Lightweight Key Technologies for Massive Multi-player Online

WebVR Conferencing

单位信息：

作者信息：

摘要

关键字

1. **引言**

梁栋补充下

1. **相关研究工作**

梁栋补充下

1. **总体技术路线**

志成和恩旸在这里补充一个技术路线图（突出几大关键技术），尝试画一下（用ppt来画图），大家一起来补充完善。

轻量化处理的核心是资源重用，我们使用该技术必须首先明白哪些资源是可重用的，并且尽可以提高资源的重用度。另外为了效果要保留丰富的多样性，参数化调整每个对象，另外可以通过对不同资源进行搭配组合来提高多样性。

首先将模型划分为多个区域（在这个问题中是划分了头部、上身、下身三个区域），每个区域可以匹配不同贴图（在这个问题中使用了男性32套贴图，女性16套贴图），这样就可以通过不同贴图搭配各种效果（男性有32^3=32768种组合，女性有16^3=4086种组合，共36864种组合）。另外，可以通过对人物对象的高矮胖瘦进行设置，还可以对对象的色调进行编辑（这个问题中主要是对下身的裤子颜色进行设置），在骨骼动画方面，可以为每个人物对象设置不同的动画播放速度。

1. **关键技术之一：大规模会议情境的轻量化预处理**

志成写一段引文

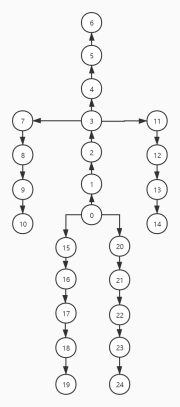
4.1**静态会场场景的轻量化预处理**

志成完善下列文字

大规模人群渲染需要存储的信息可以分为三级。第一级，所有对象都共用的数据，如模型的网格信息（网格点位置，UV等信息）；第二级，选择性使用的数据（如纹理贴图），这类数据每个对象只需要其中的一部分数据（如模型贴图有多套可供选择，但每个对象只使用其中的一套）；第二级，描述对象的参数信息（如，贴图类型，动画播放速度，高矮胖瘦，色调），这些信息每个对象都可以不同，所以每个对象的这些参数信息都要单独存放。

4.2**大规模参会人群的轻量化预处理（参考朴雪论文中相关内容）**

志成/恩旸完善这部分文字

在项目中我们只需要实现听众鼓掌的动作，这个动作只涉及到手臂的8个骨骼，其它的17个骨骼的状态没有发生变化，所以首先可以将骨骼数据分成两部分，手臂处骨骼和固定动作的骨骼。

每个骨骼的变换矩阵因为最后一行是固定值，所以一个变换矩阵只需要存储12个float数据。有17个骨骼在动画播放的过程中的状态没有发生变化，所以这可以提前计算好这17\*12=204个数据。涉及到手臂的8个骨骼，由于我们在项目中只需要实现鼓掌这个简单的动作，经过测试使用16帧就可以获得比较好的动画效果，这是16帧的鼓掌动画中后8帧可以看作前8帧的倒放，所以我们只需要8个骨骼8帧分别的变换矩阵，这8个骨骼需要的数据量为8\*8\*12=768。将这些数据在预处理阶段直接计算好可以减少在客户端的计算量。

1. **关键技术之二：细粒度化渐进式传输调度（参考李柯论文相关内容）**

5.1轻量化缓存管理

**志成这一块**

将场景中需要的数据细化拆分，相同部件只传输一次。

分级传输，提高初始加载速度。人物贴图进行了多细节层次处理，先传输低像素的纹理贴图，再传输高像素的贴图。

5.2基于兴趣度的细粒度化在线打包

**志成写这一块**

5.3 带宽自适应的渐进式传输调度

志成写这一块

为了确保复用的人物对象资源不被重复存储，需要将场景中的人群统一管理，我们用JS设计了以一个对象来管理整个的人群。

为了提高渲染效率，所有的人物在渲染时作为一个整体，一次性将人群所需的全部数据输入着色器。这就需要我们管理好每个人的所有参数信息（位置，大小，贴图类型，等等），渲染时一次性将这些信息传入shader。

**6 轻量级大规模会议场景在线渲染**

志成完成这一块

骨骼动画数据的输入格式是32的float浮点数，经过测试发现骨骼动画数据实际上并不需要这么高的精度，所以我们设计了一种16位的浮点数格式，这样就使得传入着色器的骨骼数据量减少了一半。

项目中人物的贴图左右对称，所以可以只传入着色器左半部分贴图，这样传入着色器的贴图数据量就减少了一半。

5.实验结果与性能分析

志成/恩旸搞这一块

1. 结论

志成/恩旸搞这一块

1. Reference

梁栋补充这一块