



(12)发明专利申请

(10)申请公布号 CN 113220249 A

(43)申请公布日 2021.08.06

(21)申请号 202010080672.3

(22)申请日 2020.02.05

(71)申请人 腾讯科技(深圳)有限公司

地址 518044 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72)发明人 余俊卿

(74)专利代理机构 北京同达信恒知识产权代理
有限公司 11291

代理人 李娟

(51)Int.Cl.

G06F 3/14(2006.01)

G06F 16/174(2019.01)

G06F 16/172(2019.01)

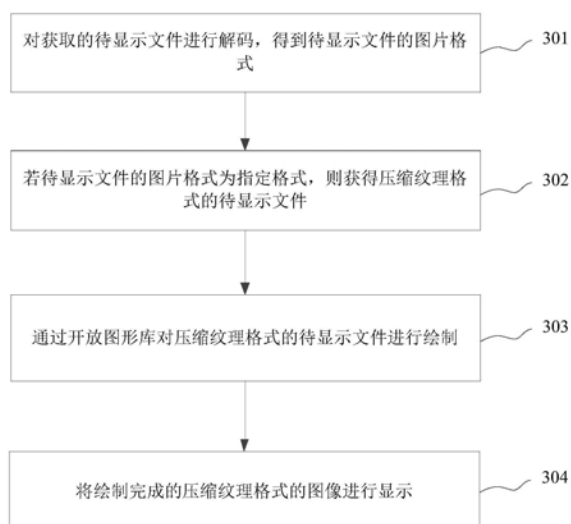
权利要求书2页 说明书9页 附图3页

(54)发明名称

一种图像显示方法、装置及存储介质

(57)摘要

本申请公开了一种图像显示方法、装置及存储介质,涉及计算机技术领域,以减少显示图像时占用的内存。该方法中,对获取的待显示文件进行解码,得到待显示文件的图片格式;若待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;通过开放图形库对压缩纹理格式的待显示文件进行绘制;将绘制完成的压缩纹理格式的图像进行显示。这样,由于压缩文件格式的图像数据较小,从而降低了内存占用。



1. 一种图像显示方法,其特征在于,所述方法包括:

对获取的待显示文件进行解码,得到所述待显示文件的图片格式;

若所述待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;

通过开放图形库对所述压缩纹理格式的待显示文件进行绘制;

将绘制完成的压缩纹理格式的图像进行显示。

2. 根据权利要求1所述的方法,其特征在于,所述获得压缩纹理格式的待显示文件,包括:

若在文件缓存中查找到所述压缩纹理格式的待显示文件,则从所述文件缓存中获取所述压缩纹理格式的待显示文件;

若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件。

3. 根据权利要求2所述的方法,其特征在于,所述若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件之后,所述方法还包括:

将所述压缩纹理格式的待显示文件存储在所述文件缓存中。

4. 根据权利要求2所述的方法,其特征在于,所述若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件的同时,所述方法还包括:

通过开放图形库对所述待显示文件进行绘制,并将绘制完成的所述指定格式的图像进行显示;

所述将绘制完成的压缩纹理格式的图像进行显示,包括:

将绘制完成的压缩纹理格式的图像替换为所述指定格式的图像进行显示。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在java层建立一个绘制方法的类;其中,所述绘制方法为调用开放图形库,使用开放图形库对压缩纹理格式的文件进行绘制;

将所述java层的类的引用传入到native层中构建的类中;

将所述native层的类的内存指针地址回传到java层中的绘制指令;

所述通过开放图形库对所述压缩纹理格式的待显示文件进行绘制,包括:

根据绘制指令中的内存指针地址,确定native层中的类;

根据native层中的所述引用,确定java层中绘制方法的类;

根据所述java层的类的绘制方法调用开放图形库,并使用开放图形库对压缩纹理格式的文件进行绘制。

6. 一种图像显示装置,其特征在于,所述方法包括:

解码模块,用于对获取的待显示文件进行解码,得到所述待显示文件的图片格式;

获得模块,用于若所述待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;

第一绘制模块,用于通过开放图形库对所述压缩纹理格式的待显示文件进行绘制;

显示模块,用于将绘制完成的压缩纹理格式的图像进行显示。

7. 根据权利要求6所述的装置,其特征在于,获得模块包括:

第一获取单元,用于若在文件缓存中查找到所述压缩纹理格式的待显示文件,则从所述文件缓存中获取所述压缩纹理格式文件的待显示文件;

第二获取单元,用于若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件。

8. 根据权利要求7所述的装置,其特征在于,所述装置还包括:

存储模块,用于第二获取单元若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件之后,将所述压缩纹理格式的待显示文件存储在所述文件缓存中。

9. 一种计算机可读介质,存储有计算机可执行指令,其特征在于,所述计算机可执行指令用于执行如权利要求1-5中任一权利要求所述的方法。

10. 一种计算装置,其特征在于,包括:

至少一个处理器;以及与所述至少一个处理器通信连接的存储器;其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行如权利要求1-5中任一权利要求所述的方法。

一种图像显示方法、装置及存储介质

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种图像显示方法、装置及存储介质。

背景技术

[0002] 随着科技的飞速发展,智能终端已成为了生活中必不可少的一部分,而在相关技术中,智能终端的原生应用在对图像进行绘制时,通常通过SDK (Software Development Kit,软件开发工具包)提供的上层接口来生成绘制命令,由于绘制的图像数据较大,从而导致智能终端的内存占用较多。

发明内容

[0003] 本申请实施例提供一种图像显示方法、装置及存储介质,以减少显示图像时占用的内存。

[0004] 第一方面,提供一种图像显示方法,包括:

[0005] 对获取的待显示文件进行解码,得到所述待显示文件的图片格式;

[0006] 若所述待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;

[0007] 通过开放图形库对所述压缩纹理格式的待显示文件进行绘制;

[0008] 将绘制完成的压缩纹理格式的图像进行显示。

[0009] 第二方面,提供一种图像显示装置,包括:

[0010] 解码模块,用于对获取的待显示文件进行解码,得到所述待显示文件的图片格式;

[0011] 获得模块,用于若所述待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;

[0012] 第一绘制模块,用于通过开放图形库对所述压缩纹理格式的待显示文件进行绘制;

[0013] 显示模块,用于将绘制完成的压缩纹理格式的图像进行显示。

[0014] 在一个实施例中,所述装置还包括:

[0015] 第二绘制模块,用于第二获取单元若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件的同时,通过开放图形库对所述待显示文件进行绘制,并将绘制完成的所述指定格式的图像进行显示;

[0016] 显示模块具体用于将绘制完成的压缩纹理格式的图像替换为所述指定格式的图像进行显示。

[0017] 在一个实施例中,所述装置还包括:

[0018] 建立模块,用于建立在java层建立一个绘制方法的类;其中,所述绘制方法为调用开放图形库,使用开放图形库对压缩纹理格式的文件进行绘制;

[0019] 传入模块,用于将所述java层的类的引用传入到native层中构建的类中;

[0020] 回传模块,用于将所述native层的类的内存指针地址回传到java层中的绘制指

令；

[0021] 第一绘制模块,包括:

[0022] 第一确定单元,用于根据绘制指令中的内存指针地址,确定native层中的类;

[0023] 第二确定单元,用于根据native层中的所述引用,确定java层中绘制方法的类;

[0024] 绘制单元,用于根据所述java层的类的绘制方法调用开放图形库,并使用开放图形库对压缩纹理格式的文件进行绘制。

[0025] 第三方面,提供一种计算装置,包括至少一个处理单元、以及至少一个存储单元,其中,存储单元存储有计算机程序,当程序被处理单元执行时,使得处理单元执行上述任意一种图像显示方法的步骤。

[0026] 在一个实施例中,计算装置可以使服务器,也可以是终端设备。

[0027] 第四方面,提供一种计算机可读介质,其存储有可由终端设备执行的计算机程序,当程序在终端设备上运行时,使得终端设备执行上述任意一种图像显示方法的步骤。

[0028] 本申请实施例提供一种图像显示方法、装置及存储介质,在确定待显示文件的压缩纹理格式后,通过开放图形库对压缩纹理格式的待显示文件进行绘制,并将绘制完成的压缩纹理格式的图像进行显示。由于压缩文件格式的图像数据较小,从而降低了内存占用。

[0029] 本申请的其它特征和优点将在随后的说明书中阐述,并且,部分地从说明书中变得显而易见,或者通过实施本申请而了解。本申请的目的和其他优点可通过在所写的说明书、权利要求书、以及附图中所特别指出的结构来实现和获得。

附图说明

[0030] 此处所说明的附图用来提供对本申请的进一步理解,构成本申请的一部分,本申请的示意性实施例及其说明用于解释本申请,并不构成对本申请的不当限定。在附图中:

[0031] 图1为本申请实施例中构建可绘制接口的流程图;

[0032] 图2为本申请实施例中构建绘制指令的示意图;

[0033] 图3为本申请实施例中一种图像显示方法的流程示意图;

[0034] 图4为本申请实施例中一种图像显示装置的结构示意图;

[0035] 图5为本申请实施方式中终端设备结构示意图。

具体实施方式

[0036] 为了使智能终端在显示图像时,减少图像的内存占用,本申请实施例中提供一种图像显示方法、装置及存储介质。为了更好的理解本申请实施例提供的技术方案,这里对该方案的基本原理做一下简单说明:

[0037] 为了便于本领域技术人员更好地理解本申请实施例中的技术方案,下面对本申请实施例涉及到的专业术语进行示例说明。

[0038] 开放图形库 (OpenGL, Open Graphics Library):是一种用于渲染2D、3D矢量图形的跨语言、跨平台的应用程序编程接口 (API, Application Programming Interface)。在本申请实施例中,可以通过开放图形库直接对压缩纹理格式的待显示文件进行绘制。

[0039] 压缩纹理:是一种由硬件支持的纹理格式,将一些纹理图片资源(如png, jpeg等格式的图像)进行纹理压缩后的产物,Gpu可以直接使用压缩纹理进行绘制。常见的压缩纹理

的格式为etc1、etc2等。

[0040] 原生APP (应用程序) : 针对IOS (苹果移动设备操作系统)、Android (一种以Linux与JAVA为基础的开放源代码操作系统, 安卓)、Windows (一种操作系统) 等不同的手机操作系统要采用不同的语言和框架进行开发, 该模式通常是由“云服务器数据+APP应用客户端”两部份构成, APP应用所有的UI元素、数据内容、逻辑框架均安装在手机终端上。在本申请实施例中, 原生APP是指没有自己创建、维护opengl渲染环境的应用。

[0041] 下面对本申请实施例的设计思想进行简要介绍。

[0042] 如前所述, 在相关技术中, 安卓系统的原生APP在绘制图片时, 是通过安卓SDK提供的上层接口来生成绘制命令 (安卓系统的原生APP并不支持直接通过开放图形库对压缩纹理格式的待显示文件进行绘制), 因此绘制的图片数据为普通的图片格式 (如png, jpeg等格式), 从而导致显示的图片所占内存较大。

[0043] 为此, 本申请实施例提供一种图像显示方法、装置及存储介质。在该方法中, 在确定待显示文件的压缩纹理格式后, 通过开放图形库对压缩纹理格式的待显示文件进行绘制, 并将绘制完成的压缩纹理格式的图像进行显示。由于压缩文件格式的图像数据较小, 从而降低了显示图像的内存占用。

[0044] 在本申请实施例中, 若绘制一张1024*1024像素的图片, 使用相关技术的绘制方法需要将该图片加载成RGBA_8888 (每个像素占四位, 即A=8, R=8, G=8, B=8, 那么一个像素点占8+8+8+8=32位) 的位图, 并对该位图进行绘制, 该位图占用的内存为4MB; 而通过本申请提供的图像显示方法, 将1024*1024像素的图片转为压缩纹理 (etc1) 格式的图片, 所占用的内存为512KB。因此, 可以极大减少图片所占用的内存。

[0045] 为便于理解, 下面结合附图对本申请提供的技术方案做进一步说明。

[0046] 在本申请实施例中, 为了实现本申请提供的图像显示方法, 需要基于原生APP的Drawable (可绘制接口), 构建一个特殊的实现类。如图1所示, 其为Drawable的构建流程图。根据图1中的流程即可将待显示文件的格式转换为压缩纹理的格式, 从而对压缩纹理格式的待显示文件进行绘制。

[0047] 在本申请实施例中, 由于相关技术中, 安卓系统的原生APP并不支持直接通过开放图形库对压缩纹理格式的待显示文件进行绘制, 因此, 在构建完Drawable后, 还需要在java层新建一个类, 以使在执行绘制命令时, 根据新建的类来执行通过开放图形库对压缩纹理格式的待显示文件进行绘制的方法。具体可包括以下步骤:

[0048] 步骤A1: 在java层建立一个绘制方法的类; 其中, 绘制方法为调用开放图形库, 使用开放图形库对压缩纹理格式的文件进行绘制。

[0049] 在本申请实施例中, 通过java层的应用框架层 (Framework) 来建立类。java层由Java语言实现。

[0050] 步骤A2: 将java层的类的引用传入到native层中构建的类中。

[0051] 在本申请实施例中, native层为本地框架层, 包括运行环境以及C语言库。native层是通过C和C++语言实现。

[0052] 在本申请实施例中, 在native层中构建类的目的是为了在native层调用的接口 (callDrawGLFunction) 要求传递一个指针函数。

[0053] 而在本申请实施例中, 将java层的类的引用传入到native层中构建的类中, 其目

的是为了简化撰写代码的难度。

[0054] 步骤A3:将native层的类的内存指针地址回传到java层中的绘制指令。

[0055] 如图2所示,其为构建绘制指令的示意图。在图2中,将java层中构建的类1的引用传入到native层构建的类2中,并将native层的类2的内存指针地址回传到java层中的绘制指令中,这样,通过java层与native层之间的联动,可以计算更加高效。

[0056] 除上述方法之外,也可以直接在native层实现扩展绘制指令的方法,本申请对此不做限定。

[0057] 这样在执行绘制指令时,通过构建好的类来实现通过开放图形库对压缩纹理格式的待显示文件进行绘制的方法。具体可实施为:

[0058] 步骤B1:根据绘制指令中的内存指针地址,确定native层中的类。

[0059] 步骤B2:根据native层中的引用,确定java层中绘制方法的类。

[0060] 步骤B3:根据java层的类的绘制方法调用开放图形库,并使用开放图形库对压缩纹理格式的文件进行绘制。

[0061] 在本申请实施例中,不同的安卓系统版本,所调用的绘制指令也不同。例如:若版本是API21,则调用HardwareCanvas#callDrawGLFunction对待显示文件进行绘制;若版本是API22时,则调用HardwareCanvas#callDrawGLFunction2,若版本是API 23时,则调用DisplayListCanvas#callDrawGLFunction2等。

[0062] 这样,通过在java层和native层构建的类,可以使开放图形库可以直接对压缩纹理格式的文件进行绘制,从而提高了绘制效率。且通过上述方法,可以直接使用开放图形库的命令,无需额外的渲染线程,也不会引起开放图形库的上下文切换,而导致的性能损耗,从而节省系统资源。

[0063] 在构建好绘制指令以及Drawable后,便可以根据构建的方法来实现图像显示方法。如图3所示,具体可包括以下步骤:

[0064] 步骤301:对获取的待显示文件进行解码,得到待显示文件的图片格式。

[0065] 在本申请实施例中,获取的待显示文件可以是在服务器上下载的,也可以是在本地中存储的数据文件。

[0066] 在本申请实施例中,图片格式包括png, jpeg等普通格式,也包括etc1、etc2等压缩纹理格式。

[0067] 步骤302:若待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件。

[0068] 在本申请实施例中,指定格式可以是普通图片格式,也可以是压缩纹理格式。根据得到的待显示文件的图片格式,确定压缩纹理格式的待显示文件。

[0069] 在本申请实施例中,步骤301-步骤302是在Drawable中执行的。

[0070] 步骤303:通过开放图形库对压缩纹理格式的待显示文件进行绘制。

[0071] 在本申请实施例中,在得到压缩纹理格式的待显示文件后,通过调用绘制指令对压缩纹理格式的待显示文件进行绘制。这样,便可以直接使用开放图形库对压缩纹理格式的待显示文件进行绘制,从而节省系统资源。

[0072] 步骤304:将绘制完成的压缩纹理格式的图像进行显示。

[0073] 在本申请实施例中,步骤303-步骤304是根据绘制指令执行的。

[0074] 需要说明的是,本申请实施例中提供的方法尤其适用于安卓系统的原生APP。这样,通过对压缩纹理格式的待显示文件进行绘制,可以使显示的图像数据较小,从而降低了内存占用。

[0075] 根据图1所示的内容,可通过两种方法来实现步骤302中的方法:

[0076] 一、若在文件缓存中查找到压缩纹理格式的待显示文件,则从文件缓存中获取压缩纹理格式的待显示文件。

[0077] 在本申请实施例中,在构建Drawable时,会预先将一些压缩纹理格式的文件存储在文件缓存中。在获取了待显示文件后,通过在文件缓存中查找是否存获取的待显示文件,若查找到存在待显示文件,则从文件缓存中获取压缩纹理格式的待显示文件。

[0078] 需要说明的是,在文件缓存中查找存在待显示文件后,还需要进行校验,其目的是为了判断获取的待显示文件是否进行过修改,例如:获取的待显示文件与文件缓存中存储的文件格式不同,或者文件中的数据不同,这些表明获取的待显示文件已进行了修改,因此不能够通过校验,从而无法获取压缩纹理格式的待显示文件。

[0079] 二、若在文件缓存中未查找到压缩纹理格式的待显示文件,则对待显示文件进行压缩,得到压缩纹理格式的待显示文件。

[0080] 在本申请实施例中,若在文件缓存中没有查找到待显示文件,或者查找到的文件校验不通过,那么需要对获取的待显示文件进行纹理压缩,从而获取压缩纹理格式的待显示文件。

[0081] 其中,可以使用NDK(Native Development Kit,一种安卓系统的开发工具包)对待显示文件进行纹理压缩,也可以使用Renderscript(一个安卓系统的组件)对待显示文件进行纹理压缩,对此,本申请不进行限定。

[0082] 在本申请实施例中,通过对获取的待显示文件进行纹理压缩后得到压缩纹理格式的待显示文件,并将获取的压缩纹理格式的待显示文件存储到文件缓存中,具体的:将压缩纹理格式的待显示文件存储在所述文件缓存中。这样,若要再次显示待显示文件时,便可以直接在文件缓存中获取到压缩纹理格式的待显示文件,从而省去了纹理压缩的过程,可以更快的显示压缩纹理格式的图像。

[0083] 在本申请实施例中,为了可以快速的在显示界面上显示图像,避免用户的等待,因此在对待显示文件进行压缩的过程中,可以同时与普通格式的待显示文件进行绘制,具体的:通过开放图形库对待显示文件进行绘制,并将绘制完成的指定格式的图像进行显示。

[0084] 而在等到压缩完成后,再对压缩纹理格式的文件进行绘制,并将绘制好的压缩纹理格式的图像进行替换,具体的:将绘制完成的压缩纹理格式的图像替换为指定格式的图像进行显示。

[0085] 这样,可以在显示页面提前显示图片,提高用户的体验。且等到压缩完成后,将绘制完成的压缩纹理格式的图像替换为指定格式的图像进行显示,可以有效地降低内存占用。

[0086] 基于相同的发明构思,本申请实施例还提供了一种图像显示装置。如图4所示,该装置包括:

[0087] 解码模块401,用于对获取的待显示文件进行解码,得到所述待显示文件的图片格式;

[0088] 获得模块402,用于若所述待显示文件的图片格式为指定格式,则获得压缩纹理格式的待显示文件;

[0089] 第一绘制模块403,用于通过开放图形库对所述压缩纹理格式的待显示文件进行绘制;

[0090] 显示模块404,用于将绘制完成的压缩纹理格式的图像进行显示。

[0091] 在一个实施例中,获得模块402包括:

[0092] 第一获取单元,用于若在文件缓存中查找到所述压缩纹理格式的待显示文件,则从所述文件缓存中获取所述压缩纹理格式文件的待显示文件;

[0093] 第二获取单元,用于若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件。

[0094] 在一个实施例中,所述装置还包括:

[0095] 存储模块,用于第二获取单元若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件之后,将所述压缩纹理格式的待显示文件存储在所述文件缓存中。

[0096] 在一个实施例中,所述装置还包括:

[0097] 第二绘制模块,用于第二获取单元若在文件缓存中未查找到所述压缩纹理格式的待显示文件,则对所述待显示文件进行压缩,得到所述压缩纹理格式的待显示文件的同时,通过开放图形库对所述待显示文件进行绘制,并将绘制完成的所述指定格式的图像进行显示;

[0098] 显示模块具体用于将绘制完成的压缩纹理格式的图像替换为所述指定格式的图像进行显示。

[0099] 在一个实施例中,所述装置还包括:

[0100] 建立模块,用于建立在java层建立一个绘制方法的类;其中,所述绘制方法为调用开放图形库,使用开放图形库对压缩纹理格式的文件进行绘制;

[0101] 传入模块,用于将所述java层的类的引用传入到native层中构建的类中;

[0102] 回传模块,用于将所述native层的类的内存指针地址回传到java层中的绘制指令;

[0103] 第一绘制模块403,包括:

[0104] 第一确定单元,用于根据绘制指令中的内存指针地址,确定native层中的类;

[0105] 第二确定单元,用于根据native层中的所述引用,确定java层中绘制方法的类;

[0106] 绘制单元,用于根据所述java层的类的绘制方法调用开放图形库,并使用开放图形库对压缩纹理格式的文件进行绘制。

[0107] 基于同一技术构思,本申请实施例还提供了一种终端设备500,参照图5所示,终端设备500用于实施上述各个方法实施例记载的方法,例如实施图2所示的实施例,终端设备500可以包括存储器501、处理器502、输入单元503和显示面板504。

[0108] 存储器501,用于存储处理器502执行的计算机程序。存储器501可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序等;存储数据区可存储根据终端设备500的使用所创建的数据等。处理器502,可以是一个中央处理单元(central processing unit,CPU),或者为数字处理单元等。输入单元503,可以用

于获取用户输入的用户指令。显示面板504,用于显示由用户输入的信息或提供给用户的信息,本申请实施例中,显示面板504主要用于显示终端设备中各应用程序的显示界面以及各显示界面中显示的控件实体。可选的,显示面板504可以采用液晶显示器(liquid crystal display,LCD)或OLED(organic light-emitting diode,有机发光二极管)等形式来配置显示面板504。

[0109] 本申请实施例中不限定上述存储器501、处理器502、输入单元503和显示面板504之间的具体连接介质。本申请实施例在图5中以存储器501、处理器502、输入单元503、显示面板504之间通过总线505连接,总线505在图5中以粗线表示,其它部件之间的连接方式,仅是进行示意性说明,并不引以为限。总线505可以分为地址总线、数据总线、控制总线等。为便于表示,图5中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0110] 存储器501可以是易失性存储器(volatile memory),例如随机存取存储器(random-access memory,RAM);存储器501也可以是非易失性存储器(non-volatile memory),例如只读存储器,快闪存储器(flash memory),硬盘(hard disk drive,HDD)或固态硬盘(solid-state drive,SSD)、或者存储器501是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。存储器501可以是上述存储器的组合。

[0111] 处理器502,用于实现如图2所示的实施例,包括:

[0112] 处理器502,用于调用存储器501中存储的计算机程序执行如实施图2所示的实施例。

[0113] 本申请实施例还提供了一种计算机可读存储介质,存储为执行上述处理器所需执行的计算机可执行指令,其包含用于执行上述处理器所需执行的程序。

[0114] 在一些可能的实施方式中,本申请提供的一种图像显示方法的各个方面还可以实现为一种程序产品的形式,其包括程序代码,当程序产品在终端设备上运行时,程序代码用于使终端设备执行本说明书上述描述的根据本申请各种示例性实施方式的一种图像显示方法中的步骤。例如,终端设备可以执行如实施图3所示的实施例。

[0115] 程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0116] 本申请的实施方式的用于一种图像显示程序产品可以采用便携式紧凑盘只读存储器(CD-ROM)并包括程序代码,并可以在计算设备上运行。然而,本申请的程序产品不限于此,在本文件中,可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0117] 可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了可读程序代码。这种传播的数据信号可以采用多种形式,包括——但不限于——电磁信号、光信号或上述的任意合适的组合。可读信号介质还可以是可读存储介质以外的任何可读介质,该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者

与其结合使用的程序。

[0118] 可读介质上包含的程序代码可以用任何适当的介质传输,包括——但不限于——无线、有线、光缆、RF等,或者上述的任意合适的组合。

[0119] 可以以一种或多种程序设计语言的任意组合来编写用于执行本申请操作的程序代码,程序设计语言包括面向实体的程序设计语言——诸如Java、C++等,还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中,远程计算设备可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算设备,或者,可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0120] 应当注意,尽管在上文详细描述中提及了装置的若干单元或子单元,但是这种划分仅仅是示例性的并非强制性的。实际上,根据本申请的实施方式,上文描述的两个或更多单元的特征和功能可以在一个单元中具体化。反之,上文描述的一个单元的特征和功能可以进一步划分为由多个单元来具体化。

[0121] 此外,尽管在附图中以特定顺序描述了本申请方法的操作,但是,这并非要求或者暗示必须按照该特定顺序来执行这些操作,或是必须执行全部所示的操作才能实现期望的结果。附加地或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,和/或将一个步骤分解为多个步骤执行。

[0122] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0123] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程图像显示设备的处理器以产生一个机器,使得通过计算机或其他可编程图像显示设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0124] 这些计算机程序指令也可存储在能引导计算机或其他可编程图像显示设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0125] 这些计算机程序指令也可装载到计算机或其他可编程图像显示设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0126] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造

性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0127] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

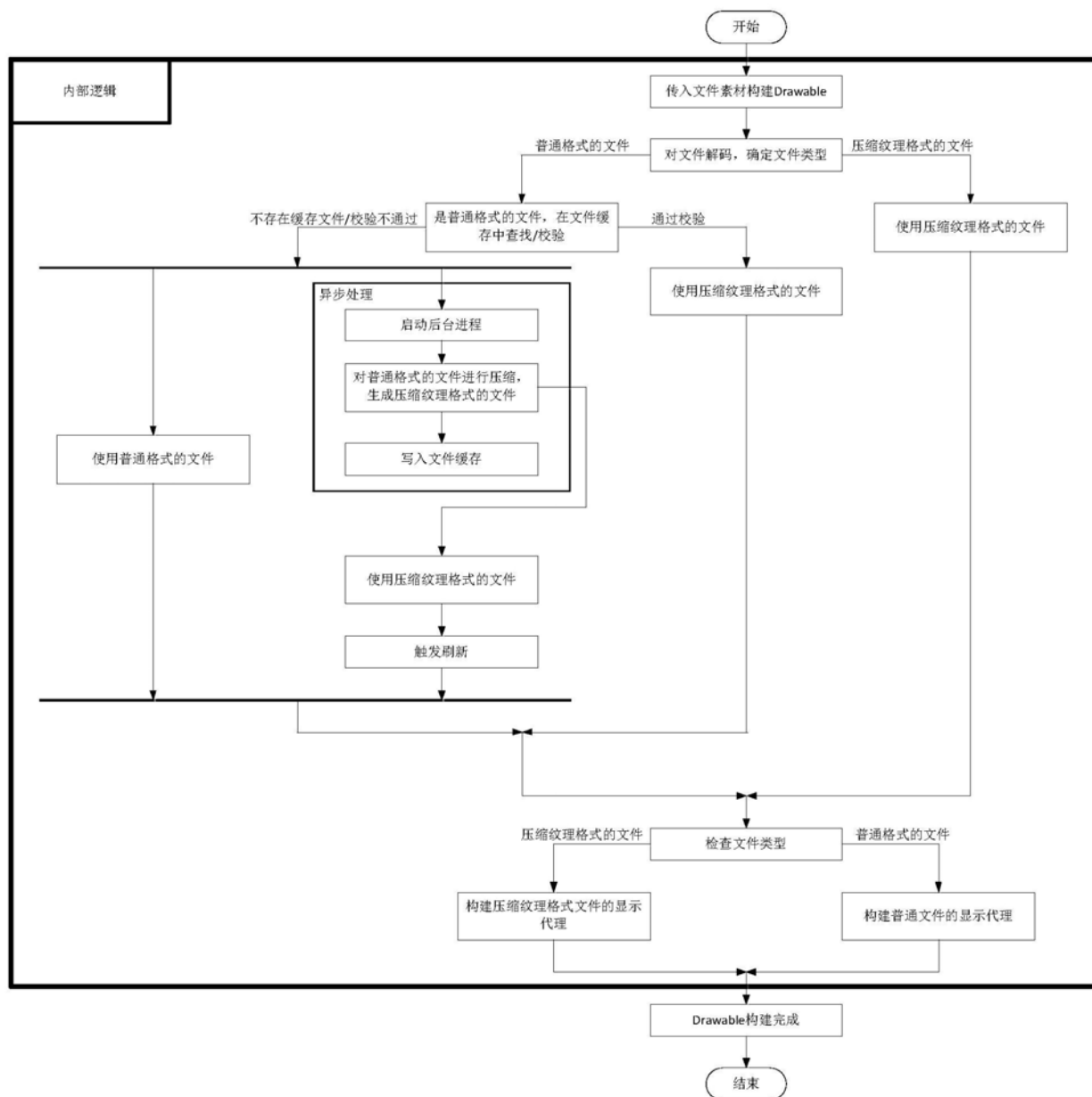


图1

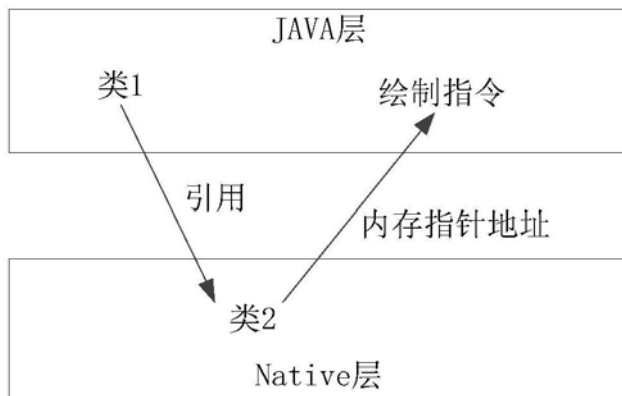


图2

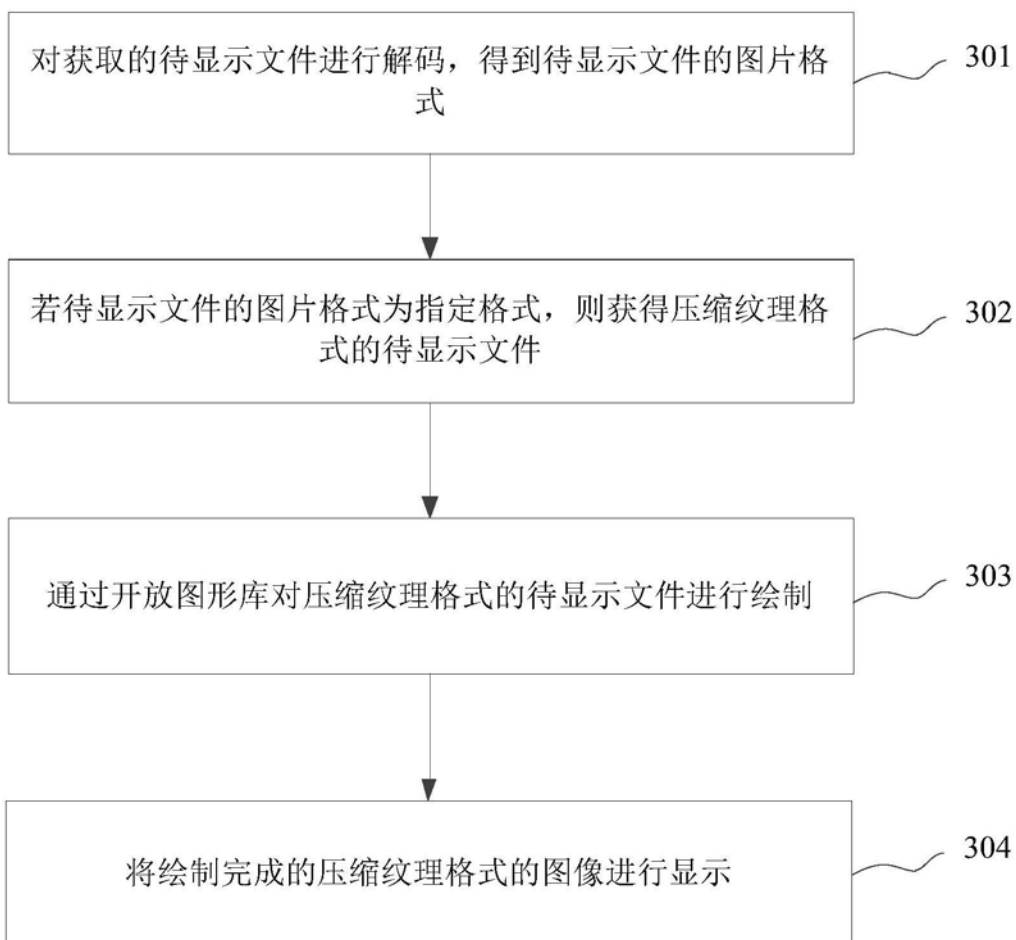


图3

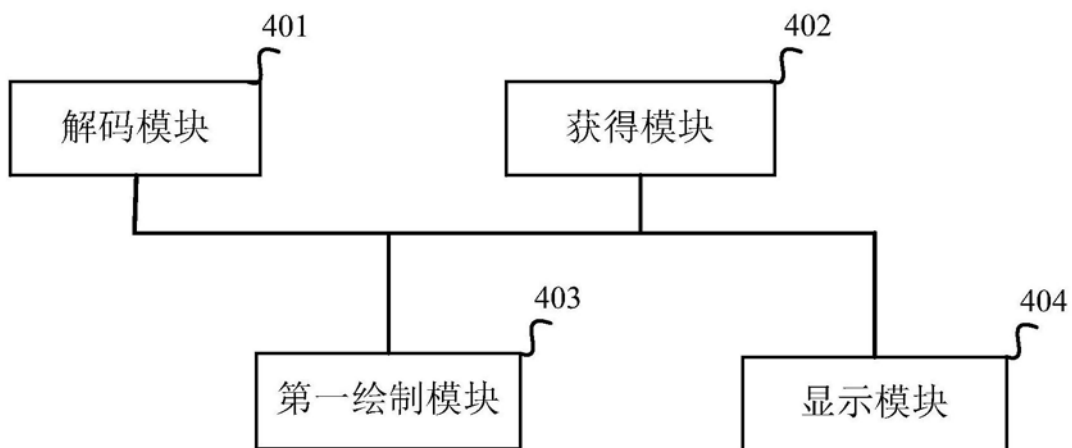


图4

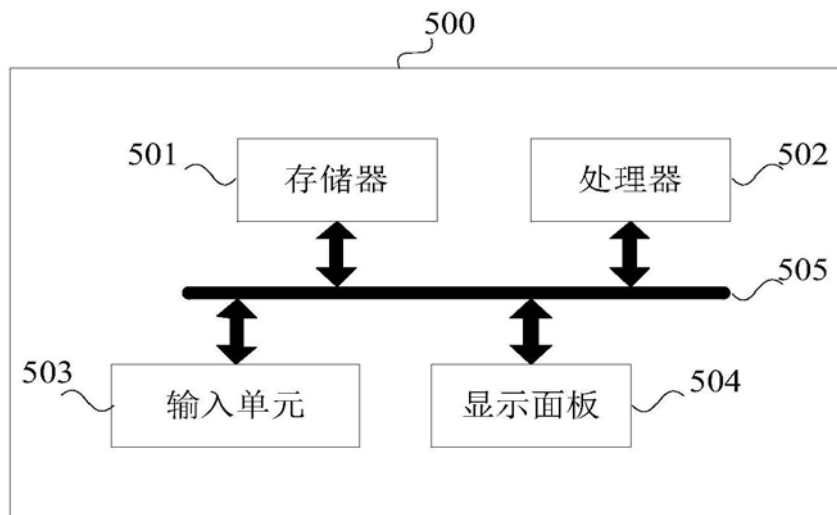


图5