360° Video Viewing Dataset in Head-Mounted Virtual Reality

Wen-Chih Lo
Department of Computer Science
National Tsing Hua University

Ching-Ling Fan
Department of Computer Science
National Tsing Hua University

Jean Lee
Department of Computer Science
National Tsing Hua University

Chun-Ying Huang
Department of Computer Science
National Chiao Tung University

Kuan-Ta Chen
Institute of Information Science
Academia Sinica

Cheng-Hsin Hsu
Department of Computer Science
National Tsing Hua University

ABSTRACT

360° videos and Head-Mounted Displays (HMDs) are getting increasingly popular. However, streaming 360° videos to HMDs is challenging. This is because only video content in viewers' Field-of-Views (FoVs) is rendered, and thus sending complete 360° videos wastes resources, including network bandwidth, storage space, and processing power. Optimizing the 360° video streaming to HMDs is, however, highly *data* and *viewer* dependent, and thus dictates real datasets. However, to our best knowledge, such datasets are not available in the literature. In this paper, we present our datasets of both content data (such as image saliency maps and motion maps derived from 360° videos) and sensor data (such as viewer head positions and orientations derived from HMD sensors). We put extra efforts to align the content and sensor data using the timestamps in the raw log files. The resulting datasets can be used by researchers, engineers, and hobbyists to either optimize existing 360° video streaming applications (like rate-distortion optimization) and novel applications (like crowd-driven camera movements). We believe that our dataset will stimulate more research activities along this exciting new research direction.

CCS CONCEPTS

•Information systems → Multimedia streaming; •Mathematics of computing → *Approximation;*

KEYWORDS

360° dataset, 360° video, virtual reality, HMD, head tracking dataset

ACM Reference format:
Wen-Chih Lo, Ching-Ling Fan, Jean Lee, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2017. 360° Video Viewing Dataset in Head-Mounted Virtual Reality. In *Proceedings of MMSys'17, Taipei, Taiwan, June 20-23, 2017,* 6 pages.
DOI: http://dx.doi.org/10.1145/3083187.3083219

1 INTRODUCTION

Augmented Reality (AR) and Virtual Reality (VR) are getting popular, e.g., a market research says that the AR/VR market will drive 108 billion USD annual revenue by 2021 [1]. Different 360° videos can be viewed with Head-Mounted Displays (HMDs), including Computer-Generated (CG) and Natural Image (NI) videos. Using conventional displays to watch 360° videos is often less intuitive, while recently released HMDs, such as Oculus Rift [3], HTC Vive [4], Samsung Gear VR [6], offer wider Field-of-Views (FoVs) and thus more immersive experience.

While service providers like YouTube [7] and Facebook [2], have put some 360° videos online, streaming these videos to HMDs is extremely challenging. One of the challenges is that 360° videos are in very high resolution, such as 4K, 8K, and higher. When watching a 360° video, a viewer wearing an HMD rotates his/her head to change the viewing *orientation*, which can be described by the angles along the x, y, and z axes. These three angles are called *yaw*, *pitch*, and *roll*. Based on the orientation, the HMD displays the current FoV, which is a fixed-size region, say 100°x100° circle. Since a viewer never sees a whole 360° video, streaming the 360° video in its full resolution wastes resources, including bandwidth, storage, and computation.

Therefore, each 360° video is often split into grids of sub-images, called tiles [13, 14]. With tiles, an optimized 360° video streaming system to HMDs would *strive* to stream only those tiles that fall in the viewer's FoV. By doing so, the system satisfies the viewer's needs and consumes less resources than streaming the whole video at its full resolution. However, getting to know *each* viewer's FoVs at *any* moment of *every* 360° video is not an easy task. The complex interplay among too many factors increases the difficulty. More specifically, both content (360° videos) and sensors (HMDs worn by viewers) affect the viewers' FoVs in the future moments. Hence, to better address the challenge, a large set of the content and sensor data from viewers watching 360° videos with HMDs is crucial.

Unfortunately, there are no public content and sensor datasets, e.g., datasets used in [11, 24] are from the industry and proprietary. To overcome such limitation, and promote reproducible research, we build up our own 360° video testbed for collecting traces from real viewers watching 360° videos using HMDs. We then use the testbed to collect content and sensor dataset. The resulting dataset can be used to, for example, predict which parts of 360° videos attract viewers to watch the most. The dataset, however, can also be leveraged in various novel applications in a much broader scope. For example, using our dataset, content provider could get to compute the most *common* FoVs among viewers, and derive the *crowd-driven camera movements*, which may be used to guide viewers through 360° videos via innovative user interfaces. Deeper investigations could even identify the essential *elements* for gaining viewers' attentions in 360° videos streamed to HMDs.

MMSys'17, Taipei, Taiwan

DOI: http://dx.doi.org/10.1145/3083187.3083219

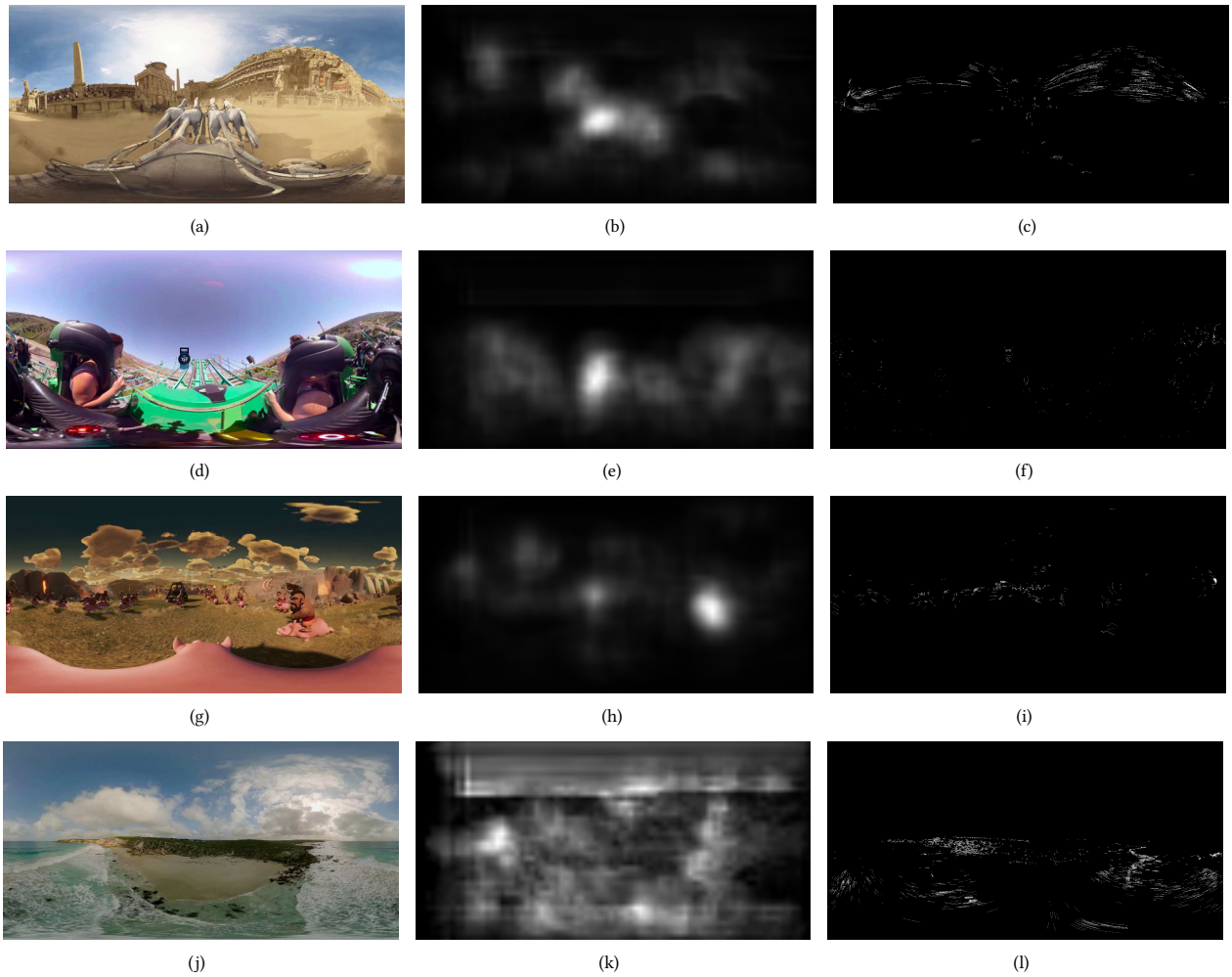

Figure 1: (a), (d), (g), (j): sample 360° video frames; (b), (e), (h), (k): image saliency maps; and (c), (f), (i), (l): motion maps. Samples from Chariot Race: (a), (b), (c); from Roller Coaster: (d), (e), (f); from Hog Rider: (g), (h), (i); and from Kangaroo Island: (j), (k), (l).

Our dataset contains content and sensor data from ten videos available on YouTube[1] and 50 viewers between ages of 20 and 48. More precisely, we analyze the 10 videos to extract the crucial features: the *image saliency* map [9] that identifies the objects attracting the viewers' attention the most; and the *motion* map [16] that high-lights the moving objects. We also log the sensor readings from the HMDs, and process them (along with the 360° videos) to derive *viewer orientation* and *viewed tile numbers*. Our dataset is unique because we collect both content and sensor data.

2 RELATED WORK

To our best knowledge, content and sensor traces of 360° video streaming to HMDs are not available to the publics. In this section, we survey some partially related content and sensor datasets.

[1]We use these videos for research purpose only.

Content traces. To know the impacts of video content on viewers' attentions, there are several content datasets that can be used for user studies. Riegler et al. [20] promote context of experience, which captures how well people percept video content, and consider the relation between video content and viewer intent. Ahmadi et al. [8] focus on gamers of side-scrolling gamers, and study the interplay between visual object regions and user attentions. While these papers [8, 20] include content traces, they are not for 360° videos, nor for viewers with HMDs.

Sensor traces. Different viewing conditions affect viewer attentions of video content. Vigier et al. [23] focus on eye tracking when viewers watching HD and UHD videos on larger screens, with wider visual angles. Ahmadi et al. [8] present an eye tracking dataset based on game-specific visual attention models. They consider the players' gaze points and mouse/keyboard commands. However, these papers [8, 23] do not consider 360° video streaming

to HMDs. Yu et al. [11] and Corbillon et al. [24] use a 360° dataset from a company, which is not publicly available.

3 CONTENT TRACE COLLECTION

In this section, we describe our content traces. Fig. 1 gives sample video frames, image saliency maps, and motion maps from four 360° videos.

3.1 Video Traces

We collect ten 360° videos with diverse characteristics from YouTube [7]. Table 1 summarizes the 360° videos. All the videos are in 4K resolution at 30 frame-per-second (fps). The videos come in different lengths, so we extract 1-min segment from each of them for experiments. The 360° videos are divided into 3 categories: (i) CG, fast-paced (ii) NI, fast-paced, and (iii) NI, slow-paced. The 360° videos are encoded in H.264 and stored in MP4 container files. We do not re-encode the videos, but report their size in Table 1. We note that H.264 codecs only support rectangular video frames. Therefore, YouTube adopts the *equirectangular* projection that maps the longitude and latitude of the sphere videos to the horizontal and vertical coordinates of the rectangular video. Although equirectangular projection leads to serious shape distortion (especially when close to the two poles), it's still widely used due to its simplicity.

3.2 Image Saliency Maps

Image saliency maps (see Figs. 1(b), 1(e), 1(h), 1(k)) indicate the attraction levels of the video frames. We process the ten 360° videos and generate the image saliency map (as videos) using Convolutional Neural Network (CNN) [21], which is widely used on images and videos. We use a deep neural network [12] based on the pre-trained VGG-16 network [21], which is combined with the weighted features from different levels of the CNN. The image saliency map is a gray-scale image (from 0 to 255), varying from black indicating the least interesting pixels to white indicating the most interesting pixels. For each 360° video, we first split each video into 1,800 images. We then apply the Keras-based [10] script developed in Cornia et al. [12] to generate the image saliency map. Last, we concatenate the 1,800 image saliency maps into a 1-min video, and encode it using H.264 in MP4 format.

3.3 Motion Maps

We analyze the optical flow [17] of the consecutive video frames from each 360° video. Optical flows indicate the relative motions between the objects in 360° videos and the viewers. They can be attributed to either local motions (individual objects move) or global motions (the camera moves), which may catch viewers' attentions. We generate the motion maps (see Figs. 1(c), 1(f), 1(i), 1(l)) using OpenCV [22]. The motion maps are black-and-white images (0 or 1), where a white pixel indicates the pixel is on one of the optical flows. In particular, we split each 360° video into 1,800 images, and generate 1,800 black-and-white images using OpenCV-based script. Last, we concatenate these motion maps into a 1-min video, and encode it using H.264 in MP4 format.

4 SENSOR TRACE COLLECTION

In this section, we describe how we collect sensor data from HMDs while viewers are watching 360° videos.

4.1 Testbed

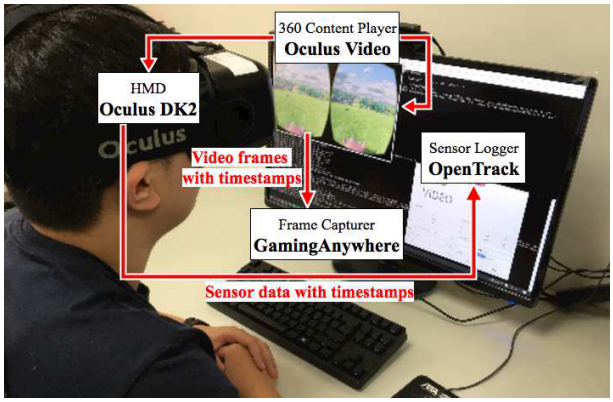

Figure 2: A photo of our 360° video streaming testbed. During the experiments, most subjects prefer to stand when watching videos.

We first give an overview on the architecture of our testbed. The 360° video testbed contains four major components: HMD, 360° video player, frame capturer, and sensor logger. Fig. 2 shows a photo of our testbed with the four components highlighted. We detail these components in the following.

- **HMD.** We use Oculus Rift DK2 [3] to be our HMD. We follow the official installation guide from Oculus to set up the hardware and install the Software Development Kit (SDK). This is done on a PC workstation with an Intel E3 CPU, 16 GB RAM, and an NVIDIA GTX 970 GPU.
- **360° video player.** Oculus Video [5] is an official app from Oculus. We configure it to render 360° videos in both HMD and a mirrored screen. We note that Oculus Video supports equirectangular 360° videos (projected to sphere surface) if the filenames have a suffix of `_360`, e.g., `coaster_360.mp4`. Otherwise, the videos are played as conventional videos instead of 360° ones.
- **Frame capturer.** We use GamingAnywhere [15] as our frame capturer, in order to record the videos rendered to the viewer. The frame capturer stamps each recorded frame with the timestamp, which will be used to align data from various sources. We configure GamingAnywhere to save YUV files at 30 fps.
- **Sensor logger.** We use OpenTrack [19], an open-source head tracking tool, to record the viewer orientations, including yaw, pitch and roll in the range of [-180, 180] from the HMD sensors. Moreover, we also record and timestamp the viewer positions, including the x, y, and z coordinates. We, however, notice that the 360° video player *ignores* the viewer positions; hence, most viewers

Table 1: Specifications of Ten 360° Videos from YouTube

Category	Videos	Used Segment	Size (MB)	Link
NI, fast-paced	Mega Coaster	1:30 - 2:30	160	https://youtu.be/-xNN-bJQ4vI
	Roller Coaster	0:20 - 1:20	153	https://youtu.be/8lsB-P8nGSM
	Driving with	0:48 - 1:48	117	https://youtu.be/LKWXHKFCMO8
NI, slow-paced	Shark Shipwreck	0:30 - 1:30	114	https://youtu.be/aQd41nbQM-U
	Perils Panel	0:10 - 1:10	60	https://youtu.be/kiP5vWqPryY
	Kangaroo Island	0:01 - 1:01	126	https://youtu.be/MXlHCTXtcNs
	SFR Sport	0:16 - 1:16	51	https://youtu.be/lo5N90TlzwU
	Hog Rider	0:00 - 1:00	138	https://youtu.be/yVLfEHXQk08
CG, fast-paced	Pac-Man	0:10 - 1:10	50	https://youtu.be/p9h3ZqJa1iA
	Chariot Race	0:02 - 1:02	149	https://youtu.be/jMyDqZe0z7M

in our dataset stay at roughly the same position. Several enhancements have been added by us into OpenTrack project. For example, we enhance the code to save timestamped logs by increasing the granularity of timers to meet our needs.

When a viewer watches a 360° video as shown in Fig. 2, the viewer can watch at any orientation by rotating his/her head. The rendered videos on the mirrored screen are captured by frame capturer and stored to disk. The sensor logger records and stores the viewing orientations. Note that the timestamps added by frame capturer and sensor logger are from the same PC workstation. Hence, they can be readily used for alignments.

4.2 Procedures and Subjects

We collect sensor traces from 50 subjects. The subjects are asked to watch all the ten 360° videos (see Table 1), where each video lasts for 1 minute. All subjects are told to stand and then are given enough space to turn around when wearing the HMD. After watching the 360° videos, subjects are asked to fill out a questionnaire. Since most participants are not familiar with HMD, collecting the dataset is time consuming. On average, it takes us 30 minutes to introduce our system to a subject, guide him/her to watch ten 1-min videos, and collect the questionnaires.

Most subjects are in their early twenties, and 52% of them are male. Around 94% of the subjects seldom use HMDs, and about 56% of them use HMDs for the first time. The responses from all subjects indicate the most interesting topic of 360° videos are gaming, simulations, and landscapes. Fig. 3 gives an overview on the popularity of three video categories from our subjects. Because of the motion sickness and discomfort of watching video in HMDs, about 51% of viewers prefer not to wear HMD for more than 30 minutes at a time.

5 DATA FORMAT AND BASIC STATISTICS

In this section, we present the data format and some statistics of both content and sensor datasets. The content dataset is compressed using H.264, while the sensor dataset is stored as Comma-Separated Values (CSV) files in ASCII.

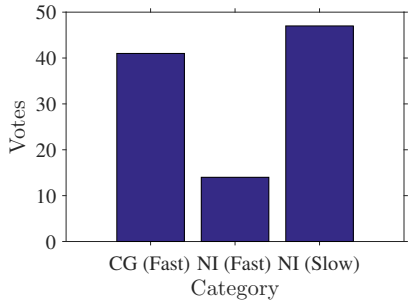

Figure 3: The popularity of three video categories.

5.1 Content Dataset

The content dataset contains twenty H.264 videos in MP4 container, where each 360° video is analyzed for two content files: (i) the image saliency map and (ii) the motion map. Table 2 gives the filenames and sizes of these 20 video files.

Table 2: Content Data Files of Ten 360° Videos

Video	Filename	Size (MB)
Mega Coaster	coaster2_saliency.mp4	45.43
	coaster2_motion.mp4	42.90
Roller Coaster	coaster_saliency.mp4	52.19
	coaster_motion.mp4	23.83
Driving with	driving_saliency.mp4	43.51
	driving_motion.mp4	71.57
Shark Shipwreck	diving_saliency.mp4	21.99
	diving_motion.mp4	7.99
Perils Panel	panel_saliency.mp4	27.07
	panel_motion.mp4	1.98
Kangaroo Island	landscape_saliency.mp4	60.39
	landscape_motion.mp4	57.62
SFR Sport	sport_saliency.mp4	28.58
	sport_motion.mp4	19.65
Hog Rider	game_saliency.mp4	45.94
	game_motion.mp4	27.70
Pac-Man	pacman_saliency.mp4	33.95
	pacman_motion.mp4	5.43
Chariot Race	ride_saliency.mp4	49.28
	ride_motion.mp4	45.17

5.2 Sensor Dataset

```
1: timestamp, raw x, raw y, raw z, raw yaw, raw pitch, raw roll
2: 1487571103.944, 26.289, 28.063, -15.581, -5.246, -4.298, -1.315
3: 1487571103.953, 26.291, 28.063, -15.567, -5.297, -4.287, -1.333
4: 1487571103.957, 26.292, 28.063, -15.559, -5.323, -4.284, -1.341
5: 1487571103.961, 26.293, 28.063, -15.552, -5.350, -4.277, -1.348
6: 1487571103.965, 26.294, 28.063, -15.545, -5.378, -4.270, -1.354
7: ...
```

Figure 4: Sample lines of a raw sensor data log file.

```
1: no. frames, raw x, raw y, raw z, raw yaw, raw pitch, raw roll, cal. yaw, cal. pitch, cal. roll
2: 00001, 16.458 ,30.032, -19.276, -9.661, 5.853, -3.068, -4.65473888889, 4.06641388889, -3.068
3: 00002, 16.458 ,30.032, -19.276, -9.661, 5.853, -3.068, -4.65473888889, 4.06641388889, -3.068
4: 00003, 16.449 ,30.02, -19.362, -9.763, 5.746, -3.184, -4.75673888889, 3.95941388889, -3.184
5: 00004, 16.449 ,30.02, -19.362, -9.763, 5.746, -3.184, -4.75673888889, 3.95941388889, -3.184
6: 00005, 16.433 ,30.007, -19.473, -9.676, 5.659, -3.308, -4.66973888889, 3.87241388889, -3.308
7: ...
```

Figure 5: Sample lines of a view orientation log file.

```
1: no. frames, tile numbers
2: 00001, 9, 28, 29, 30, 31, 47, 48, 49, 50, 51, 67, 68, 69, 70, 71, 72, 87, 88, 89, 90, 91, 92
3: 00002, 9, 28, 29, 30, 31, 47, 48, 49, 50, 51, 67, 68, 69, 70, 71, 72, 87, 88, 89, 90, 91, 92
4: 00003, 9, 28, 29, 30, 31, 47, 48, 49, 50, 51, 67, 68, 69, 70, 71, 72, 87, 88, 89, 90, 91, 92
5: 00004, 9, 28, 29, 30, 31, 47, 48, 49, 50, 51, 67, 68, 69, 70, 71, 72, 87, 88, 89, 90, 91, 92
6: 00005, 9, 28, 29, 30, 31, 47, 48, 49, 50, 51, 67, 68, 69, 70, 71, 72, 87, 88, 89, 90, 91, 92
7: ...
```

Figure 6: Sample lines of a viewed tile log file.

The sensor dataset contains 500 *raw sensor log files*, since we have 50 subjects and ten 360° videos. Fig. 4 gives a sample raw sensor log file, which contains 7 fields: (i) timestamp, (ii) raw x, (iii) raw y, (iv) raw z, (v) raw yaw, (vi) raw pitch, and (vii) raw roll. In our pilot experiments, we find that different HMD viewers tend to introduce different amount of *bias*. We then introduce a calibration procedure before each viewer starts watching 360° videos. In particular, we insert a 35-sec calibration video at the beginning of each 360° video. The calibration video sequentially displays an object (cartoon sheep) at (1920, 960), (2880, 480), (2880, 1440), (3840, 960), (960, 480), (960, 1440), and (1920, 96) of coordinates. We show the object at each position for 5 seconds, and we ask the subject to rotate his/her head in order to place the object at the center of their FoV. We then average the bias between the captured sensor data and the ground truth from calibration video. Using the bias, we compensate the raw sensor readings (yaw, pitch, and roll) for *calibrated* (cal.) sensor readings (yaw, pitch, and roll).

We note that the sensor data are collected (by OpenTrack [19]) at 250 Hz, and the captured video frames are saved (by GamingAnywhere [15]) at 30 Hz. Users of the raw sensor log files need to align the raw sensor log files with the captured video frames. To simplify the usage of our dataset, we generate *view orientation log files* at 30 Hz by aligning the timestamps in the raw sensor log files and captured video frames. Moreover, we include the calibrated sensor readings derived above in view orientation log files. Fig. 5 gives a simple view orientation log file, which contains 10 fields: (i) no. frames, (ii) raw x, (iii) raw y, (iv) raw z, (v) raw yaw, (vi) raw pitch, (vii) raw roll, (viii) cal. yaw, (ix) cal. pitch, and (x) cal. roll.

While view orientation log files give the *center* of viewer's FoVs, determining which tiles are needed to render the FoVs require extra calculations. We assume the FoVs are modeled by 100°x100° circles. Therefore, we process the view orientation log files, and generate *viewed tile log files* to further simplify the usage of our dataset. For all 360° videos, we divide each frame, which is mapped in equirectangular model, into 192x192 tiles, so there are 200 tiles in total. Then we number the tiles from upper-left to lower-right. Fig. 6 gives a sample viewed tile log file, which contains 2 fields: (i) no. frames and (ii) tile numbers. Each tile number determines a unique tile of the whole 360° video, and an FoV overlaps with multiple tiles as shown in this figure.

Last, Table 3 gives filenames of sensor data. For each video and each user, there are three sensor data files for: (i) raw sensor, (ii) view orientation, and (iii) viewed tiles. Filenames from user 0 are given as examples, while files from other users are also available in our dataset. This table also reports the total size of these log files stored in ASCII format.

Table 3: Sensor Data Files of 50 Subjects

Video	Sample Filename (User 0)	Total Size (MB)
Mega Coaster	coaster2_user00_raw.csv	224.93
	coaster2_user00_orientation.csv	16.22
	coaster2_user00_tile.csv	20.16
Roller Coaster	coaster_user00_raw.csv	228.66
	coaster_user00_orientation.csv	16.14
	coaster_user00_tile.csv	19.75
Driving with	drive_user00_raw.csv	226.99
	drive_user00_orientation.csv	15.89
	drive_user00_tile.csv	19.78
Shark Shipwreck	diving_user00_raw.csv	224.69
	diving_user00_orientation.csv	16.19
	diving_user00_tile.csv	19.03
Perils Panel	panel_user00_raw.csv	229.45
	panel_user00_orientation.csv	16.13
	panel_user00_tile.csv	20.42
Kangaroo Island	landscape_user00_raw.csv	228.71
	landscape_user00_orientation.csv	16.29
	landscape_user00_tile.csv	20.41
SFR Sport	sport_user00_raw.csv	225.77
	sport_user00_orientation.csv	15.86
	sport_user00_tile.csv	21.97
Hog Rider	game_user00_raw.csv	230.10
	game_user00_orientation.csv	15.79
	game_user00_tile.csv	19.78
Pac-Man	pacman_user00_raw.csv	218.60
	pacman_user00_orientation.csv	16.02
	pacman_user00_tile.csv	20.32
Chariot Race	ride_user00_raw.csv	230.65
	ride_user00_orientation.csv	15.79
	ride_user00_tile.csv	19.55

6 SAMPLE APPLICATIONS

Our collected dataset can be used in various 360° video applications with viewers using HMDs. More specifically, researchers, engineers, and hobbyists can: (i) analyze our dataset for some insights before designing their systems and algorithms, (ii) employ our dataset to train and fine-tune their systems and algorithms,

and (iii) adopt our dataset in their trace-driven simulations and emulations. In the rest of this section, we briefly present three sample applications of our dataset.

Viewed tile predictions for 360° video streaming to HMDs. The de-facto DASH (Dynamic Adaptive Streaming over HTTP) approach divides each video into segments, and every segment lasts for a few (say 10) seconds. The DASH client requests for the segments over HTTP/TCP connections. Compared to the UDP-based RTP (Real-time Transport Protocol) approach, DASH is less sensitive to network dynamics, but more vulnerable to long response time. For example, when a viewer with HMD rotates his/her head to new tiles that have not been requested, it may take the streaming system *several seconds* to deliver these new tiles. The state-of-the-art viewed tile prediction work [18] employs simple extrapolations, which may be less accurate as only sensor (no content) data are leveraged. Our comprehensive dataset can be used for developing and evaluating new algorithms for viewed tile predictions, so as to mitigate the limitations of DASH streaming in 360° video streaming to HMDs.

Rate-distortion optimization. Our dataset also contains video content with diverse characteristics, and can be used for Rate-Distortion (R-D) optimization for 360° video streaming to HMDs. Compared to traditional video streaming, viewers of 360° videos only see portions (FoVs) of the whole videos. Hence, the room for R-D optimization is even larger and is worth to investigate.

Crowd-driven camera movements. Some novel applications may be proposed based on observations on our dataset, which contain sensor data from many viewers, or crowds. For example, common camera movements could be identified among the *view orientation log files* among viewers of the same 360° videos. The resulting camera movements can be then used to guide viewers through the 360° videos, providing a new interaction model, which may be appealing to viewers who do *not* want to make too many decisions on where to look. Our dataset can be used to understand how *homogeneous* the viewer orientations are, so as to quantify the potential of this (and other) novel application.

7 CONCLUSION

In this paper, we presented our dataset collected from ten YouTube 360° videos and 50 subjects. Our dataset is unique, because both content data, such as image saliency maps and motion maps, and sensor data, such as positions and orientations, are provided. Extra efforts are put into aligning the content and sensor data based on the timestamps in raw log files. To our best knowledge, there exists no similar datasets in the literature. The resulting dataset can be leveraged by researchers, engineers, and hobbyists in different development phases: from design, to fine-tuning, to evaluations. Many 360° video streaming applications, both traditions ones (like R-D optimization) and novel ones (like crowd-driven camera movements) can benefit from our comprehensive dataset. Our current work can be extended in several ways. For example, the eyes movement are good hints of future head movement. Adding eye tracking data to our dataset will further broaden the applications of our dataset. Last, we thank the subjects who volunteered in our user study.

ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science and Technology of Taiwan under the grants: 104-2221-E-009-200-MY3, 105-2628-E-001-004-MY2, and 105-2221-E-007-088.

REFERENCES

[1] 2017. After mixed year, mobile AR to drive $108 billion VR/AR market by 2021. (2017). https://goo.gl/P9N0z0.
[2] 2017. Facebook. (2017). https://www.facebook.com/.
[3] 2017. Facebook Oculus Rift. (2017). https://www.oculus.com.
[4] 2017. HTC Vive. (2017). https://www.htcvive.com.
[5] 2017. Oculus Video. (2017). ://www.oculus.com/experiences/rift/926562347437041/.
[6] 2017. Samsung Gear VR. (2017). http://www.samsung.com/global/galaxy/gear-vr.
[7] 2017. YouTube. (2017). https://www.youtube.com/.
[8] H. Ahmadi, S. Tootaghaj, S. Mowlaei, M. Hashemi, and S. Shirmohammadi. 2016. GSET somi: a game-specific eye tracking dataset for somi. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*. Klagenfurt, Austria, 42:1–42:6.
[9] A. Borji, M. Cheng, H. Jiang, and J. Li. 2014. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878* (2014).
[10] François Chollet. 2015. Keras. https://github.com/fchollet/keras. (2015).
[11] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski. 2017. Viewport-Adaptive Navigable 360-Degree Video Delivery. In *Proc. of IEEE International conference on communications (ICC'17)*. Paris, France, Accepted to Appear.
[12] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. 2016. A Deep Multi-Level Network for Saliency Prediction. In *Proc. of International Conference on Pattern Recognition (ICPR'16)*. Cancun, Mexico, 3488–3493.
[13] L. D'Acunto, J. Berg, E. Thomas, and O. Niamut. 2016. Using MPEG DASH SRD for zoomable and navigable video. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*. Klagenfurt, Austria, 34:1–34:4.
[14] J. Feuvre and C. Concolato. 2016. Tiled-based adaptive streaming using MPEG-DASH. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*. Klagenfurt, Austria, 41:1–41:3.
[15] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. 2013. GamingAnywhere: An Open Cloud Gaming System. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'13)*. Oslo, Norway, 36–47.
[16] Y. Kavak, E. Erdem, and A. Erdem. 2017. A comparative study for feature integration strategies in dynamic saliency estimation. *Signal Processing: Image Communication* 51 (November 2017), 13–25.
[17] B. Lucas and T. Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'7)*. Vancouver, BC, Canada, 674–679.
[18] A. Mavlankar and B. Girod. 2010. Video streaming with interactive pan/tilt/zoom. In *Signals and Communication Technology*. 431–455.
[19] OpenTrack:head tracking software 2017. OpenTrack: head tracking software. (2017). https://github.com/opentrack/opentrack.
[20] M. Riegler, M. Larson, C. Spampinato, P. Halvorsen, M. Lux, J. Markussen, K. Pogorelov, C. Griwodz, and H. Stensland. 2016. Right Inflight?: A Dataset for Exploring the Automatic Prediction of Movies Suitable for a Watching Situation. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*. Klagenfurt, Austria, 45:1–45:6.
[21] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[22] The OpenCV Library 2000. The OpenCV Library. (2000). http://opencv.org.
[23] T. Vigier, J. Rousseau, M. Silva, and P. Callet. 2016. A new HD and UHD video eye tracking Dataset. In *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*. Klagenfurt, Austria, 48:1–48:6.
[24] M. Yu, H. Lakshman, and B. Girod. 2015. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR'15)*. Fukuoka, Japan, 31–36.