國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

量體視訊串流下之多視角轉碼最佳化

Optimization of Multiview Generation for Volumetric Video Streaming with Edge Transcoding

吳奕寶

Yi-Pao Wu

指導教授：謝宏昀 博士

Advisor: Hung-Yun Hsieh, Ph.D.

中華民國 112 年 7 月

July 2023

doi:10.6342/NTU202303932

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

量體視訊串流下之多視角轉碼最佳化

Optimization of Multiview Generation for Volumetric Video Streaming with Edge Transcoding

本論文係吳奕寶（R10942070）在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 112 年 7 月 26 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Communication Engineering on 26 / 07 / 2023 have examined a Master's thesis entitled above presented by Yi-Pao Wu (R10942070) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

謝安明　　廖婉君　　高榮鴻

（指導教授 Advisor）

系主任/所長 Director: 王鵬彥

致謝

碩士班兩年的時間過得很快，很喜歡碩班自由自在的生活，但其中也不乏很多痛苦的時刻，每次和老師討論前晚上對於報告內容的掙扎，每次計畫月報前一個禮拜要生出進度的煎熬，思考碩士論文方法的苦惱，這些辛苦的回憶都還歷歷在目。

這兩年的時間需要感謝很多人，有你們讓我這兩年可以更順利更充實的度過。謝謝指導教授謝宏昀老師耐心的教導，每次討論都願意耐心的聽我們每週的進度，也可以快速的從中聽出重點然後給出很核心的建議，除了研究相關的問題，在計畫一開始月報之前，老師也願意花好幾個小時的時間和我們一起討論報告投影片的內容，讓我更熟悉一個正式的口頭報告應該要具備什麼結構和元素，讓我具備全面的能力。謝謝實驗室同屆的同學，雖然大家做的題目不太一樣，但還是時常一起討論，也一起分擔實驗室和畢業需要完成的事項，特別感謝翊宏和定為，謝謝翊宏和我一起完成兩年的計畫，常常可以提供很獨特深入的想法，讓計畫可以順利的完成，後來每次和你討論都可以很快速的理解對方的意思，能有這樣的意見交流真的很開心，謝謝定為總是統籌實驗室大大小小的事情，也跳出來當很多學期的助教，而且真的超級認真在做研究，每次點開你的進度算是督促我認真做事的動力之一。感謝實驗室的學弟學妹，你們讓實驗室變得很溫馨，希望你們碩二的生活也可以順利。特別感謝高中好友泓毅，最後寫論文很崩潰的時候給我很實質和溫暖的建議，一起在路易莎做事都是我最有效率的時候，沒有你我一定沒有辦法把論文寫出來，很開心最後我們可以一起順利畢業。另外特別感謝珮珊，這半年常常聽我抱怨研究上遇到的事情，也一直陪我到各種地方做事，謝謝你每次溫柔的鼓勵和支持。最後感謝我的家人，讓我可以在這兩年的時間沒有負擔的專心完成學校的事情。

最後，感謝自己在最煎熬的時候沒有停下腳步，一直都覺得研究所除了是培養知識的地方，更是在訓練我們對抗壓力的意志力，每次痛苦的時刻都讓自己變得更強壯。終於結束了每天逐路易莎而居的生活，我會帶著在台大學會的事情繼續努力，謝謝台大帶給我的一切。

2023/08/09 吳奕寶 筆

doi:10.6342/NTU202303932

摘要

實現元宇宙的主要研究議題之一是量體視訊的串流，其需要極高的頻寬消耗、極低的延遲要求以及顯著的解碼負擔。本研究探索利用邊緣渲染 (edge rendering) 的串流系統，系統中根據視野預測結果對量體視訊 2D 視角進行轉碼。然而，視野預測的不準確性可能會因為其在偏移視點上降產生畫面而降低轉碼影像的品質。在最先進的邊緣輔助量體視訊串流系統中，選擇生成多個轉碼視角的位置是根據均勻步長移動預測位置，這種方法沒有將視野預測模型不同的準確性納入考慮，可能會顯著降低預渲染視圖的品質。本研究將虛擬視角合成技術納入串流系統並建立分析轉碼畫面的品質模型，該模型代表畫面品質與位置偏移之間的關係。基於充分的模擬結果，將模擬結果得到的品質模型作為目標，本研究提出建立在最佳量化問題之上的最佳化框架，用於選擇生成多個轉碼視角的位置，以最佳化期望品質，考慮了實證模擬結果而非僅依賴歐氏距離。基於這個最佳化框架，本研究設計了一種結合無梯度最佳化方法與競爭性學習向量量化的演算法，該演算法考慮用戶位置的機率分佈和品質模擬結果，動態地決定最佳的視角轉碼位置。我們的模擬結果顯示，我們提出的演算法相比最先進的量體視訊串流系統方法，在影片串流過程中可以在 55%至 83% 的時間帶來畫面品質的提升。

doi:10.6342/NTU202303932

ABSTRACT

One of the major research topics to enable the metaverse is the streaming of volumetric video, which comes with ultra-high bandwidth consumption, ultra-low latency requirement, and significant decoding overhead. This work explores the utilization of a streaming system with edge rendering, where 2D views of volumetric video are transcoded at the edge server according to the viewport prediction result. However, inherent inaccuracy of viewport prediction may degrade the quality of transcoded frames that are rendered at a deviated viewpoint. In the state-of-the-art edge-assisted volumetric streaming system, positions to generate multiple transcoded views are selected by shifting predicted position with uniform step size, in which a multiview generation approach without considering varying viewport prediction accuracy could significantly degrade the quality of pre-rendered views. This work incorporates virtual view synthesis techniques into the streaming system and establishes a quality model representing the relation between quality and position deviation based on thorough simulation results. With the quality model as a target, an optimization framework built upon the optimal quantization problem is formulated to select the positions for generating multiple transcoded views that optimize expected quality, taking into account empirical simulation results instead of relying solely on Euclidean distance. We propose an algorithm integrating concepts of gradient-free optimization with competitive learning vector quantization process to achieve maximal expected quality. The algorithm judiciously determines the best positions to transcode the views in consideration of the probability distribution of user's position and empirical simulation results. Our evaluations indicate that our proposed algorithms outperform baseline methods proposed by state-of-the-art transcoded volumetric streaming system with an improvement ratio ranging from 55% to 83% on a segment-by-segment basis.

doi:10.6342/NTU202303932

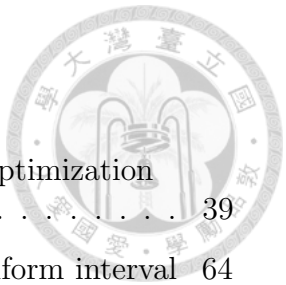
TABLE OF CONTENTS

doi:10.6342/NTU202303932

doi:10.6342/NTU202303932

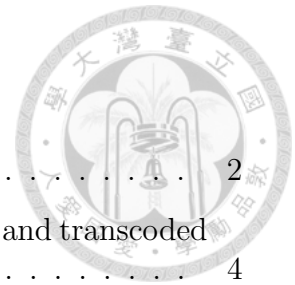
doi:10.6342/NTU202303932

LIST OF TABLES



doi:10.6342/NTU202303932

LIST OF FIGURES



doi:10.6342/NTU202303932

doi:10.6342/NTU202303932

CHAPTER 1

INTRODUCTION

With the proposal of the metaverse by Facebook, along with changes in human lifestyles owing to covid-19 pandemic, it is expected that a new version of immersive experience is going to become ubiquitous in our data lives in areas of commerce, education, and entertainment. Virtual reality (VR), augmented reality (AR), and mixed reality (MR) are the key technology to create the digital world of the metaverse, enabling multisensory interactions with virtual environments, digital objects, and people [4]. Various domains of knowledge should be leveraged to facilitate VR and AR. High-end display device, namely head-mounted display (HMD), and high-quality sensors are crucial to showcase the experience to users for the ultimate goal of reaching retina resolution. In addition, Deep learning for computer vision has been well-developed in recent years and contributes significantly to the development of VR and AR. Also, high-speed networking technologies, such as millimeter wave (mmWave) and TeraHertz communication, are necessary for the transmission of enormous amounts of data in immersive applications. Other than the mentioned research trajectories, optimization of the streaming process is also indispensable for providing a smooth experience since oscillating quality, prolonged latency, or stalling may greatly affect the user when taking pleasure in the virtual world.

The streaming techniques for 2D videos have been established for a long time. Adaptive bitrate streaming (ABS) methods, such as dynamic adaptive streaming over HTTP (DASH) and HTTP live streaming (HLS), are widely employed as the video streaming framework on Netflix, YouTube, or other media platforms. In adaptive bitrate streaming, videos are segmented into small chunks (e.g., 4 seconds) and encoded with several bitrates to generate versions of different quality. Then the server provides clients with a file containing streaming information about the video. For example, a DASH server sends an XML-based media presentation description (MPD) to clients with a list of available video streams and details such as codecs and resolutions used. Clients hold the key logic in the streaming system by choosing an adequate quality rate to download video segments matching the instant bandwidth of the wireless channel. Adaptive bitrate streaming aims to consistently provide the best video quality to users while maintaining continued video playback even with poor bandwidth by alternating to sub-optimal video versions with lower data rate requirement.

doi:10.6342/NTU202303932

(a) Example of point cloud data (b) Example of mesh data

Figure 1: Example of volumetric data from PyVista library [1]

Immersive videos come in formats including 360-degree video and volumetric video in VR and AR applications. 360-degree video allows users to rotate and view different angles, allowing them to focus on specific portions of the video that capture their interest. Volumetric video captures real objects or scenes with multiple cameras placed at different positions and pointing at different angles, creating a 3D representation of people or objects in the digital space. The user is enabled to move in 3D space along six degrees of freedom (6DoF), namely three rotational axes (roll, pitch, and yaw) and three translational axes (surge, sway, and heave). In other words, a user can view and interact with the objects by freely walking and viewing them as if the virtual objects were really there. Point cloud and mesh are two major representations of volumetric video. Point cloud data is composed of a great number of 3D points with texture and position information. Mesh data formulates 3D objects with multiple interconnected polygons, such as triangles and quadrangles, to form the object's surface. Fig. 1(a) and 1(b) demonstrate example images of point cloud data and mesh data respectively.

The concept of adaptive bitrate streaming, commonly used in conventional 2D videos, is extended to immersive video streaming. The challenges of streaming immersive content consist of two major items: ultra-large data volume and ultra-low latency constraint. The data size of 360-degree videos is estimated to be 16 times that of 2D videos, and the size of point cloud is about 2.5 times compared to 360-degree video [5]. Motion-to-photon (MTP) latency, the delay between user movement and video playback, has to be limited when streaming immersive video as users may easily encounter motion sickness when the MTP latency is too long. Much research has proposed methods to refine adaptive bitrate streaming techniques to overcome the obstacles for streaming 360-degree videos [6]. Nevertheless, innovating streaming volumetric video is challenging due to the massive amount of

doi:10.6342/NTU202303932

data it requires, which exceeds the bandwidth limitations of current wireless communication technology. Additionally, the image processing involved in volumetric video consumes more energy and increases latency on the user's mobile device. Currently, many unexplored problems remain due to the limited research focused on designing a dedicated streaming system for volumetric video that is practical within the constraints of existing networking techniques and the capabilities of mobile display devices.

Existing solutions for volumetric video systems can be categorized into two main groups: direct streaming and transcoded streaming. When employing direct transmission, clients receive data in point cloud or mesh format and ensure a satisfactory QoE by rendering the viewport corresponding to user's viewpoint during playback. However, streaming 3D content directly can lead to high bandwidth consumption. Furthermore, long decoding overhead brought about by complicated decoding process such as kd-tree [7] may surpass the limitation of MTP latency. In transcoded streaming, the edge server transcodes the 3D content into 2D video frames in real-time based on viewport prediction results and transmits encoded 2D video to the clients. Single-view transcoding is introduced in [8, 9], where the edge server pre-renders a single view from point cloud data so as to meet up with the playback time and satisfy the MTP delay constraint. In [3], multi-view transcoding for a single user is proposed. The authors recommend that inaccuracy of viewport prediction model can significantly affect the quality of pre-rendered view when applying single-view transcoding, since the actual viewport of users may displace a lot from the predicted result. Transcoded streaming allows for significant conservation of bandwidth even with multiple transcoded views. Fig. 2 presents the approximated values of required bandwidth for point cloud and transcoded views[1].

While Vues [3] provides a comprehensive blueprint for designing an edge-assisted volumetric streaming system with multiview transcoding, there are still areas where further improvements can be made. In Vues, the method proposed to select positions to render the transcoded views is based on simple heuristics, where limitations on the searching space may lead to degraded quality of pre-rendered views. This work aims to boost the QoE of clients by conducting thorough research on the relationship between the quality of transcoded 2D frames and the variations in viewport prediction results. Additionally, the virtual view synthesis

[1]The data volume of point cloud is estimated with a compression ratio of Draco library [10] indicated in [7]. As for transcoded views, the data volume of a single view is approximated with simulated result encoded with H.264 and 25 views of 720p (1280×720) are applied in total in the calculation. The bandwidth is simulated under a video frame rate of 30 FPS.

doi:10.6342/NTU202303932

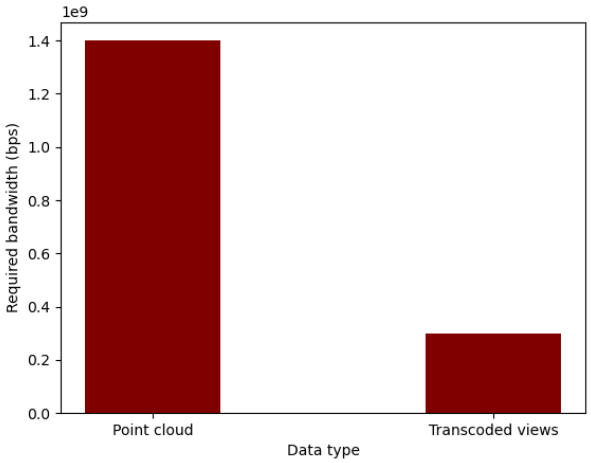


Figure 2: Comparison between required bandwidth of point cloud and transcoded views

approach is incorporated into the system on client side to generate a view of enhanced quality fitting to the actual viewport of users. With an in-depth analysis of the similarity between pre-rendered views and the actual view seen by users, we formulate the multiview generation problem into an optimal quantization problem with a distortion function built upon our empirical simulation results. Also, an effective algorithm gradient-free competitive learning vector quantization (GF-CLVQ), enhanced from a well-established high-dimensional quantization solution, is proposed for the minimization of expected distortion function. The proposed solution searches the optimal set of reference positions for transcoded views, extending the searching space with an effective process. Furthermore, our method could dynamically adapt to varying levels of viewport prediction accuracy and is compatible with all kinds of underlying virtual view synthesis techniques. The results of exhaustive simulations manifest that quality can be improved with identical required number of views, which means that the required bandwidth remains at the same level when applying optimized transcoding.

We summarize the contributions of this work as follows:

1. We apply virtual view synthesis technique to transcoded volumetric streaming system, which enhances the quality of transcoded views, and establish a quality model based on empirical simulation of the quality of transcoded views.

2. We formulate the decision of multiview generation on the edge server into an optimization framework based on optimal quantization problem that dynamically adapts to the distribution of viewport prediction error to maximize

doi:10.6342/NTU202303932

the expected value of user's perceived quality.

3. We propose gradient-free competitive learning vector quantization (GF-CLVQ) algorithm, revised from a well-established quantization solution for high-dimensional probability distribution to accommodate empirical simulation results. The proposed algorithms outperform baseline methods proposed by state-of-the-art transcoded volumetric streaming system with improvement ranging from 55% to 83% on a segment-by-segment basis.

doi:10.6342/NTU202303932

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Virtual View Synthesis

Virtual view synthesis is a process of generating novel views of an object or a scene from a given set of images and corresponding camera orientation information. In other words, the process renders virtual views that are not actually captured by cameras, which may be useful concerning increasing frame rate by rendering additional frames [11] or creating additional views from originally captured images for 3D TV system [12–14] or even free-viewpoint video [15, 16] and virtual reality. This work seeks to incorporate a virtual view synthesis technique into a volumetric video streaming system to create high-quality views with limited communication and computation resource demand for the user devices and the entire system. This section first gives a general overview of virtual view synthesis techniques and then explains the preliminary multiple view geometry model used in this work.

2.1.1 Depth Image

A depth image, also known as a depth map, is an image in which each pixel stores distance information relative to the camera as 8-bit gray values. In this representation, a gray value of 0 indicates the closest distance, while a value of 255 corresponds to the furthest distance. For each depth image, with the nearest clipping plane defined by the value Z_{min} and the furthest clipping plane specified with the value Z_{max}, the depth value of each pixel is normalized with the two main clipping planes. In the case of linear quantization, the relation between the absolute depth value Z and the pixel intensity can be given as

$$Z = Z_{min} + \frac{I}{255}(Z_{max} - Z_{min}), \tag{2.1}$$

where I represents the intensity of the pixel. Depth maps can be captured by sensors such as structured light sensors or time-of-flight sensors. Also, depth maps can be rendered directly from 3D models. An example of depth map of a volumetric object is demonstrated in Fig. 3 with the corresponding texture image.

2.1.2 Depth Image Based Rendering

Depth-image-based rendering (DIBR) is a long-standing approach of virtual view synthesis. The process generates virtual views that are not actually captured

doi:10.6342/NTU202303932

(a) Depth map (b) Texture image

Figure 3: Example of depth map and corresponding texture image

from neighboring color images and associated per-pixel depth maps [12]. Here we refer to the input images of DIBR as reference images. Performing a 3D transform to synthesize new images is also called 3D warping in computer graphics literature [11], which is applicable to increasing frame rate by rendering additional frames [11] or creating a left and right view for 3D TV system from originally captured images [12–14].

Multiple challenges must be confronted when adopting DIBR for virtual view synthesis, including occlusion handling, ghosting effects, and dealing with non-Lambertian surfaces, which occurs when the color of an object changes with different viewing directions. A lot of researchers [17–20] have been working on these problems to improve the quality of rendered images.

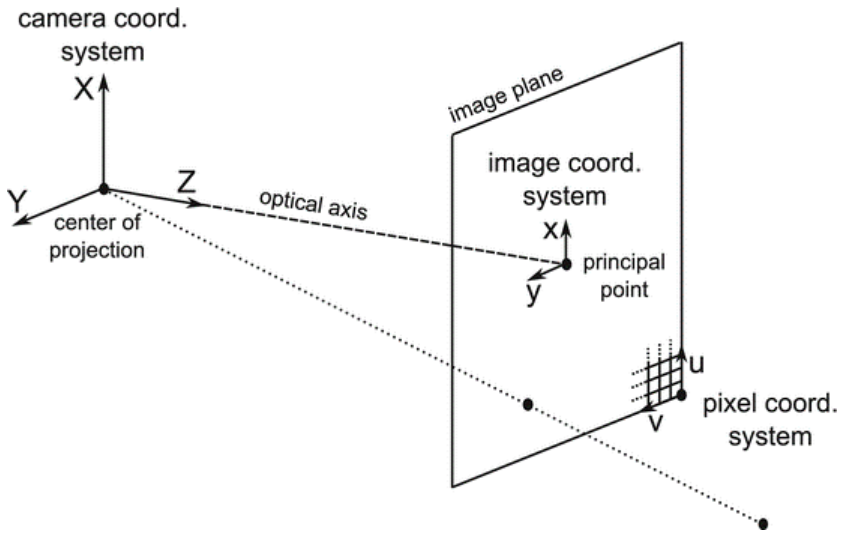


Figure 4: Illustration of pihole camera model [2]

2.1.3 Pinhole Camera Model

A general camera is modeled as a pinhole in DIBR, defining how a 3D world coordinate point is projected onto a 2D image plane. The extrinsic matrix and

doi:10.6342/NTU202303932

intrinsic matrix contain the essential parameters for a pinhole camera. The extrinsic matrix $\mathbf{E}$ contains rotation and translation information depending on its location and orientation; the intrinsic matrix $\mathbf{K}$ contains internal details. Given a point $\mathbf{P} = [x, y, z, 1]^T$ in world coordinates and a point $\tilde{\mathbf{P}} = [u, v, 1]^T$ on 2D image plane using homogeneous coordinates, where x, y, z are used for world coordinates and u, v are used for 2D image coordinates,the relation between projected pixel point $\tilde{\mathbf{P}}$ on image plane and $\mathbf{P}$ can be expressed as [21]

$$D\tilde{\mathbf{P}} = \mathbf{KEP}, \tag{2.2}$$

with intrinsic matrix and extrinsic matrix being

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}_{3\times3}, \tag{2.3}$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times4}, \tag{2.4}$$

where f is the focal length, p_x and p_y are the principal points (center of image plane), $\mathbf{R}$ and $\mathbf{T}$ are the world-to-camera rotation matrix and translation offset respectively. Given α, β, γ representing roll, yaw, and pitch with z-axis pointing forward and y-axis pointing up, rotation matrix $\mathbf{R}$ can be obtained from

$$\begin{aligned} \mathbf{R} &= \mathbf{R_z}(\alpha)\mathbf{R_y}(\beta)\mathbf{R_x}(\gamma) \\ &= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}, \end{aligned} \tag{2.5}$$

and $\mathbf{T}$ represents the coordinates of the origin of the world coordinate system in the 3-dimensional coordinates of the camera, i.e., $[t_x, t_y, t_z]^T$.

2.1.4 3D Image Warping

In general, 3D warping can be understood as a two-step procedure: deprojection of 2D pixels in the reference image into 3D world coordinates and, secondly, reprojection of the 3D points onto the 2D image plane of the target view. These two processes can also be represented in the matrix-form formula based on the pinhole camera model [19, 20]. Let D_r be the depth of pixel in the reference image,

doi:10.6342/NTU202303932

$\mathbf{K_r}$ and $\mathbf{E_r}$ be the intrinsic and extrinsic matrix of reference view, the deprojection process is given as

$$\mathbf{P} = D_r \mathbf{E_r^{-1}} \mathbf{K_r^{-1}} \tilde{\mathbf{P}}_\mathbf{r}. \tag{2.6}$$

Let D_t be the depth of pixel in the target image, $\mathbf{K_t}$ and $\mathbf{E_t}$ be the intrinsic and extrinsic matrix of target view, the reprojection process can be written as

$$\tilde{\mathbf{P}}_\mathbf{t} = \frac{1}{D_t} \mathbf{K_t} \mathbf{E_t} \mathbf{P} = \frac{D_r}{D_t} \mathbf{K_t} \mathbf{E_t} \mathbf{E_r^{-1}} \mathbf{K_r^{-1}} \tilde{\mathbf{P}}_\mathbf{r}, \tag{2.7}$$

where inverse of extrinsic matrix $\mathbf{E}$ can be proofed to be

$$\mathbf{E^{-1}} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R^T} & -\mathbf{R^T T} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{2.8}$$

2.1.5 Deep Learning-Based Method

Deep learning-based approaches have gained great attention for virtual view synthesis. In [22–24], deep learning models assist with DIBR impainting and hole-filling process. In [25, 26], convolutional networks are presented to predict images from arbitrary views given images of different views seeing the same object as training data. Recently, the method of neural radiance fields (NeRF) [27, 28] has been proposed as a new branch of view synthesis techniques. Fully connected deep networks are utilized in NeRF, taking 3D coordinates and two viewing angles as input data, which generates volume density and view-dependent emitted radiance at specific spatial locations.

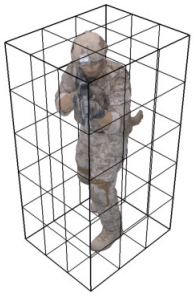
This work applies DIBR with forward warping and bilinear splatting as a virtual view synthesis module. However, it is important to note that the algorithm proposed remains flexible and adaptable to different virtual view synthesis algorithms. The proposed algorithm is adapted to the quality distribution of rendered images from variant virtual view synthesis algorithms, allowing for a versatile approach.

2.2 Related Work

As people seeking a more absorbing experience of entertainment rather than merely watching 2D videos, a number of researches regarding immersive video have been proposed by the academia. One of the major topics is the streaming of immersive video since it takes more communication and computation resource to process data formats such as 360-degree video and volumetric video. Despite the abundance of research about streaming 360-degree videos, exploration for efficiently streaming volumetric video in the form of point cloud or mesh is still in its pioneering state. Proposed as a promising approach, edge-assisted systems with

doi:10.6342/NTU202303932

(a) 360 video segmentation



(b) Point cloud segmentation

Figure 5: Segmentation of immersive video

transcoding techniques have been introduced to facilitate volumetric video streaming within the limitations of current wireless communication technology. We first introduce the basic concept of streaming 360-degree video and then present related work about volumetric video streaming. Subsequently, we delve into the existing methods of edge-assisted volumetric streaming systems. In particular, our simulation results pinpoint that there is much room for improvement regarding the multiview generation method in the current volumetric video streaming system.

2.2.1 360-Degree Video Streaming

Techniques of 360-degree video streaming involve two innovations compared with conventional 2D videos: tiling and viewport prediction. Streaming entire video to the users may cause undesirable resource wastage since human's field of view (FoV) is limited, and it is impossible to watch the whole 360-degree view at the same time. To compensate for the inefficiency and maintain the same quality of experience (QoE) of users simultaneously, researchers have proposed to divide the video into tiles spatially [29–31], as shown in Fig. 5(a). Accompanied by a model to predict users' future viewport, the tiles which are predicted within users' viewport are transmitted at high quality, with the remaining tiles being discarded or delivered at lower quality. However, viewport prediction is not always accurate, and this may result in a quality drop. Solutions introduced to deal with prediction inaccuracy include fetching additional tiles with ranking of perceptually importance [32] or with reinforcement learning [33], and so on.

2.2.2 Direct Volumetric Video Streaming

Research on volumetric video streaming also follows the concept of 360-degree video streaming. [34] and [35] incorporated dynamic rate adaptation into point cloud streaming, proposing approaches to spatially segment point cloud objects

doi:10.6342/NTU202303932

into cells and refine DASH representation format for point cloud data. An example of point cloud segmentation is shown in Fig. 5(b). In [34], the authors first introduce methods of point cloud sub-sampling and compare rendering performance along with the objective visual quality of different point densities while not applying the concept to a practical streaming system. In [35], simple heuristics of adaptive streaming with the DASH framework are applied to point cloud data by greedily or uniformly allocating bit rate to high-ranking cells, whereas this approach lacks consideration of prolonged decoding time for point cloud data. In [36], a practical volumetric streaming system named after ViVo is put forward. Three visibility-aware optimizations are proposed, namely viewport visibility, occlusion visibility, and distance visibility, where depth information is taken into account to lower resource usage when transmitting point cloud data. In detail, video content that falls into a user's viewport yet occluded by others is eliminated and video quality is adjusted with distance from a user's viewpoint and cell of point cloud as reducing point cloud density of faraway objects may bring little difference. Nevertheless, encoded bitrate of point cloud after optimization is still at the order of about 100 Mbps, even with point cloud data of moderate data volume. When it comes to streaming point cloud to multiple users, the huge bandwidth requirement may overwhelm the wireless channel without further refinements.

2.2.3 Transcoded Volumetric Video Streaming

A volumetric video streaming system with cloud/edge-rendering is first suggested in [8] and [9]. The burdensome rendering workload is offloaded to a powerful cloud or edge server to save the computation power of the mobile display device with the help of viewport prediction techniques. Meanwhile, communication resource consumption and delay time of data transmission can be reduced since rendered 2D views are delivered instead of full volumetric content. In [9], a single viewport is pre-rendered on the cloud server based on the viewport prediction result. Nevertheless, rendering only a single view may lead to poor video quality performance when prediction results are drifted away from the actual user's viewport. In [3], the authors further extend the concept of single-view transcoding and develop Vues, an edge-assisted streaming system, that transcodes the volumetric content into multiple viewports on the edge server. A practical QoE model and adaptive streaming algorithm are proposed, cutting down 95% of bandwidth usage compared to directly transmitting volumetric videos while retaining similar video quality.

doi:10.6342/NTU202303932

2.2.4 Multiview Generation Method in Transcoded Volumetric Streaming System

In the transcoded streaming system proposed in [3], one of the most important modules is the one to generate multiview data. The multiview generation module selects appropriate viewpoints to render candidate 2D views to deliver to the users and allows the user to decide which view is the best at the playback time in accordance with the actual viewpoint. In [3], rotational movements are handled by broadening the field of view from default $90° \times 45°$ to $180° \times 90°$. As for translational movements, positions for multiple views are determined by enlarging the convex hull of three predicted positions with three prediction models, which always creates a 0.4 meters×0.4 meters coverage with 21 candidate positions. The transmitted views are then chosen heuristically from all the candidate views and the number of views is decided with the difference between results given by three simple viewport prediction models. The selected positions for creating candidate views are the green points sketched in Fig. 6, where value of step size ΔS is designated to 0.1 meters.

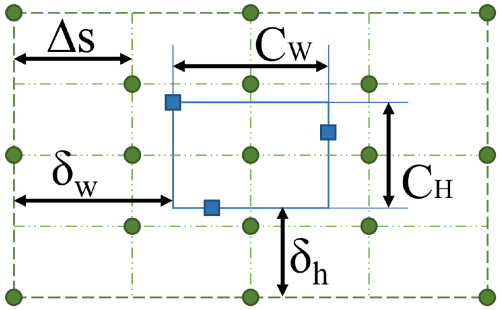
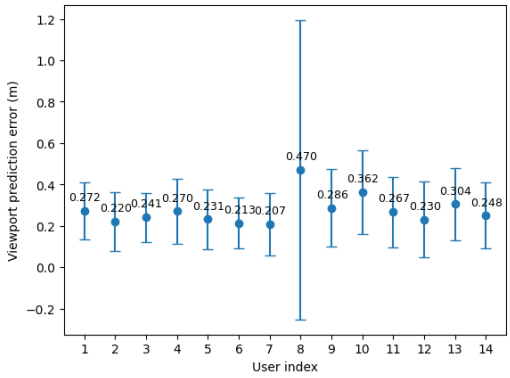


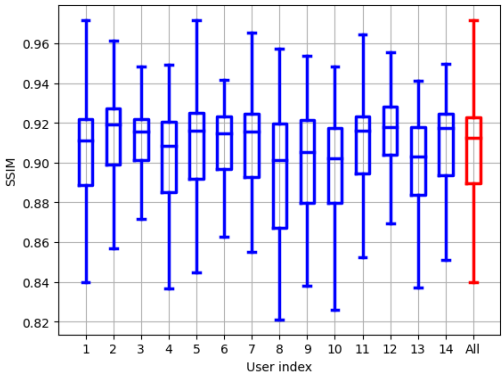
Figure 6: Multivew Generation System Proposed in [3]

Despite the usage of multiview transcoding technique, the inaccuracy and instability of viewport prediction may lead to high displacement of the presented views and actual views. Our simulation of viewport prediction result on a dataset of 14 users [37], presented in Fig. 7(a), implies that the same viewport prediction model may exhibit inconstant accuracy on different user traces. Applying an identical method to generate multiview on every case may result in inferior QoE. For example, when there is a significant viewport prediction error and the coverage area of selected positions is insufficient, the similarity between candidate views and the actual view during playback diminishes, resulting in substantial degradation of the user's QoE. In addition, implementing Vues' method leads to varying levels of quality, as indicated by the fluctuating SSIM values observed across 14 user traces with our simulated results of the SSIM over 14 users shown

doi:10.6342/NTU202303932



(a) Error bar of viewport prediction error with linear regression on dataset provided by [37]



(b) Results of method proposed in Vues [3]

Figure 7: Simulation results for motivation

in Fig. 7(b). Furthermore, the median values of SSIM fall within the range of 0.90 to 0.92. When compared to the outcomes of the dynamic volumetric streaming strategy suggested in [36], where the SSIM value exceeds 0.99, it becomes evident that there is still significant potential for enhancing the performance of the transcoded streaming system. While Vues has considerably reduced bandwidth usage, there is still an opportunity to enhance the user's experience through additional design in multiview generation. These design considerations can address challenges such as dynamic fluctuations in viewport prediction and variations in viewport prediction accuracy across multiple users. As outlined in the future work of Vues, this work intends to apply virtual view synthesis approach and propose an algorithm to render a set of views with better quality in accordance with user's actual viewport.

doi:10.6342/NTU202303932

CHAPTER 3

SYSTEM MODEL

3.1 System Architecture

The overall system architecture is presented in Fig. 8. It is important to mention that while our focus in this work is on the point cloud format, the proposed streaming system can be adapted to accommodate other volumetric video formats such as mesh-based video with minor adjustments to the rendering process. Building upon a holistic adaptive video streaming system with a client-server framework, an edge server is added to make possible efficient volumetric video streaming under the constraint of motion-to-photon delay and communication resources and even concurrent volumetric video streaming to multiple users. The system consists of a content provider, an edge server, and clients. Three entities are respectively in charge of:

- The content provider is the server storing volumetric video data. In the DASH framework, the server generates the MPD files containing metadata of video and sends requested video segments on user's demand. The underlying design of the server and format of MPD files need not be changed in the proposed system so that no additional effort will be involved when putting the system into practice.

- The edge server proactively sends requests to the content provider on behalf of the client for volumetric video segments before playback time. After receiving the volumetric video segments, the edge server transcodes the volumetric video into 2D multiview frames including texture images and depth maps before delivering video data to clients. Modules inside the edge server include viewport prediction, bandwidth estimation, multiview generation, and multiview image renderer. To reduce the adverse effects caused by inaccurate viewport prediction, the multiview generation module finds the best set of reference positions to generate reference views in order to minimize distortion caused by viewport drift based on the results of viewport prediction and bandwidth estimation. Details of viewport prediction and bandwidth estimation will be covered in the rest of this chapter and the design of multiview generation method is elaborated in the following chapters. Also, the process of offline simulation is explained in Chapter 4.

doi:10.6342/NTU202303932

- The client is the user's display device, which can include head-mounted displays, as well as other devices like mobile phones and personal computers. The client-side device first decodes the encoded multiview data. Afterward, the view selection module identifies the optimal view from the received set and determines whether virtual view synthesis should be performed. The best view will be delivered directly to the playback buffer if virtual view synthesis is not necessary. Conversely, virtual view synthesis is applied to create a better view with higher similarity to the actual view, adapting to user's actual viewpoint reactively. How the decision is made in the view selection module will be covered in the remaining sections of this chapter. Lastly, the chosen frames are displayed during playback, meanwhile, user trace data is collected and sent to the edge server for viewport prediction in the future.

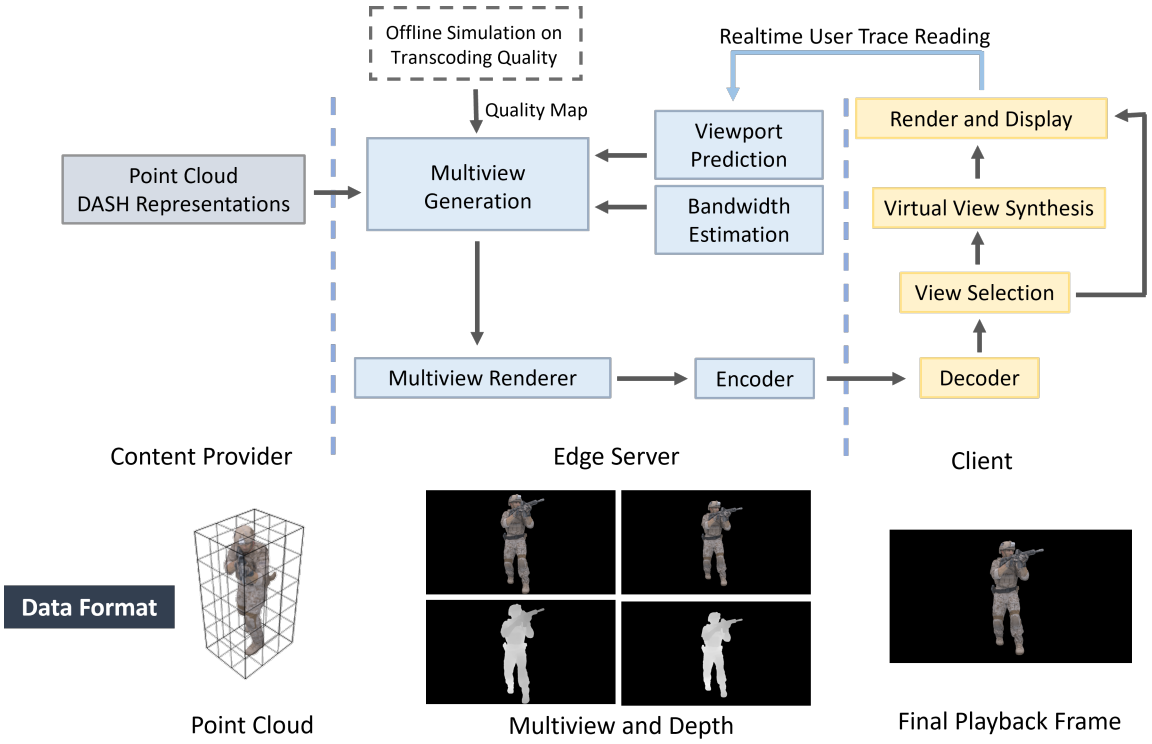


Figure 8: System Architecture

3.2 Viewport Prediction Model

Viewport prediction in 6DoF includes computing trajectories along three dimensions for translation, x, y, z, and three dimensions for rotation, roll, pitch, and yaw. The objective of viewport prediction is to analyze user's viewport in

doi:10.6342/NTU202303932

advance so that the transmission and rendering of future video frames can be initiated before the playback time to satisfy motion-to-photon delay in an immersive experience. In this work, we leverage the data of user traces in the past to predict the future user viewport. In order to preprocess the unevenly sampled data, following the recommendation of [37], the position data is upsampled using linear interpolation, and the rotation data, in quaternion format, is interpolated using spherical linear interpolation (SLERP) up to a frequency of 200Hz. Formula for SLERP on quaternions q_1 and q_2 with parameter u is defined as

$$SLERP(q_1, q_2; u) = \frac{\sin\big((1-u)\Omega\big)}{\sin\Omega} q_1 + \frac{\sin\big(u\Omega\big)}{\sin\Omega} q_2, \tag{3.1}$$

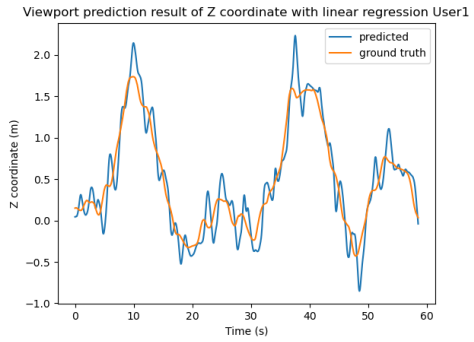
where $q_1 \cdot q_2 = \Omega$.

Then a lightweight linear regression model is applied as the viewport prediction module in the edge server in this work, which is also utilized in [3, 36]. Although more complicated models can be exploited here, linear regression serves as a suitable baseline model of viewport prediction. Our proposed algorithm is designed to dynamically adapt to the underlying accuracy of viewport prediction. The effect of linear regression is tested with 6DoF user trace data provided by [37]. The position data is with the z-axis pointing forward and the y-axis pointing up. Given the relatively low frequency of squatting movements in humans, we place particular emphasis on the results of viewport prediction for the x and z coordinates (i.e., the coordinates of the 2D plane). An illustration of viewport prediction result in comparison with the ground truth is shown in Fig. 9(a) and the cumulative distribution function (CDF) of the overall viewport prediction error is shown in Fig. 9(b). The viewport drift prediction error over 14 user traces is about 0.27 meters.

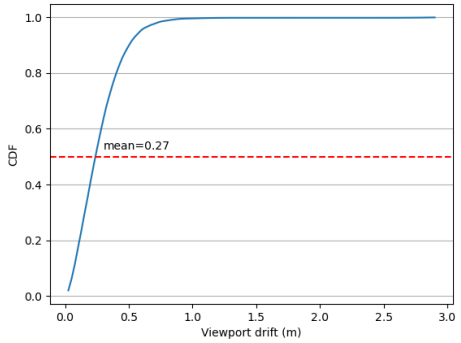
The following paragraph describes the method used for predicting the x and z coordinates in detail. Linear regression is applied to x and z coordinates separately with a history window of 0.5 seconds and a prediction window of 1.0 seconds. That is, at timestamp t seconds, we use data between $[t-0.5, t]$ seconds to find the linear relationship between position (x or z coordinates) and time. Afterward, the fitted line is used to predict the coordinate at time $t + 1.0$ seconds. Given 100 sample points of past user trace along x or z direction, the process of viewport prediction involves calculating the coefficients a_x, b_x for a linear equation $\hat{x} = a_x t + b_x$ that represents the relationship between the predicted x coordinate and the time samples, with equations

$$a_{x,t} = \frac{(\sum_{k=t-99}^{t} x_k)(\sum_{k=t-99}^{t} k^2) - (\sum_{k=t-99}^{t} k)(\sum_{k=t-99}^{t} k x_k)}{100 \times (\sum_{k=t-99}^{t} k^2) - (\sum_{k=t-99}^{t} k)^2}, \tag{3.2}$$

doi:10.6342/NTU202303932



(a) Viewport prediction result using linear regression on Z coordinate



(b) CDF of viewport prediction error on x-z plane

Figure 9: Results of viewport prediction

$$b_{x,t} = \frac{m(\sum_{k=t-99}^{t} kx_k) - (\sum_{k=t-99}^{t} k)(\sum_{k=t-99}^{t} x_k)}{100 \times (\sum_{k=t-99}^{t} k^2) - (\sum_{k=t-99}^{t} k)^2}. \tag{3.3}$$

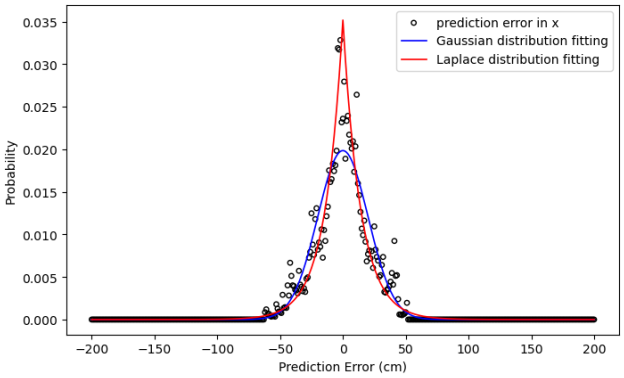
Similarly, the z coordinate can be obtained with the same approach.

3.2.1 Probability Distribution of User's Position

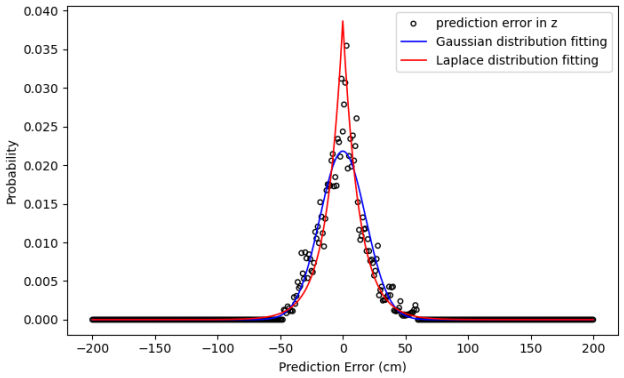
Due to the inherent inaccuracy of the viewport prediction model, the actual viewpoint of the user in the playback time should not be regarded as a deterministic value. Alternatively, user's actual viewpoint in an x-z place can be treated as a 2-dimension random variable with certain probability distribution formed by viewport prediction results and prior viewport prediction error. In 360-degree streaming problem, Gaussian distribution [38, 39] and Laplace distribution [40] are two probability models commonly used in the approximation of viewport prediction error of rotation. In this work, we determine the fitting results of both Gaussian and Laplace distributions for prediction results on user's translation. Fig. 10 presents the sample points of prediction error of x and z axis and the fitting curve following Gaussian and Laplace distributions. Results of normality tests including D'Agostino's K-squared test and Jarque–Bera test indicate that viewport prediction error on x and z direction of almost every user in the tested dataset rejects the null hypothesis of normally distributed. However, based on graphical observations, it is without confidence to assert that prediction error bears a closer resemblance to the Laplace distribution. In consequence, in this work we first utilize Gaussian distribution for its simulation simplicity, then compare the results of using Gaussian distribution and Laplace distribution to model user's trace of position. That is, for 1-dimensional trace data, the predicted position of x coordinate x can be written in the form of Gaussian distribution

$$f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\Big(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\Big), \tag{3.4}$$

doi:10.6342/NTU202303932



(a) Prediction error distribution of x axis



(b) Prediction error distribution of z axis

Figure 10: Prediction error distribution of x and z axis

where μ is the mean and σ is the standard deviation of the distribution or Laplace distribution

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right), \tag{3.5}$$

where μ is the location parameter and b is the scale parameter with variance eqaul to $2b^2$. We assume that x and z coordinates are independent, so that, with regard to 2-dimensional distribution, it becomes bivariate Gaussian distribution and bivariate Laplace distribution. Parameters of probability distribution such as μ and σ for Gaussian distribution and μ and b for Laplace distribution can be derived from prior information about viewport prediction error in practice. Evaluations of results using Gaussian and Laplace probability models will be shown in Chapter 7.

3.3 Multiview Generation

The quality of 2D frames transcoded from volumetric video may suffer from the inaccuracy of viewport prediction. When there is a significant position deviation, it can result in high distortion between the pre-rendered views and the actual views, thereby impacting the overall quality. To eliminate the influence

doi:10.6342/NTU202303932

of viewport prediction error on quality of transcoded views, multiview transcoding is incorporated into the streaming system. That is, the multiview generation module in the edge server selects a number of reference positions based on the viewport prediction result as well as the viewport prediction error prior to time t. Transcoded views are then rendered from these reference positions. We refer to the selected position as reference positions since views generated from those positions are functioned as the reference views to virtual view synthesis module.

Moreover, as mentioned in the previous section, users are less prone to move in the direction of the y-axis and it is assumed that the viewport prediction error on rotation is rather less significant in this work. This assumption is based on the observation that the prediction error for roll, pitch, and yaw angles is generally less than 10° on average [3]. Any such error can be compensated by increasing the field of view angle for pre-rendered frames, resulting in a minor increase in resource consumption. In addition, numerous studies [30,32] have been introduced to deal with viewport prediction error on 3DoF data, which is also applicable to volumetric video streaming. Thus, we consider only the displacement on the x-z plane and generate additional views by shifting viewpoints from the predicted position in the direction of forward and backward (z-axis) and the direction of left and right (x-axis).

The ultimate objective of the multiview generation module is to design an efficient approach to choose the reference positions to optimize user's quality with the least number of required views. More specifically, based on the probabilistic model on user position and simulation results on the quality of transcoded views, the expected value of SSIM between actual viewport and the pre-rendered viewport should be optimized with the selected reference positions. Details including problem formulation, proposed algorithms, and result evaluation will be covered in Chapter 5, 6, and 7 respectively.

3.4 View Selection

According to the observations, view selection modules can decide which view is the best view among all the reference views. Instead of choosing the best view simply based on the distance between the actual position of users during playback and the reference position, the view selection module compares the quality of all the reference views based on the SSIM map for neighbor views and the SSIM map for synthesized views, which will be covered in Chapter 4. The best view is determined by comparing achievable SSIM values among the neighbor views and synthesized views. If the synthesized views can achieve a higher SSIM value, virtual view synthesis is applied accordingly. In this way, we can attain a higher

doi:10.6342/NTU202303932

level of similarity between the pre-rendered view and the actual view by integrating the benefits of neighbor views and synthesized views. Afterward, the best view is delivered to playback to promise the best similarity between the reference view and the actual view.

3.5 Virtual View Synthesis Model

The virtual view synthesis model on the client side helps with generating the final video frame closely matching the actual user's viewpoint just before playback time in an attempt to further boost the quality of transcoded frames. This process is only necessary when virtual views can reach a higher similarity with actual views. In this work, we implement a virtual view synthesis module with the DIBR technique mentioned in Chapter 2 as a baseline model. Texture image and depth map captured from the reference viewpoint and parameters of reference viewpoint and target viewpoint are fed into the module, then DIBR model outputs an image from target viewpoint through projection to 3D space and re-projection to target 2D plane, which is the plane the user is looking at during playback. Although our baseline model for virtual view synthesis does not address possible issues such as cracks, ghosts, disocclusion regions, and non-Lambertian surface, it is still valuable to analyze the effects and phenomena observed with the implemented DIBR model. Fig 11 illustrates the flow of performing virtual view synthesis.

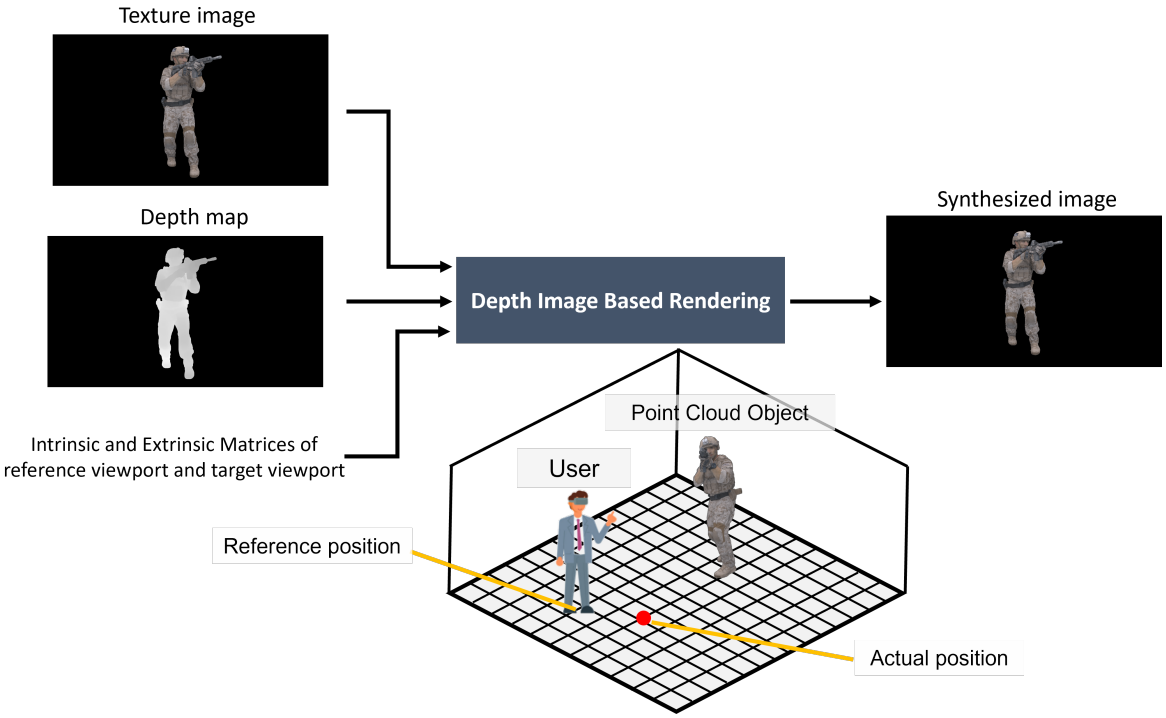


Figure 11: Illustration of virtual view synthesis model

doi:10.6342/NTU202303932

CHAPTER 4

OBSERVATION ON REALISTIC SIMULATION FOR TRANSCODING

The objective of applying transcoding with an edge server is to generate a set of 2D images based on viewport prediction results in advance, ensuring a seamless playback experience for the user. This approach aims to produce transcoded views that closely resemble the actual views observed by users during playback, ideally rendering an indistinguishable representation. However, there is an inherent deviation between the predicted position and the actual position of the user at the playback time. That is, we are using a view that is rendered from a neighboring position to represent the view that should be rendered from the actual position. In order to enhance the quality of transcoded views, virtual view synthesis is exploited in this work. To facilitate optimization on transcoded views, it is essential to inspect the relation between the quality of neighbor views as well as synthesized views and position deviation. In this chapter, we provide a detailed explanation of our simulation process. Additionally, we present our observations on the simulation results, which serve as the foundation for the design of the proposed multiview generation algorithm.

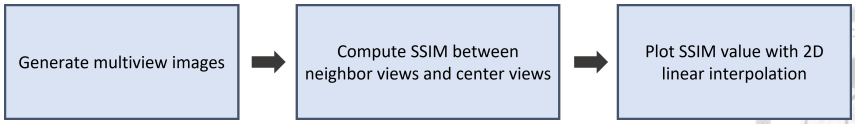
4.1 Quality Metric

In this work, we use structural similarity (SSIM) index [41] as the quality metric to evaluate the similarity between reference images, synthesized images, and actual images seen from the target position. SSIM index is a full reference metric, which is used to measure the quality of tested images based on distortion-free images as reference. The SSIM index between two signals x and y is defined as

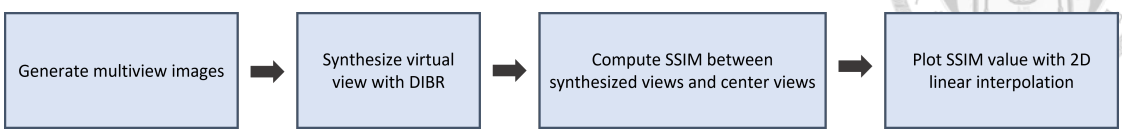
$$SSIM(x,y) = \frac{(2\mu_x\mu_y + (c_1L)^2)(2\sigma_{xy} + (c_2L)^2)}{(\mu_x^2 + \mu_y^2 + (c_1L)^2(\sigma_x^2 + \sigma_y^2 + (c_2L)^2)}, \tag{4.1}$$

where μ_x, μ_y are means of x and y, σ_x^2, σ_y^2 are variances of x and y, σ_{xy} is covariance of x and y, c_1, c_2 are adjustable constants, L is the dynamic range of pixel values. The SSIM value ranges from 0 to 1, where higher values indicate better image quality. Instead of measuring the pixel-by-pixel difference between a tested image and a reference image, the SSIM index identifies and compares the structural information. It operates by assuming that human perception is highly sensitive to

doi:10.6342/NTU202303932



(a) Generation of SSIM map for neighbor views



(b) Generation of SSIM map for synthesized views

Figure 12: Procedure of simulation

extracting structural information from a scene.

4.2 Simulation Flow

To gain a comprehensive understanding of the quality of transcoded images, we conduct a thorough simulation to reproduce the process carried out in the edge server in practice. In addition, to evaluate the benefit brought about by the virtual view synthesis technique, we compare the quality outcomes obtained when employing virtual view synthesis versus its absence. In this section, the overall procedures of simulation are explained in order of neighbor views and synthesized views, which are also summarized in Fig. 12 respectively. At the end of the simulation, an SSIM map is generated to facilitate the analysis of the relationship between quality and position deviation. As for the simulation environment, neighbor images and depth maps are rendered with PyVista library [1] by changing the translation information in the extrinsic matrix. Implementation of virtual view synthesis is developed based on source code provided with [42].

4.2.1 Generation of SSIM Map for Neighbor Views

To inspect the effect of position deviation on a similarity of images, we create an SSIM map that manifests the value of SSIM between a given center view and neighbor views rendered from neighboring positions. Throughout this study, we consider a coordinate system where the z-axis points forward and the y-axis points up. To begin, we select a coordinate on the x-z plane to be the center position and render the corresponding view as the center view. The center view servers as the target view for reference in computing the SSIM index throughout the process of creating the SSIM map. Subsequently, a number of neighbor views are generated by rendering views at positions that are shifted from the center view in four directions on the x-z plane with identical viewpoint angles. The center view and the neighbor views may also be considered as the views rendered at the predicted

doi:10.6342/NTU202303932

position and the neighboring position respectively. Also, we denote position deviation as the distance between the center position and the nearest neighboring position, which represents the spatial gap between the positions where center views and best neighbor views are rendered. Then SSIM values between each neighbor view and the center view are computed. Lastly, the SSIM map is constructed by mapping the corresponding SSIM value onto the sample positions and performing 2-dimensional linear interpolation to estimate SSIM values between the discrete data samples.

Fig. 13 illustrates the process of generating an SSIM map where the green point and orange point are the position of the volumetric object and the user's position respectively. The yellow box with a dashed line sketches the boundary of neighboring positions and the blue points are examples of neighboring positions where neighbor views are rendered. The red line represents the calculation of SSIM between the associated neighbor view and the center view. We use point cloud data provided by [43] as the volumetric object in our simulation. On the horizontal axis of the SSIM map, moving toward the right indicates a position that is more toward the right. On the vertical axis, moving upwards represents a position that is more toward the object. To clarify further, assuming the center position is at (3, 3) on the x-z plane, the SSIM of coordinates (4, 2) given in the SSIM map (i.e. Fig. 15(a)) denotes the SSIM computed with a center view and the neighbor view rendered from the position shifted from the center position 1m right and 1m backward to the object.

4.2.2 Generation of SSIM Map for Synthesized Views

To evaluate quality distribution when applying virtual view synthesis, we generate a counterpart version of the SSIM map demonstrating the relation between SSIM of synthesized views and position deviation. The procedure of creating the SSIM map for synthesized views is similar to that for neighbor views. Once the center position is selected and the center view is rendered, we proceed to generate multiple texture images by displacing the users' positions along the x and z directions. Additionally, depth maps are also rendered specifically for the virtual view synthesis process. Next, the DIBR model generates images projected to the plane viewed from the center location. These projected images are created using neighbor views, including texture images and depth maps viewed from shifted positions, along with camera parameters such as each position and the center location. Lastly, SSIM indices between the center view and each synthesized view are computed and sketched onto the corresponding position on the heap map. 2-dimensional linear interpolation is also leveraged to generate SSIM between the

doi:10.6342/NTU202303932

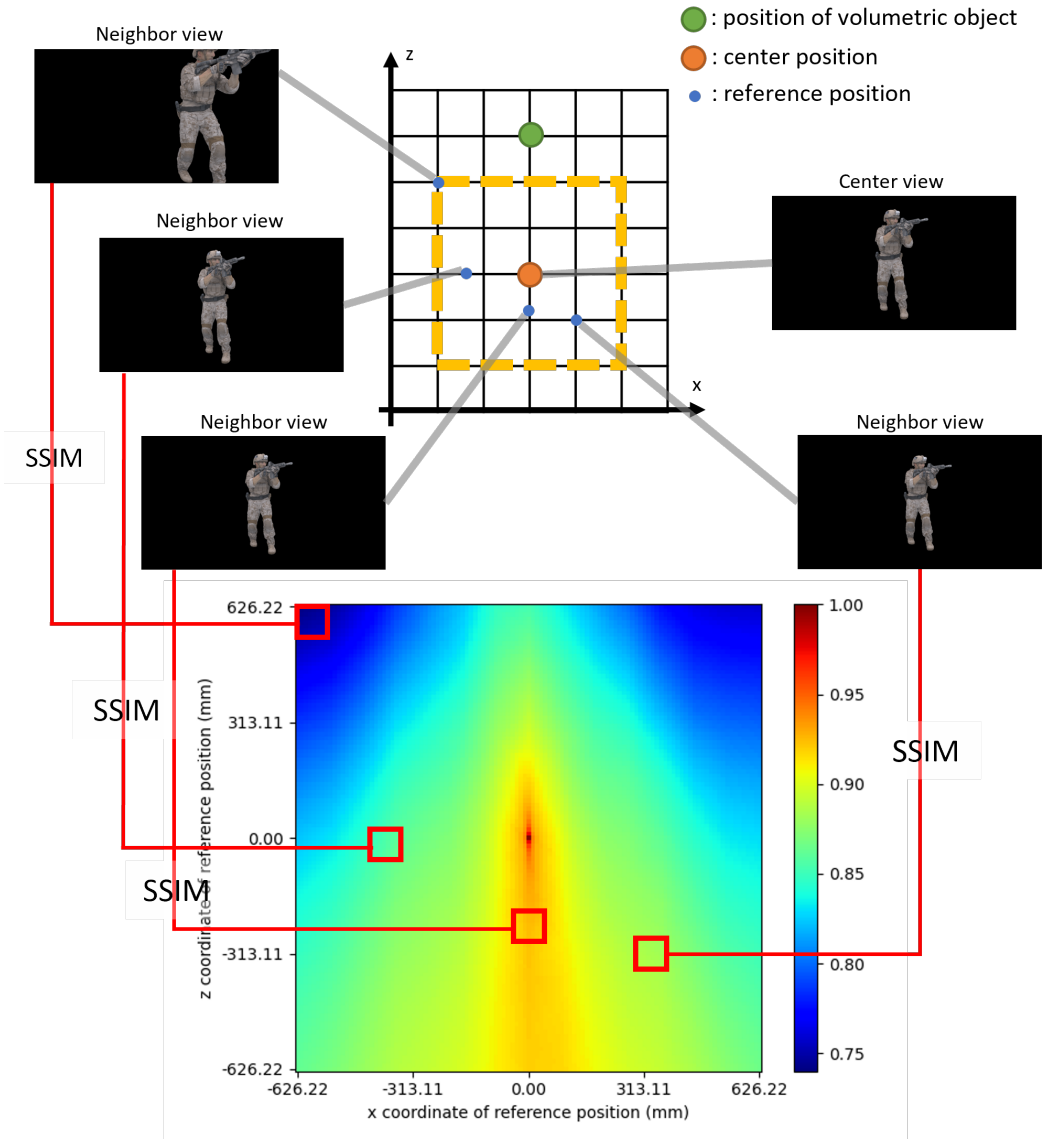


Figure 13: Process of generating SSIM map for neighbor views

doi:10.6342/NTU202303932

discrete data samples. The process of generating SSIM maps for neighbor views and synthesized views differs in a few key aspects. In the case of synthesized views, depth maps are rendered in addition to the texture images, and a DIBR model is utilized to generate the synthesized view based on the neighbor views. The general procedure for constructing the SSIM map for synthesized views is outlined in Fig. 14. The blue arrows in the diagram represent the DIBR process, which combines the texture image, depth map, and relevant camera parameters to synthesize the virtual view.

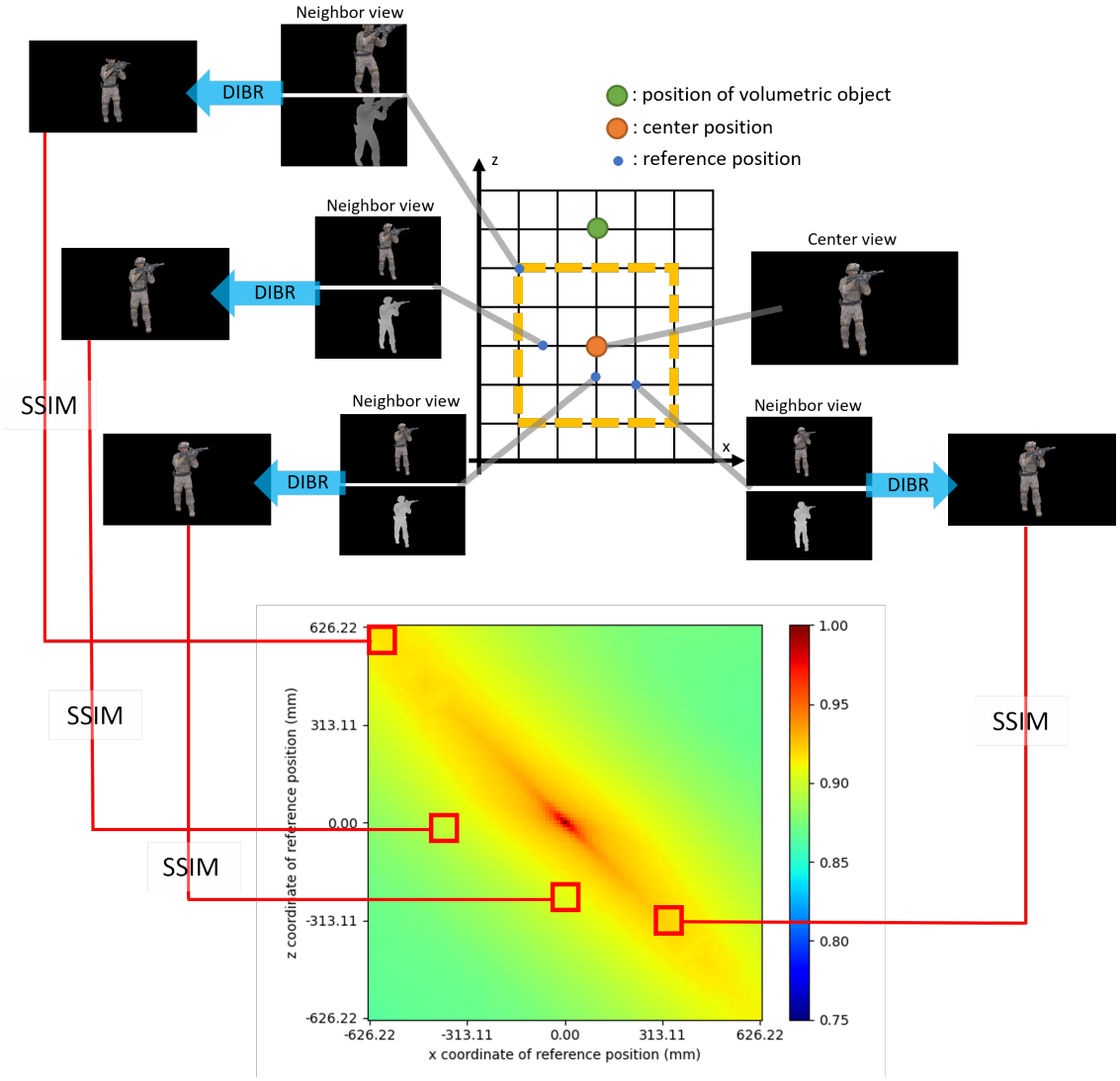
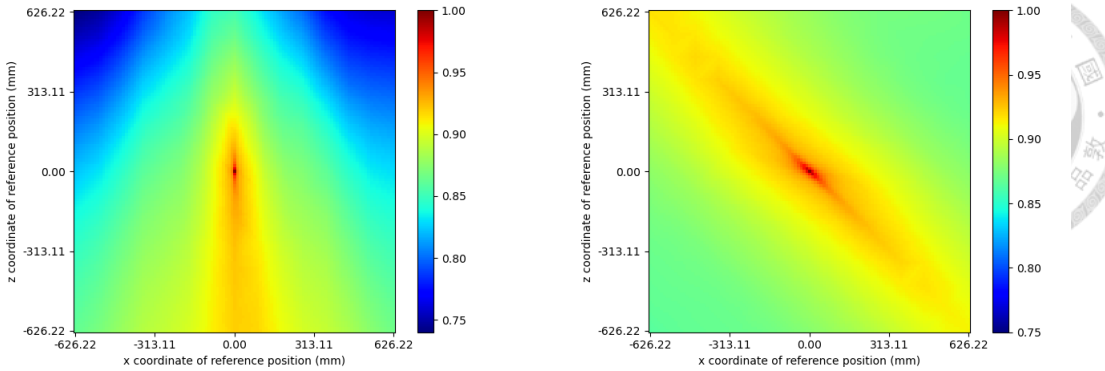


Figure 14: Process of generating SSIM map for synthesized views

4.3 Observation on SSIM Maps

In this section, we elaborate on the characteristics of both the SSIM map for neighbor views and the SSIM map for synthesized views. Intuitively, it may be

doi:10.6342/NTU202303932



(a) SSIM map for neighbor images

(b) SSIM map for synthesized image

Figure 15: SSIM distribution

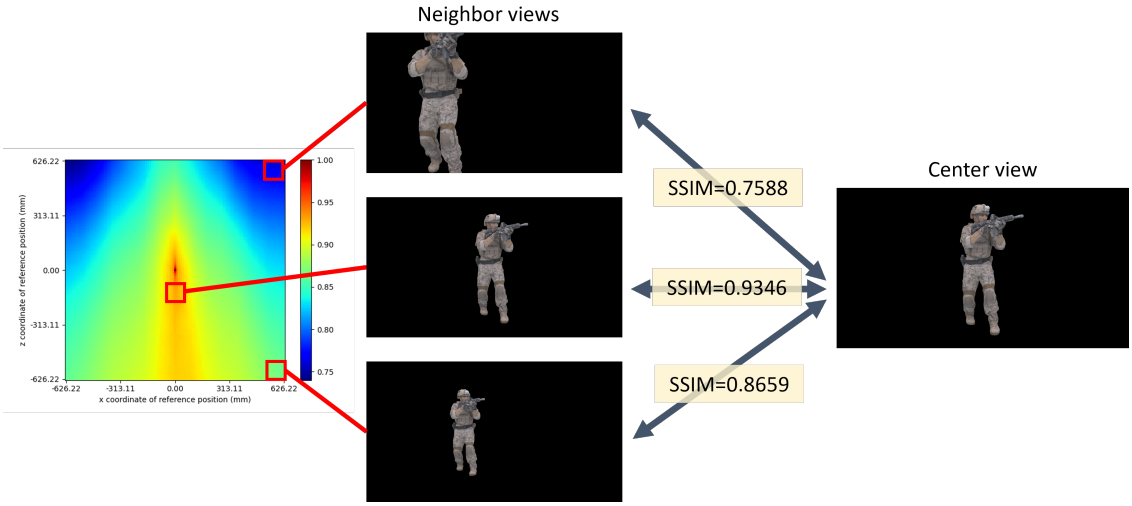


Figure 16: SSIM analysis of neighbor views

expected that a shorter distance between the center position and the neighbor position would bring about a higher SSIM value. Nonetheless, both of these maps reveal a unique pattern of high SSIM that is not solely determined by distance alone. Delving into the factors and attributes of the feature can provide valuable insights for designing the multiview generation approach which considers the optimization of empirical quality metrics. The two maps are presented side by side in Fig. 15 to make easy comparison and analysis.

4.3.1 Observation on SSIM Maps for Neighbor Views

It can be seen from Fig. 15(a) that for SSIM map with neighbor views, moving backward (z direction; vertical axis in the figure) has a minor influence on the quality of synthesized images while small sideways movements from the center viewpoint (x direction; horizontal axis in the figure) degrade much the quality

doi:10.6342/NTU202303932

of candidate views. The observed phenomenon can be attributed to the effect of moving backward, which leads to an increased viewing distance. As the viewing distance becomes greater, it becomes more challenging to perceive the details of distant objects. Consequently, the similarity between the distant view and the center view is higher, reflecting the reduced visibility of fine details at greater distances, as shown in the middle image in Fig. 16. Likewise, all neighbor views rendered at a considerable distance from the object exhibit a certain level of SSIM even with a large spatial displacement with an example displayed in the bottom image in Fig. 16. On the contrary, the dissimilarity (i.e., the blue region in the heatmap) observed in SSIM maps for neighbor views occurs when the user is in close proximity to the object. This is because even slight movements by the user near the volumetric object can result in a significant variation in the field of view. In addition, the object may be cropped in the field of view when one is too close to it. An example of cropped image is shown in the top image in Fig. 16, where the upper head of the soldier is trimmed out of view.

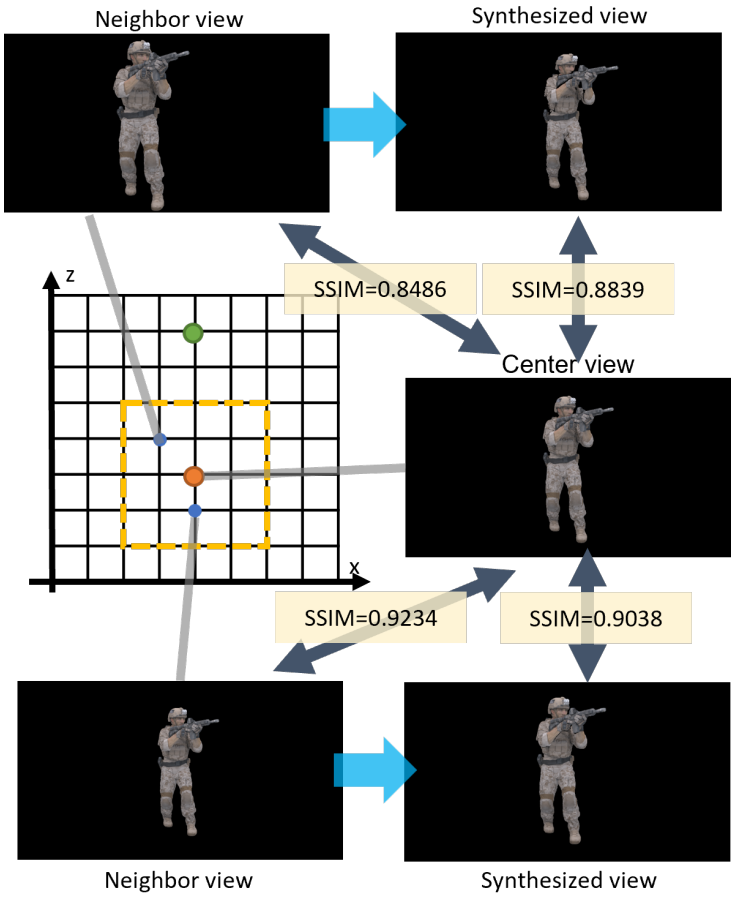


Figure 17: Illustration of neighbor view and synthesized view

doi:10.6342/NTU202303932

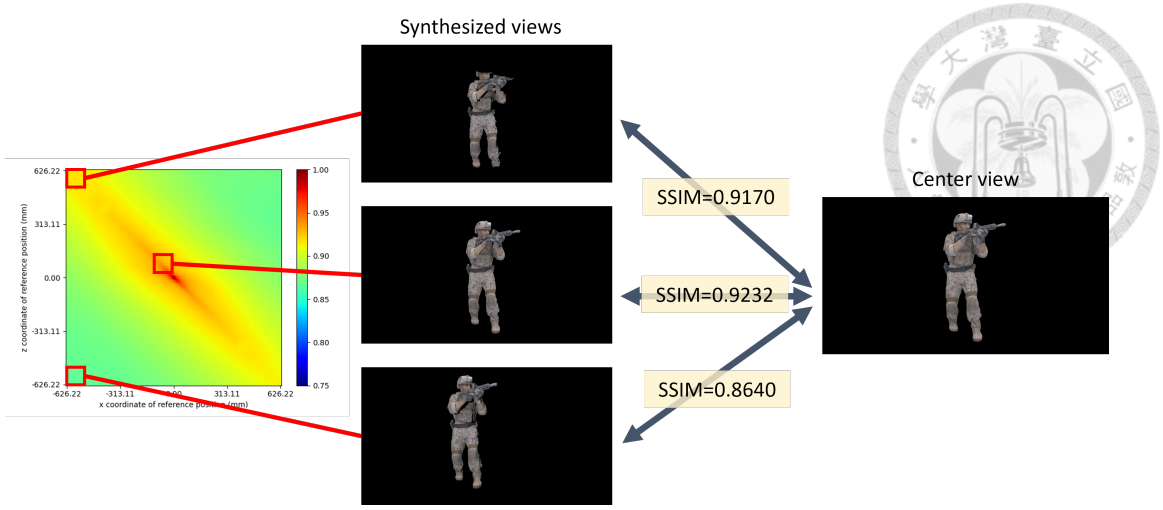


Figure 18: SSIM analysis of synthesized views

4.3.2 Observation on SSIM Maps for Synthesized Views

In contrast, the SSIM map with synthesized images reveals a distinct pattern where the highest SSIM values are maintained when moving forward and to the left. Implementing virtual view synthesis boosts the similarity of views from some neighboring positions. However, some synthesized views exhibit inferior SSIM values compared with neighbor views. As presented in the upper images in Fig. 17, when the neighbor position is located to the left and in front of the center position, the application of DIBR process brings about higher similarity with the center view. Nevertheless, when the neighbor position is directly behind the center position, utilizing virtual view synthesis unexpectedly reduces the SSIM value, which is demonstrated in the lower images of Fig. 17. This is due to the fact that applying virtual view synthesis may induce unexpected pixel-wise translation at specific viewpoints.

Upon careful examination of the attribute of the SSIM map for synthesized views, it can be pointed out that the distinctive pattern arises from the slight translation brought about by the virtual view synthesis process. Based on our observation, translation is less noticeable in synthesized views rendered with positions with direction from left-front to right-rear, hence leading to a higher SSIM value (i.e., the red region in Fig.15(b)). Furthermore, the problem that high dissimilarity emerges when the user is too close to the object is mitigated through virtual view synthesis. The top image in Fig. 18 reveals that exploitation of the DIBR technique results in a higher-quality view that exhibits greater similarity to the center view, despite the inability to recover the cropped portion. The lower image in Fig. 18 displays an example of a synthesized view with high translation.

doi:10.6342/NTU202303932

Despite the translation issue, the SSIM values of synthesized views generally show improvement compared to the SSIM map for neighbor views.

4.3.3 Observation on Different Center Views

Moreover, we perform a similar experiment using different center views to validate the reliability of SSIM maps for neighbor views and synthesized views. Fig. 19 demonstrates three instances of center views rendered from different viewpoints, along with their corresponding SSIM maps for neighbor views and synthesized views. It is evident that although the characteristics of both SSIM maps would be affected by different center views, the SSIM maps exhibit similar patterns as the SSIM maps depicted in Fig. 15. The SSIM maps for synthesized views all follow the shape where the strip along the direction from left-front to right-rear demonstrates higher SSIM value over other parts.

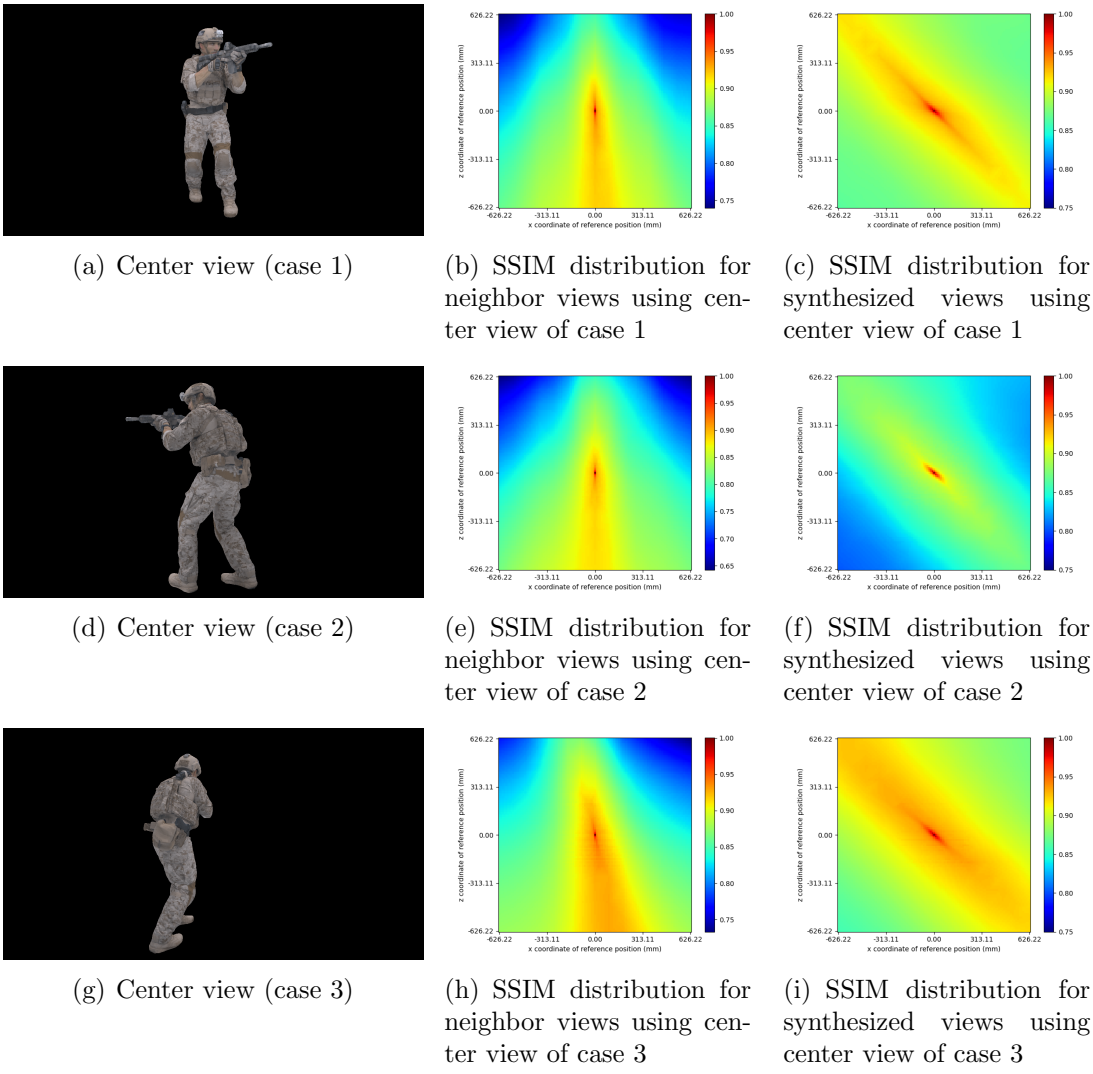
4.3.4 Observation on Different Distance to the Object

Furthermore, we examine the effects of utilizing center views rendered at varying distances between the user and the volumetric object, which is shown in Fig. 20. Left-hand side of Fig. 20 presents the center view with the distance between user and the object being 1.5, 2.5, 0.5 meters from the top to bottom. The decrease in SSIM is less pronounced when the center view is located at a greater distance, as the change of the viewpoint has a minimal impact on the perspective when the target object is far away. Also, it can be seen that SSIM maps generated with different center views follow a similar pattern, where areas with high SSIM (i.e. the red area) and low SSIM (i.e. the blue area) have comparable shapes in Fig. 20(c), 20(f), and 20(i). This observation recommends that optimization built upon one of the SSIM distributions may also be practiced on other SSIM distributions of different distances to the object.

4.4 Insight on SSIM Maps for Optimization

In previous sections, we have examined the attributes of SSIM distribution of synthesized views and neighbor views. It is shown that these two SSIM maps display distinct patterns and exhibit different advantages in terms of the similarity between pre-rendered views and actual views. Therefore, in order to make the most of the strengths offered by both kinds of views, it is logical to integrate the advantages of two SSIM maps. In particular, if the synthesized views have a higher SSIM value, the view selection module in the client side will pass the best reference view to the virtual view synthesis model; in opposition, if neighbor views exhibit better similarity, the view selection module sends the reference view directly to

doi:10.6342/NTU202303932



(a) Center view (case 1)

(b) SSIM distribution for neighbor views using center view of case 1

(c) SSIM distribution for synthesized views using center view of case 1

(d) Center view (case 2)

(e) SSIM distribution for neighbor views using center view of case 2

(f) SSIM distribution for synthesized views using center view of case 2

(g) Center view (case 3)

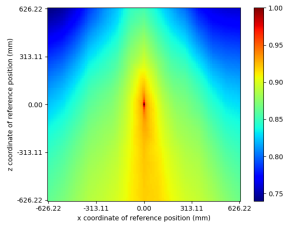
(h) SSIM distribution for neighbor views using center view of case 3

(i) SSIM distribution for synthesized views using center view of case 3

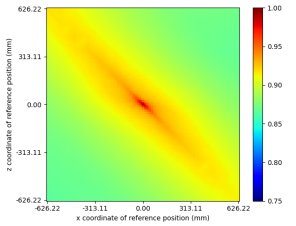
Figure 19: SSIM distribution of center views of different distance to the object

doi:10.6342/NTU202303932

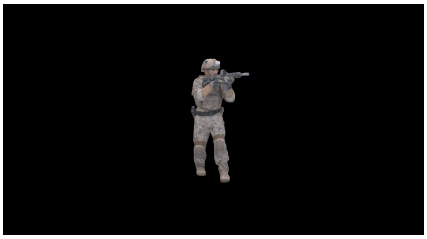
(a) Center view with distance 1.5 m between user and object



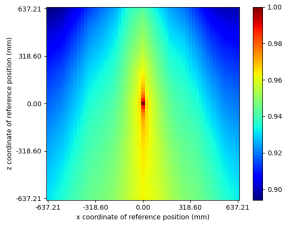
(b) SSIM map for neighbor views using center view with distance 1.5 m between user and object



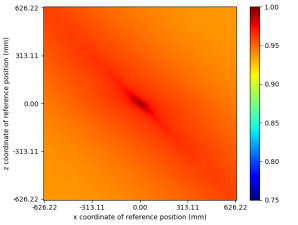
(c) SSIM distribution for synthesized views using center view with distance 1.5 m between user and object



(d) Center view with distance 2.5 m between user and object

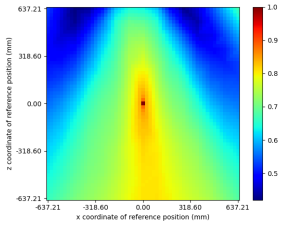


(e) SSIM map for neighbor views using center view with distance 2.5 m between user and object

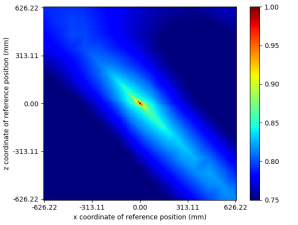


(f) SSIM distribution for synthesized views using center view with distance 2.5 m between user and object

(g) Center view with distance 0.5 m between user and object



(h) SSIM map for neighbor views using center view with distance 0.5 m between user and object



(i) SSIM distribution for synthesized views using center view with distance 0.5 m between user and object

Figure 20: SSIM distribution of center views of different distance to the object

doi:10.6342/NTU202303932

playback. In essence, this is identical to taking the point-wise maximum operation of the two SSIM maps, which is shown in Fig. 21(a). We will refer to Fig. 21(a) as the mixed SSIM map in the rest of the document. The mixed SSIM map can also be thought of as expanding the region of high SSIM values by implementing virtual view synthesis on the client side. Although it is mentioned that instances such as different center views and distance between the user and the object could influence the distribution of SSIM, here we use the version of the SSIM map shown in Fig. 15 to represent the target optimization function. This choice is made because extensive simulations could be performed offline to generate a database containing information on quality distribution. Furthermore, our proposed approach will be independent of the specific pattern of SSIM distribution.

To evaluate the loss of similarity between the actual view and transcoded view of the user, we employ structural dissimilarity (DSSIM) to represent the distortion of synthesized images or neighbor images caused by position deviation. In this work, value of DSSIM between two signals x and y is defined as

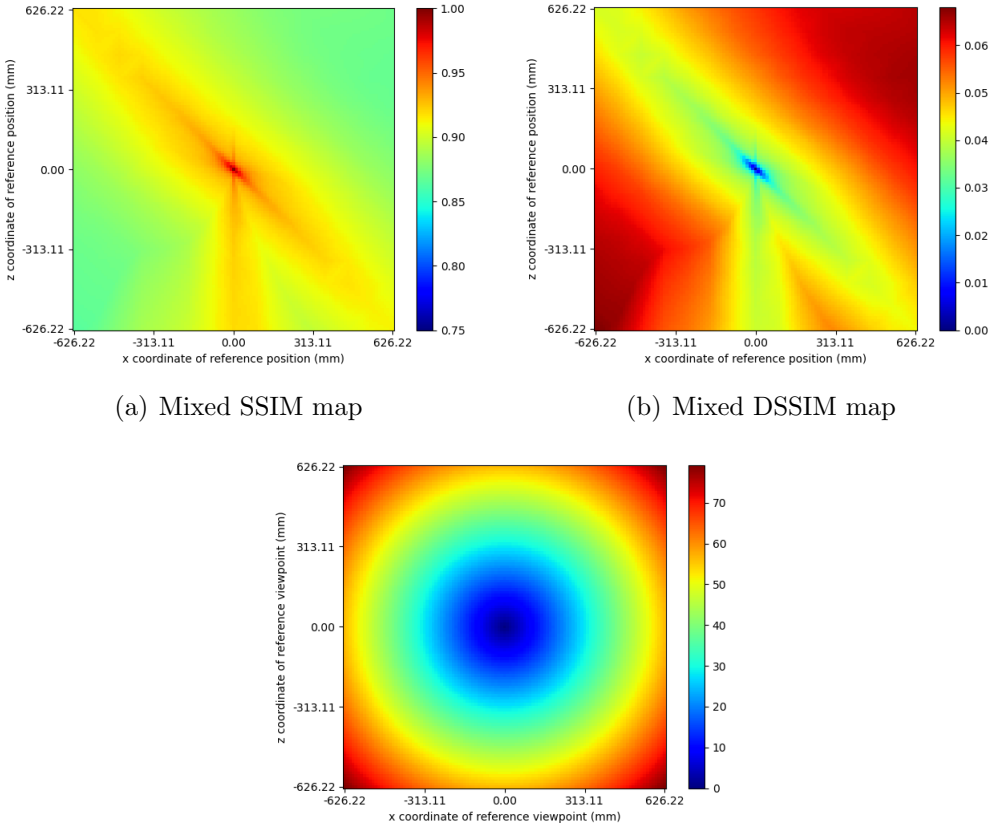
$$DSSIM(x, y) = \frac{1 - SSIM(x, y)}{2}. \tag{4.2}$$

DSSIM map is illustrated in Fig. 21(b), which is in opposite color compared to the SSIM map.

The distribution map of the L2 norm, the most common distortion function applied in general optimal quantization problems, is also shown in Fig. 21(c). It is evident that the distortion maps obtained through DSSIM and L2 norm display notable differences in several aspects. Firstly, the area of minor distortion (i.e. the blue part in the heatmap) in the distortion map of DSSIM is smaller, indicating that position deviation must be rather modest to achieve low distortion in regards to DSSIM. In contrast to the L2 norm where the value of distortion follows the shape of concentric circles, DSSIM in reality demonstrates a special shape, where the value is relatively low along a valley-shaped region. In addition, the escalation of distortion converges in the DSSIM map, whereas in the case of the L2 norm, the distortion keeps increasing with distance.

Based on the above observation, we are motivated by the distinctive attributions of DSSIM to revise the objective function used in problem formulation and modify the optimization algorithm for the purpose of minimizing the distortion of synthesized views in terms of DSSIM and thus enhance user's quality of experience. It is important to note that while this work considers SSIM as the quality metric, other evaluation standards including the self-defined QoE model can be applied as the objective function in the proposed optimization frame.

doi:10.6342/NTU202303932



(a) Mixed SSIM map

(b) Mixed DSSIM map

(c) L2 norm

Figure 21: Mixed SSIM map and DSSIM map

doi:10.6342/NTU202303932

CHAPTER 5

PROBLEM FORMULATION

In this chapter, we first introduce fundamental properties and general problem formulation of optimal quantization for probability distributions. Then we clarify that multiview generation problem considered in this work can be essentially viewed as an optimal quantization problem with certain adjustments. Finally, we formulate the multiview generation problem into three phases of the quantization sub-problem.

5.1 Optimal Quantization Problem

Quantization for probability distributions targets finding the best approximation of a d-dimensional probability distribution P by a discrete probability with a set of n points, so-called vector quantizers. Vector quantizers $a \in \alpha$ partition the whole space into Voronoi regions, which is defined as [44]

$$W(a|\alpha) = \{x \in \mathbb{R}^d : \|x - a\| = \min_{b \in \alpha} \|x - b\|\}, \tag{5.1}$$

and Voronoi diagram is referred to as

$$\{W(a|\alpha) : a \in \alpha\}. \tag{5.2}$$

A Voronoi region $W(a|\alpha)$ consists of all the points that are nearest to point a in the set α and a Voronoi diagram is formed by multiple Voronoi regions. Illustrations of Voronoi diagram in 1 and 2-dimension are shown in Fig. 22(a) and 22(b) respectively. Star-shaped dots in Fig. 22(a) and blues dots in Fig. 22(b) represent the vector quantizers; red dash lines in Fig. 22(a) and orange lines in Fig. 22(b) depicts the boundary of Voronoi regions.

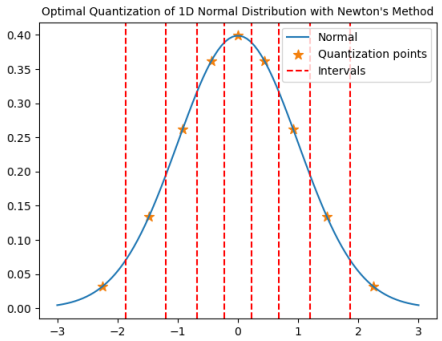
Generally, the quantization error is evaluated with the norm function. L_r norm of vector $x = (x_1, ..., x_k)$ is defined as

$$\|x\|_r = \big(\sum_{i=1}^{k} |x_i|^p\big)^{\frac{1}{p}}. \tag{5.3}$$

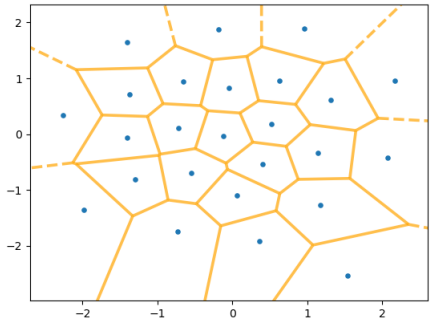
For $r = 2$, we get the Euclidean norm. The minimal quantization error of a given probability P and $n \in \mathbb{N}$ can be defined as

$$V_{n,r}(P) = \inf\{E\|X - q(X)\|^r : q : \mathbb{R}^d \to \mathbb{R}^d \quad \text{measurable}, |f(\mathbb{R}^d)| \leq n\}, \tag{5.4}$$

doi:10.6342/NTU202303932



(a) Voronoi diagram for 1-dimension



(b) Voronoi diagram for 2-dimension

Figure 22: Voronoi diagram

where $\|\cdot\|$ is the L_r norm on $\mathbb{R}^d$ and q is the quantization function attaining the inf, which is the objective for optimization in this problem. In other words, the target vector quantizers achieve minimal distortion in terms of expectation. The quantization problem as a whole can be conceptualized as the placement of n quantizers at appropriate positions in order to minimize the expected value of the distance between the target random variable and selected quantization points.

Overall quantization problem of finding n quantizers for d-dimensional random variable X with probability density function $f_X(x)$ can be formulated as [44, 45]

$$\begin{aligned} &\text{Find best } n\text{-tuple } \alpha^* = (a_1^*, a_2^*, ..., a_n^*) \\ &\text{s.t. } D_{n,r}(a_1^*, ..., a_n^*) \leq D_{n,r}(a_1, ..., a_n), \forall (a_1, ..., a_n) = \alpha \in (\mathbb{R}^d)^n, \end{aligned} \tag{5.5}$$

where $D_{n,r}(\alpha) : (\mathbb{R}^d)^n \to \mathbb{R}^+$, considering distortion of Euclidean norm, distortion function given set of quantization points, is defined as

$$D_{n,r}(\alpha) = \int_{\mathbb{R}^d} \min_{1\leq i\leq n} |a_i - \tau|^2 f_X(\tau) d\tau = \sum_{i=1}^{n} \int_{W(a_i|\alpha)} |a_i - \tau|^2 f_X(\tau) d\tau. \tag{5.6}$$

Above formulation states that no vector quantizer α performs better that the optimal one α^* with regard to the distortion function $D_{n,r}(\alpha)$.

Define the function of finding optimal vector quantizers as

$$\psi_{n,r} : (\mathbb{R}^d)^n \to \mathbb{R}_+, \quad \psi_{n,r}(a_1, ..., a_n) = E \min_{1\leq i\leq n} \|X - a_i\|^r. \tag{5.7}$$

A theorem for the existence of the optimal quantizers is stated as

Theorem 1. *(Existence, see [44]) We have $V_{n,r} < V_{n-1,r}$. The level set $\{\Psi_{n,r} \leq c\}$ is compact for every $0 \leq c < V_{n-1,r}$, hence the optimal set of centers exist and lies in a bounded set.*

doi:10.6342/NTU202303932

Let $\|h\|_p = \left(\int |h|^p d\lambda^d \right)^{\frac{1}{p}}$ for Borel measurable function $h : \mathbb{R}^d \to \mathbb{R}$ and $0 < p < \infty$. A theorem for asymptotic quantization error is developed as follows.

Theorem 2. *(Asymptotic quantization error; Zador's theorem, see [46] and [44]) Suppose* $\mathbb{E}\|X\|^{r+\delta}$ *for some* $\delta > 0$. *Let* P_a *be the absolutely continuous part of* P *and*

$$Q_r^d \triangleq \inf_{n \geq 1} n^{\frac{r}{d}} V_{n,r}\big(\mathcal{U}_{[0,1]^d}\big). \tag{5.8}$$

Then $Q_r^d > 0$ *and*

$$lim_{n\to\infty} n^{\frac{r}{d}} V_{n,r}(P) = Q_r^d \left\| \frac{dP_a}{d\lambda^d} \right\|_{\frac{d}{d+r}}. \tag{5.9}$$

With Theorem 1, it is ensured that n-optimal quantizers exist when using norm functions as quantization distortion functions. The optimal solution of quantization for 1-dimensional probability distribution can be solved straightforwardly with mathematical methods. However, followed by Theorem 2, for dimension $d \geq 2$, the target function of optimization problem becomes typically nonconvex, posing a challenge for its solution. In detail, The $n^{\frac{r}{d}}$ rate in equation (5.9) may be referred to as curse of dimensionality [47]. That is to say, the dimension of searching space may grow rapidly with the value of n of d which becomes too demanding to solve in practice. Therefore, sub-optimal methods that are easier to compute are proposed to find asymptotically reasonable solutions. Popular algorithms for solving high-dimensional quantization problems include the Lloyd-Max algorithm [48, 49] and competitive learning vector quantization [50].

5.2 Formulation of Multiview Generation Problem

In this work, we aim to minimize the distortion of transcoded 2D frames by finding the optimal set of reference viewpoints considering the viewport prediction error under the constraint of the number of total views. Given the probability model for user position as target distribution and mixed SSIM map as objective distortion function, the multiview generation problem is essentially an optimal quantization problem with a special design of probability distribution and distortion function. The selection of reference positions can be deemed as placing the quantization points at the proper location to minimize the distortion between the actual position of user during playback and the nearest reference position, as measured by DSSIM. With an examination of the distribution of viewport prediction error and quality distribution of synthesized images, we can uncover how the problem of reference position selection reflects the characteristics of the optimal quantization problem with specific modifications.

doi:10.6342/NTU202303932

To draw a comparison with the method proposed in Vues [3], we divide the multi-view generation problem into three sub-problems and introduce a corresponding algorithm to solve each sub-problem.

5.2.1 1-Dimensional Quantization with Uniform Interval

First of all, motivated by the method proposed in Vues [3], we choose the positions to create reference views by moving the prediction position toward four directions on the x-z plane with uniform step size s. In Vues, the authors set the step size s to a fixed value of 0.1 meters. The total number of views n is selected heuristically with the difference between prediction results given by three prediction models, which are chosen from a range from 1 to 7. As recommended in the motivation of this work, using preset values of s and n can substantially degrade the quality of user's view when the viewport prediction error is large. It also limits the opportunity to further enhance the similarity between the actual view and transcoded view.

As a result, we intend to identify an optimal value for step size s for the quantization problem given the number of views n in the x and z directions independently. This problem is sometimes referred to as the uniform quantization problem in certain manuscripts. With a given constant value of step size s, the coordinates of the quantization points in the x and z directions form an arithmetic sequence. Subsequently, the grid points of two 1-dimensional vector quantizers are selected as the final quantization points. Let $\alpha_x = (x_1, x_2, ..., x_L)$ and $\alpha_z = (z_1, z_2, ..., z_L)$ be the two set of 1-dimensional vector quantizers such that $x_k = x_1 + (k-1) \cdot s_x$ and $z_k = z_1 + (k-1) \cdot s_z$ where L is the number of views along one direction (i.e., there are $L^2 = n$ quantization points in total), s_x and s_z are the distances between two quantizers along x and z direction respectively. Let $f_{P_x}(p_x) : \mathbb{R} \to \mathbb{R}$ and $f_{P_z}(p_z) : \mathbb{R} \to \mathbb{R}$ be the probability density function representing user position's x and z coordinates. Given distortion function $g(y, \hat{y}) : (\mathbb{R}, \mathbb{R}) \to \mathbb{R}^+$, we can formulate the quantization problem with uniform interval as follows,

$$\begin{aligned} &\min_{x_1, s_x} D_L\big(x_1, x_2, ..., x_L\big) = \textstyle\int_{\mathbb{R}} \min_{1 \le i \le L} g\big(x_i, \tau\big) f_{P_x}(\tau) d\tau \\ &\text{s.t. } x_k = x_1 + (k-1) \cdot s_x, \end{aligned} \tag{5.10}$$

$$\begin{aligned} &\min_{z_1, s_z} D_L\big(z_1, z_2, ..., z_L\big) = \textstyle\int_{\mathbb{R}} \min_{1 \le i \le L} g\big(z_i, \tau\big) f_{P_z}(\tau) d\tau \\ &\text{s.t. } z_k = z_1 + (k-1) \cdot s_z. \end{aligned} \tag{5.11}$$

There are only 4 parameters to be optimized if we assumed that distances between every two of the quantizers are identical. In this way, the solution can be attained in a short period of time while the performance is doomed to be inferior to other optimal approaches.

doi:10.6342/NTU202303932

5.2.2 Independent Optimal 1-Dimensional Quantization

In general, quantization with uniform intervals does not give the most effective solution for a specific probability distribution. When considering a fixed number of quantization points, nonuniform spacing of quantizers can yield lower quantization distortion and less sensitivity to variations in input statistics. Therefore, we enlarge the solution space in an attempt to search for optimal quantization points in the two directions on the x-z plane independently. Similarly, we generate final 2D quantization points by creating the grid points with two sets of 1-dimensional vector quantizers. Using the same notation as mentioned in the previous subsection, we can formulate an independent optimal 1-dimensional quantization problem as

$$\min_{\alpha_x=(x_1,x_2,...,x_L)} D_L\big(x_1,x_2,...,x_L\big) = \int_{\mathbb{R}} \min_{1\le i\le L} g\big(x_i,\tau\big) f_{P_x}(\tau)d\tau, \tag{5.12}$$

$$\min_{\alpha_z=(z_1,z_2,...,z_L)} D_L\big(z_1,z_2,...,z_L\big) = \int_{\mathbb{R}} \min_{1\le i\le L} g\big(z_i,\tau\big) f_{P_z}(\tau)d\tau. \tag{5.13}$$

It may be noticed that (5.12) and (5.13) are essentially (5.10) and (5.11) without the constraint of uniform step size. By removing the constraint, smaller distortion is achievable while optimization on $2L$ parameters is required.

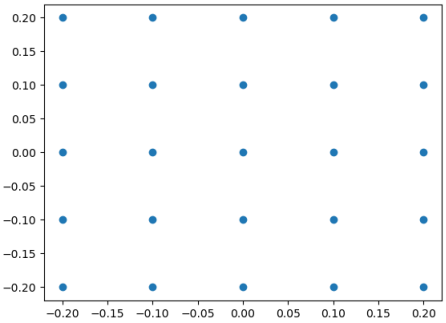
5.2.3 2-Dimensional Quantization

Last but not least, we formulate and solve directly the 2-dimensional quantization problem. Given probability density function of user's actual viewpoint $f_P(p) : \mathbb{R}^2 \to \mathbb{R}^2$ and quality distortion function $g(p,\hat{p}) : (\mathbb{R}^2,\mathbb{R}^2) \to \mathbb{R}^+$, the optimization problem for reference view selection can be written in form of optimal quantization problem,

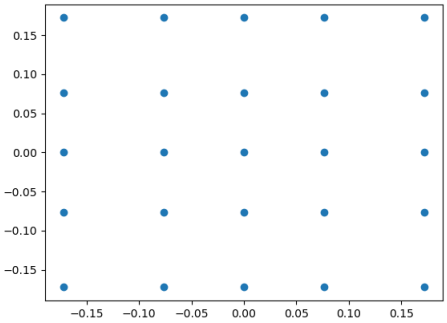
$$\begin{aligned} &\min_{\alpha=(x_1,z_1),(x_2,z_2),...,(x_n,z_n)} D_n(\alpha) \\ &\text{where } D_n(\alpha) = \int_{\mathbb{R}^2} \min_{1\le i\le n} g\big((x_i,z_i),\tau\big) f_P(\tau)d\tau \end{aligned} \tag{5.14}$$

The solution to 2-dimensional quantization problems takes into consideration the relation between position distribution between x and z direction and dissimilarity caused by viewport drift with the cost of extensive searching space. However, this approach comes at the expense of a large searching space, and it is important to recall that Theorem 2 highlights the challenges in achieving optimality for dimensions higher than two ($d > 2$). One may notice that the only difference between general optimal quantization and the proposed reference view selection problem is that the distortion function considered here is the SSIM loss rather than the norm function and that target probability distribution is designed to follow the pattern of viewport prediction error. Due to the intricate pattern of

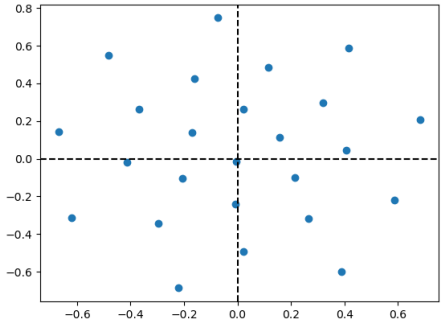
doi:10.6342/NTU202303932



(a) 1D quantization with uniform interval



(b) Optimal 1D quantization



(c) 2D quantization

Figure 23: Illustration of quantization points of different methods

SSIM distribution, it is necessary to conceive a specific algorithm to maintain high quality in terms of SSIM with a limited number of views.

Fig. 23 demonstrates the quantization points obtained with different methods. Quantizers in the x and z direction in 23(b) are acquired with Gaussian probability distribution with 0 mean and standard deviation 0.1 (i.e. $\mathcal{N}(0, 0.1)$). Quantizers in 23(c) are generated with bivariate Gaussian probability distribution with 0 means in both directions, standard deviation $\sigma_x = \sigma_z = 0.1$, and covariance $cov(x, z) = 0$. Three types of optimization problems and corresponding optimized variables are summarized in Table. 1.

Optimization problems	Optimized variables
1D quantization with uniform interval	x_1, s_x, z_1, s_z
1D optimal quantization	$(x_1, ..., x_L), (z_1, ..., x_L)$
2D quantization	$\big((x_1, z_1), ..., (x_n, z_n)\big)$

Table 1: Proposed optimization problems and corresponding optimization variables

doi:10.6342/NTU202303932

CHAPTER 6

OPTIMAL MULTIVIEW GENERATION ALGORITHMS

Algorithms for vector quantization have been developed for decades. To effectively address multiview generation to appropriately select the reference positions, we first explore the user of two independent quantizers for 1-dimensional distribution. In this work, we apply Newton's method to find optimal solutions of quantizers for 1-dimensional distribution. Subsequently, competitive learning vector quantization (CLVQ) algorithm, also known as the Kohonen algorithm, is leveraged to further boost the expected value of viewport similarity. This approach is chosen to ensure optimal adaptation to the reference position selection problem. Finally, a concept borrowed from simulation-based optimization is incorporated into the CLVQ process to attain optimality on SSIM simulation results.

6.1 1D Quantization Method

To simplify the process of achieving an optimal solution, we first determine the quantization points independently for the x and z directions. Then we generate the final reference positions by taking the grid points from these two sets of 1-dimensional quantization points. In [45], authors describe in detail the steps to solve the 1D quantization problem for Gaussian distribution and L2 norm distortion function with Newton's method. To summarize, the target is to find the vector quantizers obtaining the minimum of expected L2 distortion. Due to the convexity of the L2 norm, the goal is identical to finding the zero of derivative of the distortion function. With the derivative and second derivative of the distortion function in hand, we can reach the optimal quantizers by implementing Newton's method iteratively, which can be expressed as

$$\alpha^{n+1} = \alpha^n - \Big(\mathbf{H}(D_n(\alpha))\Big)^{-1} \cdot \nabla D_n(\alpha), \tag{6.1}$$

where $\mathbf{H}$ and ∇ represents Hessian matrix and gradient respectively; α^n is the vector quantizers in the nth iteration.

In particular, analytical forms of distortion function concerning 1-dimensional standard Normal distribution and Laplace distribution with L2 norm distortion function are derived. Let α be the set of vector quantizers and $\alpha = (a_1, ..., a_n), a_1 < a_2 < ... < a_n$. Set $a_{j+\frac{1}{2}} = \frac{a_j + a_{j+1}}{2}$ and $a_{j-\frac{1}{2}} := \frac{a_j + a_{j-1}}{2}, j = 2, ...n-1$, $a_{\frac{1}{2}} =$

doi:10.6342/NTU202303932

$-\infty, a_{n+\frac{1}{2}} = \infty$. Denote CDF function of Gaussian distribution as $\Phi(\gamma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma} \exp(-\frac{\tau^2}{2}) d\tau$. The distortion of vector quantizers α for $\mathcal{N}(0,1)$ is shown to be computed as [45]

$$\begin{aligned} D_n(\alpha) &= \sum_{j=1}^{n} \int_{a_{j-\frac{1}{2}}}^{a_{j+\frac{1}{2}}} (a_j - \tau)^2 \exp(-\frac{\tau^2}{2}) \frac{d\tau}{\sqrt{2\pi}} \\ &= \sum_{j=1}^{n} \Bigg((1 + a_j^2) \Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}}) \Big) \\ &- \frac{1}{\sqrt{2\pi}} \Big(a_{j+\frac{1}{2}} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - a_{j-\frac{1}{2}} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) \Big) \\ &+ \frac{2}{\sqrt{2\pi}} a_j \Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) \Big) \Bigg). \end{aligned} \tag{6.2}$$

6.1.1 1D Quantization with Uniform Interval

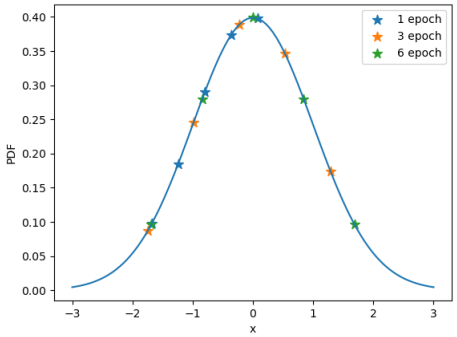
To manifest the importance of choosing adequate reference positions even with uniform intervals, we first introduce the method to choose the optimal distance between selected positions to compare with the method proposed in [3]. In the 1-dimensional quantization problem with uniform interval, which is also introduced in chapter 4.4.1, we assume that the distance between each two quantizers is constant. Let $\alpha_x = (x_1, x_2, ..., x_n)$ be the vector quantizers in x direction such that $x_k = x_1 + (k-1) \cdot s_x$ where s_x is the interval between two quantizers. In this manner, optimization terms remain only x_1 and s_x. The repeating process of Newton's method becomes

$$\begin{bmatrix} x_1^{k+1} \\ s_x^{k+1} \end{bmatrix} = \begin{bmatrix} x_1^k \\ s_x^k \end{bmatrix} - \begin{bmatrix} \frac{\partial^2 D_n(\alpha)}{(\partial x_1^k)^2} & \frac{\partial^2 D_n(\alpha)}{\partial x_1^k \partial s_x^k} \\ \frac{\partial^2 D_n(\alpha)}{\partial s_x^k \partial x_1^k} & \frac{\partial^2 D_n(\alpha)}{(\partial s_x^k)^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial D_n(\alpha)}{\partial x_1^k} \\ \frac{\partial D_n(\alpha)}{\partial s_x^k} \end{bmatrix}, \tag{6.3}$$

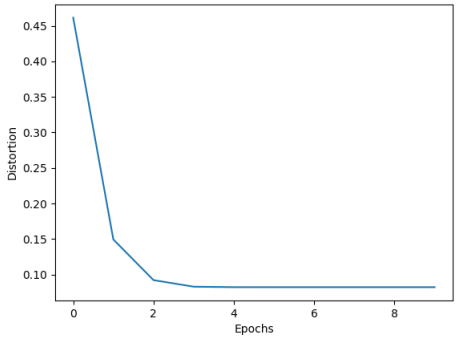
where -1 in the exponential term means the inverse of a matrix.

Executing process of Newton's method iteratively yields the optimized quantization points with a uniform interval in the x direction by moving toward the zero of the first derivative of the distortion function. The overall procedure is shown in Algorithm 1. For computation simplicity, we first calculate the vector quantizers α for standard Normal distribution (i.e. $\mathcal{N}(0,1)$) and then get the quantizers of specific Normal distribution $\mathcal{N}(\mu, \sigma)$ by shifting with formula $\alpha\sigma + \mu$. To effectively adapt to the viewport prediction error of each user, we compute σ based on the viewport prediction error in the history trace. To clarify, for quantization at time t, we use the L1 norm and L2 norm of viewport prediction error in $(0, t)$ to represent for a standard deviation of target probability distribution. Computations for derivatives of above formula concerning 1-dimensional standard Normal

doi:10.6342/NTU202303932



(a) Optimization steps of 1D quantizers with uniform distance



(b) Expected L2-norm distortion on each epoch of 1D quantizers with uniform distance

Figure 24: Results of 1D quantization with uniform interval

distribution and L2 norm distortion function are shown in Appendix B. It is clear that vector quantizers along the z direction can be solved with similar operations.

Fig. 24 presents example results of 1-dimensional quantization with a uniform interval for standard Gaussian probability distribution. Vector quantizers in different epochs of optimization over iteration of Newton's method are shown in Fig. 24(a). The value of expected distortion computed in each epoch is plotted in Fig. 24(b).

Algorithm 1 1D Quantization with Uniform Interval

Input: $\mathbb{P}$: 1-dimensional Gaussian probability distribution $\mathcal{N}(\mu, \sigma)$, ϵ_n: distortion criteria, ν: limit of epochs

Initialization: Set $\alpha_x^0 = (x_1^0, x_2^0, ..., x_L^0)$ where $x_1^0 = -2 + \frac{2}{L}, s_x^0 = \frac{2}{L}, x_j^0 = x_1^0 + (j-1)s_x, 1 \leq j \leq L$

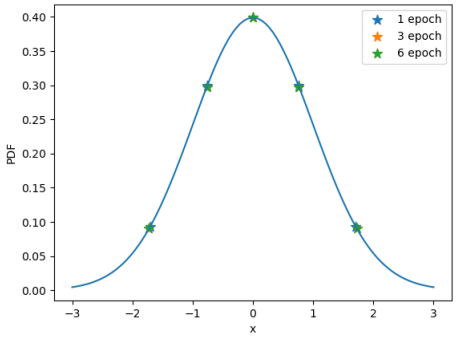
1: **repeat**
2: Update x_1^{k+1} and s_x^{k+1} with equation (6.3)
3: Compute distortion D_L with equation (5.10)
4: **until** $D_n \leq \epsilon_n$ is met or the number of epochs exceeds the limit

Output: $\sigma\alpha_x + \mu$: set of 1-dimensional vector quantizers

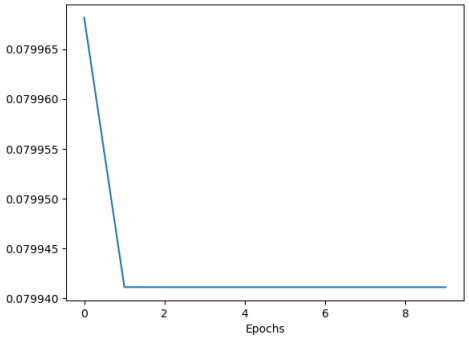
6.1.2 Independent 1D Optimal Quantization Method

Next, we aim to explore 1-dimensional optimal vector quantizers without the constraint of uniform step size so as to further reduce expected distortion. Optimal quantization points for 1-dimensional distribution can be derived by applying a similar approach. The explicit formula used in Newton's method can be written

doi:10.6342/NTU202303932



(a) Optimization steps of optimal 1D quantizers



(b) Expected L2-norm distortion on each epoch of optimal 1D quantizers

Figure 25: Results of optimal 1D quantization

as

$$\begin{bmatrix} a_1^{k+1} \\ \vdots \\ a_n^{k+1} \end{bmatrix} = \begin{bmatrix} a_1^{k} \\ \vdots \\ a_n^{k} \end{bmatrix} - \begin{bmatrix} \frac{\partial^2 D_n(\alpha)}{(\partial a_1^k)^2} & \cdots & \frac{\partial^2 D_n(\alpha)}{\partial a_1^k \partial a_n^k} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 D_n(\alpha)}{\partial a_n^k \partial a_1^k} & \cdots & \frac{\partial^2 D_n(\alpha)}{(\partial a_n^k)^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial D_n(\alpha)}{\partial a_1^k} \\ \vdots \\ \frac{\partial D_n(\alpha)}{\partial a_n^k} \end{bmatrix} \tag{6.4}$$

The overall procedure is summarized in Algorithm 2, which is similar to Algorithm 1 except for the update equation. With some calculation, the analytical form of the derivative and second derivative of distortion function can also be determined for 1D Gaussian distribution with L2 norm distortion function, which is provided in [45] and also shown in Appendix C.

Fig. 25 shows example results of optimal 1D quantization on standard Gaussian distribution (i.e. $\mathcal{N}(0,1)$). Given that the probability is higher around the mean, the points closer to the mean are more concentrated. The spacing between points with higher probability is reduced and larger spacing is allocated to points with lower occurence. Moreover, it can be observed that the process of Newton's method on 1-dimensional quantization converges rapidly, where the expected value of distortion reaches stability in 3 epochs whether or not considering the constraint of uniform interval. Since Newton's method updates all the variables in a single iteration, the number of variables to be optimized does not affect the speed of convergence. In addition, with the same number of quantizers and the same target probability distribution, optimal 1D quantization gives a lower expected distortion value compared to 1D quantization with uniform intervals.

doi:10.6342/NTU202303932

Algorithm 2 1D Optimal Quantization

Input: $\mathbb{P}$: 1-dimensional Gaussian probability distribution $\mathcal{N}(\mu, \sigma)$, ϵ_n: distortion criteria, ν: limit of epochs

Initialization: Set $\alpha_x^0 = (x_1^0, x_2^0, ..., x_L^0)$ where $x_j^0 = -2 + \frac{2(2j-1)}{L}, 1 \leq j \leq L$

1: **repeat**
2: Update $\alpha_x^{k+1} = (x_1^{k+1}, ..., x_L^{k+1})$ with equation (6.4)
3: Compute distortion D_L with equation (5.12)
4: **until** $D_n \leq \epsilon_n$ is met or the number of epochs exceeds the limit

Output: $\sigma\alpha_x + \mu$: set of 1-dimensional vector quantizers

6.2 2D Quantization Method

In the edge-assisted volumetric video streaming system introduced in this work, the multiview generation module makes a decision on choosing appropriate positions in $\mathbb{R}^2$ space to best represent the distribution of user's future position concerning the similarity between synthesized view and the actual view seen at the actual viewpoint. This process resembles the optimal quantization problem on a 2-dimensional plane. Although vector quantizers for 1D Gaussian distribution with L2 norm distortion function can be obtained directly with Newton's method, improvement of user-perceived quality requires further innovation to completely fit into the distribution of user's position and similarity loss. Therefore, it is necessary to develop an efficient approach to quantization for a 2-dimensional probability distribution. We first introduce the well-known stochastic approach, competitive learning vector quantization (CLVQ), for solving high dimensional quantization problems, then propose a new version of CLVQ which embodies the concept of simulation-based optimization to resolve the intricate pattern of SSIM distribution.

6.2.1 Competitive Learning Vector Quantization (CLVQ)

Competitive learning vector quantization procedure [51], which is proposed in the 1980s, emerges as a stochastic gradient method in machine learning nowadays. In this work, we apply the CLVQ algorithm to deal with vector quantization for 2-dimension distribution, where optimal quantization points are difficult to solve. The essence of the CLVQ algorithm is to update the vector quantizers stochastically by updating a selected quantizer at one time. The key formula used in CLVQ to update the quantizers is

$$\alpha_{k+1} = \alpha_k - \gamma_k \nabla g(\alpha_k), \tag{6.5}$$

where α_k is the quantization points at kth update, γ_k is the gain parameter for kth update, and $g(\cdot)$ is the loss function. When considering Euclidean distortion,

doi:10.6342/NTU202303932

the update function for the winning point becomes

$$a^*_{k+1} = a^*_k + \gamma_k \cdot (y - a^*_k). \tag{6.6}$$

The selection of gain parameter γ holds great significance and requires careful consideration. Considering the convergence of stochastic gradient descent, it is required that the value of γ satisfies following constraints,

$$\sum_{k\geq 1} \gamma_k = +\infty, \text{ and } \sum_{k\geq 1} \gamma_k^2 < +\infty. \tag{6.7}$$

Here for CVLQ process, we follow [45] to set the gain parameter for kth update to

$$\gamma_k = \gamma_0 \frac{a}{a + \gamma_0 bk}, a = 4k^{\frac{1}{d}}, b = \pi^2 k^{-\frac{2}{d}}, \tag{6.8}$$

for d-dimensional Gaussian distribution. The choice borrows the step size from a uniform distribution that is proved to satisfy the condition of convergence by the Central Limit Theorem.

The overall procedure of the CLVQ algorithm is presented in Algorithm 3. The third line in Algorithm 3 is called the competitive phase that selects the best point (i.e. the quantizer closest to the sample point) and the fourth line is the learning phase that updates the winning quantization point by moving toward the sample point. Fig. 26 illustrates the operation in the learning phase, where the winning point (light blue point) moves toward the sample point (orange point) with a given step size. By iteratively applying the procedure of competing and learning with decreasing gain parameters, the vector quantizers converge to a final estimation of the optimized, albeit not optimal, quantization set.

Algorithm 3 Competitive Learning Vector Quantization (CLVQ)

Input: $\mathbb{P}$: probability distribution, n: number of quantization points, ν: limit of epochs, γ_0: initial gain parameter

Initialization: Set $\alpha = \{a_1, a_2, ..., a_n\}$ to be a set of n quantizers by sampling n points from distribution $\mathbb{P}$

1: **for** $k \in \{1, 2, ..., \nu\}$ **do**
2: Take a sample y from $\mathbb{P}$
3: Find quantizer a_i that is closest to y
4: Compute gain parameter γ_k with equation (6.8)
5: Update a_i with equation $a_i \leftarrow a_i + \gamma_k \cdot (y - a_i)$
6: **end for**

Output: α: set of optimized vector quantizers

Fig. 27 demonstrates the results of 25 quantizers and transition of expected distortion of running CLVQ 10000 epochs for bivariate Gaussian distribution with both mean equal to 0 and variance equal to 1 and L2 norm distortion function.

doi:10.6342/NTU202303932

Quantization points resemble the shape of concentric circles with the origin as centers. It may be seen again that due to the higher probability distribution around the mean, there is a greater concentration of points in proximity to the mean. Fig. 28 shows the counterpart figures for bivariate Laplace distribution with both mean equal to 0 and variance equal to 1. Under identical values of mean and variance, quantization points are more concentrated around the center and more dispersed on the margin for Laplace distribution; while quantization points are more average distributed for Gaussian distribution.

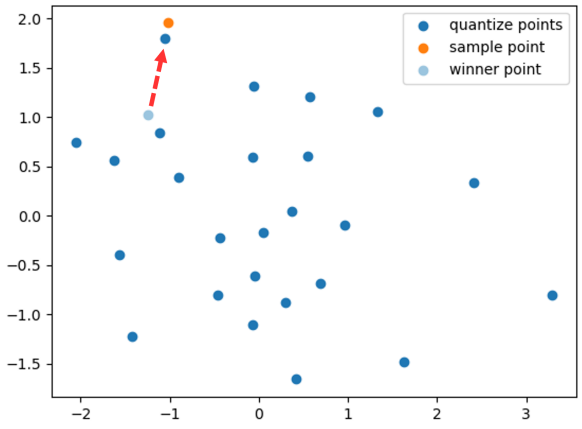
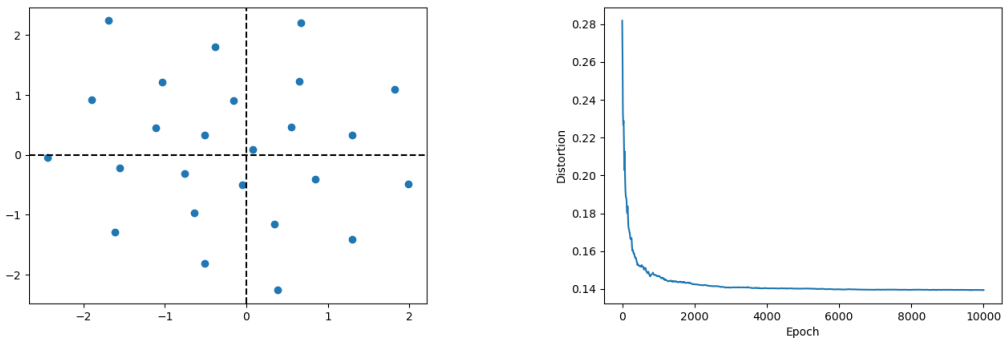


Figure 26: Illustration of learning phase in CLVQ



(a) Optimization result of quantizers on 2D Gaussian distribution

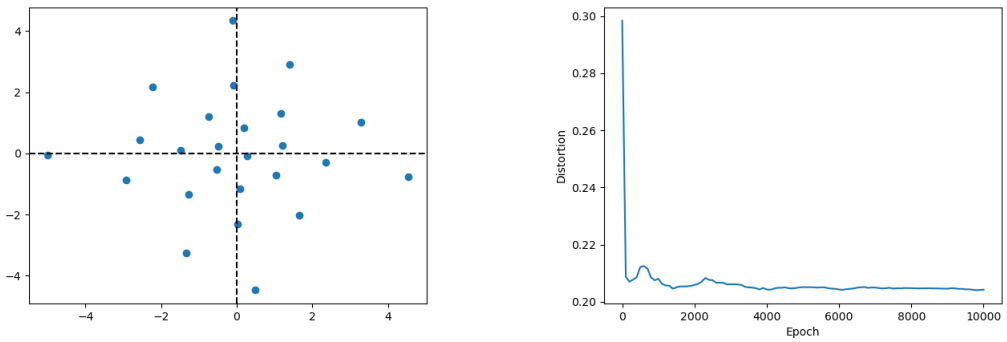
(b) Expected L2-norm distortion of quantizers on 2D Gaussian distribution

Figure 27: Results of quantization on 2D Gaussian distribution

6.2.2 Gradient-Free Competitive Learning Vector Quantization

Although classical quantization solutions mentioned above provide an effective algorithm to find the vector quantizers, it should be emphasized that established

doi:10.6342/NTU202303932



(a) Optimization result of quantizers on 2D Laplace distribution

(b) Expected L2-norm distortion of quantizers on 2D Laplace distribution

Figure 28: Results of 2D quantization on 2D Laplace distribution

quantization algorithms are built upon Euclidean distance distortion function. In the multiview generation problem considered in our system, the goal is to maximize the expected SSIM value (or minimize expected DSSIM) with a given number of views. Based on our analysis of the SSIM map in previous chapters, it is apparent that Euclidean distance and SSIM map exhibit distinctive characteristics. Moreover, different virtual view synthesis techniques applied on the client side can result in diverse patterns of SSIM and there would be other quality metrics considered in the future. Therefore, it is necessary to develop a framework that takes into account the diverse patterns observed in the quality map.

We draw inspiration from solution solving simulation-based optimization [52], where only zeroth-order information about the objective function $g(\cdot)$ rather than its gradient is available. Also, it remains unknown whether the objective function $g(\cdot)$ is convex or not. Our problem settings inherently belong to the domain of simulation-based optimization as a closed-form function expressing the SSIM map is not accessible. Hence, it is impossible for us to obtain the first-order gradient of the objective function in terms of DSSIM value.

Accordingly, we have made appropriate revisions to the original version of the CLVQ algorithm in order to fit our solution to the SSIM map considered in this work. The overall gradient-free CLVQ (GF-CLVQ) algorithm is outlined in Algorithm 4. With SSIM map and DSSIM function $g(x, \hat{x})$ in hand, we can get the DSSIM value given coordinates of reference point x and center point $\hat{x}$. The first modification we made is in the competition phase of the CLVQ procedure, where we choose the winning point based on the distortion measured with DSSIM instead of Euclidean distance. Subsequently, in the learning phase, we borrow the formula applied in simulation-based optimization proposed in [52] and make some adaptations for the CLVQ process. Applying the updating formula utilized in the

doi:10.6342/NTU202303932

zeroth-order method, we replace equation (6.6) with

$$\begin{aligned} a_{k+1}^* &= a_k^* - \gamma_k \cdot G(a_k^*, y, \gamma_k), \\ G(a_k^*, y, \gamma_k) &= \frac{g(a_k^* + \gamma_k \cdot (y - a_k^*), y) - g(a_k^*, y)}{\|y - a_k^*\|} \cdot (y - a_k^*), \end{aligned} \tag{6.9}$$

where $\|\cdot\|$ denotes the Euclidean distance and, in our case, the distortion function $g(x, \hat{x})$ is the DSSIM value when center position is at $\hat{x}$ and reference position is at x. The function $G(a_k^*, y, \gamma_k)$ can be regarded as an estimation of gradient based on the zeroth-order SSIM simulation results associated with the winning point and sample point. When updating the winning point toward the input sample, moving toward the negative direction of the gradient ensures an effective optimization step. This approach is named gradient-free since it allows for optimization in scenarios where first-order gradient information is not available. In this way, we can effectively optimize the quantization points for different distributions of quality maps. This is made possible by obtaining the distortion value at various position deviations using the available simulation results. For GF-CLVQ, we adopt a linear decay scheme for scheduling of gain parameters. That is, the gain parameter γ_k for kth update is determined by

$$\gamma_k = \gamma_0\big(1 - \frac{k}{\nu}\big). \tag{6.10}$$

Algorithm 4 Gradient-Free Competitive Learning Vector Quantization (GF-CLVQ)

Input: $\mathbb{P}$: probability distribution, n: number of quantization points, $g(x, \hat{x})$: DSSIM function, ν: limit of epochs, γ_0: initial gain parameter

Initialization: Set $\alpha = \{a_1, a_2, ..., a_n\}$ to be a set of n quantizers by sampling n points from distribution $\mathbb{P}$

1: **for** $k \in \{1, 2, ..., \nu\}$ **do**
2: Take a sample y from $\mathbb{P}$
3: Find quantizer a_i that attains the lowest DSSIM $g(a_i, y)$
4: Compute gain parameter γ_k with equation (6.10)
5: Update a_i with equation (6.9)
6: **end for**

Output: α: set of optimized vector quantizers

6.2.3 Cross-frame Optimization

In consideration of a practical multiview generation algorithm that can optimize the reference positions for transcoded views in real-time, 2D quantization may not be suitable due to the extended convergence time. Compared with 1D quantization optimization with Newton's method which yields vector quantizers

doi:10.6342/NTU202303932

within 10 epochs of updates, the 2D quantization algorithm may take up to 10000 epochs to reach optimized quantization points. Required computation time for the 2D quantization algorithm including CLVQ and GF-CLVQ may exceed the limitation for MTP delay when the prediction window is fixed. Therefore, it is essential to revise the overall procedure of optimal 2D quantization that can complete the computation in time at the edge server.

Recall that the objective of the optimization problem is to mitigate viewport prediction error when selecting the positions to transcode volumetric video. The accuracy of viewport prediction model applied is expected to remain relatively stable in the short term. On the other hand, initialization of the 2D quantization algorithm remains a challenge, as inadequate initialization can result in suboptimal outcomes. Hence, it is reasonable to optimize the quantization points based on previous results. The advantages include reducing the time required for convergence in 2D quantization and achieving a more solid set of vector quantizers against variation of viewport prediction results.

Algorithm 5 Cross-frame Optimization

Input: n: number of quantization points, $g(x, \hat{x})$: DSSIM function, ν: limit of epochs

Initialization:

for $t \in \{1, 2, ..., K\}$ **do** ▷ Cold-start
  Update parameters of $\mathbb{P}^t$
  Initialize α^t with 1D quantization
  for $k \in \{1, 2, ..., \nu\}$ **do**
    Take a sample y from $\mathbb{P}^t$
    Find quantizer a_i^t that attains the lowest DSSIM $g(a_i^t, y)$
    Compute gain parameter γ_k with equation (6.10)
    Update a_i^t with equation (6.9)
  end for
end for
Initialize α_c with 1D quantization
for $t \in \{K+1, ..., T\}$ **do** ▷ Cycle optimization
  Update parameters of $\mathbb{P}^t$
  Update γ_0 with equation (6.12)
  $\alpha^t = \alpha_c$
  for $k \in \{1, 2, ..., \nu\}$ **do**
    Take a sample y from $\mathbb{P}^t$
    Find quantizer a_i^t that attains the lowest DSSIM $g(a_i^t, y)$
    Compute gain parameter γ_k with equation (6.10)
    Update a_i^t with equation (6.9)
  end for
  $\alpha_c = \alpha^t$
end for

doi:10.6342/NTU202303932

The implementation of the proposed multiview generation problem is demonstrated in Algorithm 5. The overall procedure can be split into two stages: cold-start and cycle optimization. On a segment-by-segment basis, the GF-CLVQ algorithm is applied to find the quantization points to determine the positions of pre-rendered segments. Before the iteration process of GF-CLVQ, the target probability distribution is updated with prior information on viewport prediction accuracy, and the initial set of quantizers is chosen differently in two distinct phases.

At the beginning of the process (i.e., time before K segments), when there is no prior information available about the accuracy of viewport prediction, our confidence in the optimization during the initial phase is relatively low. As a result, during the cold-start stage, the initialization is based on optimal 1D quantization results with acceptable performance. After the initialization of vector quantizers, the quantization set α^t at segment t is passed through the GF-CLVQ process. Related code to the cold-start stage is specified from line 1 to line 10 in Algorithm 5.

From $K+1$ segments to the end of the streaming process, the algorithm enters the phase of cycle optimization. After we get a better grasp of the accuracy of viewport prediction, we can reuse the results obtained before as an initial set of quantizers since the chosen set of quantization points, corresponding to the positions for pre-rendering 2D images, is getting stable when we can better model the viewport prediction error with probability density function $\mathbb{P}$.

The parameters for the probability distribution $\mathbb{P}$ are updated in every iteration, which is specified in line 5 and 17 in Algorithm 5. We make the assumption that the viewport prediction is unbiased, thus setting the mean μ to be 0. Additionally, we utilize the L1-norm of the viewport prediction error up to time t as the standard deviation σ at that time. In our case with a sampling rate of 200Hz, the standard deviation $\sigma_{x,t}$ of the distribution used to model the x coordinate of the user at time t can be calculated as

$$\sigma_{x,t} = \frac{1}{200t}\sum_{i=1}^{200t}|e_i| = \frac{1}{200t}\sum_{i=1}^{200t}|\hat{x_i} - x_i|, \tag{6.11}$$

where e_i is the viewport prediction error of ith sample. Computation on the value of standard deviation is applied to Gaussian distribution on Laplace distribution alike.

Nevertheless, the value of the gain parameter should be determined with additional attention to ensure the stability of the quantization points. Between the

doi:10.6342/NTU202303932

optimization between each segment, we update the initial value of the gain parameter for segment t with equation

$$\gamma_0^t = \exp\big(-0.05 \times (t - K)\big) + 0.1. \tag{6.12}$$

By setting the initial value of gain parameter to a decaying value, we make certain that the variation of quantization points during the optimization process does not change significantly.

doi:10.6342/NTU202303932

CHAPTER 7

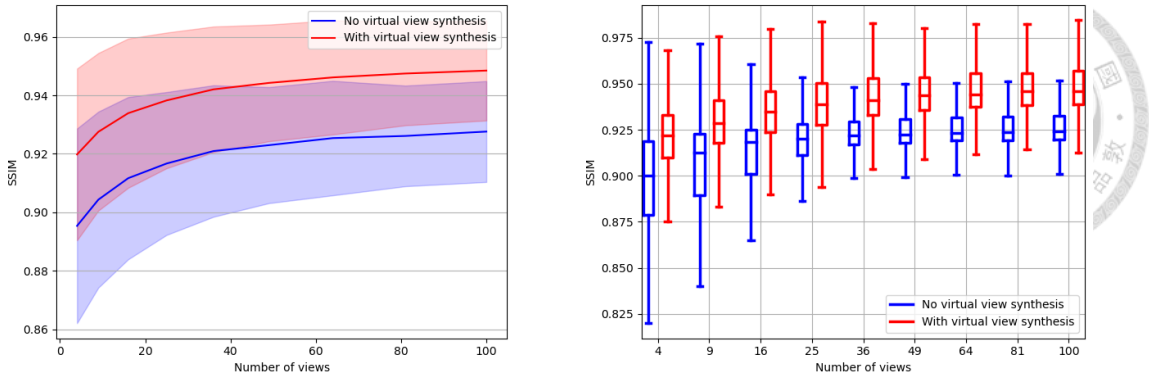
EVALUATION AND ANALYSIS

Extensive experiments are conducted based on the real-world 6DoF traces of 14 users provided by [37] along with 8i voxelized point cloud dataset [43]. We focus on evaluating the results of quantization approaches elaborated in Chapter 6 and compare the results with the state-of-the-art multiview generation method proposed in Vues [3]. We would demonstrate the benefit of applying optimal quantization solution on the multiview generation module and analyze the effectiveness by delving into the details when choosing reference positions for pre-rendering 2D views at the edge server.

7.1 Simulation Setup

We develop a simulation platform for evaluation of the performance of different multiview generation algorithms. It is assumed that users can freely roam in a room to explore the volumetric object placed in the virtual world. Meanwhile, translation and rotation traces are recorded and delivered to the edge server in real-time. Also, the SSIM map is generated offline for real-time evaluation. In the simulation process at time t seconds, viewport prediction with linear regression is first exploited to compute the predicted position of user at time $t+1$ seconds, with a prediction window of 1 second. Next, the multiview generation module selects the final reference positions on a segment-by-segment basis. Finally, the achievable similarity between transcoded views and actual views is computed by taking the maximal value of SSIM indices among all the reference views according to the SSIM map that indicates the relation between SSIM value and the position deviation. The performance evaluation of each approach primarily relies on the average SSIM value across 14 user traces, with additional insights provided through box plots depicting the median and quartiles of the SSIM value, ensuring a comprehensive assessment. We compare the performance of each method across varying numbers of views to assess the flexibility and adaptability of each algorithm when the number of views used in transcoding fluctuates. In a practical streaming system, the total number of views would be determined based on estimated bandwidth to attain optimal quality by fully utilizing the communication resource.

doi:10.6342/NTU202303932



(a) Results comparing with and without virtual view synthesis

(b) Box plot comparing with and without virtual view synthesis

Figure 29: Results comparing with and without virtual view synthesis

7.2 Performance of Incorporating Virtual View Synthesis

In this section, we showcase the benefit of employing virtual view synthesis technique within the volumetric video streaming system. As mentioned in Chapter. 4, incorporating a virtual view synthesis module on the client side addresses the limitations of directly using neighbor view, effectively mitigating the substantial variation that may arise between the pre-rendered views and the actual views experienced by users. Fig. 29 presents the impact on SSIM values when comparing the use of virtual view synthesis approach with its absence. The reference points are selected using a fixed interval of 0.1m, which aligns with the method employed in Vues [3]. Simulation results with and without the virtual view synthesis process are calculated based on the SSIM map shown in Fig. 21(a) and Fig. 15(a) respectively. Recall that Fig. 21(a) depicts the results obtained using both neighbor views and synthesized views, while Fig. 15(a) represents the results obtained solely with neighbor views. The results indicate that the utilization of virtual view synthesis enhances the SSIM value by approximately 0.2, regardless of the number of views. Therefore, it is confirmed that implementing virtual view synthesis before playback on client devices does bring an advantage to the overall streaming system.

7.3 Performance of Multiview Generation Algorithms

In this section, we compare and analyze the performance of different multiview generation algorithms. The multiview generation approach proposed by the state-of-the-art volumetric streaming system, Vues [3], acts as the baseline method to be compared with classical quantization solutions as well as proposed solutions

doi:10.6342/NTU202303932

for multiview generation problem. In order to better capture the performance of each approach, the performance is compared over different numbers of views or the number of quantizers utilized in the quantization problem.

7.3.1 Baseline Algorithm for Comparison

In Vues [3], a state-of-the-art transcoding-based edge-assisted volumetric streaming system, the multiview generation module selects reference positions by expanding the viewport prediction results with a fixed step size of 0.1m and the number of views are set to 1, 3, 5, or 7 in Vues. To facilitate a fair comparison between the methods employed in Vues and the proposed algorithms, a consistent approach of setting a fixed step size of 0.1m is applied across varying numbers of views. Although virtual view synthesis is not utilized in Vues, we simulate its effect by employing a mixed SSIM map to evaluate the performance of the multiview generation methods.

7.3.2 Performance of Proposed Methods

7.3.2.1 Average SSIM over users

Summaries of simulation results applying different multiview generation approaches are demonstrated in Fig. 30, in which performances are compared across baseline method proposed by Vues, optimal 1D quantization with Gaussian distribution, and 2D quantization with GF-CLVQ algorithm. The number of quantization points n is set to an identical value to ensure a fair comparison. It is evident that the proposed methods outperform baseline Vues' method in terms of average SSIM values regardless of the number of views applied. Additionally, the average SSIM values achieved with GF-CLVQ are slightly higher than those of other methods. When considering the DSSIM values, the GF-CLVQ method shows an improvement of approximately 6.5% compared to the Vues method when the number of views is 4 and 100. The improvement is less effective when number of views is around 20 to 40. This is because the optimal value of interval in quantization with fixed interval coincides with the value of 0.1 meters applied in Vues.

Although the improvement appears modest, it is worth noting that achieving reference views with an SSIM higher than 0.95 proves to be challenging based on the experimental findings from the SSIM map. Also, the following example serves as justification for the benefits of the progress made. In the case of the number of views being 4, the SSIM value is improved from 0.9198 to 0.9252 comparing Vues' method and GF-CLVQ. Fig. 31 showcases examples of transcoded views with SSIM values of 0.9197 and 0.9253 as well as the reference view. As can be

doi:10.6342/NTU202303932

seen, views of SSIM value 0.9197 have lower quality in comparison to the one with SSIM value 0.9253, particularly in the region highlighted by the red boxes, where some cracks of the soldier are noticeable. In addition, we compute the equivalent value of position deviation of transcoded views along x direction which leads to the average SSIM value achieved in the simulation for different numbers of views, which is shown in Fig. 30(c). It can be seen that the GF-CLVQ method corresponds to the lowest value of position deviation Therefore, it is reasonable to conclude that the advancement in quality of transcoded view is still advantageous and distinguishable by humans.

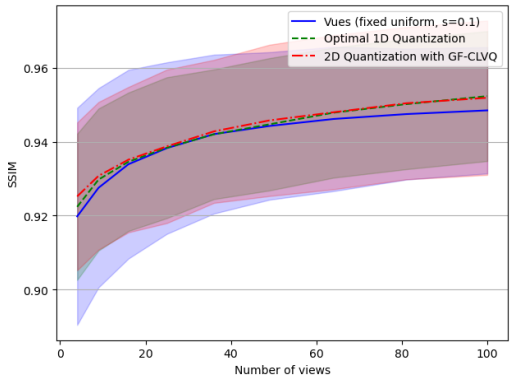
Fig. 30(d) displays the average ratio of improvement when compared segment-by-segment with the method used in Vues, across all users. The improvement ratio is computed by dividing the total number of improved segments by the total number of segments. It is clear that the GF-CLVQ algorithm has better outcomes compared with other quantization methods and also the baseline multiview generation method proposed in Vues. Generally, utilization of optimal 1-dimensional quantization method and 2D quantization with CLVQ except for the number of views is 16, 25, or 36. The application of GF-CLVQ leads to the highest improvement ratio when the number of views is 4 with 83.33%. Despite modest enhancement, when the number of views is 25 or 36, GF-CLVQ brings about an improvement ratio of 55.26% compared to the method applied in Vues.

Furthermore, an approximate number of views required to attain certain values of the SSIM index are summarized in Fig. 30(e). It is evident that the application of proposed method effectively saves computation and communication resources while ensuring the quality of transcoded views. In particular, 19.2% of views can be saved when targeting an SSIM value of 0.945. Also, Vues' method cannot attain an SSIM value of 0.95 even with 100 views while 2D quantization with GF-CLVQ yields the target result with 61 views.

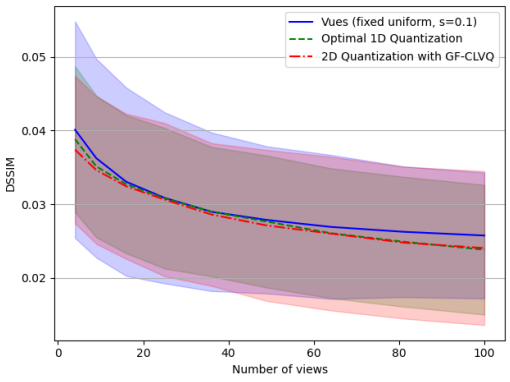
7.3.2.2 SSIM of individual user

To gain a deeper understanding of the system performance, we evaluate the quality of transcoded views for individual users. Fig. 32 illustrates the SSIM values obtained by three multiview generation methods over each user when using 9 and 81 views, arranged in ascending order of viewport prediction error. It can be observed that there is an inverse relationship between the SSIM values and the average viewport prediction error. This is logical since a higher deviation between the predicted position and the actual user position results in inferior quality of transcoded views. Also, a higher standard deviation over the quality is introduced with a larger viewport prediction error. Upon examining the SSIM

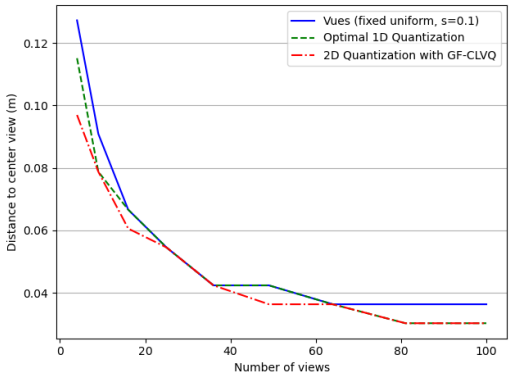
doi:10.6342/NTU202303932



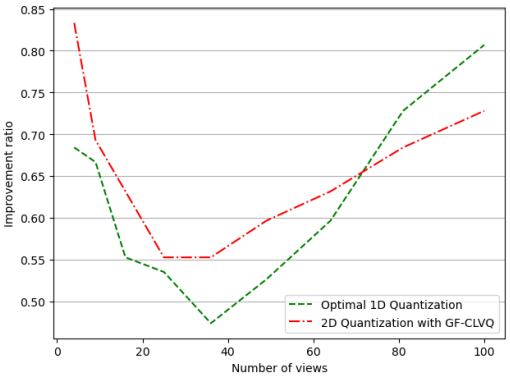
(a) SSIM



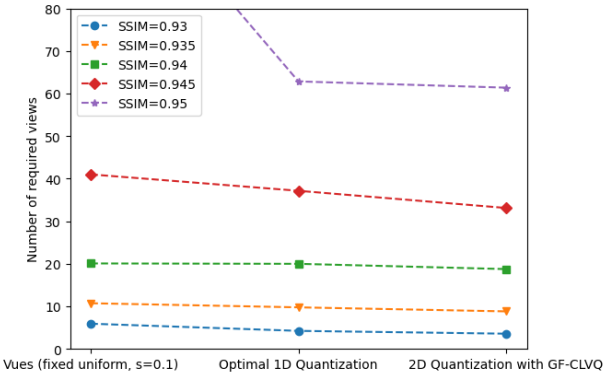
(b) DSSIM



(c) Distance to center view



(d) Improvement rate



(e) Required number of views

Figure 30: Results of different methods

doi:10.6342/NTU202303932

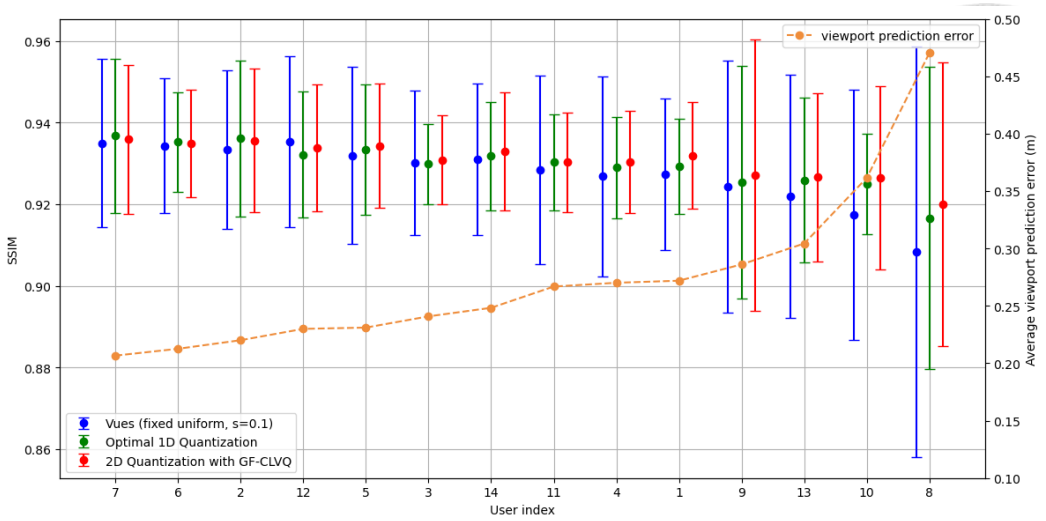
(a) Reference view

(b) SSIM=0.9253

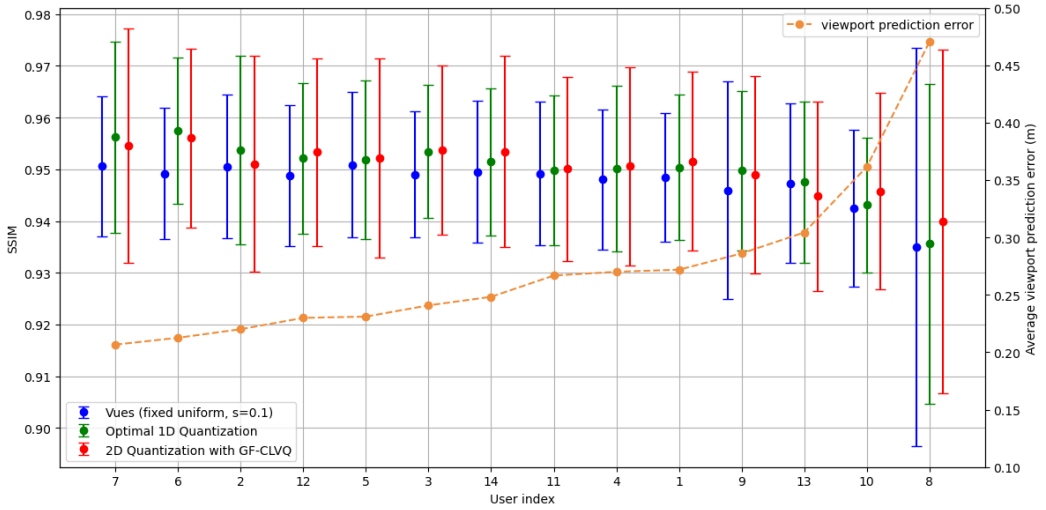
(c) SSIM=0.9197

Figure 31: Examples of transcoded views

doi:10.6342/NTU202303932



(a) n=9



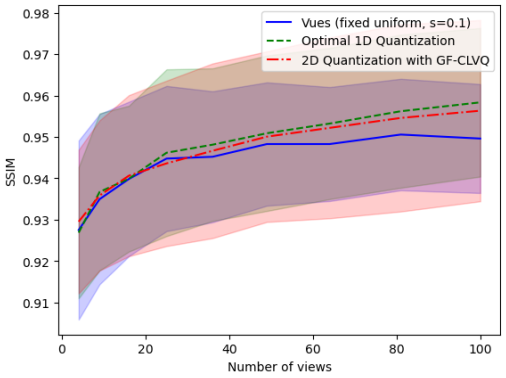
(b) n=81

Figure 32: Results of each user

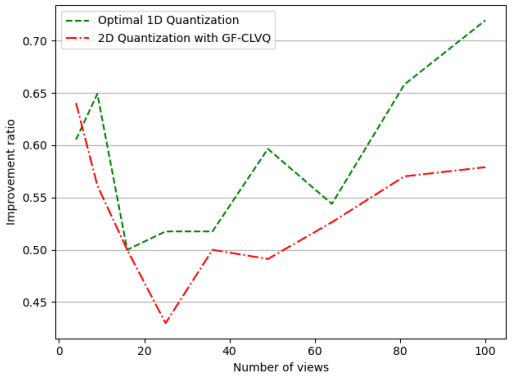
values obtained by the three methods, it is confirmed that our proposed algorithms outperform Vues' method for the majority of individual user traces, irrespective of the accuracy of the viewport prediction and the number of views applied. However, the outcomes of optimal 1D quantization and 2D quantization with GF-CVLQ vary for each user trace. This variation can be attributed to the randomness of position deviation, which is challenging to model using a single probability distribution.

To further investigate, we choose two users with varying levels of viewport prediction accuracy to verify that our proposed algorithm could dynamically adapt the reference positions to the information of prior viewport prediction errors in the multiview generation model. Tested with a lightweight linear regression viewport prediction model, the viewport trace of user7 and user8 yield the lowest and highest

doi:10.6342/NTU202303932

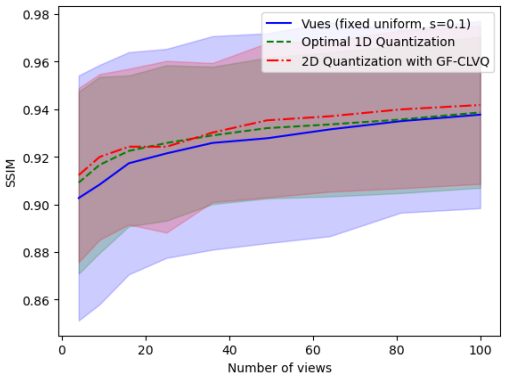


(a) Average and standard deviation

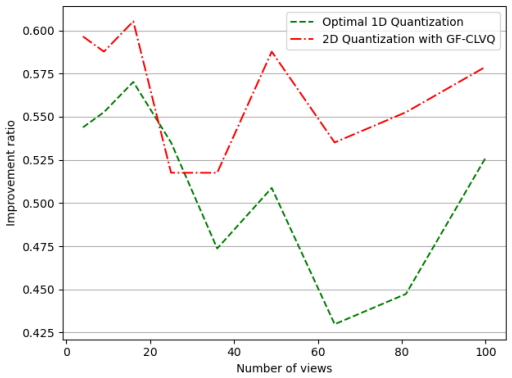


(b) Segment-by-segment improvement ratio

Figure 33: Results of user7



(a) Average and standard deviation



(b) Segment-by-segment improvement ratio

Figure 34: Results of user8

average viewport prediction error of values 0.207 and 0.470 meters. Fig. 33 and 34 present the simulation results of two users comparing the performance of the Vues method, 1D quantization, and GF-CLVQ method. As shown, there is an improvement in the average values of the SSIM index, and in some cases, the improvement can reach up to 0.2. It could be validated that the proposed GF-CLVQ algorithm is capable of boosting the SSIM values regardless of the viewport prediction accuracy and number of views.

7.4 Analysis of System Performance

In this section, we compare the performance of different methods in detail and further verify the reason for various outcomes achieved by different algorithms. Through in-depth analysis, we seek to uncover the underlying factors that contribute to the observed differences in performance.

doi:10.6342/NTU202303932

7.4.1 Observation on Step Size in 1D Quantization

7.4.1.1 Observation on step size in 1D quantization with fixed interval

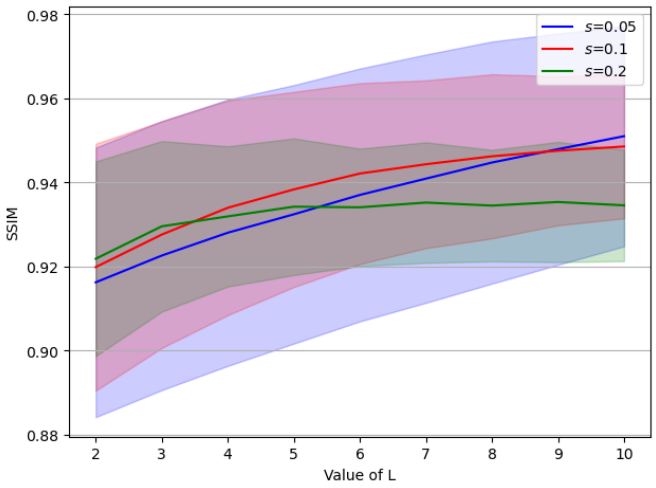


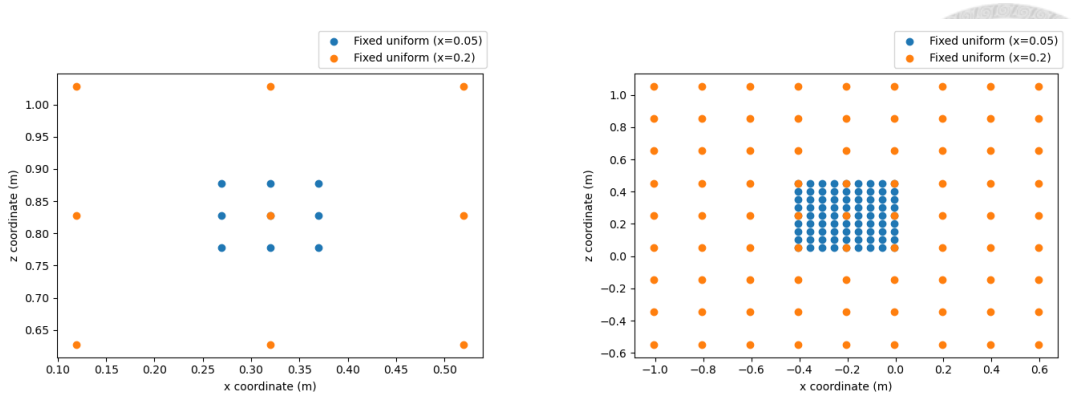
Figure 35: Results of quantization with the uniform interval setting difference values of step size (Vues method)

We present the performance of finding reference positions with a fixed value of spacing between each two quantizers in order to gain insight into how the number of views impacts the decision on the reference positions. Here we test with step size values of 0.05m, 0.1m, and 0.2m. Results of a different decision on the value of step size s are compared in Fig. 35.

In Fig. 35, it can be noted that no specific step size value consistently outperforms all the other choices over different numbers of views, where L is the number of quantization points along a 1-dimensional direction and $n = L^2$ is the total number of views used. In other words, the optimal value of step size is dependent on the number of views. While using the fixed step size of 0.1m, similar to Vues, can achieve an approximately ideal result, there is still significant potential for improvement in selecting optimal reference positions compared to this simplistic approach. In addition, imposing a maximum limit of 7 views as Vues does would significantly limit the enhancement of the user's QoE, especially when the available bandwidth has not been fully utilized, as higher SSIM can be achieved with more views.

The objective of the multiview generation problem is to minimize the distance between the actual user position and the nearest reference position when deciding positions to create transcoded views so as to increase the SSIM value (decrease the DSSIM) of pre-rendered viewports. Intuitively, it is apparent that the average SSIM value increases as the number of views used grows since we devote more

doi:10.6342/NTU202303932



(a) L=3 (b) L=9

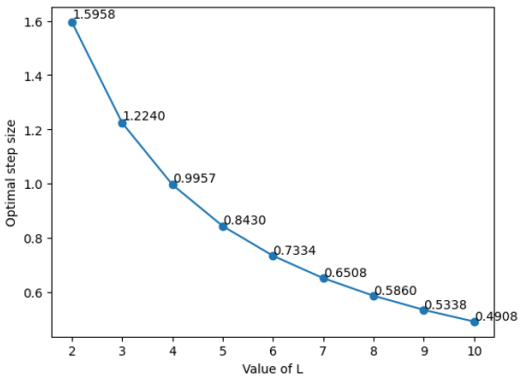
Figure 36: Illustrations of reference positions with fixed interval

resources to transcoding the 2D views and transmitting the data. To provide further insight into the results, we explore the specific factors contributing to the positive performance. When employing a limited number of views (e.g., $L = 3$), a larger step size s leads to better performance as it allows for greater coverage by expanding the distance between two reference positions. By enlarging the span of chosen points, the probability of the actual user position being encompassed by the coverage of reference positions is increased, thereby enhancing the SSIM value. Conversely, when the number of views is large (e.g., $L = 9$), a smaller value of step size s yields a superior result since concentration matters more given that the criteria of coverage is automatically met with a larger number of reference positions. That is, when the coverage provided by the reference positions is already sufficient to encompass most of the deviation in viewport prediction results, increasing the density of reference positions can further augment the overall quality. Examples of reference positions determined with different values of step size s are illustrated in Fig. 36. For $L = 3$, the green point representing the actual user's position is closer to the nearest reference positions when a step size of 0.2m is applied. On the other hand, for $L = 9$, a step size of 0.05m results in higher SSIM values. Additionally, using a smaller step size s results in a higher standard deviation. This is because extreme SSIM index values occur more frequently when the actual user position is very close to or significantly far from any of the reference positions. These situations are more likely to happen when a smaller step size is set in the multiview generation model.

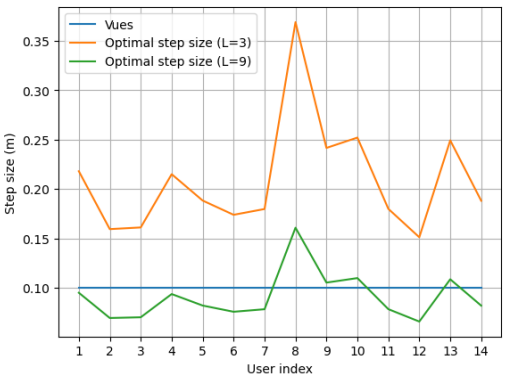
7.4.1.2 Observation on step size in optimal 1D quantization with uniform interval

Next, we evaluate the performance of acquiring optimal step size with Newton's method for the 1-dimensional quantization problem with uniform interval.

doi:10.6342/NTU202303932



(a) Optimal step size for 1D quantization with uniform interval



(b) Step size for 1D quantization with uniform interval of each user

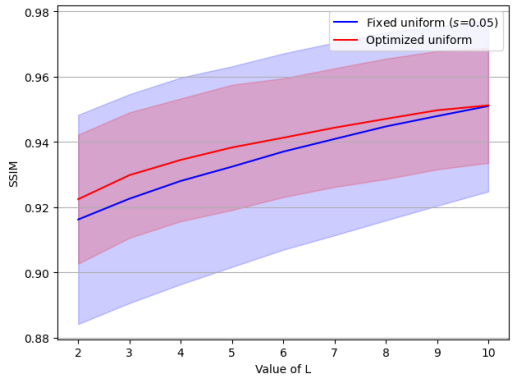
Figure 37: Simulation results of 1D quantization with uniform interval

Following the process outlined in Algorithm 1, we can obtain the optimal step size for Normal distribution $\mathcal{N}(\mu, \sigma)$.

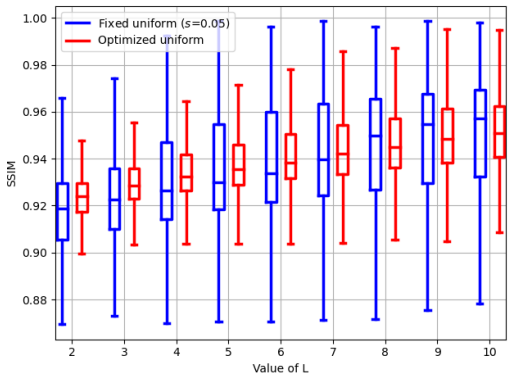
In particular, for 1-dimensional standard Normal distribution (i.e., $\mathcal{N}(0, 1)$), the value of optimal step size is shown in Fig. 37(a) and summarized in Table 2. It is evident that the optimal step size decreases as the number of quantizers increases since more concentrated quantization points can lower the expected distortion of target probability distribution with an increased number of points. On the other hand, Fig. 37(b) illustrates values of step size obtained with the proposed 1D quantization with a uniform interval algorithm. Due to the different viewport prediction accuracy of each user, the values of step size should be adjusted to accommodate the different probability distributions of user position. The results once again highlight the significance of selecting an appropriate step size based on the number of points and the parameters of the probability distribution, rather than using a fixed step size.

Fig. 38 displays the results of optimized quantization with a uniform interval in contrast with quantization with a fixed interval. It is evident that the optimized version of quantization with uniform interval exhibits a finer performance under average SSIM value compared with the approach employing fixed interval except for the particular testing case of $s = 0.1m, L = 6$. The deviation in performance for these cases could be attributed to the randomization of users' traces, and the average SSIM values remain comparable to the results obtained using a fixed interval approach. The advantage of applying optimal quantization with uniform interval is its ability to dynamically adapt the step size s based on varying numbers of views, ensuring consistent optimization of quality regardless of the number of views used.

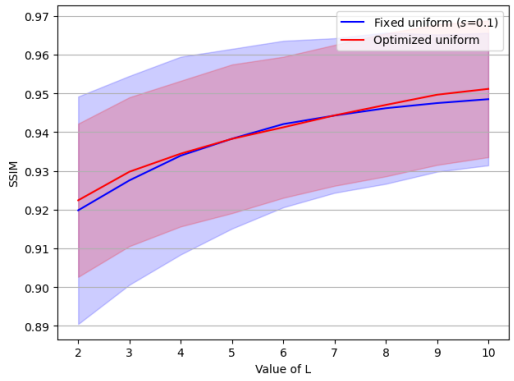
doi:10.6342/NTU202303932



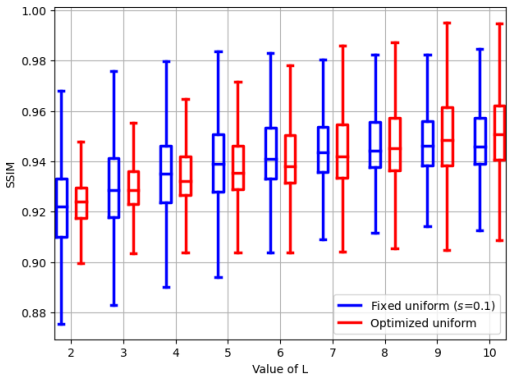
(a) Average and standard deviation comparing with fixed interval of $s = 0.05m$



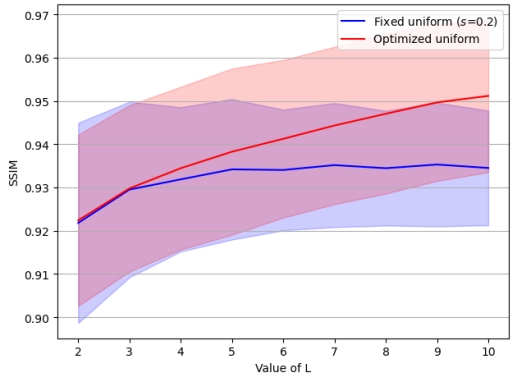
(b) Box plot comparing with fixed interval of $s = 0.05m$



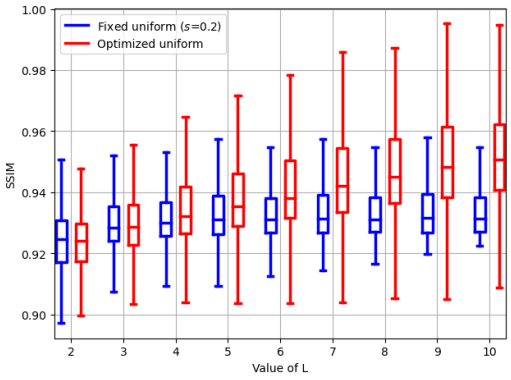
(c) Average and standard deviation comparing with fixed interval of $s = 0.1m$



(d) Box plot comparing with fixed interval of $s = 0.1m$



(e) Average and standard deviation comparing with fixed interval of $s = 0.2m$



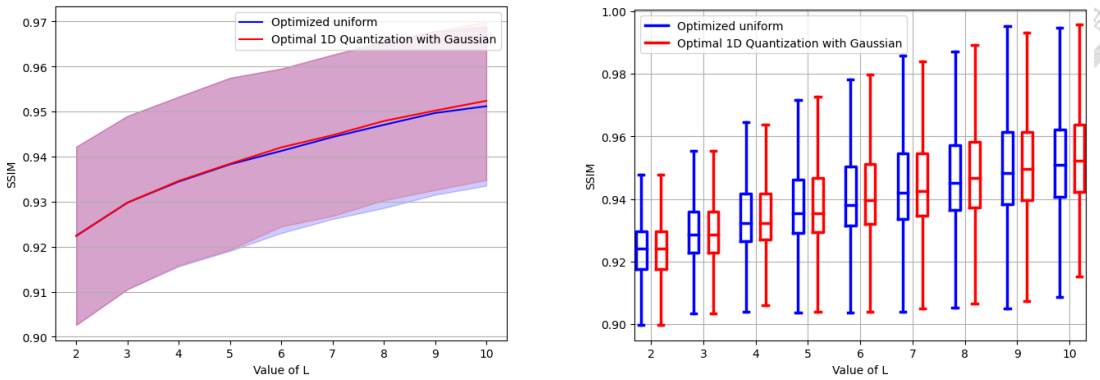
(f) Box plot comparing with fixed interval of $s = 0.2m$

Figure 38: Performance of optimized quantization with uniform interval

doi:10.6342/NTU202303932

Number of quantizers	2	3	4	5	6	7	8	9	10
Optimal step size	1.5958	1.2240	0.9957	0.8430	0.7334	0.6508	0.5860	0.5338	0.4908

Table 2: Optimal value of step size for 1D quantization with uniform interval



(a) Average and standard deviation

(b) Box plot

Figure 39: Results comparing optimized quantization with uniform interval and optimal 1D quantization

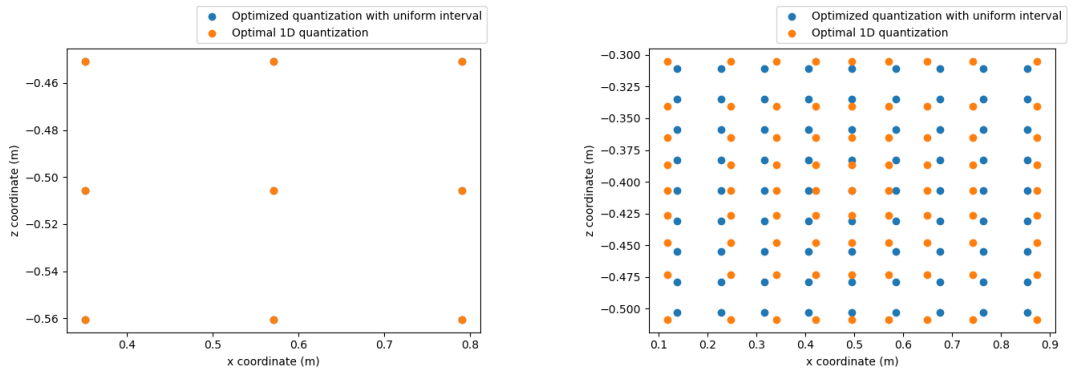
7.4.1.3 Observation on step size in optimal 1D quantization

We can obtain a more precise set of quantization points with Newton's process stated in Algorithm 2 by removing the constraint of the uniform interval when modeling the user's x and z coordinates with two 1-dimensional probability distributions. We use 1-dimensional Normal distribution $\mathcal{N}(\mu, \sigma)$ to model user's x and z coordinates here. Similarly, the parameters μ and σ are set to 0 and the L1-norm of the prediction error, respectively. Results comparing optimal 1D quantization and quantization with fixed interval are demonstrated in Fig. 39. It can be observed that optimized quantization with uniform interval and optimal 1D quantization yield comparable performance. The reason may lie in the resemblance in the selection of the reference position between the two approaches, which is shown in Fig. 40. These two methods yield identical reference positions when $L = 3$, and the selected points are very close together when $L = 9$.

7.4.2 Comparison between Different Probability Distributions

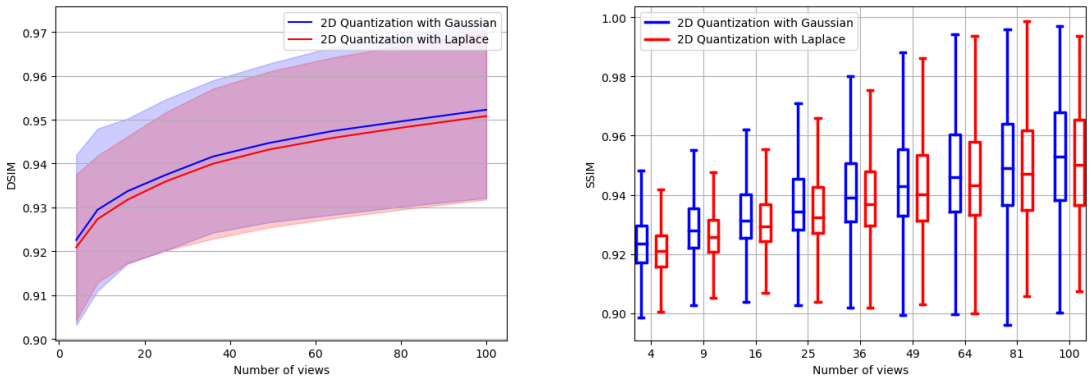
We evaluate the performance of applying the CLVQ algorithm on 2-dimensional Gaussian distribution and Laplace distribution. Based on the graphical analysis for the distribution of viewport prediction error, it is indicated that neither the Gaussian nor the Laplace probability model can fully capture the distribution. The objective of this section is to identify a more suitable probability distribution to model user position in a 6DoF scenario for our streaming system. Fig. 41 sketches the comparison between results on two distributions. According to our

doi:10.6342/NTU202303932



(a) L=3 (b) L=9

Figure 40: Reference positions selected with optimized quantization with uniform interval and optimal 1D quantization



(a) Average and standard deviation (b) Box plot

Figure 41: Results comparing 2D quantization with CLVQ on Gaussian distribution and Laplace distribution

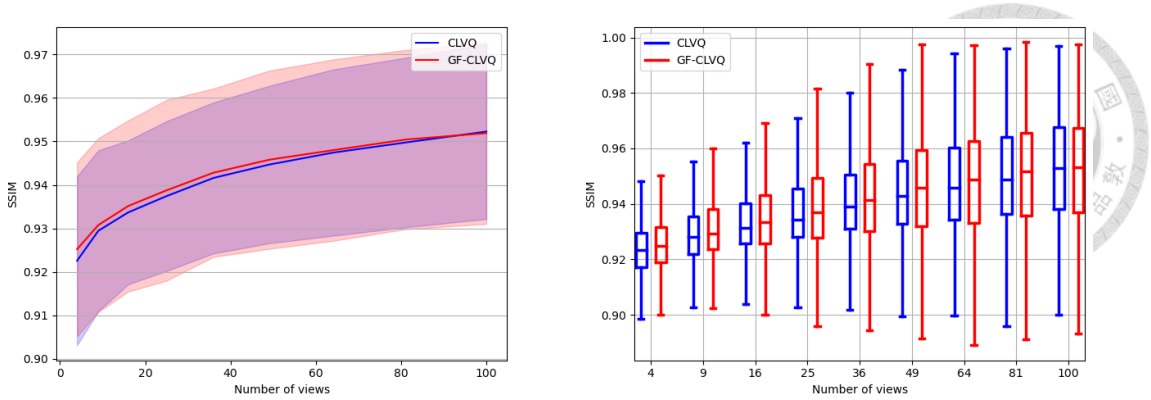
simulation results, it can be seen that exploiting Gaussian distribution leads to higher SSIM, suggesting that Gaussian distribution is a more suitable model for representing the position of user in a 6DoF scenario.

7.4.3 Observation on Algorithm Design

7.4.3.1 Comparison between CLVQ and GF-CLVQ

Fig. 42 presents the results comparing 2D quantization with CVLQ and GF-CLVQ. It is evident that additional innovations, including the integration of the simulation-based optimization approach in CLVQ and careful consideration in the initialization of the quantization set, do enhance the SSIM results in terms of the average value. To further investigate the impact of incorporating the gradient-free concept into the CLVQ algorithm, we examine the quantization points selected by these two methods. Fig. 43(a) illustrates the reference positions selected with

doi:10.6342/NTU202303932



(a) Average and standard deviation (b) Box plot

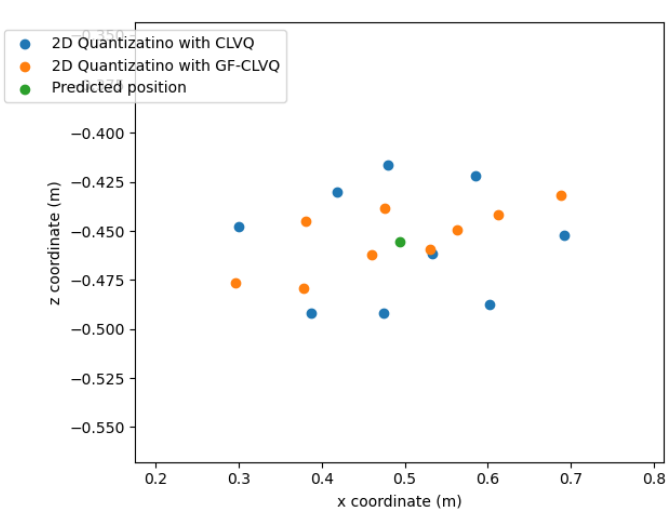
Figure 42: Results of 2D quantization with CLVQ and GF-CLVQ

CLVQ and GF-CLVQ methods when using 9 views (n=9). The reference positions selected with the proposed algorithm can be seen to be distributed in the upper-right to lower-left direction, which complements the pattern of the SSIM map. This phenomenon can be attributed to the application of GF-CLVQ, which takes into account the gradient of the SSIM map. The size of the updating step depends on the magnitude of the gradient. When the gradient is steeper, the point takes larger strides in the updating process. Fig. 43(b) and 43(c) portray the contour lines of SSIM values achieved by reference positions selected with respective methods. It can be observed that reference positions selected with GF-CLVQ result in SSIM values that distribute more evenly around the predicted position. This example manifests the underlying reason for the superior performance of the GF-CLVQ method.

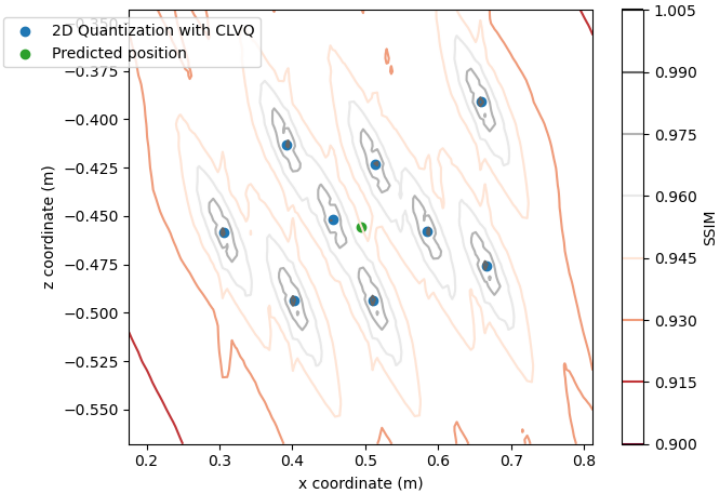
7.4.3.2 Observation on cross-frame optimization

In this subsection, we focus on the effect of applying cross-frame optimization. We first perform simulations on GF-CLVQ methods using different maximal update values ν. The results are illustrated in Fig. 44. As can be seen, the increasing number of updates can lead to a higher SSIM value, particularly when the number of quantization points is increased. The improvement becomes more pronounced as the number of views increases, as a greater number of updates are needed to reach optimality. This can be thought up as the trade-off between computation resources and optimization performance. Fig. 45 shows the performance compared between the original GF-CVLQ of 100 maximal updates and GF-CLVQ with cross-frame optimization of 100 maximal updates. In addition to the slight improvement in quality, the utilization of cross-frame optimization can result in a reduction of approximately 90% of the computation resource as well as computation time. It is important to mention that the computation time required for

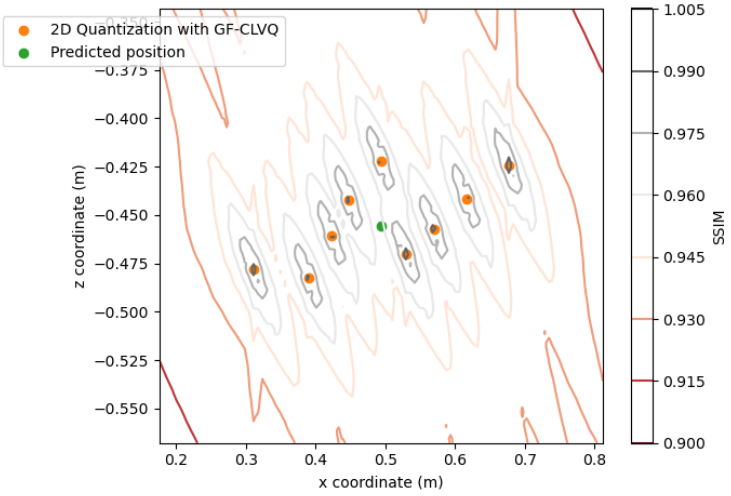
doi:10.6342/NTU202303932



(a)



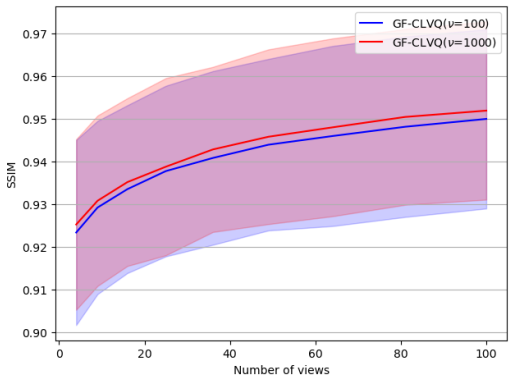
(b)



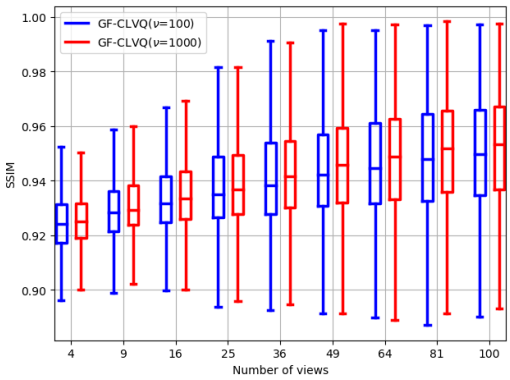
(c)

Figure 43: (a)Reference positions selected with CLVQ and GF-CLVQ when n=9 (b)Contour lines of achievable SSIM of reference positions selected with CLVQ (c)Contour lines of achievable SSIM of reference positions selected with GF-CLVQ

doi:10.6342/NTU202303932

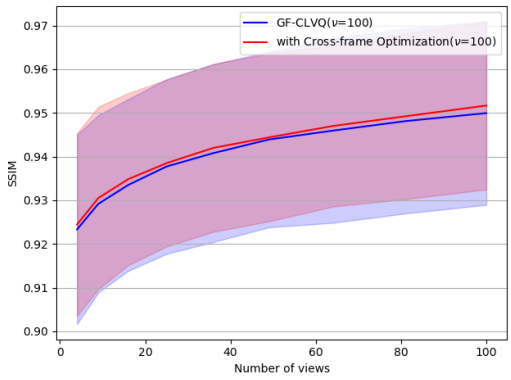


(a) Average and standard deviation

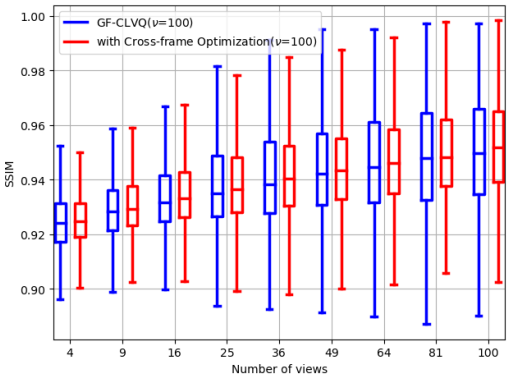


(b) Box plot

Figure 44: Results of GF-CLVQ with different number of maximal updates ν



(a) Average and standard deviation



(b) Box plot

Figure 45: Results of cross-frame optimization

1000 updates is shorter than the duration of a single segment, which meets the practical implementation constraints. However, it is always a benefit in reducing the computational energy consumed by the edge server.

doi:10.6342/NTU202303932

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Streaming of volumetric video is an unsolved problem to facilitate the immersive experience and formation of the metaverse. This work proposes to leverage edge rendering on point cloud data for streaming high-quality video frames under the bandwidth constraint of current networking technology. To solve the problem, the overall system design is built upon a holistic adaptive video streaming system with the incorporation of optimization on multiview generation as well as virtual view synthesis techniques. We formulate the multiview generation problem into an optimal quantization problem. Conventional algorithms, including Newton's method and competitive learning vector quantization, to solve quantization problems are tested. In addition, concepts of gradient-free stochastic gradient descent for simulation-based optimization are employed in the CLVQ algorithm.

With a thorough evaluation of real-world VR user traces and the SSIM map generated with real-world volumetric data, we confirm that the application of the virtual view synthesis technique can bring a 0.2 improvement in average SSIM value. Also, we validate that the proposed multiview generation algorithm could improve the average SSIM value by 0.05. In addition, the proposed algorithms outperform baseline methods proposed by state-of-the-art edge-assisted volumetric streaming system with improvement ranging from 55% to 83% on a segment-by-segment basis. The proposed optimization framework could better select the transcoded views to save computation and communication resources as well as enhance the quality of pre-rendered images.

8.2 Future Work

8.2.1 Optimality Analysis

In this work, the proposed multiview generation algorithm determines the reference positions with the GF-CLVQ process along with careful consideration of gain parameters. Although the procedure applied in this work could achieve a great improvement compared with the baseline method, the optimality and convergence of the proposed algorithm are not guaranteed. The derivation of proof for optimal solutions and convergence conditions for the stochastic process could be a new research direction.

doi:10.6342/NTU202303932

8.2.2 Practical Implementation and Experiment

While extensive simulations have been conducted, this work does not include practical implementation of the proposed streaming system using off-the-shelf devices and real-world networks. Advanced techniques for virtual view synthesis could be exploited to further enhance the quality of transcoded views. Also, Different quality evaluation metrics can be applied to the optimization framework to attain better QoE for users in real-world scenarios. Additionally, a user study could be conducted to provide an objective evaluation of the streaming system.

8.2.3 Multi-user scenario

In this study, the streaming process is simulated individually for each user. However, in real-world scenarios, multiple users often engage in immersive experiences simultaneously. In such cases, streaming for multiple users concurrently is necessary. Optimizing the reference positions of transcoded views for multiple users adds an additional dimension to the optimization process, including considerations for the similarity of viewports between users.

doi:10.6342/NTU202303932

APPENDIX A

PROOF OF EQUATION (6.2)

Let $\alpha = (a_1, ..., a_n), a_1 < a_2 < ... < a_n$, $a_{j\pm\frac{1}{2}} = \frac{a_j + a_{j\pm1}}{2}, j = 2, ..., n-1$, $a_{\frac{1}{2}} = -\infty, a_{n+\frac{1}{2}} = \infty$. By definition of distortion D_n can be computed as,

$$\begin{aligned} D_n(\alpha) &= \sum_{j=1}^{n} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (x_j - \tau)^2 \exp\big(-\frac{\tau^2}{2}\big) \frac{d\tau}{\sqrt{2\pi}} \\ &= \sum_{j=1}^{n} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (x_j^2 - 2x_j\tau + \tau^2) \exp\big(-\frac{\tau^2}{2}\big) \frac{d\tau}{\sqrt{2\pi}} \end{aligned} \tag{A.1}$$

We then solve three parts of equation (A.1) separately,

- $$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} x_j^2 \cdot \exp\big(-\frac{\tau^2}{2}\big) \frac{d\tau}{\sqrt{2\pi}} = x_j^2\big(\Phi(x_{j+\frac{1}{2}}) - \Phi(x_{j-\frac{1}{2}})\big) \tag{A.2}$$

- $$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} -2x_j\tau \cdot \exp\big(-\frac{\tau^2}{2}\big) \frac{d\tau}{\sqrt{2\pi}} = -\frac{2x_j}{\sqrt{2\pi}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tau \cdot \exp\big(-\frac{\tau^2}{2}\big) d\tau \tag{A.3}$$

  Let $t = \tau^2, dt = 2\tau d\tau$

  $$-\frac{2x_j}{\sqrt{2\pi}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tau \cdot \exp\big(-\frac{t}{2}\big)\frac{1}{2} d\tau = \frac{2x_j}{\sqrt{2\pi}}\Big(\exp\big(-\frac{x_{j+\frac{1}{2}}}{2}\big) - exp\big(-\frac{x_{j-\frac{1}{2}}}{2}\big)\Big) \tag{A.4}$$

- $$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tau^2 \cdot \exp\big(-\frac{\tau^2}{2}\big) \frac{d\tau}{\sqrt{2\pi}} \tag{A.5}$$

  Let $u = \tau, dv = \exp\big(-\frac{\tau^2}{2}\big)\tau d\tau, v = -\exp\big(-\frac{\tau^2}{2}\big)$, with integration by parts,

  $$\begin{aligned} &\tfrac{1}{\sqrt{2\pi}} \textstyle\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u dv = \tfrac{1}{\sqrt{2\pi}}\Big(uv\Big|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} - \textstyle\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v du\Big) \\ &= \tfrac{1}{\sqrt{2\pi}}\Big(-\tau\exp\big(-\tfrac{\tau^2}{2}\big)\Big|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + \textstyle\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \exp\big(-\tfrac{\tau^2}{2}\big) d\tau\Big) \\ &= -\tfrac{1}{\sqrt{2\pi}}\Big(x_{j+\frac{1}{2}}\exp\big(-\tfrac{x_{j+\frac{1}{2}}^2}{2}\big) - x_{j-\frac{1}{2}}\exp\big(-\tfrac{x_{j-\frac{1}{2}}^2}{2}\big)\Big) + \Big(\Phi(x_{j+\frac{1}{2}}) - \Phi(x_{j-\frac{1}{2}})\Big) \end{aligned} \tag{A.6}$$

By combining the three parts of integral, one can get equation (6.2).

doi:10.6342/NTU202303932

APPENDIX B

COMPUTATION OF EQUATION (6.3)

In optimal quantization problem with uniform interval, distance between each two quantizers is equal to a constant s. Let the set of quantizers $\alpha = (a_1, a_1 + s, ..., a_1+(n-1)\cdots)$. The elements for performing Newton's method (i.e. Equation (6.3) with 1-dimensional standard Normal distribution and L2 norm distortion function are calculated with basic calculus as follows,

$$
\begin{aligned}
\frac{\partial D_n}{\partial a_1}(\alpha) = \sum_{j=1}^{n} \Bigg(& 2a_j \cdot \Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) + (a_j^2+1)\frac{1}{\sqrt{2\pi}}\Big(\exp\,(-\frac{a_{j+\frac{1}{2}}}{2}) - \exp\,(-\frac{a_{j-\frac{1}{2}}}{2})\Big) \\
& - \frac{1}{\sqrt{2\pi}}\Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - a_{j+\frac{1}{2}}^2 \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) + a_{j-\frac{1}{2}}^2 \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Big) \\
& + \frac{2}{\sqrt{2\pi}}\Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Big) \\
& + \frac{2a_j}{\sqrt{2\pi}}\Big(- \exp(-\frac{a_{j+\frac{1}{2}}^2}{2})a_{j+\frac{1}{2}} + \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})a_{j-\frac{1}{2}}\Big)\Bigg) \\
= \sum_{j=1}^{n} \Bigg(& 2a_j \cdot \Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\
& + \Big(2 + \big(\frac{a_j - a_{j+1}}{2}\big)^2\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \Big(2 + \big(\frac{a_j - a_{j-1}}{2}\big)^2\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Bigg)
\end{aligned}
\tag{B.1}
$$

doi:10.6342/NTU202303932

$$
\begin{aligned}
\frac{\partial D_n}{\partial s}(\alpha) = \sum_{j=1}^{n} \Bigg(& 2a_j(j-1)\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\
& + (a_j^2+1)\frac{1}{\sqrt{2\pi}}\Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) \cdot \big(\frac{2j-1}{2}\big) - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) \cdot \big(\frac{2j-3}{2}\big)\Big) \\
& - \frac{1}{\sqrt{2\pi}}\big(\frac{2j-1}{2}\big) \cdot \exp(-\frac{a_{j+\frac{1}{2}}^2}{2})\Big(1 - a_{j+\frac{1}{2}}^2\Big) \\
& + \frac{1}{\sqrt{2\pi}}\big(\frac{2j-3}{2}\big) \cdot \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Big(1 - a_{j-\frac{1}{2}}^2\Big) \\
& + \frac{2(j-1)}{\sqrt{2\pi}}\Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Big) \\
& + \frac{2a_j}{\sqrt{2\pi}}\Big(- a_{j+\frac{1}{2}} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2})\big(\frac{2j-1}{2}\big) + a_{j-\frac{1}{2}} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\big(\frac{2j-3}{2}\big)\Big)\Bigg) \\
= \sum_{j=1}^{n} \Bigg(& 2a_j(j-1)\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\
& + \Big(\big(\frac{2j-1}{2}\big) \cdot \big(\frac{s}{2}\big)^2 + 2(j-1)\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) \\
& - \Big(\big(\frac{2j-3}{2}\big) \cdot \big(\frac{s}{2}\big)^2 + 2(j-1)\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\Bigg)
\end{aligned}
\tag{B.2}
$$

$$
\begin{aligned}
\frac{\partial^2 D_n}{\partial a_1^2} = \sum_{j=1}^{n} \Bigg(& 2\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) + 2a_j\frac{1}{\sqrt{2\pi}}\Big(\exp\big(-\frac{a_{j+\frac{1}{2}}}{2}\big) - \exp\big(-\frac{a_{j-\frac{1}{2}}}{2}\big)\Big) \\
& - \Big(2 + \big(\frac{s}{2}\big)^2\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) a_{j+\frac{1}{2}} + \Big(2 + \big(\frac{s}{2}\big)^2\Big) \cdot \frac{1}{\sqrt{2\pi}} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) a_{j-\frac{1}{2}}\Bigg)
\end{aligned}
\tag{B.3}
$$

$$
\begin{aligned}
\frac{\partial^2 D_n}{\partial s \partial a_1} = \sum_{j=1}^{n} \Bigg(& 2(j-1)\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\
& + 2a_j\frac{1}{\sqrt{2\pi}}\Big(\exp(-\frac{a_{j+\frac{1}{2}}^2}{2})\frac{2j-1}{2} - \exp(-\frac{a_{j-\frac{1}{2}}^2}{2})\frac{2j-3}{2}\Big) \\
& + \frac{1}{\sqrt{2\pi}}\frac{s}{2} \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) + \frac{1}{\sqrt{2\pi}}\Big(2 + \big(\frac{s}{2}\big)^2\Big) \exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) \cdot (-a_{j+\frac{1}{2}} \cdot \frac{2j-1}{2}) \\
& - \frac{1}{\sqrt{2\pi}}\frac{s}{2} \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) - \frac{1}{\sqrt{2\pi}}\Big(2 + \big(\frac{s}{2}\big)^2\Big) \exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) \cdot (-a_{j-\frac{1}{2}} \cdot \frac{2j-3}{2})\Bigg)
\end{aligned}
\tag{B.4}
$$

doi:10.6342/NTU202303932

$$\begin{aligned}\frac{\partial^2 D_n}{\partial a_1 \partial s} = \sum_{j=1}^{n} \Bigg(& 2(j-1)\cdot\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\ & + 2a_j(j-1)\cdot\frac{1}{\sqrt{2\pi}}\Big(\exp\left(-\frac{a_{j+\frac{1}{2}}}{2}\right) - \exp\left(-\frac{a_{j-\frac{1}{2}}}{2}\right)\Big) \\ & + \Big(\big(\frac{2j-1}{2}\big)\cdot\big(\frac{s}{2}\big)^2 + 2(j-1)\Big)\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j+\frac{1}{2}}^2}{2})(-a_{j+\frac{1}{2}}) \\ & - \Big(\big(\frac{2j-3}{2}\big)\cdot\big(\frac{s}{2}\big)^2 + 2(j-1)\Big)\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j-\frac{1}{2}}^2}{2})(-a_{j-\frac{1}{2}})\Bigg)\end{aligned} \tag{B.5}$$

$$\begin{aligned}\frac{\partial^2 D_n}{\partial s^2} = \sum_{j=1}^{n} \Bigg(& 2(j-1)^2\cdot\Big(\Phi(a_{j+\frac{1}{2}}) - \Phi(a_{j-\frac{1}{2}})\Big) \\ & + 2a_j(j-1)\cdot\frac{1}{\sqrt{2\pi}}\Big(\exp\left(-\frac{a_{j+\frac{1}{2}}}{2}\right)\big(\frac{2j-1}{2}\big) - \exp\left(-\frac{a_{j-\frac{1}{2}}}{2}\right)\big(\frac{2j-3}{2}\big)\Big) \\ & + \frac{s(2j-1)}{4}\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j+\frac{1}{2}}^2}{2}) - \frac{s(2j-3)}{4}\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j-\frac{1}{2}}^2}{2}) \\ & + \Big(\big(\frac{2j-1}{2}\big)\cdot\big(\frac{s}{2}\big)^2 + 2(j-1)\Big)\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j+\frac{1}{2}}^2}{2})(-a_{j+\frac{1}{2}})\big(\frac{2j-1}{2}\big) \\ & - \Big(\big(\frac{2j-3}{2}\big)\cdot\big(\frac{s}{2}\big)^2 + 2(j-1)\Big)\cdot\frac{1}{\sqrt{2\pi}}\exp(-\frac{a_{j-\frac{1}{2}}^2}{2})(-a_{j-\frac{1}{2}})\big(\frac{2j-3}{2}\big)\Bigg)\end{aligned} \tag{B.6}$$

doi:10.6342/NTU202303932

APPENDIX C

COMPUTATION OF EQUATION (6.4)

Let α be the set of vector quantizers and $\alpha = (a_1, ..., a_n), a_1 < a_2 < ... < a_n$. Set $a_{j+\frac{1}{2}} = \frac{a_j + a_{j+1}}{2}$ and $a_{j-\frac{1}{2}} := \frac{a_j + a_{j-1}}{2}, j = 2, ...n-1$, $a_{\frac{1}{2}} = -\infty, a_{n+\frac{1}{2}} = \infty$. Denote CDF function of Gaussian distribution as $\Phi(\gamma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma} \exp(-\frac{\tau^2}{2}) d\tau$. The elements for performing Newton's method (i.e. Equation (6.4) with 1-dimensional standard Normal distribution and L2 norm distortion function are calculated with basic calculus as follows [45],

$$\frac{\partial D_n}{\partial a_i}(\alpha) = a_i\big(\Phi(a_{i+\frac{1}{2}}) - \Phi(a_{i-\frac{1}{2}})\big) + \frac{\exp\big(\frac{-a_{i+\frac{1}{2}}^2}{2}\big) - \exp\big(\frac{-a_{i-\frac{1}{2}}^2}{2}\big)}{\sqrt{2\pi}} \tag{C.1}$$

$$\frac{\partial^2 D_n}{\partial a_i \partial a_{i-1}}(\alpha) = -\frac{1}{4\sqrt{2\pi}}(a_i - a_{i-1}) \exp\big(-\frac{a_{i-\frac{1}{2}}^2}{2}\big) \tag{C.2}$$

$$\begin{aligned}\frac{\partial^2 D_n}{\partial a_i^2}(\alpha) = \Phi(a_{i+\frac{1}{2}}) - \Phi(a_{i-\frac{1}{2}}) &- \frac{1}{4\sqrt{2\pi}}(a_{i+1} - a_i) \exp\big(-\frac{a_{i+\frac{1}{2}}^2}{2}\big) \\ &- \frac{1}{4\sqrt{2\pi}}(a_i - a_{i-1}) \exp\big(-\frac{a_{i-\frac{1}{2}}^2}{2}\big)\end{aligned} \tag{C.3}$$

$$\frac{\partial^2 D_n}{\partial a_i \partial a_{i+1}}(\alpha) = -\frac{1}{4\sqrt{2\pi}}(a_{i+1} - a_i) \exp\big(-\frac{a_{i+\frac{1}{2}}^2}{2}\big) \tag{C.4}$$

doi:10.6342/NTU202303932

REFERENCES

[1] B. Sullivan and A. Kaszynski, "PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)," *Journal of Open Source Software*, vol. 4, no. 37, p. 1450, May 2019. Online Available at: https://doi.org/10.21105/joss.01450

[2] P. Sturm, "Pinhole camera model," 2014.

[3] Y. Liu, B. Han, F. Qian, A. Narayanan, and Z.-L. Zhang, "Vues: practical mobile volumetric video streaming through multiview transcoding," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 514–527.

[4] S. Mystakidis, "Metaverse," *Encyclopedia*, vol. 2, no. 1, pp. 486–497, 2022.

[5] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, "Point cloud video streaming: Challenges and solutions," *IEEE Network*, vol. 35, no. 5, pp. 202–209, 2021.

[6] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[7] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. M. Kim, "Groot: a real-time streaming system of high-fidelity volumetric videos," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

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

[9] S. Gül, D. Podborski, T. Buchholz, T. Schierl, and C. Hellge, "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*, 2020, pp. 27–33.

[10] "Google draco," https://google.github.io/draco/.

[11] W. R. Mark, L. McMillan, and G. Bishop, "Post-rendering 3d warping," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 7–ff.

[12] C. Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Stereoscopic displays and virtual reality systems XI*, vol. 5291. SPIE, 2004, pp. 93–104.

doi:10.6342/NTU202303932

[13] W.-Y. Chen, Y.-L. Chang, S.-F. Lin, L.-F. Ding, and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *2005 IEEE International Conference on Multimedia and Expo*. IEEE, 2005, pp. 1314–1317.

[14] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453–465, 2011.

[15] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint tv," *IEEE signal processing magazine*, vol. 28, no. 1, pp. 67–76, 2010.

[16] A. Smolic, "3d video and free viewpoint video—from capture to display," *Pattern recognition*, vol. 44, no. 9, pp. 1958–1968, 2011.

[17] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3d warping using depth information for ftv," *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 65–72, 2009.

[18] I. Daribo and H. Saito, "A novel inpainting-based layered depth video for 3dtv," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 533–541, 2011.

[19] H.-Y. Huang and S.-Y. Huang, "Fast hole filling for view synthesis in free viewpoint video," *Electronics*, vol. 9, no. 6, p. 906, 2020.

[20] S. Fachada, D. Bonatto, M. Teratani, and G. Lafruit, "View synthesis tool for vr immersive video," 2022.

[21] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[22] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.

[23] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[24] S. Li, K. Wang, Y. Gao, X. Cai, and M. Ye, "Geometric warping error aware cnn for dibr oriented view synthesis," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1512–1521.

[25] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 322–337.

[26] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 286–301.

doi:10.6342/NTU202303932

[27] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[28] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, "Ibrnet: Learning multi-view image-based rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.

[29] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 1–6.

[30] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[31] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, "360 innovations for panoramic video streaming," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, 2017, pp. 50–56.

[32] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[33] S. Sengupta, N. Ganguly, S. Chakraborty, and P. De, "Hotdash: Hotspot aware adaptive video streaming using deep reinforcement learning," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 165–175.

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

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

[36] B. Han, Y. Liu, and F. Qian, "Vivo: Visibility-aware mobile volumetric video streaming," in *Proceedings of the 26th annual international conference on mobile computing and networking*, 2020, pp. 1–13.

[37] S. Gül, S. Bosse, D. Podborski, T. Schierl, and C. Hellge, "Kalman filter-based head motion prediction for cloud-based mixed reality," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3632–3641. Online Available at: https://doi.org/10.1145/3394171.3413699

[38] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 315–323.

doi:10.6342/NTU202303932

[39] Z. Xu, X. Zhang, K. Zhang, and Z. Guo, "Probabilistic viewport adaptive streaming for 360-degree videos," in *2018 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[40] J. Zou, C. Li, C. Liu, Q. Yang, H. Xiong, and E. Steinbach, "Probabilistic tile visibility-based server-side rate adaptation for adaptive 360-degree video streaming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 161–176, 2019.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[42] N. Somraj, "Pose-warping for view synthesis / DIBR," 2020. Online Available at: https://github.com/NagabhushanSN95/Pose-Warping

[43] M. Krivokuca, P. A. Chou, and P. Savill, "8i voxelized surface light field (8ivslf) dataset," *ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914*, 2018.

[44] S. Graf and H. Luschgy, *Foundations of quantization for probability distributions*. Springer, 2007.

[45] G. Pagès and J. Printems, "Optimal quadratic quantization for numerics: the gaussian case," 2003.

[46] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 139–149, 1982.

[47] H. Luschgy and G. Pagès, "Greedy vector quantization," *Journal of Approximation Theory*, vol. 198, pp. 111–131, 2015.

[48] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inf. Theory*, vol. 6, pp. 7–12, 1960.

[49] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[50] T. Kohonen, *Self-organization and associative memory*. Springer Science & Business Media, 2012, vol. 8.

[51] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[52] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

doi:10.6342/NTU202303932