**VR Viewport Pose Model for Quantifying and Exploiting Frame Correlations**

Ying Chen*, Hojung Kwon*, Hazer Inaltekin†, Maria Gorlatova*  
*Duke University, Durham, NC, †Macquarie University, North Ryde, NSW, Australia

**Abstract**—The importance of the dynamics of the viewport pose, i.e., the location and the orientation of users’ points of view, for virtual reality (VR) experiences calls for the development of VR viewport pose models. In this paper, informed by our experimental measurements of viewport trajectories across 3 different types of VR interfaces, we first develop a statistical model of viewport poses in VR environments. Based on the developed model, we examine the correlations between pixels in VR frames that correspond to different viewport poses, and obtain an analytical expression for the visibility similarity (ViS) of the pixels across different VR frames. We then propose a lightweight ViS-based ALG-ViS algorithm that adaptively splits VR frames into the background and the foreground, reusing the background across different frames. Our implementation of ALG-ViS in two Oculus Quest 2 rendering systems demonstrates ALG-ViS running in real time, supporting the full VR frame rate, and outperforming baselines on measures of frame quality and bandwidth consumption.

**Index Terms**—Virtual reality, pose model, frame correlation, game engine-based simulations

I. INTRODUCTION

Virtual reality (VR), which immerses users into computer-generated virtual environments [1], has been showing promise in many applications including gaming, education, and healthcare [2]. VR is expected to boost global GDP by $450 billion by 2030 [3]. High expectation for VR, coupled with its known resource-hungry nature [4], spurred a wide range of recent research that optimizes VR systems to reduce their communication and computing resource consumption [5]–[10].

A particular feature of VR is the tight coupling of user’s actions and the generated frames. In traditional visual media, the frames that are shown to the users are fixed. By contrast, in VR, to allow the users to independently explore virtual worlds, each frame is generated for the specific point of view of the user at a given time, i.e., for the specific viewport pose, namely the $x$, $y$, and $z$ coordinates, and polar and azimuth orientation angles $\theta$ and $\phi$, of user’s VR headset or another interface to the virtual world (mobile phone [5], [8], [11]; computer monitor [12]). Hence, the correlations between different VR frames, and the performance of approaches that exploit them to reduce resource consumption in VR [8]–[10], are intimately tied to the dynamics of user behavior within the VR experience. We examine and exploit this phenomenon in this work.

First, we develop a statistical model of users’ VR viewport pose, comprised of the models of pose components, orientation and position. To develop this model, we collected a dataset of VR viewport trajectories in 3 VR games and across 3 different types of VR user interfaces, with over 5.5 hours of user data in total.

To characterize the correlation of viewport orientations between VR frames that are $\Delta t$ seconds apart, we obtain models of the change of azimuth and polar angles over $\Delta t$. To characterize the displacement between VR viewport positions that are $\Delta t$ apart, we propose a modified random waypoint model (RWP) with random pause times (‘paused-MRWP’). We demonstrate a close fit of the developed VR viewport pose model to the experimental data. To the best of our knowledge, this is the first statistical model of viewport pose in VR.

Next, we apply the developed pose model to quantify the similarity of pixels across VR frames. For similar poses, the VR frames are highly redundant, as shown in Fig. 1. It is thus possible to reduce 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 via view projection [8], [13]. In this paper we derive analytical expressions for the visibility similarity (ViS) of pixels across different VR frames, relating the poses of the reference and novel frames through the developed viewport pose model, and accounting for the misalignments of the fields of view (FoVs) and the VR contents-to-viewpoint distance differences between the novel and the reference frames. We verify our analysis via Unity 3D [14] game engine-based simulations. Finally, we exploit the formulated ViS to adaptively divide VR frame contents into background and foreground, in order to render the foreground for the novel frames, while reusing the background. Separate treatment of background and foreground in VR frame generation has been considered in multiple lines of work [9]–[11], [15], [16], which use heuristics for this separation. In this work, we propose a lightweight algorithm, ALG-ViS, that uses the analytical ViS to adaptively determine the distance threshold beyond which the contents are treated as background. We incorporate the developed ALG-ViS in two rendering systems based on Oculus Quest 2 (also known as Meta Quest 2), one on-device and one supported by edge computing. In both systems, ALG-ViS...
runs in real time, supporting the full VR frame rate, and outperforming a set of baselines on measures of frame quality and resource consumption.

To summarize, the main contributions of this paper are: (i) the first statistical model of viewport pose in VR, (ii) the analysis of the visibility similarity between different VR frames, and (iii) the analytically grounded algorithm for determining which contents to reuse across different frames. We make the VR viewport pose dataset and our implementation codes publicly available via GitHub.

The rest of this paper is organized as follows. We review the related work in Section II and propose the viewport pose model in Section III and analyze the ViS and propose the ALG-ViS in Section IV. We present the evaluation in Section V and conclude the paper in Section VI.

II. RELATED WORK

Device pose modeling: VR frame generation requires information about the pose (position and orientation) of user’s point of view. The vast body of work that has, over the years, modeled human mobility in many different applications [17]–[19] focused on human positions but not orientations. Orientations of handheld mobile devices are starting to be modeled in context of visible light communications [20], [21]. We are unaware of existing statistical models of users’ viewport pose in VR. The position component of our developed model builds on the modified RWP proposed in [22], and one of the orientation components is related to the observations previously made in [23]. The comprehensive model we propose significantly modifies and extends these approaches.

Predicting VR viewport pose: Recently, several approaches that predict pose or its components in VR systems have been developed [11], [24]–[28]. Unfortunately, highly immersive VR experiences are known to be negatively affected by errors in the pose prediction [8]. Our statistical approach can be seen as making decisions based on the distribution of viewport poses, rather than the specific predicted pose. Our evaluation demonstrates that this approach improves image quality and bandwidth variability over prediction-based approaches.

Exploiting redundancy across VR frames: Multiple methods for reducing the required bandwidth and transmission latency in VR have been developed [5]–[10]. In a rich body of work [9], [11], [15], [16], a VR frame is classified into background that is relatively static across VR frames and foreground that is less similar from one frame to the next. The background can be rendered on the edge and prefetched by the VR device, while the foreground is rendered on the mobile device [11], [15], [16]; the background can also be reused across multiple frames [9], [10]. These studies use heuristics to separate the background and the foreground. Complementing this work, we derive an analytical expression for the inter-frame pixel similarity, which we use to split the background and the foreground adaptively, via a lightweight algorithm that can run on-device or on the edge server. Our evaluation demonstrates that our approach improves the VR image quality while consuming fewer resources.

1https://github.com/VRViewportPose/VRViewportPose.

III. VR VIEWPORT POSE MODEL

We introduce our collected dataset in Section III-A, describe our orientation model in Section III-B and our position model in Section III-C, and model the correlation between them in Section III-D.

A. Collected Dataset

To complement existing datasets of users’ head orientation in 360° videos [23], [29], [30] and a small-scale single-interface dataset of users’ head pose in untethered VR [31], we collected a dataset of users’ viewport pose in 3 different VR games listed in Table I across 3 different common VR user interface types. Specifically, we examined: (i) VR experienced through a VR headset and controlled through user head rotation and a VR controller (“headset VR”), (ii) “desktop VR” [12], experienced through the user’s desktop monitor and controlled through desktop’s mouse and keyboard, and (iii) VR experienced through a mobile phone, controlled via moving the phone and tapping on it [32]. Our institutional review board (IRB)-approved data collection, conducted under COVID-19 restrictions, involved remote desktop VR and phone-based VR data collection via apps that we distributed to remote users, and a small number of socially distanced in-lab experiments for headset and phone-based VR. In total, we recorded experiences of 5 users with headset and phone-based VR, and 20 users with desktop VR. For desktop and phone-based VR, each user explored the 3 VR games for 2–5 minutes (per game). For headset VR, the users explored each game for 2 minutes to avoid simulator sickness. Additional data collection protocol details and the dataset are provided via GitHub.

B. Orientation Model

A VR viewport is depicted in Fig. 2 where \( \mathbf{n} \) is the unit vector along the optical axis of the camera. We first introduce the representation for the viewport orientation using variables related to \( \mathbf{n} \), followed by the definition of the statistical viewport orientation model.

Definition 1 (Viewport orientation representation). The viewport orientation is the tuple \((\theta, \phi)\), where polar angle \(\theta \in [0, \pi]\) is the angle between \(\mathbf{n}\) and the positive direction of the Y-axis, and the azimuth angle \(\phi \in [-\pi, \pi]\) is the angle between the projection of \(\mathbf{n}\) in the XZ-plane and the positive direction of the X-axis in the Earth coordinates XYZ. \(\theta\) and \(\phi\) characterize how users look vertically and horizontally.

Definition 2 (Viewport orientation model). The viewport orientation model is the tuple \( (p_{\theta}(\theta), p_{\Delta\theta}(\Delta\theta), p_{\Delta\phi}(\Delta\phi)) \), where \(p_{\theta}(\theta)\), \(p_{\Delta\theta}(\Delta\theta)\), and \(p_{\Delta\phi}(\Delta\phi)\) are the probability density functions of the rotation angle \(\theta\), the change in the rotation angle \(\Delta\theta\), and the change in the azimuth angle \(\Delta\phi\).
functions (PDFs) of the polar angle $\theta$, the polar angle change $\Delta \theta \in [-\pi, \pi]$ over the time interval $\Delta t$, and the azimuth angle change $\Delta \phi \in [-\pi, \pi]$ over $\Delta t$. In this paper we assume that $p_0(\theta)$, $p_{\Delta \theta}(\Delta \theta)$, and $p_{\Delta \phi}(\Delta \phi)$ are independent.

$\Delta \theta$ is given by $\Delta \theta = \theta_{\text{nov}} - \theta_{\text{ref}}$, where $\theta_{\text{ref}}$ and $\theta_{\text{nov}}$ are the polar angles of the reference and novel frames taken $\Delta t$ seconds apart. $\Delta \phi$ is calculated as $\Delta \phi = -\pi + \text{mod} \left(\phi_{\text{nov}} - \phi_{\text{ref}} + \pi, 2\pi\right)$, where $\text{mod}(a, b) = a - b \left\lfloor \frac{a}{b} \right\rfloor$ and $\lfloor \cdot \rfloor$ is the floor function, and $\phi_{\text{nov}}$ and $\phi_{\text{ref}}$ are the azimuth angles of the reference and novel frames taken $\Delta t$ seconds apart.

We examine azimuth angle change $\Delta \phi$ rather than $\phi$ itself because $\phi$ can be assumed to be uniformly distributed when VR contents are scattered along different longitudes in the VR systems and there are no viewing preferences. Hence, the distribution of $\phi$ does not provide information about the correlation between different VR frames.

1) Distribution fit: We evaluate the distribution fit for the experimental measurements of $\theta$, $\Delta \theta$, and $\Delta \phi$. In analyzing $p_0(\theta)$ and $p_{\Delta \theta}(\Delta \theta)$, we fit the experimental data to a set of common statistical distributions. In analyzing $p_{\Delta \phi}(\Delta \phi)$, the PDFs of which have irregular shapes for some values of $\Delta t$, we fit the experimental data to the set of common distributions and mixed distributions of two different common distributions. As the error metric, we use the sum of squared errors (SSE) \[2\] between the data and the fitted distribution.

2) Statistical distribution of the polar angle $\theta$: We found Laplace distributions, with means close to $90^\circ$ (89.0°–92.3°) and scales ranging from 3.5 to 7.4, to best fit experimental data (see a summary in Table[1] and a fit example in Fig.3). This is intuitive: it corresponds to humans having a bias for looking straight ahead, to the central parts of VR contents, without frequently tilting their heads. Among the 3 VR games, the scale values are the largest for VR (4.3–7.4), which has more contents scattered along different latitudes than the other games. Among the 3 interface types, the scale values are the smallest for headset VR (3.5–4.3), corresponding to users looking straight ahead rather than up and down. We hypothesize that this is due to the discomfort associated with tilting the head drastically while wearing a headset.

3) Statistical models for polar angle change $\Delta \theta$: The experimental distributions of $\Delta \theta$ for different $\Delta t$ values closely fit zero-mean Laplace distributions. The scales $b_{1,0}$ of the Laplace distributions that yield the best fit for different $\Delta t$ values in Lite are shown in Table[11] As $\Delta t$ increases, the correlation between polar angles decreases, leading to the increase of the scale with $\Delta t$ (e.g., from 0.305 for $\Delta t = 5/60$ s to 3.407 for $\Delta t = 100/60$ s for desktop VR). Among the 3 interface types, headset VR has the largest $b_{1,0}$ when $\Delta t$ is small (e.g., 1.92 vs. 1.48 and 1.41 for $\Delta t = 30/60$ s), indicating that the polar angle changes more rapidly. This is due to the ease of changing viewpoint orientation over a small time interval in headset VR.

4) Statistical models for azimuth angle change $\Delta \phi$: In our examinations, for some $\Delta t$ the distributions of $\Delta \phi$ appeared to have canonical shapes, while for others they appeared as a mixture of distributions. Thus we fit the experimental data to both common and mixed distributions. We present the best distribution fits and their parameters, for a subset of $\Delta t$ values, in desktop VR for all 3 games jointly, in Table[IV]. For the cases of mixed distributions, logistic and Laplace in these examples, the PDF of the mixed distribution is written as

$$f_{\text{mixed}}(x) = p_1 \frac{\exp \left(-\frac{|x| - \mu_1}{b_1}\right)}{1 + \exp \left(-\frac{x - \mu_1}{b_1}\right)} + (1 - p_1) \frac{\exp \left(-\frac{|x| - \mu_2}{b_2}\right)}{1 + \exp \left(-\frac{x - \mu_2}{b_2}\right)},$$

where $\mu_1$ and $b_1$ are the mean and the scale of the Laplace distribution, $\mu_1$, $b_1$, $\mu_2$, and $b_2$ are the mean and the scale of the logistic distribution, and $p_1$ is used to alter the fractions of the logistic and the Laplace distributions.

The best distribution fit for $\Delta \phi$ changes with $\Delta t$. When $\Delta t$ is small (i.e., when $\Delta t < \beta_1$), $\Delta \phi$ is best modeled by a Laplace distribution with a relatively small scale. When $\beta_1 \leq \Delta t < \beta_2$, $\Delta \phi$ is best modeled by a mixture of logistic and Laplace distributions, corresponding to users’ tendency to change their head orientations only slightly over these time intervals (i.e., $-15^\circ < \Delta \phi < 15^\circ$). Finally, when $\Delta t \geq \beta_2$, the individual angle observations become uncorrelated and are best modeled by a uniform distribution $\Delta \phi(-180^\circ, 180^\circ)$. From the collected desktop VR pose data, we obtain $\beta_1 = 189/60$ s and $\beta_2 = 1549/60$ s. Examples of these three cases are shown in Fig.4 For the other 2 VR interfaces, we observe similar patterns, but $\beta_1$ and $\beta_2$ are different: $\beta_1 = 244/60$ s and $\beta_2 = 1003/60$ s for headset VR, $\beta_1 = 496/60$ s and $\beta_2 = 1006/60$ s for phone-based VR.

C. Position Model

1) MRWP model with random pause times: In this section we introduce our model for VR viewpoint position. We focus on viewpoint position change over time interval $\Delta t$, in order to analyze the ViS of VR frames that are $\Delta t$ apart.

Adopting the axis notation common in computer graphics \[14, 37\], the viewpoint positions in the Earth coordinates $XYZ$ (shown in Fig.2) are denoted as $(x, y, z)$, where $y$ is the height of the viewpoint, and $x$ and $z$ are the coordinates of the viewpoint positions in the $XZ$-plane (i.e., the ground plane). We assume that $y$ is constant, e.g., $y$ can be set to the human eye level. While changing $y$ is important in some specific contexts, such as exergames \[48, 39\], $y$ is fixed in the vast majority of typical VR experiences, and in native Oculus Integration app development, to avoid disorienting the users when they sink below or float above the ground in the virtual environment \[40\].

To model the change of $x$ and $z$, we propose a paused-MRWP position model in the infinite plane, based on our collected pose data and the modified RWP \[22\]. The model consists of an infinite sequence of points $W_n, n \in \mathbb{N}^+$, called waypoints, the pause time $S_n$ at each waypoint, the duration $T_n$ to move along a straight line from $W_n$ to $W_{n+1}$ with a constant velocity $v$, and the included angle $\alpha_n$ between $W_nW_{n+1}$ and the abscissa. The waypoint is expressed as
TABLE II: Mean and scale of the Laplace distribution that has the smallest SSE with the experimental measurements of polar angle $\theta$, for different interface types and VR games.

| Interface type | Desktop VR | Headset VR | Phone-based VR |
|---------------|------------|------------|----------------|
| VR game       | VK         | Lite       | Office         | VK             | Lite       | Office       |
| Mean          | 90.575     | 90.034     | 89.799         | 92.331         | 89.654     | 88.999       |
| Scale         | 7.356      | 6.057      | 6.204          | 4.319          | 3.454      | 3.646        |

TABLE III: Scale of best-fitting Laplace distributions for polar angle change $\Delta \theta$, in Lite for different interface types.

| Interface type | Desktop VR | Headset VR | Phone-based VR |
|---------------|------------|------------|----------------|
| $\Delta t$ (1/60 s) |          |            |                |
|                | 10         | 60         | 200            |
|                | 200        | 500        | 1000           |
|                |            |            |                |
|                | 1.049      | 1.276      | 1.480          |
|                | 3.407      | 4.984      | 8.005          |
|                | 9.926      | 6.797      | 6.195          |

Fig. 3: Experimentally obtained $\theta$ (desktop VR, Lite) is Laplace distributed.

Fig. 4: The distribution of the experimental data for azimuth change $\Delta \phi$ after $\Delta t$, and the best distribution fit among common distributions and mixed distributions.

Fig. 5: The collected trajectory for one user in Lite and the extracted flights.

Fig. 6: CDF of the flight time for collected samples, paused-MRWP, and classical RWP.

Fig. 7: The pause time for all users and all games.

Fig. 8: The analytical and empirical results of the $k$-th moment of position displacement.

TABLE IV: Distributions that best fit the experimental azimuth angle change $\Delta \phi$ over $\Delta t$, and the SSE between the experimental and the fitted distributions.

| $\Delta t$ (s) | Distribution | Best fit Parameters | SSE |
|---------------|--------------|---------------------|-----|
| 10/60         | Laplace      | $\mu_1 = 0.0, b_1 = 3.130$ | $1.465 \times 10^{-3}$ |
| 60/60         | Laplace      | $\mu_2 = 0.0, b_2 = 0.1806$ | $7.016 \times 10^{-3}$ |
| 200/60        | Logistic+    | $\mu_{10} = 0.1, b_{10} = 28.54$ | $3.650 \times 10^{-3}$ |
| 500/60        | Laplace      | $\mu_1 = 0.1, b_1 = 0.24$ | $0.36$ |
| 2000/60       | Uniform      | $\Delta \phi \sim U(-180^\circ, 180^\circ)$ | $1.739 \times 10^{-4}$ |

$W_n = (x_n, z_n)$ where $x_n$ and $z_n$ are the coordinates of waypoints in the $XZ$-plane. The vector $W_nW_{n+1}$ is called the $n$-th flight, and $\alpha_n$ is called the direction of the $n$-th flight. At time 0, the viewpoint is at $W_1$, and starts to move towards $W_2$. Due to the constant velocity, the flight time $T_n$ for the $n$-th flight is proportional to its length $|W_nW_{n+1}|$. The reason for assuming a constant velocity is that although acceleration can be used to produce more realistic movements, constant-velocity VR movement is known to be more comfortable than the movement with acceleration or deceleration [41]. We further assume that $T_n$, $S_n$, and $\alpha_n$ are all i.i.d. distributed over $n$. Based on the experimental data in desktop VR, we will propose the models for $T_n$, $S_n$, and $\alpha_n$.

The collected data shows that the viewpoint movement in the $XZ$-plane is well approximated by a sequence of flights. We apply the standard angle model proposed in [19] to extract flights from the trajectories. Fig. 5 plots the trajectory in $XZ$-plane of one user in Lite and the extracted flights. Although the viewport does not move in a perfectly straight line during each flight, the trajectory is close to it.

**Modeling flight duration.** Our experimental data demonstrates that $T_n$ is exponentially distributed, and confirms that the paused-MRWP better models the flight time in VR than the classical RWP models [17]. Specifically, the PDF of the flight times $T_n$, denoted as $f_{T_n}(t)$, is modeled as $\mu e^{-\mu t}$. Fig. 6 shows the cumulative distribution function (CDF) of flight times for our collected flight samples, the paused-MRWP model with the best fitted $\mu$, and the classical RWP with a constant velocity in VR game. We see that the flight times of the paused-MRWP model match the measurements better statistically, with the SSE as low as 0.0021. The exponential distribution with the best fitted $\mu$ will be used to model the flight times in §IV and §V.

**Modeling pause time.** We model the pause time $S_n$ accord-
ing to the collected data. Fig. [7] shows that the exponential distribution with a "bump" around zero is a good fit to its distribution. The PDF of the pause times, denoted as $f_{s_n}(s)$, is modeled as $(1 - c)e^{-\lambda s} + o(f(s))$, where $ae^{-\lambda s}$ stands for the PDF of the exponential distribution with parameter $\lambda$, Dirac delta function $\delta(s)$ models the "bump" around zero pause time, and $c$ represents the fraction of the "bump".

**Modeling flight direction.** From our pose dataset, the flight angles $\alpha_n$ follow the uniform distribution on $[0, 2\pi)$.

With the developed paused-MRWP model, we will focus on observation intervals of duration $\Delta t$, and derive an analytical expression for the moment generating function (MGF) of the displacement between two viewpoint positions.

**Definition 3 (MGF of position displacement).** Consider an observation interval $[t_s, t_s + \Delta t]$, where $t_s$ is a sample from $U[0, T]$. Let $X_{n}^{\text{ref}}$ and $X_{n}^{\text{nov}}$ be the viewpoint positions at $t_s$ and $t_s + \Delta t$, respectively. Then, the position displacement $\psi$ is defined as $\|X_{n}^{\text{nov}} - X_{n}^{\text{ref}}\|^2$, and the MGF of $\psi$, denoted by $M_{\psi}(\tau)$, is given by $M_{\psi}(\tau) = \lim_{\tau \to \infty} \mathbb{E} \left[ \exp \left( \tau \cdot \|X_{n}^{\text{nov}} - X_{n}^{\text{ref}}\|^2 \right) \right]$.

We note that during a small $\Delta t$ in VR systems, $X_{n}^{\text{ref}}$ and $X_{n}^{\text{nov}}$ are not necessarily at the waypoints. This is in contrast to conventional applications of RWP models, where the movement is observed at a larger timescale [17, 19].

2) Analysis: In analyzing $M_{\psi}(\tau)$, there are two mutually exclusive and exhaustive cases to consider. Setting $T_0 = S_0 = 0$ as the auxiliary variables, the Case 1 is the case that $\exists j \in \mathbb{N}^+$, $\sum_{n=0}^{j} T_n + \sum_{n=0}^{j} S_n < t_s < \sum_{n=0}^{j} T_n + \sum_{n=0}^{j} S_n$, i.e., we start observing the process when the movement is paused. Let $\Lambda$ denote the event that Case 1 holds. Let $\Lambda'$ denote the complement of $\Lambda$, and $\Lambda''$ is the event that Case 2 holds. The Case 2 is the case that $\exists j \in \mathbb{N}^+$, $\sum_{n=0}^{j} T_n + \sum_{n=0}^{j} S_n < t_s < \sum_{n=0}^{j} T_n + \sum_{n=0}^{j} S_n$, i.e., we start observing the process during a flight. We will first obtain the $k$-th ($k \geq 1$) moment of $\psi$, $m_{\lambda}(k)$ and $m_{\Lambda}(k)$, for Cases 1 and 2 in Lemmas [1] and [2]. Combined with the probability that Case 1 holds given in Lemma [3] we will obtain $M_{\psi}(\tau)$ in Theorem [1].

**Lemma 1.** Assume Case 1 holds. Let $T'_i = 0$, and $S'_i = S_i$ for $i > j$. We define the events $A_n$ and $B_n$ as

\[
A_n = \begin{cases} 
\{ \xi_1 < 0 \}, & n = 1 \\
\{ \xi_{n-1} \geq 0 \land \xi_n < 0 \}, & n \geq 2
\end{cases}
\]

\[
B_n = \begin{cases} 
\{ \xi_n \geq 0 \land \xi_n < 0 \}, & n \geq 1
\end{cases}
\]

where $\xi_n \triangleq \Delta t - \sum_{i=j}^{j+n-2} S'_i - \sum_{i=j}^{j+n-1} T'_i$ and $\hat{\xi}_n \triangleq \Delta t - \sum_{i=j}^{j+n-1} S'_i - \sum_{i=j}^{j+n} T'_i$. $A_n$ is the event that we end the observation in a pause interval ($\exists j' \in \mathbb{N}^+$, $\sum_{n=0}^{j'} T_n + \sum_{n=0}^{j'-1} S_n < t_s + \Delta t < \sum_{n=0}^{j'} T_n + \sum_{n=0}^{j'} S_n$) and that there are $n - 1$ complete flights in $[t_s, t_s + \Delta t]$. $B_n$ is the event that $\exists j \in \mathbb{N}^+$, $\sum_{n=0}^{j-1} T_n + \sum_{n=0}^{j-1} S_n < t_s + \Delta t < \sum_{n=0}^{j} T_n + \sum_{n=0}^{j} S_n$, and that there are $n - 1$ complete flights in $[t_s, t_s + \Delta t]$. Then, $m_{\lambda}(k)$ is given by

\[
m_{\lambda}(k) = \sum_{n=2}^{\infty} \mathbb{E} [\psi^k(1(A_n))] + \sum_{n=1}^{\infty} \mathbb{E} [\psi^k(1(B_n))] \tag{1}
\]

where

\[
\mathbb{E} [\psi^k(1(A_n))] = \binom{n + k + 2}{n - 2} \sum_{h=0}^{n-2} \mu \lambda g_{n-2, h, h+2, k, 2k+n+h+1} \tag{2}
\]

\[
\mathbb{E} [\psi^k(1(B_n))] = \binom{n + k - 1}{n - 1} \sum_{h=0}^{n-1} \lambda g_{n-1, h, h+1, k, 2k+n+h+1} \tag{3}
\]

and $g_{i, h, k, m, n} \triangleq \frac{(2k)\mu^{h}(\Delta t)^{m-1}e^{-n(\Delta t)}F_{1}(n; m; -\Delta t)}{(m-1)!} \times \sum_{j=0}^{\infty} T_{j}e_{i}$, where $e_i$ denotes the unit vector whose direction represents the moving direction $\alpha_i$ of the $i$-th flight. Based on the property that $\alpha_i$ is i.i.d. uniformly distributed on $[0, 2\pi)$ and on the distributions of flight and pause times, we get the expressions of $\mathbb{E} [\psi^k(1(A_n))]$. Similar techniques are used to obtain $\mathbb{E} [\psi^k(1(B_n))]$.

**Proof.** See Appendix A Proof sketch: Since the collections of events $\{A_n\}$ and $\{B_n\}$ are mutually disjoint and collectively exhaustive, we write $m_{\lambda}(k)$ as in (1). To calculate $\mathbb{E} [\psi^k(1(A_n))]$, we express the $k$-th moment of $\psi$ as

\[
u^{2k} \left[ \sum_{i=j+1}^{j+n-1} T_{i}e_{i} \right] \tag{4}
\]

where $e_i$ denotes the unit vector whose direction represents the moving direction $\alpha_i$ of the $i$-th flight. Based on the property that $\alpha_i$ is i.i.d. uniformly distributed on $[0, 2\pi)$ and on the distributions of flight and pause times, we get the expressions of $\mathbb{E} [\psi^k(1(A_n))]$. Similar techniques are used to obtain $\mathbb{E} [\psi^k(1(B_n))]$.

**Lemma 2.** Assume Case 2 holds. Let $T'_i = 0$, and $T'_j$ be the remaining flight duration after $t_s$ in the same flight interval. Let $T'_i = T_i$ for $i > j$ and $S'_i = S_i$ for $i > j$. Let the events $A_n$ and $B_n$ be defined as

\[
A_n = \begin{cases} 
\{ \xi_1 < 0 \}, & n = 0 \\
\{ \xi_n \geq 0 \land \xi_n < 0 \}, & n > 1
\end{cases}
\]

\[
B_n = \begin{cases} 
\{ \xi_n \geq 0 \land \xi_n < 0 \}, & n \geq 1
\end{cases}
\]

where $\xi_n \triangleq \Delta t - \sum_{i=j}^{j+n-2} S'_i - \sum_{i=j}^{j+n-1} T'_i$ and $\hat{\xi}_n \triangleq \Delta t - \sum_{i=j}^{j+n-1} S'_i - \sum_{i=j}^{j+n} T'_i$. Then, $m_{\Lambda}(k)$ is given by

\[
m_{\Lambda}(k) = \sum_{n=0}^{\infty} \mathbb{E} [\psi^k(1(A_n))] + \sum_{n=1}^{\infty} \mathbb{E} [\psi^k(1(B_n))]
\]
Proof. The proof follows directly from Lemmas 1-3. Then we apply the developed VR pose model to analyze the ViS(d) in IV-B and propose the ViS-based VR content splitting algorithm, ALG-ViS, in IV-C.

A. ViS Model

The analytical model for ViS(d) we develop in this section characterizes the statistical average of the inter-frame pixel similarity over different pose changes given the viewport-to-content distance d. We define the ViS(d) formally after introducing the camera model to present how viewpoint pose determines the rendered pixels.

Camera model. In VR, the virtual environment is constructed as computer-generated 3D contents, where the pixels in VR frames are generated by capturing the scenes with the camera. The camera is modeled as a standard pinhole camera following [44]. A 3D point in the virtual environment is projected through the pinhole to a pixel on the VR frame. We denote the camera’s angle of view (AoV) as ψ

We break down the reference frame’s pixels that are projected to the 3D points with distance d from the reference camera. d is expressed as

\[ d = \sqrt{(x_{3D} - x_r)^2 + (z_{3D} - z_r)^2}, \]

where \( x_{3D}, z_{3D} \) are the positions of the 3D point and the reference camera in the XZ-plane, respectively. Let \( S_{d} \) denote the number of the reference frame’s pixels that have the same pixel value as the corresponding pixels in the novel frame. The visibility similarity given d is defined as

\[ \text{ViS}(d) = \frac{1}{|S_{d}|} \sum_{i=1}^{n} \delta(x_{3D} - x_r, z_{3D} - z_r), \]

where \( x_{3D}, z_{3D} \) are the positions of the 3D point and the reference camera in the XZ-plane, respectively. Let \( S_{d} \) denote the number of the reference frame’s pixels that have the same pixel value as the corresponding pixels in the novel frame. The visibility similarity given d is defined as

\[ \text{ViS}(d) = \frac{1}{|S_{d}|} \sum_{i=1}^{n} \delta(x_{3D} - x_r, z_{3D} - z_r). \]

We introduce the model for the average visibility similarity given the viewport-to-content distance d (ViS(d)) in IV-A.

D. Correlations between Orientations and Positions

VR viewpoints’ position and orientation are correlated. Observing the azimuth angles \( \phi \) and the walking directions (i.e., the included angle between \( \mathbf{X}_{\text{ref}} \) and the positive direction of X-axis) in our collected data, we find that the azimuth angles fixate around the walking direction. Similar observations have been made about human walking patterns in non-virtual worlds [42], [43]. Supported by the pose data, we assume that the azimuth angle at observation start time is the same as the walking direction.

IV. VISIBILITY SIMILARITY

We introduce the model for the average visibility similarity given the viewport-to-content distance d (ViS(d)) in IV-A. Then we apply the developed VR pose model to analyze
result in the use of more pixels to represent the contents. Note that \( \text{ViS}_{\text{fov}} \) is independent of \( d \) while \( \text{ViS}_{\text{dist}}(d) \) is a function of \( d \). In our analysis, we ignore the influence of occlusion in \( \text{ViS}(d) \), i.e., we do not consider the case where the occluded objects are in the FoV of both reference and novel frames, are rendered in the novel frame, but occluded by the other contents in the reference frame. In our numerical results, we show that their effect is small. We have \( \text{ViS}(d) = \text{ViS}_{\text{fov}} \cdot \text{ViS}_{\text{dist}}(d) \). The \( \text{ViS}_{\text{fov}} \) and \( \text{ViS}_{\text{dist}}(d) \) are defined formally below.

**Definition 5 (FoV term).** \( \text{ViS}_{\text{fov}} \) is defined as the statistical average of the multiplication of the fraction of overlapping polar and azimuth angles of reference and novel cameras.

We analyze \( \text{ViS}_{\text{fov}} \) with the distributions of \( \Delta \theta = \theta_{\text{nov}} - \theta_{\text{ref}} \) and \( \Delta \phi = \phi_{\text{nov}} - \phi_{\text{ref}} \) obtained in \( \S III-B \). Fig. 9 depicts the positions and angles of reference and novel camera viewports in the \( XZ \)-plane. Since the far plane \( d_{fp} \) is much larger than \( \| \mathbf{X}_{\text{ref}} - \mathbf{X}_{\text{nov}} \| \), the fraction of the overlapping azimuth angles is \( \frac{\min(\theta_{\text{ref}} + \frac{w_{\text{ref}}}{2}, \theta_{\text{nov}} + \frac{w_{\text{ref}}}{2}) - \max(\theta_{\text{ref}} - \frac{w_{\text{ref}}}{2}, \theta_{\text{nov}} - \frac{w_{\text{ref}}}{2})}{w_{\text{ref}}} \). The fraction of the overlapping polar angles is obtained similarly. Hence, the FoV term is

\[
\text{ViS}_{\text{fov}} = \mathbb{E}_{\Delta \theta} \left[ 1 - \frac{|\Delta \theta|}{w_{fv}} \right] \cdot \mathbb{E}_{\Delta \phi} \left[ 1 - \frac{|\Delta \phi|}{w_{fv}} \right].
\]

**Definition 6 (Distance term).** \( \text{ViS}_{\text{dist}}(d) \) is defined as

\[
\text{ViS}_{\text{dist}}(d) \triangleq \mathbb{E}_{\psi, \theta} \left[ \left( \sqrt{d^2 + \psi - 2d\sqrt{\psi} \cos(\theta)} \right)^2 \right]
\]

where \( \sqrt{d^2 + \psi - 2d\sqrt{\psi} \cos(\theta)} \) is the distance between the VR content and the novel camera, and \( \psi \) is the included angle between \( \mathbf{X}_{\text{ref}} \mathbf{X}_{\text{fov}} \) and \( \mathbf{X}_{\text{ref}} \mathbf{X}_{\text{nov}} \), as shown in Fig. 9.

**B. The Analysis of \( \text{ViS}(d) \)**

We first provide a closed-form expression for \( \text{ViS}_{\text{fov}} \), and then approximate \( \text{ViS}_{\text{dist}}(d) \) tightly with a small error.

**Theorem 2 (FoV term).** \( \text{ViS}_{\text{fov}} \) is equal to

\[
\text{ViS}_{\text{fov}} = \left( 1 - \frac{b_{\theta} - \exp\left(-\frac{w_{fv}}{b_{\theta}}\right)}{w_{fv}} (b_{\phi} + w_{fv}) \right) \left( 1 - p_{\psi}^\phi \right)
\]

where \( b_{\theta, \phi} \) is the scale of the fitted Laplace distribution of \( \Delta \theta, \Delta \phi \) and \( p_{\psi}^\phi = \frac{2}{w_{fv}} \int_0^{w_{fv}} \Delta \phi p_{\Delta \phi}(\Delta \phi) d(\Delta \phi) \), where \( p_{\Delta \phi}(\Delta \phi) \) is the PDF of \( \Delta \phi \). \( p_{\psi}^\phi \) is equal to

\[
p_{\psi}^\phi = \begin{cases} b_{\psi} - \exp\left(-\frac{w_{fv}}{b_{\psi}}\right) (b_{\psi} + w_{fv}) & \Delta \psi < \beta_1 \\ \frac{2(1 - p_{\psi}^\phi)}{w_{fv}} \exp\left(-\frac{w_{fv}}{b_{\psi} + w_{tv}}\log\left(\frac{z}{1 + e^{-\frac{w_{fv}}{b_{\psi} + w_{tv}}}}\right)\right) & 1 + e^{-\frac{w_{fv}}{b_{\psi} + w_{tv}}} \leq \Delta \psi < \beta_2 \end{cases}
\]

**Proof.** See Appendix B.

We focus on small \( \Delta \psi \) (e.g., \( \Delta \psi \leq 1 \)) which belongs to the first case in (3), as the most relevant, in practice, to exploiting VR frame correlation. In this case, the FoV term only depends on \( w_{fv}, b_{\psi}, b_{\theta}, b_{\phi} \).

**Theorem 3 (Distance term).** For \( \varepsilon > 0 \), we can approximate \( \text{ViS}_{\text{dist}}(d) \) within error of \( \kappa \varepsilon \mathcal{M}_{\psi}\left(\frac{1}{\varepsilon^2}\right) \), where \( \kappa = \frac{4 \sin \left(\frac{\pi}{w_{fv}}\right)}{w_{fv} d_{fp} \psi} \).

Specifically, \( \text{ViS}_{\text{dist}}(d) \) can be approximated according to

\[
\left| \text{ViS}_{\text{dist}}(d) - 1 - \frac{m(1)}{d^2} + \kappa \sum_{i=0}^{\infty} g(i) \right| < \kappa \varepsilon \mathcal{M}_{\psi}\left(\frac{1}{\varepsilon^2}\right)
\]

where \( g(i) = \frac{(-1)^i e^{-(2i+1)m(i+1)}}{(1+i)!} \).

**Proof.** See Appendix D. Proof sketch: Taking the average over \( \psi \) and substituting \( m(1) \), (3) is rewritten as \( \text{ViS}_{\text{dist}}(d) = 1 + \frac{m(1)}{d^2} - \frac{4 \sin \left(\frac{\pi}{w_{fv}}\right)}{w_{fv} d_{fp} \psi} \mathbb{E}_{\psi} \left[ \psi \right] \). Since

\[
\mathbb{E}_{\psi} \left[ \psi \right] = \frac{1}{\sqrt{\pi}} \mathbb{E}_{\psi} \left[ \int_0^\infty \psi e^{-\psi^2 \psi \psi - \frac{1}{2} \psi} d\psi \right] = \sum_{i=0}^{\infty} g(i)
\]

and that the approximation error \( \mathbb{E}_{\psi} \left[ \int_0^\infty ^2 \psi e^{-\psi^2 \psi \psi - \frac{1}{2} \psi} d\psi \right] \) is smaller than \( \varepsilon \mathcal{M}_{\psi}\left(\frac{1}{\varepsilon^2}\right) \) to conclude the proof.

From (5), \( \text{ViS}_{\text{dist}}(d) \) can be approximated by \( 1 + \frac{m(1)}{d^2} - \kappa \sum_{i=0}^{\infty} g(i) \). The approximation error \( \varepsilon \mathcal{M}_{\psi}\left(\frac{1}{\varepsilon^2}\right) \) can be made arbitrarily small by choosing a small \( \varepsilon \). \( \sum_{i=0}^{\infty} g(i) \) is convergent since \( |g(i)| \) is bounded by \( 2 e^{-2(i+1)\kappa(2i+1)} \). Hence, \( \sum_{i=0}^{\infty} g(i) \) is convergent, and the sum of the first 30 terms provides a good approximation with an error \( \varepsilon \mathcal{M}_{\psi}\left(\frac{1}{\varepsilon^2}\right) < 0.01 \) in (4).

The time taken to calculate the \( \text{ViS}(d) \) is dominated by calling the confluent hypergeometric function \( _1 F_1 (n; m; z) \) in Lemmas 1 and 2. The average time, over 100 iterations, for calculating \( \text{ViS}(d) \) on a Lenovo laptop (equipped with an AMD Ryzen 7 4800H CPU and an NVIDIA GTX 1660 Ti GPU) in MATLAB is only 4.4 ms. Hence, the \( \text{ViS}(d) \) can be calculated for each frame in real time.
Algorithm 1 ALG-ViS.

1: \(\textit{count} = 0;\)
2: for each new frame do
3: \[\text{if } \text{mod} (\textit{count}, R) == 0 \text{ then}\]
4: \(\text{The frame is selected as a reference frame;}\)
5: \(\text{else}\)
6: \(\text{Classify all VR contents as foreground contents;}\)
7: \(d_t \leftarrow d_{fp};\)
8: \(\text{for } (d = 0.5; d < d_{fp}; d \leftarrow d + 0.5) \text{ do}\)
9: \(\text{Calculate } \text{ViS}(d);\)
10: \(\text{if } \text{ViS}(d) \geq \text{ViS}_{tr} \text{ then}\)
11: \(d_t \leftarrow d;\)
12: \(\text{break};\)
13: \(\text{Classify the VR contents with } d \leq d_t \text{ as foreground contents, and other contents as background contents;}\)
14: \(\text{count} \leftarrow \text{count} + 1;\)
15: \(\text{Render the foreground contents. Reuse the background pixels from the reference frame by view projection;}\)

C. ViS-Based Foreground and Background Content Splitting

Based on the average ViS for a given \(d\), i.e., \(\text{ViS}(d)\), we adaptively split the contents to background and foreground, where the background (with a larger \(d\)) has a high \(\text{ViS}(d)\) and can be reused to reduce the resource consumption. To this end, we propose ALG-ViS given in Algorithm 1. For every \(R\) consecutive frames, the first frame is selected as the reference frame; the remaining \(R - 1\) frames are the novel frames. In reference frames, all VR contents are classified as foreground contents. In novel frames, we calculate a distance threshold \(d_{tr}\) such that \(\text{ViS}(d_{tr}) = \text{ViS}_{tr}\), where \(\text{ViS}_{tr}\) is the threshold and a \(\text{ViS}(d)\) larger than \(\text{ViS}_{tr}\) indicates high similarity of pixels in reference and novel frames. The VR contents that have the distance \(d \leq d_{tr}\) from the reference camera are classified as foreground contents, the other contents – as background contents. VR system renders the foreground contents and reuses the pixels for the background contents from the reference frame by view projection [8, 13]. We only need to calculate \(\text{ViS}(d)\) for \(R - 1\) values of \(\Delta t\) when the inter-frame interval is fixed (e.g., when the system supports the full frame rate as in [V], which makes ALG-ViS even more lightweight.

V. EVALUATION

We verify the analysis of ViS via simulations in §V-A and examine the performance of ALG-ViS in real-world VR implementations in §V-B. The parameters are listed in Table V unless otherwise specified.

A. Examining ViS

1) Simulation settings: We verify the ViS analysis via simulations using Unity Engine 2019.2.14f\(1\) [14] with 3 VR games listed in Table I. The results are analyzed in MATLAB.

To simulate the ViS, we randomly sample 5000 pairs of viewport poses from the collected pose trajectories for reference and novel cameras. We obtain the pristine novel frame rendered by Unity, and the generated novel frame by view projection from the reference frame and its depth map, where each pixel in the depth map represents the distance of the VR contents to the reference camera. Among the generated novel frame’s pixels whose corresponding 3D points are at a distance \(d\) from the reference camera, the pixels with RGB values with indistinguishable differences from the pristine novel frame constitute the set \(\Phi(d)\); the other pixels form the set \(\Phi^c(d)\). The ViS given \(d\) is calculated as the proportion of matched pixels \(\text{ViS}(d) = \frac{|\Phi(d)|}{|\Phi(d)| + |\Phi^c(d)|}\). We consider pixel values as indistinguishable when the difference of the pixel values is less than \(C_{th} \in \mathbb{N}^+\) in all RGB channels [4].

2) Numerical results: Fig. 10 shows the ViS obtained in our simulations, \(\text{ViS}(d)\), and the analytically derived \(\text{ViS}(d)\), for headset VR and phone-based VR, for different \(d\) and \(\Delta t\). The results for desktop VR are similar to the results for phone-based VR and are omitted. As expected, the ViS declines with \(\Delta t\) (i.e., frames that are separated by a longer time interval are less similar), and increases with \(d\) (i.e., contents that are farther from the camera change less across different frames). We observe differences between VR interfaces as well: the ViS for headset VR (Fig. 10(a)) is smaller than the ViS for the other interface types (Fig. 10(b)). This is explained by the difference in movement dynamics we observed in our dataset: in headset VR, the users change their viewpoint orientations and flight directions more rapidly than in the other VR interfaces. Smaller ViS for headset VR implies that fewer VR contents will be classified as background contents. Finally, we note that the gap between the analytical results and the simulations is small, 4.1\% on average for \(d < 20\)m and only 1.6\% on average for \(d \geq 20\)m. The gap can be attributed to the omission of object occlusions from our analytical derivations. The achieved highly accurate analytical ViS\((d)\) in the high ViS regime (e.g., \(d \geq 20\)m) is crucial for selecting the distance threshold \(d_{tr}\) to ensure that the background has a high ViS.

The obtained ViS for the three VR games in desktop VR is shown in Fig. 11. The results for the other 2 interface types exhibit the same trends and are omitted. Although VK has a larger number of triangles and vertices, which manifests in higher visual scene complexity, the ViS\((d)\) of VK is larger than the ViS\((d)\) of Office and Lite. This is because users’ pose trajectories in VK have relatively smaller \(b_{l,\theta}\) and \(b_t\), and smaller \(\mu\) and \(\lambda\), corresponding to larger flight lengths and pause durations. In other words, users tend to change both their viewpoint orientations and positions more slowly in VK.

2) \(C_{th}\) is selected as the maximum value such that the structural similarity index measure (SSIM) between the combined and the pristine novel frame is larger than 0.95. The combined novel frame is obtained by replacing the mismatched pixels \(\cup \Phi^c(d)\) in the generated novel frame with the corresponding pixels in the pristine novel frame.

| Parameter | Value |
|-----------|-------|
| VR frame resolution | 1080×1080 |
| AoV | 90° |
| Depth encoding | Linear |
| Far clipping plane \(d_{fp}\) | 50 m |
| Unity unit: meter | 1:1 |
| Pairs of viewport poses | 5000 |
| Depth map precision | 8 bits |
| Eye level height | 1.6 m |

| Parameter | Value |
|-----------|-------|
| VR Frame rate | 60 fps |
| Mapping scheme | Equiangular |
| \(\text{SSIM}\) | 0.95 |

The obtained ViS for the three VR games in desktop VR is shown in Fig. 11. The results for the other 2 interface types exhibit the same trends and are omitted. Although VK has a larger number of triangles and vertices, which manifests in higher visual scene complexity, the ViS\((d)\) of VK is larger than the ViS\((d)\) of Office and Lite. This is because users’ pose trajectories in VK have relatively smaller \(b_{l,\theta}\) and \(b_t\), and smaller \(\mu\) and \(\lambda\), corresponding to larger flight lengths and pause durations. In other words, users tend to change both their viewpoint orientations and positions more slowly in VK.

2) \(C_{th}\) is selected as the maximum value such that the structural similarity index measure (SSIM) between the combined and the pristine novel frame is larger than 0.95. The combined novel frame is obtained by replacing the mismatched pixels \(\cup \Phi^c(d)\) in the generated novel frame with the corresponding pixels in the pristine novel frame.
and present the results for Lite, which are representative. We are replaying 30 min of headset pose trajectories we collected reproducibility, for all algorithms our evaluation is based on this system is set to the laptop’s default 60 fps. To ensure an NVIDIA GTX 1660 Ti GPU. The target frame rate of sending them to the headset over IEEE 802.11ac WiFi. The Unity Engine 2019.1.14f1 and Google VR SDK [46], and frame rate is set to the default 72 fps. The edge-assisted rendering system is implemented with build 30.0; its target frames in an Oculus Quest 2 VR headset [45]. On-device performance of ALG-ViS

We implement ALG-ViS and a set of baselines in two VR rendering systems, an on-device (“local”) one and one supported by edge computing. Both systems display the generated frames in an Oculus Quest 2 VR headset [45]. On-device rendering system is implemented with build 30.0; its target frame rate is set to the default 72 fps. The edge-assisted system generates the VR frames in a Lenovo laptop, with Unity Engine 2019.1.14f1 and Google VR SDK [46], and sends them to the headset over IEEE 802.11ac WiFi. The laptop is equipped with an AMD Ryzen 7 4800H CPU and an NVIDIA GTX 1660 Ti GPU. The target frame rate of this system is set to the laptop’s default 60 fps. To ensure reproducibility, for all algorithms our evaluation is based on replaying 30 min of headset pose trajectories we collected (see [11]-[14]. We examine the performance for all 3 games, and present the results for Lite, which are representative. We examine the required bandwidth for the edge-supported system in Fig. 12 and the achieved SSIM, frame processing time, and CPU and GPU usage (monitored using the OVR Metrics Tool [47]) for the local system in Fig. 13.

We compare ALG-ViS to different approaches with the same reference frame interval $R$: ALG-FX-S, ALG-FX-T, and ALG-ML. In ALG-ViS, $\Delta t_{V}$ is set to 0.945 to ensure high ViS values for background contents. We set $R$ as the minimum integer such that $d_{tr} < d_{fp}$ for every frame. In ALG-FX-S and ALG-FX-T, $d_{tr}$ is determined by the number of triangles $N_{tr}$ in the VR frame, similar to the near and far background splitting in [10]. Specifically, $d_{tr} = \max\{qN_{tr}, d_{fp}\}$, where $q$ is fixed for each game. For a fair comparison, in ALG-FX-T, we set $q$ to make the average frame processing time of ALG-FX-T and ALG-ViS the same to compare the SSIM; in ALG-FX-S, we set $q$ to make the average SSIM the same as ALG-ViS when comparing other metrics. In ALG-ML, we adopt an online ridge regression model, which has been shown to achieve state-of-the-art accuracy in 360° video pose prediction [24], [25]. Following [11], we predict $x, y, z, \theta$, and $\phi$ separately. We set the history and prediction windows as in [11]. We split the VR contents by calculating the ViS of the reference frame and the predicted VR frame.

The CDF of the required bandwidth of the edge-assisted rendering system shown in Fig. 12 demonstrates that ALG-ViS requires less bandwidth on average than ALG-FX-S (11.3% difference; ALG-FX-S consistently transmits more pixels to maintain the same SSIM as ALG-ViS), and has significantly smaller bandwidth variance than ALG-ML (88.4% reduction). Although ALG-ML can save bandwidth when it accurately predicts the viewport pose, the required bandwidth increases drastically when the prediction is erroneous. Generating less bursty traffic, ALG-ViS prevents transmission resource over-provisioning and potential TCP incast problems in edge-assisted VR systems, which helps supporting these systems better when compared with ALG-ML.

Fig. 13 shows the CDFs of the SSIM, frame processing time, and CPU and GPU usage for the local rendering system. It shows that ALG-ViS improves frame quality and frame processing time while consuming fewer resources. The ALG-ViS ensures high average SSIM, outperforming ALG-FX-T by 3.2% and ALG-ML by 5.9%. The ALG-ML exhibits lower frame quality. This is because ALG-ML splits the foreground and background contents based on the predicted pose, and prediction errors lead to severe performance degradation [8].
The ALG-ViS decreases the frame processing time by 16.1% and 33.4% compared to ALG-FX-S and ALG-ML. The CPU usage of ALG-ML is 20.5% higher than that of ALG-ViS due to the extra computation required to tune the regularization parameter and conduct the pose prediction. The GPU usage of ALG-FX-S is 33.7% higher than that of ALG-ViS because ALG-FX-S classifies more VR contents as foreground contents on average. These results demonstrate that the developed ALG-ViS is lightweight yet effective.

VI. CONCLUSION

In this paper, we first propose a viewport pose model for VR systems based on the experimental measurements. We apply the pose model to adaptively select background contents that are reused across VR frames to reduce the communication and computation resource consumption, via quantifying the similarity of pixels across VR frames. Numerical results verify the pose model and the inter-frame pixel similarity analysis. Oculus Quest 2-based implementations of our adaptive background content selection approach show that it improves the image quality by 5.6% and reduces the variance of the required bandwidth by 88.4% compared to the method based on viewport pose prediction.

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] S. M. LaValle, Virtual Reality. Cambridge University Press, 2016.
[2] 3GPP. (2021) Technical specification group services and system aspects; Extended reality (XR) in 5G. TR 26.928 V16.1.0. https://www.3gpp.org/ftp/Specs/archive/26_series/26_928
[3] PricewaterhouseCoopers. (2019) Seeing is believing. https://www.pwc.com/seeingsbelieving
[4] E. Cuervo, K. Chintalapudi, and M. Kotaru, “Creating the perfect illusion: What will it take to create life-like virtual reality headsets?” in Proc. ACM/IEEE IPSN, 2018.
[5] Z. Tan, Y. Li, Q. Li, Z. Zhang, Z. Li, and S. Lu, “Supporting mobile VR in LTE networks: How close are we?” in Proc. ACM SIGMETRICS, 2018.
[6] J. Jeong, S. Lee, I. Ryu, T. Le, and E. Ryu, “Towards viewport-dependent 6DoF 360 video tiled streaming for virtual reality systems,” in Proc. ACM MM, 2020.
[7] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gutwemer, “Cutting the cord: Designing a high-quality untethered VR system with low latency remote rendering,” in Proc. ACM MobiSys, 2018.
[8] Y. Li and W. Gao, “DeltaVR: Achieving high-performance mobile VR dynamics through pixel reuse,” in Proc. ACM/IEEE ASILOMIR, 2019.
[9] Y. Li and W. Gao, “MUVR: Supporting multi-user mobile virtual reality with resource constrained edge cloud,” in Proc. IEEE/ACM SEC, 2018.
[10] 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.
[11] X. Liu, C. Vlachou, F. Qian, C. Wang, and K.-H. Kim, “Firefly: Untethered multi-user VR for commodity mobile devices,” in Proc. USENIX ATC, 2020.
[12] P. Srivastava, A. Rimzhim, P. Vijay, S. Singh, and S. Chandra, “Desktop VR is better than non-ambulatory HMD VR for spatial learning,” Frontiers Front. Robot. AI, vol. 6, no. 50, pp. 1–15, 2019.
[13] Y. Liu, J. Liu, A. Argyrilou, 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.
[14] Unity Technologies. (2021) The leading platform for creating interactive, real-time content. [Online]. Available: https://unity.com/
A. Proof of Lemma 7

When Case 1 holds, there are two possible ways to end the observation interval at time $t_s + \Delta t$. If $\exists j^* \in \mathbb{N}^+$, $\sum_{n=1}^{j^*-1} S_n < t_s + \Delta t < \sum_{n=0}^{j^*-1} T_n + \sum_{n=0}^{j^*} S_n$, the observation ends. Otherwise, $t_s + \Delta t$ lies in a flight interval. This leads to the first summation in (1).

If $t_s + \Delta t$ lies in a flight interval, the movement includes some number of complete flights without any fractional flights. We do not observe any movement when $n = 1$. For $n \geq 1$, $A_n$ is the event that $t_s + \Delta t$ lies in a pause interval and there are $n - 1$ complete flights in $[t_s, t_s + \Delta t]$. This leads to the second summation in (1).

It only remains to calculate $\mathbb{E}[\psi^k 1(A_n)]$ and $\mathbb{E}[\psi^k 1(B_n)]$ to complete the proof. We start with $\mathbb{E}[\psi^k 1(A_n)]$. Given $T_i, e_i$, the $k$-th moment of $\psi$ is given by $\nu^2k \sum_{i=j+1}^{j+n-1} T_i e_i$. According to the flight direction model in (III-C), the direction of $e_i$ is i.i.d. uniformly distributed on $[0, 2\pi)$. Thus, $\mathbb{E}_{e_i} e_i e_i^* = 1$ for $i = k$ and 0 for $i \neq k$. Thus, expanding $\mathbb{E}[\psi^k 1(A_n)]$ and averaging over $e_{j+1}, e_{j+2}, \ldots, e_{j+n-1}$, only the terms in which all $T_i e_i$ ($j + 1 \leq i \leq j + n - 1$) have some powers are non-zero. The number of these non-zero terms is $(n_k + k - 2)$. Further, let $h(l) = \sum_{i=l}^{j+n-1} T_i$ denote the number of non-zero complete pause intervals (excluding the first and last fractional pause intervals). Denote the set of the non-zero complete pause intervals as $\{S_i^k\}, 1 \leq i \leq h$. Let $s = \Delta t - \sum_{i=j+1}^{j+n-1} T_i$ be the sum of pause intervals in the observation interval. $\mathbb{E}[\psi^k 1(A_n)]$ is given by (6), shown at the top of the previous page. Using these observations, the expressions for $\mathbb{E}[\psi^k 1(B_n)]$ can be obtained similarly, which completes the proof.

B. Proof of Lemma 3

Let $\Gamma_n$ be the end time of the $n$-th flight, i.e., $\Gamma_n = \sum_{i=0}^{n-1} S_i + \sum_{i=0}^{n-1} T_i$. Then, for any $T > 0$, there exists $n \in \mathbb{N}$ such that $\Gamma_n \leq T < \Gamma_{n+1}$, and we can upper and lower bound $p_T$ according to

$$\mathbb{E} \left[ \frac{n+1}{\sum_{i=0}^{n} S_i + \sum_{i=0}^{n+1} T_i} \right] \leq p_T \leq \mathbb{E} \left[ \frac{n}{\sum_{i=0}^{n-1} S_i + \sum_{i=0}^{n+1} T_i} \right].$$

The lower bound includes an extra pause interval between $\Gamma_n$ and $\Gamma_{n+1}$ by ignoring any possible fractional flight duration. On the other hand, the upper bound includes one complete flight duration between $\Gamma_n$ and $\Gamma_{n+1}$ by ignoring $S_n$.

When $T \to \infty$, using Lebesgue’s dominated convergence theorem to replace the order of limit and expectation operators and the law of large numbers (as $T$ grows large, $n$ tends to infinity), it can be seen that both upper and lower bounds converge to $p = \frac{\lambda/(1-c)}{\lambda(1-c)+\mu}$.

C. Proof of Theorem 2

Seen from Fig. 9, the overlapping azimuth angle is

$$\mathbb{E}_{\Delta \phi} \left[ \min(\phi_{ref}, \phi_{nov}) + \frac{w_{\phi}}{w_{\phi}'} - (\max(\phi_{ref}, \phi_{nov}) - \frac{w_{\phi}}{w_{\phi}'} \right] = 1 - \frac{1}{w_{\phi}'} \int_{-w_{\phi}'}^{w_{\phi}'} |\Delta \phi| p_{\Delta \phi}(\Delta \phi) d(\Delta \phi) \leq \frac{2}{w_{\phi}'} \int_{-w_{\phi}'}^{w_{\phi}'} \Delta \phi p_{\Delta \phi}(\Delta \phi) d(\Delta \phi) = p_f^\phi.$$ (7)

where $\Delta \phi = \phi_{ref} - \phi_{nov}$, and (a) is because $\min(\phi_{ref}, \phi_{nov}) + \frac{w_{\phi}}{w_{\phi}'} - (\max(\phi_{ref}, \phi_{nov}) - \frac{w_{\phi}}{w_{\phi}'})$ if $\phi_{ref} < \phi_{nov}$ and $\min(\phi_{ref}, \phi_{nov}) + \frac{w_{\phi}}{w_{\phi}'} - (\max(\phi_{ref}, \phi_{nov}) - \frac{w_{\phi}}{w_{\phi}'}) = \phi_{nov} + \frac{w_{\phi}}{w_{\phi}'} - (\phi_{ref} - \frac{w_{\phi}}{w_{\phi}'})$ if $\phi_{ref} \geq \phi_{nov}$. $p_{\Delta \phi}(\Delta \phi)$ is the PDF of $\Delta \phi$ obtained in Section [III-B].

$$p_{\Delta \phi}(\Delta \phi) = \begin{cases} \frac{1}{\tau_{\phi}} \exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right), & \Delta \phi < \beta_1 \\ \left(1 - p_L^\phi \right) \frac{h_{\tau}(1 + \exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right))}{\exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right)} \exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right) - \frac{1}{1 + \exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right)}, & \beta_1 \leq \Delta \phi < \beta_2 \\ \frac{2}{\tau_{\phi}^2} \exp \left( -\frac{\Delta \phi}{\tau_{\phi}} \right), & \Delta \phi > \beta_2. \end{cases}$$

Substituting $p_{\Delta \phi}(\Delta \phi)$ into (7), we obtain the results for $p_f^\phi$ (the fraction of overlapping azimuth angles) in (4).
\[
E[\psi^k | \mathbf{A}(n)] = \sum_{h=0}^{n-2} \frac{(n-2)}{h} (1+\epsilon)^h e^{n^2-h^2} v^2k \left( \mathbb{I}[n=2] \cdot \int_0^{\Delta t} (\Delta t - s)^{2k} + \mathbb{I}[n \geq 3], \int_0^{\Delta t} \int_0^{\Delta t - s} \right)
\]