# A Viewport Prediction Framework for Panoramic Videos

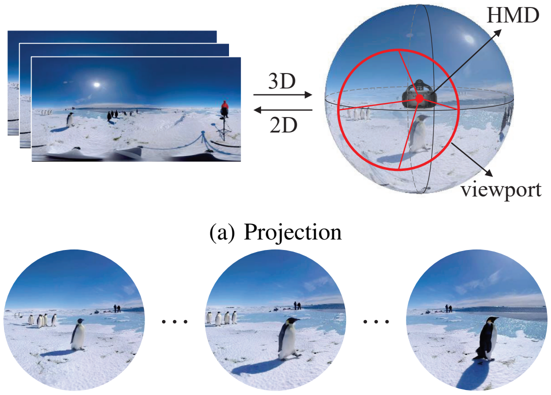
Jinting Tang, Yongkai Huo, Shaoshi Yang and Jianmin Jiang

***Abstract*—Panoramic video is considered to be an attractive video format, since it provides the viewers with an immersive experience, such as virtual reality (VR) gaming. However, the viewers only focus on part of panoramic video, which is referred to as viewport. Hence, the resources consumed for distributing the remaining part of the panoramic video are wasted. It is intuitive to only deliver the video data within this viewport for reducing the distribution cost. Empirically, viewports within a time interval are highly correlated, hence the historical trajectory may be used for predicting the future viewports. On the other hand, a viewer tends to sustain attention on a specific object in a panoramic video. Motivated by these findings, we propose a deep learning-based viewport Prediction scheme, namely HOP, where the Historical viewport trajectory of viewers and Object tracking are jointly exploited by the long short-term memory (LSTM) networks. Additionally, our solution is capable of predicting multiple future viewports, while a single viewport prediction was supported by the state-of-the-art contributions. Simulation results show that our proposed HOP scheme outperforms the benchmarkers by up to 33.5% in terms of the prediction error.**

***Index Terms*—panoramic video, viewport prediction, object tracking, deep learning**

## I. Introduction

Panoramic video has been attracting substantial research attention, since it enables 360 degree experience of the designated scenes. It may be utilized in numerous scenarios, such as sports, social network, advertisement and virtual reality (VR) gaming. The rapid expansion of networks, such as the fifth generation (5G) wireless networks [1], [2], may further motivate the applications of panoramic videos. Each panoramic frame may cover a range of 360°×180° video data in the horizontal and vertical directions, respectively. As shown in Fig. 1a, the planar panoramic video is projected onto a spherical surface for achieving immersive experience, where a viewer equipped with head-mounted displays (HMDs) is positioned at the center of the rendered sphere. The terminology viewport means the specific region of panoramic video, which attracts sustained attention of the viewer. The viewer may freely change the viewport by altering the orientation of his head or body, which is detected by the embedded positioning sensors of the HMDs. The size of viewport may range from 60° to 110° depending on the HMD, while the remaining part of the panoramic video is invisible to the viewer. Three sampled viewports are shown in Fig. 1b, and they are extracted from three panoramic frames. Fig. 2 exemplifies viewports of two different viewers of a panoramic frame. Note that different viewers may have different viewports depending on their interests in the panoramic videos. Below, we will review the researches of the panoramic video in various aspects.



(a) Projection

(b) Sampled viewports

Fig. 1: Illustration of watching a panoramic video using HMD.

- **Projection:** Equirectangular projection (ERP) is one of the most popular formats for panoramic videos, which is illustrated by Fig. 1a. Cube map projection (CMP) [3] was proposed by Wang *et al.* to replace ERP, since the CMP scheme can reduce the distortion and enhance the region-of-interest (ROI) signals of panoramic videos.
- **Compression:**Sánchez *et al.* [4] advocated an H.265/HEVC based video streaming algorithm for improving the quality of the ROI.
- **Viewport-adaptive delivery:** For reducing the bandwidth requirement while maintaining the quality of experience, an viewport-adaptive video distribution scheme was proposed in [5]–[10]. It aimed to deliver the video data according to the viewer's viewport, where more bandwidth may be allocated to the viewport region in comparison to sending the full panoramic video. Firstly, the viewport-adaptive algorithm is more bandwidth-efficient than delivering the full panoramic videos. Moreover, it may be readily integrated with dynamic adaptive streaming over

This work was supported in part by the National Natural Science Foundation of China (No.61702335), in part by the Science and Technology Innovation Commission of Shenzhen (JCYJ20170302154149766), in part by the startup funding of Shenzhen University (2110315). J. Tang, Y. Huo (Corresponding Author) and J. Jiang are with the National Engineering Laboratory for Big Data System Computing Technology, and the Research Institute for Future Media Computing, School of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. (e-mail:tangjinting2017@email.szu.edu.cn, ykhuo@szu.edu.cn, jianmin.jiang@szu.edu.cn) S. Yang is with the School of Information and Communication Engineering, and the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China. (e-mail:shaoshi.yang@ieee.org)

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

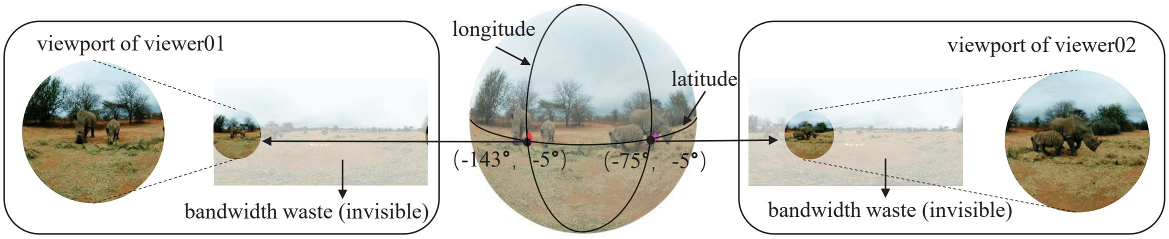


Fig. 2: Demonstration of two viewers' viewports.

HTTP (DASH) protocol and the HTTP protocol by using tile-based delivering schemes.

- **Virtual cinematography:** Chang *et al.* [11] proposed a non-heuristic algorithm to model the composition styles of professional photographs within a panoramic panoramic scene by analyzing the structural features and the layout of visual saliency. Su *et al.* [12] utilized dynamic programming to find the optimal human-like camera trajectory, which may be used for deriving the viewers' viewport. Hu *et al.* [13] proposed a 360° pilot scheme to predict the next viewport using the selected historical viewports and the characteristics of objects, which include their appearance, motion and locations.

Viewport prediction may be utilized in various applications of panoramic videos, such as ROI based compression [4], virtual cinematography [11]–[13], rendering [14] and viewport-adaptive delivery [6]–[10]. The viewports of viewers only occupy a small portion of panoramic videos, while the remaining parts are invisible to the viewers as exemplified by Fig. 2. Hence, massive bandwidth is required for distributing these panoramic videos intactly. As a solution, viewport-adaptive delivery schemes adaptively allocate the bandwidth for the ROI signals and ignore the remaining regions, thereby reducing the required bandwidth for delivering panoramic videos. Qian *et al.* [15] investigated the weighted linear regression (WLR) model for predicting the future viewport using historical viewport trajectory, which verifies that future viewports are correlated with their historical trajectory. Based on the panoramic content, saliency map [16] and optical flow [17] were investigated to predict the ROI by Fan *et al.* [18].

Motivated by the viewport-adaptive panoramic video delivery solutions, this paper aims to predict multiple future viewports of a single viewer. Therefore, the accuracy of predicted viewport may significantly affect the performance of viewport-adaptive panoramic systems, while larger number of predicted viewports correspond to lower network delay. For determining the ROI of a viewer, we employ the historical viewport trajectory and the foreground object[1] attracting the focus of the viewer. We observe from the dataset [19], [20] that viewers sustain their attention on some foreground objects, hence their viewports may change in a similar pace with these objects. Furthermore, in a panoramic video containing multiple objects, the viewers may focus on some of these objects, while their viewports tend to follow their favorite objects. We employ the tool of OpenCV for multi-object tracking [21]–[24]. Since the long short-term memory (LSTM) network [25], [26] is a powerful mechanism of solving sequence problems [27], [28], we propose a deep learning-based viewport prediction scheme for estimating the future viewport trajectory for a sequence of future frames, namely HOP, where the historical viewport trajectory and the tracking of objects are jointly exploited by the LSTM network.

Our main contributions are listed as follows:

(1) We propose a HOP scheme for predicting multiple future viewports, and the experiments validate the effectiveness of our solution.

(2) We are the first to exploit object tracking along with the historical viewport trajectory for improving the accuracy of the viewport prediction.

(3) We propose a "trajectory translation" algorithm to solve the discontinuity issue of viewport trajectory.

The rest of this paper is organized as follows. In section II, we detail the architecture of the proposed HOP scheme. Then the performance of the HOP scheme is presented in Section III. Finally, Section IV concludes the paper.

## II. Proposed Approach

In this section, we commence by introducing the problem formulation, followed by detailing our framework. The "trajectory translation" and "trajectory selector" blocks of the architecture will be detailed in section II-C and section II-D, respectively. Notations employed are defined in Table I.

### A. Problem Formulation

Since panoramic video carries a 360-degree view of the designated scene, typically it is substantially larger than the traditional video. Moreover, the invisible region of panoramic video is also delivered through the network, which causes

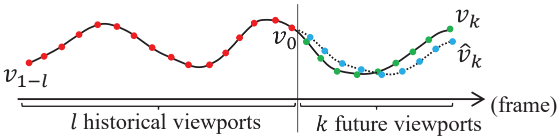


Fig. 3: Timeline of a viewport trajectory

[1] Take a movie as an example, people may always focus on their favorite stars.

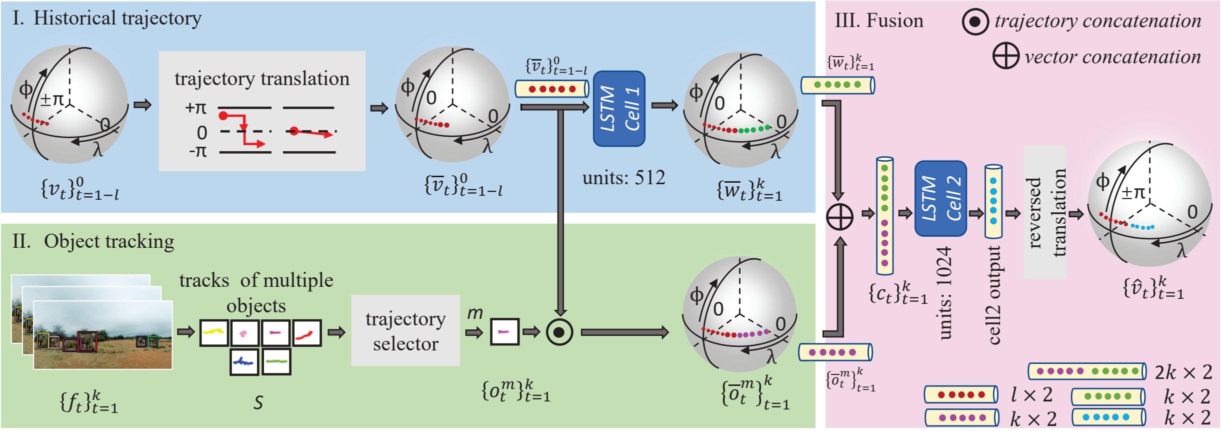


Fig. 4: Architecture of our HOP scheme

massive bandwidth waste. Hence, we aim for predicting future viewports for viewport-adaptive panoramic video delivery, where more accurately predicted viewports lead to lower bandwidth cost.

For predicting the viewport of a future frame, we exploit two aspects: the historical viewport trajectory and the content of panoramic video. As shown in Fig. 2, viewers may be attracted by different regions of the same panoramic frame. Hence, the historical viewport trajectory for different viewers are also different. Again, viewports of a historical trajectory may be highly correlated, hence a future viewport may be estimated from the viewer's historical trajectory. In spherical coordinate system, we denote the viewer's viewport of the $t^{th}$ frame as $v_t = (\lambda, \phi)$, where $\lambda \epsilon [-\pi, \pi]$ and $\phi \epsilon [-\pi/2, \pi/2]$ indicate the corresponding longitude and latitude, respectively. Fig. 3 illustrates the timeline of a viewport trajectory. Furthermore, $l$ historical viewports may be expressed as

$$\{v_t\}_{t=1-l}^{0} = \{v_{1-l}, v_{2-l}, ..., v_0\}\,. \tag{1}$$

Additionally, viewers tend to sustain their attention on specific content of the panoramic video. Although two viewers may have different viewports as shown in Fig. 2, the overlapped ROI of them may be extracted and utilized for predicting future viewport trajectory. Initially, we extract the common ROI of the next $k$ frames $\{f_t\}_{t=1}^{k}$ in the panoramic video. We observe from viewport trajectory that viewers are more likely interested in foreground objects. Viewers' viewport trajectories are associated with the motions of multiple foreground objects. Therefore, we use the multi-object tracking tool of OpenCV for recording motions of these objects in the panoramic video. The motions of $d$ objects in the next $k$ panoramic frames may be expressed as

$$\mathcal{S} = \left\{\left\{o_1^1, ..., o_k^1\right\}, ..., \left\{o_1^d, ..., o_k^d\right\}\right\}. \tag{2}$$

Our HOP scheme aims to combine the historical trajectory $\{v_t\}_{t=1-l}^{0}$ and the motion tracks of the multi-object $\mathcal{S}$. Then the predicted viewport of $k^{th}$ frame may be formulated as

$$\hat{v}_k = \mathrm{HOP}\left(\{v_t\}_{t=1-l}^{0}, \mathcal{S}\right). \tag{3}$$

TABLE I: Symbol definition

| Symbol | Definition |
|---|---|
| $l$ | the length of the historical viewport trajectory |
| $k$ | the index of the viewport to be predicted |
| $v_t$ | the viewport $v_t = (\lambda, \phi)$ of the $t^{th}$ frame, where $\lambda$ and $\phi$ denote the longitude and the latitude, respectively |
| $v_0$ | the viewport of the current frame |
| $\{v_t\}_{t=1-l}^{0}$ | $l$ historical viewports, containing the current viewport $v_0$ |
| $\overline{v}_t$ | the translated version of $v_t$ |
| $\{\overline{w}_t\}_{t=1}^{k}$ | the viewports generated by the LSTM Cell-1 network with $\{\overline{v}_t\}_{t=1-l}^{0}$ as the input |
| $\{f_t\}_{t=1}^{k}$ | $k$ panoramic frames of the future |
| $o_t^i$ | the location of the $i^{th}$ object in the $t^{th}$ frame |
| $\alpha_i$ | the spatial angle between the object $o_t^i$ and the viewport $v_t$ |
| $m$ | the index of the object having minimum spatial angle $\alpha$ |
| $\overline{o}_t^m$ | the translated version of $o_t^m$ |
| $\{c_t\}_{t=1}^{k}$ | the fused vector of $\{\overline{w}_t\}_{t=1}^{k}$ and $\{\overline{o}_t^m\}_{t=1}^{k}$ |
| $\{\hat{v}_t\}_{t=1}^{k}$ | the viewports of the predicted $k$ frames |

### *B. Architecture*

The architecture of the HOP scheme is illustrated in Fig. 4, which consists of three parts, namely the historical trajectory, the object tracking and the fusion.

*1) Historical trajectory:* The "historical trajectory" part of Fig. 4 aims to predict the future viewport trajectory depending on the corresponding historical trajectory. The viewport trajectory $\{v_t\}_{t=1-l}^{0}$, as exemplified in Fig. 4, contains $l$ historical viewports including the current viewport $v_0$. The viewport $v_0$, namely the rear of the historical trajectory $\{v_t\}_{t=1-l}^{0}$, is the starting point of the predicted viewport trajectory $\{\hat{v}_t\}_{t=1}^{k}$ for the $k$ future frames. However, we observe that there is a dramatic overturn around $\pm\pi$ in longitude, which breaks the continuity of this historical trajectory. Hence, the "trajectory translation" block of Fig. 4, as will be detailed in Section II-C, is introduced for mitigating this "longitude overturn" issue of the historical trajectory. The translated trajectory generated by the "trajectory translation" block, namely $\{\bar{v}_t\}_{t=1-l}^{0}$, is utilized by the LSTM Cell-1 network for generating the predicted trajectory $\{\bar{w}_t\}_{t=1}^{k}$, as shown in Fig. 4. Note that each viewport is determined by a pair of longitude and latitude, as defined in Table I. At this stage, the resultant of predicted trajectory purely depends on the historical trajectory.

*2) Object tracking:* The "object tracking" part of Fig. 4 is designed for obtaining the motion trajectories of foreground objects, which will be utilized for predicting the viewport trajectory. Firstly, the motion trajectories of multiple objects $\mathcal{S}$ are extracted from the successive $k$ panoramic frames $\{f_t\}_{t=1}^{k}$. The "trajectory selector" block of Fig. 4, as will be detailed in Section II-D, aims to find the motion trajectory of the "key" object[2]. Generally, the object closest to the current viewport will be selected as the "key" object. The motion trajectory of the "key" object is denoted as $\{o_t^m\}_{t=1}^{k}$, which will be translated to $\{\bar{o}_t^m\}_{t=1}^{k}$ and be concatenated with $\{\bar{v}_t\}_{t=1-l}^{0}$ by the "trajectory translation" block of Fig. 4.

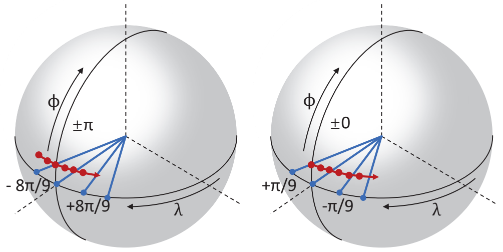
*3) Fusion:* The "fusion" part aims to derive the refined version of the predicted trajectory, namely $\{\hat{v}_t\}_{t=1}^{k}$ in Fig. 4, where the predicted trajectory $\{\bar{w}_t\}_{t=1}^{k}$ and the selected motion trajectory $\{\bar{o}_t^m\}_{t=1}^{k}$ are jointly exploited. Specifically, the vector $\{c_t\}_{t=1}^{k}$ is concatenated from the predicted trajectory and the selected motion trajectory. Given the vector $\{c_t\}_{t=1}^{k}$, the predicted viewport trajectory is generated by the LSTM Cell-2 network of Fig. 4. The "reversed translation" block of Fig. 4 reversely performs the translation for obtaining the rectified predicted trajectory $\{\hat{v}_t\}_{t=1}^{k}$.

### *C. Trajectory Translation*

Since the viewer may watch the panoramic video from arbitrary direction, the viewport trajectory is equivalent to a spherical arc as exemplified in Fig. 5a. Again, the viewport trajectory may have an overturn in the longitude coordinate around $\pm\pi$. Alternatively speaking, the viewport trajectory may cross the $\pm\pi$ in longitude, as shown in Fig. 5a. This discontinuity issue may confuse the LSTM network when generating the predicted trajectory. Therefore, the "trajectory translation" block of Fig. 4 is introduced for avoiding this discontinuity.

Generally, the "trajectory translation" block aims to concatenate multiple discontinuous trajectories to a single continuous trajectory. An initial spherical arc is shown in Fig. 5a, which consists of two discontinuous longitude sections, namely [-8π/9,$-\pi$] and [+$\pi$,+8π/9]. The processes of the "trajectory translation" block are listed as follows:

(1) Translate the longitude $\lambda_{1-l}$ of the first historical viewport $v_{1-l}\,(\lambda, \phi)$ to 0 radian in longitude, and then the rest of the spherical arc $\{\lambda_t\}_{2-l}^{0}$ is translated to $\{\bar{\lambda}_t\}_{2-l}^{0}$ with the same size.



(a) Before translation (b) After translation

Fig. 5: Trajectory translated to 0 radian in longitude.

(2) Estimate the continuity of the $i^{th}$ and the $i-1^{th}$ viewports using the distance $\parallel \bar{\lambda}_i - \bar{\lambda}_{i-1} \parallel$. If the longitude difference between $\bar{\lambda}_{i-1}$ to $\bar{\lambda}_i$ is larger than $\pi$, $\bar{\lambda}_i$ will be rectified with a deviation of $\pm 2\pi$.

The "trajectory translation" block may be formulated as follows:

$$\{\bar{\lambda}_t\}_{t=1-l}^{0} = \{\lambda_t\}_{t=1-l}^{0} - \lambda_{1-l}, \tag{4a}$$

$$\bar{\lambda}_i = \begin{cases} \bar{\lambda}_i + 2\pi & \bar{\lambda}_i - \bar{\lambda}_{i-1} \leq -\pi, \\ \bar{\lambda}_i - 2\pi & \bar{\lambda}_i - \bar{\lambda}_{i-1} \geq \pi, \\ \bar{\lambda}_i & \parallel \bar{\lambda}_i - \bar{\lambda}_{i-1} \parallel < \pi. \end{cases} \tag{4b}$$

Upon completing the trajectory translation, the longitude of the translated historical trajectory will be around 0 radian, thereby avoiding the discontinuity issue. The translated trajectory $\{\bar{v}_t\}_{t=1-l}^{0}$ is shown in Fig. 5b, where its latitude coordinate remains the same.

### *D. Trajectory Selector*

In this section, we detail the "trajectory selector" block of Fig. 4, which generates the motion trajectory $\{o_t^m\}_{t=1}^{k}$ of the "key" object. Again, different viewers may have different types of viewports, as shown in Fig. 6, where three types of viewports are exemplified. The considered panoramic frame $f_0$ contains multiple (six) foreground objects, where the location of the $i^{th}$ object is denoted as $o_0^i$. We summarize these types of viewports as follows:

- Single object: There is only one object in the current viewport $v_0$, which means that only a single object is visible to the viewer. Therefore, this single object is the "key" object.
- Multiple objects: There are multiple objects in the current viewport $v_0$. The object closest to the viewport center is deemed as the "key" object, since viewers tend to focus on the center of their visual field.
- None object: None object is visible to the viewer, which may be interpreted as that the viewer is switching his "key" object.

[2] Viewers may focus on some specific objects, such as a movie star, a car or an animal. In this paper, we refer to it as the "key" object.

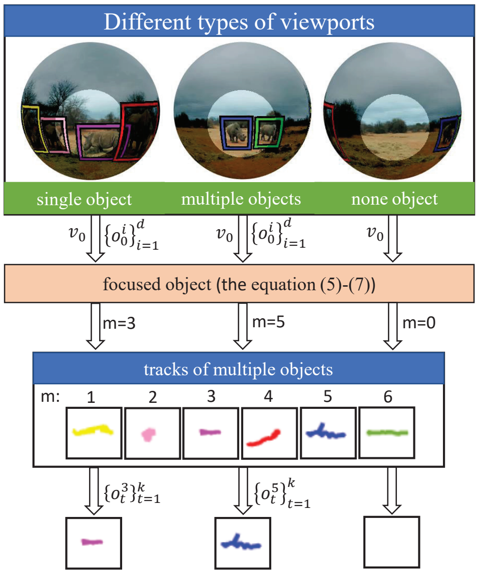


Fig. 6: Different types of viewports and the "key" object selection.

Given the locations $\{o_0^i\}_{i=1}^d$ of $d$ objects in frame $f_0$, the "key" object $o_0^m$ may be found by minimizing the spatial angle $\alpha_i$, which is calculated as

$$\alpha_i = \arccos\left(v_0\left(x,y,z\right)\odot o_0^i\left(x,y,z\right)\right), \tag{5}$$

where the symbol $\odot$ means vector dot product. Additionally, $v_0\left(x,y,z\right)$ and $o_0^i\left(x,y,z\right)$ are the corresponding planar coordinates of viewport $v_0\left(\lambda,\phi\right)$ and object $o_0^i(\lambda,\phi)$, where the Geographic coordinates may be transformed to the Cartesian coordinates as follows

$$\begin{aligned} x &= \cos\phi\cos\lambda, \\ y &= \cos\phi\sin\lambda, \\ z &= \sin\phi. \end{aligned} \tag{6}$$

Then, the "key" object is given by

$$\begin{aligned} \alpha_m = \min\left\{\alpha_1,...,\alpha_i,...,\alpha_d\right\}, \\ s.t. \quad \alpha_m \le \pi/2. \end{aligned} \tag{7}$$

Therefore, the index of the "key" object is $m$, while its corresponding trajectory is expressed as $\{o_t^m\}_{t=1}^k$. The "key" object locates in the current viewport $v_0$. In the "none object" case, the predicted viewport trajectory may be independent of these objects, since the "trajectory selector" block may not output any motion trajectory.

## III. Experiments

In this section, we conmerce by defining the the evaluation metrics, followed by introducing the benchmarkers. Afterwards, we describe the panoramic video dataset employed in the simulations. Finally, the system performance is presented and analyzed.

### A. Evaluation Metrics

Since the predicted viewport $\hat{v}_k$ and the ground truth $v_k$ are on the sphere surface, three-dimensional coordinates are required to express the linear growth of the distance. For evaluating the distance between the predicted viewport $\hat{v}_k$ and the ground truth $v_k$ of the $k^{th}$ frame, we define the angle error (AE) as

$$AE = \arccos\left(\hat{v}_k\left(x,y,z\right)\odot v_k\left(x,y,z\right)\right), \tag{8}$$

where $\hat{v}_k\left(x,y,z\right)$ and $v_k\left(x,y,z\right)$ are the Cartesian coordinate version of $\hat{v}_k\left(\lambda,\phi\right)$ and $v_k(\lambda,\phi)$. The range of AE is $[0,\pi]$, while smaller AE indicates more accurate prediction. Furthermore, mean angle error (MAE) may be employed for measuring the average prediction accuracy, which is defined as

$$MAE = \frac{1}{N}\sum_{n=0}^{N-1} AE_n. \tag{9}$$

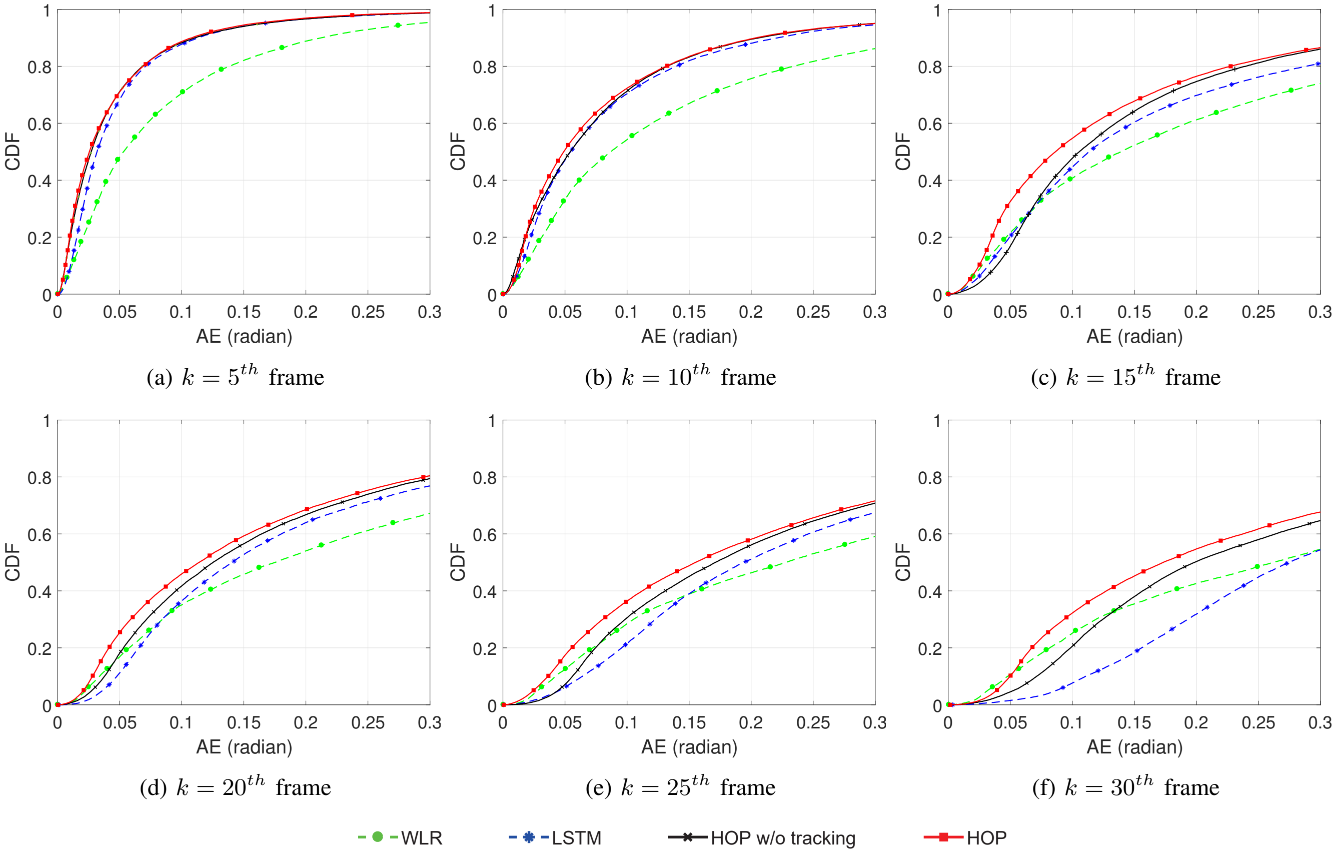
Generally, with smaller AE, lower bandwidth is required for delivering panoramic videos. However, the AE may fluctuate for different panoramic frames. For example, the AE may be more stable, when the viewer is focusing on a slow-moving object. To associate the AE with the fluctuation of the required bandwidth, we define the metric mean square deviation (MSD) to evaluate the stability of AE. The MSD is formulated as

$$MSD = \frac{1}{N}\sum_{n=0}^{N-1}\left(AE_n - MAE\right)^2. \tag{10}$$

### B. Benchmarkers

There is a paucity of contributions on the viewport trajectory prediction. We benchmark our HOP scheme against the weighted linear regression (WLR) [15] and the LSTM based prediction schemes [18].

- WLR [15]: As an enhanced version of linear regression, WLR has unequal coefficient weights, which increases along with the time-axis of the historical viewport trajectory. In other words, temporally closer viewports tend to exhibit higher correlations. However, WLR predicts the future trajectory purely relying on the historical viewport trajectory, while ignoring the features of the panoramic video content.
- LSTM [18]: Considering the expertise of LSTM in sequence generation, the tile-based LSTM network is conceived for viewport prediction using the orientation, saliency map and optical flow map of historical frames. However, the saliency map and the optical flow map are prohibited in this benchmarker.
- HOP without object tracking (HOP w/o tracking): HOP w/o tracking is a simplified version of HOP, where the object tracking part is disabled. This benchmarker is considered for providing further insights into the object tracking part of Fig. 4.



(a) $k = 5^{th}$ frame

(b) $k = 10^{th}$ frame

(c) $k = 15^{th}$ frame

(d) $k = 20^{th}$ frame

(e) $k = 25^{th}$ frame

(f) $k = 30^{th}$ frame

Fig. 7: CDF for different prediction length.

TABLE II: MAE and MSD comparison of benchmarkers

| Method | MAE(radian) | | | | | | MSD(radian) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $5^{th}$ | $10^{th}$ | $15^{th}$ | $20^{th}$ | $25^{th}$ | $30^{th}$ | $5^{th}$ | $10^{th}$ | $15^{th}$ | $20^{th}$ | $25^{th}$ | $30^{th}$ |
| WLR [15] | 0.090 | 0.147 | 0.229 | 0.280 | 0.349 | 0.392 | 0.106 | 0.170 | 0.254 | 0.302 | 0.365 | 0.402 |
| LSTM [18] | 0.052 | 0.095 | 0.192 | 0.232 | 0.317 | 0.404 | 0.067 | 0.120 | 0.222 | 0.270 | 0.368 | 0.391 |
| HOP w/o tracking | **0.048** | **0.090** | **0.168** | **0.207** | **0.276** | **0.321** | **0.066** | **0.115** | **0.186** | **0.235** | **0.302** | **0.329** |
| HOP | **0.047** | **0.088** | **0.152** | **0.195** | **0.261** | **0.298** | **0.067** | **0.115** | **0.189** | **0.239** | **0.304** | **0.340** |

### *C. Dataset*

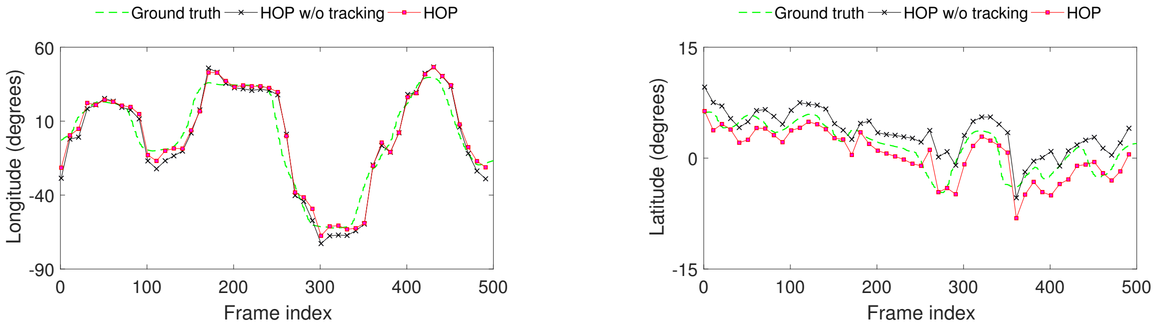
Two datasets [19], [20] are employed, and they contains head movements of 21˜50 viewers when watching panoramic videos. More specifically, each video has a length of 60 to 70 seconds with frames per second (FPS) of 30. In numerous practical applications, the motion objects may occupy a large proportion of panoramic videos, such as sports, crowded streets etc. We mainly focus on these panoramic videos, while panoramic videos without any motion object are not considered. For keeping data consistency, the head movement data represented in four elements [20] is transformed to euler angle.

### *D. Performance*

This section evaluates the performance of our HOP scheme for predicting the future viewports of a single viewer. We benchmark the HOP scheme against the WLR scheme, the LSTM scheme and the HOP w/o tracking scheme. Specifically, the performance of $k = [5, 10, 15, 20, 25,30]$ is presented.

**Evaluation of the CDF of AE**. Fig. 7 shows the cumulative distribution function (CDF) of AE, where the $y$ axis indicates the cumulative probability and $x$ axis means the AE. We observe from Fig. 7 that our HOP scheme outperforms the benchmarkers in all scenarios in terms of the cumulative probability. The CDF value decreases with an increasing $k$, which means that the predicted viewport becomes less accurate, when the predicted frame length is larger. Specifically, in Fig. 7f, the HOP scheme achieves a CDF gain of about 14% in comparison to both of the WLR and LSTM schemes, when the AE ranges from 0 to 0.3.

**Evaluation of the MAE and MSD.** Table II shows the MAE and MSD results of all benchmarkers with various pre-



(a) Longitude

(b) Latitude

Fig. 8: Longitude and latitude comparison of our HOP scheme and the HOP w/o tracking scheme.

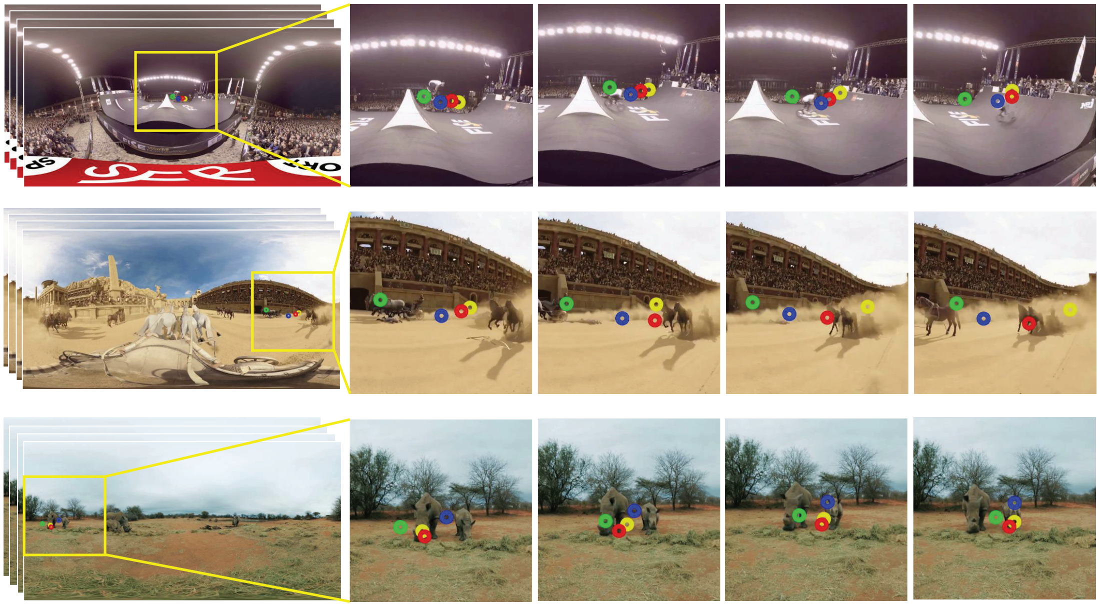


Fig. 9: Comparison of viewport prediction for the $15^{th}$ frame using various panoramic videos: the HOP, WLR and LSTM schemes are benchmarked. The leftmost column shows the original panoramic frames, while the other columns demonstrate the sampled results of predicted viewports within 1 second. The red, green and blue circles indicate the viewports generated by the HOP, WLR and LSTM schemes, respectively, while the yellow circle represents the ground truth viewport.

diction length. In terms of MAE, our HOP scheme outperforms the WLR scheme by 23.9% to 47.7%, while gains of 7.3% to 26.2% are observed against the LSTM scheme, when the prediction length $k$ increases from 5 to 30. In comparison to the HOP w/o tracking scheme, our HOP scheme reduces MAE by 0.023 at the prediction length of 30. In terms of MSD, the HOP w/o tracking scheme exhibits lower MSD than our HOP scheme and outperforms the other benchmakers. Our HOP scheme is only second to the HOP w/o tracking scheme and is more stable in AE compared with the WLR and LSTM schemes. The performance improvement of our HOP scheme is attributed to: (1) Our HOP scheme combines object tracking and the historical viewport trajectory, while the WLR and LSTM schemes generate future viewports purely relying on the historical viewport trajectory. (2) Our HOP scheme may benefit from the trajectory translation for avoiding the "longitude overturn" issue.

**Evaluation of the "trajectory translation".** We benchmark the HOP w/o tracking scheme against the WLR and LSTM schemes. As shown in Table II, by averaging the range of $k = 5$ to 30, the HOP w/o tracking scheme decreases the MAE values by 29.4% and 11.6% in comparison to the WLR and LSTM schemes, respectively, which validates the effectiveness of the trajectory translation. Meanwhile, the HOP w/o tracking scheme decreases the MSD by 25.8% and 11.4% in comparison to the WLR and LSTM schemes, respectively. Therefore, we may benefit from the "trajectory translation" block for decreasing the prediction error while maintaining stable bandwidth consumption.

**Evaluation of the object tracking.** In Fig. 8, we compare longitude and latitude trajectory of our HOP sheme and the

HOP w/o tracking scheme for predicting the future $15^{th}$ viewport. Observe from Fig. 8, the longitude exhibits larger range than latitude does for the 500 frames, which indicates that viewers tend to move more in the horizontal direction instead of vertical direction. When the orientation of the ground truth viewport changes, the HOP w/o tracking scheme exhibits an increasing AE, while our HOP scheme is capable of effectively adjusting the predicted viewport. Hence, our HOP scheme may benefit from the object tracking, especially when the orientation of the viewport trajectory changes.

To provide further insights into our experiments, a variety of panoramic frames are visualized in Fig. 9 for predicting the $15^{th}$ viewport. In the first row of Fig. 9, the main motion object is a fast-moving biker. The LSTM and WLR schemes exhibit modest AE, while our HOP scheme provides more accurate viewport prediction with the aid of the object tracking. In the middle and bottom rows of Fig. 9, the panoramic videos contain multiple objects, where the middle row containing the horses exhibits larger motion in comparison to the bottom row containing a slow-moving rhino. Therefore, the predicted result of the rhino sequence is better than that of the horses panoramic video. In both scenarios, our HOP scheme exhibits better performance than the other schemes. The results reveal that our HOP scheme predicts more accurately than the WLR and LSTM schemes, since the multi-object tracking part is employed.

## IV. Conclusion

In this paper, we proposed our HOP scheme to predict the viewport trajectory for the future frames. The multi-object tracking and the historical viewport trajectory were jointly exploited. Moreover, considering the fact that multiple objects may exist in a panoramic video, an object selection algorithm was designed. Simulation results shows that our HOP scheme substantially outperforms the benchmarkers. Our future work will extend the HOP scheme to three-dimensional panoramic videos, where the depth map may be exploited.

## References

[1] S. Yang, C. Zhou, T. Lv, and L. Hanzo, "Large-scale MIMO is capable of eliminating power-thirsty channel coding for wireless transmission of HEVC/H.265 video," *IEEE Wireless Communications*, vol. 23, pp. 57–63, Jun. 2016.

[2] X. Zhang, T. Lv, and S. Yang, "Near-optimal layer placement for scalable videos in cache-enabled small-cell networks," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 9047–9051, Sep. 2018.

[3] Z. Wang, X. Jin, F. Xue, X. He, R. Li, and H. Zha, "Panorama to cube: A content-aware representation method," in *ACM special interest group on Computer GRAPHics and Interactive Techiques*, vol. 6, pp. 1–4, Nov. 2015.

[4] Y. Sánchez de la Fuente, R. Skupin, and T. Schierl, "Video processing for panoramic streaming using HEVC and its scalable extensions," *Multimedia Tools and Applications*, vol. 76, pp. 5631–5659, Feb. 2017.

[5] Z. Alireza, A. Alireza, M. Miska, and G. Moncef, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *ACM Multimedia*, pp. 601–605, Oct. 2016.

[6] Z. Tu, T. Zong, X. Xi, L. Ai, Y. Jin, X. Zeng, and Y. Fan, "Content adaptive tiling method based on user access preference for streaming panoramic video," in *IEEE International Conference on Consumer Electronics*, pp. 1–4, Jan. 2018.

[7] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *IEEE International Conference on Communications*, pp. 1–7, May. 2017.

[8] G. Mario, T. Christian, and M. Christopher, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *ACM Multimedia Systems Conference*, pp. 261–271, Jun. 2017.

[9] K. Sreedhar, A. Aminlou, M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *IEEE International Symposium on Multimedia*, pp. 583–586, Dec. 2016.

[10] A. TaghaviNasrabadi, A. Mahzari, J. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," in *IEEE Virtual Reality*, pp. 347–348, Mar. 2017.

[11] Y. Chang and H. Chen, "Finding good composition in panoramic scenes," in *IEEE International Conference on Computer Vision*, pp. 2225–2231, Sep. 2009.

[12] Y. Su, J. Dinesh, and G. Kristen, "Pano2vid: Automatic cinematography for watching 360° videos," in *Asian Conference on Computer Vision*, pp. 154–171, Mar. 2016.

[13] H. Hu, Y. Lin, M. Liu, H. Cheng, Y. Chang, and M. Sun, "Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1396–1405, Jul. 2017.

[14] M. Stengel and M. Magnor, "Gaze-contingent computational displays: Boosting perceptual fidelity," *IEEE Signal Processing Magazine*, vol. 33, pp. 139–148, Sep. 2016.

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

[16] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3177, Jun. 2013.

[17] B. Thomas, B. Andrés, P. Nils, and W. Joachim, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, pp. 25–36, May 2004.

[18] C. Fan, J. Lee, W. Lo, C. Huang, K. Chen, and C. Hsu, "Fixation prediction for 360° video streaming in head-mounted virtual reality," in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 67–72, Jun. 2017.

[19] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, "360° video viewing dataset in head-mounted virtual reality," in *ACM Multimedia Systems Conference*, pp. 211–216, Jun. 2017.

[20] X. Corbillon, S. De, and G. Simon, "360 degreee video head movement dataset," in *ACM Multimedia Systems Conference*, pp. 199–204, Jun. 2017.

[21] X. Yu, A. Alexandre, and S. Silvio, "Learning to track: Online multi-object tracking by decision making," in *IEEE International Conference on Computer Vision*, pp. 4705–4713, Dec. 2015.

[22] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.

[23] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 33, pp. 1806–1819, Sep. 2011.

[24] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1201–1208, Jun. 2011.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Neural Information Processing Systems*, pp. 1–9, Dec. 2014.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, pp. 1–15, May 2015.

[28] G. Alex and J. Navdeep, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, vol. 32, pp. 1764–1772, Jan. 2014.