Supplementary Material for CEAP-360VR: A Continuous Physiological and Behavioral Emotion Annotation Dataset for 360° VR Videos

I. Dataset Description

Our dataset includes the raw and processed data from all 32 participants and eight seleced videos, along with the processing and validating scripts. All data were saved in JavaScript Object Notation (JSON) [1], a well-known file format that has native support by most programming languages. This makes the data accessible and easy to process. We will provide a website that can enable researchers to easily locate this dataset, where they can request our public dataset.

In our dataset, we gave a unique identifier to 32 participants ($P1-P32$) and eight videos ($V1-V8$). The following description of the data uses the letter *PXX* to denote the IDs of the participants (*XX* are natural numbers in the set {1,2,...32}) and *VXX* to denote the IDs of the videos (*XX* are natural numbers in the set {1,2,...8}).

A. Stimuli and Questionnaire

The *1_Stimuli* folder contains a *VideoInfo.json* file including all eight videos with detailed information (video name, resolution, duration, framerate, link[1]), as well as the mean V-A ratings from Li et al.'s dataset [2] and our annotation study for the clipped videos.

The *2_QuestionnaireData* directory contains one JSON file *PXX_QuestionnaireData.json* per participant for all videos. For the SSQ, IPQ and NASA-TLX questionnaire data records, we used integer numbers in the following ranges $[1,4],[-3,3],[1,20]$, respectively. Each file consists of participants' responses to all questionnaires, V-A Within-VR SAM ratings, the video playback order and the peripheral feedback order in which *1* represents the first block and *2* means the second block. We also extracted the start and end timestamps for each video watched by each participant, which were recorded in four formats: the local UTC/GMT time, the Unix time with second accuracy, the Unix time with millisecond accuracy, and the HMD device time.

B. Participant Data

In our dataset, participant data are recorded in three directories: *3_AnnotationData*, *4_BehaviorData*, and *5_PhysioData*.

1) Data Pre-processing: We first summarize the pre-processing procedures for the logged data.

Step 1. Video Playback Time: Based on the video start and end timestamps in *PXX_QuestionnaireData.json* file, we filter out the logged data during each video playback period (by clipping from video start and end time). Then the video playback time relative to the video start time 0 is calculated with millisecond accuracy, which is added to each sample data as Timestamp in second.

Step 2. Transforming Raw Data: We transform the raw data based on the data attributes and the logged instruments and sensors. The collected annotation data are rescaled to $[1,9]$ to be consistent with the SAM rating range. The raw HM/EM data are converted into longitude and latitude of viewing directions for further analysis. For physiological data, we normalize the values of each signal after filtering out noise following previous work [3]. The specific conversions are described in the next sections in detail.

Step 3. Data Re-sampling: The data logged in our experiment are at different sampling frequencies because they were collected from different instruments and sensors. In data acquisition and recording, latency is a common problem [4], [5], which is also evident in our data. To align and synchronize different types of data among participants, we first calculate the new sampling timestamps in each frame for each video. If the sampling frequency of the raw data is less than the video frame rate, a linear interpolation is performed following prior work [4] to determine the values at new sampling timestamps by fitting a line using the corresponding discrete samples in the raw data. For the raw sampling frequency higher than the video frame rate, we select the maximum value less than the specified timestamp of the re-sampled data. For other researchers who may prefer different alignment methods, we also provide the original non-interpolated raw data in our dataset.

Step 4. Data Saving format: Step 1 is applied to get the raw data from the logged data. Step 2 is performed differently to acquire the transformed data and Step 3 is applied to get the frame data based on the transformed data. Therefore, the annotation, behavioral and physiological data are divided into the following sub-directories: *Raw*, *Transformed*, and *Frame*.

2) Continuous Annotation Data: We logged the continuous X-axis u and Y-axis v values of the joystick head position in range $[-1,1]$ during the course of videos at 10Hz. Then based on Step 1 in the pre-processing, the raw annotation data was saved into *PXX_Annotation_RawData.json* file per participant, containing three variables *{Timestamp, X_Value, Y_Value}*. Since the movement of the joystick head is in a circular disc, the simple stretch method [6] is used to map the data into a square region as follows ($sgn(u)$ is the function

[1]The eight videos used in our dataset are also available per request from the corresponding author (for non-commercial research purposes only).

that extracts the sign of u):

$$x = \begin{cases} 0 & u = 0 \\ sgn(u)\sqrt{u^2+v^2} & |u| \geq |v| \\ sgn(v)\frac{u}{v}\sqrt{u^2+v^2} & |u| < |v| \end{cases}$$
$$y = \begin{cases} v & u = 0 \\ sgn(u)\frac{v}{u}\sqrt{u^2+v^2} & |u| \geq |v| \\ sgn(v)\sqrt{u^2+v^2} & |u| < |v| \end{cases} \quad (1)$$

Then x and y values in the interval $[-1,1]$ are rescaled to $[1,9]$ to be consistent with the SAM-rating annotation.

$$\begin{aligned} Valence &= 4*x+5 \\ Arousal &= 4*y+5 \end{aligned} \quad (2)$$

The format of the transformed continuous annotation samples is the following: *{TimeStamp, Valence, Arousal}* saved into *PXX_Annotation_TransformedData.json*. The frame annotation data was provided in *PXX_Annotation_FrameData.json* based on Step 3.

3) Behavior Data: Data acquired from the HMD Tobii eye tracker were sampled at 120Hz. Each sample contains the following information: the camera Euler angles for the HMD/Head rotation, $rotation(x,y,z)$ $(x,y,z$ values are in range of $[0,360])$; the left, right and combined eye gaze direction as a normalized vector in world space, $direction(x,y,z)$ $(x,y,z$ values are in range of $[-1,1])$; the left pupil diameter (LPD) and right pupil diameter (RPD) are in millimeters. These data were extracted and saved in *PXX_Behavior_RawData.json* file per participant.

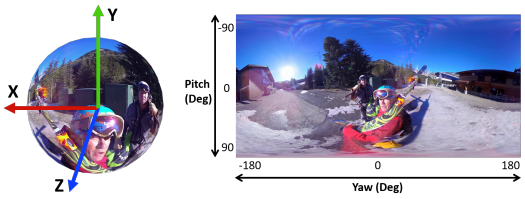

Fig. 1. One frame mapped to sphere and the coordinate system (left). One frame in Equirectangular format (right).

We show the coordinate system and the equirectangular video format in Fig. 1. Note that the left image on the surface of the sphere is mirrored, for the video is viewed from the inside of the sphere. When the viewer looks in the direction of Z, Z-axis points out to the center of the equirectangular video, Y-axis points up and X-axis points right. To obtain the points the participants are looking at, we extracted pitch and yaw values based on the raw head rotation and eye direction data. The reported yaw values are between $(-180,180)$ with 0 indicating the horizontal center of the original equirectangular video, and the pitch values are between $(-90,90)$ with 0 indicating the vertical center, as shown in Fig. 1(right). For each participant and each video, the head orientations (HM) including pitch H_{pitch} and yaw H_{yaw} were computed based on the raw head rotation data $rot(x,y,z)$ as follows:

$$H_{pitch} = \begin{cases} rot.x & rot.x \leq 180 \\ rot.x - 360 & rot.x > 180 \end{cases}$$
$$H_{yaw} = \begin{cases} rot.y & rot.y \leq 180 \\ rot.y - 360 & rot.y > 180 \end{cases} \quad (3)$$

Equation 4 was used to convert left, right and combined eye gaze direction (x,y,z) respectively to pitch E_{pitch} and yaw E_{yaw} as LEM, REM and EM:

$$E_{pitch} = sgn(y)\arccos\sqrt{\frac{x^2+z^2}{x^2+y^2+z^2}}\frac{180}{\pi}$$
$$E_{yaw} = \begin{cases} \arctan(\frac{x}{z})\frac{180}{\pi} & z > 0 \\ 180 + \arctan(\frac{x}{z})\frac{180}{\pi} & x > 0, z < 0 \\ -180 + \arctan(\frac{x}{z})\frac{180}{\pi} & x < 0, z < 0 \end{cases} \quad (4)$$

We did not do any conversion processing on LPD and RPD. The transformed behavior data were saved into *PXX_Behavior_TransformedData.json* containing: *{HM, EM, LEM, REM}* with variables *{TimeStamp, Pitch, Yaw}* and *{LPD, RPD}* with variables *{TimeStamp, PD}*. Frame behavior data was provided in *PXX_Behavior_FramedData.json* by Step 3.

4) Processed Behavior Data: It is necessary to detect users' fixation while watching 360° VR videos, especially for viewing patterns research. Based David et al.'s [7] work, it is not suitable to define concepts like head fixation and saccade, since the head-only movement without eye data cannot reflect the actual visual perception. In our dataset, we parse the addition of head and eye gaze direction data to fixation and saccade, along with head rotation data to head scanpaths, saved in *EM_Fixation* and *HM_ScanPath* folders, respectively.

To generate gaze fixation, the velocity and acceleration of the sample points by frame were calculated first. We determined the distance $\Delta\delta$ between two points by the Orthodromic Distance, λ_1, ϕ_1 and λ_2, ϕ_2 are the yaw and pitch values of two adjacent sampling points:

$$\Delta\sigma = 2\arcsin\sqrt{\sin^2(\frac{\Delta\phi}{2}) + \cos\phi_1\cos\phi_2 + \sin^2(\frac{\Delta\lambda}{2}))} \quad (5)$$

The velocity and acceleration of the first sample point are set as 0, and others are calculated by:

$$v_2 = \frac{\Delta\delta}{\Delta t}, \qquad a_2 = \frac{v_2 - v_1}{\Delta t} \quad (6)$$

A threshold-based method [8] defined the gaze saccade onset and offset using speed and acceleration of $75^o/s$ and $200^o/s^2$. We logged the sample data between the offset of one saccade and the onset of the next saccade. By setting the minimum duration to 150ms [9], the centroid of the sample data during this period was output as fixation center. The processed gaze fixation samples were saved in *PXX_Behavior_GazeFixationData.json* with 4 variables: *{StartFrame, EndFrame, Pitch, Yaw}*.

The method presented in [7] was used to achieve head scanpaths. We first segmented head rotation data using a sequential window of 200 msec. Then the centroid of these samples within said time windows was calculated. The processed head rotation samples were saved in *PXX_Behavior_HeadScanPathData.json* with 3 variables: *{PointID, Pitch, Yaw}*.

5) Physiological Data: With embedded sensors, the Empatica E4 collects ACC data from 3-axis accelerometer sensor (64Hz), BVP data from photoplethysmography (64Hz), EDA data from the electrodermal activity sensor in μS (4Hz), SKT data from temperature sensor in Celsius (4Hz). The HR data indicate average heart rate value (1Hz) and IBI data indicate the time interval between individual beats of the heart, and are computed by BVP data. All data are exported from the E4 Wristband in CSV format[2]. As previously mentioned in pre-processing Step 1, we kept the raw physiological data during the video playback period from each participant and saved it in the *PXX_Physio_RawData.json* file with the following variables: *{TimeStamp, ACC, EDA, SKT, BVP, HR, IBI}*. Note that IBI data from P2 and P12 were missing, possibly due to the internal algorithm of E4 deriving IBI from BVP automatically, which discard the obtained value when its reliability is below a certain threshold.

A common problem in physiological data logging is the signal noise that influences the stability of features and accuracy [10]. A third-order low-pass filter with a cut-off frequency of 2Hz was used to remove the artifacts in EDA, BVP and SKT [11]. Then we normalized the filtered values of EDA, BVP and SKT to range $[0,1]$, to overcome individual difference associated with physiological signals by using the min $signal_{min}$ and max $signal_{max}$ values of each participant [12], [13]:

$$Normalize_{signal}(i) = \frac{signal(i) - signal_{min}}{signal_{max} - signal_{min}} \tag{7}$$

The transformed and frame physiological data were saved in *PXX_Physio_TransformedData.json* file and *PXX_Physio_FrameData.json* file, respectively.

C. Scripts

The *6_Scripts* directory contains the Unity project (C#), as well as the code (Python scripts) for data processing and validation.

For the *CEAP-360VR_Project*, we published the software to playback 360° video, show the feedback visualization, and output Joy-Con controller data as well as record head rotation and eye direction data.
(1) *CEAP-360VR_Controller.cs* - implement 360° video playback, the annotation feedback display, within-VR SAM rating, and the data output.
(2) *SteamVR, JoyconLib, TobiiPro SDK* - used for HMD rendering, recognizing Joy-Con input and collecting HM/EM/PD data, respectively.

In the *CEAP-360VR_Process* folder, we show the scripts for pre-processing the acquired raw data for further analysis, which have been divided across the following scripts:
(1) *3_Get_Annotation_TransData.py, 4_Get_Behavior_TransData.py, 5_Get_Physio_TransData.py*, - implement Step 2 of pre-processing. The resulting files are saved in the *Transformed* directories.
(2) *3_Get_Annotation_FrameData.py, 4_Get_Behavior_FrameData.py, 5_Get_Physio_FrameData.py*, - implement Step 3 of pre-processing. The resulting files are saved in the *Frame* directories.
(3) *4_Get_Behavior_FixationData.py, 4_Get_Behavior_ScanPath.py*, - are used to process the behavior data to get gaze fixations and head scanpaths.
(4) *3_Analyze_Annotation_Data.py*, - is used to generate the annotation trajectories and calculate the mean value of continuous annotations for each participant watching each video.
(5) *4_Analyze_Behavior_Data.py*, - is used to analyze the consistency of participants' viewing behavior, plot the heatmaps and histograms, and calculate the mean values.
(6) *5_Analyze_Physio_Data.py*, - is used to standardize the physiological signals and extract the features of different signals.

In the *CEAP-360VR_Baseline* folder, we provide scripts to prepare data and features for running ML experiments, which include the following scripts:
(1) *1_Classifier_DataExtraction.py*, - is used to generate processed behavioral and physiological data with V-A labels for deep learning (DL) experiments.
(2) *2_Classifier_FeatureExtraction.py*, - is used to extract behavioral and physiological features with V-A labels for hand-crafted machine learning (ML) experiments.
(3) *3_Classifier_Main.py*, - is used to run the ML and DL scripts.
(4) *4_Classifier_Models.py*, - provides ML and DL models and implementation for metrics.
(5) *5_Classifier_Validation.py*, - provides subject-dependent and subject-independent model implementation.

[2]https://support.empatica.com/hc/en-us/articles/201608896-Data-export-and-formatting-from-E4-connect-

REFERENCES

[1] T. Bray *et al.*, "The javascript object notation (json) data interchange format," 2014.

[2] B. J. Li, J. N. Bailenson, A. Pines, W. J. Greenleaf, and L. M. Williams, "A public database of immersive vr videos with corresponding ratings of arousal, valence, and correlations between head movements and self report measures," *Frontiers in psychology*, vol. 8, p. 2116, 2017.

[3] B. Zhao, Z. Wang, Z. Yu, and B. Guo, "Emotionsense: emotion recognition based on wearable wristband," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2018, pp. 346–355.

[4] K. Sharma, C. Castellini, E. L. van den Broek, A. Albu-Schaeffer, and F. Schwenker, "A dataset of continuous affect annotations and physiological signals for emotion analysis," *Scientific data*, vol. 6, no. 1, pp. 1–13, 2019.

[5] S. Friston and A. Steed, "Measuring latency in virtual environments," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 4, pp. 616–625, 2014.

[6] C. Fong, "Analytical methods for squaring the disc," *arXiv preprint arXiv:1509.06344*, 2015.

[7] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet, "A dataset of head and eye movements for 360 videos," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 432–437.

[8] Y. Fang, R. Nakashima, K. Matsumiya, I. Kuriki, and S. Shioiri, "Eye-head coordination for visual cognitive processing," *PloS one*, vol. 10, no. 3, 2015.

[9] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, 2000, pp. 71–78.

[10] J. Wagner, J. Kim, and E. André, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," in *2005 IEEE international conference on multimedia and expo*. IEEE, 2005, pp. 940–943.

[11] M. Nabian, Y. Yin, J. Wormwood, K. S. Quigley, L. F. Barrett, and S. Ostadabbas, "An open-source feature extraction tool for the analysis of peripheral physiological data," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–11, 2018.

[12] J. Fleureau, P. Guillotel, and I. Orlac, "Affective benchmarking of movies based on the physiological responses of a real audience," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE, 2013, pp. 73–78.

[13] T. Zhang, A. El Ali, C. Wang, A. Hanjalic, and P. Cesar, "Rcea: Real-time, continuous emotion annotation for collecting precise mobile video ground truth labels," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–15.

or up by placing