Term_project

作者為 元駿 孫

提交日期: 2022年01月03日 01:36上午 (UTC+0800)
作業提交代碼: 1736899997
文檔名稱: RD_model.pdf (7.85M)
文字總數: 2451
字符總數: 13316

Rate-Distortion Modeling of Synthesizer Immersive Video for 6DoF Interaction

Yuan Jun Sun
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Recently, Virtual Reality (VR) becomes increasingly more popular. It enables a wide array of novel applications in many domains, such as video streaming, computer games, occupational training, healthcare, manufacturing, etc. The market research also reports that foresee explosive growth of the VR market in the upcoming years [1]. More and more companies devote their effort to the VR industry such as Meta [2] or Google [3], [4].

One way to classify the VR applications is through the different *interaction techniques*, including *3DoF* and *6DoF* interactions. Because of the high popularity of 360° video streaming services, most users are familiar with the *3DoF* (Degree-of-Freedom) interactions, in which a user's viewport is determined by his/her head/HMD *orientation*. In the 3DoF VR application, the user's position in his/her coordinate such as standing up and walking around, his/her HMD viewport would not reflect the changes of positions. Therefore, the user will *not* feel he/she is moving in the virtual world, leading to an inferior *immersive* user experience. In *6DoF* interactions, the application will render the viewports depending on the user position and orientation. Different from the 3DoF interaction, the 6DoF interaction allows users to walk around the virtual world which optimizes the immersive user experience. Fig. 1 illustrates the difference between 3DoF and 6DoF interactions.

Supporting 6DoF Extended Reality (XR) using 360° videos is not an easy task, because, for every single position, a new 360° video needs to be captured. Even if we deploy dense 360° cameras, users may still miss smooth transitions at the positions between any two adjacent cameras. Hence, more descriptive 3D representations are required for enabling the truly immersive experience of 6DoF VR applications. Recently, MPEG-I (Moving Picture Expert Group - Immersive Group) has been actively developing MPEG Immersive Video (MIV) standard [5], [6] which can use for 6DoF video compression. It uses multi-view RGB-D video as the data representation and includes the integrated pipeline for encoding, decoding, synthesizing, and rendering. The Test Model for Immersive Video (TMIV) [7], [8], which is the reference software of MIV standard, has been released to show a reference implementation of MIV.

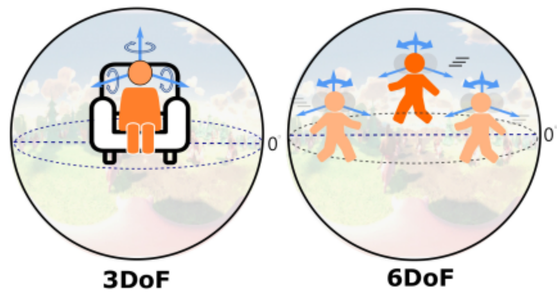


Fig. 1. Difference between 3DoF and 6DoF interactions.

Besides, reducing bandwidth and maintaining high view quality is a bottleneck of 6DoF real-time streaming. Because of the limitation of bandwidth, performing the best quality by adaptive quality model for 6DoF immersive video is quite a challenging work. There are many existing rate control algorithms for video compression [9] [10] [11]. Based on the existed rate control algorithms, it is possible to optimize those models to predict the 6DoF video performance. Although, there are still many effects that could impact the 6DoF video quality, e.g., tile sizes [12], camera placements, complexity of scenes [13], number of groups, synthesizer, and a quantization parameter. In this paper, we conduct a rate-distortion model (RD-model) to predict the performance of 6DoF immersive video in common situations as we show in Fig. 2. The model will design in an empirical way and aim at some vital TMIV parameters. The main goal of this model is to generate relevance between TMIV parameters and quality metrics, e.g., PSRN, SSIM, and VMAF. Once the model generates, it is possible to use on estimate the 6DoF immersive video performance in different camera placements and also use it on real-time 6DoF streaming.

II. Related Work

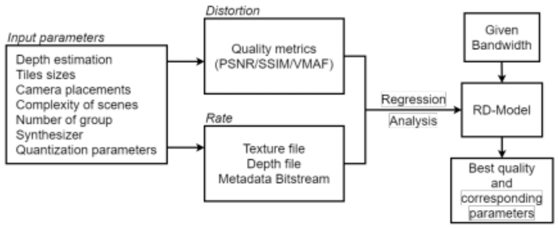


Fig. 2. RD-model workflow

A. 2D videos

B. 360 videos

C. Synthesized video

Yangang Cai [14] compressed the depth map by AVC, HEVC, and AVS3. Their results show that using the AVS3 encoder to compress the depth maps can provide better virtual view performance and less bitrate. Basel Salahieh [13] evaluate the performance of the object-based solution. Their results show the pixel rate saving and bitrate distortion in the object-based situation. Xavier Corbillon [15] implemented a 6DoF VR application with a multi-camera system and analyzed two extreme optimal algorithms. Their results show that tiling is able to improve the service performance and the high cost for the proactive optimizing strategies By the previous experiments, both 6DoF video quality and bitrate have a negative correlation with QP.

III. Data collection

A. Implementations

The purpose of collecting our own dataset is to investigate the impacts of diverse settings on the synthesized target views. This dataset has to cover a wide variety of usage scenarios with: (i) different camera placement strategies, (ii) random target view trajectories mimicking real HMD users, (iii) scenes with diverse characteristics (e.g., lighting conditions, color tone, and dynamics), we capture the source views by extending AirSim [16], which is an open source project that provides Application Programming Interface (API) for programmers. For example, we develop tools using camera control API to capture source views. Moreover, we adopt the random waypoint as the mobility model to generate random target view trajectories. In addition to random trajectories, we also implement tools to collect real HMD user trajectories using Unreal Engine. In a pilot test, we recruit three HMD users, play a random scene to them, and analyze their target view trajectories. We found that the dynamic ranges of roll/pitch/yaw are 80/80/100 degrees, and the angular velocity is between 8 and 80 degree/s. We also found the HMD users moves at a speed between 0.1 and 1 m/s. We use these statistics to generate random target view trajectories.

B. The Dataset

We select five scenes from Unreal Engine marketplace [17], and modify one of them into a dynamic scene. Fig. 3 gives sample video frame of these scenes (except the dynamic one). Table I summarizes the scenes with key characteristics. The synthesized target views from these scenes exhibit different complexities in terms of Temporal Information (TI) and Spatial Information (SI). The dynamic scene, *RealD*, is generated by adding a howling wolf and a rolling ball to *Real*. For each scene, we manually choose direction that have the richest set of visual features. We then place 36 cameras with one of the four placements: 6X6, 9X4, 12X3, and 18X2. Following MPEG's recommendations, all cameras and trajectories are confined in a $0.35 \times 1 \times 1\ m^3$ bounding box. We set the resolution and Field-of-View (FoV) of each camera to be 1280×720 and 90°, respectively. We generate 10 target view trajectories: t_1 to t_{10} using the random waypoint model. We also selected a real HMD user's trajectory from our pilot test that covers the largest surface of the scene. We refer to it as u_1. Each trajectory contains 90 samples of positions and orientations at 30 Hz.

TABLE I
SCENES IN OUR DATASET

Scene	# Obj.	# Mesh.	Space	Lighting	Color Tone	TI	SI
Light	52	51.3 K	Narrow	Bright	Warm	19.1	35.0
Arch	282	5.5 M	Wide	Bright	Warm	27.1	57.4
Xoio	125	2.8 M	Wide	Bright	Cold	26.2	57.8
Office	96	100.4 K	Narrow	Dark	Cold	29.6	62.2
Real	352	221.1 K	Narrow	Medium	Warm	34.7	66.7

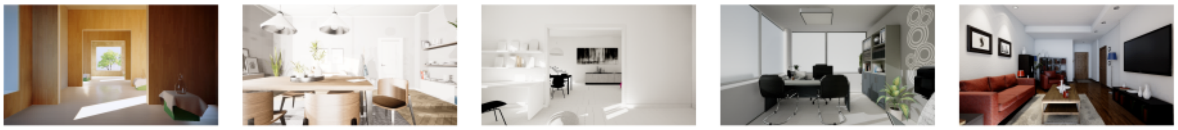
One last complication is the different coordinate systems used by: (i) Unreal Engine (North-Eastern-up, left-handed), (ii) AirSim (North-Eastern-down, right-handed), and (iii) TMIV (North-Western-up, right-handed). We have implemented scripts to convert the coordinate systems. For each combination of the scene and camera placement (trajectories), we capture RGBD videos from individual cameras, which are *source views*. We also captured the RGBD video clips following individual target view trajectories, which are *target views*, or ground truth. Both source and target views are stored as raw videos. In summary, our dataset contains: (i) source view placements (trajectories), (ii) target view (trajectories), (iii) source views, and (iv) target views. We plan to make our tools and sample dataset public.

IV. MPEG Immersive Video Standard

In this section, we briefly introduce the workflow and components of MIV codec [5], [7].

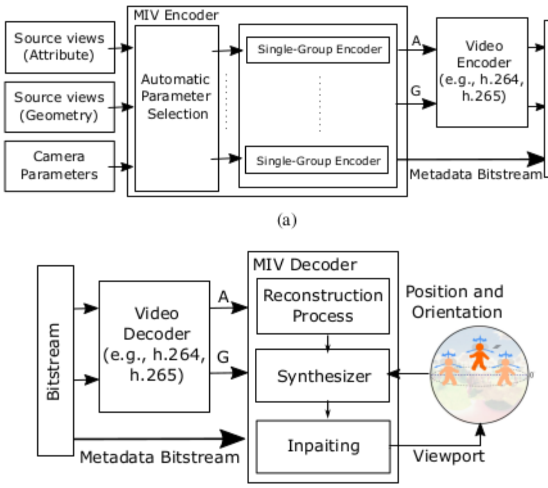
Fig. 4(a) shows the high-level workflow of MIV encoder. The inputs of MIV encoder are *source views*. Each source view is composed of attribute (texture) videos, geometric (depth) videos, and camera parameters. MIV encoder do the following process to compress source views:

- **Automatic parameter selection.** MIV encoder automatically calculate the parameters for compression, e.g., assessing geometric video quality, splitting source views into multiple group according to configuration, and labeling source views in each group.
- **Single-group encoders.** MIV encoder encodes each group of source views separately. In each group, the en-



(a) (b) (c) (d) (e)

Fig. 3. The considered scenes sorted in increasing complexity levels: (a) *Light*, (b) *Arch*, (c) *Xoio*, (d) *Office*, and (e) *Real*.



(a)

(b)

Fig. 4. The high-level overview of process flow of TMIV: (a) Encoder and (b) Decoder.

Fig. 5. The example of atlases. The left picture is attribute atlas, and the right picture is geometric altas.

coder chooses several views as the basic view according to the label of source view, and remove the duplicate area in other source views. The basic view and remaining area of other views are packed into rectangle video frames, which are called *atlases*. Fig. 5 show the example of atlases.

The outputs of MIV encoder are attribute atlases, geometric atlases, and metadata bitstream. The atlases are further compressed by video codec, and multiplexed with metadata bitstream as a single bitstream.

Fig. 4(b) shows the high-level workflow of MIV decoder. The inputs pf MIV decoder is the bitstream contains atlases bitstream, and metadata bitstream. The video decoder first be employed to decompress attribute atlases and geometric atlases. After that MIV decoder do the following process to decompress altases and synthesize the user's viewport.

- **Reconstruction process.** The MIV decoder reconstruct the source view by using the data in atlases.
- **Synthesizer.** The MIV decoder employ view synthesis techniques to synthesize the user's viewport according to user's position and orientation. Specifically, the synthesizer warp the pixel of each source view to user's viewport according to depth information, and blending the pixel values from each source views.
- **Inpainting.** After synthesis, the synthesized result may contain holes without information. The inpainting process uses the information from neighbor pixels to calculate pixel value for holes.

The outputs of MIV decoder are user's viewport synthesized according to user's position and orientation.

V. RD-Modeling

The target of our model is to predict the objective video quality which we choose VMAF as our main quality metric. VMAF is a quality metric developed by Netflix. It predicted the video quality by existing image metrics and some other features, such as Visual Information Fidelity (VIF), Detail Loss Metric (DLM), and Mean Co-Located Pixel Difference (MCPD).

We vary the following inputs of the model in our experiments:

- **QP** trades off the bitrate and RGBD video quality. MPEG group suggests using a smaller QP for depth video (80% of the corresponding RGB videos) because depth imposes significant impacts on synthesized quality [18]. We set (RGB video) QP $\in \{20, 36, 44, 48, 50\}$, targeting 5 to 50 Mbps total bitrates.
- **Target view trajectory** includes: (i) yaw, roll, and pitch, which specify the user *orientation* and (ii) surge, heave, and sway, which specify the user *position*.

References

[1] (2018) Augmented reality and virtual reality market. [Online]. Available: https://reurl.cc/WLaNp9

[2] Meta. (2021) Meta. [Online]. Available: https://about.facebook.com/meta/
[3] Google. (2021) Google ar and vr. [Online]. Available: https://arvr.google.com/
[4] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. DuVall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec, "Immersive light field video with a layered mesh representation," vol. 39, no. 4, pp. 86:1–86:15, 2020.
[5] J. M. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, "Mpeg immersive video coding standard," *Proceedings of the IEEE*, 2021.
[6] MPEG. (2021) Mpeg immersive video. [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-i/immersive-video
[7] B. Salahieh, J. Jung, and A. Dziembowski, "Test Model 10 for MPEG Immersive Video," International Organization for Standardization Meeting Document ISO/IEC JTC1/SC29/WG04 N0112, 2021.
[8] MPEG. (2019) The gitlab of mpeg test model for immersive video. [Online]. Available: https://gitlab.com/mpeg-i-visual/tmiv/-/tree/v10.0.1
[9] M. A. Papadopoulos, F. Zhang, D. Agrafiotis, and D. Bull, "An adaptive qp offset determination method for hevc," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 4220–4224.
[10] Z. Wu, H. Yu, B. Tang, and C. W. Chen, "Adaptive initial quantization parameter determination for h.264/avc video transcoding," *IEEE Transactions on Broadcasting*, vol. 58, no. 2, pp. 277–284, 2012.
[11] Z. He and S. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1221–1236, 2001.
[12] J.-B. Jeong, S. Lee, I.-W. Ryu, T. T. Le, and E.-S. Ryu, "Towards viewport-dependent 6dof 360 video tiled streaming for virtual reality systems," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3687–3695. [Online]. Available: https://doi.org/10.1145/3394171.3413712
[13] B. Salahieh, W. Cochran, and J. Boyce, "Delivering object-based immersive video experiences," pp. 103–1–103–8(8), 2021.
[14] Y. Cai, R. Wang, K. Qiu, R. Peng, Z. Cheng, and Q. Wang, "Depth map video compression performance evaluation for ieee 1857.9," in *2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2021, pp. 1–6.
[15] X. Corbillon, F. De Simone, G. Simon, and P. Frossard, "Dynamic adaptive streaming for multi-viewpoint omnidirectional videos," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 237–249. [Online]. Available: https://doi.org/10.1145/3204949.3204968
[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*. Springer, November 2018, pp. 621–635.
[17] Epic Games. (2021) Unreal engine marketplace. [Online]. Available: https://www.unrealengine.com/marketplace/en-US/store
[18] J. Jung, B. Kroon, and J. Boyce, "Common test conditions for mpeg immersive video," *ISO/IEC JTC 1/SC 29/WG 11 N19484*, 2020.

Term_project

原創性報告

19%	16%	16%	10%
相似度指數	網際網絡來源	出版物	學生文稿

主要來源

1	icitst.org 網際網絡來源	3%
2	trepo.tuni.fi 網際網絡來源	2%
3	koreascience.or.kr 網際網絡來源	1%
4	etd.lib.nsysu.edu.tw 網際網絡來源	1%
5	www4.comp.polyu.edu.hk 網際網絡來源	1%
6	Daniel J Ringis, Francois Pitie, Anil Kokaram. "Near Optimal Per-Clip Lagrangian Multiplier Prediction in HEVC", 2021 Picture Coding Symposium (PCS), 2021 出版物	1%
7	Submitted to Texas A&M University, College Station 學生文稿	1%
8	www.fruct.org 網際網絡來源	1%

9	Patrick Garus, Felix Henry, Joel Jung, Thomas Maugey, Christine Guillemot. "Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?", IEEE Transactions on Circuits and Systems for Video Technology, 2021 出版物	1%
10	Chen Zhang, Qiang Cao, Jie Yao, Changsheng Xie. "VRefine: Refining Massive Surveillance Videos for Efficient Store and Fast Analyzing", 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2021 出版物	1%
11	Jin Young Lee, Sang-hyo Park. "Adaptive fractional motion and disparity estimation skipping in MV-HEVC", Journal of Visual Communication and Image Representation, 2021 出版物	1%
12	www.dis.cwi.nl 網際網絡來源	1%
13	www.sciencegate.app 網際網絡來源	1%
14	Submitted to University College Falmouth 學生文稿	1%

15	Tuan Thanh Le, Jong-Beom Jeong, SangSoon Lee, Jaehyoun Kim, Eun-Seok Ryu. "An Efficient Viewport-Dependent 360 VR System Based on Adaptive Tiled Streaming", Computers, Materials & Continua, 2021 出版物	<1 %
16	Basel Salahieh, Mengyu Chen, Jill Boyce. "An Overview of MPEG Immersive Video", OSA Imaging and Applied Optics Congress 2021 (3D, COSI, DH, ISA, pcAOP), 2021 出版物	<1 %
17	Submitted to Cardiff University 學生文稿	<1 %
18	Catarina Brites, Joao Ascenso, Fernando Pereira. "Lenslet Light Field Image Coding: Classifying, Reviewing and Evaluating", IEEE Transactions on Circuits and Systems for Video Technology, 2020 出版物	<1 %
19	Hong-Chang Shin, Jun-Young Jeong, Gwangsoon Lee, Muhammad Umer Kakli, Junyoung Yun, Jeongil Seo. "Enhanced pruning algorithm for improving visual quality in MPEG immersive video", ETRI Journal, 2021 出版物	<1 %
20	Submitted to University of Lincoln 學生文稿	<1 %

21	Yangang Cai, Ronggang Wang, Ke Qiu, Rui Peng, Zhipeng Cheng, Qi Wang. "Depth Map Video Compression Performance Evaluation For Ieee 1857.9", 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2021 出版物		<1 %
22	doi.org 網際網絡來源		<1 %

排除引述	關閉	排除相符處	關閉
排除參考書目	關閉		