MouseLabTracker

User guide (v0.2.9)

Introduction

This document serves as a user guide for the ***MouseLabTracker*** software, which is a Matlab based application that uses imaging processing techniques to analyze video recordings of behavioral experiments and extract information about the animals positions at each time. Tracking the animals' position as a function of time then allows for studying locomotion, velocity, acceleration, time spent in regions of interest (ROI) or any other behavioral parameter under study.

The MouseLabTracker was created as a rewrite of the *Mousetracker* software (J Neurosci Methods 2006 157:91-7). Mousetracker was a similar solution with the same objectives, running exclusively on Windows platform. As a Matlab application, MouseLabtracker can run on many

Brain Institute - UFRN.
mouselabtracker@neuro.ufrn.br
http://www.neuro.ufrn.br/incerebro/

different platforms, such as Linux, Windows and OSX. We have successfully been using this software in our own research group at the Brain Institute, UFRN - Brazil. Starting from version 0.2.9, we decided to release MouseLabtracker to the general public.

This guide is updated each time we release a new version of MouseLabTracker. You should receive a copy of this guide along with the software; alternatively, this guide can be directly downloaded from the project website (http://www.neuro.ufrn.br/incerebro/), where one can obtain the latest version of the software.

What's new?

- Version 0.2.9 (11/08/2011)
    - Print detailed log information, with timestamps, to the Matlab command window.
    - Some names in the GUI were changed to better indicate the components function.
    - Keep the same tracking settings across different videos.
    - Bug fixed in the suggested file name when saving results (remove video file extension).
    - Bug fixed in the spreading of `last_*_dir.mat` files in the user file system.
    - Auto close the "Tracking Done" dialog when the user clicks the user saves the results.
    - Show the total elapsed time during tracking.
    - Bug fixed in the tracking progress display.

Getting started

Here we describe how to set up your MouseLabTracker environment and get it running.

Requirements

- Matlab 2009b or newer.
    - With support for the `mmreader` component.
- Matlab Image Processing Toolbox.

Downloading and Installing

The latest version of MouseLabTracker can be obtained by requesting a copy from mouselabtracker@neuro.ufrn.br. Since we like to keep track of the MouseLabTracker users, please do not distribute the software to others; rather, ask your colleagues to write to this email address and we will promptly provide them with the latest version. Users can also send suggestions and bug reports to this same address. We are currently working to allow direct download of MouseLabTracker from our webpage.

After getting a copy, simply decompress the *mouselabtracker.zip* package in the desired location, which will create a `mouselabtracker/` directory.

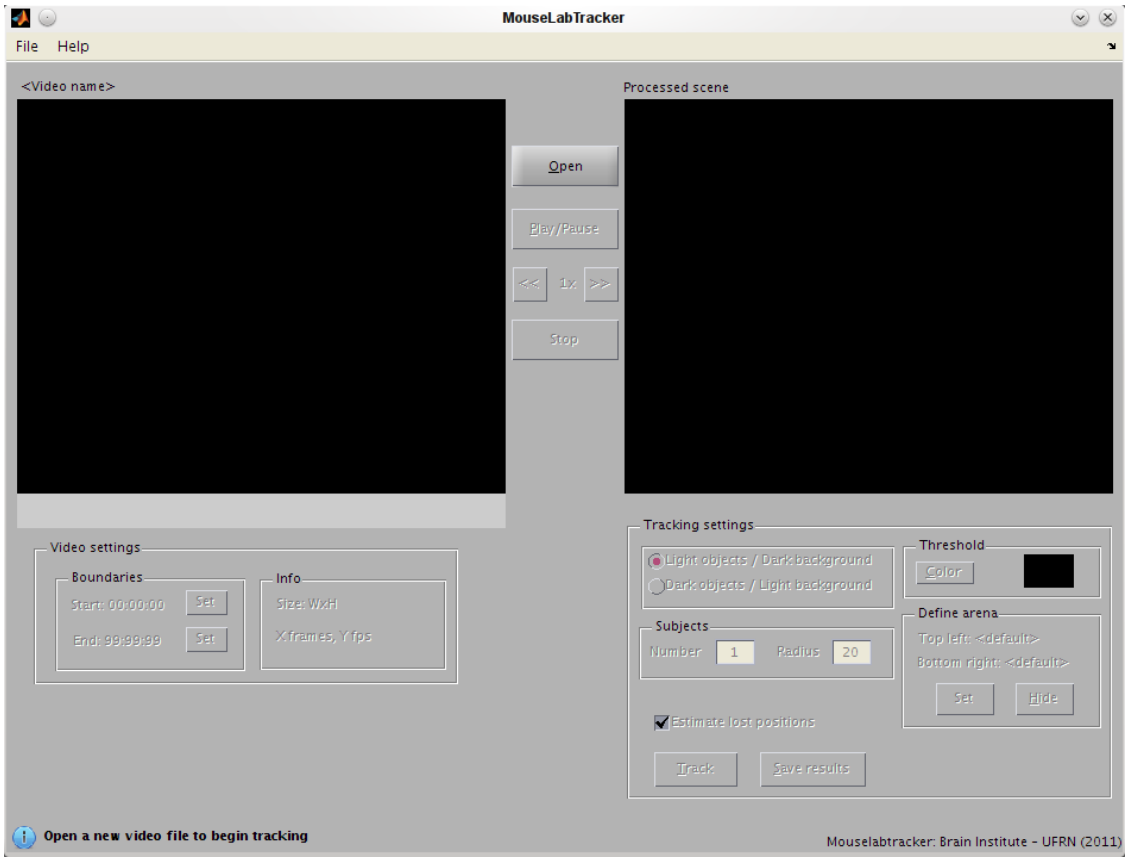
First run

- Open your Matlab environment and set `mouselabtracker/` as the current directory.
- Add mouselabtracker and all its sub-directories to the Matlab path
    - Select the directory in Matlab, right click it and select “Add to path” → “Selected folder and sub-folders”.
- Type "main()" to call the main function in the command window to start the software. You should see the following:

```
>> main()
[10:02:08] Mouselabtracker initiating
```

User interface

After calling `main()`, the following interface appears:



Note that, due to technical constraints, the interface looks different for each operating system (this is the Linux version), but retains the same functionality.

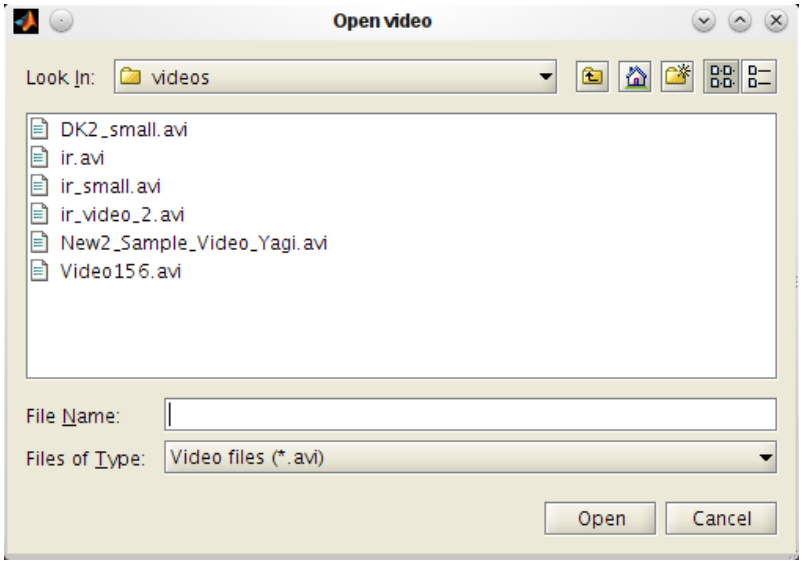
- There are 2 display panels (the black rectangles in the screenshot above). The left panel is the *Video Display* and it shows the original video. The right panel is the *Tracking Display*, which shows the processed frames.
- The components at the bottom of the left display panel are the *Video Components*. They show information about the video file and allow you to set the desired video boundaries (explained below).
- The 5 buttons at the top middle of the screen, between the display panels, are the *Video Controls* (Open, Play/Pause and Stop). They can be used to control video playback before tracking. When the software is tracking an animal, these buttons allow pausing and resuming the analysis.
- The components at the bottom of the right display panel are the *Tracking Components*. Here the user can set all tracking parameters and start the tracking process.

There are some keyboard shortcuts that activate interface buttons:

- Open: Alt+O
- Play/Pause: Alt+P
- Stop: Alt+Z
- Rewind (<<): Alt+Left
- Forward (>>): Alt+Right
- Color: Alt+C
- Define arena (“Set”): Alt+A
- Hide arena: Alt+H
- Track: Alt+T
- Save results: Alt+S

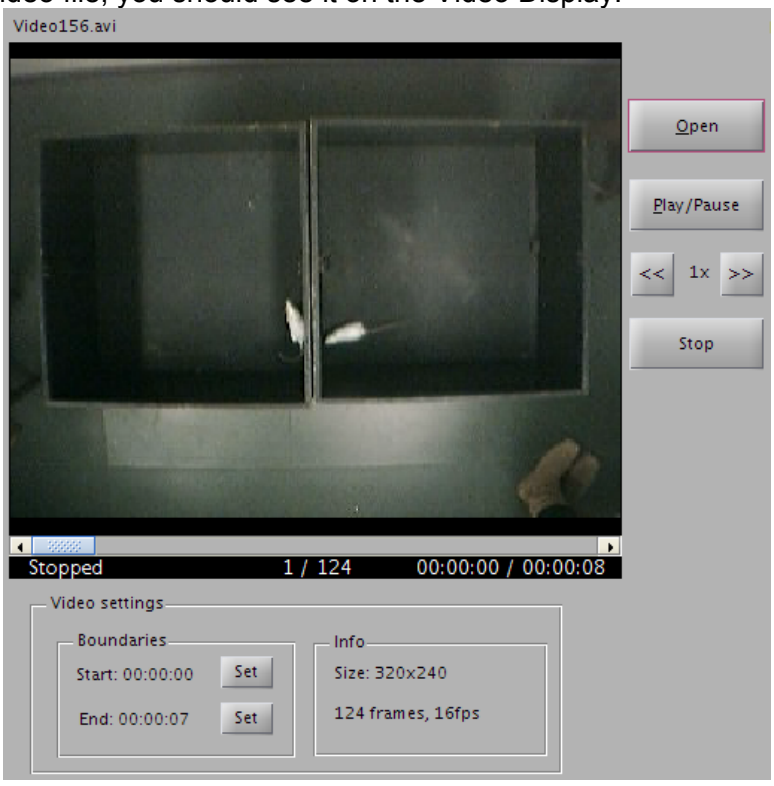
Opening a video file

Before tracking the animals' position, you first need to open the video file. If you don’t have a video file and just want to test the software, you can use the following example video: http://dl.dropbox.com/u/7028726/Video156.avi. To open the video, click "Open"; you should see the dialog below:



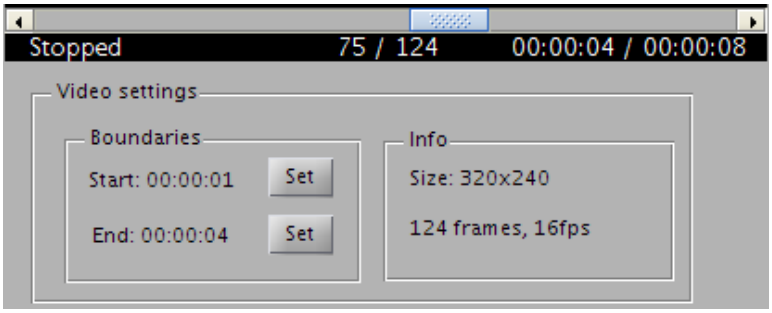
Using this dialog, use the mouse to select a supported video file and open it. The list of supported video formats and codecs can be found in Matlab webpage: http://www.mathworks.com/help/techdoc/ref/mmreaderclass.html.

After loading the video file, you should see it on the Video Display:



You can use the Video Controls (Play/Pause, <<, >>, Stop) to control the video playback parameters. The “Info” subpanel shows some metadata about the video file, such as frame size,

total number of frames and frame-rate. The “Boundaries” subpanel shows the current values for the start and end times to be analyzed. The default values are the start and end time of the video itself. To change the start time, pause the video at the desired start point, and then click on the button "Set" to the right of the start time label. The end time can be changed in a similar way. You should see the modified values instantly:



Tracking the animals' position

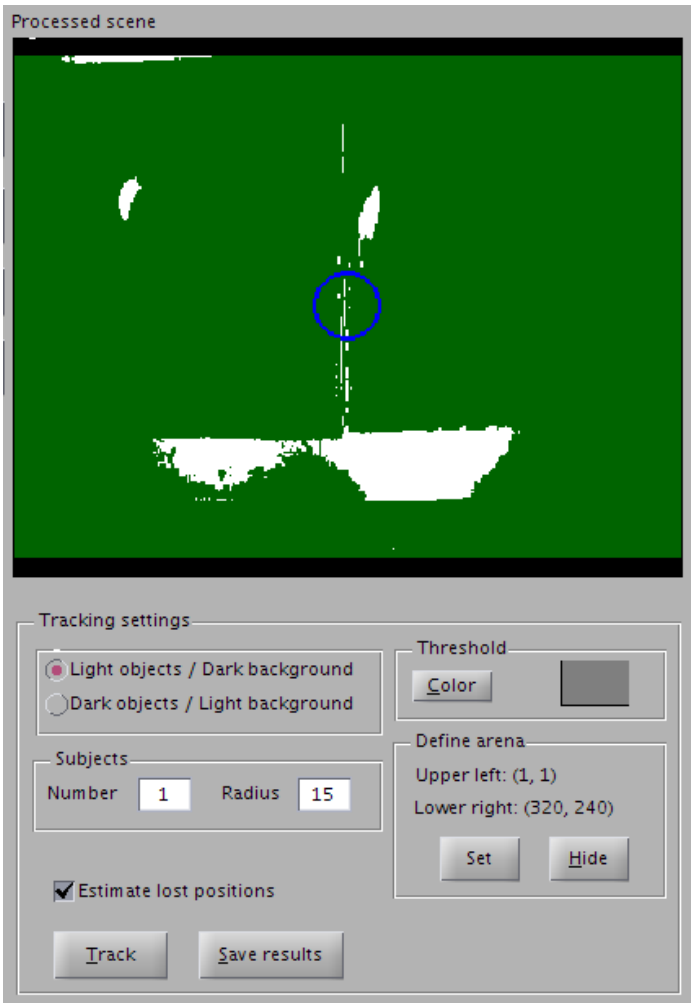
The algorithm we use to locate the animals on each frame is essentially the one specified in *J Neurosci Methods 2006 157:91-7*. We refer the reader to this paper for an in depth explanation of the algorithm and all its parameters.

Parameters

After loading the video and setting the start and end times, one should next choose the tracking parameters. When opening a video file for the first time, the software uses default values for the tracking parameters, which can be manually changed.

At any point the user can save a *settings file* containing the current parameter values (press Ctr+S or, alternatively, go to the “File” -> “Save Settings” menu). In this way, when working with another video file that requires similar tracking parameters, one can open the settings file (using Ctr+O or going to the “File” -> “Open Settings” menu) and the interface reloads the saved parameters. Another way to load tracking parameters is to have a settings file named `settings.mat` in the same directory as the video file being opened. The software recognizes this file and automatically loads the parameter settings.

The tracking parameters are found on the right side of the interface, under the "Tracking settings" panel:



They can be changed before starting to track the animals position and the changes will be reflected immediatly on the *Tracking Display*.

What follows is a brief explanation of all tracking parameters:

Animal / Background colors

Mouselabtracker works by differentiating between the color of the animals and the background color when one is significantly brighter than the other. The user has to inform whether the animals are lighter or darker than the background.

In order to analyze our test video (Video156.avi), keep the default option: “Light objects / Dark background”.

Number of animals

In the text field labeled “Number”, inside the “Subjects” grouping panel, the user can set the

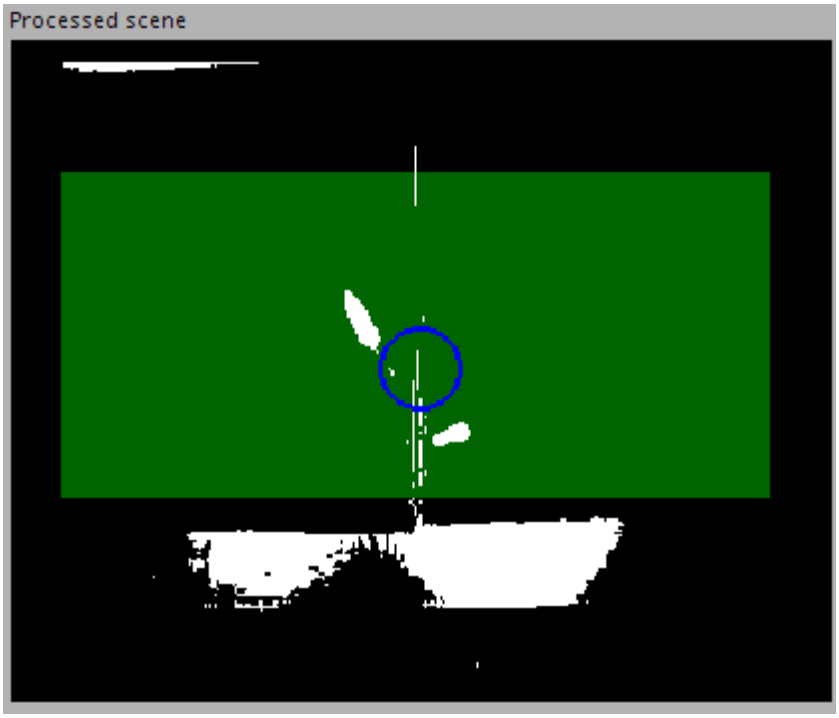
number of animals that will be tracked by the software. Note that, if desired, the user can track fewer animals than the total number of animals in the video.

Experiment arena

The user can set the points that define the corners of the experiment arena. Initially, the arena is defined as the whole rectangular area of the video. Constraining the arena space is useful if the lightning outside the arena has a brightness similar to the animals, which can interfere with the tracking results. By defining the experiment arena, the tracking algorithm will ignore the video portions outside it. Currently we only support a rectangular arena, but we plan to support other geometrical shapes in the future.

To define the arena, click in “Set”. The status bar will inform the user to click on the left video panel to define two points: the top-left and bottom-right corners of the rectangular arena.

When the user clicks on the bottom-right corner, the tracking parameters are updated and the right panel shows the new arena with a translucent green color.



Notice in the example video above that, after setting the corners of the video arena, the bright spots at the bottom of the video (the two white clusters of pixels at the bottom of the image) will

be ignored during tracking.

Threshold

The threshold color is the most important parameter of the algorithm. It determines, for each pixel of the experiment arena, whether the software considers it as part of the animal or part of the background. Pixels that are lighter than the threshold color (in grayscale) are considered as part of an animal (or the other way around if the user selects the “Dark subjects/Light background” option).

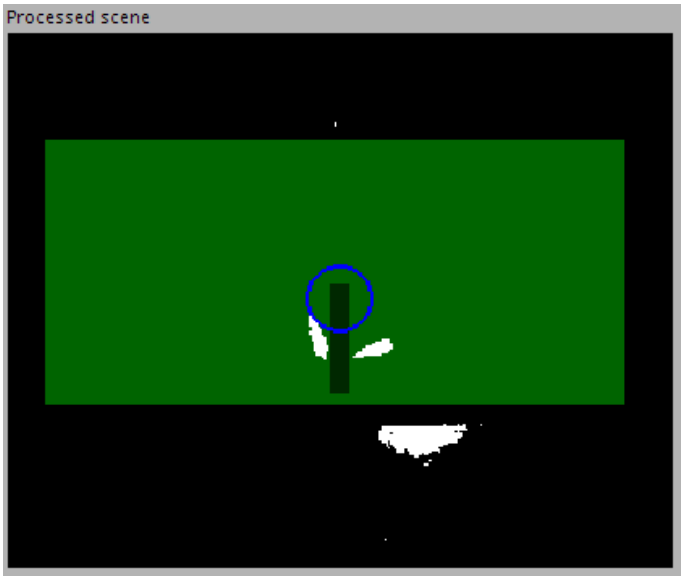
In the example shown in the previous figure, the white pixels represent what the software recognizes as animals and the dark pixels represent what is considered background; notice that there is significant noise in the scene, that is white pixels that don't actually correspond to the animals. This is an indication that the threshold could be improved. In this case, the threshold was set too low. By using a higher threshold (in this video, this means a lighter color), we can better separate the animals from the background, reducing the noise.

As you can see in the figure above, much of the noise that was apparent earlier was removed. At the same time, the white clusters representing the two animals became smaller and less well defined. The trade-offs involved in the threshold selection are discussed in the “Tracking Quality” section.

Hidden arena portions

In the example above, further increases in the threshold color will cause less pixels to be associated to each animal, that is, parts of the animals will be considered as background. This can potentially lead to spurious tracking results. To avoid this, the user can keep a threshold color value the properly tracks the animals and use the “Hide” function to tell the software to ignore specific background areas that are excessively light. After clicking the "Hide" button,

the user selects two points in the Tracking Display, the same way as when defining the arena corners. The selected rectangle will be painted black in the tracking display, and will be ignored during tracking (irrespective of the absence or presence of animals in this area).



In the figure above, the hidden area is represented by the rectangle between the two animals. This area will be ignored during the analysis.

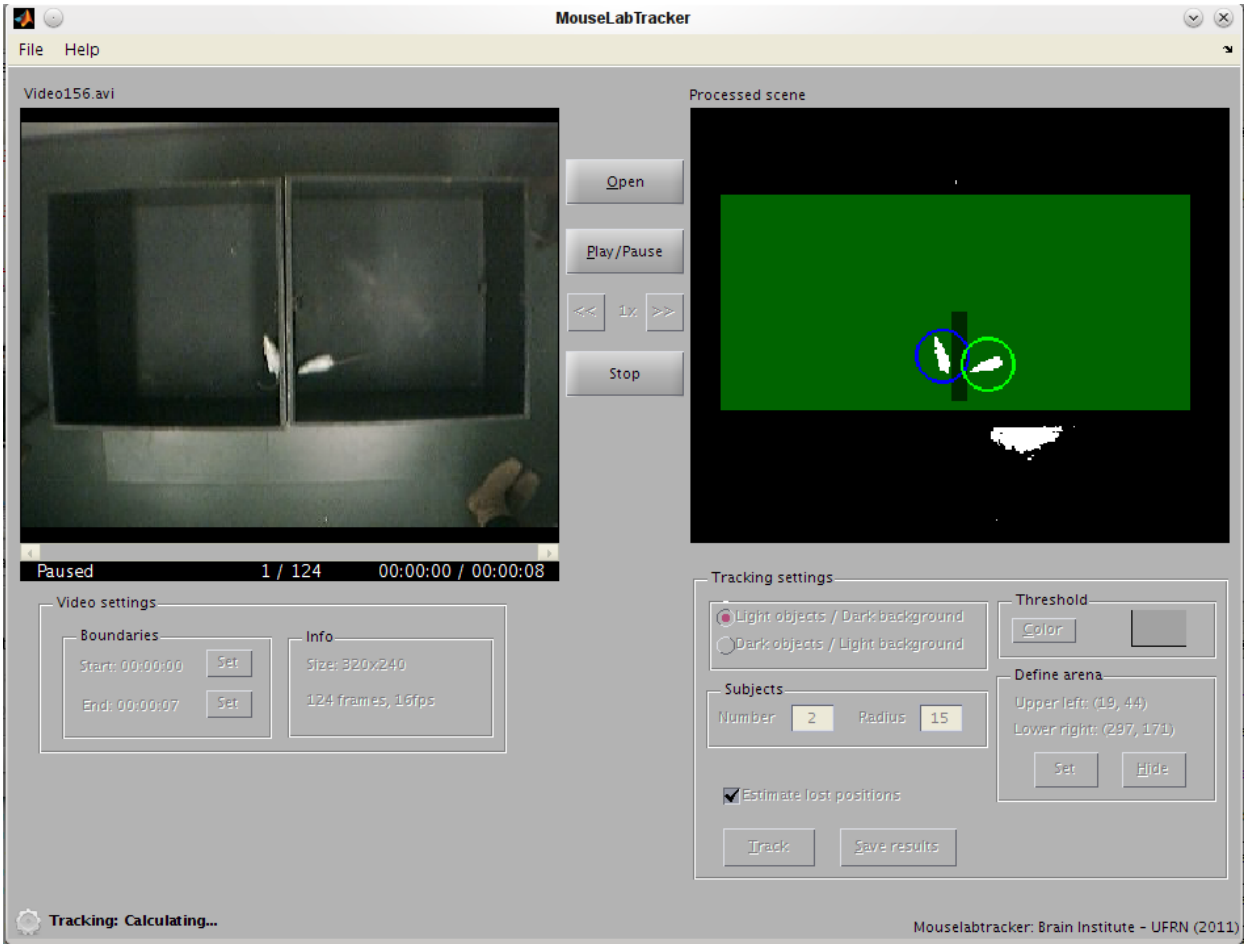
Radius

After defining the threshold color, the user can set the radius size used by the algorithm. That is, the radius of a circle centered on the current position of the animal. On each frame, the algorithm will search for the animal within the area defined by this circle in order to update its position (see J Neurosci Methods 2006 2006 157:91-7 for further details).

The current circle size is shown in the middle of the Tracking Panel, as can be seen in the previous figures. The circle should be large enough to cover the entire animal; in case of videos with multiple animals, the radius should be small enough so that the circles of different animals in different portions of the arena do not overlap.

Tracking

After setting the parameter values, the user can start tracking of the animals position by selecting the “Track” button (or Alt+T). The software asks for the initial position of all animals in the video; each position is manually informed by directly clicking on the current animal position in the Video Display. After setting the position of the last animal, tracking automatically starts:



As shown above, notice that the software disables several control components during tracking (such as the Tracking setting panels). When tracking ends, the software shows a pop-up dialog informing the elapsed time.

After tracking is completed, the user can use the video playback buttons to see the results in real-time in the Tracking Display. A color circle shows the calculated position of each animal in the current frame. The user can then click “Save Results” to create a Matlab file with the position results. If the visual inspection of the results reveals that the tracking was not properly performed, the user can redefine the parameter values and re-analyze the video.

Working with the results in Matlab

The user can load the results in Matlab from the generated .mat file (see: http://www.mathworks.com/help/techdoc/import_export/braietb-1.html for further instructions).

Inside the mat-file there is a *result* variable holding a structure, which contains general information about the video, the tracking parameters employed, and the calculated positions. An

example of a result structure for our test video is shown below:

```
>> result
result =

            video_name: 'Video156.avi'
     arena_upper_left: [19 44]
    arena_lower_right: [297 171]
       excluded_areas: [1x2x2 double]
      number_subjects: 2
     subject_is_white: 1
               radius: 15
    initial_positions: [2x2 double]
          start_frame: 1
            end_frame: 124
            threshold: 0.6366
             duration: 8.3125
           frame_rate: 16
            positions: [124x2x2 double]
```

The structure fields are:

- `video_name`: filename of the analyzed video
- `arena_upper_left, arena_lower_right`: top-left and bottom-right arena corners
- `excluded_areas`: list of the excluded areas from the arena (using the “Hide” button)
- `number_subjects`: number of analyzed animals
- `subject_is_white`: 1 if the animals are lighter than the background, 0 otherwise
- `radius`: radius size employed in the analysis
- `initial_positions`: initial positions for the animals, as selected by the user
    - `initial_positions(i,:)` is the `[x y]` initial position of animal `i`
- `start_frame, end_frame`: start and end times of the analysis
- `threshold`: threshold color selected by the user, normalized to the `[0,1]` range
- `duration, frame_rate`: total video duration and video frame sampling rate (in Hz)
- **`positions`**: calculated positions for all animals at each frame
    - `result.positions(j,:,i)` is the `[x y]` position of animal `i` in the j-th frame (with respect to the first analyzed frame).
    - `result.positions(:,:,i)` is an array containing all positions for animal `i`

Frequently asked questions

What are the supported video codecs?

Similar to: What codec should I use? My video opened with strange colors and video artifacts, what happened?

The supported codecs are determined by the local Matlab installation. We used the *mmreader* class for better compatibility with older Matlab versions. The list of supported codecs for each platform can be found at: http://www.mathworks.com/help/techdoc/ref/mmreaderclass.html

Why the Tracking Display isn't updated when I change the animal color option (light/ dark)?

Due to some technical restrictions of the Matlab platform relating to the GUI code, the interface can't be automatically updated when a radio button changes. Nevertheless, the function is still active. You can change it normally and when another parameter changes, the interface will update itself with the new animal color setting.

What happens if I try to load settings made for a previous video with a different size and duration?

The software will try to adapt the settings to the new video. For example, if the previous arena is too large for the current video, it will be clipped.

Does the software work with unusual arena formats or only rectangles?

You can emulate support for those arenas using the "Hide" option. Select a rectangle containing the entire structure for the arena. Later, you can use "Hide" to remove certain portions of the arena, leaving only the area with the desired shape. We note that defining the experiment area within a video is only required when the there are noise influences (e.g., light pixels) outside the arena. In principle any arena shape can be used if the background area outside the arena has no noise interference. For example, one can analyze a plus maze experiment using a rectangular area encompassing the whole maze.

Contact

To send suggestions or bug reports, please write to mouselabtracker@neuro.ufrn.br

Credits

The MouseLabTracker was developed by Giuliano Vilela, under the supervision of Prof. Adriano Tort. We thank Fábio Caixeta, Robson Teixeira and Cláudio Queiroz for alpha-testing the software. Please cite J Neurosci Methods 2006 157:91-7 in any research work that used this software.

This software was created at the Brain Institute at the Federal University of Rio Grande do Norte (UFRN), Natal, Brazil.

September 2011.