

FAVR - Accelerating Direct Volume Rendering for Virtual Reality Systems

Dezeming Family

2023 年 7 月 10 日

正常字体：表示论文的基本内容解释。

粗体：表示需要特别注意的内容。

红色字体：表示容易理解错误或者混淆的内容。

蓝色字体：表示额外增加的一些注释。

绿色字体：表示额外举的一些例子。

目录

一 引文	1
二 技术	1
2 1 管线的描述	1
2 2 纹理查找	1
2 3 整个渲染管线	3
参考文献	3

一 引言

近年来,使用坚固且价格合理的头戴式显示器(HMD)系统,虚拟现实(VR)已经变得容易获得。最近提出了几种基于虚拟现实的科学可视化解决方案,但直接体绘制(DVR)仅在少数虚拟现实应用中被考虑。我们将此归因于 DVR 的高计算需求和 VR 系统可用的有限渲染预算对于 DVR 这样的高度碎片绑定(fragment-bound)的方法来说,实现 VR 所必需的非常高的更新率是具有挑战性的。我们提出了一种加速技术,旨在利用 HMD 系统的特定特性。我们使用了一种基于 HMD 的镜头失真和人眼视觉感知的自适应渲染方法。我们的实现降低了 DVR 的渲染成本,同时提供了与标准渲染技术无法区分的体验。

运动自由度以高更新率和低响应时间的必要性的形式给 VR 系统带来了限制,没有达到这些强制性限制会导致疲劳和恶心,也称为 cybersickness。VR 设备的高分辨率显示器对于在眼睛附近呈现清晰的图像和减少 screendoor-effects 效果的出现是必要的。直接体绘制与虚拟现实相结合的挑战是必要的高刷新率。实现高更新率的一种常见方法是渐进渲染方法,该方法在交互过程中渲染快速预览,并在交互停止后细化图像。然而,在虚拟现实中,交互从未停止,这些技术并不直接适用于头部跟踪环境。

因此,我们提出了一种渲染方法,该方法可以减少 fragment 工作量,同时立即产生高质量的结果。为了实现这一点,我们根据 HMD 透镜和人类视觉系统的非均匀特性调整了光线投射器采样率。Scholl 等人也提出了直接加速技术不足以用于 VR 设置的认识。他们还提出了一种在 HMD 上直接体绘制系统的解决方案,并认为 90Hz 是适合 VR 应用的最小更新率。由于 DVR 的复杂性,他们不得不将光线遍历过程中的采样率限制为沿单个光线的 150 个样本,从而限制了整体图像质量。他们还限制在比我们发现的日常实践中常用的数据集更小的数据集内,以保持必要的刷新率。Usher 等人专注于利用 VR 中的直接体绘制进行神经元跟踪的可视化工具,虽然在沉浸式环境中提供了一个有用的工具,但他们只能实时可视化原始体积数据的小块,以保持足够的更新率。Meng 等人提出了另一种加速 HMD 渲染的方法,该方法降低了 non-dominant 眼的实际渲染分辨率。

为了加速 DVR 实现最先进的消费级 VR,我们引入了聚焦自适应体渲染(Focus Adaptive Volume Rendering, FAVR)。我们的方法有效地减少了渲染过程中处理的 fragment 的数量,而不会感知到图像质量的损失。我们调整了人类视觉系统的周边视觉中的渲染分辨率,这对用户来说并不明显,并利用了 HMD 提供的固定镜头设置。与我们的方法类似,已经为桌面应用程序开发了其他解决方案,并表明渲染性能大幅提高。应用我们的技术,我们可以提供一个远远超过 VR 要求的更新率的交互式系统。

二 技术

2.1 管线的描述

我们的管线是延迟渲染(deferred rendering),论文来自于 Acceleration techniques for gpu-based volume rendering,包括早期光线终止和空域跳跃技术。我们的方法在计算密集的光线遍历之前添加了一个子采样阶段,以及最后添加了最终重建。这个子采样阶段是我们加速的基本 fragment 减少部分,使我们能够实现 HMD 设备所需的刷新率。

在第一步中,基于 GPU 的光线投射器以进入和退出缓冲区的形式可视化数据的边界几何图形。然后,这些缓冲区用于遍历实际数据。就 HMD 而言,我们必须考虑本机显示分辨率的帧缓冲区,以获得尽可能好的质量,这将严重降低系统的整体性能。由于 HMD 内置透镜,在整个像素密度中我们只能感知显示的中心部分。我们可以利用这一特性,在不引人注目的情况下大幅降低用户周边视觉的实际像素覆盖率。

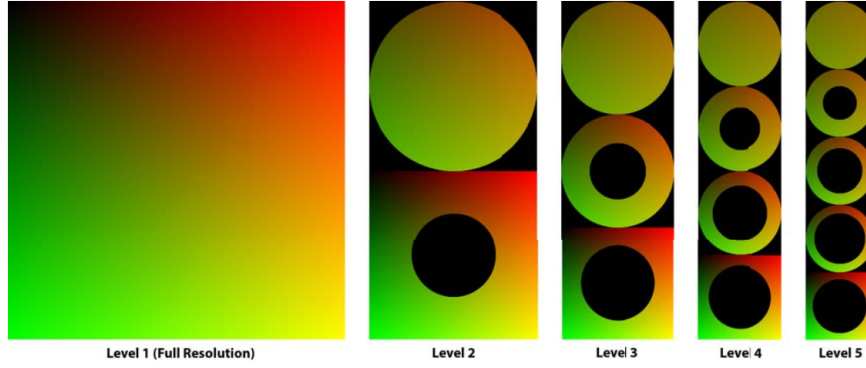
因此,我们预先计算查找纹理,以减少和重建全分辨率入口和出口缓冲区。我们将在下一节中介绍这些缓冲器的构造和使用。我们的系统可以在无需用户任何交互的情况下实时完全自动调整子采样级别,并为 VR 环境保持稳定的刷新率。

2.2 纹理查找

这里我先分开描述,再讲述整个管线。

我们的系统在启动时预先计算多个子采样和重建纹理。这些子采样（下图第一张）和重建纹理（下图第二张）表示查找表，将来自边界几何体的进入和退出缓冲区的 fragment 映射到我们的光线投射器输入的目标缓冲区。

预计算纹理的数量，也称为子采样级别，可以由用户指定。在运行时，我们的系统评估先前帧的计算时间，并在不同级别之间动态切换，以保持 90Hz 以上的刷新率。



为了减少 fragment，我们为子采样阶段预先计算多个查找纹理。这些用于生成外围视觉中信息密度降低的入口和出口缓冲区。图中的子采样纹理代表了我们的加速方法的五个级别（注意，每次渲染时我们只会使用其中的一个级别，这会根据你的实际渲染速度来动态选取）。在 1 级，我们的系统将原始全分辨率缓冲区复制到射线投射器目标中，减少 0 个碎片。

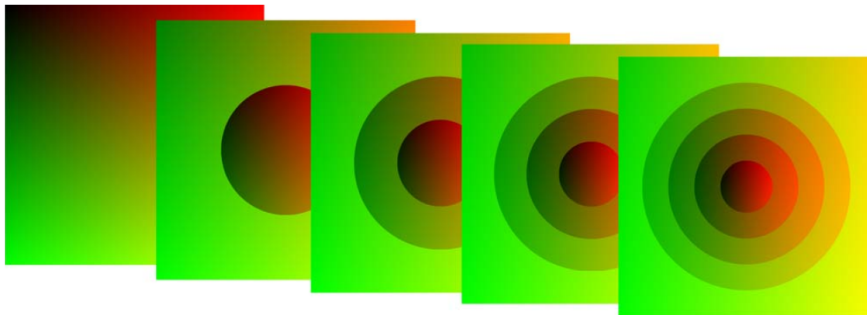
从第 2 层开始，系统将此处所示的圆形结构引入管线，这些结构是基于在 HMD 透镜上发现的圆形形式。在我们的系统中，我们将这些结构称为 stages，每个纹理本身都是子采样级别。对于每个级别 l ，我们在纹理中都有相同数量的 stages（比如第 4 级有 4 个 stages）。级别 1 由一个 stage 组成，即原始输入纹理的直接查找纹理。第 2 级引入两个 stages，每个子采样纹理的宽度是： $\frac{1}{l} \cdot w$ ，其中， w 表示输入纹理的最初宽度。在我们的实现中，纹理的高也是一样的。

从顶部以 0 开始的每个连续阶段 s 表示输入缓冲的缩放部分。每个 stage 的分辨率按比例缩放：

$$1.0 - \frac{s}{l} \quad (二.1)$$

总 fragment 数就会变为 $\frac{w \cdot h}{l}$ ，对于每个子采样层，由于这些纹理中的圆形结构和空空间，光线遍历过程中评估的碎片数量甚至进一步减少。

我们评估了每个层的剩余片段的百分比，发现二级子采样已经消除了 60% 以上的片段，三级子采样大约消除了 80%。在评估光线遍历后，使用子采样的入口和出口缓冲区必须重新创建原始大小的图像并将其呈现给用户。

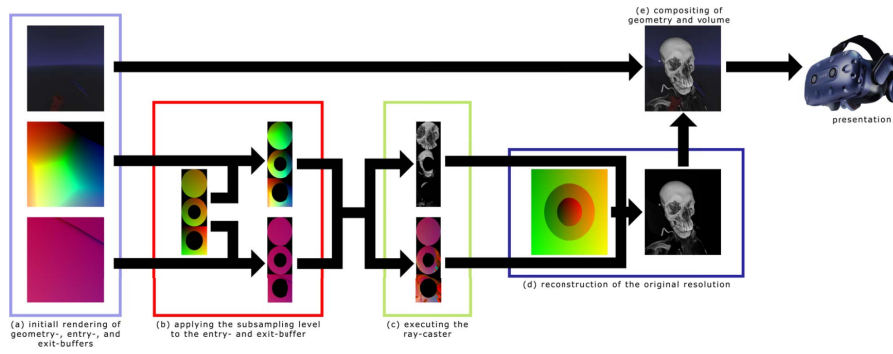


上图是我们为最终图像的重建准备的查找纹理。这些纹理的每个数据点都对应于我们的源子采样纹理的位置。

因此，我们的系统计算重建纹理，其将二次采样的射线投射器图像的纹素映射到全尺寸的帧缓冲区。光线投射器的最终图像部分渲染的分辨率低于我们的最终目标图像，因此必须进行插值以填充缺失的数据。在我们的例子中，我们使用双线性插值，使图像的外围区域变得模糊。此外，为了确保用户看不到明显的间隙，并平滑阶段之间的过渡，我们在 stage 之间添加了一个小的重叠，用于在阶段之间混合并去除可见边缘。

2.3 整个渲染管线

这个流程管线描述的是三级分辨率，最中央的部分是全分辨率渲染，周边区域是降低了分辨率渲染的结果，然后最终将渲染结果合并：



参考文献

- [1] Waschke A, Krüger J. Favr-accelerating direct volume rendering for virtual reality systems[C]//2020 IEEE Visualization Conference (VIS). IEEE, 2020: 106-110.