**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

**摘要**

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、分级复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。

请给出几个关键数据: 轻量级压缩率，传输效率，响应延迟等等

关键字

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. **引言**

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. 【代入感差】当前的在线会议系统不具有3D可视化。传统的音视频的方式无法充分还原会议的真实场景，代入感较弱，沉浸感差，缺少VR/AR等功能。
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。
3. 【规模性太小】因为浏览器计算能力严重受限，WebVR会议系统的在线参会人数非常受限，无法支持大规模多人在线参会，部分在线会议系统较为重量级，对用户设备及网络环境要求高，会议加载速度慢，响应不及时等。

基于后疫情时代下的在线远程会议需求，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. **大型会议情景中网络传输太慢**。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，受制于有限的内存空间与弱计算能力，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。对渲染速度、渲染精度、资源占用等都有着极高的要求。

本文的研究目的是提出一套大规模轻量级WebVR在线会议技术解决方案，该方案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. **相关研究工作**

**2.1 虚拟在线会议系统的现状**

2020年，可以说是各大厂商开始频繁地将VR相关技术应用在展览展会、线上会议中的一年。从谷歌、苹果、Facebook、MWC、E3、Unity等等每年备受关注的大会纷纷宣布改为线上举办后，利用科技手段在线上举办发布会逐渐形成一种潮流。

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PC客户端的VR会议平台的研究，如VSWork[1]、Engage[2]等。第二类是对基于Web端的在线社交会议平台的研究，如Mozilia hubs[3]。

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。

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。

总的来说，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；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。

**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[4]等人提出了一种基于WebBIM场景的数据轻量化方法。孙[5]等人提出了一种基于内容的可比较压缩算法，还有一些方法着重考虑模型数据之间的相似性，并将这种相似性与差异性进行度量，从而达到压缩数据的目的[6][7].特别地，许多学者将语义信息引入了模型检索过程，提出了一些基于语义的检索与冗余数据检测的间接性方案[8][9]。还有一些学者，在语义信息之外，还将几何处理方法纳入了他们的解决方案中，如laga[10]等人与zheng[11]等人，在几何识别中加入了一些语义规则.

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 大型在线会议动态情境的轻量化处理**

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。 目前，模型参数化的概念已应用于Web3D建筑物信息建模（BIM）中的建筑物简化中[14]。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。朴[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。

**2.4 大型在线会议情境的在线可视化调度与架构部署**

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[18][19][20]，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[21]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。最终证明，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。

1. **总体技术路线**

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，并且优先传输优点度高的数据，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。

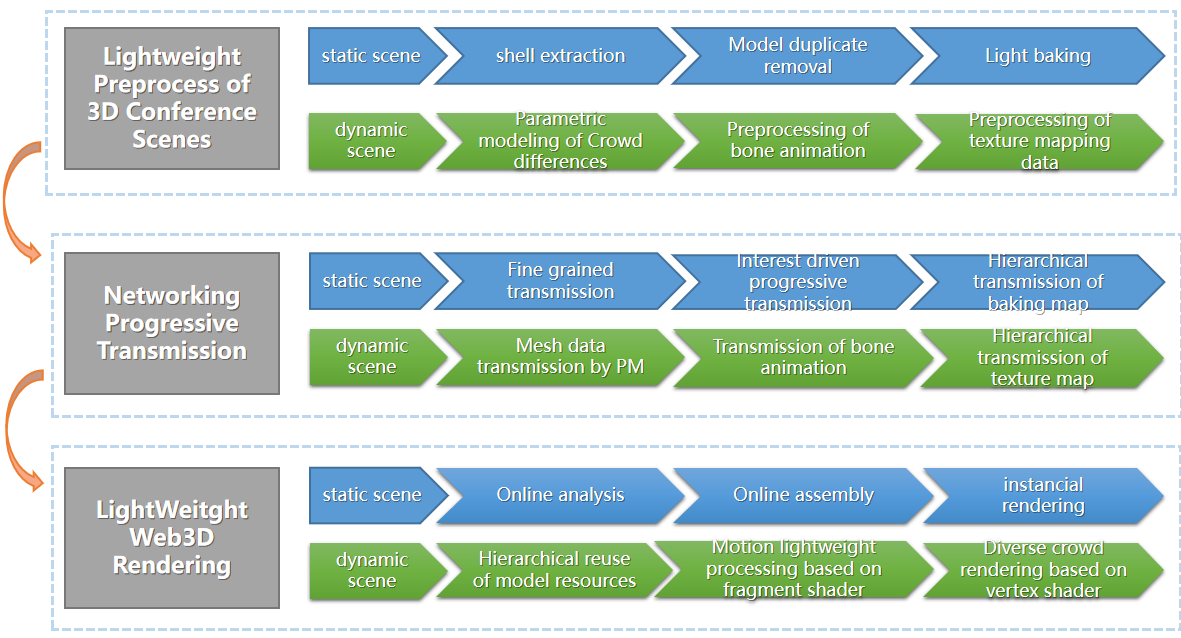


图 1 方法概要，简要解释每个模块的功能

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，因为镜头初始时位于房间外，首先需要看到房间的整体外部结构，然后进入房间后才会需要内部细节，所以房间模型的外壳数据需要优先传输。然后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。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，我们在文中使用了兴趣度这个概念，兴趣度可以描述模型单元的重要程度，越重要的单元越需要被优先传输，我们通过兴趣度分析各个部分的重要性，从而确定这些单元的传输优先次序。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。

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复用，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。

1. **关键技术之一：大规模会议情境的轻量化预处理**

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复用度，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**

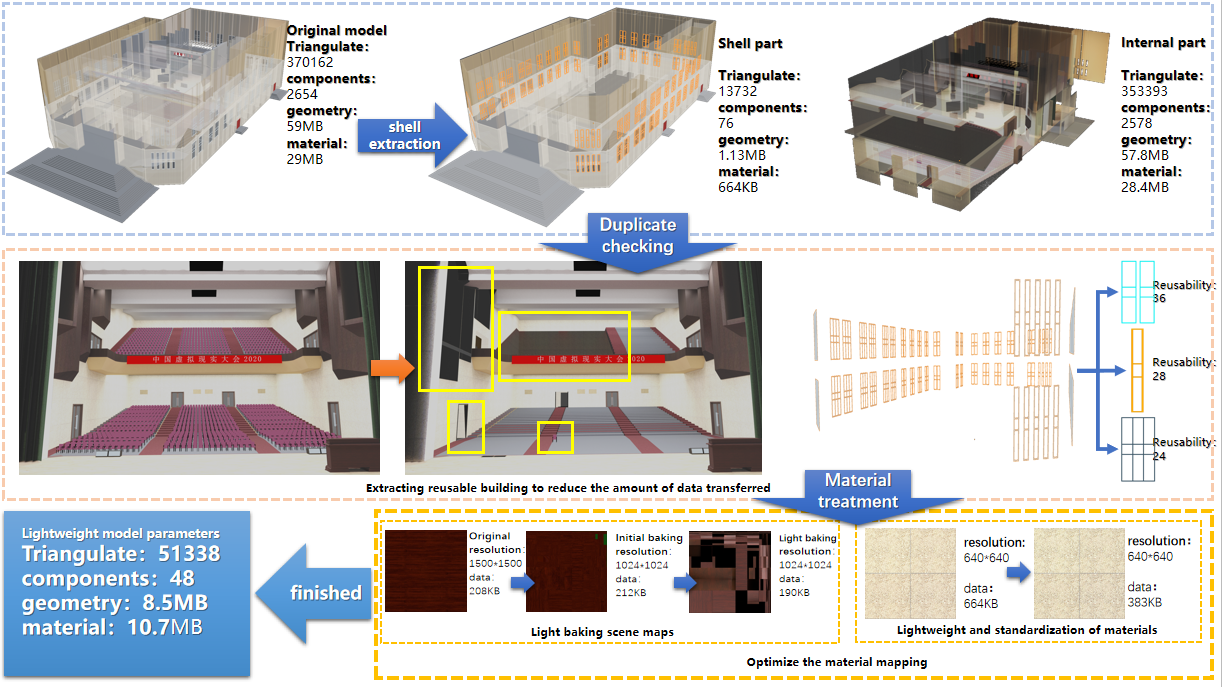


图 2 **静态会场场景的轻量化处理， 简要解释每个模块**

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。

通过上述处理，将原本有782650个三角面片构成，数据量约为80MB的模型文件。轻量化为仅有49395三角面片，数据量仅有约19MB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2252把座椅数据量约28.5MB。通过上述处理，系统仅保留了一把座椅作为重用样本，因此传输数据仅有10KB。

4.2**大规模参会人群的轻量化预处理**

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。

1.模型资源的分级复用。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）；第三级，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。

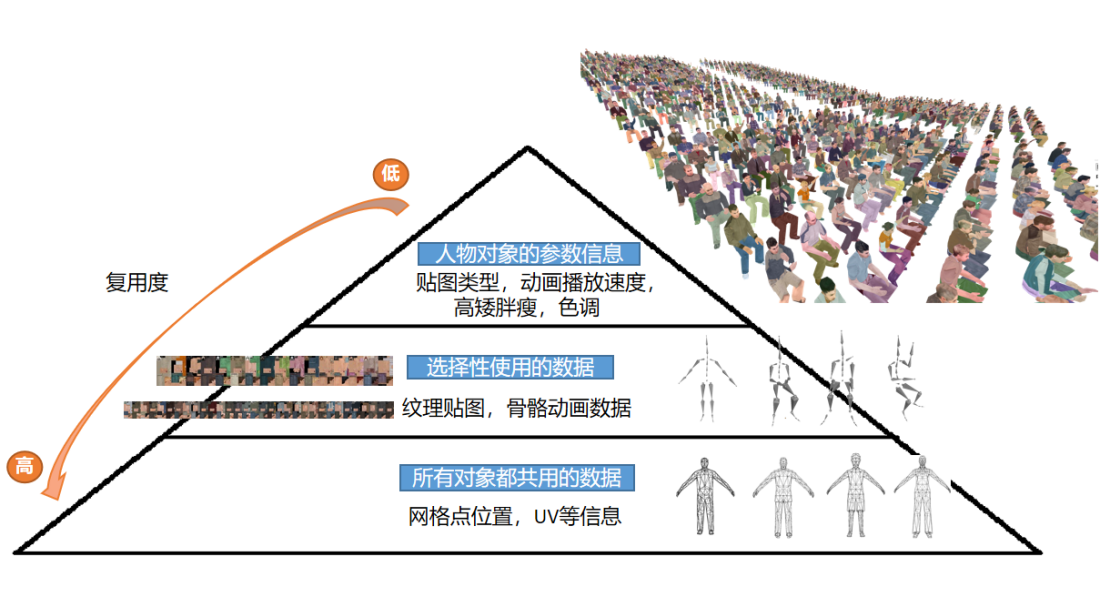


图 3 大规模参会人群的轻量级行为建模

2.增大临近人物对象间的差异性

（1）需求

为了使得人群渲染的多样性效果更加明显，我们希望使用相同头部贴图的模型距离越远越好。

（2）问题分析

可以将这一需求抽象为以下数学问题：

人和椅子位置相同，椅子的摆放为规则的矩阵，可以用一个长宽固定的矩阵A记录信息。头部贴图共用48种，可以用0-47这48给整数表示矩阵A中每个元素。

我们定义一个叫**分散度**概念，用来描述相似对象的分散程度。分散度的值为相同类型相同的元素间距离的最小值。我们要求一个最佳的人群设置方式A，使得分散度最大。

公式中的每个参数与变量都应该做说明。

我们以平均散度作为我们的目标函数: ，为总体参会人数， 为欧式距离，为用户 的行列坐标， 为距离用户 最近且使用相同类型元素用户的坐标。

（4）优化效果

我们进行了1000次不经过处理的测试，分散度f的值始终为1，既存在相邻的两个人物对象贴图相同。经过我们的处理后分散度约为5.831，所以在我们的场景中任意使用相同贴图的人物对象之间的最小距离是5.831，他们之间相隔4人以上。

请说明采用哪种方法优化？ 贪心算法， 动态规划，或者其他？

1. **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。由于有动作的物体往往更容易吸引人的注意，并且动作越剧烈越容易吸引人，所以我们在考虑构件传输优先级（即构件的重要性）的时候，应将物体的运动速度作为一个衡量指标，运动速度与传输优先级正相关，并将没有动画的静止部件视为运动速度为零。

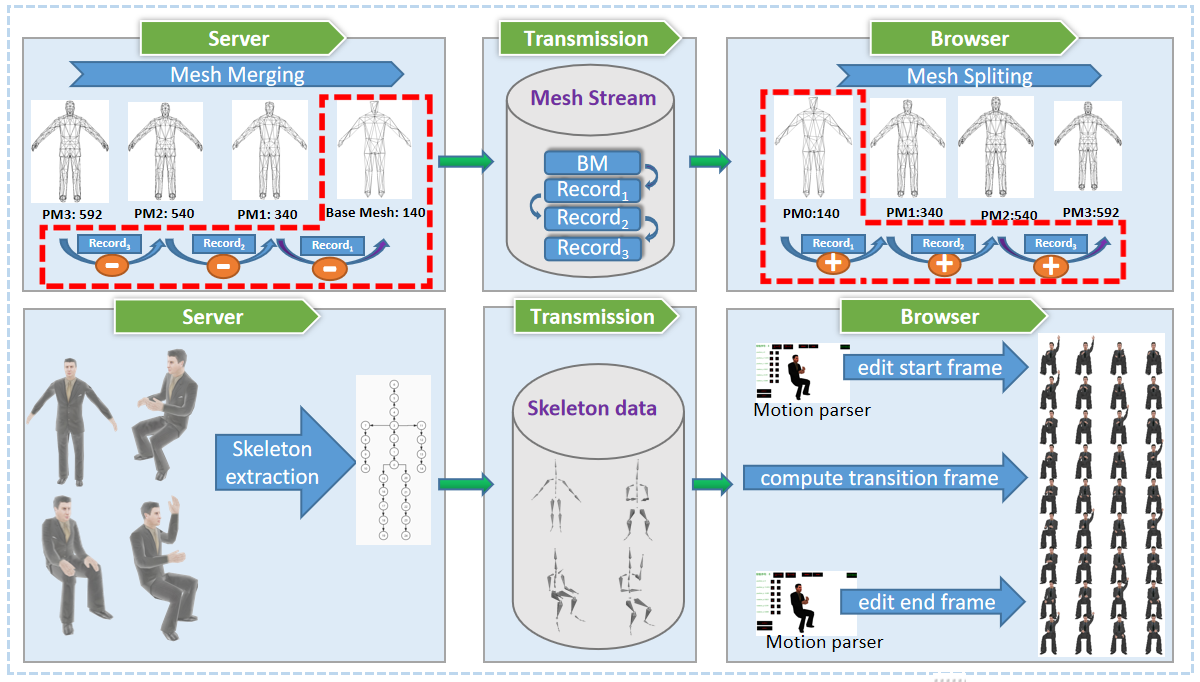


图 4 **细粒度化渐进式传输调度**

5.1轻量化内存管理

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。我们将人群作为一个整体，单个人物的操作通过人群对象提供的接口来进行。这样可以集中管理人群渲染所需的资源，单个人物对象只对应参数信息但不具有资源，只有整个人群对象才拥有资源。

人群对象管理的资源主要有3D模型资源、纹理贴图资源、骨骼动画数据、shader代码文件。对于3D模型资源，一个人群对象只有一个这种资源，即整个人群都对应同一个网格模型，如果要使用不同的网格需要创建新的人群对象，在项目中我们就是男性模型对应一个人群对象，女性模型对应另一个人群对象。一个人群可以对应多套纹理贴图资源以及多套骨骼动画，单个人物对象可以使用其中的任意一套贴图以及任意一个动画，并且身体的不同部位可以使用不同的贴图，如头部使用1号贴图下身的裤子使用2号贴图。由于动态文件和静态的shader代码是分别存放在两个文件中的，所以人群对象还要负责shader代码文件的管理。

为了方便将我们的人群轻量化处理技术应用于其它项目，我们进行了完整的方法封装和提供了丰富的接口，增强安全性和简化编程，使用者不必了解轻量化处理的具体实现细节，而只是要调用接口就可以将轻量化技术用于其它项目。我们提供的接口主要分为人物多样性设置接口和人物操控接口两类，多样性设置接口可以设置身体不同部位的贴图还可以调整人物的色调，人物操控接口可以操纵人物进行移动、旋转，设置人物的高矮胖瘦、动画的种类和播放速度。

5.2基于兴趣度的细粒度化处理

将3D场景资源拆分为小的组件，进行细粒度化传输，通过兴趣度分析各个部分的重要性，求出每个单元的兴趣的，按照兴趣度由高到低的次序来依次传输这些单元对应的资源，从而确定这些单元的传输优先次序。

兴趣度的定义：动画播放速度、填充度、重用度以及关注度的加权结果，综合反应了该模型对于用户视觉影响的贡献大小以及其文件传输的代价，计算公式如下。

ID=a×AS+b×FD+c×RD+d×AD

公式中a、b、c、d为权重。ID(Interest Degree)是兴趣度。AS(Animation Speed)是动画播放速度。填充度（Fill Degree, FD）指模型所占的包围盒体 。重用度(Reuse Degree, RD）指模型的实例构件数量与其文件大小的比值。关注度（Attention Degree, AD）反应了模型在单 位传输消耗下于用户对其可能的关注程度高低积与其文件大小的比值。

5.3 带宽自适应的渐进式传输调度

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 轻量级大规模会议场景在线渲染**

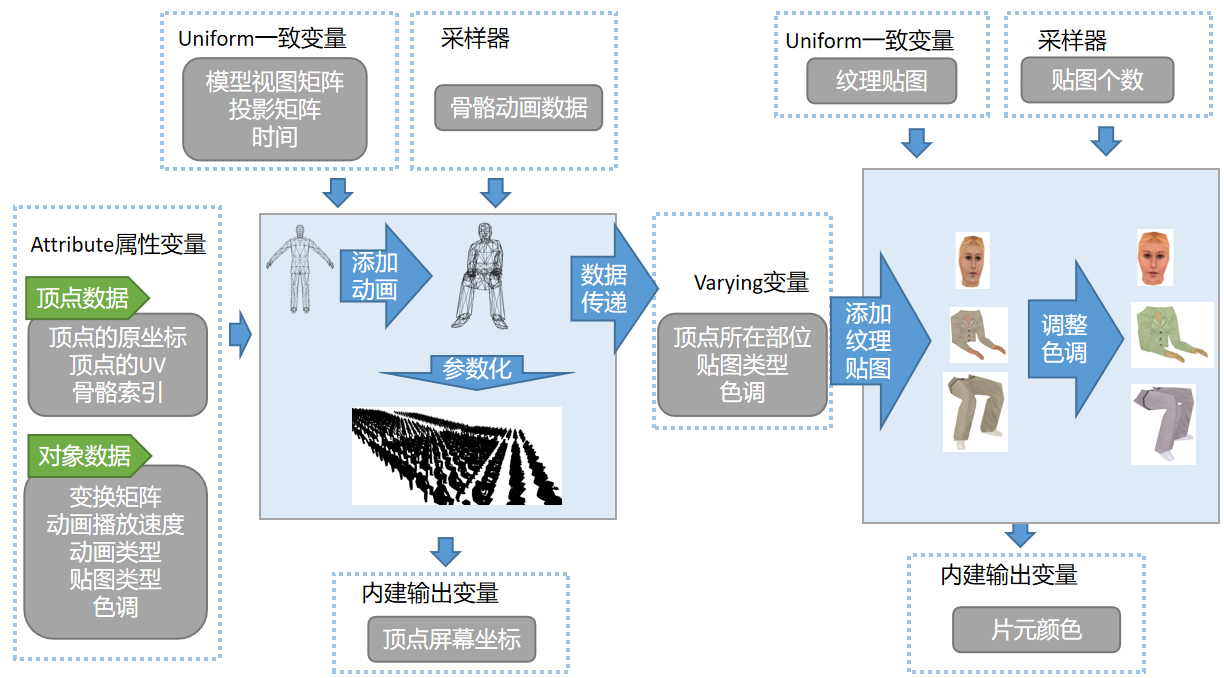


图 5 大规模人群的轻量化在线渲染

首先将模型划分为多个区域，每个区域可以进行不同的操作，这样就可以使得搭配更加多样化。如果为每个区域匹配不同贴图，就可以通过不同贴图搭配各种效果。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，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。

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。

项目中的优化处理：

（1）设计了专门用于存储骨骼动画数据的浮点数格式

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。

（2）利用人物贴图的对称性

大多数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。



7.实验结果与性能分析

|  |  |  |
| --- | --- | --- |
| 测试环境的硬件配置 | 台式机 | 笔记本电脑 |
| CPU | I7-10700U | I5-10210U |
| Memory | 16GB | 8GB |
| GPU | Nvidia GeForce RTX 2060 SUPER | Nvidia GeForceMX330 |
| OS | Windows 10  64Bit | Windows 10  64Bit |
| Networks | 4G Wireless  Network | 4G Wireless  Network |
| Browser | Chrome | Chrome |

表 1 测试环境的硬件配置

我们的优化方法可以将一部分场景渲染所需要的计算力分散给GPU，所以渲染性能有很大的提升。新技术的优化效果和GUP的硬件性能有很大关系。

|  |  |  |
| --- | --- | --- |
| PM的处理效果 | 低模 | 高模 |
| 基模的三角面片数 | 140 | 3039 |
| 原模型三角面片数 | 535 | 15990 |

表 2 PM处理效果

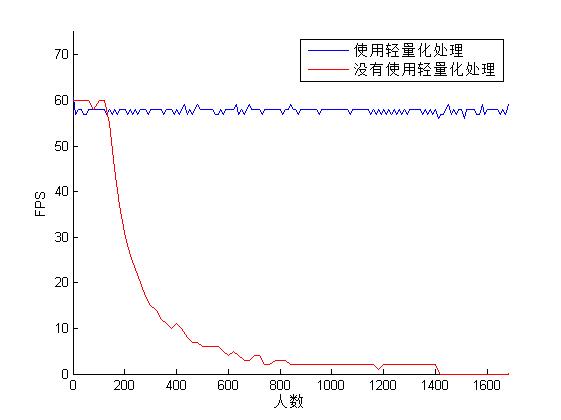


图6 人群的轻量化处理技术在项目中的效果， 解释FPS ()



图 7 同样帧率下能支持的在线虚拟参会的最大人数

从以上结果可以明显看出，新的方法优于先前的方法。虽然在人数较少时新的方法表现不是很理想，但随着场景中人数的增加，轻量化处理后的优越性就体现了出来，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。

【也请补充若干WebVR会议的截图，网址链接，英文简介】



列一个表格：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参会规模 | 预处理耗时 | 传输耗时 | 渲染延迟 | 总体延迟 | 重用效率(%) |
| 10 |  |  |  |  |  |
| 100 |  |  |  |  |  |
| 200 |  |  |  |  |  |
| 500 |  |  |  |  |  |
| 1000 |  |  |  |  |  |

8.结论

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。

在对动态资源的处理上，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资源的分级复用和通过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。

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]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.

[5] G. G. X.-G. H. Jing Sun, Yu-Shen Liu, IFCCompressor: A

content-based compression algorithm for optimizing Industry Foundation

Classes files, Automation in Construction 50 (2015) 1–15.

doi:10.1016/j.autcon.2014.10.015.

[6] J. L. G. Arthaud, Automatic semantic comparison of step product models,

in: Innovations in Design & Decision Support Systems in Architecture

and Urban Planning, Springer Netherlands, 2006, pp. 447–463.

doi:10.1007/978-1-4020-5060-2 29.

[7] S. H. Y. S. Ghang Lee, Jongsung Won, Metrics for quantifying the

similarities and differences between ifc files, Journal of Computing in

Civil Engineering 25 (2) (2011) 172–181. doi:10.1061/(ASCE)CP.1943-

5487.0000077.

[8] R. R. A. I. Le Zhang, Development of ifc-based construction industry

ontology for information retrieval from ifc models, in: Proceedings of

the 2011 EG-ICE Workshop, University of Twente, The Netherlands,

2011, pp. 6–8.

[9] M. W. M. G. J.-H. Y. Ge Gao, Yu-Shen Liu, A query expansion

method for retrieving online bim resources based on industry

foundation classes, Automation in Construction 56 (2015) 14–25.

doi:10.1016/j.autcon.2015.04.006.

[10] M. S. Hamid Laga, Michela Mortara, Geometry and context for semantic

correspondences and functionality recognition in man-made 3d shapes, ACM Transactions on Graphics (TOG) 32 (5) (2013) 150.

doi:10.1145/2516971.2516975.

[11] M. A. N. J. M. Youyi Zheng, Daniel Cohen-Or, Recurring part arrangements

in shape collections, Computer Graphics Forum 33 (2) (2014)

115–124. doi:10.1111/cgf.12309.

[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]Oğuzcan Oğuz, Ateş Akaydın, Türker Yılmaz, and Uğur Güdükbay. 2010. Emergency

crowd simulation for outdoor environments. Computers & Graphics 34, 2 (2010),

136–144.

[14]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).

[15]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.

[16]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.

[17]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.

[18]M Satyanarayanan. 2017. The Emergence of Edge Computing. IEEE Computer 50, 1(2017), 30–39.

[19]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.

[20]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.

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