PupilLabs HSILab

Generated by Doxygen 1.10.0

Chapter 1

Pupil Labs Data Extraction

This project is to extract data from pupil labs recording file. The extraction is available in 2 forms, which is a semi-colon seperated file and an hdf5 file.

The input parameters is in config.json where user can and should input the local path of the computer.

1.1 HDF5 file

Pandas data structure is saved in this file format to allow easy access to the results withut recalculating values. an example on how to read the file is available in

```
examples/hdf5_reader.py
```

1.2 Doxygen documentation

this project uses the library doxygen, installing link can be found `here`, full documentation can be found here `doxygen_docu`.

in this repository, a doxygen file is already initialize. To regenerate documentation, run `doxygen Doxyfile`

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 getData Namespace Reference

Functions

- setup_logging ()
- unproject_points (points_2d, camera_matrix, distortion_coefs, normalize=False)
- cart_to_spherical (points_3d, apply_rad2deg=True)
- find_ranged_index (values, left_boundaries, right_boundaries)
- export_gaze (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_blinks (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_fixations (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_saccades (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_eyestates (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_imu (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_events (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- export_info (recording, export_path)
- export_scene_camera_calibration (recording, export_path)
- export_world_timestamps (recording, export_path)
- export (bool csv=True, bool hdf5=True, str recording_file=None, str recording_number=None, str export_↩ path=None)

4.1.1 Detailed Description

```
This script exports data from a Pupil Labs Neon recording into various formats such as CSV and HDF5.
It includes functions to export gaze data, blinks, fixations, saccades, eye states, IMU data, events, recordin
```

4.1.2 Function Documentation

4.1.2.1 cart_to_spherical()

```
getData.cart_to_spherical (
              points_3d,
              apply_rad2deg = True )
```

4.1.2.2 export()

```
getData.export (
             bool  csv = True,
             bool  hdf5 = True,
             str  recording_file = None,
             str  recording_number = None,
             str  export_path = None )
```

```
main callable function to export data from a Pupil Labs recording.

:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param recording_file: The path to the recording file. If None, defaults to the parent directory of the curren
:param recording_number: The recording number. If None, defaults to the first directory in the recording file
:param export_path: The path where the exported files will be saved. If None, defaults to a directory named "e
:return: None
```

4.1.2.3 export_blinks()

```
getData.export_blinks (
              recording,
              export_path,
             bool  csv = True,
             bool  hdf5 = True,
              hdf5_path = None )
```

```
Exports blink data from a recording to CSV and/or HDF5 format.

:param recording: The recording object containing blink data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.4 export_events()

```
getData.export_events (
              recording,
              export_path,
             bool  csv = True,
             bool  hdf5 = True,
              hdf5_path = None )
```

```
Exports event data from a recording to CSV and/or HDF5 format.

:param recording: The recording object containing event data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.5 export_eyestates()

```
getData.export_eyestates (
               recording,
               export_path,
              bool  csv = True,
              bool  hdf5 = True,
               hdf5_path = None )
```

```
Exports eye state data from a recording to CSV and/or HDF5 format.

:param recording: The recording object containing eye state data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.6 export_fixations()

```
getData.export_fixations (
               recording,
               export_path,
              bool  csv = True,
              bool  hdf5 = True,
               hdf5_path = None )
```

```
Exports fixation data from a recording to CSV and/or HDF5 format.

:param recording: The recording object containing fixation data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.7 export_gaze()

```
getData.export_gaze (
               recording,
               export_path,
              bool  csv = True,
              bool  hdf5 = True,
               hdf5_path = None )
```

```
Exports gaze data from a recording to CSV and/or HDF5 format.

:param recording: The recording object containing gaze data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.8 export_imu()

```
getData.export_imu (
              recording,
              export_path,
             bool  csv = True,
             bool  hdf5 = True,
              hdf5_path = None )
```

```
Exports IMU data from a recording to CSV and/or HDF5 format.


:param recording: The recording object containing IMU data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.9 export_info()

```
getData.export_info (
              recording,
              export_path )
```

```
Exports recording information to a JSON file.

:param recording: The recording object containing information.
:param export_path: The path where the exported file will be saved.
```

4.1.2.10 export_saccades()

```
getData.export_saccades (
              recording,
              export_path,
             bool  csv = True,
             bool  hdf5 = True,
              hdf5_path = None )
```

```
Exports saccade data from a recording to CSV and/or HDF5 format.


:param recording: The recording object containing saccade data.
:param export_path: The path where the exported files will be saved.
:param csv: Boolean indicating whether to export to CSV format.
:param hdf5: Boolean indicating whether to export to HDF5 format.
:param hdf5_path: The path for the HDF5 file if exporting to HDF5.
:return: None
```

4.1.2.11 export_scene_camera_calibration()

```
getData.export_scene_camera_calibration (
              recording,
              export_path )
```

```
Exports scene camera calibration data to a JSON file.

:param recording: The recording object containing camera calibration data.
:param export_path: The path where the exported file will be saved.
```

4.1.2.12 export_world_timestamps()

```
getData.export_world_timestamps (
              recording,
              export_path )
```

```
Exports world timestamps from a recording to a CSV file.
:param recording: The recording object containing world timestamps.
:param export_path: The path where the exported file will be saved.
```

4.1.2.13 find_ranged_index()

```
getData.find_ranged_index (
              values,
              left_boundaries,
              right_boundaries )
```

4.1.2.14 setup_logging()

```
getData.setup_logging ( )
```

```
Sets up logging configuration. Create a log directory if it does not exist, and configure logging to output to
```

4.1.2.15 unproject_points()

```
getData.unproject_points (
              points_2d,
              camera_matrix,
              distortion_coefs,
              normalize = False )
```

```
Undistorts points according to the camera model.

:param pts_2d, shape: Nx2
:return: Array of unprojected 3d points, shape: Nx3
```

4.2 runfile Namespace Reference

Variables

- config = json.load(f)
- recording_file = config["recording_file"]
- recording_number = config["recording_number"]
- export_path = config["export_path"]
- export_csv = config.get("csv", True)
- export_hdf5 = config.get("hdf5", True)
- csv
- hdf5

4.2.1 Detailed Description

```
Program runfile to export data from Pupil Labs recordings to CSV and HDF5 formats. referencing lib/getData.py
hdf5 files can be read again as a pandas DataFrame.An example of how to do this is in examples/extractHDF5.py
```

4.2.2 Variable Documentation

4.2.2.1 config

```
runfile.config = json.load(f)
```

4.2.2.2 csv

```
runfile.csv
```

4.2.2.3 export_csv

```
runfile.export_csv = config.get("csv", True)
```

4.2.2.4 export_hdf5

```
runfile.export_hdf5 = config.get("hdf5", True)
```

4.2.2.5 export_path

```
runfile.export_path = config["export_path"]
```

4.2.2.6 hdf5

```
runfile.hdf5
```

4.2.2.7 recording_file

```
runfile.recording_file = config["recording_file"]
```

4.2.2.8 recording_number

```
runfile.recording_number = config["recording_number"]
```

Chapter 5

File Documentation

5.1 lib/getData.py File Reference

Namespaces

- namespace getData

Functions

- getData.setup_logging ()
- getData.unproject_points (points_2d, camera_matrix, distortion_coefs, normalize=False)
- getData.cart_to_spherical (points_3d, apply_rad2deg=True)
- getData.find_ranged_index (values, left_boundaries, right_boundaries)
- getData.export_gaze (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_blinks (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_fixations (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_saccades (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_eyestates (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_imu (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_events (recording, export_path, bool csv=True, bool hdf5=True, hdf5_path=None)
- getData.export_info (recording, export_path)
- getData.export_scene_camera_calibration (recording, export_path)
- getData.export_world_timestamps (recording, export_path)
- getData.export (bool csv=True, bool hdf5=True, str recording_file=None, str recording_number=None, str export_path=None)

5.2 readme.md File Reference

5.3 runfile.py File Reference

Namespaces

- namespace runfile

Variables

- runfile.config = json.load(f)
- runfile.recording_file = config["recording_file"]
- runfile.recording_number = config["recording_number"]
- runfile.export_path = config["export_path"]
- runfile.export_csv = config.get("csv", True)
- runfile.export_hdf5 = config.get("hdf5", True)
- runfile.csv
- runfile.hdf5