



(12) 发明专利申请

(10) 申请公布号 CN 105678680 A

(43) 申请公布日 2016. 06. 15

(21) 申请号 201511030578. 2

(22) 申请日 2015. 12. 30

(71) 申请人 魅族科技(中国)有限公司

地址 519080 广东省珠海市香洲区科技创新
海岸魅族科技楼

(72) 发明人 朱和泉 崔英云 黄涛

(74) 专利代理机构 广州三环专利代理有限公司
44202

代理人 郝传鑫 熊永强

(51) Int. Cl.

G06T 1/20(2006. 01)

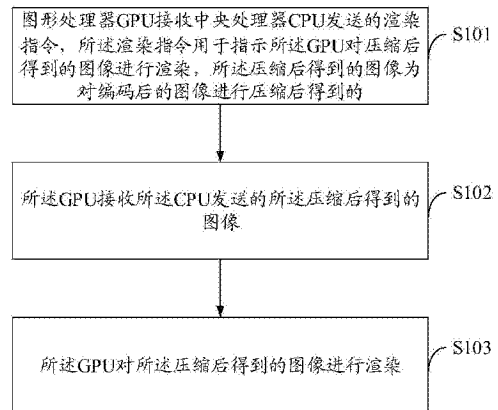
权利要求书2页 说明书10页 附图5页

(54) 发明名称

一种图像处理的方法和装置

(57) 摘要

本发明实施例提供了一种图像处理的方法和装置,其中,所述方法包括:图形处理器 GPU 接收中央处理器 CPU 发送的渲染指令,所述渲染指令用于指示所述 GPU 对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;所述 GPU 接收所述 CPU 发送的所述压缩后得到的图像;所述 GPU 对所述压缩后得到的图像进行渲染。采用本发明,可节省终端存储图像的内存空间,提升图像处理的效率以及终端的运行性能。



1. 一种图像处理的方法,其特征在于,包括:

图形处理器GPU接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

所述GPU接收所述CPU发送的所述压缩后得到的图像;

所述GPU对所述压缩后得到的图像进行渲染。

2. 根据权利要求1所述的方法,其特征在于,所述GPU对所述压缩后得到的图像进行渲染,包括:

所述GPU调用函数callDrawGLFunction对所述压缩后得到的图像进行渲染。

3. 根据权利要求1或2所述的方法,其特征在于,所述GPU对所述压缩后得到的图像进行渲染,包括:

所述GPU将所述压缩后得到的图像作为纹理图像进行渲染。

4. 根据权利要求1-3中任一项所述的方法,其特征在于,所述GPU接收所述CPU发送的所述压缩后得到的图像,包括:

所述GPU接收所述CPU通过函数glCompressedTexImage2D发送的所述压缩后得到的图像。

5. 根据权利要求1-4中任一项所述的方法,其特征在于,在所述GPU接收所述CPU发送的所述压缩后得到的图像之后,所述方法还包括:

所述GPU将所述压缩后得到的图像保存在所述GPU创建的内存空间中;

则,所述GPU对所述压缩后得到的图像进行渲染,包括:

所述GPU对所述内存空间中保存的压缩后得到的图像进行渲染。

6. 一种图像处理的方法,其特征在于,包括:

中央处理器CPU向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

所述CPU向所述GPU发送所述压缩后得到的图像。

7. 根据权利要求6所述的方法,其特征在于,所述方法还包括:

所述GPU对所述压缩后得到的图像进行渲染。

8. 一种图像处理的装置,其特征在于,所述装置包括:

第一接收模块,用于接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

第二接收模块,用于接收所述CPU发送的所述压缩后得到的图像;

处理模块,用于对所述压缩后得到的图像进行渲染。

9. 根据权利要求8所述的装置,其特征在于,

所述处理模块,具体用于调用函数callDrawGLFunction对所述压缩后得到的图像进行渲染。

10. 根据权利要求8或9所述的装置,其特征在于,

所述处理模块,具体用于将所述压缩后得到的图像作为纹理图像进行渲染。

11. 根据权利要求8-10中任意一项所述的装置,其特征在于,

所述第二接收模块,具体用于接收所述CPU通过函数glCompressedTexImage2D发送的所述压缩后得到的图像。

12.根据权利要求8-11中任意一项所述的装置,其特征在于,所述装置还包括:
保存模块,用于将所述压缩后得到的图像保存在所述GPU创建的内存空间中;
所述处理模块,具体用于对所述内存空间中保存的压缩后得到的图像进行渲染。

13.一种图像处理的装置,其特征在于,所述装置包括:

第一发送模块,用于向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

第二发送模块,用于向所述GPU发送所述压缩后得到的图像。

14.如权利要求14所述的装置,其特征在于,所述装置还包括:
渲染模块,用于对所述压缩后得到的图像进行渲染。

一种图像处理的方法和装置

技术领域

[0001] 本发明涉及图像处理技术领域,尤其涉及一种图像处理的方法和装置。

背景技术

[0002] 目前,很多应用程序App为达到华丽、精美的视觉效果,会使用大量的图像。现有的图像渲染技术大多是对各类图像格式(如PNG、JPG等)的图像进行解码处理得到对应的位图图像,然后所述终端的中央处理器(Central Processing Unit,简称CPU)再将位图图像发送给渲染引擎进行渲染。

[0003] 在实践中发现,存储位图图像所占用的内存空间比较大,一个像素的位图占用4个字节的内存,对于1080P的手机一张全屏的位图就需要占用8M内存,对终端的内存和带宽的开销都很大,从而影响终端在进行图像处理过程中的处理效率。

发明内容

[0004] 本发明实施例所要解决的技术问题在于,提供一种图像处理的方法和装置,能够节省终端的内存和解码时间,提升图像处理的效率和终端的运行性能。

[0005] 第一方面,本发明实施例提供了一种图像处理的方法,所述方法包括:

[0006] 图形处理器GPU接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0007] 所述GPU接收所述CPU发送的所述压缩后得到的图像;

[0008] 所述GPU对所述压缩后得到的图像进行渲染。

[0009] 其中可选地,所述GPU对所述压缩后得到的图像进行渲染,包括:

[0010] 所述GPU调用函数callDrawGLFunction对所述压缩后得到的图像进行渲染。

[0011] 其中可选地,所述GPU对所述压缩后得到的图像进行渲染,包括:

[0012] 所述GPU将所述压缩后得到的图像作为纹理图像进行渲染。

[0013] 其中可选地,所述GPU接收所述CPU发送的所述压缩后得到的图像,包括:

[0014] 所述GPU接收所述CPU通过函数glCompressedTexImage2D发送的所述压缩后得到的图像。

[0015] 其中可选地,在所述GPU接收所述CPU发送的所述压缩后得到的图像之后,所述方法还包括:

[0016] 所述GPU将所述压缩后得到的图像保存在所述GPU创建的内存空间中;

[0017] 则,所述GPU对所述压缩后得到的图像进行渲染,包括:

[0018] 所述GPU对所述内存空间中保存的压缩后得到的图像进行渲染。

[0019] 第二方面,本发明实施例提供了另一种图像处理的方法,所述方法包括:

[0020] 中央处理器CPU向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到

的；

[0021] 所述CPU向所述GPU发送所述压缩后得到的图像。

[0022] 其中可选地,所述方法还包括:

[0023] 所述GPU对所述压缩后得到的图像进行渲染。

[0024] 第三方面,本发明实施例提供了一种图像处理的装置,所述装置包括:

[0025] 第一接收模块,用于接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0026] 第二接收模块,用于接收所述CPU发送的所述压缩后得到的图像;

[0027] 处理模块,用于对所述压缩后得到的图像进行渲染。

[0028] 其中可选地,

[0029] 所述处理模块,具体用于调用函数callDrawGLFunction对所述压缩后得到的图像进行渲染。

[0030] 其中可选地,

[0031] 所述处理模块,具体用于将所述压缩后得到的图像作为纹理图像进行渲染。

[0032] 其中可选地,

[0033] 所述第二接收模块,具体用于接收所述CPU通过函数glCompressedTexImage2D发送的所述压缩后得到的图像。

[0034] 其中可选地,所述装置还包括:

[0035] 保存模块,用于将所述压缩后得到的图像保存在所述GPU创建的内存空间中;

[0036] 所述处理模块,具体用于对所述内存空间中保存的压缩后得到的图像进行渲染。

[0037] 第四方面,本发明实施例提供了另一种图像处理的装置,所述装置包括:

[0038] 第一发送模块,用于向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0039] 第二发送模块,用于向所述GPU发送所述压缩后得到的图像。

[0040] 其中可选地,所述装置还包括:

[0041] 渲染模块,用于对所述压缩后得到的图像进行渲染。

[0042] 本发明实施例中,图形处理器GPU可通过接收中央处理器CPU发送的压缩后得到的图像,并对所述压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以调用并保存对编码后的图像进行压缩后得到的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端存储图像的存储空间,同时还提升了图像处理的效率以及终端的运行性能。

附图说明

[0043] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以

根据这些附图获得其他的附图。

- [0044] 图1是本发明实施例的一种图像处理方法的流程示意图；
- [0045] 图2是本发明实施例的另一种图像处理方法的流程示意图；
- [0046] 图3是本发明实施例的另一种图像处理方法的流程示意图；
- [0047] 图4是本发明实施例的另一种图像处理方法的流程示意图；
- [0048] 图5是本发明实施例的一种图像处理装置的结构示意图；
- [0049] 图6是本发明实施例的另一种图像处理装置的结构示意图；
- [0050] 图7是本发明实施例的另一种图像处理装置的结构示意图；
- [0051] 图8是本发明实施例的另一种图像处理装置的结构示意图。

具体实施方式

[0052] 本发明实施例提供了一种图像处理的方法和装置,以期可以节省图像渲染过程中的所占用的内存空间,同时提高图像渲染效率。

[0053] 为了使本技术领域的人员更好地理解本发明方案,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,所描述的实施例仅仅是本发明一部分的实施例,而不是全部的实施例。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前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0054] 本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”和“第三”等是用于区别不同对象,而非用于描述特定顺序。此外,术语“包括”以及它们任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有局限于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0055] 本发明实施例提供一种图像处理的方法,包括:

[0056] 图形处理器GPU接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0057] 所述GPU接收所述CPU发送的所述压缩后得到的图像;

[0058] 所述GPU对所述压缩后得到的图像进行渲染。

[0059] 具体的,请参见图1,图1是本发明实施例的一种图像处理方法的流程示意图,本发明实施例的所述方法可以包括如下步骤。

[0060] S101、图形处理器GPU接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。本发明实施例中,当中央处理器(Central Processing Unit,简称CPU)检测到需要进行图像处理时,CPU可以加载需要进行渲染处理的各种格式(如PNG、JPG、GIF等格式)的图像,所述中央处理器CPU还可以通过纹理压缩工具对加载的所述图像进行特定压缩格式(如ASTC、PVRTC、ETC1、ATITC、S3TC等)的压缩,得到压缩后的图像,CPU可以将所述压缩后得到的图像存储至本CPU自己创建的一个或多个CPU内存空间中。

[0061] 需要说明的是,为了节省压缩时间,本发明可以预先对需要渲染的图像进行处理,

如原本应用会将便携式网络图片(Portable Network Graphics,简称PNG)格式的图形打包到应用中,为了节约运行时的压缩时间,本方案将PNG等格式文件用压缩工具压缩成特定格式,如ETC,ETC2,ASTC,MALI官网提供的Mali Texture Compression Tool。然后将压缩后的文件替换原来的图形文件,比如将原本的xxx.png替换成压缩后的xxx.etc,并随着应用程序的其它资源一起打包。

[0062] 当终端需要运行应用程序时,由CPU调用应用程序包文件(application package file,简称APK),并加载该apk中压缩的图像文件,将该图像文件保存在CPU的内存空间,然后将内存空间中保存的压缩的图像文件发送给图形处理器(Graphic Processing Unit,简称GPU),具体实现中,CPU可以通过调用函数glCompressedTexImage2D将所述压缩后得到的图像发送给GPU。

[0063] CPU还可以向GPU发送渲染指令,其中,所述渲染指令用于指示所述图形处理器GPU对所述压缩后得到的图像进行渲染,其中,所述压缩后得到的图像可以为本CPU加载的对图像进行特定格式压缩之后得到的图形文件(如上述的XXX.ete图像);GPU接收所述中央处理器CPU发送的所述渲染指令。

[0064] 当然CPU可以在向GPU发送压缩后得到的图像之前发送所述渲染指令,也可以在向GPU发送压缩有得到的图像之后发送所述渲染指令。S102、所述GPU接收所述CPU发送的所述压缩后得到的图像。

[0065] 本发明实施例中,GPU可以解析S101中接收到的所述渲染指令,得到所述压缩之后得到的图像;或者,GPU可以接收S101中CPU发送的所述压缩后得到的图像。

[0066] GPU还可以自己创建一个或多个GPU内存空间,GPU可以将接收到的或者解析到的所述压缩后得到的图像存储至本GPU创建或分配的内存空间中,以便GPU对GPU创建或分配的内存空间中保存的所述压缩后得到的图像进行渲染。

[0067] S103、所述GPU对所述压缩后得到的图像进行渲染。

[0068] 本发明实施例中,GPU可以对S102中解析或接收到的所述压缩后得到的图像进行渲染,GPU还可以将S102中解析或接收到的所述压缩后得到的图像作为纹理图像进行渲染;GPU通过调用函数callDrawGLFunction对所述压缩后得到的图像进行渲染。

[0069] 安卓Android渲染架构HWUI提供了一种callDrawGLFunction函数方法,其中,该callDrawGLFunction方法为专门针对网络视图WebView设计使用的,WebView会在主渲染线程通过浏览器渲染引擎渲染自己内容,而浏览器渲染引擎通过callDrawGLFunction这个方法可以执行自定义的渲染命令,将内容显示在屏幕上。即基于callDrawGLFunction实现了一套不依赖系统的渲染架构,能让所有的View都像WebView一样可以在主渲染线程执行自定义的渲染命令来渲染自己的内容。callDrawGLFunction

[0070] 具体实现中,CPU所述压缩后得到的图像为纹理图像,并将所述纹理图像和渲染位置信息发送给GPU,以便GPU根据所述渲染位置信息对所述纹理图像进行渲染,并将纹理图像显示在显示屏上。

[0071] 本发明实施例中,图形处理器GPU可通过接收中央处理器CPU发送的压缩后得到的图像,并对所述压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以调用并保存对编码后的图像进行压缩后得到的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所

述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端存储图像的存储空间,同时还提升了图像处理的效率以及终端的运行性能。

[0072] 请参见图2,是本发明实施例的另一种图像处理方法的流程示意图,本发明实施例的所述方法还包括如下步骤。

[0073] S201、图形处理器GPU接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。

[0074] S202、GPU接收所述CPU通过glCompressedTexImage2D发送的所述压缩后得到的图像。

[0075] 示例性地,如下给出有关glCompressedTexImage2D函数的结构示意图。

[0076] void glCompressedTexImage2D(GLenum target,

[0077] GLint level,

[0078] GLenum internalformat,

[0079] GLsizei width,

[0080] GLsizei height,

[0081] GLint border,

[0082] GLsizei imageSize,

[0083] const GLvoid*data);

[0084] 本发明可以调用上述glCompressedTexImage2D函数来传输发送所述压缩后得到的图像,glCompressedTexImage2D指明了一个二维的压缩纹理图像,其中,上述的target指定活动纹理单元的目标纹理,level指定细节级别数,internalformat指定压缩图形数据的存储格式,即为所述压缩后得到的图像对应的特定压缩格式(如ASTC、PVRTC、ETC1、ATITC、S3TC等),width和height分别指所述压缩后图像的宽度、高度,上述的border为指定边框宽度,在OpenGL ES中border为0;上述的imageSize为所述压缩后得到的图像的大小,一般单位为字节byte。CPU可以通过调用glCompressedTexImage2D将包括宽度、高度、大小等信息的所述压缩后得到的图像发送给GPU,GPU接收CPU通过调用glCompressedTexImage2D函数发送的所述压缩后得到的图像。

[0085] S203、GPU创建一个或多个内存空间。

[0086] 本发明实施例中,图形处理器GPU可以在安装有本图形处理器GPU的终端中,或者在本图形处理器GPU所控制的终端中创建一个或多个内存空间;或者,所述图形处理器GPU还可以在本图形处理器GPU所占用的内存资源中创建/分配一个或多个内存空间,以便所述图形处理器所创建/分配的内存空间用于存储所述压缩后得到的图像。

[0087] S204、GPU将所述压缩后得到的图像保存在GPU创建的内存空间中。

[0088] S204、GPU将所述内存空间中保存的压缩后得到的图像作为纹理图像,并调用函数callDrawGLFunction对所述纹理图像进行渲染。

[0089] 需要说明的是,callDrawGLFunction可以抽象为:Android应用程序渲染引擎(HWUI)提供给应用层的执行应用层自定义渲染命令的接口;

[0090] callDrawGLFunction的声明为:public int callDrawGLFunction(long drawGLFunction),其中,drawGLFunction指向的是一个函数适配器地址,这个地址可以传

给HWUI,由HWUI去执行。

[0091] callDrawGLFunction的具体使用方法:

[0092] 1、首先在C++代码定义一个GLFunction,比如:

```
class GLFunctor:public Functor{
public:
    GLFunctor(){};
[0093]     ~GLFunctor(){};
    int operator()(int what,void*data){//do something render};
};
```

[0094] 2、在java代码通过native调用创建Functor并将返回的函数地址保存,比如为drawGLFunction,

```
static jlong GLFunctor_create(JNIEnv*env,object this,object weak_this)
{
[0095]     GLFunctor *functor=new GLFunctor();
    return(jlong) functor;
};
```

[0096] 3、调用canvas的callDrawGLFunction(drawGLFunction),HWUI就是执行上面定义的函数:int operator()(int what,void*data){//do something render}。

[0097] 在本发明实施例中引入callDrawGLFunction函数实现渲染,可以让应用在主渲染线程执行自定义的OpenGL代码,只有主渲染线程才能把内容渲染到屏幕,在主渲染线程执行OpenGL代码就能不进行额外的拷贝和渲染开销就能把内容渲染到屏幕,如果不使用callDrawGLFunction的话,就需要开设一个额外的渲染线程,在这个线程用OpenGL把内容渲染到一个临时framebuffer,然后把这个framebuffer内容拷贝出来,再传给主线程渲染到屏幕,所以多了一次拷贝和把framebuffer渲染到屏幕的开销,这是极其影响性能的。

[0098] 本发明实施例中,图形处理器GPU可通过接收中央处理器CPU发送的压缩后得到的图像,并对所述压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以调用并保存对编码后的图像进行压缩后得到的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端存储图像的存储空间,同时还提升了图像处理的效率以及终端的运行性能。

[0099] 请参见图3,是本发明实施例的另一种图像处理方法的流程示意图,本发明实施例的所述方法还包括如下步骤。

[0100] S301、中央处理器CPU向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。

[0101] S302、所述CPU向所述GPU发送所述压缩后得到的图像。

[0102] 本发明实施例中,当中央处理器CPU(Central Processing Unit,简称CPU)检测到图像处理事件时,所述中央处理器CPU可以加载需要进行渲染处理的各种图像格式(如PNG、JPG、GIF等格式)的图像,所述中央处理器CPU还可以通过纹理压缩工具(如Mali官网提供的Mali_Texture_Compression_Tool)对加载的所述图像进行特定压缩格式(如ASTC、PVRTC、ETC1、ATITC、S3TC等)的压缩,得到压缩后得到的图像(如将XXX.png图像压缩成XXX.ete图像),所述中央处理器CPU可以将所述压缩后得到的图像存储至本中央处理器CPU自己创建的一个或多个CPU内存空间中。所述中央处理器CPU还可以将所述压缩后得到的图像和应用的资源(如应用中存在的所有或者部分图像)一起打包发送给图形处理器GPU。

[0103] 具体实现中,所述中央处理器CPU可以通过调用glCompressedTexImage2D函数将所述压缩后得到的图像发送给所述图形处理器GPU。

[0104] 所述GPU、所述CPU可以为应用在诸如智能手机、平板电脑、智能可穿戴设备等带通信网络功能的终端上的处理器,所述通信终端可以包括个人电脑、智能手机(如Android手机、IOS手机等)、平板电脑、掌上电脑、移动互联网设备(MID, Mobile Internet Devices)或穿戴式智能设备等互联网设备,本发明实施例不作限定。

[0105] 本发明实施例中,图形处理器GPU可通过接收中央处理器CPU发送的压缩后得到的图像,并对所述压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以对编码后的图像进行压缩得到压缩后的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端的存储空间,同时还提升了图像处理的效率以及终端的运行效能。

[0106] 请参阅图4,是本发明实施例的另一种图像处理方法的流程示意图,本发明实施例的所述方法可以应用在诸如智能手机、平板电脑、智能可穿戴设备等带通信网络功能的终端中,具体可由这些通信终端的处理器来实现。本发明实施例的所述方法包括如上所述的S301、S302步骤,还包括如下步骤。

[0107] S303、所述GPU对所述压缩后得到的图像进行渲染。

[0108] 本发明实施例中,终端的图像处理器GPU可以接收本终端或者其他终端的中央处理器CPU在上述S302步骤中通过调用glCompressedTexImage2D函数发送的所述压缩后得到的图像,所述图像处理器GPU还可以将接收到的所述压缩后得到的图像存储至本图形处理器GPU创建的内存空间中,继而与所述压缩后得到的图像进行渲染。

[0109] 具体实现中,所述图形处理器GPU可以将接收到的所述压缩后得到的图像作为纹理图像,并通过调用callDrawGLFunction函数对所述纹理图像进行渲染。

[0110] 本发明实施例中,终端的图形处理器GPU可通过接收本终端或者其他终端的中央处理器CPU发送的压缩后得到的图像,并对所述压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以对编码后的图像进行压缩得到压缩后的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端的存储空间,同时还提升了终端对图像处理的效率以及终端的运行性能。

[0111] 本发明实施例还提供一种图像处理的装置,包括:

[0112] 第一接收模块,用于接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0113] 第二接收模块,用于接收所述CPU发送的所述压缩后得到的图像;

[0114] 处理模块,用于对所述压缩后得到的图像进行渲染。

[0115] 具体的,请参见图5,是本发明实施例的一种图像处理装置的结构示意图,所述图形处理的装置5包括:

[0116] 第一接收模块50,用于接收中央处理器CPU发送的渲染指令,所述渲染指令用于指示GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0117] 第二接收模块51,用于接收所述CPU发送的所述压缩后得到的图像;

[0118] 处理模块52,用于对所述压缩后得到的图像进行渲染。

[0119] 本发明实施例中涉及各个模块、单元的具体实现可参考图1至图4对象实施例中相关功能模块的描述,在此不赘述。

[0120] 本发明实施例的图像处理的装置中的,第一接收模块和第二接收模块分别接收CPU发送的渲染指令和压缩后得到的图像,由处理模块对压缩后得到的图像进行渲染;其中,所述压缩后得到的图像为对编码后的图像进行压缩后得到的。中央处理器CPU可以调用并保存对编码后的图像进行压缩后得到的图像,并将所述压缩后的图像发送给图形处理器处理模块进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端存储图像的存储空间,同时还提升了图像处理的效率以及终端的运行性能。

[0121] 请参见图6,是本发明实施例的另一种图像处理装置的结构示意图,所述图像处理的装置6包括上述的第一接收模块50、第二接收模块51、处理模块52,

[0122] 处理模块52,具体用于调用函数`callDrawGLFunction`对所述压缩后得到的图像进行渲染。

[0123] 其中可选地,在本发明实施例中,

[0124] 处理模块52,具体用于将所述压缩后得到的图像作为纹理图像进行渲染。

[0125] 其中可选地,在本发明实施例中,

[0126] 第二接收模块51,具体用于接收所述CPU通过调用函数`glCompressedTexImage2D`发送的所述压缩后得到的图像。

[0127] 其中可选地,在本发明实施例中,所述装置还包括:

[0128] 保存模块53,用于将所述压缩后得到的图像保存在所述GPU创建的内存空间中;

[0129] 所述处理模块52,具体用于对所述内存空间中保存的所述压缩后得到的图像进行渲染。

[0130] 本发明实施例中涉及各个模块、单元的具体实现可参考图1至图4对象实施例中相关功能模块的描述,在此不赘述。

[0131] 请参见图7,是本发明实施例的另一种图像处理装置的结构示意图,所述图像处理的装置7包括:

[0132] 第一发送模块70,用于向图形处理器GPU发送渲染指令,所述渲染指令用于指示所述GPU对压缩后得到的图像进行渲染,所述压缩后得到的图像为对编码后的图像进行压缩后得到的;

[0133] 第二发送模块71,用于向所述GPU发送所述压缩后得到的图像。

[0134] 本发明实施例中涉及各个模块、单元的具体实现可参考图1至图4对象实施例中相关功能模块的描述,在此不赘述。

[0135] 本发明实施例中,中央处理器CPU可以对编码后的图像进行压缩得到压缩后的图像,并将所述压缩后的图像发送给图形处理器GPU进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端的存储空间,还可提升终端对图像处理的效率。

[0136] 请参见图8,是本发明实施例的另一种图像处理装置的结构示意图,本发明实施例的所述装置可以设置在诸如智能手机、平板电脑、智能可穿戴设备等带通信网络功能的终端中,图像处理的装置8包括上述的第一发送模块70、第二发送模块71,还可以包括:

[0137] 渲染模块72,用于对所述压缩后得到的图像进行渲染。

[0138] 本发明实施例中涉及各个模块、单元的具体实现可参考图1至图4对象实施例中相关功能模块的描述,在此不赘述。

[0139] 本发明实施例中,由第一发送模块和第二发送模块分别向GPU发送渲染指令和压缩后得到的图像,然后由渲染模块对压缩后的图像进行渲染,避免了中央处理器CPU对所述编码后的图像进行解码,这样既节省终端对所述编码后的图像进行解码的解码时间,又节省终端的存储空间,还可提升终端对图像处理的效率以及终端的运行性能。

[0140] 本发明实施例还提供一种计算机存储介质,其中,该计算机存储介质可存储有程序,该程序执行时包括上述方法实施例中记载的任何音频播放应用的操作方法的部分或全部步骤。

[0141] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本发明并不受所描述的动作顺序的限制,因为依据本发明,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本发明所必须的。

[0142] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述的部分,可以参见其他实施例的相关描述。

[0143] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置,可通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性或其它的形式。

[0144] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目

的。

[0145] 另外,在本发明的各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0146] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0147] 以上所述,以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

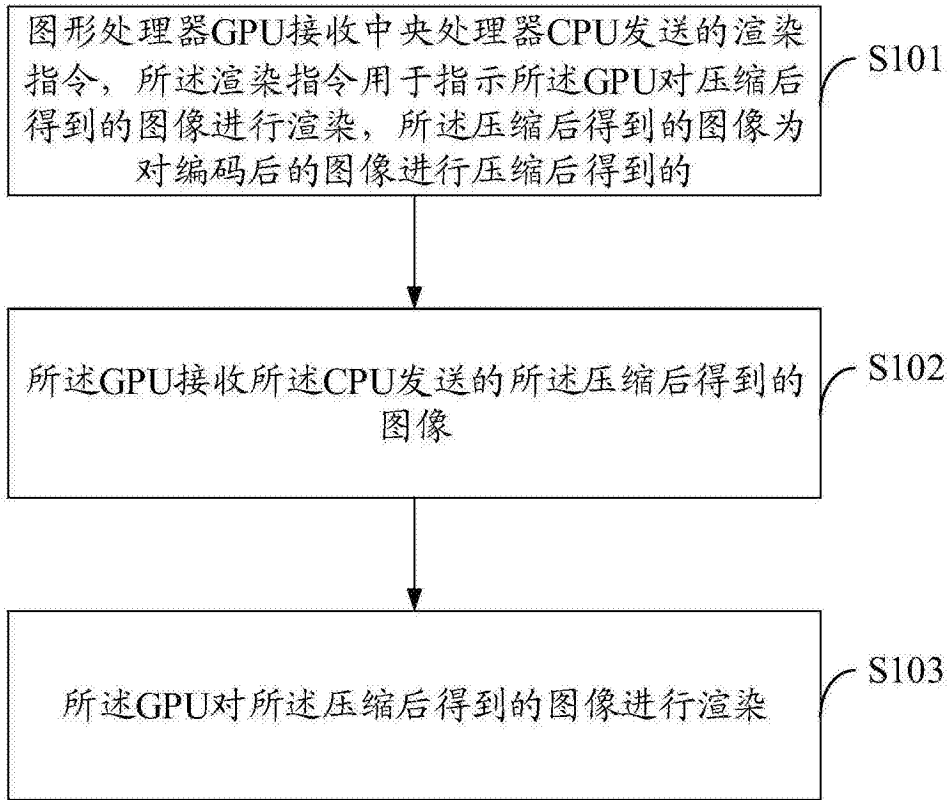


图1

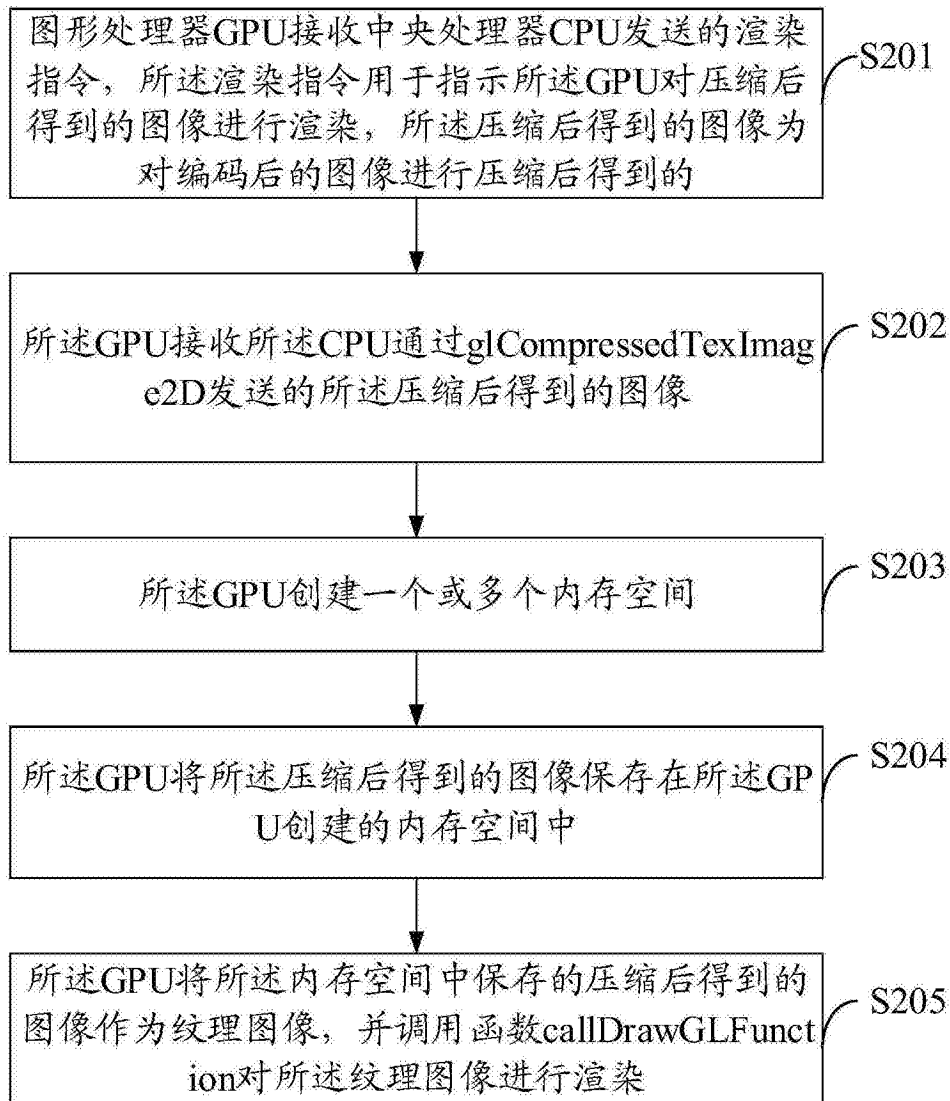


图2

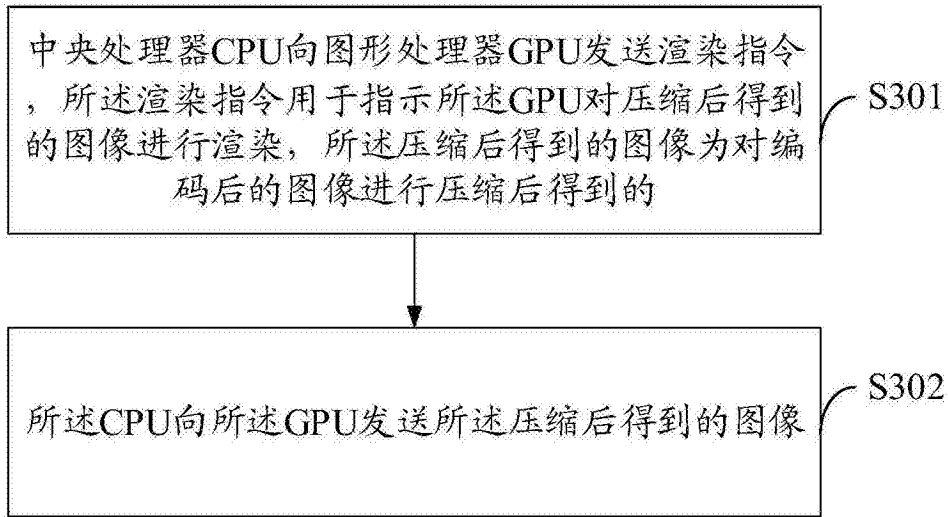


图3

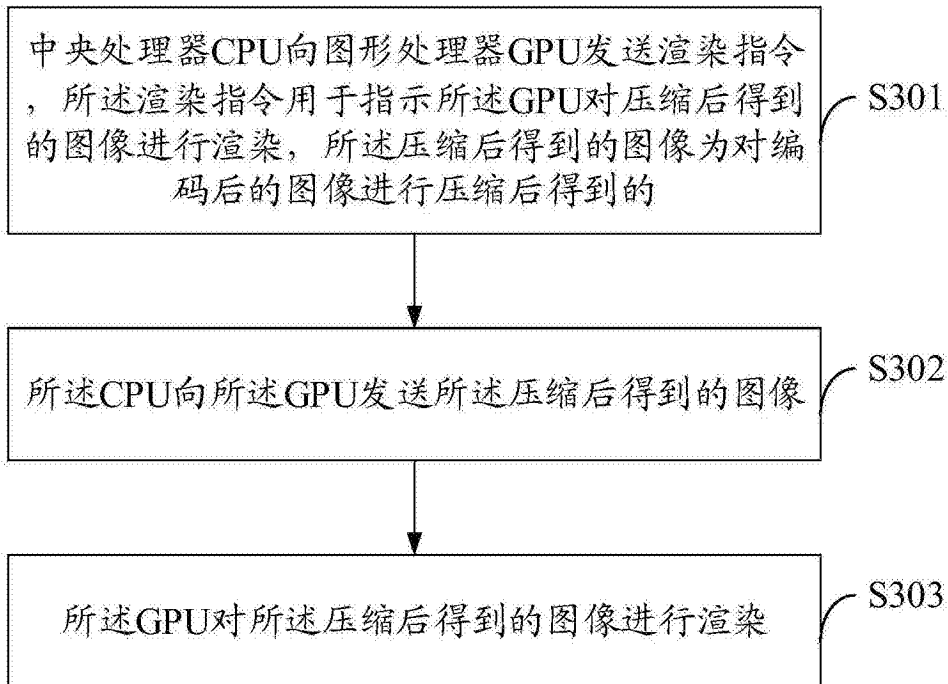


图4

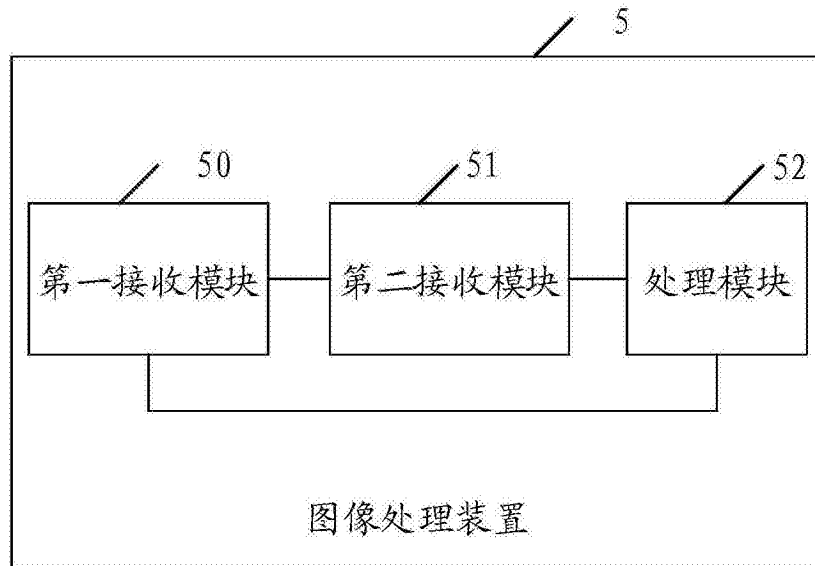


图5

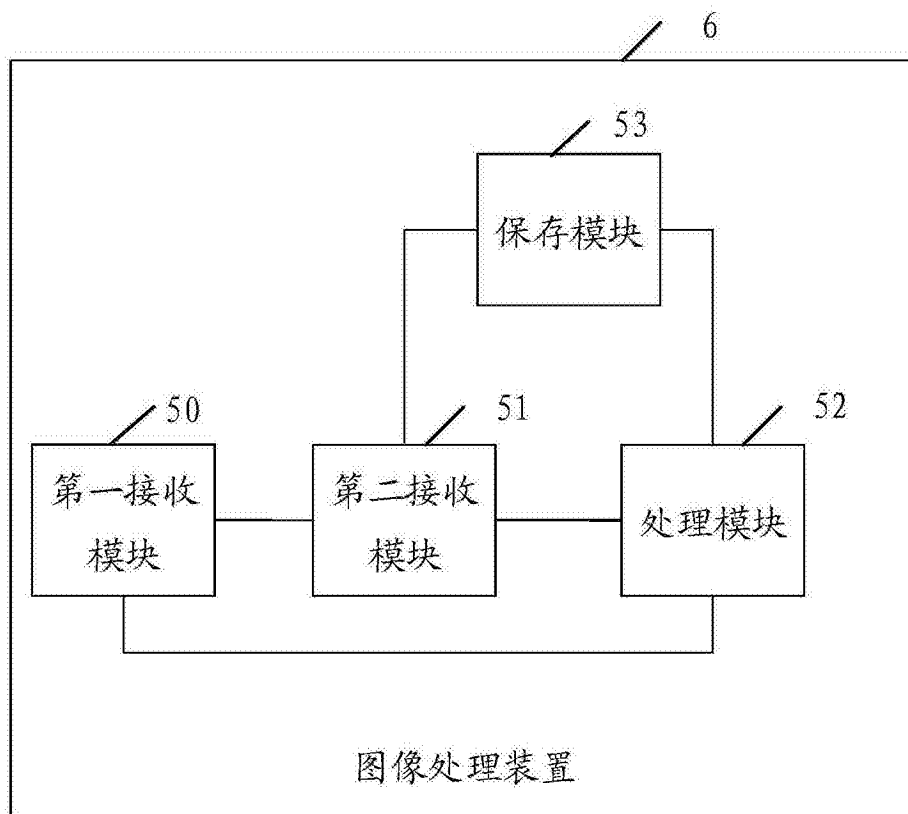


图6

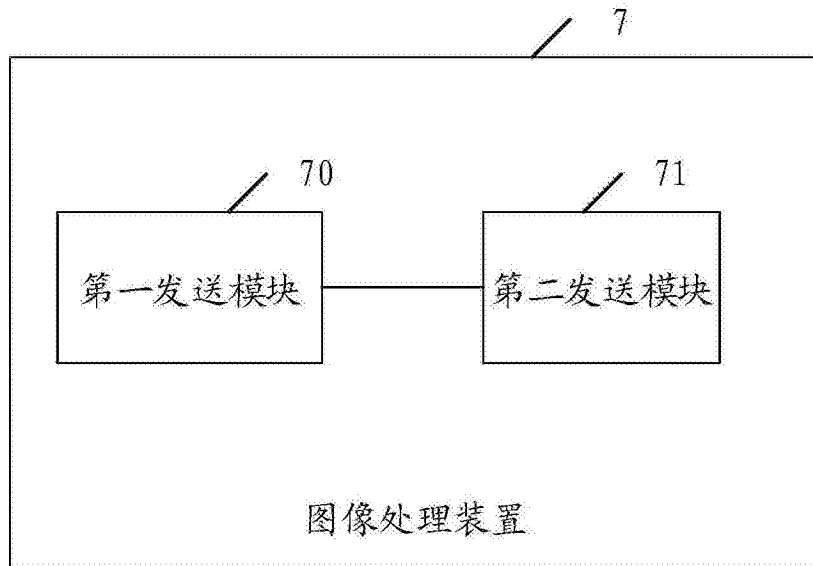


图7

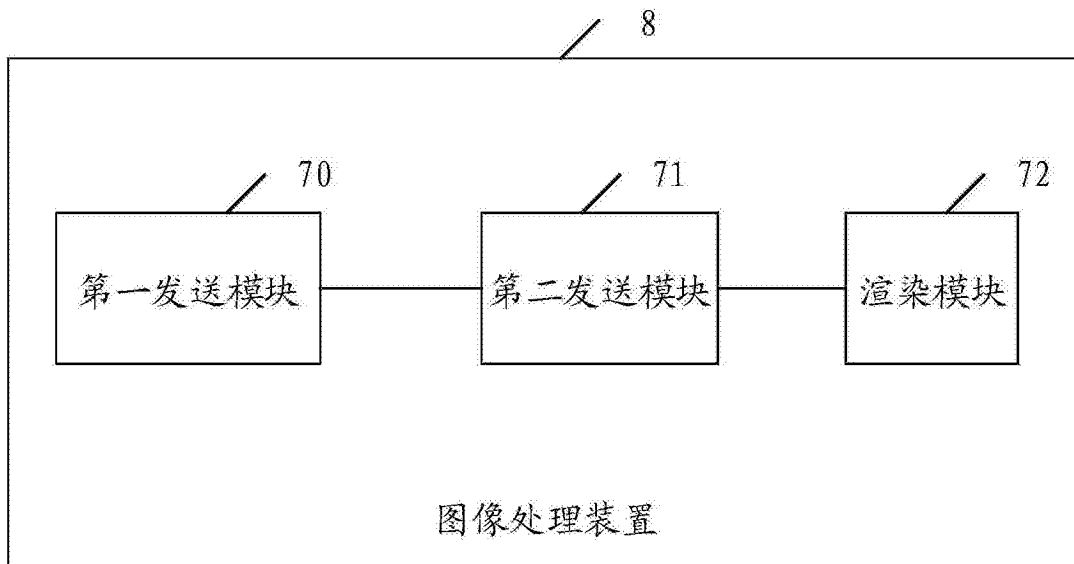


图8