How to load data

Directory tree

When “Default directory tree” checkbox is not marked all data paths can be edited, but it’s recommended to use the default paths.

The “Default directory tree” checkbox is marked by default meaning that just the “main data path” has to be set. Then every path will be filled automatically. For instance, if we are downloading data for the subject number twenty one cooking a salad of the Main Dataset, then each specific signal path will be filled as [in most cases key sensitive!]:

Main data directory	…/SXX/
Wired IMU sensors	…/SXX/3DMGX1/
Bluetooth IMU Sensors	…/SXX/6DOFv4/
Microphone	…/SXX/Audio/
MOCAP data	…/SXX/mocap/
RFID data	…/SXX/RFID/
eWatch data	…/SXX/eWatch/
Video timestamps	…/SXX/Video
Camera 1 frames	…/SXX/video/out6510211/
Camera 2 frames	…/SXX/video/out7150996/
Camera 3 frames	…/SXX/video/out8421130/
Camera 4 frames	…/SXX/video/out7151020/
Camera 5 frames	…/SXX/video/out7151062/
Camera 6 frames	…/SXX/video/out7150991/

IMPORTANT: To be able to show the data on the application, the video timestamps must be loaded and located in the subjects folder (..SXX/video or root of the subject, both work).

You can download a zip file with all the folders created; remember to delete the folders where you don’t put data: http://kitchen.cs.cmu.edu/Tools/directorytree.zip

Once the directory tree has been created just select the main data directory on the LoadFile (Matlab) application and all the paths will full fill automatically, if the files/folders haven’t been found, a message will appear on the textbox instead of the path, so… don’t select this data! You have to check manually if the cameras are in the folder or not. Remember to “clear all” before you start the application.

Once everything is loaded close the LoadFile Gui and start the “Control” application in Matlab, this will show another Gui, when pressing “Play” it will show all the signals you loaded.

Functions used to load data

getIMUData

Input

IMUPath	String with path of directory with all IMU data files

Output

eachDataIMU	Cell containing one data matrix per sensor file in the IMUPath.
TimeStampIMU	Vector including all timestamps in seconds
timeString	Cell which is a list of all timestamps
numSensorFiles	Double with the number of sensors found in the IMUPath
id	Vector containing the identity of each sensor

This function is used both for wired and Bluetooth IMUs and is called once for reading data of all wired sensors and once more for all the Bluetooth ones. Only one timestamp is generated for all processed IMU during the function call (e.g. for all IMU of the same kind, wired or Bluetooth).

getCamTimeStamp

Input

pathFile	String with the whole path to the timestamps directory
cameraName	Name of this camera

Output

textTimeStampCam	Cell of strings including all timestamps
num1stFrame	Number of the first frame in the video file, which may be taken into account in the case images generated from the video are numbered starting in #1 instead of the right number
numlastFrame	Number of the last frame in the video file, which may be taken into account in the case images generated from the video which contains less frames than lines in timestamp file
numFrames	Number of frames in the timestamp file

This function is called once per camera generating one timestamp for each one.

getAudioTimeInit

Input

audioPath	String with the whole path to the audio directory which includes the *“recording.log”* file

Output

timeStart	String timestamp of the beginning of the record
timeStop	String timestamp of the end of the record

As all microphones record at the same time, this function is called just once.

readDecimateSamplesAudio

Input

filename	String with the whole path to the audio file
decimate	Factor to decimate the number of samples. As our main goal is to visualize all data and check its synchronization less samples are requested (original *fs*=44100Hz)
timeStart	Timestamp when recording started. It is found in the *“recording.log”* file

Output

data	Decimated data samples
timing	Vector with timestamps in seconds

This function is called once per audio file generating one data and timing for each microphone.

ReadDataMOCAP

Input

path	String with the whole path to the MoCap directory, which includes skeleton file (*.ASF), markers' position for each frame (*.AMC) and start timestamp (*mocapTime.txt*).

Output

wsDOF	Register including 3D position for all frames and for all markers
wsFeat	Register including processed feature (log of quaternions, angular velocity,

	absolute position or relative position) for all frames but just for selected markers (according to parameter *filter* in *paraF*)
wsPart	Register including processed feature (log of quaternions, angular velocity, absolute position or relative position) only for some decimated frames and just for selected markers (according to parameter *filter* in *paraF*)
paraF	Some parameters

This function takes long so it is very convenient to store the resulting variables in a MAT file.

ReadRFIDData

Input

RfidPath	String with path to the RFID text file

Output

data	Cell of strings with tag ID (if there is one) at each timestamp
timing	Vector with all timestamps in seconds

This function takes long so it is very convenient to store the resulting variables in a MAT file.

ReadeWatchData

Input

ewatchPath	String with path to the eWatch directory

Output

accel_X	Matrix with all data of X axis accelerometer
accel_Y	Matrix with all data of Y axis accelerometer
accel_Z	Matrix with all data of Z axis accelerometer
light	Matrix with all data of light sensor
timing	Vector with all timestamps in seconds

This function takes long so it is very convenient to store the resulting variables in a MAT file.