标分别按照风格分为几个集合，每个集合图标的风格相同

5.2 然后以两个界面中风格相同的图标为点集构图进行最大匹配，最大匹配中判定边存在的方式为两个图标重合部分大于设定阈值δ；

5.3 两个界面的相似度：两个界面中图标被判定相同的个数所占两个界面总图标数量的比例η；

6．根据权利要求 1 步骤 3 所述的重打包检测方法，其特征在于，软件相似度计算方法为：将一对图标-风格-界面树转化为多个二部图，通过计算每个二部图的最大匹配值得到图标-风格-界面树之间的相似度。

7．根据权利要求 6 所述的重打包检测方法，其特征在于：

图标-风格-界面树转为多个二部图的特征为：设 2 个图标-风格-界面树G_1和G_2，每个图根据层计数可以分为δ_l个集合。设第i层包含的界面集合为G_{1i}和G_{2i}，则二部图点集为$G_{1i}.V\cup G_{2i}.V$，边存在的判定方式为：使用权利要求 3 中计算一组界面的方式计算出相似度η，若η大于一个给定阈值δ_η则判定相同。每层构建一个二部图，可以得到多个二部图组成的集合$\{G_{b1},G_{b2},\cdots,G_{b\delta_l}\}$。对每个二部图，可以通过最大匹配计算出相似度：第$i$个二部图的相似度为最大匹配判定相同的界面个数占二部图总界面个数的比例，记

批注 [jm4]: 到底是 u 还是 v？前后不一致

批注 [jm5]: 这句话感觉表述不是太清楚，最大匹配往往是在一个图中进行的，你这里是否隐含构图了？

为φ_i。树相似度计算公式如下：

$$Sim_g(G_1, G_2) = \frac{\sum_{i=1}^{\delta_l} |\varphi_i \cdot (|G_{1i}.V| + |G_{2i}.V|)|}{\sum_{i=1}^{\delta_l} (|G_{1i}.V| + |G_{2i}.V|)}$$

批注 [jm6]:

批注 [jm7R6]: \phi_i 这个公式看不懂，定义域是整数？

8．根据权利要求 6 所述的重打包检测方法，其特征在于：当一对图标-风格-界面树之间的相似度大于阈值δ_G时，对应的两个应用被判断为重打包对。

说　　明　　书

一种基于界面图标特征的安卓应用重打包检测方法

技术领域

本发明专利涉及软件应用，图像分类，软件胎记，重打包检测领域，尤其涉及一种基于界面图标特征的安卓应用重打包检测方法。

背景技术

以智能手机为代表的智能设备的普及以及移动互联网技术的快速发展，推动了移动互联网软件产业的蓬勃发展。较传统PC软件而言，诸如支付宝、微信等新兴移动应用更加贴近用户日常生活，其市场规模庞大且增长迅速。而应用重打包已经成为恶意软件传播的主要途径之一，威胁到用户的安全和隐私，侵犯了开发者/发布者的知识产权、经济利益，阻碍了移动应用产品的创新，严重危害移动互联网行业的正常秩序和生态。软件胎记技术是一种重要的、有良好应用前景的软件知识产权保护技术，受到普遍关注，目前该技术取得了一定的研究进展和应用效果。然而通过加密与混淆的手段，进行重打包的攻击者们会在代码中加入迷惑自动化分析软件的代码与控件。对此，静态分析手段几乎完全失效。而采用非基于视觉特征的动态分析方法在应用经过了代码混淆和界面混淆的情况下容易误判，两个对于人来说相似的界面在软件实现上可能完全不同。

发明内容

本发明主要针对上述不足，提出了一种基于界面图标特征的安卓应用重打包检测方法。

该方法基于以下现实：攻击者在进行重打包操作时，会确保重打包后的应用在视觉特征上与原始应用一致。因此我们选择视觉特征中极为重要的界面图标特征来提取软件胎记，并用此判定安卓软件的重打包。

本发明具体提供的功能包括：

1）自动化地执行并遍历安卓应用；

2）收集安卓应用用户界面图标特征信息；

3）反馈安卓应用界面之间的相似度；

4）反馈安卓应用的软件胎记；

5）提供安卓应用之间的相似度；

6）检测安卓应用是否重打包。

说　明　书

本发明的技术方案为：基于界面图标特征的安卓应用重打包检测方法，主要过程包括：

1. 动态执行安卓应用，收集用户界面图标；
2. 通过获得的图标生成图标-风格-界面树；
3. 通过比较图标-风格-界面树的相似度来判断应用之间是否重打包。

安卓应用的图标是指安卓应用界面中抽象化的，用于标示某个功能而使用的小型图片。图标是应用开发者为了提示用户操作当前界面的特定功能而设计的，因此随着界面的跳转会发生改变。

基于界面图标特征的安卓应用重打包检测方法，其描述安卓应用运行时界面图标特征的数据模型图标-风格-界面树的特征在于：

数据模型图标-风格-界面树是一个有向图，$G=(V,E)$。其中，V表示界面图标及特征的集合，E表示界面的转移；

G的节点$v=(I,S,l)\in V$，表示一个特定界面的图标及特征的集合。其中I是图标的集合；S为一个映射$I\to\mathbb{N}$，表示图标的风格；l标示界面的层计数。这个界面视为树的一个节点。而G的边$e\in E\subseteq V\times V$，$e=(v_1,v_2)$意味着存在从界面$v_1$到界面$v_2$的转移。

动态执行安卓应用并构造图标-风格-界面树的过程的特征在于一个有限循环算法。所述的有限算法具体特征为：

在循环开始前，用户指定循环层深层δ_l与宽度δ_a。计数器l归零，当前选中界面$u=(C,I,a)$置为软件的初始化界面u_i，其中C为所有控件的集合，a为已经在当前界面执行的操作数；

进入主循环后：如果l小于一个给定阈值δ_l，则返回上一级界面，l减去 1。否则首先令a为零，筛选出当前界面u中控件集合C中的有效控件集合$C_v\in C$，根据有效控件集合C_v选择出有效图标集合$I_v\in I$；如果在现有的图标-风格-界面树中找到了相似的界面，则认定重复，返回上一层界面，l减去 1；否则将有效图标分类出风格后与界面和层数一同加入输出的图标-风格-界面树。

接下来如果a小于给定阈值δ_a，则在C_v中找到任意一个没有选择过的可交互控件c，随机选择一个该控件所支持的行为在控件c上执行操作，a增加 1，并认为转移到了下一层界面$u'=(C',I',a')$，l增加 1，重复执行主循环。如果a大于等于给定阈值δ_a，若当前界面为初始化界面u_i，则退出循环，算法结束；否则返回上一级界面，l减去 1。

这样可以得出枚举的界面个数不超过

$$\frac{\delta_a^{\delta_l}-1}{\delta_a-1}$$

步骤 3 的特征在于一个图标-风格-界面树相似度计算方法，当一对图标-风格-界面树之间的相似度大于阈值δ_G时，对应的两个应用被判断为重打包。

所述图标-风格-界面树相似度计算方法是指：将一对图标-风格-界面树转化为多个二部图，通过加权计算多个二部图最大匹配值得到图标-风格-界面树之间的相似度。

所述的图标-风格-界面树转为多个二部图的特征为：设 2 个图标-风格-界面树G_1和G_2，每个图根据层计数可以分为δ_l个集合。设第i层包含的界面集合为G_{1i}和G_{2i}，则二部图点集为$G_{1i}.V \cup G_{2i}.V$，边存在的判定方式为：使用权利要求 3 中计算一组界面的方式计算出相似度η，若η大于一个给定阈值δ_η则判定相同。每层构建一个二部图，可以得到多个二部图组成的集合$\{G_{b1}, G_{b2}, \cdots, G_{b\delta_l}\}$。对每个二部图，可以通过最大匹配计算出相似度：第$i$个二部图的相似度为最大匹配判定相同的界面个数占二部图总界面个数的比例，记为φ_i。树相似度计算公式如下：

$$Sim_g(G_1, G_2) = \frac{\sum_{i=1}^{\delta_l} \varphi_i(|G_{1i}.V| + |G_{2i}.V|)}{\sum_{i=1}^{\delta_l}(|G_{1i}.V| + |G_{2i}.V|)}$$

本发明与现有技术相比，其显著优点在于：现有技术难以对被加密了的应用及做了界面混淆的应用进行有效的检测，使得重打包检测的对象受到很大的限制。而本发明则提供了动态执行安卓应用结合界面视觉特征的策略，提高了对代码混淆与加密这两类反检测方式的抵抗性，降低了对重打包检测对象的限制要求，提高了安卓应用重打包检测的精确性。

附图说明

图 1 为本发明实施例的基于界面图标特征图的安卓应用重打包检测方法的系统框图。

图 2 为本发明实施例的实现框架图。

图 3 为本发明实施例的图生成算法流程图。

图 4 为本发明实施例的图相似度计算流程图。

具体实施方式

说　明　书

下面结合本发明实例中的附图，对本发明实施例中的技术方案进行清楚、完整地的描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部实施例。基于本发明的实施例，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，都属于本发明的保护范围。

1. 主要过程

图 1 所示为本发明提供的一种基于界面图标特征的安卓应用重打包检测方法的系统框图，图 1 中：

对于待比较的两个安卓应用，我们分别动态执行这两个应用，收集并筛选应用运行时的用户界面图标控件位置，并从界面截图中挖取出图标。然后通过风格识别器、树生成器，将界面图标转化为图标-风格-界面树，将该树作为应用的软件胎记。最后，计算两个安卓应用图标-风格-界面树的相似度，用于确定应用是否重打包。

图 2 所示为本发明基于上述系统的一个实现框架图，主要由三个部分组成：安卓系统端、中间代理端和策略执行端。安卓系统端负责在安卓设备上的行为执行和数据获取，行为执行包括安装/卸载应用、启动/结束应用、对当前应用的操作；数据获取包括获取系统会话堆栈、系统窗口堆栈、当前界面的控件信息、系统日志；策略执行端负责筛选与挖取图标，并生成图标-风格-界面树，具体的生成策略参见实施方法 4：树生成算法；中间代理端负责为策略执行端和安卓系统端提供数据交互，该部分通过 Socket 与安卓系统端连接，向安卓系统端发送和接受命令与数据，接受到的数据传递给策略执行端供其进行策略执行。

2. 图标-风格-界面树

安卓应用的图标是指安卓应用界面中抽象化的，用于标示某个功能而使用的小型图片。图标是应用开发者为了提示用户操作当前界面的特定功能而设计的，因此随着界面的跳转会发生改变。

数据模型图标-风格-界面树是一个有向图，$G = (V, E)$。其中，V表示界面图标及特征的集合，E表示界面的转移；

G的节点$v = (I, S, l) \in V$，表示一个特定界面的图标及特征的集合。其中I是图标的集合；S为一个映射$I \to \mathbb{N}$，表示图标的风格；l标示界面的层计数。这个界面视为树的一个节点。而G的边$e \in E \subseteq V \times V$，$e = (v_1, v_2)$意味着存在从界面$v_1$到界面$v_2$的转移。

3. 界面相似度算法

本发明所述的界面相似度算法可用于计算安卓应用的界面相似程度。首先我们获得界面的布局，无论该布局是静态布局 XML 文件中的还是运行时动态获取的布局 XML 数据。我们获取到的布局都是 XML 格式，其数据结构可被看作为一棵树，整棵树代表该布局层次，树的节点代表布局中对应的控件。我们将这棵树以深度优先遍历，遍历每一个叶子节点。这是因为图标控件作为不可划分的控件一定处于叶子节点。获取到了控件的位置信息后，我们进行混淆控件筛选。控件的左上角坐标与右下角坐标框定了控件的边缘，这个边缘上的信息熵与控件中心的信息熵的比值如果大于给定阈值则通过筛选。这是因为控件作为整体，其信息应当集中于中心而不是边缘。一旦控件判定有效，则判定其是否为图标。判定为图标后，从应用界面的截图中直接按照上述坐标挖取出图标，并计算出其风格。

我们首先将两个应用的图标$\{i_{11}, i_{12}, \cdots, i_{1m}\}$和$\{i_{21}, i_{22}, \cdots, i_{2n}\}$按照风格分别分类到不同的集合里$\{\{i_{111}, i_{112}, \cdots, i_{11s}\}, \cdots, \{i_{1k1}, i_{1k2}, \cdots, i_{1kt}\}\}$和$\{\{i_{211}, i_{212}, \cdots, i_{21s'}\}, \cdots, \{i_{2k1}, i_{2k2}, \cdots, i_{2kt'}\}\}$。我们对每对风格相同的集合求最大匹配，最大匹配的结果记为$\text{max_match}(\{i_{1j1}, i_{1j2}, \cdots, i_{1ja}\}, \{i_{1j1}, i_{1j2}, \cdots, i_{2ja'}\})$。其中最大匹配的边存在规则为：对两个图标$i_1$和$i_2$，若存在一个$i$为$i_1$和$i_2$的共同子图，且$i$占$i_1$和$i_2$的比例均大于给定阈值$i_1$和$\delta_i$，则认定边存在。

4. 树生成算法

本发明所述的树生成算法流程图如图 3 所示。该算法为一个有限循环遍历策略，其总体思想为：

数据模型图标-风格-界面树是一个有向图，$G = (V, E)$。其中，V表示界面图标及特征的集合，E表示界面的转移；

G的节点$v = (I, S, l) \in V$，表示一个特定界面的图标及特征的集合。其中I是图标的集合；S为一个映射$I \to \mathbb{N}$，表示图标的风格；l标示界面的层计数。这个界面视为树的一个节点。而G的边$e \in E \subseteq V \times V$，$e = (v_1, v_2)$意味着存在从界面$v_1$到界面$v_2$的转移。

在循环开始前，用户指定循环层深层δ_l与宽度δ_a。计数器l归零，当前选中界面$u = (C, I, a)$置为软件的初始化界面u_i，其中C为所有控件的集合，a为已经在当前界面执行的操作数；

进入主循环后：如果l小于一个给定阈值δ_l，则返回上一级界面，l减去 1。否则首先令a为零，筛选出当前界面u中控件集合C中的有效控件集合$C_v \in C$，根据有效控件集合

C_v选择出有效图标集合$I_v \in I$；如果在现有的图标-风格-界面树中找到了相似的界面，则认定重复，返回上一层界面，l减去 1；否则将有效图标分类出风格后与界面和层数一同加入输出的图标-风格-界面树。

接下来如果a小于给定阈值δ_a，则在C_v中找到任意一个没有选择过的可交互控件c，随机选择一个该控件所支持的行为在控件c上执行操作，a增加 1，并认为转移到了下一层界面$u' = (C', I', a')$，l增加 1，重复执行主循环。如果a大于等于给定阈值δ_a，若当前界面为初始化界面u_i，则退出循环，算法结束；否则返回上一级界面，l减去 1。

5. 树相似度计算

本发明所述的图相似度计算如图 4 流程图所示。我们考虑 2 个图标-风格-界面树G_1和G_2，每个图根据层计数可以分为δ_l个集合。设第i层包含的界面集合为G_{1i}和G_{2i}，则二部图点集为$G_{1i}.V \cup G_{2i}.V$，边存在的判定方式为：使用权利要求 3 中计算一组界面的方式计算出相似度η，若η大于一个给定阈值δ_η则判定相同。每层构建一个二部图，可以得到多个二部图组成的集合$\{G_{b1}, G_{b2}, \cdots, G_{b\delta_l}\}$。对每个二部图，通过最大匹配计算出相似度：第$i$个二部图的相似度为最大匹配判定相同的界面个数占二部图总界面个数的比例，记为φ_i。树相似度计算公式如下：

$$Sim_g(G_1, G_2) = \frac{\sum_{i=1}^{\delta_l} \varphi_i(|G_{1i}.V| + |G_{2i}.V|)}{\sum_{i=1}^{\delta_l}(|G_{1i}.V| + |G_{2i}.V|)}$$

以上的实施例仅为说明本发明的技术思想，不能以此限定本发明的保护范围，凡是按照本发明提出的技术思想，在技术方案基础上所做的任何改动，均落入本发明保护范围之内。本发明未涉及的技术均可通过现有的技术加以实现。

说 明 书 附 图

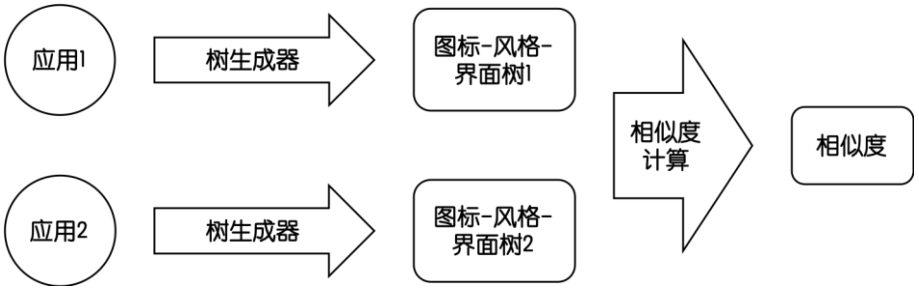

应用1
树生成器
图标-风格-界面树1
应用2
树生成器
图标-风格-界面树2
相似度计算
相似度

图 1

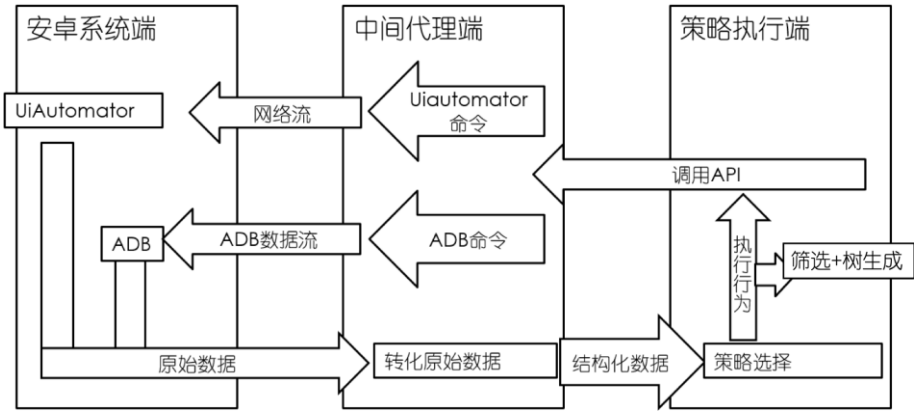

安卓系统端
中间代理端
策略执行端
UiAutomator
网络流
Uiautomator命令
调用API
ADB
ADB数据流
ADB命令
执行行为
筛选+树生成
原始数据
转化原始数据
结构化数据
策略选择

图 2

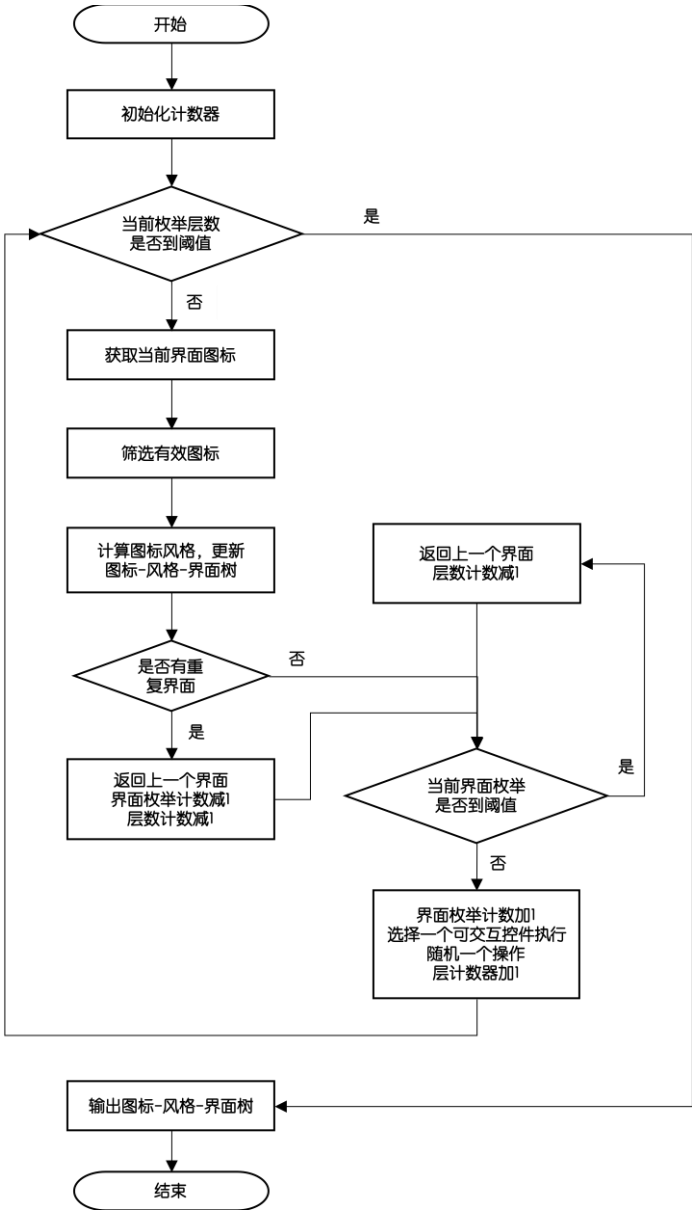

开始
初始化计数器
当前枚举层数
是否到阈值
是
否
获取当前界面图标
筛选有效图标
计算图标风格，更新
图标-风格-界面树
是否有重
复界面
否
是
返回上一个界面
界面枚举计数减1
层数计数减1
返回上一个界面
层数计数减1
当前界面枚举
是否到阈值
是
否
界面枚举计数加1
选择一个可交互控件执行
随机一个操作
层计数器加1
输出图标-风格-界面树
结束

图 3

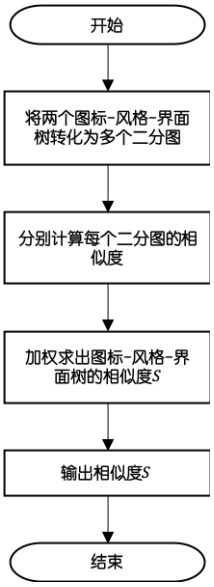

开始
将两个图标-风格-界面树转化为多个二分图
分别计算每个二分图的相似度
加权求出图标-风格-界面树的相似度S
输出相似度S
结束

图 4