**Lightweight Key Technologies for Massive Multi-player Online**

**WebVR Conferencing**

单位信息：同济大学-智慧Web3D实验室 吉林动画学院-智慧虚拟现实研究院

Smart Web3D Media Lab, Tongji University

Smart VR R&D Center, Jilin Animation Institute

作者信息：根据大家的贡献度来定

ABSTRACT

WebVR与在线会议是提当下高远程办公效率的必然途径，然而大规模多人移动在线WebVR会议仍然存在多个技术瓶颈。针对这些瓶颈，本文提出了一个系统性的方法，该方法包括：1) 服务端3D模型的轻量化；2)渐进式传输调度；3) 客户端轻量级在线渲染。服务端采用静态3D会场的轻量级细粒度化处理与流式化处理、分级重用与参数化差异处理、参会者虚拟化及动作行为的轻量级建模等方法。传输端采用了基于兴趣度的大规模静态与动态会议场景的渐进式传输调度机制。针对频繁的客户端请求，我们通过对Shader的高效调度，将部分渲染计算迁移到GPU，并合理避免了CPU，GPU以及内存之间频繁的信息交换。基于本文提出的方法，我们实现了大规模WebVR在线会议的轻量性、多样性、真实性与高效性。根据数千人的WebVR在线会议实践验证了我们所提出方法的可行性与先进性。 静态场景经过轻量化处理后，在不改变模型的精度、不损失细节的情况下，将整体数据量减少了97.6%。动态资源经过轻量化处理后，在保证场景帧数不低于30帧的情况下，在台式机中可容纳的最大人群规模提高为原来的180.19倍，在笔记本电脑中可容纳的最大人群规模提高为原来的12倍。

KEYWORDS

WebVR，虚拟会议，大规模群体参会行为建模，轻量化预处理，渐进式传输调度，轻量级在线渲染

1. INTRODUCTION

新冠疫情在多个维度上改变了人们生活、工作、学习以及协作的模式，传统的线下行需要迁移到线上环境中。在疫情的影响之下，远程在线会议的需求激增，例如ZOOM会议、腾讯会议、钉钉会议等在线会议平台的使用量增幅巨大，但是目前的在线会议系统具有一下几点限制：

1.【现场感差】当前的主流在线会议系统不具有3D可视化的能力。传统的音视频的方式无法充分还原会议的真实场景，缺少VR那样的代入感较和沉浸感。

2.【便捷性差】当前会议系统的便利性及用户友好性较差。用户需额外安装插件。目前大多用户都不喜欢安装插件或APP，而是喜欢Down-and-Play。大多数VR会议系统较为重量级，需要用户在进行虚拟参会前安装PC客户端或插件，给手机参会用户或电脑配置较低的用户带来了不便，与在线会议“随时随地，即时上会，快速开会”的目标相背离。

3.【规模性太小】大规模多人在线参会已经逐渐成为趋势，近年来线上会议与会人数已达千人级别，传统的小规模在线会议系统已经不能满足当前的需求。但是，由于浏览器渲染能力较弱，VR会议系统若不采用优化算法，在线参会人数将会非常受限，无法支持大规模多人在线参会。目前大多数在线会议系统对用户设备及网络环境要求高，未采用轻量化技术，会议情景需要加载的模型及动画数据量较大，导致会议加载速度慢，响应不及时等。

基于后疫情时代下的WebVR在线会议需求，想要解决当前远程会议系统存在的上述问题，主要有以下几个技术瓶颈：

1. **会议情景的3D模型数据量太大**。为满足用户对在线会议的真实性、便利性等需求，给用户的提供较为良好的与会体验，势必需要大量精细的、高数据量的会议场景模型、人物与场景动画等等，并且数据量会随着用户对场景美观度、交互方式的丰富度等的提高而进一步提高。
2. **对于大型会议情景来说网速与带宽严重不足**。对于大型web端3D会议来说，即使是使用5G网速和带宽也略有不足，尤其是无线传输。由于美观的场景模型、精细的人物动作带来的模型数据量的问题，会导致这些模型的传输出现比较大的网络延迟，从而对用户的与会体验造成负面影响。
3. **网页端在线渲染太卡顿**。网页浏览器相对于桌面应用来说，受制于有限的缓存空间与计算能力，不仅要渲染大规模、高数据量的会议场景、人物模型及动画等，还要时刻对用户的交互请求做出及时的响应。大规模WebVR会议对渲染速度、渲染精度、资源占用等都有着极高的要求。

本文针对这些挑战进行了长期有针对性的研究。首先，本文将服务端3D模型的轻量化、渐进式传输调度、高效shader调度应用于静态3D会场轻量化处理、大规模会议场景传输、客户端在线渲染中。同时采用虚拟现实的方式进行远程会议环境的呈现，给体验者带来更强的参与感与沉浸感，使得线上与线下的距离进一步拉近，采用Web+VR的方式进一步减低VR的体验与使用门槛，用户无需安装任何插件，即可快速上会，并且支持大规模的人群在线参会，同时给用户提供丰富的、友好的、快速的与会交互操作，提升用户使用的友好度与易用性。

1. RELATED WORK

Web-based virtual reality (WebVR) and online conference is an active research area during the past decades, with the purpose to move the entire vision of physical world online that enable reviewers and individuals to create/edit immersive and interactive VR applications in real-time without extensive efforts. Recently, Nguyen and Dang proposed a framework that utilize VR and AR to present 3D objects for online learning environment with Unity [20, 21]. Miyata et al. introduced an educational framework for students to collaboratively design VR applications [22]. The approach of [23] used a three-year industrial VR projects to study the key factors that contribute to the failure/success of VR products. WebVR and Web conferences have been used to explore alternatives for microteaching in rural regions [24].

Unlike these approaches which focus on content visualization and online education for limited group size, our approach provides a systematic framework could offer a cost-effective and efficient VR interaction for large scale users by integrating lightweight 3D meshes modeling, progressive transmission, and real-time online rendering with Unity. Teh S et al. developed a virtual conference management system [26]. Nguyen and Dang [21] also used Unity for setting up 3D VR and AR environments, unlike our approach, their methods do not have the lightweight modeling and progressive transmission modules. By contrast, online conference system could handle similar users do not have VR features.

**2.1 虚拟在线会议系统的现状**

当下，对VR在线会议系统的研究主要分为两大类。第一类是对基于Web端的在线社交会议平台的研究，如Spatial[5]。第二类是对基于PC客户端的VR会议平台的研究，如VSWork[1]、Engage[2]等。第三类是结合了前两者的WebVR会议平台，如Mozilia hubs[3]。

Web在线会议系统比较常见的有Spatial，Web在线会议系统能够容纳大规模的人群，但是其真实性和多样性较VR会议来说极差，会议没有带入感，难以达到真实线下会议的效果。

VR虚拟会议系统比较常见的有VSWork和Engage，VSWork是国内的一款VR虚拟会议系统，其实现了丰富的用户交互操作，如自定义虚拟角色，人物模型与场景物体的互动等。在软件架构方面，VSWORK的全部内容都采用云端部署，终端设备的本地负担很小，不同客户的内容都对应独立的服务器，保证了用户文件与信息的安全性与私密性。Engage由VR教育公司IVRE开发，同样支持与会者一对一的丰富交互操作，同时，其对VR设备也有着较好的支持。基于PC的VR在线会议系统，大多是采用Unity进行开发，较为重量级；发布模式仍是PC端运行的可执行程序，用户使用门槛较高。

Web3D虚拟会议平台比较常见的有Mozilla hubs，Mozilla hubs是运行在浏览器中的虚拟协作平台。其技术优势在于无需安装其他插件，通过链接邀请他人进入自创的虚拟空间中进行相关的交互操作，并提供对VR设备的支持。并且，其拥有着高自由度的化身定制、场景定制功能，以及高自由度的交互方式。不过在部分场景下，场景的美观度、加载速度上仍然较慢。支持有限度的互动与远程协助即使是目前像Mozilla hubs这样的WebVR会议平台，虽然有着更加轻量级、用户交互操作更多等特点，但是在某些场景的美观性和精度不足、加载速度上较慢，卡顿较为频繁，仍然不足以为用户提供一个较为良好的与会体验。

表 1 各种会议系统间的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **轻量性** | **多样性** | **真实性** | **规模性** |
| Spatial | 轻 | 差 | 差 | 大 |
| Engage | 重 | 优 | 优 | 中 |
| VSWork | 重 | 优 | 中 | 中 |
| Mozilia hubs | 轻 | 中 | 中 | 小 |

**2.2 大型会议静态场景的轻量化处理**

大规模在线会议由于参会人数众多，会议场景、人物模型数据量非常庞大，巨量数据会对网络条件、用户配置及浏览器渲染速度造成非常大的压力。因此，降低会议情景中的数据量方法就变得十分必要。刘[11]等人提出了一种基于WebBIM场景的数据轻量化方法。

由于模型组件之间的相似性，我们可以进一步利用它来提高效率。模型的预处理当然可以缓解问题，例如3D压缩和渐进式网格（PM）。但是他们都没有考虑模型组件之间的相似性，因此我们可以利用它来进一步提高效率。Wen[12]等人提出了一种类似感知的数据约简方法，称为轻量级渐进网格（LPM）。 LPM旨在挖掘模型中的相似组件，在删除冗余组件后生成剩余每个组件的PM表示，并使用称为轻量级场景图的结构组织所有处理后的数据。这种方法具有四个显着优势。首先，它可以极大地减小模型的文件大小，而几乎不会造成任何精度损失。其次，PM使传递变得渐进，即所谓的流式传输。第三，在客户端进行渲染时，由于使用了轻量级的场景图，因此无需进行解压缩，并且可以充分执行实例渲染。第四，它在非常有限的带宽下非常有效，尤其是在交付大型场景时。

**2.3 大规模人群在线可视化处理**

当下，大规模人群的可视化技术通常是通过基于代理的模拟技术实现[13]。使用单一模型的缺点是加载时间可能很长，一旦人物数量增加，帧率就会显著下降。之前的方法着重于简化模型本身或提高硬件的性能。通常，这些方法会忽略大规模人群仿真的多样性和真实性。并且，GPU实例机制和LOD（详细级别）技术已经可以用于实现数千个化身的实时渲染，但模拟化身是同质的[15]。大多数研究都没有考虑多样性和现实因素。因此，一种在保证多样性和真实性要求的情况下，进一步扩大人群规模来灵活提高效率的方法是非常有必要的。Daniel[16]等人提出了一种基于形状空间的参数化技术来对化身的外观进行多样化。同时使用异步传输渲染元素，助于减少带宽压力。其提出的多级克隆实例化方法可以在短时间内生成大量的异构化身。

由于在线会议情景需传输大量模型及动画数据，为了避免出现比较大的网络延迟，减轻对用户的与会体验所造成的负面影响。有学者尝试使用边缘计算方式来代替集中式云计算，如李[17]等人提出的一种新的云端边缘浏览器计算框架与大型场景数据自适应传输调度算法。该算法通过综合用户视角的重复度、填充程度与所注意的场景来定义兴趣度，基于计算出的用户兴趣来优化传输机制、进行路径预测，以实现网络传输的最佳效果。与云计算的集中式结构不同，边缘计算采用分布式结构[17][18]，通过在本地设备端附近部署几个计算节点。来自本地设备的数据直接传输到边缘节点进行计算和处理。与集中式云计算相比，这种转变提高了服务器的响应速度和可靠性。侯[19]等人实现了轻量级VR眼镜及边缘云设备之间的无线连接，并在这些设备之间实现了远程渲染，并将各种解决方案在延迟和其他方面进行了比较。通过其分散的结构，它对于Web3D可视化的实时计算和处理更加有效。

3.METHOD OVERVIEW

针对目前大型WebVR在线会议的三大技术瓶颈问题，本文提出的轻量化技术通过服务器预处理、网络传输、浏览器渲染三个阶段的处理来完成。服务器预处理阶段通过通过模型去除和场景的参数化处理等方法，主要解决资源数据量大的问题；网络传输阶段将数据分批传输，并且优先传输更重要的数据，着重解决传输慢的问题。浏览器渲染阶段通过优化shader中的处理过程，主要解决了渲染卡顿的问题。

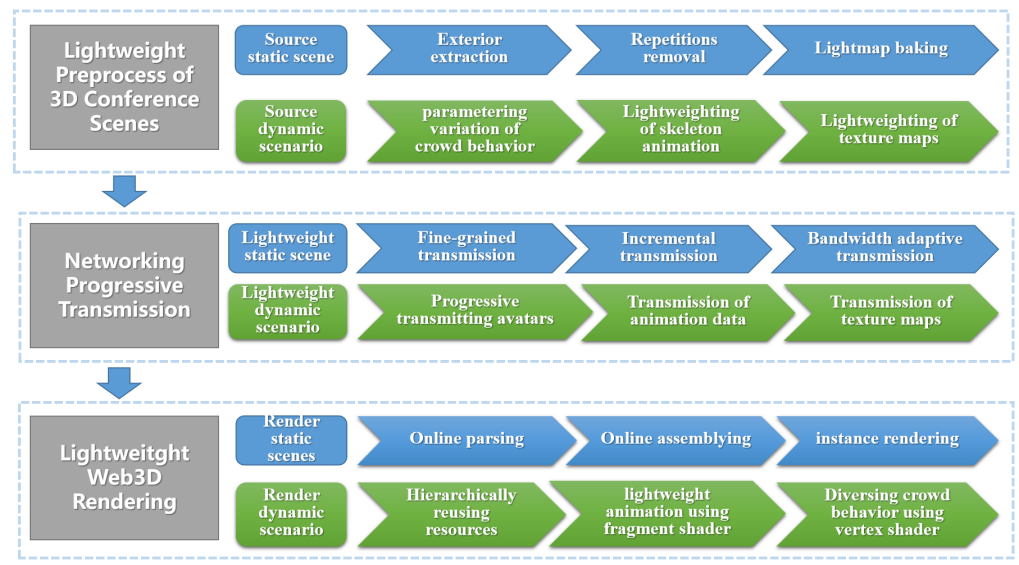


Figure 1 general technical route

服务器阶段要实现大规模会议情境的轻量化预处理。对于静态场景，这一阶段首先要提取房间模型的外壳。然后要找出并标注场景中所有的重复单元。之后再对场景进行光照烘焙，烘焙处理可以提高场景的真实感，并且对渲染性能的影响较低，是提高整体渲染效果的绝佳手段。对于动态情景，首先要对大规模人群进行参数化处理，将人物间的差异使用参数化的方式体现出来，对于整个人群每个人物表示为一组参数，接下来传输数据时只需要传输重用的资源和这些参数就可以在浏览器中还原出整个人群。为了方便渲染阶段在shader中处理我们需要提取出骨骼动画的数据并进行处理，先对项目中的骨骼动画进行分析，对于坐着的动画，主要是手臂动作，像腿部的骨骼基本是不动的，我们要把运动和不运动的骨骼分开，还需要把不运动骨骼一帧的数据和运动骨骼每帧的数据都计算出来。我们对场景中的纹理贴图进行了分级预处理，尽可以降低初始加载速度，并逐步提高贴图的像素。

网络传输阶段要实现细粒度化渐进式传输调度。对于静态场景，将场景分解为更小的单元进行细粒度化传输。对于动态情景，考虑到人物模型的精细度比较高，所以采用了PM的方式来传输网格数据，首先传输一个精度较低的基网格，然后不断传输网格的增量信息，不同调高网格的精细度，直至恢复为原网格。

浏览器阶段要实现轻量级大规模会议场景在线渲染。对于静态场景，首先要分析收到的数据，然后将各个部分的资源组装到一起，最后利用实例化渲染技术生成场景中的重复单元。

对于动态情景，首先要对人群的模型资源进行分级重用，不光减少了占用的内存资源还可以通过实例化渲染技术减少GPU访问内存的次数，从而极大地提高了项目的渲染性能。本文在对于shader里的操作进行了改进，使用fragment shader实现人群的动态化，然后再使用vertex shader实现人群的多样性，通过这些操作，我们将骨骼动画功能和人物间丰富的多样性融入到了实例化渲染技术中。

4.关键技术之一：大规模会议场景的轻量化预处理

对于静态资源，在预处理阶段主要进行外壳提取和模型去重。因为初始视点在模型的外面，我们希望模型中部件的渲染顺序是由内到外，所以外部模型的优先级要比内部的高，模型的。另外我们希望模型中相同的部件只被传输一次，并且使用实例化渲染的方式将这些模型添加到场景中，为了后续实现这些操作，我们必须在预处理阶段将所有重复单元找出，并单独处理。对于动态资源的处理，在这个场景中主要指的是人群的轻量化处理，一方面我们要尽可能的提高资源的重用度，另一方还要确保人物模型间的差异性不丢失。对骨骼动画数据进行预处理可以减少在浏览器上的计算量，从而提高渲染性能。

4.1**静态会场场景的轻量化预处理**

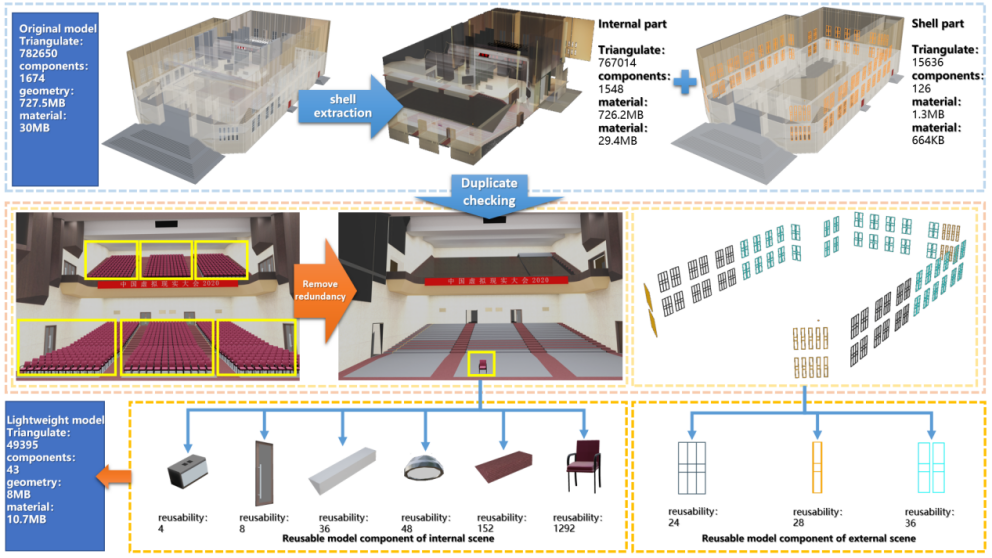


Figure 2 Lightweight processing of static venue scene

我们在静态模型的预处理阶段主要完成了两项工作，外壳抽取和模型去除。由于在项目中我们的初始视角看到的是场景的整体外貌，然后才会需要场景的内部细节，所以我们需要进行外壳抽取，优先传输这些数据。另外场景中有许多重复构件，我们要找出并标注场景中所有的重复单元，标注这些单元有两个作用，其一是为了接下来进行去重处理，去除场景中的重复单元，去除重复单元后再进行网络传输可以减少传输的数据量，其二是为了方便在渲染阶段对相同的单元进行实例化渲染处理。最后再对场景进行光照烘焙，烘焙处理可以提高场景的真实感，并且对渲染性能的影响较低，是提高整体渲染效果的绝佳手段。

有些部件的烘焙贴图所占的空间相较于模型网格比较大，这时对烘焙贴图的分级传输处理就非常重要，对这些部件我们将网格资源和烘焙贴图资源分开进行加载，并初始加载一个低像素的烘焙贴图，再逐步提高烘焙贴图的像素，处理方式与动态情景中对纹理贴图的处理类型。

在本项目案例中，由于场景数据量较大且模型结构复杂，会对模型传输与渲染带来很大的阻力，为例能够突破这些阻碍，因此就需要对模型场景进行进行轻量化处理：

1．对于模型材质的轻量化与标准化：由于在本案例中无透明材质的需求，因此需要将贴图中所带有的透明通道进行去除，以达到模型轻量化与标准化的目的，大大降低材质文件的数据量。

2．对于室内场景材质的光照烘焙处理：为加强模型的观赏性和轻量型，因此需要对模型进行光照烘焙。光照烘焙可以加强模型的观赏性，且由于模型材质并非会占据整块贴图，因此大大提高了贴图的利用率。

3．对于模型构建的查重处理：由于场景建筑模型中存大量重复构件，包括门窗、室内摆设、礼堂座椅等。因此通过查重处理，系统会将重复的模型构件仅保留一份作为样本进行传输，并在完成传输后，在网页前端进行重用构建，以达到还原场景的效果。

4.为了提高初始加载速度，对贴图资源进行了分级传输。对静态场景的烘焙贴图和动态人群的纹理贴图都进行了分级处理，先传输分辨率低的贴图，再逐渐提高贴图的分辨率。

4.2**大规模参会人群的轻量化预处理**

1.模型资源的分级重用。

资源的分级复用处理有两个目的，从数据传输的角度，复用的资源只传输一次提高了传输速度，从场景渲染的角度，采用渲染的人群

（1）根据通过geometry来划分不同的人群中

geometry是一个化身对象最复杂的资源，所以geometry的复用比其它资源更重要，为了尽可能提高资源的复用度，我们让整个人群对象共用一个geometry，如果一些化身使用不同的geometry，必须为每个geometry创建一个人群对象。人群对象是渲染的单位，整个人群被作为一个整体渲染到场景中。

即便使用相同的geometry我们也可以让化身间保持形体上的多样性，通过对化身进行放缩可以实现化身整体的形变，对骨骼进行放缩可以实现化身的局部形体差异。

（2）提取复用资源

一个人群对象共用一个geometry，除此之外其它的资源也都或多或少被复用。人群对象除了geometry外主要是纹理贴图资源和骨骼动画资源。

两个人群对象之间可以共用骨骼动画资源，不过人群间共用资源只是减少了网络传输的数据量，并不能提高渲染效率从而提高渲染质量，因为人群对象被视为一个整体进行渲染。

（3）参数化表示人群

将人群的全部资源进行整理并编号。

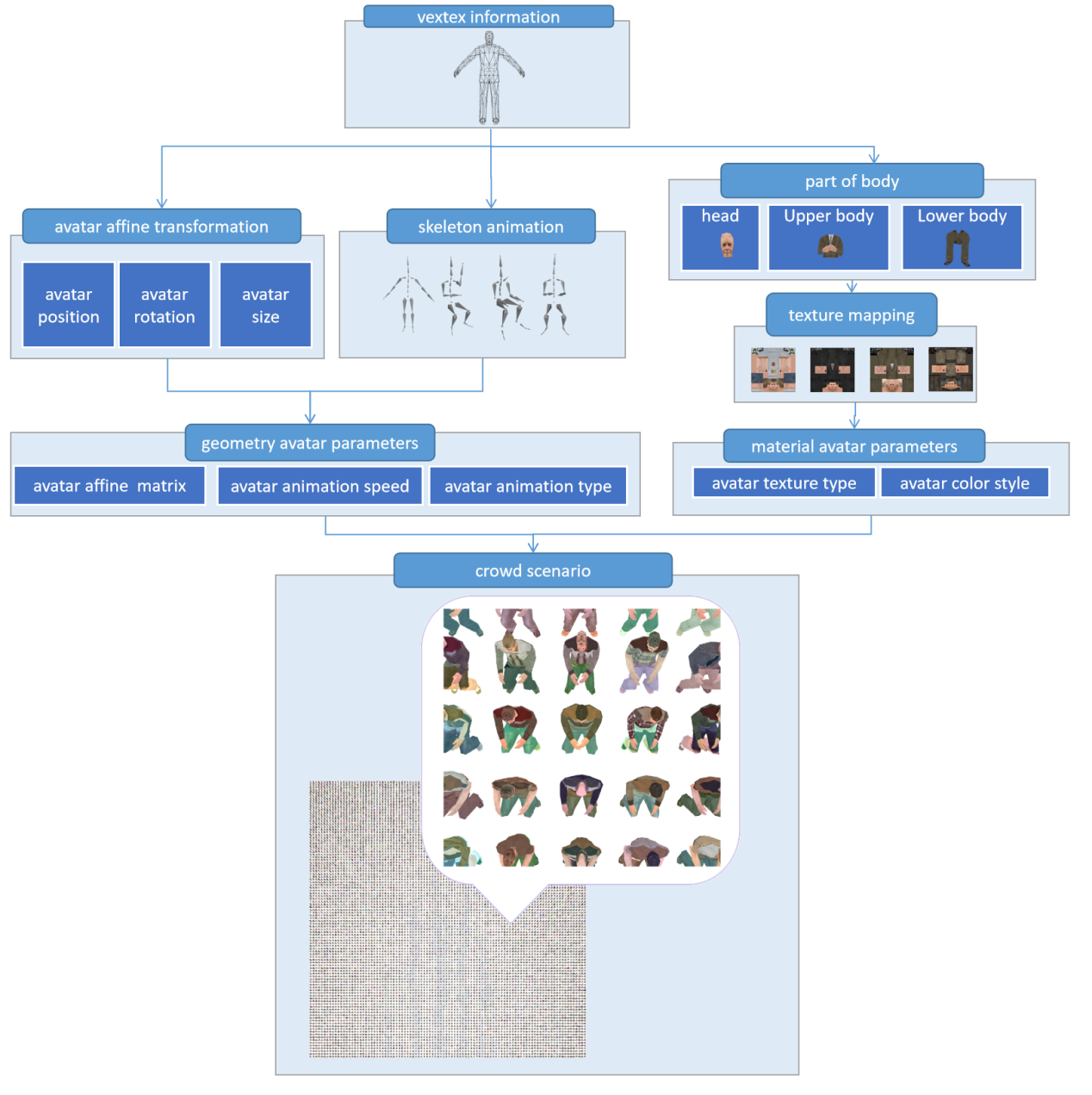


Figure 3 Lightweight behavior modeling of participants

2.增大临近人物对象间的差异性

为了使得人群渲染的多样性效果更加明显，我们希望相似的化身距离越远越好。我们要求一个最佳的人群设置方式，使得相似的化身进可能分散开，并且化身间的差别尽可能大，因此我们定义一个叫**分散度Disperse**概念，用来描述整个场景中相似对象的分散程度，Disperse的取值范围为0到1，我们希望场景的Disperse尽可能大。

**定义 1：性别区分度（Different degrees of gender, Diff\_G）：**

Diff\_G(,)表示和两个化身性别的区分度。如果和相同性别Diff\_G为0，否则Diff\_G为1。性别区分度的计算公式如下，公式中和分别表示第i个化身和第j个化身：

 (1)

**定义 2：形态区分度（Different degrees of shape, Diff\_S）：**

Diff\_S(,)表示和两个化身高矮胖瘦的区分度，取值为0到1，为1的时候差异度最大，为0的时候两个化身形态完全相同、没有差异。形态区分度的计算公式为：

 (2)

和分别表示和两个化身包围盒的长度，lMax表示化身包围盒的最大长度限制。和分别表示和两个化身包围盒的宽度，wMax表示化身包围盒的最大宽度限制。和分别表示和两个化身包围盒的高度，hMax表示化身包围盒的最大高度限制。

**定义 2：颜色区分度（Different degrees of color, Diff\_C）：**

Diff\_C(,)表示和两个化身色调的区分度，取值为0到1，为1的时候差异度最大，为0的时候两个化身色调完全相同、没有差异。形态区分度的计算公式为：

 (3)

和分别表示和两个化身红色色调的增强程度，取值范围为0到255。和分别表示和两个化身绿色色调的增强程度，取值范围为0到255。和分别表示和两个化身蓝色色调的增强程度，取值范围为0到255。

**定义 3：纹理区分度（Different degrees of texture, Diff\_T）：**

Diff\_T(,)表示和两个化身纹理的区分度，取值为0到1，为1的时候差异度最大，为0的时候两个化身色调完全相同、没有差异。形态区分度的计算公式为：

 (4)

公式中w5,w6,w7,w8表示权重，并且权重之和为1。

Diff\_TH、Diff\_TC、Diff\_TT、Diff\_TS分别为面部纹理区分度（Different degrees of head texture）、上衣纹理区分度（Different degrees of coat texture）、裤子纹理区分度（Different degrees of trousers texture）、鞋子纹理区分度（Different degrees of shoes texture）的区分度，使用相同的纹理值为0，使用不同的纹理值为1。它们的计算公式为：

 (5)

 (6)

 (7)

 (8)

其中MapHead、MapCoat、MapTrousers、MapShoes分别表示头部、上身、裤子、鞋子的纹理贴图。PSNR(Peak Signal to Noise Ratio)表示两张纹理贴图之间的峰值信噪比，PSNR为0表示两张贴图差异最大，PSNR为正无穷表示两张贴图完全相同。

**定义 4：分散度（Disperse）：**

 (9)

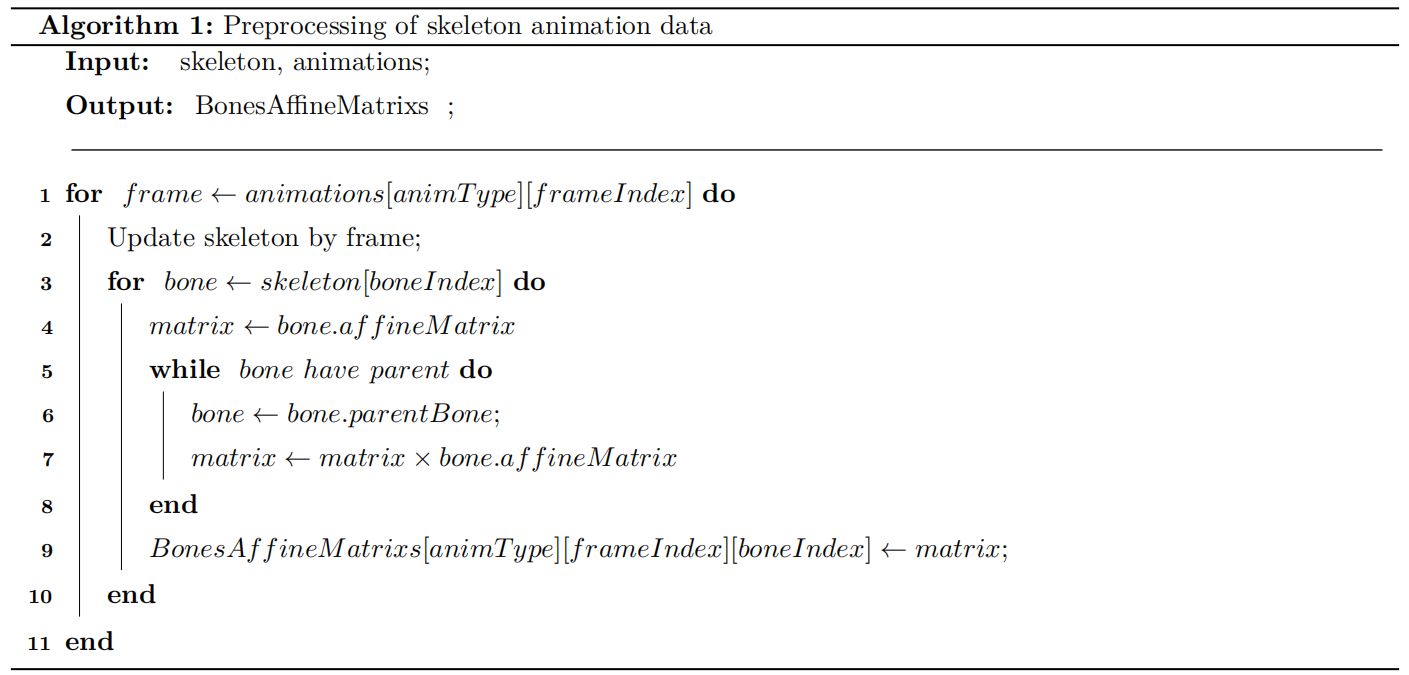
公式中w1,w2,w3,w4是权重，并且权重之和为1。i和j表示化身的编号。n表示化身的总个数。Dist(,)表示和两个化身间的距离。Diff\_G描述了对象间gender的区分度。

3骨骼动画数据的预处理

对于传统的骨骼动画实现方法，需要在浏览器上为每个化身计算骨架中每一个骨骼的仿射变换矩阵，每个仿射矩阵的计算为了分析骨骼间的父子关系都需要进行多次矩阵乘法，并且每帧的仿射矩阵都需要重新计算，当人数非常多时，这种方法所耗费的计算量是无法容忍的。

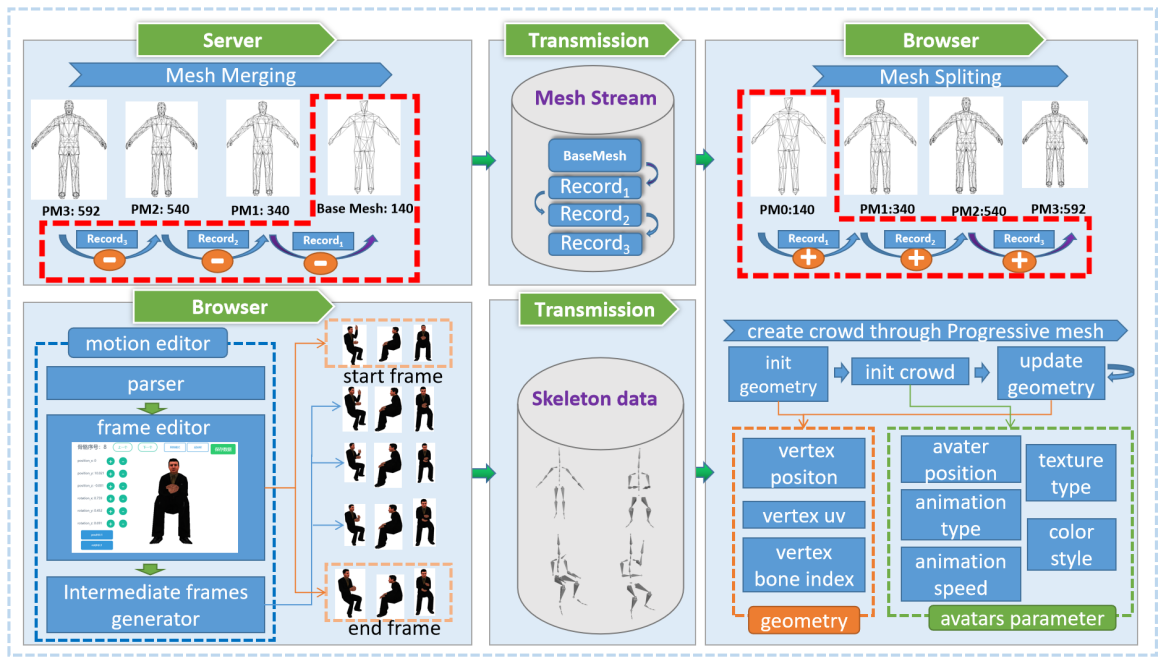
我们的处理方法是人群中所有化身都不再单独计算骨骼的仿射变换矩阵，所有使用同一种动画的化身共用一组骨骼动画数据。我们为每种动画选定了一个可以流程展现出这段动画的最小帧数，然后计算出每一帧的骨骼仿射变换矩阵，每一个使用这种骨骼动画的化身根据自己的动画播放速度和播放时间计算出当前应当在场景中呈现的那一帧的编号，然后根据帧编号从共用的动画数据中查询对应的仿射变换矩阵。

这些骨骼动画数据是固定的，我们可以在服务器的预处理阶段就计算好这些数据。为了节省在浏览器中的计算量，我们计算出了每种动画每一帧中每一个骨骼的仿射变换矩阵，这些数据作为我们进行人群渲染所需的动画数据，具体实现方法见算法1。



5.关键技术之二：细粒度化渐进式传输调度

该步骤的目的在于将场景中的数据细化拆分，提高调度的灵活性，并且减少不必要的信息传输，重用的部件只传输一次。粒度指的是所有位于可视区域的构件单元集合，而细粒度化处理指的是对构件单元进行细化差分，从而使得我们可以进行更加灵活的资源调度。传输调度的核心目标是提高初始加载速度，主要思想是优先传输重要的信息，为了实现这些功能，我们需要首先进行细粒度化的处理，以提高资源调度的灵活性，然后要判断出不同构件单元的重要性，优先传输更重要的部分。



**Figure 4** **Fine grained progressive transmission scheduling**

在预处理阶段，首先通过对3D模型进行压缩处理来获得基网格，并记录压缩处理的操作过程，PM就是模型压缩处理操作的逆过程。压缩处理可以看成不断将相邻两个点合并的过程，如果这两个点都在边缘经过合并后会减少一个三角面，如果这两个点在网格内部经过合并操作后会减少两个三角面。压缩处理的每一步操作都可以表示为被合并两个点的索引加合并后那个点的位置，PM传输和使用的增量信息就是压缩处理过程中每一步合并点的索引和合并后点的位置，PM的过程中可以根据这些信息将一个点分裂为两个点，从而不断增加三角面的个数。

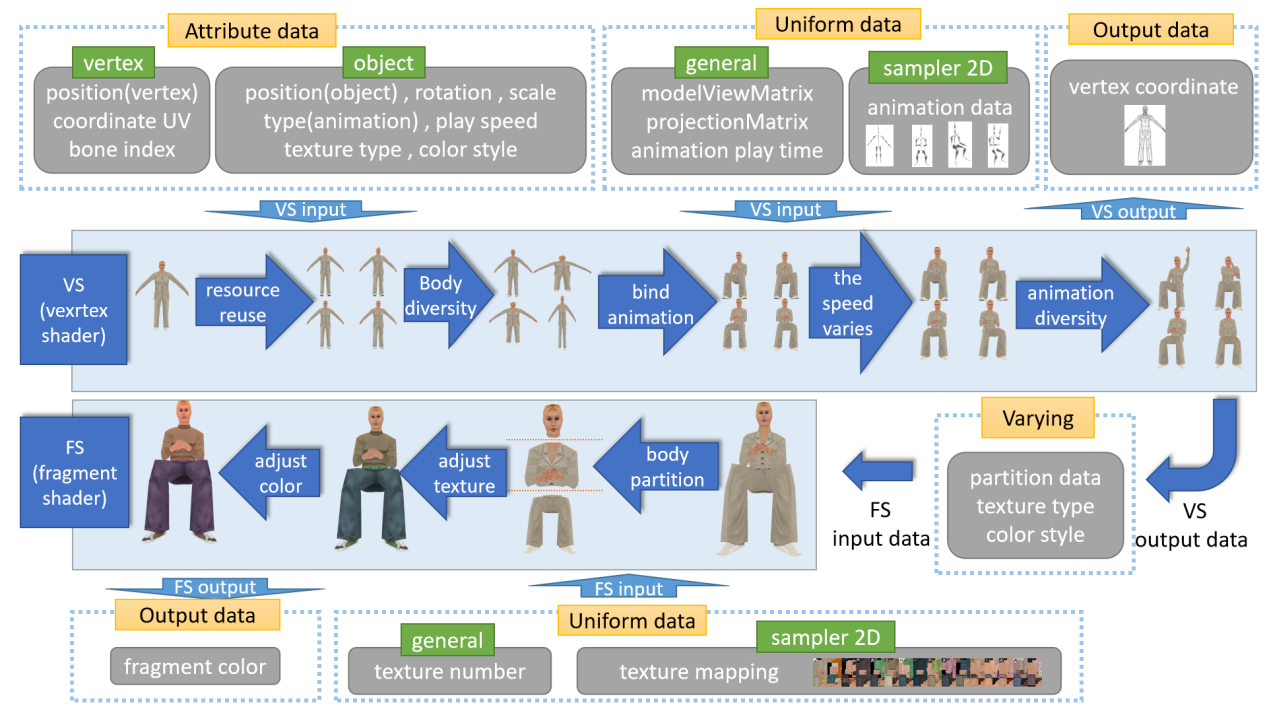
在网络传输阶段，首先传输基网格信息，然后传输增量信息。每一组增量信息，记录了一个等待分裂点的索引，和这个点分裂后的两个点的信息。为了提高传输效率，我将多组增量信息合并为一个数据包发送，一个数据包包含多少组增量信息是可以自由设置的，应当结合项目中的具体情况来设置数据包的大小，如果数据包过大会导致客户端网格更新间隔过大从而显得过渡不自然，如果数据包过小会使得数据包数量过多，频繁地发送和接收处理数据包会降低项目整体的性能。

数据的传输能够自适应具体的网络带宽。渐进式网格的增量数据被划分为多个数据包，客户端浏览器只有接收到上一个数据包后，才会请求服务器发送下一个包。当网络质量较高时，每个数据包的传输时间更短，客户端请求的频率更高，服务器发送数据包的频率也就更高；当网络质量较差时，每个数据包的传输时间更长，客户端请求的频率更低，服务器发送数据包的频率也就更低。

在浏览器阶段，接收到基网格数据后就将基网格渲染到场景中，然后每接收到一个数据包的增量信息就更新一次场景中的模型。一个数据包中包含多组增量信息，每一组增量信息记录了一个待分裂的点，经过一次分裂网格中的顶点个数增加一个。经过一次分裂，一个点分裂为两个点，如果分裂后的两个点位于网格内部，三角面的个数经过一次分裂会增加两个，如果分裂后的两个点都位于网格边缘，三角面的个数经过一次分裂会增加一个。

6. 关键技术之三：轻量级大规模会议场景在线渲染

大规模动态场景的轻量化渲染需要vextex-shader和fragment-shader的相互配合。vextex-shader和fragment-shader都是可编程的处理单元，vextex-shader的主要功能是对geometry的顶点位置进行变换，fragment-shader的主要功能是计算光栅化后每个片元的颜色。它们的操作是由GPU中的专门硬件实现的，执行速度很快。

Figure 5 Lightweight rendering pipeline of large scale virtual conferencing crowd

**6.1动态情景的实例化渲染技术**

实例化渲染技术被广泛应用于在场景中重复渲染静态的3D对象，我们借用实例化渲染的思想实现了动态人群情景的高效渲染，并且保留了人群动作的多样性。实例化渲染是通过在shader中的操纵实现的优化，主要是通过在vextex-shade里重复渲染geometry中的点可以达到在场景中渲染出多个对象的目的。

（1）让实例化渲染的人群做出动作

为了用实例化渲染技术渲染出动态情景，我们将所有骨骼的仿射变换矩阵都输入到vextex-shader中，在vextex-shader对geometry中的顶点位置进行实例化渲染变换之前，先进行骨骼动画变换，通过将顶点位置乘以对应骨骼的仿射矩阵来实现人群中所有化身的骨骼动画。

（2）实现动作间的过渡，让实例化渲染人群动起来

为了实现动作间的过渡，我们需要向vextex-shader中传入一个描述动画播放时间的变量，并且不断定时更新这个变量，在shader中根据时间计算相应的帧序号，并根据不同的帧序号使用对应的帧下的骨骼仿射变换矩阵。随着我们不断更新传入shader的时间变量，场景里的化身也就呈现出不同帧下的效果。

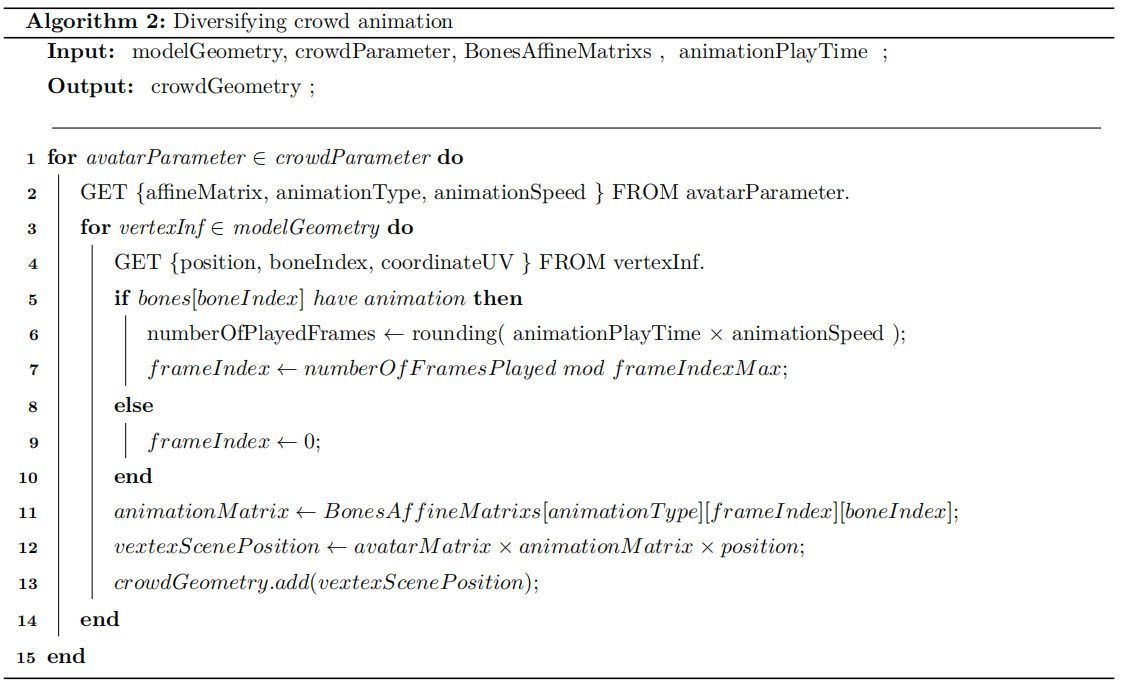
（3）实现化身间的动画不同步

实现以上的这些处理后你会发现人群中所有化身的动作都是一致的，这是因为整个人群共用一个时间变量，同一时刻所有化身计算出来的帧序号是相同的。为了让动画不同步，我们需要为每个化身设置一个动画播放速度，然后将所有化身的播放速度参数传入shader，shader中在计算化身的帧序号时除了时间还要考虑化身的动画播放速度，这样在同一时刻每个化身都呈现出不同帧的效果。

（4）实现动画类型的多样性

为了实现让每个化身可以播放不同的动画，我们需要为每个化身设置一个描述动画类型的参数，然后将所有化身的动画类型参数传入shader，shader中根据每个化身不同的动画类型，为该化身的顶点选取对应的骨骼仿射变换矩阵。

以上操纵的具体实现方法见算法2。



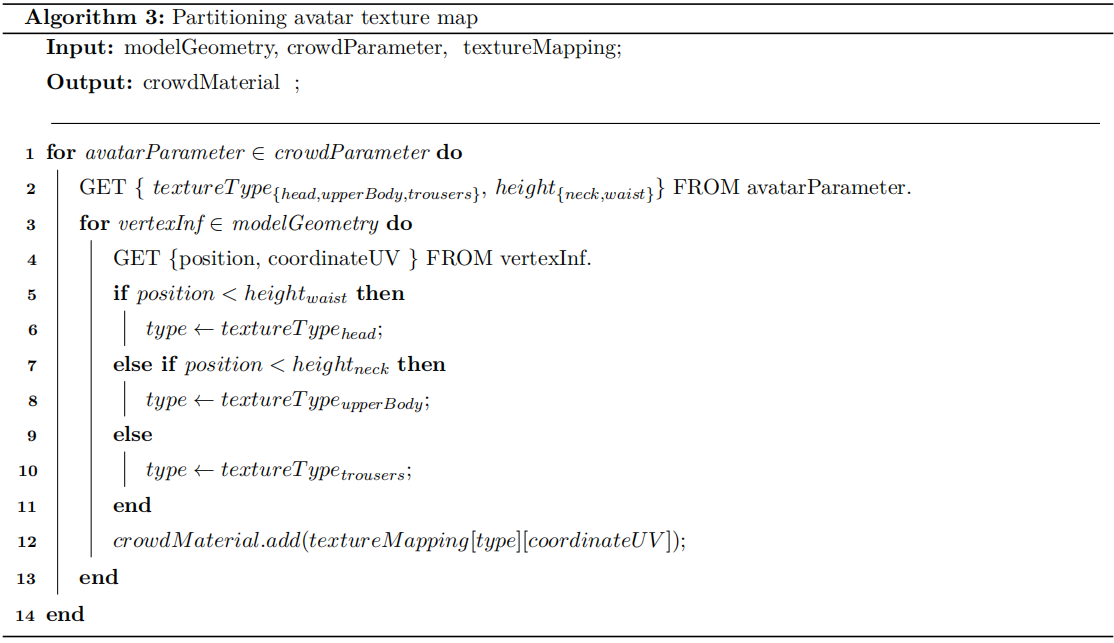
**6.2轻量化人群中化身纹理多样化的实现方法**

（1）为实例化人群绑定贴图

为动态实例化人群绑定贴图的方法和为静态实例化场景绑定贴图的方法类似。所有化身都使用同一组UV数据，在vextex-shader获取顶点的UV坐标，然后fragment-shader通过这个UV坐标从纹理贴图中获取像素。

（2）化身纹理贴图的分区绑定

我们在项目中将纹理贴图划分成头部、上衣、裤子三部分，分别与化身的对应区域绑定。分区绑定纹理贴图可以丰富纹理贴图的搭配效果，如：男性贴图有32套，采用分区绑定后这些贴图可以搭配出32768种效果。

我们需要为每个化身增设三个参数，分别用于描述化身的头部、上衣、裤子这三个区域的贴图类型，然后将所有化身的贴图类型参数传入shader，在vextex-shader中根据顶点坐标判断出顶点所处位置到地面的高度，通过高度值来区分当前位置位于肢体的哪个部位，然后fragment-shader根据判断结果选取对应部位的贴图。具体实现方法见算法3。

（3）化身的色调调整

我们首先需要为每个化身添加三个参数，这三个参数分别表示化身红、绿、蓝的颜色增强程度。Fragment-shader中在获取纹理贴图的像素后，将像素的红绿蓝三个颜色值分别加上化身的三个颜色参数，然后将计算结果作为片元颜色输出。

**6.3人群渲染方面其它的优化处理**

将每个顶点依赖的骨骼数量由四个变为一个。根据3D骨骼动画的格式规范，每个顶点的运动最多受到4个骨骼的影响，受到每个骨骼的影响程度用权重来描述，对应四个骨骼的四个权重之和为1。实际上大多数顶点的运动只受到1个骨头的影响，也就是说其它3个骨头对应的权重为0。因此我们尝试了在计算过程中只考虑一个骨头，把所有的顶点都看成只受到单个骨头的影响，结果渲染性能有了较大的提高，并且对动画质量的影响不大。

设计了专门用于存储骨骼动画数据的16位浮点数格式，相较于float类型传入shader的数据量减少了一半。

我们只传入化身的左半部分纹理贴图进入shader，然后利用化身的左右对称的性质读取纹理贴图的像素。

7. Experimental Results and Performance Analysis

Table2 Hardware configuration of test environment

|  |  |  |  |
| --- | --- | --- | --- |
|  | PC | Laptop | Handset |
| Computer type | desktop computer | Notebook computer | mobile phone |
| CPU | I7-10700U | I5-10210U | HUAWEI Kirin990 5G |
| Memory | 16GB | 8GB | 8GB |
| GPU | Nvidia GeForce RTX 2060 SUPER | Nvidia GeForceMX330 | Mali-G76 |
| OS | Windows 10  64Bit | Windows 10  64Bit | Android10 |
| Networks | 4G Wireless  Network | 4G Wireless  Network | 4G Wireless  Network |
| Browser | Chrome | Chrome | WeChat |

对于静态场景，通过轻量化处理处理，将原本由782650个三角面片构成、数据量约为727.5MB的模型文件，轻量化为仅有49395三角面片、数据量仅有约18.7MB的可传输模型数据包，轻量化非常效果明显。以场景中的座椅为例，原始场景中共有1292把座椅，模型数据量约699MB。通过上述处理，系统仅保留了一把座椅作为重用样本，因此座椅模型的传输数据仅有约10KB。经过我们的轻量化处理虽然数据量减少了，但是模型的精度没有变、细节也没有损失。

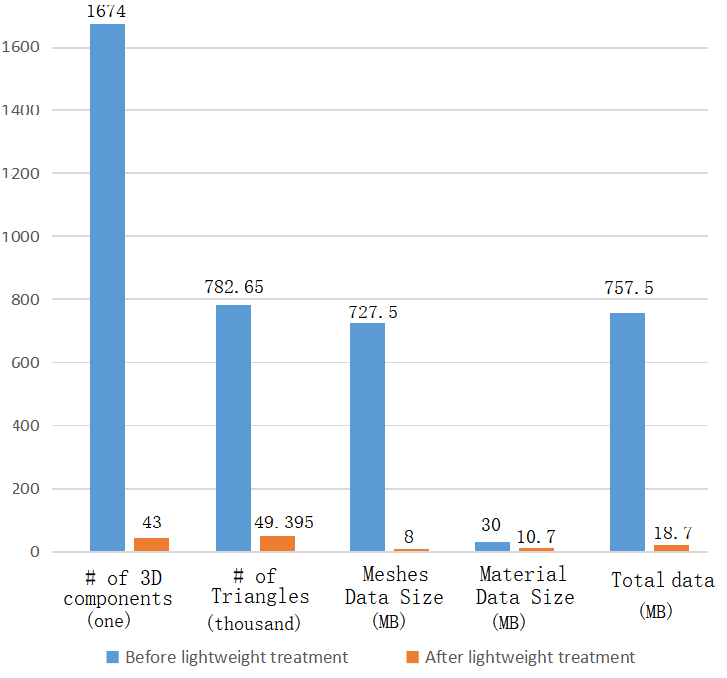


Figure 6 Performance data of lightweighting conference sceneriao

Table 2 Data reduction after lightweight processing of static scene data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **# of 3D components** | **# of Triangles** | **Meshes Data Size** | **Material Data Size** | **Total data** |
| Without lightweight treatment | 1674 | 782650 | 727.5 MB | 30 MB | 757.5 MB |
| with lightweight treatment | 43 | 49395 | 8 MB | 10.7 MB | 18.7 MB |
| Reduced amount of data | 1631 | 733255 | 719.5 MB | 19.3 MB | 738.8 MB |
| Reduced ratio of data | 97.40% | 93.70% | 99% | 64% | 97.60% |

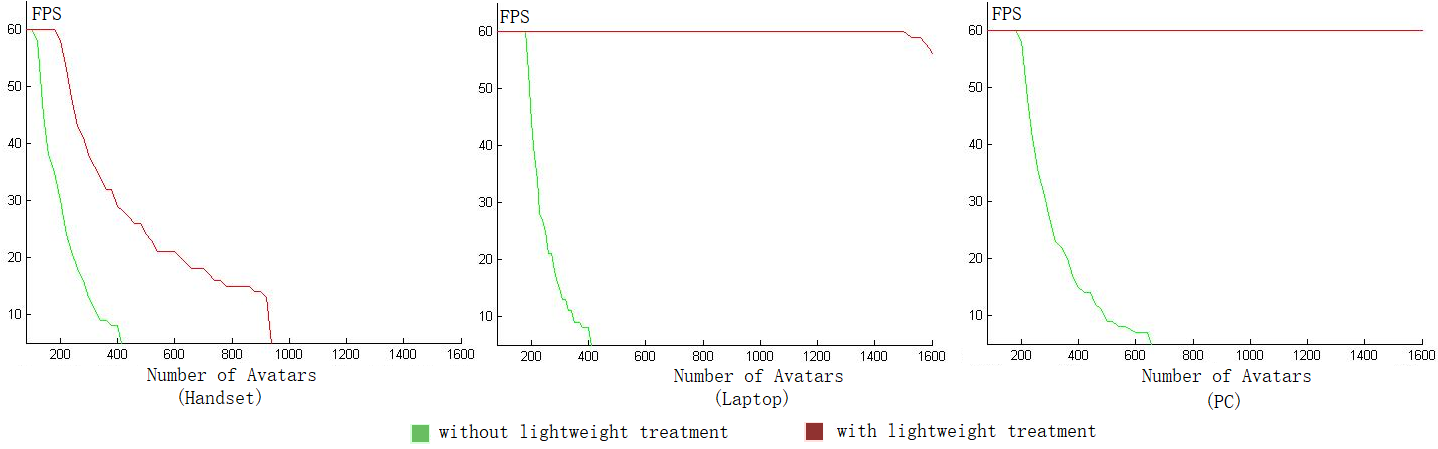


Figure 7 Performance data on population of Web3D conferencing crowds

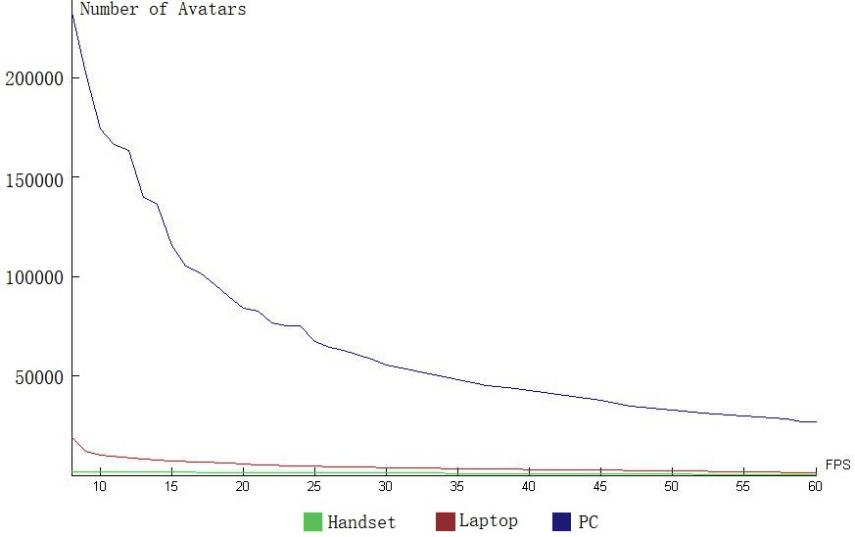


Figure 8 Comparison of rendering with lightweight effects of different devices

从图7左图可以看出经过轻量化处理后项目的渲染效果得到了明显的改善，从图7右图可以看出人群的轻量化处理技术对硬件环境比较敏感，不同硬件环境下的渲染效果有较大差别，但无论是台式机设备还是笔记本电脑设备，通过轻量化处理后项目的渲染性能都有了极大的提升。

Table 3 The maximum size of the crowd when the number of frames is not less than 30

|  |  |  |  |
| --- | --- | --- | --- |
| **Test equipment** | **without lightweight treatment** | **with lightweight treatment** | **Rate of increase** |
| PC | 308 | 55501 | 180.19 |
| Laptop | 304 | 3667 | 12.06 |
| Handset | 153 | 327 | 2.13 |

从表4可以看出，在保证场景帧数不低于30帧的情况下，使用测试设备1场景中可容纳的最大人群规模提高为原来的180.19倍，使用测试设备2场景中可容纳的最大人群规模提高为原来的12倍。从该表还可以看出硬件配置越好的设备采用轻量化处理后的优化效果越明显。新的方法可以达到十万人基本，但是传统的方法只能进行百人基本的渲染。该框架确实可以满足大规模多人在线webVR会议的千人级别渲染要求。



Figure 9 Monocular snapshots of our large-scale Web3D online conferencing Scenario

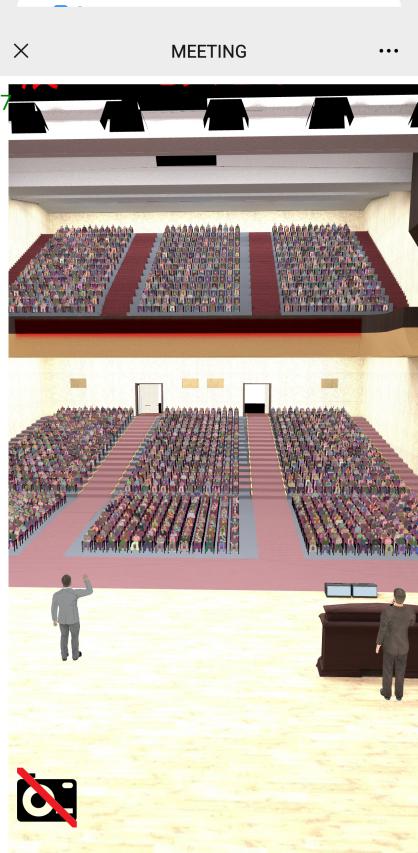


Figure 10 Mobile on mobile phones effects of our large-scale Web3D online conference Scenario

【**艾子豪**】下面再来插入一张VR双目显示的会议场景（分别左右眼两张截图）

**先把截图弄好**

Figure 11 Binocular snapshots of our large-scale WebVR online conferencing Scenario

图9和图10和图11是测试时项目运行的效果截图，项目的地址是 [http://youku3d.com/smart3d/meeting5/](http://youku3d.com/smart3d/meeting4/) 。

**8.Conclusion**

在本研究中，我们提出了对模型的轻量化处理、传输和渲染的一整套解决方案，以解决在大规模多人在线webVR会议中遇到的挑战。 提出的解决方案满足了轻量化、多样性的需要。在对静态资源的处理上，我们通过外壳提取、模型去除、烘焙处理、实例化渲染等手段，实现了资源的快速传输和渲染的真实性。在对动态资源的处理上，我们的方法通过对3D资源的分级重用和通过shader使用GPU计算等方式，在保证人群中人物多样性的前提下拥有良好的渲染性能。

我们的工作还有一些需要进一步完善的地方，由于shader语言严重依赖硬件，所以我们的项目对硬件的兼容性不是非常理想，在移动端对骨骼动画数据的处理还有一些问题，人群的轻量化处理效果也受到硬件环境的影响，这是我们接下来要继续处理的问题。另外我们在项目中使用的许多骨骼动画不是通过动作捕捉设备录制的，而是通过自制插件制作的，由于我们的骨骼动画制作系统不够完善，做来的动画不够生动，这也需要我们接下来继续完善。

9.Reference

[1] VSWORK虚拟空间[EB/OL] <http://www.vswork.com/>

[2] ENGAGE Virtual Reality Education & Corporate Training. VR Education Holdings PLC [EB/OL] https://engagevr.io/

[3] Mozilla Hubs [EB/OL] <https://hubs.mozilla.com/>

[4] MeetinVR[EB/OL] <https://www.meetinvr.com/>

[5] Spatial[EB/OL] <https://spatial.io/>

[6] Glue[EB/OL] <https://glue.work/>

[7] FrameVR[EB/OL] <https://framevr.io/>

[8] BigScreenVR[EB/OL] https://www.bigscreenvr.com/

[9] AltSpace[EB/OL] https://altvr.com/

[11] Xiaojun Liu, Ning Xie, Kai Tang, Jinyuan Jia. “Lightweight for Web3D Visualization of Large-scale BIM Scenes in Real-time”. Graphical Models, vol. 88, pp. 40-56, November，2016.

[12] Laixiang Wen, Ning. Xie, and Jinyuan. Jia. “Fast accessing Web3D contents using lightweight progressive meshes”. Computer Animation and Virtual Worlds. 27(5): 466-483, May, 2016.

[13] Alessandro Pluchino, Cesare Garofalo, Giuseppe Inturri, Andrea Rapisarda, and Matteo

Ignaccolo. 2013. Agent-based simulation of pedestrian behaviour in closed spaces:

a museum case study. arXiv preprint arXiv:1302.7153 (2013).

[14] Fengting Yan, Jinyuan Jia, Yonghao Hu, Qinghua Guo, and Hehua Zhu. 2019. Smart

fire evacuation service based on Internet of Things computing for Web3D. Journal

of Internet Technology 20, 2 (2019), 521–532.

[15] Daniel P Savoy, Marcio C Cabral, and Marcelo K Zuffo. 2015. Crowd simulation

rendering for web. In Proceedings of the 20th International Conference on 3D Web

Technology. 159–160.

[16]Ke Li, Qian Zhang, Hantao Zhao and Jinyuan Jia. User Interests Driven Collaborative Cloud-Edge-Browser Architecture for WebBIM Visualization. ACM Web3D, Nov. 13-15, 2020.

[17] Tianchu Zhao, Sheng Zhou, Xueying Guo, Yun Zhao, and Zhisheng Niu. 2015. ACooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-Aware Mobile Cloud Computing. (2015), 1–6.

[18] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge Computing:Vision and Challenges. IEEE Internet of Things Journal 3, 5 (2016), 637–646.

[19] Xueshi Hou, Yao Lu, and Sujit Dey. 2017. Wireless VR/AR with Edge/Cloud Computing.(2017), 1–8.

[20] Nguyen, V.T. and Dang, T., 2017, October. Setting up virtual reality and augmented reality learning environment in unity. In 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct) (pp. 315-320). IEEE.

[21] Nguyen, V.T., Hite, R. and Dang, T., 2018, December. Web-based virtual reality development in classroom: From learner's perspectives. In 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR) (pp. 11-18). IEEE.

[22] Miyata, K., Umemoto, K. and Higuchi, T., 2010. An educational framework for creating VR application through groupwork. Computers & Graphics, 34(6), pp.811-819.

[23] Häfner, P., Häfner, V. and Ovtcharova, J., 2013. Teaching methodology for virtual reality practical course in engineering education. Procedia Computer Science, 25, pp.251-260.

[24] Dixon, R.A., Hall, C. and Shawon, F., 2019. Using Virtual Reality and Web Conferencing Technologies: Exploring Alternatives for Microteaching in a Rural Region. Northwest Journal of Teacher Education, 14(1), p.4.

[25] Thomas Erickson, N. Sadat Shami, Wendy A. Kellogg, and David W. Levine. 2011. Synchronous interaction among hundreds: an evaluation of a conference in an avatar-based virtual environment. ACM CHI’ 11, New York, NY, USA, 503–512. doi:https://doi.org/10.1145/1978942.1979013.

[26] Yeh S, Chen Y, Wang T H, et al. Virtual conference management system. ICOIN 2001. 776 - 781.

[27] Gurunathan, Pradeep , and S. Pandian . "A Novel Approach for Web-based Conference Management System." International Conference on Computer Engineering & Technology IEEE, 2009.

[28] James A E , Nanos A G , Thompson P . V-ROOM: a virtual meeting system with intelligent structured summarisation[J]. Enterprise Information Systems, 2015:863-892.

[29] Virbela[EB/OL] https://www.virbela.com/