



(12) 发明专利

(10) 授权公告号 CN 1804900 B

(45) 授权公告日 2010.08.04

(21) 申请号 200510138038.6

(22) 申请日 2005.09.22

(30) 优先权数据

10/947852 2004.09.22 US

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 S·库鲁帕蒂 B·尼克尔森 S·王

S·乔德哈里 J·刘

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

代理人 程天正 王忠忠

(51) Int. Cl.

G06T 3/40(2006.01)

G06T 5/00(2006.01)

H04N 5/14(2006.01)

H04N 7/26(2006.01)

(56) 对比文件

US 6144700 A, 2000.11.07, 全文.

US 6005580 A, 1999.12.21, 全文.

US 5689343 A, 1997.11.18, 全文.

US 5528704 A, 1996.06.18, 全文.

US 5650858 A, 1997.07.22, 全文.

审查员 梁燕

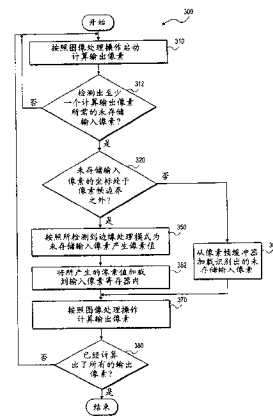
权利要求书 3 页 说明书 11 页 附图 6 页

(54) 发明名称

用于图像处理中的边缘处理的设备和方法

(57) 摘要

一种视频后置处理中基于硬件的边缘处理的方法和设备。在一种实施例中,该方法包括在输出像素计算期间识别出至少一个未存储的计算输出像素所需的输入像素。一旦识别到,就按照检测到的边缘处理模式对所述至少一个未存储的输入像素产生像素值。在一种实施例中,为未存储的输入像素产生像素值是在未存储的输入像素处于像素帧边界之外的情况下进行的。例如,在一种实施例中,对于缩放操作的输出像素计算,帧边界包括左(上)边缘和右(下)边缘,对于这些边缘,在帧边界上或帧边界附近,并不存在计算输出像素所需的输入像素。还介绍了其它一些实施例,并且要求保护这些实施例。



1. 一种用于图像处理中的边缘处理的方法,包括:
在输出像素计算期间识别出至少一个计算输出像素所需的未经存储的输入像素;然后
如果未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式来为所识别出的未经存储的输入像素产生像素值。
2. 按照权利要求 1 所述的方法,还包括:
将所产生的像素值加载到输入像素寄存器内,以存储计算输出像素所需的 N 个输入像素,此处 N 是整数。
3. 按照权利要求 1 所述的方法,还包括:
如果所识别出的未经存储的输入像素的坐标处于像素帧边界之内,则从像素帧缓冲器加载所识别出的未经存储的输入像素。
4. 按照权利要求 1 所述的方法,其中所述产生像素值的步骤还包括:
访问模式寄存器,以将边缘处理模式确定为基于零的边缘处理模式、复制边缘处理模式和镜像边缘处理模式中的一种;和
按照所识别出的边缘处理模式形成像素值。
5. 按照权利要求 1 所述的方法,其中所述产生像素值的步骤还包括:
确定像素加载计数;
如果像素加载计数等于规定的加载计数,则检测左像素帧边缘;和
如果像素加载计数大于或等于像素帧的线宽,则检测右像素帧边缘。
6. 按照权利要求 5 所述的方法,其中检测左像素帧边缘还包括:
在多个相邻输入像素寄存器之间对输入像素进行移位;
将输入像素从像素帧缓冲器存储到最低有效位输入像素寄存器内;
识别出至少一个空输入像素寄存器;和
将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。
7. 按照权利要求 5 所述的方法,其中检测右像素帧边缘还包括:
在每个相邻输入像素寄存器之间对输入像素值进行移位,以空出输入像素寄存器;和
将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。
8. 按照权利要求 1 所述的方法,其中输出像素计算包括视频缩放操作。
9. 按照权利要求 1 所述的方法,其中所述识别输入像素的步骤包括:
确定计算输出像素所需的 N 个输入像素;和
如果计算输出像素所需的 N 个输入像素中的至少一个并不包含在 N 个输入像素寄存器中的输入像素寄存器内,则检测至少一个未经存储的输入像素,此处 N 是整数。
10. 按照权利要求 5 所述的方法,其中检测右像素帧边缘还包括:
访问线宽寄存器,以确定像素帧的线宽。
11. 一种用于图像处理中的边缘处理的方法,包括:
在输出像素计算期间检测出计算输出像素所需的一个或多个输入像素,所检测到的输入像素均具有处于像素帧边界之外的坐标;
按照所检测到的边缘处理模式为每个所检测到的未经存储的输入像素产生像素值;和
按照图像处理操作计算输出像素。
12. 按照权利要求 11 所述的方法,还包括:

如果所识别出的未经存储的输入像素的坐标处于像素帧边界之内,则从像素帧缓冲器加载一个或多个所识别出的计算输出像素所需的未经存储的输入像素;和

将每个所产生的像素值加载到输入像素寄存器内,以存储计算输出像素所需的 N 个输入像素,此处 N 是整数。

13. 按照权利要求 11 所述的方法,其中为了产生像素值,所述方法还包括:

确定像素加载计数;

如果像素加载计数等于规定的加载计数,则检测左像素帧边缘;和

如果像素加载计数大于或等于像素帧的线宽,则检测右像素帧边缘。

14. 按照权利要求 13 所述的方法,其中为了检测左像素帧边缘,所述方法还包括:

在多个相邻输入像素寄存器之间对输入像素进行移位;

将输入像素从像素帧缓冲器存储到最低有效位输入像素寄存器内;

识别出至少一个空输入像素寄存器;和

将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

15. 按照权利要求 13 所述的方法,其中为了检测右像素帧边缘,所述方法还包括:

在每个相邻输入像素寄存器之间对输入像素值进行移位,以空出输入像素寄存器;和

将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

16. 一种用于图像处理中的边缘处理的设备,包括:

多个输入像素寄存器;和

与所述多个输入像素寄存器相耦合的控制器,该控制器用来在输出像素计算期间识别出计算输出像素所需的、至少一个未存储在所述多个输入像素寄存器内的输入像素,并且,如果所识别出的未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式为所识别出的未经存储的输入像素产生像素值。

17. 按照权利要求 16 所述的设备,还包括:

边缘处理模式寄存器,用于将为具有处于像素帧边界之外的坐标的输入像素产生像素值所用的边缘处理模式识别为基于零的边缘处理模式、复制边缘处理模式和镜像边缘处理模式中的一种。

18. 按照权利要求 17 所述的设备,还包括:

线宽寄存器,该线宽寄存器用于存储像素帧的线宽和像素加载计数。

19. 按照权利要求 16 所述的设备,还包括:

第一逻辑门,用于接收来自控制器的控制信号,并且依照控制信号,当检测到右边缘时,用所产生的像素值填充第一输入像素寄存器,而在其它情况下将从像素帧缓冲器中取得的输入像素值加载到第一输入像素寄存器内。

20. 按照权利要求 19 所述的设备,其中第一逻辑门用于在每个相邻输入像素寄存器之间对像素值进行移位以空出第一输入像素寄存器之后,对第一输入像素寄存器进行加载。

21. 按照权利要求 19 所述的设备,还包括:

与多个输入像素寄存器中的一个或多个相耦合的一个或多个第二逻辑门,所述一个或多个第二逻辑门用于接收来自控制器的控制信号,以在检测到左边缘的时候将像素值加载到输入像素寄存器内,并且在其它情况下在相邻的输入像素寄存器之间对像素值进行移位。

22. 按照权利要求 18 所述的设备,还包括:

装置驱动器,所述装置驱动器对边缘处理模式寄存器和线宽寄存器进行加载。

23. 按照权利要求 21 所述的设备,其中第一逻辑门是多路复用器,并且所述一个或多个第二逻辑门是多路复用器,作为多路复用器的所述第一逻辑门和作为多路复用器的所述一个或多个第二逻辑门用于按照从控制器接收的右边缘控制信号和从控制器接收的左边缘控制信号之一选择输入。

24. 按照权利要求 16 所述的设备,其中所述控制器用于在检测到输入像素帧的左边缘和右边缘之一时产生用来将所产生的像素值加载到输入像素寄存器内的控制信号。

25. 按照权利要求 18 所述的设备,还包括:

控制处理器,用于对边缘处理模式寄存器和线宽寄存器进行加载。

26. 一种用于图像处理中的边缘处理的系统,包括:

嵌入式处理器;

与该嵌入式处理器相耦合的后置处理逻辑,该后置处理逻辑包括缩放器,该缩放器包括:

多个输入像素寄存器,和

控制器,用于在输出像素计算期间识别出计算输出像素所需的、未存储在输入像素寄存器内的至少一个输入像素,并且,如果未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式为未经存储的输入像素产生像素值;和

显示装置,用于显示经过后置处理的视频帧。

27. 按照权利要求 26 所述的系统,还包括:

存储器控制器;

系统存储器;

与显示装置相耦合的直接存储器访问 DMA 缓冲器;和

与 DMA 缓冲器相耦合的 DMA 控制器,用于向所述存储器控制器发出用来自所述系统存储器的输入像素加载 DMA 缓冲器的请求。

28. 按照权利要求 26 所述的系统,其中该系统包括芯片上系统。

29. 按照权利要求 26 所述的系统,其中该系统还包括:

视频解码器,用于对来自芯片外系统存储器的数字视频进行解码,并且将经过解码的数字视频帧提供给后置处理逻辑。

30. 按照权利要求 26 所述的系统,还包括:

边缘处理模式寄存器和线宽寄存器,其中所述嵌入式处理器用来对边缘处理模式寄存器和线宽寄存器进行加载。

用于图像处理中的边缘处理的设备和方法

技术领域

[0001] 本发明的一种或多种实施例涉及视频后置处理和图像增强领域。更加具体地讲，一种实施例涉及用于图像处理中的边缘处理的方法和设备。

背景技术

[0002] 目前的数字电视 (DTV) 可以在编码的媒体信号解码之后但显示解码的数字视频帧之前的 DTV 显示处理流水线中提供视频后置处理和 / 或图像增强。视频后置处理的例子包括水平和垂直视频缩放。在 DTV 芯片内进行的水平和垂直视频缩放可以使用多相滤波器来实现。总体上, 可以将视频缩放看成内插 / 抽取过程, 这一过程包括从一组输入像素产生更大 / 更小 (取决于缩放系数) 数量的输出像素。当缩放是使用多相滤波器实现之时, 所产生的各个输出像素是 N 个输入像素的函数。

[0003] 一般来说, N 是整数, 指的是由多相滤波器给出的用来进行缩放过程的抽头的数量。产生输出像素所需的 N 个输入像素一般以输出像素的位置为中心, 例如, 在使用对称多相滤波器的时候就是如此。虽然这对大多数输出像素是很有作用的, 但是却在像素帧的边缘造成了问题。实际上, 至少有一个产生像素帧边缘上的输出像素所需的输入像素是不存在的。

[0004] 举例来说, 考虑从 720 水平像素到 1920 像素的缩放 (放大)。假设这个例子中的抽头数量也就是 N 等于九。根据这一抽头数量, 为了产生输出像素零 (用 0-1919 对像素进行编号), 需要九个输入像素, 这九个输入像素以零为中心。虽然输入像素 0-4 是存在的, 但像素 -4 到 -1 却并不存在。类似地, 对输出像素 1、2 和 3 进行的输出像素计算也有缺失的输入像素。虽然缺失的输入像素的数量在计算后续输出像素的时候会减少并且最终会消失, 但是在像素帧的右边缘 (上下边缘 (对于垂直缩放)) 会再次出现这一问题, 即, 输入像素 719 (对于 720 线宽像素帧) 右侧的输入像素也不存在。

[0005] 为了解决这一问题, 目前存在数种边缘处理方法。这些方法专注于在像素帧周围填补额外的边缘像素。这些方法包括: 复制 (在上述例子中, 像素 -4 到 -1 全部取像素 0 的值); 镜像, 其中像素 -4 到 -1 取像素 1-4 的值; 和基于零, 在这种情况下, 将像素 -4 到 -1 简单地设为零。

[0006] 上述方法中的任何一种的实现一般来说都牵涉到在进行缩放操作之前通过软件或其它硬件单元真正产生这些填补像素并且将它们保存在存储器中。结果, 视频缩放器要读取整个经过填补的图像。硬件厂商常常会完全忽略边缘处理问题并且放弃像素帧边缘处缩放的质量。

发明内容

[0007] 在本发明的第一个方面, 提供了一种用于图像处理中的边缘处理的方法, 包括:

[0008] 在输出像素计算期间识别出至少一个计算输出像素所需的未经存储的输入像素; 然后

[0009] 如果未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式来为所识别出的未经存储的输入像素产生像素值。

[0010] 优选地,所述方法还包括:

[0011] 将所产生的像素值加载到输入像素寄存器内,以存储计算输出像素所需的 N 个输入像素,此处 N 是整数。

[0012] 优选地,所述方法还包括:

[0013] 如果所识别出的未经存储的输入像素的坐标处于像素帧边界之内,则从像素帧缓冲器加载所识别出的未经存储的输入像素。

[0014] 优选地,所述产生像素值的步骤还包括:

[0015] 访问模式寄存器,以将边缘处理模式确定为基于零的边缘处理模式、复制边缘处理模式和镜像边缘处理模式中的一种;和

[0016] 按照所识别出的边缘处理模式形成像素值。

[0017] 优选地,所述产生像素值的步骤还包括:

[0018] 确定像素加载计数;

[0019] 如果像素加载计数等于规定的加载计数,则检测左像素帧边缘;和

[0020] 如果像素加载计数大于或等于像素帧的线宽,则检测右像素帧边缘。

[0021] 优选地,检测左像素帧边缘还包括:

[0022] 在多个相邻输入像素寄存器之间对输入像素进行移位;

[0023] 将输入像素从像素帧缓冲器存储到最低有效位输入像素寄存器内;

[0024] 识别出至少一个空输入像素寄存器;和

[0025] 将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

[0026] 优选地,检测右像素帧边缘还包括:

[0027] 在每个相邻输入像素寄存器之间对输入像素值进行移位,以空出输入像素寄存器;和

[0028] 将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

[0029] 优选地,输出像素计算包括视频缩放操作。

[0030] 优选地,所述识别输入像素的步骤包括:

[0031] 确定计算输出像素所需的 N 个输入像素;和

[0032] 如果计算输出像素所需的 N 个输入像素中的至少一个并不包含在 N 个输入像素寄存器中的输入像素寄存器内,则检测至少一个未经存储的输入像素,此处 N 是整数。

[0033] 优选地,检测右像素帧边缘还包括:

[0034] 访问线宽寄存器,以确定像素帧的线宽。

[0035] 在本发明的第二个方面,提供了一种用于图像处理中的边缘处理的方法,包括:

[0036] 在输出像素计算期间检测出计算输出像素所需的一个或多个输入像素,所检测到的输入像素均具有处于像素帧边界之外的坐标;

[0037] 按照所检测到的边缘处理模式为每个所检测到的未经存储的输入像素产生像素值;和

[0038] 按照图像处理操作计算输出像素。

[0039] 优选地,所述方法还包括:

[0040] 如果所识别出的未经存储的输入像素的坐标处于像素帧边界之内,则从像素帧缓冲器加载一个或多个所识别出的计算输出像素所需的未经存储的输入像素;和

[0041] 将每个所产生的像素值加载到输入像素寄存器内,以存储计算输出像素所需的N个输入像素,此处N是整数。

[0042] 优选地,为了产生像素值,所述方法还包括:

[0043] 确定像素加载计数;

[0044] 如果像素加载计数等于规定的加载计数,则检测左像素帧边缘;和

[0045] 如果像素加载计数大于或等于像素帧的线宽,则检测右像素帧边缘。

[0046] 优选地,为了检测左像素帧边缘,所述方法还包括:

[0047] 在多个相邻输入像素寄存器之间对输入像素进行移位;

[0048] 将输入像素从像素帧缓冲器存储到最低有效位输入像素寄存器内;

[0049] 识别出至少一个空输入像素寄存器;和

[0050] 将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

[0051] 优选地,为了检测右像素帧边缘,所述方法还包括:

[0052] 在每个相邻输入像素寄存器之间对输入像素值进行移位,以空出输入像素寄存器;和

[0053] 将按照所检测到的边缘处理模式产生的像素值加载到空输入像素寄存器内。

[0054] 在本发明的第三个方面,提供了一种用于图像处理中的边缘处理的设备,包括:

[0055] 多个输入像素寄存器;和

[0056] 与所述多个输入像素寄存器相耦合的控制器,该控制器用来在输出像素计算期间识别出计算输出像素所需的、至少一个未存储在所述多个输入像素寄存器内的输入像素,并且,如果所识别出的未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式为所识别出的未经存储的输入像素产生像素值。

[0057] 优选地,所述设备还包括:

[0058] 边缘处理模式寄存器,用于将为具有处于像素帧边界之外的坐标的输入像素产生像素值所用的边缘处理模式识别为基于零的边缘处理模式、复制边缘处理模式和镜像边缘处理模式中的一种。

[0059] 优选地,所述设备还包括:

[0060] 线宽寄存器,该线宽寄存器用于存储像素帧的线宽和像素加载计数。

[0061] 优选地,所述设备还包括:

[0062] 第一逻辑门,用于接收来自控制器的控制信号,并且依照控制信号,当检测到右边缘时,用所产生的像素值填充第一输入像素寄存器,而在其它情况下将从像素帧缓冲器中取得的输入像素值加载到第一输入像素寄存器内。

[0063] 优选地,第一逻辑门用于在每个相邻输入像素寄存器之间对像素值进行移位以空出第一输入像素寄存器之后,对第一输入像素寄存器进行加载。

[0064] 优选地,所述设备还包括:

[0065] 与多个输入像素寄存器中的一个或多个相耦合的一个或多个第二逻辑门,所述一个或多个第二逻辑门用于接收来自控制器的控制信号,以在检测到左边缘的时候将像素值加载到输入像素寄存器内,并且在其它情况下在相邻的输入像素寄存器之间对像素值进行

移位。

[0066] 优选地,所述设备还包括:

[0067] 装置驱动器,所述装置驱动器对边缘处理模式寄存器和线宽寄存器进行加载。

[0068] 优选地,第一逻辑门是多路复用器,并且所述一个或多个第二逻辑门是多路复用器,作为多路复用器的所述第一逻辑门和作为多路复用器的所述一个或多个第二逻辑门用于按照从控制器接收的右边缘控制信号和从控制器接收的左边缘控制信号之一选择输入。

[0069] 优选地,所述控制器用于在检测到输入像素帧的左边缘和右边缘之一时产生用来将所产生的像素值加载到输入像素寄存器内的控制信号。

[0070] 优选地,所述设备还包括:

[0071] 控制处理器,用于对边缘处理模式寄存器和线宽寄存器进行加载。

[0072] 在本发明的第四个方面,提供了一种用于图像处理中的边缘处理的系统,包括:

[0073] 嵌入式处理器;

[0074] 与该嵌入式处理器相耦合的后置处理逻辑,该后置处理逻辑包括缩放器,该缩放器包括:

[0075] 多个输入像素寄存器,和

[0076] 控制器,用于在输出像素计算期间识别出计算输出像素所需的、未存储在输入像素寄存器内的至少一个输入像素,并且,如果未经存储的输入像素的坐标处于像素帧边界之外,则按照所检测到的边缘处理模式为未经存储的输入像素产生像素值;和

[0077] 显示装置,用于显示经过后置处理的视频帧。

[0078] 优选地,所述系统还包括:

[0079] 存储器控制器;

[0080] 系统存储器;

[0081] 与显示装置相耦合的直接存储器访问 DMA 缓冲器;和

[0082] 与 DMA 缓冲器相耦合的 DMA 控制器,用于向所述存储器控制器发出用来自所述系统存储器的输入像素加载 DMA 缓冲器的请求。

[0083] 优选地,该系统包括芯片上系统。

[0084] 优选地,该系统还包括:

[0085] 视频解码器,用于对来自芯片外系统存储器的数字视频进行解码,并且将经过解码的数字视频帧提供给后置处理逻辑。

[0086] 优选地,该系统还包括:

[0087] 边缘处理模式寄存器和线宽寄存器,其中所述嵌入式处理器用来对边缘处理模式寄存器和线宽寄存器进行加载。

附图说明

[0088] 在附图中,为了举例而非为了限制,图解说明了本发明的各种实施例,其中:

[0089] 附图 1 是表示按照一种实施例的边缘处理逻辑的框图;

[0090] 附图 2 是表示按照一种实施例的像素帧边界处的边缘处理的框图;

[0091] 附图 3 是表示按照一种实施例的包括边缘处理逻辑(例如,如附图 1 所示的边缘处理逻辑)的系统的框图;

- [0092] 附图 4 是表示按照一种实施例的图像处理中的边缘处理方法的流程图；
- [0093] 附图 5 是表示按照一种实施例在像素帧边界处的边缘处理方法的流程图。
- [0094] 附图 6 是表示使用所公开的技术的设计方案的仿真、模拟和制造的各种不同设计方案表达方式或格式的框图。

具体实施方式

[0095] 在下面的说明中,给出了大量的具体细节,比如逻辑实现、信号和总线的大小和名称、系统组件的类型和相互关系以及逻辑划分 / 整合选择,为的是提供更加全面的理解。不过,本领域技术人员将会意识到,在没有这些具体细节的情况下,也能够实践本发明。在其它例子中,没有详细给出控制结构和门电平电路,以避免使本发明重点不突出。本领域的普通技术人员在使用所包含的说明书的情况下,将能够实现合乎要求的逻辑电路,而不需要过度的试验。

[0096] 系统

[0097] 附图 1 是表示按照一种实施例的边缘处理逻辑 100 的框图。在一种实施例中,边缘处理逻辑 100 在诸如视频后置处理和 / 或图像增强之类的图像处理期间执行边缘处理。在一种实施例中,边缘处理逻辑 100 并入在视频后置处理逻辑(比如,显示器群集器)的水平 and 垂直缩放器中。本文介绍的实施例并不局限于将边缘处理逻辑并入在水平和垂直缩放器中,并且可以用在其它类型的后置处理逻辑中。实例包括用在数字电视(DTV)的显示处理流水线、计算机系统内具有集成图形控制器的系统芯片组的显示引擎或其它显示装置中。

[0098] 具有代表性地,边缘处理逻辑 100 包括控制器 110,可以对该控制器 110 进行编程或硬连接,以指示输入逻辑门 120 以及第二逻辑门 130(130-1、103-2、130-3 和 130-4) 填充输入像素寄存器 140(140-1、140-2、140-3、140-4、140-5、140-6、140-7、140-8 和 140-9)。如附图 1 所示,边缘处理逻辑 100 是在视频后置处理逻辑的水平 / 垂直缩放器内工作的。不过,本文所介绍的实施例并不局限于在垂直或水平缩放器内实现边缘处理逻辑 100,而是可以应用于对像素帧进行处理以从处于像素帧边缘之外的输入像素产生输出像素的其它视频后置处理和图像处理技术,例如,如附图 2 所示。

[0099] 附图 2 是表示包括多个输入像素 $166(Z_{0,0}, \dots, Z_{i,j})$ 的像素帧缓冲器 150 的框图。如图所示,像素帧缓冲器 150 包括左边缘 152、右边缘 154、上边缘 156 和下边缘 158。因此,在一种实施例中,其中边缘处理逻辑(EHL)100(附图 1)用于在 DTV 处理流水线内进行缩放,EHL 100 可以在视频后置处理逻辑的水平缩放器内实现。

[0100] 如上所述,在 DTV 中的视频缩放一般来说是使用多相滤波器实现的。可以将视频缩放看作内插 / 抽取过程,这一过程专注于从一组输入像素产生更大 / 更小(取决于缩放系数)数量的输出像素。当缩放是使用多相滤波器实现之时,各个输出像素是作为 N 个输入像素的函数而产生的,其中 N 是多相滤波器抽头的数量。所使用的 N 个输入像素将与输出像素以相同的位置为中心,例如,如附图 2 所示。

[0101] 假设与输入像素 $Z_{0,0}$ 的位置相对应的输出像素 160 是从像素帧 150 产生的,则计算输出像素 160 所需的输入像素可能是不可用的。如附图 2 所示,假设 N 的值等于九,要产生输出像素 160,需要九个以输出像素 160 为中心的输入像素。虽然需要九个输入像素,但是却只有五个输入像素可用,具体来说,像素 $-Z_{0,4}$ 、 $-Z_{0,3}$ 、 $-Z_{0,2}$ 、 $-Z_{0,1}$ 不可用,因为输出像素

160 在像素帧 150 的左边缘上。对于位于像素帧 150 的右边缘的输出像素 162 以及位于像素帧 150 的下边缘 158 的输出像素 164 的输出像素计算,同样会出现这个问题。

[0102] 为了计算像素帧边缘处或附近的输出像素,有三种边缘处理方法。这些方法中的每一种都专注于在像素帧 150 的外边缘 170 周围填补额外的边缘像素。外边缘 170 的尺寸 (D) 178 (178-1、178-2、178-3 和 178-4) 是按下式确定的:

$$[0103] \quad D = (N-1)/2 \quad \text{如果 } N \text{ 是奇数} \quad (1)$$

$$[0104] \quad D = N/2 \quad \text{如果 } N \text{ 是偶数} \quad (2)$$

[0105] 为了计算输出像素 160,基于零的边缘处理方法会用零值 172 填补像素帧缓冲器 150 的外边缘 170。通过填补像素帧 150 的外边缘 170,可以得到计算输出像素 160 所需的九个输入像素。还可以认识到,对输出像素 160 右边的输出像素 ($Z_{0,1}$ 、 $Z_{0,2}$ 、 $Z_{0,3}$) 进行水平缩放也要求边缘处理方法提供缺失的输入像素。其它的技术牵涉到复制,其中对帧边缘处的像素的值进行重复使用,如 174 所示。在镜像方法中,对输入像素的值进行镜像,如 176 所示。

[0106] 再次参照附图 1,按照所示的实施例,边缘处理逻辑 100 是为九抽头多相滤波器而提供的。在一种实施例中,输入像素寄存器 140 保存与多相滤波器的抽头数量 N 相应的九个输入像素。在一种实施例中,各个像素是与亮度值 (Y) 和色度值 (Cr/Cb) 相应的 8/10/12 比特量。如图所示,控制器 110 包括边缘处理模式 (EHM) 寄存器 116,用来提供边缘处理模式。同样,控制器 110 与线宽寄存器 118 相耦合,该线宽寄存器提供像素帧的当前线的线宽度。在一种实施例中,EHM 寄存器 116 和线宽寄存器 118 是由例如诸如设备驱动器或其它类似的特许软件之类的控制软件进行填充的,以提供边缘处理模式和像素帧的当前线的线宽度。

[0107] 在一种实施例中,信号 114 是提供给第二逻辑门 130 的左(上)边缘控制信号。在一种实施例中,信号 112 是右(下)边缘控制信号,该信号提供给第一逻辑门 120。基于线宽度并且取决于所检测到的边缘处理方法,按照正在进行的一种实施例,控制器 110 判断当前输出像素是否位于像素帧边界上或处于像素帧边界附近。在一种实施例中,控制器 110 使控制信号 112 和 114 有效以便指示用输入像素寄存器 140 来存储输入像素,从而能够按照视频后置处理和/或图像增强操作实现输出像素的计算。

[0108] 在一种实施例中,第一逻辑门 120 是例如多路复用器 (MUX),它接收输入像素数据信号(数据输入) 122 以及来自输入像素寄存器 140-1、140-2、140-3、140-6 和 140-8 的反馈信号 124 (124-1、124-2、124-3、124-4、124-5),以辅助像素帧右边缘的处理。在正常操作状态下,控制器 110 指示 MUX 120 用来自数据输入 122 的输入像素填充输入像素寄存器 140-1。对于线行中剩下的像素,MUX 120 简单地选取数据输入 122,一旦对各个输入像素寄存器 140 内的值进行了移位,例如,右移,则将数据输入 122 存储在输入寄存器 140-1 内。当检测到右边缘时,控制器 110 依据从寄存器 116 中检测的边缘处理模式,选取到 MUX 120 的反馈输入 124 之一。

[0109] 在一种实施例中,第二逻辑门 130 也是多路复用器 (MUX),这些多路复用器是由信号 114 控制的,信号 114 由控制器 110 驱动。如图所示,第二 MUX 130 用来从寄存器 140-6 到 140-9 选取数据。如图所示,MUX 130-4 为基于零的边缘处理取输入零,以及取得来自数据输入 122、输入像素寄存器 140-4 和输入寄存器 140-8 的输入。同样地,MUX 130-3 取输

入零,以及从输入像素寄存器 140-1、140-4 和 140-7 取得输入。MUX130-2 取输入零以及从输入像素寄存器 140-2、140-4 和 140-6 取得输入。最后, MUX 130-1 取输入零并且从输入像素寄存器 140-3、140-4 和 140-5 取得输入。

[0110] 在一种实施例中,提供第二 MUX 130 用来处理左(上)边缘的输出像素计算。对于线中剩下的像素,第二 MUX 起到简单的移位寄存器的作用。如图所示,所给出的操作是右移操作。在线的左边缘上,控制器 110 使控制信号 114 有效以便按照从 EHM 寄存器 116 中检测到的边缘处理模式指示输入像素寄存器 140-6、140-7、140-8 和 140-9 的加载。在一种实施例中,附图 1 中所示的边缘处理逻辑 100 的操作是按照表 1 中给出的伪码进行的。

[0111] 表 1

[0112] // 伪码参数 :9 抽头多相滤波器

[0113] 总是 @(每时钟边缘) 开始

[0114] // 左边缘专用代码

[0115] // 这个代码仅对第 5 个输入像素操作 1 个周期。

[0116] // 寄存器 P[0] 到 P[3] 具有基于非移位寄存器的功能。

[0117] // 其它 5 个寄存器继续起移位寄存器的作用。

[0118] if(no_pixel_shifted_in == 4)begin// 对于这个例子为 4(Number_of_taps-1/2)

[0119] If(EHA = ZERO_BASED)begin

[0120] For(i = 0 ; i < 4 ; i++)P[i] <= 0 ;// 这里完成赋零

[0121] For(i = 4 ; i < 8 ; i++)P[i] <= P[i+1] ;

[0122] P[8] <= data_in ;

[0123] End

[0124] If(EHA = REPLICATION)begin

[0125] For(i = 0 ; i < 4 ; i++)P[i] <= P[5] ;// 这里完成复制

[0126] For(i = 4 ; i < 8 ; i++)P[i] <= P[i+1] ;

[0127] P[8] <= data_in ;

[0128] End

[0129] If(EHA = MIRRORING)begin

[0130] P[0] <= data_in ;// 对 P[0] 进行镜像

[0131] For(i = 1 ; i < 4 ; ; i++)P[i] <= P[9-i] ;// 对 P[1] 到 P[3]

进行镜像

[0132] For(i = 4 ; i < 8 ; i++)P[i] <= P[i+1] ;

[0133] P[8] <= data_in ;

[0134] End

[0135] End

[0136] // 右边缘专用代码

[0137] // 这个代码进行 4 个周期的操作 (例如, 周期第 721 到 724, 如果输入具有 720 线输入像素的话)。

[0138] // 仅对 P[8] 进行改变。

```

[0139] // 其它 11 个寄存器继续起移位寄存器的作用。
[0140]         if(no_pixel_shifted_in >= line_width)begin
[0141]             For(i = 0 ;i < 8 ;i++)P[i] <= P[i+1] ;// 寄存器 0 到 7
[0142]             if(no_pixel_shifted_in == line_width)begin
[0143]                 if(EHA == ZERO_BASED)P[8] <= 0 ;
[0144]                 if(EHA == REPLICATION)P[8] <= P[8]
[0145]                 if(EHA == MIRRORING)P[8] <= P[7]]
[0146]             end
[0147]             if(no_pixel_shifted_in == line_width+2)begin
[0148]                 if(EHA == ZERO_BASED)P[8] <= 0 ;
[0149]                 if(EHA == REPLICATION)P[8] <= P[8]
[0150]                 if(EHA == MIRRORING)P[8] <= P[1] ;
[0151]             end
[0152]         End
[0153] // 最后否则子句（简单移位寄存器）
[0154]         If((edge == none)\ \
[0155]           ((edge == left)&((pixel_shifted_in != 5)\ \
[0156]           ((edge == right)&((pixel_shifted_in < bundle_width)))begin
[0157]             For(i = 0 ;i < 8 ;i++)P[i] <= P[i+1] ;
[0158]             P[8] <= date_in ;
[0159]         End
[0160]     End

```

[0161] 虽然附图 1 表示用于进行视频后置处理操作的按照针对九抽头多相滤波器的实施例的边缘处理逻辑 100,但是应当意识到,本文所介绍的关于边缘处理逻辑 100 的这些实施例包括但不局限于取决于视频后置处理操作所要求的多相滤波器的抽头数量 N 的各种不同的配置。同样地,虽然是用输入 MUX 120 和第二 MUX 130 加以图解说明的,但是可以根据需要增加额外的 MUX,以适应各种不同的边缘处理模式,来为输出像素计算提供输入像素。

[0162] 因此,按照本文所介绍的实施例,用来产生缺失的输入像素的边缘处理会在后置处理操作期间发生,而不是象在像素帧的周围填补像素来为输出像素计算消除像素帧边缘处缺失的输入像素的传统技术所进行的那样在视频后置处理操作之前进行。例如,如附图 2 所示,传统技术在提供用于显示的像素帧的后置处理之前用缺失的像素填补解码的像素帧。

[0163] 附图 3 是表示包括具有水平缩放器 (H 缩放器) 284 和垂直缩放器 (V 缩放器) 286 的后置处理逻辑 280 的芯片上系统 (SOC) 200 的框图,水平缩放器和垂直缩放器包括附图 1 的边缘处理逻辑 (EHL) 100。典型地,系统 200 包括与存储器控制器 292 和 DMA 控制器 290 相耦合的嵌入式控制处理器 202。在一种实施例中,后置处理逻辑 280 可以包括噪声修正块 282 以及水平缩放器 (H 缩放器) 284 和垂直缩放器 (V 缩放器) 286。在一种实施例中,运动图像专家组 (MPEG) 块 206 向帧缓冲器 208 提供用于后置处理逻辑 280 的解码的数字

视频帧,该后置处理逻辑 280 例如可以对所接收到的解码的数字视频帧进行缩放或图像增强。在一种实施例中,MPEG 块 206 根据从外部系统存储器 294 接收到的编码的视频信号产生解码的视频帧。

[0164] 在一种实施例中,后置处理逻辑 280 依靠控制处理器 202 或预设软件来确定边缘处理模式和当前像素帧线的线宽。在一种实施例中,控制处理器 202 是嵌入式处理器,该处理器可以用汇编语言代码进行编程。按照一种可选实施例,可以提供装置驱动器来辅助控制处理器 202 填充 EHM 寄存器 116 和线宽寄存器 118,以便指示控制器 110 填充输入像素寄存器 140,从而来实现例如变形缩放操作。一旦进行了后置处理,就向显示器 296 提供输出像素帧。在所描述的实施例中,显示器 296 可以是模拟阴极射线管 (CRT) 监视器、数字监视器(比如具有数字显示接口的平板显示器)、DTV 单元等等。

[0165] 在一种实施例中,为了符合例如后置处理逻辑 280 每秒 30 帧的带宽要求,芯片上直接存储器访问 (DMA) 控制器 290 向芯片上存储器控制器 292 发出 DMA 请求,该芯片上存储器控制器 292 将来自芯片外系统存储器 294 的像素帧提供给 DMA 控制器 290。然后将这些像素帧提供给帧缓冲器 208,比如,先入先出 (FIFO) 缓冲器等。在一种实施例中,可以将这种初始帧提供给 MPEG 块 206,该 MPEG 块 206 将所接收到的帧解码成解码的数字视频信号帧,并且经由帧缓冲器 208 将这种帧提供给后置处理逻辑 280。

[0166] 在一种实施例中,系统 200 可以并入在机顶盒或电视机内,用以提供高清晰度电视 (HDTV) 信号处理,以及可以并入在 DTV 等的显示流水线内。在这样的一种实施例中,系统存储器 294 是由机顶盒或电视机系统提供的。不过,应当认识到,可以将 SOC 202 提供在用于提供图像或视频信号显示的其它系统内,例如,家用个人计算机 (PC)、桌面或笔记本单元、手持装置等等。

[0167] 在一种实施例中,可以将 SOC 200 实现在例如需要在显示解码图像或视频帧之前进行后置处理或图像增强的计算机系统或显示装置的芯片组或者图形控制器的显示引擎(它可能需要在像素帧边界处进行边缘处理)内。在这样做的过程中,SOC 200 消除了现有技术的局限性,现有技术会按照边缘处理模式画出像素帧缓冲器的外周界,这需要在存储器内存储额外的系统开销和日期。现在将介绍实现一种或多种实施例的方法。

[0168] 操作过程

[0169] 附图 4 是表示按照一种实施例的用于视频后置处理和/或图像增强的基于硬件的边缘处理方法 300 的流程图。在处理块 310,依照视频后置处理和/或图像增强操作开始输出像素的计算。在处理块 312,判断计算输出像素所需的至少一个输入像素是否未存储在输入像素寄存器内。当检测到这一情况时,在处理块 320 中,判断未存储的输入像素是否处于像素帧边界之外。

[0170] 在一种实施例中,在处理块 350,按照检测到的边缘处理模式为未存储的输入像素产生像素值。在一种实施例中,输入像素的产生是由控制器进行的,例如,由附图 1 的边缘处理逻辑 100 的控制器 110 进行。一旦产生了像素值,则将该像素值加载到输入像素寄存器内。如果计算输出像素所需的多个输入像素都未存储,并且处于像素帧边界之外,则重复进行处理块 350 和 352,直到计算输出像素所需的 N 个输入像素都得到存储。另外,在处理块 360,将未存储的输入像素从像素帧缓冲器加载到输入像素寄存器内。在处理块 370,计算输出像素。在处理块 380 中,重复进行处理块 310-360,直到计算出所有输出像素。

[0171] 附图 5 是表示视频缩放器（例如，如附图 3 所示）内基于硬件的边缘处理方法 330 的流程图。在处理块 332 中，确定像素加载计数。在一种实施例中，像素加载计数是依据线宽寄存器（例如，附图 1 的边缘处理逻辑 100 的线宽寄存器 118）来进行跟踪的。在处理块 334 中，如果像素加载计数等于规定的加载计数，则检测左像素帧边缘。在一种实施例中，所规定的加载计数是按照等式 (1) 和 (2) 确定的。因此，按照参照附图 2 介绍的实施例，所规定的像素加载计数等于四。

[0172] 当处理块 334 的条件评定为真时，在处理块 336 中，在一个或多个相邻输入像素寄存器之间对输入像素进行移位。在处理块 338 中，将来自像素帧缓冲器的输入像素存储在最低有效位输入像素寄存器中，比如，存储在附图 1 的输入像素寄存器 140-1 中。一旦加载了，在处理块 340 中，识别至少一个空的输入像素寄存器。提供输入像素寄存器的识别是为了加载针对处于像素帧边界之外的缺失输入像素而产生的值，以按照视频后置处理或图像增强操作计算输出像素。

[0173] 在处理块 342 中，判断像素加载计数是否大于或等于当前像素帧线的线宽。当这种条件评定为真时，在处理块 344 中，在各个相邻输入像素寄存器之间对输入像素值进行移位，以空出一个输入像素寄存器。输入像素寄存器的移位空出了一个输入像素寄存器，以便加载至少一个为处于右像素帧边界之外的输入像素产生的值。随后，控制流程分支到附图 4 的处理块 350。否则，控制流程分支到处理块 360，从而将计算输出像素所需的各个输入像素包含在输入像素寄存器或像素帧缓冲器内。

[0174] 在一种实施例中，未存储输入像素的识别是通过确定计算输出像素所需的输入像素来进行的。一旦得到确定，在一种实施例中，附图 1 的控制器 110 可以判断计算输出像素所需的输入像素中的任何一个是否没有存储在输入像素寄存器内。当确是这种情况时，检测至少一个未经存储的、计算输出像素所需的输入像素。随后，如果未存储的输入像素处于像素帧边界之外，则控制器可以按照所检测到的边缘处理模式产生像素值，而不是使用传统的技术，传统技术在视频后置处理操作之前填补处于像素帧边界之外的输入像素，以实现边缘处理。

[0175] 附图 6 是表示对使用所公开的技术的设计方案的仿真、模拟和制造的各种表示或格式的框图。代表设计方案的数据可以代表多种方式的设计方案。首先，由于在仿真中很有用，因此硬件可以使用硬件描述语言或者其它功能描述语言来表示，这种语言基本上给出了所设计的硬件应该如何实现的计算机化模型。可以将硬件模型 410 存储在存储介质 400 中，比如计算机存储器，从而可以使用仿真软件 420 对该模型进行仿真，该仿真软件 420 对硬件模型 410 应用特定的测试组，来判定它是否确实起到了所预期的功能。在某些实施例中，不将仿真软件记录、采集或包含在介质中。

[0176] 按照设计方案的任何一种表示，可以将数据存储在任何形式的机器可读介质中。经过调制或采用其它方式产生的用来传输这种信息的光波或电波 460、存储器 450 或者磁性或光学存储装置（比如盘）可以是机器可读介质。这些介质中的任何一种可以携带设计信息。术语“携带”（例如，机器可读介质携带信息）因此涵盖了存储在存储装置上的信息或编码成载波或调制在载波上的信息。描述设计或设计的细节的数据位集（当嵌入在机器可读介质（比如载波或存储介质）中时）是可以封入其内外的物品，或者由他人用来进行进一步设计或制造。

[0177] 可供选用的其它实施例

[0178] 应意识到,对于其它的实施例,可以使用不同的系统配置。例如,虽然将系统 200 配置为芯片上系统,但是对于其它实施例,单处理器或多处理器系统(在这种情况下,一个或多个处理器可以类似于上面介绍的处理器 202 那样配置和操作)可以受益于各种实施例的基于硬件的边缘处理。此外,对于其它的实施例,可以使用不同类型的系统或不同类型的计算机系统,例如,服务器、工作站、台式计算机系统、游戏系统、嵌入式计算机系统、刀片服务器等。

[0179] 虽然公开了具体实施例和最佳模式,但是可以在保持在所附权利要求定义的本发明的实施例的范围内的同时,对所公开的实施例进行修改和改变。

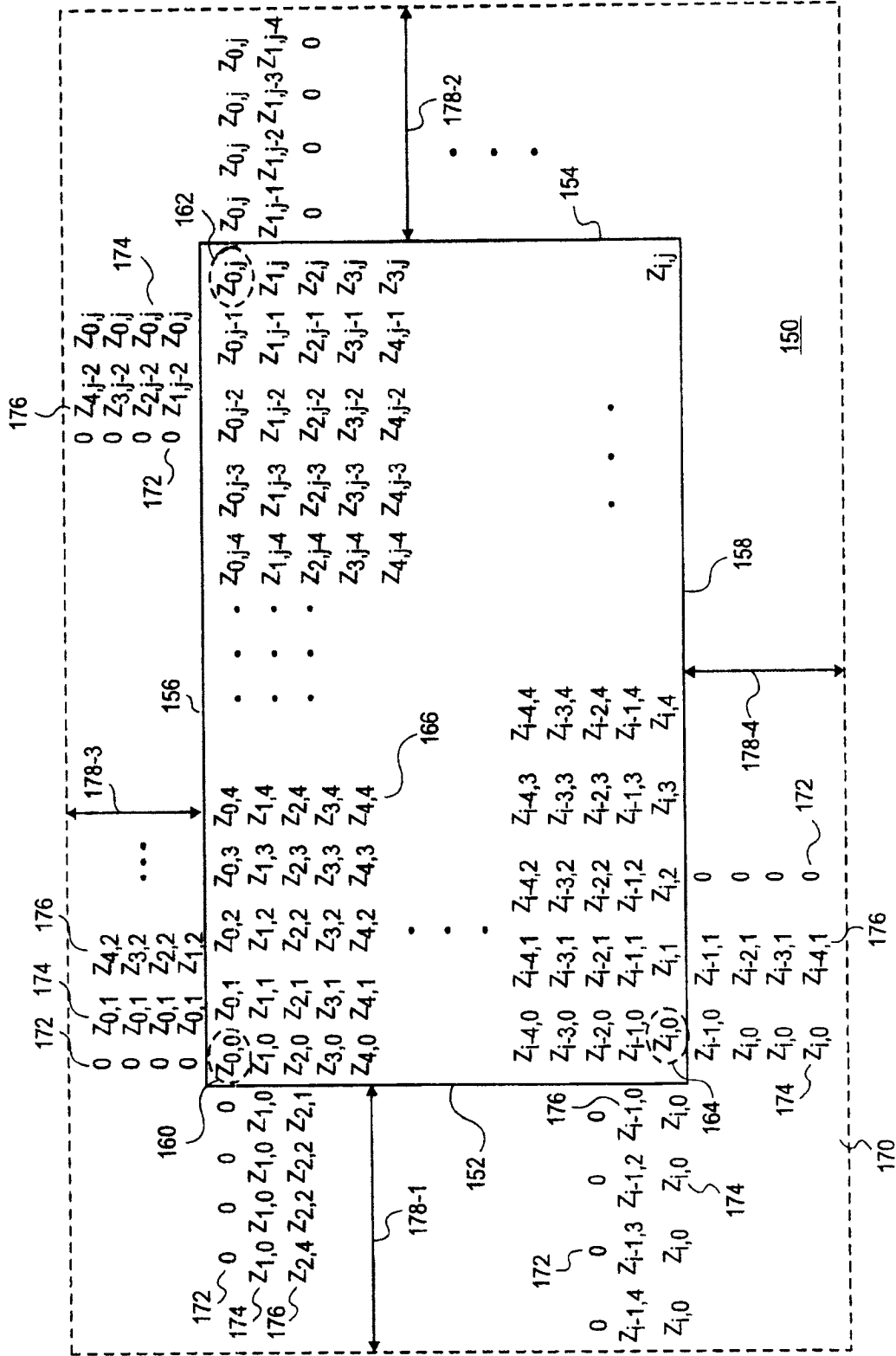


图 2

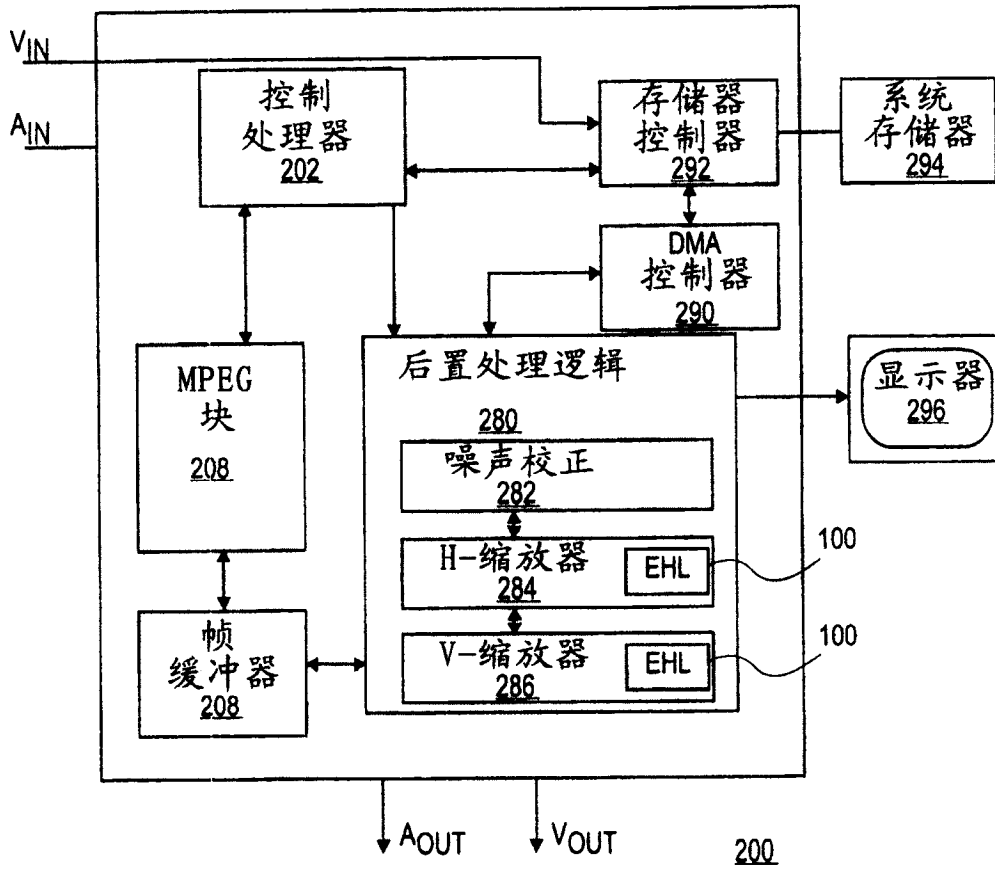


图 3

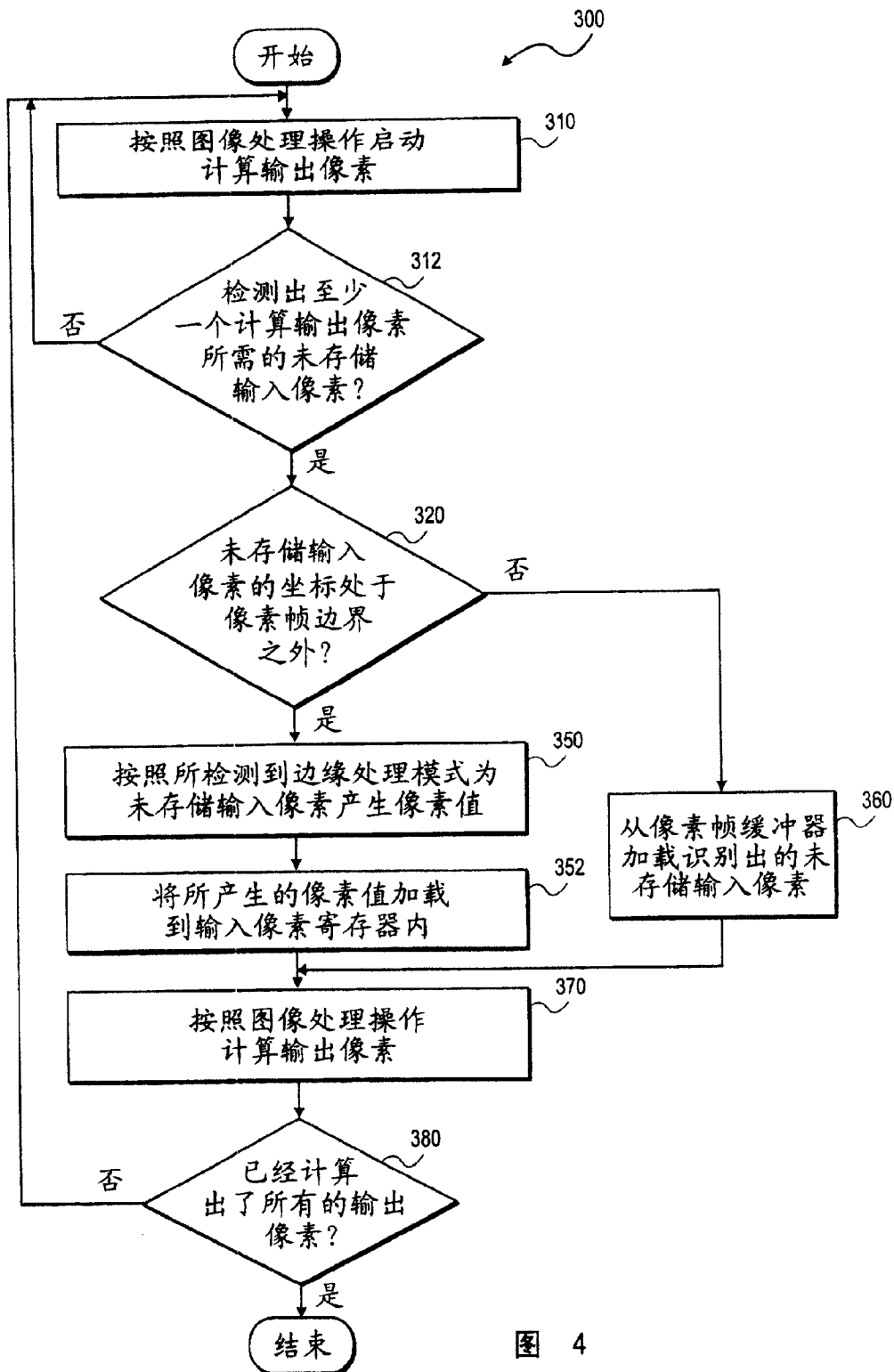


图 4

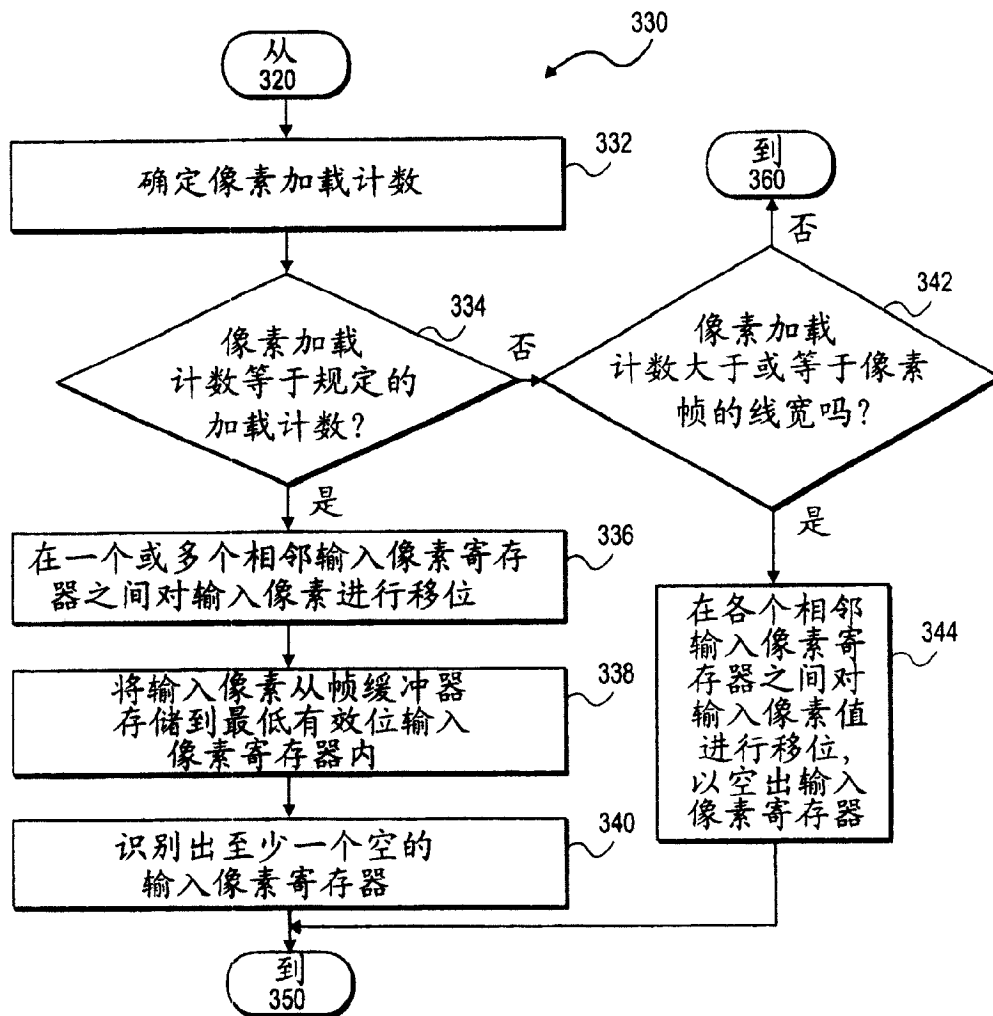


图 5

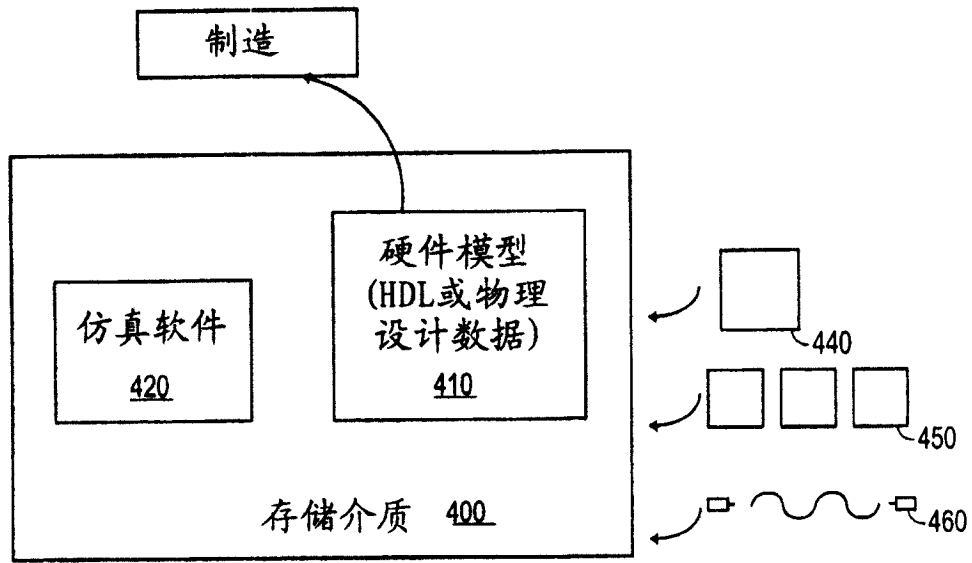


图 6