electronics

MDPI

Article

MNCATM: A Multi-Layer Non-Uniform Coding-Based Adaptive Transmission Method for 360° Video

Xiang Li [1], Junfeng Nie [2,*], Xinmiao Zhang [3], Chengrui Li [4], Yichen Zhu [2], Yang Liu [2], Kun Tian [5] and Jia Guo [2]

[1] College of Politics and Public Administration, Tianjin Normal University, Tianjin 300387, China; yylixiang@tjnu.edu.cn
[2] College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China; 2311090070@stu.tjnu.edu.cn (Y.Z.); 002909@tjnu.edu.cn (Y.L.); c04s316@bupt.cn (J.G.)
[3] Office of Network Security and Information Technology, Tianjin Normal University, Tianjin 300387, China; zxm1983@tjnu.edu.cn
[4] School of Software, Tiangong University, Tianjin 300387, China; 2111320108@tiangong.edu.cn
[5] CS Department, California State University Chico, Chico, CA 95929, USA; ktian@csuchico.edu
* Correspondence: 2130200069@stu.tjnu.edu.cn

Abstract: With the rapid development of multimedia services and smart devices, 360-degree video has enhanced the user viewing experience, ushering in a new era of immersive human–computer interaction. These technologies are increasingly integrating everyday life, including gaming, education, and healthcare. However, the uneven spatiotemporal distribution of wireless resources presents significant challenges for the transmission of ultra-high-definition 360-degree video streaming. To address this issue, this paper proposes a multi-layer non-uniform coding-based adaptive transmission method for 360° video (MNCATM). This method optimizes video caching and transmission by dividing non-uniform tiles and leveraging users' dynamic field of view (FoV) information and the multi-bitrate characteristics of video content. First, the video transmission process is formalized and modeled, and an adaptive transmission optimization framework for a non-uniform video is proposed. Based on this, the optimization problem required by the paper is summarized, and an algorithm is proposed to solve the problem. Simulation experiments demonstrate that the proposed method, MNCATM, outperforms existing transmission schemes in terms of bandwidth utilization and user quality of experience (QoE). MNCATM can effectively utilize network bandwidth, reduce latency, improve transmission efficiency, and maximize user experience quality.

Keywords: 360-degree video; multi-layer non-uniform coding; QoE ; FoV

check for updates

Citation: Li, X.; Nie, J.; Zhang, X.; Li, C.; Zhu, Y.; Liu, Y.; Tian, K.; Guo, J. MNCATM: A Multi-Layer Non-Uniform Coding-Based Adaptive Transmission Method for 360° Video. *Electronics* **2024**, *13*, 4200. https://doi.org/10.3390/electronics13214200

Academic Editor: Krzysztof Okarma

Received: 8 October 2024
Revised: 21 October 2024
Accepted: 23 October 2024
Published: 26 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, with the rapid advancement of multimedia services and smart devices, two emerging technologies—virtual reality (VR) 360-degree video and augmented reality (AR)—have introduced a new era of immersive human–computer interaction. These technologies have garnered increasing attention within the Internet industry and are gradually being integrated into various aspects of daily life, including gaming, education, and healthcare, where they play a crucial role. According to a recent press release by the International Data Corporation (IDC), global shipments of VR and AR headsets are projected to grow by 46.4% in 2024, with AR product shipments reaching 845,000 units, reflecting an 85.6% increase compared to 2023 [1]. Users can experience 360-degree panoramic videos by wearing head-mounted displays (HMDs) such as Apple Vision Pro, HTC Vive, and Oculus, and can adjust their field of view (FoV) by rotating their heads [2].

High-resolution 360-degree video can provide users with an immersive experience. However, the transmission of ultra-high-definition 360-degree video streaming faces significant challenges: ultra-high bandwidth requirements and ultra-low latency (the delay between head movement and video rendering) requirements. First, compared to traditional

2D videos, the bandwidth required for 360-degree videos is an order of magnitude larger due to its spherical nature, which results in a large amount of data needing to be stored and transmitted. Additionally, because the display is very close to the user's eyes (usually only a few centimeters away), high spatial resolution is necessary. For example, providing a video with a resolution of 8000 and a temporal resolution of at least 90 Hz ensures user comfort [3]. Second, the uneven spatiotemporal distribution of wireless resources becomes a major issue, as wireless resources become severely constrained when a large number of users simultaneously access the network.

To address these challenges, edge caching [4] and edge computing [5] technologies have been proposed to optimize the transmission of 360-degree videos. By proactively storing content that users may request at network edge nodes, edge caching reduces the burden on network backhaul traffic, improves network throughput and user experience, and lowers deployment costs and energy consumption [6]. In this edge caching transmission architecture, the original video is provided by the edge server, which runs algorithms to predict the user's subsequent video data and requests video content from the media server, caching it at the edge server in advance. The cached video is then transmitted to the user through the edge server, thereby reducing the load on the core network [7]. Additionally, since users can only view the images within their FoV when watching panoramic videos, prioritizing the transmission of video content within the FoV has been proposed [8]. Some studies have designed tile-based caching strategies [9–16], where 360-degree videos are encoded into independently encoded segments, meaning that the videos are spatially divided into non-overlapping rectangular areas called tiles [17].

Recently, BT and EE have conducted several trials in the UK on 5G technology and its application in 360-degree video transmission, aiming to launch commercial services based on 5G. For example, BT Sports used 360-degree VR technology to broadcast live English Premier League matches, providing viewers with an immersive experience (as described in BT Sports' live coverage [18] and BT Sports 360-degree VR live broadcast [19]). These cases demonstrate the industry's demand and application potential for adaptive tile-based 360-degree immersive and interactive visual content streaming technology. Through these practical applications, BT and EE have made positive contributions to promoting the development of 5G technology and improving user experience.

However, the aforementioned transmission optimizations for 360° video are based on uniform tile partitioning, which presents certain issues. For instance, if the tile size is too small, it increases the overall transmission volume, while if the tile size is too large, it leads to increased latency during viewpoint switching. One of the problems this paper aims to address is how to ensure the video quality in regions where user attention is concentrated while balancing the overall video size and maintaining the flexibility of viewpoint switching. Although edge computing and edge caching can partially alleviate network transmission pressure, edge storage capacity and computing resources remain limited. On one hand, the dynamic nature of user preferences makes it challenging to design intelligent caching strategies, specifically determining which video content should be cached on edge servers. On the other hand, the algorithms of varying complexity impose different performance requirements on edge computing. One of the problems this paper aims to solve is how to pre-cache the necessary content at the edge to efficiently meet user demands.

In this paper, we propose a multi-layer non-uniform coding-based adaptive transmission method for 360-degree video (MNCATM). The non-uniform encoding of the 360° video, as shown in Figure 1, indicates that, when dividing into tiles, the tiles are partitioned into different sizes. This transmission strategy takes into account each user's attention mechanism, dynamic FoV information, and the multi-bitrate nature of video content, fully leveraging available network bandwidth, reducing latency, and maximizing user quality of experience (QoE) [20]. The main contributions of this paper are summarized as follows:

- This paper proposes a non-uniform 360-degree video encoding scheme, where the viewing screen is divided into regions of different sizes based on the varying distribution of user attention.

- This paper introduces an adaptive transmission architecture based on non-uniformly encoded multi-layer for 360° video, mathematically modeling the adaptive transmission process, and formulating the optimization problems that need to be addressed. Finally, the MNCATM method is proposed to solve the optimization problem, fully utilizing the available network bandwidth, thereby avoiding resource wastage while ensuring optimal user's QoE.
- In the simulation experiments, the proposed scheme is compared and evaluated against four other transmission schemes, and the impact of key performance indicators is analyzed. The results demonstrate that MNCATM outperforms in terms of bandwidth utilization and user's QoE.

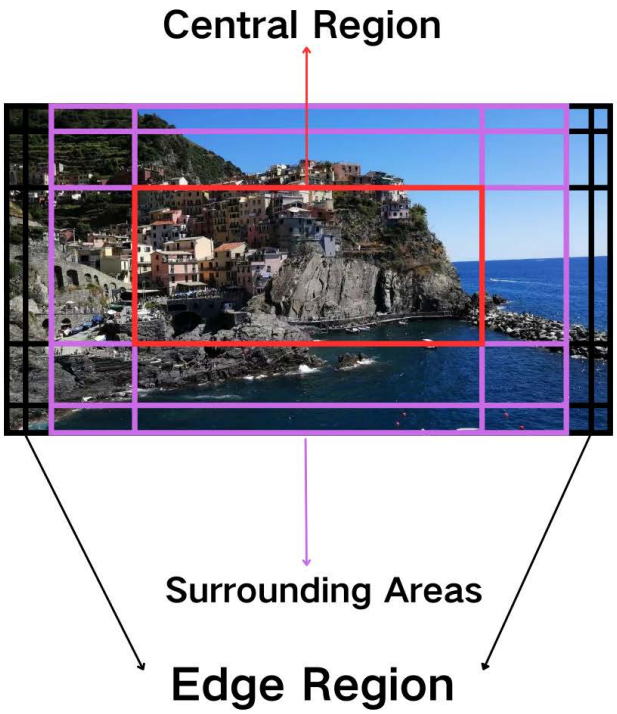


Figure 1. Non-uniformly encoded 360-degree video.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 proposes an uneven 360-degree video screen partitioning scheme and redefines the parameters for evaluating QoE. In Section 4, a collaborative coding adaptive transmission strategy based on multi-layer non-uniform coding and tiles is proposed. Section 5 analyzes the complexity of the algorithm and the real-time feasibility at the edge, and Section 6 presents the simulation experiments and discusses the results. Finally, the paper is concluded in Section 7.

2. Related Work

Extensive research has been conducted on optimizing the delivery of 360-degree videos, yielding significant advancements. These efforts can be categorized as follows:

2.1. Optimizations of Caching Techniques

2.1.1. Edge Caching

X. Zhang et al. proposed a learning-based edge caching scheme (LECS), which modeled the cooperative content caching problem as an optimization problem, using temporal convolutional networks to model cooperative edge caching, and passed dynamic programming proposes a near-optimal cache location scheme [21]. Liu et al. proposed an online caching algorithm (VIE) for the optimization of 360° video streaming in mobile edge caching systems. In terms of caching strategy, a caching algorithm VIE with the unknown

future requested content is designed, which can effectively improve the content hit rate and reduce the time of prediction, calculation, and transmission [22].

2.1.2. DRL-Based Caching

Yang et al. designed a collaborative edge caching and transcoding solution for 360° video streaming based on deep reinforcement learning (DRL) algorithms. This solution models the collaborative transcoding and caching process as a Markov decision process (MDP), and uses the deep deterministic policy gradient (DDPG) method to obtain cache replacement and computing resource allocation strategies, and designs a horizon-oriented scheme to accelerate the training process [2]. P. Maniotis and N. Thomos proposed a viewport-aware DRL method for 360° video caching, introduced the new concept of a virtual viewport and proposed an active caching scheme. The content placement problem of 360° video content in the edge cache network is modeled as an MDP, and then a Deep Q-Network (DQN) algorithm is used to determine the optimal cache placement scheme [4].

2.1.3. Collaborative Caching

Fu and their team proposed a joint communication–computation–caching (3C) optimization strategy that first analyzes user interests using reviews collected from users' local devices to predict their request probability for all 360° content. Next, the optimization problem is decomposed into joint caching and calculation sub-problems and bandwidth allocation sub-problems, and these two sub-problems are solved separately through alternating iterative methods to obtain the optimal joint 3C optimization strategy [23]. J. Xia et al. proposed a 360° video transmission and caching method based on the cooperation of mobile edge computing (MEC) servers. They considered a multi-MEC 5G architecture and proposed an optimized k-shortest paths (OKSP) algorithm to solve the cooperative caching problem of multiple MEC servers [24].

2.1.4. Proactive Caching and Viewport Prediction

Viewport prediction is a novel transmission scheme [25,26]. Q. Cheng et al. proposed a 360-degree mobile virtual reality video (MVRV) streaming solution that comprehensively considers video encoding, active caching, computation offloading, and data transmission. In order to solve the end-to-end delay problem, a convolution neural network (CNN) + long short-term memory (LSTM) + Gaussian mixture model (GMM) is used in advance to obtain the saliency mapping, and the calculation task is offloaded to the MEC server, so that the tiles with the highest probability are actively cached in the MEC server [27]. J. Yang et al. proposed a cloud-edge-end collaborative caching method based on graph learning, designed an active collaborative caching algorithm using graph neural network (GNN) to predict user needs, and proposed a cache update strategy. In the cloud-edge collaborative service architecture, background content and interactive content are processed through independent encoding mechanisms to achieve efficient and low-latency 360° content distribution. Experiments have proven that this GNN model is more accurate than LSTM and motion-based prediction methods [28].

2.2. Optimization of Multicast Delivery Technology

2.2.1. Tile-Based Multicasting

Taking into account the heterogeneity of users, K. Long et al. proposed a method to optimize the wireless transmission of multi-quality block 360-degree videos. They considered two field-of-view quality variation requirements: the absolute smoothness requirement (all tiles within the field of view must be of the same quality level) and relative smoothness requirement (tiles can vary in quality within a specified range), as well as two video playback modes: direct-playback mode and transcode-playback mode. In addition to natural multicast opportunities, this approach introduces two new multicast opportunities: relatively smooth-enabled multicast opportunities and transcoding-enabled multicast opportunities. For these four situations with different quality change requirements and video

playback modes, the transmission resource allocation, playback quality level selection, and transmission quality level selection are optimized by maximizing the use of potential multicast opportunities, thereby minimizing energy consumption. By comparing the optimal values in these four cases, it was demonstrated that the energy consumption of wireless 360° streaming is reduced when more multicast opportunities can be exploited [29]. T. Okamoto et al. introduced a new edge-assisted multi-user 360-degree video transmission scheme and proposed a bit allocation algorithm based on available bandwidth tiles. Redundant video transmission is reduced by dynamically adjusting each user's video quality in edge servers to accommodate changes in their visual viewpoint, and by employing hybrid unicast and multicast tile delivery. The resulting viewport normalized peak signal-to-noise ratio (PSNR) and associated traffic are evaluated in simulation experiments, demonstrating the benefits of this solution compared to uniform bit allocation schemes [30].

2.2.2. Proactive Multicasting and Viewport Prediction

D. Nguyen et al. proposed a new method that combines scalable video coding and multicast, first using linear regression (LR) to produce tile weight estimates, and then encoding the tiles into multiple layers through scalable video coding (SVC) technology and transmitting them to user [31–33]. The authors proposed two online algorithms, one for deciding the appropriate tile hierarchy each user should receive and its transmission mode, and another for allocating network resources. The simulation results show that the proposed method can significantly improve the average viewport quality compared to existing techniques [34].

2.2.3. Multicasting in Innovative 5G Networks

Zhong et al. studied a framework called MEC-DC, and a new buffer evolution model. They adopt a scenario that includes one macro-cell-based station (MBS) and multiple small-cell-based stations (SBSs) with edge computing capabilities, as well as mobile users with device-to-device (D2D) communication enabled between them, and model the multicast problem in 5G-HetNet as one that maximizes the average buffer level of all viewers. There is also a charging issue with the random buffer. Furthermore, they describe a multicast-aware transcoding offloading algorithm (MATO) to optimize multicast and transcoding tasks, and a crowd-assisted delivery algorithm (CAD) to supplement the discussed algorithm in case of segment loss. It turns out that the MATO-CAD solution can find near-optimal buffer sizes. To our knowledge, this is the first study to improve real-time 360° service quality through innovative joint computing and transmission resource allocation in 5G-HetNet [35]. Chakareski et al. proposed an optimized 360° real-time video streaming method for heterogeneous mobile 360° clients in 5G networks, they use scalable tiling compression to process 360° video, and use edge servers to adapt the data to the transmission rate and computing power of each client, using formal rate-distortion calculations for optimization. The single-connectivity client receives the base content layer, while the dual-connectivity client receives the content layer on both LTE and NR (new radio) connections, and uses a geometric programming optimization strategy to reduce complexity. This method shows significant performance improvements over existing technologies and is robust to inaccurate user navigation predictions, temporary loss of NR links, LTE bandwidth changes, and diverse 360° video content [36].

Although the aforementioned research has significantly advanced the efficient transmission of 360-degree VR videos, these methods are often based on uniform tile partitioning, which lacks flexibility in balancing video quality and viewpoint switching. They fail to fully consider the uneven distribution of user attention and its changes in dynamic viewing environments. Therefore, this study proposes a non-uniform encoding and transmission method based on multi-layer coding. By dividing the video into non-uniform regions and optimizing caching and transmission strategies according to the user's dynamic viewpoint information and the characteristics of multi-bitrate video content, this method fully utilizes the available network bandwidth, reduces latency, and enhances the user's QoE. Experi-

mental results demonstrate that MNCATM outperforms traditional methods (such as LFU, LRU, VIE, and 3C) in key performance indicators, including bandwidth utilization, PSNR, and structural similarity (SSIM). These results not only verify MNCATM's advantages in handling complex dynamic scenes but also highlight its effectiveness in improving the user's QoE. Consequently, the MNCATM method not only fills the gap in current research but also provides new insights and solutions for optimizing 360-degree video transmission in the future.

3. System Model

This section introduces the system model, covering three main aspects: network model, content model, and caching model. The overall delivery process is divided into four key stages: request, caching, computation, and transmission.

3.1. Network Model

As depicted in Figure 2, the 360° video service delivery system architecture is composed of three primary components: the server, the base station, and the user device. These components work together to provide users with a seamless 360° video experience. The user device, typically an HMD, receives 360° video data transmitted by the base station, decodes, and plays back the content to create an immersive 360° experience. The algorithm dynamically adjusts the quality of buffered 360° video in different regions in real time based on the state of the network channel, thereby enhancing the user's QoE. The 5G base station is equipped with an edge computing module. Assuming that the algorithm proposed in this paper runs within the base station, it benefits from powerful computational capabilities and can pre-fetch cached data from the video server, thereby reducing stuttering caused by network jitter.

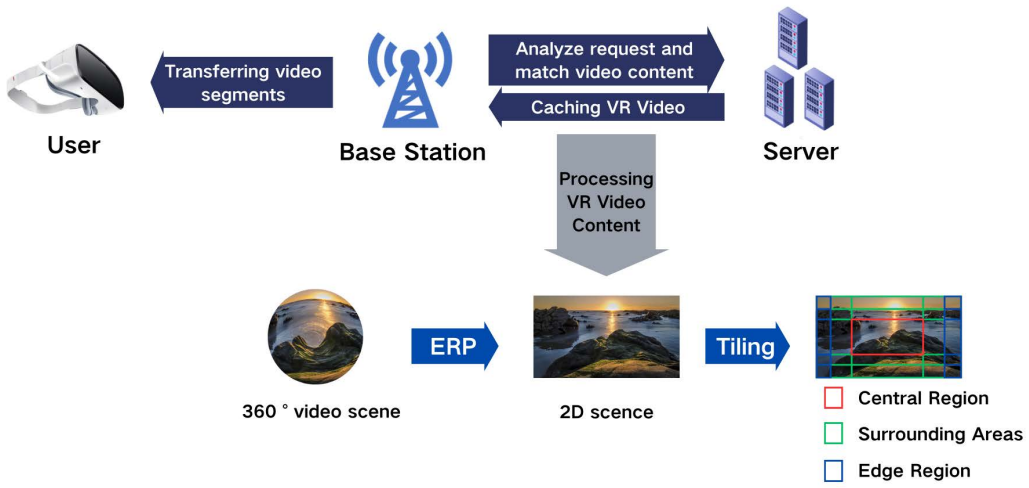


Figure 2. A 360° video service scenario architecture.

The server stores a vast repository of high-quality 360° video content, catering to the diverse demands of users. Before the base station retrieves the video data, the server preprocesses the 360° content by projecting the 3D images onto a 2D plane, dividing these into three distinct viewing areas, and segmenting these areas into smaller blocks. In addition, the multi-layer non-uniform encoding technology is used to encode video content into multiple layers, including a base layer and multiple enhancement layers. The base layer provides the lowest quality video data, including the basic frame rate, resolution, and quality of the video; the enhancement layer is a supplement to the base layer, which improves the visual effect of the video by increasing the frame rate, resolution, or quality of the video, and each enhancement layer provides incremental data on top of the base layer. Therefore, multi-layer non-uniform coding technology provides flexible bandwidth adaptability and terminal device compatibility, providing strong support for QoE optimization in 360° video transmission.

3.2. Content Model

The 360° video content is pre-cached at the base station based on the algorithm proposed in this paper. Consider a video library I containing a set of 360° videos with cardinality $|I|$. Each 360° video V in the video library is evenly divided into n segments, each of equal duration, denoted by $V = \{S_1, S_2, S_3, \ldots, S_n\}$, where S_i represents the i-th segment of video V. As illustrated in Figure 2, this study adopts the equirectangular projection (ERP) format, which is widely used for 360° video, to project the spherical 3D content onto a 2D rectangular plane [37].

Each video segment is further divided into three regions, denoted by $S_i = \{T_1, T_2, T_3\}$. Here, T_1 represents the central region, the primary focus of the user's field of view; T_2 covers the surrounding areas; and T_3 corresponds to the edge region, which attracts the least user attention. Each region is non-uniformly encoded into non-overlapping rectangular blocks of varying sizes known as tiles, denoted by $T_1 = \{t_1, t_2, \ldots, t_{i_1}\}$, $T_2 = \{t_1, t_2, \ldots, t_{i_2}\}$, and $T_3 = \{t_1, t_2, \ldots, t_{i_3}\}$.

The objective of this study is to maximize the user's quality of experience, denoted by Max(QoE). In this paper, QoE is defined as the sum of the QoE values across the three regions, as shown in Equation (1). The QoE of each region is calculated by multiplying the video quality weight (α) of that region by the sum of the QoE values of all tiles within the region, as detailed in Equation (2). The higher the weight, the more important the region, and the greater its impact on QoE.

$$\mathrm{QoE} = \sum_{i=1}^{3} \mathrm{QoE}_{T_i}, \tag{1}$$

$$\mathrm{QoE}_{T_i} = \alpha_i \cdot \sum_{j=1}^{m} \mathrm{QoE}_{t_j}, \quad i \in [1,3], \tag{2}$$

$$\mathrm{QoE}_{t_j} = \mathrm{QoS}_{t_j} - \beta \cdot k. \tag{3}$$

The QoE for each tile is derived from its objective quality minus the jitter, where β controls the trade-off between video quality stability, and k represents the current video quality stability, as expressed in Equation (3). QoS represents the objective quality of the video, which is typically measured using PSNR or SSIM.

$$k = 1 - \frac{\sum_{i=1}^{N} \left(\left| R_{\max} - Q_i' \right| * \phi(x) \right)}{\sum_{i=1}^{N} (R_{\max} * \phi(x))}, \tag{4}$$

where the parameter k represents the instability index, which is the weighted sum of all switch steps in the previous segments divided by the weighted sum of the highest received rate level during the transmission time. In Equation (4), N denotes the number of segments, while $R_{\max}$ refers to the highest received rate level of the transmitted video during transmission time. The term Q_i' corresponds to the rate value related to the time segment. Additionally, $\phi(x) = N - i$ assigns higher penalties to more recent bit-rate switches. A value of k closer to 1 indicates better stability.

Let K represent the evaluation of the jitter for each tile in the 360° video transmission. The overall jitter of the 360° video is calculated by averaging the jitter of all tiles within different regions, weighted by the relevant parameters. These parameters correspond to the trade-off parameters in the QoE evaluation for each region.

$$K = \sum_{i=1}^{3} \alpha_i \cdot \frac{1}{N_i} \sum_{j=1}^{N_i} k_{ij} \tag{5}$$

where k_{ij} is the jitter of the j-th tile in region i, N_i is the number of tiles in region i, α_i is the weight parameter for region i, reflecting its importance in the QoE calculation. The symbols involved in the model can be found in Table 1.

Table 1. The meaning of notation.

Symbol	Notations
QoE	The subjective video quality perceived by the user.
k	The instability index of video playback.
I	Video library
S_i	The i-th segment of the 360° video
V	A 360° video
T_i	The 3 areas divided in the screen
t_m	The m th tile
α_i	The weights for each region
$Bitrate$	The bit rate of the tiles transferred
$bandwidth_t$	Network bandwidth at time t

3.3. Caching Model

In the video service delivery process, ensuring efficient video transmission to users necessitates careful consideration of network bandwidth constraints. The bitrate of each video tile is determined by the relationship function between the tile's bitrate and its objective quality, as described in Equation (6).

$$\mathrm{QoS}_{t_j} = f(\mathrm{Bitrate}), \tag{6}$$

where f represents the function that corresponds the bitrate to the objective quality of the video.

$$\mathrm{Bitrate} = \sum_{j=1}^{i_n} Bitrate_j, \tag{7}$$

n represents the number of tiles that need to be cached currently. $Bitrate_j$ represents the bitrate of tile j.

The bitrates of all tiles are summed to obtain the total bitrate for the transmitted video segment, ensuring that this total does not exceed the available network bandwidth, as outlined in Equation (8).

$$\mathrm{bandwidth}_t \geq Bitrate \tag{8}$$

The current network bandwidth is defined as the average of the bandwidth measurements taken at the previous three instances, as represented by Equation (9).

$$\mathrm{bandwidth}_t = \frac{\mathrm{bandwidth}_{t-1} + \mathrm{bandwidth}_{t-2} + \mathrm{bandwidth}_{t-3}}{3} \tag{9}$$

Consequently, the optimization objective of this paper is as follows:

$$\begin{aligned} &\mathrm{maximize}(QoE) \\ &\text{subject to Equations (1)–(9)} \end{aligned} \tag{10}$$

4. Algorithm

The transmission algorithm is shown in Algorithm 1. In the first stage, as shown in lines 3–7, this paper preprocesses the video segment and converts it into a three-dimensional matrix. Then, the matrix is looped to calculate the efficiency of different layers of each tile, as shown in lines 8–14. Finally, we sort all the calculated results and set the limiting conditions to determine the final transmission plan, as shown in lines 15–24.

Algorithm 1 MNCATM algorithm.

Require: The bandwidth: B, the bitrate: b. $p_1 = 1$.

1: **for** $p_2 = 1$ to 10 **do**
2: **for** $p_3 = 1$ to 10 **do**
3: **for** $i = 1$ to n **do**
4: Project each video segment to 2D plane and divide into 3 regions $\{T_i, 1 \leq i \leq 3\}$
5: Divide the video frame into tiles non-uniformly and put them into a $m \times n$ matrix V $\{t_{11}, t_{12}, \ldots, t_{mn}\}$
6: Encode each tile with multi-layer non-uniform coding into one base layer and $(o-1)$ enhancement layers, turning matrix V into 3D $\{m \times n \times o\}$
7: **end for**
8: **for** j in V **do**
9: $QoS = f(b)$
10: $QoE = QoS - \beta k$
11: $QoE = \alpha n \times QoE$
12: $E = QoE/b$
13: Insert E into $arry[]$
14: **end for**
15: Sort $arry[]$ in descending order
16: **for** k in $arry[]$ **do**
17: **if** $B \geq b$ **then**
18: Transmit $arry[k]$
19: Delete $arry[k]$
20: $B = B - b$
21: **else**
22: Delete $arry[k]$
23: **end if**
24: **end for**
25: **end for**
26: **end for**
27: Select the optimal values of p_1, p_2, and p_3, which represent the number of tiles in different viewing regions, in order to achieve the best QoE.

In the first stage, it corresponds to lines 3–7 in the algorithm. First, traverse the n segmented video clips, project the screen of each video segment on a 2D plane, and divide it into three areas: the center area, the viewing area, and the edge area. Then, each video screen is non-uniformly encoded into rectangular blocks of non-fixed size and non-overlapping, namely tiles, and these tiles are stored in a $m \times n$ matrix V, where m represents the number of rows and n represents the number of columns, that is, a total of $m \times n$ elements. Next, multi-layer non-uniform coding is performed, and each tile is encoded as a base layer and $(o-1)$ enhancement layers, so that the matrix V becomes a three-dimensional matrix $(m \times n \times o)$, where o = 1 represents the base layer, o = 2 represents the base layer plus the first enhancement layer, o = 3 represents the base layer plus the first enhancement layer and the second enhancement layer, and so on, ending the loop.

In the second stage, corresponding to lines 8–14 in the algorithm, after creating the three-dimensional matrix V, each element in the matrix is traversed. Based on the user experience evaluation parameters defined above, the user media experience corresponding to each tile at different layers is estimated using parameters such as the bitrate. According to the area where each tile is located, the QoE is updated by multiplying it with the corresponding weight of that area. Then, the QoE is divided by the bitrate to obtain the efficiency value of the tile, which is stored in the array arry, ending the loop.

In the third stage, corresponding to lines 15–24 in the algorithm, the elements in the array arry are sorted in descending order, and the array is then traversed. A condition is set such that, if the current bandwidth is greater than or equal to the bitrate corresponding to the element, the element is transmitted. After transmission, the element is deleted, and the

bandwidth is updated (current bandwidth minus the transmitted bitrate). Otherwise, the element is deleted. The loop continues until the remaining network bandwidth is insufficient to meet the transmission of the remaining video segments.

The final step involves predicting the number of tiles in different viewing regions, calculating the optimal tile partitioning method, and determining the best QoE.

5. Computational Complexity Analysis

The MNCATM algorithm involves several stages that contribute to the overall computational complexity. We analyze the complexity of the algorithm and assess its feasibility for real-time execution at the edge.

5.1. Algorithmic Complexity

The computational complexity of the MNCATM algorithm is determined by the following components:

Each 360-degree video frame is divided into tiles of different sizes based on the regions of interest (RoIs), and each tile is encoded using multi-layer coding. Let n represent the number of tiles in the video frame, and o represent the number of layers per tile. The complexity of non-uniform partitioning and multi-layer encoding is $O(n \cdot o)$. This step is performed during preprocessing and does not affect real-time transmission, as the video content can be pre-encoded and cached prior to streaming.

To predict the user's field of view (FoV), the algorithm processes real-time data, such as head movement. Let m represent the number of parameters involved in the prediction (e.g., head orientation, speed). The complexity of this step is $O(m)$. This step must be executed in real-time to adapt the transmission to the user's changing perspective.

The algorithm optimizes the transmission strategy by sorting the tiles based on their quality-of-experience (QoE) efficiency, which accounts for user attention and tile bitrate. Sorting the n tiles requires $O(n \log n)$. After sorting, the tiles are transmitted in descending order of importance, ensuring optimal QoE for the user.

The algorithm evaluates the different combinations of tile numbers p_1, p_2, and p_3 (where p_1, p_2, and p_3 represent the number of tiles in different viewing regions) to find the best partitioning scheme. Given that p_1, p_2, and p_3 are iterated within a defined range (e.g., 1 to 10), this results in an additional complexity of $O(P_1 \cdot P_2 \cdot P_3) = O(10 \cdot 10 \cdot 10) = O(1000)$.

Thus, the overall time complexity of the MNCATM algorithm can be expressed as $O(P_1 \cdot P_2 \cdot P_3 \cdot (n \cdot o + m + n \log n)) = O(1000 \cdot (n \cdot o + m + n \log n))$ where n is the number of tiles, o is the number of coding layers per tile, m is the number of parameters for user perspective prediction, and P_1, P_2, P_3 are the range of values for tile partitioning.

5.2. Real-Time Feasibility at the Edge

To determine whether the MNCATM algorithm can be executed in real-time at the network edge, we evaluate the complexity of each phase relative to the capabilities of modern edge computing environments:

This phase has a complexity of $O(n \cdot o)$ and is performed offline during the preprocessing stage. Since this step is completed prior to transmission, it does not affect the real-time performance.

The complexity of $O(m)$ for real-time perspective prediction is manageable, as the number of parameters m is typically small. Edge computing devices, especially in 5G networks, are equipped to handle such real-time data processing tasks efficiently with minimal delay, allowing for real-time adaptation to user movements.

Sorting the tiles based on their QoE efficiency has a complexity of $O(n \log n)$. Given that the number of tiles n is moderate (e.g., tens or hundreds), the sorting operation can be performed quickly on edge servers with sufficient computational resources. Furthermore, the proximity of edge servers to end-users helps minimize latency, ensuring timely transmission.

The evaluation of different tile partitioning schemes introduces an additional complexity of $O(1000)$, which is a fixed factor. Given the computational capabilities of edge servers, this overhead is relatively small and does not significantly impact real-time performance.

In summary, the MNCATM algorithm is computationally feasible for real-time execution at the edge. The offline preprocessing of video content reduces the real-time computational burden, and the real-time components of the algorithm (such as user perspective prediction and transmission optimization) have manageable complexities that can be efficiently handled by edge computing devices. As a result, the algorithm is suitable for real-time deployment in edge computing environments, particularly in modern 5G networks where low latency and high bandwidth are available.

6. Experimental Results

To demonstrate the efficiency of MNCATM, we perform a series of trace-driven simulations. We first introduce the experiment settings and then show the results.

6.1. Experiment Settings

This paper designs three simulation scenarios, namely the case of good network bandwidth (wired network), the case of poor network bandwidth (such as on a high-speed train), and the case of normal network bandwidth (stationary state). Each case includes one user and two 360-degree videos, each of which has six different quality versions, including quality level, resolution, bit rate (kb/s), PSNR, and SSIM. Each 360 video is 1 min long and divided into 60 parts, that is, S = 60.

We divide each frame into $l \times h$ tiles, a total of N tiles, that is, $l \times h = N$, and each tile can be represented by a two-dimensional matrix as t_{lh}. Since the number of tiles in each frame is calculated in real time according to the algorithm, the number of tiles in each segment is different, that is, the values of l and h in each division are uncertain. Among them, we define the tiles in the center of the picture as the central area, denoted by T_1; other tiles in the FoV are defined as the surrounding area, denoted by T_2; the tiles close to the sides are the edge area, denoted by T_3.

In addition, we set the weights of T_1, T_2, and T_3 to 0.6, 0.3, and 0.1, respectively, to reflect the importance of different areas, i.e., $\alpha_1 = 0.6$, $\alpha_2 = 0.3$, and $\alpha_3 = 0.1$. The values of α_1, α_2, and α_3 are derived from the subjective experimental results of our team. Due to the variability in individual user experiences, these parameters can also be customized by users themselves. Through multi-layer non-uniform coding, each tile is encoded into a base layer and 4 enhancement layers, and each layer of each tile can be represented by a three-dimensional matrix as t_{lho}. Among them, $o = 1$ represents the base layer, $o = 2$ represents the base layer plus enhancement layer 1, $o = 3$ represents the base layer plus enhancement layer 1 and enhancement layer 2, and so on.

To verify the performance of the proposed MNCATM algorithm, this paper evaluates it from the following four aspects: QoE, SSIM, mean opinion score (MOS), and PSNR. QoE refers to the overall experience and satisfaction felt by users during 360° video transmission. For a better comparison of the results, the QoE values are normalized in this paper, as shown in Equation (11). SSIM is an indicator for measuring image quality, which is used to evaluate the degree of structural information loss during compression or transmission. MOS refers to collecting a group of users' evaluation scores on video quality and then using the average of these scores to represent the video quality. PSNR is a commonly used image quality evaluation metric, used to measure the quality of images after reconstruction or compression. Additionally, the MNCATM algorithm is compared with the four following 360° video caching transmission algorithms:

$$\mathrm{QoE}_{\mathrm{norm}} = \frac{\mathrm{QoE} - \mathrm{QoE}_{\mathrm{min}}}{\mathrm{QoE}_{\mathrm{max}} - \mathrm{QoE}_{\mathrm{min}}} \tag{11}$$

1. Least frequent use (LFU) [4]: The number of times each tile has been requested is recorded, and caching newly arrived tiles by removing the least frequently used tiles.

2. Least recently used (LRU) [4]: The last requested time of each tile is recorded, and caching newly arrived tiles by removing the least recently used tiles.
3. The recent victor download and recent failure deletion (VIE) [22]: By predicting users' viewing behavior, popular video tiles are cached, and infrequently used tiles are dynamically removed from the cache based on their popularity.
4. A joint communication–computation–cache (3C) optimization algorithm (3C) [23]: The overall optimization problem is decomposed into three sub-problems through the decomposition method: joint caching and computing problem, bandwidth allocation problem, and request probability solution problem.

6.2. Evaluation Results

In Figure 3, the QoE performance of the MNCATM algorithm is evaluated under normal network bandwidth conditions. The results demonstrate that the five algorithms exhibit similar trends across both video samples, with the QoE values of each algorithm fluctuating throughout the transmission process. Among them, MNCATM consistently outperforms the others, delivering higher QoE and exhibiting less fluctuation. The QoE for MNCATM predominantly remains within the range of 0.86–0.99, indicating its superior ability to provide an optimal user experience. In contrast, the QoE values for the LRU and LFU algorithms show significant fluctuations and are notably lower than those of the other three algorithms, with maximum QoE values reaching only up to 0.75. The QoE performance of the 3C and VIE algorithms is comparatively better, with QoE values fluctuating between 0.79 and 0.95, positioning them as the second best after MNCATM.

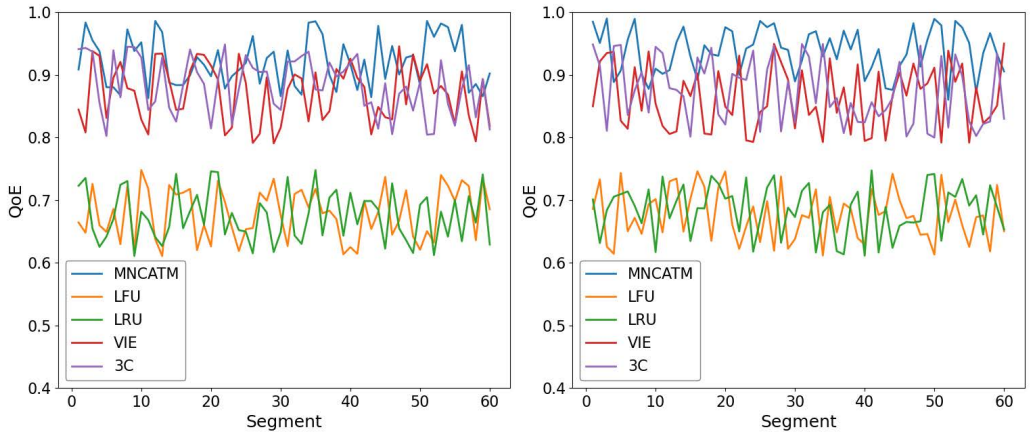
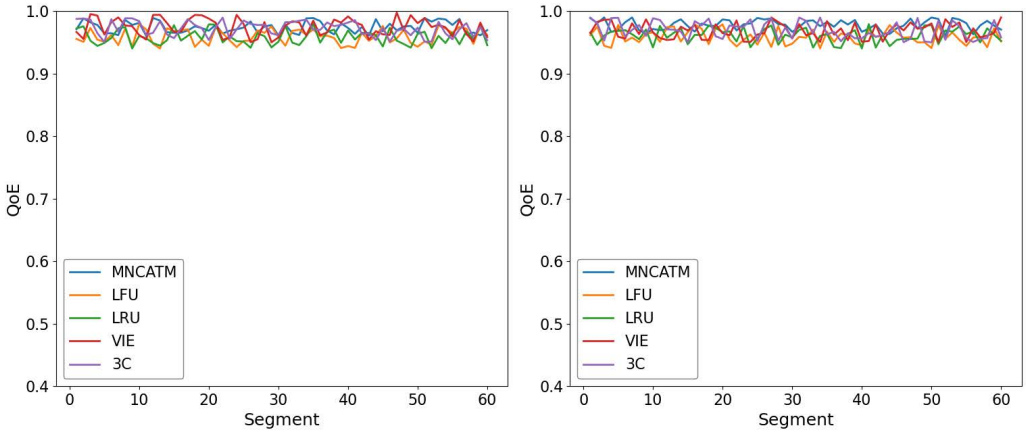


Figure 3. QoE comparison of different methods under normal bandwidth.

In Figures 4 and 5, we also compare the QoE performance of the five algorithms under good and poor network bandwidth conditions. When the network bandwidth is good, the QoE values of all algorithms are between 0.8 and 1.0, and the QoE values hardly fluctuate, showing good stability. Therefore, all algorithms can effectively support video transmission and provide high-quality QoE. On the contrary, when the network bandwidth is poor, the QoE values of each algorithm fluctuate violently between 0.4 and 0.8, especially the LFU and LRU algorithms, which are often lower than 0.5, indicating that there are serious problems in the video processing and transmission process under low-quality bandwidth environment, resulting in poor user experience.

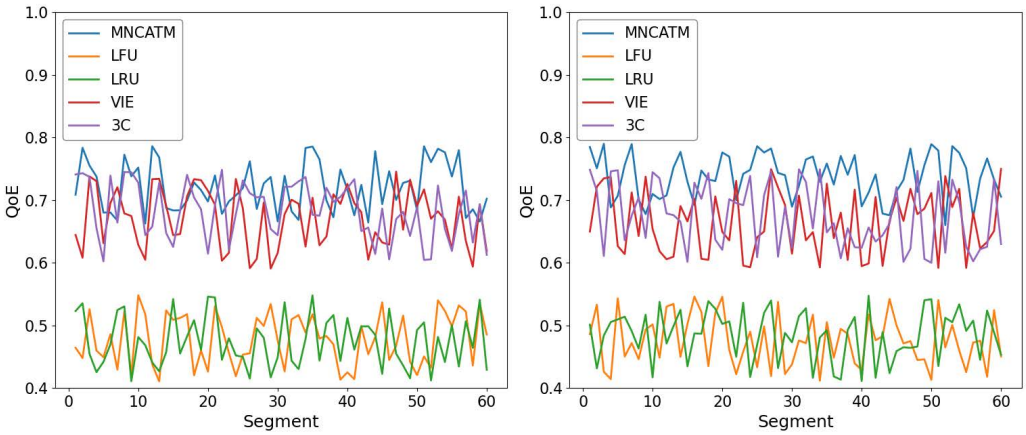
Figure 6 shows the evaluation results of the PSNR performance of the five algorithms when applied to the two videos of a boat and a crosswalk under normal bandwidth. A higher PSNR value indicates better image quality. Across all video clips, MNCATM maintains a PSNR generally within the range of 35–40. While there are some fluctuations, they are relatively minor, allowing MNCATM to deliver consistently high visual quality in most clips. In contrast, the PSNR values for the LFU and LRU algorithms are notably

lower than those of the other three algorithms, typically ranging between 25 and 30, with some instances where the PSNR dips even lower and exhibits considerable fluctuations. The PSNR performance of the 3C and VIE algorithms is closer to that of MNCATM, with both occasionally matching MNCATM's performance, particularly in certain clips. However, these algorithms also experience occasional large fluctuations in PSNR.



(a) Comparison of video Boat (b) Comparison of video Crosswalk

Figure 4. QoE comparison of different methods under good bandwidth.



(a) Comparison of video Boat (b) Comparison of video Crosswalk

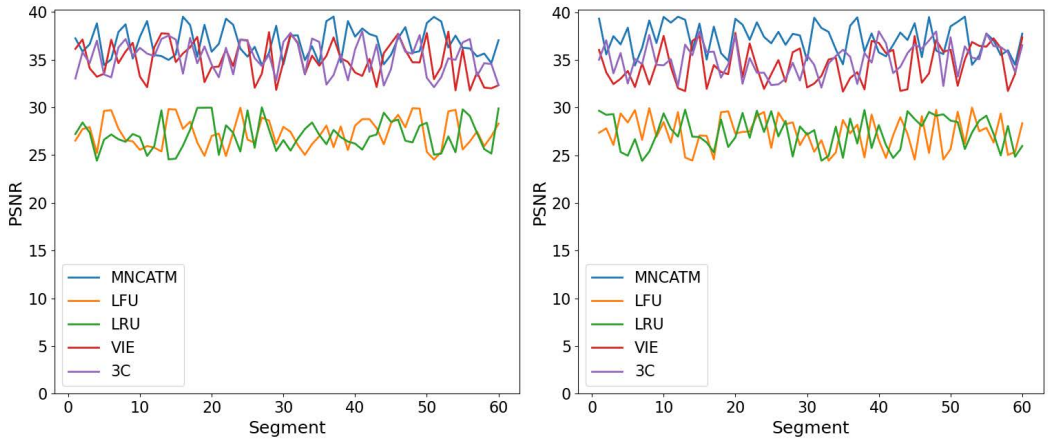
Figure 5. QoE comparison of different methods under poor bandwidth.

It can be seen from Figure 7 that, when bandwidth resources are sufficient, the PSNR values of the five algorithms are almost consistent, floating between 0.94 and 1, so they can all provide users with high-definition video quality. However, in the case of poor bandwidth, the MNCATM algorithm still performed the most prominently in the two videos, always maintaining a high and stable PSNR value, indicating that it has significant advantages in maintaining video quality. Second only to MNCATM, the PSNR values of the LFU and LRU algorithms are much lower than the other three algorithms, as shown in Figure 8.

Figure 9 illustrates the results of SSIM, which evaluates the quality of an image or video when the network bandwidth is normal. An SSIM value closer to 1 indicates a higher similarity between the reconstructed video and the original, signifying superior video quality. As observed in the bar graph, the SSIM value for MNCATM exceeds 0.9 in both videos, which is substantially higher than that of the other algorithms, indicating exceptionally high video quality.

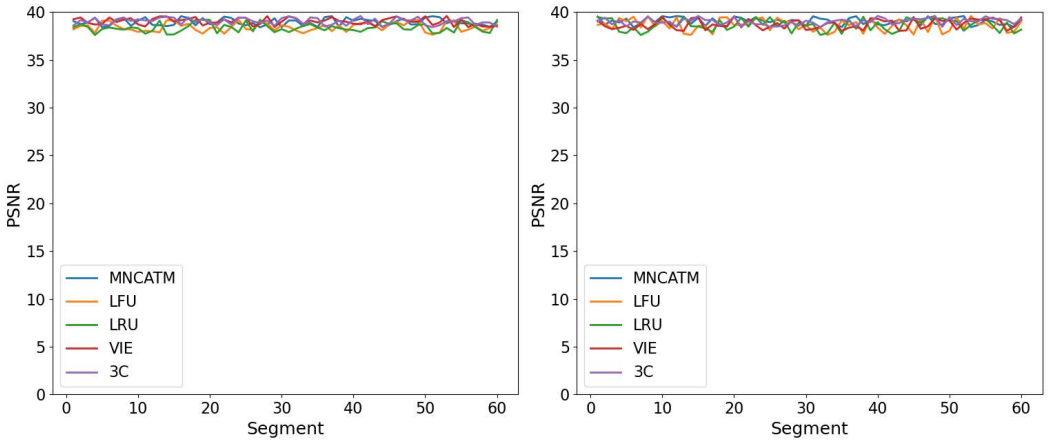
In Figure 10, due to the abundant bandwidth resources, the five algorithms can transmit a video quality close to the original video, so their SSIM values are excellent,

almost above 0.95. However, in Figure 11, the SSIM values of the five algorithms have dropped significantly because the bandwidth resources are very limited, but the MNCATM algorithm still outperforms the other four algorithms in terms of SSIM performance.



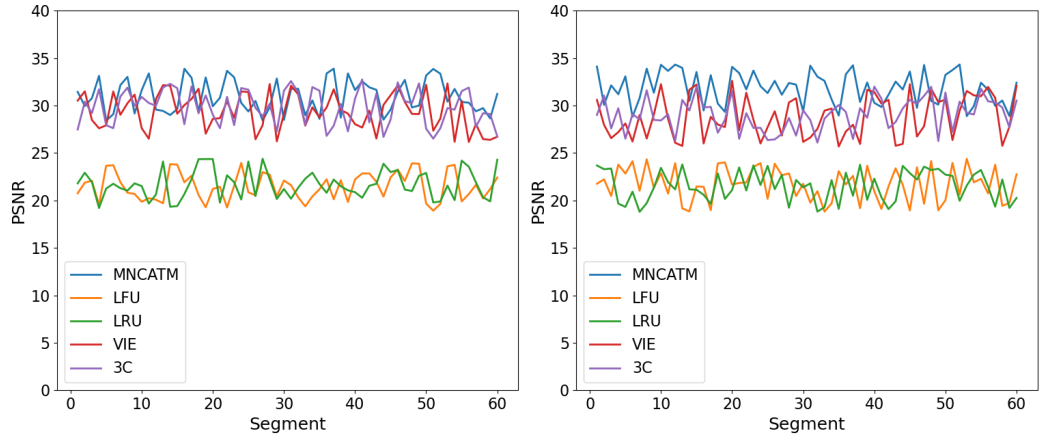
(a) Comparison of video Boat (b) Comparison of video Crosswalk

Figure 6. PSNR comparison of different methods under normal bandwidth.



(a) Comparison of video Boat (b) Comparison of video Crosswalk

Figure 7. PSNR comparison of different methods under good bandwidth.



(a) Comparison of video Boat (b) Comparison of video Crosswalk

Figure 8. PSNR comparison of different methods under poor bandwidth.

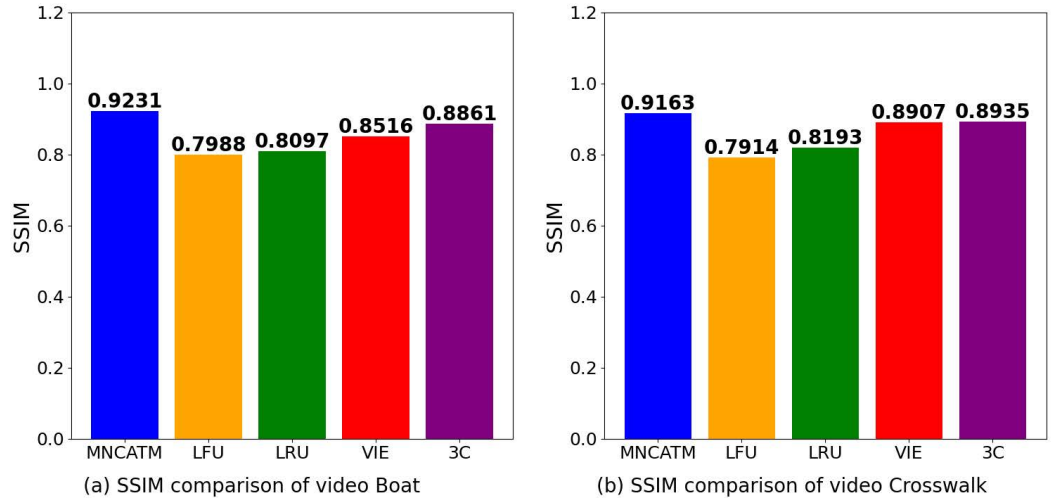


Figure 9. SSIM comparison of different methods under normal bandwidth.

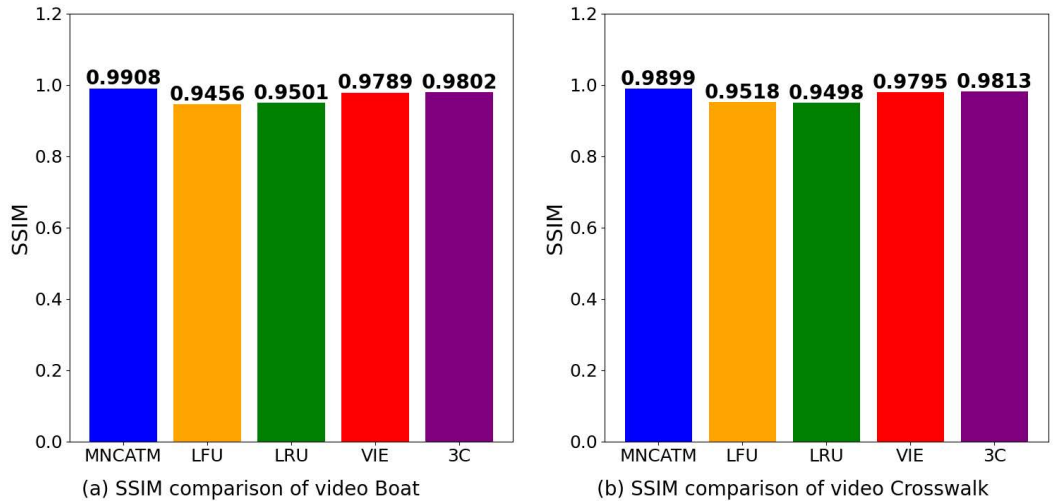


Figure 10. SSIM comparison of different methods under good bandwidth.

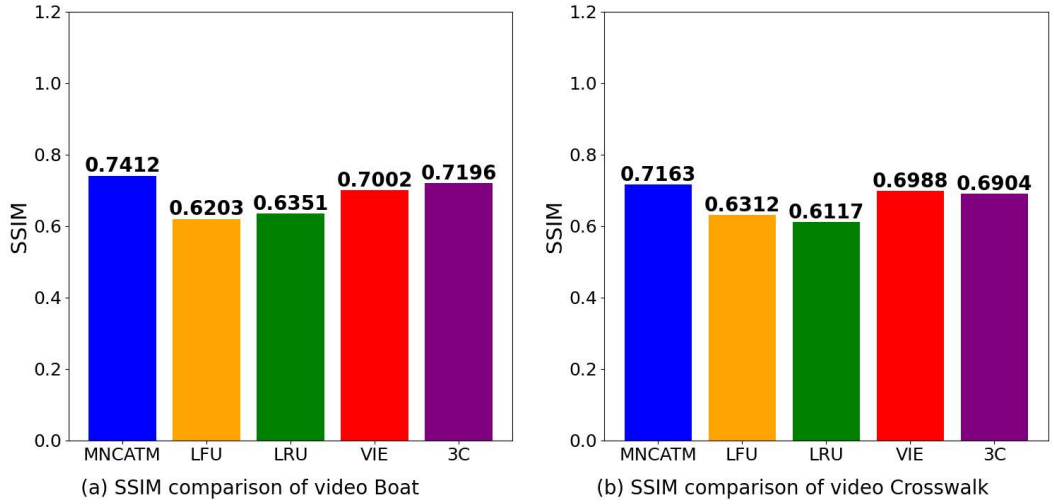


Figure 11. SSIM comparison of different methods under poor bandwidth.

To assess the impact of 360-degree VR video quality on user subjective experiences under different conditions, we designed an MOS evaluation experiment, as depicted in Figures 12–14. Based on the MOS standard specified by the International Telecommunication Union (ITU), we recruited 30 participants, equally divided between men and women,

aged between 15 and 60 years old and free of vision-related diseases [38]. All participants had experience watching VR videos and were arranged to watch the videos in the same environment to minimize the impact of external factors (such as light, equipment quality, etc.). Under three conditions of good, average, and poor network bandwidth, the participants were asked to watch two videos, "Boat" and "Crosswalk", under five different transmission schemes, namely LFU, LRU, VIE, 3C, and MNCATM. We distributed rating sheets to 30 users and asked them to rate the video quality of each solution from 1 to 5 in three scenarios to ensure the validity of the ratings. Finally, for the two 360-degree videos, the arithmetic average of the 30 user ratings for each solution in each case (client_avg) was calculated. In addition, we also participated in the experiment and provided scores to calculate the arithmetic average of the author ratings (author_avg) under the same conditions.

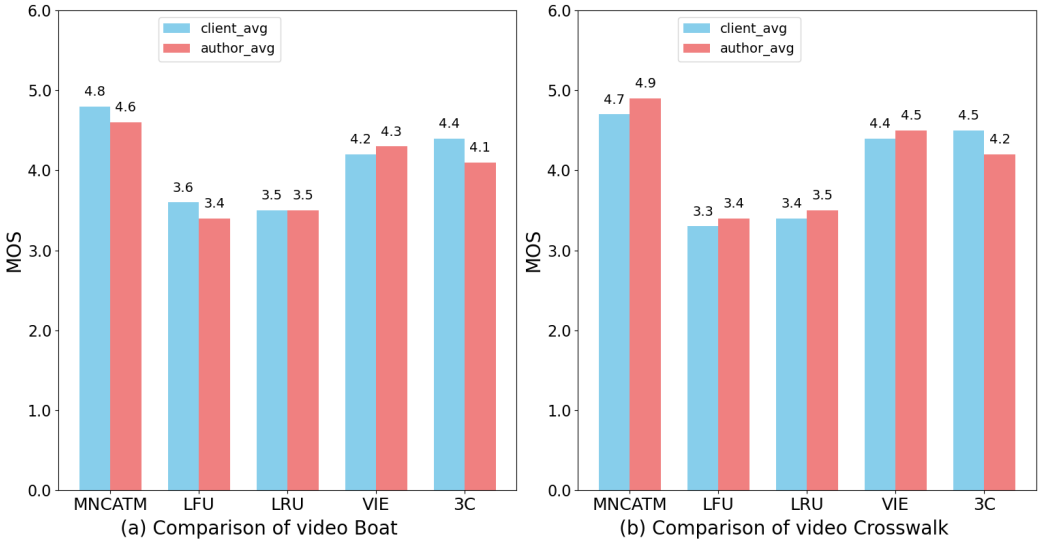


Figure 12. MOS comparison of different methods under normal bandwidth.

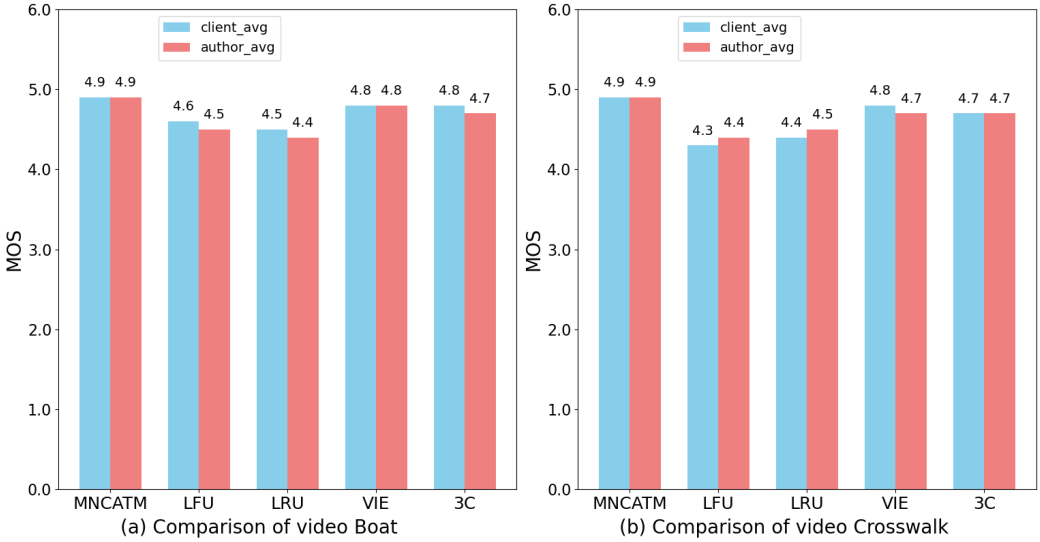


Figure 13. MOS comparison of different methods under good bandwidth.

Under normal network bandwidth conditions, the average user scores closely matched the average author scores, with differences typically ranging from 0.1 to 0.3 points, indicating that both the authors and the 30 participants had similar overall perceptions of video quality. Among the schemes, MNCATM performed the best, achieving the highest scores in both videos. Notably, the average author score for the Crosswalk video reached 4.9, nearly a perfect score. This suggests that the MNCATM scheme excels in handling

complex dynamic scenes and delivers a high-quality visual experience. The VIE scheme also demonstrated strong performance, with scores ranging from 4.2 to 4.5 across both videos, second only to MNCATM. The 3C scheme showed a relatively stable performance, with consistently high average user scores, suggesting that it provides a good viewing experience, though there may be room for improvement in handling finer details. However, the LFU and LRU schemes significantly underperformed compared to the other solutions, receiving low scores in both the Boat and Crosswalk videos. This indicates that these schemes have clear deficiencies in handling video details and complex scenes, and they fail to provide a satisfactory viewing experience.

When bandwidth resources are sufficient, the average scores of users and authors both increase. The scores of MNCATM, VIE, and 3C algorithms are close to full marks, indicating that users are very satisfied with the quality of video transmission. Although the scores of LFU and LRU methods are lower, they also reach more than 4 points. On the contrary, when bandwidth resources are limited, the scores of five algorithms all drop significantly, but MNCATM can maintain more than 4 points, showing its superiority.

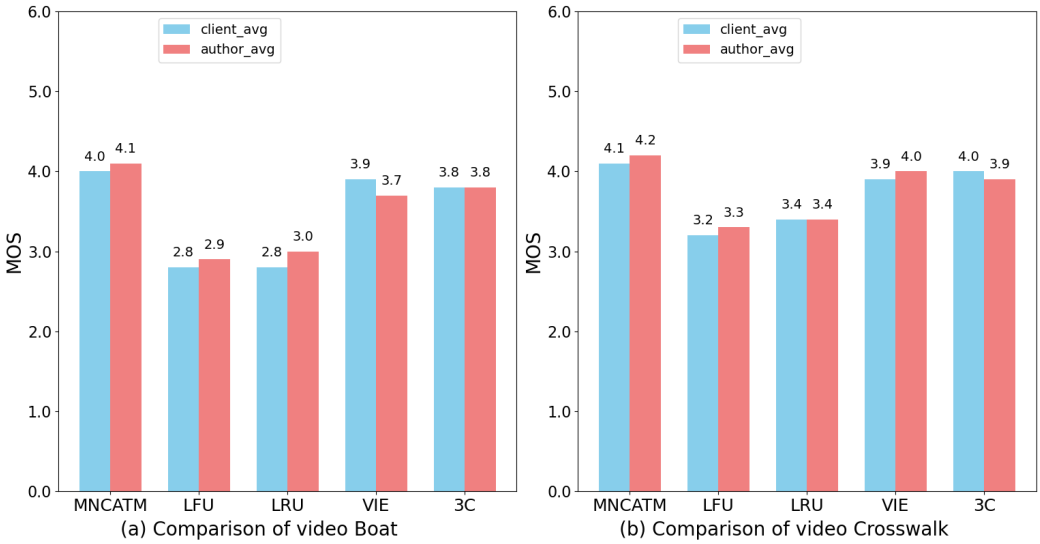


Figure 14. MOS comparison of different methods under poor bandwidth.

6.3. Statistical Analysis and Results

In order to evaluate the differences in performance metrics such as QoE, PSNR, and SSIM across different transmission methods (MNCATM, LFU, LRU, VIE, 3C), we employed one-way analysis of variance (ANOVA). ANOVA is used to test for significant differences in the means across multiple groups, with the formula given by:

$$\mathrm{F} = \frac{MS_{\text{between}}}{MS_{\text{within}}} \tag{12}$$

where MS_{between} represents the mean square variance between groups, indicating the variation between different methods, and MS_{within} represents the mean square variance within groups, indicating the variation within the same method. By comparing the variance between and within groups, ANOVA helps determine whether the differences in the performance metrics across different transmission methods are statistically significant.

In this experiment, the ANOVA results showed statistically significant differences across different transmission methods for QoE ($F(4, 295) = 23.45$, $p < 0.001$). Similarly, for PSNR and SSIM, the F-values were 19.87 and 17.34, respectively, with p-values less than 0.01, indicating that the differences in video quality and structural similarity between MNCATM and the other methods are also significant.

To further investigate these differences, we conducted **Tukey HSD post hoc tests**, which allow pairwise comparisons between groups. The formula for Tukey HSD is:

$$q = \frac{\bar{X}_i - \bar{X}_j}{SE} \tag{13}$$

where $\bar{X}_i$ and $\bar{X}_j$ are the mean values of groups i and j, respectively, and SE represents the standard error. This test enables us to determine which specific methods differ significantly from each other.

The post hoc test results showed that MNCATM had the most significant differences compared to LFU and LRU across all performance metrics. Specifically, MNCATM had a mean QoE of 0.92 (SD = 0.04), which was significantly higher than that of LFU (0.78, SD = 0.06) and LRU (0.79, SD = 0.07), with p-values less than 0.01. Similarly, in terms of PSNR, MNCATM had an average value of 38.5, which is significantly higher than LFU's 28.7 and LRU's 29.2. The SSIM analysis also indicated that MNCATM had an average value of 0.91, while LFU and LRU had SSIM values of 0.81 and 0.82, respectively.

The ANOVA and Tukey HSD post hoc test results clearly demonstrate that MNCATM outperforms traditional methods in enhancing user experience and video quality. The minor fluctuations in PSNR and SSIM observed with MNCATM have a minimal impact on user experience, as confirmed by the high QoE scores. Additionally, MNCATM's multi-layer non-uniform coding strategy effectively maintains high-quality transmission even under constrained bandwidth conditions, allowing it to deliver a stable performance in complex scenarios, further improving the overall viewing experience for users.

7. Conclusions

This paper presented MNCATM, an innovative 360-degree video transmission and caching algorithm leveraging multi-layer non-uniform coding to optimize the delivery of immersive VR experiences. By dynamically adjusting to users' FoV and efficiently utilizing the available network bandwidth, MNCATM significantly enhances user QoE while minimizing latency. Comparative simulation results validate MNCATM's superior performance over traditional algorithms in various metrics, including bandwidth utilization, video quality, and user satisfaction. The proposed method addresses key challenges in 360° video streaming, paving the way for more efficient and scalable solutions in the future. Further research will explore advanced bitrate adaptation and multi-path transmission strategies to enhance real-time 360-degree video streaming systems.

Author Contributions: Conceptualization, J.G. and X.L.; methodology, X.L.; software, C.L. and Y.Z.; validation, K.T. and Y.Z.; formal analysis, Y.L.; investigation, J.G.; resources, X.L.; data curation, K.T.; writing—original draft preparation, X.L. and Y.L.; writing—review and editing, J.N.; visualization, J.N.; supervision, X.Z.; project administration, X.L.; funding acquisition, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was substantially supported by Tianjin Normal University Cybersecurity and Informatization Development Project No.52WT2327 and No.52WT2328, the National Natural Science Foundation of China under Grant No. 62002263, and the Tianjin Municipal Education Commission Research Program Project under 2022KJ012.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available as they are related to our subsequent research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. AR/VR Headset Market Forecast to Decline 8.3% in 2023 But Remains on Track to Rebound in 2024, According to IDC. Available online: https://www.idc.com/getdoc.jsp?containerId=prUS51574023 (accessed on 5 March 2024).
2. Yang, T.; Tan, Z.; Xu, Y.; Cai, S. Collaborative edge caching and transcoding for 360° video streaming based on deep reinforcement learning. *IEEE Internet Things J.* **2022**, *9*, 25551–25564. [CrossRef]

3. Mahmoud, M.; Rizou, S.; Panayides, A.S.; Kantartzis, N.V.; Karagiannidis, G.K.; Lazaridis, P.I.; Zaharis, Z.D. A survey on optimizing mobile delivery of 360° videos: Edge caching and multicasting. *IEEE Access* **2023**, *11*, 68925–68942. [CrossRef]
4. Maniotis, P.; Thomos, N. Viewport-Aware Deep Reinforcement Learning Approach for 360° Video Caching. *IEEE Trans. Multimed.* **2021**, *24*, 386–399. [CrossRef]
5. Zheng, C.; Liu, S.; Huang, Y.; Yang, L. Hybrid policy learning for energy-latency tradeoff in MEC-assisted VR video service. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9006–9021. [CrossRef]
6. Han, S.; Su, H.; Yang, C.; Molisch, A.F. Proactive edge caching for video on demand with quality adaptation. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 218–234. [CrossRef]
7. Guo, Y.; Yu, F.R.; An, J.; Yang, K.; Yu, C.; Leung, V.C.M. Adaptive bitrate streaming in wireless networks with transcoding at network edge using deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3879–3892. [CrossRef]
8. Sun, L.; Duanmu, F.; Liu, Y.; Wang, Y.; Ye, Y.; Shi, H. A two-tier system for on-demand streaming of 360 degree video over dynamic networks. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2019**, *9*, 43–57. [CrossRef]
9. Hu, M.; Chen, J.; Wu, D.; Zhou, Y.; Wang, Y.; Dai, H. TVG-streaming: Learning user behaviors for QoE-optimized 360-degree video streaming. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 4107–4120. [CrossRef]
10. Ozcinar, C.; Abreu, A.D.; Smolic, A. Viewport-aware adaptive 360 video streaming using tiles for virtual reality. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
11. Son, J.; Ryu, E.-S. Tile-based 360-degree video streaming for mobile virtual reality in cyber physical system. *Comput. Electr. Eng.* **2018**, *72*, 361–368. [CrossRef]
12. Nguyen, D.V.; Tran, H.T.T.; Pham, A.T.; Thang, T.C. An optimal tile-based approach for viewport-adaptive 360-degree video streaming. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2019**, *9*, 29–42. [CrossRef]
13. Jeppsson, M. Efficient live and on-demand tiled HEVC 360 VR video streaming. *Int. J. Semant. Comput.* **2019**, *13*, 367–391. [CrossRef]
14. Carreira, J.; de Faria, S.M.M.; Tavora, L.M.N.; Navarro, A.; Assuncao, P.A. 360° video coding using adaptive tile partitioning. In Proceedings of the 2021 Telecoms Conference (ConfTELE), Leiria, Portugal, 11–12 February 2021.
15. Yaqoob, A.; Muntean, C.; Muntean, G.-M. Flexible Tiles in Adaptive Viewing Window: Enabling Bandwidth-Efficient and Quality-Oriented 360° VR Video Streaming. In Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Bilbao, Spain, 15–17 June 2022.
16. Nguyen, D.V.; Tran, H.T.T.; Thang, T.C. An evaluation of tile selection methods for viewport-adaptive streaming of 360-degree video. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2020**, *16*, 1–24. [CrossRef]
17. Ye, Z.; Li, Q.; Ma, X.; Zhao, D.; Jiang, Y.; Ma, L. VRCT: A viewport reconstruction-based 360 video caching solution for tile-adaptive streaming. *IEEE Trans. Broadcast.* **2023**, *69*, 691–703. [CrossRef]
18. EE and BT Unveil New 5G-Enabled XR Sports Experiences. Available online: https://www.broadcastnow.co.uk/production/ee-and-bt-unveil-new-5g-enabled-xr-sports-experiences/5168508.article (accessed on 15 March 2024).
19. EE and BT Unveil New Sports and Performing Arts Experiences Based on 5G and Extended Reality. Available online: https://newsroom.bt.com/ee-and-bt-unveil-new-sports-and-performing-arts-experiences-based-on-5g-and-extended-reality/ (accessed on 15 March 2024).
20. Shafi, R.; Shuai, W.; Younus, M.U. 360-degree video streaming: A survey of the state of the art. *Symmetry* **2020**, *12*, 1491. [CrossRef]
21. Zhang, X.; Qi, Z.; Min, G.; Miao, W.; Fan, Q.; Ma, Z. Cooperative edge caching based on temporal convolutional networks. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 2093–2105. [CrossRef]
22. Liu, Q.; Chen, H.; Li, Z.; Bai, Y.; Wu, D.; Zhou, Y. Online Caching Algorithm for VR Video Streaming in Mobile Edge Caching System. *Mob. Netw. Appl.* **2024**, 1–13. [CrossRef]
23. Fu, B.; Tang, T.; Wu, D.; Wang, R. Interest-Aware Joint Caching, Computing, and Communication Optimization for Mobile VR Delivery in MEC Networks. *arXiv* **2024**, arXiv:2403.05851.
24. Xia, J.; Chen, L.; Tang, Y.; Wang, W. Multi-MEC cooperation based VR video transmission and cache using K-shortest paths optimization. In Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services, Pittsburgh, PA, USA, 14–17 November 2022; Springer Nature: Cham, Switzerland, 2022; pp. 334–355.
25. Shafi, R.; Shuai, W.; Younus, M.U. MTC360: A multi-tiles configuration for viewport-dependent 360-degree video streaming. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020.
26. Chen, H.Y.; Lin, C.S. Tiled streaming for layered 3D virtual reality videos with viewport prediction. *Multimed. Tools Appl.* **2022**, *81*, 13867–13888. [CrossRef]
27. Cheng, Q.; Shan, H.; Zhuang, W.; Yu, L.; Zhang, Z.; Quek, T.Q.S. Design and Analysis of MEC-and Proactive Caching-Based 360° Mobile VR Video Streaming. *IEEE Trans. Multimed.* **2021**, *24*, 1529–1544. [CrossRef]
28. Yang, J.; Guo, Z.; Luo, J.; Shen, Y.; Yu, K. Cloud-edge-end collaborative caching based on graph learning for cyber-physical virtual reality. *IEEE Syst. J.* **2023**, *17*, 5097–5108. [CrossRef]
29. Long, K.; Cui, Y.; Ye, C.; Liu, Z. Optimal wireless streaming of multi-quality 360 VR video by exploiting natural, relative smoothness-enabled, and transcoding-enabled multicast opportunities. *IEEE Trans. Multimed.* **2020**, *23*, 3670–3683. [CrossRef]

30. Okamoto, T.; Ishioka, T.; Shiina, R.; Fukui, T.; Ono, H.; Fujiwara, T. Edge-assisted multi-user 360-degree video delivery. In Proceedings of the 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2023; pp. 194–199.
31. Zhang, G.; Wu, C.; Gao, Q. Exploiting layer and spatial correlations to enhance SVC and tile based 360-degree video streaming. *Comput. Netw.* **2021**, *191*, 107985. [CrossRef]
32. Nasrabadi, A.T.; Mahzari, A.; Beshay, J.D.; Prakash, R. Adaptive 360-degree video streaming using layered video coding. In Proceedings of the 2017 IEEE Virtual Reality (VR), Los Angeles, CA, USA, 18–22 March 2017; pp. 347–348.
33. Zhang, X.; Hu, X.; Zhong, L.; Shirmohammadi, S.; Zhang, L. Cooperative tile-based 360 panoramic streaming in heterogeneous networks using scalable video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *30*, 217–231. [CrossRef]
34. Nguyen, D.; Hung, N.V.; Phong, N.T.; Huong, T.T.; Thang, T.C. Scalable multicast for live 360-degree video streaming over mobile networks. *IEEE Access* **2022**, *10*, 38802–38812. [CrossRef]
35. Zhong, L.; Chen, X.; Xu, C.; Ma, Y.; Wang, M.; Zhao, Y. A multi-user cost-efficient crowd-assisted VR content delivery solution in 5G-and-beyond heterogeneous networks. *IEEE Trans. Mob. Comput.* **2022**, *22*, 4405–4421. [CrossRef]
36. Chakareski, J.; Khan, M. Live 360° Video Streaming to Heterogeneous Clients in 5G Networks. *IEEE Trans. Multimed.* **2024, 26**, 1–14. [CrossRef]
37. Chiariotti, F. A survey on 360-degree video: Coding, quality of experience and streaming. *Comput. Commun.* **2021**, *177*, 133–155. [CrossRef]
38. Mean Opinion Score (MOS) Terminology in the ITU-T P.800 Standard Released by ITU in 1996. Available online: https://www.itu.int/rec/T-REC-P.800.1-201607-I/en (accessed on 22 March 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.