Demo Abstract: Pixel Similarity-Based Content Reuse in Edge-Assisted Virtual Reality

Ying Chen*, Hazer Inaltekin†, Maria Gorlatova*
*Duke University, Durham, NC, †Macquarie University, North Ryde, NSW, Australia
*{ying.chen151, maria.gorlatova}@duke.edu, †hazer.inaltekin@mq.edu.au

***Abstract*—Offloading the computation-intensive virtual reality (VR) frame rendering to the edge server is a promising approach to providing immersive VR experiences in mobile VR devices with limited computational capability and battery lifetime. However, edge-assisted VR systems require the data delivery at a high data rate, which poses challenges to the wireless communication between the edge and the device. In this demo, to reduce the communication resource consumption, we present PixSimVR, a pixel similarity-based content reuse framework for edge-assisted VR. PixSimVR analyzes the similarity of the pixels across different VR frames that correspond to different *viewport poses*, i.e., users' points of view in the virtual world. Based on the pixel similarity level, PixSimVR adaptively splits the VR content into the foreground and the background, reusing the background that has a higher similarity level across frames. Our demo showcases how PixSimVR reduces bandwidth requirements by adaptive VR content reuse. Demo participants will develop an intuition for the potential of exploiting the correlation between VR frames corresponding to similar viewport poses specifically, and for the promises and the challenges of edge-assisted VR as a whole. This demonstration accompanies [1].**

I. Introduction

Immersive virtual reality (VR) systems are computation-intensive, where simulated visual content is constantly updated to remain consistent with the user's movements [2], [3]. An important research direction in VR is enabling high-quality immersive VR experiences, which require both a high frame rate and a high frame resolution, via generating VR frames on the cloud or edge and transmitting these frames to mobile VR devices. Such an approach, however, is challenging in facilitating the wireless data delivery at a high data rate. In this demo we reduce the communication resource consumption by exploiting the similarity between different VR frames.

Several recent studies [2], [3] show that exploiting these correlations by reusing the pixels can reduce bandwidth requirements of VR systems. Especially, the VR frames are highly correlated for similar *viewport poses*, namely the positions and orientations of users' VR headset devices. This is because each frame is generated for the specific viewport pose when the users explore virtual worlds. Exploiting inter-frame pixel similarity, these studies [2], [3] reuse similar VR content to cut down wireless network demand. However, the selection of the reused content is based on heuristics. Complementing this work, PixSimVR adaptively reuses the VR content based on the pixel similarity level that quantifies the similarity of pixels across VR frames.

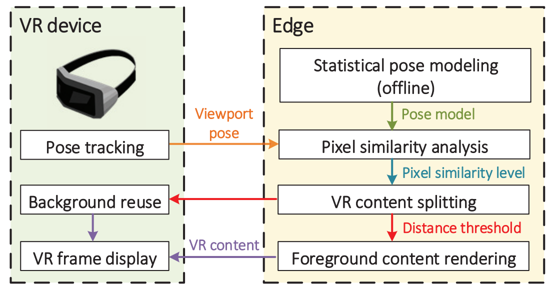

Fig. 1. PixSimVR system overview. The VR device sends the tracked pose to the edge. The edge splits the VR content into the background and the foreground based on the pixel similarity level. For displaying the VR frame corresponding to the tracked pose, the VR device receives the foreground rendered on the edge and reuses the background.

In this demo, we present PixSimVR, a **Pix**el **Sim**ilarity-based content reuse framework for edge-assisted **V**irtual **R**eality. The demo accompanies [1], in which we analytically modeled the statistical average of the inter-frame pixel similarity over different pose changes. PixSimVR reduces the resource consumption by rendering a set of 'reference' frames, and generating other, 'novel', frames by rendering only a portion of the frame while generating the rest by reusing the reference frame. PixSimVR first models the statistics of the viewport poses (both positions and orientations) offline with a pre-collected viewport pose dataset. PixSimVR then quantifies the similarity of pixels across VR frames, relating the poses of the reference and novel frames using the statistical pose model. Based on the pixel similarity level, PixSimVR adaptively splits the VR content of novel frames to the foreground and the background, where the foreground is rendered and transmitted from the edge and the background is reused. We showcase that PixSimVR can achieve higher image quality for VR frames while requiring fewer communication resources, compared to the baseline which sets a fixed cutoff value for the viewport-to-content distance to select the reused content.

II. System Design

The system overview of PixSimVR is shown in Fig. 1. We implement an edge computing-based architecture to render VR frames on the edge and wirelessly transmit the frames to the mobile VR device. To enable high resolutions and high frame rates, PixSimVR reduces the required wireless network bandwidth by adaptively splitting the VR frame into the dynamic 'foreground' and the static 'background' according to the pixel similarity, and reusing background pixels that are similar across VR frames.

1) The VR device: The VR device tracks the pose in real-time, transmits the pose to the edge, generates the background content of the novel frame via viewport projection [4] from the reference frame, and merges the background with the foreground received from the edge for the final display.

2) The PixSimVR edge: The edge is deployed on the PC which communicates with the VR device over a one-hop wireless local area network via an HTTP request using a UnityWebRequest. The edge in PixSimVR performs four tasks: (1) statistical pose modeling, (2) pixel similarity analysis, (3) VR content splitting, and (4) foreground content rendering.

Statistical pose modeling. Since the user's viewport poses (positions and orientations) determine the rendered VR frames, the edge first performs statistical pose modeling offline before analyzing the pixel similarity across VR frames. Based on a pre-collected dataset of VR viewport poses, PixSimVR obtains a statistical model for x, y, z coordinates of viewport positions, and for azimuth and polar angles of viewport orientations.

Pixel similarity analysis. This module analyzes the pixel similarity of two VR frames, the reference and novel frames. The analysis is based on the statistical pose model that characterizes the viewport change from the reference frame to the novel frame. This module outputs the pixel similarity level as a function of the viewport-to-content distance.

VR content splitting. To reduce the required bandwidth, PixSimVR renders the whole reference frames, while generating novel frames by rendering the foreground content and reusing the background content. In this module, for every consecutive R frames, the first frame is selected as the reference frame and other $R-1$ frames are the novel frames. The whole reference frame is classified as the foreground content and is rendered on the edge. Based on the pixel similarity level, PixSimVR adaptively sets a threshold for the viewport-to-content distance. In this way, PixSimVR splits the VR content of novel frames to the foreground and the background, where the background has a higher viewport-to-content distance and hence a higher pixel similarity level.

Foreground content rendering. The edge renders the foreground, that is, the whole reference frame and a portion of the novel frame that is classified as the foreground content. The rendered foreground content is then wirelessly transmitted to the VR device.

III. Demonstration

The demonstration follows the workflow shown in Fig. 1. The participants of our demo will develop an intuition for the correlation between VR frames corresponding to similar points of view of the user in virtual worlds, and will see how adaptive VR content reuse reduces bandwidth requirements in edge-assisted VR. A video of the demo is available online.[1]

The VR application is built using Unity 2019.2.14f and the open-source VR game Lite [5]. A Lenovo laptop with an AMD Ryzen 7 4700H CPU and an NVIDIA GTX 1660 Ti GPU serves as the edge and a Meta Quest 2 acts as the VR device.

[1]Link to the demo video: https://youtu.be/MK3w8jvgVNE

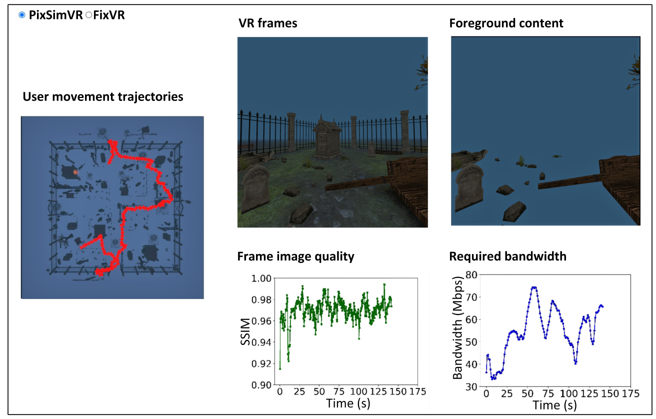

Fig. 2. A screenshot of PixSimVR in action. When exploring the virtual world, the demo participant can view the graphical content rendered using PixSimVR for the specific point of view.

Similar to the illustration in Fig. 2, we show a scenario where a user is exploring a virtual world. When the user moves around and looks in different directions within the virtual world, the trajectories of the user's position and orientation will be recorded and shown. In the meantime, the graphical content will be rendered for the user's specific points of view *at the full VR frame rate* of the Meta Quest 2. The participants can observe how the background VR content is correlated for similar points of view of the user. To reuse the background VR content, the participants can choose from two methods, the PixSimVR and the baseline FixVR method. PixSimVR adaptively reuses the VR content based on the pixel similarity level, while FixVR reuses the VR content farther than a fixed cutoff value from the user position. For both methods, the participants will view what VR content is classified as the dynamic foreground (which is wirelessly transmitted between the edge and the VR device). To demonstrate the performance of PixSimVR and FixVR, we will show real-time performance graphs that display the metrics of the graphical rendering quality (i.e., the structural similarity index measure (SSIM)) and the bandwidth requirement of the VR systems. PixSimVR ensures high SSIM (larger than 0.95 with 95.3% probability), outperforming FixVR by 3.2%.

Acknowledgments

This work is supported in part by NSF grants CSR-1903136, CNS-1908051, and CAREER-2046072, and by an IBM Faculty Award.

References

[1] Y. Chen, H. Kwon, H. Inaltekin, and M. Gorlatova, "VR viewport pose model for quantifying and exploiting frame correlations," in *Proc. IEEE INFOCOM*, 2022.

[2] Y. Li and W. Gao, "DeltaVR: Achieving high-performance mobile VR dynamics through pixel reuse," in *Proc. ACM/IEEE IPSN*, 2019.

[3] J. Meng, S. Paul, and Y. C. Hu, "Coterie: Exploiting frame similarity to enable high-quality multiplayer VR on commodity mobile devices," in *Proc. ACM ASPLOS*, 2020.

[4] Y. Liu, J. Liu, A. Argyriou, L. Wang, and Z. Xu, "Rendering-aware VR video caching over multi-cell MEC networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2728–2742, 2021.

[5] Xiaolianhua Studio. (2017) Lite. [Online]. Available: https://assetstore.unity.com/packages/3d/environments/fantasy/make-your-fantasy-game-lite-8312