

保密级： 保密期限：

北京邮电大学

硕士学位论文



题目： 基于视口预测的全景视频传输优化

关键技术研究

学 号： _____

姓 名： _____

专 业： _____

导 师： _____

学 院： _____

2020 年 02 月 25 日



**BEIJING UNIVERSITY OF
POSTS AND
TELECOMMUNICATIONS**

Thesis for Master Degree

**Topic: Research on Panoramic Video Transmission
based on Viewport Prediction**

Student No.: _____

Candidate: _____

Subject: _____

Supervisor: _____

Institute: _____

February. 25st. 2020

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

本人完全了解并同意北京邮电大学有关保留、使用学位论文的规定，即：北京邮电大学拥有以下关于学位论文的无偿使用权，具体包括：学校有权保留并向国家有关部门或机构送交学位论文，有权允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，有权允许采用影印、缩印或其它复制手段保存、汇编学位论文，将学位论文的全部或部分内容编入有关数据库进行检索。（保密的学位论文在解密后遵守此规定）

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

基于视口预测的全景视频传输关键技术研究

摘要

由于计算机科学和多媒体技术的不断创新,虚拟现实因其沉浸式构想式特性得到越来越多的关注,全景视频作为虚拟现实技术重要的组成部分也在休闲、教育和医学等领域得到了应用。但是,全景视频数据量大、时延敏感的特点对现有传输网络造成了巨大的压力。因此,如何在有限的网络带宽中最大程度地满足多个用户的观看体验至关重要。本文主要从全景视频视口预测和物理资源调度两方面来优化全景视频的传输。

近年来,为了降低传输全景视频时的数据量,基于视口自适应的全景视频传输方案得到越来越多的研究。该方案的关键前提就是视口位置的精准预测。我们提出一种基于用户历史观看轨迹的视口预测算法,首先根据当前用户的历史视口数据,使用长短期记忆模型初步预测出视口中心位置,然后结合其他用户的视口数据对初步预测结果进行调整以调高准确度。实验结果表明,我们提出的预测算法在准确度方面比起基准算法有了一定的提升。

另一方面,如何科学地调度带宽资源满足不同类型业务的用户需求也是研究重点。我们首先研究了 LTE 系统中几种常见的下行资源调度算法,考虑到这些算法在调度全景视频等实时业务时有一些不足之处,我们提出了一种使用 Q 学习来进行调度决策的改进算法,可以根据实时变化的网络状态在每个调度间隔自适应地选择最合适的调度算法。仿真实验表明我们提出的算法在吞吐量、丢包率和时延等指标上优于经典算法。

关键词 全景视频 视口预测 LTE 资源调度

RESEARCH ON PANORAMIC VIDEO TRANSMISSION BASED ON VIEWPORT PREDICTION

ABSTRACT

Due to the continuous innovation of computer science and multimedia technology, virtual reality has received more and more attention due to its immersive conception. As an important part of virtual reality, panoramic video has also been applied in the fields of leisure, education and medicine. However, the large amount of panoramic video and delay-sensitive features have put tremendous pressure on existing transmission networks. Therefore, how to maximize the viewing experience of multiple users under a limited network bandwidth is crucial. This paper mainly optimizes the transmission of panoramic video from the perspective of viewport prediction and physical resource scheduling.

In recent years, more and more researches have been made on the viewport-based adaptive transmission scheme for panoramic video. The key premise of this scheme is the accurate prediction of the viewport position. Therefore, we improve the viewport prediction algorithm based on the historical viewing trajectory of the user. First, based on the historical viewport data of the current user, we use the long-short term memory model to initially predict the coordinates of the center of the viewport. Then we combine the viewing trajectory of other users to adjust the preliminary forecast results. The experimental results show that the accuracy of the

proposed prediction algorithm is improved compared with the benchmark algorithm.

On the other hand, how to scientifically schedule physical resources to meet the needs of multiple users is also a research focus. We first study several classical downlink resource scheduling algorithms in LTE systems. Considering that these algorithms have some shortcomings when scheduling real-time services such as panoramic video, we propose an improved algorithm that uses Q-Learning to make scheduling decisions. The most appropriate scheduling algorithm is adaptively selected at each scheduling interval according to the network state. Simulation experiments show that our proposed algorithm is superior to classical algorithms in terms of throughput, packet loss rate and delay.

KEY WORDS: panoramic video, viewport prediction, LTE, resource scheduling

目录

第一章 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	3
1.2.1 全景视频视口预测研究现状	3
1.2.2 LTE 资源调度算法研究现状	5
1.3 主要研究内容	7
1.4 本文组织结构	7
第二章 基于历史观看轨迹的视口预测算法	9
2.1 视口预测相关技术	9
2.1.1 相关定义	9
2.1.2 常用预测模型	12
2.2 基于历史观看轨迹的视口预测算法	16
2.2.1 数据集介绍	16
2.2.2 LSTM 介绍	16
2.2.3 视口预测模型介绍	18
2.3 实验与结果分析	22
2.3.1 实验过程	22
2.3.2 评价指标	23
2.3.3 实验与结果分析	24
2.4 本章小结	27
第三章 基于 Q-Learning 的 LTE 资源调度算法	29
3.1 LTE 架构与资源调度	29
3.1.1 LTE 网络架构与协议架构	29
3.1.2 LTE 架构关键技术	31
3.1.3 无线资源调度	32
3.2 基于 Q-Learning 的资源调度算法	35
3.2.1 经典的资源调度算法	35
3.2.2 Q-Learning 介绍	36

3.2.3 DSQL 算法介绍.....	37
3.3 实验与结果分析	41
3.3.1 仿真工具介绍	41
3.3.2 仿真环境与参数	42
3.3.3 仿真结果与分析	43
3.4 本章小结	47
第四章 总结与展望	49
4.1 工作总结	49
4.2 未来展望	50
参考文献	51
致谢	55
攻读学位期间发表的学术论文	57

第一章 绪论

1.1 研究背景与意义

由于计算机科学和多媒体技术的不断创新，VR（Virtual Reality, 虚拟现实）技术开始逐步进入大众的关注范围。VR 技术构建的虚拟环境与真实环境真假难辨，用户从传统的以视觉为主的局限体验扩展到包含视觉、听觉、触觉甚至嗅觉的综合感官体验，在虚拟环境中体验到仿佛身处在真实世界的感觉。

VR 具有沉浸性、交互性和构想性的特点[1]。沉浸性是指当用户进入到虚拟环境时，听到的看到的体验到的感受与真实世界完全一致，这是衡量 VR 技术优劣的关键指标；交互性指用户能够与虚构世界互动。区别于传统的只有观感的用户体验，VR 技术通过力反馈装置、陀螺仪、跟踪器等多种传感设备使得用户可以跟虚拟环境进行交互。例如，用户可以在射击游戏中拿起枪支对目标进行射击，可以蹲起躲避敌人，且整个交互过程遵循基本物理学定律，与真实世界一致；构想性也被称作想象性，用户在虚拟环境中不仅能够体验到实际环境中的感觉，还可以自由丰富地创建出真实世界不可能发生的事情，激发用户的创新性。

VR 的概念可以追溯到 20 世纪 80 年代初，发展至今已经在影视休闲、教育学习和远程医疗等领域得到了广泛的应用。工业和信息化部表示，2018 年全世界内 VR 的产业规模已将近千亿元，同时中国的 VR 产业规模也在一直扩大，预计到 2021 年 VR 的市场规模将达到 554 亿元。工信部还表示将给予制造业和零售业等多方面的支持，助力 VR 技术在教育、商业、文娱等领域得到更多运用[2]。VR 技术巨大的发展市场使得人们不断对其进行探索与研究，其中最流行的应用方向之一的全景视频也逐渐成为人们争相研究的重点。

全景视频，又被称作 360 度视频，是一种用户可以随意转变观看方向的三维视频。传统二维视频的主要缺点是缺乏互动性，用户只能被动观看视频内容，无法像在真实世界中那样可以随意转动头部观看不同的场景画面，全景视频的出现则打破了这种僵局。目前在各大网络平台看到的全景视频主要由特殊的全景摄像机拍摄（全景摄像机通常由若干个普通相机组成，可以拍摄多个角度的画面），然后通过特征点检测、特征匹配和融合等一系列步骤将各个角度的视频画面拼接成 360 度得到的。用户在观看全景视频时需要佩戴特殊的硬件设备 VR 眼镜或者 HMD（Head Mounted Display, 头戴式显示仪），然后用户处在全景摄像机的位置

观看 360 度的视频画面。用户可以自由地转动头部，视频画面也会随着变化，如同在真实环境中体验的一样。

相对于一般的平面视频而言，全景视频更加贴近用户的眼睛而对视频分辨率有着更为苛刻的要求。即使在相同的单眼分辨率情况下，由于需要保留 360 度的画面信息，全景视频的数据量也要比普通视频的数据量高许多，一般为前者的 5~6 倍。同时，全景视频对网络系统的传输时延有着较为严格的要求，行业认为传输系统的 MTP (Motion To Photos Latency, 头动到画面延迟)，即转动头部和视频画面对应切换之间的响应时间必须控制在 20ms 以内[3]，否则由于生理原因，用户会感到头晕恶心从而造成极差的观看体验。因此，全景视频数据量大、时延敏感的传输特性对现有的无线传输网络造成了极大的压力，如何在现有传输网络中为多个用户提供高水平的全景视频业务成为了棘手问题。而在实际观看场景中，由于人眼角度的特性，在某一时刻，用户实际只能看到全景视频中的部分内容即 FOV (Field of Viewport, 视口区域)。如果以同等质量传输所有的画面信息势必会造成网络资源的巨大浪费，如何根据视口区域自适应传输全景视频内容也受到越来越多的关注。

除此之外，不断增长的用户数量使得接入无线网络的终端设备逐年增多，无线网络传输的业务类型也从传统的二维视频、音频及图文扩展到新兴的全景视频等等。不同用户业务的 QoS (Quality of Service, 服务质量) 参数不同，如全景视频这种实时业务对系统的传输时延有着较高的要求，而文本这种非实时业务则要求传输系统有着较低的丢包率来保证信息的完整度，如何合理利用有限的带宽资源来满足不同业务的需求也是全景视频传输优化极为关键的问题之一。LTE (Long Term Evolution, 长期演进) 系统基站侧的无线资源管理有着十分重要的功能，其中涉及的资源调度算法则是关键。所谓资源调度，就是通过一定方式计算小区内等待调度的业务的优先等级，然后根据等级顺序将带宽资源依次分配给业务，直到所有待调度业务都满足了传输需求或者带宽资源全被占用。通常而言，资源调度算法需要考虑网络时延、吞吐量、时延以及用户间的公平性等，一个高效的调度算法能够考虑多个方面的因素，使得综合效益最大化。因此，如何通过合理的调度与分配来有效利用短缺的带宽资源，为多用户提供高质量低时延的全景视频业务并且保证其他传统业务的质量，也是目前全景视频传输优化工作的重点与难点。

1.2 国内外研究现状

全景视频作为一种新兴的多媒体业务，涉及到多方面的关键技术，包括图像拼接、三维投影、自适应传输以及客户端的渲染重建等等。本文主要聚焦于全景视频的视口预测算法和底层物理资源调度两方面。

1.2.1 全景视频视口预测研究现状

目前全景视频的传输方案主要分为全部传输和自适应传输两种[4]。全部传输是类似于传统视频传输方案，即将全景视频完整的传输给用户，目前主流的流媒体平台（例如 YouTube）都是采用这种传输方式。这种传输方案可以保证用户看到全部的视频内容但是对网络负担极大，且会造成较多的带宽浪费。自适应传输方案则是针对视口区域进行差别传输，其中基于瓦片（tile）的传输方式被认为是传输全景视频的理想方案[5]。不同于传输整个画面，基于 tile 的方式以较高质量传输视口区域所在的 tile，而以低质量传输非视口区域的 tile 甚至选择不传输。这种传输方式可以很好的降低带宽需求，但很依赖准确的视口预测算法，一旦视口区域估计错误，将会导致用户观看区域内出现明显的低质量画面甚至是黑块区域从而严重降低用户体验，因此，如何准确有效地预测出用户的视口区域对全景视频的自适应传输有着极为重要的作用。

对视口进行预测，首先要明确的问题是视口是否可预测。文献[6]对视口中心的位置，即视点（viewpoint）数据进行了实验分析。文献主要研究表征视点位置的三个参数：偏航角（yaw）、俯仰角（pitch）和翻滚角（roll）的相关特性。首先分析了三个角度的累积分布图，实验结果表明偏航角的变化幅度最大，最难预测，其次是俯仰角；然后又分析了三个角度和各自差值的自相关函数，结果显示在 500ms 内有较强的自相关性，这表明视点数据是可以预测的。

目前全景视频视口预测的方法主要有基于运动轨迹和基于内容显著性检测两种[7]。

基于轨迹预测主要是根据该用户历史一段时间内的视口位置来预测接下来的视口位置，视口位置主要通过头部移动轨迹或者眼球移动轨迹来确定，而这些轨迹数据可以通过 HMD 的定位功能采集到。

由于具有时间相关性的运动轨迹从本质上可以看作是时间序列，因此很多时间序列预测的方法被应用进来，如滑动平均、线性回归和机器学习等。文献[8]对比了滑动平均、线性回归和加权线性回归三种预测算法分别预测 yaw、pitch 和

roll 三个角度的准确度。其中滑动平均复杂度最低,即使用历史窗口中所有视点数据的平均值作为下一时刻的预测值,当数据变化幅度较小时这种预测方法也可以获得不错的预测准确度。线性回归则是应用统计学中的回归模型来对历史一定时间内的用户视点数据进行拟合,然后通过拟合出来的函数模型获取下一时刻的预测值。加权线性回归则是给予历史数据不同的权重值,如越近的数据拥有越高的权重值,以此来获得不同的影响程度,一定程度上提升预测准确度。[8]的结果表明,线性回归与加权线性回归预测结果要好于滑动平均。当使用历史 1s 的视点数据预测接下来 1s 的视点数据时,加权线性回归的预测准确度高达 90% 以上,但是当预测未来长时间段的数据时准确度会显著降低,当预测接下来 2s 的视点时,预测准确度则不足 70%。文献[6]则对比了朴素法(即使用当前时刻的角度值当作下一时刻的预测值)、线性回归和神经网络预测未来 0.1s~1s 范围内的偏航角,实验结果证明神经网络的预测效果要好于前两个算法。

除了使用当前用户的视点数据外,一些研究还探索了跨用户的行为。文献[9]使用了结合 KNN (K-Nearest Neighbor, K 最邻近) 聚合的线性回归模型,基于当前用户的历史视点,使用线性回归模型预测出当前用户下一时刻的视点后,再结合观看同一视频的其他用户的历史数据,使用 KNN 来对预测结果进行纠正,进一步提升预测准确度。文献[10]对三个角度的轨迹进行建模,具有相似观看行为的轨迹被聚合在一起,并且为每个聚类计算出不同的轨迹函数。文献[11]则首次使用一种基于密度的聚类算法 DBSCAN,在服务器端先对用户进行分类,然后在客户端,通过 SVM (Support Vector Machine, 支持向量机) 分类器来预测用户的类别,最后预测该类别所对应的观看轨迹。

另外一类是基于内容显著性检测的预测算法。对视频而言,显著性表征了用户对视频不同内容的关注程度,一般来说,视频内容的显著性越强则说明用户对该内容越感兴趣,内容被观看的可能性也越大。近年来,基于深度 CNN (Convolutional Neural Networks, 卷积神经网络) 的显著性检测方法相比于传统方法可以取得较优的结果[12,13]。文献[14]训练了一个浅层网络进行显著性检测,并应用了转移学习生成一个深度网络;同样的作者在文献[15]中介绍了 SalGAN,这是一种通过对抗进行训练的深度网络;但是,这些网络模型都是应用在传统二维图像或者视频中,直接应用到全景视频时可能会造成中心偏差或者多目标混淆的问题,造成错误的检测。

近些年来一些学者开始关注面向全景视频的显著性检测问题。由于用户观看全景视频时更倾向于观看赤道附近的内容,一些算法就将该特性应用到显著性检测中。文献[16]将全景视频的每帧图像分割成若干个区块后给每一部分区块设置

不同的显著性权重；由于全景视频投影为二维视频的时候会存在边界失真的情况，文献[17]分别计算出几个投影平面的显著性图后计算平均值作为全景视频的显著性结果；文献[18]则是将全景图像映射为 6 个视口子图像输入到 CNN 网络中，输出子图像的显著性图，最后将 6 个显著性图映射到矩形平面后得到全景图像的显著性。

为了更准确地捕获未来视口与过去视口之间的非线性和长期依赖关系，文献[19]提出了针对固定视点的预测模型，该模型同时利用过去的视点位置和视频内容特征来预测接下来 n 帧中的视口轨迹。文献[20]提出了两种深度增强学习模型：首先使用一个基于视觉特征估计视频中每帧热图的离线模型，然后使用一个在线模型，根据过去观察到的头部移动位置以及离线模型得到的热图来预测头部运动。文献[21]使用 CNN 开发了一个视口预测模型，模型删除了池化层，并添加了更多的卷积层以增强非线性拟合能力。

在处理长时间内数据关系时常被使用的 LSTM(Long-Short Term Memory, 长短期记忆)模型也被逐渐应用在视口预测中。文献[22]使用 LSTM 编码视口区域扫描路径的历史记录，并将隐藏状态特征与视觉特征结合起来进行未来 1s 的预测，结果表明基于 LSTM 模型的预测优于几种基准方案。

1.2.2 LTE 资源调度算法研究现状

基于视口的自适应传输可以显著降低全景视频数据量，但在传统业务中引入时延敏感的全景视频业务后，如何在现有的无线网络中通过有效的资源调度来满足多用户不同业务的服务质量需求也一直是学术人员的重点关注领域，本文主要聚焦于 LTE 系统中下行链路的资源调度问题。

资源调度领域的相关研究能够追溯到 20 世纪 80 年代左右，最初学者们提出的算法较为简单，忽略了无线信道的传输状况，其中最为典型的是 RR(Round Robin, 轮询)[23]算法。RR 算法将资源轮流公平地分配给网络中的用户，而不考虑实时变化的信道条件，RR 算法可以给与用户平等的调度机会，但带宽利用率低，导致系统吞吐量较小；与 RR 算法形成对比的 Max C/I(Maximum Carrier to Interference, 最大载干比)算法[24]则是以系统吞吐量最大化为目标，只考虑用户信道条件的好坏，造成信道条件差的边缘用户一直得不到系统资源；结合 RR 和 Max C/I 算法的优点学者们提出了 PF(Proportional Fairness, 比例公平)算法[25]。但是上述这些算法都忽略了业务的 QoS 需求导致在实时业务的调度场景中表现较差。鉴于视频和语音通话这类实时业务需求的增加，研究人员们又提出了一些面向实时业务的调度算法，比如 EDF 算法(Earliest Deadline First, 最早截至

优先) [26], 该算法根据每个业务的最大时延对所有任务队列进行排序, 每次优先调度截止时间最小的业务, 保证了时延敏感业务的调度; MLWDF (Modified Largest Weighted Delay, 改进的最大权重时延优先) 算法[27]在 PF 算法的基础上同时衡量了分组数据的时延阈值和信道条件两方面的信息, 可以在系统吞吐量和实时业务丢包率之间取得较好的平衡。

由于经典算法性能的局限性, 大量改进的优良算法也被陆续提出。基于 PF 算法文献[28]对非实时业务和实时业务进行了区分, 对于实时业务, 改进的算法考虑了其时延特性, 保证了时延较为敏感的业务可以优先被分配资源得以调度; 文献[29]也在 PF 算法的基础上引入保障参数, 降低了实时业务的丢包率但提高了非实时业务的丢包率。

除了从网络系统方面对算法进行优化外, 研究人员也以 QoE (Quality of Experience, 用户体验) 为优化目标改进调度算法。文献[30]根据平均意见得分指标评估了 PF、EXP-PF 和 MLWDF 算法。实验结果表明, 最适合的下行调度算法是 MLWDF, 其端到端时延小于 50ms, 并且在可以接受的平均意见得分 (超过 3.5 分) 下运行最大数量的用户接入 (超过 50 个用户)。

为了更好地结合多方面的因素来达到高效调度的目的, 综合考虑多个协议栈参数的跨层调度也得到了进一步的研究。文献[31]协同无线协议栈的应用层、MAC 层和物理层参数提出了一种跨层优化方案, 最大化利用网络资源并提高用户的体验水平。仿真结果显示提出的跨层框架可以最大程度的提高用户的感知质量并且在用户的公平性方面得到了显著的改进。

目前完全针对全景视频的资源调度研究十分有限。文献[32]面向全景视频业务提出了基于视点反馈的跨层资源调度算法, 比起经典算法在视频缓存长度和播放卡顿等方面有了一定的性能提升; 文献[33]以多路径协作路由的方案来促进 5G 小型小区网络中的全景视频的传输优化; 文献[34]则设计了面向多用户的全景视频自适应传输框架, 并通过部署移动边缘计算来增强基站的功能。

在最近的研究中, 深度学习的相关方法也被应用在无线网络传输中。针对资源调度问题, 文献[35]使用 PF 算法计算用户的优先级, 然后基于 Q 学习 (Q-Learning) 在每个调度间隔根据用户的 CQI (Channel Quality Index, 信道质量指标) 来决定物理资源块的分配机制。

1.3 主要研究内容

综上所述，全景视频的准确预测是传输优化问题的重要前提之一，然而，基于运动轨迹的预测没有考虑视频内容对用户观看行为的影响，在长期预测场景中存在较大问题；基于视频显著性检测的算法复杂度不利用对时延比较严格要求的全景视频，同时显著性检测也未能考虑观看者在年龄、性别、兴趣爱好等方面的区别，在内容提取方面也存在一些误差。因此，本文在研究了视口预测的常见算法后对基于运动轨迹的预测算法进行了改进优化。

随着全景视频等视频业务在无线网络中的增加，如何科学调度现有的带宽资源来满足多用户多业务的需求也是亟待解决的问题之一。本文聚焦于物理层面的数据特性，考虑到现有的调度算法在整个调度期间只使用一个调度策略，不能很好地适应动态变化的网络情况，本文根据吞吐量、时延和公平性指标定义了若干种状态，然后基于 Q-Learning 算法来决定在每种状态下应该选择何种调度策略来达到系统的性能最优化。

1.4 本文组织结构

本文共分为四个章节，各章节安排如下：

第一章：绪论。该章节主要介绍了全景视频传输的研究背景和研究意义，针对传输框架中的视口预测和资源调度两方面介绍了国内外研究现状和研究内容，最后给出本文的组织结构；

第二章：基于历史观看轨迹的视口预测算法。本章介绍了视口预测的常见方案并分析了优劣之处，然后详细介绍了本章提出的基于历史观看轨迹的视口预测算法，最后阐述了视口预测算法的实验过程与详细的实验结果分析。

第三章：基于 Q-Learning 的 LTE 资源调度算法。本章首先分析了 LTE 网络的基础架构和关键技术，然后介绍了常见的资源调度算法与资源调度流程，在此基础上，本章提出一种基于 Q-Learning 算法的下行资源调度算法并详细介绍了该算法的设计思想。最后介绍了仿真平台与仿真环境，从系统吞吐量、丢包率和用户业务时延等方面对实验结果进行了分析与对比。

第四章：总结与展望。本章节对论文做了回顾总结与未来展望。

第二章 基于历史观看轨迹的视口预测算法

根据第一章的介绍，现阶段国内外视口预测主要分为两种方式：基于用户历史观看轨迹和基于视频内容。由于基于视频内容的预测算法时间复杂度较高，同时，不同年龄、性别的用户在观看视频时的感兴趣区域差异较大造成预测偏差，因此，本章只聚焦于基于历史观看轨迹的预测算法。首先介绍视口预测相关技术的理论基础，然后详细介绍基于当前用户视口数据的预测算法，并结合其他用户的视口数据辅助预测结果，最后，对比分析了其他算法与我们算法的预测准确度。

2.1 视口预测相关技术

视口预测作为全景视频传输优化问题的前提之一，已经出现了一些针对该问题的研究。为了方便后续算法的描述，本节首先介绍视口的相关定义，然后概述了现阶段的视口预测算法。

2.1.1 相关定义

(1) 视口区域

在给出视口区域的定义前，首先介绍人眼和 VR 眼镜（或 HMD）的特性。

人类眼睛单眼在水平方向最大可以观看到 150 度左右的画面，双眼在水平方向则可高达 188 度左右。但是双眼重合视区为 120 度左右，即在这个角度范围内观看到的物体才会呈现出立体感。超过水平视角 30 度之外的视角称为诱导视角，即我们常说的余光，在这些区域的事物人眼是不敏感的，如图 2-1（a）所示。而在垂直方向上，单眼感光区域约为 120 度，视觉敏感区为 60 度，如图 2-1（b）所示。

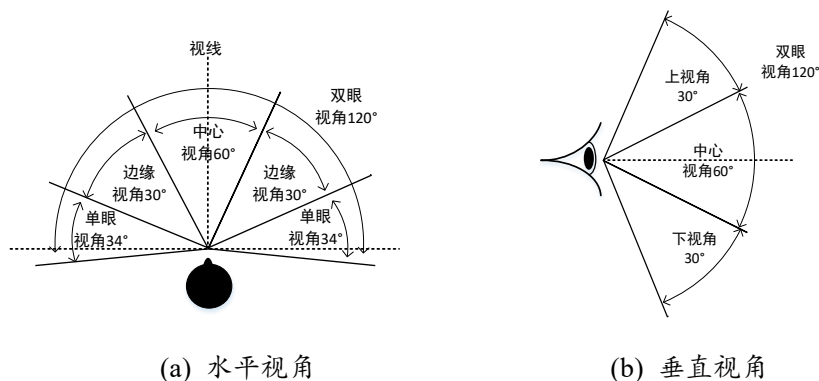


图 2-1 人眼视角区域

由于全景视频的特性，必须佩戴相关的观看设备才能观看，这些硬件设备在生产时都会有各自固定的视角区域。目前市面上入门级的 VR 眼镜的视角区域水平为 90 度，垂直区域为 90 度左右；像 Facebook 的 Oculus 和 HTC Vive 之类昂贵的头显设备的视角区域则为水平 110 度，垂直 90 度。

全景视频是一个横向呈 360 度，纵向成 180 度的三维视频，但由于人眼特性和观看设备的参数，在某个特定时刻，用户只能观看到全景视频的一部分内容，这部分就称为视口区域 FOV，如图 2-2 所示：

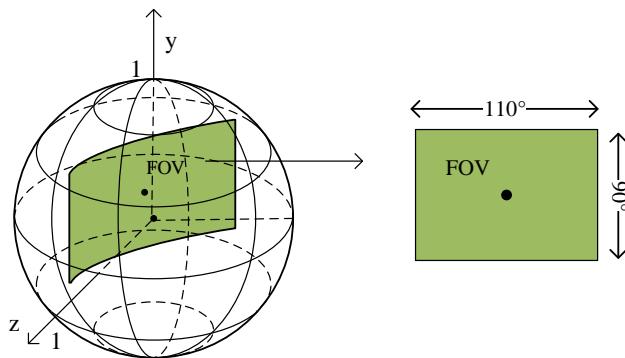


图 2-2 视口区域

为了方便后续的实验说明，本文中所指的视口区域特定为水平 110 度，垂直 90 度的矩形区域。

(2) 视点

视口区域是一个矩形区域，我们定义该矩形的中心点所在位置为视点。当视口大小固定时，只要知道视点位置即可获取到视口区域。我们给出可以描述视点位置的几种常用参数如下：

a) x、y、z 坐标

我们将全景视频构建成笛卡儿坐标系（Cartesian Coordinates）中的一个单位圆，视点位置即为圆上一点，可以使用空间三维坐标 x、y、z 来表示。

b) 经纬度

全景视频可以通过多种投影方式展开为二维平面视频，常见的投影格式有 ERP（Equirectangular Projection，等角投影）和 CMP（Cube Map Projection，立方体投影）等。我们使用国内外主流内容平台都使用的 ERP 方式将全景视频投影为二维平面视频，投影方式如图 2-3 所示。以地球仪展开为地图为例，ERP 就是假设将地球仪嵌入一个圆柱体内，然后在地球仪的球心处放置一个灯泡，灯打开后地球仪上的画面照应到圆柱上，再把圆柱水平展开为二维平面即地图。这种投影方式在赤道部分的误差较小，越往两级误差越大。

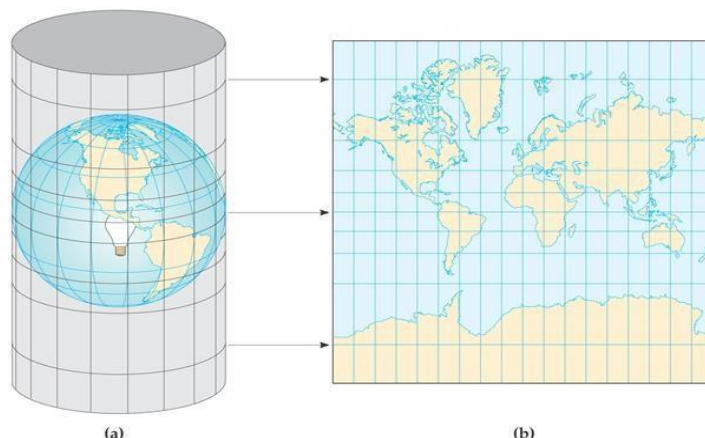


图 2-3 ERP 投影示意图

经投影展开后,三维球面上的视点就落在了平面上,我们使用经纬度来表示。经纬度与三维坐标可以通过下面的公式互相转化:

$$\begin{cases} x = R * \cos(phi) * \cos(theta) \\ y = R * \cos(phi) * \sin(theta) \\ z = R * \sin(phi) \end{cases} \quad (2-1)$$

其中, R 代表三维球体的半径, phi 为纬度, $theta$ 为经度。

c) 偏航角、俯仰角和翻滚角

用户在佩戴 VR 设备观看全景视频时,头部从某个位置转到另外一个位置的旋转量可以使用三个欧拉角 (Euler Angles) 来表示,即偏航角 (yaw)、俯仰角 (pitch) 和翻滚角 (roll),如图 2-4 所示。俯仰角为头部水平左右摆动的角度,向右转为正方向,范围为-180 度到 180 度;俯仰角为头部垂直上下摆动的角度,向上抬头为正方向,范围为-90 度到 90 度;翻滚角则是视线看向正前方,头顶在视线的正交平面左右摆动,以向右摆为正方向,范围为-90 度到 90 度。

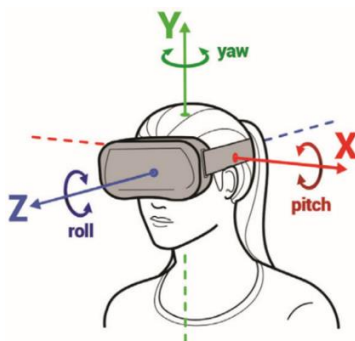


图 2-4 方向角示意图

(3) 视频 tile

现阶段,全景视频的传输主要有两种方式,一种是传输全部内容,一种是传输部分内容。第二种传输方案中,需要从空间上将全景视频帧分割成不同的区域,这些区域就称为 tile,如图 2-5 所示。我们将全景视频分成了 $16*8$ 个区块,其中

绿色区域为视口区域，则基于 tile 的传输方式只需要传输视口区域所覆盖的 6*4 块 tile，如图中橙色区域所示：



图 2-5 tile 示意图

2.1.2 常用预测模型

全景视频视口预测主要有两种方法，基于历史运动轨迹和基于视频显著性检测。考虑到后一种方法的时间复杂度不太适合持久预测，本文只探讨第一种预测算法。

基于历史运动轨迹的预测，通常是使用历史的视点数据来预测接下来一段时间的视点数据，本质上可以划分为时间序列预测问题。时间序列的预测是数据挖掘领域热门的研究方向，研究人员们已经提出了许多优秀的预测模型，主要可以分为基于统计分析、基于机器学习和基于神经网络，本节分别介绍这三种类别中的常见的预测模型。

(1) 移动平均

MA (Moving Average, 移动平均) 是一种简单的时间预测模型，主要思想是根据历史一段时间(窗口)内序列的平均值来预测接下来某一时刻的序列值，然后移动窗口继续预测，计算公式如下：

$$x'_t = \frac{1}{n}(x_{t-1} + x_{t-2} + \cdots + x_{t-n}) \quad (2-2)$$

其中， x'_t 表示 t 时刻的预测值， $x_{t-1} \sim x_{t-n}$ 为历史 n 个序列值。

当数据变化范围较小，整体呈现平稳状态时，该方法可以取得较好的预测准确度。在移动平均预测的基础上，又提出了加权移动平均法，即给历史窗口中的每个序列不同的权重值，计算公式如下：

$$x'_t = \frac{1}{n}(\delta_1 * x_{t-1} + \delta_2 * x_{t-2} + \cdots + \delta_n * x_{t-n}) \quad (2-3)$$

其中， δ_n 为第 n 个序列的权重值。使用该方法进行预测时，权重值的选择尤为重要，最为简单的是经验法和试验法，通常情况下，最近的序列值最能影响未来预测值，所以权重值应该大一些。

(2) ARIMA

ARIMA (Auto-Regressive Integrated Moving Average, 整合移动平均自回归模型) 是基于统计学的十分流行的一种预测模型。该模型有着完善的理论依据, 简单高效。

ARIMA 模型是在 ARMA (Auto Regression and Moving Average, 移动平均自回归模型) 的基础上进行 d 阶差分得到的。差分是将不平稳序列转为平稳序列的常用方法之一。所谓差分, 就是使用后一个序列值减去前一个序列值后的结果, 一阶差分即进行一次减法运算, 二阶差分则在一阶差分的结果上再进行一次减法运算, d 阶差分依此类推, 计算公式如下:

$$\begin{cases} u_t^1 = x_t - x_{t-1} \\ u_t^2 = u_t^1 - u_{t-1}^1 \end{cases} \quad (2-4)$$

其中, x_t 代表 t 时刻的序列值, u_t^1 和 u_t^2 分别代表一阶差分值和二阶差分值。

ARMA 模型则是由 AR (Auto Regression, 自回归模型) 和 MA 模型两部分组成, 表达式如下:

$$x'_t = \sum_{i=1}^p \phi_i \cdot x_{t-i} + \sum_{j=1}^q \theta_j \cdot \varepsilon_{t-j} + \varepsilon_t \quad (2-5)$$

其中, x'_t 是模型的输出, 即 t 时刻的预测值, $\sum_{i=1}^p \phi_i \cdot x_{t-i}$ 是 AR 模型的公式, ϕ_i 是系数, x_t 代表时间序列值; AR 模型主要是探索预测值与历史序列值之间的线性关系。而 MA 模型主要是探索预测值与预测误差的线性关系。由公式可以看出, ARIMA 模型的关键是确定三个超算数 p, q, d 的值, 差分中的 d 可以通过平稳性检验来确定, p 和 q 一般通过观察时间序列的自相关图和偏相关图[36]的截尾和拖尾来确定。

ARIMA 模型简单高效, 但仅适用于线性规律, 同时关键的 p 和 q 的值需要人工判断, 且参数一旦固定后就无法应对数据的变动。在实际的视口预测模型中, 由于用户观看不同视频时的观看行为不同, 需要对每一个视频都进行参数估计建立新的模型, 工程量大效率低下, 因此该模型并不适用于本文探讨的预测场景。

(3) 线性回归

随着机器学习技术的发展, 研究人员也开始使用一些机器学习的模型来进行预测。LR (Linear Regression, 线性回归)[37] 是机器学习中较为基础的一种算法。线性回归主要是找到数据之间的规律函数, 对于单变量而言, 线性回归拟合出一条直线, 方程可表示为: $y = a + bx$, 对于多个变量的话, 拟合出来的表达式可表示为:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n \quad (2-6)$$

其中, b_0 是常数项, b_1, b_2, \dots, b_n 为 x_1, x_2, \dots, x_n 对应的偏回归系数。

基于机器学习的预测过程主要是模型的训练, 一般是根据历史窗口的大小将数据集划分为多个子序列段 $x_{t-n} \sim x_t$, 每个子序列段后的 h 个值就是待预测的真实值 $x_t \sim x_{t+h}$, 模型输出的即为对应的 h 个预测值 $x'_t \sim x'_{t+h}$ 。在这种情况下, 我们通常使用真实值与预测值的 MSE (Mean Squared Error, 均方误差) 作为训练时的损失函数, 计算公式如下:

$$MSE = \frac{1}{n} \sum_{i=1}^n w_i (y_i - y'_i)^2 \quad (2-7)$$

其中, n 为预测值个数, w_i 为系数, y_i 为真实值, y'_i 为对应的预测值。

可以看出, 回归问题最后就转变成了优化问题, 即找到使损失函数最小的解, 划分好训练集与测试集之后训练模型即可。

除了 LR 模型外, 常见的机器学习模型还有 SVM 模型 (Support Vector Machine, 支持向量机) [38] 和 RF 模型 (Random Forest, 随机森林) [39]。相比于基于统计学的预测方法, 基于机器学习的方法可以处理非线性关系, 同时训练出来的预测模型具有普适性。

(4) MLP

由于深度学习的崛起, 一些神经网络的模型也被应用到时间序列的预测中来。与机器学习算法类似, 我们也可以通过训练神经网络模型来获取到历史序列值与预测值之间的规律关系。

MLP (Multilayer Perception, 多层感知机) [40] 是最基本的人工神经网络模型, 除了输入层和输出层外 MLP 还包括若干个隐藏层, 而最简单的 MLP 模型只有一层隐藏层, 不同层之间使用全连接, 如图 2-6 (a) 所示。

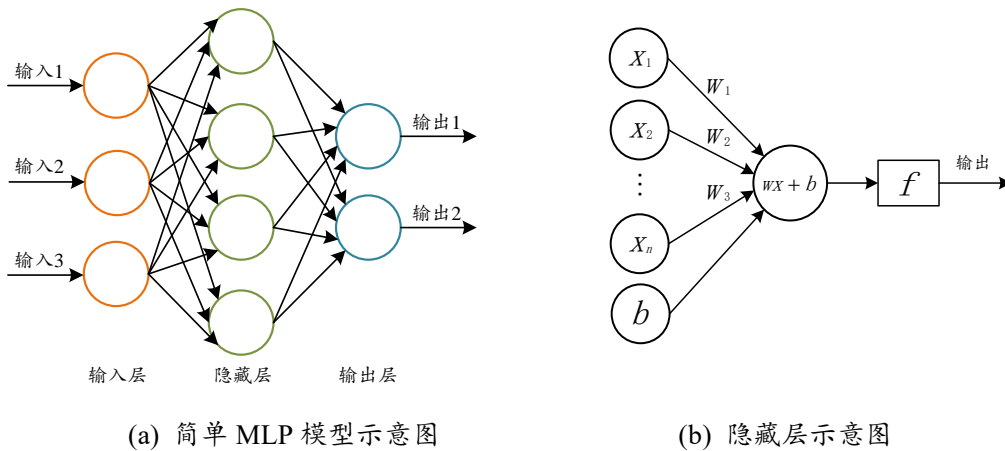


图 2-6 MLP 示意图

图 2-6 (b) 是隐藏层的示意图, 假设输入层有 n 个输入, 记为 x_1, x_2, \dots, x_n ,

则隐藏层的输出可表示为 $f(w_i x_i + b)$ ，其中 w_i 为权重，代表着两个神经元之间的连接程度，权重越大，可能性越大； b 是偏置，偏置的设置是为了正确分类样本； f 是激活函数，如果不使用激活函数，每一层的输出都和输入呈线性关系，激活函数在神经网络中引入非线性关系，可以将神经元的输出映射到有限范围内，如 $-1 \sim 1$ 或者 $0 \sim 1$ 。常见的激活函数有 Sigmoid 函数、Tanh 函数和 ReLU 函数等，根据实际环境来选择适合的激活函数。

MLP 被应用到了许多预测场景中。文献[41]对比了 ARIMA 模型和 MLP 模型在时间序列预测方面的性能差异，实验结果表明，短时间内 MLP 可以获得更高的预测准确度。文献[42]使用 MLP 预测了河流流量，文献[43]则使用 MLP 预测了金融领域的的数据。

(5) RNN

RNN (Recurrent Neural Network, 循环神经网络) [44]是以序列数据为输入，建模学习序列关系的深度神经网络。RNN 在一般神经网络的基础上，在不同层的神经元之间也建立了互连关系，结构示意图如下：

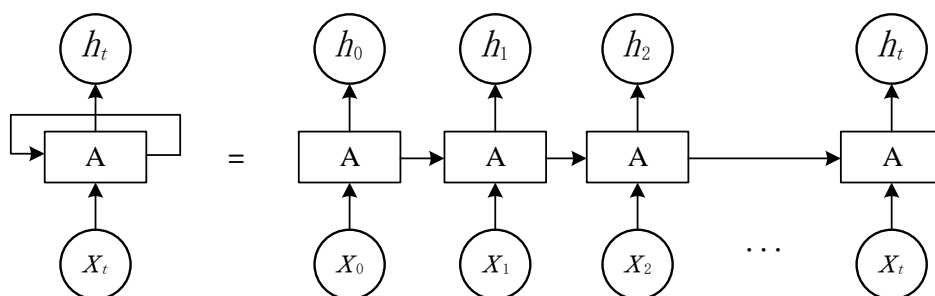


图 2-7 RNN 结构图

从图中可以看出，在计算当前的最优解时，前一时刻的隐藏状态信息 h_{t-1} 通过互联传递到了当前时刻，与当时时刻的序列值 x_t 共同影响输出结果。A 为神经元，假设激活函数为 \tanh ，则当前的隐藏状态 $h_t = \tanh (wx_t + uh_{t-1})$ ，其中 w 为权重值， u 为转换参数，因此，历史序列的部分信息可以被保存，即“记忆”，这是 RNN 可以处理序列之间依赖关系的关键所在。

比起统计学模型和机器学习模型，神经网络在处理非线性关系的序列数据问题中有着更为突出的能力。文献[45]证明一般的神经网络可以拟合任意一个非线性函数。虽然 RNN 有着强大的学习能力，但是在处理长期依赖（即时间序列中距离较长的数据）时存在梯度消失的问题。梯度消失是指在训练神经网络模型的过程中，梯度传递到某个隐藏层时已经减小到极小值甚至消失，造成权重无法更新模型无法训练的问题。

2.2 基于历史观看轨迹的视口预测算法

我们将基于历史观看轨迹的视口预测问题归类为时间序列预测问题,根据上述介绍可以看出,针对视口数据这种呈现非线性关系的时间序列进行预测,使用神经网络更能取得较好的预测准确度。目前的研究文献中主要提出了基于线性回归和一般神经网络模型的预测算法,但这些算法的预测准确度随着预测时间的增长下降明显,无法做到长期预测。

为解决上述问题,本文提出了改进的视口预测算法,首先使用 LSTM 模型分别预测当前用户视口中心位置的经度和纬度坐标,然后结合其他用户观看同一个视频时的观看轨迹,计算出最终视口区域所落在的 tile 位置。

2.2.1 数据集介绍

视点数据集的合理选择是预测算法成功的重要前提。由于视频内容、用户特性对用户的观看轨迹影响很大,因此合理的数据集应该包含各种类型的全景视频,观看视频的用户也应该分散到各个群体。

我们分析了现有公开的几个视点数据集[46,47,48],最终使用 wu 等人采集的全景视频用户头部移动数据集[48]。该数据集为 48 个用户观看 18 个全景视频的头部移动数据,48 名用户包括 24 名女性和 24 名男性,覆盖了各个年龄范围,18 个全景视频则包含了演出、体育、电影、脱口秀和纪录片 5 种最流行的类别。

18 个全景视频的时长从 2 分 44 秒到 10 分 55 秒不等,帧率为 30fps,而头部移动采集设备每秒可以采集 60~80 个数据,且每两个采样点之间的时间间隔有轻微的随机性。为了后续数据分析的便捷性,本文使用线性插值法将原始数据下采样为每秒 30 个数据,与视频的帧率保持一致。最后我们将原始数据转换为经度和纬度值,并分别将经度的取值范围映射到 0~360 度,纬度的取值范围映射到 0~180 度。

2.2.2 LSTM 介绍

由于后续的视口预测算法是基于 LSTM 模型的,本节先介绍一下 LSTM 的相关理论知识。

通过前面的介绍可知,RNN 在对长距离的序列数据建模时会出现梯度消失的问题,为了应对这个问题文献[49]中提出了 LSTM 模型。

LSTM 是 RNN 的一种变体。传统 RNN 的神经元只有一层 tanh 层,而 LSTM 的神经元引入了门函数来控制信息流,以规避因为过多的记忆信息而造成梯度消

失。图 2-8 为 LSTM 结构的示意图，其中，每个绿色区域称为一个单元（cell），中间的 cell 表示了 t 时刻的数据 x_t 输入后的操作路径， σ 代表 *sigmoid* 激活函数， \tanh 代表 *tanh* 激活函数， \times 为点乘操作， $+$ 为加法操作， x_t 为输入值， h_t 为隐藏层的输出值。

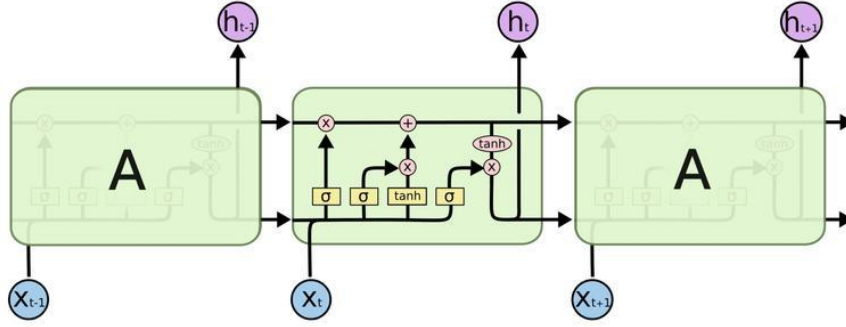


图 2-8 LSTM 结构示意图

LSTM 的门函数有三种，即输入门，输出门和忘记门，其中忘记门是 LSTM 设计思想的精髓所在，三个门的主要功能如下：

(a) 忘记门

忘记门在当前时刻使用 *sigmoid* 函数对输入数据和上一时刻的隐藏状态进行取舍，*sigmoid* 函数的输出范围为 0~1，当输出接近 0 时代表丢弃（忘记）该信息，接近 1 是代表使用（记忆）该信息，表达式如下：

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2-8)$$

其中 f_t 为忘记门的输出， w_f 和 b_f 分别代表权重和偏置。

(b) 输入门

输入门用来更新 cell 的状态。首先输入当前信息 x_t 和上一时刻的隐藏状态 h_{t-1} ，根据 *sigmoid* 的输出值来决定更新多少信息，根据 *tanh* 来压缩数值调整网络，最后将两者的输出相乘来决定哪些信息是重要的需要被记忆的，表达式如下：

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2-9)$$

$$C'_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2-10)$$

最后，将忘记门的输出与上一时刻的 cell 状态相乘，加上需要被记忆的信息，就控制了当前信息和历史信息的忘记和记忆，因此当前 cell 的状态可表示为：

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t \quad (2-11)$$

(c) 输出门

输出门给出最后的输出值。同样的，将当前信息 x_t 和上一时刻的隐藏状态 h_{t-1} 输入到 *sigmoid* 中，将当前的 cell 状态 C_t 输入到 *tanh* 中，最后将两者的输出值相乘，即得到了当前的隐藏状态 h_t 。表达式如下：

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (2-12)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (2-13)$$

基于 LSTM 对时间序列进行预测的过程与一般神经网络的过程一样，即划分训练集与测试集，对模型进行训练，训练结束后即可进行预测。

2.2.3 视口预测模型介绍

由前面的分析可知，当视点位置确定时，视口区域也可以确定，因此，本节聚焦于视点的预测，根据历史的经纬度数据来预测未来一段时间内的经纬度数据。

假设 x_t 和 y_t 分别为 t 时刻的经度和维度值， x_0 和 y_0 为用户初始的经纬度， $x_{t_1:t_2}$ 和 $y_{t_1:t_2}$ 分别为 t_1 到 t_2 时刻之间的经纬度值，即 $x_{t_1:t_2} = x_{t_1}, x_{t_1+1}, x_{t_1+2}, \dots, x_{t_2}$ 和 $y_{t_1:t_2} = y_{t_1}, y_{t_1+1}, y_{t_1+2}, \dots, y_{t_2}$ 。在 t 时刻的时候，用户的头显设备记录下来从初始值到 t 时刻的全部的经纬度值 $(x_{0:t_1}, y_{0:t_1})$ 。视点预测即使用历史窗口 hw 中的若干经纬度值 $x_{t-hw:t}$ 来预测接下来 pw 时刻的经纬度值，记为 $(x'_{t:t+pw}, y'_{t:t+pw})$ ，其中 pw 为预测窗口，然后移动历史窗口继续预测，直至视频观看结束。文献[7]认为，视点的几个特征维度之间的自相关性大于互相关性，因此本文也采用独立预测经纬度的方式。考虑到经纬度序列的时间特性和非线性特性，我们引入 LSTM 模型进行长期时间的预测，历史窗口的数据即为模型的输入，而模型的输出则为经纬度的预测值，模型结构图如下所示：

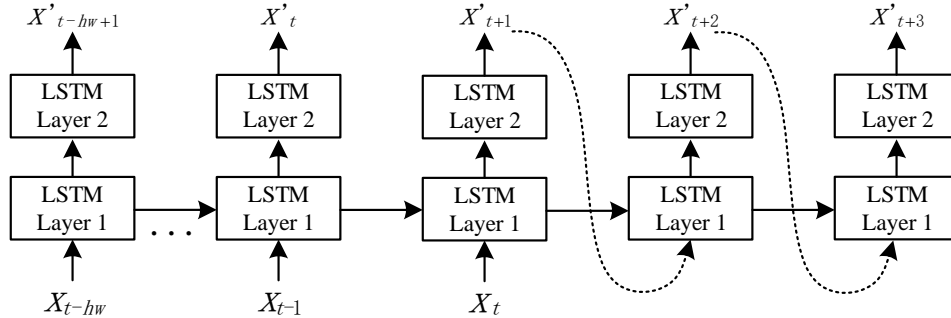


图 2-9 基于 LSTM 的预测模型结构图

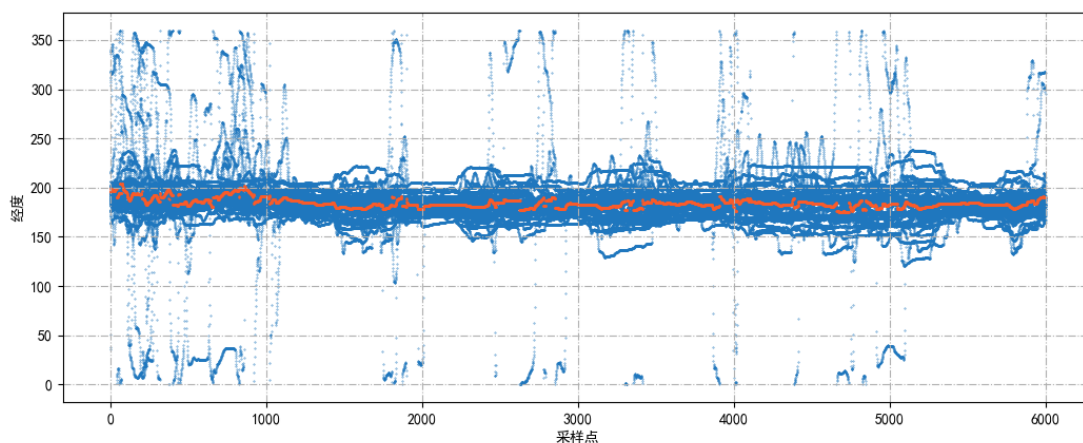
从图中可以看出，在 t 时刻可以根据历史的一段数据预测出下一个时刻的数据，表示为 $x'_{t+1} = h(x_{t-hw}, x_{t-hw+1}, \dots, x_t)$ ，其中 $h(\cdot)$ 为 LSTM 模型的输入输出函数；为了继续预测接下来的数据值，预测值 x'_{t+1} 将作为额外的输入值，组成新的历史数据，则 $x'_{t+2} = h(x_{t-hw+1}, \dots, x_t, x'_{t+1})$ ，依次类推，直至预测出需要长度的数据值。

可以想见，随着预测时间的增加，预测误差将会逐渐增大，为了进一步提高预测准确率，我们参考基于内容显著性检测的视点预测方法。该方法根据视频内

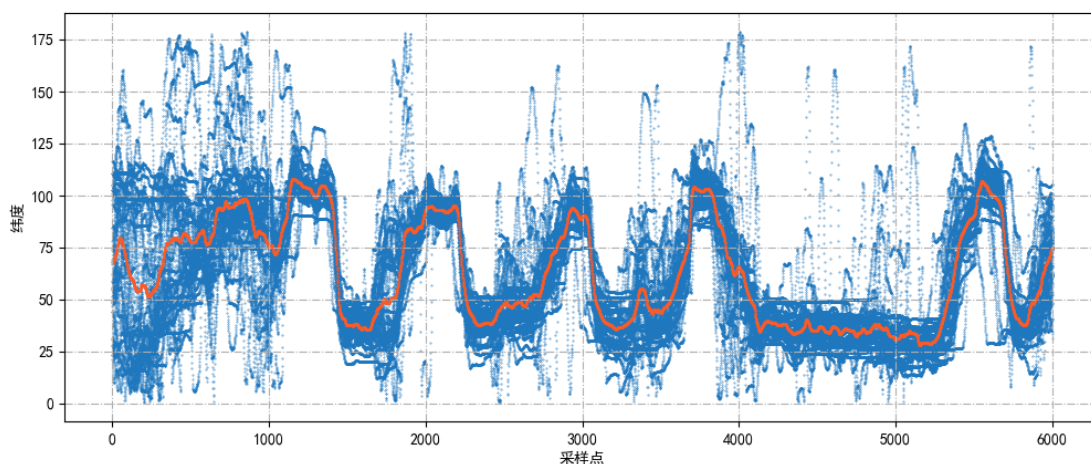
容的显著性来辅助预测，由于视频各个区域内容的显著性不同，人们在某个时刻观看某个区域的概率也不相同。当该区域视频内容较为显著性时，用户观看的可能性也较大，所以，多个用户的观看轨迹可以从侧面反映视频内容的显著性。

为了验证这一假设，我们从数据集中随机抽取了 2 个视频文件，按照用户观看时长分别截取了 6000 和 5000 个采样点，绘制了 48 名用户观看这两个视频的轨迹散点图，对应的经纬度散点图如图 2-10 和图 2-11 所示，其中蓝色散点为各个用户的观看位置，红色散点为所有用户数据的平均值。

从图 2-10 中可以看出，视频开始时，由于大部分用户第一次观看全景视频，对全景视频体验感到新奇，所以会随意转动头部以探索更多的画面，用户观看位置较为分散。而当新鲜感降低完全沉浸在视频内容中后（图中采样点大概在 1000 左右时），用户会密切关注视频的显著性内容，所以几乎所有的用户都保持着相同的观看轨迹，用户的视点分布十分集中，这种情况下，其他用户的观看行为对当前用户有着很强的参考价值。

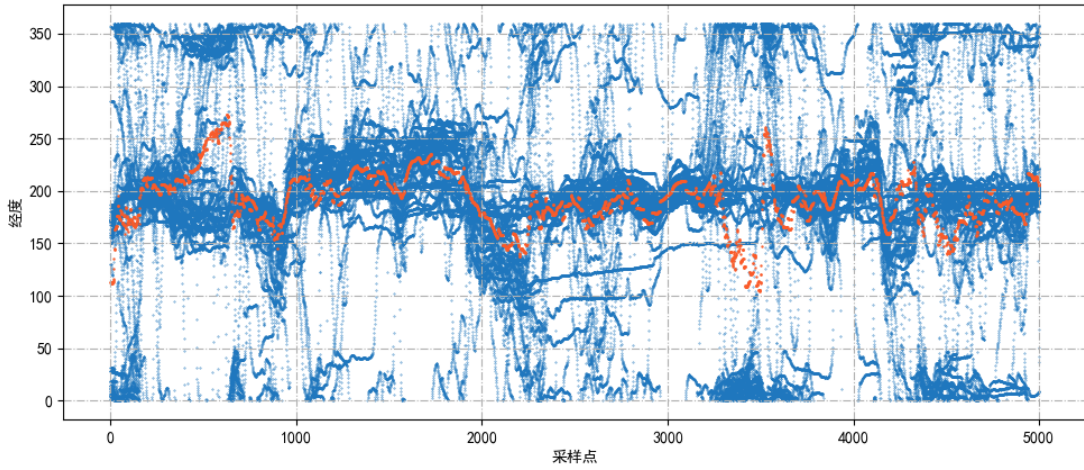


(a) 多用户的经度轨迹

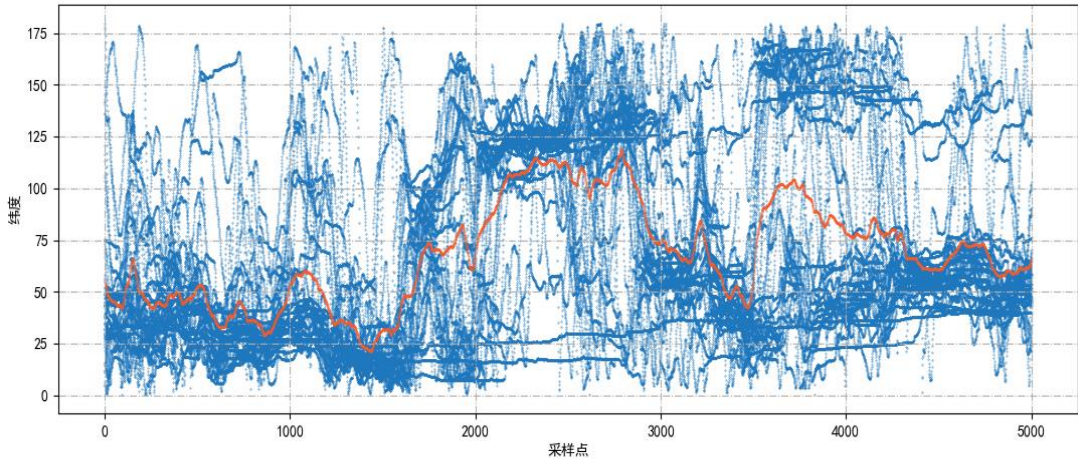


(b) 多用户的纬度轨迹

图 2-10 多用户观看视频 1 的轨迹图



(a) 多用户的经度轨迹



(b) 多用户的纬度轨迹

图 2-11 多用户观看视频 2 的轨迹图

图 2-11 显示的用户观看轨迹相对图 2-10 的轨迹来说杂乱些, 说明视频 2 中可能存在多个显著性区域。经度轨迹图中有一些看似波动很大的散点 (从 0 度波动到 360 度), 这主要是因为有些用户在 0 度经线 (也是 360 度经线) 附近观看, 经平面投影后呈现出图中所示的现象。虽然轨迹杂乱一些, 但超过一半的用户也会倾向观看同一位置, 这种情况下其他用户的观看轨迹对当前用户有着一定的参考价值。因此, 我们认为不管哪种全景视频其他用户的观看轨迹都与当前用户的观看轨迹存在一定的相似性。

基于上述分析, 使用 LSTM 模型初步预测出当前用户的视点数据后, 假设视频服务器存储了观看同一视频的其他用户的视口数据的前提下, 我们结合其他用户的观看行为, 对初步预测出来的数据进行调整, 以提高预测准确度。

为了衡量其他用户历史数据的重要性, 我们引入基于 tile 的传输机制, 不再只预测视点的经纬度值, 而是预测视口区域所在 tile。我们将视频每帧划分为 16×8 块 tile, 记第 i 行 j 列的 tile 的观看次数为 c_{ij} , 初始值设为 0。首先, 我们根据当前

用户的历史数据分别预测出接下来的经纬度值，如第 2.1.1 介绍的，根据经纬度值即可确定出视口区域以及视口所覆盖的 tile 块，这些区域每块 tile 的观看次数在初始值上增加 c_p 。然后根据其他用户在同一时刻的视口区域位置，将覆盖的 tile 的观看次数依次加 c_o ，最后，每个 tile 的观看次数即为两者之和，计算公式如下：

$$c_{ij} = \alpha \cdot c_p + \sum_{k=1}^n c_o \quad (2-14)$$

其中， α 取值 0 或 1，表示根据 LSTM 模型预测出来的区域是否覆盖该 tile， n 为视口区域覆盖该 tile 的其他用户的个数。

根据前面对其他用户观看轨迹的分析可知，不同的全景视频场景下，其他用户的轨迹与当前用户轨迹的相似程度不同，因此， c_p 和 c_o 取值的比例关系代表了其他用户数据的参考性。例如当用户轨迹比较分散时，可以设置 $c_p = 1, c_o = 1$ ，即我们认为最终预测的视口区域由 LSTM 模型预测出来的视口区域和其他用户的视口区域以相同的重要性共同确定，而当用户轨迹十分集中时，我们应该要适当提高 c_p 的比重，例如可以设置 $c_p = 5, c_o = 1$ 。另一方面，考虑到 LSTM 预测出来的视口数据的准确度会随着预测时间 pw 的增加而降低， c_p 的取值也应该随着 pw 的变化而变化，为了减小后续计算的复杂度，我们定义 c_p 和 c_o 的计算公式如下：

$$c_o = 1 \quad (2-15)$$

$$c_p = \left\lfloor \delta * \frac{1}{pw} * N * c_o \right\rfloor \quad (2-16)$$

其中， N 为其他用户的总个数， δ 为系数，可以根据实验结果对 c_p 的取值进行调整以提高预测准确度， $\lfloor \cdot \rfloor$ 为向下取整函数。

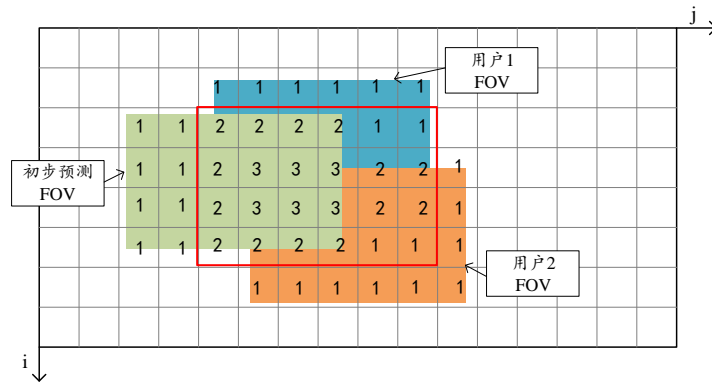


图 2-12 tile 观看次数示意图

图 2-12 为视频一帧的观看次数示意图，我们将视频帧划分为 16×8 块 tile，视口区域占 6×4 块 tile，设置 $c_p = 1$ 。我们假设共有 3 名用户观看该全景视频，绿色区域为使用 LSTM 初步预测出来的当前用户的视口位置，蓝色和橙色区域为其他 2 名用户同一时刻的的视口位置，按照公式 2-14 计算出所有 tile 的观看

次数后，我们可以获得一个二维数组，然后按照移动窗口的思想，遍历整个二维数据，计算出 $6*4$ 块 tile 的和最大区域即为最终预测的视口区域所占 tile，如图 2-12 中红色边框所框出来的区域所示。

2.3 实验与结果分析

本节介绍视口预测算法的实验过程和结果分析，并且介绍衡量预测算法准确度的两个评价指标，最后对比我们提出的预测算法与其他基准算法在两个评价指标下的预测准确度。

2.3.1 实验过程

根据之前的介绍，我们对经纬度坐标值分别预测，两者的预测过程一样，以纬度预测为例，主要包含以下几步：

（1）数据处理

我们首先对原始的纬度序列进行归一化处理，归一化可以将数据分布映射到一个确定的区间上，提高模型的收敛速度。由于我们经纬度的范围均处理成了正数，所以我们使用线性函数转换方法将纬度特征映射到 $0\sim 1$ 的区间，线性函数转化方法可以在 python 的 sklearn 包中调用 preprocessing 库中的 MinMaxScaler 类实现。

归一化处理后，我们需要将时间序列转化为监督学习数据。为了验证我们的算法在长时间预测方面的准确度，使用历史 1s 的 30 个纬度数据来预测接下来 1~6s 的纬度，即将历史窗口 pw 的数据作为监督学习中的输入 x ，将预测窗口 hw 的数据作为监督学习中的输出 y ，并通过移动窗口的方式构造出 n 组样本数据，然后将数据格式转化成 LSTM 模型期望的 3 维格式 [n 组, 时间步长 30, 特征值 1]。

为了有足够的数据进行学习，我们将所有用户的纬度数据均进行上述处理，产生 $\{n_1, n_2, \dots, n_{48}\}$ 组三维数据，然后将观看同一视频的所有用户的处理数据拼接起来作为我们最终的数据集，采用交叉验证的方式，取前 80% 的数据作为训练集，后 20% 的数据作为测试集。

（2）生成 LSTM 模型

我们调用 Keras 库定义了由 2 层 LSTM 共 128 个神经元构成的网络结构，采用均方误差 MSE 作为损失函数，ADAM 作为优化函数（比起传统的优化算法 SGD, ADAM 有着更快的收敛速度和更低的错误率），采用 ReLU 作为激活函数。设定好损失函数和优化函数后，调用 compile() 函数编译模型，编译好模型后调用

fit()函数来训练模型，该网络模型采用参数分享的反向传播算法进行训练。

在模型的训练过程中，因为训练数据量巨大，没办法将全部数据一次性输入模型进行计算，因此需要设定不同的周期（epoch）和批次（batch）。一个 epoch 是指全部数据训练并输出一次。当一个 epoch 的数据过多给计算机造成计算压力时，就需要分割数据，分割后的每块数据称为 batch，然后将数据一块一块地传递给计算机进行计算。而在神经网络的训练过程中，只传递一次完整的数据集是完全不够的，我们需要将有限的数据集多次传递给神经网络，多次更新权重值，使得拟合曲线从欠拟合到拟合。本节最终设置 epoch 为 20，batch 为 400。

（3）预测数据

当模型训练完毕后，便可调用 predict()输入测试数据进行预测了，预测出来的数据区间也在 0~1 之间，使用 MinMaxScaler 类再将数据转换到原始区间，即可得到预测的纬度值。我们将预测的纬度值跟同一时刻真实的纬度值对应起来存储在 csv 文件中，用于后续准确度的计算。

通过上述几步分别预测出经度和纬度值后，可以根据经纬度计算出视口所在位置以及对应的 tile 块。然后我们将同一视口其他用户的经纬度数据也按照历史窗口和预测窗口的大小划分成若干组，与预测数据的时间对应，并计算出每个用户视口所在 tile，按照公式 2-14 更新 tile 数组。最终我们获得一个 16*8 的数组，以 6*4 的窗口去遍历该数组，计算出窗口中所有 tile 值的总和，总和最大的窗口即最终预测出来的视口区域。如果存在多个最大值，我们认为这种情况是无法区分哪个最大值更准确，因此就以最后一个最大值作为输出。同样的，我们将预测视口所占的 tile 与同一时刻真实视口所占的 tile 对应起来，存储在新的 csv 文件中，用于后续的数据分析。

2.3.2 评价指标

为了对比我们提出的算法与其他基准算法的预测准确度，本节采用了两种评价指标来计算预测准确度：经纬度绝对差值和 tile 重叠比。

（1）经纬度绝对差值

为了衡量 LSTM 模型在预测经纬度时的预测准确度，我们使用同一时刻真实的经纬度与预测的经纬度的绝对差值来代表预测误差。由于全景视频的球体特性，例如在经度上，-150 度和 150 度直接取差值为 300 度，但实际两者只差了 60 度，因此，我们使用文献[6]中定义的角度差值的计算方式，公式如下：

$$\Delta(\theta_1, \theta_2) = \text{mod}(\theta_1 - \theta_2 + 180^\circ, 360^\circ) - 180^\circ \quad (2-17)$$

其中， $\Delta(\theta_1, \theta_2)$ 代表角度 θ_1 和 θ_2 的差值， $\text{mod}(\cdot)$ 为取模运算。基于此我们定

义预测时刻经纬度的绝对差值如下：

$$e_{t+pw}^x = |\Delta(x_{t+pw} - x'_{t+pw})| \quad (2-18)$$

$$e_{t+pw}^y = |\Delta(y_{t+pw} - y'_{t+pw})| \quad (2-19)$$

其中， e_{t+pw}^x 和 e_{t+pw}^y 代码预测时刻的经纬度绝对差值，绝对差值越大说明预测越不准确。

(2) tile 重叠比

由于我们提出的预测算法最终预测的是视口区域所占 tile 的情况，所以我们使用同一时刻预测的视口所占 tile 在真实视口所在 tile 中的重叠比作为最终的预测准确度指标，如真实视口所在 tile 编号为{(00,01,02,03), (10,11,12,13)}，预测视口所在 tile 编号为{(02,03,04,05), (12,13,14,15)}，则 tile 重叠比为 4/8=0.5。在实验中我们把该值作为预测算法的准确度。

2.3.3 实验与结果分析

为了对比预测准确度，我们研究了两种回归方法，线性回归-LR 和非线性回归-MLP，这两种回归法将时间和经纬度值视为独立变量，目标是根据历史窗口的数据拟合出时间和序列值的回归函数 $f(t)$ ，然后根据预测窗口得出预测数据 $y = f(pw)$ 。我们对经纬度分别建模，对于 MLP，我们采用了含有 64 个神经元的模型结构，同样使 ReLU 作为激活函数，ADME 作为优化函数。最后，我们使用移动平均算法作为最基准的算法。

首先，考虑到只选用一个视频可能造成预测结果的随机性，我们选取了数据集中三种代表性视频，分别对比了几种算法在预测经纬度方面的准确度。我们定义经度的绝对差值 e_{t+pw}^x 小于 20 度为预测准确，纬度采用同样的判定方法，实验结果如图 2-13~2-15 所示。

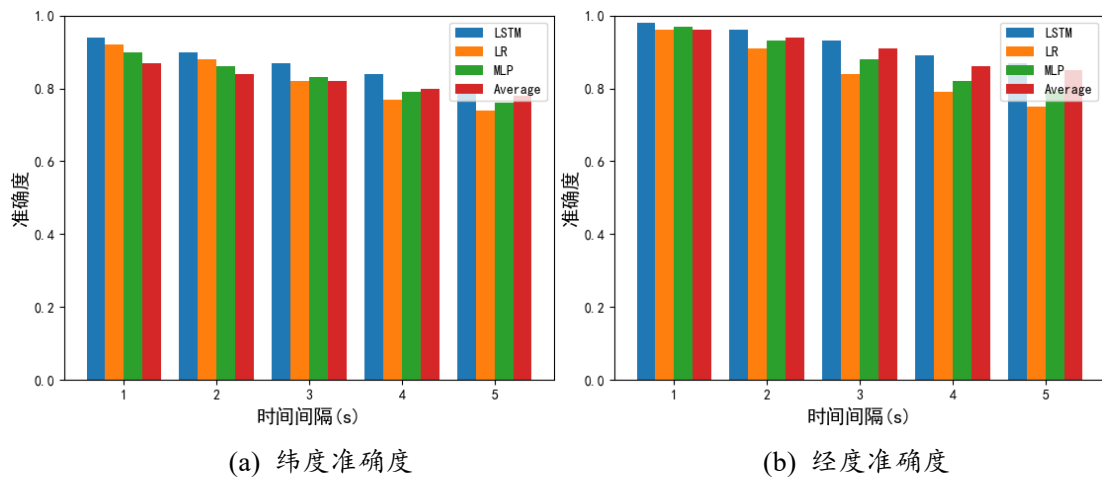


图 2-13 女子篮球比赛预测经纬度准确度

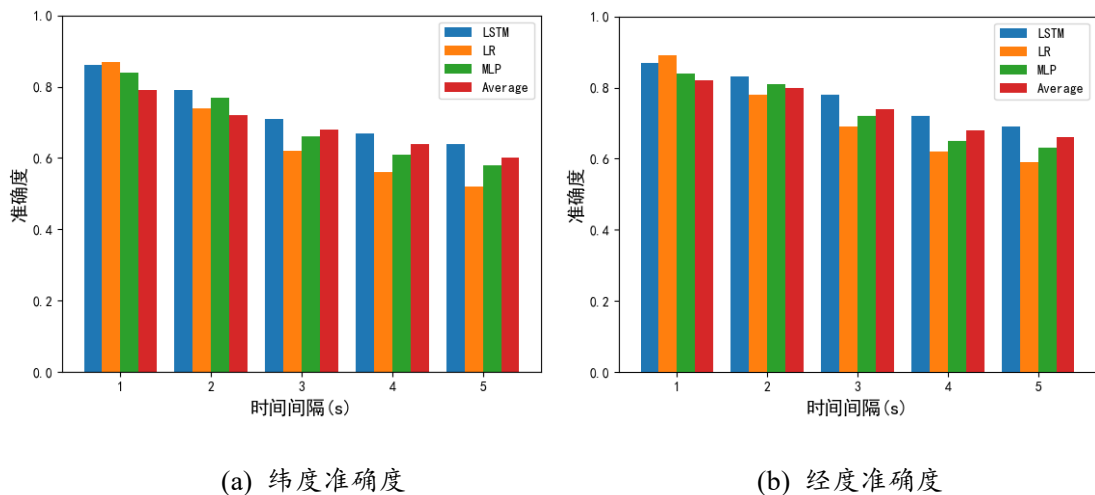


图 2-14 自由滑雪预测经纬度准确度

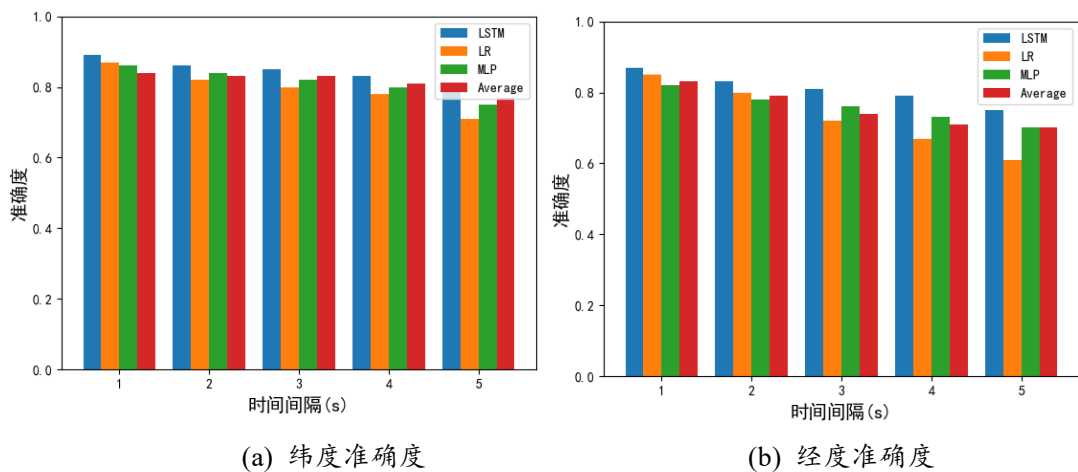


图 2-15 台表演预测经纬度准确度

从上面的多张图中可以看出,随着预测时间的增加,所有算法的预测准确率都在降低,尤其是线性回归算法的预测准确度显著下降,这说明视口的长期预测是比较困难的,这主要是因为用户在观看全景视频内容的过程中,随着时间长度的增加,用户会随着内容的变换而自由地切换感兴趣区域,造成移动的随机性增强,从而降低预测的准确度。

对比了三个视频在经度和纬度方面的预测准确度后,区别于文献[50]认为的经度的预测难度高于纬度,我们认为经纬度的预测难度没有绝对的高低之分。我们分析,虽然通常情况下,人们更倾向于左右移动头部而不是上下移动,但当用户完全沉浸在全景视频中后,头部移动轨迹跟视频内容密切相关,同时由于眩晕感,在短时间内用户并不会迅速地转动头部,所以纬度的变化范围在很多情况下也有可能大于纬度的变化范围。

对比四种预测算法可以看出,无论是预测经度还是纬度,基于 LSTM 模型的预测算法的准确度在各个预测窗口下普遍优于回归算法,尤其在长时间预测时。LR 在短时间(1s)内有较高的预测准确度,在某些情况下甚至高于 LSTM 的准确度,但是随着预测窗口的增加,LR 的预测准确度显著下降,低于 MLP 甚至移动平均,这说明视口数据在短时间内呈现较强的线性关系,而在长时间内则呈现非线性关系。由于视口数据更多的是以赤道为中心上下分布[50],所以长时间预测时移动平均算法会优于 LR。

另外,虽然都是运动场景,图 2-14 中的预测准确度却要明显低于图 2-13,这主要是由于,女子篮球比赛是直播场景,视频内容只有一个感兴趣区域位于视口位置,用户以较小幅度左右移动头部,在纬度范围上,用户头部随着篮球上下移动而呈现规律性移动,因此预测准确度较高。而自由滑雪的视频中有多个目标,目标移动幅度较大且分散,造成了预测的困难。舞台表演视频由于感兴趣区域集中,但目标移动的规律性小,所以预测准确度在前两者之间。

由实验结果图可以看出, LSTM 算法的预测效果更加优越,但在某些场景下的预测结果仍差强人意,为了提高预测度,我们在 LSTM 预测的基础上引入其他用户的视口数据对结果进行调整,结果如图 2-16 所示。

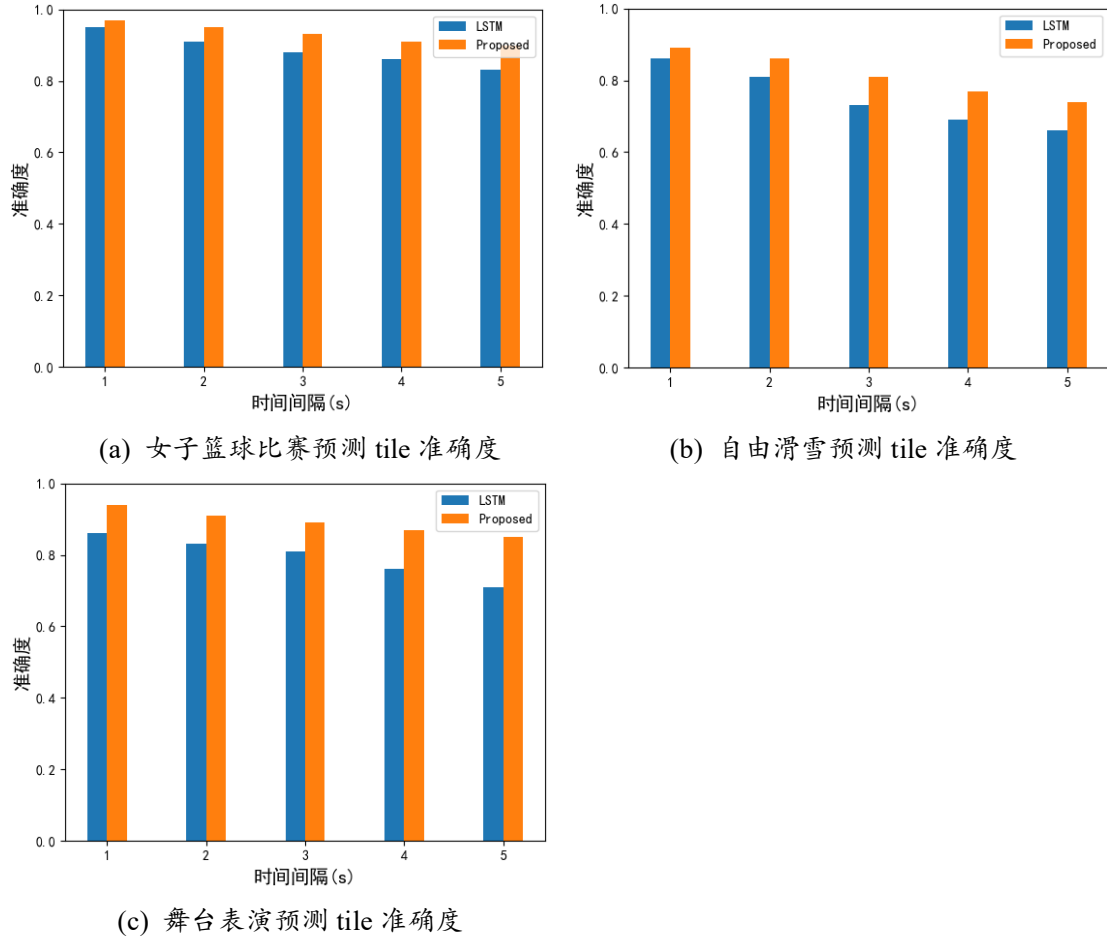


图 2-16 结合其他用户预测 tile 准确度

从图 2-16 中可以看出，我们提出的结合其他用户的视口数据来预测 tile 时要优于只基于当前用户的预测算法，由于舞台表演视频的视点轨迹呈现出波动大但用户之间分布集中的特点，所以结合其他用户的视点数据后比较显著地提高了预测准确度。从数值上看，当预测窗口为 5s 时，我们提出的算法的预测准确率分别为 83%、64%和 68%，比单用户预测算法的预测准确度分别提高了 8%、10%和 16%，平均提高了 11.3%。这表明，无论是哪种类型的全景视频，我们提出的算法在预测准确度上均取得了一定的提升效果。

2.4 本章小结

全景视频的视口预测是全景视频传输优化问题的重要前提，优秀的预测算法可以显著降低全景视频的传输数据量。本节针对现阶段视口预测算法长时间预测准确度明显下降的问题，提出了一种基于历史观看轨迹的预测算法。首先，该算法根据当前用户的历史观看轨迹，使用 LSTM 模型对轨迹的经纬度分别预测得

到初步预测结果。为了进一步提高预测准确度，在视频服务器存储了观看同一视频的其他用户的轨迹数据的情况下，我们计算其他用户在每块 tile 的观看次数，再结合 LSTM 模型的输出结果，综合计算出预测视口所覆盖的 tile 块。实验结果表明，基于当前用户和其他用户历史观看轨迹的预测算法比起其他常用的预测算法有着较高的准确度。

第三章 基于 Q-Learning 的 LTE 资源调度算法

准确的视口预测可以显著降低传输全景视频时的数据量，但当引入全景视频业务后网络系统会变得更加复杂动态，由于经典的调度算法在整个调度过程中只使用一种调度策略导致算法不能很好地适应多变的系统状态。本章聚焦于物理层面，基于 Q-Learning 来决定每个调度间隔选择何种调度策略使得网络性能最大化，并通过仿真实验对比了提出的算法与经典算法的在系统吞吐量、丢包率和用户公平性方面的性能优劣。

3.1 LTE 架构与资源调度

本节主要介绍了 LTE 网络架构、协议架构和关键技术，然后介绍了无线资源的帧结构与资源块概念，最后介绍了资源调度过程。

3.1.1 LTE 网络架构与协议架构

LTE 是 3G 技术的演进版本，相较于 3G 网络 LTE 主要为了实现提高链路的峰值速率、减小系统时延和尽量满足小区中所有用户的 QoS 需求等几个目标。

为了实现上述目标，LTE 网络采用了一种扁平化的架构即只由 EPC (Evolved Packet Core, 演进核心网) 和 E-UTRAN (Evolved Universal Terrestrial Radio Access Network, 演进通用陆地无线接入网) 两部分组成，如图 3-1 所示。这种架构一方面可以降低基础设备的数量，另一方面也可以降低系统的传输时延。

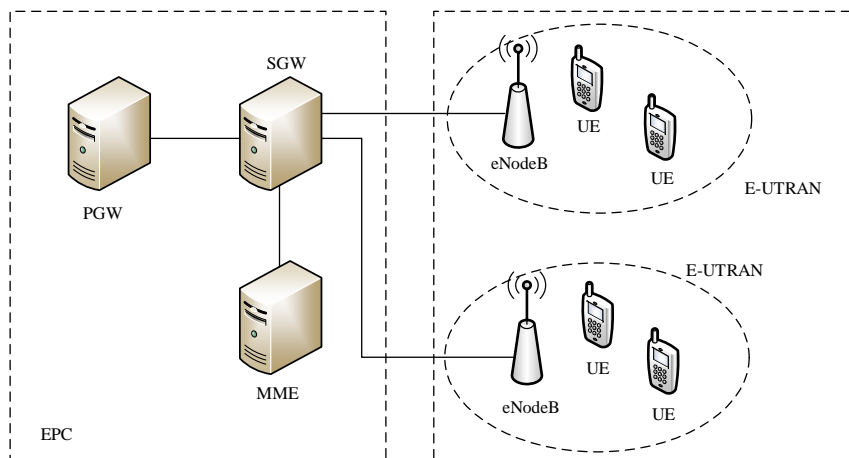


图 3-1 LTE 网络架构图

其中，EPC 的网络节点的功能如下：

(1) MME (Mobility Management Entity, 移动管理体) 主要负责用户移动、用户跨小区时切换小区和对信令进行加密控制等控制面功能；

(2) PGW (Packet Data Network Gateway, 分组数据网关) 主要是给 LTE 系统和其他非 3GPP 技术提供交互连接支持；

(3) SGW (Serving Gateway, 服务网关) 主要给用户数据包提供路由和传输支持并保证 QoS 需求；

与 3G 网络相比, LTE 系统中的 E-UTRAN 仅由演进型基站(eNodeB)组成, eNodeB 具有基站的全部功能和大部分无线网络控制器在接入层的功能。

接口定义了网络中不同单元之间进行数据沟通的方式, 接口协议的架构被称为协议栈。LTE 系统中的协议栈由功能不同的控制平面协议和用户平面协议两大块组成。下图为 LTE 系统协议栈示意图：

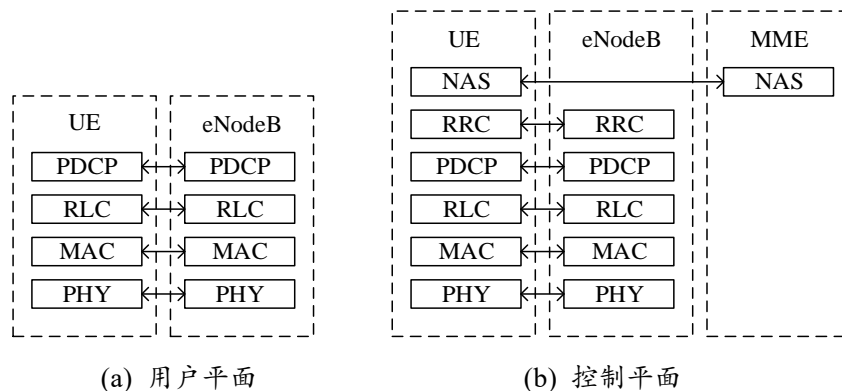


图 3-2 协议栈

用户平面各层主要功能如下：

(1) PHY (Physical Layer, 物理层) 主要负责编解码、调制解调、多天线等处理以及为 MAC 层的数据传输提供信道支持；

(2) MAC (Media Access Control, 媒体接入控制层) 主要负责处理逻辑信道和传输信道的对应关系, 处理用户 CQI 报告以及上下行链路的资源调度问题；

(3) RLC (Radio Link Control, 无线链路控制层) 主要负责分段重组用户的业务数据、实现错误重传机制和进行重复检查等等；

(4) PDCP (Packet Data Convergence Protocol, 分组数据汇聚协议层) 在发送端对流经的 IP 数据包进行头压缩以降低传输的数据量; 对控制平面的信令进行加密和保护；同时在接收端可以进行解压缩和解密等逆操作。

而控制平面则在用户平面上增加了用户承载的 RRC (Radio Resource Control, 无线资源控制层) 和用于控制非接入层的 NAS (Non-Access Stratum, 非接入层)。

3.1.2 LTE 架构关键技术

LTE 架构中主要的关键技术如下：

(1) OFDM

LTE 有两项关键的物理层技术：OFDM（Orthogonal Frequency Division multiplexing, 正交频分复用）和 MIMO（Multiple Input Multiple Output, 多输入输出）。其中，OFDM 是应用广泛的多载波接入方案，核心思想是在频域内将无线信道划分为若干个正交的互不干扰的子信道，接收端接收到信息后，可以通过一定的技术手段将正交信号分开，这就类似于把高速传输的串行数据信号转换成低速传输的多个并行信号，OFDM 的频谱示意图如下图所示：

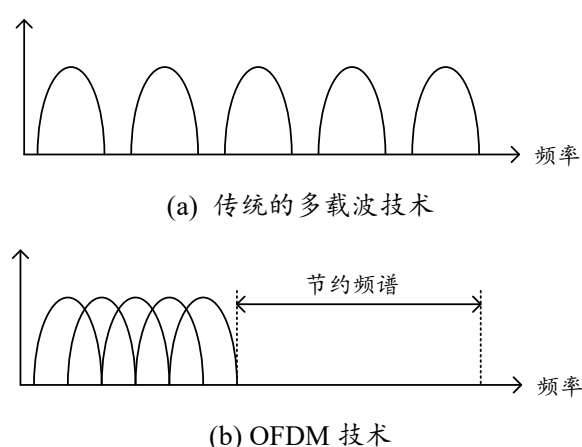


图 3-3 OFDM 频谱图

在以往频分复用的系统中，为了更好地消除子信道之间的干扰通常会在相邻的两个子信道之间增加足够的保护频带，而正交的技术允许没有保护频带，甚至相互覆盖仍不会造成干扰，比起传统的系统既降低了子信道之间的干扰性又极大提高了带宽利用率；由于在每个子载波上传输信号的时间要比单载波信号传输的时间长很多，使得 OFDM 在对抗信道衰落方面有着较好的表现。另一方面 OFDM 使用了计算简单易于实现的快速傅立叶反变化和快速傅立叶变化来进行调制解调。

不过 OFDM 仍存在一些缺点。一方面，由于 OFDM 的核心是依赖子载波之间严格的正交特性，而频偏或者相位噪声会造成不准确的正交，使得 OFDM 的性能大幅降低。另一方面，多个子载波信号合成的 OFDM 信号也会造成较高的峰均功率比。

(2) MIMO

区别于传统系统中发射端和接收端只有一根天线的形式，MIMO 则在两端同时采用了多个天线。理论上来说信道容量与天线数量呈线性正比关系，这使得

MIMO 可以在带宽和发射功率都保持不变的情况下显著提高 LTE 系统的吞吐量和信号的传送质量。图 3-4 展示了 MIMO 系统的框架结构：

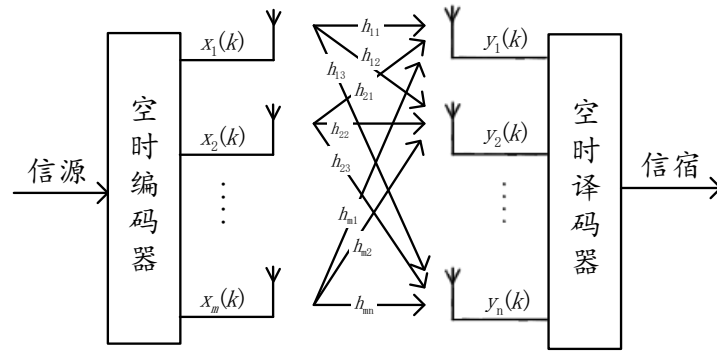


图 3-4 MIMO 系统框架图

MIMO 的关键技术主要包括：a) 空间分集，由于多天线技术在实现方面存在着一定的复杂度，空间分集可以把复杂度放在发射端，然后在接收端只使用单根天线即可获得增益；b) 空间复用，可以在多根天线上并行发射多个数据信号，提高了数据的峰值速率；c) 波束赋形，指利用信号的相干特性将能量汇聚在某个或某几个指定方向上产生波束来降低信号之间的干扰性。

(3) 高阶调制技术

调制技术是把要传输的原始信号处理为适合在信道中传输的信号。理论上而言，调制方式越高阶，带宽的利用率也越高，高阶是指每个调制符号可以承载较多的信息，在信噪比较好的情况下（比如离基站比较近的用户）可以通过提高调制阶数来提升信道的使用率。LTE 中主要有 QPSK、16QAM 和 64QAM 三种调制方式，其中 QPSK 的理论速率为 2bit/符号，16QAM 的理论速率是 QPSK 的 2 倍，64QAM 的理论速率则是 QPSK 的 3 倍。

3.1.3 无线资源调度

3.1.3.1 无线资源概述

无线资源分为时域和频域两个维度，在时域上表现为无线帧。LTE 系统中定义了两种帧结构：TDD（Time Division Duplexing, 时分双工）帧结构和 FDD（Frequency Division Duplexing, 频分双工）帧结构。

TDD 帧结构如图 3-5 所示，每个 TDD 帧的长度固定为 10ms 并且可以等分为 10 个长度为 1ms 的子帧，其中有 8 个常规子帧和 2 个特殊帧。

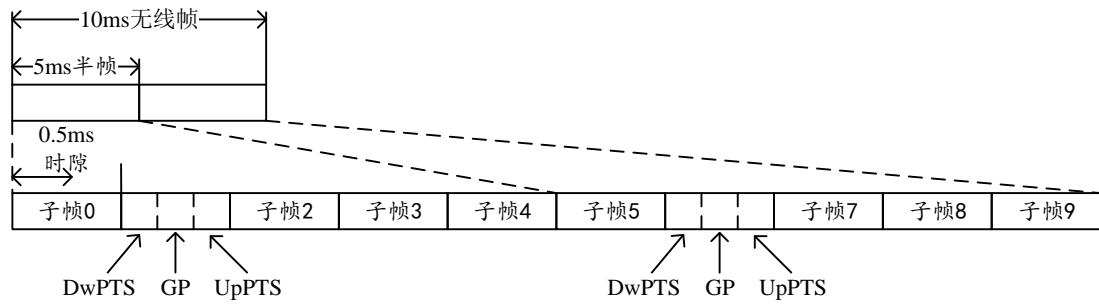


图 3-5 TDD 帧结构示意图

FDD 帧结构如图 3-6 所示，每个 FDD 帧的长度也为 10ms，不同于 TDD 的是 FDD 的无线帧由 10 个常规子帧组成。

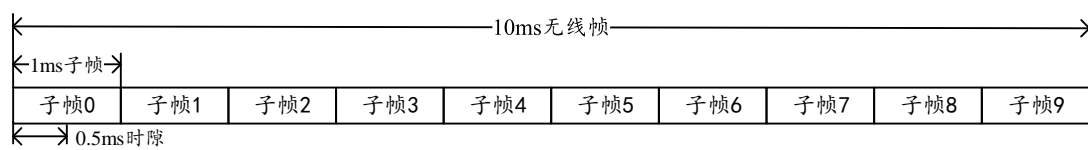


图 3-6 FDD 帧结构示意图

频域上无线资源则是由若干个 RB（Resource Block，资源块）构成。我们将占一个时隙（0.5ms）、12 个子载波（180kHz）的物理资源称作一个资源块，如图 3-7 所示，RB 是资源调度中可分配的最小单元。

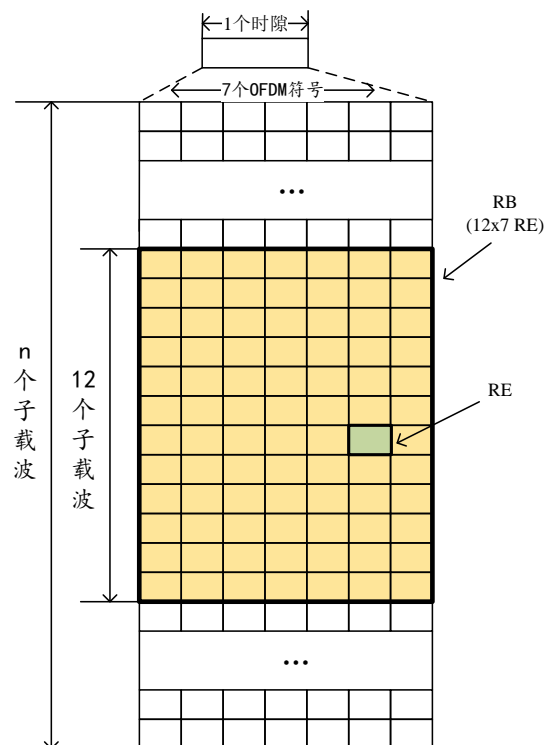


图 3-7 资源块结构图

系统带宽不同，物理资源块的数量也不同。在 LTE 系统中可以通过动态配置子载波数量以满足不同场景的需求。

3.1.3.2 资源调度流程

LTE 系统中上下行链路的调度过程是分开的, 本文重点探讨下行链路的调度过程。当小区内的多个用户请求不同业务时, 基站处的缓冲池就保存着等待传输的不同应用的任务队列。总结来说, 资源调度就是决策当前时刻给哪些待调度的任务队列分配哪些 RB, 主要分为两个流程, 即计算用户优先级和分配资源块。

(1) 计算用户优先级

优先级计算定义的是第 i 个数据流在第 j 个资源块上的调度优先顺序。不同的调度算法计算优先级的公式不同, 以此来达到各自的分配目标。例如 RR 算法, 每个数据流的优先级相同, 资源就被依次分配。

(2) 分配资源块

计算出优先级顺序后则要进行资源块的分配。假设系统带宽有 4 个资源块, 记为 RB1-RB4, 有 3 个待调度的用户, 记为 user1-user3, 根据调度算法计算出优先级结果如表 3-1 所示:

表 3-1 优先级矩阵

	RB1	RB2	RB3	RB4
user1	15	8	10	8
user2	11	12	14	13
user3	9	4	7	11

我们假设每个用户分配到两个资源块后即可达到传输需要。首先调度器开始遍历资源块 RB1 上的各个用户的优先级大小, 发现 user1 的优先级最高, 就将 RB1 分配给 user1, 然后在遍历 RB2, 发现 user2 的优先级最高, 将 RB2 分配给 user3, 然后再遍历 RB3, 发现 user2 的优先级最高, 将 RB3 分配给 user2, 最后遍历 RB4, 发现 user2 的优先级依旧最高, 但是此时 user2 已经分配到两个资源块了, 无需再分配, 调度器就会比较剩下的用户 user1 和 user3 的优先级, 发现 user3 的优先级较高, 因此将 RB4 分配给 user3。最后各个资源块的分配结果如下表所示:

表 3-2 资源块分配结果

user1	RB1
user2	RB2、RB3
user3	RB4

调度器就按照上述步骤依次遍历用户和资源块, 当所有的 RB 都被使用掉或者系统中所有等待的用户无需再分配资源时停止调度。

3.2 基于 Q-Learning 的资源调度算法

本节首先介绍了常见的四种资源调度算法的原理以及优缺点,在这些研究的基础上提出一种改进算法并详细阐述了改进算法的设计思想和系统框架。

3.2.1 经典的资源调度算法

在有限的带宽资源条件下如何有效地采取措施进行分配调度来尽可能满足小区内不同用户的需求至关重要。通常需要考虑用户公平性、系统吞吐量、系统丢包率和分组时延等用来衡量调度算法好坏的常用指标,接下来从这几个角度来介绍四个经典的调度算法,这些算法对后续改进算法的提出奠定了基础。

(1) PF 算法

最初研究人员提出的轮询算法 RR 和最大载干比 Max C/I 算法都只考虑单个方面的性能最优,应用到实际网络中存在着一定问题,因此,研究人员提出了同时考虑吞吐量和公平性两方面参数的比例公平算法 PF。该算法在定义用户优先级上同时考虑了用户的历史吞吐量和信道质量两个因素,优先级公式定义如下:

$$m_{i,j} = \frac{r_{i,j}}{\bar{R}_i} \quad (3-1)$$

其中, $m_{i,j}$ 表示第 i 个用户在第 j 个资源块上的优先级; $r_{i,j}$ 表示将第 j 个资源块分配给用户 i 时用户所获取到的即时传输速率(可以通过用户的 CQI 反馈计算出来), $r_{i,j}$ 的值越大说明用户的信道质量条件越好; \bar{R}_i 则表示用户 i 的历史平均传输速率,在仿真中,该值的计算一般遵循以下公式:

$$\bar{R}_i = 0.8\bar{R}_{(i-1)} + 0.2r_i \quad (3-2)$$

从公式中可以清楚的看出,如果刚开始某个用户的信道条件比较好时瞬时速率较高,用户的优先级较大从而可以优先调度分配到资源,但随着资源的逐渐积累,该用户的历史平均速率也会升高,用户优先级随之下降,系统则会有可能将资源分配给小区边缘信道条件不好的用户。

PF 算法在吞吐量和公平性方面较好的性能表现使得其成为了 LTE 下行链路中最常使用的调度算法。但是该算法没有考虑到各个业务 QoS 参数的不同,尤其是实时业务对时延的敏感性,所以需要更加适合实时业务的调度策略。

(2) EDF 算法

为了解决实时业务的调度问题,最早截止先 EDF 算法被提出。该算法给予即将到达时延阈值被丢弃的业务最高优先级,给出最简单的计算公式如下:

$$m_{i,j} = \frac{1}{\tau_i - D_{HOL,i}} \quad (3-3)$$

τ_i 为第*i*个用户业务流的时延阈值，超过该阈值的数据包将会被丢弃， $D_{HOL,i}$ 为用户*i*的缓存队列首个数据包的等待时间。可以看出，越临近时延阈值的用户优先级越高。

(3) MLWDF 算法

应对上述问题，改进的最大权重延迟优先算法 MLWDF 被提出。对于非实时业务，该算法计算优先级的公式等同于 PF 算法，而对于实时业务，该算法在 PF 算法计算优先级的公式的基础之上，引进了服务质量参数。优先级公式定义如下：

$$m_{i,j} = \alpha_i \times D_{HOL,i} \times \frac{r_{i,j}}{\bar{R}_i} \quad (3-4)$$

$$\alpha_i = -\frac{\log(\delta_i)}{\tau_i} \quad (3-5)$$

δ_i 为队头时延超过时延阈值的最大概率，在仿真过程中一般设置为常数。

(4) EXP/PF 算法

EXP/PF 算法是对 M-LWDF 算法的一种改进，也同样适用于非实时业务和实时业务。该算法引入了缓冲时间来保证对实时业务对优先调度。对于实时业务，优先级公式定义如下：

$$m_{i,j} = \exp\left(\frac{\alpha_i * D_{HOL,i} - X}{1 + \sqrt{X}} * \frac{r_{i,j}}{\bar{R}_i}\right) \quad (3-6)$$

其中的参数含义等同于公式 (3-5)，X 计算公式如下：

$$X = \frac{1}{N_{rt}} * \sum_{i=1}^{N_{rt}} \alpha_i * D_{HOL,i} \quad (3-7)$$

其中， N_{rt} 表示系统中等待调度的实时用户数。对于非实时业务，该算法的优先级公式与 PF 算法相同。

3.2.2 Q-Learning 介绍

上述这些经典算法虽然被广泛使用，但在整个调度过程中始终使用单一的调度策略。当引入全景视频、传统视频和网页等多种业务时网络系统会变得十分复杂动态，这导致这些算法不能很好地适应不断变化的网络情况。我们希望能够根据不同的网络状态来选择最佳的调度策略以达到性能最优化。

近年来开始出现一些基于增强学习进行优化的研究。对调度而言，增强学习是一个很好的切入点，因其可以较好地学习动态变化的网络状态然后做出更高效的策略，Q-Learning 算法就是这方面的翘楚。

Q 学习 (Q-Learning) 是增强学习领域最广为人知的算法之一。Q-Learning 与模型状态无关, 可以在环境未知的情况下, 根据“状态-动作”获得环境的奖励, 然后根据奖励不断调整动作使得奖励最大化, 最后学习到最佳动作。可以看出, Q-Learning 主要包含三个要素: (1) 状态空间 $S = \{s_1, s_2, \dots, s_n\}$ (2) 动作空间 $A = \{a_1, a_2, \dots, a_m\}$ (3) 环境的奖励 $r(s_t, a_t)$ 。

我们将实施上述过程的客体称为智能体 (Agent), 在每一次迭代中, Agent 根据当前的状态 s_t 从动作空间中选择一个动作 a_t , 然后获得环境的即刻奖励 r_t , 然后环境的状态转移到 s_{t+1} , 继续迭代, 记 $Q(s_t, a_t)$ 为多次迭代后的累计回报, 使用一个 $n \times m$ 的二维表来记录该值, 该表称为 Q 表 (Q-table), 在每次迭代中根据公式 3-8 来不断更新 Q 表的值。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot [r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3-8)$$

其中, α 为学习率, 范围在 0~1 之间, 代表之前训练结果的影响权重, 学习率越大, 之前训练结果的影响越小; γ 则为折扣因子, 范围在 0~1 之间, 代表了未来奖励的重要性, 如果 $\gamma=0$, 只有当前的奖励会被考虑。 $\max_a Q(s_{t+1}, a)$ 代表了下一个状态时 Agent 可以获得的最大的累计奖励。经过多次更新后, Q 值收敛到最优值。

3.2.3 DSQL 算法介绍

需要说明的一点是, 由于我们思考优化方案的时候, 考虑更多的是如何改进调度底层物理资源的算法, 以更好地适应引入了全景视频视口数据后的混合业务场景, 而没有只针对全景视频业务, 实现视口数据端到端的传输流程。所以我们忽略了跨层信息 (如从应用层面区分视口数据与传统视频数据), 暂时先只聚焦于底层物理数据层面, 将视口区域 (或者全部传输的全景视频) 视为数据量较大、时延阈值较小的实时业务而不再进行额外的区分。本节基于此假设提出 DSQL 算法 (Downlink Scheduler based on Q-Learning, 基于 Q 学习的下行调度算法)。

从上述分析可以看出, 将 Q-Learning 应用到资源调度场景中, 负责调度过程的基站就充当 Agent 的角色, 动态变化的无线网络为环境的状态 S , 动作 A 则是不同的调度算法, 我们希望在每一个 TTI (Transmission Time Interval, 传输时间间隔, LTE 系统中设置为 1ms) 都可以选择一个最适合当前环境的调度算法, 最后使得系统的累计性能最大, 而常用的衡量系统性能的网络参数如系统吞吐量等即为奖励, 基于 Q-Learning 的调度过程如下图所示:

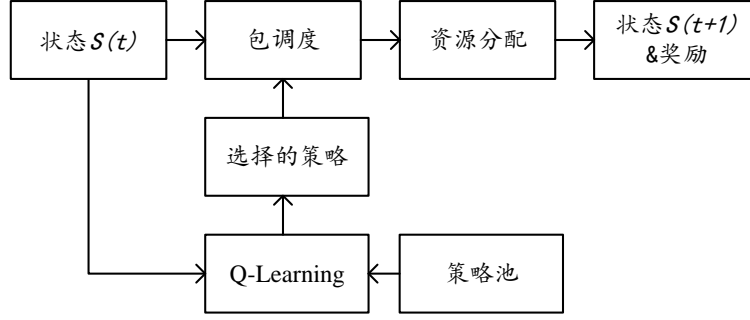


图 3-8 基于 Q-Learning 的调度过程

首先，基站从环境中获取到信息来判断当前所处状态 s_t ，然后从定义好的若干个调度算法中选择一个，计算用户优先级完成资源分配后，环境状态从 s_t 变化到 s_{t+1} ，并且可以获得此次调度过程的即刻奖励，按照公式 3-8 更新 Q 表的值并开始下一次的迭代，直至调度完成。

根据上述分析，基于 Q-Learning 的调度算法重要的是需要定义状态空间、决策空间和奖励函数，接下来将详细介绍这几个关键点。

(1) 状态空间 S

状态空间的划分是 Agent 可以学习到动作策略的基础，合理的状态应该满足以下两点要求：一是状态可以被 Agent 感知和处理，二是后一个状态只能依赖于前一个状态和选择的决策。一次资源调度后，系统中用户需要的数据量和等待时间都会发生变化，同时考虑到我们算法的目的主要是为了满足实时业务的 QoS 需求，所以我们将定义状态空间为 $\langle dataIndex, delayIndex \rangle$ ，其中：

$$dataIndex = \frac{Th_{req}}{Th_{max}} \quad (3-9)$$

$$delayIndex = \sum_{i=1}^N (\delta_i - D_{HOL,i}) \quad (3-10)$$

Th_{req} 为等待调度的所有用户的数据量总和， Th_{max} 为系统可传输的最大的数据量， $dataIndex$ 表征了无线资源的短缺程度； N 为等待调度的用户数， $delayIndex$ 表征了待调度应用流的时间紧迫度。当 $dataIndex$ 和 $delayIndex$ 都比较小时，说明更多的用户快要到了时间阈值且系统的资源较为充足，此时应该给时间紧迫的用户更高的优先级；而当 $dataIndex$ 和 $delayIndex$ 都比较大时，应该给信道质量较好的用户更高的优先级。

如果直接使用上述公式来定义状态的话，由于实际的网络情况比较随机，状态空间的数量将是巨大的。为了降低计算的复杂度提高收敛速度，我们首先统计了测试环境下 $dataIndex$ 和 $delayIndex$ 的数值范围和分布概率，然后将状态划分为 16 个区间，落在同一个区间的状态被赋予相同的值。

(2) 动作空间A

通过前面的介绍,显而易见的,决策空间应该被定义为不同的资源调度算法。为了保证 QoS 需求,计算方式需要综合考虑信道质量、用户公平性和实时业务的时延需求。因此,基于 MLWDF 算法,定义优先级的一般计算公式如下:

$$m_{i,j} = \frac{(r_{i,j})^\theta}{(\bar{R}_i)^\beta} \cdot \left(\frac{1}{\delta_i - D_{HOL,i}} \right)^\omega \quad (3-11)$$

其中参数的相关含义与公式 3-4 相同。

考虑到系统吞吐量、用户公平性和业务时延的权重关系,我们选择其中特定 4 种情况构成动作空间,如下表所示:

表 3-3 动作空间

动作	θ	β	ω
A1	1	1	1
A2	1	1	2
A3	1	0	1
A4	2	1	1

每个调度动作可以得到不同的 RB 矩阵,根据 RB 矩阵带宽资源依次分配给用户完成一次调度过程。

(3) 即刻奖励 r

即刻奖励是用来评估前一个动作是否合适的重要指标,奖励函数的设计需要基于系统性能参数。我们希望在每个 TTI 选择某个调度策略后,系统的 QoS 指标(吞吐量、公平性和时延)可以得到提升,因此,奖励与这三个指标紧密相连。由于这三个指标的数量级不同,需要将取值映射到同一区间,下面依次给出三个指标的计算公式,首先是系统吞吐量指标 STI:

$$STI(t) = \sum_{i=1}^N \frac{Th_i(t)}{Th_{all}} \quad (3-12)$$

其中, N 为当前调度流程时系统中待调度的用户数, $Th_i(t)$ 为 t 时刻用户 i 实际获得的吞吐量, Th_{all} 则为当前调度流程可以调度的资源总和,可以看出 UTI 的取值范围在 0~1 之间。

我们使用 AJFI (Average Jain's Fairness Index, 平均 Jain 公平指标) [51] 作为公平性指标,定义如下:

$$AJFI(t) = \frac{(\sum_{i=1}^N Th_i(t))^2}{[N \cdot \sum_{i=1}^N (Th_i(t))^2]} \quad (3-13)$$

其中参数的含义与公式 3-12 相同。

我们定义系统时延指标 SDI 如下：

$$SDI(t) = \frac{\sum_{i=1}^N D_{HOL,i}}{\sum_{i=1}^N \delta_i} \quad (3-14)$$

SDI 的值越小，用户的等待时间越短。

由于学习目标是使得奖励最大化，所以即刻奖励应该跟吞吐量和公平性成正比，跟时延成反比，因此，我们定义即刻奖励的计算公式如下：

$$reward = \eta \cdot STI + \lambda \cdot AJFI + \mu \cdot SDI \quad (3-15)$$

其中， η 、 λ 和 μ 为权重值，可以通过设置不同的权重来达到所期望的权衡。因此，这个奖励函数期望系统吞吐量和公平性最大化，同时降低系统时延。

(4) 探索与开发

除了上述三个要素外，Q-Learning 算法还需要确定搜索策略。如果每次都是选择 Q 值最大的动作，则称这种搜索方式为开发（exploit），开发模式会导致 Q 值小的动作一直不被执行，每次只选择当前阶段的最优动作而忽略了全局。探索（explore）则使得系统去尝试更多没有被选择过的策略。考虑到我们算法使用的仿真环境，我们使用一个简单的 ε -贪婪算法，按照概率 ε 随机去选择动作，然后以 $1-\varepsilon$ 的概率根据 Q 值来选择最优的策略。在仿真实验时， ε 初始为 1，即最开始完全随机的探索，然后逐渐降低到 0.1 后保持不变。

最后，我们给出 DSQL 算法的调度流程，如表 3-4 所示：

表 3-4 DSQL 算法的调度流程

1: 初始化 Q 表: $Q_0 = \{0\}$, 设置权重值 γ 和 α
2: for t=0 to T-1 do
3: 根据公式 3-9、3-10 从状态空间中选择当前的状态 S_t
4: 根据 ε 贪婪算法选择一种调度策略
5: 进行包调度和资源分配
6: 根据公式 3-15 计算奖励
7: 根据公式 3-8 更新 Q 表
8: end for

3.3 实验与结果分析

3.3.1 仿真工具介绍

由于 LTE 系统涉及的技术繁杂，在真实的网络环境中进行实验论证存在着诸多困难，而使用仿真工具可以显著提高实验的可行性，节约实验时间。仿真方法主要有面向物理层的链路级仿真和面向网络架构的系统级仿真两种。为了验证 DSQL 算法的性能，我们需要进行系统级的仿真实验。

目前开源的 LTE 系统级仿真工具主要由三个：维也纳大学开发的 Vienna LTE-A Simulators，基于 Matlab 进行开发；意大利都灵理工大学开发的 LTE-Sim[52]，基于 C++，运行在 Linux 环境下和 NS-3。考虑到仿真工具的复杂度和对 C++ 开发语言的熟悉度，我们使用 LTE-Sim 作为仿真工具完成实验。

LTE-Sim 是一个开源的 LTE 系统级仿真平台，使用 C++ 作为开发语言，因其具有封装与继承的特点，易于扩展开发，近些年来被研究资源调度的学者广泛使用。该平台以事件驱动，定义了 90 个类函数，共有 220 个文件大约 2 万多行代码。

LTE-Sim 平台再现了完整的 LTE 网络结构，包括 E-UTRAN 和 EPS，支持单/多小区、单/多用户环境、QoS 管理、多小区切换、频分复用技术、AMC 调制等多个场景和技术。LTE-Sim 主要包含四个模块：Simulator、NetworkManager、FlowsManager 和 FrameManager，主要负责开始、处理和结束事件；定义并处理 LTE 帧和子帧；管理应用流；创建小区节点、用户设备并设置用户位置以及实现小区切换和频分复用。LTE-Sim 还实现了三个网络节点：UE、eNodeB 和 MME/GW，这三个节点是由 NetworkManager 创建和处理的。

LTE-Sim 仿真了四种最常见的应用类：（1）Trace-based，即视频流，主要是根据真实的视频文件获得到的数据包；（2）VoIP，语音流，根据 G.729 标准构建的 on/off 马尔科夫链模型；（3）CBR，有固定比特率的数据包，可指定数据包的到达时间；（4）Infinite-buffer，模拟的持续发送的数据包，一般用户模拟背景业务。

针对资源调度算法的仿真问题，LTE-Sim 基本实现了资源调用与分配的整个过程，DownlinkPacketScheduler 类包含的函数实现了从设定用户流承载到计算优先级再到分配资源的全部功能，其中，与资源调度算法有着最直接关系的是 ComputeSchedulingMetric 函数，在该函数里可以编写不同的计算用户优先级的公式，默认实现了 PF、MLWDF 和 EXP/PF 算法。由于 CQI 反馈是用户与基站之

间重要的交互过程，LTE-Sim 也实现了该功能。

在 LTE-Sim 中对资源调度过程进行仿真的主要步骤如下：

- (1) 初始化仿真器、网络管理器、帧管理器和流管理器
- (2) 创建小区、基站、网关和信道，创建用户并设置位置和移动模式
- (3) 为用户指定承载的应用流，定义 QoS 参数和持续时间
- (4) 指定资源调度算法，分配无线资源
- (5) 指定仿真运行时间，开始仿真

仿真开始后，用户开始请求应用，基站侧的调度器根据数据包的 QoS 参数和用户发过来的 CQI 反馈等信息选择调制编码方式，然后根据资源调度算法计算用户优先级，最后分配无线资源块后完成一次调度过程。在整个仿真运行时间内，基站在每一个 TTI 执行一次上述过程。

3.3.2 仿真环境与参数

我们模拟了单个小区多用户的环境。小区半径为 1km，小区中央有一个基站，用户数量从 10 递增到 40，随机分布在不同位置，同时用户在小区内以 3km/h 的移动速度选择随机的方向运动。为了防止仿真随机性对结果造成的影响，我们进行多次实验取平均值作为最终的实验结果，仿真环境具体参数见表 3-5。

表 3-5 仿真环境参数设置

参数	参数值
带宽	10MHz
小区半径	1Km
复用模式	FDD
帧长	1ms
仿真时长	66s
业务流时长	55s
用户数	10-40
用户速度	3km/h
应用	Video, VoIP
视频时延	20-100ms
VoIP 时延	100ms
学习速率	0.05
折扣因子	0.01

3.3.3 仿真结果与分析

首先介绍衡量调度算法性能的几种指标。

(1) 系统吞吐量

系统吞吐量表征了单位时间内系统传输的数据量总和,在其他条件相同的情况下,系统吞吐量越大表明无线资源的利用率越高。本文使用如下公式来计算系统吞吐量大小:

$$\text{System throughput} = \frac{1}{T} \sum_{i=1}^N \sum_{t=1}^T p_{\text{transmit}_i}(t) \quad (3-16)$$

其中, T 表示仿真时间内传输间隔的数量, N 为等待调度的用户总数, $p_{\text{transmit}_i}(t)$ 为用户 i 在 t 时刻传输的数据量。

(2) 用户公平性

追求系统吞吐量最大化可以充分利用带宽资源,但是会导致一些边缘用户一直分配不到资源,为了保证同一小区内的用户得到自由的机会平等,还需要定义用户公平性。公平性的评定方式有多种,本文采用使用 AJFI 作为公平性指标,计算公式如 3-13 所示。可以看出,系统吞吐量跟用户公平性是互相矛盾的。

(3) 丢包率

丢包率是丢失数据在总传输数据中的占比,丢包率小可以很好保证传输数据的完整度。丢包率主要取决于数据包的长度和排队时间。丢包率表示如下:

$$PLR = \sum_{i=1}^N \sum_{t=1}^T p_{\text{discard}_i}(t) / \sum_{i=1}^N \sum_{t=1}^T p_{\text{size}_i}(t) \quad (3-17)$$

其中, $p_{\text{discard}_i}(t)$ 为用户 i 在时刻 t 丢弃的数据包大小, $p_{\text{size}_i}(t)$ 则为用户 i 在 t 时刻成功到达基站缓存队列的数据包大小。

(4) 系统时延

系统时延主要是衡量实时业务的参数,即数据端到端传输所消耗的时间,实时业务的等待时间超过时延阈值时会被丢弃。定义系统的平均时延如下:

$$\text{average system delay} = \frac{1}{T} \sum_{t=1}^T \frac{1}{N} \sum_{i=1}^N D_{\text{HOL},i} \quad (3-18)$$

为了强调我们对全景视频类实时业务时延性能的考虑,首先模拟了小区内有 30 个用户且每个用户都请求视频业务的场景。由于我们假设全景视频视口区域为数据量较大、时延阈值很小的实时业务,我们给用户加载了不同数据量的视频业务,并给这些数据设置了不同的时延阈值,以此来模拟系统中混合视口数据和传统视频数据的场景,我们选择经典的实时业务资源调度算法 EXP、MLWDF 和 EXP/PF 作为对比算法,仿真结果如图 3-9~3-11 所示。

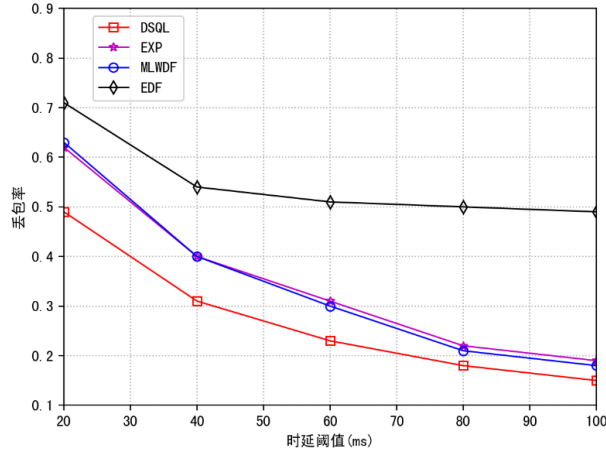


图 3-9 不同时延阈值时系统丢包率

通常来说,丢包率是评估实时业务调度算法的重要指标之一。如图 3-9 所示,显而易见的,所有算法的丢包率都与时延阈值呈反比。EDF 算法只根据业务的时间紧迫度计算优先级,理论上会降低因超过时延阈值而被丢弃的数据包数量,但在实际调度过程中,可能由优先级最高的用户的信道质量很差,导致一次调度过程中用户只能传输少部分数据,只能继续等待下次调度,资源利用率较小导致后面资源紧缺,造成大量用户得不到调度被丢弃。MLWDF 和 EXP 均考虑了时延和信道质量参数,且计算优先级公式类似,因此两者的丢包率相似。DSQL 算法可以自适应的选择最优的调度策略因而丢包率最小,尤其在时延阈值小于 40ms 时显著优于其他三种算法,这说明 DSQL 算法在调度全景视频这类时延敏感的业务时有着较好的性能。

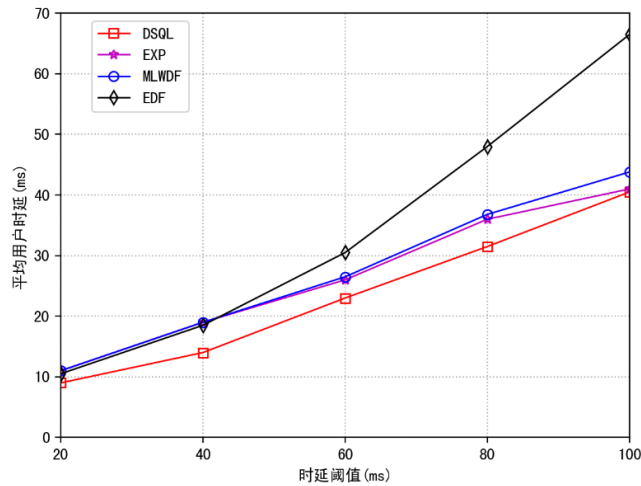


图 3-10 不同时延阈值下平均用户时延

图 3-10 展示了不同时延阈值下的平均用户时延。在时延阈值小于 40ms 的时候,可以看出 EDF、EXP 和 MLWDF 的平均用户时延基本一致,EDF 略低于后两者,但这并不是说明 EDF 算法在时延方面优于后两者,结合之前的丢包率可

以分析出，由于 EDF 的丢包率较大，许多时延较大的数据被丢弃了，导致参与计算的都是时延较小的用户。由于 EXP 算法中优先级与时延呈指数关系，对时延的反应更加灵敏，所以它在时延方面的性能要略优于 MLWDF 算法。与丢包率类似，在用户平均时延上 DSQL 算法仍保持着最优的性能，这说明在复杂变化的网络状态下灵活选择不同的调度算法要优于只使用一种特定的算法。

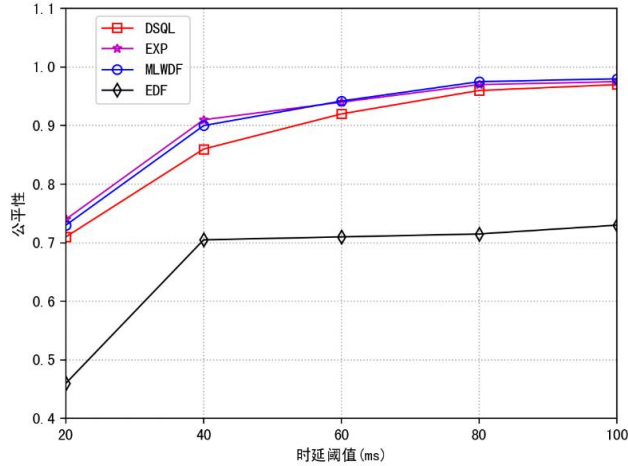


图 3-11 不同延迟阈值下用户公平性

图 3-11 展示了各算法在用户公平性方面的性能。理所当然，EDF 算法有着较差的公平性，这也说明，通常情况下只考虑一个方面的调度算法的性能比较差。由于我们在设计决策空间和奖励函数的时候，重点考虑了系统吞吐量和时延参数，所以在公平性方面 DSQL 算法略差于 EXP 和 MLWDF 算法，而要做到可以在这几个指标上都得到优化是十分困难的。所以考虑到 DSQL 算法在系统丢包率和用户时延方面取得的优异表现，用户公平性方面的轻微牺牲是可以接受的。

为了进一步评估我们算法的性能，我们改变了仿真环境，用户数范围从 10 个以 5 个的间隔递增到 40 个，并且每个用户都请求实时视频业务和语音业务。同样的，为了模拟全景视频与传统视频混合的场景，设置了多种数据量的视频业务，并设置了 20ms 和 40ms 两种时延阈值；VoIP 业务的时延阈值则设置为 100ms。图 3-12~3-13 显示了混合实时业务下的算法性能。

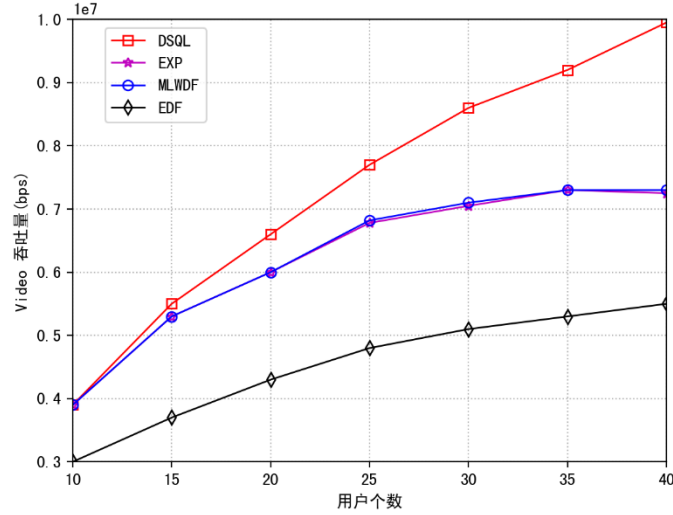


图 3-12 不同用户时 video 吞吐量

从图 3-12 可以看出，当系统中有 10 个用户时，DSQL、EXP 和 MLWDF 的吞吐量一样，这是因为此时用户数少，系统资源充足，所有用户都分配到了资源，系统吞吐量即为所有用户所需的数据量总和。随着用户数的增加，系统吞吐量逐渐上升，但随之丢包率也开始上升，所以当用户数达到 35 时，EXP 和 MLWDF 算法开始出现轻微下降的趋势，而 DSQL 算法仍在持续上升。EDF 依旧表现最差。

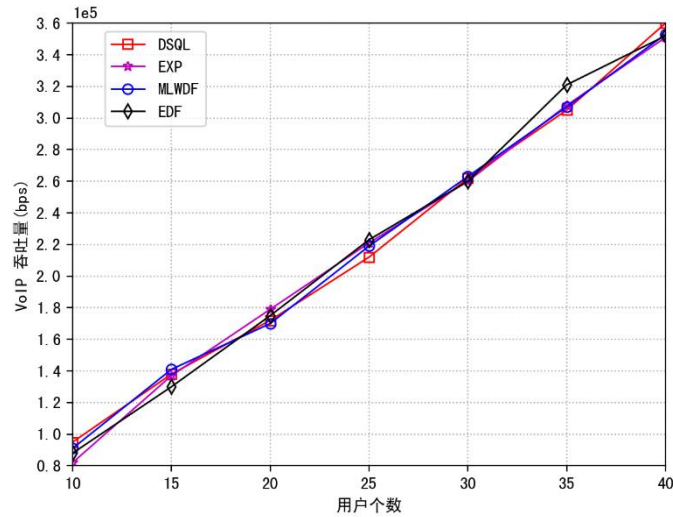


图 3-13 不同用户时 VoIP 吞吐量

图 3-13 则为混合业务中 VoIP 的吞吐量，可以看出几种算法有着相似的性能。这表明，相比于其他算法，DSQL 可以在显著提高视频类业务性能的同时维持着 VoIP 业务的性能。

从上述实验结果和分析可以看出，我们提出的 DSQL 算法在调度混合全景视频视口业务、传统视频业务和 VoIP 业务的场景时可以表现出较为优异的网络

性能,尤其当系统中存在较多时延敏感业务时的性能要显著优于经典算法。不过如第 3.2.2 节所解释的,目前我们设计 DSQL 算法时只聚焦于物理层参数,在仿真实验过程中只是通过设置数据量和时延阈值来模拟全景视频视口数据和混合传统视频数据混合的场景,同时我们目前只使用了吞吐量和丢包率等网络性能指标来对比调度算法,所以在评估 DSQL 算法性能时也存在一定的片面性。

在未来的研究工作中可以进一步实现从视口数据端到端传输的整个流程,在底层物理资源调度时,尝试引入跨层信息来标记全景视频视口数据、非视口数据、传统视频数据和非实时业务数据,然后在划分系统状态 S 和调度策略 A 时根据标记信息进行更细粒度的区分。同时可以尝试使用视频卡顿次数、视频画面质量、客户端缓存长度等 QoE 性能指标对奖励函数 R 进行调整,使得算法的用户性能参数而不是网络性能参数最大,进一步优化 DSQL 算法以更好地面向全景视频业务。

3.4 本章小结

本章首先介绍了 LTE 架构和几种关键技术;然后介绍了 TDD 和 FDD 两种无线帧结构,引出 LTE 系统中资源块的定义,描述了资源调度和资源分配的过程,并着重介绍了经典的资源调度算法。

本章针对全景视频这类实时业务,结合增强学习中的 Q-Learning 算法,从物理数据层面提出一种改进的下行资源调度算法 DSQL,该算法可以根据变化的网络状态和用户状态,动态地选择不同的调度策略,在系统吞吐量、系统时延和用户公平性之间取得平衡使得综合性能最大化。然后,本章介绍了使用的仿真平台 LTE-Sim,描述了常用的仿真功能函数和仿真过程。最后,通过仿真实验从系统吞吐量、丢包率、时延和用户公平性几个方面对比了提出的算法与经典算法的性能优劣,最终验证了 DSQL 算法的性能优越性。

第四章 总结与展望

4.1 工作总结

随着计算机科学和多媒体技术的发展,全景视频作为一种全新的信息媒介以其沉浸式构想性的特点给人们带来了身临其境的观感,并逐渐应用到各个领域。然而,全景视频数据量大、对时延要求严格的特性也给现有的传输网络带来了一些挑战,如何优化全景视频的传输一直是受人关注的研究领域。因此,本文从降低传输数据量和高效调度无线资源两个方面进行改进,主要工作研究与成果如下:

(1) 为了降低全景视频的传输量,本文首先研究了视口预测算法。考虑到基于内容分析的预测算法的复杂性,本文只聚焦于基于历史轨迹的预测算法,将视口预测问题划分为时间序列预测,首先介绍了几种常见的预测算法,基于视口的二维特性,即经纬度,初步提出了基于 LSTM 模型的预测算法,在数据集上初步验证了该算法的预测准确度高于其他一些算法。为了进一步提高预测准确度,本文分析了观看同一个视频的其他用户的视口轨迹,发现轨迹之间的相似度较高,基于此前提,本文结合其他用户的视口数据对初步预测的结果进行矫正,最终,实验结果证明联合其他用户视口数据的预测算法在长时间预测时可以有效提高预测准确度。

(2) 减少全景视频的传输量之后,本文又考虑了混合业务下 LTE 网络中基站侧的无线资源调度。首先介绍了 LTE 的相关理论,包括系统架构和关键技术。然后介绍了无线资源块的概念,给出了资源调度和分配流程。针对全景视频这种对时延敏感的业务,本文研究了经典的调度算法的优缺点后,基于强化学习中的 Q-Learning 算法,从物理数据层面出发提出了一种创新的下线资源调度算法 DSQL,该算法根据动态变化的网络状态和用户状态,使用 Q-Learning 算法在每个传输间隔选择不同的调度策略,最终达到系统性能最大化目的。为了验证 DSQL 算法的性能优越性,本文介绍了仿真平台 LTE-Sim 的相关功能和仿真流程,通过仿真实验,从系统吞吐量、系统丢包率、时延和用户公平性几个方面综合对比了 DSQL 和经典算法的性能,最终验证了 DSQL 算法的优越性,尤其在时延敏感的实时业务调度方面。

4.2 未来展望

全景视频传输优化是一个庞大的研究领域，设计到计算机视觉、数据挖掘、无线网络传输优化等多个研究学科，本文只在现有研究基础上做了部分改进，仍有一些问题需要未来做进一步深入研究：

（1）本文只聚焦于基于历史轨迹的视口预测算法，没有充分考虑到影响视口的多个因素，如视频内容的多样性和用户的观看兴趣与习惯，没有建立充分完善的视口模型；在对视口预测时，分别预测了视口中心位置经纬度的规律，没有进一步探索其余预测模式，如联合预测经纬度，或直接对 tile 进行预测，这些因素都将会影响预测准确度，后续需要综合考虑深入研究。

（2）由于本文没有真正实现视口数据端到端的传输流程，所以在资源调度方面，本文只是使用数据量较大、时延阈值小这两个特征来代表全景视频视口数据进行调度，没有考虑全景视频的其他特性，如视口区域数据和非视口数据的重要性差别；没有考虑视频 I/P/B 帧的重要性差别，应当提高这些重要区域的优先级。同时在仿真过程中只考虑了单小区多用户的场景，而实际中更多的是多小区场景，存在小区间干扰等现象，之后应当更加紧密结合全景视频的特性，针对多小区的场景，进一步优化资源调度算法的性能。

（3）本文在研究资源调度算法时聚焦于 LTE 网络，而在研究后期，5G 网络的发展逐渐加快，相关技术研究也不断完善，未来 5G 网络的覆盖将改善全景视频的传输问题，因此后续可以考虑结合新型网络架构与技术，进一步优化全景视频的传输。

参考文献

- [1] 孟祥翔. 大视场虚拟现实头盔显示器光学系统研究[J]. 2015.
- [2] http://www.gov.cn/xinwen/2019-06/20/content_5401891.htm
- [3] Jerald J. Scene-motion-and latency-perception thresholds for head-mounted displays [D]. North Carolina: The University of North Carolina at Chapel Hill, 2009.
- [4] 赖立冬. 全景视频基于视点的自适应传输技术研究[D]. 北京邮电大学, 2018.
- [5] L. Xie, Z. Xu, Y. Ban, X. Zhang. 360probdash: Improving qoe of 360 video streaming using tile based http adaptive streaming[C]. ACM Multimedia, 2017.
- [6] Yanan Bao, Huasen Wu, Tianxiao Zhang, et al. Shooting a Moving Target: Motion-Prediction-Based Transmission for 360-Degree Videos[C]. 2016 IEEE International Conference on Big Data. IEEE, 2016.
- [7] C. Cortes, VN Vapnik. Support Vector Networks[J]. Machine Learning, 1995, 20(3):273-297.
- [8] Feng Qian, Lusheng Ji, Bo Han, et al. Optimizing 360 video delivery over cellular networks[C]. the 5th Workshop, ACM, 2016.
- [9] Ban Y, Xie L, Xu Z, et al. CUB360: Exploiting Cross-Users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming[C]. 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2018.
- [10] Petrangeli, Stefano, Simon. Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos[J]. 2018, 157-160. 10.1109/AIVR.2018.00033.
- [11] Xie, Lan, Zhang, Xinggong, et al. CLS: A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming[C]. the 9th ACM on Multimedia Systems Conference (MMSys), 2018.
- [12] M. Cornia, L. Baraldi, G. Serra, et al. A deep multi-level network for saliency prediction[C]. In Pattern Recognition (ICPR), 2016 23rd International Conference on, pages 3488–3493. IEEE, 2016.
- [13] Jetley S, Murray N, Vig E. End-to-End Saliency Mapping via Probability Distribution Prediction[J]. 2018.
- [14] J. Pan, E. Sayrol, X. Giro-i-Nieto, et al. Shallow and deep convolutional networks for saliency prediction[C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

- [15]J. Pan, C. Canton, K. McGuinness, et al. Salgan: Visual saliency prediction with generative adversarial networks. arXiv preprint arXiv:1701.01081, 2017.
- [16]Battisti, Federica, Baldoni, et al. A feature-based approach for saliency estimation of omni-directional images[J]. Signal Processing Image Communication,2018.
- [17]Lebreton P, Raake A. GBVS360, BMS360, ProSal: Extending existing saliency prediction models from 2D to omnidirectional images[J]. Signal Processing Image Communication,2018,69:69-78
- [18]Monroy, Rafael, Lutz, et al. SalNet360: Saliency maps for omni-directional images with CNN[J]. Signal Processing Image Communication, 2018,69:26-34
- [19]Fan C, Lee J, Lo W, et al. Fixation prediction for 360° video streaming in head-mounted virtual reality[C]. Proceedings of the 27th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video. 2017.
- [20]Mai X, Yuhang S, Jianyi W, et al. Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2018:1-1.
- [21]Yang Q, Zou J, Tang K, et al. Single and Sequential Viewports Prediction for 360-Degree Video Streaming[C]. 2019 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2019.
- [22]Y. Xu, Y. Dong, J. Wu, et al. Gaze prediction in dynamic 360 immersive videos[C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.
- [23]Zhu X J, Zeng H, Huang K, et al. Round-robin based scheduling algorithms for FIFO IQ switch[C]. Proceedings of the IEEE International Conference on Networking, Sensing and Control, ICNSC, 2008.
- [24]Cui Chunfeng, Du Lei, Zhang Ping. A CDMA based scheduling algorithm with IP QoS guarantee[C]. Vehicular Technology Conference, 2003.
- [25]Jin-Ghoo Choi, Saewoong Bahk. Cell-Throughput Analysis of the Proportional Fair Scheduler in the Single-Cell Environment[J]. IEEE Transactions on Vehicular Technology, 2007, 56(2):766-778.
- [26]Kargahi M, Movaghar A. A Method for Performance Analysis of Earliest-Deadline-First Scheduling Policy[J]. Journal of supercomputing, 2006, 37(2):197-222.

- [27] P. Ameigeiras, J. Wigard, P. Mogensen. Performance of the M-LWDF scheduling algorithm for streaming services in HSDPA[C], Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th. IEEE, 2004.
- [28]Mahfoudi M, Bekkali M E, Najd A, et al. A New Downlink Scheduling Algorithm Proposed for Real Time Traffic in LTE System[J]. International Journal of Electronics and Telecommunications, 2015, 61(4).
- [29]冯卿. LTE 中多业务优先级调度算法的研究与改进[J].电子制作, 2014.
- [30]Alfayly A, Mkwawa I H. QoE-based performance evaluation of scheduling algorithms over LTE[C]. Globecom Workshops (GC Wkshps). IEEE, 2012.
- [31]Oyman O, Singh S. Quality of experience for HTTP adaptive streaming services[J]. IEEE Communications Magazine, 2012, 50(4): 20-27.
- [32]王一迪.面向 VR 视频传输的无线网络跨层优化[D]. 北京邮电大学, 2019
- [33]Ge X, Pan L, Li Q, et al. Multipath Cooperative Communications Networks for Augmented and Virtual Reality Transmission[J]. IEEE transactions on multimedia, 2017, 19(10):2345-2358.
- [34]邓瑞. 虚拟现实视频无线传输研究现状及发展动态分析[J]. 移动通信, 2019, 43(03):58-67.
- [35]Ioan-Sorin Comşa, Zhang S, Aydin M E, et al. A novel dynamic Q-learning-based scheduler technique for LTE-advanced technologies using neural networks[C]. Local Computer Networks (LCN), 2012 IEEE 37th Conference on. IEEE, 2012.
- [36]Dickey, David. Time Series Theory and Methods[J]. Technometrics, 1989, 31(1):121-121.
- [37]Dennis Arjo. Statistical Models: Theory and Practice[J]. Technometrics, 2009, 48(2):315-315.
- [38]Ho Tin. Random decision forests[C]. Document Analysis and Recognition,1995.
- [39]B. W. White, Frank Rosenblatt. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms[J]. The American Journal of Psychology, 1963, 76(4):705.
- [40]Oancea Bogdan, Ciucu Stefan. Time series forecasting using neural networks. 2013.
- [41]Peter G, Zhang. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003.
- [42]Sahoo G B, Ray C. Flow forecasting for a Hawaii stream using rating curves and neural networks[J]. Journal of Hydrology, 2006, 317(1-2):0-80.

- [43] Celik A E , Karatepe Y . Evaluating and forecasting banking crises through neural network models: An application for Turkish banking sector[J]. Expert Systems with Applications, 2007, 33(4):809-815.
- [44] Schmidhuber, Jürgen. Deep learning in neural networks: An overview[J]. Neural Networks, 2015, 61:85-117.
- [45] Hornik K. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5):359-366.
- [46] David Erwan, Gutiérrez Jesús, Coutrot Antoine, et al. A dataset of head and eye movements for 360° videos[C]. the 9th ACM. ACM, 2018
- [47] Corbillon X, F. De Simone, G. Simon. 360-degree video head movement dataset[C]. the 8th ACM. ACM, 2017.
- [48] Wu C, Tan Z, Wang Z, et al. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming[C]. the 8th ACM. ACM, 2017.
- [49] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural computation, 1997, 9(8):1735-1780.
- [50] Xu Tan, Qian Feng, Han Bo. Content Assisted Viewport Prediction for Panoramic Video Streaming. 2019
- [51] Cavalcanti, Porto F R. Resource Allocation and MIMO for 4G and Beyond[J]. 2014, 10.1007/978-1-4614-8057-0.
- [52] Piro, G, Grieco, L.A, et al. Simulating LTE Cellular Systems: An Open-Source Framework[J]. IEEE Transactions on Vehicular Technology, 2011, 60(2):498-513.

致谢

时光如白驹过隙，转眼间充实又紧张的三年研究生生涯即将告一段落，之后将要告别校园步入职场，论文行文至此，内心不禁思绪万千。回顾这三年来，收到过许多人的关心与帮助，在此向激励我、支持我、帮助我，让我不断成长的老师、家人、同学和朋友致以我最真诚的感谢。

首先由衷地感谢我的导师。老师在学术上对我帮助甚多，从如何高效地查找科研资料、如何正确地阅读参考论文到如何进行科研实验，指导我从只会学习的本科阶段顺利过渡到可以独立科研的研究生阶段；针对我的研究课题，老师总能提出关键性问题，在研究瓶颈期引导我开拓思路，正是老师的谆谆教导才让我取得了一定的科研成果；此外，老师严肃认真的科研态度也不断影响着我，真的很幸运可以遇到这样一个良师。

其次，我要感谢老师。在科研学习与研究生生活中老师也给予了我许多的支持与帮助。在准备发表论文成果期间，老师细致耐心地辅导我撰写论文，纠正论文中的语法错误，给我提出了许多有用的修改建议。另外我也要感谢老师，老师积极组织我们参加科研讲座、参与科研项目、阅读最新研究成果，正是因为有老师的督促和鼓舞才扩宽了我科研的方向。当然，每一位老师都给予过我各种各样的帮助，再次表达我最诚挚的感谢。

由衷感谢实验室的每一位小伙伴。感谢师兄，两位师兄的热情与风趣让我迅速融入了这个大家庭，感谢你们在科研、工作和生活上的建议，祝福你们工作顺利；感谢小伙伴，是你们的陪伴才让紧张枯燥的科研生活变得精彩缤纷，尤其感谢同学，我们工位紧挨一起，一起科研，一起吃饭，一起出差开会，我们一起度过了许许多多快乐的时光，衷心祝福你们可以心想事成；感谢可爱的师弟师妹们，你们的加入让我们这个家庭不断扩大，祝福你们在科研道路上越走越顺畅。

感谢我的室友，我们来自五湖四海，十分幸运与你们成为了亲密无间的室友，感谢你们在我科研困顿的时候，在我找工作迷茫的时候给予我最坚定温暖的安慰与支持，感谢你们的参与给我的研究生生涯留下了无数快乐的美好回忆，祝福你们未来可期，前程似锦。

最后，要感谢默默支持我关心我的家人，让我可以毫无顾忌的追求自己的梦想，祝福家人身体健康，万事如意。

攻读学位期间发表的学术论文

- [1] A Novel Downlink Scheduler Based on Q-Learning for Video Traffic in LTE Networks. 115-119. 10.1109/ICNIDC, 2018.