LSTM-based Viewpoint Prediction for Multi-Quality Tiled Video Coding in Virtual Reality Streaming

Mohammadreza Jamali
Department of Software Engineering and IT
École de technologie supérieure
Montréal, Canada
mohammadreza.jamali.1@ens.etsmtl.ca

Stéphane Coulombe
Department of Software Engineering and IT
École de technologie supérieure
Montréal, Canada
stephane.coulombe@etsmtl.ca

Ahmad Vakili
R&D Department
Summit Tech Multimedia
Montréal, Canada
vakili@summit-tech.ca

Carlos Vazquez
Department of Software Engineering and IT
École de technologie supérieure
Montréal, Canada
carlos.vazquez@etsmtl.ca

***Abstract*—Virtual reality (VR) streaming is impaired by the large amount of data required to deliver 360-degree video resulting in low-quality end user experience when network bandwidth is limited, or latency is high. To address these challenges, proposed in this paper is a novel method for viewpoint prediction for long-term horizons in VR streaming. This method uses a long short-term memory (LSTM) encoder-decoder network to carry out a sequence-to-sequence prediction. To enhance the results obtained by this network, experiments are performed using viewpoint information from users on low-latency networks. By applying an effective tile-based quality assignment after viewpoint prediction, a 61% average bandwidth saving, with respect to the transmission of the whole ERP frame, is achieved along with a high-quality viewport rendered to the end user.**

***Index Terms*—Virtual reality streaming, 360-degree video, Long short-term memory (LSTM), Viewpoint prediction, Sequence-to-sequence prediction**

I. Introduction

Virtual reality (VR) offers a unique immersive video experience by providing 360-degree video in a panoramic view. Limited bandwidth, high quality requirements, encoder complexity and network latency are the main challenges to deliver reasonable quality of experience (QoE) to the end users [1]. Although there have been improvements in computing, storage/memory, networks and video coding, the community still needs to deliver improved methods and techniques to overcome the above-mentioned problems. The principal challenge in deploying effective VR streaming technologies is the massive amount of data required to transmit high-quality video to end users. High-resolution video at high frame rates is necessary to offer a genuine immersive experience [1]. The simplest technique to stream VR video is by sending the entire 360-degree frame, which is much larger than the user's viewport, at a uniform quality. This results in sending data for the invisible parts of the frame, i.e. areas outside the viewport, leading to inefficient bandwidth usage. In view of this, despite all the advances made in VR technologies in recent years, there is a strong need for a streaming approach providing efficient bandwidth usage while keeping the quality of the viewport as high as possible.

A successful method in this regard is tile-based video coding [2]–[5]. Tiled coding divides video frames into rectangular regions. Although it was originally designed for parallel coding, it is now used for viewport-aware coding in 360-degree video [6]–[8]. In this approach, tiles are encoded at varying qualities. Those which overlap with the viewport are encoded at the highest quality. Other tiles, which do not overlap with the viewport, are encoded at a lower quality depending on their distance to the viewport. As a result of this non-uniform quality encoding, VR streaming service utilizes the bandwidth efficiently since there is much less data to transmit. A main requirement for a successful viewport-aware tile-based video coding is to determine the viewpoint position in a short period of time [9]. Since there are varying degrees of latency in streaming networks between servers and end users, the server cannot always determine in a timely manner the user's current viewpoint. Thus, it needs to predict the viewpoint position to make the tile-based coding effective. Although it is called *viewport* prediction in the literature, the problem is actually *viewpoint* prediction. By viewpoint, we mean the center of the viewport which is determined by knowing the yaw and pitch angles. Viewport is the area of the frame seen by the user on the end device. In [10], a machine translation model and a convolutional long short-term memory (LSTM) are used for viewpoint prediction. The prediction is made for mean and standard deviation of the viewpoint position over periods of one second (not on a frame level) which is more appropriate for HTTP based adaptive video streaming. In [11], a method has been proposed to predict the motion and the viewpoint position using the past motion of the user as a feature. To this end, three regression models are used: naive, linear regression and neural networks. In [12], a reinforcement learning (RL) method using contextual bandit is used to make the viewpoint

prediction. To show its usability, the method is deployed on an adaptive tile-based VR streaming testbed. In [13], a method has been proposed to predict user's head turnings to optimize the delivery of 360-degree video over cellular networks by sending only the portion of the frame that the user is looking at. This prediction is done using three methods: average, linear regression and weighted linear regression (WLR).

In this paper, a novel method for multi-quality tile-based VR streaming is proposed. This method is based on viewpoint prediction using an LSTM encoder-decoder network along with viewpoint information form users on low-latency networks (nearly zero-latency). After viewpoint prediction, a tile-based quality assignment is applied to send tiles in a frame in three different qualities. In the prediction phase, we use a sequence-to-sequence prediction model. The input of this model is a sequence of viewpoint positions and the output of the model is the predicted viewpoint positions. The proposed method considers users on heterogeneous networks with different and variable latencies and performs the viewpoint prediction for a long-term horizon of up to 4 seconds. This allows streaming high-quality VR content to users on low to high-latency networks. In addition, the proposed method relies only on users' viewpoints and not on content processing. In other words, it does not analyze video content to determine the regions of interest. This leads to a low-complexity method which does not burden the encoder/server with heavy computations.

The remainder of this work is organized as follows. In section II, our proposed method for high-quality VR streaming is introduced. Section III presents the experimental results and finally, section IV concludes the paper.

II. Proposed Method

To send the viewport in high quality, the encoder/server should be able to know the user's viewpoint at the time of tile quality assignment. Since, at this time, for users on high-latency networks, the viewpoint information is not available, viewpoint prediction is an inevitable step for tile-based streaming. In the following, a method for high-quality VR streaming is presented based on a viewpoint prediction approach and a tile-based quality assignment policy.

A. Viewpoint prediction

In this section, we develop a sequence-to-sequence predictive model using an LSTM encoder-decoder which achieves excellent performance in sequence-to-sequence prediction problems [14]. The predictive model takes the user's viewpoint position history as a sequence and predicts the future viewpoint positions as a sequence as well. Figure 1 shows the architecture of the network used for viewpoint prediction. The encoder takes a sliding window of M frames' features (yaw or pitch angles) and a time distributed dense layer, implemented after the decoder, outputs a prediction window of N frames' features (yaw or pitch angles). We thus obtain the predicted viewpoint for future frames ranging from the next one up to the N^{th} one. This allows the system to support users with different and variable latencies.

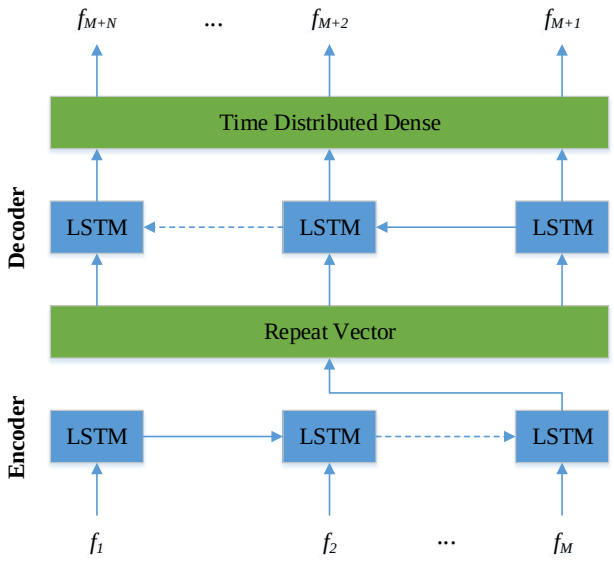

Fig. 1: LSTM encoder-decoder network for predicting yaw and pitch angles. f_1 to f_M are M input frames' features (yaw or pitch angles) and f_{M+1} to f_{M+N} are N output frames' features (yaw or pitch angles).

As reported in the literature, roll angle is most of the time around zero degree [11]. Thus, in this work, the prediction is made based on the viewpoint and only for yaw and pitch angles. Moreover, yaw and pitch angles are predicted as two independent variables since autocorrelation is much stronger than the correlation between them [11]. Since angles -180 and 179 are only one degree distant, we use sine and cosine of yaw and pitch angles to map them on a unit circle which makes it easier for the network to understand the spherical nature of frames in equirectangular projection (ERP) (e.g, when we cross the left border we enter from the right border). Since sine and cosine functions are restricted between -1 and 1, we use *tanh* as the activation function of all layers of the network. In the rest of this work, we use the yaw notation and discuss the process for yaw angle but the same applies to the pitch angle as well.

B. Prediction adjustment

Although LSTM is an appropriate model for sequence prediction, the prediction error increases significantly for long-term horizons. The error occurs because the user's current head direction is not an accurate predictor for the directions in the next 3-4 seconds. Content usually changes for long-term horizons resulting in new attractive regions which makes users turn their heads unexpectedly. This makes it difficult to carry out predictions far in the future.

To improve the results obtained by the LSTM model, we consider an additional step. Assuming that some users are on low-latency networks, and that they are viewing the same

content, their viewpoint information is used to adjust the output of the predictive model. The adjustment is made based on circular mean and circular variance [15] of these *guide users'* yaw and pitch angles. The new adjusted yaw angle is computed as:

$$\begin{aligned} Y_a &= (1-W) \times Y_p + W \times Y_g \\ W &= (1-V_g^{1/3}) \times (h/P_{max}) \\ & 0 \leq V_g \leq 1 \\ & 0 \leq h \leq P_{max}, \end{aligned} \tag{1}$$

where Y_p is the predicted yaw angle obtained by the LSTM model, Y_g is the average yaw angle of the guide users and Y_a is the adjusted yaw angle of a target user. V_g is the circular variance [15] of guide users' viewpoints. W is a confidence factor, between 0 and 1, of the guide users' information. W is high when the circular variance is low (guide users look at a similar positions) and we predict far in the future where LSTM is unlikely to perform well. A low variance suggests that there is one region of interest in the frame. On the other hand, a high variance means it is less likely that there is one region of interest; thus, the viewpoint predicted by the LSTM network is considered more important by applying a larger weight. h is the time horizon (the frame in the future we are predicting the viewpoint for) and P_{max} is the maximum size of the prediction window. Applying this time horizon term makes sure that when the prediction is made for the near future, the target user's LSTM-predicted viewpoint has a dominant role and when the prediction is made for the far future, given the guide users' variance is low, the guide users are considered more important.

C. Viewport generation

To generate the predicted viewport, we use the rectilinear projection. This projection, which is also called 'gnomic', 'gnomonic' and 'tangent-plane', is used to map a part of a sphere surface to a flat plane [16]. Figure 2 shows how a viewport is generated using rectilinear projection. For viewport generation the viewpoint angle is assumed to be along the Z axis. When this is not the case, the sphere is rotated to align the viewpoint with the Z axis, and then viewport generation is carried out. Viewport generation starts from pixels on the corresponding 2D coordinates in the source projection plane (ERP in our problem) which are mapped to the corresponding 3D (X, Y, Z) coordinates on the sphere surface. Then, every pixel is processed to see whether it is on the viewport plane generated by rectilinear projection. If it is on this plane, it is considered part of the viewport. At the end, all pixels belonging to the viewport on the 2D frame are known.

D. Tiles quality assignment

After viewpoint prediction and viewport generation, different qualities are assigned to the frame tiles. We consider three levels of quality. Tiles that overlap with the predicted viewport will be sent with the highest quality. Their neighboring tiles are sent with the second level of quality and the remaining

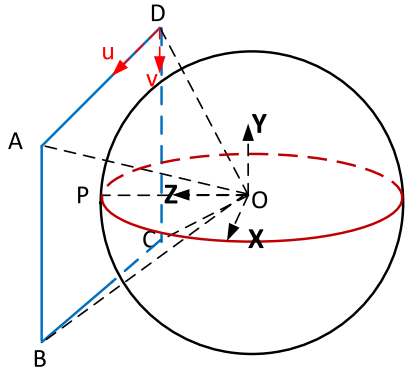

Fig. 2: Viewport generation with rectilinear projection [16].

tiles are sent with the lowest quality. In our experiments the viewport is an area with a width of 110 degrees and a height of 90 degrees [13]. The frame, in ERP format, has a height of 180 degree and a width of 360 degree.

To compute the bandwidth saving, we consider the non-tiled streaming sent at b Mbps. For the tiled streaming, each of the highest quality tiles is sent at $b/(m \times n)$ Mbps, where $(m \times n)$ is the number of tiles. Each of the second-level tiles is sent at $b/(2 \times m \times n)$ Mbps and each of the third-level tiles is sent at $b/(8 \times m \times n)$ Mbps. This assures that when we have prediction errors, the user still sees the content (although at a lower quality) instead of a black image as is the case if only the predicted viewport's tiles are sent.

III. Experimental Results

In this section, we present the results of our experiments by comparing our proposed method to some baseline methods. Thus the results are reported for 'linear regression' [11], [13], 'persistence' [10], 'LSTM encoder-decoder' and 'LSTM + guide users'. In the persistence prediction the last know viewpoint is used as predictor, which means it assumes a no-motion model for the user. In linear regression, a liner model is created to fit all the data points in the sliding window and then this model is used to predict the future viewport trajectory.

The results are reported based on root mean square error (RMSE) between the real and predicted viewpoint positions, viewport overlap with high-quality and medium-quality tiles, and bandwidth saving. For our experiments, we consider a dataset provided in [17] since the content in this experiment is related to live VR streaming such as a basketball match or a talk show. This dataset includes 48 users watching 9 videos. For each video, one minute of the content is selected. We consider 7 videos for training and two videos for testing (Basketball Match and Showtime Boxing). We conduct 48 experiments. In each of them one user is considered on a high-latency network and the other 47 users on a low-latency network. Then the average results over these 48 experiments are taken as the final results to provide a cross validation over all users. The data in this dataset is provided in unit quaternion which shows the head-mounted display's (HMD) direction.

TABLE I: Prediction RMSE error and high- and medium-quality viewport for long-term horizon (3.6 s)

Method	Yaw RMSE (°)	Pitch RMSE (°)	HQV(%)	HQV + MQV(%)
Linear regression	41.6	13.1	70.2	93.8
Persistance	26.4	4.2	84.4	98.6
LSTM enc.-dec.	23.4	4.0	87.6	98.9
LSTM + guide users	16.5	3.6	95.9	99.7

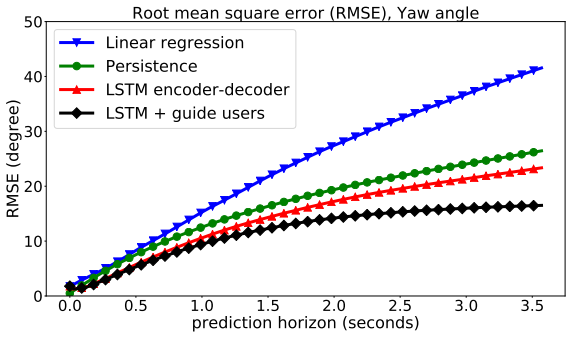

Fig. 3: Prediction error based on RMSE for the entire prediction window.

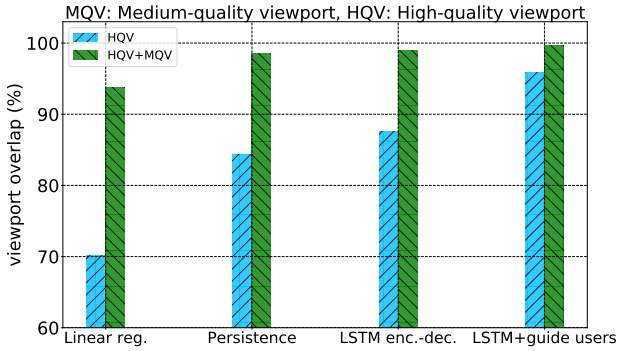

Fig. 4: Viewport overlap with high- and medium-quality regions for long-term horizon (3.6 s).

We convert these coordinates to X, Y and Z on a unit sphere and then to yaw and pitch angles. To train our model and to make predictions, we use the sine and cosine of yaw and pitch angles. Input sliding window and output prediction window are set to 20 and 120 frames respectively (M=20, N=120) and P_{max} is set to 200. A grid of 12×12 tiles is applied to each frame in these experiments. The videos are sampled at every 30 ms which means we are using a window of 600 ms to predict all the frames in the next 3.6 s. This makes our method suitable for a long-term horizon prediction and helps to deliver high-quality VR content to users on high-latency networks.

Based on our tiles quality assignment method discussed in section II-D, the average bandwidth saving for all four methods is the same and is equal to 61%. Bandwidth saving is not dependent on the viewport prediction method but on the target bitrate ratio for each quality level. Figure 3 shows the yaw prediction error based on RMSE, for the entire prediction window. Pitch error is negligible and does not significantly affect the quality. Table I shows the error for both yaw and pitch angles for a long-term horizon (3.6 s). This table also shows high-quality viewport (HQV) and the combination of HQV and medium-quality viewport (MQV). HQV and MQV show the overlap (in percentage) between the viewport seen by the user and the high-quality and medium-quality tiles, respectively. This table shows that 'LSTM + guide users' is highly effective when it is applied to a distant future. In addition, Fig. 4 shows HQV and HQV+MQV as a bar graph for a long-term horizon. Based on the experimental results, our proposed method achieves a high-accuracy prediction and delivers a high-quality viewport. It is only based on viewpoint trajectories which makes it a low-complexity method as it does not need expensive content processing computations or high-complexity saliency detection algorithms. Compared to [10], while their method is proposed for on demand streaming and video streamed in segments, our method is applicable to both on demand and live streaming. Moreover, they assume the viewport is fixed over a period of one second and make the prediction for mean and standard deviation of the viewpoint over each second, while in our method the prediction is made for each frame which makes it possible to adapt faster to the users' head movements. Compared to [11], we predict a long-term horizon (3.6 s) while they make predictions for the 100-500 ms in the future. In summary, our proposed method predicts future viewpoint positions of up to 4 seconds. Thus all frames in the prediction window can be encoded using the predicted positions. It is scalable and latency-adaptive as it is able to predict multiple frames ahead.

IV. Conclusion

In this paper, we proposed a sequence-to-sequence LSTM prediction model for the problem of viewpoint prediction in virtual reality streaming. To enhance the results obtained by this model, we used guide users' viewpoint information to adjust the output of the predictive model. These guide users are assumed to be on low-latency networks while target users are on high-latency networks. Based on this prediction, a tile-based quality assignment is proposed to send tiles of a frame in three different qualities resulting in a significant bandwidth saving. By providing a large overlapping area between the viewport and high-quality tiles, the proposed method delivers a high-quality VR experience to the end users.

REFERENCES

[1] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, pp. 110–117, June 2017.

[2] M. Jeppsson, H. Espeland, C. Griwodz, T. Kupka, R. Langseth, A. Petlund, P. Qiaoqiao, C. Xue, K. Pogorelov, M. Riegler, D. Johansen, and P. Halvorsen, "Efficient live and on-demand tiled HEVC 360 VR video streaming," in *2018 IEEE International Symposium on Multimedia (ISM)*, pp. 81–88, Dec 2018.

[3] J. Fu, X. Chen, Z. Zhang, S. Wu, and Z. Chen, "360SRL: A sequential reinforcement learning approach for ABR tile-based 360 video streaming," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 290–295, July 2019.

[4] S. Zhao and D. Medhi, "SDN-assisted adaptive streaming framework for tile-based immersive content using MPEG-DASH," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–6, Nov 2017.

[5] L. Jongmin, L. Joohyung, L. Jeongyeon, and K. Maro, "Bandwidth-efficient live virtual reality streaming scheme for reducing view adaptation delay," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 1, pp. 291–304, 2019.

[6] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC tile based streaming to head mounted displays," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 613–615, Jan 2017.

[7] C. Ozcinar, J. Cabrera, and A. Smolic, "Viewport-aware omnidirectional video streaming using visual attention and dynamic tiles," in *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, Nov 2018.

[8] C. Ozcinar, J. Cabrera, and A. Smolic, "Visual attention-aware omnidirectional video streaming using optimal tiles for virtual reality," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, pp. 217–230, March 2019.

[9] J. Jerald, "Latency compensation for head-mounted virtual reality," Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[10] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very long term field of view prediction for 360-degree video streaming," *CoRR*, vol. abs/1902.01439, 2019.

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

[12] J. Heyse, M. T. Vega, F. de Backere, and F. de Turck, "Contextual bandit learning-based viewport prediction for 360 video," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 972–973, March 2019.

[13] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ATC '16, (New York, NY, USA), pp. 1–6, ACM, 2016.

[14] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[15] "Circular data analysis." NCSS, https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Circular_Data_Analysis.pdf, [Accessed: 1-Oct-2019].

[16] Y. Ye, E. Alshina, and J. Boyce, "Algorithm descriptions of projection format conversion and video quality metrics in 360lib version 5," *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*.

[17] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, (New York, NY, USA), pp. 193–198, ACM, 2017.