2018. Oct. 2nd

MoBI Analysis Tutorial

Hiroyuki Kambara

Tokyo Institute of Technology

Outline

I. Getting Started

(I-i) About This Tutorial

This tutorial is made to explain our pipeline for juggling experiment data analysis as an example of data analysis using MoBILAB toolbox. Since some of the processes are highly customized for juggling data analysis, I will not explain those process in detail nor provide the programs for those processes. Instead, I will provide outcomes of those programs necessary to follow the steps in our pipeline. Also I will not explain about EEG signal processing. Please refer to EEGLAB Tutorial to study how to process EEG signal by EEGLAB toolbox.

(I-ii) Goal of Data Analysis

The goal of our data analysis is to find characteristic features in EEG signal related to behavior events during juggling task.

(I-iii) Experiment and Recorded Data

Subjects were asked to perform three-balls cascade jugging as long as possible.

Data recorded during experiment are

- EEG (Biosemi 205ch, 2048 Hz sampling),
- Motion Capture (Phasespace, 13 channels, 420 Hz sampling),
- Video (Sony 640 x 480 pixels, 60fps).

(I-iv) Computer System

The OS and Matlab and EEGLAB versions that I used are

- Mac OSX El Capitan (10.11.6),
- MATLAB 2014a (MATLAB 2017b for ball tracking process),
- eeglab13_6_5b.

(I-v) Data Set and Programs Provided for Tutorial

Since the size of original data (.xdf) file is to large, most of the recording period in the original data are trimmed for this tutorial. The data set after trimming includes first 4 juggling trials. In addition, MoBILAB format data set files, instead of .xdf file, are provided.

The data set provided for this tutorial are listed below.

- MoBILAB data stream files
  - Folder : mobi_tutorial/data/juggling_MoBI
  - Files : three sets of binary data (.bin) and header (.hdr) files for three data streams after trimming period
    - eeg_CESlab102_xxxx.bin
    - eeg_CESlab102_xxxx.hdr
    - mocap_CESlab102_yyyy.bin
    - mocap_CESlab102_yyyy.hdr
    - videostream__untitled_zzzz.bin
    - videostream__untitled_zzzz.hdr

- Movie files
  - Folder : mobi_tutorial/data/moviefiles
  - Files : recorded movie files after trimming period
    - juggling.mov (for MacOS)
    - juggling.asf (for WinOS)

- Supplemental data files
  - Folder : mobi_tutorial/data/supplemental_datafiles
  - Files : balls trajectory, events information, motion capture marker connection data , and channel 3D location files
    - ball_pos.mat
    - ball_vel.mat,
    - ball_acc.mat
    - event_catch.mat
    - event_release.mat
    - event_balltop.mat
    - event_ballbottom.mat
    - mocap_markerConnection.mat
    - channel_location.sfp

The Matlab program files provided for this tutorial are listed below. All programs are stored in mobi_tutorial/programs folder.

- Matlab scripts adding new data streams to MoBILAB data set
    - script_addStream_BallsPosition.m
    - script_addStream_BallsVelocity.m
    - script_addStream_BallsAcceleration.m
- Matlab script adding juggling related events to MoBILAB data streams
    - script_insertEvents.m
- Matlab function launching GUI for event marking
    - mobi_gui_addEvent.m
- Matlab function launching GUI for event modification and removal
    - mobi_gui_modifyEvent.m
- MoBILAB class slightly modified from "streamBrowserHandle" class
    - streamBrowserHandle_HK.m
- MoBILAB class slightly modified from "videoStreamBrowserHandle1" class
    - videoStreamBrowserHandle1_HK.m

The last two classes are used in GUIs for event marking and event modification and removal.

II. Overview of Analysis

(II-i) Framework of MoBI Analysis

Fig. II-1is a schematic diagram showing a framework of MoBI analysis using MoBILAB toolbox. Note that this is one of the examples of the way MoBILAB toolbox can be utilized.

Analysis starts by loading experiment data by MoBILAB toolbox. The main purpose of analysis by MoBILAB is to mark events. To find timing of events, you can browse multi-modal data streams and apply basic preprocessing, e.g. filtering and temporal differentiation, to originally recorded data streams. If you need to process data in more specific way, you can use your own programs and/or other toolbox. Once the events are marked, EEG data stream can be exported as .set format which can be loaded by EEGLAB toolbox. Then you can analyze EEG data with EEGLAB.

When you want to analyze data in MoBILAB data set with your own scripts or other toolbox, you need to load/access the data in MoBILAB format in some way. Data in

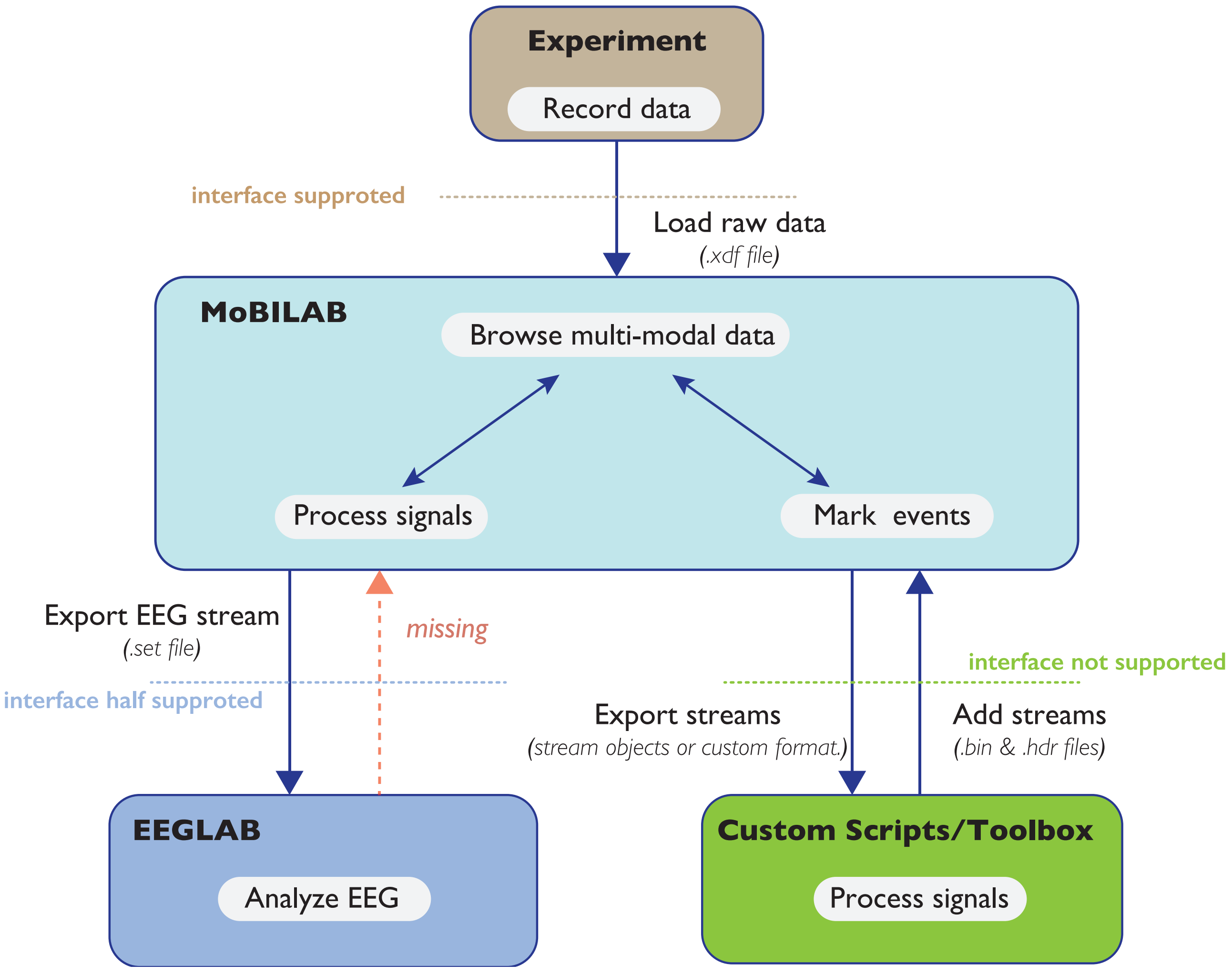


Fig. II-1 : Framework of juggling data analysis

MoBILAB format can be loaded as data stream class objects if custom-made scripts are written in MATLAB code. If custom-made scripts are written in different program language, data must be exported to other formats suitable for the scripts. Similarly, when using ready-made toolbox (e.g. EEGLAB), data should be exported to format suitable for the toolbox. The data format of EEGLAB is one of the formats MoBILAB can converted into. In other word, the ''eeg'' stream object data can be exported as .set file that can be loaded with EEGLAB programs.

(II-ii) Flow chart of Juggling Data Analysis

Fig. II-2 is a flow chart of juggling data analysis pipeline. Each of the steps in the pipeline is placed under the name of toolbox used to process data.

step (i)	Raw recording data in .xdf file format is loaded to create MoBILAB data set folder which contains sets of binary file (.bin) and matlab .mat format file (.hdr). Each set of .bin and .hdr files is made for each data stream.
step (ii)	Missing samples in motion capture stream due to marker occlusion are reconstructed by interpolation.
step (iii)	The events corresponding to start and end of each trial are marked with browsing motion capture and video data streams.
step (iv)	Trajectories of three balls are computed by tracking the position of the balls from frame images in movie file. In addition, timings of catching, releasing, and balls peaks are automatically detected from balls position, velocity, and acceleration data.
step (v)	New data streams corresponding to balls position, velocity, and acceleration are added to MoBILAB data set by making sets of .bin and .hdr files for balls position, velocity and acceleration data.
step (vi)	The events information of catching, releasing and balls peaks are saved to MoBILAB data set.
step (vii)	Timings of catching and releasing events automatically detected from balls trajectory are checked by browsing balls trajectory streams and video data stream and corrected if the timing seems wrong.
step (viii)	"eeg" data stream is exported as .set file.

step (ix) EEG data is processed by EEGLAB toolbox.

step (x) Continuous EEG data is segmented to epochs time locked to juggling event. Epochs can be selected and/or sorted according to some juggling performance indices computed from balls trajectory data.

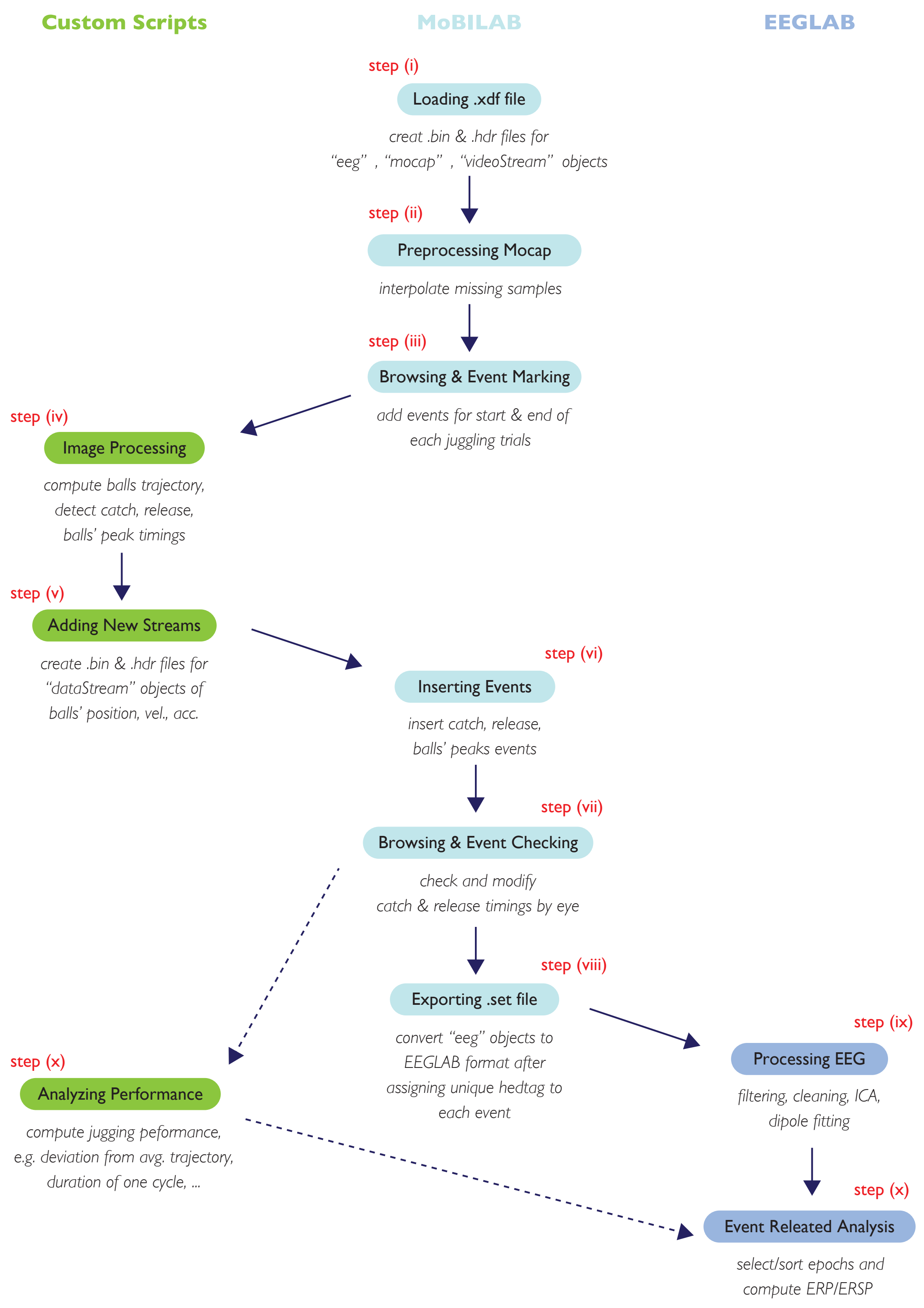

Custom Scripts
MoBILAB
EEGLAB
step (i)
Loading .xdf file
creat .bin & .hdr files for "eeg", "mocap", "videoStream" objects
step (ii)
Preprocessing Mocap
interpolate missing samples
step (iii)
Browsing & Event Marking
add events for start & end of each juggling trials
step (iv)
Image Processing
compute balls trajectory, detect catch, release, balls' peak timings
step (v)
Adding New Streams
create .bin & .hdr files for "dataStream" objects of balls' position, vel., acc.
step (vi)
Inserting Events
insert catch, release, balls' peaks events
step (vii)
Browsing & Event Checking
check and modify catch & release timings by eye
step (viii)
Exporting .set file
convert "eeg" objects to EEGLAB format after assigning unique hedtag to each event
step (ix)
Processing EEG
filtering, cleaning, ICA, dipole fitting
step (x)
Analyzing Performance
compute jugging peformance, e.g. deviation from avg. trajectory, duration of one cycle, ...
step (x)
Event Releated Analysis
select/sort epochs and compute ERP/ERSP

Fig. II-2 : Flow chart of juggling data analysis

(III) Step-by-Step Process in Juggling Data Analysis

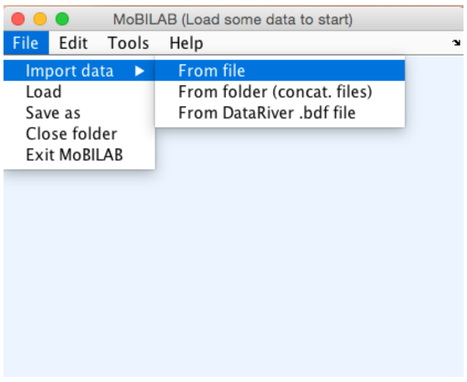
(III-i) Importing Experiment Data to MoBILAB

The first step is to import raw data to MoBILAB by following steps.

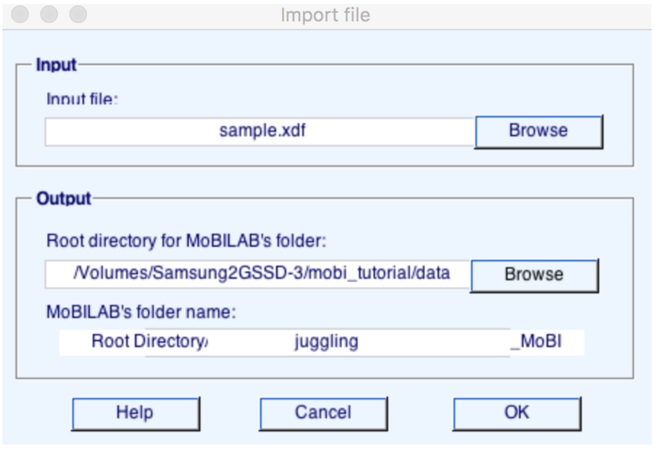
1. launch MoBILAB GUI window by entering ***runmobilab*** at MATLAB command line.

```
>> runmobilab
```

2. click ***File*** >> ***Import data*** >> ***From file***



3. select ***Input file*** (raw data file (.xdf)) and ***Root directory for MoBILAB's folder*** (directory where MoBILAB format data folder is saved)


Import file

Input

Input file:

sample.xdf

Output

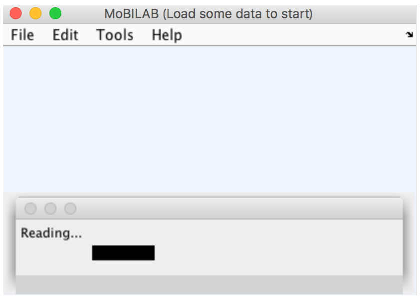
Root directory for MoBILAB's folder:

/Volumes/Samsung2GSSD-3/mobi_tutorial/data

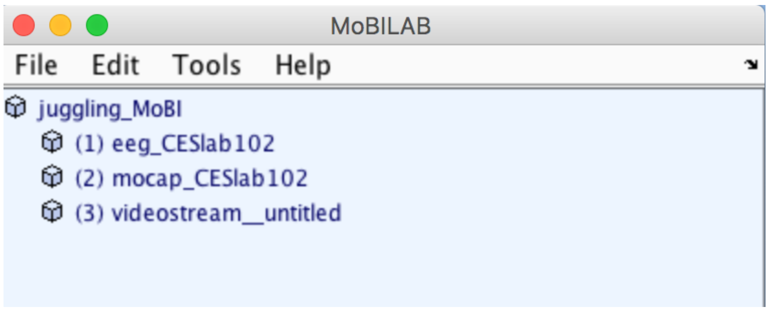
MoBILAB's folder name:

Root Directory/ juggling _MoBI

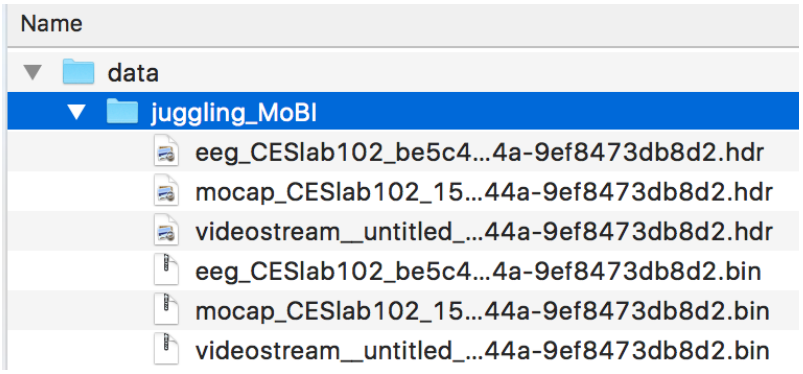
4. wait until data import is finished and streams information appears on GUI



When data import is finished, names of MoBILAB data streams are shown on MoBILAB GUI window.

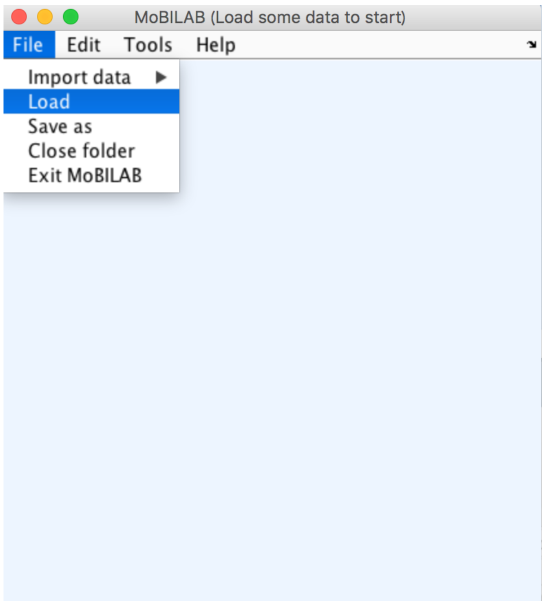


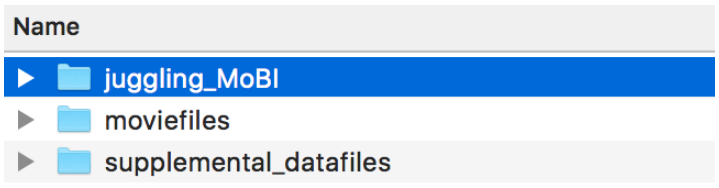
In addition, a folder containing sets of binary file (.bin) and matlab .mat format hearder file (.hdr) is created. Each set of .bin and .hdr files is made for each data stream. The MoBILAB data set folder must have letters "_MoBI" at the end of folder name.



Name
- data
  - juggling_MoBI
    - eeg_CESlab102_be5c4...4a-9ef8473db8d2.hdr
    - mocap_CESlab102_15...44a-9ef8473db8d2.hdr
    - videostream__untitled_...44a-9ef8473db8d2.hdr
    - eeg_CESlab102_be5c4...4a-9ef8473db8d2.bin
    - mocap_CESlab102_15...44a-9ef8473db8d2.bin
    - videostream__untitled_...44a-9ef8473db8d2.bin

Once the MoBILAB data set folder is created, streams data can be loaded by clicking ***File*** >> ***Load*** and selecting the data set folder.

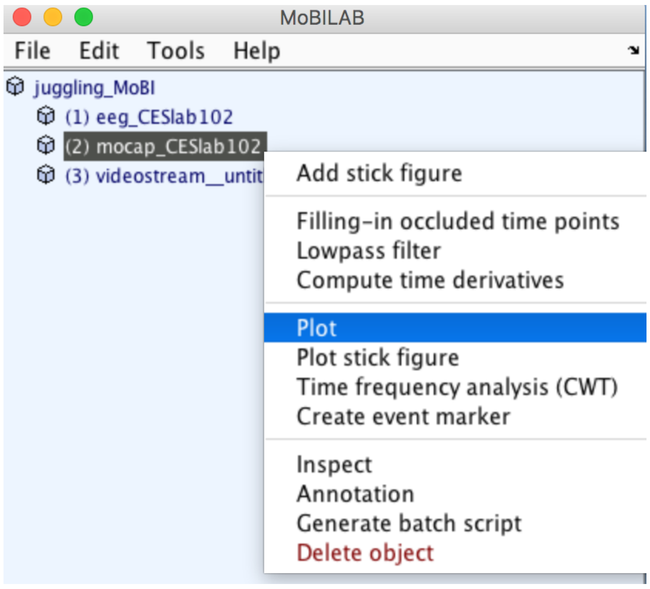




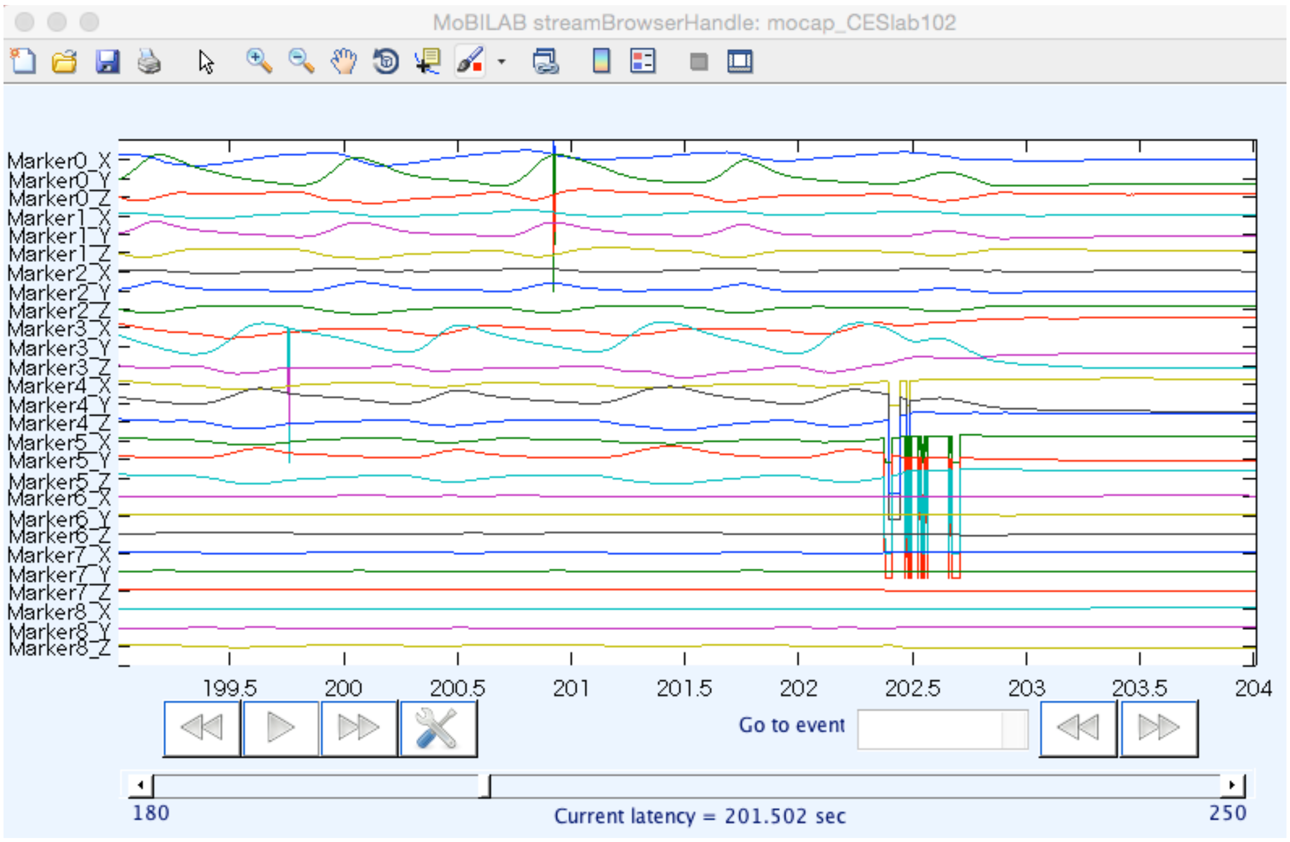
Name
- juggling_MoBI
- moviefiles
- supplemental_datafiles

(III-ii) Interpolating Motion Capture Data

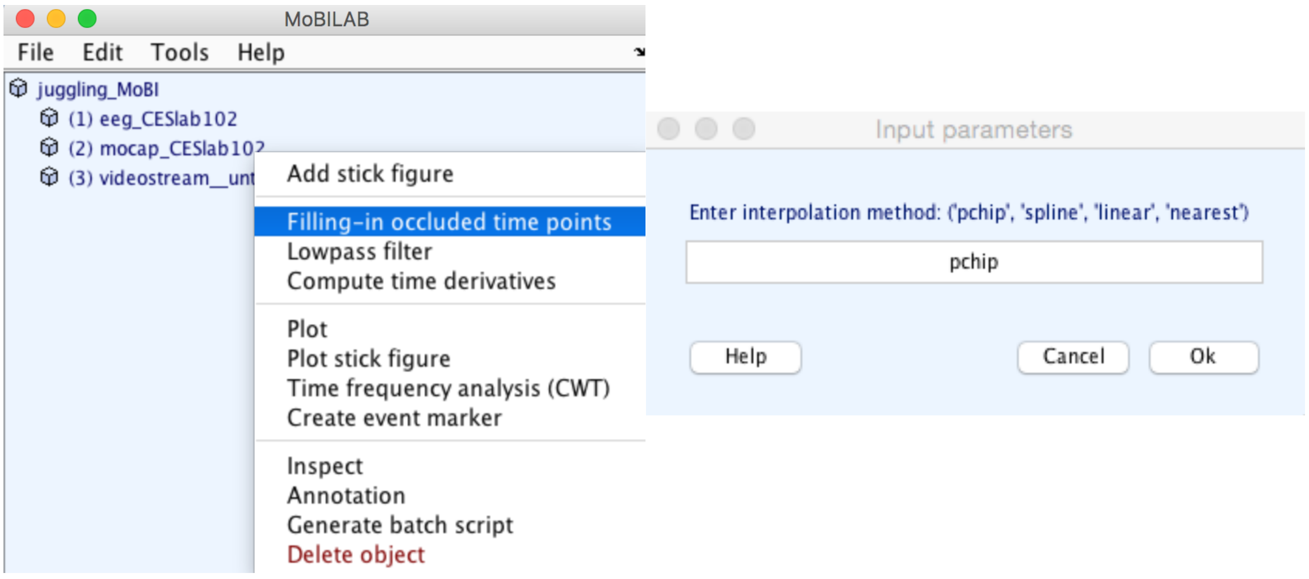
After loading MoBILAB data set folder, you can browse each data stream independently, or several data streams simultaneously. For example, if you want to browse motion capture data, right click the stream of motion capture and select ***Plot*** menu. A new window showing time series of markers position will pop up.



The raw position data may be a bit noisy and missing some sampling points due to occlusion of markers during recording. For example, the rectangle bumps seen in marker position signals correspond to the missing data points.



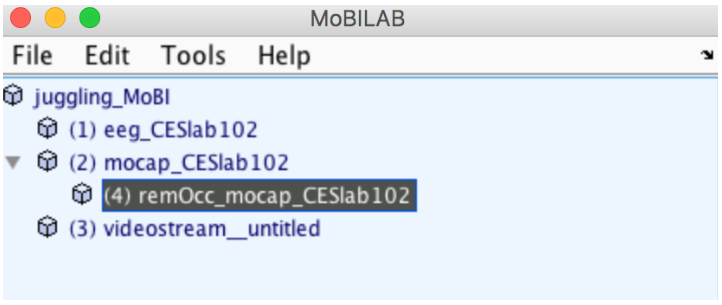
MoBILAB toolbox has the function to filling-in these missing data points. To do that , right click mocap data stream and select ***Filling-in occluded time points*** menu. The missing points will be interpolated by the algorithm chosen at the window appearing after selecting the menu.



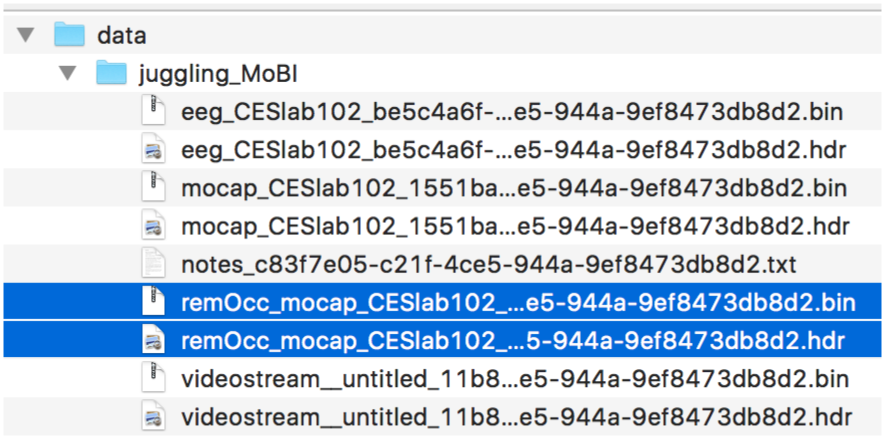
Enter interpolation method: ('pchip', 'spline', 'linear', 'nearest')

pchip

After interpolation process, a new data stream with the name starting from letters “remOcc” followed by original mocap stream name is made. You can see the name of new data stream in MoBILAB GUI window. In addition, .bin and .hdr files for the new data stream are saved in the data set folder.

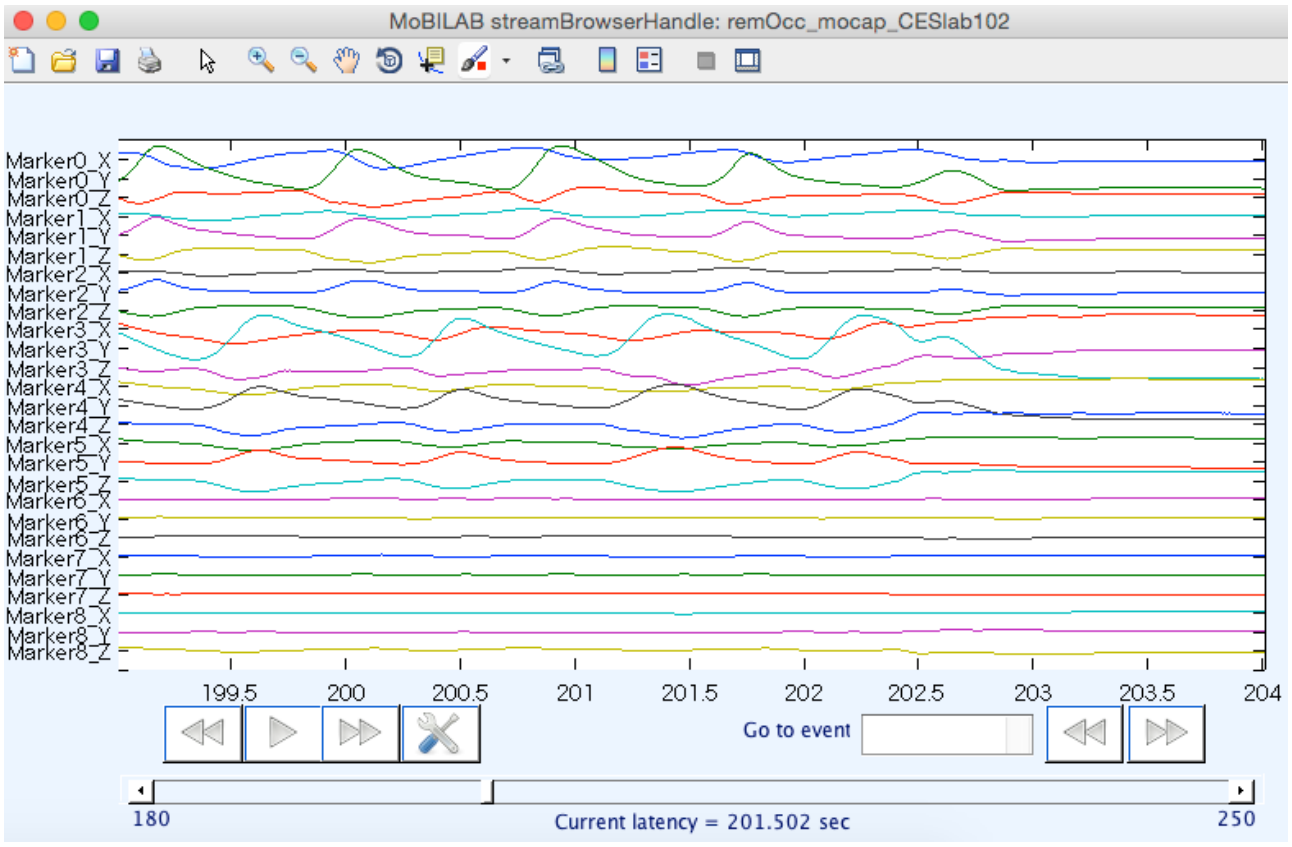


juggling_MoBI
(1) eeg_CESlab102
(2) mocap_CESlab102
(4) remOcc_mocap_CESlab102
(3) videostream__untitled



data
juggling_MoBI
eeg_CESlab102_be5c4a6f-...e5-944a-9ef8473db8d2.bin
eeg_CESlab102_be5c4a6f-...e5-944a-9ef8473db8d2.hdr
mocap_CESlab102_1551ba...e5-944a-9ef8473db8d2.bin
mocap_CESlab102_1551ba...e5-944a-9ef8473db8d2.hdr
notes_c83f7e05-c21f-4ce5-944a-9ef8473db8d2.txt
remOcc_mocap_CESlab102_...e5-944a-9ef8473db8d2.bin
remOcc_mocap_CESlab102_...5-944a-9ef8473db8d2.hdr
videostream__untitled_11b8...e5-944a-9ef8473db8d2.bin
videostream__untitled_11b8...e5-944a-9ef8473db8d2.hdr

The new data stream can be browsed by selection ***Plot*** menu after right clicking the stream name in the GUI window. Now you will observe that the rectangle bumps seen in original mocap data stream have been disappeared.



(III-iii) Browsing Multi-Modal Streams and Marking Events

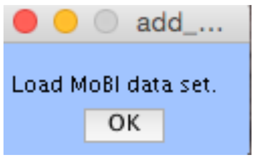
Next step is to detect start and end timings of juggling trial by browsing video stream data and mark those timings as events in all of the data streams.

MoBILAB toolbox support browsing multi-modal data streams simultaneously and also inserting events from multi-modal stream browsing GUI window (see MoBILAB Wiki).

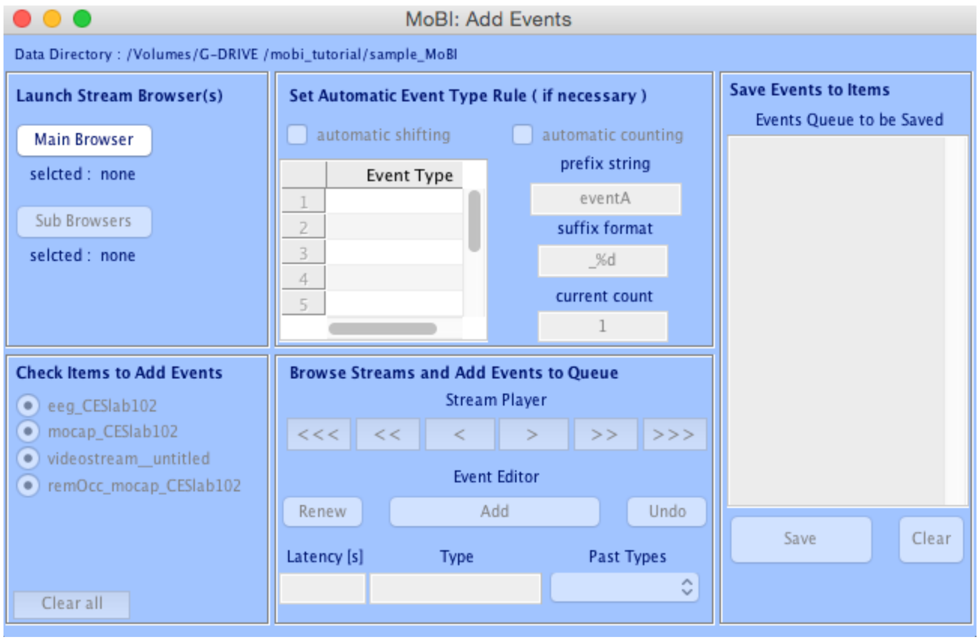
I also made a GUI for marking events with browsing several streams simultaneously. The original multi-modal browsing function of MoBILAB toolbox is focusing more on browsing than marking events. Since I needed to mark a lot of events for juggling data analysis, I made own GUI that focuses more on events marking. Program for the GUI is written in MATLAB code and using data stream classes and data stream browser classse defined in original MoBILAB toolbox. Since the browsing windows are made by data stream browser class objects in original MoBILAB toolbox, browsers' properties can be changed in the same way as original MoBILAB.

The GUI can be launched by running "mobi_gui_addEvent.m" function. Basic steps to mark events by the GUI is the following.

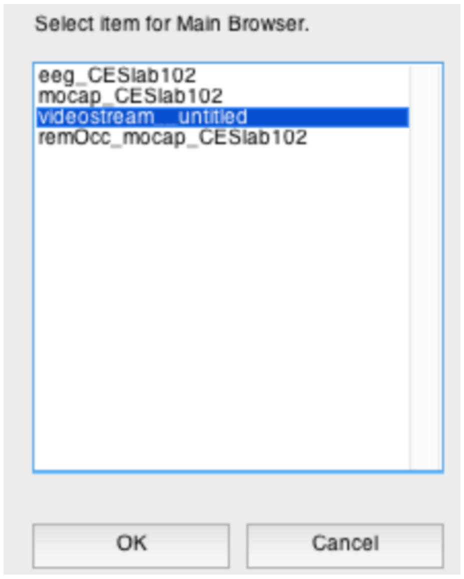
1. Type and enter "***mobi_gui_addEvent();***" in Matlab command line.
2. Click ***OK*** button in a pop-up window asking for loading MoBILAB data set.



3. Select MoBILAB data set folder, then the GUI window pops up.



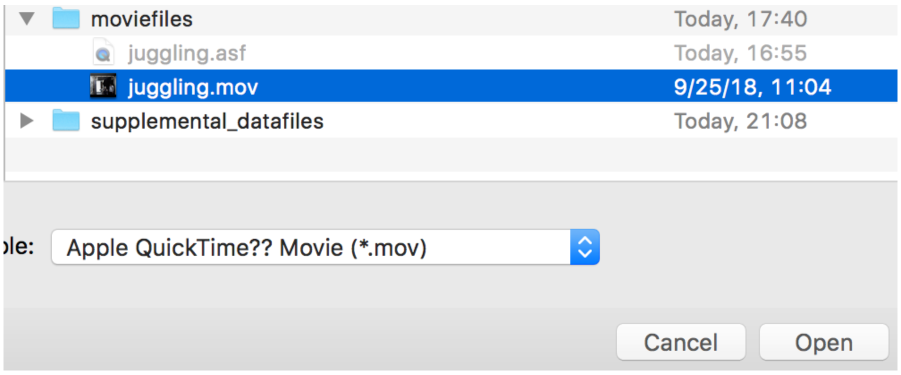
4. Click ***Main Browser*** button in the upper-left panel, and select video data stream in a list selection dialog box.



Select item for Main Browser.

eeg_CESlab102
mocap_CESlab102
videostream__untitled
remOcc_mocap_CESlab102

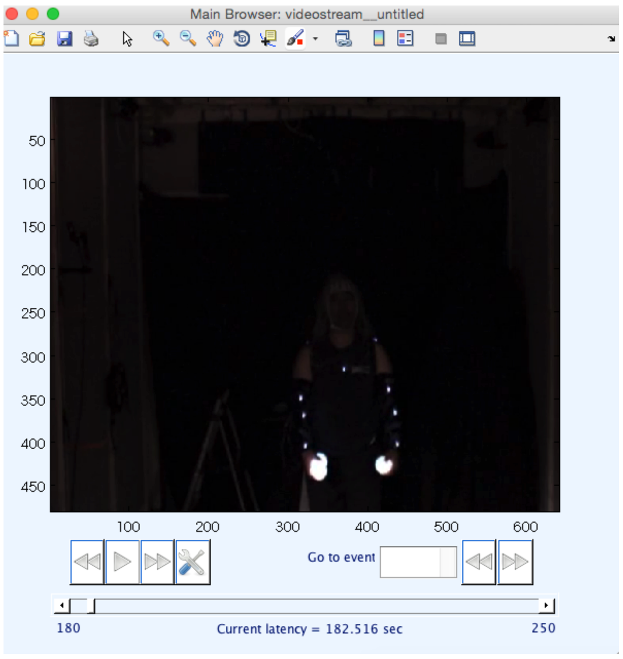
5. Select movie file at a file selection dialog box.



moviefiles	Today, 17:40
juggling.asf	Today, 16:55
juggling.mov	9/25/18, 11:04
supplemental_datafiles	Today, 21:08

le: Apple QuickTime?? Movie (*.mov)

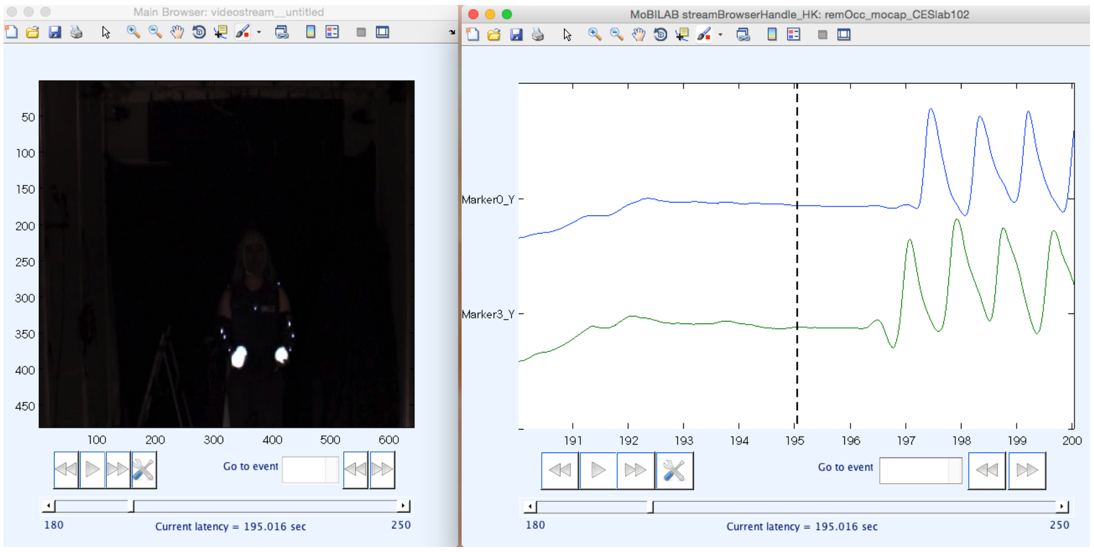
6. Wait for a while video data stream browser appears.



Main Browser: videostream__untitled

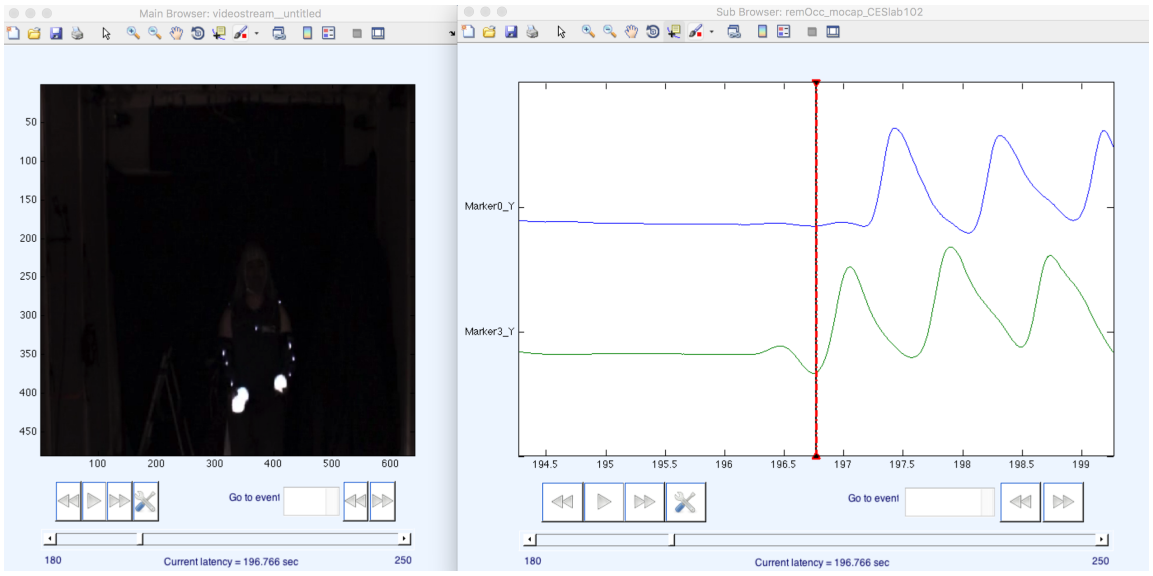
Current latency = 182.516 sec

7. If you want to browse other data streams, click ***Sub Browsers*** button and select data stream(s) you want to browse.


Main Browser: videostream__untitled
MoBILAB streamBrowserHandle_HK: remOcc_mocap_CESlab102
Marker0_Y
Marker3_Y
Go to event
Current latency = 195.016 sec
180
250

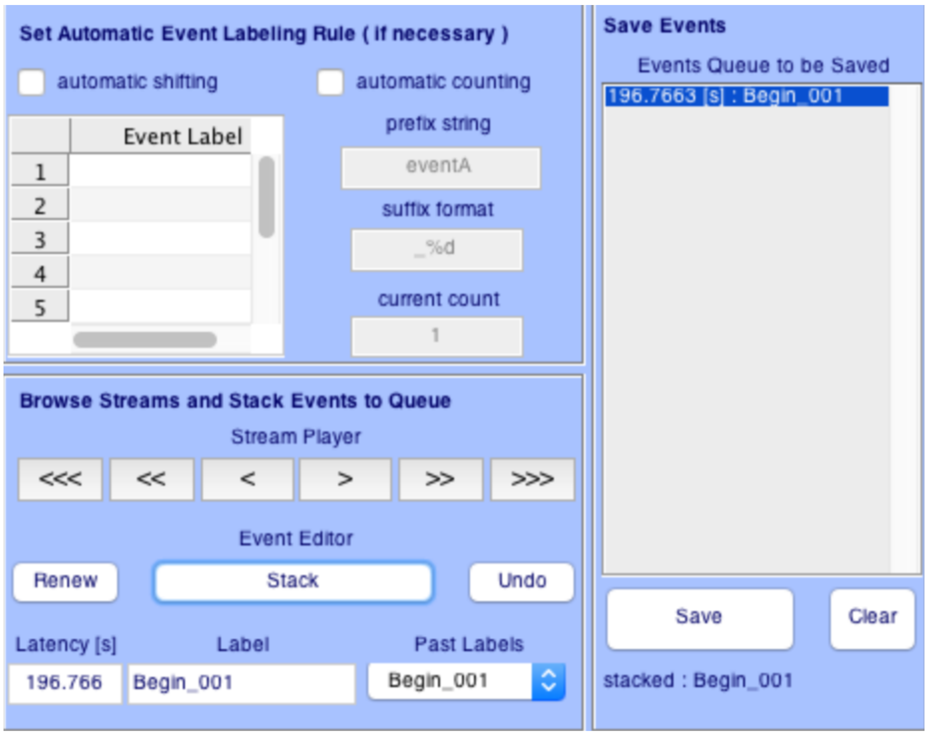
MEMO : To detect start and end of juggling trials, motion capture's markers position signals are useful because they show cycle waves during juggling trials. So I usually launch motion capture browser and set its properties to plot vertical position of the markers attached to left and right wrists. (In sample data set, channel #2 and #11 corresponds to position of left and right wrist in vertical direction, respectively.) Then I detect rough timing of trial start and end from motion capture browser.

8. Click >, >>, >>>, < , << ,<< buttons in the bottom-mid panel to rewind/forward current latency of the browser and stop at the moment of trial start (or trial end).


Main Browser: videostream__untitled
Sub Browser: remOcc_mocap_CESlab102
Marker0_Y
Marker3_Y
Go to event
Current latency = 196.766 sec
180
250

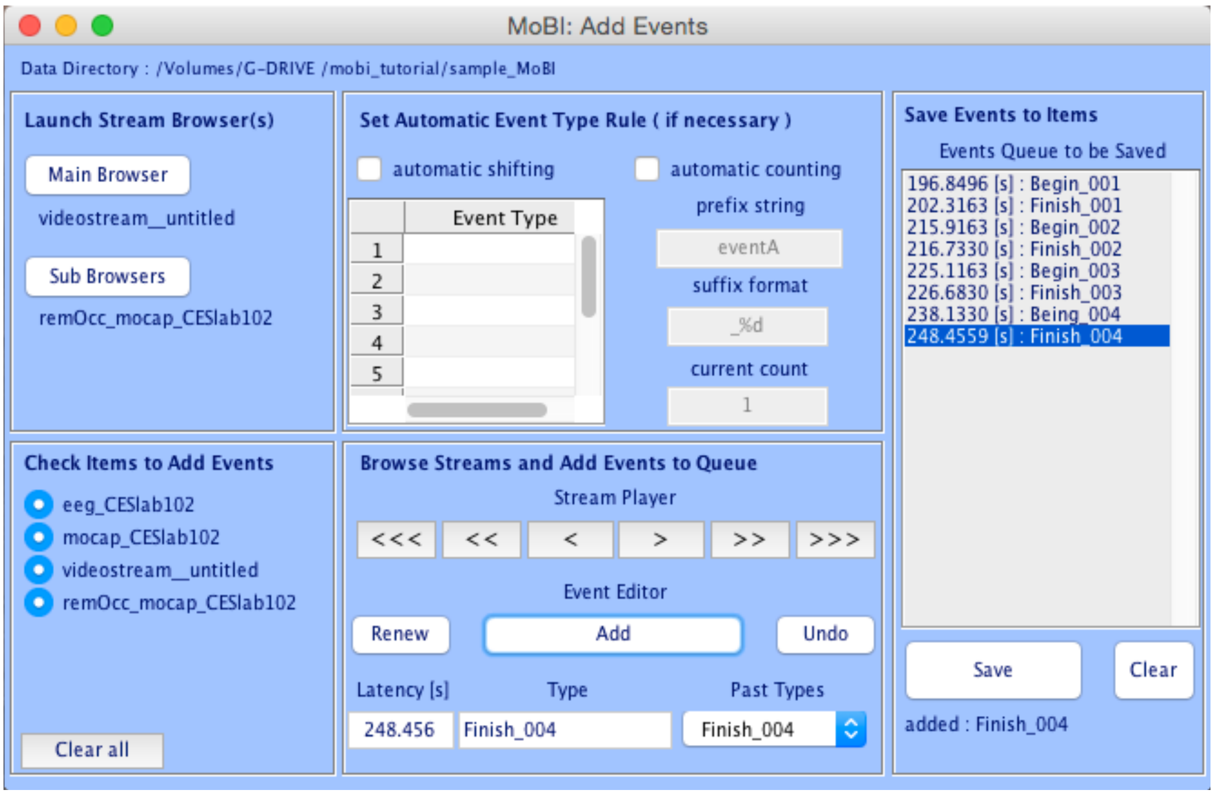
MEMO : I defined the timing of juggling trial start as the moment the ball in the right hand starts moving upward for the first throw. The timing of trial end is more ambiguous than trial start since the way trial ends differs by trials. Examples of trial end definitions are, the moments a ball is miss catched, two balls hit with each other, subject quit juggling, and etc.

9. Enter the name of event into the text box under ***Type*** and click ***Stack*** button to stack the event into a events queue shown in the right panel. Note that the event is not saved to MoBILAB data set at this point.



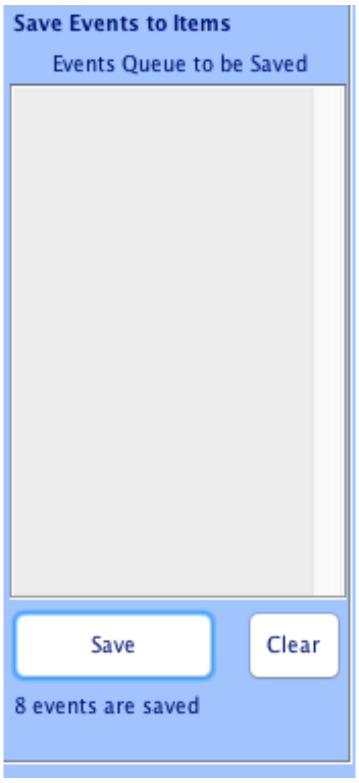
MEMO : The name of trial start events is defined as "Begin_001", "Begin_002", …, and those of trial end is defined as "Finish_001', "Finish_002", ….

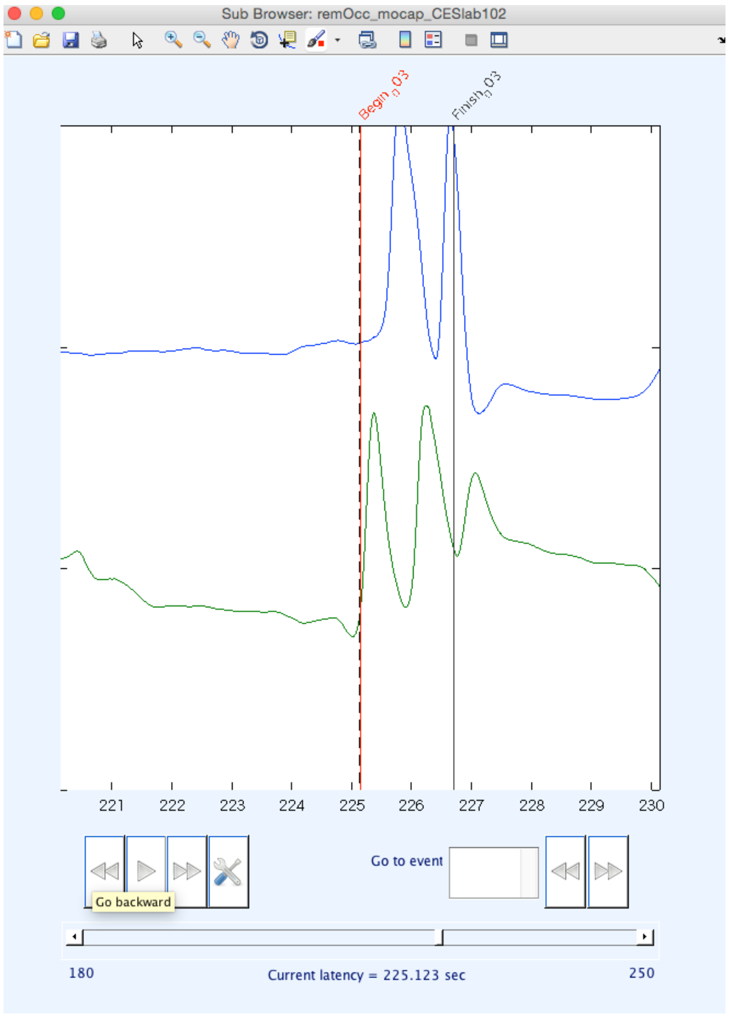
10. Iterate steps 8 and 9 and stack events to be inserted to the data streams.



```
196.8496 [s] : Begin_001
202.3163 [s] : Finish_001
215.9163 [s] : Begin_002
216.7330 [s] : Finish_002
225.1163 [s] : Begin_003
226.6830 [s] : Finish_003
238.1330 [s] : Being_004
248.4559 [s] : Finish_004
```

11. Click ***Save*** button to save events in the events queue to MoBILAB data set



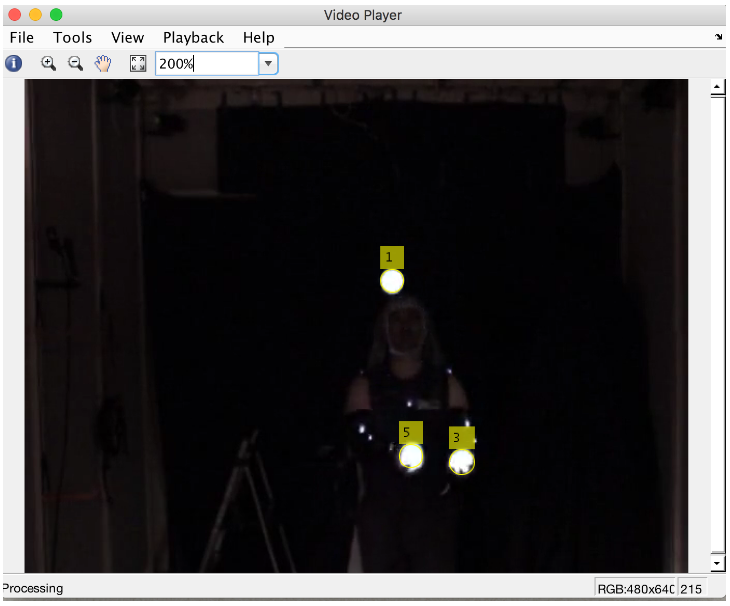


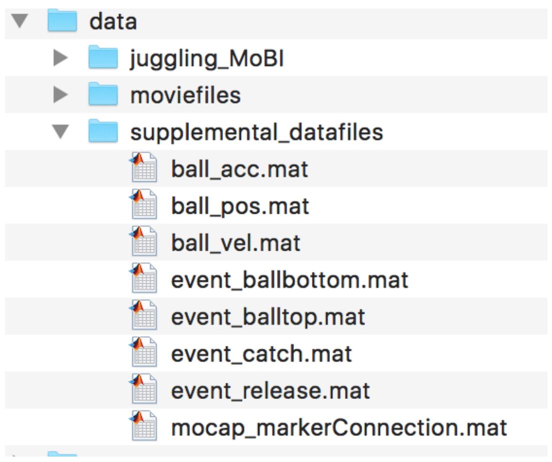
MEMO : Since saving events to MoBILAB data set takes several seconds, I do not save event one-by-one. Instead, I usually stack about 10 ~ 20 events in the queue, and save them at once.

(III-iv) Detecting Events Timing by Custom Programs

To detect timing of events that occur during juggling, we made custom Matlab scripts. The aims of the scripts are to 1) track three balls in movie frames and get time series of balls' position, velocity, and acceleration, 2) automatically detect timing of catching and throwing, and timing when a ball reaches hight/lowest points.

Since computational processes of ball tracking and timing detection are specific to data analysis for juggling, I do not explain in detail about those processes in this tutorial. Briefly speaking, the scripts for ball tracking utilizes functions in Matlab's Image Processing Toolbox. The juggling related events are automatically detected by searching characteristic points in balls trajectory data corresponding to the events. In those scripts, MoBILAB data set is loaded and the video stream is referred to assign time stamp to each of the video frame images.





- data
  - juggling_MoBI
  - moviefiles
  - supplemental_datafiles
    - ball_acc.mat
    - ball_pos.mat
    - ball_vel.mat
    - event_ballbottom.mat
    - event_balltop.mat
    - event_catch.mat
    - event_release.mat
    - mocap_markerConnection.mat

Instead of providing ball tracking programs, Matlab data files containing time sequences of balls position ("ball_pos.mat"), velocity ("ball_vel.mat"), and acceleration ("ball_acc.mat") are provided as supplemental data files. Also, Matlab data files containing events information for catching ("event_catch.mat"), throwing ("event_release.mat"), balls apex ("event_balltop.mat"), and balls bottom ("event_ballbottom.mat") events are provided.

Note that the balls are only tracked during jugging trials. The position, velocity, and acceleration are set to 0 during inter-trial periods.

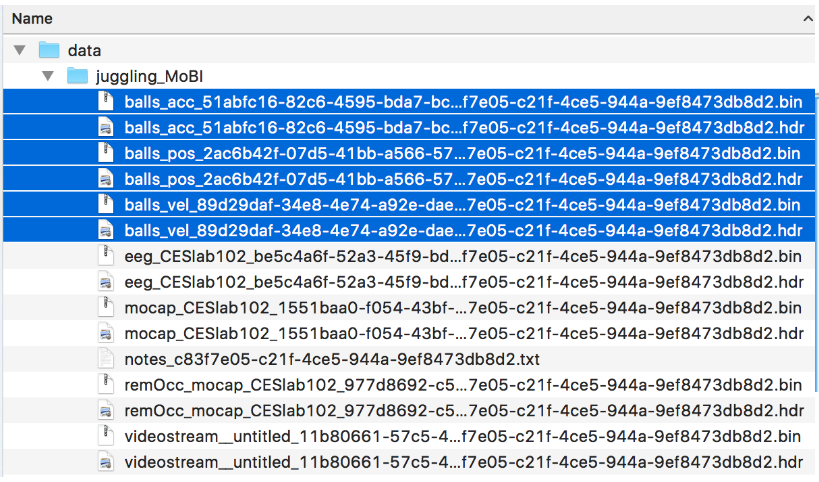
(III-v) Adding New Streams to MoBILAB Data Set

If you want to browse signal created outside MoBILAB toolbox in synchronization with MoBILAB data streams, it is better to add the signal as a new data stream to MoBILAB data set. To do that, you have to create a set of binary data (.bin) and header information (.hdr) files for the new data stream and save them to MoBILAB data set folder.

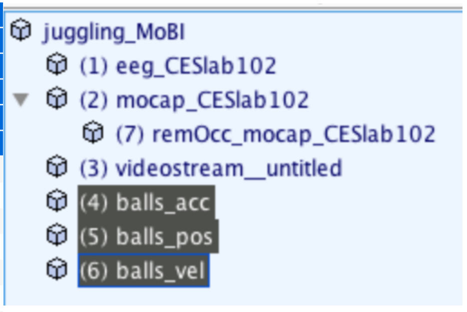
Matlab scripts “script_addStream_BallsPosition.m”, “script_addStream_BallsVelocity.m”, and “script_add_Stream_BallsAcceleration.m” creates set of binary and header files for the data streams of balls’ position, velocity, and acceleration, respectively.

Please run the scripts, after modifying the code at line 29 where path to the tutorial data directory is defined.

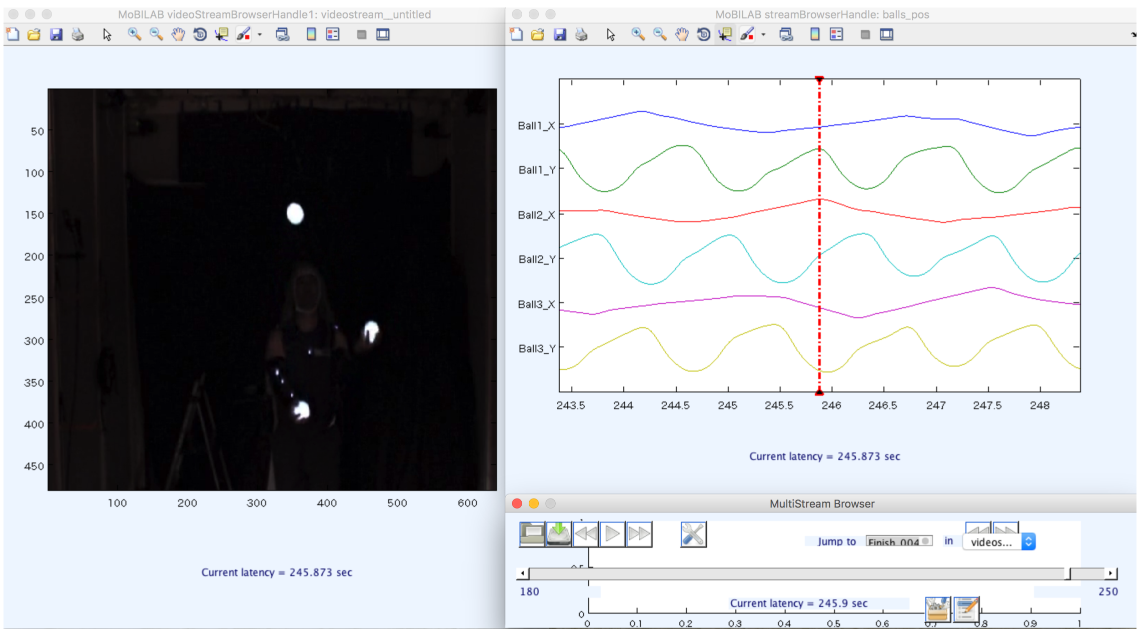
After successful run, you will find new .bin and .hdr files in MoBILAB data set folder. And the new data streams will be included when you re-load MoBILAB data set folder.



```
Name
▼ data
  ▼ juggling_MoBI
      balls_acc_51abfc16-82c6-4595-bda7-bc...f7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      balls_acc_51abfc16-82c6-4595-bda7-bc...f7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      balls_pos_2ac6b42f-07d5-41bb-a566-57...7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      balls_pos_2ac6b42f-07d5-41bb-a566-57...7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      balls_vel_89d29daf-34e8-4e74-a92e-dae...7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      balls_vel_89d29daf-34e8-4e74-a92e-dae...7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      eeg_CESlab102_be5c4a6f-52a3-45f9-bd...f7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      eeg_CESlab102_be5c4a6f-52a3-45f9-bd...f7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      mocap_CESlab102_1551baa0-f054-43bf-...7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      mocap_CESlab102_1551baa0-f054-43bf-...7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      notes_c83f7e05-c21f-4ce5-944a-9ef8473db8d2.txt
      remOcc_mocap_CESlab102_977d8692-c5...7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      remOcc_mocap_CESlab102_977d8692-c5...7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
      videostream__untitled_11b80661-57c5-4...f7e05-c21f-4ce5-944a-9ef8473db8d2.bin
      videostream__untitled_11b80661-57c5-4...f7e05-c21f-4ce5-944a-9ef8473db8d2.hdr
```



```
juggling_MoBI
  (1) eeg_CESlab102
▼ (2) mocap_CESlab102
      (7) remOcc_mocap_CESlab102
  (3) videostream__untitled
  (4) balls_acc
  (5) balls_pos
  (6) balls_vel
```


MoBILAB videoStreamBrowserHandle1: videostream__untitled
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
Current latency = 245.873 sec
MoBILAB streamBrowserHandle: balls_pos
Ball1_X
Ball1_Y
Ball2_X
Ball2_Y
Ball3_X
Ball3_Y
243.5
244
244.5
245
245.5
246
246.5
247
247.5
248
Current latency = 245.873 sec
MultiStream Browser
Jump to
Finish 004
in
videos...
180
250
Current latency = 245.9 sec

(III-vi) Inserting Events to MoBILAB Data Set

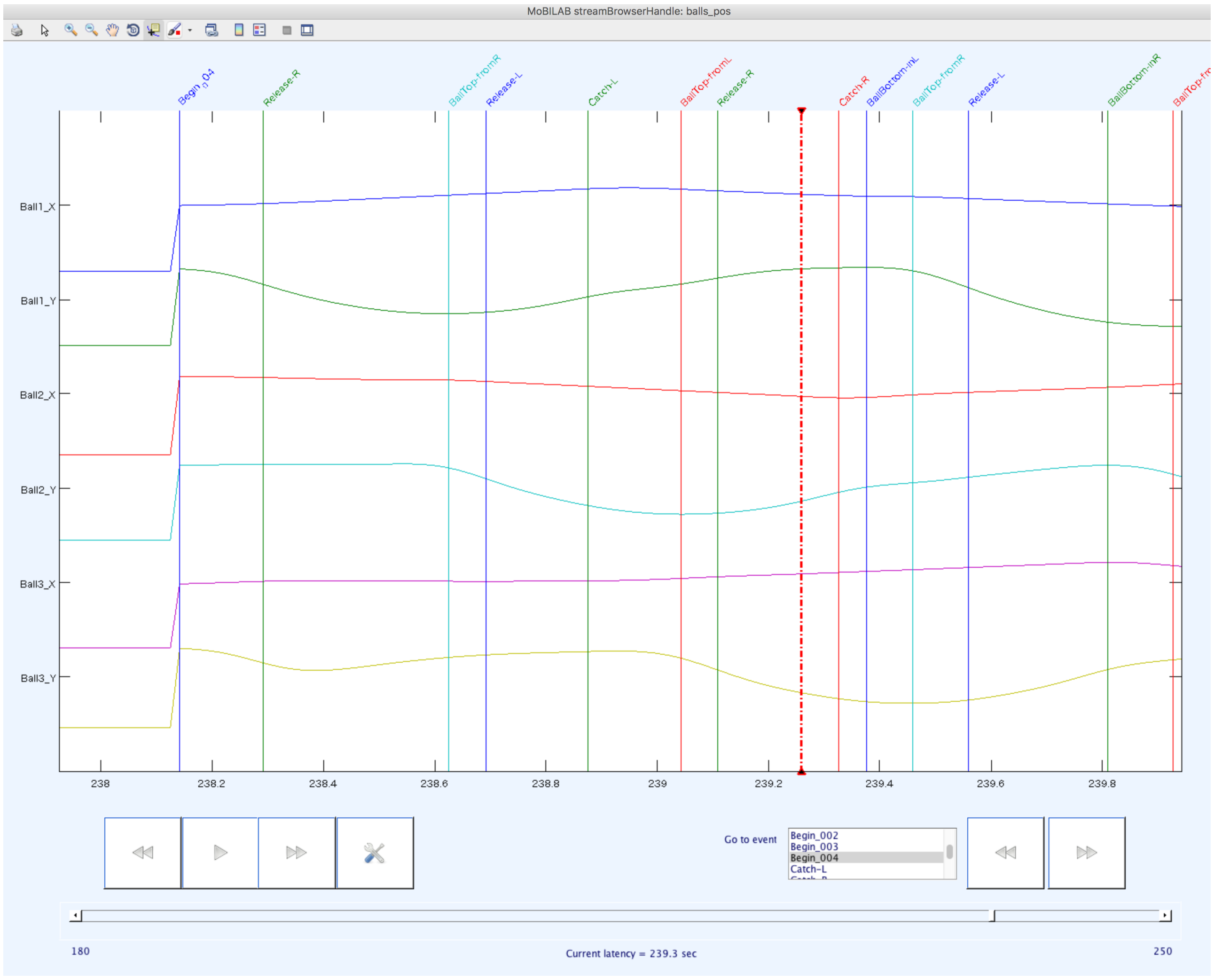
MoBILAB toolbox supports inserting events not only from GUI window but also from program script.

Matlab script “script_insertEvent.m” inserts events to all data streams in MoBILAB data set. The types of events inserted are catching, throwing, balls highest peaks and balls lowest peaks.

Please run the scripts, after modifying the code at line 36 where path to the tutorial data directory is defined.

You will see messages in command window tell you that header files containing events information are saved due to changes in event class object inherent in data streams.

Please load MoBILAB data set folder again, and confirm that the new events appear in data stream browser.



Current latency = 239.3 sec

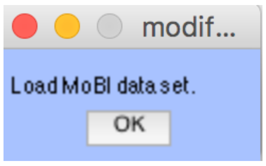
(III-vii) Checking and Modifying Events

At step-iv, catching and throwing events were automatically detected by analyzing balls trajectory data. Here we will browse video data stream and check whether catching and throwing really happened at automatically detected timings. If the timings are incorrect then we would like to modify temporal latencies of those events.

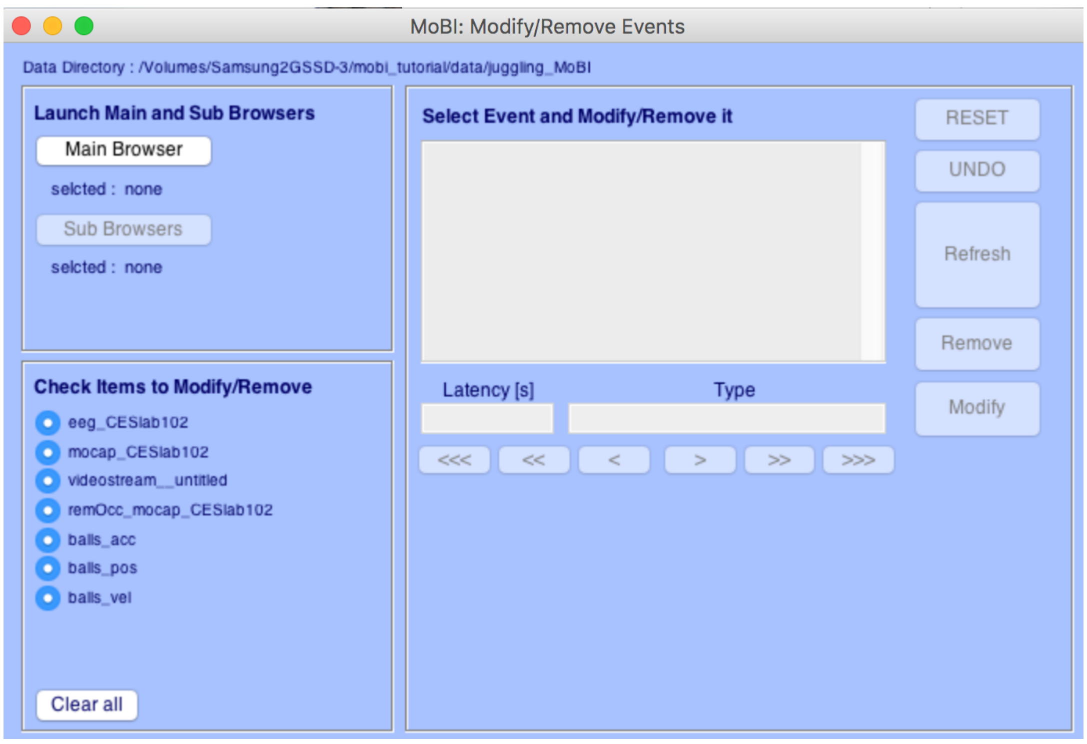
Since MoBILAB toolbox does not have a function to modify events information with browsing data stream, I made another GUI for that purpose. The GUI program is based on MoBILAB classes and utilized the browser classes with slight modification.

The GUI can be launched by running “mobi_gui_modifyEvent.m” function. Basic steps to modify events information by the GUI is the following.

1. Type and enter “***mobi_gui_modifyEvent()***” in Matlab command line.
2. Click ***OK*** button in a pop-up window asking for loading MoBILAB data set.

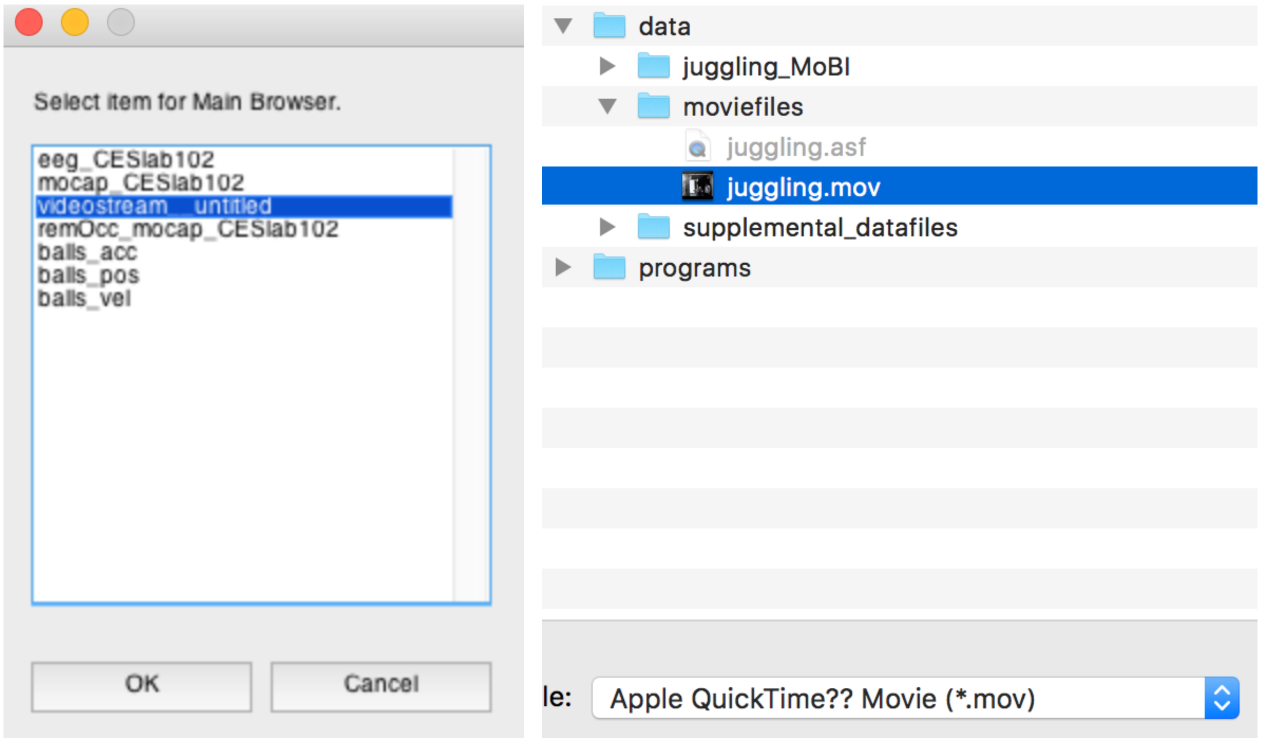


3. Select MoBILAB data set folder, then the GUI window pops up.


MoBI: Modify/Remove Events

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/data/juggling_MoBI

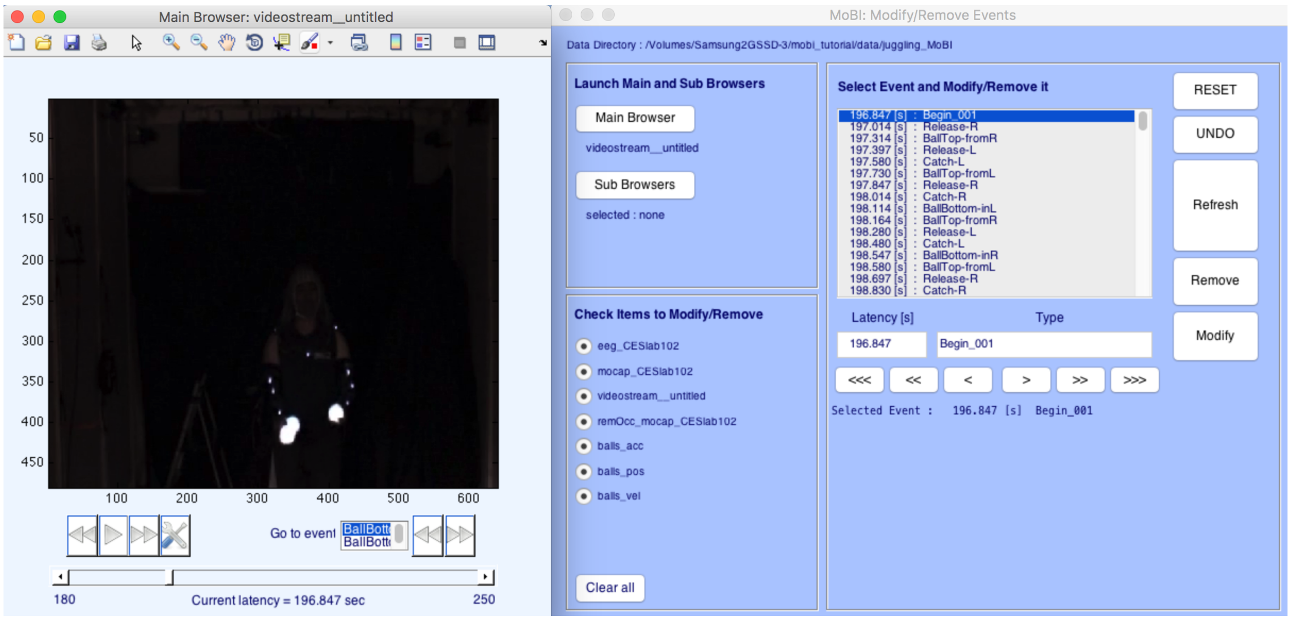
4. Click ***Main Browser*** button in the upper-left panel, and select video data stream in a list selection dialog box.



Select item for Main Browser.

```
eeg_CESlab102
mocap_CESlab102
videostream__untitled
remOcc_mocap_CESlab102
balls_acc
balls_pos
balls_vel
```

5. Video data stream browser will appear and events information (latency and label) is displayed to the list box of the right panel in the GUI.



Main Browser: videostream__untitled

Current latency = 196.847 sec

MoBI: Modify/Remove Events

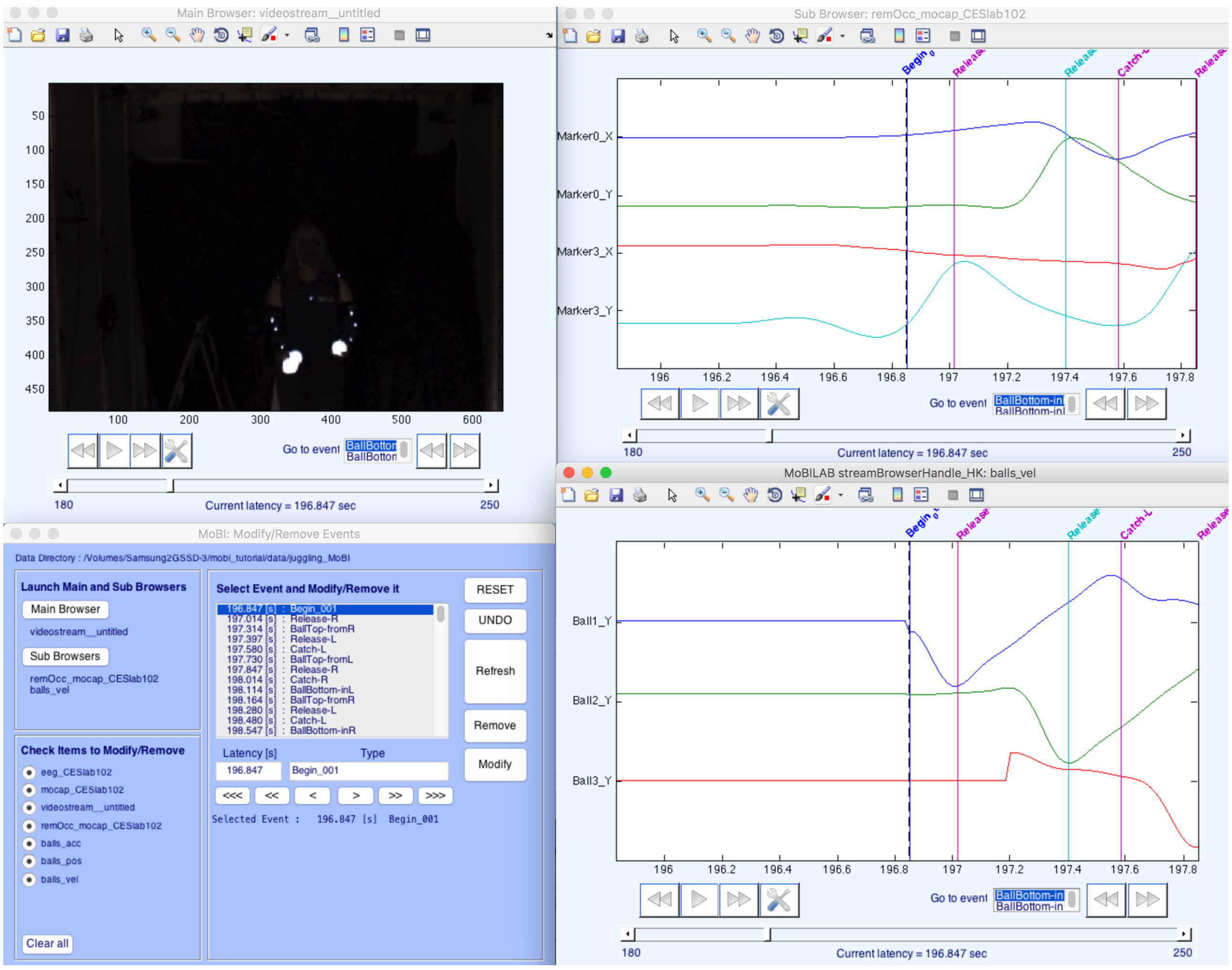
Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/data/juggling_MoBI

```
196.847 [s]  :  Begin_001
197.014 [s]  :  Release-R
197.314 [s]  :  BallTop-fromR
197.397 [s]  :  Release-L
197.580 [s]  :  Catch-L
197.730 [s]  :  BallTop-fromL
197.847 [s]  :  Release-R
198.014 [s]  :  Catch-R
198.114 [s]  :  BallBottom-inL
198.164 [s]  :  BallTop-fromR
198.280 [s]  :  Release-L
198.480 [s]  :  Catch-L
198.547 [s]  :  BallBottom-inR
198.580 [s]  :  BallTop-fromL
198.697 [s]  :  Release-R
198.830 [s]  :  Catch-R
```

Latency [s]: 196.847
Type: Begin_001

Selected Event :   196.847 [s] Begin_001

6. If you want to browse other data streams, click ***Sub Browsers*** button and select data stream you want to browse.


Main Browser: videostream__untitled
Sub Browser: remOcc_mocap_CESlab102
MoBILAB streamBrowserHandle_HK: balls_vel
MoBI: Modify/Remove Events
Marker0_X
Marker0_Y
Marker3_X
Marker3_Y
Ball1_Y
Ball2_Y
Ball3_Y
Begin_0
Release
Catch-L
Go to event
BallBottom-in
180
250
Launch Main and Sub Browsers
Main Browser
Sub Browsers
Check Items to Modify/Remove
Clear all
Select Event and Modify/Remove it
RESET
UNDO
Refresh
Remove
Modify
Latency [s]
Type
<<<
<<
<
>
>>
>>>

Current latency = 196.847 sec

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/data/juggling_MoBI

videostream__untitled

remOcc_mocap_CESlab102
balls_vel

● eeg_CESlab102
● mocap_CESlab102
● videostream__untitled
● remOcc_mocap_CESlab102
● balls_acc
● balls_pos
● balls_vel

```
196.847 [s] : Begin_001
197.014 [s] : Release-R
197.314 [s] : BallTop-fromR
197.397 [s] : Release-L
197.580 [s] : Catch-L
197.730 [s] : BallTop-fromL
197.847 [s] : Release-R
198.014 [s] : Catch-R
198.114 [s] : BallBottom-inL
198.164 [s] : BallTop-fromR
198.280 [s] : Release-L
198.480 [s] : Catch-L
198.547 [s] : BallBottom-inR
```

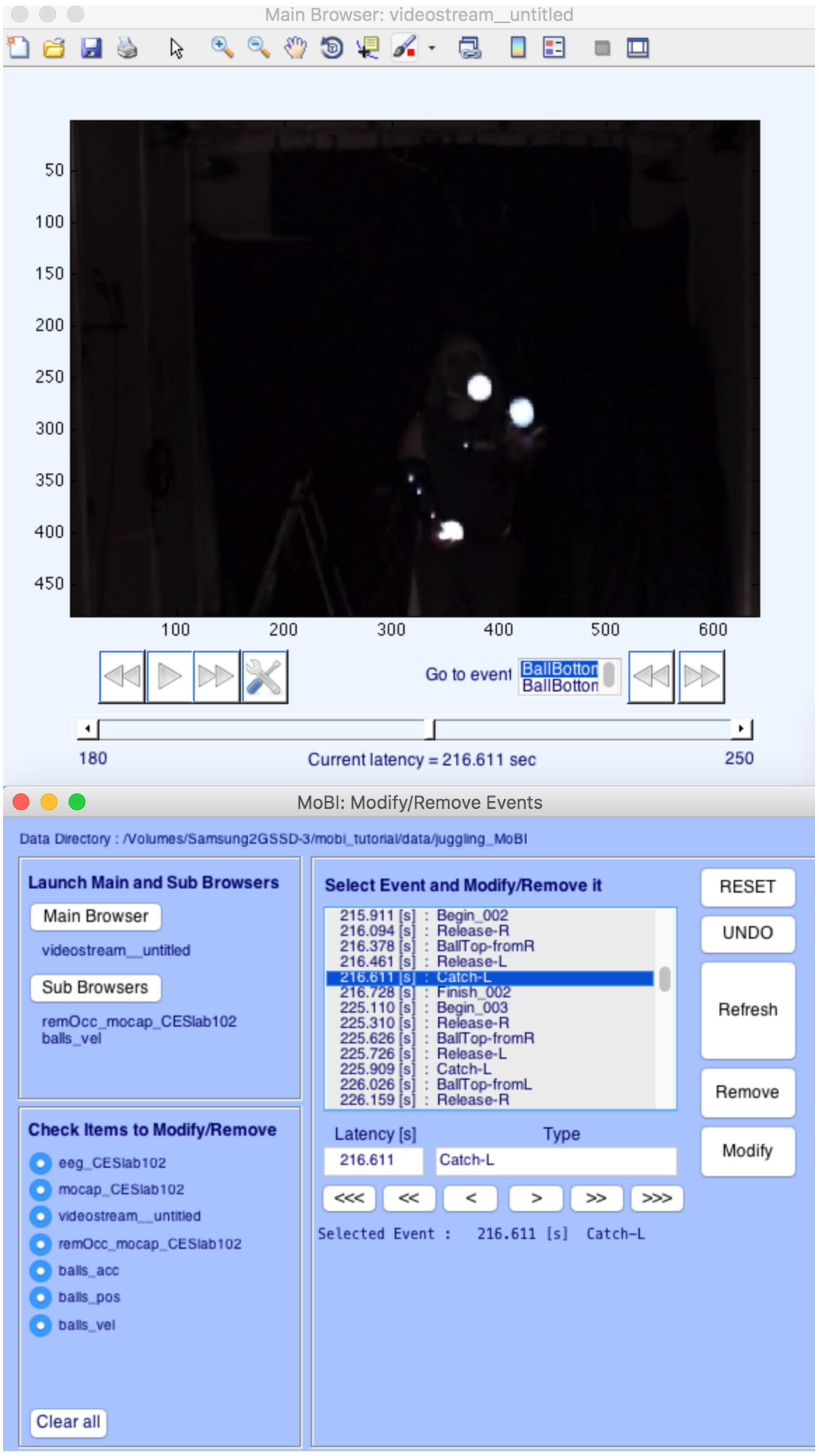
Latency [s]: 196.847 Type: Begin_001

```
Selected Event :   196.847 [s]  Begin_001
```

MEMO : As additional browsers, I selected occlusion removed motion capture data stream (“remOcc_mocap_CESlab102”) and balls velocity data stream (“balls_vel”). The figure above is a snapshot after changing browsers properties as below.

- time window width (both browser) : 2 seconds
- events subset displayed (both browser) :
  - Begin_00X, Finish_00X, Release-R/L, Catch-R/L
- Channels to plot (mocap) : [1 2 10 11]
- Channels to plot (balls_vel) : [2 4 6]

7. Click the event in the list box you want to check, then the current latency of the browsers will set at the latency of the selected event.



Current latency = 216.611 sec

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/data/juggling_MoBI

videostream__untitled

remOcc_mocap_CESlab102
balls_vel

eeg_CESlab102
mocap_CESlab102
videostream__untitled
remOcc_mocap_CESlab102
balls_acc
balls_pos
balls_vel

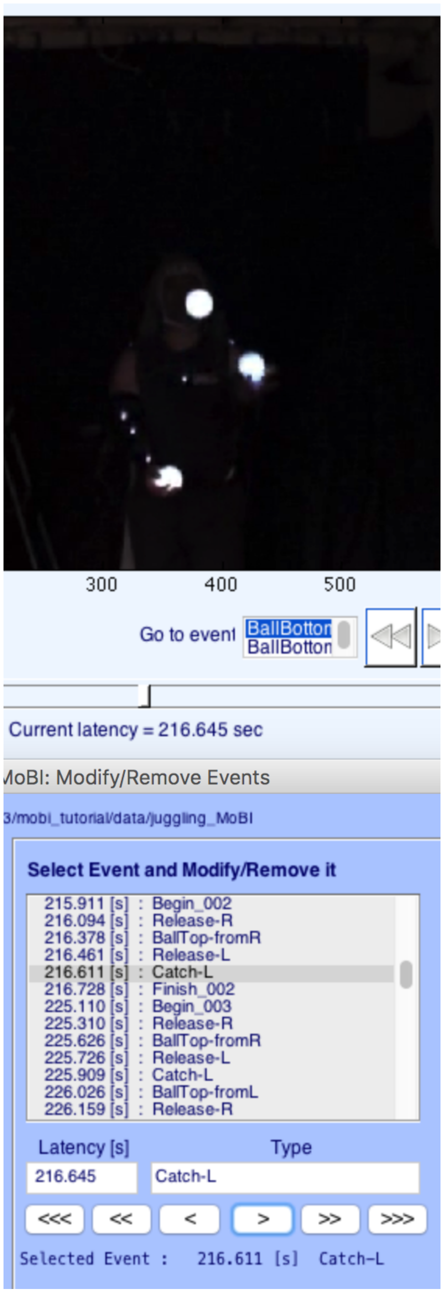
```
215.911 [s] : Begin_002
216.094 [s] : Release-R
216.378 [s] : BallTop-fromR
216.461 [s] : Release-L
216.611 [s] : Catch-L
216.728 [s] : Finish_002
225.110 [s] : Begin_003
225.310 [s] : Release-R
225.626 [s] : BallTop-fromR
225.726 [s] : Release-L
225.909 [s] : Catch-L
226.026 [s] : BallTop-fromL
226.159 [s] : Release-R
```

Latency [s]	Type
216.611	Catch-L

Selected Event : 216.611 [s] Catch-L

MEMO : The figure above is a snapshot after clicking “Catch-L” event at latency of 216.511 [s]. You can see that a ball is still in the air, which means that automatically detected timing was slightly earlier than actual catch timing.

8. To modify latency of the event, you can directly type desired latency in the box named ***Latency [s]***, or forward/rewind in time by clicking **>, >>, >>>, < ,<< ,<<<** buttons. After correcting the latency, click ***Modify*** button to save modified event data in MoBILAB data set.



Current latency = 216.645 sec

MoBI: Modify/Remove Events

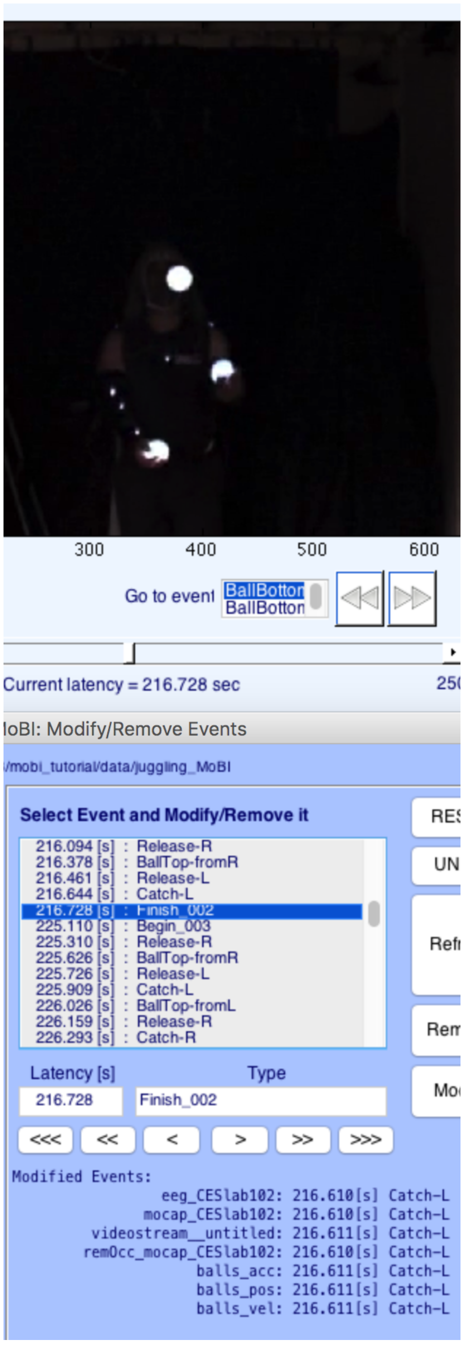
3/mobi_tutorial/data/juggling_MoBI

Select Event and Modify/Remove it

```
215.911 [s] : Begin_002
216.094 [s] : Release-R
216.378 [s] : BallTop-fromR
216.461 [s] : Release-L
216.611 [s] : Catch-L
216.728 [s] : Finish_002
225.110 [s] : Begin_003
225.310 [s] : Release-R
225.626 [s] : BallTop-fromR
225.726 [s] : Release-L
225.909 [s] : Catch-L
226.026 [s] : BallTop-fromL
226.159 [s] : Release-R
```

Latency [s]: 216.645 Type: Catch-L

```
Selected Event :   216.611 [s]  Catch-L
```



Current latency = 216.728 sec

MoBI: Modify/Remove Events

/mobi_tutorial/data/juggling_MoBI

Select Event and Modify/Remove it

```
216.094 [s] : Release-R
216.378 [s] : BallTop-fromR
216.461 [s] : Release-L
216.644 [s] : Catch-L
216.728 [s] : Finish_002
225.110 [s] : Begin_003
225.310 [s] : Release-R
225.626 [s] : BallTop-fromR
225.726 [s] : Release-L
225.909 [s] : Catch-L
226.026 [s] : BallTop-fromL
226.159 [s] : Release-R
226.293 [s] : Catch-R
```

Latency [s]: 216.728 Type: Finish_002

```
Modified Events:
          eeg_CESlab102: 216.610[s] Catch-L
        mocap_CESlab102: 216.610[s] Catch-L
    videostream__untitled: 216.611[s] Catch-L
    remOcc_mocap_CESlab102: 216.610[s] Catch-L
               balls_acc: 216.611[s] Catch-L
               balls_pos: 216.611[s] Catch-L
               balls_vel: 216.611[s] Catch-L
```

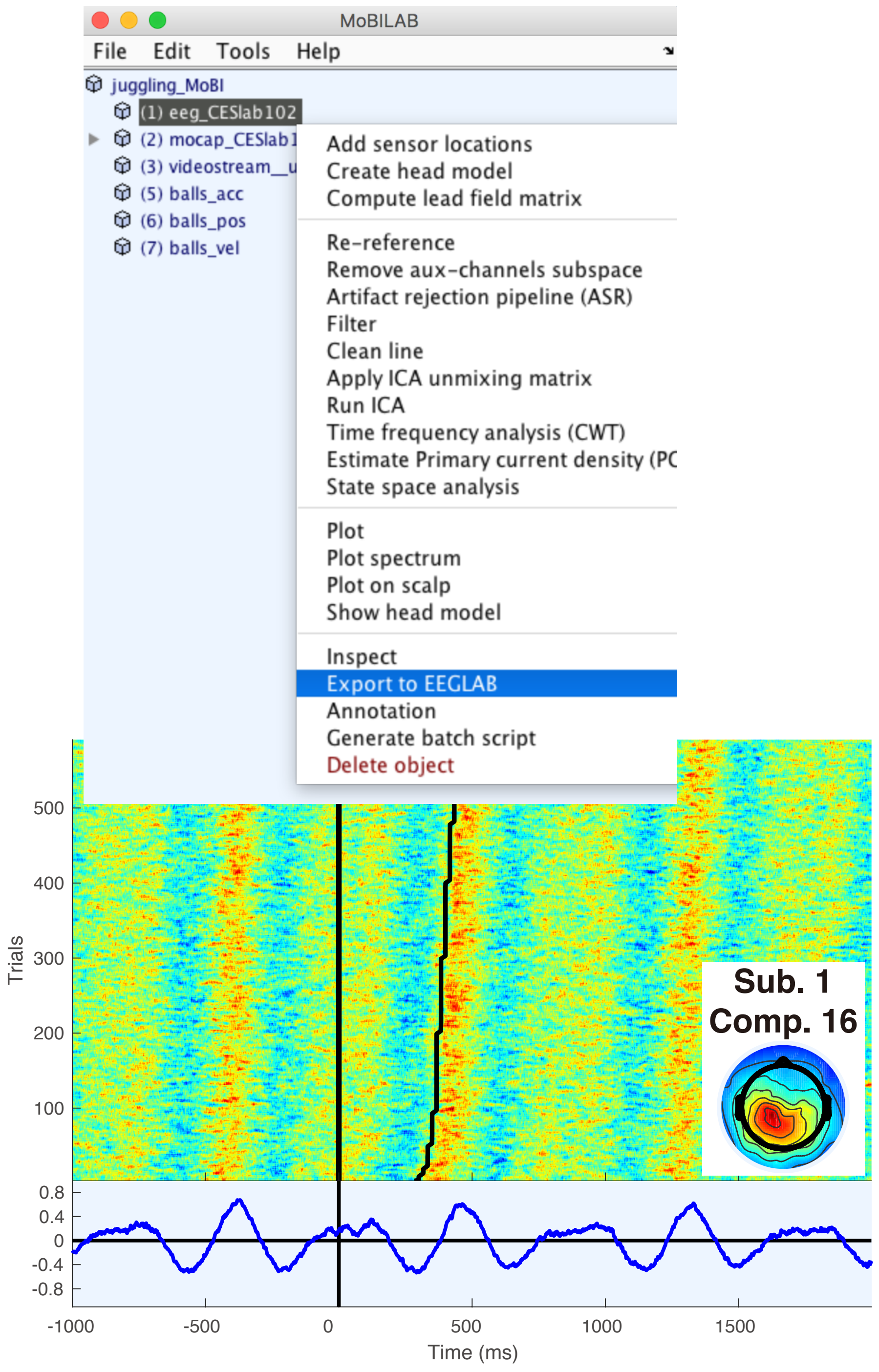
MEMO : The left figure above is a snapshot after advancing two frames by clicking > button twice. The right figure is a snapshot after clicking ***Modify*** button then. You can see that the latency of corresponding "Catch-L" event has been changed from 216.611 to 216.644 [s].

9. If you want to change label of events, click an event you want to modify in the list box and type desired label in the box named ***Type***, then click ***Modify*** button. You will see the message at bottom part of the panel that the events has been modified.

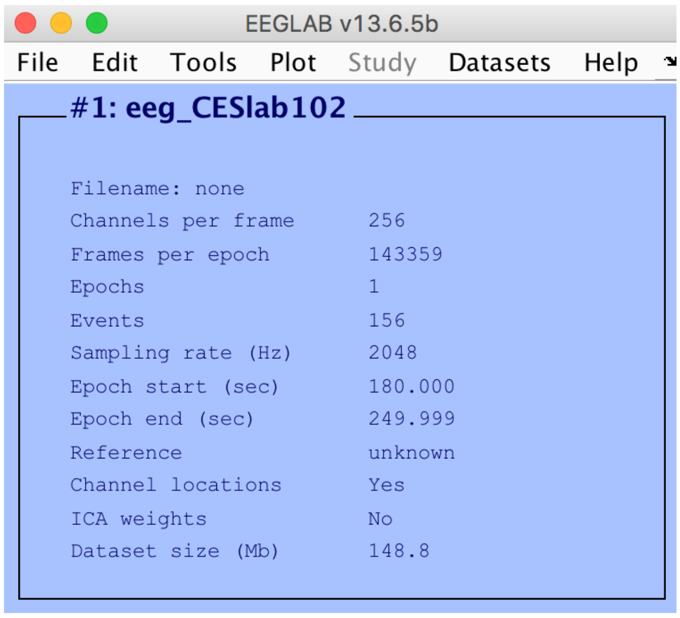
(III-viii) Exporting EEG Data from MoBILAB to EEGLAB

After checking and modifying events information, we are going to export EEG data stream to EEGLAB .set format by the following steps.

1. Launch MoBILAB GUI and load MoBILAB data set folder.
2. Right click eeg data stream and select ***Export to EEGLAB.***



3. EEG data will be automatically imported to EEGLAB toolbox and you can start analyzing EEG signals by EEGLAB GUI.



```
Filename: none
Channels per frame      256
Frames per epoch        143359
Epochs                  1
Events                  156
Sampling rate (Hz)      2048
Epoch start (sec)       180.000
Epoch end (sec)         249.999
Reference               unknown
Channel locations       Yes
ICA weights             No
Dataset size (Mb)       148.8
```

(III-ix) Preprocessing EEG Data

There is nothing special about preprocessing of EEG signals for juggling data. So I am not going to explain here. Please refer EEGLAB Wiki pages.

(III-x) Event Related Analysis

Now the last step in this tutorial is to do event related analysis of EEG data. In EEGLAB GUI, you can chose one type of events of interest and extract epochs time locked to the events. Then you can compute ERP, ERSP, ITC, and so on. Again, this step is also typical process in the analysis using EEGLAB. So I will leave detail explanation to EEGLAB Wiki pages.

I will finish this tutorial by showing an example of ERP acquired by juggling data analysis. The figure below is an ERP of one independent component from left sensory-motor area. Each epochs are time-locked to left hand catching and sorted by the latency of right hand catching. We can see that activity of this independent component is time-locked to right hand catching.

2018. Oct. 2nd

Appendix:
MoBILAB GUIs for Event Marking and Modification

For juggling data analysis, I made two Matlab GUI programs that can handle events in MoBILAB data set with browsing data streams. One GUI is for inserting new events, and the other is for modifying label and/or latency of existing events. The programs basically utilizes classes in MoBILAB toolbox. I also made two classes for browsing by slightly modifying codes from original MoBILAB toolbox.

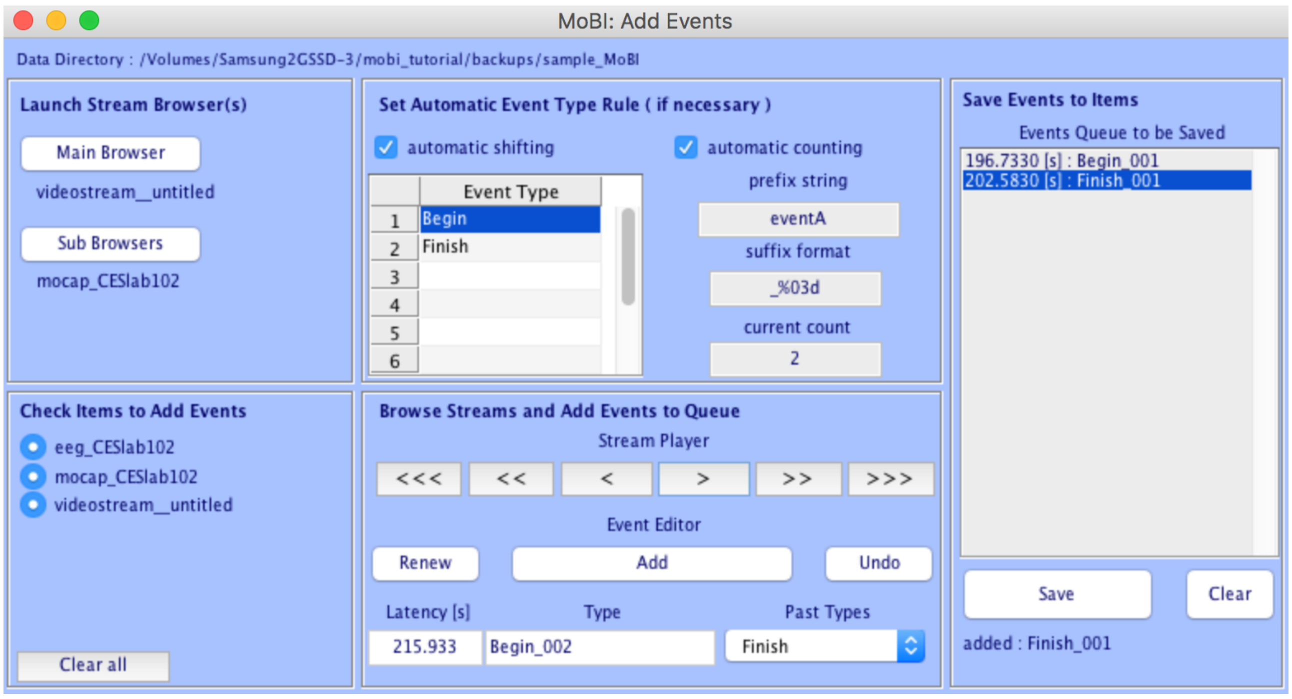
Here are instructions for using the two GUIs.

(Appendix-i) Event Marking GUI

GUI for event marking can be launched by running Matlab “mobi_gui_addEvent.m” function. This GUI can be used when you want to detect some events by browsing data streams and insert events to MoBILAB data set.

Features

- Multiple data stream browsing
- New event insertion
- Automatic event name labelling
    - automatic alteration between registered event labels
    - automatic count-up for event label index



Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI

videostream__untitled

mocap_CESlab102

✓ automatic shifting

✓ automatic counting

	Event Type
1	Begin
2	Finish
3	
4	
5	
6	

prefix string: eventA

suffix format: _%03d

current count: 2

196.7330 [s] : Begin_001
202.5830 [s] : Finish_001

● eeg_CESlab102
● mocap_CESlab102
● videostream__untitled

Latency [s]	Type	Past Types
215.933	Begin_002	Finish

added : Finish_001

Overview of Basic Steps

1) Start GUI program
2) Load MoBILAB data set
3) Launch data stream browsers
4) Select data streams to which new events are saved
5) Detect event by browsing data streams
6) Set event name
7) Stack event information to events queue
8) Insert events in the queue to data streams after stacking one or more events

How To

1) Start GUI program

GUI window will be launched by running “mobi_gui_addEvent” function from Matlab command line.

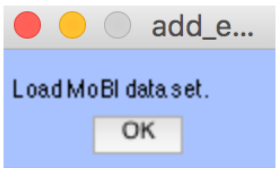
```
>> objMoBI = add_event_gui();
```

The function returns a “dataSourceMoBI” class object. The object can be given as input argument when you run the function again. Then MoBILAB data set loading (step 2) will be skipped.

```
>> objMoBI = add_event_gui(objMoBI);
```

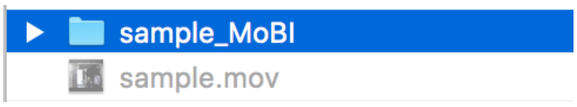
2) Load MoBILAB data set

If you run “mobi_gui_addEvent” function without input argument, you will be asked to select MoBILAB data set folder. Click ***OK*** button in the pop-up window.



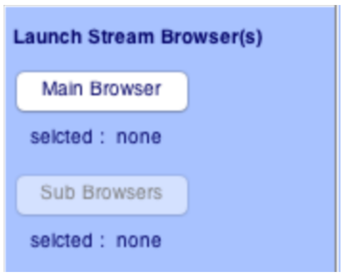
Load MoBI data set.

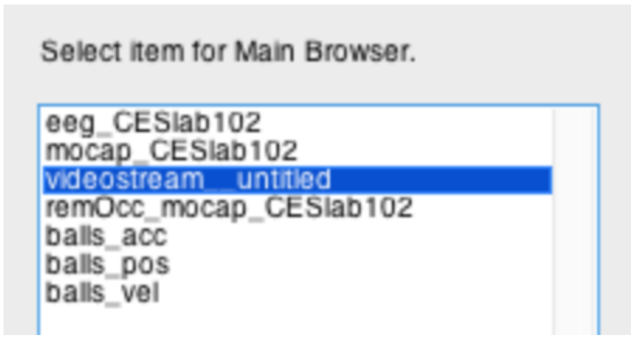
At the folder selection dialog box appearing next, select MoBILAB data set folder you want to load.



3) Launch data stream browsers

You can launch one or more data stream browsers for detecting events. Click ***Main Browser*** button in the upper-left panel. Then select the name of data stream you want to browse.

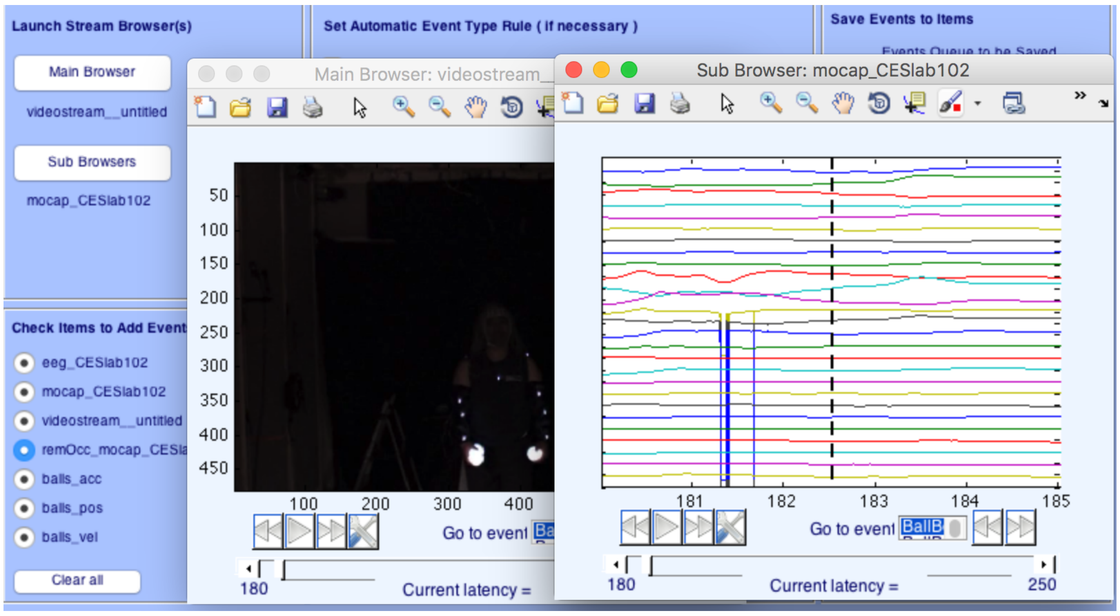




Select item for Main Browser.

```
eeg_CESlab102
mocap_CESlab102
videostream__untitled
remOcc_mocap_CESlab102
balls_acc
balls_pos
balls_vel
```

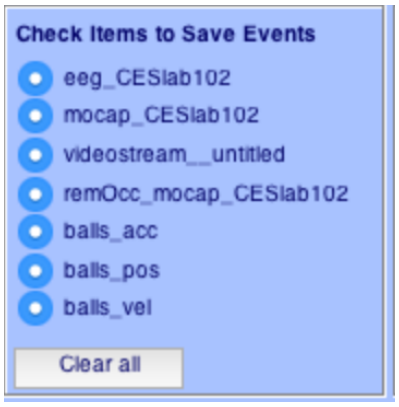
You can launch additional browsers by clicking ***Sub Browsers*** button.



NOTE : Which data stream should be selected for Main Browser?
The data stream most related to the events that you are looking for is recommended to be browsed with Main Browser. In the GUI, there are buttons changing current latency of browsers. Timestamp of the data stream selected for Main Browser is referred when those buttons are clicked. For example, when >> button in the GUI is clicked, current time in each browser will be shifted forward by the time corresponds to 10 sampling points of the "Main" data stream.

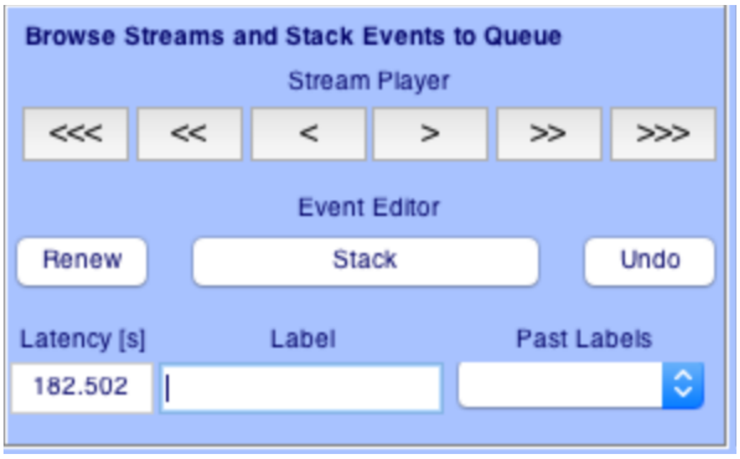
4) Select data streams to which new events are saved

By checking on/off the radio buttons in bottom-left panel, you can select data streams to which new events will be saved. Initially, all data streams are selected.



5) Detect event by browsing data streams

Current latency in browsers can be changed by clicking buttons under ***Stream Player*** text in bottom-mid panel.

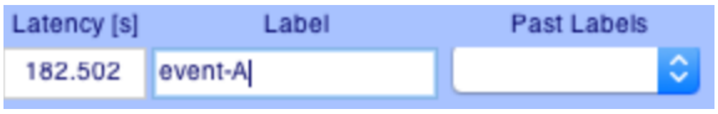


When you click < **,** << **,** or <<< button, current latency in all browsers will be rewinded by the time corresponding to 1, 10, 100 sampling points of the data stream selected for Main Browser. When you click > **,** >> **,** >>> button, current latency will be forwarded by the time corresponding to 1, 10, 100 sampling points.

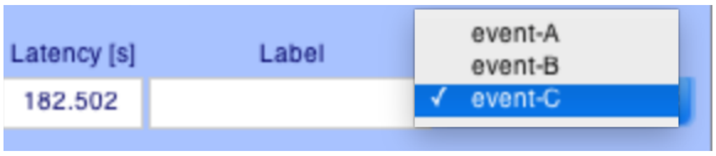
You can also play browsers by clicking playing buttons in each browser, but the current latency will not be synchronised with each other. To set the current latency in all browser to the same latency, you can click ***Renew*** button in bottom-mid panel. Then current latency in each browser will be set to that of Main Browser.

6) Set event name

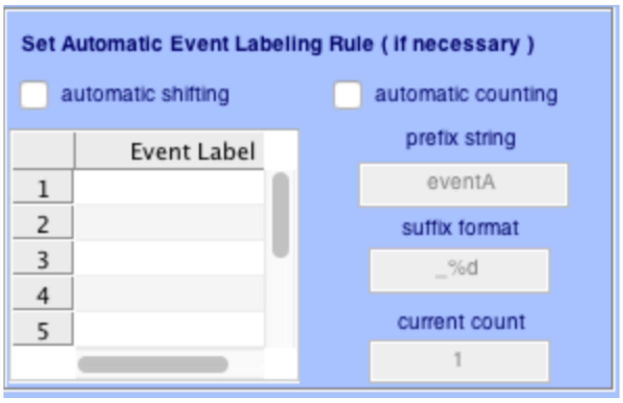
Event name (label) can directly be typed in the box under ***Label*** text in bottom-mid panel.



Also you can chose a label from the pop-up menu listing the ones stacked before.



In addition, event label can be automatically determined by either or both of two rules set in upper-mid panel.


Set Automatic Event Labeling Rule (if necessary)

☐ automatic shifting ☐ automatic counting

	Event Label
1	
2	
3	
4	
5	

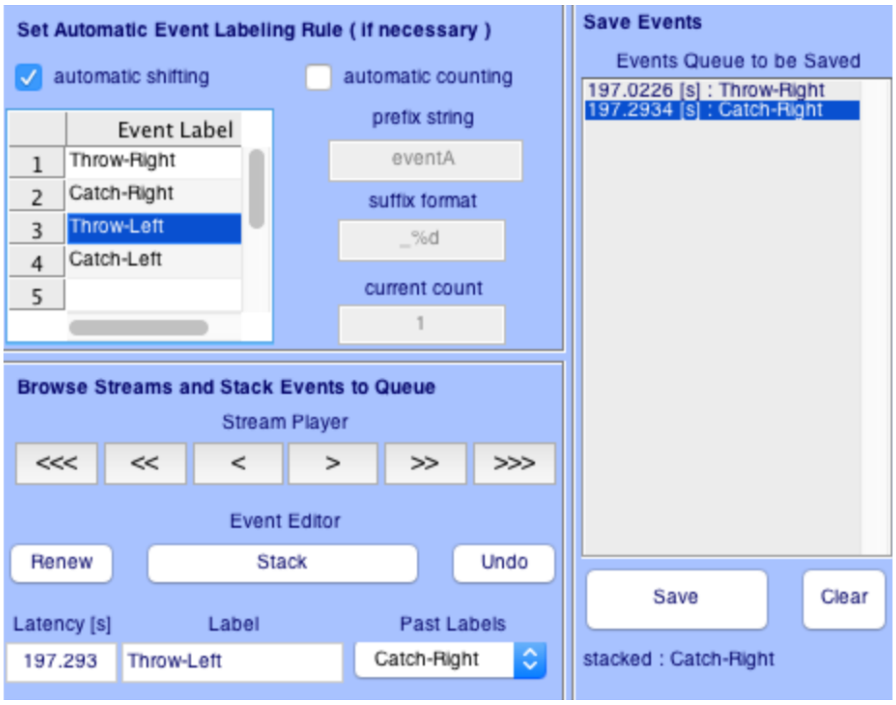
prefix string: eventA

suffix format: _%d

current count: 1

First rule is for automatic event name shifting. This rule is useful If you want to mark several types of event iteratively occurring in the same order. For example, during 3 ball cascade juggling, catching and throwing events occurs periodically in the fixed order as right-hand throw, right-hand catch, left-hand throw, and left-hand catch.

If you want to use this rule, check on ***automatic shifting*** check box and enter event labels to the list box in the order of appearance. Then event label in bottom-mid panel will automatically changed after one event is stacked in the queue.


Set Automatic Event Labeling Rule (if necessary)

☑ automatic shifting ☐ automatic counting

	Event Label
1	Throw-Right
2	Catch-Right
3	Throw-Left
4	Catch-Left
5	

prefix string: eventA

suffix format: _%d

current count: 1

Save Events

Events Queue to be Saved

```
197.0226 [s] : Throw-Right
197.2934 [s] : Catch-Right
```

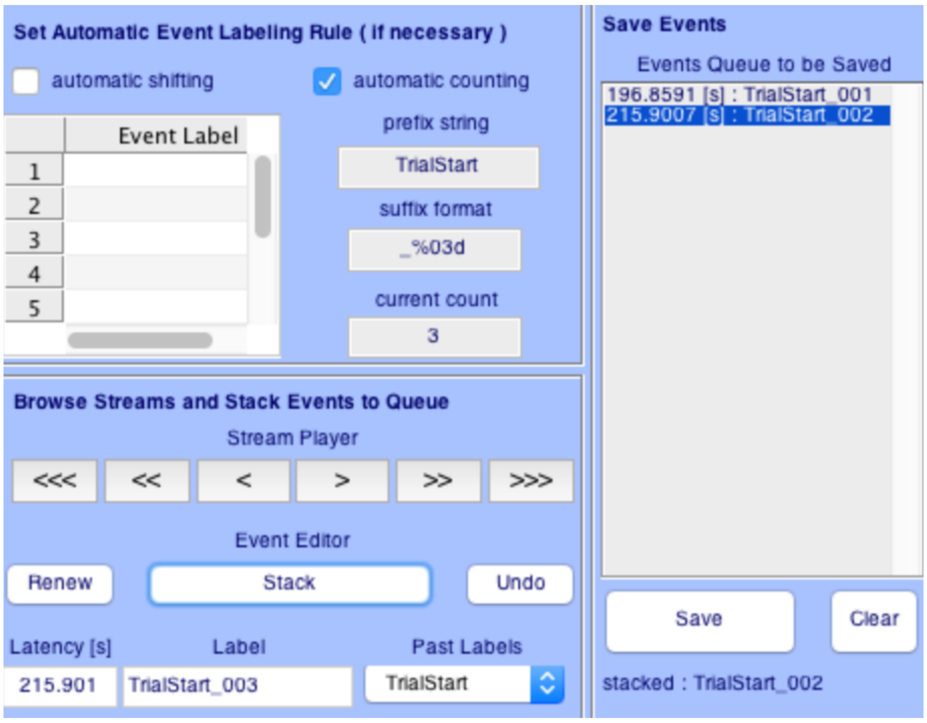
Browse Streams and Stack Events to Queue

Latency [s]: 197.293 Label: Throw-Left Past Labels: Catch-Right

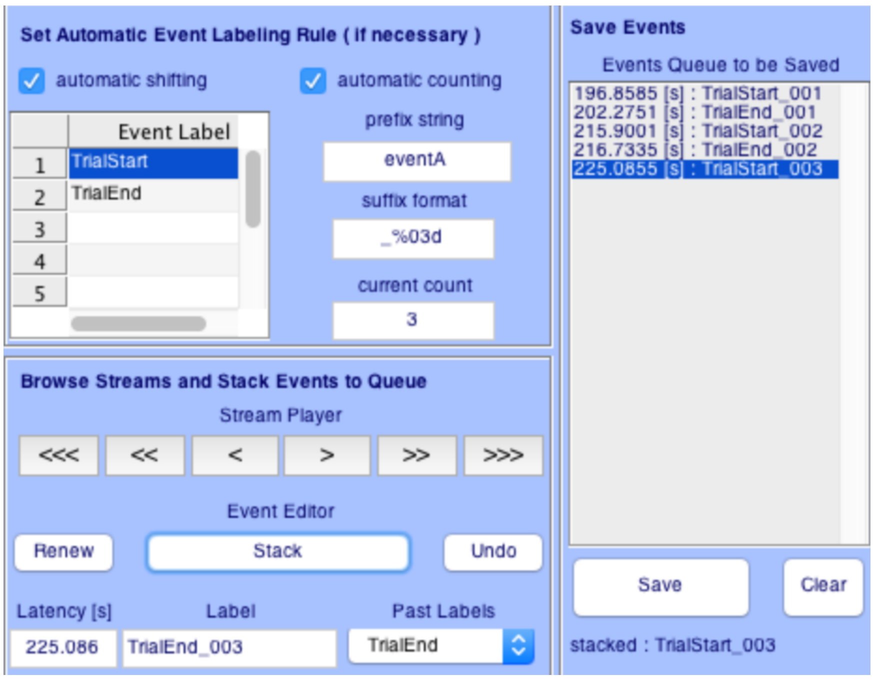
stacked : Catch-Right

The second rule is for automatic event label indexing. This rule is useful if you want to add incremental index to event labels. For example, in the case of juggling data analysis, I use this rule to set the event labels of beginning of trials.

If you want to use this rule, check on ***automatic counting*** check box and enter prefix of label, string format of event indexing, and initial index number to ***prefix string***, ***suffix format***, and ***current count*** boxes respectively. Then suffix of the event label in bottom-mid panel will automatically be counted up after one event is stacked in the queue.

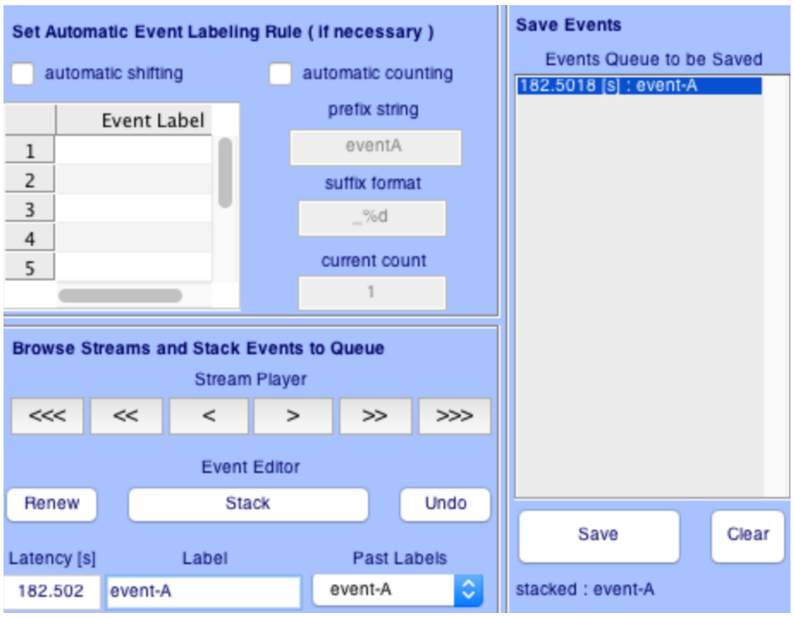


You can also combine the two rules. If you check on both checking boxes, prefix of the event label will be shifted from one label to the next in the list box each time event is stacked. On the other hand, index number of the event will be counted up, after prefix of the event label came back to the first one in the list box. Note that the label typed in ***prefix string*** box will be ignored.



7) Stack event information to events queue

After setting event latency and label, click ***Stack*** button in bottom-mid panel. Then, the event will be stacked to events queue show in right panel. You can also remove the event in the queue by clicking ***Undo*** button.



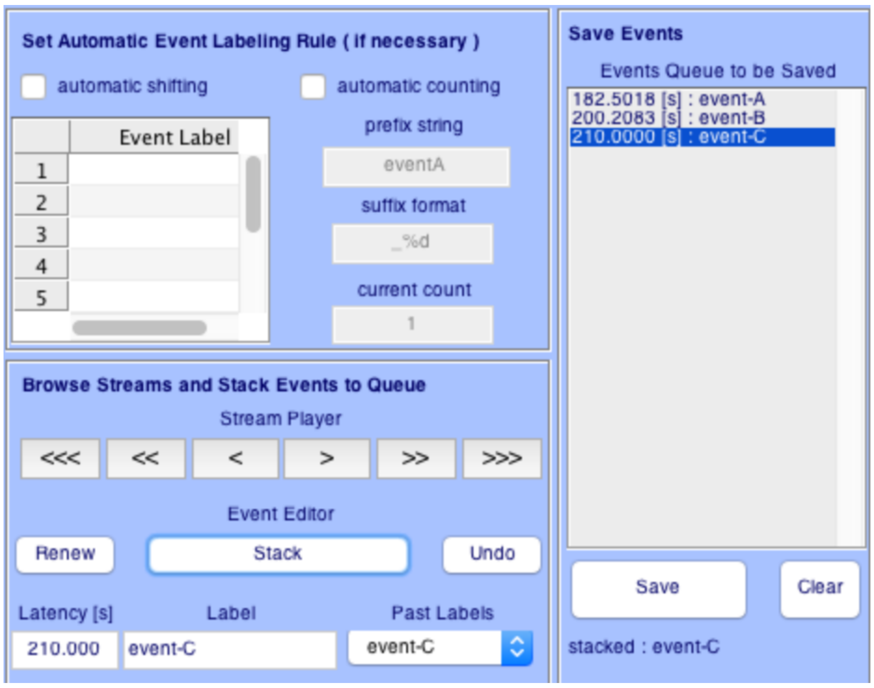
Latency [s]: 182.502
Label: event-A
Past Labels: event-A

182.5018 [s] : event-A

stacked : event-A

8) Insert events in the queue to data streams after stacking one or more events

You can stack one or more events to the events queue by iterating steps 5) to 7).



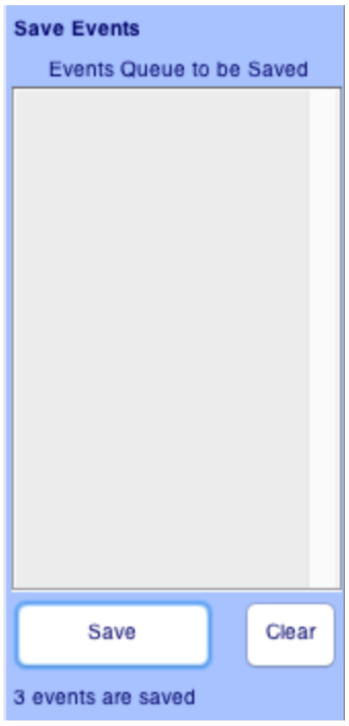
Latency [s]: 210.000
Label: event-C
Past Labels: event-C

182.5018 [s] : event-A
200.2083 [s] : event-B
210.0000 [s] : event-C

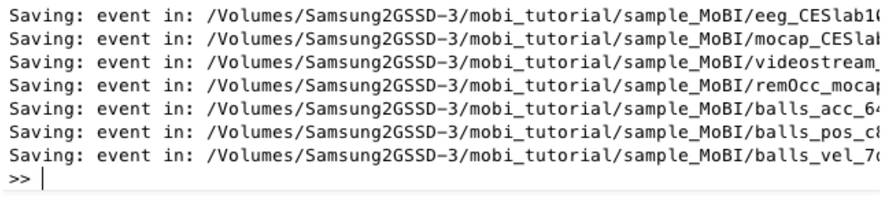
stacked : event-C

Note that the events are not saved to the data streams just by stacking events. To save the events, click ***Save*** button in right panel. You will see messages in

Matlab command line and lower part of right panel noticing that events are saved to the data streams. Also events in the queue will be cleared.



3 events are saved



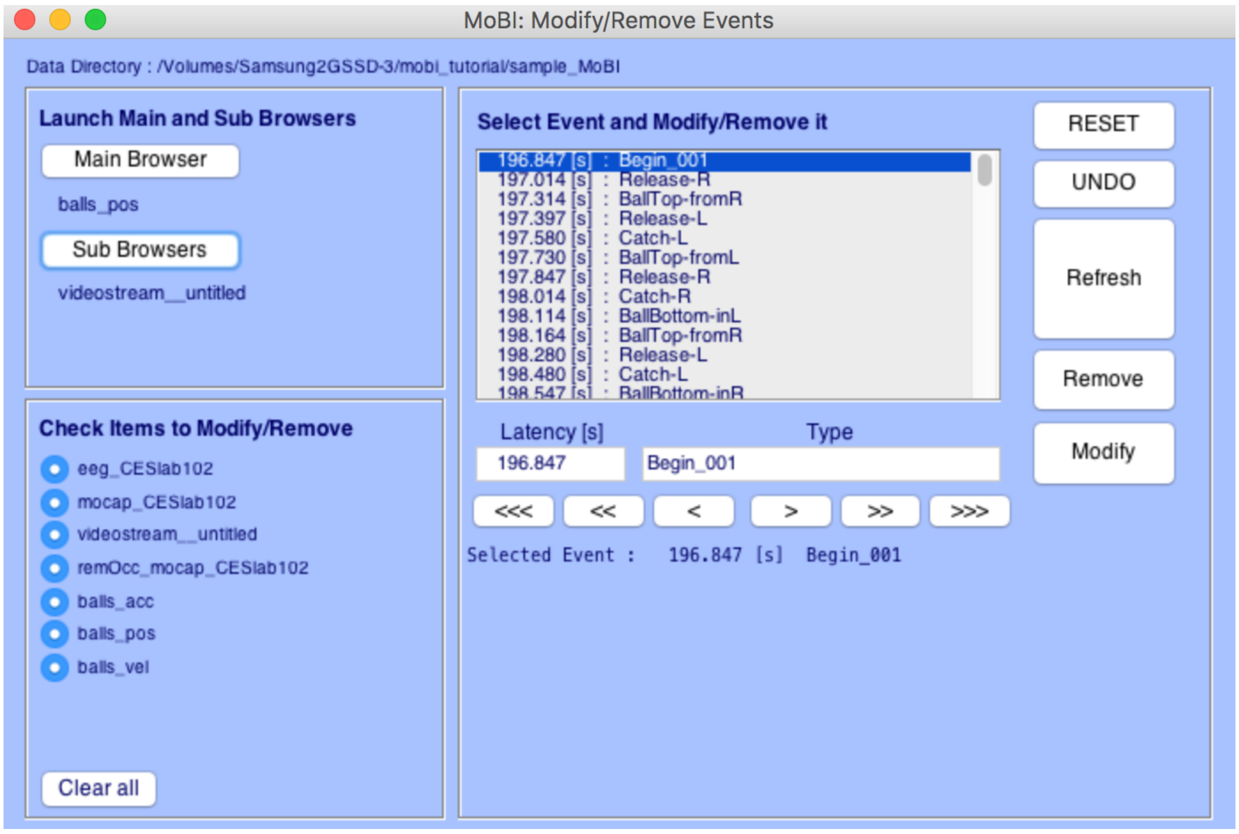
```
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/eeg_CESlab1(
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/mocap_CESlal
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/videostream_
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/remOcc_mocap
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/balls_acc_6
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/balls_pos_c
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI/balls_vel_7
>> |
```

(Appendix-ii) Event modifying GUI

GUI for event modification can be launched by running Matlab "mobi_gui_modifyEvent.m" function. This GUI can be used when you want to modify event latency and/or label with browsing data streams. In addition, event can be removed from this GUI.

Features

- Multiple data stream browsing
- Event latency and label modification
- Event removal


MoBI: Modify/Remove Events
Launch Main and Sub Browsers
Main Browser
Sub Browsers
Check Items to Modify/Remove
Clear all
Select Event and Modify/Remove it
RESET
UNDO
Refresh
Remove
Modify
<<<
<<
<
>
>>
>>>

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/sample_MoBI

balls_pos

videostream__untitled

```
196.847 [s] : Begin_001
197.014 [s] : Release-R
197.314 [s] : BallTop-fromR
197.397 [s] : Release-L
197.580 [s] : Catch-L
197.730 [s] : BallTop-fromL
197.847 [s] : Release-R
198.014 [s] : Catch-R
198.114 [s] : BallBottom-inL
198.164 [s] : BallTop-fromR
198.280 [s] : Release-L
198.480 [s] : Catch-L
198.547 [s] : BallBottom-inR
```

- eeg_CESlab102
- mocap_CESlab102
- videostream__untitled
- remOcc_mocap_CESlab102
- balls_acc
- balls_pos
- balls_vel

Latency [s]: 196.847
Type: Begin_001

```
Selected Event :   196.847 [s]   Begin_001
```

Over View of Basic Steps

1) Start GUI program
2) Load MoBILAB data set
3) Launch data stream browsers
4) Select data streams of which events are modified
5) Select event to be modified/removed
6) Set correct event latency and/or label
7) Save modification or remove event

2018. Oct. 2nd

How To

1) Start GUI program

GUI window will be launched by running "mobi_gui_modifyEvent" function from Matlab command line.

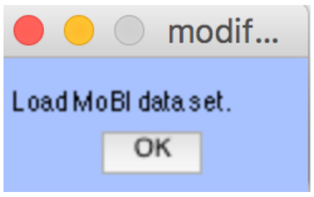
```
>> objMoBI = modify_event_gui();
```

The function returns a "dataSourceMoBI" class object. The object can be given as input argument when you run the function again. Then MoBILAB data set loading (step 2) will be skipped.

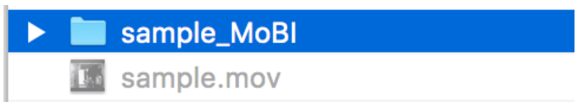
```
>> objMoBI = modify_event_gui(objMoBI);
```

2) Load MoBILAB data set

If you run "mobi_gui_modifyEvent" function without input argument, you will be asked to select MoBILAB data set folder. Click ***OK*** button in the pop-up window.

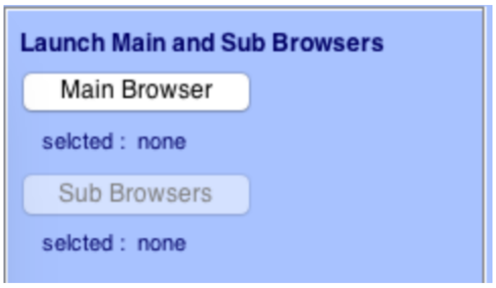


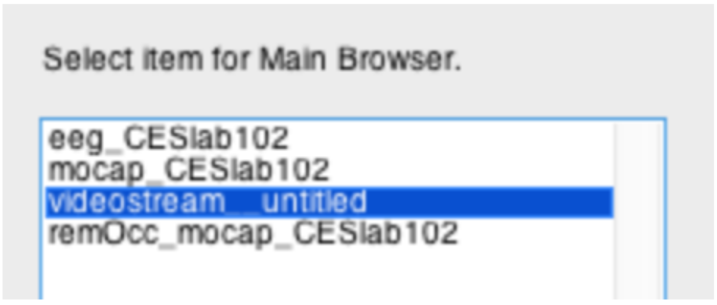
At the folder selection dialog box appearing next, select MoBILAB data set folder you want to load.



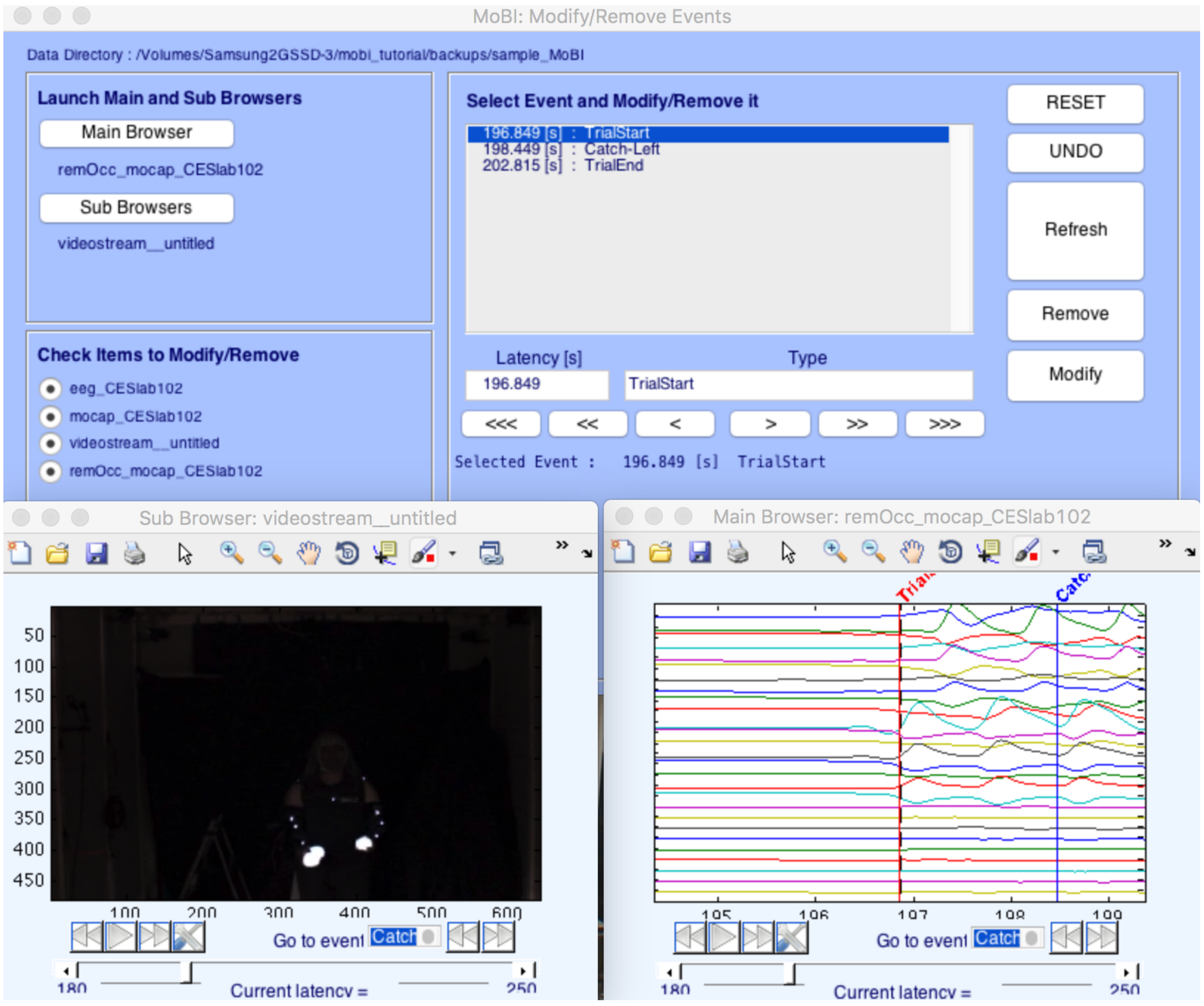
3) Launch data stream browsers

You can launch one or more data stream browsers for checking events. Click ***Main Browser*** button in the upper-left panel. Then select the name of data stream you want to browse.





You can launch additional browsers by clicking ***Sub Browsers*** button.



MoBI: Modify/Remove Events

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI

Launch Main and Sub Browsers

Main Browser

remOcc_mocap_CESlab102

Sub Browsers

videostream__untitled

Check Items to Modify/Remove

● eeg_CESlab102
● mocap_CESlab102
● videostream__untitled
● remOcc_mocap_CESlab102

Select Event and Modify/Remove it

```
196.849 [s] : TrialStart
198.449 [s] : Catch-Left
202.815 [s] : TrialEnd
```

Latency [s]: 196.849

Type: TrialStart

<<< << < > >> >>>

Selected Event : 196.849 [s] TrialStart

RESET

UNDO

Refresh

Remove

Modify

Sub Browser: videostream__untitled

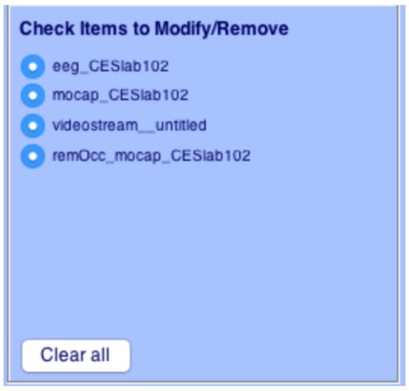
Main Browser: remOcc_mocap_CESlab102

NOTE : Which data stream should be selected for Main Browser?

As same as GUI for event marking, there are buttons changing current latency of browsers. Timestamp of the data stream selected for Main Browser is referred when those buttons are clicked. In addition, event list shown in the GUI is made by referring to events data of the "Main" data stream. Therefore, the data stream most related to the events that you are looking for is recommended to be browsed with Main Browser.

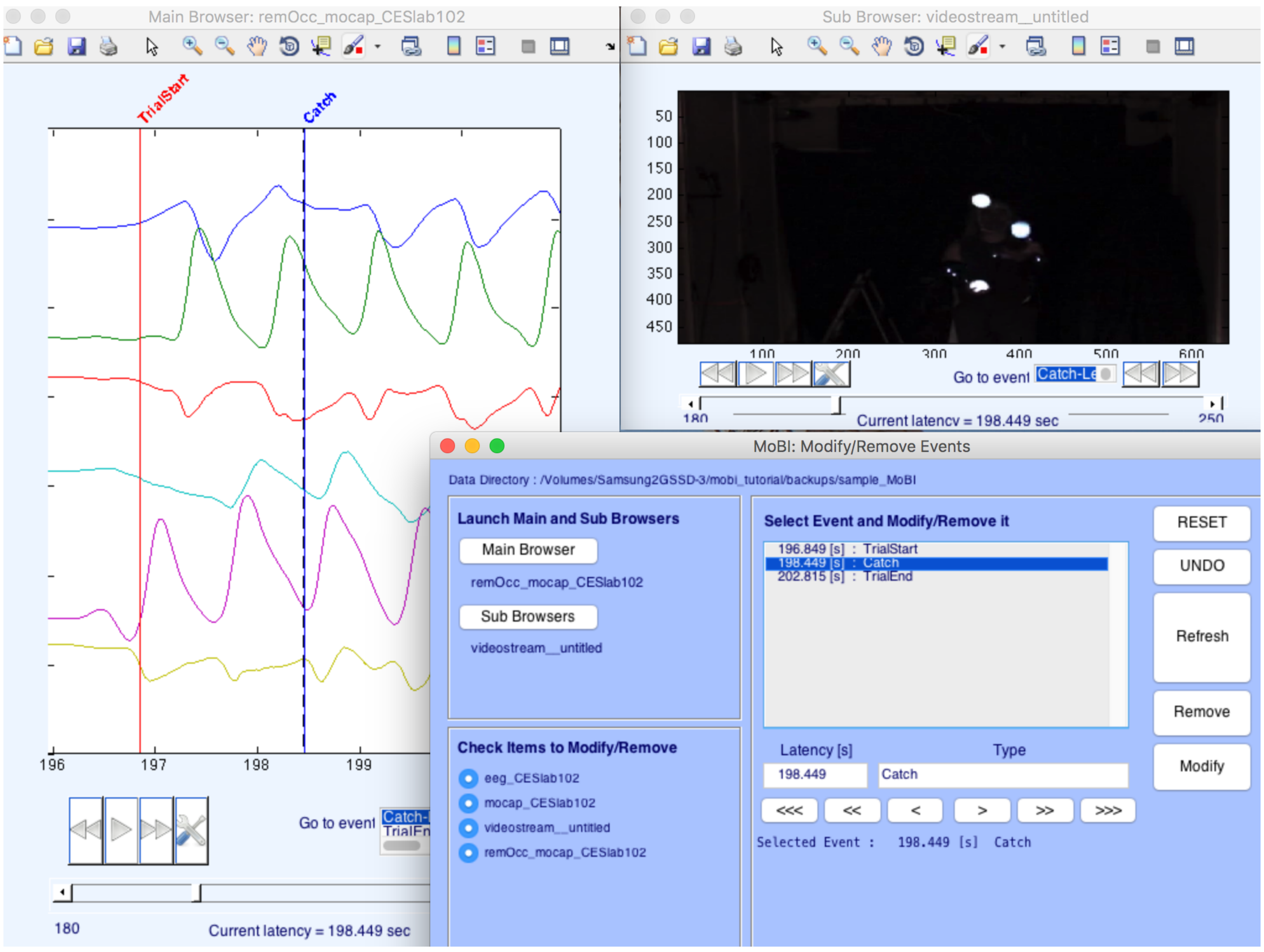
4) Select data streams of which events are modified

By checking on/off the radio buttons in bottom-left panel, you can select data streams of which events will be modified. Initially, all data streams are selected.



5) Select event to be modified/removed

To select event you want to modify or remove, click that event in the list box in the right panel. Latency and label of the event will appear in the text boxes under ***Latency [s]*** and ***Type*** texts, respectively. In addition, the current latency of each browser becomes as the latency of the event.



Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI

```
196.849 [s] : TrialStart
198.449 [s] : Catch
202.815 [s] : TrialEnd
```

Latency [s]: 198.449 Type: Catch

```
Selected Event :   198.449 [s]  Catch
```

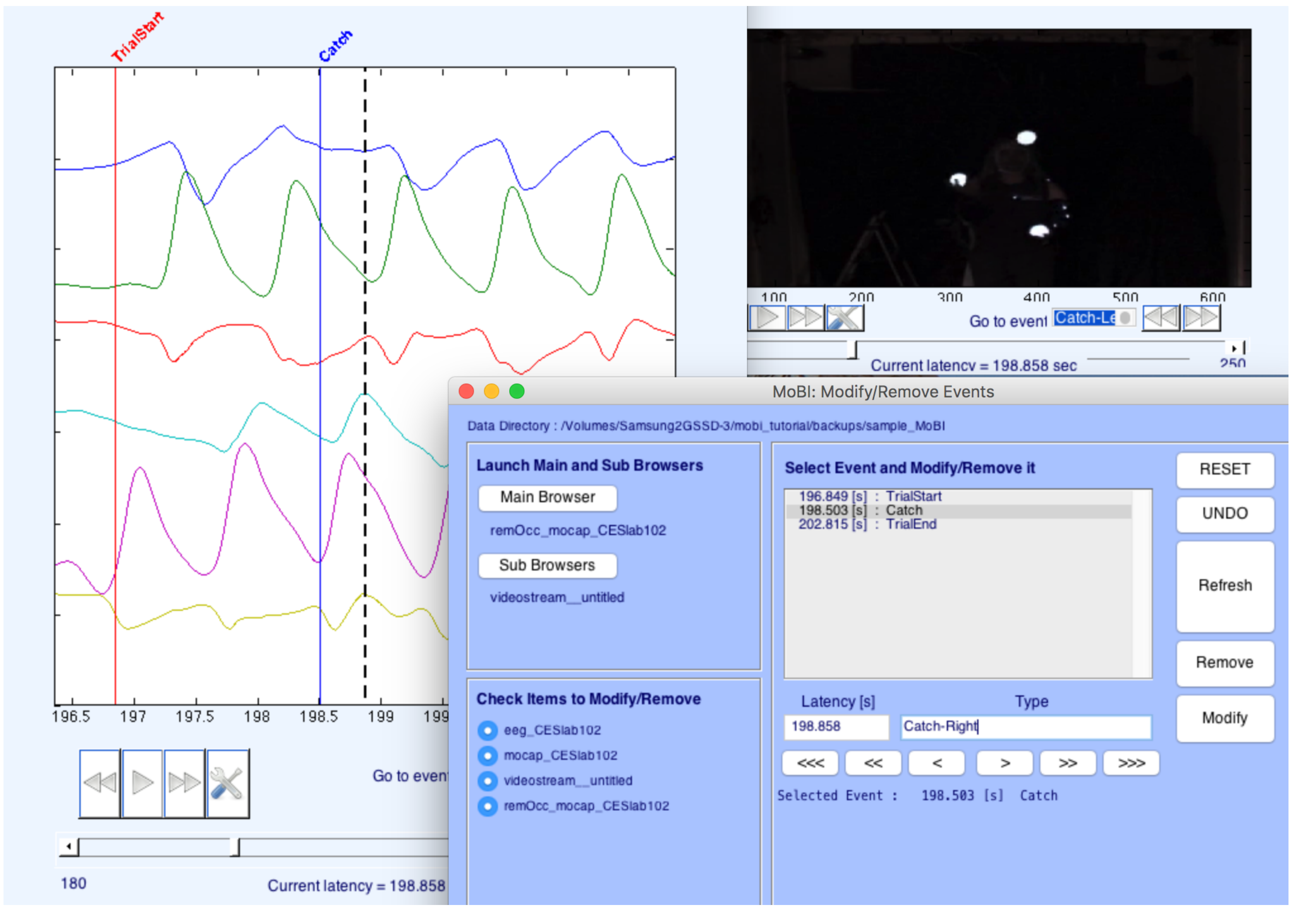
If you are not sure which event in the list corresponds to the event you are looking for, browse the "Main" data stream by playing buttons in the browser until you find the event. Then click ***Refresh*** button in right panel in the GUI. The event whose latency is closest to the current latency of Main Browser will be selected.

6) Set correct event latency and/or label

You can skip this step if you want to remove event.

To modify event latency, you can directly enter the latency in the text box under ***Latency [s]*** text. Or you can adjust latency by clicking <<< , << , < , > , >> , >>> buttons. The current latency of the browsers will be changed to the latency in the text box.

To modify event label, you can directly change the label in the text box.



Current latency = 198.858 sec

Current latency = 198.858

Data Directory : /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI

```
196.849 [s] : TrialStart
198.503 [s] : Catch
202.815 [s] : TrialEnd
```

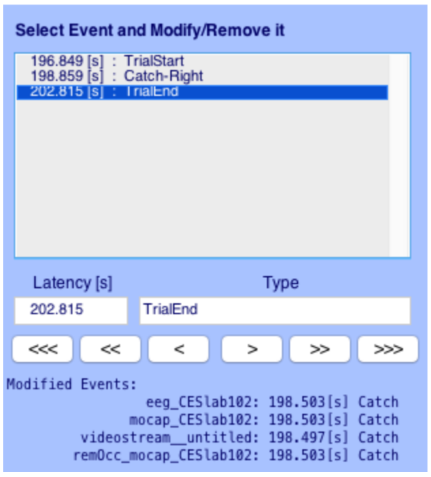
Latency [s]: 198.858

Type: Catch-Right

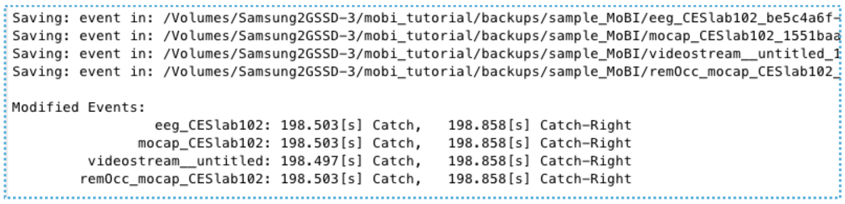
```
Selected Event :   198.503  [s]   Catch
```

7) Save modification or remove event

If you want to modify the event information, click ***Modify*** button to save the modification. The event information in the list will be modified. And you will see the messages in the GUI and Matlab command line that event data in the data streams are modified.



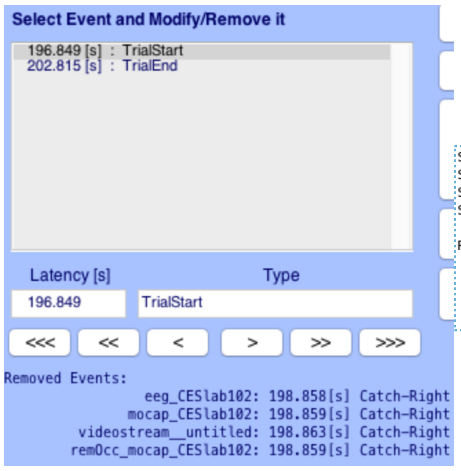
```
Modified Events:
          eeg_CESlab102: 198.503[s] Catch
        mocap_CESlab102: 198.503[s] Catch
   videostream__untitled: 198.497[s] Catch
  remOcc_mocap_CESlab102: 198.503[s] Catch
```



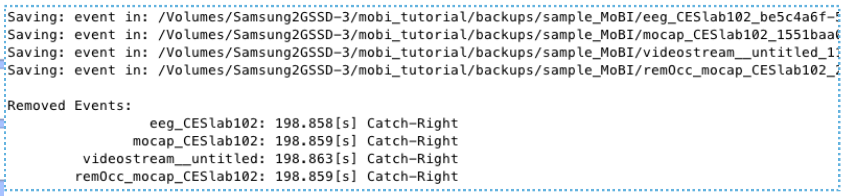
```
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/eeg_CESlab102_be5c4a6f-
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/mocap_CESlab102_1551baa
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/videostream__untitled_1
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/remOcc_mocap_CESlab102_

Modified Events:
                eeg_CESlab102: 198.503[s] Catch,   198.858[s] Catch-Right
              mocap_CESlab102: 198.503[s] Catch,   198.858[s] Catch-Right
        videostream__untitled: 198.497[s] Catch,   198.858[s] Catch-Right
       remOcc_mocap_CESlab102: 198.503[s] Catch,   198.858[s] Catch-Right
```

To remove event, select ***Remove*** button. The event will be removed from the list. And you will the see messages in the GUI and Matlab command line that event in the data streams are removed.



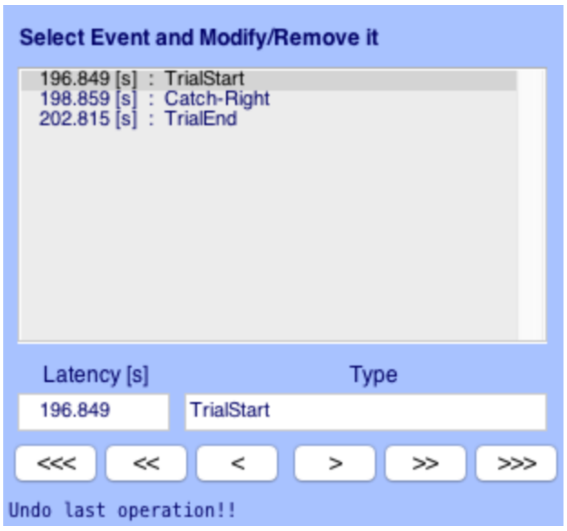
```
Removed Events:
          eeg_CESlab102: 198.858[s] Catch-Right
        mocap_CESlab102: 198.859[s] Catch-Right
   videostream__untitled: 198.863[s] Catch-Right
  remOcc_mocap_CESlab102: 198.859[s] Catch-Right
```



```
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/eeg_CESlab102_be5c4a6f-5
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/mocap_CESlab102_1551baa6
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/videostream__untitled_1
Saving: event in: /Volumes/Samsung2GSSD-3/mobi_tutorial/backups/sample_MoBI/remOcc_mocap_CESlab102_

Removed Events:
                eeg_CESlab102: 198.858[s] Catch-Right
              mocap_CESlab102: 198.859[s] Catch-Right
        videostream__untitled: 198.863[s] Catch-Right
       remOcc_mocap_CESlab102: 198.859[s] Catch-Right
```

If you want to undo the last modification you made, click ***UNDO*** button.



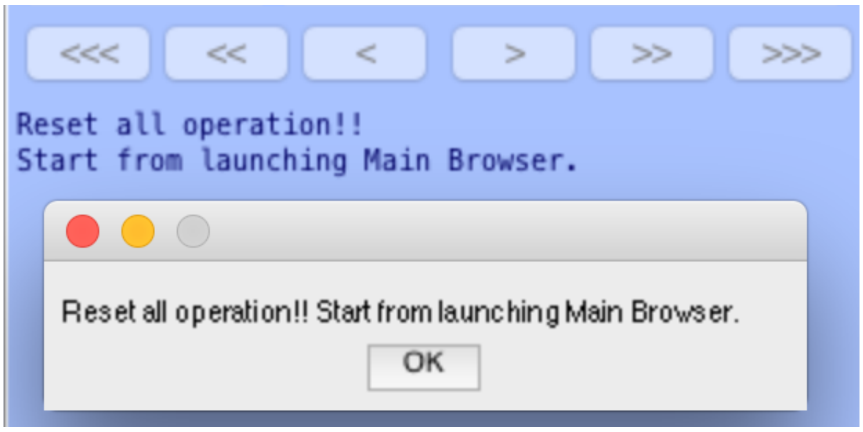
Select Event and Modify/Remove it

```
196.849 [s] : TrialStart
198.859 [s] : Catch-Right
202.815 [s] : TrialEnd
```

Latency [s]: 196.849
Type: TrialStart

```
Undo last operation!!
```

Also ***RESET*** button will withdraw all the modification you made since the GUI has been launched. You will be asked to re-start from launching Main Browser.



```
Reset all operation!!
Start from launching Main Browser.
```

Reset all operation!! Start from launching Main Browser.