VoLUT: Efficient Volumetric streaming enhanced by LUT-based super-resolution

Chendong Wang[*1] **Anlan Zhang**[2] **Yifan Yang**[3] **Lili Qiu**[3] **Yuqing Yang**[3] **XINYANG JIANG**[3] **Feng Qian**[2] **Suman Banerjee**[1]

Abstract

3D volumetric video provides immersive experience and is gaining traction in digital media. Despite its rising popularity, the streaming of volumetric video content poses significant challenges due to the high data bandwidth requirement. A natural approach to mitigate the bandwidth issue is to reduce the volumetric video's data rate by downsampling the content prior to transmission. The video can then be upsampled at the receiver's end using a super-resolution (SR) algorithm to reconstruct the high-resolution details. While super-resolution techniques have been extensively explored and advanced for 2D video content, there is limited work on SR algorithms tailored for volumetric videos.

To address this gap and the growing need for efficient volumetric video streaming, we have developed VoLUT with a new SR algorithm specifically designed for volumetric content. Our algorithm uniquely harnesses the power of lookup tables (LUTs) to facilitate the efficient and accurate upscaling of low-resolution volumetric data. The use of LUTs enables our algorithm to quickly reference precomputed high-resolution values, thereby significantly reducing the computational complexity and time required for upscaling. We further apply adaptive video bit rate algorithm (ABR) to dynamically determine the downsampling rate according to the network condition and stream the selected video rate to the receiver. Compared to related work, VoLUT is the first to enable high-quality 3D SR on commodity mobile devices at line-rate. Our evaluation shows VoLUT can reduce bandwidth usage by 70% , boost QoE by 36.7% for volumetric video streaming and achieve 8.4× 3D SR speed-up with no quality compromise.

1 Introduction

Empowered by advances in 3D capture and rendering technologies, volumetric video applications are revolutionizing how we experience digital content across entertainment (vol, 2024), education (Emad et al., 2022), virtual reality (efe, 2024), *etc.* These applications enable users to freely navigate and interact with dynamic 3D scenes with six degrees of freedom (6DoF), providing an unprecedented level of immersion that traditional 2D videos cannot match. Among various 3D representations, point cloud has emerged as a preferred format for volumetric content due to its rendering efficiency and capturing availability (Yang et al.; Zhang et al.). The recent breakthrough in 3D Gaussian splatting (Luiten et al.), which can be viewed as a specialized point cloud format, further demonstrates the potential of point-based representations for high-quality volumetric content delivery.

A key challenge in deploying volumetric video applications is the substantial bandwidth requirement for streaming high-quality point cloud content at line-rate (*i.e.,* 30 FPS), which can demand as much as 720Mbps for high-quality content with 200K points per frame (Zhang et al.). As a result, viewers watching volumetric content over the Internet always suffer from drastically compromised quality-of-experience (QoE). While various approaches have been proposed to address this challenge, each has significant limitations. Viewport-adaptive streaming techniques reduce bandwidth usage by streaming only the content within the user's future viewport (Han et al., 2020; Lee et al., 2020; Liu et al., 2024), but suffer from quality degradation under rapid viewer movement or when rendering wide-angle scenes. Remote-rendering-based solutions employ a cloud/edge server to transcode 3D scenes to regular 2D frames that consume much less bandwidth (Gül et al., 2020b; Liu et al., 2022b). However, these approaches introduce non-negligible latency (50-200ms) and require complex infrastructure support to scale up to multiple concurrent users.

Recently, super-resolution (SR) based approaches (Zhang

[1]University of Wisconsin–Madison, Madison, WI, USA [2]University of Southern California, Los Angeles, CA, USA [3]Microsoft Research Asia, Beijing, China. Correspondence to: Chendong Wang <cwang747@wisc.edu>. [*]This work is done while Chendong Wang's internship at Microsoft Research.

Proceedings of the 8th MLSys Conference, Santa Clara, CA, USA, 2025. Copyright 2025 by the author(s).

et al.) have demonstrated their potentials in reducing the bandwidth consumption while boosting the users QoE for volumetric video streaming. These approaches offload the network-side burden to the client-side computation overhead, by deploying a deep 3D SR model on the client devices to enhance the visual quality of low-resolution point cloud content. Nevertheless, off-the-shelf deep 3D SR models (Li et al., 2019; Qian et al.; Yu et al.; Wang et al., 2021; Long et al., 2022) are still too heavyweight to perform real-time (*i.e.,* 30 FPS) SR on resource-constrained mobile devices such as Meta Quest 3 (met, 2024), even after extensive optimizations for inference acceleration (Zhang et al.). In addition, existing SR-based solutions struggle with limited upsampling ratios and growing training cost, as they require training an SR model for each upsampling ratio per video.

This paper presents VoLUT, a novel SR-enhanced volumetric video streaming system for commodity mobile devices by addressing all the above limitations. We make two vital insights for developing VoLUT. First, the computational complexity of 3D SR can be drastically reduced by decomposing this task into two stages (Liu et al., 2020): (1) a traditional interpolation phase that performs basic SR on the input point cloud, followed by (2) a refinement deep neural network (DNN) model that fine-tunes the SR output in (1). This brand new 3D SR pipeline also provides several side benefits, such as good generalization across different content types, and flexible upsampling ratios with only a single refinement DNN model (Liu et al., 2020) . Second, the on-device memory is not well-utilized in previous SR-based volumetric video streaming systems. Therefore, we have rich opportunities to trade the on-device memory for efficient model inference thus further SR speedup, by transferring a DNN model to a lookup table (LUT) offline.

We face several key challenges when building VoLUT. First, vanilla kNN-based interpolation introduces noticeable visual distortions (see Figure 4) and high computation latency, while diminishing these distortions and accelerating the interpolation simultaneously is a non-trivial task. Second, while LUT has been explored to speed up 2D image SR (Jo & Joo Kim, 2021; Liu et al., 2022a; Yang et al., 2022), applying it to 3D SR poses a unique difficulty given the continuous nature of point cloud: converting the refinement DNN to a LUT that preserves the SR quality and fits the limited on-device memory requires sophisticated designs to balance the trade-off. Furthermore, the arbitrary SR ratio ensured by our two-stage SR brings a necessity for an adaptive bitrate (ABR) streaming algorithm that can rapidly determine the highly fine-grained {to-be-fetched point density, SR ratio} for the volumetric frames, to fully utilize the network/compute resource. We address these challenges through three core technical innovations:

• **Enhanced dilated interpolation:** We develop a novel interpolation technique that introduces controlled dilation, octree-based parallelism and neighbor relationship reuse in neighbor point selection. Our approach achieves 4× faster processing compared to traditional k-NN interpolation with no quality compromise.

• **Position-aware LUT refinement:** We enable efficient 3D LUT application through a novel position encoding scheme that effectively normalizes and quantizes the continuous 3D space into discrete bins. Our method reduces the refinement computation latency by over 99.9% compared to neural network inference (sub-milliseconds v.s. seconds) while maintaining the quality benefits.

• **Continuous-ratio streaming pipeline:** We design an end-to-end system that integrates our SR pipeline with continuous quality adaptation, providing fine-grained control over the quality-bandwidth tradeoff.

We implement VoLUT and evaluate it on both desktop PCs and mobile devices like Orange Pi (ora, 2024) that has similar computation and memory capability as Meta Quest 3 (met, 2024). Our evaluation shows that VoLUT achieves real-time performance (30+ fps) on mobile devices while reducing bandwidth requirements by up to 70% compared to raw point cloud streaming. Notably, our LUT-based SR achieves 8.4× speedup compared to YuZu (Zhang et al.)'s neural SR approach, the state-of-the-art SR-based volumetric video streaming system, and our system achieves 36.7% better QoE than Yuzu system under both stable and LTE traces. Our key contributions include:

• An dilated interpolation technique with spatial information pruning and reusing that significantly improves both visual quality and processing speed for point cloud super-resolution

• A position encoding mechanism designed for applying LUT to 3D continuous space that enables efficient 3D super-resolution on resource-constrained devices

• An end-to-end system design that orchestrates the SR pipeline with continuous quality adaptation for volumetric video streaming

2 Background and Motivation

2.1 Point Cloud Super-Resolution

Point cloud super-resolution (PCSR) has emerged as a promising technique for enhancing volumetric video quality while reducing bandwidth requirements. Early PCSR methods like PU-Net (Yu et al., 2018), MPU (Yifan et al., 2019), and PUGAN (Li et al., 2019) demonstrated the potential of learning-based approaches but were constrained by fixed upsampling ratios and high computational demands. The recent GradPU (He et al., 2023) overcame the ratio limitation by introducing a two-stage approach (shown in Figure 1): first performing midpoint interpolation in Euclidean space,

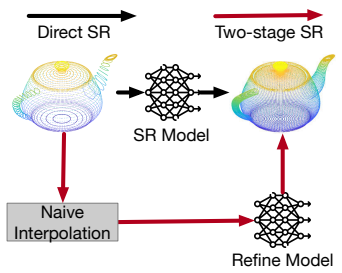

Figure 1. DL-based Point Cloud Super-resolution: Direct vs. Two-stage.

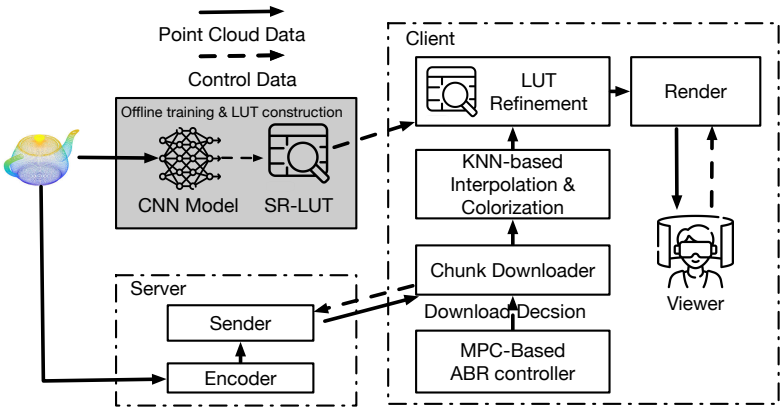

Figure 2. The System Architecture of VoLUT.

then refining point positions through iterative optimization to minimize point-to-point distances with ground truth.

While GradPU's flexible upsampling ratio and improved generalization make it particularly attractive for volumetric video streaming, its computational requirements remain prohibitive for consumer devices. YuZu (Zhang et al.), the first system to apply PCSR for volumetric video streaming, demonstrates this challenge—despite achieving significant quality improvements, it requires high-end GPUs and introduces substantial latency due to neural network inference. This computational barrier has limited the practical deployment of PCSR-based streaming solutions on resource-constrained devices like mobile VR headsets.

2.2 LUT-based Image Super-resolution

Look-up table (LUT) based approaches have recently shown promise in accelerating 2D image super-resolution by replacing expensive neural network computations with efficient table lookups (Tang et al., 2023; Jo & Joo Kim, 2021; Liu et al.; Yang et al., 2022). These methods work by training a deep SR network with a constrained receptive field, then transferring the learned mappings to a lookup table. During inference, the system uses the local input pattern as an index to retrieve pre-computed high-resolution outputs, dramatically reducing computational overhead compared to full neural network inference.

However, extending LUT-based optimization to 3D point clouds introduces fundamental challenges not present in 2D scenarios. Unlike image pixels that exist on a discrete grid, point clouds occupy continuous 3D space, making it impossible to directly index all potential local point configurations. The indexing space grows exponentially with the number of neighboring points considered, and naive quantization schemes can lead to significant quality degradation. Additionally, while 2D images have regular neighborhoods defined by fixed pixel grids, point cloud neighborhoods vary in size and spatial distribution, further complicating the design of an effective lookup mechanism.

These challenges motivate our development of positional encoding (§ 4.2.1) that can effectively bridge the gap between 2D LUT-based approaches and 3D point cloud processing while maintaining the quality benefits of learning-based PCSR methods. By carefully addressing the continuous space quantization and neighborhood encoding problems, we can make PCSR practical for real-time volumetric video streaming on consumer devices.

3 System Overview

VoLUT enables high-quality streaming of volumetric video content by combining adaptive bitrate streaming with super-resolution enhancement. The server segments videos into fixed-length chunks and encodes them at requested point densities.

As shown in Figure 2, the client processes received low-resolution frames through three stages: kNN-based interpolation with dilation to increase point density (§4.1), colorization based on spatial relationships (§4.1), and LUT-based fine-tuning to enhance visual quality (§4.2). To achieve real-time performance, VoLUT transforms neural networks into memory-efficient LUTs and reuses kNN results across pipeline stages, ensuring stable quality and latency across different upscaling ratios.

The system dynamically selects optimal downsampling ratios based on network conditions and buffer status (§5). Through this combination of adaptive downsampling and efficient super-resolution, VoLUT delivers high-quality volumetric video while optimizing bandwidth utilization across varying network conditions.

4 LUT Based Point Cloud Super-resolution

4.1 Enhanced Interpolation with Colorization

Given a downsampled point cloud along with a desired upsampling ratio, we first perform interpolation to increase the point density. The quality of the final super-resoluted point cloud critically depends on the initial interpolation stage. Our qualitative results reveal that poor initial point

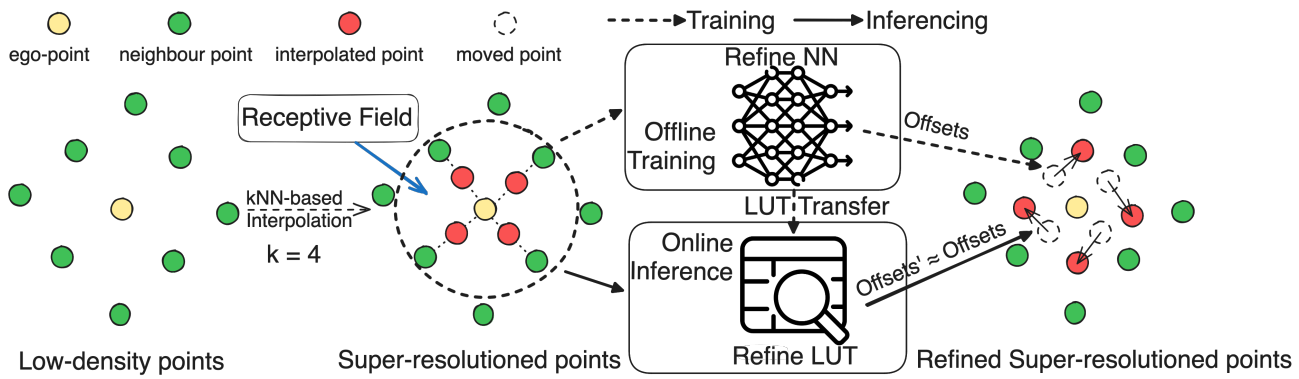

Figure 3. The pipeline of two-stage Super-resolution with LUT refinement

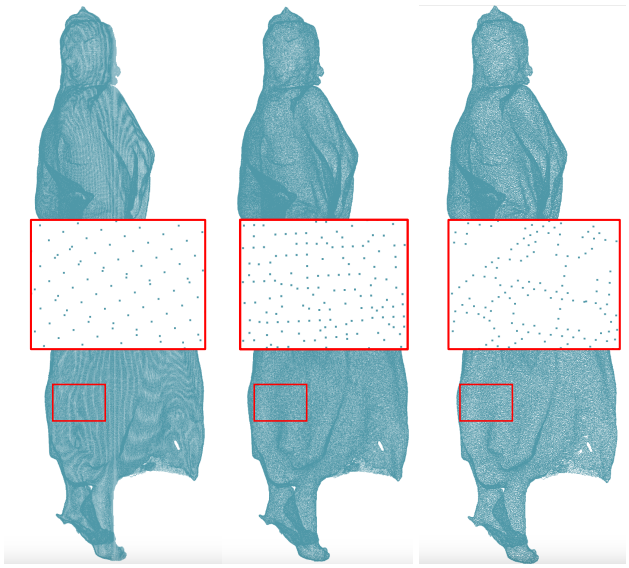

Figure 4. Qualitative upsampling results (from left to right): *Groundtruth*, *Interpolation with dilation*, *Naive knn-based interpolation*. Our method achieves more uniform point distribution while preserving geometric details.

distributions create artifacts (Figure 4) that persist even after neural refinement. Additionally, traditional interpolation methods create a severe performance bottleneck—our measurements on GradPU show that naive kNN-based interpolation consumes over 70% of the frame time, making real-time operation infeasible.

Central to both interpolation and subsequent refinement is the concept of receptive fields (RF)—the local spatial regions considered when processing each point. As illustrated in Figure 5, the receptive field determines which neighboring points influence the position and attributes of newly generated points. In traditional kNN approaches, each new point is generated by considering only its k closest neighbors, which causes two distinct problems. First, this method tends to reinforce existing density patterns because points in dense regions have closer neighbors than those in sparse

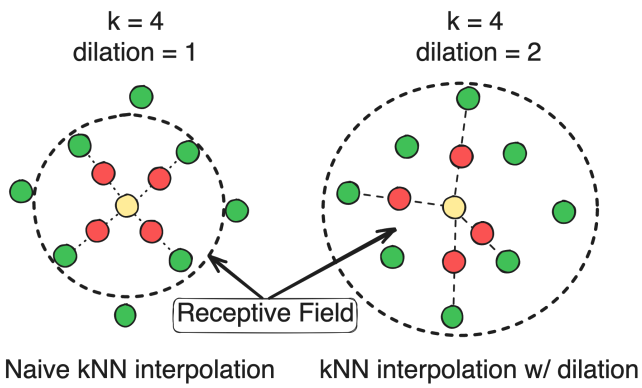

Figure 5. Interpolation with and without dilation. Receptive Field size = $k\times$ dilation. The dilated approach significantly improves point distribution uniformity and surface coverage.

regions, leading to uneven point distributions. Second, the neighbor search operations required for each new point are computationally expensive, especially as the point cloud size grows. VoLUT enables two optimizations in interpolation stage.

Dilated Interpolation for Uniform Upsampling Our key insight is that by carefully expanding the sampling neighborhood through dilation, we can break the artifact introduced by traditional kNN interpolation while still preserving important geometric features. As shown in Figure 5, our dilated approach examines a broader spatial region during interpolation, defined by a receptive field of size $k\times d$, where k is the number of neighbors and d is the dilation factor.

For a point cloud $P = \{p_1, p_2, \ldots, p_n\}$, we define a dilated neighborhood for each point p_i as:

$$N_{dk}(p_i) = \{p_j \in P_n | P_n = \text{Top}_{d\times k}(||p_j - p_i||_2)\} \quad (1)$$

where d is the dilation factor, k is the desired neighbor count, and $||x||_2$ denotes Euclidean distance. The Top function orders points by distance increasingly and keeps the first

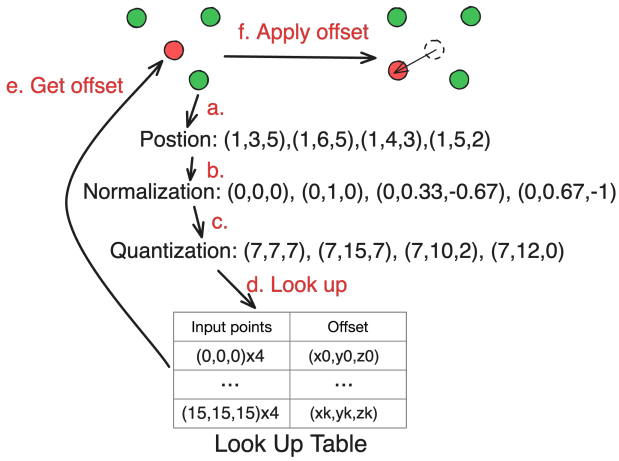

Figure 6. LUT look up example.

$d \times k$ ones, similar as the request of vanilla kNN. From this expanded neighborhood, we randomly select a subset S_i of points for interpolation based on the target upsampling ratio requirement.

An alternative solution may interpolate the point cloud to a higher density and perform Farthest Point Sampling (Li et al., 2022) (FPS) to the target upsampling ratio. FPS iteratively samples the farthest point and updates the distance, which can preserve geometry feature but introduce unacceptable computation latency (≥ 5 minutes to downsample a 200K points to 100K points on a commodity desktop).

Hierarchical kNN Computation with Relationship Reuse To achieve real-time performance on mobile devices, we adopt an efficient two-layer octree (Schnabel & Klein) structure that balances spatial organization against traversal overhead. Our measurements show that naive dilated interpolation takes over 100ms per 100K-points-frame on an Orange Pi, making optimization essential for real-time operation.

The octree divides the point cloud into eight major regions at the first layer, with each region further subdivided into eight sub-regions. While the construction of the octree takes limited effort, its leaf nodes store a subset of the points whose neighbour points are highly likely self-contained. This hierarchical structure enables rapid neighbor search through efficient spatial pruning.

We further accelerate computation through neighbor relationship reuse. For each interpolated point p' generated between points p and q, we observe that:

$$N_k(p') \approx \text{MergeAndPrune}(N_k(p), N_k(q)) \tag{2}$$

where $N_k(p')$ represents the k-nearest neighbors of p'. This approximation eliminates redundant neighbor searches while maintaining accuracy.

We colorize new points based on the nearest original point, reusing spatial relationships from geometric interpolation to avoid redundant computations.

4.2 Interpolation Refinement with LUT

As discussed in § 2.1, a refinement function is required to adjust the interpolated point clouds for better visual quality. We propose an LUT-based refinement approach (shown in Figure 3) that first captures refinement patterns through offline neural network training, then transfers this knowledge into an efficient lookup table for real-time inference.

4.2.1 Position Encoding and LUT Construction

The key challenge in constructing a lookup table for point refinement is converting continuous 3D point positions into discrete indices while preserving geometric relationships. As shown in Figure 6, we address this through a systematic encoding pipeline that transforms raw 3D coordinates into quantized indices for efficient lookup and refinement.

Position Encoding Pipeline Our encoding process consists of three key steps:

• **Position Input(Stage (a)):** The pipeline takes as input a neighborhood of 3D points represented as (x, y, z) coordinates. For a receptive field of size n, we process the target point along with its $n-1$ neighboring points' positions in 3D space.

• **Normalization(Stage (b)):** To ensure consistent lookup behavior, we normalize the coordinates relative to the center point:

$$\mathbf{n}_i = \frac{\mathbf{r}_i - \mathbf{r}_c}{R} \tag{3}$$

where $\mathbf{r}_c$ is the center point coordinates and R is the neighborhood radius (maximum distance from any point to the center). This transforms ensuring all points lie within the unit cube $[-1, 1]^3$.

• **Quantization(Stage (c)):** The normalized coordinates are then discretized into fixed-size b bins to create lookup indices:

$$\mathbf{q}_i = \lfloor (\frac{\mathbf{n}_i + 1}{2}) \times (b-1) \rfloor \tag{4}$$

This step converts continuous normalized values into discrete integer indices suitable for table lookup, effectively creating a finite set of possible neighborhood configurations.

LUT Construction and Usage The lookup table stores precomputed refinement offsets for all possible quantized neighborhood configurations. For a receptive field of size n points with 3 coordinates in 3D space and b quantization bins per dimension, the total number of possible combinations for input indices is:

$$N_{entries} = b^n \times 3 \tag{5}$$

As illustrated in step (d) of Figure 6, each LUT entry maps a sequence of quantized coordinates to a refinement offset

RF Size (n)	Bins (b)	Entries ($b^n \times 3$)	Size (2B/offset)
3	128	$128^3 \times 3$	12 MB
3	64	$64^3 \times 3$	1.5 MB
4	128	$128^4 \times 3$	1.61 GB
4	64	$64^4 \times 3$	100 MB
5	128	$128^5 \times 3$	201 GB
5	64	$64^5 \times 3$	6.25 GB

Table 1. Memory analysis for different LUT configurations with float16 (2B) storage

in three dimensions:

$$\text{LUT}[\text{quantize}(\mathbf{q}_1, \ldots, \mathbf{q}_n)] = \text{NN}(\mathbf{q}_1, \ldots, \mathbf{q}_n) \quad (6)$$

The memory requirement for storing these entries with 2-byte floating-point values (float16) for three coordinate offsets is:

$$M = N_{entries} \times 2 \text{ bytes} \quad (7)$$

Table 1 analyzes the memory requirements for different configurations. The selection of bin size b presents a crucial trade-off between memory efficiency and refinement precision. Our implementation uses $b = 128$ (7-bit quantization) with a receptive field size $n = 4$, resulting in a 1.6 GB lookup table that stores 3D coordinate offsets in $float16$ format. This configuration achieves a good balance between memory efficiency and refinement precision, making the approach practical for real-world deployment.

During runtime operation (Stage (a-f) in Figure 6), we follow this encoding pipeline: given an interpolated point along with its neighbors, we normalize and quantize the coordinates to obtain lookup indices, retrieve the corresponding offset from the table, and apply it to refine the center point's position. Notably, the interpolated point will be placed at first in the index.

4.2.2 *Neural Network for LUT Construction*

The refinement network follows the design from GradPU (He et al., 2023). Given a interpolated point as the central point $\mathbf{p}_c$ and its $n-1$ nearest neighbors $\{\mathbf{p}_i\}_{i=1}^{n-1}$, our network *NN* computes a refinement offset through:

$$\boldsymbol{\delta} = NN(\mathbf{p}_c, \{\mathbf{p}_i\}_{i=1}^{n-1}) \quad (8)$$

The receptive field size n is specifically chosen to balance LUT memory requirements and refinement quality. The offset $\boldsymbol{\delta}$ represents the mean displacement between the interpolated points and their groundtruth counterparts:

$$\boldsymbol{\delta} = \frac{1}{|P|} \sum p_c \in P ||\mathbf{p}_{gt} - \mathbf{p}_c||_2 \quad (9)$$

Training *NN* is equivalent to minimize the target loss function $\boldsymbol{\delta}$.

To assist robust LUT construction, we incorporate two key design elements into the network training. First, Gaussian noise injection ($\sigma = 0.02$) to interpolated points during training improves the network's resilience to quantization artifacts that may arise during the discretization process. Second, we constrain the network's prediction space through normalized coordinate inputs, ensuring the learned function maps well to the LUT's discrete indexing scheme. The complete network training process is detailed in Section 7.1.

5 Continuous Adaptive Bitrate Streaming

Existing volumetric video streaming systems (Han et al., 2020; Zhang et al.; Lee et al., 2020) are constrained by discrete quality levels, typically offering fixed point densities (e.g., 100K, 200K points per frame). We introduce a continuous adaptive bitrate (ABR) mechanism that dynamically optimizes streaming quality through fine-grained point density adjustments. This approach is made possible by our upsampling algorithm's consistent latency across varying upsampling ratios (detailed in § 7.3). The ability to support arbitrary downsampling ratios through our super-resolution pipeline enables more precise adaptation to network conditions, allowing for smoother quality transitions and better bandwidth utilization compared to traditional discrete-level approaches.

5.1 MPC-based Quality Optimization

We formulate quality adaptation using Model Predictive Control (MPC) (Yin et al.), which optimizes streaming quality over a finite horizon of k future frames. We borrow the QoE formulation from Yuzu (Zhang et al.) since it provides an SR-targeting definition validated by real user study. The optimization objective balances three key components: visual quality, quality variation, and stall:

$$\max_{r_t, \ldots, r_{t+k}} \sum_{i=t}^{t+k} (\alpha Q(r_i) - \beta V(r_i, r_{i-1}) - \gamma S(r_i)) \quad (10)$$

The quality term $Q(r)$ denotes the post-SR point density viewed by the user. The variation penalty $V(r_i, r_{i-1})$ prevents rapid quality fluctuations by penalizing changes between consecutive frames, with higher weights for quality drops that are more noticeable to viewers. The stall term $S(r_i)$ ensures smooth playback by maintaining sufficient buffer levels above a minimum threshold.

The MPC solver takes network throughput estimates (computed via harmonic mean over sliding windows) and current buffer levels as input, outputting the optimal {to-be-fetched point density, SR ratio} pair by solving a simple constrained optimization problem.

5.2 Random Downsampling

Given a target ratio r from the MPC solver, we employ random point selection for downsampling with a simple selection probability $P_{select}(p_i) = r$ for each point $p_i \in \mathcal{P}$. Similarly, we choose random sampling approach over FPS stated in § 4.1 to avoid the high computation cost. Combined with our robust upsampling pipeline, simple random downsampling provides sufficient quality while meeting the strict latency requirements of VoD streaming.

6 Implementation

We integrate all the components described in §4 into VoLUT. Our implementation consists of 8.1K lines of code (LoC) in total, with 2.8K LoC for the c++ version client and 3.4K LoC for the cuda version client.

For offline training of the point cloud super-resolution models, we use PyTorch 3.7.11 (Paszke et al., 2017) as the deep learning framework. We modify the source code of GradPU (He et al., 2023), to incorporate our proposed interpolation with dilation and multi-LUT fusion techniques.

Our Look Up Table is generated using c++ code and stored as an npy file which is language- and platform- neutral, faciliating for future use. The c++ verison client pipelined is optimized for performance by leveraging multi-threading and system pipelining. The CUDA client features parallel kNN search, interpolation, and colorization kernels based on cuKDTree (Wald, 2023), along with efficient LUT lookup. The server is also implemented in C++ for efficient processing and serving of volumetric video content. We develop a custom DASH-like protocol over TCP for client-server communication.

7 Evaluation

7.1 Evaluation Setup

Volumetric Videos. We use four point-cloud-based volumetric videos in our evaluations:

- **The Long Dress (Dress) and Loot Videos:** Each has 300 frames lasting 10 seconds and containing approximately 100K points. We loop these videos ten times in our evaluations due to their short duration.

- **The Haggle Video:** Comprises 7,800 frames (4.3 minutes) each containing approximately 100K points.

- **The Lab Video:** Features 3,622 frames (2 minutes) each with approximately 100K points.

Model Training and LUT Generation. We use GradPU (He et al., 2023) as our reference model, training it exclusively on the Long Dress video. The training process involves downsampling the original frames to different densities and using pairs of low/high-resolution point clouds as training data. The trained model is then transformed into a single LUT table with $RF = 4$ and $bin = 128$ (approximately 1.5GB) following the process described in § 4.2. We apply this LUT for super-resolution across all test videos to evaluate its generalization capability.

Evaluation Metrics. We assess VoLUT's performance using both geometric and perceptual metrics: Point-to-point (P2P) Chamfer Distance (CD) (Wu et al.; Li et al., 2019) measure geometric accuracy between upsampled and ground truth point clouds. Peak Signal-to-Noise Ratio (PSNR) evaluates the visual quality of upsampled points. These metrics are computed per-frame and averaged over all frames. Runtime performance is measured through CPU/GPU memory utilization, frame processing latency and frame per second (FPS). For streaming evaluation, we assess Quality of Experience (QoE) discussed in § 5 and data usage during transmission. § 7.2.1 and § 7.2.2 present quality results, § 7.3 examines computational efficiency, while § 7.4 § 7.5 analyzes end-to-end streaming performance.

Network traces We consider the following network conditions that are representative of today's wired and wireless networks. (1) Wired network with stable bandwidth (*e.g.,*, 50, 75, and 100 Mbps) and a round-trip time (RTT) of approximately 10ms. (2) Fluctuating bandwidth captured from real-world LTE networks, with average bandwidths varying from 32.5 to 176.5 Mbps and standard deviations ranging from 13.5 to 26.8 Mbps. Among these traces, we include a LTE trace with an average throughput of 32.5 Mbps to represent lower-bandwidth wireless network scenarios.

Devices. Our server setup includes a commodity machine with an Intel Xeon Gold 6230 CPU @ 2.10GHz and 32GB of RAM. We utilize two client hosts: (1) a desktop with an Intel Core i9-10900X CPU @ 3.70GHz, an NVIDIA GeForce RTX 3080Ti GPU, and 32GB of RAM, serving as our standard evaluation client; (2) an Orange Pi embedded system equipped with a Rockchip RK3588S 8-core 64-bit processor @ 2.4GHz and 8GB of RAM, comparable to the Meta Quest 3 (Meta, 2024) with a Qualcomm XR2 chip (Qualcomm, 2024).

User Traces. We employ multi-user 6DoF motion traces during video playback, replicating user movements in some experiments.

Baselines for SR Quality Evaluation: For evaluating the quality of super-resolution (SR) techniques, we employ three primary baselines: GradPU (He et al., 2023), and a naive interpolation method with a dilation factor of 1. GradPU serves not only as a baseline for assessing SR quality but also as a benchmark for runtime performance. Additionally, we implemented Yuzu (Zhang et al.) with its SR pipeline, for runtime comparisons and end-to-end evaluations. To ensure a fair comparison, we disable Yuzu's cache and delta-coding mechanisms, as these features are orthogonal to our SR approach. We also compare with Vivo (Han

et al., 2020), a visibility-aware volumetric video streaming system with preemptive viewport adaptation.

7.2 SR Quality

To rigorously evaluate the impact of super-resolution (SR) techniques on image and geometric quality, we conduct a comprehensive set of experiments. These experiments involve multiple users watching videos while their 6 Degrees of Freedom (6DoF) motion traces are recorded. For each SR setting, we render viewports as images, denoted as $\{\text{ISR}\}$, and repeat this process with the original videos to capture the baseline images, denoted as $\{\text{Igt}\}$. We then compute the Peak Signal-to-Noise Ratio (PSNR) by comparing each image in $\{\text{ISR}\}$ against its corresponding image in $\{\text{Igt}\}$. Additionally, we measure the Chamfer Distance to evaluate the geometric accuracy by comparing the SR-enhanced point clouds with the corresponding ground truth point clouds. The corresponding videos are downsampled to 100K and upsampled to $\times 2$ and $\times 4$.

7.2.1 Interpolation with dilation

In this analysis, we explore the effect of varying the dilation factor ("d") within the kNN interpolation process used in SR. Specifically, we assess the PSNR outcomes for both $\times 2$ and $\times 4$ super-resolution settings, presented in Figures 7 and 9. The results indicate a clear improvement in PSNR values when dilation is increased from $K4d1$ to $K4d2$, suggesting better image quality across different upsampling ratios. Concurrently, the Chamfer Distance results, shown in Figures 8 and 10, reveal a reduction in geometric discrepancies as dilation is incorporated. These findings illustrate that enhanced dilation provides a broader spatial context during interpolation which not only improves visual clarity but also significantly enhances the geometric accuracy of the super-resolved images.

7.2.2 LUT refinement

The LUT refinement process targets the optimization of interpolated point cloud data by looking up the precomputed offsets stored in the Look-Up Table. This step is crucial for enhancing the final SR quality. $K4d2 - lut$ represents our approach using network generated LUT. By analyzing both PSNR and Chamfer Distance metrics post-refinement, as depicted in Figures 7, 9, 8, and 10, we observe noticeable improvements in image fidelity and geometric accuracy. By comparing the GradPU and our lut results, we show that the integration of our interpolation with dilation adjustments and subsequent LUT refinement ensures that the accelerated SR process does not compromise on visual or geometric quality.

Overall, our experimental analysis demonstrates that the applied SR techniques not only preserve but significantly enhance both the visual and geometric qualities of the images. Notably, achieving consistent PSNR values over 30 dB across various settings underscores the excellent visual quality of our SR process (Thomos et al., 2005; Dasari et al., 2020).

7.3 Runtime Performance

We now focus on analyzing how different parts of our SR system contribute to the overall latency in both desktop and mobile settings. The experiments are conducted using 100Mbps wired netork with our continuous ABR disabled.

Interpolation Speedup: As shown in Figures 11, our optimized interpolation achieves significant speedups across different platforms. On the Orange Pi, we maintain 3.7×-3.9× speedup over vanilla interpolation, reaching 31.2 FPS at 8× upsampling (vs vanilla's 8.0 FPS). The improvement is even more substantial on the commordity GPU empowered by the cuda implementation, where we achieve 7.5×-8.1× speedup, processing at 357.1 FPS for 2× upsampling and maintaining 138.9 FPS even at 8× upsampling. This consistent performance across upsampling ratios demonstrates the effectiveness of our spatial-parallel optimization in reducing the kNN search overhead.

GPU Memory Usage: As shown in Figure 15, Our approach using one LUT can improve the GPU memory usage by 86% compared to GradPU and is comparable to Yuzu with frozen tensorflow model in c++, which is particularly beneficial for devices with limited GPU resources.

Runtime Breakdown: Figure 16 provides a detailed breakdown of the time spent in each stage of the SR process on both desktop and Orange Pi platforms. On both GPU (desktop) and mobile scenario, kNN search takes the most significant portion of time, followed by interpolation, with LUT refinement consuming the least time.

SR Performance on Different Platforms: Figure 17 and Figure 18 further illustrate the SR runtime on a commodity GPU and the impact of various upsampling ratios on the Orange Pi, respectively. We show the average upsampling rate(in FPS). On the desktop (Figure 17), our method outperforming Yuzu by $8.4\times$ and outperforming GradPU by $46400\times$. Our approach is mainly benefited by the efficient LUT look up compared to heavy neural network inferencing even if accelerated in a frozen cpp implementation as Yuzu (Zhang et al.) did. As shown in Figure 18, the upsampling speed on the Orange Pi maintains relatively stable even as the upsampling ratio increases. This is due to the fact that the main bottleneck (shown in Figure 16) of our SR approach lies in the kNN based interpolation which is mainly related to the number of input points.

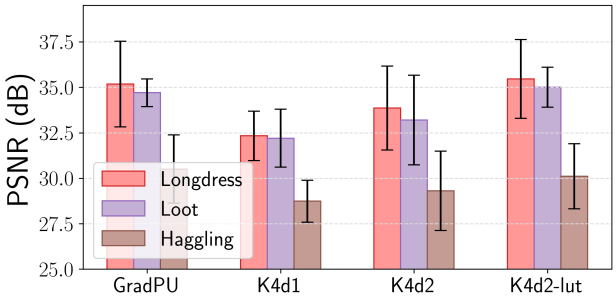

Figure 7. PSNR for ×2 SR

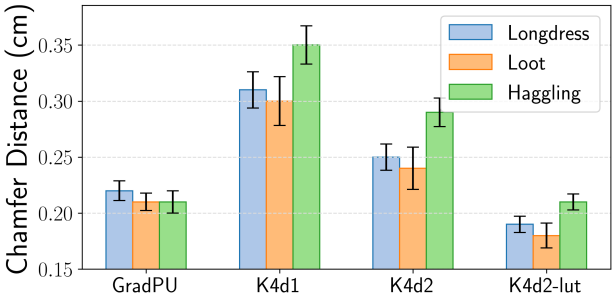

Figure 8. Chamfer Distance for ×2 SR

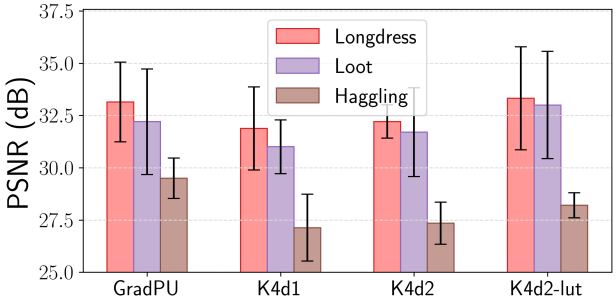

Figure 9. PSNR for ×4 SR

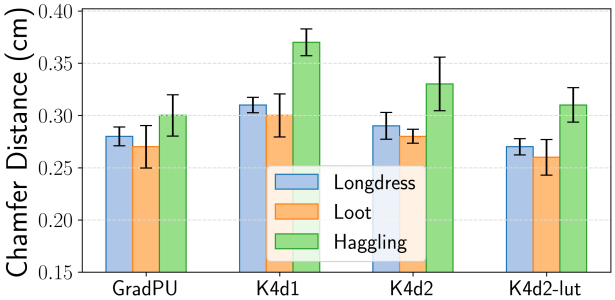

Figure 10. Chamfer Distance for ×4 SR

7.4 QoE measurement under various network conditions

We evaluate the QoE of our system under different network conditions, using various videos and associated motion traces. We compare our approach with Yuzu-SR, a re-implementation of Yuzu (Zhang et al.) with caching and delta coding disabled for fair comparison. The normalized QoE results are shown in Figure 12 and the data usage (defined by the total downloaded bytes including SR models for yuzu SR and meta data) results are shown in Figure 13.

Stable Bandwidth. We first consider a stable bandwidth scenario with a throughput of 50Mbps. Under this condition, our system achieves a normalized QoE of 100, while Yuzu-SR and Vivo (Han et al., 2020), on the other hand, achieves a normalized QoE of 75.8 and 43.2, respectively. VoLUT is mainly benefited from the fast SR speed for any-scale upsampling compared to Yuzu-SR and the efficacy of SR compared to Vivo.

In terms of data usage, VoLUT can reduce it by 23% compared to Yuzu-SR because our fine-grained ABR algorithms allows any-scale downsampling rate for transmission while Yuzu-SR's discrete SR options (1x2, 2x2, 1x3, 1x4, 4x1, 2x1) provide less optimal decision. VoLUT can reduce data usage by 31% compared to Vivo due to the significant point cloud reduction during streaming with our downsampling and SR manner.

Fluctuating Bandwidth. We also evaluate the performance of our system under fluctuating bandwidth conditions using real-world LTE traces(§ 7.1). In this scenario, our system achieves a normalized QoE of 83 while consuming only 17% of the data. In comparison, Yuzu-SR achieves a normalized QoE of 57 but requires 31% of the data. Notably, QoE performance gain is higer under LTE (26) traces compared to the stable trace (24) is due to the the limited bandwidth, which pushes the systems to fetch content with lower density and introduces more SR workload. Thus VoLUT's fast SR will benefit more under limited network resources.

7.5 Ablation study

How the system is compared to Yuzu and simple Adaptation in terms of FPS and resource consumption?

Our ablation study evaluate three variants of our system (In Table 2). Figure 14 shows the normalized QoE vs. data usage trade-off for these system variants under fluctuating bandwidth conditions.

Our proposed system (H1) achieves the best balance between QoE and data usage. It maintains a high normalized QoE of 98 while consuming only 31% of the data compared to the baseline. Using discrete ABR (H2) instead of continuous ABR leads a reduction of normalized QoE by 15.3% and increases the data usage by 14% compared to H1. This highlights the advantage of our continuous ABR approach in fine-grained bitrate adaptation, which allows better utilization of available bandwidth and reduces data consumption.

Replacing our faster SR method with Yuzu's SR (H3) results in a notable drop in QoE by 36.7% compared to H1 while still consuming 48% of the data. This emphasizes the faster SR speed will also benefit the stall time which is a major

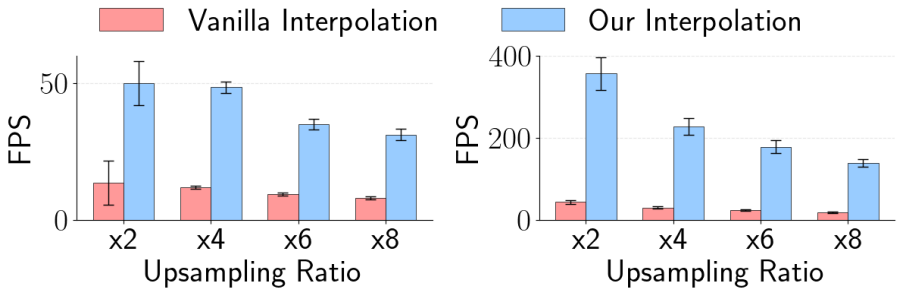

Figure 11. Interpolation FPS on Orange Pi (Left) and NVDIA 3080Ti (Right)

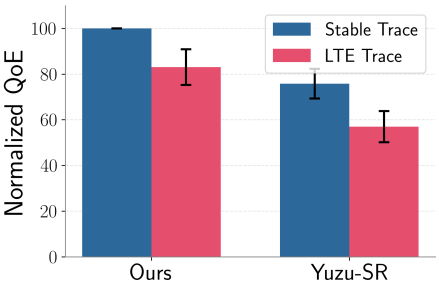

Figure 12. Normalized QoE

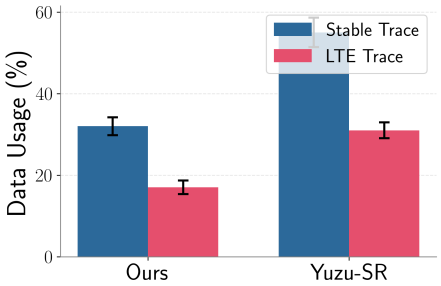

Figure 13. Data usage

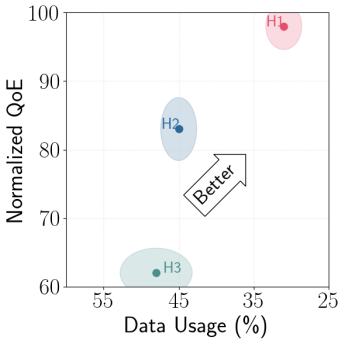

Figure 14. QoE vs. Data usage over fluctuating bandwidth (LTE traces)

H1	VoLUT with Continuous ABR
H2	VoLUT with Discrete ABR
H3	VoLUT with Discrete ABR and Yuzu SR

Table 2. Variants of VoLUT

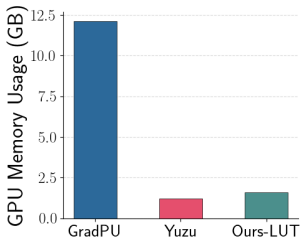

Figure 15. GPU memory usage (3080Ti Desktop).

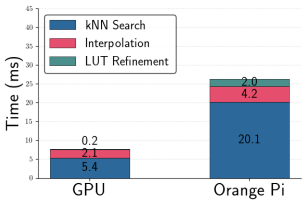

Figure 16. End to End SR Runtime breakdown

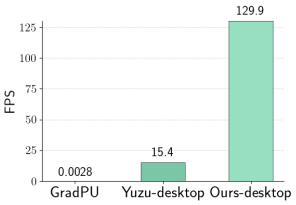

Figure 17. SR Runtime on commodity GPU

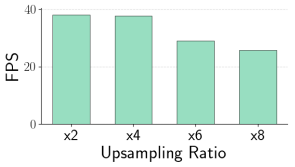

Figure 18. SR Runtime on OrangePi under various upsampling ratio

components of the QoE(§ 5).

8 Related Work

Volumetric video streaming: Volumetric video streaming has gained significant attention in recent years due to its ability to provide immersive and interactive experiences. Several studies have focused on point-cloud-based volumetric video streaming (Lee et al., 2020; Han et al., 2020; Gül et al., 2020a;b; Hosseini & Timmerer, 2018; Park et al., 2018; Qian et al., 2019; Van Der Hooft et al., 2019). DASH-PC (Hosseini & Timmerer, 2018) extends the Dynamic Adaptive Streaming over HTTP (DASH) protocol to support volumetric videos. ViVo (Han et al., 2020) introduces visibility-aware optimizations to improve the streaming efficiency of volumetric videos. GROOT (Lee et al., 2020) focuses on optimizing point cloud compression for volumetric video streaming.MuV2 (Liu et al., 2024) and Vues (Liu et al., 2022b) applies transcoding to 3D contents at server and transmit 2D frames to clients. YuZu (Zhang et al.) is a recently proposed volumetric video streaming system that employs deep learning-based point cloud super-resolution to enhance the visual quality of low-resolution content at the receiver's end. However, these existing works do not explore the potential of interpolation and lut-based approaches for efficient point cloud super-resolution in volumetric video streaming.

Look-up table (LUT) based inference speed-up: Look-up tables have been widely used in various domains to accelerate computation and reduce memory footprint. In the context of image processing, several works have explored LUT-based approaches. Jo *et al.* (Jo & Joo Kim, 2021) and Liu*et al.* (Liu et al.) propose LUT-based method for efficient single-image super-resolution. LUT-NN (Tang et al., 2023) introduces a LUT-based neural network inference framework that achieves significant speedup and memory reduction compared to traditional neural network inference. DLUX (Gu et al., 2020) presents a LUT-based near-bank accelerator for efficient deep learning training in data centers. Sutradhar *et al.* (Sutradhar et al., 2021) explores the use of LUTs in processing-in-memory architectures for deep learn-

ing workloads. These works demonstrate the effectiveness of LUT-based approaches in various domains. However, to the best of our knowledge, no prior work has investigated the application of LUTs for point cloud super-resolution in volumetric video streaming.

9 Conclusion

In this paper, we present VoLUT, a novel system that leverages LUT-based point cloud super-resolution for efficient and high-quality volumetric video streaming. Through combining accelerated dilated interpolation and Look Up table based refinement, VoLUT achieves real-time performance even on mobile devices, significantly reduces bandwidth requirements, and enhances the user experience. Our extensive evaluations demonstrate the effectiveness of VoLUT in delivering high-quality volumetric video content while adapting to network conditions and user preferences. The contributions of our work lay the foundation for future research and development in the field of volumetric video streaming, opening up new possibilities for immersive and accessible volumetric experiences.

Acknowledgements

We thank the anonymous reviewers for their insightful comments. This work was supported in part by NSF CNS-2112562, CNS-2107060, CNS-2213688, CNS-2312716, and the US Department of Commerce award 70NANB21H043.

References

Ef eve - volumetric video platform (vr & desktop), 2024. URL https://store.steampowered.com/app/774721/EF_EVE__Volumetric_Video_Platform_VR__Desktop/.

Compare Meta Quest VR headsets, 2024. URL https://www.meta.com/quest/compare/. Product comparison page for Meta Quest virtual reality headsets.

Orange Pi 5B with 32gb emmc, 2024. URL http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5B-32GB.html. Single board computer product specifications.

Volumetric video games, 2024. URL https://www.volumetricvideogames.ca/.

Dasari, M., Bhattacharya, A., Vargas, S., Sahu, P., Balasubramanian, A., and Das, S. R. Streaming 360-degree videos using super-resolution. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1977–1986. IEEE, 2020.

Emad, M., Peemen, M., and Corporaal, H. Moesr: blind super-resolution using kernel-aware mixture of experts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3408–3417, 2022.

Gu, P., Xie, X., Li, S., Niu, D., Zheng, H., Malladi, K. T., and Xie, Y. Dlux: A lut-based near-bank accelerator for data center deep learning training workloads. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(8):1586–1599, 2020.

Gül, S., Podborski, D., Buchholz, T., Schierl, T., and Hellge, C. Low-latency cloud-based volumetric video streaming using head motion prediction. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 27–33, 2020a.

Gül, S., Podborski, D., Son, J., Bhullar, G. S., Buchholz, T., Schierl, T., and Hellge, C. Cloud rendering-based volumetric video streaming system for mixed reality services. In *Proceedings of the 11th ACM multimedia systems conference*, pp. 357–360, 2020b.

Han, B., Liu, Y., and Qian, F. ViVo: visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–13, London United Kingdom, April 2020. ACM. ISBN 978-1-4503-7085-1. doi: 10.1145/3372224.3380888. URL https://dl.acm.org/doi/10.1145/3372224.3380888.

He, Y., Tang, D., Zhang, Y., Xue, X., and Fu, Y. Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions, April 2023. URL http://arxiv.org/abs/2304.11846. arXiv:2304.11846 [cs].

Hosseini, M. and Timmerer, C. Dynamic adaptive point cloud streaming. In *Proceedings of the 23rd Packet Video Workshop*, pp. 25–30, 2018.

Jo, Y. and Joo Kim, S. Practical Single-Image Super-Resolution Using Look-Up Table. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 691–700, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00075. URL https://ieeexplore.ieee.org/document/9577923/.

Lee, K., Yi, J., Lee, Y., Choi, S., and Kim, Y. M. GROOT: a real-time streaming system of high-fidelity volumetric videos. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–14, London United Kingdom, September 2020. ACM. ISBN 978-1-4503-7085-1. doi: 10.1145/3372224.3419214. URL https://dl.acm.org/doi/10.1145/3372224.3419214.

Li, J., Zhou, J., Xiong, Y., Chen, X., and Chakrabarti, C. An Adjustable Farthest Point Sampling Method for Approximately-sorted Point Cloud Data, August 2022.

Li, R., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. PU-GAN: a Point Cloud Upsampling Adversarial Network, July 2019. URL http://arxiv.org/abs/1907.10844. arXiv:1907.10844 [cs].

Liu, C., Yang, H., Fu, J., and Qian, X. 4D LUT: Learnable Context-Aware 4D Lookup Table for Image Enhancement. URL http://arxiv.org/abs/2209.01749.

Liu, C., Yang, H., Fu, J., and Qian, X. 4D LUT: Learnable Context-Aware 4D Lookup Table for Image Enhancement, September 2022a. URL http://arxiv.org/abs/2209.01749. arXiv:2209.01749 [cs, eess].

Liu, Y., Jiang, B., Guo, T., Sitaraman, R. K., Towsley, D., and Wang, X. Grad: Learning for Overhead-aware Adaptive Video Streaming with Scalable Video Coding. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 349–357, Seattle WA USA, October 2020. ACM. ISBN 978-1-4503-7988-5. doi: 10.1145/3394171.3413512. URL https://dl.acm.org/doi/10.1145/3394171.3413512.

Liu, Y., Han, B., Qian, F., Narayanan, A., and Zhang, Z.-L. Vues: practical mobile volumetric video streaming through multiview transcoding. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pp. 514–527, Sydney NSW Australia, October 2022b. ACM. ISBN 978-1-4503-9181-8. doi: 10.1145/3495243.3517027. URL https://dl.acm.org/doi/10.1145/3495243.3517027.

Liu, Y., Zhou, P., Zhang, Z., Zhang, A., Han, B., Li, Z., and Qian, F. MuV2: Scaling up Multi-user Mobile Volumetric Video Streaming via Content Hybridization and Sharing. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 327–341, Washington D.C. DC USA, May 2024. ACM. ISBN 9798400704895. doi: 10.1145/3636534.3649364.

Long, C., Zhang, W., Li, R., Wang, H., Dong, Z., and Yang, B. PC^2-PU: Patch Correlation and Point Correlation for Effective Point Cloud Upsampling, July 2022. URL http://arxiv.org/abs/2109.09337. arXiv:2109.09337 [cs].

Luiten, J., Kopanas, G., Leibe, B., and Ramanan, D. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. URL http://arxiv.org/abs/2308.09713.

Meta. Compare meta quest products, May 2024. URL https://www.meta.com/quest/compare/.

Park, J., Chou, P. A., and Hwang, J.-N. Volumetric media streaming for augmented reality. In *2018 IEEE Global communications conference (GLOBECOM)*, pp. 1–6. IEEE, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Qian, F., Han, B., Pair, J., and Gopalakrishnan, V. Toward practical volumetric video streaming on commodity smartphones. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pp. 135–140, 2019.

Qian, G., Abualshour, A., Li, G., Thabet, A., and Ghanem, B. PU-GCN: Point Cloud Upsampling using Graph Convolutional Networks. URL http://arxiv.org/abs/1912.03264.

Qualcomm. Snapdragon xr2 gen 2 platform product brief. Online, 2024. URL https://docs.qualcomm.com/bundle/publicresource/87-73689-1_REV_A_Snapdragon_XR2_Gen_2_Platform_Product_Brief.pdf. Accessed: May 6, 2024.

Schnabel, R. and Klein, R. Octree-based Point-Cloud Compression.

Sutradhar, P. R., Bavikadi, S., Connolly, M., Prajapati, S., Indovina, M. A., Dinakarrao, S. M. P., and Ganguly, A. Look-up-table based processing-in-memory architecture with programmable precision-scaling for deep learning applications. *IEEE Transactions on Parallel and Distributed Systems*, 33(2):263–275, 2021.

Tang, X., Wang, Y., Cao, T., Zhang, L. L., Chen, Q., Cai, D., Liu, Y., and Yang, M. LUT-NN: Empower Efficient Neural Network Inference with Centroid Learning and Table Lookup, September 2023. URL http://arxiv.org/abs/2302.03213. arXiv:2302.03213 [cs].

Thomos, N., Boulgouris, N. V., and Strintzis, M. G. Optimized transmission of jpeg2000 streams over wireless channels. *IEEE Transactions on image processing*, 15(1): 54–67, 2005.

Van Der Hooft, J., Wauters, T., De Turck, F., Timmerer, C., and Hellwagner, H. Towards 6dof http adaptive streaming through point cloud compression. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2405–2413, 2019.

Wald, I. cudaKDTree: A CUDA-based k-d Tree Implementation. https://github.com/ingowald/cudaKDTree, 2023.

Wang, G., Xu, G., Wu, Q., and Wu, X. Two-Stage Point Cloud Super Resolution with Local Interpolation and Readjustment via Outer-Product Neural Network. *J Syst Sci Complex*, 34(1):68–82, February 2021. ISSN 1009-6124, 1559-7067. doi: 10.1007/s11424-020-9266-x. URL https://link.springer.com/10.1007/s11424-020-9266-x.

Wu, T., Pan, L., Zhang, J., Wang, T., Liu, Z., and Lin, D. Density-aware Chamfer Distance as a Comprehensive Metric for Point Cloud Completion. URL http://arxiv.org/abs/2111.12702.

Yang, C., Jin, M., Jia, X., Xu, Y., and Chen, Y. AdaInt: Learning Adaptive Intervals for 3D Lookup Tables on Real-time Image Enhancement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17501–17510, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01700. URL https://ieeexplore.ieee.org/document/9879870/.

Yang, M., Luo, Z., Hu, M., Chen, M., and Wu, D. A Comparative Measurement Study of Point Cloud-Based Volumetric Video Codecs. 69(3):715–726. ISSN 0018-9316, 1557-9611. doi: 10.1109/TBC.2023.3243407. URL https://ieeexplore.ieee.org/document/10050256/.

Yifan, W., Wu, S., Huang, H., Cohen-Or, D., and Sorkine-Hornung, O. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5958–5967, 2019.

Yin, X., Jindal, A., Sekar, V., and Sinopoli, B. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 325–338. ACM. ISBN 978-1-4503-3542-3. doi: 10.1145/2785956.2787486. URL https://dl.acm.org/doi/10.1145/2785956.2787486.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. PU-Net: Point Cloud Upsampling Network. URL http://arxiv.org/abs/1801.06761.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. PU-Net: Point Cloud Upsampling Network, March 2018. URL http://arxiv.org/abs/1801.06761. arXiv:1801.06761 [cs].

Zhang, A., Wang, C., Han, B., and Qian, F. YuZu: Neural-Enhanced Volumetric Video Streaming.