

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2022年5月19日 (19.05.2022)

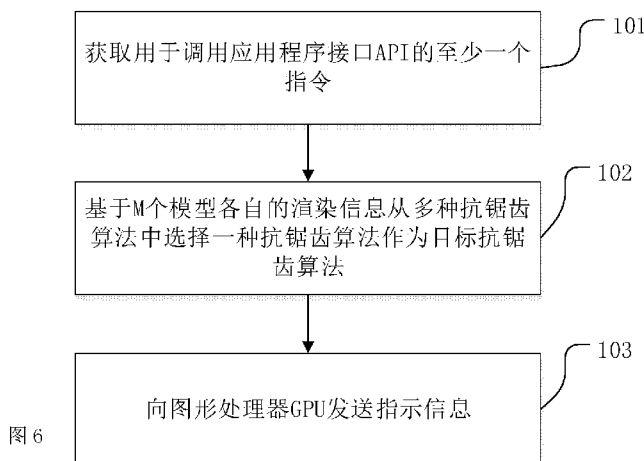


(10) 国际公布号
WO 2022/100663 A1

- (51) 国际专利分类号:
G06T 15/20 (2011.01)
- (21) 国际申请号: PCT/CN2021/130070
- (22) 国际申请日: 2021年11月11日 (11.11.2021)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202011273391.6 2020年11月13日 (13.11.2020) CN
- (71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (72) 发明人: 姜泽成 (JIANG, Zecheng); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 颜锦(YAN, Jin); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 罗备(LUO, Bei); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (74) 代理人: 深圳市深佳知识产权代理事务所(普通合伙) (SHENPAT INTELLECTUAL PROPERTY AGENCY); 中国广东省深圳市罗湖区南湖街道春风路庐山大厦B座18C2、18D、18E、18E2, Guangdong 518001 (CN)。
- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,

(54) Title: METHOD, APPARATUS AND DEVICE FOR SELECTING ANTI-ALIASING ALGORITHM AND READABLE STORAGE MEDIUM

(54) 发明名称: 一种选择抗锯齿算法的方法、装置、设备及可读存储介质



- 101 ACQUIRE AT LEAST ONE INSTRUCTION FOR CALLING AN APPLICATION PROGRAM INTERFACE (API)
- 102 ON THE BASIS OF RESPECTIVE RENDERING INFORMATION OF M MODELS, SELECT AN ANTI-ALIASING ALGORITHM AS A TARGET ANTI-ALIASING ALGORITHM FROM AMONG A PLURALITY OF ANTI-ALIASING ALGORITHMS
- 103 SEND INSTRUCTION INFORMATION TO A GRAPHICS PROCESSING UNIT (GPU)

(57) Abstract: A method, apparatus and device for selecting an anti-aliasing algorithm and a readable storage medium, which are used for dynamically selecting an anti-aliasing algorithm suitable for a current image scene, so as to improve the anti-aliasing effect and improve image quality. The method comprises: a CPU acquires at least one instruction for calling an application program interface (API), the at least one instruction bearing respective rendering information of M models, the M models belonging to the same image frame, and M being a positive integer; then, on the basis of the respective rendering information of the M models, an anti-aliasing



WO 2022/100663 A1

GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

algorithm is selected as a target anti-aliasing algorithm from among a plurality of anti-aliasing algorithms; and the CPU sends instruction information to a graphics processing unit (GPU), so that the GPU renders at least one image frame on the basis of the target anti-aliasing algorithm.

(57) 摘要: 一种选择抗锯齿算法的方法、装置、设备及可读存储介质, 用于动态选择适合当前图像场景的抗锯齿算法, 以提高抗锯齿效果, 提高画质。该方法包括: CPU获取用于调用应用程序接口API的至少一个指令, 该至少一个指令中携带M个模型各自的渲染信息, 且M个模型属于同一帧图像, M为正整数; 然后基于M个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法; CPU向图形处理器GPU发送指示信息, 使得GPU基于目标抗锯齿算法对至少一帧图像进行渲染。

一种选择抗锯齿算法的方法、装置、设备及可读存储介质

本申请要求于 2020 年 11 月 13 日提交的、中国申请号为 202011273391.6、发明名称为“一种选择抗锯齿算法的方法、装置、设备及可读存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请实施例涉及计算机技术领域，尤其涉及一种选择抗锯齿算法的方法、装置、设备及可读存储介质。

10

背景技术

在信号处理以及相关领域中，走样（Aliasing）是指在对不同的信号进行采样时，采样结果显示信号相同的现象。

15

在图像领域，走样可以包括几何体走样、着色走样以及时间走样。以几何体走样为例，几何体走样通常表现为锯齿现象。锯齿现象是指显示器中显示的图像边缘会出现凹凸锯齿。例如，一条直线如图 1 所示，将图 1 所示的直线放大，放大后的直线图像如图 2 所示，可以看出，这条直线两侧均出现了凹凸锯齿。

20

由于锯齿现象会影响图像的显示效果，所以诞生了抗锯齿（Anti-Aliasing, AA）技术。抗锯齿也可以称为反走样边缘柔化、消除混叠、抗图像折叠有损等，是一种能够消除图像边缘出现的凹凸锯齿的技术。例如，在渲染直线的过程中采用抗锯齿技术进行处理，则直线的显示结果如图 3 所示。对比图 2 和图 3 可知，抗锯齿技术将锯齿柔化，使得直线显示地更加平滑。

25

抗锯齿技术有很多算法。但在现有的游戏中，通常会选择一种固定的抗锯齿算法对游戏中的所有图像进行抗锯齿处理。由于每种抗锯齿算法都有不同的适用场景，所以以固定的一种抗锯齿算法进行处理，会导致部分图像的抗锯齿效果较差。

发明内容

本申请实施例提供了一种选择抗锯齿算法的方法、装置、设备及可读存储介质，该方法能够动态选择适合当前图像场景的抗锯齿算法，从而提高抗锯齿效果，提高画质。

30

本申请实施例第一方面提供了一种选择抗锯齿算法的方法，可以应用于中央处理器 CPU，包括：获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，该渲染信息包括但不限于顶点数据、模型的标识、模型对应的 VBO 的 ID 以及模型对应的 EBO 的 ID，M 个模型属于同一帧图像，M 为正整数；基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法，多种抗锯齿算法可以包括 SSAA 算法、MSAA 算法、CSAA 算法、CFAA 算法、FXAA 算法、TXAA 算法和 DLAA 算法中的至少两种；向图形处理器 GPU 发送指示信息，指示信息指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染。

35

应理解的是，这里选择出来的抗锯齿算法应用的所述“至少一帧图像”可以包括所述 M 个模型所属的那一帧，也可能来不及应用于这一帧，而是应用于这一帧后续的一帧或多帧。

由于 M 个模型属于同一帧图像，所以 M 个模型各自的渲染信息能够反映出该帧图像的具体场景，那么基于 M 个模型各自的渲染信息确定目标抗锯齿算法则能够实现对目标抗锯齿算法的动态选择，且能得到适用于当前图像场景的目标抗锯齿算法，以提高抗锯齿效果，提高画质。

作为一种实现方式，多种抗锯齿算法包括快速近似抗锯齿 FXAA 算法，相应地，基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法包括：从 M 个模型中确定渲染信息包含纹理信息的 N 个模型，纹理信息指示在渲染模型的过程中使用纹理数据，N 为小于或等于 M 的正整数；计算 N 个模型各自的顶点重复度，顶点重复度可以采用顶点重复使用的次数表示，也可以用顶点的总数与顶点重复使用的次数表示，具体地，顶点重复度可以等于模型中顶点重复使用的次数与和的比值，该和为顶点的总数与顶点重复使用的次数的和；基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择 FXAA 算法作为目标抗锯齿算法，条件可以的内容可以有多种，例如可以包括模型的数量阈值，K 为小于或等于 N 的正整数。

基于 FXAA 算法适用于通过纹理贴图绘制且顶点重复度较小的模型的渲染过程，所以在该实现方式中，从 M 个模型中先筛选得到通过纹理贴图绘制的 N 个模型，再从 N 个模型中筛选得到顶点重复度小于目标重复度的 K 个模型，最终当 K 个模型满足目标条件时选择 FXAA 算法，从而保证 FXAA 算法适用于 M 个模型所属的图像的场景。

作为一种实现方式，条件包括 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例，目标比例的取值可以根据经验、游戏应用程序的种类以及具体的游戏场景进行设定，例如，目标比例可以设置为 1/3、2/5 或 1/2。

该实现方式提供了条件的另一种示例，由于 FXAA 算法适用于通过纹理贴图绘制且顶点重复度较小的模型的渲染过程，所以当通过纹理贴图绘制且顶点重复度较小的 K 个模型占屏幕面积的比例较大时，选择 FXAA 算法作为目标抗锯齿算法，能够保证当前帧图像具有较好的抗锯齿效果。

作为一种实现方式，在计算 N 个模型各自的顶点重复度之后，在基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择 FXAA 算法作为目标抗锯齿算法之前，方法还包括：计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例，计算比例的方法有多种。

计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例，当该比例大于目标比例时，选择 FXAA 算法作为目标抗锯齿算法，能够保证当前帧图像具有较好的抗锯齿效果。

作为一种实现方式，计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例包括：获取包围 K 个模型的 P 个包围盒，每个包围盒为包围 K 个模型中至少一个模型的几何体，一个包围盒可以对应一个模型，也可以对应多个模型；包围盒的形状可以有多种，例如为长方体、正方体、球体等；基于转换矩阵计算 P 个包围盒在屏幕空间中的投影面积，转换

矩阵用于将 P 个包围盒从模型空间投影到屏幕空间，不同包围盒可以对应不同的转换矩阵；计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

5 由于模型本身通常为不规则几何体，因此如果直接计算模型在屏幕空间中的投影面积与屏幕面积的比例，则计算过程会较为繁琐；所以在该实现方式中，先获取包围 K 个模型的 P 个包围盒，然后计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例的近似值，能够简化比例的计算过程，提高比例的计算效果，从而
10 可以提高确定出目标抗锯齿算法的效率。

 作为一种实现方式，M 个模型各自的渲染信息还包括 M 个模型各自对应的顶点缓冲对象 VBO，相应地，获取包围 K 个模型的 P 个包围盒包括：从 K 个模型各自对应的顶点缓冲对象 VBO 中获取 K 个模型各自对应的顶点数据，其中，顶点数据可以包括顶点坐标、顶点法向量；基于 K 个模型各自对应的顶点数据确定 P 个包围盒，具体地，可以基于 K 个模型
15 各自的顶点坐标，便可以确定每个模型在各个方向对应的最大坐标值和最小坐标值，然后再确定 P 个包围盒。

 在该实现方式中，基于 K 个模型各自对应的顶点数据确定 P 个包围盒，使得确定出的包围盒的体积与模型的实际体积相差不大，这样，P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例便可以近似等于 K 个模型在屏幕空间中的投影面积与屏幕面积的比例，从而
20 保证条件的准确性，进而使得选择出的 FXAA 算法适用于 M 个模型所属的图像的场景。

 作为一种实现方式，N 个模型包含目标模型，目标模型的渲染信息还包括目标模型对应的索引缓冲对象 EBO，相应地，计算 N 个模型各自的顶点重复度包括：从目标模型对应的 EBO 中获取目标模型对应的顶点索引的数量和顶点索引键值的数量，其中，顶点索引可以理解为顶点的存储位置的索引，顶点索引键值是指索引的取值；基于顶点索引的数量和
25 顶点索引键值的数量计算目标模型的顶点重复度，目标模型的顶点重复度为目标差值与顶点索引的数量的比值，目标差值等于于顶点索引的数量与顶点索引键值的数量的差值。

 利用顶点索引的数量和顶点索引键值的数量计算顶点重复度，相比于利用顶点数据计算顶点重复度，计算量较小，计算过程简化，能够提高选择目标抗锯齿算法的效率。

 作为一种实现方式，M 个模型为所属图像中的所有模型，多种抗锯齿算法还包括多重
30 采样抗锯齿 MSAA 算法；基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法还包括：基于 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足条件，选择 MSAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

 由于 FXAA 算法和 MSAA 算法属于适用性较高的两种抗锯齿算法，因此在该实现方式中，先根据 M 个模型的渲染信息判断 FXAA 算法是否适用于当前图像的场景，若适用，则选择
35 FXAA 算法作为目标抗锯齿算法；若不适用，则选择 MSAA 算法作为目标抗锯齿算法；根据不同的场景对 FXAA 算法和 MSAA 算法进行不断切换，可以提高游戏应用程序的抗锯齿效果。

 作为一种实现方式，在基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条

件，选择 FXAA 算法作为目标抗锯齿算法之前，方法还包括：获取多种特定的场景下的多帧图像，每种特定的场景可以根据游戏应用程序的种类设定且对应至少一帧图像，多种特定的场景可以包含游戏应用程序中的所有的主要场景；计算多帧图像中特定的至少一个模型的顶点重复度，该特定的至少一个模型可以是适用于 FXAA 算法的模型，例如，可以通过纹理贴图绘制的模型，也可以是顶点重复度小的模型；基于特定的至少一个模型的顶点重复度确定目标重复度。

选取特定场景下的多帧图像，使得确定出的目标重复度适用于游戏应用程序中所有场景下的目标抗锯齿算法的选择；并且，根据多帧图像中特定的至少一个模型的顶点重复度确定目标重复度，例如当该特定的至少一个模型适用于 FXAA 算法的模型时，使得最终确定出的目标重复度可以用于 FXAA 算法的选择。

本申请实施例第二方面提供了一种选择抗锯齿算法的装置，包括：

获取单元，用于获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，M 个模型属于同一帧图像，M 为正整数；

选择单元，用于基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；

发送单元，用于向图形处理器 GPU 发送指示信息，指示信息指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染。

作为一种实现方式，多种抗锯齿算法包括快速近似抗锯齿 FXAA 算法；选择单元，用于从 M 个模型中确定渲染信息包含纹理信息的 N 个模型，纹理信息指示在渲染模型的过程中使用纹理数据，N 为小于或等于 M 的正整数；计算 N 个模型各自的顶点重复度；基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择 FXAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

作为一种实现方式，条件包括 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例。

作为一种实现方式，选择单元还用于计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

作为一种实现方式，选择单元用于获取包围 K 个模型的 P 个包围盒，每个包围盒为包围 K 个模型中至少一个模型的几何体；基于转换矩阵计算 P 个包围盒在屏幕空间中的投影面积，转换矩阵用于将 P 个包围盒从模型空间投影到屏幕空间；计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

作为一种实现方式，M 个模型各自的渲染信息还包括 M 个模型各自对应的顶点缓冲对象 VBO；选择单元，用于从 K 个模型各自对应的顶点缓冲对象 VBO 中获取 K 个模型各自对应的顶点数据；基于 K 个模型各自对应的顶点数据确定 P 个包围盒。

作为一种实现方式，N 个模型包含目标模型，目标模型的渲染信息还包括目标模型对应的索引缓冲对象 EBO；选择单元，用于从目标模型对应的 EBO 中获取目标模型对应的顶点索引的数量和顶点索引键值的数量；基于顶点索引的数量和顶点索引键值的数量计算目

标模型的顶点重复度，目标模型的顶点重复度为目标差值与顶点索引的数量的比值，目标差值等于于顶点索引的数量与顶点索引键值的数量的差值。

作为一种实现方式，M 个模型为所属图像中的所有模型，多种抗锯齿算法还包括多重采样抗锯齿 MSAA 算法；选择单元还用于基于 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足条件，选择 MSAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

作为一种实现方式，该装置还包括：确定单元，用于获取多种场景下的多帧图像，每种场景对应至少一帧图像；计算多帧图像中特定的至少一个模型的顶点重复度；基于特定的至少一个模型的顶点重复度确定目标重复度。

其中，以上各单元的具体实现、相关说明以及技术效果请参考本申请实施例第一方面的描述。

本申请实施例第三方面提供了一种终端设备，包括：一个或多个处理器和存储器；其中，所述存储器中存储有计算机可读指令；所述一个或多个处理器读取所述计算机可读指令，以使所述计算机设备实现如第一方面任一实现方式所述的方法。

本申请实施例第四方面提供了一种计算机可读存储介质，包括计算机可读指令，当所述计算机可读指令在计算机上运行时，使得所述计算机执行如第一方面任一实现方式所述的方法。

本申请实施例第五方面提供了一种芯片，包括一个或多个处理器。所述处理器中的部分或全部用于读取并执行存储器中存储的计算机程序，以执行上述第一方面任意可能的实现方式中的方法。

可选地，该芯片该包括存储器，该存储器与该处理器通过电路或电线与存储器连接。进一步可选地，该芯片还包括通信接口，处理器与该通信接口连接。通信接口用于接收需要处理的数据和/或信息，处理器从该通信接口获取该数据和/或信息，并对该数据和/或信息进行处理，并通过该通信接口输出处理结果。该通信接口可以是输入输出接口。

在一些实现方式中，所述一个或多个处理器中还可以有部分处理器是通过专用硬件的方式来实现以上方法中的部分步骤，例如涉及神经网络模型的处理可以由专用神经网络处理器或图形处理器来实现。

本申请实施例提供的方法可以由一个芯片实现，也可以由多个芯片协同实现。

本申请实施例第六方面提供了一种计算机程序产品，该计算机程序产品包括计算机软件指令，该计算机软件指令可通过处理器进行加载来实现上述第一方面中任意一种实现方式所述的方法。

从以上技术方案可以看出，本申请实施例具有以下优点：

获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，M 个模型属于同一帧图像，M 为正整数；基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；向图形处理器 GPU 发送指示信息，指示信息指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染；

由于 M 个模型属于同一帧图像，所以 M 个模型各自的渲染信息能够反映出该帧图像的

具体场景，那么基于 M 个模型各自的渲染信息确定目标抗锯齿算法则能够实现对目标抗锯齿算法的动态选择，且能得到适用于当前图像场景的目标抗锯齿算法，以提高抗锯齿效果，提高画质。

5 附图说明

- 图 1 为直线未经抗锯齿处理的图像示意图；
图 2 为将图 1 中的直线放大后的图像示意图；
图 3 为直线经抗锯齿处理后的图像示意图；
图 4 为本申请实施例中的系统架构示意图；
10 图 5 为包含抗锯齿开关选项的显示设置界面的示意图；
图 6 为本申请实施例提供的选择抗锯齿算法的方法的一个实施例示意图；
图 7 为本申请实施例提供的选择抗锯齿算法的方法的另一个实施例示意图；
图 8 为本申请实施例中一个模型的示意图；
图 9 为本申请实施例中草的模型的示意图；
15 图 10 为本申请实施例中人的模型的示意图；
图 11 为本申请实施例中鞋子的模型的示意图；
图 12 为本申请实施例中计算 K 个模型的投影面积与屏幕面积的比例示意图；
图 13 为本申请实施例中包围盒的一个实施例示意图；
图 14 为本申请实施例中包围盒的另一个实施例示意图；
20 图 15 为本申请实施例中应用例的示意图；
图 16 为本申请实施例第一场景下的第一种抗锯齿效果示意图；
图 17 为本申请实施例第一场景下的第二种抗锯齿效果示意图；
图 18 为本申请实施例第二场景下的第一种抗锯齿效果示意图；
图 19 为本申请实施例第二场景下的第二种抗锯齿效果示意图；
25 图 20 为本申请实施例中选择抗锯齿算法的装置的示意图；
图 21 为本申请实施例中终端设备的结构示意图。

具体实施方式

下面结合附图，对本申请的实施例进行描述，显然，所描述的实施例仅仅是本申请一
30 部分的实施例，而不是全部的实施例。本领域普通技术人员可知，随着技术的发展和场景的出现，本申请实施例提供的技术方案对于类似的技术问题，同样适用。

本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别
类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情
况下可以互换，以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实
35 施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，
例如，包含了一系列步骤或模块的过程、方法、系统、产品或设备不必限于清楚地列出的

那些步骤或模块，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或模块。在本申请中出现的对步骤进行的命名或者编号，并不意味着必须按照命名或者编号所指示的时间或逻辑先后顺序执行方法流程中的步骤，已经命名或者编号的流程步骤可以根据要实现的技术目的变更执行次序，只要能达到相同或者相类似的技术效果即可。

本申请实施例可以应用于如图 4 所示的系统架构中。该系统架构包括应用程序层、图形应用程序接口 (Application Program Interface, API) 层、内核层和硬件层。其中，该系统框架可以应用于任意能够渲染图像的终端设备中，例如，该系统框架可以应用的终端设备包括但不限于手机、平板电脑、笔记本电脑和车载电脑。

如图 4 所示，应用程序层包括三个游戏应用程序。需要说明的是，在该实施例中，应用程序层中的应用程序主要是指游戏应用程序，即本申请实施例主要应用于游戏中的图像渲染场景。但在本申请其它实施例中，应用程序层中的应用程序也可以是其他适用的应用程序，本申请对此不做具体限定；另外，游戏应用程序的数量也不限于三个，更少或更多都可以。

图形 API 层包括但不限于开放图形库 (Open Graphics Library, OpenGL)、OpenGL ES (OpenGL for Embedded Systems) 和 Vulkan 这三种图形 API。其中，图形 API 层可以用于将 2D 或 3D 场景渲染至终端设备的屏幕上；例如，OpenGL 用于渲染 2D 场景，OpenGL ES 和 Vulkan 用于渲染 3D 场景。

内核层包括系统内核以及驱动等。

硬件层包括但不限于中央处理器 CPU、图形处理器 GPU 和内部存储器等，其中，内部存储器可以是双倍速率 SDRAM (Double Data Rate Synchronous Dynamic Random-access Memory, DDR SDRAM)，也可以是其他类型的内部存储器。

基于图 4 所示的系统框架，当游戏应用程序运行时，以一幅图像为例，图像的渲染流程 Pipe Line 包括：

第一步，游戏应用程序中的游戏引擎对游戏场景进行遍历，裁剪、剔除游戏场景中的不可见物体。

第二步，对应游戏场景中的每个可见物体，游戏引擎准备数据资源，并发送用于调用图形 API 的指令，该用于调用图形 API 的指令会经过图形 API 层到达内核层。

第三步，内核层的驱动在接收到用于调用图形 API 的指令后，会将数据资源传递到 GPU 的内部存储器中，并向 GPU 发送渲染指令，以指示 GPU 进行渲染。

第四步，GPU 在接收到渲染指令后，会从内部存储器中读取数据资源，并执行顶点着色器 Vertex shader、片段着色器 Fragment shader 或者 Compute shader 等操作，以得到渲染后的一幅图像。

其中，渲染后的图像中的每个物体都是一个网格 Mesh 模型，每个 Mesh 模型由大量的三角形组成，每个三角形都具有三个顶点。不同的三角形的顶点可以重叠。

顶点着色器是指在 GPU 上处理顶点变换的程序片段。对应 Mesh 模型的每个顶点，均会执行一次顶点着色器。

片段着色器是指在 GPU 上处理像素着色的程序片段。Mesh 模型经过顶点变换后，对屏幕上每个被 Mesh 模型覆盖的像素点执行一次像素着色器。

Compute shader 是指在 GPU 上执行的程序片段，可用于图像渲染之外的计算。具体地，可通过 OpenGL ES 或者 Vulkan API 将 Compute shader 传输至 GPU，以被 GPU 执行。

5 然而，由于渲染得到的图像存在锯齿现象，而锯齿现象会影响图像的显示效果，所以会在图像渲染的过程中，采用抗锯齿技术进行抗锯齿处理，以降低锯齿现象对图像的显示效果的影响，提高画质。例如，如图 5 所示，游戏应用程序会在显示设置界面中提供一个抗锯齿开关选项，用户可以在游戏游行前或游戏运行过程中选择是否开启抗锯齿；在用户选择开始抗锯齿后，终端设备将在后续图像渲染的过程中，采用抗锯齿技术进行抗锯齿处
10 理。

抗锯齿技术有多种，具体包括超级采样抗锯齿（Super-Sampling Anti-aliasing, SSAA）、多重采样抗锯齿（Multisampling Anti-Aliasing, MSAA）、覆盖采样抗锯齿（Coverage Sampling Anti-Aliasing, CSAA）、可编程过滤抗锯齿（Custom Filter Anti-Aliasing, CFAA）、快速近似抗锯齿（Fast Approximate Anti-Aliasing, FXAA）、时间混叠抗锯齿
15 （Temporal Anti-Aliasing, TXAA）和深度学习抗锯齿（Deep Learning Anti-Aliasing, DLAA）。

下面具体介绍上述抗锯齿技术中适用性较好的两种抗锯齿技术。

MSAA：MSAA 来自于 OpenGL，只对 Z 缓存（Z-Buffer）和模板缓存（Stencil Buffer）中的数据进行超级采样抗锯齿的处理；这可以简单理解为，MSAA 只对多边形的边缘进行抗
20 锯齿处理。这样的话，相比 SSAA 对图像中所有数据进行处理，MSAA 对资源的消耗需求大幅减少，并能够实现较好的抗锯齿效果。因此，MSAA 的应用范围广泛，几乎所有游戏都支持 MSAA。

FXAA：FXAA 是一种后处理 Post-Processing 抗锯齿技术，其中，后处理是指在普通场景渲染结束后对结果进行处理，即对一张或多张纹理图像进行处理，得到一张新的纹理图
25 像。所以，FXAA 是对初步渲染的图像进行处理，不依赖于任何图形处理器 GPU 计算接口，对显卡没有特殊要求。FXAA 通过分析图像中的像素，提取并平滑图像的锯齿边缘，只需消耗很小的计算资源，就能获得不错的抗锯齿效果。但是 FXAA 也会对图像产生一定的模糊效果，同时还可能加剧边缘闪烁的问题。

基于上述说明可知，MSAA 适用于对 Mesh 模型中多边形的边缘进行抗锯齿处理；由于
30 FXAA 是对初步渲染的图像进行处理，所以 FXAA 适用于对纹理贴图绘制的 Mesh 模型进行抗锯齿处理。

其他抗锯齿技术同上述两种抗锯齿技术一样，都具有各自的适用场景。这里的场景可以理解游戏中的场景，随着游戏中场景的不断变化，场景中物体种类、物体数量以及物体的渲染方法等都会发生变化。所以，在游戏运行过程中，若仅采用一种抗锯齿技术进行
35 抗锯齿处理，当游戏场景变为抗锯齿技术不适用的场景，抗锯齿效果则会变差。

为此，本申请实施例在图 4 所示的系统框架中增加了算法选择层，该算法选择层能从图形 API 层实时获取到调用图形 API 的指令，然后从获取到的调用图形 API 的指令提取 Mesh

模型的渲染信息，然后基于 Mesh 模型的渲染信息动态地从多种抗锯齿算法中选择抗锯齿算法，使得选择出的抗锯齿算法能够适应当前的场景，从而可以提高抗锯齿效果，提高画质。其中，一种抗锯齿算法可以对应一种抗锯齿技术。

需要说明的是，本申请实施例可以应用于任意能够应用抗锯齿技术的终端设备，例如，
5 可以应用于手机、平板电脑、台式电脑、笔记本电脑和车载电脑等终端设备。

然而，由于抗锯齿算法通常由 GPU 执行，不同终端设备的 GPU 架构不同，所以当本申请实施例应用于不同的终端设备时，可能需要进行适应性调整，适应性调整后的方案也属于本申请实施例的保护范围。

10 应理解，手机和平板电脑等移动终端的 GPU 架构与台式电脑等固定终端的 GPU 架构不同；并且，由于移动终端需要考虑硬件资源以及功耗等因素，所以通常只能选择一种抗锯齿技术进行抗锯齿处理，而固定终端由于不需要考虑硬件资源以及功耗等因素，所以可以同时选择多种抗锯齿技术进行抗锯齿处理。因此，本申请实施例尤其适用于手机和平板电脑等移动终端，具体可用于移动终端的游戏应用程序中。

下面对本申请实施例提供的选择抗锯齿算法的方法进行介绍。为了便于描述，下文用
15 模型代替 Mesh 模型。

具体地，请参阅图 6，本申请实施例提供了一种选择抗锯齿算法的方法的一个实施例，该实施例可以应用于中央处理器 CPU；在游戏运行过程中，该实施例包括：

步骤 101，获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，M 个模型属于同一帧图像，M 为正整数。

20 基于图 1 的相关说明可知，应用程序会向图形应用程序接口 API 发送指令，该指令用于调用图形 API；而在本申请实施例中，可以从图形 API 获取到该指令。

在本申请实施例中，获取到调用图形 API 的至少一个指令，该至少一个指令可以包括一个指令，也可以包括多个指令。

下面对至少一个指令与 M 个模型的对应关系进行说明。

25 一个指令可以对应多个模型。例如，一个指令可以指示对同一种物体的多个模型进行渲染；具体地，当前游戏场景中包含多棵相同的树，那么便可以通过一个指令指示对这多棵树的模型进行渲染。

一个模型可以对应多个指令。通常情况下，一个模型的渲染过程需要发送多个指令，该多个指令从类型上可以被划分准备指令和 Drawcall 指令。

30 其中，准备指令可以用于将模型渲染所需的资源数据存入相应的 buffer 缓冲器；具体地，准备指令可以用于将模型与纹理数据绑定，可以用于将模型的顶点数据等存入顶点缓冲对象 (Vertex Buffer Object, VBO)，还可以用于将顶点索引存入索引缓冲对象 (Element Buffer Object, EBO)。

35 VBO 是在显卡内存 (简称显存) 中开辟出的一块内存缓存器，用于存储顶点的各类属性数据，顶点数据包括顶点坐标、顶点法向量、顶点颜色数据等。在渲染时，GPU 可以直接从 VBO 中读取顶点的各类属性数据，由于 VBO 在显存而不是在内存中，不需要从 CPU 传输数据，处理效率更高。在显存中可以开辟很多个 VBO，每个 VBO 对应唯一标识 ID，这个

ID 对应着具体的显存地址，通过这个 ID 可以对特定的 VBO 内的数据进行存取操作。

EBO 与 VBO 类似，也是在显存中的一块内存缓冲器，只是 EBO 保存的是顶点索引，顶点索引可以理解为顶点的存储位置的索引。EBO 相当于 OpenGL 中的顶点数组的概念，可以解决同一个顶点多次重复调用的问题，从而减少内存空间浪费，提高执行效率。当需要使用重复的顶点时，可以通过顶点索引来调用顶点，而不是对重复的顶点数据重复记录和重复调用。

Drawcall 指令可以用于指令 GPU 对模型进行渲染，其中，Drawcall 指令中可以包含模型的标识、模型对应的 VBO 的 ID 以及模型对应的 EBO 的 ID 等。

因此，本申请实施例对至少一个指令的数量、种类以及至少一个指令与 M 个模型的对应关系均不做具体限定。

模型的渲染信息可以包括多种信息，本申请实施例对此不做具体限定。基于前述说明可知，渲染信息包括但不限于顶点数据、模型的标识、模型对应的 VBO 的 ID 以及模型对应的 EBO 的 ID。

M 个模型属于同一帧图像，这一帧图像属于待渲染的图像，所以 M 个模型各自的渲染信息能够反映出这一帧图像的具体场景。其中，M 个模型可以是一个模型，也可以包括多个模型；M 个模型可以是该帧图像中的部分模型，也可以是该帧图像中的全部模型。

需要说明的是，渲染一帧图像的过程包括渲染该帧图像中所有模型的过程，所以在渲染一帧图像的过程中，通常会持续发送几十甚至几百个调用图形 API 的指令。因此，步骤 101 可以是一次获取，得到至少一个指令；当至少一个指令的数量为多个时，步骤 101 还可以是通过多次获取，得到多个指令。

步骤 102，基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法。

基于步骤 101 的相关说明可知，M 个模型各自的渲染信息能够反映出 M 个模型所属的待渲染的图像的具体场景，所以基于 M 个模型各自的渲染信息选择目标抗锯齿算法，可以理解为基于待渲染的图像的具体场景选择目标抗锯齿算法。

本申请实施例对多种抗锯齿算法的种类不做具体限定，包括但不限于 SSAA 算法、MSAA 算法、CSAA 算法、CFAA 算法、FXAA 算法、TXAA 算法和 DLAA 算法中至少两种。

由于 M 个模型各自的渲染信息可以包括多种信息，所以对应不同的渲染信息，选择目标抗锯齿算法的方法也不同。

因此，在步骤 102 中，选择目标抗锯齿算法的方法有多种，本申请实施例对此不做具体限定。

步骤 103，向图形处理器 GPU 发送指示信息，指示信息指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染。

在确定目标抗锯齿算法后，CPU 向 GPU 发送指示信息，使得 GPU 基于目标抗锯齿算法进行抗锯齿处理。

需要说明的是，假设 M 个模型所属的图像为当前帧图像，本申请实施例在确定目标抗锯齿算法的同时，GPU 也在渲染该当前帧图像中的模型；所以在确定目标抗锯齿算法后，

已经完成了当前帧图像中部分模型的渲染，或已经完成了当前帧图像的渲染。

在游戏渲染过程中，一秒内需要多帧图像，相邻帧图像的场景相差不大。

因此，当在确定目标抗锯齿算法后，已经完成了当前帧图像中部分模型的渲染时，可以利用目标抗锯齿算法对当前帧图像进行处理，也可以利用目标抗锯齿算法对当前帧图像后的一帧或多帧图像进行渲染；当在确定目标抗锯齿算法后，已经完成了当前帧图像的渲染时，也可以利用目标抗锯齿算法对当前帧图像后的一帧或多帧图像进行渲染。

综上，指示信息可以指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染。

在本申请实施例中，由于 M 个模型各自的渲染信息能够反映出 M 个模型所属的待渲染的图像的具体场景，所以基于至少一个指令中携带 M 个模型各自的渲染信息确定目标抗锯齿算法，能够实现对目标抗锯齿算法的动态选择，且得到适用于当前图像场景的目标抗锯齿算法，以提高抗锯齿效果，提高画质。

基于上述实施例可知，选择目标抗锯齿算法的方法有多种，下面以一具体实施例对此进行说明。

请参阅图 7，本申请实施例提供了一种选择抗锯齿算法的方法的另一个实施例，该实施例也可以应用于中央处理器 CPU，该实施例包括：

步骤 201，获取多种场景下的多帧图像，每种场景对应至少一帧图像。

需要说明的是，多种场景可以是预先设定的，本申请实施例对多种场景的具体内容不做具体限定；多种场景的具体内容可以与游戏应用程序相关，不同的游戏应用程序，对应的多种场景不同。

由于多种场景下的多帧图像是用于确定目标重复度，所以为了保证确定出的目标重复度适用于游戏应用程序中所有场景下的目标抗锯齿算法的选择，多种场景是游戏应用程序中具有代表性的场景，可以包含游戏应用程序中的所有的主要场景。

通常情况下，对于一款特定的游戏应用程序，主要场景的数量是有限的，因此可以将游戏应用程序的几种主要场景设置为步骤 201 中的多种场景。

下面以具体的示例对多种场景进行说明。例如，对于一款野外枪击类的游戏应用程序，多种场景可以包括山上的场景、水中的场景、桥上的场景以及房间内的场景等；对于一款塔防类的游戏应用程序，多种场景可以包括基地的场景、对线小兵的场景、防御塔的场景以及对线玩家的场景等。

每种场景对应至少一帧图像，示例性地，每种场景对应关键帧图像，其中关键帧图像可以理解为角色或者物体在运动或变化中的关键动作所处的那一帧图像。

步骤 202，计算多帧图像中特定的至少一个模型的顶点重复度。

需要说明的是，计算特定的至少一个模型的顶点重复度是为了确定目标重复度，而目标重复度是用于选择目标抗锯齿算法，所以可以根据所需选择的目标抗锯齿算法，并结合人工经验预先选择特定的至少一个模型。

例如，为了确定选择 FXAA 算法的条件，所以根据 FXAA 算法适用的模型选择特定的至少一个模型。具体地，FXAA 算法适用于贴图绘制的模型，所以可以选择渲染过程中通过纹理贴图绘制的模型，作为特定的至少一个模型，例如可以选择草的模型、栅栏的模型以及

树叶的模型中的一种或多种作为特定的至少一个模型。

模型的顶点重复度越高，模型的三角形数量就越多，边也就越多；基于此，由于 MSAA 算法适用于对多边形的边缘进行抗锯齿处理，且 FXAA 算法会对图像产生模糊的效果，因此对比来看，MSAA 算法比 FXAA 算法更适用于顶点重复度高的模型的渲染过程，而 FXAA 算法更适用于顶点重复度低的模型的渲染过程。为此，可以选择通常情况下顶点重复度低的模型作为特定的至少一个模型；例如，草的模型、栅栏的模型以及树叶的模型的顶点重复度通常较低，所以可以选择草的模型、栅栏的模型以及树叶的模型中的一种或多种作为特定的至少一个模型。

其中，顶点重复度指示一个模型中顶点重复使用的情况。假设一模型如图 8 所示，该模型由两个三角形构成，这两个三角形的一条边重合，所以该模型包含的顶点的数量为 4，有两个顶点分别重复使用一次。

顶点重复度的形式可以有多种，本申请实施例对此不做具体限定。作为一种实现方式，顶点重复度可以为模型中顶点重复使用的次数，但这对于顶点数量较多的模型来说，顶点重复度必然较高。因此，作为另一种实现方式，顶点重复度可以等于模型中顶点重复使用的次数与和的比值，该和为顶点的总数与顶点重复使用的次数的和。

下面以具体的示例说明草的模型、栅栏的模型以及树叶的模型的顶点重复度通常较低。

如图 9、图 10 和图 11 所示，分别示出了草的模型、人物的模型以及鞋子的模型，对比三张图可以明显地看出草的模型的顶点重复度较低。

步骤 203，基于特定的至少一个模型的顶点重复度确定目标重复度。

需要说明的是，基于特定的至少一个模型的顶点重复度确定目标重复度的方法有多种，本申请实施例对此不做限定。作为一种实现方式，将目标重复度设置为大于特定的至少一个模型中任一模型的顶点重复度。例如，特定的至少一个模型包括两个模型，且顶点重复度分别为 30% 和 31%，则目标重复度可以设置为 35%。

步骤 201 至步骤 203 提供了一种确定目标重复度的具体方案，即基于选择的特定的至少一个模型的顶点重复度确定目标重复度，该目标重复度用于步骤 207 中确定目标抗锯齿算法的过程。

相应地，步骤 201 至步骤 203 需要在步骤 207 之前执行；具体地，可以在游戏离线状态下执行步骤 201 至步骤 203，也可以在用户选择抗锯齿之前但在游戏运行过程中执行步骤 201 至步骤 203，本申请实施例对此不做具体限定。

需要说明的是，除了上述确定目标重复度的方法外，还可以使用直接设定目标重复度等其他方法确定目标重复度，因此在本申请实施例中，步骤 201 至步骤 203 是可选地。

下面的步骤 204 至步骤 207 是在游戏运行过程中执行。

步骤 204，获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，M 个模型属于同一帧图像，M 为正整数。

基于步骤 101 的相关说明可知，获取到的指令可以把包括将模型与纹理数据绑定的指令。

可以理解的是，在模型的渲染过程中，通常可以通过两种方法对该模型进行着色。

第一种着色方法：使用纹理数据对模型进行着色，也可以称为通过纹理贴图绘制；相应地，在渲染前，需要发送将纹理数据与模型绑定的指令。当该指令被获取到时，渲染信息则包含纹理信息，纹理信息指示在渲染模型的过程中使用纹理数据。

第二种着色方法：通过设置像素点的颜色来对模型进行着色，即将模型中的每个像素点都设置为一种颜色；相应地，在 CPU 发送给 GPU 的资源数据中会包含模型中各顶点以及各像素点的颜色数据。这种情况下，CPU 便不会发送将纹理数据与模型绑定的指令，因此也无法获取到将纹理数据与模型绑定的指令，相应地，模型的渲染信息中则不会包含纹理信息。

本申请实施例中的步骤 204 的其他内容与前一实施例中的步骤 101 类似，具体可参看前述实施例中步骤 101 的相关说明对本申请实施例中的步骤 204 进行理解。

下面通过步骤 204 至步骤 207 介绍选择目标抗锯齿算法的具体过程，在本申请实施例中，多种抗锯齿算法包括快速近似抗锯齿 FXAA 算法，相应地，前述实施例中的步骤 102 可以包括步骤 204 至步骤 207。

步骤 205，从 M 个模型中确定渲染信息包含纹理信息的 N 个模型，纹理信息指示在渲染模型的过程中使用纹理数据，N 为小于或等于 M 的正整数。

基于步骤 204 的相关说明可知，若模型需要通过纹理数据进行着色，该模型的渲染信息中则会包含纹理信息；若模型不需要通过纹理数据进行着色，该模型的渲染信息中则不会包含纹理信息。

因此，可以从 M 个模型中筛选出包含纹理信息的 N 个模型；其中，N 个模型可以是全部 M 个模型，此时，N 等于 M；N 个模型也可以是 M 个模型中的一部分，此时，N 小于 M。

步骤 206，计算 N 个模型各自的顶点重复度。

基于步骤 202 的相关说明可知，模型的顶点重复度可以采用顶点重复使用的次数表示，也可以基于顶点的总数与顶点重复使用的次数计算，即顶点重复度=顶点重复使用的次数/（顶点的总数+顶点重复使用的次数）。

可以理解的是，若通过模型的顶点数据来确定被重复使用的顶点以及每个被重复使用的顶点的重复使用的次数，则计算量将会较大。为了，本申请实施例提供了计算顶点重复度的另外一种方式，下面对此进行介绍。

示例性地，N 个模型包含目标模型，即目标模型为 N 个模型中的一个，目标模型的渲染信息还包括目标模型对应的索引缓冲对象 EBO。其中，EBO 可参照步骤 101 的相关说明进行理解。

基于上述条件，计算 N 个模型各自的顶点重复度包括：

从目标模型对应的 EBO 中获取目标模型对应的顶点索引的数量和顶点索引键值的数量；基于顶点索引的数量和顶点索引键值的数量计算目标模型的顶点重复度，目标模型的顶点重复度为目标差值与顶点索引的数量的比值，目标差值等于于顶点索引的数量与顶点索引键值的数量的差值。

下面结合图 9 至图 11 对模型的顶点重复度的计算过程进行说明。

对于图 9 所示的模型，EBO 中的顶点索引包括：[0, 1, 2, 3, 4, 5, 6, 4, 3, 7, 8,

9, 8, 7, 10, 11, 12, 13, 12, 11, 14, 15, 16, 17, 17, 18, 15, 19, 20, 21, 22, 20, 19, 23, 24, 25, 24, 23, 26, 27, 28, 29, 29, 30, 27, 31, 32, 33, 33, 34, 31], 其中每个索引值又可以称为顶点索引键值, 表示顶点的存储位置, 根据该顶点索引值可以从相应的 VBO 中获取顶点数据。

5 对于图 9 所示模型的顶点索引, 顶点索引的数量为 51, 顶点索引键值可以为 0 至 34, 所以顶点索引键值的数量为 35; 相应地, 顶点重复使用的次数为 $51-35=16$, 该模型的顶点重复度= $16/51=31.4\%$ 。

使用相同的方法对图 10 以及图 11 所示的模型的顶点重复度进行计算。

10 对于图 10 所示的模型, 顶点索引的数量为 4669, 顶点索引键值的数量为 877; 相应地, 该模型的顶点重复度= $(4669-877) / 4669=81.2\%$ 。

对于图 11 所示的模型, 顶点索引的数量为 1002, 顶点索引键值的数量为 254; 相应地, 该模型的顶点重复度= $(1002-254) / 4669=74.6\%$ 。

基于上述数据可以看出, 图 9 所示的草的模型的顶点重复度明显低于图 10 和图 11 所示的模型的顶点重复度。

15 在本申请实施例中, 利用顶点索引的数量和顶点索引键值的数量计算顶点重复度, 相比于利用顶点数据计算顶点重复度, 计算量较小, 计算过程简化, 能够提高选择目标抗锯齿算法的效率。

步骤 207, 基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件, 选择 FXAA 算法作为目标抗锯齿算法, K 为小于或等于 N 的正整数。

20 基于步骤 203 中对特定的至少一个模型的相关说明可知, FXAA 算法适用于通过纹理贴图绘制且顶点重复度较小的模型的渲染过程。

因此, 在本申请实施例中, 从 M 个模型中先筛选得到通过纹理贴图绘制的 N 个模型, 然后再从 N 个模型中筛选得到顶点重复度小于目标重复度的 K 个模型; 最终, 基于 K 个模型以及预设的条件来选择目标抗锯齿算法。

25 其中, 预设的条件可以有多种, 本申请实施例对此不做具体限定。

作为一种实现方式, 预设的条件可以为模型的数量阈值。

可以理解的是, 即使 K 个模型的数量很多, 但若 K 个模型在屏幕空间中的尺寸很小, 那么即使无论选择哪种抗锯齿算法对 K 个模型进行处理, 对 K 个模型的抗锯齿效果影响都不大。

30 作为另一种实现方式, 条件包括 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例。

可以理解的是, K 个模型在屏幕空间中的尺寸越大, 那么对 K 个模型所属的该帧图像的显示效果影响就越大; 而 K 个模型的渲染过程适用于 FXAA 算法, 所以当 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例时, 选择 FXAA 算法作为目标抗锯齿算法, 对整帧图像的抗锯齿效果提高较大。

35 其中, 目标比例可以根据经验、游戏应用程序的种类以及具体的游戏场景进行设定, 例如, 目标比例可以设置为 $1/3$ 、 $2/5$ 或 $1/2$ 。

相应地，在步骤 206 之后，在步骤 207 之前，方法还包括：计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例的方法有多种，本申请实施例对此不做具体限定，下面对次进行具体介绍。

5 作为一种实现方式，如图 12 所示，计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例包括：

步骤 301，获取包围 K 个模型的 P 个包围盒，每个包围盒为包围 K 个模型中至少一个模型的几何体。

10 对于一个模型来说，包围盒则为在模型空间中包围该模型的几何体。其中，对于同一模型，包围盒的种类可以有多种，只要能够包围该模型即可。因此，本申请实施例对 P 个包围盒的结构不做具体限定。

例如，对于一个兔子的模型来说，包围盒可以如图 13 所示，即包围盒为一个球体；包围盒也可以如图 14 所示，即包围盒为一个正方体。

下面对 K 个模型和 P 个包围盒的对应关系进行说明。

15 一个包围盒可以对应一个模型，即一个包围盒包围一个模型。如图 12 和图 13 所示，该包围盒仅包围一个兔子的模型。

一个包围盒也可以对应多个模型。例如，在模型空间中，两个模型距离比较近甚至有重叠，此时则可以确定包围这两个模型的一个包围盒。

20 需要说明的是，模型可以对应多种包围盒，所以获取 P 个包围盒的方法有多种。示例性地，M 个模型各自的渲染信息还包括 M 个模型各自对应的顶点缓冲对象 VBO，具体可参照前述实施例中的步骤 101 的相关说明对 VBO 进行理解。

基于此，获取包围 K 个模型的 P 个包围盒包括：

从 K 个模型各自对应的顶点缓冲对象 VBO 中获取 M 个模型各自对应的顶点数据，基于 K 个模型各自对应的顶点数据确定 P 个包围盒。

25 其中，顶点数据可以包括顶点坐标、顶点法向量。

可以理解的是，基于 K 个模型各自的顶点坐标，便可以确定每个模型在各个方向对应的最大坐标值和最小坐标值，这相当于确定了模型的边缘，基于每个模型在各个方向对应的最大坐标值和最小坐标值便可以获取到 P 个包围盒。

30 步骤 302，基于转换矩阵计算 P 个包围盒在屏幕空间中的投影面积，转换矩阵用于将 P 个包围盒从模型空间投影到屏幕空间。

模型空间也称对象空间或者局部空间，在模型空间中，每个模型都有自己独立的坐标空间，当模型移动或者旋转时，模型空间也跟着它移动和旋转。

屏幕空间是一个二维空间，可以将模型的顶点坐标投影到屏幕空间，以得到对应的二维坐标。

35 其中，转换矩阵则用于将模型空间中的模型投影（也可以称为转换）到屏幕空间。

P 个包围盒中任意两个包围盒可以对应相同的转换矩阵，也可以对应不同的转换矩阵，因此可以基于一个或多个转换矩阵将 P 个包围盒从模型空间投影到屏幕空间。

需要说明的是，计算 P 个包围盒在屏幕空间中的投影面积为较成熟的技术，本申请实施例在此不做详述。

步骤 303，计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

可以理解的是，模型本身通常为不规则几何体，因此如果直接计算模型在屏幕空间中的投影面积与屏幕面积的比例，则计算过程会较为繁琐。

本申请实施例先获取包围 K 个模型的 P 个包围盒，然后计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例的近似值，能够简化比例的计算过程，提高比例的计算效果，从而可以提高确定出目标抗锯齿算法的效率。

基于前述各个实施例的说明可知，对于一帧图像来说，只要该帧图像包含的所有的模型中，存在使用纹理数据且顶点重复度小于目标重复度的 K 个模型，且这 K 个模型满足预先设定的条件，则可以选择 FXAA 算法作为目标抗锯齿算法。

而获取调用图形 API 的指令的过程是一个持续的过程，所以可以在获取到一部分指令后，便可以根据这一部分指令中包含的模型的信息判断，是否存在满足预先设定的条件 K 个模型。若存在，则可以选择 FXAA 算法作为目标抗锯齿算法，若不存在，则可以继续获取调用图形 API 的指令，并继续进行判断。

因此，M 个模型可以是所属图像中的所有模型，也可以是所属图像中的所有模型的一部分模型。

例如，渲染一帧图像需要渲染 100 个模型；则可以在获取到 20 个模型的渲染信息后，通过这 20 个模型的渲染信息判断是否存在满足预先设定的条件 K 个模型，此时 M 为 20；若不满足，则可以继续获取模型的渲染信息，在获取到 30 个模型的渲染信息后，通过这 30 个模型的渲染信息判断是否存在满足预先设定的条件 K 个模型，此时 M 为 30；重复上述过程，直到当 M 为 100 时，再进行一次判断，此时，若存在满足预先设定的条件 K 个模型，则选择 FXAA 算法作为目标抗锯齿算法。

可以理解的是，当 M 个模型为所属图像中的所有模型时，若仍不存在满足条件的 K 个模型，则可以不选择 FXAA 算法作为目标抗锯齿算法，而选择其他算法作为目标抗锯齿算法，下面对此进行具体介绍。

示例性地，M 个模型为所属图像中的所有模型，多种抗锯齿算法还包括多重采样抗锯齿 MSAA 算法。

基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法还包括：

基于 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足条件，选择 MSAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

可以理解的是，若 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足条件，则意味着使用纹理数据且顶点重复度较小的 K 个模型不满足条件，则 FXAA 算法不适用于当前

图像的场景；相应地，说明 M 个模型中存在较多的重复度较高的模型或不适用纹理数据的模型，基于前述步骤 202 的相关说明可知，此时可以选择 MSAA 算法作为目标抗锯齿算法。

由于 FXAA 算法和 MSAA 算法属于适用性较高的两种抗锯齿算法，因此在本申请实施例中，先基于 M 个模型的渲染信息判断 FXAA 算法是否适用于当前图像的场景，若适用，则选择 FXAA 算法作为目标抗锯齿算法；若不适用，则选择 MSAA 算法作为目标抗锯齿算法；根据不同的场景对 FXAA 算法和 MSAA 算法进行不断切换，可以提高游戏应用程序的抗锯齿效果。

为了便于理解本申请实施例提供的选择抗锯齿算法的方法，下面提供一具体应用例。

如图 15 所示，该应用例的场景中游戏运行阶段，具体包括：

第一步，获取 M 个模型的渲染信息，具体的获取过程可参阅步骤 101 和步骤 204 的相关说明；

第二步，基于 M 个模型的渲染信息从 M 个模型中识别出使用纹理信息的 N 个模型，具体识别过程可参阅步骤 205 的相关说明；

第三步，从 N 个模型中识别出顶点重复度小于目标重复度的 K 个模型，具体识别过程可参阅步骤 206 的相关说明；

第四步，通过遍历 K 个模型对应的 VBO，获取 K 个模型对应的 P 个包围盒，具体可参阅步骤 301 的相关说明；

第五步，基于 P 个包围盒和转换矩阵计算 K 个模型在屏幕空间的投影面积，具体可参阅步骤 302 的相关说明；

第六步，判断 K 个模型在屏幕空间的投影面积与屏幕面积的比例是否大于 $1/3$ ，当大于 $1/3$ 时，选择 FXAA 算法作为目标抗锯齿算法；当小于 $1/3$ 时，选择 MSAA 算法作为目标抗锯齿算法。

下面通过游戏中具体的图像说明本申请实施例中选择抗锯齿算法的方法的效果。

具体地，将本申请实施例中选择抗锯齿算法的方法应用于一款游戏应用程序，对于第一种场景，由于草等通过贴图绘制且顶点重复度低的模型占屏幕的比例较大，因此选择了 FXAA 算法作为目标抗锯齿算法，利用 FXAA 算法进行抗锯齿处理后的图像如图 16 所示；对于第二种场景，由于草等通过贴图绘制且顶点重复度低的模型占屏幕的比例较小，因此选择了 MSAA 算法作为目标抗锯齿算法，利用 MSAA 算法进行抗锯齿处理后的图像如图 17 所示。

为了体验本申请实施例的抗锯齿效果，申请人做了对比试验：采用 MSAA 算法对第一种场景进行抗锯齿处理，得到如图 18 所示的图像；采用 FXAA 算法对第二种场景进行抗锯齿处理，得到如图 19 所示的图像。

对比图 16 和图 18 可以看出，在本申请实施例选择 FXAA 算法进行抗锯齿处理得到的图像中，草的边缘的锯齿较小，画质更好；而在选择 MSAA 算法进行抗锯齿处理得到的图像中，草的边缘的锯齿较大，画质更差。

同样地，对比图 17 和图 19 可以看出，在本申请实施例选择 MSAA 算法进行抗锯齿处理得到的图像中，电线的边缘的锯齿较小，画质更好；而在选择 FXAA 算法进行抗锯齿处理得

到的图像中，电线的边缘的锯齿较大，画质更差。

请参阅图 20，本申请实施例还提供了一种选择抗锯齿算法的装置的一个实施例，包括：

获取单元 401，用于获取用于调用应用程序接口 API 的至少一个指令，至少一个指令中携带 M 个模型各自的渲染信息，M 个模型属于同一帧图像，M 为正整数；

5 选择单元 402，用于基于 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；

发送单元 403，用于向图形处理器 GPU 发送指示信息，指示信息指示 GPU 基于目标抗锯齿算法对至少一帧图像进行渲染。

10 作为一种实现方式，多种抗锯齿算法包括快速近似抗锯齿 FXAA 算法；选择单元 402，用于从 M 个模型中确定渲染信息包含纹理信息的 N 个模型，纹理信息指示在渲染模型的过程中使用纹理数据，N 为小于或等于 M 的正整数；计算 N 个模型各自的顶点重复度；基于 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择 FXAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

15 作为一种实现方式，条件包括 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例。

作为一种实现方式，选择单元 402 还用于计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

20 作为一种实现方式，选择单元 402 用于获取包围 K 个模型的 P 个包围盒，每个包围盒为包围 K 个模型中至少一个模型的几何体；基于转换矩阵计算 P 个包围盒在屏幕空间中的投影面积，转换矩阵用于将 P 个包围盒从模型空间投影到屏幕空间；计算 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，并将 P 个包围盒在屏幕空间中的投影面积与屏幕面积的比例，作为 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

25 作为一种实现方式，M 个模型各自的渲染信息还包括 M 个模型各自对应的顶点缓冲对象 VBO；选择单元 402，用于从 K 个模型各自对应的顶点缓冲对象 VBO 中获取 K 个模型各自对应的顶点数据；基于 K 个模型各自对应的顶点数据确定 P 个包围盒。

30 作为一种实现方式，N 个模型包含目标模型，目标模型的渲染信息还包括目标模型对应的索引缓冲对象 EBO；选择单元 402，用于从目标模型对应的 EBO 中获取目标模型对应的顶点索引的数量和顶点索引键值的数量；基于顶点索引的数量和顶点索引键值的数量计算目标模型的顶点重复度，目标模型的顶点重复度为目标差值与顶点索引的数量的比值，目标差值等于于顶点索引的数量与顶点索引键值的数量的差值。

作为一种实现方式，M 个模型为所属图像中的所有模型，多种抗锯齿算法还包括多重采样抗锯齿 MSAA 算法；选择单元 402 还用于基于 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足条件，选择 MSAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

35 作为一种实现方式，该装置还包括：确定单元 404，用于获取多种场景下的多帧图像，每种场景对应至少一帧图像；计算多帧图像中特定的至少一个模型的顶点重复度；基于特定的至少一个模型的顶点重复度确定目标重复度。

其中，以上各单元的具体实现、相关说明以及技术效果请参考本申请实施例方法部分

的描述。

本申请实施例还提供了另一种终端设备，如图 21 所示，为了便于说明，仅示出了与本申请实施例相关的部分，具体技术细节未揭示的，请参照本申请实施例方法部分。该终端可以为包括手机、平板电脑、个人数字助理（英文全称：Personal Digital Assistant，英文缩写：PDA）、销售终端（英文全称：Point of Sales，英文缩写：POS）、车载电脑等任意终端设备，以终端为手机为例：

图 21 示出的是与本申请实施例提供的终端相关的手机的部分结构的框图。参考图 21，手机包括：射频（英文全称：Radio Frequency，英文缩写：RF）电路 1010、存储器 1020、输入单元 1030、显示单元 1040、传感器 1050、音频电路 1060、无线保真（英文全称：wireless fidelity，英文缩写：WiFi）模块 1070、中央处理器 CPU1080、图形处理器 GPU1090 以及电源等部件。本领域技术人员可以理解，图 21 中示出的手机结构并不构成对手机的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。

下面结合图 21 对手机的各个构成部件进行具体的介绍：

RF 电路 1010 可用于收发信息或通话过程中，信号的接收和发送，特别地，将基站的下行信息接收后，给 CPU1080 处理；另外，将设计上的数据发送给基站。通常，RF 电路 1010 包括但不限于天线、至少一个放大器、收发信机、耦合器、低噪声放大器（英文全称：Low Noise Amplifier，英文缩写：LNA）、双工器等。此外，RF 电路 1010 还可以通过无线通信与网络和其他设备通信。上述无线通信可以使用任一通信标准或协议，包括但不限于全球移动通讯系统（英文全称：Global System of Mobile communication，英文缩写：GSM）、通用分组无线服务（英文全称：General Packet Radio Service，GPRS）、码分多址（英文全称：Code Division Multiple Access，英文缩写：CDMA）、宽带码分多址（英文全称：Wideband Code Division Multiple Access，英文缩写：WCDMA）、长期演进（英文全称：Long Term Evolution，英文缩写：LTE）、电子邮件、短消息服务（英文全称：Short Messaging Service，SMS）等。

存储器 1020 可用于存储软件程序以及模块，CPU1080 通过运行存储在存储器 1020 的软件程序以及模块，从而执行手机的各种功能应用以及数据处理。存储器 1020 可主要包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需的应用程序（比如声音播放功能、图像播放功能等）等；存储数据区可存储根据手机的使用所创建的数据（比如音频数据、电话本等）等。此外，存储器 1020 可以包括高速随机存取存储器，还可以包括非易失性存储器，例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

输入单元 1030 可用于接收输入的数字或字符信息，以及产生与手机的用户设置以及功能控制有关的键信号输入。具体地，输入单元 1030 可包括触控面板 1031 以及其他输入设备 1032。触控面板 1031，也称为触摸屏，可收集用户在其上或附近的触摸操作（比如用户使用手指、触笔等任何适合的物体或附件在触控面板 1031 上或在触控面板 1031 附近的操作），并根据预先设定的程式驱动相应的连接装置。可选的，触控面板 1031 可包括触摸检测装置和触摸控制器两个部分。其中，触摸检测装置检测用户的触摸方位，并检测触摸操

作带来的信号，将信号传送给触摸控制器；触摸控制器从触摸检测装置上接收触摸信息，并将它转换成触点坐标，再送给 CPU1080，并能接收 CPU1080 发来的命令并加以执行。此外，可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触控面板 1031。除了触控面板 1031，输入单元 1030 还可以包括其他输入设备 1032。具体地，其他输入设备 1032 可以包括但不限于物理键盘、功能键（比如音量控制按键、开关按键等）、轨迹球、鼠标、操作杆等中的一种或多种。

显示单元 1040 可用于显示由用户输入的信息或提供给用户的信息以及手机的各种菜单。显示单元 1040 可包括显示面板 1041，可选的，可以采用液晶显示器（英文全称：Liquid Crystal Display，英文缩写：LCD）、有机发光二极管（英文全称：Organic Light-Emitting Diode，英文缩写：OLED）等形式来配置显示面板 1041。进一步的，触控面板 1031 可覆盖显示面板 1041，当触控面板 1031 检测到在其上或附近的触摸操作后，传送给 CPU1080 以确定触摸事件的类型，随后 CPU1080 根据触摸事件的类型在显示面板 1041 上提供相应的视觉输出。虽然在图 21 中，触控面板 1031 与显示面板 1041 是作为两个独立的部件来实现手机的输入和输入功能，但是在某些实施例中，可以将触控面板 1031 与显示面板 1041 集成而实现手机的输入和输出功能。

手机还可包括至少一种传感器 1050，比如光传感器、运动传感器以及其他传感器。具体地，光传感器可包括环境光传感器及接近传感器，其中，环境光传感器可根据环境光线的明暗来调节显示面板 1041 的亮度，接近传感器可在手机移动到耳边时，关闭显示面板 1041 和/或背光。作为运动传感器的一种，加速计传感器可检测各个方向上（一般为三轴）加速度的大小，静止时可检测出重力的大小及方向，可用于识别手机姿态的应用（比如横竖屏切换、相关游戏、磁力计姿态校准）、振动识别相关功能（比如计步器、敲击）等；至于手机还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器，在此不再赘述。

音频电路 1060、扬声器 1061，传声器 1062 可提供用户与手机之间的音频接口。音频电路 1060 可将接收到的音频数据转换后的电信号，传输到扬声器 1061，由扬声器 1061 转换为声音信号输出；另一方面，传声器 1062 将收集的声音信号转换为电信号，由音频电路 1060 接收后转换为音频数据，再将音频数据输出 CPU1080 处理后，经 RF 电路 1010 以发送给比如另一手机，或者将音频数据输出至存储器 1020 以便进一步处理。

WiFi 属于短距离无线传输技术，手机通过 WiFi 模块 1070 可以帮助用户收发电子邮件、浏览网页和访问流式媒体等，它为用户提供了无线的宽带互联网访问。虽然图 21 示出了 WiFi 模块 1070，但是可以理解的是，其并不属于手机的必须构成，完全可以根据需要在不改变发明的本质的范围内而省略。

CPU1080 是手机的控制中心，利用各种接口和线路连接整个手机的各个部分，通过运行或执行存储在存储器 1020 内的软件程序和/或模块，以及调用存储在存储器 1020 内的数据，执行手机的各种功能和处理数据，从而对手机进行整体监控。可选的，CPU1080 可包括一个或多个处理单元；优选的，CPU1080 可集成应用处理器和调制解调处理器，其中，应用处理器主要处理操作系统、用户界面和应用程序等，调制解调处理器主要处理无线通

信。可以理解的是，上述调制解调处理器也可以不集成到 CPU1080 中。

GPU1090 主要用于将存储器 1020 中的数据渲染成图像，在显示单元 1040 中显示。

手机还包括给各个部件供电的电源（比如电池），优选的，电源可以通过电源管理系统与 CPU1080 逻辑相连，从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。

5 尽管未示出，手机还可以包括摄像头、蓝牙模块等，在此不再赘述。

在本申请实施例中，该终端所包括的 CPU1080 还具有以下功能：

获取用于调用应用程序接口 API 的至少一个指令，所述至少一个指令中携带 M 个模型各自的渲染信息，所述 M 个模型属于同一帧图像，M 为正整数；

10 根据所述 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；

向图形处理器 GPU 发送指示信息，所述指示信息指示所述 GPU 基于所述目标抗锯齿算法对至少一帧图像进行渲染。

在本申请实施例中，该终端所包括的 GPU1090 还具有以下功能：

15 接收来自 CPU1080 的指示信息，并基于该指示信息采用目标抗锯齿算法在图像渲染过程中进行抗锯齿处理。

本申请实施例还提供一种芯片，包括一个或多个处理器。所述处理器中的部分或全部用于读取并执行存储器中存储的计算机程序，以执行图 6、图 7 和图 12 对应的各实施例的方法。

20 可选地，该芯片该包括存储器，该存储器与该处理器通过电路或电线与存储器连接。进一步可选地，该芯片还包括通信接口，处理器与该通信接口连接。通信接口用于接收需要处理的数据和/或信息，处理器从该通信接口获取该数据和/或信息，并对该数据和/或信息进行处理，并通过该通信接口输出处理结果。该通信接口可以是输入输出接口。

25 在一些实现方式中，所述一个或多个处理器中还可以有部分处理器是通过专用硬件的方式来实现以上方法中的部分步骤，例如涉及神经网络模型的处理可以由专用神经网络处理器或图形处理器来实现。

本申请实施例提供的方法可以由一个芯片实现，也可以由多个芯片协同实现。

本申请实施例还提供了一种计算机存储介质，该计算机存储介质用于储存为上述计算机设备所用的计算机软件指令，其包括用于执行为计算机设备所设计的程序。

该计算机设备可以如前述图 15 所描述的运行应用程序的装置。

30 本申请实施例还提供了一种计算机程序产品，该计算机程序产品包括计算机软件指令，该计算机软件指令可通过处理器进行加载来实现前述图 6、图 7 和图 12 所示的方法中的流程。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

35 在本申请所提供的几个实施例中，应该理解到，所揭露的系统，装置和方法，可以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，机械或其它的形式。

5 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。

另外，在本申请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。

10 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，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介质中，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）执行本申请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、移动硬盘、只读存储器（ROM，Read-Only Memory）、
15 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。

权利要求

1. 一种选择抗锯齿算法的方法，其特征在于，包括：

获取用于调用应用程序接口 API 的至少一个指令，所述至少一个指令中携带 M 个模型各自的渲染信息，所述 M 个模型属于同一帧图像，M 为正整数；

5 基于所述 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；

向图形处理器 GPU 发送指示信息，所述指示信息指示所述 GPU 基于所述目标抗锯齿算法对至少一帧图像进行渲染。

2. 根据权利要求 1 所述的方法，其特征在于，所述多种抗锯齿算法包括快速近似抗锯齿 FXAA 算法；

所述基于所述 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法包括：

从所述 M 个模型中确定渲染信息包含纹理信息的 N 个模型，所述纹理信息指示在渲染模型的过程中使用纹理数据，N 为小于或等于 M 的正整数；

15 计算所述 N 个模型各自的顶点重复度；

基于所述 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择所述 FXAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

3. 根据权利要求 2 所述的方法，其特征在于，所述条件包括所述 K 个模型在屏幕空间中的投影面积与屏幕面积的比例大于目标比例。

20 4. 根据权利要求 2 或 3 所述的方法，其特征在于，在所述计算所述 N 个模型各自的顶点重复度之后，在基于所述 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择所述 FXAA 算法作为目标抗锯齿算法之前，所述方法还包括：

计算所述 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

25 5. 根据权利要求 4 所述的方法，其特征在于，所述计算 K 个模型在屏幕空间中的投影面积与屏幕面积的比例包括：

获取包围 K 个模型的 P 个包围盒，每个包围盒为包围所述 K 个模型中至少一个模型的几何体；

基于转换矩阵计算所述 P 个包围盒在屏幕空间中的投影面积，所述转换矩阵用于将所述 P 个包围盒从模型空间投影到所述屏幕空间；

30 计算所述 P 个包围盒在所述屏幕空间中的投影面积与屏幕面积的比例，并将所述 P 个包围盒在所述屏幕空间中的投影面积与屏幕面积的比例，作为所述 K 个模型在屏幕空间中的投影面积与屏幕面积的比例。

6. 根据权利要求 5 所述的方法，其特征在于，所述 M 个模型各自的渲染信息还包括所述 M 个模型各自对应的顶点缓冲对象 VBO；

35 获取包围 K 个模型的 P 个包围盒包括：

从所述 K 个模型各自对应的顶点缓冲对象 VBO 中获取所述 K 个模型各自对应的顶点数据；

基于所述 K 个模型各自对应的顶点数据确定 P 个包围盒。

7. 根据权利要求 2 至 6 中任一项所述的方法，其特征在于，所述 N 个模型包含目标模型，所述目标模型的渲染信息还包括所述目标模型对应的索引缓冲对象 EBO；

所述计算所述 N 个模型各自的顶点重复度包括：

5 从所述目标模型对应的 EBO 中获取所述目标模型对应的顶点索引的数量和顶点索引键值的数量；

基于所述顶点索引的数量和所述顶点索引键值的数量计算所述目标模型的顶点重复度，所述目标模型的顶点重复度为目标差值与所述顶点索引的数量的比值，所述目标差值等于于所述顶点索引的数量与所述顶点索引键值的数量的差值。

10 8. 根据权利要求 2 至 7 中任一项所述的方法，其特征在于，所述 M 个模型为所属图像中的所有模型，所述多种抗锯齿算法还包括多重采样抗锯齿 MSAA 算法；

所述基于所述 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法还包括：

15 基于所述 N 个模型中顶点重复度小于目标重复度的 K 个模型不满足所述条件，选择所述 MSAA 算法作为目标抗锯齿算法，K 为小于或等于 N 的正整数。

9. 根据权利要求 2 至 8 中任一项所述的方法，其特征在于，在所述基于所述 N 个模型中顶点重复度小于目标重复度的 K 个模型满足条件，选择所述 FXAA 算法作为目标抗锯齿算法之前，所述方法还包括：

获取多种场景下的多帧图像，每种场景对应至少一帧图像；

20 计算所述多帧图像中至少一个模型的顶点重复度；

基于所述至少一个模型的顶点重复度确定所述目标重复度。

10. 一种选择抗锯齿算法的装置，其特征在于，包括：

获取单元，用于获取用于调用应用程序接口 API 的至少一个指令，所述至少一个指令中携带 M 个模型各自的渲染信息，所述 M 个模型属于同一帧图像，M 为正整数；

25 选择单元，用于基于所述 M 个模型各自的渲染信息从多种抗锯齿算法中选择一种抗锯齿算法作为目标抗锯齿算法；

发送单元，用于向图形处理器 GPU 发送指示信息，所述指示信息指示所述 GPU 基于所述目标抗锯齿算法对至少一帧图像进行渲染。

30 11. 一种终端设备，其特征在于，包括：一个或多个处理器和存储器；其中，所述存储器中存储有计算机可读指令；

所述一个或多个处理器读取所述计算机可读指令，以使所述计算机设备实现如权利要求 1 至 9 中任一项所述的方法。

12. 一种计算机可读存储介质，其特征在于，包括计算机可读指令，当所述计算机可读指令在计算机上运行时，使得所述计算机执行如权利要求 1 至 9 中任一项所述的方法。

35 13. 一种计算机程序产品，其特征在于，包括计算机可读指令，当所述计算机可读指令在计算机上运行时，使得所述计算机执行如权利要求 1 至 9 中任一项所述的方法。

— 1/12 —



图 1



图 2

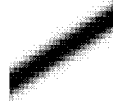


图 3



图 4

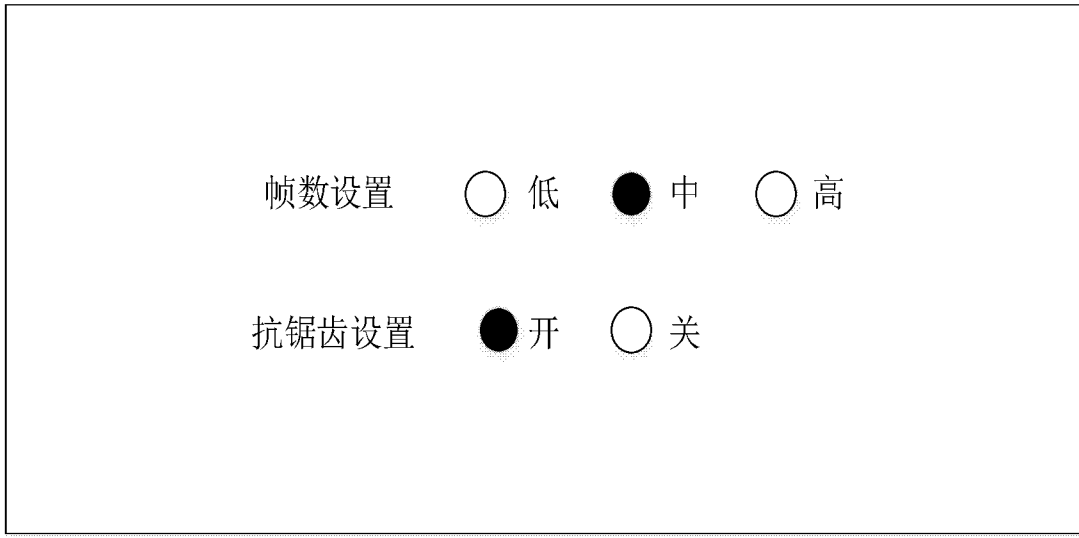


图 5

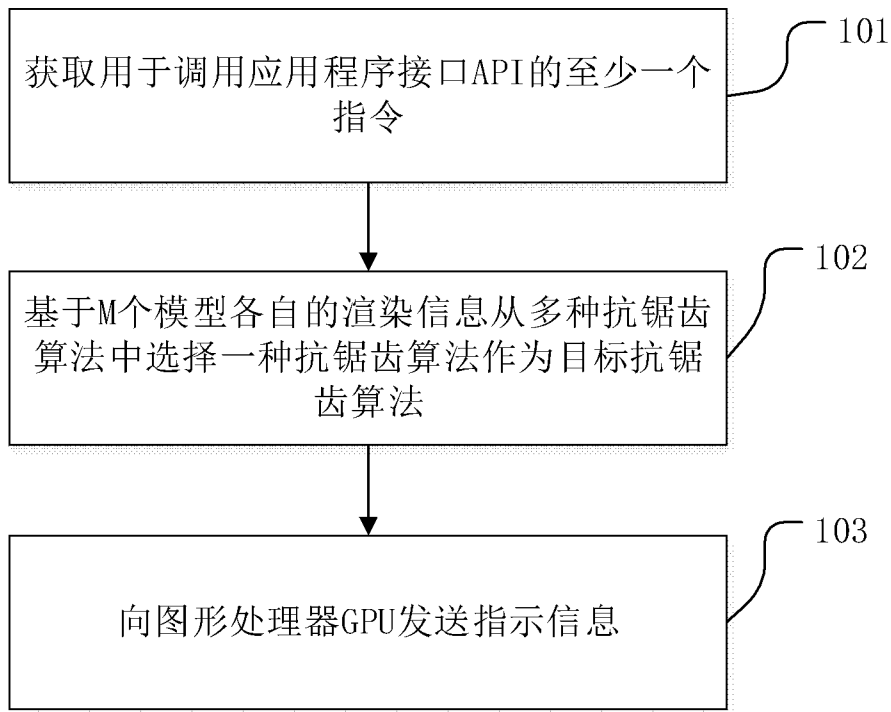


图 6

- 5/12 -

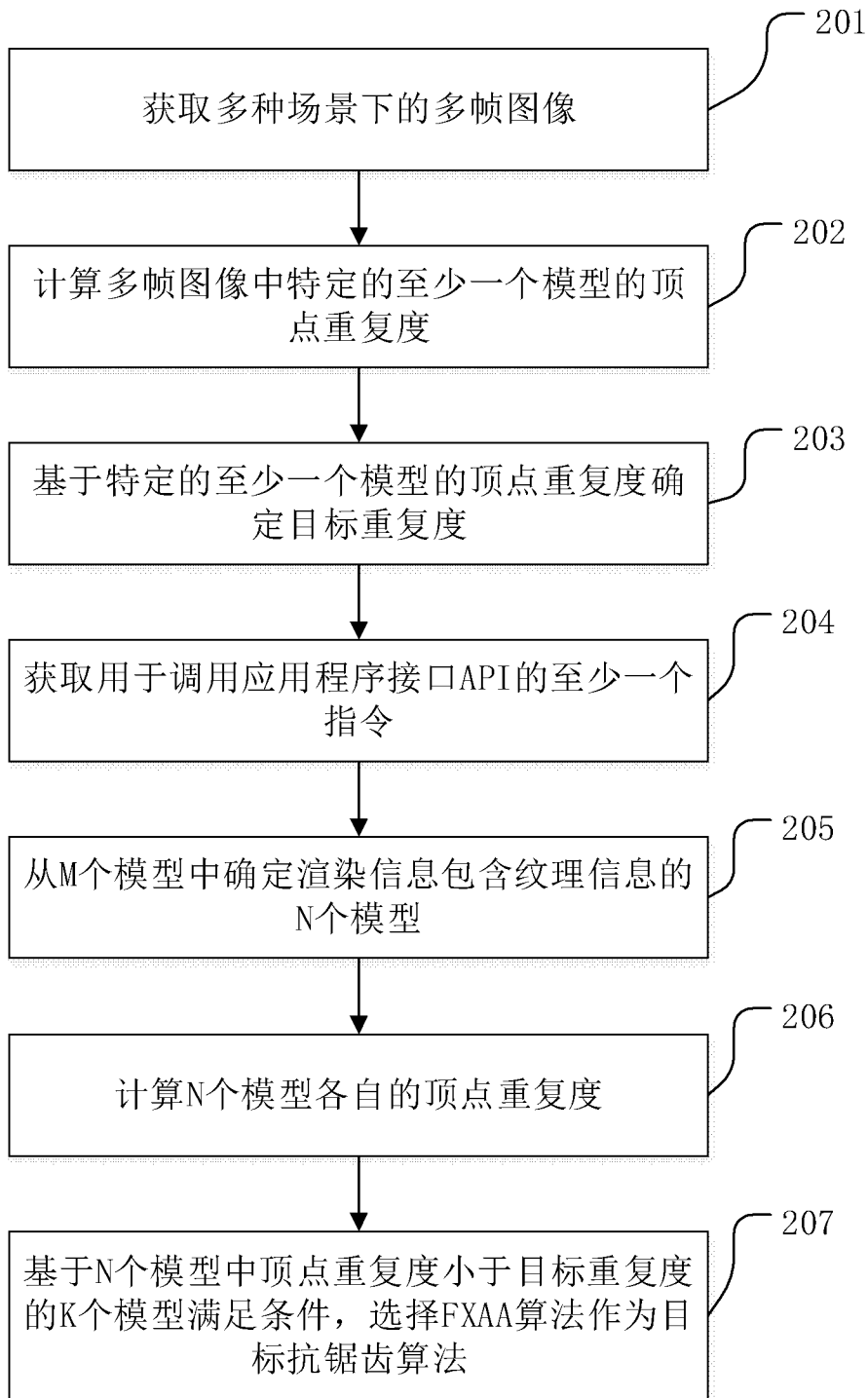


图 7

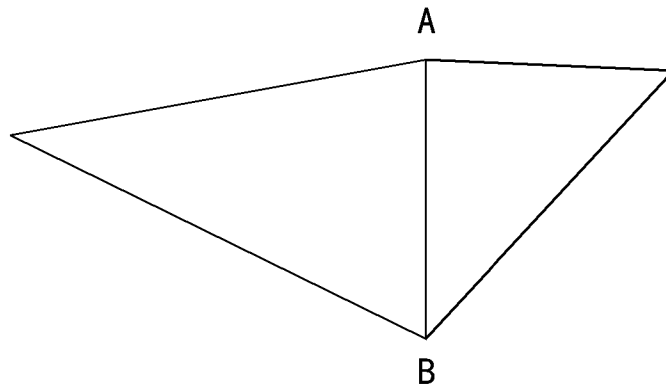


图 8

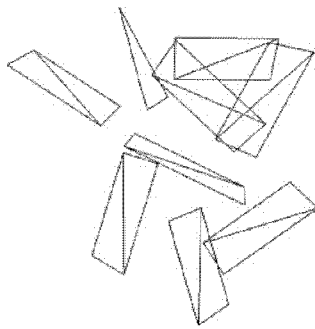


图 9

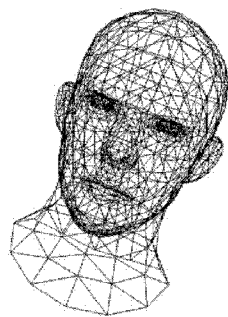


图 10

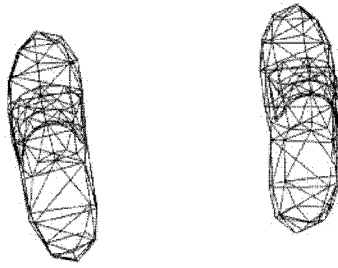


图 11

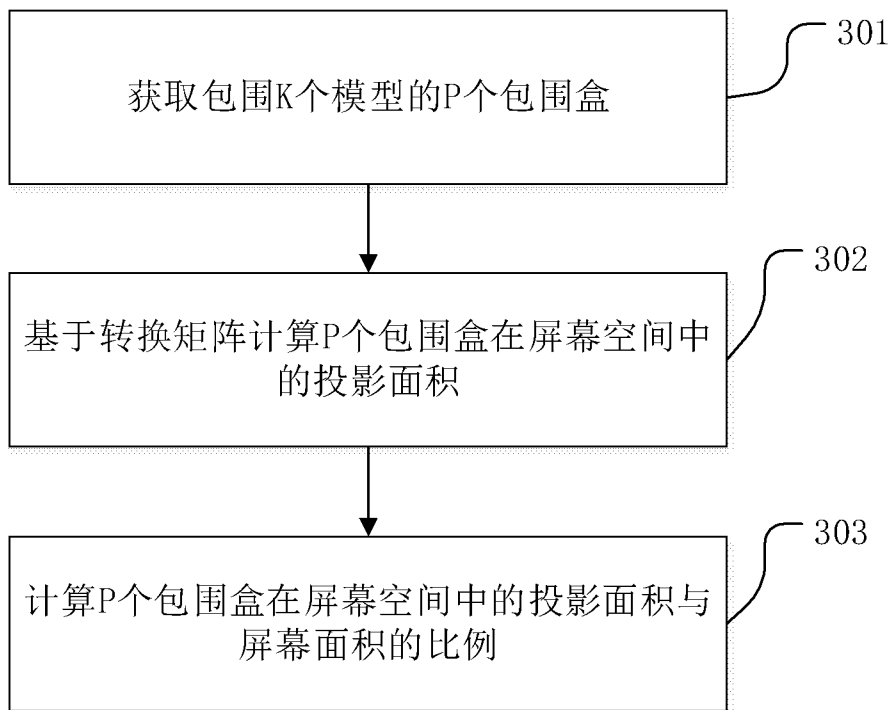


图 12

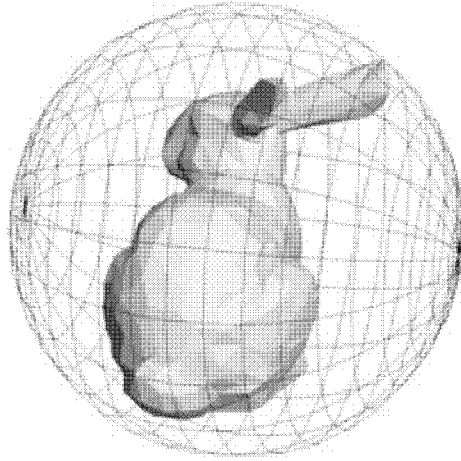


图 13

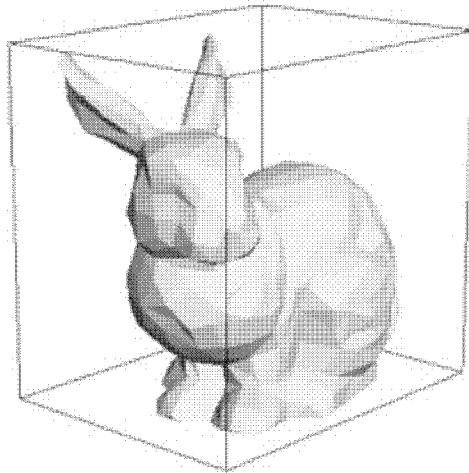


图 14

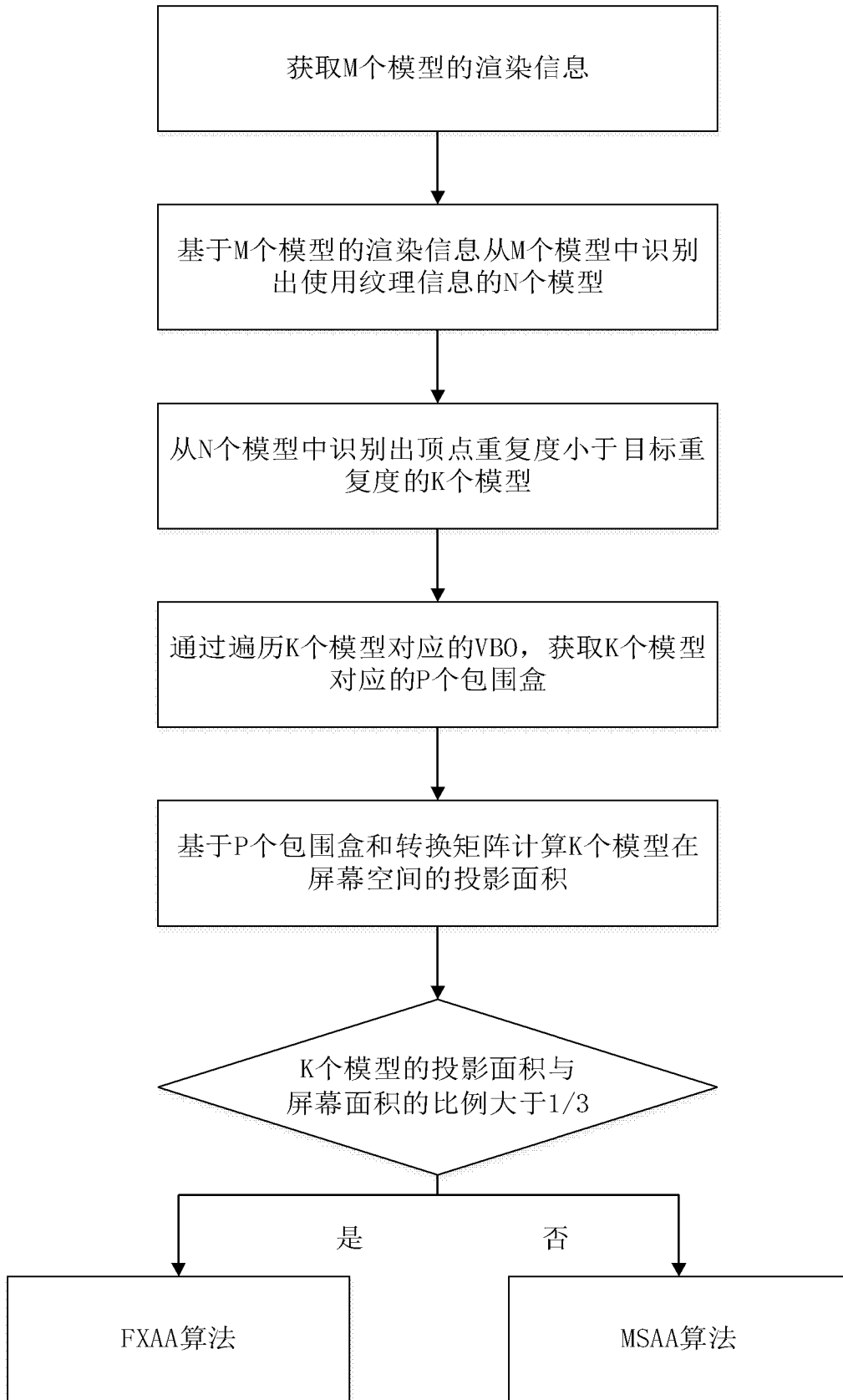


图 15

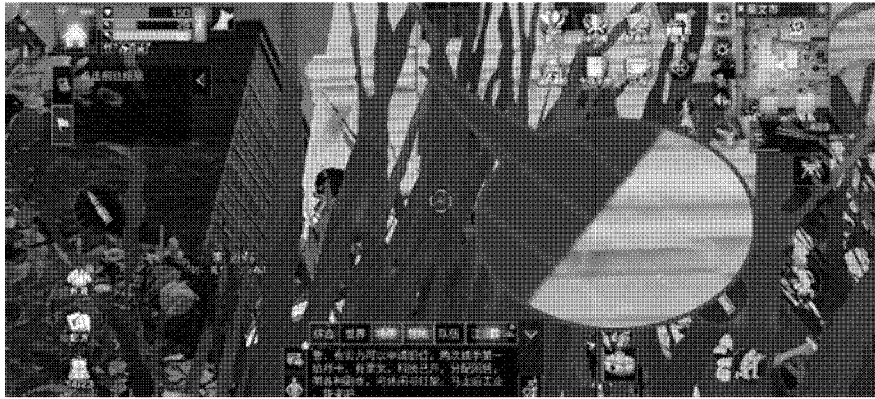


图 16

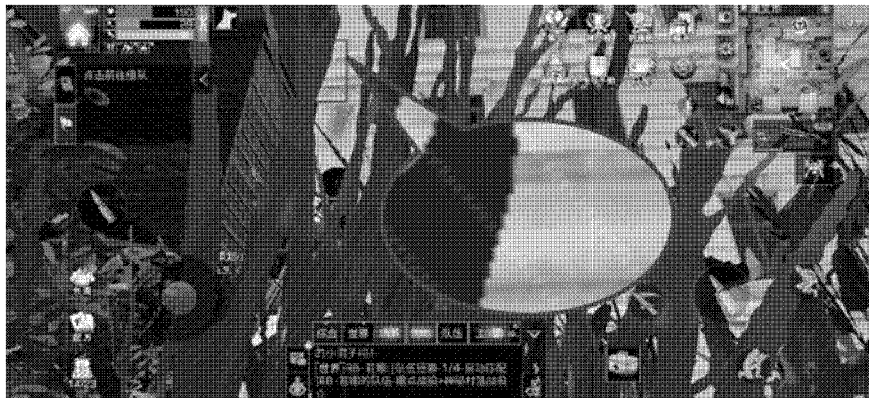


图 17

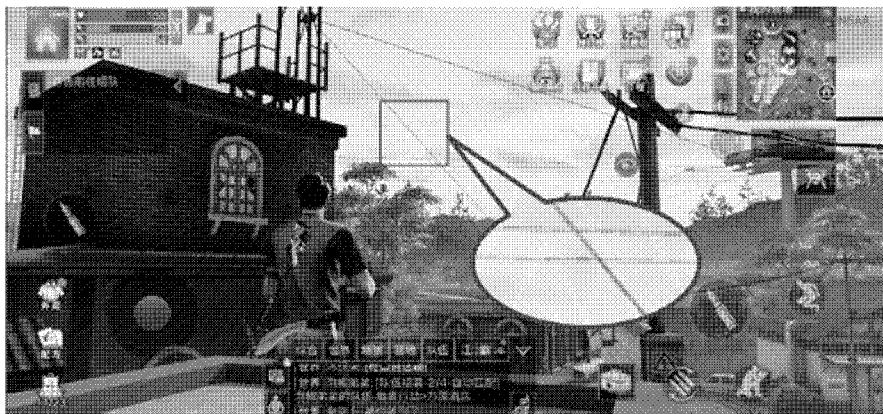


图 18

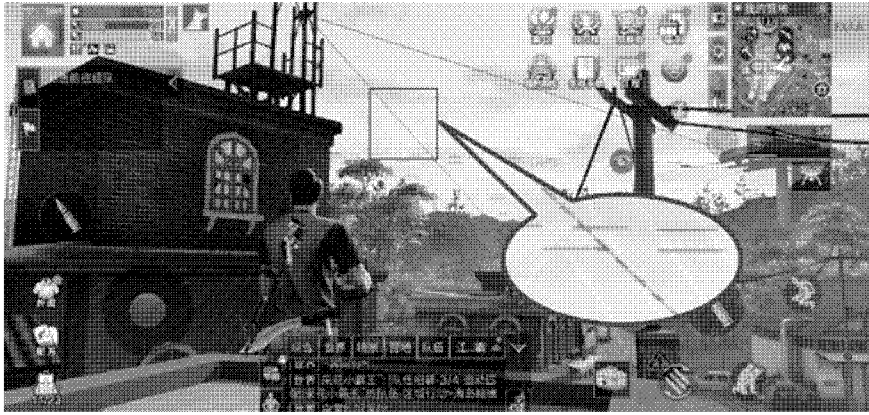


图 19



图 20

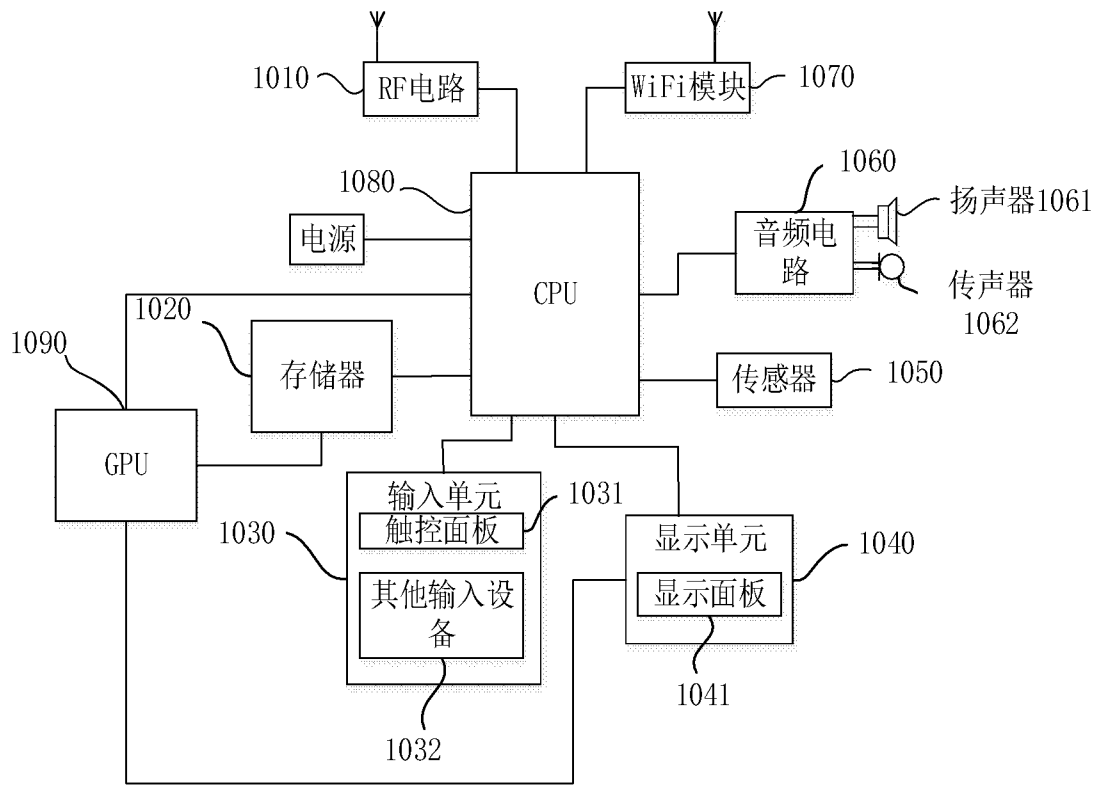


图 21

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/130070

A. CLASSIFICATION OF SUBJECT MATTER G06T 15/20(2011.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06T Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WPI, EPODOC, CNPAT, CNKI, IEEE, GOOGLE: 抗锯齿, 抗走样, 反走样, 算法, 选择, 渲染, 模型, 顶点, 重复度, 包围盒, 索引, 串, 纹理, 阈值, 面积比, 级别; VBO, FXAA, MSAA, EBO, anti-aliasing, antialiasing, algorithm?, select+, choos+, render +, model, vertex?, degree s repetition, bounding box, index, string, texture, threshold, area ratio, level		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 108846815 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 20 November 2018 (2018-11-20) claims 1-15, description paragraphs [0037]-[0238], figures 1-16	1, 10-13
A	CN 105574817 A (HUAWEI TECHNOLOGIES CO., LTD.) 11 May 2016 (2016-05-11) entire document	1-13
A	CN 108022218 A (XI'AN AVIATION INSTITUTE OF COMPUTING TECHNOLOGY, AVIATION INDUSTRY CORPORATION OF CHINA) 11 May 2018 (2018-05-11) entire document	1-13
A	CN 109712225 A (SUPERMAP SOFTWARE CO., LTD.) 03 May 2019 (2019-05-03) entire document	1-13
A	GB 201518260 D0 (SAMSUNG ELECTRONICS CO., LTD.) 02 December 2015 (2015-12-02) entire document	1-13
A	CN 102708585 A (BEIJING PIXEL SOFTWARE TECHNOLOGY JOINT-STOCK CO., LTD.) 03 October 2012 (2012-10-03) entire document	1-13
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 24 January 2022		Date of mailing of the international search report 11 February 2022
Name and mailing address of the ISA/CN China National Intellectual Property Administration (ISA/CN) No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088, China Facsimile No. (86-10)62019451		Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/130070

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	108846815	A	20 November 2018	None			
CN	105574817	A	11 May 2016	US	2016110847	A1	21 April 2016
				JP	2016081042	A	16 May 2016
				JP	6135952	B2	31 May 2017
				EP	3009982	A1	20 April 2016
				KR	20160045561	A	27 April 2016
				KR	101710966	B1	28 February 2017
CN	108022218	A	11 May 2018	None			
CN	109712225	A	03 May 2019	None			
GB	201518260	D0	02 December 2015	GB	2531936	A	04 May 2016
CN	102708585	A	03 October 2012	CN	102708585	B	20 May 2015

<p>A. 主题的分类</p> <p>G06T 15/20 (2011.01) i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p>																																			
<p>B. 检索领域</p> <p>检索的最低限度文献(标明分类系统和分类号)</p> <p>G06T</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))</p> <p>WPI, EPODOC, CNPAT, CNKI, IEEE, GOOGLE: 抗锯齿, 抗走样, 反走样, 算法, 选择, 渲染, 模型, 顶点, 重复度, 包围盒, 索引, 串, 纹理, 阈值, 面积比, 级别; VBO, FXAA, MSAA, EBO, anti-aliasing, antialiasing, algorithm?, select+, choos+, render+, model, vertex?, degree s repetition, bounding box, index, string, texture, threshold, area ratio, level</p>																																			
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>CN 108846815 A (腾讯科技深圳有限公司) 2018年11月20日 (2018 - 11 - 20) 权利要求1-15, 说明书第[0037]-[0238]段, 图1-16</td> <td>1, 10-13</td> </tr> <tr> <td>A</td> <td>CN 105574817 A (华为技术有限公司) 2016年5月11日 (2016 - 05 - 11) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>CN 108022218 A (中国航空工业集团公司西安航空计算技术研究所) 2018年5月11日 (2018 - 05 - 11) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>CN 109712225 A (北京超图软件股份有限公司) 2019年5月3日 (2019 - 05 - 03) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>GB 201518260 D0 (SAMSUNG ELECTRONICS CO., LTD) 2015年12月2日 (2015 - 12 - 02) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>CN 102708585 A (北京像素软件科技股份有限公司) 2012年10月3日 (2012 - 10 - 03) 全文</td> <td>1-13</td> </tr> </tbody> </table> <p><input type="checkbox"/> 其余文件在C栏的续页中列出。 <input checked="" type="checkbox"/> 见同族专利附件。</p> <table border="0"> <tr> <td>* 引用文件的具体类型:</td> <td>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</td> </tr> <tr> <td>“A” 认为不特别相关的表示了现有技术一般状态的文件</td> <td>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</td> </tr> <tr> <td>“E” 在国际申请日的当天或之后公布的在先申请或专利</td> <td>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</td> </tr> <tr> <td>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</td> <td>“&” 同族专利的文件</td> </tr> <tr> <td>“O” 涉及口头公开、使用、展览或其他方式公开的文件</td> <td></td> </tr> <tr> <td>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</td> <td></td> </tr> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 108846815 A (腾讯科技深圳有限公司) 2018年11月20日 (2018 - 11 - 20) 权利要求1-15, 说明书第[0037]-[0238]段, 图1-16	1, 10-13	A	CN 105574817 A (华为技术有限公司) 2016年5月11日 (2016 - 05 - 11) 全文	1-13	A	CN 108022218 A (中国航空工业集团公司西安航空计算技术研究所) 2018年5月11日 (2018 - 05 - 11) 全文	1-13	A	CN 109712225 A (北京超图软件股份有限公司) 2019年5月3日 (2019 - 05 - 03) 全文	1-13	A	GB 201518260 D0 (SAMSUNG ELECTRONICS CO., LTD) 2015年12月2日 (2015 - 12 - 02) 全文	1-13	A	CN 102708585 A (北京像素软件科技股份有限公司) 2012年10月3日 (2012 - 10 - 03) 全文	1-13	* 引用文件的具体类型:	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件	“A” 认为不特别相关的表示了现有技术一般状态的文件	“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性	“E” 在国际申请日的当天或之后公布的在先申请或专利	“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性	“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)	“&” 同族专利的文件	“O” 涉及口头公开、使用、展览或其他方式公开的文件		“P” 公布日先于国际申请日但迟于所要求的优先权日的文件	
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求																																	
X	CN 108846815 A (腾讯科技深圳有限公司) 2018年11月20日 (2018 - 11 - 20) 权利要求1-15, 说明书第[0037]-[0238]段, 图1-16	1, 10-13																																	
A	CN 105574817 A (华为技术有限公司) 2016年5月11日 (2016 - 05 - 11) 全文	1-13																																	
A	CN 108022218 A (中国航空工业集团公司西安航空计算技术研究所) 2018年5月11日 (2018 - 05 - 11) 全文	1-13																																	
A	CN 109712225 A (北京超图软件股份有限公司) 2019年5月3日 (2019 - 05 - 03) 全文	1-13																																	
A	GB 201518260 D0 (SAMSUNG ELECTRONICS CO., LTD) 2015年12月2日 (2015 - 12 - 02) 全文	1-13																																	
A	CN 102708585 A (北京像素软件科技股份有限公司) 2012年10月3日 (2012 - 10 - 03) 全文	1-13																																	
* 引用文件的具体类型:	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件																																		
“A” 认为不特别相关的表示了现有技术一般状态的文件	“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性																																		
“E” 在国际申请日的当天或之后公布的在先申请或专利	“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性																																		
“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)	“&” 同族专利的文件																																		
“O” 涉及口头公开、使用、展览或其他方式公开的文件																																			
“P” 公布日先于国际申请日但迟于所要求的优先权日的文件																																			
国际检索实际完成的日期	国际检索报告邮寄日期																																		
2022年1月24日	2022年2月11日																																		
ISA/CN的名称和邮寄地址	授权官员																																		
中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088	李敏																																		
传真号 (86-10)62019451	电话号码 86-(10)-53961526																																		

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/130070

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	108846815	A	2018年11月20日	无			
CN	105574817	A	2016年5月11日	US	2016110847	A1	2016年4月21日
				JP	2016081042	A	2016年5月16日
				JP	6135952	B2	2017年5月31日
				EP	3009982	A1	2016年4月20日
				KR	20160045561	A	2016年4月27日
				KR	101710966	B1	2017年2月28日
CN	108022218	A	2018年5月11日	无			
CN	109712225	A	2019年5月3日	无			
GB	201518260	D0	2015年12月2日	GB	2531936	A	2016年5月4日
CN	102708585	A	2012年10月3日	CN	102708585	B	2015年5月20日