(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 111033519 B
(45)授权公告日 2021.07.27

(21)申请号 201880041154.1
(22)申请日 2018.04.20
(65)同一申请的已公布的文献号
申请公布号 CN 111033519 A
(43)申请公布日 2020.04.17
(30)优先权数据
62/488,526 2017.04.21 US
62/653,056 2018.04.05 US
(85)PCT国际申请进入国家阶段日
2019.12.19
(86)PCT国际申请的申请数据
PCT/US2018/028645 2018.04.20
(87)PCT国际申请的公布数据
WO2018/195477 EN 2018.10.25
(73)专利权人 泽尼马克斯媒体公司
地址 美国马里兰州
(72)发明人 迈克尔·科皮茨
(74)专利代理机构 北京柏杉松知识产权代理事务所(普通合伙) 11413
代理人 侯丽英 刘继富
(51)Int.Cl.
G06K 9/36(2006.01)
(56)对比文件
US 2016227172 A1,2016.08.04
US 2014226490 A1,2014.08.14
US 8145451 B2,2012.03.27
US 2013083163 A1,2013.04.04
CN 103905836 A,2014.07.02
CN 103428488 A,2013.12.04
CN 105917649 A,2016.08.31
审查员 王晶

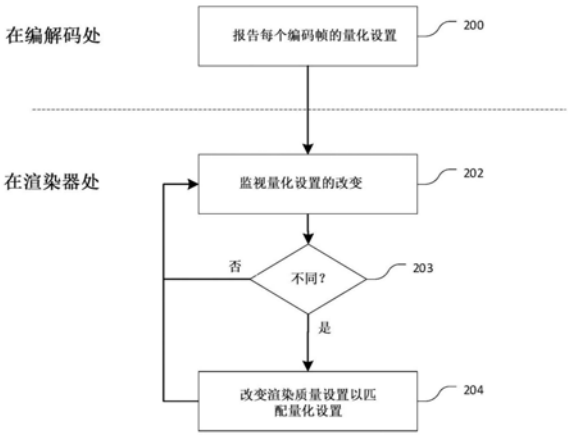
权利要求书3页 说明书8页 附图5页

(54)发明名称

用于编码器引导的自适应质量渲染的系统和方法

(57)摘要

本申请公开了用于改善与图像的渲染和编码有关的、优选地用于视频游戏环境中的计算机技术的系统和方法。在某些实施例中，编解码器用于针对编码器设置的部分范围编码一个或更多个参考图像，并且渲染器用于生成一个或更多个渲染质量设置配置文件、生成一个或更多个参考图像、针对一个或更多个参考图像中的每个，计算感知质量、针对一个或更多个渲染质量设置配置文件中的每个，重新渲染一个或更多个参考图像、以及针对一个或更多个重新渲染的参考图像中的每个，计算感知质量。渲染器对参考图像的感知质量与重新渲染的图像的感知质量进行比较并且匹配它们。这些匹配导致将一个或更多个编码器设置和与其匹配的渲染质量设置配置文件关联到查找表中。



CN 111033519 B

1.一种用于渲染的计算机实现的方法,包括以下步骤:

生成一个或更多个渲染质量设置配置文件;

生成一个或更多个参考图像;

针对编码器设置的部分范围,编码所述一个或更多个参考图像;

针对一个或更多个编码后的参考图像中的每个,计算第一感知质量;

针对一个或更多个渲染质量设置配置文件中的每个,重新渲染所述一个或更多个参考图像;

针对一个或更多个重新渲染的参考图像中的每个,计算一个或更多个第二感知质量;

将一个或更多个第一感知质量与一个或更多个第二感知质量进行比较,其中,一个或更多个第一感知质量与一个或更多个第二感知质量之间的匹配导致一个或更多个编码器设置和与其匹配的渲染质量设置配置文件在一个或更多个查找表中的关联;以及

基于所述查找表,以与编码帧基本相同的感知质量生成渲染图像。

2.根据权利要求1所述的计算机实现的方法,还包括步骤:计算重新渲染的参考图像中的每个的计算成本。

3.根据权利要求2所述的计算机实现的方法,还包括步骤:对参考图像和重新渲染的参考图像,应用所述重新渲染的参考图像中的每个的计算成本作为一个或更多个第一感知质量与一个或更多个第二感知质量的多个匹配的决胜点。

4.根据权利要求2所述的计算机实现的方法,其中,以渲染时间或时钟周期来衡量所述计算成本。

5.根据权利要求1所述的计算机实现的方法,其中,计算算法图像质量评估并且然后将其用于计算所述第一感知质量和第二感知质量。

6.根据权利要求1所述的计算机实现的方法,其中,通过应用决策树以编程方式缩小概率空间来执行将一个或更多个编码器设置和与其匹配的渲染质量设置配置文件的关联。

7.根据权利要求1所述的计算机实现的方法,其中,结构相似性指数被用于测量所述第一感知质量和所述第二感知质量。

8.根据权利要求1所述的计算机实现的方法,其中,使用量化参数的多个整数值或非整数部分范围来生成一个或更多个查找表。

9.根据权利要求8所述的计算机实现的方法,其中,量化参数的整数值为0到51。

10.根据权利要求1所述的计算机实现的方法,其中,所述质量设置配置文件是每个可用渲染质量设置的值的列表。

11.一种用于渲染的系统,包括:

编解码器,其针对编码器设置的部分范围编码一个或更多个参考图像;以及

渲染器;其中,所述渲染器用于:

生成一个或更多个渲染质量设置配置文件;

生成一个或更多个参考图像;

针对一个或更多个编码后的参考图像中的每个,计算第一感知质量;

针对所述一个或更多个渲染质量设置配置文件中的每个,重新渲染所述一个或更多个参考图像;

针对所述一个或更多个重新渲染的参考图像中的每个,计算一个或更多个第二感知质

量；

将一个或更多个第一感知质量与一个或更多个第二感知质量进行比较，其中，一个或更多个第一感知质量与一个或更多个第二感知质量之间的匹配导致一个或更多个编码器设置和与其匹配的渲染质量设置配置文件在查找表中的关联；以及

基于所述查找表，以与编码帧基本相同的感知质量生成渲染图像。

12.根据权利要求11所述的系统，还包括步骤：计算重新渲染的参考图像中的每个的计算成本。

13.根据权利要求12所述的系统，还包括步骤：对参考图像和重新渲染的参考图像，应用所述重新渲染的参考图像中的每个的计算成本作为一个或更多个第一感知质量与一个或更多个第二感知质量的多个匹配的决胜点。

14.根据权利要求12所述的系统，其中，所述计算成本是以渲染时间或时钟周期来衡量的。

15.根据权利要求11所述的系统，其中，计算算法图像质量评估并且然后将其用于计算所述第一感知质量和第二感知质量。

16.根据权利要求11所述的系统，其中，一个或更多个编码器设置和与其匹配的渲染质量设置配置文件的关联是通过应用决策树以编程方式缩小概率空间来执行的。

17.根据权利要求11所述的系统，其中，结构相似性指数被用于测量所述第一感知质量和所述第二感知质量。

18.根据权利要求11所述的系统，其中，一个或更多个查找表是使用量化参数的多个整数值或非整数部分范围来生成的。

19.根据权利要求18所述的系统，其中，量化参数的整数值为0到51。

20.根据权利要求11所述的系统，其中，所述质量设置配置文件是每个可用渲染质量设置的值的列表。

21.一种用于渲染的计算机实现的方法，包括以下步骤：

生成一个或更多个参考图像；

针对编码器设置的部分范围，编码一个或更多个参考图像；

针对一个或更多个编码后的参考图像中的每个，计算第一感知质量；

针对一个或更多个渲染质量设置配置文件中的每个，重新渲染所述一个或更多个参考图像；

针对一个或更多个重新渲染的参考图像中的每个，计算一个或更多个第二感知质量；

针对每个编码参考图像，将一个或更多个第一感知质量与一个或更多个第二感知质量进行比较，其中，一个或更多个第一感知质量与一个或更多个第二感知质量之间的匹配导致一个或更多个编码器设置和与其匹配的渲染质量设置配置文件相关联；以及

以与编码帧基本相同的感知质量生成渲染图像。

22.根据权利要求21所述的计算机实现的方法，其中，所述方法的步骤在渲染器或编解码器处执行。

23.根据权利要求22所述的计算机实现的方法，其中，所述渲染器能够具有用于按像素质量控制的多个设置，所述像素质量控制包括屏幕分辨率、纹理映射选择、细节层次选择、阴影质量和后处理质量。

24.根据权利要求21所述的计算机实现的方法,其中,所述质量设置配置文件定义为每个可用质量设置的值的列表。

25.根据权利要求21所述的计算机实现的方法,还包括优化质量设置配置文件的步骤。

26.根据权利要求25所述的计算机实现的方法,其中,使用决策树以编程方式缩小概率空间来优化质量设置配置文件。

27.根据权利要求21所述的计算机实现的方法,其中,所述质量设置配置文件储存在一个或更多个查找表中。

用于编码器引导的自适应质量渲染的系统和方法

[0001] 相关申请

[0002] 本申请要求于2017年4月21日提交的No.62/488,526和2018年4月5日提交的No.62/653,056的美国临时申请的优先权。

背景技术

[0003] 由客户端玩家控制服务器端游戏的远程游戏应用程序已尝试使用现有或定制的编解码器(也称为编码器)对来自三维(3D)图形引擎的视频输出进行实时编码。然而,视频游戏的交互性质、特别是在视频输出与玩家输入之间的玩家反馈循环使得游戏视频流比传统视频流对延迟更加敏感。现有的视频编码方法可以为减少编码时间而牺牲计算能力,而别无其他。用于将编码过程集成到视频渲染过程中的新方法能够显著减少编码时间,同时还可以降低计算能力、改善编码视频的质量、并保留原始比特流数据格式以保留现有硬件设备的互操作性。

[0004] 当视频游戏实例在玩家本地的硬件上运行时,期望游戏以最高质量输出每个像素。然而,在渲染输出被编码并传输到远程客户端的服务器端游戏实例中,编码器可能会降低图像质量以适合有限的带宽。如果渲染质量显著高于解码输出的质量,则将大量损失服务器端渲染工作。

[0005] 通过基于来自编码器的反馈使服务器端渲染的质量与量化后的质量相匹配,游戏能够减少浪费的服务器端计算而不会有任何明显的客户端质量损失。减少服务器端计算浪费还可以导致额外的益处,包括减少能源消耗、减少渲染时间、以及减少播放器反馈延迟。在多个游戏实例在同一台服务器上运行的环境中,服务器端的计算节省更为复杂。

[0006] 在涉及多个玩家的游戏、特别是诸如大型多人在线游戏(“MMOG”)的游戏的流环境中,确保不浪费服务器端渲染工作变得越来越重要。由于MMOG玩家可用的带宽有限,因此在防止游戏速度下降的同时使渲染质量最大化的编码器特别重要。如下所述,当前的技术采用各种方法来尝试解决该问题,但是仍然不足。

[0007] 公开号US20170132830A1(“'830公开”)的美国专利公开了一种系统及方法,用于在执行着色的3D场景中确定选择着色点、在确定的着色点上执行着色、并且基于在确定的着色点上所执行的着色的结果来确定3D场景的着色信息。基于场景的时间特性调整场景的着色。然而,该技术没有解决基于服务器端渲染功能和可用带宽来优化编码的根本问题。

[0008] 公开号US20170200253A1(“'253公开”)的美国专利公开了用于改善图形处理器的渲染性能的系统和方法。在图形处理器处,能够设置上限阈值,以便在遇到大于设置的阈值的帧时,图形处理器将采取适当的操作以减少渲染时间。然而,该技术仅基于设置的阈值并且无法动态调整以适应服务器端渲染功能和可用带宽。

[0009] 公开号US2017/0278296A1(“'296公开”)的美国专利公开了系统和方法,其中生成确定场景的每个部分处的纹理的场景的初始渲染,并且通过追踪光线的初始样本来生成场景的光线追踪渲染。该参考文献公开了基于场景纹理的预知和识别由于在光线追踪期间欠采样而引起的噪声来智能地确定每个像素的最佳采样数量。再次,该技术受限于最佳光线

采样并且无法动态调整以适应服务器端渲染功能和可用带宽。

[0010] 从对该技术的现有状态的以上讨论中明显看出，本领域中需要改善与游戏的渲染和编码有关的现有计算机技术。

发明内容

[0011] 因此，本发明的目的是公开用于通过以下操作优化渲染的系统和方法：通过使用编解码器(在本文中也可称为编码器)来针对编码器设置的部分范围编码一个或更多个参考图像，并且使用渲染器生成一个或更多个渲染质量设置配置文件、生成一个或更多个参考图像、针对一个或更多个参考图像中的每个，计算感知质量、针对一个或更多个参考图像中的每个，重新渲染一个或更多个参考图像、以及针对一个或更多个重新渲染的参考图像中的每个，计算感知质量。渲染器将参考图像的感知质量与重新渲染图像的感知质量进行比较并且匹配它们。这些匹配结果导致将一个或更多个编码器设置和与其匹配的渲染质量设置配置文件关联到查找表中。查找表用于在游戏期间以与编码帧基本相同的感知质量生成渲染图像。

[0012] 本发明的另一个目的是公开用于通过使用渲染器来计算重新渲染的参考图像中的每个的计算成本来优化编码和渲染的系统和方法。

[0013] 本发明的又一个目的是公开用于通过应用结构相似性指数计算图像的感知质量来优化编码和渲染的系统和方法。

附图说明

[0014] 当结合附图考虑时，通过参考以下详细描述将会更好地理解并容易获得对本发明及其许多伴随优点的更完整的认识，其中：

[0015] 图1是根据本发明实施例的示例性环境的示图，在所述示例性环境中，实时流编解码器能够将设置传达回产生视频的渲染器；

[0016] 图2是根据本发明实施例的概述了编码器引导的自适应质量渲染的示例性阶段的流程图；

[0017] 图3是根据本发明实施例的概述了查找表的示例性预生成的流程图，所述查找表将渲染质量设置配置文件分配给编码器设置的每个部分范围；

[0018] 图4是根据本发明实施例的用于渲染质量设置配置文件的示例性查找表生成的示图，所述渲染质量设置配置文件仅由一个设置组成；以及

[0019] 图5是根据本发明实施例的用于渲染质量设置配置文件的示例性查找表生成的示图，所述渲染质量设置配置文件包含多个设置。

具体实施方式

[0020] 在描述附图中示出的本发明的优选实施例时，为了清楚起见将采用特定术语。然而，本发明不旨在限于如此选择的特定术语，并且应当理解，每个特定术语包括以相似方式操作以完成相似目的的所有技术等同物。为了说明的目的描述了本发明的若干优选实施例，应该理解，本发明可以以未在附图中具体示出的其他形式实施。

[0021] 现代渲染引擎(诸如视频游戏中使用的那些)具有在运行期间基于诸如玩家与目

标的距离、前一帧的渲染时间、或其他运行时测量值等因素来适应某些质量设置的能力。渲染引擎可以提供多种方法来调整质量，从而允许更精细地控制总体渲染质量。一些示例包括：偏置纹理采样以使用模糊纹理映射(Mipmap)；在阴影上使用更低质量的级联或更少的样本；在着色模型上运行简化的路径(例如看起来像漫反射的镜面反射的DCT变换)；以及使用更少的样本进行后处理(例如用于高斯、体积雾等)。在实时流应用中，响应于编码器设置的改变来更改一个或更多个渲染质量设置可以提供最佳的渲染成本节省而不影响编码输出质量。

[0022] 图1是实时渲染视频被实时流传输到远程观看者的示例性环境的示图。服务器100可以是能够同时运行实时渲染过程102(以下也称为渲染器)和流编解码器104的任何硬件。编解码器104还必须具有通过直接报告或本领域已知的一些其他监视过程将其量化设置传达回渲染过程102的能力。编码视频流通过网络传输到客户端设备106。客户端106可以是能够解码和显示视频流的任何硬件。

[0023] 图2是概述了编码器引导的自适应质量渲染的示例性阶段的流程图。使用H.264标准兼容的编码器的实时流编码通常采用恒定速率因子(“CRF”)模式，所述模式在“报告每个编码帧的量化设置，”(步骤200)处将编码帧的有效量化设置报告为量化参数(“QP”)。在某些实施例中，使用的H.264标准兼容库是ffmpeg，其输出量化参数作为变量f_crf_avg。量化参数是从0到51的指数，其定义了编码期间压缩的有损程度。更低的QP值表示更低的压缩，而更高的QP值表示更高的压缩。为了保持恒定的比特率，在CRF模式下运行的编码器将提高帧的QP，这能够提供更高的压缩，并且降低需要更高的质量的帧的QP。编码器利用了这样的事实，即人眼通过在具有相对高运动的区域中增加压缩并且在相对静止的区域中减少压缩而不太能够辨别在移动对象上的细节。这允许编码器在减小一些编码帧的尺寸的同时保持目标感知质量。

[0024] 渲染器在“监视量化设置的改变，”(步骤202)处在渲染帧之前读取报告的QP。在“不同?”(步骤203)处，如果自先前渲染的帧以来有效量化设置没有改变，则渲染器不采取任何操作来适应渲染质量，并且将在下一帧上再次检查。如果渲染器读取的QP值与先前渲染的帧不同，或者如果这是正在执行编码器引导的自适应质量渲染的第一个编码帧，则在“改变渲染质量设置以匹配量化设置”(步骤204)处更改渲染质量。如果自先前渲染的帧以来QP值已提高，则渲染器将降低质量以匹配在编码器处的压缩级别。同样，如果自先前渲染的帧以来QP值已降低，则编码器将提高质量。为了改变渲染设置，渲染器将检查预生成的查找表，所述查找表提供用于编码器提供的QP值的渲染质量设置配置文件。通常，每个编码器质量设置应该仅有一个条目。渲染器使用编码器提供的QP，找到该一个条目，并且使用相关联的渲染质量设置配置文件。通常，应用整个渲染质量设置配置文件。渲染质量设置配置文件限定为每个可用渲染质量设置的值的列表。参考图3更详细地描述了该查找表的预生成。预定义的查找表可以对QP的整数值定义渲染设置，这需要渲染器将读取的QP值四舍五入到最接近的整数，或者查找表可以对在0到51之间的QP值的每个部分范围限定渲染设置。图4和图5中的示例假设渲染器将在使用查找表之前将QP四舍五入到最接近的整数，但是替代地，可以将示例修改成使用QP的部分范围来限定查找表。在渲染下一帧之前，渲染器将根据从查找表获取的渲染质量设置配置文件来更改质量设置。降低渲染质量将减少在编码器出现质量瓶颈时浪费的渲染工作量。

[0025] 图3是概述了查找表的示例性预生成的流程图，所述查找表将渲染质量设置配置文件分配给编码器设置的每个部分范围。参考图像将用作基准，以在改变编码设置或渲染设置时测量对感知质量的影响。参考图像应该代表视频输出的典型帧，并包括所选游戏背景特有的渲染元素，诸如模型、纹理或视觉效果。游戏背景可以包括特定区域、特定地图、特定级别、或一些特定游戏。所选的参考图像将用于生成估算在与参考图像相同的背景内所渲染的视频的感知质量的查找表。例如，由包含来自游戏级别的代表性元素组的参考图像所生成的查找表可以用于估算来自同一级别内的相似场景所渲染的视频的感知质量。下面将进一步讨论将多个查找表组合成通用查找表的方法。在识别了游戏背景之后，应该选择具有代表性的场景并以全质量渲染，如“选择并生成参考图像”(步骤300)处示出的。代表性场景的全质量渲染场景在本文中称为参考图像。

[0026] 上面结合图2的描述讨论的渲染器的运行时行为的优选实施例需要渲染器在读取查找表之前将接收的QP值四舍五入到最接近的整数。因此，将仅使用QP的整数值生成查找表。在编码器处，针对编码器中每个整数值的质量设置(量化参数(QP)整数值0到51)编码全质量参考图像，如在“针对编码器设置的每个部分范围编码参考图像”(步骤302)处示出的。在优选实施例中，有由渲染器执行的四舍五入操作所限定的52个部分范围。能够修改实施方式，以对更常见的QP值(0到51的中间处的值)创建更多的部分范围、或者对更罕见的QP值(0到51的极端处的值)创建更少的部分范围。

[0027] 感知质量是尝试量化人眼感知压缩图像与全质量源图像之间的质量损失的程度。存有多种用于估算感知质量的方法，包括仅使用两个图像之间的亮度和对比度值的差来计算压缩编解码器的质量的均方误差(MSE)和峰值信噪比(PSNR)。如由Z.Wang、A.C.Bovik、H.R.Sheikh以及E.P.Simoncelli在“Image quality assessment:From error visibility to structural similarity,”IEEE Transactions on Image Processing,vol.13,no.4,pp.600-612,Apr.2004所公开的，结构相似性(SSIM)指数是一种方法，它增加了一种假设——即人眼也擅长从场景中提取结构性信息，并定义了一种计算以估算感知的质量。SSIM通过比较两个图像的像素数据:未压缩的全质量参考图像和编码图像之间的像素数据来工作。该算法比较8x8像素“窗口”上的亮度、对比度、结构、以及有时的色度。因为SSIM的计算成本低，并且优于MSE和PSNR等方法，所以其是用于计算感知质量的优选工具。为了对编码器设置的每个值生成感知质量(优选地在渲染器和/或游戏引擎处)，在每个编码参考图像与参考图像之间计算SSIM指数，如“对每个编码参考图像计算感知质量”(步骤304)处示出的。在优选实施例中，对于每个量化参数(QP)整数中的一个，为0到51的值计算52个SSIM值。参考图3、图4以及图5的示例性描述使用标准的SSIM计算来比较两个静止图像，但是存在能够比较两个视频片段并且可以以提高计算成本来替代的SSIM方法的变体。一种这样的SSIM变体是时空SSIM，如Anush K.Moorthy和Alan C.Bovik公开的“Efficient Motion Weighted Spatio-Temporal Video SSIM Index,”Human Vision and Electronic Imaging XV,vol.7527,Mar.2010(从

[0028] http://live.ece.utexas.edu/publications/2010/moorthy_spie_jan10.pdf可获得)。

[0029] 渲染器可以具有可用于每像素质量控制的多个设置，所述设置包括屏幕分辨率、纹理映射(mipmap)选择、多细节层次(LOD)选择、阴影质量、后处理质量、或其他设置。质量

设置配置文件限定为每个可用质量设置的值的列表。在某些实施例中，在渲染器处，收集了能够自适应更改的所有渲染设置的列表以及它们的可能值。然后生成自适应质量渲染设置及其值的所有排列组合，以创建渲染质量设置配置文件的列表，如在“生成渲染质量设置配置文件列表”(步骤306)处示出的。因为渲染器可以是具有许多可能值的许多质量设置，所以质量设置配置文件的排列组合的数量可能会过长。图5的示例讨论了用于限制和优化列表中的质量设置配置文件的数量的示例性方法。

[0030] 对列表中的每个渲染质量设置配置文件，在渲染器处应该使用指定的渲染设置重新渲染的参考图像，如在“对每个渲染质量设置配置文件重新渲染参考图像”(步骤308)处示出的。如果渲染质量设置配置文件由多于一个的设置组成，则对每个重新渲染的参考图像的渲染时间也应该记录为计算成本的测量值，示例性地，以渲染时间或时钟周期测量。如果存在任何SSIM值冲突，则该计算成本的测量值可以在以后步骤中用作决胜点。

[0031] 使用与先前在步骤304中所使用的相同的感知质量的测量值，通过将重新渲染图像中的每个与原始参考图像比较来测量感知质量，如在“针对每个重新渲染的参考图像计算感知质量”(步骤310)处示出的。在优选实施例中，结构相似性指数(SSIM)被用于测量编码器结果的感知质量，并且将被用于测量重新渲染结果的感知质量。

[0032] 在渲染器处，将两个感知质量值组(在步骤304处计算的编码参考图像的SSIM值和在步骤310处计算的每个配置文件重新渲染的参考图像的SSIM值)在两个图像组之间比较，以在两组之间找到匹配的SSIM值。理想地，对于每个编码参考图像的SSIM值，每个配置文件重新渲染的图像组中存在一个精确匹配的SSIM值。如果没有精确匹配，则选择的每个配置文件重新渲染图像的SSIM值应该大于并尽可能接近目标编码参考图像的SSIM值。跨两个感知质量值组的匹配的SSIM值将对QP的每个值标识渲染质量设置配置文件，如在“查找编码器设置的每个部分范围的质量设置配置文件，”(步骤312)处示出的。在发生冲突(从每个配置文件重新渲染图像的SSIM值组中存有两个或更多个精确匹配)的情况下，在步骤308中记录的计算成本可以用作决胜点和为编码器设置选择的成本更低的渲染质量设置配置文件。图5示出了示例冲突。

[0033] 编码器设置和其匹配的渲染质量设置配置文件应该被组织到查找表中，如在“创建对每个编码器设置分配渲染质量设置配置文件的查找表”(步骤314)处示出的。该查找表可以在运行时期间在渲染器处用于改变渲染质量设置，以匹配如图2中的步骤204所描述的量化设置。查找表提供渲染质量设置配置文件，其生成与编码帧具有相同感知质量的图像，并为给定的参考帧提供最大的计算节省。图4和图5中示出了示例查找表。

[0034] 通过结合图3描述的方法生成的查找表可以在类似的游戏背景、场景、或环境中用作参考图像。可以针对多个参考图像重复结合图3概述的过程(每个参考图像代表特定的环境、场景类型、或其他有意义的游戏背景)。例如，可以从游戏中的每个地图中选择参考图像，以生成多个地图指定的查找表。查找表也可以被组合以创建能够在游戏环境中更普遍使用的查找表。例如，地图指定的查找表可以被组合以生成可以用于游戏中的所有地图的一个查找表。为了组合查找表，可以组合每个QP的渲染质量设置配置文件，以找到对包含在配置文件中的每个设置的平均值。例如，对三个参考图像生成三个查找表。渲染质量设置配置文件由三个设置值组成：后处理质量设置、阴影质量设置、以及分辨率设置。为了组合QP值为4的渲染质量设置配置文件，配置文件从每个查找表中读取并被表示为$P4_1=\{3,MED,$

95％}、$P4_2$＝{4,LOW,90％}、以及$P4_3$＝{2,MED,90％}。找到对每个设置的平均值以生成PAvg＝{3,MED,92％}。配置文件平均过程应该四舍五入，以使渲染过程永远不会以低于当前编码质量设置的感知质量级别生成图像。将每个QP值的配置文件取平均值，并组织成一个新的查找表。

[0035] 图4是用于仅由一个设置组成的渲染质量设置配置文件的查找表生成的示例。在该示例中，响应于编码器质量设置中的改变来适应单个渲染质量设置。通过更改视图的3D部分(在“3D视图”400处所示)的分辨率，在渲染器处适应3D场景的第一人视角的渲染，然而为保持任何面向玩家的文本的可读性，用户界面(UI)元素(示出为“UI”402)的分辨率未更改。这种类型的选择性分辨率缩放称为动态分辨率缩放，并且是渲染引擎越来越常见的特征。参考图像(在“参考图像”404处所示)代表来自以尽可能最高的分辨率所渲染的典型视频输出的单个帧，并且根据在图3的步骤300处所概述的准则来选择。在编码器处，对QP的每个整数值编码参考图像(在“参考图像”404处所示)，如结合图3的步骤302描述的，以在“编码参考图像”406处生成编码参考图像的列表。如结合图3的步骤304描述的，在渲染器处，对每个编码参考图像406计算SSIM值(示出为“SSIM”408)。因为渲染质量配置文件仅由一个质量设置组成，所以质量配置文件排列组合的数量受限于可用于3D视图(示出为“3D视图”400)分辨率的可能值的数量。可能分辨率值的数量上限由3D视图的最大可能分辨率限定并且下限由3D视图的最小可行分辨率限定。纵横比可以限定在最小分辨率与最大分辨率之间存在多少分辨率值。例如，最大分辨率3840x 2160具有纵横比16:9，并且在该纵横比中的最小可行分辨率被选为1280x 720。在这些上限和下限之间存在160种具有纵横比为16:9的可能分辨率。可替代地，可以任意选择在上限与下限之间相同分辨率的一些数字作为分辨率样本。例如，可以在3840和1280之间沿x方向逐渐减小分辨率，以选择样本分辨率尺寸的一些数字。

[0036] 在渲染器处，针对可用分辨率尺寸中的每个或所选的样本分辨率尺寸中的每个重新渲染参考图像(如在“重新渲染的参考序列”410处示出的)，如结合图3的步骤308描述的。在渲染器处对每个重新渲染图像计算“SSIM”值(示出为“SSIM”412)，如图3的步骤310描述的。将两个SSIM值组：编码参考图像的SSIM值(如在“SSIM”408处示出的)和每个配置文件重新渲染的参考图像的SSIM值(如在“重新渲染的参考序列”410处示出的)比较来跨图像组找到匹配，以对QP的每个整数值提供分辨率设置。结果被组织成将在运行时期间所使用的如在“查找表”414处所示的查找表。通过降低3D视图分辨率来匹配量化设置，能够显著减少浪费的渲染工作，这可以导致其他益处，包括：服务器上的能源消耗的减少、渲染时间的减少、以及播放器反馈延迟的改善。在多个游戏实例运行在单个服务器上的环境中，这些益处是复合的。

[0037] 图5是用于包含多个设置的渲染质量设置配置文件的查找表生成的示例。结合图3描述的过程对选择参考图像并测量每个编码器设置的感知质量没有改变(如结合步骤300、302和304描述的)。因为渲染器可以缩放与QP值相关的一个或更多个渲染质量设置，所以生成的渲染质量设置配置文件的列表(结合图3中的步骤306描述的)可能太长以至于无法重新渲染参考图像和为每个渲染质量设置配置文件计算感知质量。由于可能存有大量的渲染设置排列组合，所以决策树可以有助于以编程方式缩小概率空间。例如，可能不期望具有后处理质量非常低但是其他每个设置都非常高的渲染质量设置配置文件。在某些实施例中，

可能期望利用低质量的后处理覆盖高质量的阴影。在其他实施例中，可为相反的。这种决策是主观的，但是基于的标准包括但不限于与特定渲染设置相关联的计算成本、在两个设置值之间的感知质量差、一个渲染设置相对于另一个渲染设置的相对明显性(诸如消耗了屏幕大部分的特写效果与仅几个像素宽的远处细节相比)、或相对的游戏重要性(诸如用于传达回玩家的重要的视觉效果)。

[0038] 图5是示例性决策树(如在“决策树”500处所示)，所述决策树由用于四个可能后处理质量设置、三个可能阴影质量设置、以及五个可能3D视图分辨率的每个排列组合的叶子组成。该示例决策树明显小于实际示例，因为可能有更多的自适应渲染设置或每个设置有更多选项，这对本领域任何普通技术人员来说是显而易见的。优选地，根据任何限制条件(诸如在后处理质量非常低而所有其他设置高的地方避开叶子)遍历决策树。对于没有被限制条件去除的每个叶子，参考帧可以利用与叶子相关联的渲染质量设置配置文件重新渲染，如图3中的308描述的。在此处可以记录以渲染时间或时钟周期测量的计算成本，以在感知质量值冲突的情况下用作潜在的决胜点。然后，对每个重新渲染图像可以测量感知质量，如结合图3的步骤310描述的。在对编码器设置计算的组中对于每个计算的感知质量值(SSIM)，可以生成具有匹配的SSIM值的所有渲染质量设置配置文件的列表，如结合图3的步骤312描述的。图5的示例示出了对QP值为16所生成的该列表。

[0039] 利用QP值16编码的参考图像的SSIM值为0.997，为些存在具有匹配的SSIM值的三个渲染质量设置配置文件，示出计算成本为16.004、15.554以及15.402。由于对于感知的质量值存在三个冲突，所以较早记录的计算成本用作决胜点并且可以用于确定哪个渲染质量设置配置文件最便宜，在该情况中是成本为15.402的那个。应该生成查找表(如在“查找表”502处所示)以将最便宜的渲染质量设置配置文件分配给QP的每个值，如图3中步骤314描述的。对QP值16选择的渲染质量设置配置文件在图5中示出为“配置文件16”。

[0040] 示例1:将渲染时间作为计算浪费的代理的影响

[0041] 在示例实施方式中，响应于编码器质量的改变，仅分辨率被线性地缩放。例如，如果编码器质量下降50％，分辨率将降低50％作为响应。因为渲染时间节省与计算能力节省直接互相关，所以在缩放分辨率的同时检查渲染时间。测量值是在包括玩家的手、武器、以及固定墙壁的第一人视角的视角的低运动环境下取得的。选择该低运动视角是为了限制可能影响测量结果的渲染时间而污染测量结果的因素的数量。这些因素可以包括诸如运动模糊的后处理、渲染对象数量的改变、屏幕上纹理的改变、或在高运动视角中可能改变的视图的其他要素。固定场景的固定视角还可以直接比较以缩放的分辨率取得的各种测量值。渲染引擎被强制以逐渐降低的分辨率输出视频，并且测量结果如下面表1所示。

[0042]

分辨率缩放	不透明处理时间	总渲染时间
100%	0.4 ms	1.4 ms
50%	0.3 ms	1.0 ms
25%	0.2 ms	0.8 ms

[0043] 表1:分辨率缩放对渲染时间的影响

[0044] 不透明处理(opaque pass)是用于在视图中绘制不透明几何图形的渲染管线的部

分。这是渲染管线的对分辨率改变最敏感的部分。通过缩放分辨率获得的任何渲染时间节省或计算成本节省将大部分来自不透明渲染处理。

[0045] 如表1所示，在60帧的全分辨率1280x 720下，总渲染时间1.4ms，其中不透明处理的渲染时间为0.4ms。当分辨率降低到全分辨率的50％时，总渲染时间为1.0ms，其中不透明处理的渲染时间为0.3ms。因此，将分辨率缩小50％导致渲染时间显著节省约30％。当分辨率降低到全分辨率的25％时，总渲染时间为0.8ms，其中不透明处理的渲染时间为0.2ms。因此，将分辨率缩小75％导致渲染时间显著节省40％以上。

[0046] 前述描述和附图应被认为仅是对本发明原理的说明。本发明不旨在受限于优选实施例并且可以以对本领域普通技术人员显而易见的各种方式来实施。本领域技术人员将容易想到本发明的许多应用。因此，不需将本发明受限于所公开的特定示例或所示出和描述的确切构造和操作。相反，在本发明的范围内，可以采用所有合适的修改和等同物。

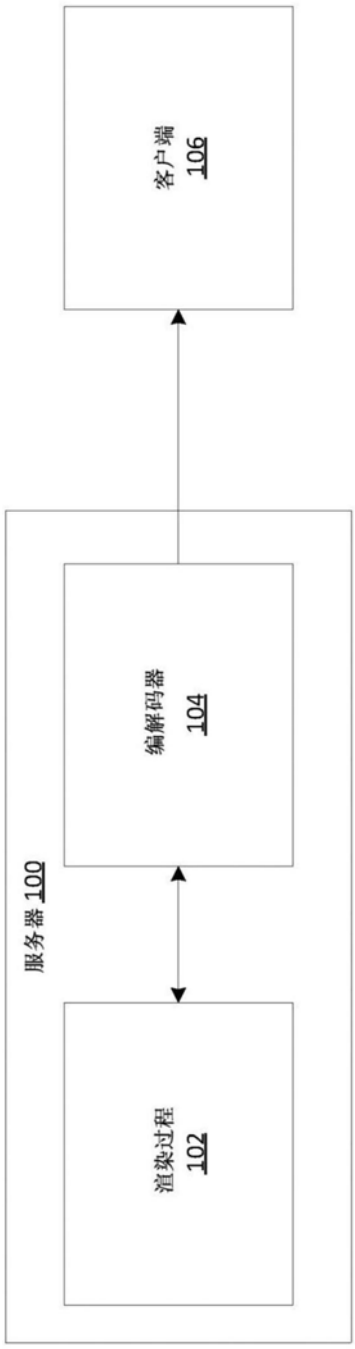


图1

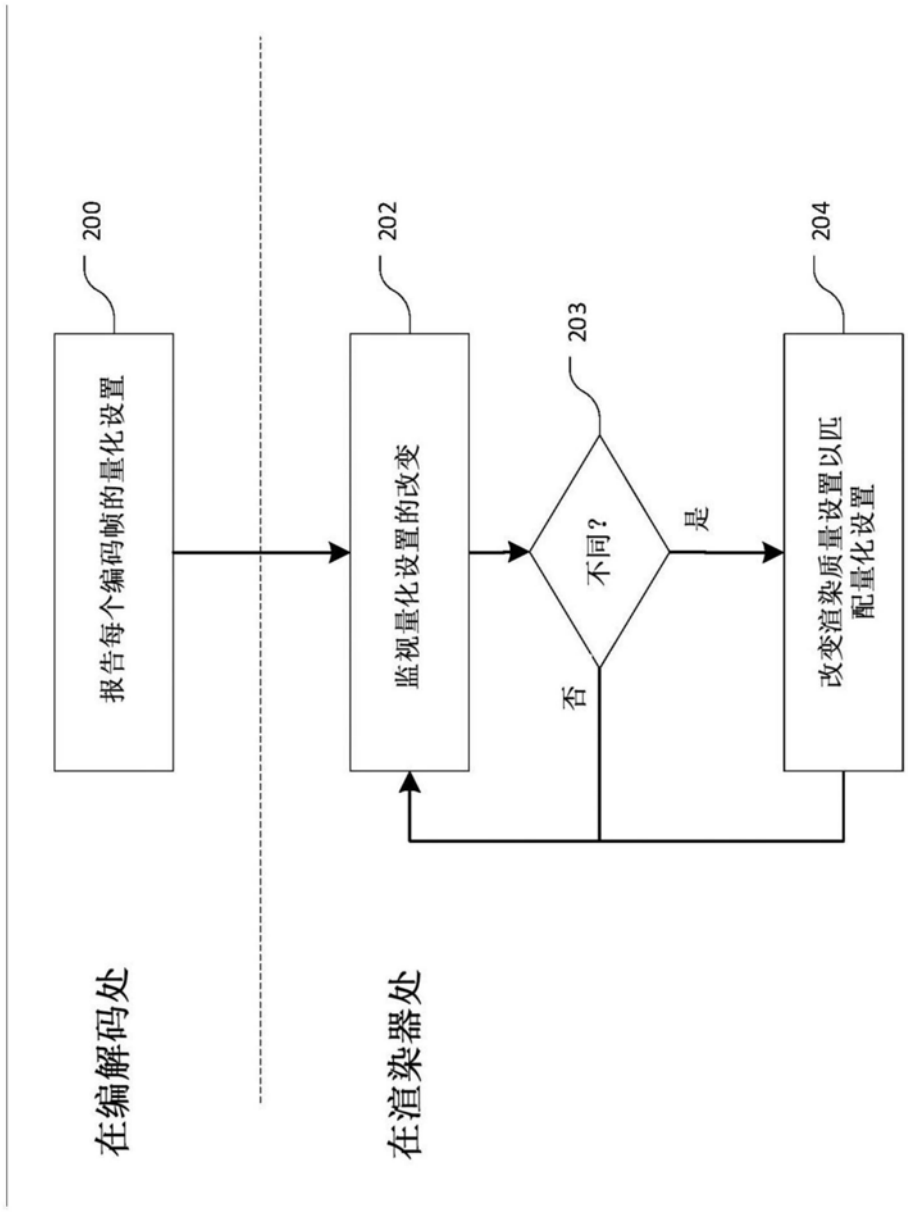

200
报告每个编码帧的量化设置
202
监视量化设置的改变
203
不同?
否
是
204
改变渲染质量设置以匹配量化设置
在编解码处
在渲染器处

图2

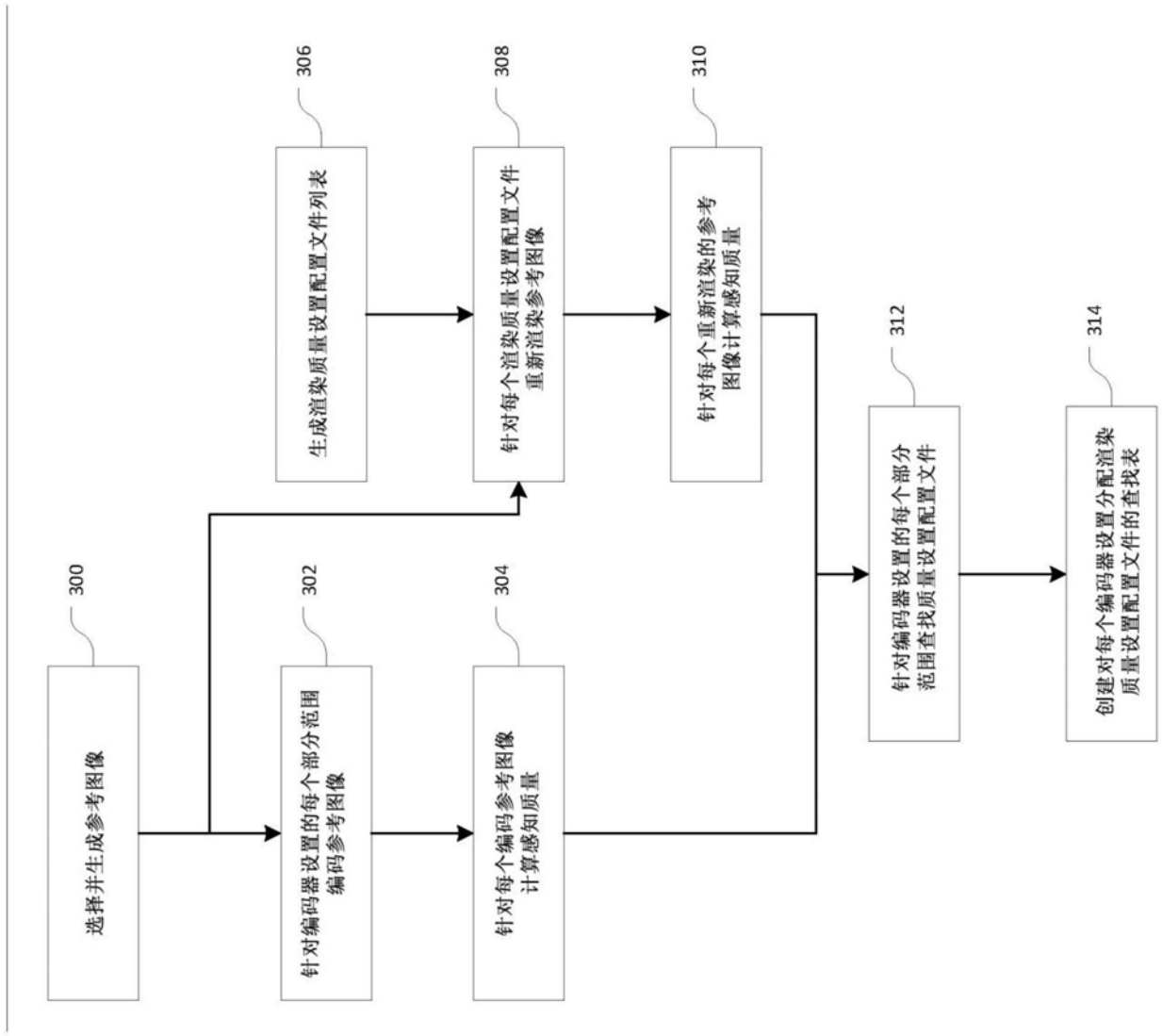

300
选择并生成参考图像
302
针对编码器设置的每个部分范围编码参考图像
304
针对每个编码参考图像计算感知质量
306
生成渲染质量设置配置文件列表
308
针对每个渲染质量设置配置文件重新渲染参考图像
310
针对每个重新渲染的参考图像计算感知质量
312
针对编码器设置的每个部分范围查找质量设置配置文件
314
创建对每个编码器设置分配渲染质量设置配置文件的查找表

图3

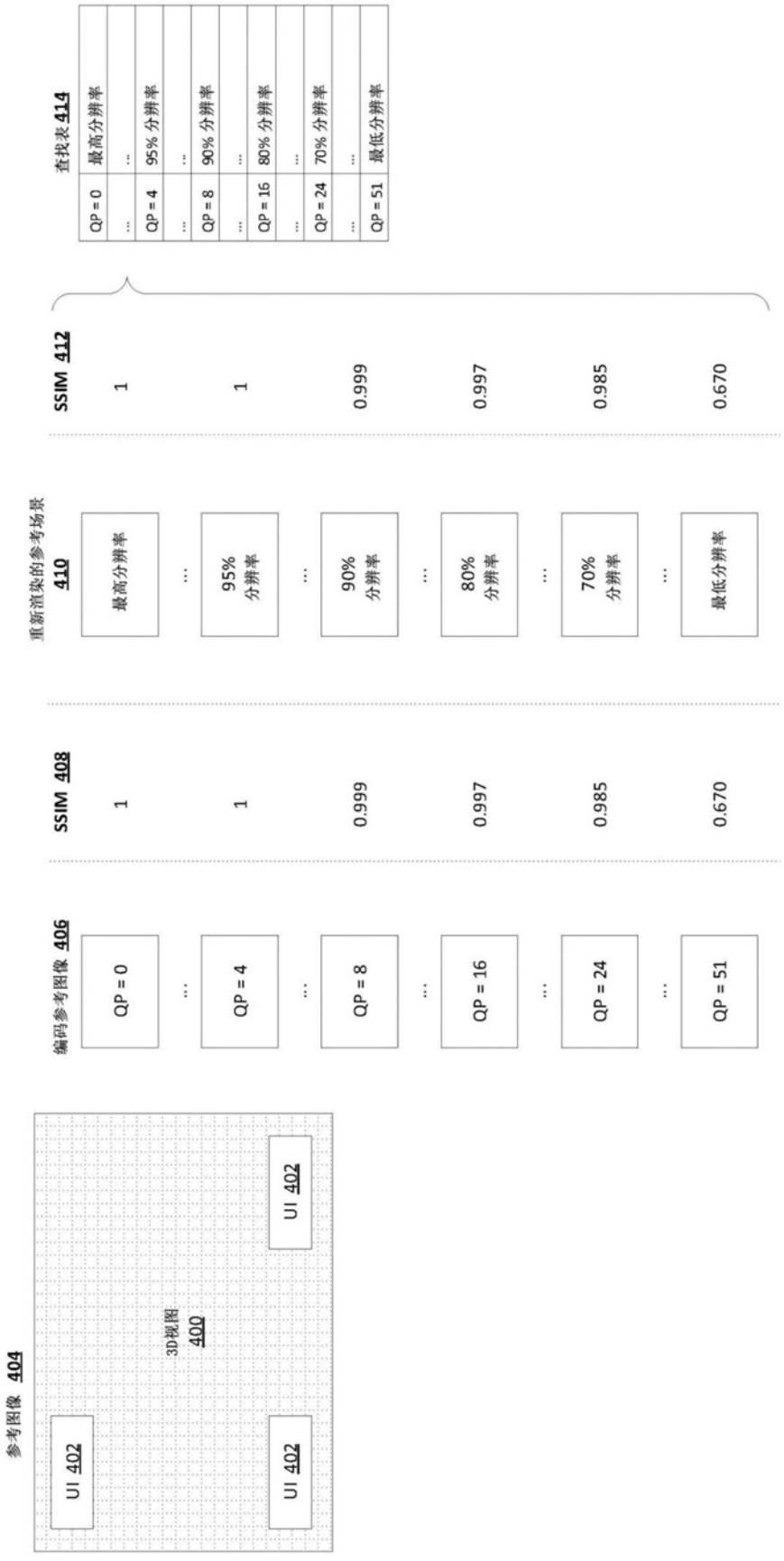

参考图像 404
UI 402
UI 402
UI 402
3D视图 400
编码参考图像 406
QP = 0
QP = 4
QP = 8
QP = 16
QP = 24
QP = 51
SSIM 408
1
1
0.999
0.997
0.985
0.670
重新渲染的参考场景 410
最高分辨率
95% 分辨率
90% 分辨率
80% 分辨率
70% 分辨率
最低分辨率
SSIM 412
1
1
0.999
0.997
0.985
0.670
查找表 414
QP = 0 最高分辨率
QP = 4 95% 分辨率
QP = 8 90% 分辨率
QP = 16 80% 分辨率
QP = 24 70% 分辨率
QP = 51 最低分辨率

图4

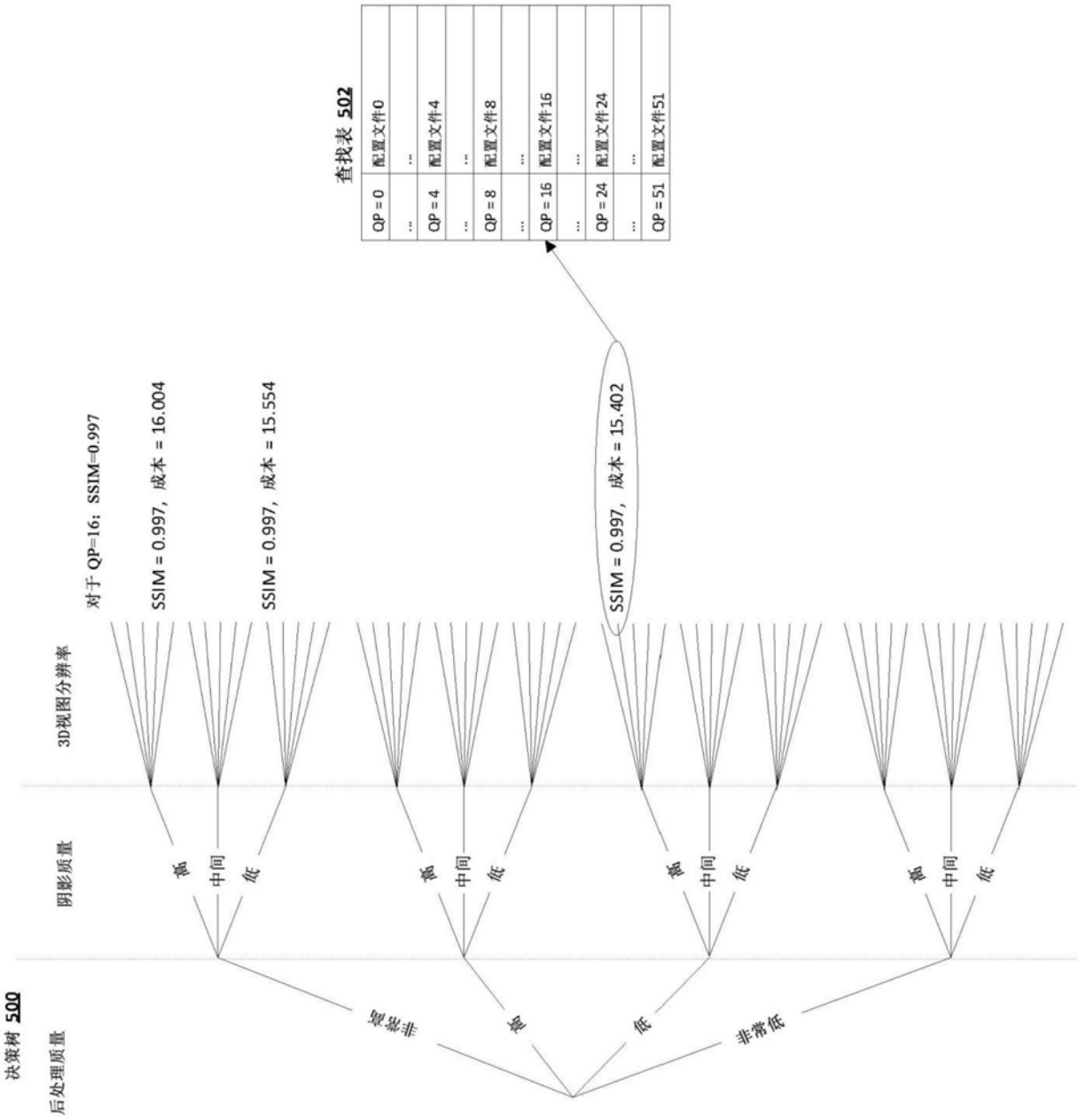

查找表 502
QP = 0
配置文件0
QP = 4
配置文件4
QP = 8
配置文件8
QP = 16
配置文件16
QP = 24
配置文件24
QP = 51
配置文件51
对于 QP=16；SSIM=0.997
SSIM = 0.997，成本 = 16.004
SSIM = 0.997，成本 = 15.554
SSIM = 0.997，成本 = 15.402
3D视图分辨率
阴影质量
后处理质量
决策树 500
高
中间
低
非常高
非常低

图5