README.md

RatCAVE VR Experiments, Round 2 Data

Author: Nicholas A. Del Grosso

File Naming Convention

These filenames are very long; they contain many pieces of metadata, each seperated by an underscore. (**Note**: It will not be necessary to parse the filenames to get metadata--each is already available inside the **Attributes** section of the HDF5 File (described below))

Filename sections are as follows:

- Experiment Name
- Date (YYY-MM-DD)
- Time Session Started (HH-MM-SS)
- Rat Name (always starts with "VR-", with cagemates sharing number codes)
- (Experiment-Specific Session Values)
- Experimenter (N: Nicholas Del Grosso; E: Eduardo Blanco)
- Recording Day Code (XXX-XXX).
  - This code matches the hand-written observation log sheet from that Day. Experimenter Comments from the session and observer-coded data can be found in the office binder, and digitally-scanned copies at ***____(Fill in Path)***

Original Data Formats

Each session originally consisted of __ files, which have been merged into a single hdf5 file and avi video file. They can be found in //theta/storage/nickdg/data/VR_Experiments_Round_2/Motive Files/VR_Experiment_Data

- **'session_name.tak' file**: These are NaturalPoint's Optitrack Motive software proprietary motion capture file format. They can be read into Motive, but that's about it. They contain all motion capture data and video data.

- **'session_name_log.csv' file**: The event log for the experiment, it contains each event that changed during the session, including the Motive Timestamp when it happened, the name of the event type, and the arguments for that event. In most analyses, we only need to know when the **"set_scene"** event occurred, because this is when a new phase of the experiment happens.

- **'session_name_log.json' file**: The session settings. These contain all of the session data, including things like the location of virtual objects, what kind of session (e.g. "Real" or "Virtual" cliff), the rat name, and more.

Middle Data Formats

Motive Take files were exported to standard text (CSV) and video (AVI) files, before the data was further preprocessed. These can be found at /home/nickdg/theta_storage/data/VR_Experiments_Round_2/Converted Motive Files.

However, the HDF5 files (listed below) are intended to be the starting point for further analysis.

Preprocessed Data Formats

- **'session_name.h5' file**: These are the primary data files we will be analyzing from. They contain all of the above data, plus the tracking data has been re-aligned so that each session is in the same space relative to each other, making analysis simpler.

- **'session_name.avi' file**: Video Files from overhead Motive camera. These aren't the prettiest (they're pretty noisy and dark), because of some tradeoffs with tracking quality imposed by the Motive software. Their noise also makes them pretty large files because of compression difficulty--processing these files will help a lot. Still, they are useful references for checking results and better-interpreting patterns in the tracking data.

HDF5 File Organization

Although the HDF5 files (*.h5 files) contain lots of datasets, they don't all need be loaded into RAM; you can specify which data you want to read, making analysis faster.

HDF5 files are internally organized like a filesystem. They contain their own directories ("**Groups**"), files ("**Datasets**"), symlinks ("**Softlinks**"), and **Attributes**, which are metadata attached to a Group or Dataset. All of these are used in the VR files. Below is the organization, starting with the root Group, which is called '/':

These files can be found in //theta/storage/nickdg/data/VR_Experiments_Round_2/Motive Files/VR_Experiment_Data/preprocessed.

They have also been sorted by experiment through symlinks. This can be more convenient for analyzing a specific experiment's data, and can be found in //theta/storage/nickdg/data/VR_Experiments_Round_2/Motive Files/VR_Experiment_Data/preprocessed_by_experiment.

Following is a Directory Tree view and details of the HDF5 files:

```
/: Root Folder.  Contains all session metadata from Motive and the VR experiment as Attributes.
|   README.md
|   file001.txt
|
└───raw: This Group contains the nmodified tracking data.
|   |
|   └───Rigid Body: Contains Position and Rotation of the set of markers corresponding to a rigid body.
|   |   └───Arena
|   |   |   └───Dataset: Error Per Marker
|   |   |   |      - Description: Provides a rough estimate of overall tracking accuracy (meters)
|   |   |   |      - Dimensions: NFrames x 3 (Frame, Time, Error Per Marker)
|   |   |   |
|   |   |   └───Dataset: Rotation
|   |   |   |      - Description: Instantaneous Body Rotation from original orientation (quaternions)
|   |   |   |      - Dimensions: NFrames x 6 (Frame, Time, X, Y, Z, W)
|   |   |   |
|   |   |   └───Dataset: Position
|   |   |          - Description: Instantaneous Body Position, as registered in Motive (meters)
|   |   |          - Dimensions: NFrames x 5 (Frame, Time, X, Y, Z)
|   |   |          - Note: Motive uses Y as the Height. For 2D analysis, use X and Z.
|   |   |          - Note: For most Bodies, this the mean Marker position.
|   |   |          - Note: For the Rat, this is a position between the rat's eyes.
|   |   |
|   |   └───Rat
|   |   |   └───Dataset: Error Per Marker
|   |   |   |
|   |   |   └───Dataset: Rotation
|   |   |   |
|   |   |   └───Dataset: Position
|   |   |
|   |   └───(All Other Tracked Rigid Bodies)
|   |
|   |
|   └───Rigid Body Markers: Contains Position of each marker tracked.
|       |
|       └───Arena_1: Name of the marker ("BodyName_MarkerNumber").
|       |   |  - Note: This marker name is the same across all sessions.
|       |   |
|       |   └───Dataset: Marker Quality
|       |   |      - Description: Provides a rough estimate of overall tracking accuracy (meters)
|       |   |      - Dimensions: NFrames x 3 (Frame, Time, Marker Quality)
|       |   |
|       |   └───Dataset: Position
|       |          - Description: XYZ Position of the Marker over Time (meters)
|       |          - Dimensions: NFrames x 5 (Frame, Time, X, Y, Z)
|       |
|       └───Arena_2
|       |   └───Dataset: Marker Quality
|       |   |
|       |   └───Dataset: Position
|       |
|       └───(All Other Markers)
|
└───preprocessed: This Group contains all re-aligned tracking data, plus an Orientation dataset.
|   |
|   └───Rigid Body
```

```
│   │   └───(All Tracked Rigid Bodies, same as in "raw")
│   │       └───Dataset: Error Per Marker
│   │       │
│   │       └───Dataset: Rotation
│   │       │     -Note: This is unchanged from the "raw" Rotation, but is still valid for analysis.
│   │       │
│   │       └───Dataset: Position (re-aligned with X and Z arena-centered and the arena floor height at Y=0
│   │       │
│   │       └───Dataset: Orientation: An XYZ vector describing which direction the rigid body is facing.
│   │       │     - Note: For the Rat, this is direction the rat is facing.
│   │       │
│   │       └───Markers:
│   │          └───Arena_1: A Softlink to the Rigid Body Markers, specific to that Rigid Body.
│   │
│   └───Rigid Body Markers:
│       └───(All Tracked Markers, same as in "raw")
│           └───Dataset: Marker Quality
│           └───Dataset: Position (realigned, same as Rigid Bodies)
│
└───events: This Group contains all VR event log info (anything that chabged over time in the experiment)
    │
    └───Dateset: eventLog
    │      - Description: When each event occured
    │      - Dimensions: Nx3 (Frame, Time, MotiveTime)
    │
    └───Dateset: eventNames
    │      - Description: The name of each event.
    │      - Dimensions: Nx1 (array of strings)
    │
    └───Dateset: eventArguments
           - Description: The data that went into each event function in the original VR script.
           - Dimensions: Nx1 (array of roughly-JSON-formatted strings)
```

Note: Only the Root folder was given Attribute data, so there's no need to search for that info throughout the file.

Note: All Markers correspond to a given Rigid Body--unlabeled markers were not exported to HDF5, but are available in the original Motive Take Files.

Attributes

There are dozens of attributes in each file, but most can be safely ignored for most analysis. The experiment settings are all in capital letters and start with the experiment name.

Important attributes for each experiment are as follows:

All Experiments

- **RAT**: The rat name.
- **EXPERIMENT**: The Experiment name.
  - *Note*: For the VR_SpatialNovelty experiment, it also contains info on whether a Black Background was used for the session.
- **EXPERIMENTER**: The Experimenter's name (the person that conducted that session, either Nicholas or Eduardo)
- **Capture Frame Rate**: The Frame rate of the session's motion tracking (Hz)
- **PAPER_LOG_CODE**: The observation log code for that day. This matches the paper log, which contains additional experimenter comments. (XXX-XXX) It shares the same value as all sessions for that rat on a given day, allowing Cliff and Object/Wall experiments to be easily matched up, for cross-experiment analyses.

VR_Cliff Experiment

- **CLIFF_SIDE**: Which side of the arena the cliff was on, on the X axis along the arena length (Either "L" or "R").
- **CLIFF_TYPE**: Which condition the session was ("Real": A real visual cliff, "VR": the virtual cliff, or "Static": a stationary virtual cliff, which didn't update with the rat's movement).

-**CLIFF_COVER_RAT_WITH_UMBRELLA**: Whether the rat had a black circle over its head, which followed its movements (1: yes, 0: no).

- **CLIFF_FILENAME**: The .obj file (loadable in Blender3D) that contains the 3D meshes rendered in the VR system.

VR_Wall Experiment

- **VR_WALL_X_OFFSET**: The Arena X position of the wall during Phase 2 (meters). This wall stretched across the whole arena width, and could only be -0.4, -0.2, 0.2, or 0.4.
- **VR_OBJECT_FADE_SPEED**: The speed of one inter-phase fade. Each Phase had a "Fade to Black", then "Fade Back" stage. To calculate the duration of each step, divide 1 second by this value (1/value).

VR_Object Experiment

- **VR_OBJECT_FADE_SPEED**: The speed of one inter-phase fade. Each Phase had a "Fade to Black", then "Fade Back" stage. To calculate the duration of each step, divide 1 second by this value (1/value).
- **VR_OBJECT_FILENAME**: The filename, without an extension.
- **VR_OBJECT_NAME**: Which object shape was used for this session.
- **VR_OBJECT_ROBO_ARM_WAIT_DURATION_SECS**: How many seconds there were between fades, to wait for the robotic arm to move. (Note: this duration was used even in Virtual Object sessions.)
- **VR_OBJECT_SIDE**: Which side the object appeared on, on the X axis along the arena length (Either "L" or "R").
- **VR_OBJECT_TYPE**: Which type of object was present (Either "Real" or "VR")
- **VR_OBJECT_POSITION_L**: XYZ Array, the position of the Left Object location (should be roughly [-.2, .07, 0])
- **VR_OBJECT_POSITION_R**: XYZ Array, the position of the Right Object location (should be roughly [.2, .07, 0])

VR_Spatial_Novelty Experiment

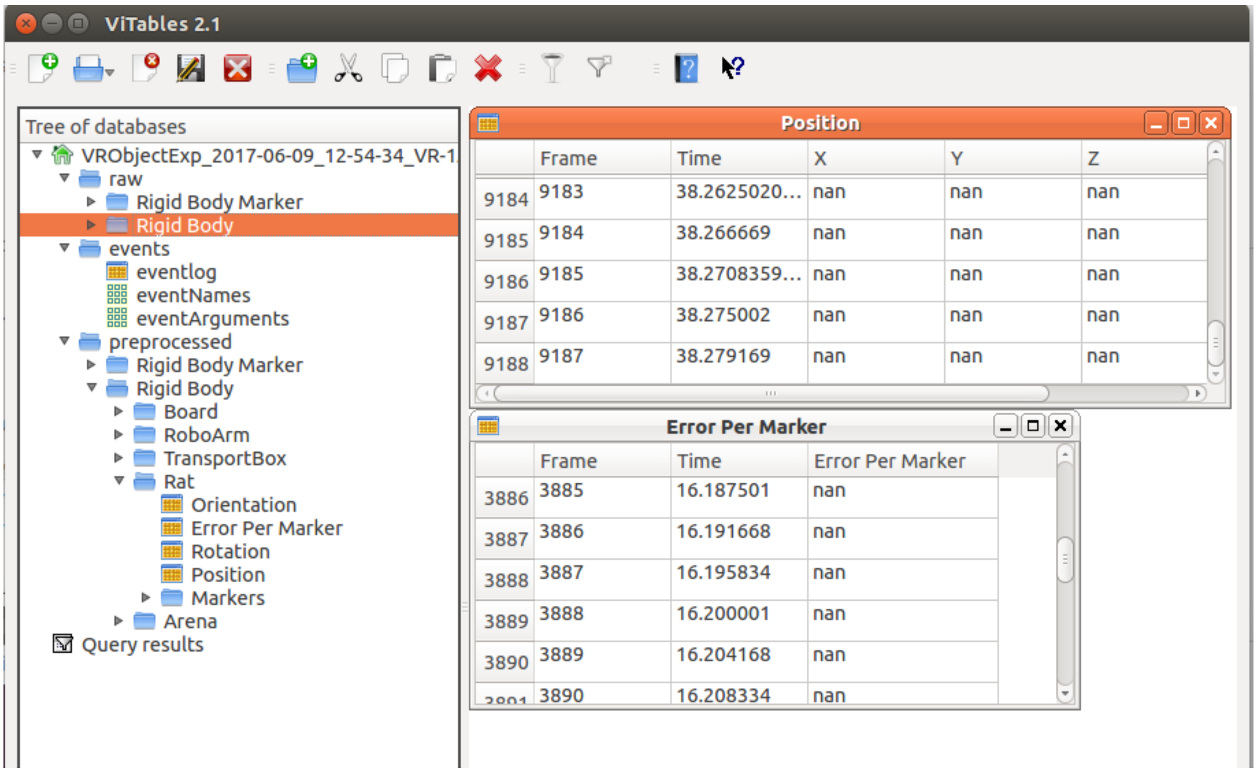
- **VR_SPATIAL_NOVELTY_FADE_SPEED**: The speed of one inter-phase fade. Each Phase had a "Fade to Black", then "Fade Back" stage. To calculate the duration of each step, divide 1 second by this value (1/value).
- **VR_SPATIAL_FIXED_OBJECT_POSITIONS**: An Nx3 XYZ position array for each of the possible "fixed" objects in the experiment.
- **VR_SPATIAL_NOVELTY_FIXED_POSITION**: A 1-indexed Integer, corresponding to which Fixed object position was used in the session.
- **VR_SPATIAL_NOVELTY_OBJECT_POSITIONS**: An Nx3 XYZ position array for each of the possible "novel" and "familiar" objects in the experiment.
- **VR_SPATIAL_NOVELTY_NOVEL_POSITION**: A 1-indexed Integer, corresponding to which "Novelty" object position was used in the session.
- **VR_SPATIAL_NOVELTY_FAMILIAR_POSITION**: A 1-indexed Integer, corresponding to which "Novelty" object position was used in the session.
- **VR_SPATIAL_NOVELTY_OBJECT_NAME**: Object Name, describing the shape of the object.
- **VR_SPATIAL_NOVELTY_OBJECT_TYPE**: Whether the object was a 3D hovering one, or just a flat image ("3D", "2D", or "Real")

Reading HDF5 Data

Visualizing the Data

Because HDF5 is a standard file format, several readers are available, including GUI readers, allowing you to easily browse the data and get a feel for what is inside it. The one I've been using is called ViTables (http://vitables.org/), and should be cross-platform. Very convenient!

Here is a screenshot of ViTables in use:



- VRObjectExp_2017-06-09_12-54-34_VR-1
  - raw
    - Rigid Body Marker
    - Rigid Body
  - events
    - eventlog
    - eventNames
    - eventArguments
  - preprocessed
    - Rigid Body Marker
    - Rigid Body
      - Board
      - RoboArm
      - TransportBox
      - Rat
        - Orientation
        - Error Per Marker
        - Rotation
        - Position
        - Markers
      - Arena

Position

	Frame	Time	X	Y	Z
9184	9183	38.2625020…	nan	nan	nan
9185	9184	38.266669	nan	nan	nan
9186	9185	38.2708359…	nan	nan	nan
9187	9186	38.275002	nan	nan	nan
9188	9187	38.279169	nan	nan	nan

Error Per Marker

	Frame	Time	Error Per Marker
3886	3885	16.187501	nan
3887	3886	16.191668	nan
3888	3887	16.195834	nan
3889	3888	16.200001	nan
3890	3889	16.204168	nan
3891	3890	16.208334	nan

MATLAB Functions

All Matlab HDF5 functions start with **h5**. Most of them take a filename and a **location**, a string that says which Group or Dataset to read (This looks just like a file path. For example: '/preprocessed/Rigid Body/Rat/Position'). They return structure arrays for the most part.

- **h5read()** and **h5info()**: These files return Dataset data or metadata, respectively, as structure arrays.

For example, to calculate the mean Rat position in a session:

```
ratpos = h5read('MyFile.h5', '/preprocessed/Rigid Body/Rat/Position');
xyz = [ratpos.X, ratpos.Y, ratpos.Z];
xyz_mean = nanmean(xyz);
sprintf('Mean Position: (X=%.2f, Y=%.2f, Z=%.2f)', xyz_mean)
```

- **h5reatattr()**: A shortcut for getting a specific attribute. Takes the attribute name as a third input. **Note**: In our files, all of the attributes are located at the Root group, indicated by a single slash ("/").

For example, to get the Experimenter Name in a session:

```
experimenter = h5readattr('MyFile.h5', '/', 'EXPERIMENTER')
```

You can also explore the file's Attributes and directory structure using the h5info() function. I find it a bit unwieldy, though, in Matlab to explore structure arrays. One solution is to convert it to a table to quickly see what attributes there are:

```
session = h5info('MyFile.h5', '/')
struct2table(session.Attributes)
```

Note: Please do not modify the HDF5 data files or the directories they are inside. Further analysis files should be saved somewhere else.

Python Functions

HDF5 Files can be worked with in Python using the **h5py**, and **Pandas** packages. h5py is a low-level reading and writing library that allows dict-like file browsing and Numpy array reading and writing, and Pandas is used for working with data as a pandas.DataFrame object.

To read a Dataset with Pandas and calculate the mean position:

```
import pandas as pd

rat_position = pd.read_hdf('MyFile.h5', '/preprocessed/Rigid Body/Rat/Position')
xyz_mean = rat_position[['X', 'Y', 'Z']].mean()
print(xyz_mean)
```

To do the same with h5py:

```
import h5py

with h5py.File('MyFile.h5') as f:
    rat_position = f['/preprocessed/Rigid Body/Rat/Position'][:]
xyz_mean = rat_position[['X', 'Y', 'Z']].mean()
print(xyz_mean)
```

h5py is needed for attribute access, since there is no Dataframe equivilent for HDF5 metadata. For example, to get the Experimenter Name in a session:

```
import h5py

with h5py.File('MyFile.h5') as f:
        experimenter = f.attr['Experimenter']
```

Browsing the file in h5py is straightforward, as it uses a dict-like interface for both the directory tree and for the Attributes (found as the "attr" attribute). For example, to get a list of all attributes in a file:

```
import h5py

with h5py.File('MyFile.h5') as f:
        attribute_names = [name for name in f.attr]
print(attribute_names)
```

Note: Please do not modify the HDF5 data files or the directories they are inside. Further analysis files should be saved somewhere else.