This document introduces the process of making the timestamp sheet and using Matlab and Python to segment data based on that timestamp sheet.

# Timestamp sheet

Timestamp.xlsx is the raw timestamp data annotated by Yuchen Li. Based on it, I made Timestamp\_zi.xlsx contains the timestamp for each step. Some mistakes and typo are found in the raw data, which is fixed and marked red in this file.

There are four time need to be figured out, they are:

Step 3 start time: the time the participants enter the tea room to start reading timestamp

Step 4 start time: the time participants encounter with actor.

Step 5 start time: the time the conversation between actor and participant is over.

Step 6 start time: the time first interview question is asked by interviewer.

Following is the approaches to identify the timestamp of each step.

S3 start: listen the audios (subject microphone record) manually, the moment subject pick up the instruction is considered as S3 start. The microphone start time is known, so that the S3 start real time can be calculated

S4 start: For subject 1- 31: In Mic data, the start of communication time is known, plus the known microphone start time, it is the S4 start real time; For subject 32-49, they are annotated by Yuchen Li

S5 start: For subject 1-31: known mic end of communication time, plus the start time in real, it is S5 start time in real; For subject 32-49, they are annotated by Yuchen Li

S5 end: S5 start time + 90s.

S6 start: For subject 1-31: Mic-Q1 time is known, plus the known microphone start time, it is the s6 start time in real; For subject 32-49, annotated by Yuchen Li

NB: For subject 32-49, the step 3 start time cannot be identified since the microphone data is lost.

For every timestamp in this sheet, they have up to 1 second error ideally. Because it is calculated based on two raw timestamps and they are annotated manually, which are only accurate to second and for each manually annotated timestamp, there is up to 0.5 second error ideally, if this stopwatch is rounder. If not, still ~1s error, because first click can have 0.99s error while second one has no error, that is 0.99s error totally.

For all types of data, the interview part is also segmented by questions if the data is collected during the interview. If you want to use that, please keep the ~1s error in mind.

There are 49 subjects’ data involved in this segmentation process totally.

# E4 data segmentation

Subject 5, 11 and 12’s data is invalid. The data stored in /participant 5 belongs to participant 4 actually. The data stored in /participant 11 is participant 7’s actually and the files in /participant 12 is participant 8’s.

For each segmented csv file, first column is data while the second column is timestamp of this frame. Only BVP data is segmented.

6 parts are segmented, they are:

Baseline (baseline data.csv): Cut 40s before step3 start.

Step3 (S3 starts.csv): read the instruction

Step4 (s4 starts.csv): encounter the actor

Step5 (S5 starts.csv): go down stairs and sign the paper

Prepare for Step6 (S5 ends.csv): device setup and greeting with interviewer.

Step6 (S6 starts.csv): interview, start from asking the first question, that is the greeting part is not included. This part is also separated into 9 parts (Q1.csv – Q9.csv), each part contain the question asking and answer period. Please keep in mind, this may have up to 1 second error theoretically.

Code: /segmentation\_code/python/E4

# Myo data segmentation

Subject 4,5,32 and 33’s data is lost.

Subject 2,3,14,16,17,25,26,27,30 and 39’s data is invalid. Detailed information see ‘myo’ sheet in segmentation\_summary.xslx

Five types of data are recorded by this device, they are accelerometer, emg, gyro, orientation and orientationEuler. During the segmentation progress, they are completely same. They share same timestamp for each frame and have same frequency.

The device is not stable at all. For each type of data, if there are more than one csv files, it means some data must be lost. For example, for orientationEuler data in /myo/Participant 22, there are two files, orientationEuler-1513819137.csv (cover the data from 12:18:57 to 12:20:11) and another one is orientationEuler-1513819276.csv (cover from 12:21:18 to 12:21:31). We can see the data from 12:20:11 to 12:21:18 is lost.

According to the dataset introduction document, Myo data should cover from step3 until end of the experiment, but actually, it only cover step 6, the interview part. Totally, 2 parts are segmented, they are:

Prepare for Step6 (accelerometer\_S5 ends.csv): device setup and greeting with interviewer. Baseline data is covered in this part.

Step6 (S6 starts.csv): interview, start from asking the first question, that is the greeting part is not included. This part is also separated into 9 parts (Q1.csv – Q9.csv), each part contain the question asking and answer period. However, this may have up to 1 second error theoretically. Because the timestamp for each step is annotated manually, which is only accurate to second.

Code: /segmentation\_code/python/E4

# Epoc data segmentation

This device contain two types of data, they share same start time but record different data with different frequency.

1. For example P11.edf, 40 data each frame and frequency is 128Hz.
2. For example P11.md.edf, 12 data each frame and frequency is 64Hz

Subject 1,6,7,8,9,10,18 and 19’s data is lost. Theoretically, another device MUSE should be used when EPOC down, but no MUSE data is stored in this dataset repo except a summary picture.

Subject 11 and 13’s data is invalid. Only 5s data is recorded.

Subject 32 only has the 128Hz data, the 64Hz one lost.

Detailed information see ‘Epoc’ sheet in segmentation\_summary.xslx

The epoc data is read and segmented by Matlab package ‘Matlab .edf reader’, output format is Matlab data format .mat

Code: segmentation\_code\matlab

Segmentation\_edf\_byquestion.m and Segmentation\_edf.m

Only step 6 is segmented, interview, start from asking the first question, in which the greeting part is not included. This part is also separated into 9 parts (for example, P4\_byquestion.mat), each part contain the question asking and answer period. However, this may have up to 1 second error theoretically. Because the timestamp for each step is annotated manually, which is only accurate to second.

# Webcam video segmentation

For each subject, at least two angles of webcam videos are available except subject no. 49, there are 33 subjects, all four angles of webcam videos are available. Detailed information see ‘webcam’ sheet in segmentation\_summary.xslx

For each subject, the raw data is one or several 5 minutes video clips. The file name indicates the video start time, for example /webcam1/participant 33/ MultiCorder2 - Microsoft® LifeCam Cinema(TM) - 12 January 2018 - 09-45-14 AM - 00011.mp4. The last index in the file name, 00011 means this is 11st video clips, so that the video begin time is 09:45:14 + 11 \* minutes = 10:40:11

Firstly, I used Python + ffmpeg merged the video if there are more than one video in this folder and calculated the video start time (/python/webcam/merge.py), then read the timestamp sheet and cut the video (/python/webcam/cut\_off.py).

Only step5 is segmented. All segmented videos start from the ending time of step 4, which the conversation between subject and actor over. Mostly, subject will appear in the webcam within 10s in the segmented videos, and the length of all videos is 90s except subject no. 18, since the subject spend more time to sign the paper.

# Thermal video segmentation

Thermal video covers step 4 ‘encounter with actor’ and step 6 ‘interview’. For subject 1 and 2, both are invalid, no subject appears in the video, for subject 10 and 42, step 4 is valid but step 6 is invalid, since the video stopped too early.

The raw data, .ats file can be played by software FLIR system ResearchIR, it also can be read by a Matlab package segmentation\_code\matlab\FilrMoiveReader.m. Each frame is represented by a matrix in Matlab, so that a segmented video is a list of matrix, or 3d array in Matlab. play\_single\_video.m can ‘play’ the output video, used to check if the segmentation is correct.

segment\_single\_video.m is used to segment single video, in case we need to cut a video manually.

Segmentation\_thermal\_byquestion.m is used to cut the interview part and output ten parts, they are baseline, Step 4 (encounter) and from Q1 to Q9 respectively.

Segmentation\_thermal\_interview.m is used to cut the interview part.

The FPS is 125.70 exactly. However, for all subject's record except 32-38, some frames are removed to simulate ~30FPS, but sometimes, a large piece of frames (e.g 5 seconds ) are removed incorrectly. The correct process of segmentation is, read the frame count of each frame, then calculate the time of each frame based on the difference from the start frame, video start time and frequency.

Another problem is, for subject 32-38, the data file is too large because of the high frequency. The RAM is not enough for cutting the entire interview part, therefore, only one quarter frames are kept, which also just in line with the rest.

Subject 32, is not segmented by question, since lack the video start time, but step 4 and entire step 6 is segmented roughly.

Three parts are segmented totally, they are:

Baseline: all participants are asked to stand in front of the camera for a minute before step 3, in the thermal video, 500 frames, around 17s data is segmented as baseline data.

Step 4: encounter the actor, this part does not include any conversation beyond the script, for example, participant asks what is next after the actor finishes the script, this is not included. Participant did not always stand in the middle of the camera, for participant 7,8,10 and 13, only half face is tracked.

Step 6: interview, greeting is not included. This part is also segmented by question (from Q1 to Q9), but it may have ~1s error, since the timestamp of video start time and question start time both is accurate to second.