GoPro Visor Write-up

June 24th, 2019

Table of Contents

1. **Overview**
2. General Intro
3. Experimental Goal
4. Methods
5. Proposed Analysis
6. University of Lethbridge - Computational Neuroscience Workshop

a. Focus

b. Background

c. Workflow

d. Results

e. Script locations

1. **Conducting the experiment**
2. **Data Structure (Locations and Descriptions)**
3. **Example Analysis Workflow (Based** on GoPro Pilot Participant 003)

**Resources**

<https://github.com/APPLabUofA/GoPro_Visor_Eye_Pi>

1. **Overview**
2. General Introduction

Experimental Goal

Replicate previous results (2017) showing that timings derived from a GoPro video recording can be used to determine stimuli onset and create subsequent ERPs. Such times first need to be corrected based on triggers simultaneously sent to a V-Amp EEG amplifier.

Methods

* Each participant will wear a pair of safety glasses containing six evenly-spaced LEDs placed along the top rim of the glasses
* Participants will be seated in a dimly lit EEG chamber with a GoPro 5 camera placed directly in front of them
* The GoPro will be placed such that it can fully record all six LEDs placed on the visor
* Participants will complete a visual oddball task administered by a Raspberry Pi 3 B+ computer
* The LEDs will be connected to the GPIO pins of the Pi
* Each participant will complete 1000 trials whereby the LEDs will randomly flash either green (standards) or blue (targets)
* Targets will be presented on 20% of trials and standards will be presented on 80% of trials
* Each flash will be presented for 500ms, followed by a variable delay randomly selected between 1000-1500ms
* Each participant will be fitted with a 16 channel EEG electrode cap containing active electrodes and passive EOG electrodes to monitor horizontal and vertical eye movements
* These electrodes will be connected to a Brainvision V-Amp amplifier which will be used to record EEG and trigger data
* The Pi will also be connected to the serial input port of the V-Amp via eight GPIO pins in order to send trigger information
* Voltages will be sent through the eight GPIO pins connected to the V-Amp, turning said pins on or off, coincident with the onset and offset of each LED flash
* Participants will be asked to press a button, attached to the GPIO pins, when they are shown targets and to withhold responses to standards
* Triggers will be sent to mark the onset of each response
* To mark the beginning and end of the experiment the LEDs will turn red and a trigger will be simultaneously sent to the amplifier
* The LED flashes and triggers will allow us to sync the GoPro recordings to the recorded EEG data

Proposed Analyses

* Analysis techniques used will be identical to those used in previous P3 papers
* For each participant, ERPs will be generated following the onset and offset of each target and standard
* ERPs will be generated based on the trigger timings recorded by the amplifier, LED onset and offset based on the GoPro video, and the timings recorded by the Pi
* As the amplifier trigger times are considered the ideal standard, GoPro and pi times will be adjusted to best match the amplifier trigger times
* Times will be adjusted using the matlab function mdl = fitlm(X,y)
* ERPs will be generated using the adjusted and unadjusted timings
* Participant ERPs will then be averaged to form grand-average ERPs for targets and standards
* Difference waveforms will also be generated by subtracting the standard grand-average ERP from the target grand-average ERP
* Topographies will be created for our standards, targets, and difference waveforms using the average voltage across the P3 waveform
* A P3 response, a positive-going deflection typically around 300-600ms, is expected to be observed in the grand-average target ERPs and the difference waveform
* The average voltage during this P3 time window will be determined for each participant and will be used for subsequent analyses
* T-tests of these average voltages will be used to show if a significant P3 response has been measured using each of the different timing conditions (amplifier, GoPro, Pi; adjusted, unadjusted)
* The t-tests will also be used to compare voltages across our conditions
* Following adjustment, we expect the P3 responses to not be significantly different across conditions
* Root Mean Squared (RMS) values will also be calculated during the baseline period (first 200ms prior to stimulus onset) on both the individual participant ERPs and the grand-average ERP waveforms
* RMS will be calculated using a permutation test similar to previous P3 papers and provides an indication of the amount of noise present in our signal
* Ideally during the baseline period our voltage activity should be close to 0, resulting in small RMS values
* These RMS values will then be compared across timing condition using a Wilcoxon Rank Sum Test
* RMS values for 10,000 permutations will be calculated along with average RMS values
* An FFT will also be calculated from our recorded data to understand power at various frequencies
* The FFT results will be compared across timing conditions and we expect all conditions to show the typical peak in the alpha band (8-12Hz)
* Power calculations will be done to determine the number of trials needed to measure a P3 response with 80% power (as per previous P3 papers)

Expected Results

* A significant P3 response is expected following targets but not standards
* The P3 response should be largest based on EEG trigger timings
* GoPro video timings are expected to differ significantly from the trigger timings, results in a diminished or non-existent P3 response
* Stimulus onset times recorded by the Raspberry Pi are expected to be comparable to the EEG trigger times, resulting in a similar P3 response
* Following adjustment of the GoPro timings based on the EEG trigger times, ERPs and then expected to be comparable
* GoPro ERPs will be the most accurate if the adjustments are calculated for each participant compared to calculating the adjustments using the grand-average difference between timings
* Similar results will be expected for all other types of analyses (eg. Different RMS values before adjustment, similar RMS values after adjustment)

1. University of Lethbridge - Computational Neuroscience Workshop

The following work represents one of two exploratory in-class projects between June 9th and 17th, 2019

Focus

* Recomputing auditory oddball derived ERPs based on surrounding stimuli.

Background

* Utilizing data recorded from 2017 which included EEG, GoPro, and GPS data.
* Auditory oddball task while cycling in quiet vs. traffic conditions
* Alignment of data streams is derived from the recording of all other components with the GoPro and standardizing to the EEG times

Workflow

* Video was recorded at 240 fps, this was converted to 24 fps to speed up image processing
* EEG embedded events were transformed into frame numbers
* Epochs of videos were extracted and classified using YOLOv3 object detector
  + Combines object location and recognition

Results

* Currently working to categorize epochs based on extracted video objects attributes
  + Restricting classification groups and differentiating into:
    - Pedestrians
    - Cyclists
    - Vehicles
* Stuck working in Matlab for ML, up until the last day
* Progress with Google Collab was much faster, would strongly recommend against Matlab for ML.

Scripts

If you are interested in using any of the scripts developed over my time at the workshop you can find them here:

M:\Analysis\GoPro\_Park

|--- EEG

| |--- Load\_In\_EEG

| |--- Recategorize\_ERPs

| |--- Recategorize\_Rt

|--- Video

| |--- Load\_In\_Video

| |--- MP4\_Convert\_Reduced\_FPS

| |--- CT\_CUT\_Frame (Cuts out all target frames out one by one - constructs a data

| | structure with positioning for each frame)

| |--- CT\_CUT\_Video (Cuts out a separate video for each identified event)

|--- Watch

|--- P3\_BikePark\_Watchdata (Script for cutting out and plotting appropriate segments)

1. **Conducting the GoPro Visor Experiment**
2. The day before the experiment, make sure the GoPro is either fully charged or plugged in when you leave for the day
3. Before the participant arrives setup up equipment as follows:
   1. RaspPi HDMI to ViewPixx DVI
   2. Custom RaspPi GPIO ribbon cable to VAMP via 25 pin serial port
   3. 2.5 Amp power cable to RaspPi
   4. Make sure the GoPro is set to 720 resolution and 240 fps
4. Consent form (form #1)
5. Head measurements (circumference & midline)
6. Enter Database info + Assign subject and study IDs
7. Set up EOG and place on subject
8. Set up EEG cap and place on subject
9. Obtain sufficient impedance level and clean signals (<5 KOhm)
10. Explain tension, motion, and eye artifacts
11. Explain task to the subject
12. Hand them the button and place visor over the EEG cap
13. Open experiment program
    1. Open python terminal
    2. cd /documents/GitHub/GoProVisor/Experiment/
    3. python3 Visual\_P3\_GoPro\_Visor
14. Make sure participant understands the experiment fully
15. Unplug laptop/surface and start saving EEG data (ensure correct workspace and adherence to correct naming convention is followed)
16. Take extraneous electronics from participant and turn off lights
17. Ensure they know to call you when you are done
18. Begin Experiment
19. Monitor EEG signals + fill out Run Sheet
20. Stop saving EEG data
21. Unplug electrodes, remove EEG cap and have the participant remove the EOGs
22. Clean up the subject
23. Wash and Sanitize electrodes (tips up!!!) + wash cap
24. Hang electrodes (tips up!!!) and net to dry
25. Debrief subject
26. Plug in the surface
27. Return all electronic to where they were previously
28. Plug in the GoPro
29. Turn off ViewPixx monitor
30. Clean up
31. **Data Structure**

M:\Data\GoPro\_Visor\Experiment\_1

|--- EEG\_Data (Output from AMP)

| |--- 00x\_GoPro\_Visor\_Eye\_Pi.eeg

| |--- 00x\_GoPro\_Visor\_Eye\_Pi.vhdr

| |--- 00x\_GoPro\_Visor\_Eye\_Pi.vmrk

| |--- Camera\_P3 (Preprocessing outputs these segments)

| |--- Camera\_Latencies

| | |--- …00x\_High\_Tones

| | |--- …00x\_Low\_Tones

| |--- EEG\_Latencies

| |--- …00x\_High\_Tones

| |--- …00x\_Low\_Tones

|--- Pi\_Times

| |--- Times saved locally from pi

|--- Video

|--- Converted

| |--- Whole (One video per participant)

| |--- output

| |--- 00x.avi

|--- Original (GoPro cuts videos into ~4 GB segments)

|--- 00x\_01.mp4

|--- 00x\_02.mp4

|--- Conversion\_Combine (Commands to format video)

**4. Example Analysis Workflow**

The following example takes you through processing GoPro Pilot Participant 003 All analysis scripts can be found in the accompanying GitHub repo

All python scripts ran from a python virtual environment (see requirements.txt in GitHub repo)

**1. Convert Video**

* From Windows Command Prompt run each command from Conversion\_Combine.txt

or

* Automated\_Conversion.py from a python terminal (convenient for converting multiple videos)

**2. Analyze Video** (Python)

Pull\_Events\_Preprocess.py

* + This pulls out and differentiates flash events in the video
  + Preprocesses by
    - subtracting first event latency from all events
    - converts from frames to seconds
    - pulls out specific event types
  + Depending on the video can take a significant amount of time to run
  + Automatically saves the start frame and event type of each flash to a .csv file

**3. Compare EEG events vs. Camera events + Transform** (Python)

a. If you would like to comparing timing before **and** after transforms on a relevant sampling rate (ms):

Run Compare\_Timing.py (uses .csv output of Pull\_Events\_Preprocess.py)

* + Aligns and compares EEG events to pulled events
    - Plots latency and difference distributions for raw aligned and linearly transformed events
  + Saves all relevant data (incl. differences) to a single .csv
    - ERPs can be created from both transformed and untransformed (aligned and in milliseconds) as shown in Step #4

b. If you would like to skip comparing times and transform directly from frames to EEG sampling points:

Run Pull\_Frames.py

* + Takes out extraneous events and transforms
  + Saves frame time and event type to a .csv
  + If you want to look at ERPs derived from frame number, this script also provides that option

**4. Plot Group Distributions**

Group\_Figures.py

* + If you followed step 3.a, you can plot the group distribution of untransformed and transformed times
  + Currently no stats scripts have been made

**4. Compute ERPs** (Matlab)

Wrapper\_Camera\_P3 (Change settings for Preprocessing and Wrapper)

exp.preprocess = 2; %%%

1 for EEG

2 for Camera

3 for LOADING PROPROCESSED DATA

exp.corrected = 2; %%%

1 = Uncorrected (Camera events are in milliseconds/aligned),

2 = Corrected (milliseconds to milliseconds)

3 = Frames Offset (not aligned)

4 = Frames Aligned

5 = Corrected (Frames to milliseconds)

Preprocessing\_Camera\_P3

Will recode events according to condition and event type

Will output high and low tones separately for each camera and EEG

Analysis\_Camera\_P3

Loads both Camera and EEG into an ALLEEG structure

Camera\_P3\_ERP

Plots ERPs, difference ERPs, topoplots, difference topoplots