

epm-tracker documentation

Ross McKinney

2017-11-20

Contents

intro	1
dependencies	2
installation	2
launching the gui	2
using the gui	3
opening a video file	3
navigating through a video file	4
tracking a video file	4
setting an arena mask	4
setting the pixel threshold	5
setting the background image frame count	7
setting the save file	7
validating your tracking settings	7
tracking	7
analyzing your tracking data	7

intro

epm-tracker is a simple bit of software to track the position of a mouse in an elevated plus maze (though it could technically be used to track any dark object in almost any shape arena). It works by calculating a background image and subtracting this image off of every frame contained in a video file. Then, all pixels (1) below a user-defined threshold and (2) outside of a user-defined arena are removed from the image, and the remaining pixels are taken as objects of interest. The object of interest with the largest area is subsequently selected as the mouse, and its position is recorded.

dependencies

This software uses [ffmpeg](#) to convert between different video formats. Specifically, if any video format other than a [FlyMovieFormat \(.fmf\)](#) is selected, then ffmpeg will convert the video file to a .fmf video and save this video in the same location as the selected file.

This software also uses [conda](#) for package and environment management. Though not absolutely required, the setup (described below) will assume that conda is installed.

installation

- Note: I will describe how to install this software using the Anaconda Prompt, though any terminal should do.

First, download the epm-tracker .zip file and extract its contents to a memorable location. Then, open an Anaconda Prompt (on Windows, search for Anaconda Prompt from the Start Menu), and navigate into the epm-tracker folder. Finally, use conda to setup an environment that will contain all of the necessary python packages needed to use this software. To do this, type the following into the Anaconda Prompt:

```
conda env create -f environment.yml
```

This will use the environment.yml file (located in the epm-tracker folder) to setup an environment called “epm”. Once all of the required packages have finished downloading/installing, type the following into the Anaconda Prompt to activate the environment:

On Windows:

```
activate epm
```

On MacOS/Linux:

```
source activate epm
```

In parentheses, and to the left of the current directory within the Anaconda Prompt, you should see something like “(epm)”, this means that you have successfully activated the epm environment.

launching the gui

With the epm environment activated, you should be able to launch the GUI by typing the following into the Anaconda Prompt (this assumes that you are still in the epm-tracker folder):

```
python epm/main.py
```

using the gui

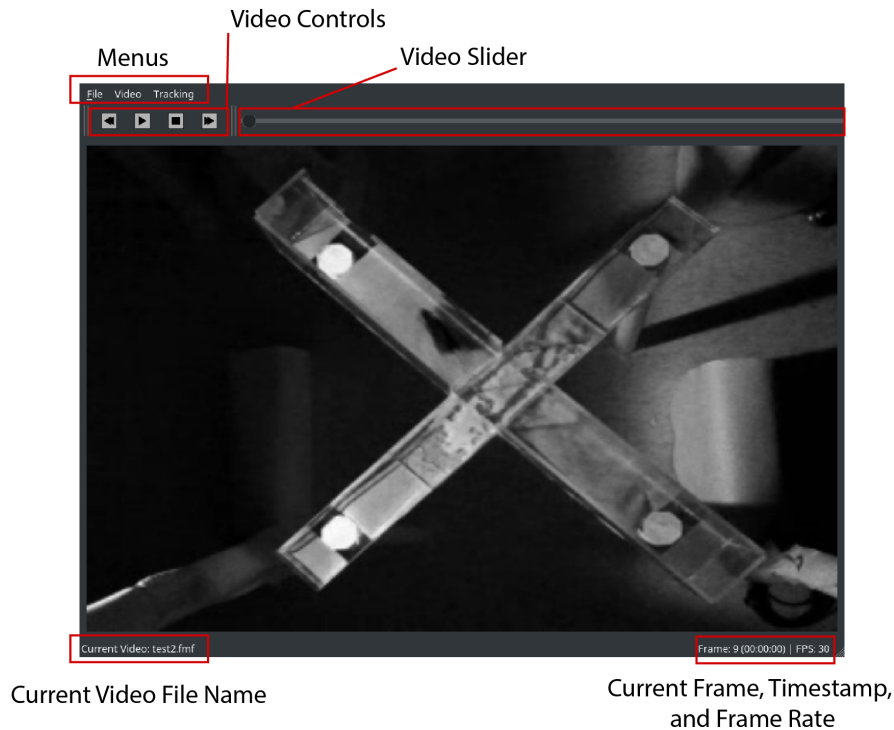


Figure 1: Main page of GUI. The main page consists of the media viewer, containing controls for navigating through the video and some simple video statistics.

opening a video file

The main window is used as a media viewer (See Figure 1). To open a video, go to File -> Open, then select the video file you want to view/track. Currently, you are allowed to select a video file with any of the following formats: **.fmf**, **.avi**, **.mp4**, **.mov**, and **.wmv**. If you select any file type other than a **.fmf** file, ffmpeg will launch within the Anaconda Prompt and convert the selected video to a **.fmf** video. You can monitor the progress of this conversion by looking at the output within the Anaconda Prompt. Following successful conversion, the

video will open in the media viewer. If you select a .fmf file, no video conversion will occur, and the video will automatically open in the media viewer.

navigating through a video file

To navigate through the video file, you can use the video control buttons or slider located above the current frame in the video. By pressing the following hotkeys, you can navigate through the video just using your keyboard:

1. “spacebar” -> toggle play/pause
2. “]” -> play
3. “[” -> pause
4. “.” -> next frame
5. “,” -> previous frame

tracking a video file

To track a video file, ensure that a video is currently open and visible in the media viewer. Then go to Tracking -> Track Video. This should open the Tracking Dialog window. Within the Tracking Dialog window, there are currently three main things that need to be set: (1) the arena mask, (2) the pixel threshold, and (3) the number of frames used to calculate a background image.

setting an arena mask

Within the first page of the Tracking Dialog window, you are asked to set an arena mask, which will be used to keep only those pixels (within each of the video frames) that fall within the EPM arena (See Figure 2).

The top three images in the “EPM Arena Mask” group box are as follows: (1) a raw image of the first frame of the selected video, (2) a binary image of the currently selected arena mask, and (3) an image of the arena mask placed on top of the raw image of the first frame of the selected video. Note that the white area in image (2) is retained, whereas the black area is discarded. Images (2) and (3) are initially black, and only show the specified arena when the user clicks the “Update Mask” button, described below.

The image just below the “EPM Arena Mask” group box is where the user specifies the arena. Use your mouse cursor to move each of the colored circles to one of the corners on the EPM. It shouldn’t matter which dot is placed at which corner, though I haven’t fully tested this yet. **Be careful to place the center of each dot within the image, placing them outside of the image will likely result in a bug that I haven’t dealt with yet.** Once you are satisfied with the position of each of the colored circles, hit the “Update Mask” button. You should see that images (2) and (3) in the “EPM Arena Mask” group box

are updated. If you like the generated mask, you can proceed to the next page, or save the mask for later use, by hitting the “Save Mask” button. This will save the locations of each colored circle in a .xlsx file. Similarly, you can load a mask by hitting the “Load Mask” button, and selecting the appropriate .xlsx file.

If you do not set a mask, all image pixels will be used to find the mouse in the EPM.

Use the “>>” button to proceed to the next page in the Tracking Dialog window.

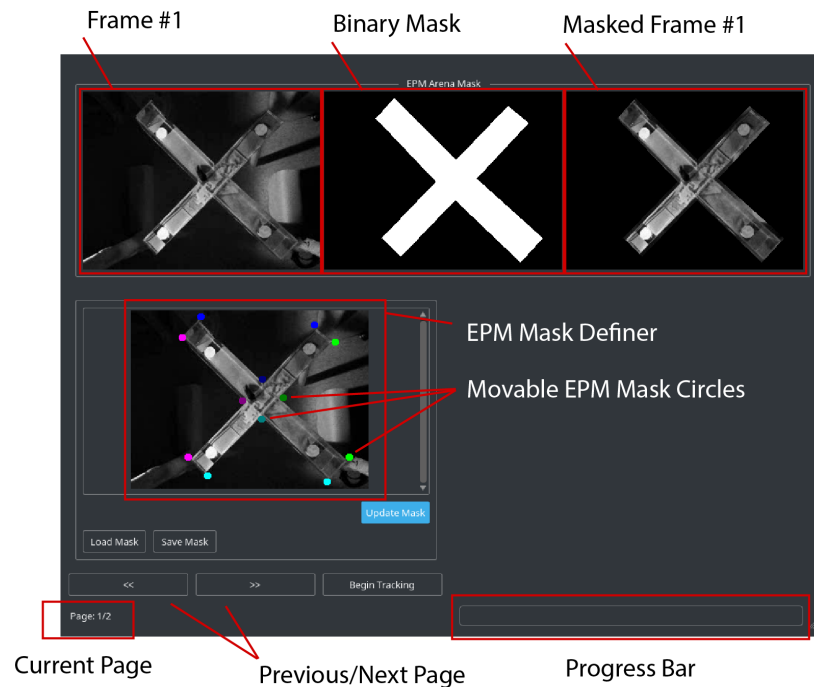


Figure 2: The first page of the Tracking Dialog window. This page allows the user to set a mask for the EPM arena by positioning small circles (“Movable EPM Mask Circles”) on the corners of the EPM. These circles do not have to be in any particular order. The “Binary Mask” and “Masked Frame #1” images are initially blank, and only display the binary mask and masked frame once the user hits the “Update Mask” button.

setting the pixel threshold

On the second page of the Tracking Dialog window, you will be asked to specify a threshold used to exclude pixels (See Figure 3). Any pixels below the currently

set threshold will be discarded during tracking.

Similar to the first page of the Tracking Dialog window, there are three images at the top of the page, in a group box labeled “Images”. (1) The first image is the calculated background image, (2) the second image is a raw image of the currently selected video frame, and (3) the third image is a background-subtracted and thresholded image of the currently selected video frame.

Using the “Tracking Settings” group box, the user can specify the tracking settings. Specifically, the spin box to the right of “Set threshold” can be changed, which will update the threshold to use for tracking. Following updating of the threshold, images (2) and (3) will also be updated to show you which pixels will be retained. The goal is to make the pixels representing the mouse as large as possible, while minimizing non-mouse pixels.

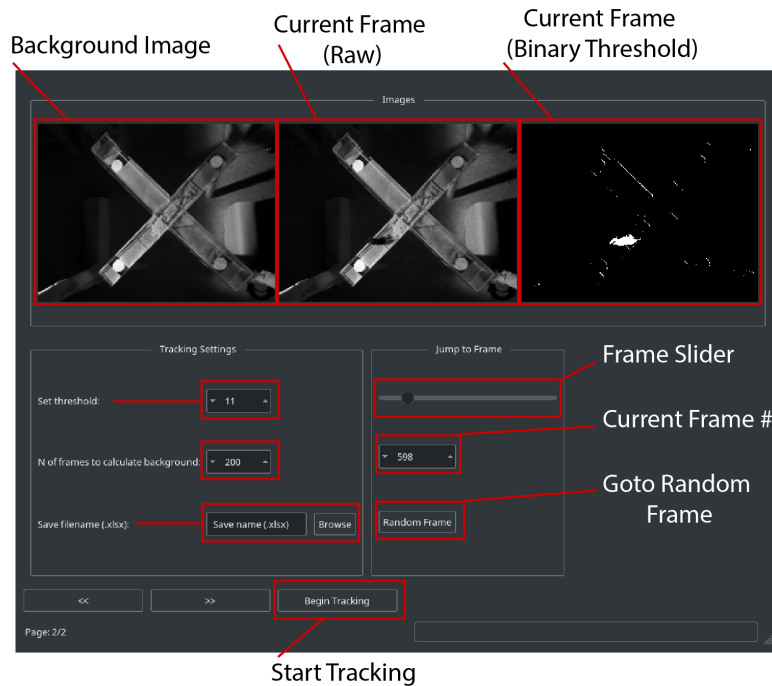


Figure 3: The second page of the Tracking Dialog window. This page allows the user to set a threshold to exclude pixels beneath a specified value. It also allows the user to set the number of frames used in calculating the background image. Finally, the user can set a save file name for the tracked mouse, and use the “Jump to Frame” group box to validate their settings.

setting the background image frame count

Just beneath the threshold spin box you can specify the number of frames to use in calculating the background image (“N frames to calculate background”). The background image is calculated by taking the mean image from N randomly selected frames contained in the video. The default value of 200 seems to work well in most situations.

setting the save file

Use the “Browse” button to the right of the “Save name (.xlsx)” text edit to select a location to save the tracked file. If you forget to specify a save file, one will automatically be generated and saved for you in your home directory.

validating your tracking settings

You can use the slider, spinbox, and “Random Frame” button in the “Jump to Frame” group box to navigate through each of the frames in the video being tracked. This is useful to make sure that your selected settings will work for the majority of the frames contained in the video. Note that if there are large fluctuations in lighting conditions (such as shadows appearing at certain points in the video), the selected settings for those frames will likely not work very well. Choose settings that work for the majority of the frames contained in the video to get the best results.

tracking

Once you have specified all of the tracking settings, hit the “Begin Tracking” button to start tracking. The tracking progress is monitored via the progress bar on the bottom right side of the Tracking Dialog window. Once tracking has completed, the Tracking Dialog window will disappear, and the tracked video will be displayed in the media viewer.

analyzing your tracking data

All of the tracking data is saved into a .xlsx that contains the following columns: (1) frame, (2) rr, (3) cc, (4) area, (5) maj, (6) min. Each row represents the data from a single tracked frame, and any missing data is represented as a “NA” value.

During tracking, the mouse is identified as the largest blob of connected pixels. Subsequently, this “blob” is modeled as an ellipse, which can be defined by its centroid (center of mass), and major and minor axis lengths (orientation of the

ellipse should also be included in this). The centroid position (x, y) is saved in pixel coordinates as (cc, rr) , which are columns (3) and (2) in the .xlsx file. To plot the “true” position of the mouse in the video, you should plot cc versus $-rr$ (this is because pixel coordinates of an image have an inverted y -axis relative to the normal Cartesian system). The major and minor axis lengths of the ellipse fitted to the blob are saved in columns (5) and (6), and the area of the ellipse fitted to the blob is saved in column (4).

Note that all of these values are currently in pixel units. Therefore, to compare between videos, you should ensure that you are not changing the camera’s position or any of the camera’s settings (such as zoom) between recordings.