autoTRAM
User Manual

Corson Areshenkoff

November 27, 2014

Contents

1 Introduction and information

TO DO

2 Usage

Participant log files should be placed in the `log_files` directory. No alteration needs to be made to the contents of the logs, though the file names will be used as labels in the output, and so they should uniquely identity the participants from whence they came. The (x, y) coordinates of the platform should be placed in a text file `support_files/platform_coords.txt`. Coordinates should be separated by a single space.

To run the program, **R** should be directed to the autoTRAM folder using

```
setwd('<path-to-autoTRAM>')
source('autoTRAM.R')
```

The program can then be run by entering

```
autoTRAM()
```

The program will then automatically load the entire `log_files` directory and output a collection of `.csv` files to the `batch_data` folder. A complete summary of the output can be found in the followng section.

3 Log structure

The log file is generated by UDK and contains output from both Kismet and the playerGPS mutation. The latter outputs time stamped vectors of player coordinates (x, y, z) and rotation vectors (pitch, roll, yaw) at a rate of approximately 60hz. The former outputs a notification whenever a particular condition is satisfied (trial started, platformed touched etc). The autoTRAM import function reads the log file and separates the location and rotation data based on the Kismet notifications. An *epoch* is then defined to be the space between two successive Kismet notifications, with the epoch name corresponding to the Kismet message at the beginning of the epoch.

For example, Kismet may output a notification *t1 start* when trial 1 begins, and another notification *platform touched* when the participant finds the platform. The epoch *t1 start* then corresponds to the time between starting the first trial and touching the platform. The epoch *platform touched* corresponds to the time between touching the platform and starting trial two.

It is possible for multiple epochs to be given the same name, as will be the case if Kismet outputs *Platform touched* on each trial. This should cause no

ambiguity, however, since epoch are listed in order in the autoTRAM output. Still, it's best to choose Kismet messages to be informative.

4 Output and analyses

autoTRAM output is stored in the `batch_files` directory, and consists of separate `.csv` files for each summary. Output is structured so that each row corresponds to a participant (labeled with the name of the corresponding log file) and each column corresponds to an epoch. The analyses and resulting files are listed below:

4.0.1 Trial Latency (`trial_latency.csv`)

The total time, in seconds, from the beginning to the end of an epoch.

4.0.2 Path Length (`path_length.csv`)

The length of the path traversed from the beginning to the end of an epoch. This is calculated as the sum of the Euclidean distances between each successive (x, y) coordinate pair.

4.0.3 Ten-percent Time (`ten_perc_time.csv`)

The time taken to travel 10% of the total distance to the platform (estimated as the last coordinate in the epoch). First, the total distance from the first to the last position in the epoch is calculated. The output is then the first time at which the path length is equal to 10% of this distance.

4.0.4 Quadrant Dwell Time (`dwelltime_<quad>.csv`)

The proportion of time spent in each of the 4 quadrants of the environment. This function assumes that the center of the maze has coordinates $(0, 0, z)$. First, the player's position is converted to polar coordinates (r, θ), and then the time points are binned according to their location on the unit circle. The quadrants are then top-right, top-left, bottom-left, and bottom-right.

4.0.5 Platform Deviation (`platform_deviation.csv`)

The Euclidean distance between the participant's final location in each epoch and the true location of the platform. The (x, y) coordinates of the platform are stored in the file `platform_coords.txt` in the `support_files` directory, separated by a space.

4.0.6 Path Curvature (`path_curvature.csv`)

The estimated unsigned total curvature of the path traversed during the epoch. The (unsigned) curvature κ of a plane curve is defined as

$$\kappa = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

However, differences in navigation speed (particularly pauses or dwells in one location) may result in vanishing derivatives, and so we first compute a velocity-homogeneous path by sampling the observed path at fixed distances. The derivatives of this new path are then estimated by the first and second differences of the data series. The output is the sum over the curvature at every point in the path and is strictly non-negative, with values very near zero suggesting a straight-line path.

5 Implementation details

autoTRAM consists of three primary layers that are called in order from the `autoTRAM()` **R** function. They are

1. Batch level processing (`batch_functions.R`)
2. Import (`import_functions.R`)
3. Trial level processing (`analysis_functions.R`)

The workflow proceeds as follows: First,

TO DO