



## (12)发明专利申请

(10)申请公布号 CN 110012336 A

(43)申请公布日 2019.07.12

(21)申请号 201910191143.8

(22)申请日 2019.03.12

(71)申请人 广州虎牙信息科技有限公司

地址 510000 广东省广州市番禺区南村镇  
万博二路79号万博商务区万达商业广  
场北区B-1栋28层10单元

(72)发明人 陈俊城

(74)专利代理机构 深圳市威世博知识产权代理  
事务所(普通合伙) 44280

代理人 李庆波

(51)Int.Cl.

H04N 21/431(2011.01)

H04N 21/44(2011.01)

H04N 21/4402(2011.01)

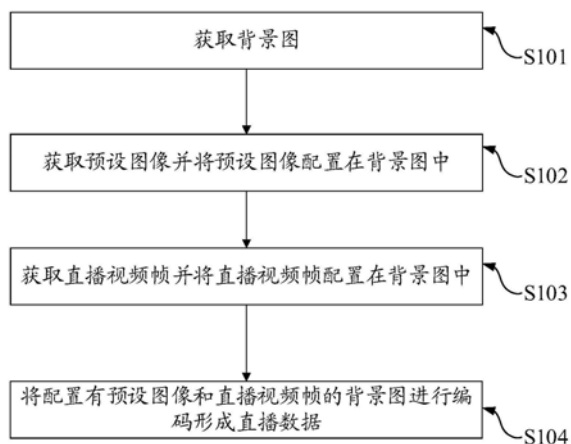
权利要求书2页 说明书9页 附图6页

### (54)发明名称

直播界面的图片配置方法、终端及装置

### (57)摘要

本申请公开了直播界面的图片配置方法、终端及装置。其中所述方法包括：获取背景图；获取预设图像并将预设图像配置在背景图中；获取直播视频帧并将直播视频帧配置在背景图中；将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。通过上述方式，本申请能够实现直播视频和图片的交互。



1. 一种直播界面的图片配置方法,其特征在于,包括:  
获取背景图;  
获取预设图像并将所述预设图像配置在所述背景图中;  
获取直播视频帧并将所述直播视频帧配置在所述背景图中;  
将配置有所述预设图像和所述直播视频帧的所述背景图进行编码形成直播数据。
2. 根据权利要求1所述的图片配置方法,其特征在于:所述获取背景图包括:  
渲染出纯色图,将所述纯色图的RGB数值进行位图格式转换得到所述背景图。
3. 根据权利要求2所述的图片配置方法,其特征在于:所述获取预设图像并将所述预设图像配置在所述背景图中包括:  
将所述预设图像进行格式转换得到所述预设图像的位图数据;  
将所述预设图像的位图数据渲染至所述背景图中。
4. 根据权利要求3所述的图片配置方法,其特征在于:所述将所述预设图像的位图数据渲染至所述背景图中包括:  
将所述预设图像的位图数据拷贝至所述背景图的内存空间内以进行渲染。
5. 根据权利要求4所述的图片配置方法,其特征在于:获取直播视频帧并将所述直播视频帧配置在所述背景图中包括:  
通过预设SDK工具包获取所述直播视频帧并返回所述直播视频帧的位图数据;  
将所述直播视频帧的位图数据渲染至所述背景图中。
6. 根据权利要求5所述的图片配置方法,其特征在于:所述将所述直播视频帧的位图数据渲染至所述背景图中包括:  
将所述直播视频帧的位图数据拷贝至所述背景图对应的内存空间内以进行渲染。
7. 根据权利要求6所述的图片配置方法,其特征在于:将所述直播视频帧的内存空间内的位图数据拷贝至所述背景图的内存空间内以进行渲染包括:  
获取所述预设图像的起始内存地址和所述背景图的起始内存地址;  
将所述预设图像的位图数据所述预设图像的起始内存地址开始依次拷贝至所述背景图对应的从所述背景图的起始内存地址开始的内存空间内,在拷贝完毕时得到配置有所述预设图像的第一背景图。
8. 根据权利要求7所述的图片配置方法,其特征在于:所述将所述直播视频帧的位图数据拷贝至所述背景图对应的内存地址中以进行渲染包括:  
获取所述直播视频帧的起始内存地址和所述第一背景图的起始内存地址;  
将所述直播视频帧的位图数据从所述直播视频帧的起始内存地址依次进拷贝至所述第一背景图对应的从所述第一背景图的起始内存地址开始的内存空间内,在拷贝完毕时获得配置有所述预设图像和所述直播视频帧的第二背景图。
9. 根据权利要求8所述的图片配置方法,其特征在于:所述第二背景图的终止内存地址为所述背景图的终止内存地址。
10. 根据权利要求5所述的图片配置方法,其特征在于:所述将所述直播视频帧的位图数据渲染至所述背景图中包括:  
将所述直播视频帧的位图数据加入待渲染队列;  
从所述待渲染队列中依次将所述直播视频帧的位图数据取出,并将所述直播视频帧的

位图数据拷贝至所述背景图对应的内存空间内以进行渲染。

11. 根据权利要求1所述的图片配置方法, 其特征在于: 所述获取预设图像并将所述预设图像配置在所述背景图中包括:

获取直播设备的相册;

响应选择指令从所述相册中获取所述预设图像。

12. 一种移动终端, 其特征在于, 包括处理器、通信电路和显示器, 所述处理器通过通信总线分别耦接所述通信电路和显示器;

所述处理器用于获取背景图;

所述处理器用于获取预设图像并将所述预设图像配置在所述背景图中;

所述处理器用于获取直播视频帧并将所述直播视频帧配置在所述背景图中;

所述处理器用于将配置有所述预设图像和所述直播视频帧的所述背景图进行编码形成直播数据。

13. 一种具有存储功能的装置, 其特征在于, 存储有程序数据, 所述程序数据能够被执行以实现如权利要求1-11任一项所述的方法。

## 直播界面的图片配置方法、终端及装置

### 技术领域

[0001] 本申请涉及直播技术领域,特别是涉及直播界面的图片配置方法、终端及装置。

### 背景技术

[0002] 随着互联网技术的发展和智能设备的应用发展,直播平台具有多元化的直播内容,例如在线娱乐或者游戏直播。特别智能移动终端已经是直播的主战场,在直播市场占有率极大,甚至超越了PC端的直播,这种趋势在未来仍旧是延续。

[0003] 但是目前在移动终端上进行直播的直播技术中,所呈现的直播界面只是单纯的视频直播,导致直播界面的功能单一,趣味性不强。

### 发明内容

[0004] 本申请主要解决的技术问题是提供直播界面的图片配置方法、终端及装置,能够实现直播视频和图像显示的交互。

[0005] 为解决上述技术问题,本申请采用的一个技术方案是:提供一种直播界面的图片配置方法,包括:

[0006] 获取背景图;

[0007] 获取预设图像并将预设图像配置在背景图中;

[0008] 获取直播视频帧并将直播视频帧配置在背景图中;

[0009] 将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0010] 为解决上述技术问题,本申请采用的另一个技术方案是:提供一种移动终端,该移动终端包括处理器、通信电路和显示器,处理器通过通信总线分别耦接通信电路和显示器;其中,

[0011] 处理器用于获取背景图;

[0012] 处理器用于获取预设图像并将预设图像配置在背景图中;

[0013] 处理器用于获取直播视频帧并将直播视频帧配置在背景图中;

[0014] 处理器用于将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0015] 为解决上述技术问题,本申请采用的另一个技术方案是:提供一种具有存储功能的装置,该装置存储有程序数据,程序数据能够被执行以实现如本申请直播界面的图片配置方法的步骤。

[0016] 与现有技术相比,本申请的有益效果是:本申请以背景图作为配置的基础,将预设图像和直播视频帧配置在背景图中,并将配置有预设图像和直播视频帧的背景图进行编码形成直播数据,直播数据可以推流至服务器,该直播数据对应的直播界面能够同时显示预设图像和直播视频帧,从而实现了图片和直播视频同时显示的交互,如此能够在直播界面上提供图片展示的窗口,从而提高直播的灵活性以及丰富直播体验。

## 附图说明

- [0017] 图1是本申请直播界面的图片配置方法第一实施例中直播系统的结构示意图；
- [0018] 图2是本申请直播界面的图片配置方法第一实施例的流程示意图；
- [0019] 图3是本申请直播界面的图片配置方法第一实施例的过程示意图；
- [0020] 图4是本申请直播界面的图片配置方法第二实施例的流程示意图；
- [0021] 图5是本申请直播界面的图片配置方法第三实施例的流程示意图；
- [0022] 图6是本申请直播界面的图片配置方法第三实施例的过程示意图；
- [0023] 图7是本申请移动终端第一实施例的电路示意框图；
- [0024] 图8是本申请移动终端第二实施例的电路示意框图；
- [0025] 图9是本申请具有存储功能的装置实施例的结构示意框图。

## 具体实施方式

[0026] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅是本申请的一部分实施例,而不是全部的实施例。基于本申请中的实施例,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,都属于本申请保护的范围。

[0027] 本申请的发明人经过长期研究发现,移动终端作为直播终端进行直播时,能够产生直播视频供观众进行实时观看,而图片是处于静态的,在现有的直播技术上,两者是难以进行结合,导致在直播时也仅能单纯地观看到视频直播界面,直接界面单调,功能单一,不存在将图片等素材与视频直播界面进行交互配合的技术,导致直播的交互效果较差。本申请为了解决上述技术问题,提出以下实施例。

[0028] 参阅图1,本申请直播界面的图片配置方法第一实施例,本实施例的技术方案可以应用在直播系统1中,直播系统1包括服务器10,多个直播设备20。例如直播设备20运行有直播应用程序或者APP(简称直播程序),直播设备20可以运行IOS操作系统,也可以运行Android操作系统,当然还可以是其他移动操作系统。如图2所示,可以通过直播设备20的直播程序进行直播或者观看直播。作为主播端的直播设备20可以进行直播活动将直播数据以数据流的形式推流至服务器10上,以便于作为观众端的直播设备20进行获取观看。在本实施例中,运行IOS操作系统的直播设备比如Ipad,Iphone,Itouch,ipod等,运行Android操作系统的直播设备20在此不进行举例。

[0029] 参阅图2和图3,本申请直播界面的图片配置方法第一实施例包括:

[0030] S101:获取背景图。

[0031] 在本实施例中,获取背景图可以是直播设备自行渲染生成而获取到的,也可以是从互联网或者其他设备处接收得到的。例如获取的背景图是纯黑色的,纯黑色像素值为0,作为配置的基础,以便于后续对预设图像和视频帧的配置。当然获取的背景图也可以是纯白色的,或者其他纯色。如图3所示的背景图中并未示出颜色。

[0032] 在一些实施方式中,例如直播设备渲染出纯黑色图,将纯黑色图的RGB数值进行位图格式转换得到包含位图数据的背景图。在本实施例中,位图数据可以称为一个像素数组,数组中的每个像素就代表着图片中的一个点。背景图的图像可以认为是二维的像素数组。

[0033] S102:获取预设图像并将预设图像配置在背景图中。

[0034] 预设图像可以是直播设备接收到选择指令(比如语音指令或者触摸指令)后选定的,当然也可以由直播设备自动选定的,比如根据直播设备自行生成的或者预置的。

[0035] 在一个具体的应用场景中,直播程序在直播设备上启动,自动读取直播设备的相册的图片,接收到选择指令(例如用户通过点击屏幕形成选择指令)选定选择指令对应的图片,将被选定的图片作为预设图像。

[0036] 直播设备在获取到预设图像后,在后台将预设图像配置在背景图中。例如将预设图像的像素替换背景图中的部分像素,从而使得背景图的一部分位置显示预设图像的图案,实现将预设图像配置在背景图中。

[0037] S103:获取直播视频帧并将直播视频帧配置在背景图中。

[0038] 在本实施例中,直播视频帧例如是通过直播设备的摄像头去拍摄而形成的,或者是通过录屏方式录制形成的。

[0039] 例如在获取到直播视频帧后,将直播视频帧的各帧以时间序列的方式依次各自配置在背景图中,配置完成的背景图也是可以以帧的方式编码合成视频。例如直播视频帧每形成一帧,可以将直播视频帧配置在背景图中,配置完成后的背景图形成的帧数与直播视频帧的帧数可以一一对应。在S102中背景图中已配置完预设图像,即在已配置完预设图像的背景图中进一步配置直播视频帧,从而在已配置预设图像的背景图中进一步显示有直播视频帧。

[0040] 当然,在其他实施例中,S102和S103可以调换顺序,先在背景图中配置直播视频帧,再配置预设图像。

[0041] S104:将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0042] 将配置有预设图像和直播视频帧的背景图进行编码,例如通过H264进行编码,形成直播数据。直播数据所对应的直播界面包含有预设图像和直播视频帧,也即在直播界面中同时显示了预设图像和直播视频帧的播放,如此在直播界面实现了预设图像和直播视频帧的交互。在本实施例中,作为主播端的直播设备,完成预设图像和直播视频帧的配置后进行编码形成的直播数据,可以数据流的形式进行推流至服务器,从而观众在其观看设备上可以看到直播界面,所观看到的直播界面同时显示了预设图像和直播视频,而直播视频是一直处于播放状态,与预设图像相对独立显示。

[0043] 本实施例通过以背景图作为配置的基础,先将预设图像和直播视频帧配置在背景图中,并将配置有预设图像和直播视频帧的背景图进行编码形成待推流的直播数据,从而能够进一步将直播数据推流至服务器,即能够实现直播数据对应的直播界面能够同时显示预设图像和直播视频帧,从而实现了图片和直播视频同时显示的交互,解决了现有直播技术中难以在直播时的同一直播界面实现图片和直播视频同时显示的问题,如此能够在直播界面上提供图片展示的途径,丰富了直播界面的功能,从而提高直播的灵活性以及丰富直播体验。

[0044] 在一个具体的应用场景中,主播在进行游戏直播时,可以将游戏的精彩瞬间截图保存在相册中。可以将某些精彩瞬间的截图设置为预设图像。直播设备将游戏直播的视频与预设图像配置在直播界面中,从而观众能够观看到主播进行游戏过程的精彩瞬间,特别是对于错过精彩瞬间的观众可以在直播界面上看到精彩瞬间,如此可以弥补错过的遗憾。本实施例能够实现图片和直播视频同时显示的交互,从而提高直播体验和观看体验。

[0045] 在另一个具体的应用场景中,主播在通过摄像头进行现场视频直播时,比如美妆主播的直播,可以设置美妆主播各种可爱的自拍图像或者各种妆容的自拍图像作为预设图像,预设图像和直播视频能够同时显示在直播界面上,观众可以知道主播想要展示的美妆秀或者观看到主播各种精致的自拍等,实现了良好的直播互动效果。

[0046] 本实施例中的S101-S104可以部分步骤或者全部步骤在后台执行,从而不影响直播设备的使用。当然,本实施例的方案也可以在作为观众端的直播设备上实现。

[0047] 可选的是,预设图像配置在背景图的第一区域,直播视频帧的图像配置在背景图的第二区域,第一区域与第二区域不重合。在实际处理中,可以对预设图像和预设视频帧的长宽比例进行相应的处理,例如裁剪等,以使得两者较好地显示在直播界面上,例如在直播界面上预置相应的区域显示预设图像,从而能够规范预设图像的长宽标准。

[0048] 参阅图3和图4,本申请直播界面的图片配置方法第二实施例,基于本申请直播界面的图片配置方法第一实施例的基础上,对预设图像和直播视频帧配置在背景图进行详细阐述。因此本实施例与本申请直播界面的图片配置方法第一实施例相同的步骤在此不再赘述。如图4所示,本实施例包括:

[0049] S201:渲染出纯色图,将纯黑色图的RGB数值进行位图格式转换得到背景图。

[0050] 本申请直播界面的图片配置方法第一实施例的S101进一步包括S201。在本实施例中,比如直播设备渲染出纯黑色图,通过将纯黑色图的RGB数值进行位图格式转换得到背景图,也即包含位图数据的背景图。具体地,渲染好一张黑色的纯色背景图,通过YUV (256级别)的转换公式(例如createPixelBufferI420),将纯色的RGB数值转换成对应的I420格式的CVPixelBuffer结构体。其中,CVPixelBufferRef是一种像素图片类型,由于CV开头,所以它是属于IOS对象中CoreVideo模块的,也即CVPixelBuffer结构体相当于位图,包含位图数据。例如背景图标记为backBuffer。其中,I420(即YUV标准格式4:2:0)的数据量是 $size = width \times height \times 1.5Byte$ 。YUV是一种颜色格式,其中“Y”表示明亮度,也就是灰阶值;而“U”和“V”表示的则是色度,作用是描述影像色彩及饱和度,用于指定像素的颜色。对于Android操作系统环境而言,可以将纯色图的RGB数值转换成Bitmap,Bitmap同样也是一种像素图片类型,是使用像素阵列表示的图像。

[0051] S202:将预设图像进行格式转换得到预设图像的位图数据。

[0052] 在IOS操作系统环境下,例如将预设图像转换成YUV I420格式的CVPixelBuffer结构体,从而得到预设图像的位图数据,例如标记为imgBuffer。例如在Android操作系统环境下,同样可以将预设图像进行格式转换得到Bitmap类型的位图。本实施例将预设图像转换成位图数据比如CVPixelBuffer结构体或者Bitmap,便于后续进行高性能可靠的视频处理,例如编码形成直播数据,制作直播视频流。

[0053] 具体地,例如先将预设图像加载出来,通过CoreGraphics的CGBitmapContext将预设图像(比如预设图像标记为UIImage)转成ARGB格式的CVPixelBuffer。由于UIImage里面可能包含CGImage和CIImage,CIImage一般是描述图片处理或产生的过程。而CGImage才属于图片数据,是CoreGraphics框架的对象,因此可以通过CoreGraphics将UIImage转换成CVPixelBuffer。接着通过Libyuv库在ARGBToI420函数下将ARGB格式的CVPixelBuffer转换成I420格式的CVPixelBuffer。其中,ARGB是一种色彩模式,也就是RGB色彩模式附加上Alpha(透明度)通道,常见于32位位图的存储结构。其中,libyuv库是Google开源的实现各

种YUV与RGB之间相互转换、旋转、缩放的库。它是跨平台的,可在Windows、Linux、Mac、Android、IOS等操作系统,x86、x64、arm架构上进行编译运行,支持SSE、AVX、NEON等SIMD指令加速。

[0054] 在本实施例中,将预设图像转换成位图数据格式,能够便于后续进行图像渲染,提高完成配置过程的效率。

[0055] S203:将预设图像的位图数据渲染至背景图中。

[0056] 在本实施例中,渲染是指将预设图像的像素点对应显示在背景图上,从而在背景图显示整个预设图像的过程。将预设图像的位图数据渲染至背景图中,例如是将预设图像的像素数组替换到背景图的像素数组中。由于背景图是纯色的,其像素数组中的每个像素点的像素值均为相同的,例如纯黑色的背景图,其每个像素点的数值都为0,因此便于实现预设图像的像素点替换背景图的像素点,从而实现将预设图像配置在背景图中。其中本申请直播界面的图片配置方法第一实施例中的S102包括本实施例的S202和S203。

[0057] S204:通过预设SDK工具包获取直播视频帧并返回直播视频帧的位图数据。

[0058] 对于录屏方式而言,在IOS操作系统中,预设SDK工具包比如是基于Airplay的录屏SDK,能把运行IOS操作系统的直播设备上的屏幕内容(包括系统UI)直接投影到其它设备,因此也称为AirPlay镜像。通过预设SDK工具包进行录制后,视频数据流会以回调的形式返回给直播程序处理。SDK支持两种形式的视频数据:一种是AirPlay镜像传递的原始H264流,另外一种YUV数据,也即可以通过预设SDK工具包返回YUV格式的CVPixelBuffer结构体,从而得到直播视频帧的位图数据。在Android操作系统中,例如通过MediaProjection接口进行录屏的SDK可以获取到直播视频帧并返回到直播视频帧的位图数据。

[0059] 当然对于摄像头拍摄的方式而言,同样可以通过预设SDK工具包获取到摄像头所拍摄的直播视频帧并可以返回直播视频帧的位图数据。本实施例通过预设SDK工具包可以便利地获取到直播视频帧并返回直播视频帧的位图数据,可以不需要通过额外的函数或者格式转换,提高视频处理和渲染的效率。

[0060] S205:将直播视频帧的位图数据渲染至背景图中。

[0061] 在本实施例中,将直播视频帧各帧的位图数据依次渲染至背景图的位图数据中,例如是将每一帧直播视频帧的像素数组替换进背景图的像素数组中,从而根据直播视频帧的帧顺序形成以时间序列的一系列的背景图。由于背景图是纯黑色的,其像素数组中的每个像素点的像素值均为0,因此便于实现直播视频帧各帧的像素点依次替换背景图的像素点,最后编码形成直播数据的数据流,再进行推流处理。其中本申请直播界面的图片配置方法第一实施例中的S103包括本实施例的S204和S205。

[0062] S206:将渲染有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0063] 本实施例通过将纯黑色图像进行位图格式转换得到背景图,以黑色的背景图作为配置的基础,如此能够便于渲染从而提高渲染的速度,且将预设图像的位图数据渲染至背景图中,通过预设SDK工具包返回直播视频帧的位图数据并渲染至背景图的位图数据,能够提高预设图像和直播视频帧的渲染速度,能够快捷地在各种移动操作系统下实现预设图像和直播视频的交互,两者同时显示,增加了直播的趣味性和体验。

[0064] 参阅图5和图6,本申请直播界面的图片配置方法第三实施例,基于本申请直播界面的图片配置方法第一实施例和第二实施例的基础上,对预设图像和直播视频帧配置在背



景图进行进一步详细阐述。因此本实施例与本申请直播界面的图片配置方法第一实施例或第二实施例相同的步骤在此不再赘述。如图5所示,本实施例包括:

[0065] S301:渲染出纯色图,将纯色图的RGB数值进行格式转换得到背景图的位图数据。

[0066] S302:将预设图像进行格式转换得到预设图像的位图数据。

[0067] S303:将预设图像的位图数据拷贝至背景图对应的内存空间中以进行渲染。

[0068] 可选的是,调用预设函数将预设图像所对应的内存空间中的位图数据对应拷贝至背景图对应的内存空间中以进行渲染。例如预设函数为LibyuvI420Copy函数,将预设图像对应YUV三个颜色通道所分别对应的数据拷贝至背景图的对应的内存空间中,从而实现将预设图像配置在背景图中。当然,在其他实施例中,可以将预设图像的内存地址拷贝至背景图内存地址中,使得背景图和预设图像的内存地址相同,让背景图这一对象和预设图像这一对象指向同一片内存空间,相当于将预设图像复制在图中,从而实现了将预设图像渲染在背景图上。其中本申请直播界面的图片配置方法第二实施例中的S203包括本实施例的S303。对于Andriod操作系统而言,同样可以通过将预设图像的位图数据拷贝至背景图对应的内存空间中以实现预设图像的配置。

[0069] 可选的是,S303包括:获取预设图像的起始内存地址和背景图的起始内存地址。

[0070] 将预设图像的位图数据从预设图像的起始内存地址开始依次拷贝至从背景图的起始内存地址开始的背景图对应的内存空间中,在拷贝完毕时得到配置有预设图像的第一背景图。

[0071] 如图6所示,例如在背景图像素点的起始坐标(0,0)即从0行0列开始取,代表从(0,0)开始渲染,坐标(0,0)对应的内存地址可以标记为起始内存地址。而且,从预设图像的起始内存地址开始将数据对应拷贝至背景图从坐标(0,0)对应位图数据的内存地址开始的内存空间中。拷贝完成后,就可以得到在背景图的一个区域例如是左侧区域显示预设图像,另一区域仍然是黑色的背景图,此时包括有预设图像和部分黑色区域的背景图即为第一背景图,第一背景图例如标记为wrapBuffer=(0,0,imgW,imgH) imgBuffer+backBuffer,其中(0,0,imgW,imgH)中的(0,0)和(imgW,imgH)分别表示预设图像的位图数据拷贝的起始坐标和终止坐标,imgW,imgH分别表示预设图像的长度和宽度。对于YUV420格式而言,Y是 $W \times H$ ,U和V是 $W/2 \times (H/2)$ ,也即imgW等于Y通道的每行字节数,imgH等于Y通道的每列字节数,U和V通道是Y通道的二分之一宽,二分之一长。如果 $W=4,H=2$ ,则内存则是:yyyyyyyyuuuvv,需要占用的内存: $W \times H \times 3/2$ 。

[0072] 具体地,例如调用LibyuvI420Copy函数先获取预设图像的Y通道的起始内存地址,从而获取Y通道每行数据所占用的字节,接着获取U通道的起始内存地址和V通道的起始内存地址,根据Y通道与U通道和V通道之间的关系可以知道U通道和V通道所占用字节,从而可以分别将YUV三通道的对应的字节数拷贝到背景图中对应的YUV通道的内存空间中,根据YUV三通道所分别对应的数据,例如各自对应的字节数,可实现在背景图上配置预设图像。

[0073] 如图6所示,对于第一背景图而言,未被渲染的部分仍是黑色的,也即在二维平面中,该部分的起始坐标为(imgW,0),在本实施例中,可以以坐标wrapBuffer(imgW,0)所对应位图数据的内存地址作为第一背景图(wrapBuffer)的起始内存地址。

[0074] S304:通过预设SDK工具包获取直播视频帧并返回直播视频帧的位图数据。

[0075] S305:将直播视频帧的位图数据拷贝至背景图对应的内存空间中以进行渲染。

[0076] 同理,可以调用预设函数将直播视频帧所对应的内存空间中的位图数据对应拷贝至背景图对应的内存空间中以进行渲染。例如预设函数为LibyuvI420Copy函数,将直播视频帧对应YUV三个颜色通道所分别对应的数据拷贝至背景图的对应的内存空间中,从而实现将直播视频帧配置在背景图中。当然在其他实施例中,可以将直播视频帧的位图数据对应的内存地址拷贝至背景图对应的内存地址,如此背景图这一对象和直播视频帧这一对象指向同一片内存空间,能够实现将预设视频帧渲染在背景图上。

[0077] 可选的是,本实施例的S305进一步包括:将直播视频帧的位图数据加入待渲染队列。

[0078] 从待渲染队列中依次将直播视频帧的位图数据取出并将所取出的直播视频帧的位图数据拷贝至背景图对应的内存空间中以进行渲染。

[0079] 由于直播视频帧是逐帧产生的,而且是随着直播时间的推移是不断产生的,但是由于渲染所需的时间可能存在稍慢于直播视频帧产生的时间的情况。因此本实施例将在直播视频帧逐帧产生时加入到待渲染队列,以等待渲染。换言之,本实施例中的第一背景图wrapBuffer是可以重复使用。每一帧直播视频帧均各自渲染在wrapBuffer中。例如将直播视频帧的位图数据标记为targerBuffer。

[0080] 例如从待渲染队列中取出targerBuffer,调用libyuvI420Copy函数从坐标targetBuffer (0,0)对应的位图数据的内存地址开始依次将targetBuffer拷贝到背景图对应的内存空间中。

[0081] 可选的是,S305包括获取直播视频帧的起始内存地址和第一背景图的起始内存地址。

[0082] 将直播视频帧的位图数据从直播视频帧的起始内存地址依次进拷贝至从第一背景图的起始内存地址开始的第一背景图对应的内存空间内,在拷贝完毕时获得配置有预设图像和直播视频帧的第二背景图。

[0083] 在本实施例中,直播视频帧的位图数据拷贝至第一背景图对应的内存空间内,从首先拷贝至第一背景图的起始内存地址,依次进行拷贝。。

[0084] 如图6所示,例如从待渲染队列中依次取出targerBuffer,调用libyuvI420Copy函数从targetBuffer的起始坐标targetBuffer (0,0)对应的位图数据的内存地址(可以视为targetBuffer的内存地址)开始依次将targerBuffer拷贝至以wrapBuffer (imgW,0)为起始坐标地址的位图数据对应内存地址为开始的wrapBuffer的内存空间中,得到了配置有预设图像和直播视频帧的第二背景图。

[0085] 例如第二背景图标记为resultBuffer= (0,0,imgW,imgH) imgBuffer+(imgW,0,targetW,targetH) targetBuffer。也即,在第二背景图中直播视频帧与预设图像实现了拼接交互。

[0086] 可选的是,第二背景图的终止内存地址为背景图的终止内存地址。换言之,背景图相当于已经完全被预设图像和直播视频帧的画面占据,例如预设图像位于背景图的左侧,直播视频帧位于背景图的右侧,实现了两者拼接且不重合。

[0087] S306:将渲染有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0088] S307:将直播数据以预设协议进行推流至服务器。

[0089] 在本实施例中,预设协议例如是指RTMP协议,当然还可以是其他适用于推流的协

议。

[0090] 现有的Android等移动操作系统,对接口权限都有一定的要求,并不能随意实现接口的调用,在后台情况下,对于接口权限的调用较为困难。相对于Android等操作系统而言,IOS操作系统对权限控制更为严格,IOS操作系统限制了在后台的情况下,不允许访问GPU允许渲染的命令,因此不能通过常规的GPU进行渲染。本实施例通过在后台将预设图像的位图数据、以及直播视频帧各帧的位图数据依次拷贝至背景图对应的内存空间中,而且通过待渲染队列实现第一背景图的重复使用,不涉及创建新对象,不需要用到GPU进行渲染,而是通过CPU和内存进行操作实现渲染,因此无需请求Android或者IOS等操作系统的相应的接口权限,渲染耗时较少从而可以快捷地实现了预设图像和直播视频帧配置在背景图中,提高了渲染和处理的效率,同时能够实现图像和视频在直播界面中同时显示的交互,提升直播体验。

[0091] 参阅图7,本申请移动终端第一实施例,其中本实施例的移动终端运行IOS操作系统或者Android等操作系统,本实施例包括处理器201、通信电路202和显示器203,处理器201通过通信总线分别耦接通信电路202和显示器203。

[0092] 处理器201用于获取背景图。处理器201用于获取预设图像并将预设图像配置在背景图中。处理器201用于获取直播视频帧并将直播视频帧配置在背景图中。处理器201用于将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0093] 在本实施例中,通信电路202是处理器201与外部设备进行通信的接口。显示器203具有显示功能,可以用于显示预设图像和直播视频帧,还可以显示直播界面。

[0094] 在本实施例中,处理器201还可以称为CPU (Central Processing Unit,中央处理单元)。处理器201可能是一种集成电路芯片,具有信号的处理能力。处理器201还可以是通用处理器、数字信号处理器 (DSP)、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0095] 本实施例中关于移动终端更多执行过程等可以参照上述本申请直播界面的图片配置方法第一实施例至第三实施例。在本实施例中,移动终端比如是智能手机、平板、智能穿戴设备等。

[0096] 参阅图8,本申请移动终端第二实施例,其中本实施例的移动终端运行IOS操作系统或者Android等操作系统,本实施例包括第一获取模块211、第二获取模块212、配置模块213以及编码模块214。其中第一获取模块211、第二获取模块212、配置模块213以及编码模块214相互连接。

[0097] 其中,第一获取模块211用于获取背景图。第二获取模块212用于获取预设图像。配置模块213用于将预设图像配置在背景图中。第二获取模块212用于获取直播视频帧。配置模块213用于将直播视频帧配置在背景图中。编码模块214用于将配置有预设图像和直播视频帧的背景图进行编码形成直播数据。

[0098] 作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块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模块来实现本实施方式方案的目的。

[0099] 另外,在实施例各功能模块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。

[0100] 本实施例中关于各模块更多功能或者更多的模块等可以参照上述本申请直播界面的图片配置方法第一实施例至第三实施例。

[0101] 参阅图9,本申请具有存储功能的装置实施例中的装置30,存储有程序数据,程序数据能够被执行以实现上述本申请直播界面的图片配置方法第一实施例至第三实施例所涉及的步骤。具体参照本申请直播界面的图片配置方法第一实施例至第三实施例,在此不再赘述。

[0102] 本申请直播界面的图片配置方法第一实施例至第三实施例所涉及的方法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,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,本申请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装置30中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(processor)执行本发明各个实施方式方法的全部或部分步骤。而前述的装置30包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质,或包括存储介质的电脑等其他装置。

[0103] 以上所述仅为本申请的实施方式,并非因此限制本申请的专利范围,凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本申请的专利保护范围内。

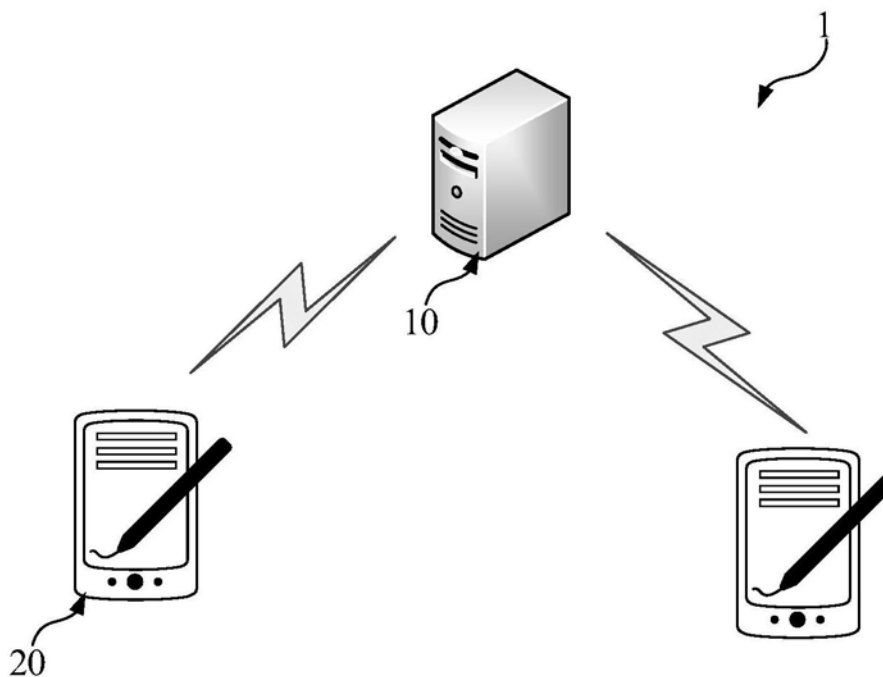


图1

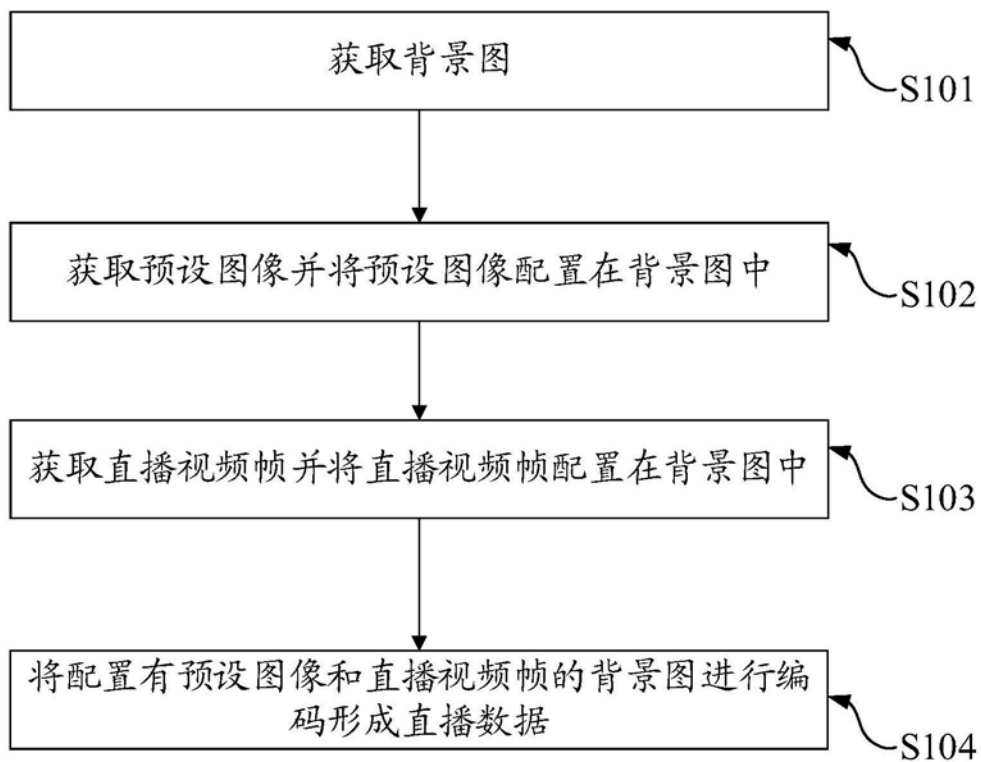


图2

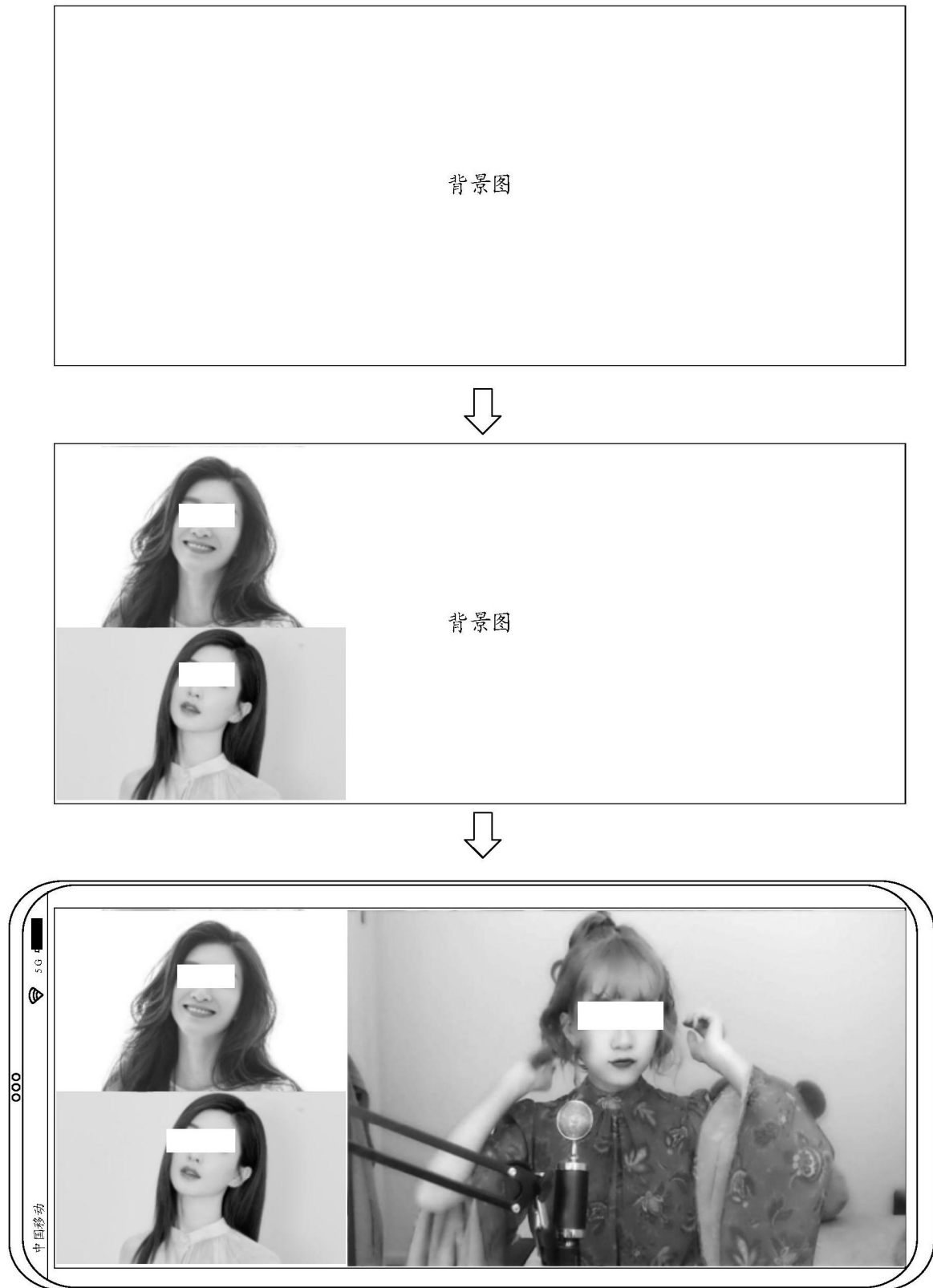


图3

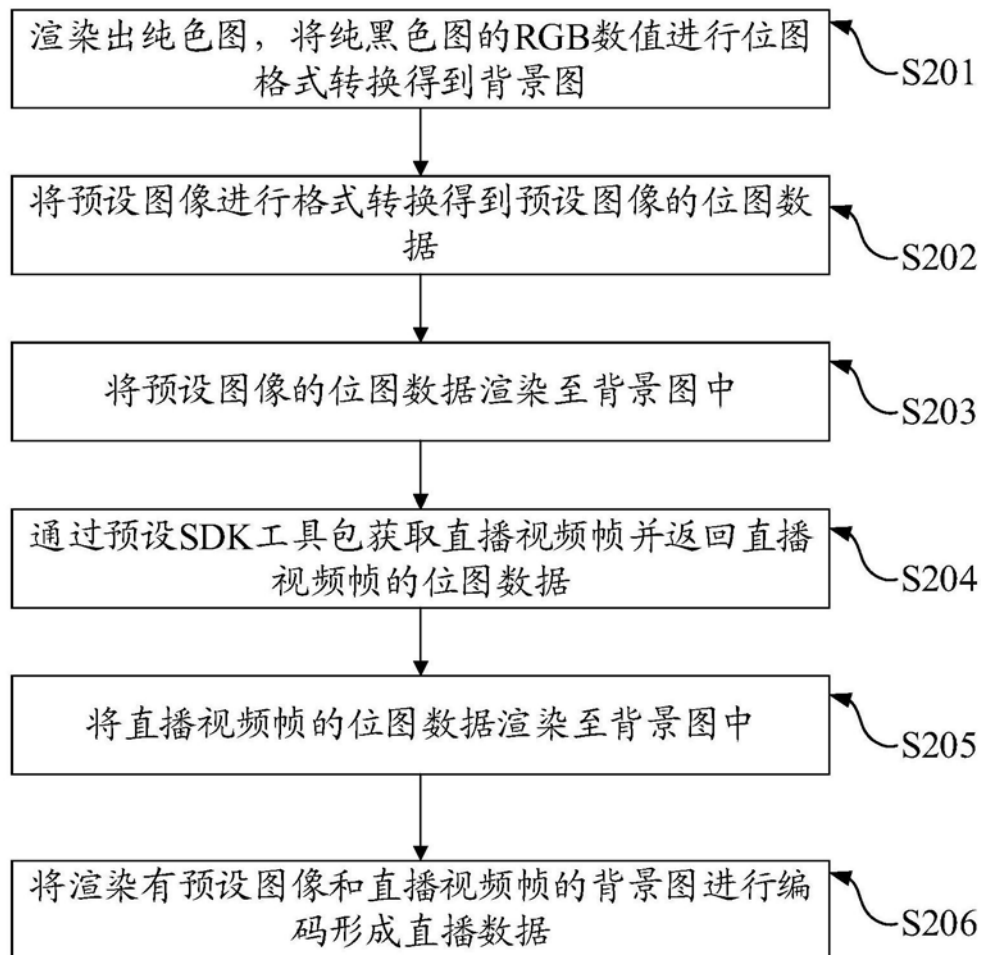


图4

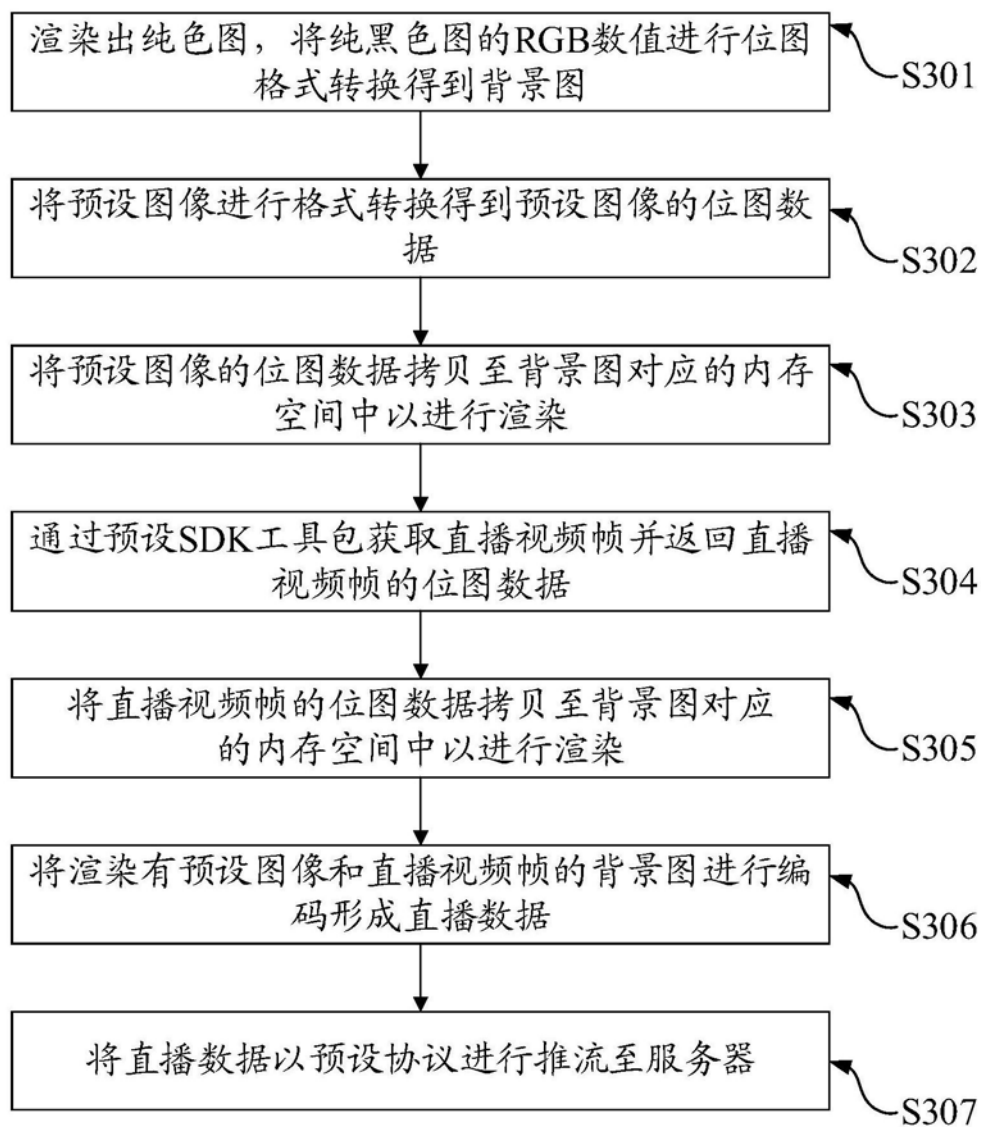


图5



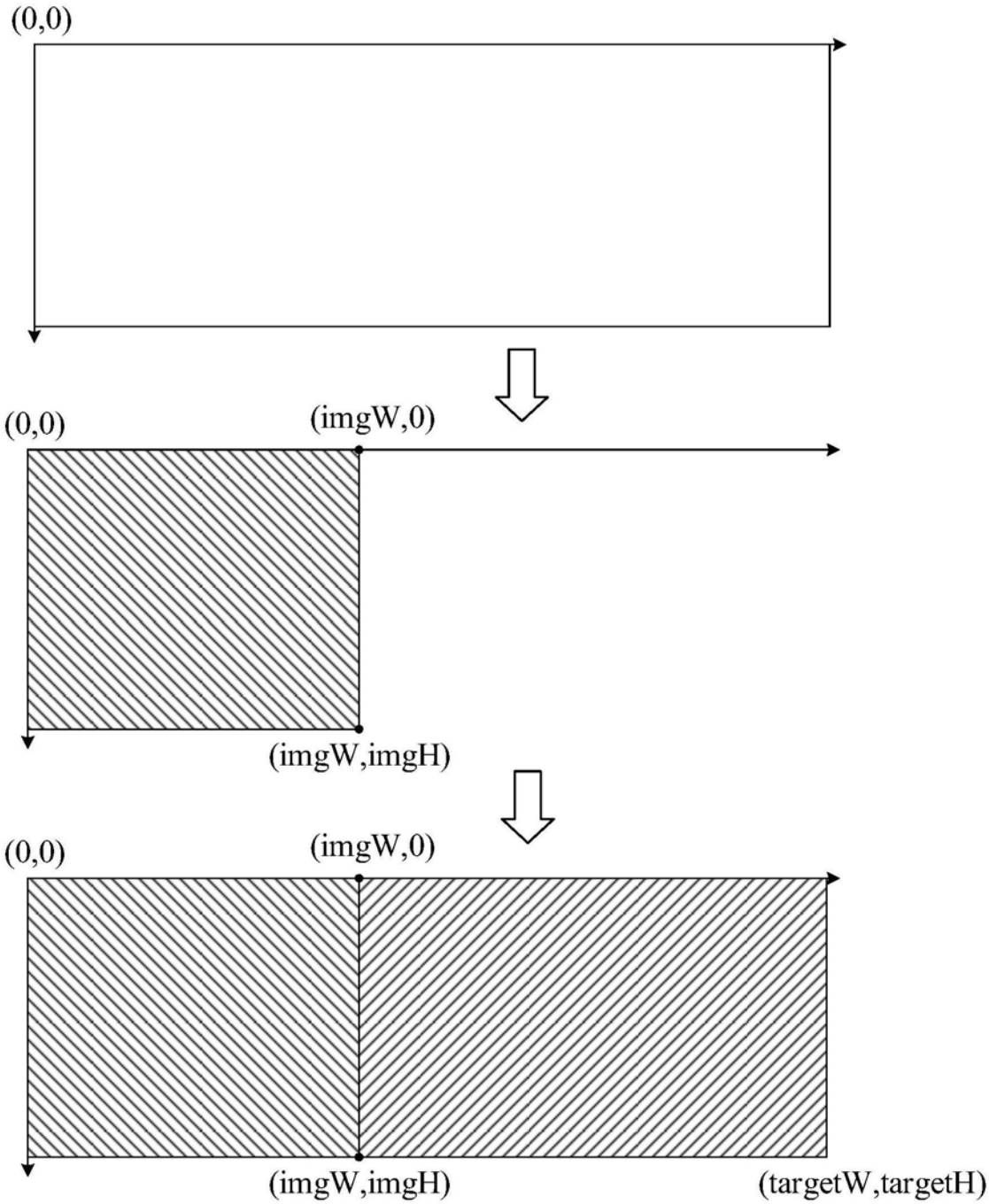


图6

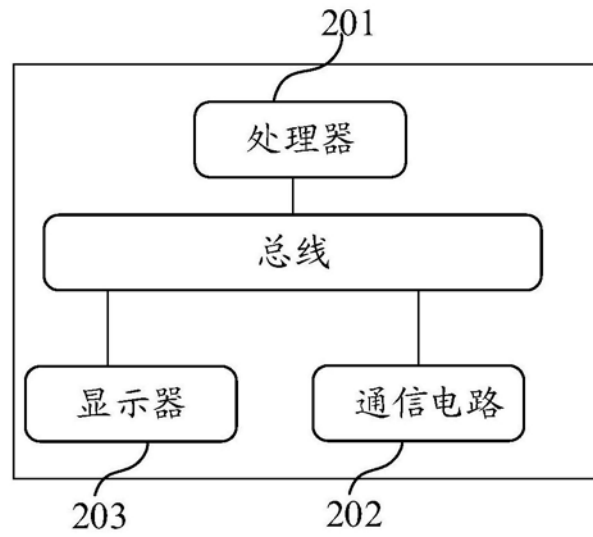


图7

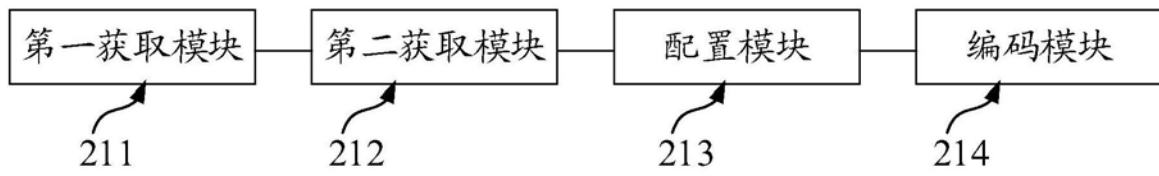


图8

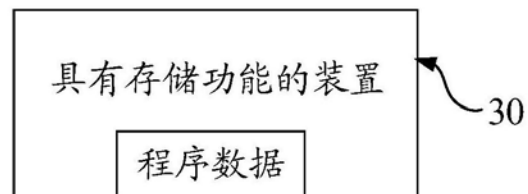


图9