HAL
open science

# TRACK: A Multi-Modal Deep Architecture for Head Motion Prediction in 360-Degree Videos

Miguel Fabian Romero Rondon, Lucile Sassatelli, Ramon Aparicio-Pardo, Frédéric Precioso

## To cite this version:

Miguel Fabian Romero Rondon, Lucile Sassatelli, Ramon Aparicio-Pardo, Frédéric Precioso. TRACK: A Multi-Modal Deep Architecture for Head Motion Prediction in 360-Degree Videos. ICIP 2020 - IEEE International Conference on Image Processing, Oct 2020, Abu Dhabi / Virtual, United Arab Emirates. hal-02615980

**HAL Id: hal-02615980**

**https://hal.science/hal-02615980**

Submitted on 23 Jul 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# TRACK: A MULTI-MODAL DEEP ARCHITECTURE FOR HEAD MOTION PREDICTION IN 360° VIDEOS

*Miguel Fabián Romero Rondón, Lucile Sassatelli, Ramón Aparicio Pardo, Frédéric Precioso*

Université Côte d'Azur, CNRS, I3S

## ABSTRACT

Head motion prediction is an important problem with 360° videos, in particular to inform the streaming decisions. Various methods tackling this problem with deep neural networks have been proposed recently. In this article, we introduce a new deep architecture, named TRACK, that benefits both from the history of past positions and knowledge of the video content. We show that TRACK achieves state-of-the-art performance when compared against all recent approaches considering the same datasets and wider prediction horizons: from 0 to 5 seconds.

***Index Terms*—** 360° videos, head motion prediction, deep recurrent networks, content analysis

## 1. INTRODUCTION

Despite the exciting prospects of Virtual Reality (VR), the development is persistently hindered by the difficulty to access immersive content through Internet streaming. Indeed, owing to the closer proximity of the screen to the eye and to the width of the content, the data rate to stream 360° videos may be two orders of magnitude that of a regular video [1]. To decrease the amount of data to stream, a solution is to send in high resolution only the portion of the sphere the user has access to at each point in time, named the Field of View (FoV). To do so, recent works have proposed to either segment the video spatially into tiles and set the quality of the tiles according to their proximity to the FoV [2], [3], [4], or use projections enabling high resolutions of regions close to the FoV [5], [6]. These approaches however require to know the user's head position in advance, that is at the time of sending the content from the server. Various methods tackling this problem with deep neural networks have therefore been proposed in the last couple of years (e.g., [7, 8, 9, 10, 11]).

**Contributions**:

• We propose a new architecture, TRACK, that (i) processes individually the time series of positions, (ii) processes the saliency map estimated from the content with recurrent unit to vary its importance over the prediction window, and (iii) then merges these two embeddings.

• We show that TRACK indeed benefits both from the position and the video modalities, achieving state-of-the-art performance on all considered datasets and wide prediction horizons: from 0 to 5 seconds.

In our concern for reproducibility, the entire code used to generate the results, as well as the experimental setup and datasets of each assessed method are detailed and available online at [12] to generate the presented results and plots.

## 2. REVIEW OF EXISTING HEAD MOTION PREDICTION METHODS

We first rigorously formulate the prediction problem which consists, at each video playback time $t$, in predicting the future user's head positions between $t$ and $t+H$, as represented in Fig. 1, with the only knowledge of this user's past positions and the (entire) video content. We then provide a description and classification of each of the existing methods we compare with.

### 2.1. Problem formulation

Let us first define some notation. Let $\mathbf{P}_t = [\theta_t, \varphi_t]$ denote the vector coordinates of the FoV at time $t$. Let $\mathbf{V}_t$ denote the considered visual information at time $t$: depending on the models' assumptions, it can either be the raw frame with each RGB channel, or a 2D saliency map resulting from a pre-computed saliency extractor (embedding the motion information). Let $T$ be the video duration. We now refer to Fig. 1. Let H be the *prediction horizon*. We define the terms *prediction step* and video *time-stamp* as predicting for all *prediction steps* $s \in [0, H]$ from video *time-stamp* $t$. For every *time-stamp* $t \in [T_{start}, T]$, we run predictions $\hat{\mathbf{P}}_{t+s}$, for all *prediction steps* $s \in [0, H]$.

We formulate the problem of trajectory prediction as finding the best model $\mathbf{F}_H^*$ verifying:

$$\mathbf{F}_H^* = \arg\min \mathbb{E}_t \Big[ D\Big( \big[\mathbf{P}_{t+1}, \dots, \mathbf{P}_{t+H}\big], \mathbf{F}_H\big( [\mathbf{P}_t, \mathbf{P}_{t-1}, \dots, \mathbf{P}_0, \mathbf{V}_{t+H}, \mathbf{V}_{t+H-1}, \dots, \mathbf{V}_0] \big)\Big)\Big]$$

The authors are with Université Côte d'Azur, CNRS, I3S, 06900 Sophia Antipolis, France. E-mail: {first.last}@univ-cotedazur.fr. Lucile Sassatelli is also with Institut Universitaire de France. This work has been supported by the French government, through the UCA JEDI and EUR DS4H Investments in the Future projects ANR-15-IDEX-0001 and ANR-17-EURE-0004.

where $D(\cdot)$ is the chosen distance between the ground truth series of the future positions and the series of predicted positions given by the prediction model $F_H$. For each $s$, we average the errors $dist(\hat{\mathbf{P}}_{t+s}, \mathbf{P}_{t+s})$ over all $t \in [T_{start}, T]$.

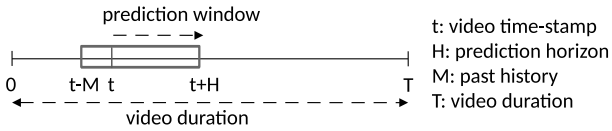


**Fig. 1**. For each time-stamp $t$, the next positions until $t + H$ are predicted.

### 2.2. Methods for head motion prediction

Various approaches to predict user motion in 360° video environments have been published in the last couple of years. Here we consider that the users' statistics for the specific video are *not* known at test time, hence we do not consider methods relying on these per-video statistics, such as [13, 14]. Each considered method from the literature is named according to the name of the conference or journal it was published in, appended with the year of publication.

**PAMI18**: Xu et al. in [7] design a Deep Reinforcement Learning model to predict head motion. Their deep neural network only receives the viewer's FoV and has to decide to which direction and with which magnitude the viewer's head will move. The prediction horizon is only one frame, around 30ms. By only injecting the FoV, the authors make the choice not to consider the positional information explicitly as input.

**IC3D17**: The strategy presented by Aladagli et al. in [15] extracts the saliency from the current frame with an off-the-shelf method, identifies the most salient point, and predicts the next FoV to be centered on this most salient point. It then builds recursively. We therefore consider this method to be a sub-case of PAMI18 to which we compare.

**ICME18**: Ban et al. in [16] use a linear regressor first learned to get a prediction of the displacement, which it then adjusts by computing the centroid of the k nearest neighbors corresponding to other users' positions at the next time-step, and hence assume more information than our case of study.

**CVPR18**: In [8], Xu et al. predict the gaze positions over the next second in 360° videos based on the gaze coordinates in the past second, and saliency and motion information on the entire equirectangular projection and the FoV of the current frame and next frame.

**MM18**: Nguyen et al. in [9] first construct a saliency model based on a deep convolutional network and named PanoSalNet. The so-extracted saliency map is then fed, along with the position encoded as a mask, into a doubly-stacked LSTM, to finally predict the tiles that pertain to the FoV.

**NOSSDAV17**: Fan et al. in [11] propose to concatenate the positional information, saliency and motion maps then fed into LSTM, they propose two networks to predict the head orientations or the likelihood that tiles pertain to future FoV in the future $H$ time-steps.

**ChinaCom18**: Li et al. in [10] present a similar approach as NOSSDAV17, adding a correction module to compensate for the fact that tiles predicted to be in the FoV may not correspond to the actual FoV shape. We consider this model to be a sub-case of NOSSDAV17 to which we compare.

**Selected methods for comparison.** The above methods can be classified into two groups, depending on the way the fusion of the visual features ($V_t$) and the positional features ($P_t$) is handled: either both position and visual information are fed to a single recurrent unit (case of MM18, Chinacom18, NOSSDAV17), or the time series of positions is first processed with a dedicated recurrent unit, the output of which then gets fused with visual features (case of CVPR18). In Sec. 4 we compare with the following methods: PAMI18, NOSSDAV17, MM18 and CVPR18, the remaining methods are either considered sub-cases of the selected methods or assume there is more information available than for our case study. However, all relevant comparisons can be found online at [12]. In Sec. 4.1, we consider the most recent representatives of both groups: MM18 and CVPR18.

## 3. OUR ARCHITECTURE: TRACK

Let us first present the analysis we make of the problem, to build our architecture. For each prediction task starting from time $t$ until $t+H$, we assume in the first prediction time steps, the user's motion is mostly driven by inertia. Hence it should be most predictable from the time series of the past positions. In the later prediction time steps however, the motion may be diverted by attractors from the content. Therefore the saliency map (or features from the content generally) should help the prediction. However, it is important that the visual features do not noise the prediction that a simple linear prediction could make for the first milliseconds.

We therefore make the following assembly, depicted in Fig. 2. First, we produce a position-only embedding by processing the time series of positions with a dedicated recurrent unit (LSTMs). Second, if the content-based (CB) saliency holds relevant (yet noisier) information that may help the prediction in the late prediction time-steps, then having a dedicated recurrent unit to process the visual features (saliency) before merging them with the positional embedding should help. Indeed, it would let information from the saliency map enter the fusion layers in the late time-steps only, fading it in the first prediction time-steps. Third, using another recurrent unit in the fusion layer would provide more lever to enable a time-dependent correction of the inertia-based prediction with the saliency information. Making these architectural choices, we come up with the architecture depicted in Fig. 2.

The main components of TRACK are (i) a doubly-stacked LSTM with 256 units each, processing the flattened CB-

saliency map pre-generated for each time-stamp; (ii) another set of doubly-stacked LSTM with 256 units to process the head orientation input; (iii) a third set of doubly-stacked LSTM with 256 units to handle the multimodal fusion; and finally (iv) a Fully-Connected (FC) layer with 256 and a FC layer with 3 neurons is used to predict the head displacement on the (x,y,z) coordinates. We assembled this building block into a seq2seq framework to get the prediction for all time-steps in the prediction window, as shown in Fig. 2. We trained our model for 500 epochs, with a batch size of 128, we used Adam optimization algorithm with a learning rate of $0.0005$ and we used the Mean Squared Error of the 3D coordinates $(x, y, z) \in \mathbb{R}^3$ as loss function to obtain the results.

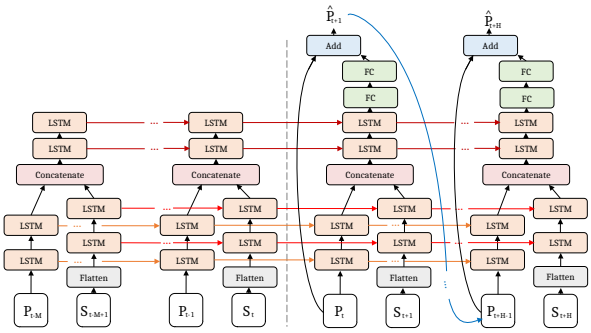


**Fig. 2**. The proposed TRACK architecture.

## 4. RESULTS

We now present the comparisons of the state-of-the-art methods presented in Sec. 2.2 with the proposed TRACK architecture defined above. We report the exact results of the original articles, along with the results of our network trained and tested on the exact same train and test subsets of the original dataset as the original method. The test metrics are those from the original articles, so are the considered prediction horizons. For the sake of completeness, we summarize below information on the metrics of each experiment.

• **NOSSDAV17** [11] considers the following metrics:

− *Accuracy*: ratio of correctly classified tiles to the union of predicted and viewed tiles.

− *Ranking Loss*: number of tile pairs that are incorrectly ordered by probability normalized to the number of tiles.

− *F-Score*: harmonic mean of *precision* and *recall*, where *precision* is the ratio of correctly predicted tiles to the total number of predicted tiles, and *recall* is the ratio of correctly predicted tiles to the number of viewed tiles.

• **PAMI18** [7] uses as metric the Mean Overlap (MO) defined as: $MO = \dfrac{A(FoV_p \cap FoV_g)}{A(FoV_p \cup FoV_g)}$, where $FoV_p$ is the predicted FoV, $FoV_g$ is the ground truth FoV, and $A(\cdot)$ is the area of a panoramic region.

• **CVPR18** [8] uses the Intersection Angle Error $IAE$ for each gaze point $(\theta, \varphi)$ and its prediction $(\hat{\theta}, \hat{\varphi})$, defined as $IAE = arccos(\langle P, \hat{P} \rangle)$, where $P$ is the 3D coordinate in the unit sphere:
$P = (x, y, z) = (cos(\theta)cos(\varphi), cos(\theta)sin(\varphi), sin(\theta))$.

• **MM18** [9] takes the tile with the highest viewing probability as the center of the predicted viewport, and assigns it and all the neighboring tiles that cover the viewport, label 1. Tiles outside the viewport are assigned 0. Then, the score is computed on these labels as $IoU = TP/TT$, the intersection between prediction and ground-truth of tiles with label 1 ($TP$) over the union of all tiles with label 1 in the prediction and in the ground-truth ($TT$).

Results for PAMI18 are shown in Table 1, for CVPR18 in Fig. 3-Left, for MM18 in Fig. 3-Right, and for NOSSDAV17 in Table 2. Given these results, we can conclude that our model outperforms all existing methods by running the same experiments with the same settings, metrics and datasets of the original works. For the case of CVPR18, on which our results are much similar, we emphasize the significantly higher complexity of this model compared to ours. To better compare with CVPR18 and its architectural choices, we propose a more challenging configuration with a longer prediction horizon, as presented next.

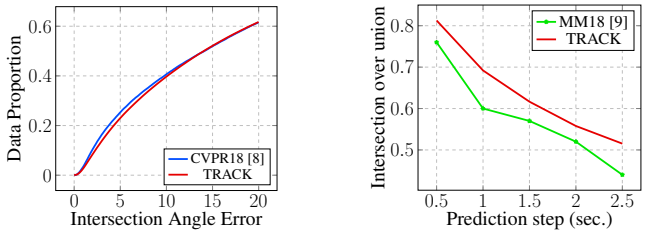


**Fig. 3**. **Left:** Comparison with CVPR18 [8], prediction horizon $H = 1$ sec. **Right:** Comparison with MM18 [9], $H = 2.5$ sec.

### 4.1. Results on a more challenging configuration

To better compare with CVPR18 and MM18 (representatives of the groups defined in Sec. 2.2), we consider more challenging settings:

• **Longer prediction horizon**: $H = 5$ sec. This way, both short-term where the motion is mostly driven by inertia, and long-term where the saliency impacts the trajectory, are covered.

• **Dataset with higher user motion**: we consider the dataset from [17], where after 5 sec., 50% of the users have shifted their FoV by more than its width (100°), while in the datasets of CVPR18, NOSSDAV17, MM18 and PAMI18, the percentage is 30%, 20%, 20% and 15%, respectively.

As for MM18, we consider that the saliency map is estimated from the video content using PanoSalNet [18, 9] for CVPR18 and TRACK. This way, we identify the most suited architectural assembly to fuse position and visual content modalities:

| Method | KingKong | SpaceWar2 | StarryPolar | Dancing | Guitar | BTSRun | InsideCar | RioOlympics | SpaceWar | CMLauncher2 | Waterfall | Sunset | BlueWorld | Symphony | WaitingForLove | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAMI18 [7] | 0.809 | 0.763 | 0.549 | 0.859 | 0.785 | 0.878 | 0.847 | 0.820 | 0.626 | 0.763 | 0.667 | 0.659 | 0.693 | 0.747 | 0.863 | 0.753 |
| TRACK | **0.974** | **0.964** | **0.912** | **0.978** | **0.968** | **0.982** | **0.974** | **0.965** | **0.965** | **0.981** | **0.972** | **0.964** | **0.970** | **0.969** | **0.977** | **0.968** |

**Table 1**. Comparison with PAMI18 [7]: Mean Overlap scores of FoV prediction, prediction horizon $H \approx 30ms$ (1 frame).

| Method | Accuracy | F-Score | Rank Loss |
|---|---|---|---|
| NOSSDAV17-Tile [11] | 84.22% | 0.53 | 0.19 |
| NOSSDAV17-Orient. [11] | 86.35% | 0.62 | **0.14** |
| TRACK | **95.48%** | **0.85** | 0.15 |

**Table 2**. Comparison with NOSSDAV17: Performance of Tile- and Orientation-based networks of [11] compared with our TRACK network, prediction horizon $H = 1$ second.

(i) MM18 first concatenates both modalities then processes them with the same recurrent unit, (ii) CVPR18 dedicates a recurrent unit to process the positions before merging these features with saliency with a Fully Connected (FC) layer, and (iii) TRACK first processes separately both modalities with a dedicated recurrent unit for each, before merging the outputs not with an FC layer, but with another recurrent unit. PanoSalNet is not the saliency extractor considered in the original CVPR18 [8] (whose implementation is not available online nor was communicated on request). Therefore, we also plot a lower bound on its prediction error by plotting the performance of the CVPR18 assembly when provided with the Ground-Truth saliency (line denoted GT Sal-CVPR18 in Fig. 4). The Ground-Truth saliency is the 2D distribution of the viewing patterns, obtained from the users' traces.

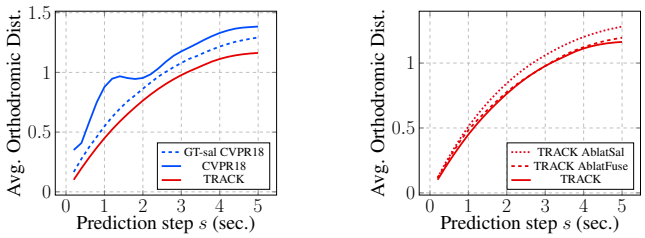


**Fig. 4**. **Left:** Comparison of TRACK with CVPR18 over a longer prediction horizon, $H$ =1 sec., on the dataset of [17]. **Right:** Ablation study of TRACK, same prediction horizon and dataset.

In Fig. 4-Left, the results of MM18 are omitted for clarity because they are all worse than CVPR18 and TRACK (but the complete figures can be generated from [12]). Fig. 4 shows that our proposed architecture TRACK is able to significantly outperform all state-of-the-art methods for both short- and long-term predictions, on the previous datasets and this new dataset. This is remarkable.

Finally, we perform an ablation study of TRACK to confirm the analysis that led us to introduce this new architecture TRACK. We either replace the LSTMs processing the saliency with two FC layers (line named AblatSal), or replace the fusion LSTMs with two FC layers (line named AblatFuse). The results are shown in Fig. 4-Right. They confirm the analysis we developed in Sec. 3: the greater degradation with AblatSal shows that the major improvement of TRACK over CVPR18 comes from the recurrent unit (a doubly-stacked LSTM) dedicated to pre-processing the estimated saliency map, and vary its weight in the prediction over time. This is all the more clear in Fig. 5 where the results are not averaged over all the videos anymore, but instead show two videos that can be categorized as *Ride* and *Static focus*, according to the taxonomy introduced in [19]. In these categories, the user's position is much predictable from the content, therefore it is important not to cancel out the noised information the estimated saliency holds, and that can help for late prediction time-steps. We can see that when there is not pre-processing of the saliency with a recurrent unit, the performance severely degrades.

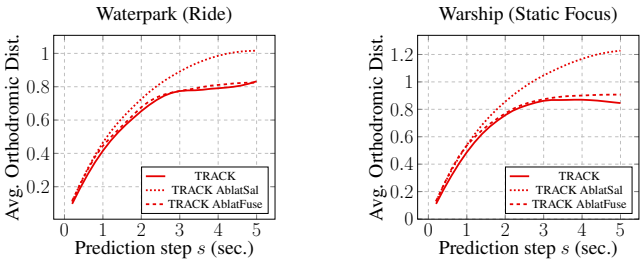


**Fig. 5**. Ablation of TRACK for specific video categories with concentrated saliency, helpful for late prediction time steps.

## 5. CONCLUSION

In this article we have reviewed existing methods for the dynamic head prediction problem in 360° videos. We proposed an architecture, named TRACK, that (i) has a dedicated recurrent network to process individually the time series of positions, before merging the obtained embedding with visual content features, and (ii) has another recurrent network processing the saliency map estimated from the content, to be resilient to, while benefiting from, noisy saliency information. We show that TRACK achieves state-of-the-art performance on all considered datasets and prediction horizons: from 0 to 5 seconds, and we conduct an ablation study to justify its assets.

## 6. REFERENCES


[1] Jounsup Park, Philip A Chou, and Jenq-Neng Hwang, "Rate-utility optimized streaming of volumetric media for augmented reality," *arXiv preprint arXiv:1804.09864*, 2018.

[2] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[3] Cagri Ozcinar, Ana De Abreu, and Aljosa Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178.

[4] Mengbai Xiao, Chao Zhou, Viswanathan Swaminathan, Yao Liu, and Songqing Chen, "BAS-360: Exploring spatial and temporal adaptability in 360-degree videos over HTTP/2," INFOCOM, 2018.

[5] H. Hristova, X. Corbillon, G. Simon, V. Swaminathan, and A. Devlic, "Heterogeneous spatial quality for omnidirectional video," in *IEEE MMSP*, Aug 2018, pp. 1–6.

[6] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[7] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Trans. on PAMI*, 2018.

[8] Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao, "Gaze prediction in dynamic 360° immersive videos," in *IEEE CVPR*, 2018, pp. 5333–5342.

[9] Anh Nguyen, Zhisheng Yan, and Klara Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," in *ACM Int. Conf. on Multimedia*, 2018, pp. 1190–1198.

[10] Yunqiao Li, Yiling Xu, Shaowei Xie, Liangji Ma, and Jun Sun, "Two-layer fov prediction model for viewport dependent streaming of 360-degree videos," in *EAI Int. Conf. on Communications and Networking (ChinaCom)*, Chengdu, China, Oct. 2018.

[11] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu, "Fixation prediction for 360 video streaming in head-mounted virtual reality," in *ACM NOSSDAV*, 2017, pp. 67–72.

[12] M. F. Romero-Rondon, L. Sassatelli, R. Aparicio, and F. Precioso, "head-motion-prediction," https://gitlab.com/miguelfromeror/head-motion-prediction/tree/master, 2020.

[13] S. Petrangeli, G. Simon, and V. Swaminathan, "Trajectory-based viewport prediction for 360-degree virtual reality videos," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, Dec 2018, pp. 157–160.

[14] S. Rossi, F. De Simone, P. Frossard, and L. Toni, "Spherical clustering of users navigating 360-degree content," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4020–4024.

[15] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz, "Predicting head trajectories in 360° virtual reality videos," in *IEEE Int. Conf. on 3D Immersion (IC3D)*, 2017.

[16] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2018.

[17] Erwan J David, Jesús Gutiérrez, Antoine Coutrot, Matthieu Perreira Da Silva, and Patrick Le Callet, "A dataset of head and eye movements for 360-degree videos," in *ACM MMSys*, 2018, pp. 432–437.

[18] Anh Nguyen, Zhisheng Yan, and Klara Nahrstedt, "Panosalnet - source code for Your Attention is Unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," https://github.com/phananh1010/PanoSalNet, 2019.

[19] Mathias Almquist, Viktor Almquist, Vengatanathan Krishnamoorthi, Niklas Carlsson, and Derek Eager, "The prefetch aggressiveness tradeof in 360 video streaming," in *ACM MMSys*, 2018.