



(12)发明专利

(10)授权公告号 CN 105493151 B

(45)授权公告日 2018.09.25

(21)申请号 201480047503.2

(22)申请日 2014.07.02

(65)同一申请的已公布的文献号

申请公布号 CN 105493151 A

(43)申请公布日 2016.04.13

(30)优先权数据

61/871,260 2013.08.28 US

14/321,409 2014.07.01 US

(85)PCT国际申请进入国家阶段日

2016.02.26

(86)PCT国际申请的申请数据

PCT/US2014/045309 2014.07.02

(87)PCT国际申请的公布数据

W02015/030933 EN 2015.03.05

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 维尼特·戈尔 尤萨梅·杰伊兰

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

G06T 11/20(2006.01)

G06T 11/40(2006.01)

G06T 15/04(2006.01)

(56)对比文件

US 2011/0285724 A1,2011.11.24,说明书第18-28,61-66,99-106,126段.

US 8094157 B1,2012.01.10,说明书第15列第19-30行、图7A-8B.

CN 101802874 A,2010.08.11,全文.

Mark J. Kilgard 等.GPU-accelerated Path Rendering.《ACM TRANSACTIONS ON GRAPHICS》.2012,第31卷(第6期),1-10.

审查员 王爽

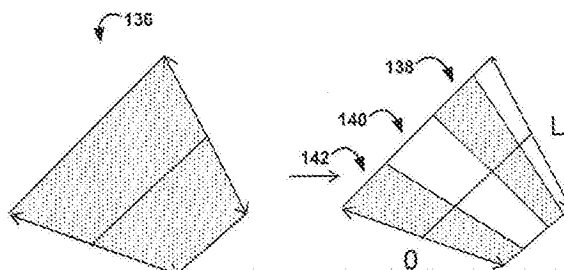
权利要求书4页 说明书27页 附图12页

(54)发明名称

图形处理中的像素化求和长度

(57)摘要

在实例中,再现图形数据包含:使用图形处理单元GPU确定虚线的多个经排序片段中的当前片段的纹理偏移,其中所述多个经排序片段中的所述当前片段的所述纹理偏移基于在次序中先于所述当前片段的片段的长度累加;及像素着色所述当前片段,包含应用所述纹理偏移以确定所述当前片段的位置。



1. 一种再现图形数据的方法,所述方法包括:

使用图形处理单元GPU对先于虚线的多个经排序片段中的当前片段的片段进行像素着色;

使用所述图形处理单元GPU确定所述虚线的所述多个经排序片段中的所述当前片段的纹理偏移,其中所述多个经排序片段的所述当前片段的所述纹理偏移是基于在次序中先于所述当前片段的所述片段的长度累加;

应用所述纹理偏移以确定所述当前片段的位置;及

在经确定的位置对所述当前片段进行像素着色。

2. 根据权利要求1所述的方法,其中所述多个片段包含一或多个可见片段和一或多个不可见片段,所述方法进一步包括:

基于所述当前片段的所确定的位置来确定所述当前片段是否是可见片段;及

基于所述当前片段是可见片段而保持所述当前片段或基于所述当前片段不是可见片段而舍弃所述当前片段。

3. 根据权利要求1所述的方法,其进一步包括:

在确定所述当前片段的所述纹理偏移之前,几何形状着色所述虚线以形成所述多个经排序片段;及

确定所述当前片段的长度值,以使得基于所述长度累加确定所述纹理偏移包括基于所述长度值确定所述纹理偏移。

4. 根据权利要求3所述的方法,其中确定所述长度值包括产生指定在所述次序中的先前片段的长度的线长标量值。

5. 根据权利要求1所述的方法,其进一步包括:

光栅化所述多个经排序片段中的每一所述片段,

其中应用所述纹理偏移包括在所述当前片段已光栅化之后将所述纹理偏移应用到所述光栅化当前片段。

6. 根据权利要求1所述的方法,其中应用所述纹理偏移包括确定指示所述位置的用于所述当前片段的纹理坐标值。

7. 根据权利要求1所述的方法,其中所述像素着色包含在针对所述虚线的包含描画所述虚线的路径再现过程中。

8. 根据权利要求1所述的方法,其进一步包括:在所述片段的几何形状着色期间确定所述片段的所述次序,以使得所述片段的所述次序是基元次序。

9. 一种用于再现图形数据的设备,所述设备包括图形处理单元GPU,所述图形处理单元经配置以:

对先于虚线的多个经排序片段中的当前片段的片段进行像素着色;

确定所述虚线的所述多个经排序片段中的所述当前片段的纹理偏移,其中所述多个经排序片段中的所述当前片段的所述纹理偏移是基于在次序中先于所述当前片段的所述片段的长度的累加;

应用所述纹理偏移以确定所述当前片段的位置;及

在经确定的位置对所述当前片段进行像素着色。

10. 根据权利要求9所述的设备,其中所述多个片段包含一或多个可见片段和一或多个

不可见片段,所述GPU经进一步配置以:

基于所述当前片段的所确定的位置来确定所述当前片段是否是可见片段;及

基于所述当前片段是可见片段而保持所述当前片段或基于所述当前片段不是可见片段而舍弃所述当前片段。

11.根据权利要求9所述的设备,其中所述GPU进一步经配置以:

在确定所述当前片段的所述纹理偏移之前,几何形状着色所述虚线以形成所述多个经排序片段;及

确定所述当前片段的长度值,以使得基于所述长度累加确定所述纹理偏移包括基于所述长度值确定所述纹理偏移。

12.根据权利要求11所述的设备,其中为了确定所述长度值,所述GPU经配置以产生指定在所述次序中的先前片段的长度的线长标量值。

13.根据权利要求9所述的设备,其中所述GPU进一步经配置以:

光栅化所述多个经排序片段中的每一所述片段,

其中为了应用所述纹理偏移,所述GPU经配置以在所述当前片段已光栅化之后将所述纹理偏移应用到所述光栅化当前片段。

14.根据权利要求9所述的设备,其中为了应用所述纹理偏移,所述GPU经配置以确定指示所述位置的用于所述当前片段的纹理坐标值。

15.根据权利要求9所述的设备,其中所述像素着色包含在针对所述虚线的包含描画所述虚线的路径再现过程中。

16.根据权利要求9所述的设备,其中所述GPU经进一步配置以在所述片段的几何形状着色期间确定所述片段的所述次序,以使得所述片段的所述次序是基元次序。

17.一种用于再现图形数据的设备,所述设备包括:

用于使用图形处理单元GPU对先于虚线的多个经排序片段中的当前片段的片段进行像素着色的装置;

用于使用所述图形处理单元GPU确定所述虚线的所述多个经排序片段中的所述当前片段的纹理偏移的装置,其中所述多个经排序片段的所述当前片段的所述纹理偏移是基于在次序中先于所述当前片段的所述片段的长度累加;

用于应用所述纹理偏移以确定所述当前片段的位置的装置;及

用于在经确定的所述位置对所述当前片段进行像素着色的装置。

18.根据权利要求17所述的设备,其中所述多个片段包含一或多个可见片段和一或多个不可见片段,所述设备进一步包括:

用于基于所述当前片段的所确定的位置来确定所述当前片段是否是可见片段的装置;及

用于基于所述当前片段是可见片段而保持所述当前片段或基于所述当前片段不是可见片段而舍弃所述当前片段的装置。

19.根据权利要求17所述的设备,其进一步包括:

在确定所述当前片段的所述纹理偏移之前,用于几何形状着色所述虚线以形成所述多个经排序片段的装置;及

用于确定所述当前片段的长度值,以使得基于所述长度的累加确定所述纹理偏移包括

基于所述长度值确定所述纹理偏移的装置。

20. 根据权利要求19所述的设备,其中所述用于确定所述长度值的装置包括用于产生指定在所述次序中的先前片段的长度的线长标量值的装置。

21. 根据权利要求17所述的设备,其进一步包括:

用于光栅化所述多个经排序片段中的每一所述片段的装置,

其中所述用于应用所述纹理偏移的装置包括用于在所述当前片段已光栅化之后将所述纹理偏移应用到所述光栅化当前片段的装置。

22. 根据权利要求17所述的设备,其中所述用于应用所述纹理偏移的装置包括用于确定指示所述位置的用于所述当前片段的纹理坐标值的装置。

23. 根据权利要求17所述的设备,其进一步包括:用于在所述片段的几何形状着色期间确定所述片段的所述次序以使得所述片段的所述次序是基元次序的装置。

24. 一种非暂时性计算机可读媒体,其上存储有在执行时引起图形处理单元GPU经配置以进行以下操作的指令:

使用图形处理单元GPU对先于虚线的多个经排序片段中的当前片段的片段进行像素着色;

确定所述虚线的所述多个经排序片段中的所述当前片段的纹理偏移,其中所述多个经排序片段中的所述当前片段的所述纹理偏移是基于在次序中先于所述当前片段的片段的长度的累加;

应用所述纹理偏移以确定所述当前片段的位置;及

在经确定的所述位置对所述当前片段进行像素着色。

25. 根据权利要求24所述的非暂时性计算机可读媒体,其中所述多个片段包含一或多个可见片段和一或多个不可见片段,所述指令进一步引起所述GPU进行以下操作:

基于所述当前片段的所确定的位置来确定所述当前片段是否是可见片段;及

基于所述当前片段是可见片段而保持所述当前片段或基于所述当前片段不是可见片段而舍弃所述当前片段。

26. 根据权利要求24所述的非暂时性计算机可读媒体,其中所述指令进一步引起所述GPU进行以下操作:

在确定所述当前片段的所述纹理偏移之前,几何形状着色所述虚线以形成所述多个经排序片段;及

确定所述当前片段的长度值,以使得基于所述长度累加确定所述纹理偏移包括基于所述长度值确定所述纹理偏移。

27. 根据权利要求26所述的非暂时性计算机可读媒体,其中为了确定所述长度值,所述指令引起所述GPU产生指定在所述次序中的先前片段的长度的线长标量值。

28. 根据权利要求24所述的非暂时性计算机可读媒体,其中所述指令进一步引起所述GPU进行以下操作:

光栅化所述多个经排序片段中的每一所述片段,

其中为了应用所述纹理偏移,所述指令引起所述GPU在所述当前片段已光栅化之后将所述纹理偏移应用到所述光栅化当前片段。

29. 根据权利要求24所述的非暂时性计算机可读媒体,其中为了应用所述纹理偏移,所

述指令引起所述GPU确定指示所述位置的用于所述当前片段的纹理坐标值。

30. 根据权利要求24所述的非暂时性计算机可读媒体,其中所述指令进一步引起所述GPU在所述片段的几何形状着色期间确定所述片段的所述次序,以使得所述片段的所述次序是基元次序。

图形处理中的像素化求和长度

[0001] 本申请案主张2013年8月28日申请的第61/871,260号美国临时申请案的权利要求,所述申请案的全部内容在此以引用的方式并入。

技术领域

[0002] 本发明涉及图形处理,且更具体地说涉及用于路径再现的技术。

背景技术

[0003] 路径再现可指代二维(2D)向量图形路径(在本文中替代地被称作“路径”)的再现,所述路径中的每一者可包含一或多个路径片段。当路径包含两个或两个以上路径片段时,个别路径片段可具有相同类型或不同类型。路径片段的类型可包含(例如)线、椭圆形弧、二次贝塞尔曲线和三次贝塞尔曲线。在一些实例中,路径片段类型可根据例如开放向量图形(OpenVG) API等标准向量图形应用程序编程接口(API)来定义。

[0004] 路径再现可在中央处理单元(CPU)中实施。然而,此方法可为CPU密集的,且因此可限制可用于其它CPU任务的CPU处理循环的量。此外,在一些情况下,相对大量的数据可能需要传送到图形处理单元(GPU)以用所要的细节水平再现路径片段。相对大量的数据在存储数据时可消耗大量存储器存储空间,且在将数据传送到GPU时可消耗大量的存储器带宽。

发明内容

[0005] 本发明包含用于使用路径填充和划短划线产生图形数据的技术。举例来说,在填充路径时,根据本发明的各方面,GPU可能以一速率执行模板印刷(stenciling)操作,所述速率不同于为所再现数据(被称作再现目标)分配存储器的速率。也就是说,可独立于用于存储所再现数据的再现目标参数指定用于执行模板操作的模板参数。

[0006] 另外,关于划短划线,根据本发明的各方面,GPU可在单一再现遍次中确定短划线特性并执行划短划线。举例来说,所述GPU可在确定片段时计算所述片段中的每一者的长度并应用所述长度信息来确定每一短划线片段的起始位置(例如,纹理坐标)。

[0007] 在实例中,一种再现图形数据的方法包含:确定模板参数,所述模板参数指示用于确定图像的路径的每一图形保真像素的覆盖值的取样速率;单独地根据所述模板参数确定再现目标参数,所述再现目标参数指示用于所述路径的每一图形保真像素的存储器分配;及使用模板参数和再现目标参数再现路径。

[0008] 在另一实例中,一种用于再现图形的设备包含图形处理单元(GPU),所述GPU经配置以确定模板参数,所述模板参数指示用于确定图像的路径的每一图形保真像素的覆盖值的取样速率;单独地根据模板参数确定再现目标参数,所述再现目标参数指示用于路径的每一图形保真像素的存储器分配;及使用模板参数和再现目标参数再现路径。

[0009] 在另一实例中,一种用于再现图形数据的设备包含:用于确定模板参数的装置,所述模板参数指示用于确定图像的路径的每一图形保真像素的覆盖值的取样速率;用于单独地根据模板参数确定再现目标参数的装置,所述再现目标参数指示用于路径的每一图形保

真像素的存储器分配;及用于使用模板参数和再现目标参数再现路径的装置。

[0010] 在另一实例中,一种非暂时性计算机可读媒体上存储有在执行时引起图形处理单元(GPU)进行以下操作的指令:确定模板参数,所述模板参数指示用于确定图像的路径的每一图形保真像素的覆盖值的取样速率;单独地根据所述模板参数确定再现目标参数,所述再现目标参数指示用于路径的每一图形保真像素的存储器分配;及使用模板参数和再现目标参数再现路径。

[0011] 在另一实例中,一种再现图形数据的方法包含:使用图形处理单元(GPU)确定虚线的多个经排序片段中的当前片段的纹理偏移,其中所述多个经排序片段中的当前片段的纹理偏移基于次序先于所述当前片段的片段的长度累加;及像素着色(pixel shading)所述当前片段,包含应用纹理偏移以确定所述当前片段的位置。

[0012] 在另一实例中,一种用于再现图形数据的设备包含图形处理单元(GPU),所述GPU经配置以进行以下操作:确定虚线的多个经排序片段中的当前片段的纹理偏移,其中所述多个经排序片段中的当前片段的纹理偏移基于次序先于当前片段的片段的长度累加;及像素着色当前片段,包含应用纹理偏移以确定当前片段的位置。

[0013] 在另一实例中,一种再现图形数据的设备包含:用于使用图形处理单元(GPU)确定虚线的多个经排序片段中的当前片段的纹理偏移的装置,其中所述多个经排序片段中的当前片段的纹理偏移基于次序先于所述当前片段的片段的长度累加;及用于像素着色所述当前片段的装置,所述像素着色包含应用纹理偏移以确定所述当前片段的位置。

[0014] 在另一实例中,一种非暂时性计算机可读媒体上存储有在执行时引起图形处理单元(GPU)经配置以进行以下操作的指令:确定虚线的多个经排序片段中的当前片段的纹理偏移,其中所述多个经排序片段中的当前片段的纹理偏移基于次序先于所述当前片段的片段的长度累加;及像素着色所述当前片段,包含应用纹理偏移以确定当前片段的位置。

[0015] 在附图和下文描述中陈述本发明的一或多个实例的细节。本发明的其它特征、目标和优点将从所述描述和图式以及权利要求书显而易见。

附图说明

[0016] 图1为说明可用于实施本发明的技术的实例计算装置的框图。

[0017] 图2是进一步详细说明图1中的计算装置的CPU、GPU和存储器的框图。

[0018] 图3是说明可用于执行本发明的技术的实例图形管线的概念图。

[0019] 图4是待再现的实例路径的图。

[0020] 图5A至5C是说明用于图4中所展示的路径的填充操作的实例序列的图。

[0021] 图6是说明模板印刷操作的概念图。

[0022] 图7是说明根据本发明的各方面的实例填充操作的概念图。

[0023] 图8是说明根据本发明的各方面的再现期间的带宽的曲线图。

[0024] 图9A至9D是说明用于图4中所展示的路径的实例划短划线操作的一系列图。

[0025] 图10是说明根据本发明的各方面的用于再现图形数据的实例过程的流程图。

[0026] 图11是说明根据本发明的各方面的用于划短划线的实例过程的流程图。

具体实施方式

[0027] 本发明是针对用于执行基于GPU的路径再现的技术。路径再现可指代二维 (2D) 向量图形路径 (在本文中替代地被称作“路径”) 的再现, 所述路径中的每一者可包含一或多个路径片段。当路径包含两个或两个以上路径片段时, 个别路径片段可具有相同类型或不同类型。路径片段的类型可包含 (例如) 线、椭圆形弧、二次贝塞尔曲线和三次贝塞尔曲线。在一些实例中, 路径片段类型可根据例如开放向量图形 (OpenVG) API 等标准向量图形应用程序编程接口 (API) 来定义。

[0028] GPU通常实施三维 (3D) 图形管线, 其经设计以与一或多个3D图形API顺应。因为当今在使用的流行的3D图形API并不需要顺应装置支持路径再现命令, 所以现代的GPU常常提供极少甚至不提供用于路径再现命令的硬件加速。举例来说, 现代的GPU中实施的典型3D图形管线可包含光栅化器, 其经设计以光栅化低阶非弯曲3D图形基元 (例如点、线和三角形), 但不能够直接再现弯曲路径再现基元 (例如, 椭圆形弧和贝塞尔曲线)。

[0029] 用于路径再现的一个方法可涉及使用3D GPU管线来提供部分GPU硬件加速以用于路径再现命令的执行。此方法涉及以中央处理单元 (CPU) 预处理路径片段以便将所述路径片段转换为可由GPU光栅化的一或多个低阶非弯曲3D图形基元。举例来说, CPU可将弯曲路径片段 (例如, 椭圆弧或贝塞尔曲线) 曲面细分为接近所述路径片段的曲率的一组相对较小三角形, 且可引起使用GPU再现所述三角形集合。然而, 此方法可为CPU密集的, 且因此可限制可用于其它CPU任务的CPU处理循环的量。此外, 在一些情况下, 可需要相对较大的三角形来以所要细节水平再现路径片段。相对较大的三角形在存储数据时可消耗大量存储器存储空间, 且在将数据传送到GPU时可消耗大量的存储器带宽。

[0030] 用于提供部分到全部GPU硬件加速以用于路径再现命令的执行的另一方法可涉及修改GPU的架构以支持专用硬件加速路径再现管线。然而, 因为流行的3D图形API (例如, Microsoft® DirectX 11 (DX) API) 并不需要GPU架构包含专用路径再现管线, 所以此方法并不产生将保证由与特定3D图形API (例如, DX 11 API) 顺应的全部GPU支持的跨平台硬件加速路径再现解决方案。

[0031] 在一些实例中, 可使用基于GPU的路径再现技术, 其中所述GPU经配置以将所接收路径片段曲面细分为多个线段及使用3D图形管线再现经曲面细分的线段。通过使用GPU将路径片段曲面细分为线段, 预处理路径片段的负担从CPU松开, 进而释放处理资源以用于其它CPU任务。此外, 在一些实例中, GPU可利用高度平行的现代GPU曲面细分架构来执行曲面细分操作, 其在一些实例中可允许GPU以比CPU更高效的方式再现路径片段。另外, 因为曲面细分在GPU中发生而不是在CPU中发生, 所以众多的经曲面细分基元不需要存储于系统存储器中且不需要从CPU传递到GPU, 进而减少用于路径再现需要的存储器占据面积以及用于路径再现需要的存储器带宽。

[0032] 在一些实例中, GPU可使用多样本图形保真 (multi-sample anti-aliasing; MSAA) 技术来执行图形保真。举例来说, 像素经均匀地染色且始终具有相同形状, 这可导致所再现的图像的线在外观上变为锯齿状。通过MSAA, 可对单个像素产生多个样本。可接着组合 (例如, 平均化) 所述样本以确定最终像素值。

[0033] 因此, 在一些情况下, GPU可能以比所显示的解析度高的解析度再现图像。GPU可接着在显示之前将图像下取样到适当大小。可使结果沿着对象的边缘从一行像素到另一行像素较平滑地转变。可使用因数4、8、16或其它值来执行MSAA。在执行MSAA时, 所述GPU可能以

MSAA速率取样深度和模板操作、以MSAA速率分配存储器并以MSAA速率光栅化像素(例如, 16x MSAA包含每像素16x深度/模板样本和每像素16x存储器分配以及每像素16x光栅化样本)。

[0034] 一般来说,“目标”可指代所再现的像素分配的存储器。通常,关于经图形保真的图像,用于执行例如光栅化等的图形操作的取样速率和用于所再现的目标的存储器分配彼此对应,例如,1:1。因此,在出于说明目的的实例中,GPU可使用每像素16x的取样速率用于光栅化并分配存储器来存储每像素16个样本。然而,在目标独立光栅化(TIR)中,可独立于为所再现的图像分配的存储器指定光栅化过程的取样速率。举例来说,每像素四个样本的取样速率可用于光栅化,而用于存储图像的像素的色彩的存储器分配可为在所述图像中每像素一个色彩。

[0035] 虽然TIR允许独立于为所述目标分配的存储器指定光栅化速率,但其它再现操作可保持相互关联。举例来说,深度和模板印刷操作(如下文更详细描述)可通常与再现目标相关联。因此,每像素指定单一再现目标,还可以相同速率(即,1x取样速率)执行深度和模板操作。

[0036] 根据本发明的各方面,GPU可在模板印刷操作中充分利用TIR概念。举例来说,GPU可能以比分配用于特定像素的存储器量高的速率执行模板印刷。也就是说,在超取样模板操作(例如,每一像素将具有16个样本)的过程中,GPU可通过计算每像素的覆盖值来进行再现,所述像素的(经超取样像素的)取样基于所述覆盖值通过模板测试(例如,在特定路径内部)。为了性能改良,再现目标可为1x取样的,而模板可为16x取样的。GPU可基于每个取样模板测试为每一像素指派一覆盖值。独立于目标指定模板印刷取样速率和指定光栅化速率在本文中可被称作经模板印刷TIR。

[0037] 可在路径再现期间应用经模板印刷TIR过程。举例来说,在路径再现时,GPU可通常执行以下实例功能来填充路径:将路径曲面细分为线段,将线段连接到枢转点以形成三角形,并将三角形再现到模板缓冲器(在一些情况下,包含执行深度测试),其中所述模板缓冲器指示所述图像的可见像素。所述填充过程的接下来的和可能的最终步骤为在启用模板测试的情况下再现限界框并将模板缓冲器的内容复制到帧缓冲器。此方法需要两个再现遍次,例如,用以再现限界框的一个遍次和用以再现纹理的一个遍次。

[0038] 根据本发明的各方面,GPU可在单一再现遍次中填充路径,而无需预处理限界框。举例来说,在一些实例中,GPU可并入有限限界框单元,所述限界框单元可包含在光栅化器级处使用的硬件。举例来说,当将基元再现到模板缓冲器时,限界框单元可跟踪给定路径的最外坐标点(例如,顶部极值、底部极值、左极值和右极值)。在最外坐标点指示路径的最外边界的意义上来说,所述点也可被称作最大边界点。在完成模板印刷后,限界框单元已基于最外坐标点确定限界矩形。

[0039] 在以上实例中,由于路径的基元经再现到模板缓冲器(所述基元仅影响模板),所以GPU并不着色所述基元。所述GPU可接着使用模板缓冲器再现限界框以指派色彩。根据本发明的各方面,在执行模板印刷和确定限界框后并不需要另一绘图调用。确切地说,所述GPU在单一遍次中使用经模板印刷TIR光栅化限界框。

[0040] 以此方式,所述GPU可在单一遍次中进行填充(例如,执行模板和色彩操作),而不是(例如)在GPU处确定基元、在CPU处确定限界框并在GPU上执行色彩操作。也就是说,本发

明的技术包括限界框最佳化,其允许GPU确定限界框(例如,在曲面细分期间,GPU可接着将所述限界框主动分发到光栅化器)以使得可在单一遍次中执行模板和色彩。

[0041] 本发明的其它方面涉及划短划线(例如虚线)。举例来说,在用短划线划出所描画的路径时,所述GPU可依序(被称作片段次序)再现短划线片段,并且可产生一个片段,先前片段在所述片段处中断。也就是说,所述GPU仅在着色先前片段后确定短划线图案中的每一片段的起始位置。因为需要处理短划线的每一区段的位置以便确定正确的起始位置,所以这些计算可降低图形处理的并行度并且需要执行超过一个再现遍次。

[0042] 根据本发明的各方面,所述GPU可在单一遍次(例如单一再现遍次)中确定短划线特性并执行划短划线。举例来说,(例如)在几何形状着色期间,所述GPU可在确定片段时计算所述片段中的每一者的长度。也就是说,所述GPU可累加片段(例如,片段次序在当前片段之前的片段)的长度,以确定当前片段的起始位置。此长度累加在本文中可被称作“前缀长度”或“前缀求和长度”。所述GPU还可确定线的总长度。

[0043] 在出于说明目的的实例中,所述GPU可确定虚线的第一片段。所述GPU还可确定虚线的第二片段。所述GPU可基于先前片段的前缀求和长度确定第二片段的起始位置。也就是说,所述GPU可基于先前片段(即,第一片段)的长度的累加确定第二片段的起始位置。所述GPU还可确定虚线的第三片段。再次,所述GPU可基于先前片段的前缀求和长度确定第三片段的起始位置。也就是说,所述GPU可基于先前片段(即,第一片段和第二片段)的长度的累加确定第三片段的起始位置。所述GPU可以这种方式继续直至已确定线的片段中的每一者的起始位置为止。

[0044] 在一些实例中,所述虚线可包含可见片段和不可见片段。举例来说,GPU 12可确定可见的片段(例如,线的短划线)的色彩并舍弃不可见的片段(例如,染色短划线之间的虚线的部分)。GPU 12可基于被着色的片段的位置确定是否保留一片段(其在本文中可互换地被称作分段,例如,在像素着色期间)。关于上文作为实例所描述的三个片段,假定虚线的第一及第三片段,并且第二片段为未染色的分隔第一及第三片段的不可见片段。GPU 12可在像素着色期间基于片段的位置确定是保留(例如,用色彩着色)还是舍弃片段。也就是说,GPU 12可基于第一片段的位置确定保留第一片段,基于第二片段的位置确定舍弃第二片段,并基于第三片段的位置确定保留第三片段。

[0045] 根据本发明的各方面,所述GPU可在再现期间应用每一片段的前缀求和长度来作为纹理偏移。举例来说,在光栅化片段后,所述GPU可将所述片段的前缀求和长度的值馈入到像素着色器来作为纹理偏移值。所述GPU可将纹理偏移应用到线的开始的纹理坐标来确定被着色的片段的位置。

[0046] 图1是说明可用于实施本发明的技术的实例计算装置2的框图。计算装置2可包括个人计算机、桌上型计算机、膝上型计算机、计算机工作站、视频游戏平台或控制台、无线通信装置(例如,移动电话、蜂窝式电话、卫星电话和/或移动电话手持机)、陆线电话、因特网电话、手持式装置(例如,便携式视频游戏装置或个人数字助理(PDA))、个人音乐播放器、视频播放器、显示装置、电视、电视机顶盒、服务器、中间网络装置、主机计算机、或处理及/或显示图形数据的任何其它类型的装置。

[0047] 如图1的实例中所示,计算装置2包含用户接口4、CPU 6、存储器控制器8、存储器10、图形处理单元(GPU) 12、GPU高速缓冲存储器14、显示接口16、显示器18和总线20。用户接

口4、CPU 6、存储器控制器8、GPU 12及显示接口16可使用总线20彼此通信。应注意,图1中展示的不同组件之间的总线和通信接口的特定配置仅是示范性的,并且具有相同或不同组件的计算装置和/或其它图形处理系统的其它配置可用于实施本发明的技术。

[0048] CPU 6可包括控制计算装置2的操作的通用或专用处理器。用户可将输入提供到计算装置2以引起CPU 6执行一或多个软件应用程序。在CPU 6上执行的软件应用程序可包含(例如)操作系统、字处理器应用程序、电子邮件应用程序、电子数据表应用程序、媒体播放器应用程序、视频游戏应用程序、图形用户接口应用程序或另一程序。用户可经由一或多个输入装置(未展示)(例如,键盘、鼠标、麦克风、触摸垫或经由用户接口4耦合到计算装置2的另一输入装置)将输入提供到计算装置2。

[0049] 在CPU 6上执行的软件应用程序可包含指示GPU 12引起图形数据到显示器18的再现的一或多个图形再现指令。在一些实例中,软件指令可符合图形应用程序编程接口(API),例如开放图形库(OpenGL[®])API、开放图形库嵌入系统(OpenGL ES)API、Direct3D API、DirectX API、RenderMan API、WebGL API或任何其它公众或专有标准图形API。为了处理图形再现指令,CPU 6可将一或多个图形再现命令发布到GPU 12以引起GPU 12执行图形数据的再现的一些或全部。在一些实例中,待再现的图形数据可包含图形基元列表,例如点、线、三角形、四边形、三角形条带、片等。在其它实例中,待再现的图形数据可包含一或多个路径再现基元,例如线段、椭圆形弧、二次贝塞尔曲线和三次贝塞尔曲线。

[0050] 存储器控制器8促进数据进出存储器10的传送。举例来说,存储器控制器8可接收来自CPU 6和/或GPU 12的存储器读取请求和存储器写入请求,并且服务于关于存储器10的这些请求以便为计算装置2中的组件提供存储器服务。存储器控制器8以通信方式耦合到存储器10。尽管图1的实例计算装置2中的存储器控制器8经说明为与CPU 6、GPU 12和存储器10中的每一者分离的处理模块,但在其它实例中,存储器控制器8的一些或全部功能性可实施在CPU 6、GPU 12和存储器10中的一或多者上。

[0051] 存储器10可存储可由CPU 6存取以用于执行的程序模块和/或指令及/或由在CPU 6上执行的程序使用的数据。举例来说,存储器10可存储用户应用程序和与所述应用程序相关联的图形数据。存储器10还可存储用于由计算装置2的其它组件使用和/或由其产生的信息。举例来说,存储器10可充当用于GPU 12的装置存储器并且可存储将在GPU 12上操作的数据以及由GPU 12执行的操作而产生的数据。举例来说,存储器10可存储路径数据、路径片段数据、表面、纹理缓冲器、深度缓冲器、模板缓冲器、顶点缓冲器、帧缓冲器或类似者的任何组合。另外,存储器10可存储命令流以供GPU 12处理。举例来说,存储器10可存储路径再现命令、3D图形再现命令和/或通用GPU计算命令。存储器10可包含一或多个易失性或非易失性存储器或存储装置,例如随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、同步动态随机存取存储器(SDRAM)、只读存储器(ROM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。

[0052] GPU 12可经配置以执行由CPU 6发布到GPU 12的命令。由GPU 12执行的命令可包含图形命令、绘制调用命令、GPU状态编程命令、存储器传送命令、通用计算命令、核心执行命令等。

[0053] 在一些实例中,GPU 12可经配置以执行图形操作以将一或多个图形基元再现到显示器18。在此等实例中,当在CPU 6上执行的软件应用程序中的一者需要图形处理时,CPU 6

可将图形数据提供到GPU 12用于再现到显示器18且将一或多个图形命令发布到GPU 12。图形命令可包含例如绘制调用命令、GPU状态编程命令、存储器传送命令、传图命令等。图形数据可包含顶点缓冲器、纹理数据、表面数据等。在一些实例中,CPU6可通过将命令及图形数据写入到存储器10而将所述命令及图形数据提供给GPU 12,存储器10可由GPU 12存取。

[0054] 在其它实例中,GPU 12可经配置以针对在CPU 6上执行的应用程序执行通用计算。在此等实例中,当在CPU 6上执行的软件应用程序中的一者决定将计算任务卸载到GPU12时,CPU 6可将通用计算数据提供到GPU 12,并且将一或多个通用计算命令发布到GPU 12。通用计算命令可包含例如核心执行命令、存储器传送命令等。在一些实例中,CPU 6可通过将命令和图形数据写入到可由GPU 12存取的存储器10而将命令和通用计算数据提供到GPU 12。

[0055] 在一些情况下,GPU 12可内置有相比CPU 6提供对向量操作更有效的处理的高度并行的结构。举例来说,GPU 12可包含多个处理元件,其经配置以用并行方式对多个顶点、控制点、像素和/或其它数据进行操作。GPU 12的高度并行性质在一些情况下可允许GPU 12与使用CPU 6再现图像相比更快地将图形图像(例如,GUI和二维(2D)和/或三维(3D)图形场景)再现到显示器18上。另外,GPU 12的高度并行性质可允许GPU 12比CPU 6更快速地处理用于通用计算应用的特定类型的向量及矩阵操作。

[0056] 在一些实例中,可将GPU 12集成到计算装置2的母板中。在其它情况下,GPU 12可存在于图形卡上,所述图形卡安装在计算装置2的母板中的端口中或可以其它方式并入经配置以与计算装置2交互操作的外围装置内。在其它情况下,GPU 12可位于与CPU6相同的微芯片上,从而形成芯片上系统(SoC)。GPU 12可包含一或多个处理器,例如一或多个微处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、数字信号处理器(DSP)或其它等效的集成或离散逻辑电路。

[0057] 在一些实例中,GPU 12可直接耦合到GPU高速缓冲存储器14。因此,GPU 12可在不必使用总线20的情况下从GPU高速缓冲存储器14读取数据并将数据写入到GPU高速缓冲存储器14。换句话说,GPU 12可使用本地存储装置而不是片外存储器在本地处理数据。此允许GPU 12通过消除GPU 12经由总线20读取及写入数据的需要而以更有效的方式操作,其中经由总线操作可经历繁重的总线业务。然而,在一些情况下,GPU12可不包含单独的高速缓冲存储器,而是经由总线20利用系统存储器10。GPU高速缓冲存储器14可包含一或多个易失性或非易失性存储器或储存装置,例如随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。

[0058] CPU 6和/或GPU 12可将再现的图像数据存储于存储器10内所分配的帧缓冲器中。关于向量图形,所再现的图像数据可包含待再现的路径片段的所再现填充区域和描画区域。显示接口16可从帧缓冲器检索数据并配置显示器18以显示由所再现的图像数据表示的图像。在一些实例中,显示接口16可包含经配置以将从帧缓冲器检索的数字值转换为可由显示器18消耗的模拟信号的数/模转换器(DAC)。在其它实例中,显示接口16可将数字值直接传递到显示器18以进行处理。

[0059] 显示器18可包含监视器、电视、投影装置、液晶显示器(LCD)、等离子显示面板、发光二极管(LED)阵列、阴极射线管(CRT)显示器、电子纸、表面传导电子发射显示器(SED)、激

光电视显示器、纳米晶体显示器或另一类型的显示单元。显示器18可集成于计算装置2内。举例来说,显示器18可为移动电话手持机或平板计算机的屏幕。替代地,显示器18可为经由有线或无线通信链路而耦合到计算机装置2的独立装置。举例来说,显示器18可为经由电缆或无线链路而连接到个人计算机的计算机监视器或平板显示器。

[0060] 总线20可使用总线结构和总线协议的任何组合来实施,包含第一、第二和第三代总线结构和协议、共享总线结构和协议、点对点总线结构和协议、单向总线结构和协议以及双向总线结构和协议。可用以实施总线20的不同总线结构和协议的实例包含例如超传输总线、InfiniBand总线、高级图形端口总线、外围组件互连(PCI)总线、PCI高速总线、高级微控制器总线架构(AMBA)、高级高性能总线(AHB)、AMBA高级外围总线(APB),和AMBA高级eXtensible接口(AXI)总线。也可以使用其它类型的总线结构和协议。

[0061] 在一些情况下,GPU 12可经配置以提供各种路径再现命令的部分到全部基于GPU的执行。举例来说,CPU 6可将一或多个路径再现命令发布到GPU 12,并且GPU 12可执行所述路径再现命令。作为一个实例,CPU 6可向GPU 12发布一或多个路径填充命令,其指示GPU 12执行路径填充操作,并且GPU 12可执行所述路径填充命令。作为另一实例,CPU 6可向GPU 12发布一或多个路径描画(stroking)命令,其指示GPU 12执行路径描画操作,并且GPU 12可执行所述路径描画命令。

[0062] 在一些实例中,GPU 12可经配置以接收指示待再现的路径的路径片段的数据,将路径片段曲面细分为多个基元,并基于所述多个基元再现用于所述路径片段的填充区域和描画区域中的至少一者。GPU可在执行填充操作时再现用于路径片段的填充区域,并且可在执行描画操作时再现用于路径片段的描画区域。在一些实例中,所述多个基元可为多个线段。

[0063] 在一些实例中,GPU 12可使用两遍次再现方法来执行路径填充操作。举例来说,作为第一再现遍次的一部分,GPU 12可接收指示待再现路径的路径片段的数据,将所述路径片段曲面细分为多个线段,并基于所述多个线段产生多个三角形基元。GPU 12可基于所述多个线段中的相应者产生所述多个三角形基元中的每一者。GPU 12可将所述多个三角形基元中的每一者再现到共同模板缓冲器中使得所述共同模板缓冲器存储指示哪些像素在用于路径片段的填充区域内部的数据。在将基元再现到共同模板缓冲器之后,GPU 12可执行第二再现遍次。在第二再现遍次期间,GPU 12可基于存储于模板缓冲器中的数据和填充色彩来再现包含处于用于路径片段的填充区域内部的像素的一或多个基元,以便产生用于路径片段的填充区域的光栅化版本。

[0064] 为了产生用于路径填充操作的多个三角形基元,在一些实例中,GPU 12可产生多个三角形基元以使得所述三角形基元中的每一者具有对于针对路径片段产生的全部三角形基元相同的共同顶点。在所述实例中,GPU 12可产生所述多个三角形基元以使得所述三角形基元中的每一者具有对应于所述多个线段中的相应者的端点的两个额外顶点(即,除共同顶点之外的两个顶点)。每一额外顶点可对应于对应线段的端点中的相应者。

[0065] 因此,在执行路径再现时,GPU 12可执行以下实例功能来填充路径:将路径曲面细分为线段,将线段连接到枢转点以形成三角形基元,并将三角形再现到模板缓冲器。填充过程的接下来的和可能的最终步骤为:在启用模板测试的情况下再现包含所述路径的限界框(如(例如)关于图5C较详细描述),并将模板内容复制到帧缓冲器。在一些情况下,可基于从

CPU 6接收的命令确定所述限界框。如上所述,此方法需要两个再现遍次和路径的预处理以计算限界框。

[0066] 另外,在执行例如MSAA等的图形保真时,GPU 12可能以与再现目标相同的速率取样模板缓冲器。举例来说,如果模板缓冲器和再现目标都以MSAA速率取样,那么在将模板缓冲器复制到帧缓冲器时所消耗的存储器带宽可能相对较大。如果GPU 12执行TIR并使用用于再现目标的相对较小分配,那么模板取样速率也可能受到影响,进而降低模板缓冲器的准确度。

[0067] 根据本发明的各方面,GPU 12可执行经模板印刷TIR。举例来说,GPU 12可确定模板参数,所述模板参数指示用于确定图像的路径的每一经图形保真像素的覆盖值的取样速率。GPU 12还可单独地根据所述模板参数确定再现目标参数,所述再现目标参数指示所述路径的每一经图形保真像素的存储器分配。CPU 12可使用所述模板参数和所述再现目标参数再现所述路径。

[0068] 在一些实例中,GPU 12可能以比分配用于像素的存储器量高的速率执行模板印刷。举例来说,关于16x MSAA,GPU 12可执行模板操作,其(例如)在每一像素具有16个样本的情况下经超取样。GPU 12可通过基于通过模板测试(例如,经确定为在路径内部)的像素的样本的数目计算每像素的覆盖值来再现给定路径。根据本发明的各方面,GPU 12像素的再现目标可为1x取样的,而模板为16x取样的。

[0069] 另外,根据本发明的各方面,GPU 12可在单一再现遍次中填充路径,而无需预处理限界框。举例来说,GPU 12可在模板印刷操作期间确定限界框。在此实例中,由于GPU 12在模板印刷期间再现基元(例如,不着色像素),GPU 12可确定路径的基元的最外点(例如,最外边界点)。在一些实例中,GPU 12可确定上点(在路径的相对顶部处)、下点(在路径的相对底部处)、右点(在路径的最右点处)和左点(在路径的最左点处)。GPU 12可使用在模板印刷期间确定的最外点来确定限界框。也就是说,GPU 12可确定包含路径的所有基元的限界框。在一些实例中,所述限界框可由两个三角形组成。

[0070] 在完成限界框后,GPU 12可进一步通过在限界框的顶部上执行模板印刷TIR来处理(例如,在同一再现遍次中)限界框。也就是说,如上所述,GPU 12可确定每一像素的覆盖值并着色GPU 12确定位于模板内的像素。在此实例中,GPU 12不必对像素执行单独的深度测试。

[0071] 关于描画,GPU 12可在一些情况下用短划线划出所描画路径。也就是说,GPU 12可确定所描画路径的多个片段,使得所再现的路径看似虚线。通常,GPU 12可依序确定用短划线划出的路径的片段。举例来说,GPU 12可接收来自CPU 6的命令以再现一个片段,随后移动到上路径的下一片段。所述过程可妨碍并行度(例如,在特定时间个例中光栅化及/或着色一个以上片段)并且可阻止GPU 12独立再现所述路径。

[0072] 根据本发明的各方面,GPU 12可确定路径的每一片段的位置(以及路径的长度)并在再现期间应用所述长度信息。举例来说,GPU 12可确定虚线的多个经排序片段中的每一片段的纹理偏移。在一些情况下,如下文更详细描述,可在几何形状着色期间确定片段次序。在此实例中,所述多个经排序片段中的当前片段的纹理偏移可基于次序先于当前片段的片段的长度累加。GPU 12还可像素着色所述片段,包含将纹理偏移应用到每一片段以确定片段的位置。举例来说,GPU 12可基于片段的位置确定片段是可见的还是不可见的。GPU

12可保留可见片段(例如,确定其色彩)并舍弃不可见的短划线的片段(例如,可见短划线之间的空间)。以此方式,GPU 12可例如在不接收来自CPU 6的划短划线命令的情况下执行虚线的路径再现。

[0073] 虽然关于路径再现进行了描述,但上文所描述的前缀总和操作不限于向量图形。举例来说,用于确定前缀总和的技术可用于其中GPU 12跟踪累加值的任何应用中。在出于说明目的的实例中,GPU 12可在确定梯度时执行上文所描述的前缀总和操作。举例来说,在图像处理中,产生梯度可能需要累加一些长度信息来确定色彩。在此实例中,GPU12可应用上文所描述的前缀总和操作来确定长度信息。

[0074] 本发明中描述的再现技术可实施在图1中所示的计算装置2的组件中之任一者中,包含(例如)CPU 6、GPU 12和存储器10。在一些实例中,所述路径再现技术可完全或差不多完全由GPU 12(例如,在如关于图3所描述的GPU 12的图形管线中)实施。在额外实例中,CPU 6可实施用于配置图形管线的状态并将着色器程序绑定到图形管线以实施GPU 12中执行本发明的路径再现技术的路径再现管线的技术。在其它实例中,CPU 6可经配置以将指示待再现路径的数据放置到一或多个缓冲器(例如,一或多个顶点缓冲器)中,所述缓冲器可由GPU 12存取以再现一或多个路径。

[0075] 图2是进一步详细说明图1中的计算装置2的CPU 6、GPU 12和存储器10的框图。如图2中所展示,CPU 6以通信方式耦合到GPU 12和存储器10,并且GPU 12以通信方式耦合到CPU 6和存储器10。在一些实例中,GPU 12可与CPU 6一起集成到母板上。在额外实例中,GPU 12可实施在安装在包含CPU 6的母板的端口中的图形卡上。在其它实例中,GPU 12可并入在经配置以与CPU 6交互操作的外围装置内。在额外实例中,GPU 12可位于与CPU 6相同的微芯片上,从而形成芯片上系统(SoC)。

[0076] CPU 6经配置以执行软件应用程序24、图形API 26、GPU驱动程序28和操作系统30。软件应用程序24可包含引起显示图形图像的一或多个指令和/或引起在GPU 12上执行非图形任务(例如,通用计算任务)的一或多个指令。软件应用程序24可将指令发布到图形API 26。图形API 26可为将从软件应用程序24接收的指令转译为可由GPU驱动程序28消耗的格式的运行时服务。GPU驱动程序28经由图形API 26从软件应用程序24接收指令,并控制GPU 12的操作以服务所述指令。举例来说,GPU驱动程序28可制定一或多个命令38、将所述命令38放置到存储器10中并指示GPU 12执行命令38。在一些实例中,GPU驱动程序28可将命令38放置到存储器10中,并经由操作系统30(例如经由一或多个系统调用)与GPU 12通信。

[0077] GPU 12包含命令引擎32和一或多个处理单元34。在一些实例中,所述一或多个处理单元34可形成及/或实施3D图形再现管线,例如,DX 11图形再现管线(即,与DX 11图形API顺应的3D图形管线)。

[0078] 命令引擎32经配置以从CPU 6(例如,经由存储器10)接收命令及引起GPU 12执行命令。响应于接收到状态命令,命令引擎32可经配置以基于所述状态命令将GPU 12中的一或多个状态寄存器设定到特定值,及/或基于所述状态命令配置固定功能处理单元34中的一或多个者。响应于接收到绘制调用命令,命令引擎32可经配置以引起处理单元34基于定义待再现的一或多个路径片段的几何形状的数据并基于指示待再现路径片段中的每一者的路径片段类型的数据来再现一或多个路径片段。在一些实例中,定义待再现的一或多个路径片段的几何形状的数据和定义路径片段中的每一者的路径片段类型的数据可存储于存

存储器10中的一或多个顶点数据结构中。命令引擎32还可接收着色器程序绑定命令,并基于着色器程序绑定命令将特定着色器程序加载到可编程处理单元34中的一或多个者中。

[0079] 处理单元34可包含一或多个处理单元,其中的每一者可为可编程处理单元或固定功能处理单元。可编程处理单元可包含(例如)经配置以执行从CPU 6下载到GPU 12上的一或多个着色器程序的可编程着色器单元。在一些实例中,着色器程序可为以高级着色语言(例如,OpenGL着色语言(GLSL)、高级着色语言(HLSL)、用于图形的C(Cg)着色语言等)编写的程序的经编译版本。

[0080] 在一些实例中,可编程着色器单元可包含经配置以并行操作的多个处理单元(例如,SIMD管线)。可编程着色器单元可具有存储着色器程序指令的程序存储器和执行状态寄存器,例如指示程序存储器中正执行的当前指令或待提取的下一指令的程序计数器寄存器。处理单元34中的可编程着色器单元可包含(例如)顶点着色器单元、像素着色器单元、几何形状着色器单元、壳着色器单元、域着色器单元、曲面细分控制着色器单元、曲面细分评估着色器单元、计算着色器单元和/或统一着色器单元。如图2中所展示,处理单元34还可包含限界框单元40和前缀总和单元。

[0081] 固定功能处理单元可包含经硬连线以执行某些功能的硬件。尽管固定功能硬件可经由例如一或多个控制信号配置以执行不同功能,但所述固定功能硬件通常并不包含能够接收用户编译程序的程序存储器。在一些实例中,处理单元34中的固定功能处理单元可包含(例如)执行光栅操作的处理单元,所述光栅操作例如深度测试、剪刀测试、 α 掺杂等。

[0082] 存储器10可存储路径数据36和一或多个命令38。在一些实例中,路径数据36可作为多个顶点(或控制点)存储在存储器10中分配的一或多个顶点缓冲器中。在一些实例中,路径数据可存储在片列表数据结构(例如,四控制点片列表)。命令38可存储于一或多个命令缓冲器(例如,环缓冲器)中。CPU 6(例如,经由操作系统30的GPU驱动程序28)可将路径数据36和命令38放置到存储器10中以供GPU 12消耗。GPU 12(例如,命令引擎32)可检索并执行存储于存储器10中的命令38。

[0083] 在其中路径数据36存储为顶点(例如,控制点)的实例中,所述顶点可包含几何学上定义待再现路径片段的一或多个属性。举例来说,对于线,片控制列表中的顶点可包含指示所述线的端点的坐标(例如,(x_0, y_0)和(x_1, y_1))的数据。对于三次贝塞尔曲线,片控制列表中的顶点可包含指示定义所述曲线的四个控制点的坐标(例如,(x_0, y_0)、(x_1, y_1)、(x_2, y_2)、(x_3, y_3))的数据。对于二次贝塞尔曲线,片控制列表中的顶点可包含指示三个控制点而非四个控制点的坐标的数据。对于椭圆形弧,片控制列表中的顶点可包含指示椭圆形弧的端点参数化的数据或指示椭圆形弧的中心参数化的数据。

[0084] 在一些情况下,几何学上定义待再现路径片段的一或多个属性可为分辨率独立的。换句话说,几何学上定义路径片段的属性可独立于当再现路径片段时执行的曲面细分的量和/或独立于当再现路径片段时产生的顶点的量。

[0085] CPU 6还可将指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)放置到顶点缓冲器中的一或多个原本未使用的顶点属性中。在一些实例中,不同路径片段类型可对应于由向量图形API定义并且可用于由软件应用程序24使用的一组路径片段类型。在一些实例中,不同路径片段类型可对应于由OpenVG API定义的一组路径片段类型。

[0086] 命令38可包含一或多个状态命令和/或一或多个绘制调用命令。状态命令可指示

GPU 12改变GPU 12中的状态变量中的一或多者,例如绘制色彩、填充色彩、描画色彩等。在一些实例中,状态命令可包含路径再现状态命令,其经配置以设定与再现路径相关联的一或多个状态变量。举例来说,状态命令可包含图画模式命令,其经配置以指示是否将填充待再现路径、描画待再现路径或两者。作为另一实例,状态命令可包含指定将用于填充操作的色彩的填充色彩命令,和/或指定用于描画操作的色彩的描画色彩命令。作为另一实例,状态命令可指定用于描画操作的一或多个参数,例如描画宽度、端盖样式(例如,圆形、正方形)、线接合点样式(例如,斜接、圆形、斜边)、斜接限制等。在一些实例中,除使用状态命令来设定一或多个状态参数之外或代替使用状态命令来设定一或多个状态参数,所述状态参数中的一或多者可通过使用绘制调用命令或通过状态指示符放置到含有路径数据36的顶点缓冲器中来设定。

[0087] 绘制调用命令可指示GPU 12再现由存储于存储器10中的一或多个顶点的群组定义(例如,在顶点缓冲器中定义)的几何形状。在一些实例中,绘制调用命令可调用GPU 12以再现存储于存储器10的经定义区段(例如,顶点缓冲器或路径数据36)中的全部顶点。换句话说,一旦GPU 12接收到绘制调用命令,控制便传递到GPU 12以用于再现由存储器10的经定义区段(例如,顶点缓冲器或路径数据36)中的顶点表示的几何形状和基元。

[0088] 绘制调用命令可包含3D绘制调用命令和路径再现绘制调用命令中的一者或两者。对于3D再现绘制调用命令,由顶点缓冲器中的一或多个顶点的群组定义的几何形状可对应于待再现的一或多个3D图形基元(例如,点、线、三角形、四边形、三角形条带、片等),并且3D再现绘制调用命令可指示GPU 12再现所述一或多个3D图形基元。对于路径再现绘制调用命令,由顶点缓冲器中的一或多个顶点的群组定义的几何形状可对应于待再现的一或多个路径基元(例如,线段、椭圆形弧、二次贝塞尔曲线和三次贝塞尔曲线等),并且路径再现绘制调用命令可指示GPU 12再现所述一或多个路径基元。在一些实例中,能够由GPU 12再现的路径基元可对应于本发明中描述的不同类型的路径片段。

[0089] 在一些实例中,本发明中描述的路径再现技术可实施在图2中所展示的组件中的任一者中,包含(例如)图形API 26、GPU驱动程序28、命令引擎32和处理单元34。在一些实例中,全部或几乎全部路径再现技术可实施在由处理单元34形成的GPU 12中的图形管线中。在额外实例中,CPU 6的软件应用程序24、图形API 26和/或GPU驱动程序28可实施技术以配置图形管线的状态并将着色器程序绑定到图形管线以实施执行本发明中所描述的路径再现技术的GPU 12中的路径再现管线。在其它实例中,CPU 6的软件应用程序24、图形API 26和/或GPU驱动程序28可经配置以将指示待再现路径的数据放置到可由GPU 12存取以再现一或多个路径的一或多个缓冲器(例如,一或多个顶点缓冲器)中。

[0090] 根据本发明的各方面,处理单元34包括限界框单元40。限界框单元40可包含用于确定限界框的一或多个可编程和/或固定功能单元。举例来说,本发明的技术包括在单一再现遍次中确定限界框和再现限界框(如(例如)下文关于图3较详细描述)。在GPU 12执行路径填充操作时,限界框单元40可负责确定路径的边界。

[0091] 根据本发明的各方面,可使用API调用起始所述限界框单元40。举例来说,图形API 26可包含用于在路径的再现期间触发限界框单元40的使用的一或多个指令。所述API调用可允许GPU 12跳过基元的着色直至限界框单元40已确定所述限界框为止。如上所述,GPU 12可接着在限界框的顶部上执行模板印刷TIR。另外,通过并入有限限界框单元40,GPU 12可

在不使用深度缓冲器的情况下在单一遍次中填充路径。

[0092] 限界框单元40可允许GPU 12在不预处理限界框的情况下填充路径。举例来说,限界框单元40可例如在CPU 6处使用控制多边形确定所述限界框。也就是说,限界框单元40可基于所有所产生基元的边界确定所述限界框。

[0093] 根据本发明的各方面,GPU 12可经配置以跳过基元的着色直至限界框单元40已确定所述限界框为止。也就是说,在产生限界框期间,GPU 12可将路径的基元写入到GPU12的模板缓冲器而不着色所述基元。通过并入有限限界框单元40,GPU 12可在不使用深度缓冲器的情况下在单一遍次中填充路径。举例来说,GPU 12可在限界框的顶部上执行模板印刷TIR。

[0094] 在出于说明目的的实例中,在GPU 12已光栅化由限界框单元40确定的限界框后,GPU 12可确定所述限界框中的每一像素的覆盖值。在一些实例中,GPU 12可确定像素中的四者的(同时确定4各像素的)覆盖值。在此等实例中,在形成用于处理的像素波之前,GPU 12可对四者中的每一像素的样本执行模板测试。GPU 12可基于测试的结果更新每一像素的覆盖蒙板(mask)。此覆盖值可被称为GPU 12接着可在着色期间使用的stenciled_TIR属性。举例来说,每一像素的InputCoverage值可基于所述stenciled_TIR。举例来说,GPU 12可像素着色(例如,染色)通过模板测试的每一像素(例如,其中当像素的更多样本是可见的(着色的)而较少样本不可见时,像素通过模板测试)。也就是说,GPU 12可将覆盖蒙(用于图心)和模板测试后的取样蒙板(用于InputCoverage)从分布式处理器(DProc)传递到取样器。

[0095] 根据本发明的一些方面,API调用可用于支持再现的模板印刷TIR模式。举例来说,图形API 26可包含用于在路径的再现期间触发模板印刷TIR的使用的一或多个指令。当模板印刷TIR在作用中时,色彩缓冲器和深度/模板缓冲器(其可分配在GPU 12的存储器和/或存储器10中)可不同。举例来说,在执行MSAA时,GPU 12可再现到为1x MSAA的色彩缓冲器和为16x MSAA的模板缓冲器。

[0096] 根据本发明的其它方面,处理单元34还包括用于再现用短划线划出的片段(例如,描画用短划线划出的路径)的前缀总和单元42。前缀总和单元42可确定虚线的多个经排序片段中的每一片段的纹理偏移。在一些实例中,曲面细分或几何形状着色器级可确定产生片段时的片段次序。所述多个经排序片段中的当前片段的纹理偏移可基于次序先于当前片段的片段的长度累加。前缀总和单元42可提供纹理偏移到着色器级,例如像素着色器级。着色器级可应用纹理偏移并在适当位置中再现片段。

[0097] 因此,前缀总和单元42可包含累加虚线的片段长度的一或多个可编程或固定功能单元。在一些实例中,前缀总和单元42可并入在光栅化器级处。举例来说,GPU 12可曲面细分路径,并且几何形状着色器级可确定路径长度。在其它实例中,所述长度可由一或多个其它着色器单元确定。举例来说,根据本发明的各方面,前缀总和单元42可以类似于用于属性点大小(pointsize)(指示点基元的大小)的系统解译值的方式来计算线长(linlength)值。也就是说,线长可为指示用短划线划出的图案中的片段(还被称作分段)的位置的系统解译值。

[0098] 当GPU 12的像素着色器(例如,如下文关于图3所描述)接收前缀求和线长值时,所述像素着色器可确定短划线图案中的经着色分段的位置。所述像素着色器可接着基于所确定的位置保留分段(如果其形成可见短划线的部分)或舍弃所述分段(如果其并非可见短划

线的部分)。在任何情况下,前缀总和单元42可累加长度信息来作为前缀总和并将所述前缀总和作为纹理偏移提供到下游着色器级,例如像素着色器。

[0099] 在再现期间,GPU 12可发送事件presum_start到曲面细分引擎(TSE)(其可包含壳着色器、曲面细分器和/或域着色器,如下文关于图3较详细描述)以重设prefix_sum参数。对于每一基元,前缀总和单元42可将基元的标量值(例如,与点大小相同的字段)作为新的值添加到prefix_sum。前缀总和单元42可将每像素的旧前缀总和值作为纹理偏移来传递。

[0100] 在一些实例中,所述曲面细分引擎可并入有寄存器以累加prefix_sum参数。前缀总和单元42可通过事件presum_start重设寄存器。所述曲面细分引擎将prefix_sum作为每基元属性传递以类似于基元面(其可类似于发送纹理偏移)而在重心平面接口中再现后端(RB)。在此实例中,属性可添加到RB以提供具有表示所述每基元属性的接口的高级定序器(HLSQ)。

[0101] 图3是说明可执行本发明的路径再现技术的实例图形管线43的概念图。在一些实例中,所述图形管线可对应于Microsoft® DirectX (DX) 11图形管线。如图3中所展示,图形管线43包含多个处理级,包含输入汇编程序(IA) 44、顶点着色器(VS) 46、壳着色器(HS) 48、曲面细分器50、域着色器(DS) 52、几何形状着色器(GS) 54、光栅化器56、像素着色器(PS) 58和输出合并器60。壳着色器48、曲面细分器50和域着色器52可形成图形管线43的曲面细分级62。另外,管线43还包含资源块64。在一些实例中,管线43可通过GPU 12实施和/或并入在GPU 12中,如下文所述。

[0102] 资源块64可对应于由图形管线43使用的一或多个存储器资源,例如一或多个纹理和/或一或多个缓冲器。资源块64可存储将由图形管线43中的处理级中的一或多者处理的输入数据和/或来自图形管线43中的处理级中的一或多者的输出数据。作为一个实例,资源块64可存储用于执行如本发明中所描述的路径填充操作的模板缓冲器。作为另一实例,资源块64可存储帧缓冲器,其如本发明中所描述保持用于路径片段的填充区域的光栅化版本和/或用于路径片段的描画区域的光栅化版本。在一些实例中,形成资源块64的存储器资源可驻留在计算装置2的存储器10和/或GPU高速缓冲存储器14中。

[0103] 图3中以直拐角所描绘的处理级表示固定功能处理级,并且图3中以圆形拐角所描绘的处理级表示可编程处理级。举例来说,如图3中所展示,输入汇编程序44、曲面细分器50、光栅化器56和输出合并器60是固定功能处理级,并且顶点着色器46、壳着色器48、域着色器52、几何形状着色器54和像素着色器58是可编程处理级。可编程级中的每一者可经配置以执行特定类型的着色器程序。举例来说,顶点着色器46可经配置以执行顶点着色器程序,壳着色器48可经配置以执行壳着色器程序等。不同类型的着色器程序中的每一者可在GPU 12的共同着色器单元上或在专用于执行一或多个特定类型的着色器程序的一或多个专用着色器单元上执行。

[0104] 如图3中所展示,输入汇编程序44、顶点着色器46、壳着色器48、域着色器52、几何形状着色器54、像素着色器58和输出合并器60通信地耦合到资源块64。输入汇编程序44、顶点着色器46、壳着色器48、域着色器52、几何形状着色器54、像素着色器58和输出合并器60经配置以从资源块64检索和/或接收输入数据。几何形状着色器54和输出合并器60经配置以将输出数据写入到资源块64。图形管线43中的处理级与资源块64之间的通信的上文所描述配置仅是在图形管线43的处理级与资源块64之间可如何配置通信的一个实例。在其它实

例中,在图形管线43的处理级与资源块64之间可提供更多或更少的单向和/或双向通信信道。

[0105] 关于DirectX 11图形管线的一般操作的额外背景信息可在<http://msdn.microsoft.com/en-us/library/windows/desktop/ff476882%28v=vs.85%29.aspx>处找到。关于DirectX 11图形管线的一般操作的进一步信息可在Zink等人的“Practical Rendering&Computation with Direct3D 11”(CRC Press (2011))中找到。

[0106] 两个主要路径再现操作可包含:(1)填充路径片段;及(2)描画路径片段。在一些情况下,所述填充操作可利用可一般涉及以下步骤的双遍次方法:

[0107] 遍次1

[0108] 1.将路径片段曲面细分为多个线段。

[0109] 2.产生每个线段的三角形基元。

[0110] 3.将全部三角形基元再现到模板缓冲器中。

[0111] 遍次2

[0112] 4.使用模板缓冲器再现路径片段的限界框。

[0113] 对于第一遍次,CPU 6可将指示待再现路径片段的数据放置到顶点缓冲器的一或多个顶点中。在一些实例中,顶点缓冲器可对应于图2中所展示的路径数据36。用于顶点缓冲器中的顶点的基元拓扑在一些实例中可为片控制列表。对于线,片控制列表中的顶点可包含指示所述线的端点的坐标(例如,(x0,y0)和(x1,y1))的数据。对于三次贝塞尔曲线,片控制列表中的顶点可包含指示定义所述曲线的四个控制点的坐标(例如,(x0,y0)、(x1,y1)、(x2,y2)、(x3,y3))的数据。对于二次贝塞尔曲线,片控制列表中的顶点可包含指示定义曲线的三个控制点而非四个控制点的坐标的数据。对于椭圆形弧,片控制列表中的顶点可包含指示椭圆形弧的端点参数化的数据或指示椭圆形弧的中心参数化的数据。CPU 6还可将指示待再现路径片段的类型的数据放置到片控制列表的原本未使用的顶点属性中。

[0114] 现将描述由GPU 12接收及使用以执行路径再现的路径数据36的一个实例格式。应理解这仅是指示待再现路径和/或待再现路径片段的数据可如何由CPU 6提供到GPU 12的一个实例并且其它实例是可能的且在本发明的范围内。在此实例中,GPU 12接收每一路径片段作为四(4)个控制点片列表基元。在此实例中,片列表中的顶点(例如,控制点)中的每一者包含定义相应顶点(例如,控制点)的属性的三(3)个浮动属性。

[0115] 对于线路径片段,输入路径数据可采取以下形式或类似形式:

[0116] {XMFLOAT3(X0,Y0,2.0f)},

[0117] {XMFLOAT3(X1,Y1,1.0f)},

[0118] {XMFLOAT3(0.0f,0.0f,1.0f)},

[0119] {XMFLOAT3(0.0f,0.0f,1.0f)},

[0120] 在此实例中,每一行表示四个控制点片列表的顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)。具体来说,在此实例中路径片段类型指示符是2.0f,其意味着所述路径片段是线路径片段。X0、Y0、X1、Y1是线路径片段的端点的坐标,其中(X0,Y0)表示第一端点并且(X1,Y1)表示第二端点。

[0121] 在此实例中剩余顶点和属性可未使用和/或可用以指示路径片段的其它属性。路

径片段的其它属性可包含例如路径片段是否为开放路径的开始或末端、路径片段是否应针对路径显示、端盖是否应放置于路径片段的末端上、应使用何种类型的端盖(如果存在)、是否应在路径片段的末端上放置接合点,以及将使用何种类型的接合点(如果存在)。

[0122] 用于三次贝塞尔路径片段的输入路径数据可采取以下形式或类似形式:

[0123] {XMFLOAT3 (X0,Y0,3.0f)} ,

[0124] {XMFLOAT3 (X1,Y1,1.0f)} ,

[0125] {XMFLOAT3 (X2,Y2,1.0f)} ,

[0126] {XMFLOAT3 (X3,Y3,1.0f)} ,

[0127] 在此实例中,每一行表示四个控制点片列表的顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)。具体来说,在此实例中路径片段类型指示符是3.0f,其意味着所述路径片段是三次贝塞尔曲线路径片段。X0-X3和Y0-Y3是用于三次贝塞尔路径片段的控制点的坐标,其中(X0,Y0)表示第一控制点,(X1,Y1)表示第二控制点等。在此实例中剩余顶点和属性可未使用和/或可用以指示路径片段的其它属性。在一些实例中,路径片段的其它属性可包含类似于上文关于线路径片段描述的属性。

[0128] 类似输入可用于二次贝塞尔路径片段,不同之处在于可提供三个控制点而非四个控制点,并且路径片段类型指示符可为不同的以区分所述基元与三次贝塞尔路径片段。

[0129] 举例来说,用于二次贝塞尔路径片段的输入路径数据可采取以下形式或类似形式:

[0130] {XMFLOAT3 (X0,Y0,1.0f)} ,

[0131] {XMFLOAT3 (X1,Y1,1.0f)} ,

[0132] {XMFLOAT3 (X2,Y2,1.0f)} ,

[0133] {XMFLOAT3 (0.0f,0.0f,1.0f)} ,

[0134] 在此实例中,每一行表示四个控制点片列表的顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)。具体来说,在此实例中路径片段类型指示符是1.0f,其意味着所述路径片段是二次贝塞尔路径片段。X0-X2和Y0-Y2是用于二次贝塞尔路径片段的控制点的坐标,其中(X0,Y0)表示第一控制点,(X1,Y1)表示第二控制点等。在此实例中剩余顶点和属性可未使用和/或可用以指示路径片段的其它属性。在一些实例中,路径片段的其它属性可包含类似于上文关于线路径片段描述的属性。

[0135] 在一些实例中,用于椭圆形弧路径片段的输入路径数据可包含指示所述椭圆形弧路径片段的中心参数化的数据。举例来说,用于椭圆形弧路径片段的输入路径数据可采取以下形式或类似形式:

[0136] {XMFLOAT3 (X0,Y0,4.0f)} ,

[0137] {XMFLOAT3 (X1,Y1,1.0f)} ,

[0138] {XMFLOAT3 (c.x,c.y,1.0f)} ,

[0139] {XMFLOAT3 (theta0,theta1,角度)} ,

[0140] 在此实例中,每一行表示四个控制点片列表的顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再

现路径片段的类型的数据(即,“路径片段类型指示符”)。在此实例中,路径片段类型指示符可为4.0、4.1、4.2或4.3中的任一者,其分别对应于大顺时针(LCW)椭圆形弧、大逆时针(LCCW)椭圆形弧、小顺时针(SCW)椭圆形弧以及小逆时针(SCCW)椭圆形弧。 X_0 、 X_1 和 Y_0 、 Y_1 是椭圆形弧路径片段的端点坐标,其中 (X_0, Y_0) 表示弧的初始端点且 (X_1, Y_1) 表示弧的最终端点。另外,theta0表示椭圆形弧的初始点的角度(如在未按比例缩放的圆上所测量的),并且theta1表示椭圆形弧的最终点的角度(如在未按比例缩放圆上所测量的)。值得注意的是,即使以上指定的实例输入数据形式是中心参数化,输入数据形式仍可包含弧的初始和最终端点的坐标(即, (X_0, Y_0) 、 (X_1, Y_1))。在一些实例中,此类坐标可用以确保所得几何形状的水密性。

[0141] 在其它实例中,用于椭圆形弧路径片段的输入路径数据可包含指示所述椭圆形弧路径片段的端点参数化的数据。举例来说,用于椭圆形弧路径片段的输入路径数据可采取以下形式或类似形式:

[0142] {XMFLOAT3($X_0, Y_0, 4.0f$)},

[0143] {XMFLOAT3($X_1, Y_1, 1.0f$)},

[0144] {XMFLOAT3($rH, rV, 1.0f$)},

[0145] {XMFLOAT3(角度, $0.0f, 1.0f$)},

[0146] 在此实例中,每一行表示四个控制点片列表的顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)。在此实例中,路径片段类型指示符可为4.0、4.1、4.2或4.3中的任一者,其分别对应于大顺时针(LCW)椭圆形弧、大逆时针(LCCW)椭圆形弧、小顺时针(SCW)椭圆形弧以及小逆时针(SCCW)椭圆形弧。 X_0 、 X_1 和 Y_0 、 Y_1 是椭圆形弧路径片段的端点坐标,其中 (X_0, Y_0) 表示弧的初始端点且 (X_1, Y_1) 表示弧的最终端点。另外,角度表示在通过(rH, rV)缩放之前所测量的椭圆相对于x轴线的逆时针旋转角。

[0147] 在其中输入路径数据包含以端点参数形式表示的椭圆形弧的实例中,在一些实例中,CPU 6可将椭圆形弧的表示从端点参数形式转换为以中心参数形式,之后将指示椭圆形弧的数据发送到GPU 12用于再现。举例来说,CPU 6可基于椭圆形弧的端点参数化产生椭圆形弧的中心参数化,并将椭圆形弧的中心参数化发送到GPU 12。椭圆形弧的中心参数化可符合以上指定的实例输入数据形式。中心参数化可由CPU 6使用以找到椭圆形弧的端点切线和/或法线,其又可由CPU 6使用以产生用于由GPU 12再现的接合点基元。

[0148] 在一些实例中,描画操作可使用顶点路径数据输入的三个额外字段来处置端盖、接合点和开放路径。举例来说,某些顶点坐标可存储指示路径片段是否是开放路径的开始、开放路径的末端以及路径片段是否可丢弃((例如,路径片段是开放路径的闭合路径片段)的数据。以下是包含上文所描述的顶点属性的实例模板:

[0149] {XMFLOAT3($X_0, Y_0, 2.0f$)},

[0150] {XMFLOAT3($X_1, Y_1, 2.0f$)},

[0151] {XMFLOAT3($0.0f, 0.0f, 2.0f$)},

[0152] {XMFLOAT3($0.0f, 0.0f, 2.0f$)},

[0153] 在此模板中,第二顶点的z坐标(即第三坐标或属性)上的 $2.0f$ 指示所述路径片段是开放路径的开始并且可发信号到GPU 12以在所述路径片段的开始处放置端盖(即,起始

盖)。第三顶点的z坐标上的2.0f指示所述路径片段是开放路径的结束并且可发信号到GPU 12以在所述路径片段的末端处放置端盖。最后顶点的z坐标上的2.0f指示当前基元将丢弃(例如,其是开放路径的闭合线或路径片段)。

[0154] 为了执行路径填充操作,输入汇编程序44从存储器10获得路径数据36,并将路径数据传递到图形管线43的一或多个后续级上以再现由路径数据36指定的路径片段(例如,路径基元)。举例来说,输入汇编程序44可从存储于存储器10中的顶点缓冲器获得多个顶点并引起顶点着色器46处理所述顶点。在一些实例中,输入汇编程序44可将待处理顶点直接传递到顶点着色器46。在额外实例中,输入汇编程序44可引导顶点着色器46以从资源块64中的顶点缓冲器检索用于处理的特定顶点。

[0155] 顶点着色器46经配置以处理从输入汇编程序44和/或资源块64接收的顶点并针对由顶点着色器46处理的每一输入顶点产生输出顶点。举例来说,对于每一输入顶点,顶点着色器46可执行GPU 12的着色器单元上的顶点着色器程序的个例。在一些实例中,顶点着色器46可执行用于每一输入顶点的“穿通”顶点着色器程序。所述“穿通”顶点着色器程序可引起顶点着色器46针对每一输入顶点输出对应于所述输入顶点的顶点。在此情况下,如果输出顶点具有与输入顶点相同的属性,那么输出顶点可对应于输入顶点。为了实施“穿通”顶点着色器程序,在一些实例中,顶点着色器46可将恒等变换应用于每一输入顶点以产生具有相同属性的输出顶点。由顶点着色器46接收的输入顶点和由顶点着色器46产生的输出顶点可替代地分别被称作输入控制点和输出控制点。

[0156] 在其它实例中,顶点着色器46可产生与对应输入顶点的输入属性不等同的输出顶点的一或多个输出属性。举例来说,顶点着色器46可对输入顶点的属性中的一或多个者执行实质性的处理以产生输出顶点的一或多个属性。作为一个实例,顶点着色器46可对输入顶点的位置属性执行世界变换、视图变换、投影变换或其任何组合中的一或多个者以产生输出顶点的一或多个属性。作为另一实例,顶点着色器46可从输入属性集合添加和/或删除属性以产生输出顶点的输出属性的集合。

[0157] 曲面细分级62(即,壳着色器48、曲面细分器50和域着色器52)可形成曲面细分引擎并且可将由输入路径数据定义的路径片段曲面细分为多个线段。所述多个线段可近似待再现路径片段的曲率。一般来说,壳着色器48可将从顶点着色器46接收的控制点传递到域着色器52用于进一步处理,并将配置数据提供到曲面细分器50。曲面细分器50可确定表示特定类型的路径片段的一或多个参数方程式应评估的值。域着色器52可在由曲面细分器50确定的值处评估参数方程式,并针对每一评估输出顶点。在一些实例中,由域着色器52输出的顶点中的每一者可包含指示所述顶点的位置的一或多个属性。在额外实例中,由域着色器52输出的顶点中的每一者可包含指示与所述顶点相关联的路径再现基元的类型的一或多个属性。

[0158] 在一些实例中,壳着色器48可处理从顶点着色器46和/或资源块64接收的控制点并且可针对由壳着色器48执行的壳着色器程序的每一个例产生输出控制点。举例来说,对于将由壳着色器48产生的每一输出控制点,壳着色器48可执行GPU 12的着色器单元上的壳着色器程序的个例。在一些实例中,壳着色器48可针对每一输出控制点执行“穿通”壳着色器程序。“穿通”壳着色器程序可引起壳着色器48针对每一输出控制点输出对应于输入控制点中的相应者的控制点。在此情况下,如果输出控制点具有与输入控制点相同的属性,那么

输出控制点可对应于输入控制点。

[0159] 在其它实例中,壳着色器48可产生与输入控制点中的相应者的输入属性不等同的输出控制点的一或多个输出属性。举例来说,壳着色器48可对输入控制点的属性中的一或多个者执行实质性处理以产生输出控制点的一或多个属性。作为另一实例,壳着色器48可从输入属性集合添加和/或删除属性以产生输出控制点的输出属性的集合。在一些实例中,如果GPU 12接收采用端点参数化形式的椭圆形弧的路径数据,那么壳着色器48可如下文进一步详细描述将椭圆形弧的端点参数化转换成椭圆形弧的中心参数化。

[0160] 在额外实例中,壳着色器48可丢弃针对特定再现操作将不再现的基元。丢弃基元可指代引起对应于基元的数据不传递到图形管线43的其它级上进而有效地引起此基元不由管线的剩余部分再现的过程。举例来说,当图形管线43正执行填充操作时,壳着色器48可丢弃接合点基元和盖基元。作为另一实例,当图形管线43正执行描画操作时,壳着色器48可针对开放路径丢弃闭合路径基元。闭合路径基元可指代表示闭合环的线路径片段的基元。闭合路径基元通常用于闭合路径而不是开放路径的路径。在一些实例中,闭合路径基元可由与用于识别路径中的其它线路径片段的基元类型识别符不同的基元类型识别符识别。举例来说,闭合路径基元可由基元类型识别符2.1f而非2.0f识别。

[0161] 壳着色器48还可执行用于每一路径片段的片恒定功能的个例。片恒定功能可确定并提供配置参数到曲面细分器50以在产生输出值时由曲面细分器50使用。举例来说,片恒定功能可引起壳着色器48将曲面细分因数提供到曲面细分器50。曲面细分因数可指定曲面细分器50适用于特定曲面细分域的曲面细分程度(例如,应细分域的精细程度和/或应将域细分成的较小对象的数目)。在一些实例中,壳着色器48可引起曲面细分器50执行用于三次贝塞尔曲线的4x曲面细分、用于圆形接合点和盖的4x曲面细分以及用于线段的1x曲面细分。

[0162] 作为另一实例,片恒定功能可引起壳着色器48将在曲面细分期间将使用的曲面细分域的类型提供到曲面细分器50。曲面细分域可指代由曲面细分器50使用以产生用于由域着色器52使用的多个坐标的对象。曲面细分域在概念上可对应于由曲面细分器50细分为多个较小对象的对象。较小对象的顶点的位置坐标随后发送到域着色器52用于进一步处理。在一些实例中,曲面细分域的类型可选择为方形、三角形和等值线中的一者。在一些实例中,域细分为的较小对象可对应于三角形、线段或点。在一些实例中,壳着色器48可指定等值线曲面细分域类型并指定曲面细分器50应将等值线域细分为线段。

[0163] 曲面细分器50可针对由曲面细分级62处理的每一路径片段产生多个输出值。所述输出值可确定表示特定类型路径片段的一或多个参数方程式应由域着色器52评估的值。在一些实例中,曲面细分器50可基于由壳着色器48提供到曲面细分器50的一或多个曲面细分因数和/或曲面细分域类型而产生多个输出值。举例来说,曲面细分器50可将等值线细分为多个线段,并针对归一化坐标系中的多个线段的每一端点产生输出值。

[0164] 域着色器52可从曲面细分器50接收输出值并从壳着色器48接收用于路径片段的控制点,并且产生对应于近似路径片段的曲率和/或形状的多个经曲面细分线段的输出顶点。举例来说,针对从曲面细分器50接收的输出值中的每一者,域着色器52可执行GPU 12的着色器单元上的域着色器程序的个例。所述域着色器程序可引起域着色器52针对从曲面细分器50接收的输出值中的每一者在基于相应输出值确定的特定值处评估一或多个参数方

程式以产生对应于所述相应输出值的输出顶点的位置坐标。用以产生输出顶点坐标的参数方程式的系数中的一或多者可基于从壳着色器48接收的控制点中的一或多者而定义。每一输出顶点可对应于多个经曲面细分线段中的一者的端点。两个连续输出顶点可对应于单个经曲面细分线段的端点。

[0165] 在额外实例中,域着色器程序可引起域着色器52产生对应于从曲面细分器50接收的输出值中的每一者的输出顶点的法线坐标。举例来说,域着色器程序可引起域着色器52针对从曲面细分器50接收的输出值中的每一者在基于相应输出值确定的特定值处评估一或多个额外参数方程式以便产生对应于所述相应输出值的输出顶点的切线坐标。输出顶点的切线坐标可指示在输出顶点处与路径片段交叉的路径片段的切线的方向。域着色器52可基于对应于相应输出顶点的切线坐标产生输出顶点中的每一者的法线坐标。针对特定输出顶点产生的法线坐标可指示法向量,所述法向量指示垂直于在输出顶点处与路径片段交叉的路径片段的切线的方向。

[0166] 在一些实例中,当图形管线43正执行填充操作时,域着色器52可产生对应于经曲面细分线段的端点的位置的顶点而无需产生此类位置的任何法线。在此等实例中,当图形管线43正执行描画操作时,域着色器52可在一些实例中产生对应于经曲面细分线段的端点的位置的顶点并产生对应于此类位置的法线。

[0167] 域着色器52可以经排序序列输出顶点,其中邻近顶点的每一集合表示经曲面细分线段。线段可共同地近似在顶点缓冲器中定义的路径片段。举例来说,域着色器52可输出以下顶点集合{0,1,2,3,4,5},其定义以下线段:{0,1}、{1,2}、{2,3}、{3,4}、{4,5}。在额外实例中,域着色器52可输出以下顶点集合{0,1,1,2,2,3,3,4,4,5},其可定义与先前实例中所列相同的线段。

[0168] 在一些实例中,曲面细分器50和域着色器52可经配置以根据以下技术均匀地将路径片段曲面细分为多个线段。具体来说,曲面细分器50可输出用于参数评估的坐标(例如, $t=0/T, 1/T, 2/T \dots T/T$,其中 T 是曲面细分因数)。取决于基元的类型,域着色器52可在由曲面细分器50输出的值处评估一或多个参数方程式。

[0169] 几何形状着色器54可从域着色器52接收经曲面细分线段并基于经曲面细分线段产生多个基元。以此方式,几何形状着色器54可确定线段的片段次序。针对经曲面细分线段中的每一者,几何形状着色器54可执行GPU 12的着色器单元上的几何形状着色器程序的个例,并基于相应经曲面细分线段产生用于经曲面细分线段的三角形基元。在一些实例中,针对经曲面细分线段中的每一者,几何形状着色器54可从域着色器52接收对应于相应经曲面细分线段的两个顶点,并产生对应于三角形基元的一组三个顶点。

[0170] 在一些实例中,三角形基元的所述顶点中的两者可为与两个所接收顶点相同的顶点(例如,具有相同位置坐标)。在此些实例中,几何形状着色器54可基于对于与待再现路径片段相关联的全部经曲面细分线段为共同共同顶点产生第三顶点。所述共同顶点可或可不对应于经曲面细分线段的端点中的一者。在一些实例中,所述共同顶点可对应于对应于用于待再现路径片段的经曲面细分线段的一组顶点中的第一顶点。

[0171] 几何形状着色器54可针对由域着色器52产生的经曲面细分线段中的每一者调用一次。针对经曲面细分线段中的每一者,几何形状着色器54可使用共同控制点作为三角形的第一顶点并使用相应经曲面细分线段的两个端点作为三角形的第二和第三顶点来产生

三角形基元。举例来说,以上提供实例,其中域着色器52产生以下顶点集合{0,1,2,3,4,5},其定义以下线段:{0,1}、{1,2}、{2,3}、{3,4}、{4,5}。对于以上所列线段序列,几何形状着色器54可产生以下三角形:{C,0,1}、{C,1,2}、{C,2,3}、{C,3,4}、{C,4,5}、{C,4,5},其中C是对于全部三角形为共同的任何单一顶点。

[0172] 光栅化器56可经配置以将多个3D图形基元(例如,点、线和三角形)转换为对应于3D图形基元的多个像素。举例来说,光栅化器56可接收对应于三角形基元的三个顶点,并将所述三个顶点转换为对应于由三角形基元覆盖的屏幕像素位置的多个像素。由三角形基元覆盖的屏幕像素位置可包含对应于三角形的顶点、三角形的边缘和三角形的内部的屏幕像素位置。

[0173] 像素着色器58可从光栅化器56接收像素,并根据像素着色器程序基于所接收的像素产生经着色像素。举例来说,针对从光栅化器56接收的每一像素,像素着色器58可执行GPU 12的着色器单元上的像素着色器程序的个例。在一些实例中,像素着色器58可针对每一像素执行“穿通”像素着色器程序。“穿通”像素着色器程序可引起像素着色器58针对每一像素输出对应于输入像素中的相应者的像素。在此情况下,如果输出像素具有与输入像素相同的属性,那么输出像素可对应于输入像素。

[0174] 在其它实例中,像素着色器58可产生与输入像素中的相应者的输入属性不等同的输出像素的一或多个输出属性。举例来说,像素着色器58可对输入像素的属性中的一或多个者执行实质性处理以产生输出像素的一或多个属性。作为另一实例,像素着色器58可从输入属性集合添加和/或删除属性以产生输出像素的输出属性的集合。

[0175] 输出合并器60可将从像素着色器58接收的像素数据放置到再现目标(例如,帧缓冲器或模板缓冲器)中。在一些实例中,输出合并器60可基于光栅操作将从像素着色器58接收的像素数据与已经存储于再现目标中的像素数据合并。

[0176] 为了执行路径填充操作,光栅化器56可将由几何形状着色器54接收的三角形中的每一者光栅化到共同模板缓冲器(例如,存储于资源块64中的缓冲器)中。在第一遍次期间,像素着色器58可停用或设定成“穿通”模式以将输入像素直接传递到输出合并器60。输出合并器60可经配置以根据一或多个模板缓冲器填充技术填入模板缓冲器以使得模板缓冲器存储指示用于路径片段的填充区域的值。

[0177] 根据本发明的各方面,如上所述,GPU 12可用使用模板印刷TIR和限界框的单一遍次方法执行填充操作,所述方法涉及以下步骤:

[0178] 1.将路径片段曲面细分为多个线段。

[0179] 2.产生每个线段的三角形基元。

[0180] 3.将全部三角形基元再现到模板缓冲器中。

[0181] 4.在模板印刷期间确定限界框。

[0182] 5.使用模板印刷TIR再现限界框。

[0183] 在以上实例中,GPU 12可将指示应重设限界框参数(例如,bb_box)的事件(例如,bb_start)发送到曲面细分62(在本文中也称作曲面细分引擎)。GPU 12可接着在使用上文所描述的过程更新模板缓冲器时产生三角形基元。另外,曲面细分62通过比较最小最大参数与顶点数据来更新限界框参数(bb_box)。也就是说,曲面细分62可检查顶点中的每一者以例如使用笛卡尔坐标确定顶点是否具有进一步位于任何先前确定的顶点之上、

下、右或左的位置。如果顶点定位在相对于其它顶点的最外位置处,那么曲面细分级62可更新所述限界框参数(bb_box)。

[0184] 当曲面细分级62接收限界框结束事件(例如,bb_end)时,所述曲面细分级可产生对应于所确定的限界框坐标的rectlist,例如,形成包含路径的三角形基元的限界框。光栅化器56可接着光栅化所述限界框。根据本发明的各方面,光栅化器56可执行模板印刷TIR,其中所述经模板印刷像素相对于再现目标经超取样,并且像素着色器58仅着色经模板印刷像素。当像素着色器58着色像素,可自模板缓冲器清除用于所述像素的模板值。

[0185] 因此,在上文所描述的实例中,曲面细分级62维持限界框开始事件(bb_start)和限界框结束事件(bb_end)之间的限界框参数(bb_box)的累加。所述再现后端(例如,包含几何形状着色器54、光栅化器56、像素着色器58和/或输出合并器60)预期限界框开始事件(bb_start)与限界框结束事件(bb_end)之间的固定操作。也就是说,所述再现后端可执行与确定所述限界框相关联的操作而无需驱动程序(例如GPU驱动程序28(图2))编程可分配在资源块64中的再现后端寄存器。虽然关于曲面细分级62进行了描述,但应理解所述以上技术可由图形管线的一或多个其它级执行。以此方式,GPU 12可使用图形管线43在单一遍次中填充路径并且在单独遍次期间不需要再现限界框。

[0186] 根据本发明的其它方面,图形管线43可经配置以针对所描画路径片段执行划短划线。在出于说明目的的实例中,几何形状着色器54可从域着色器52接收经曲面细分线段并基于所述经曲面细分线段产生多个基元。所述多个基元可包含待着色的短划线片段,并且所述多个基元可按特定次序,例如,按片段次序。几何形状着色器54(或图形管线43的另一组件)还可确定短划线中的每一者的长度。

[0187] 另外,几何形状着色器54在产生每一短划线时累加所述短划线的长度并且为每一短划线片段指派先前短划线片段(例如,片段次序先于所述当前片段的短划线片段)的长度的前缀总和。举例来说,第一短划线片段可指派有为零的前缀总和,第二短划线片段可指派有为第一短划线片段的长度的前缀总和,第三短划线片段可指派有为第一和第二短划线片段的组合长度的前缀总和,等等。

[0188] 光栅化器56通常接收短划线片段并在光栅化期间遵循基元次序,其中所述基元次序指代用于再现的次序。在光栅化后,可将每一短划线片段的前缀总和发送到像素着色器58以在着色所述短划线片段时使用。举例来说,为了着色适当位置中的短划线片段,像素着色器58可将每一短划线片段的前缀总和和应用作为纹理偏移。所述纹理偏移指示先前短划线片段的位置,进而允许像素着色器58着色相对于先前片段在适当位置中的下一短划线片段。

[0189] 图4为待再现的实例路径80的图。举例来说,路径80表示具有圆形顶部和细长底部的“冰淇淋锥体”形状。路径80可为由两个三次曲线组成的闭合路径。片段可组装成片的4个基元(基元)。举例来说,用于路径80的输入路径数据可采取以下形式或类似形式:

[0190] {XMFLOAT4(0.0f,0.0f,CUBIC,0.0f)},

[0191] {XMFLOAT4(0.4f,1.2f,0.0f,0.0f)},

[0192] {XMFLOAT4(1.4f,1.2f,0.0f,0.0f)},

[0193] {XMFLOAT4(1.8f,0.0f,0.0f,0.0f)},

[0194] {XMFLOAT4(1.8f,0.0f,CUBIC,0.0f)},

[0195] {XMFLOAT4 (0.5f, -3.0f, 0.0f, 0.0f) },

[0196] {XMFLOAT4 (1.3f, -3.0f, 0.0f, 0.0f) },

[0197] {XMFLOAT4 (0.0f, 0.0f, 0.0f, 0.0f)

[0198] 在此实例中,每一行表示顶点或控制点,并且圆括号中的每一参数表示相应顶点或控制点的属性。在此实例中,第一控制点的最后属性存储指示待再现路径片段的类型的数据(即,“路径片段类型指示符”)。具体来说,在此实例中路径片段类型指示符是0.0f,其意味着所述路径片段是三次贝塞尔曲线路径片段。在一些实例中,路径片段的其它属性可包含类似于上文关于线路径片段描述的属性。

[0199] 图5A至5C是说明用于图4中所展示的路径80的实例填充操作的一系列图。路径80在出于说明目的的图5A至5C的实例中是欠曲面细分的(例如,具有比典型的少的片段)。另外,虽然出于解释的目的关于GPU 12进行了描述,但图5A至5C中进行的过程可由多种其它处理器来执行。

[0200] 如图5A中所展示,GPU 12曲面细分路径80以包含以线条方式84连接的数个顶点82。如图5B中所展示,GPU 12产生连接到枢转点88以形成数个三角形基元的数个线段86。在图5B的实例中,路径80的相对第一顶点用作枢转点88。三角形的卷绕次序将确定适当模板操作。举例来说,每一所产生线段连接到枢转点88。三角形的所得定向(例如,顺时针或逆时针)可确定三角形基元的卷绕次序。所述卷绕次序可以不同方式影响模板值(例如,针对顺时针卷绕次序递增模板值或针对逆时针卷绕次序递减模板值)。

[0201] 在此实例中,GPU 12在模板印刷期间并不着色图5B中所展示的三角形基元。确切地说,如上所述,在模板印刷期间所再现的三角形基元仅影响模板纹理90。也就是说,模板纹理90指示图像中呈现的路径的部分,例如,所再现并着色的部分。

[0202] 如图5C中所展示,GPU 12确定包含模板纹理90的限界框92。也就是说,所述限界框覆盖待填充的整个路径。GPU 12接着对限界框92执行模板印刷TIR来产生经填充路径96。以此方式,GPU 12在单一再现遍次中确定限界框92并填充路径80。

[0203] 图6是说明模板印刷操作的概念图。举例来说,出于说明的目的假定GPU 12使用16x MSAA再现基元100。在此实例中,每一正方形表示像素102的样本。

[0204] 根据本发明的各方面,GPU 12可执行经模板印刷TIR。因此,GPU 12可独立于再现目标参数(例如,用于所再现的像素的存储器分配)确定模板参数(例如,模板取样速率)。在此实例中,GPU 12可基于像素是否通过每样本模板测试来确定用于再现所述像素的覆盖值。

[0205] 在一些实例中,GPU 12可执行模板测试以确定样本是否具有非零值。举例来说,GPU 12可执行零/非零模板测试,其中再现具有非零模板值的样本。在另一实例中,GPU 12可执行奇/偶模板测试,其中再现具有奇数(或偶数)值的样本。因此,在一些实例中,GPU 12可执行模板测试以确定样本是否具有奇数值。在其它实例中,GPU 12可执行模板测试以确定样本是否具有偶数值。

[0206] 在任何情况下,在图6中所展示的实例中,像素102的16个样本(再次,由方框表示)中的10个位于基元100内。因此,用于基元100的覆盖蒙板可包含像素102,并且GPU 12可在再现期间着色像素102。

[0207] 图7是根据本发明的各方面说明实例填充操作的概念图。举例来说,图7说明确定

三角形基元和基于基元110的定向更新模板缓冲器,基于所述基元的最外点、模板缓冲器114的内容以及所绘制限界框和经模板印刷像素116确定限界框112。

[0208] 根据一些实例,在模板印刷期间确定限界框和再现模板中所包含的基元的所述序列可由GPU 12使用以下API调用执行:

[0209] Draw_BB();//compute bounding box of primitives within the draw call

[0210] //or it can be BeginQuery()...EndQuery()

[0211] //if pixel shader is not bound,then only bounding box is

[0212] //computed and no primitives are rendered

[0213] Render_BB();//render the previously computed bounding box,or it can

[0214] //DrawIndirect-like method

[0215] //same as draw call,except no primitives are sent,but pixel

[0216] //shader and other back-end states are specified

[0217] 其中Draw_BB指示GPU 12在模板印刷期间确定限界框112(而不再现像素),并且Render_BB指示GPU 12在限界框的顶部上执行模板印刷TIR。以此方式,GPU 12可在单一再现遍次中确定限界框并执行路径填充。

[0218] 图8是说明根据本发明的各方面的再现期间的存储器带宽的曲线图。举例来说,图8针对以下三个再现方案中的每一者说明三个不同图形保真速率(4x、8x和16x)以及与所述速率相关联的相关联数据传送:MSAA方案120,其中执行曲面细分和几何形状着色;模板印刷TIR方案122;和保守光栅化方案124(例如,Loop-Blinn过程)。图8中所展示的带宽需求与以60帧每秒(fps)再现的图像的测试序列和具有32位元的色彩、24位元的深度和8位元的模板的缓冲器格式相关联。

[0219] 如图8中的曲线图所展示,MSAA 120的存储器带宽需求明显地大于模板印刷TIR122和保守光栅化124的存储器带宽需求。下文展示的表1说明MSAA 120、保守光栅化124和模板印刷TIR 122的比较。

[0220] 表1

[0221]

	MSAA	保守光栅化	模板印刷 TIR
品质	类似	稍微较佳	类似
存储器带宽	高	低(每样本 1 像素)	低(模板+每样本 1 像素)
像素着色	无	极高	无
HW 改变	已存在	小	中等
API 改变	已存在	小	中等
几何形状操作	CPU (移动到曲面细分引擎)	CPU (复杂算法)	较少 CPU (移动到曲面细分引擎)
一般路径再现	是	否	是
重叠路径	是	否	是

[0222] 下文展示的表2说明MSAA 120、模板印刷TIR 122和保守光栅化124之间的额外比较。

[0223] 表2

[0224]

	Nx MSAA:	Nx TIR + 模板:	保守光栅化
样本测试(HW)	Nx	Nx	1x

[0225]

深度缓冲器更新	Nx	无	无
模板缓冲器更新	Nx	Nx	无
色彩缓冲器更新	Nx	1x	1x
像素着色器启动	1x	1x	1x (相对复杂的像素着色级)

[0226] 图9A至9D是说明用于图4中所展示的路径的实例划短划线操作的一系列图。再次，路径80在出于说明目的的图9A至9D的实例中欠曲面细分。另外，虽然出于解释的目的关于GPU 12进行了描述，但图9A至9D中进行的过程可由多种其它处理器来执行。

[0227] 如图9A中所展示，GPU 12曲面细分路径80以包含以线条84连接的数个顶点82。另外，GPU 12确定数个法线130 (展示为从顶点延伸的箭头)。在图9A的实例中，双法线130指示接合点位置。产生接合点可能需要线条中的下一基元的端点切线。图9A还包含实例片段132。

[0228] 图9B说明GPU 12对片段132执行充实操作，其中所述片段132在+/-法线方向上经充实描画宽度/2。出于说明的目的在图9B的实例中展示附加稠密描画线。图9B说明经充实片段134和经描画充实片段136。

[0229] 图9C说明用短划线划出经描画充实片段136。举例来说，GPU 12可确定第一短划线片段138、第二短划线片段140和第三短划线片段142。在此实例中，第一短划线片段138和第三短划线片段142是可见片段，而第二短划线片段140是不可见短划线片段。对于划短划线，GPU 12确定每一短划线片段(线) 138、140和142的起始位置。如上所述，在一些实例中，前缀总和单元42可在几何形状着色期间累加短划线片段138至142的长度。

[0230] GPU 12可将长度应用作为纹理坐标从0到线长度L的纹理偏移。举例来说，根据本发明的各方面，前缀总和单元42可计算指示片段138至142中的每一者的位置的线长值。前缀总和单元42可将前缀求和线长值发送到像素着色器级，所述像素着色器级在像素着色期间确定片段138至142的相应位置。GPU 12保留形成可见短划线图案的部分的可见片段138和142 (作为着色分段) 并舍弃在所述短划线图案中不可见的片段140 (未着色)。

[0231] 在一些实例中，支持确定例如点大小的每基元标量值的图形API可用于指示GPU 12确定标量长度。根据本发明的各方面，所述图形API可支持线长值。此线长值可为基元的同一平坦属性，但可将所述属性提供到像素着色器级。举例来说，GPU 12可应用每经曲面细分基元的前缀总和路径来在像素着色期间确定偏移坐标。另外，API调用prsum_start、prsum_end (类似于query_start/end) 可为一或多个绘图调用加括号 (bracketize) 来指示虚线的相对开始和结束。

[0232] 图9D说明用短划线划出的描画线142与填充路径144的相加以产生经填充和用短划线划出的路径146。

[0233] 图10是说明根据本发明的用于执行填充操作的实例技术的流程图。虽然出于说明的目的描述为由GPU 12执行，但应理解图10中所展示的技术可由多种其它处理器执行。另外，与所展示的步骤相比，较少、额外或不同步骤可用于执行所述技术。

[0234] 在图10的实例中, GPU 12接收路径数据(160)。所述路径数据可指示待再现的路径的一或多个路径片段。GPU 12还确定模板参数(162)。在一些实例中, 所述模板参数可指示取样速率, 所述取样速率用于确定所述路径的每一经图形保真的像素的覆盖值。GPU 12还单独地根据模板参数确定再现目标参数(164)。所述再现目标参数可指示所述路径的每一经图形保真的像素的存储器分配。

[0235] GPU 12将由路径数据定义的路径片段曲面细分为多个线段(166)。举例来说, GPU 12可将路径数据曲面细分为例如图5A中所展示的线条的线条。GPU 12接着基于所述多个线段产生多个三角形基元(168)。可基于所述多个线段中的相应者产生所述多个三角形基元中的每一者。用于给定路径片段的所述多个三角形基元中的每一者可共享共同顶点。三角形基元中的每一者的另两个顶点可对应于所述多个线段中的相应者的端点。

[0236] GPU 12使用所述模板参数将所述多个三角形基元中的每一者再现到共同模板缓冲器并确定限界框(170)。举例来说, 如上所述, GPU 12在模板印刷期间并不着色所述三角形基元。然而, GPU 12可确定三角形基元的最外点以确定用于基元的限界框。在一些实例中, GPU 12可确定每一三角基元的坐标并在每当基元延伸超过先前三角基元的最外点时覆盖上边界点、下边界点、右边界点和/或左边界点。

[0237] 在将全部三角形基元再现到模板缓冲器中之后, 模板缓冲器可存储指示哪些像素在用于路径片段的填充区域内部的数据。另外, 所述限界框包含三角形基元中的每一者。

[0238] GPU 12接着使用所述再现目标参数和模板缓冲器光栅化限界框(172)。举例来说, 根据本发明的各方面, GPU 12对限界框执行模板印刷TIR以确定路径数据的每一像素的色彩值。模板缓冲器中的数据可引起用填充色彩着色填充区域内部的像素并引起将填充区域外部的像素保持为不着色的。一旦完成限界框的再现, 所述再现目标(例如, 帧缓冲器)就可存储使用再现目标参数的用于路径片段的填充区域的光栅化版本。

[0239] 图11是说明根据本发明的用于执行描画操作的实例技术的流程图。再次, 虽然出于说明的目的描述为由GPU 12执行, 但应理解图11中所展示的技术可由多种其它处理器执行。另外, 与所展示的步骤相比, 较少、额外或不同步骤可用于执行所述技术。

[0240] GPU 12接收路径数据(180)。路径数据可指示待再现路径的一或多个路径片段。GPU 12将由路径数据定义的路径片段曲面细分为多个线段(182)。举例来说, GPU 12可将路径数据曲面细分为例如图9A中所展示的线条的线条。

[0241] GPU 12产生在空间上对应于用于路径片段的描画区域的多个基元(184)。举例来说, 对于多个经曲面细分线段中的每一者, GPU 12可产生在空间上对应于用于相应线段的描画区域的一或多个基元。GPU 12可在线段的几何形状着色期间确定每线段的经曲面细分基元的数目。也就是说, 在几何形状着色期间, GPU 12可产生描画线“而无需划短划线”(例如, 无需着色描画线的特定片段)。

[0242] 在划短划线时, GPU 12确定每经曲面细分基元的路径长度(186)。举例来说, GPU 12可确定在几何形状着色期间产生的每一短划线片段(基元)的长度的累加。也就是说, 可以特定次序(例如, 在曲面细分和/或几何形状着色期间确定的次序)来排序所述短划线片段。对于每一基元, GPU 12可累加次序在其之前的基元的长度。

[0243] GPU 12可基于长度累加确定正再现的每一基元的纹理坐标的纹理偏移(188)。举例来说, 如上所述, GPU 12可使用长度信息确定基元中的每一者的开始的纹理坐标。GPU 12

可在像素着色期间应用所述纹理偏移(190)。举例来说, GPU 12应用所述纹理偏移并使用用于所描画路径数据的适当色彩来着色短划线的片段中的每一者。

[0244] 在一些实例中, 本发明的技术可允许DirectX 11硬件的用户使用DirectX 11硬件或以具有类似性能特性的硬件执行路径再现。在其它实例中, 本发明的技术可提供对路径再现的全部GPU再现解决方案。

[0245] 尽管本发明的技术已经主要关于由DX 11图形API定义的硬件架构进行描述, 但本发明的技术也可以在根据其它芯片上具有曲面细分功能的图形API定义的硬件架构中执行, 例如OpenGL图形API (例如, OpenGL版本4.0、4.1、4.2、4.3和更晚的版本)。在本发明的技术在根据OpenGL图形API定义的硬件架构中实施的实例中, 本发明中归于壳着色器48的功能中的一或多者可由曲面细分控制着色器执行, 和/或本发明中归于域着色器52的功能中的一或多者可由曲面细分评估着色器执行。

[0246] 本发明中所描述的技术可至少部分实施于硬件、软件、固件或其任何组合中。举例而言, 所描述技术的各种方面可实施于一或多个处理器中, 包含一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA), 或任何其它等效集成或离散逻辑电路, 以及此等组件的任何组合。术语“处理器”或“处理电路”可一般指代前述逻辑电路中的任一者(单独或结合其它逻辑电路)或例如执行处理的离散硬件的任何其它等效电路。

[0247] 此硬件、软件和固件可实施于同一装置内或单独装置内以支持本发明中所描述的各种操作和功能。另外, 所描述单元、模块或组件中的任一者可一起或单独作为离散但可互操作逻辑装置而实施。将不同特征描述为模块或单元意在强调不同功能方面并且未必暗示此等模块或单元必须由单独硬件或软件组件实现。确切地说, 与一或多个模块或单元相关联的功能性可由单独硬件、固件和/或软件组件执行, 或集成到共用或单独硬件或软件组件内。

[0248] 本发明中所描述的技术也可存储、体现或编码于计算机可读媒体(例如, 存储指令的计算机可读存储媒体)中。嵌入或编码于计算机可读媒体中的指令可引起一或多个处理器执行本文中所描述的技术(例如, 当由一或多个处理器执行指令时)。计算机可读存储媒体可包含随机存取存储器(RAM)、只读存储器(ROM)、可编程只读存储器(PROM)、可擦除可编程只读存储器(EPROM)、电子可擦除可编程只读存储器(EEPROM)、快闪存储器、硬盘、CD-ROM、软盘、卡盒、磁性媒体、光学媒体或其它有形计算机可读存储媒体。

[0249] 计算机可读媒体可包含计算机可读存储媒体, 其对应于例如上文所列的有形存储媒体等有形存储媒体。计算机可读媒体也可包括通信媒体, 其包含有助于计算机程序从一个地点到另一地点的传送(例如, 根据通信协议)的任何媒体。以此方式, 短语“计算机可读媒体”一般可对应于(1)非暂时性有形计算机可读存储媒体, 和(2)例如暂时性信号或载波等非有形计算机可读通信媒体。

[0250] 已描述各种方面和实例。然而, 可在不脱离以下权利要求书的范围的情况下对本发明的结构或技术作出修改。

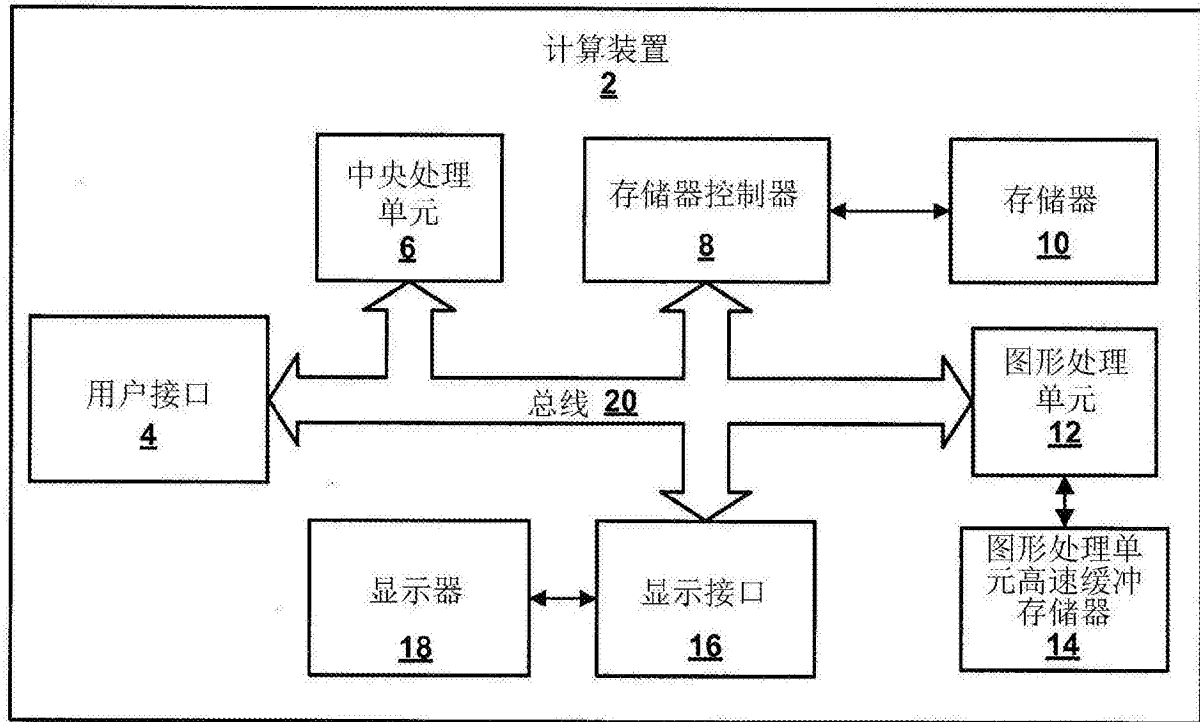


图1

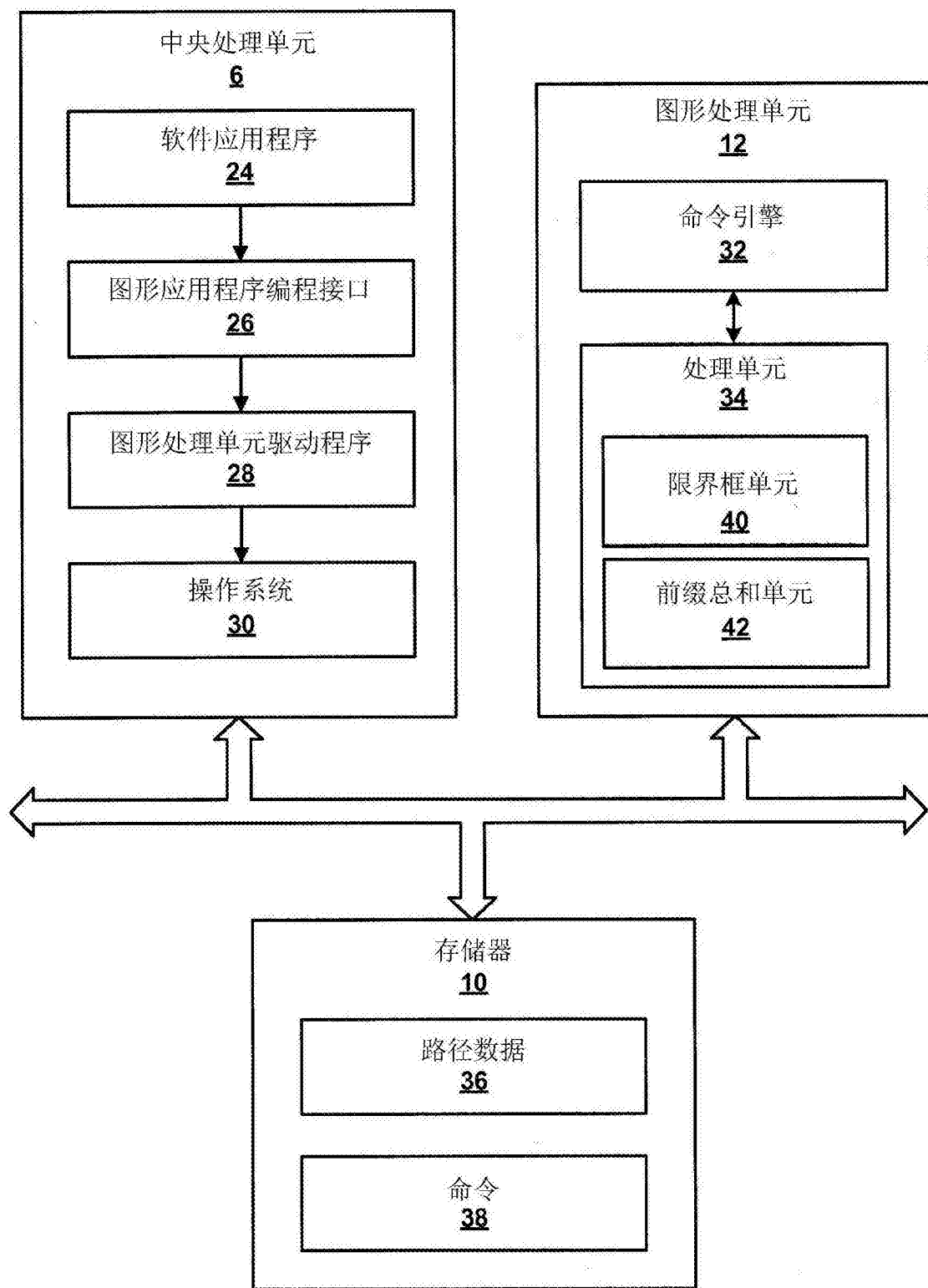


图2

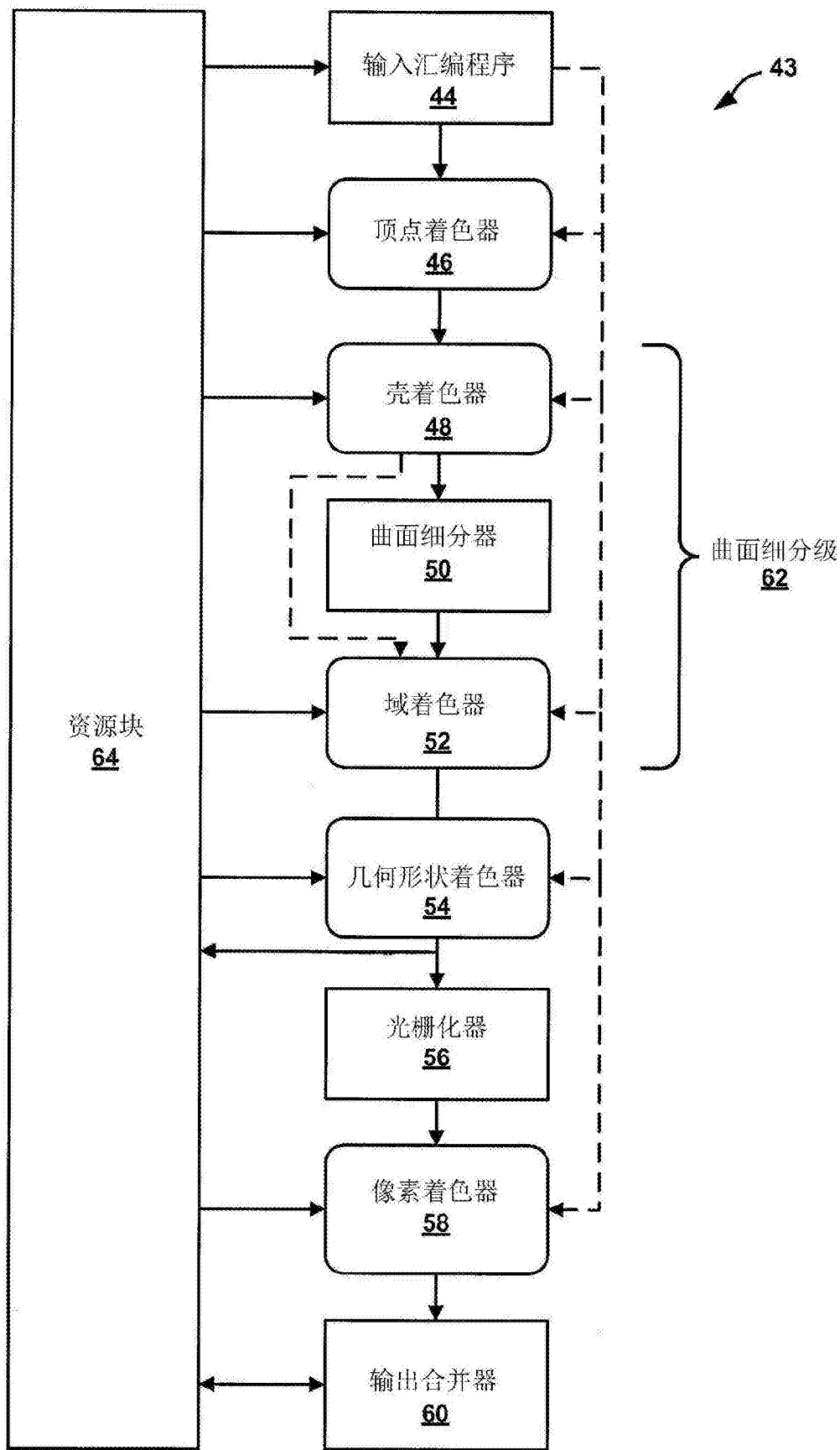


图3

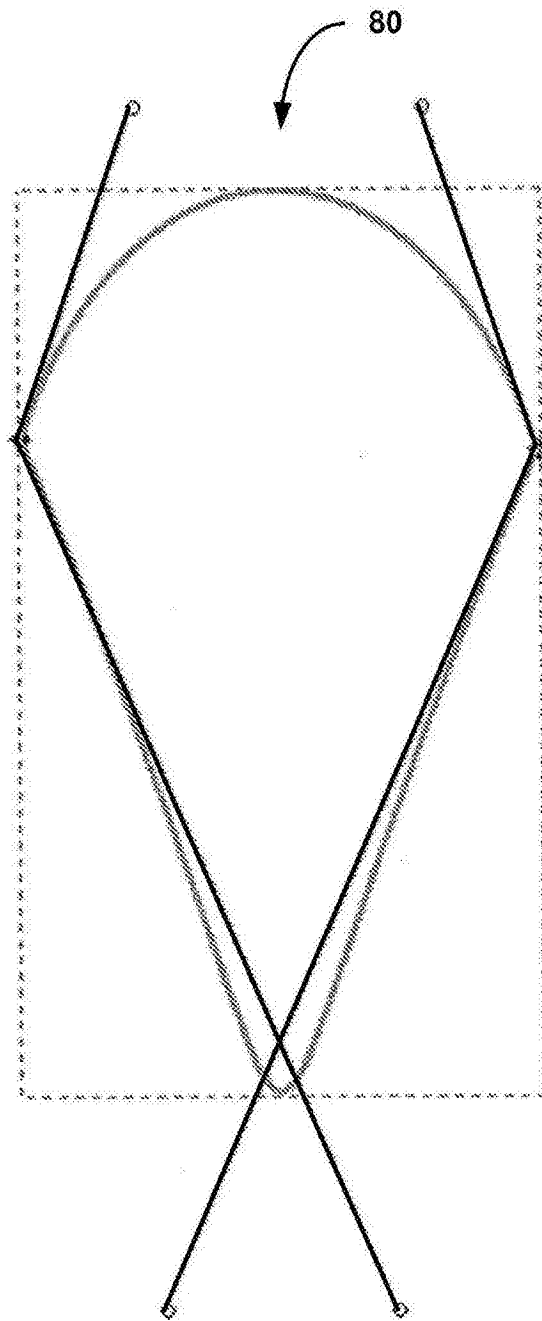


图4

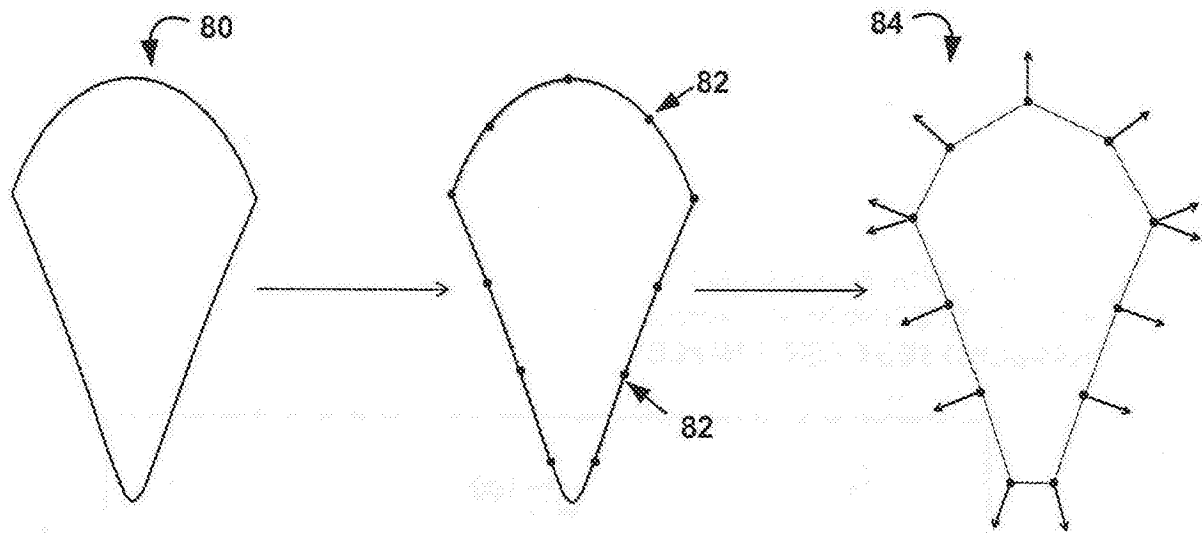


图5A

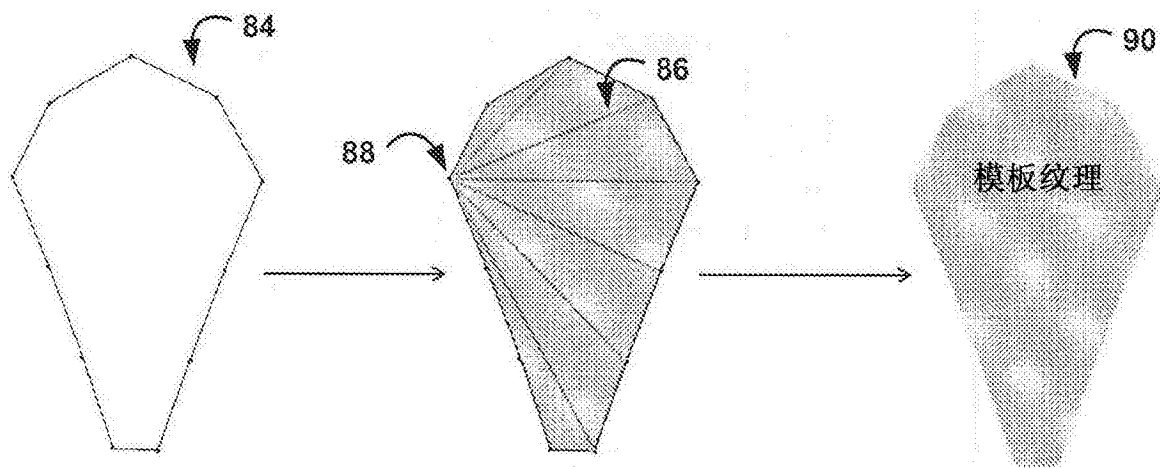


图5B

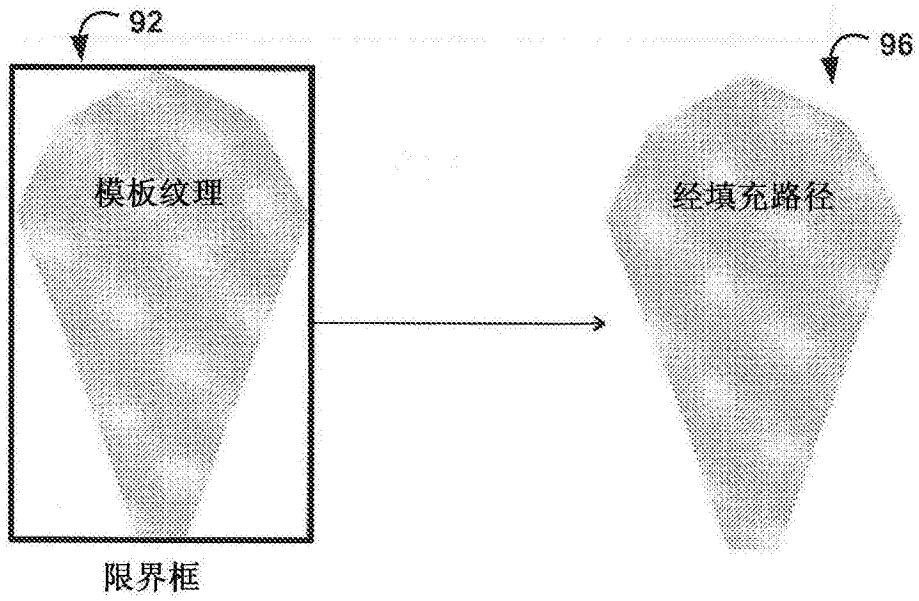


图5C

覆盖为 $10/16=5/8$
基于每样本通过非零测试的模板

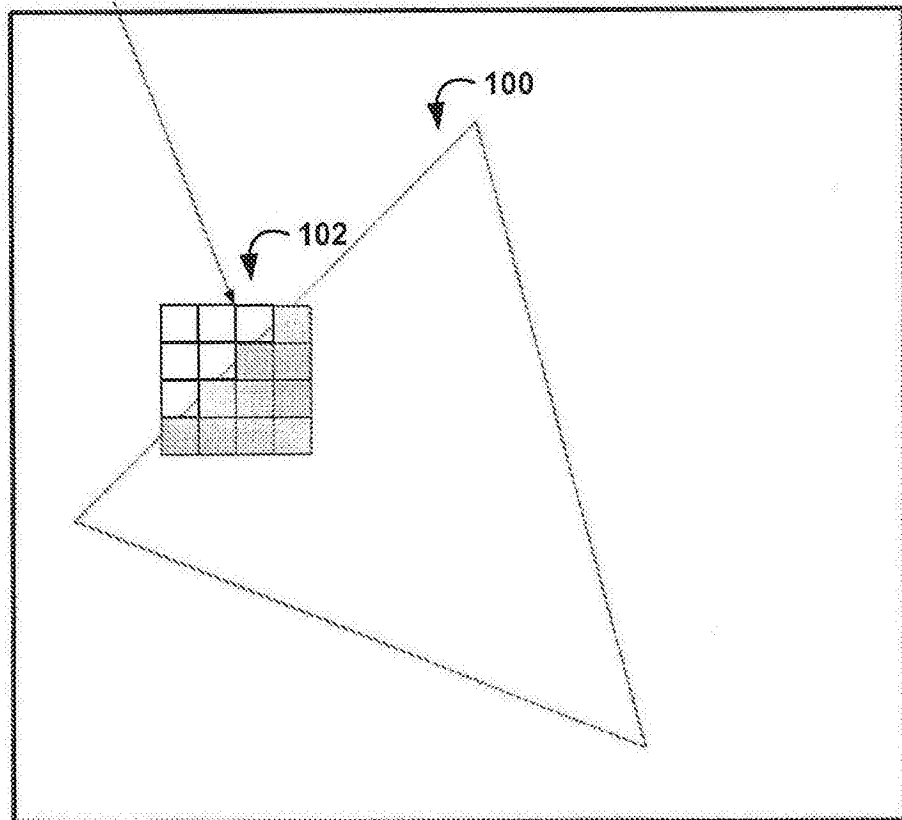


图6

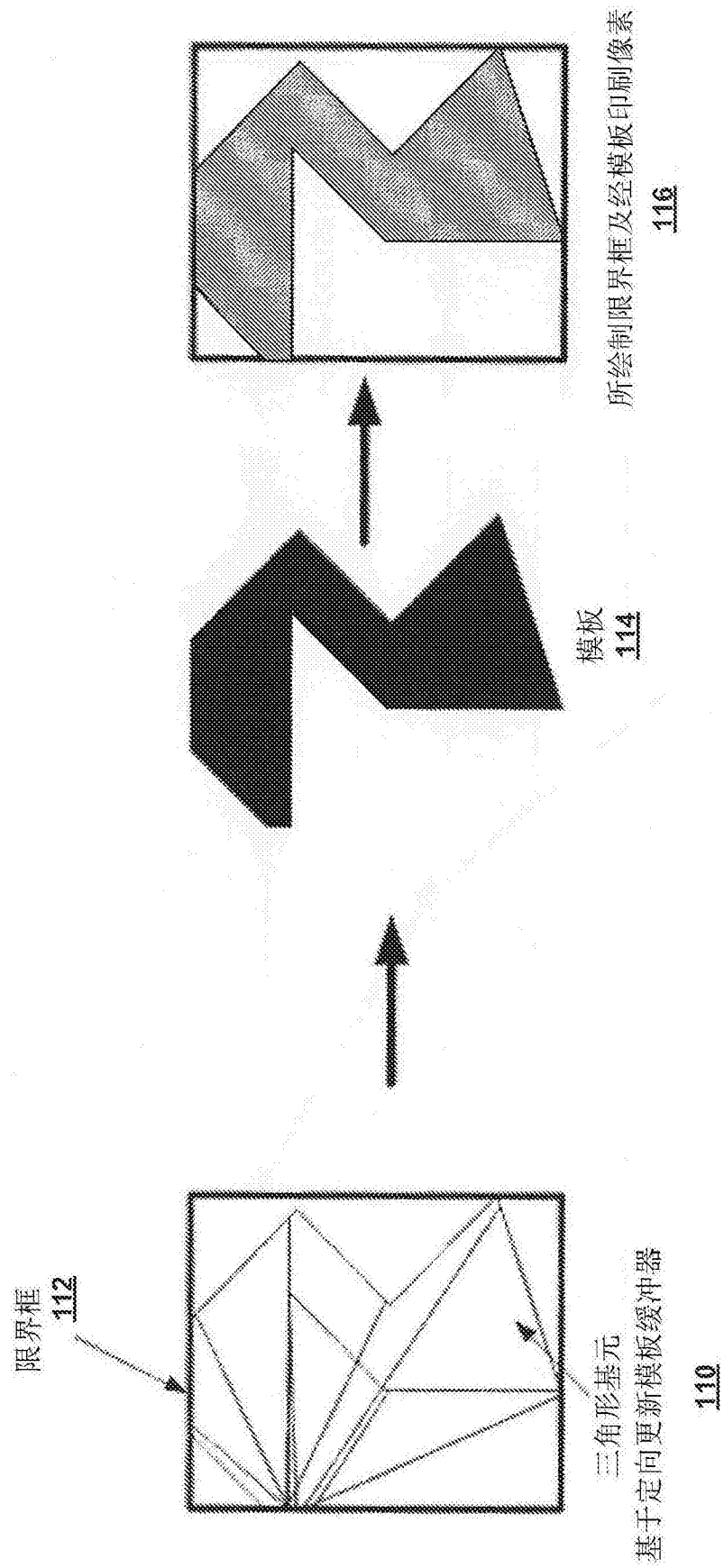


图7

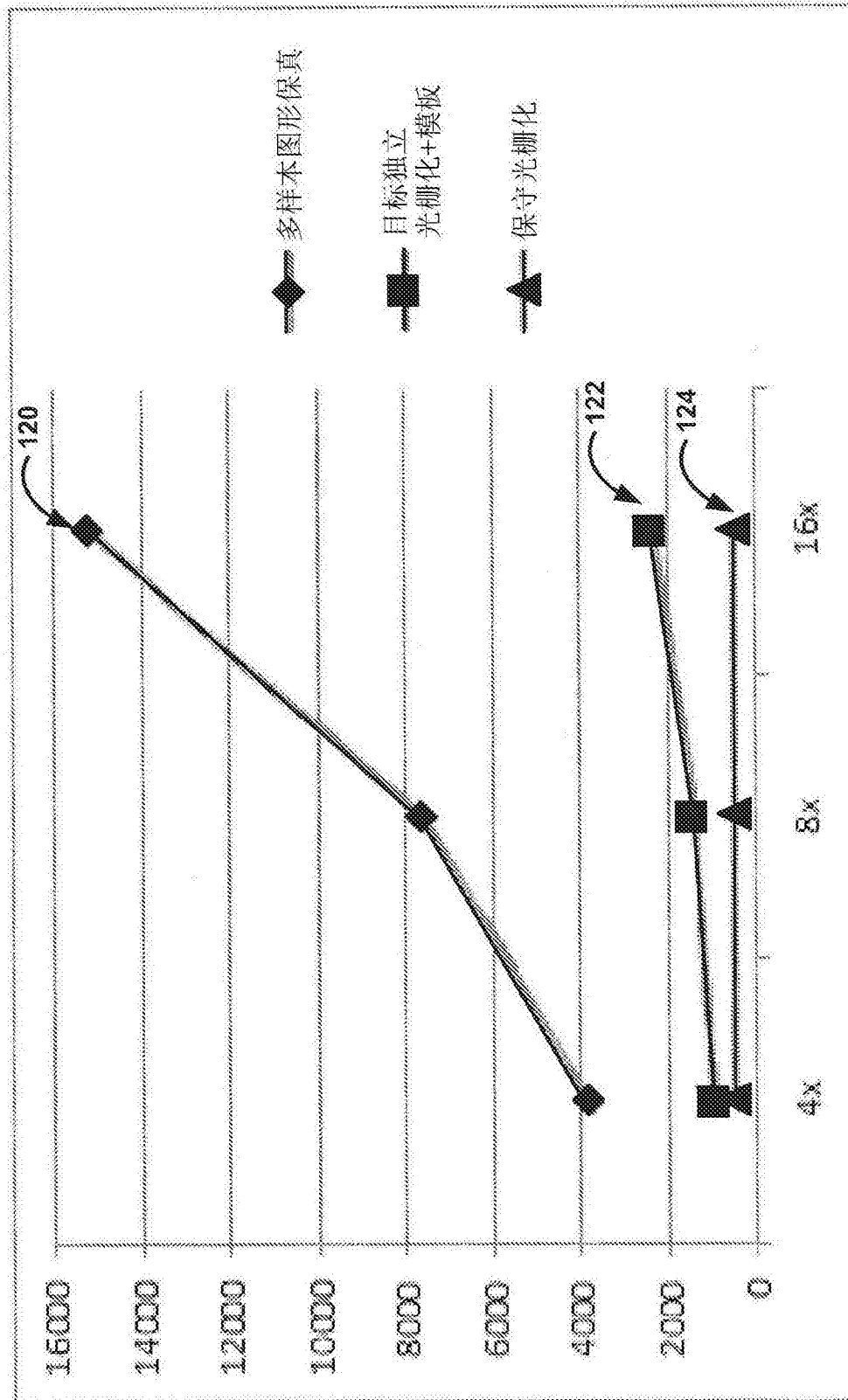


图8

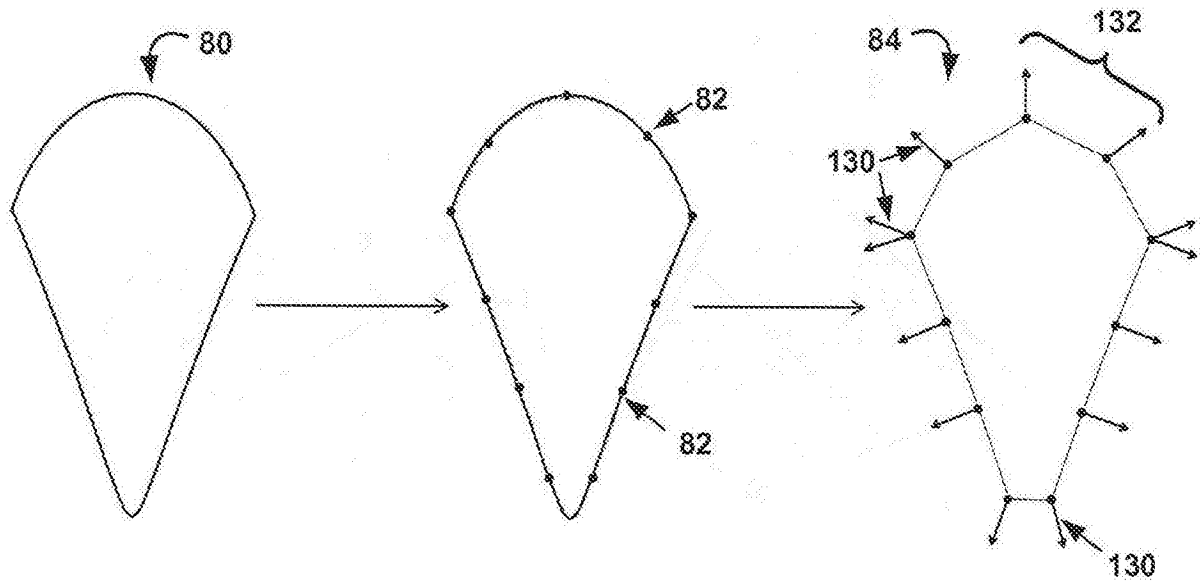


图9A

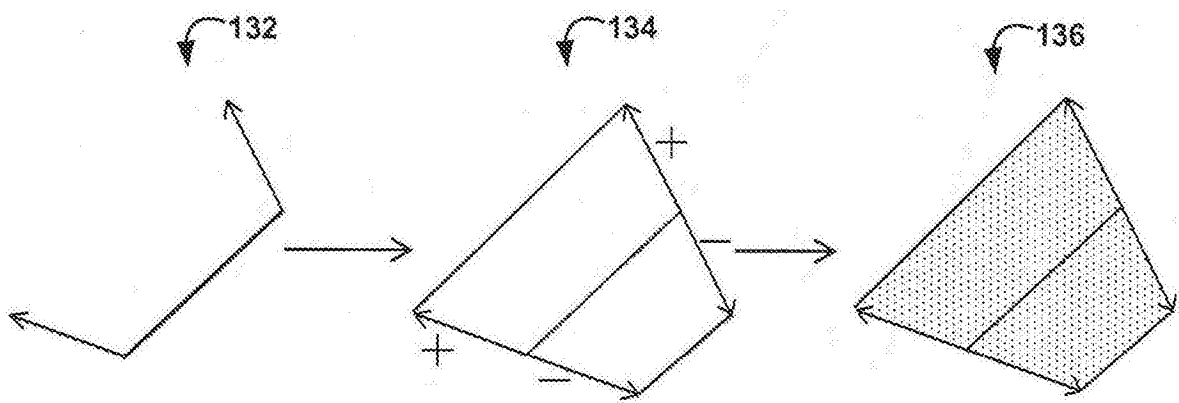


图9B

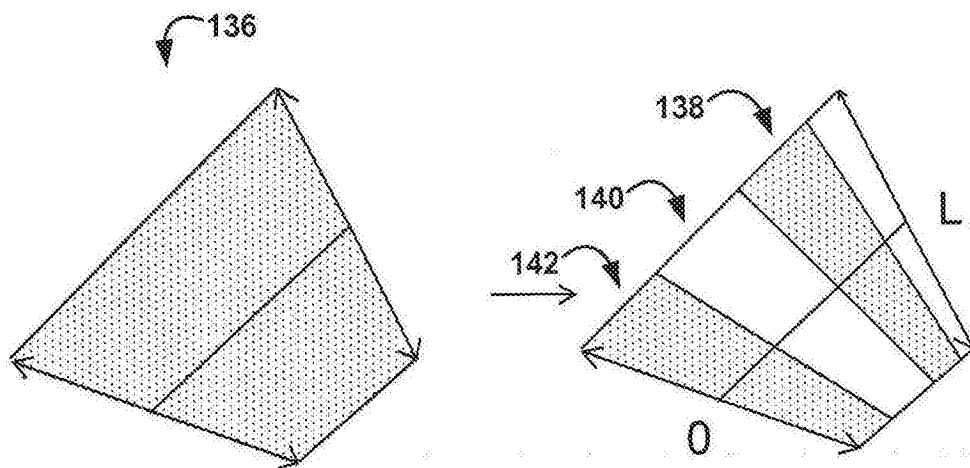


图9C

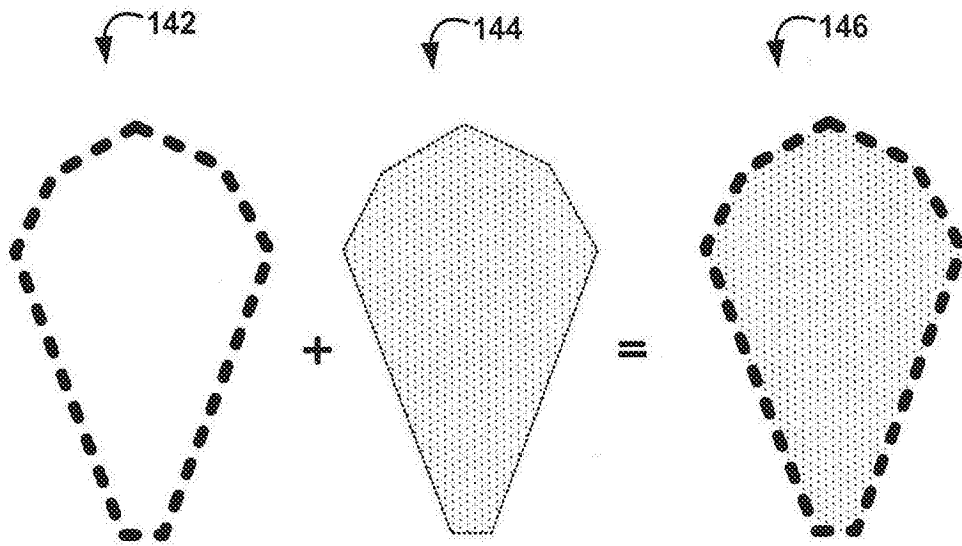


图9D

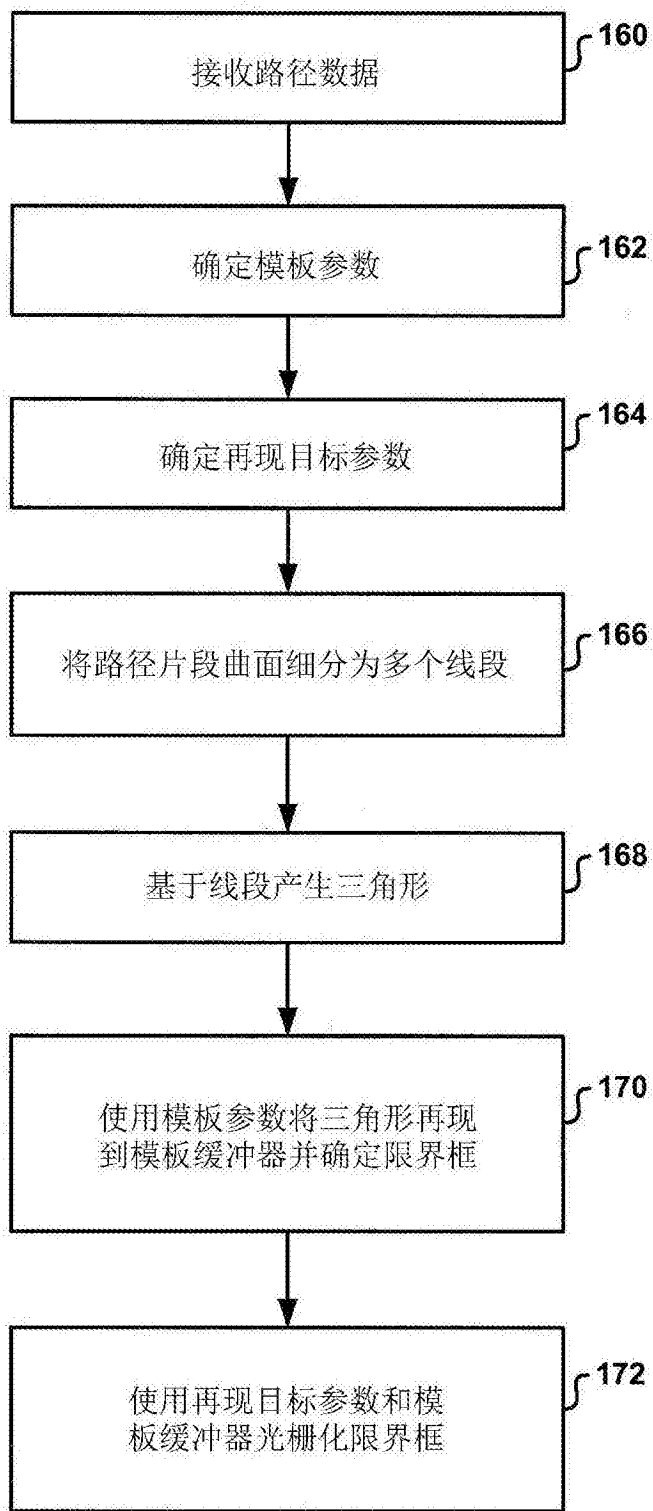


图10

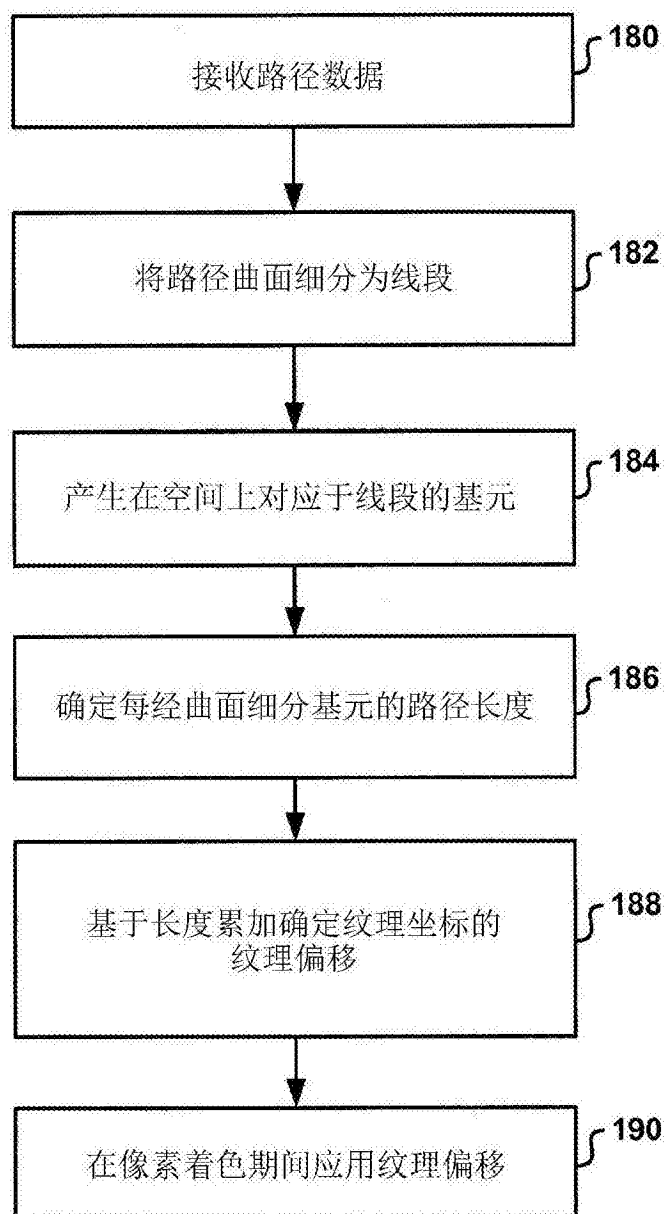


图11