

# RatTracker: A software platform for the Conditioned place preference test



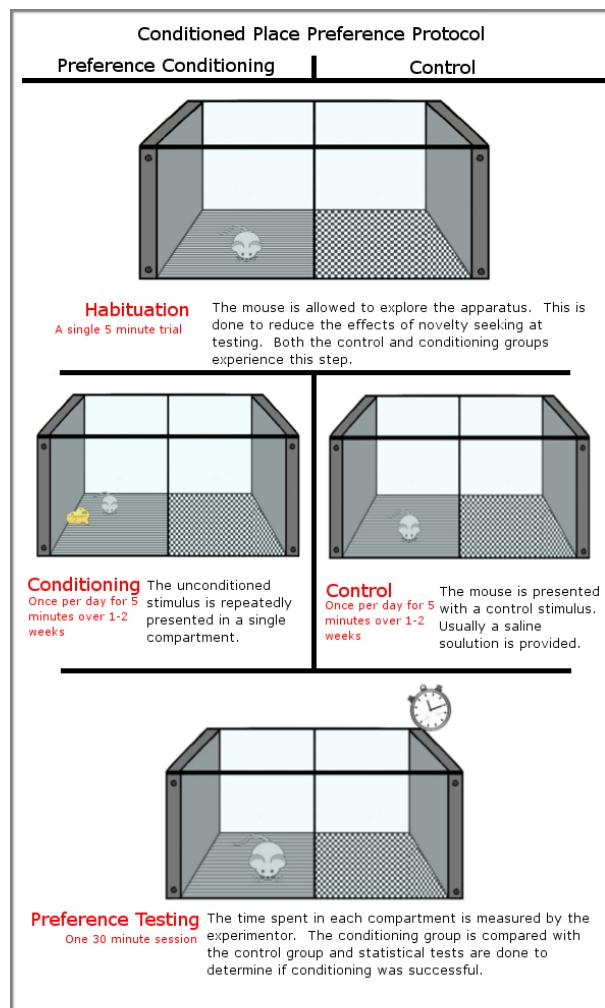
*Zahra Sadeghi Adl, Fatemeh Karimkhani*

Winter 2018

# Introduction

Conditioned place preference (CPP) is a form of Pavlovian conditioning used to measure the motivational effects of objects or experiences. By measuring the amount of time an animal spends in an area that has been associated with a stimulus, researchers can infer the animal's liking for the stimulus.

One of the challenges is calculation the time; manual measuring is too hard and some times impossible because in every simple research, there are about one hundred tests at least. So a platform for measuring this time with the help of computer vision can help researchers dramatically. In this project, we decide to develop a software platform that can automatically detect areas and measure the time spent in each of them. In this project, we use the techniques learned in computer vision course to develop a platform to identify the areas and the time that rats spent in each area. It identifies areas irrespective of angle, size, or number, and then by detecting rats and tracking them reports the time of presence in each area with 3 milliseconds accuracy.



# Problem assumptions

The video file being processed in this project contains the arbitrary number of boxes that we see in Figure 1. Each box has three areas:

1. Barberry or orange area (corridor)
2. Black area
3. Check pattern area

There is a rat in each box. They can move to the other two areas (black and Check) through the corridor area (colored). The gap between the black area and the chessboard is covered with walls and the rat can not move between the two areas.

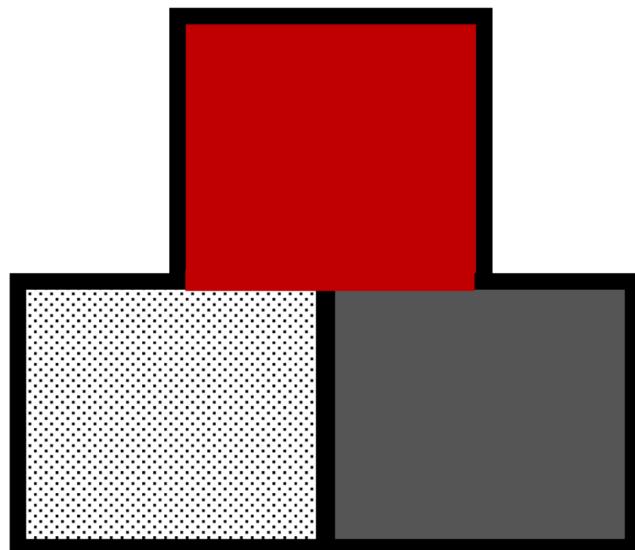


Figure 1: Schematic image of boxes from top view. Each box has three areas of different colors. The corridor area can be barberry or orange and the other two areas are black and Check pattern (black and white). Boxes can be placed in any direction.

# The purpose

The ultimate goal of the problem is to find out how long each rat has spent in different areas. For this purpose, after finding the number of boxes (which is equal to the number of rats), we must report three numbers per rat/box, which represent the time which the rat spent in three zones (approximately 3 milliseconds accuracy).

# Methodology

In order to simplify the solution, we break down the problem into a number of subproblems:

- A. Background Extraction: Finding the background is achieved through the median filter on different video frames. The resulting image (background) is used to identify areas as well as to detect image changes.
- B. Detect the Boxes and Areas: After finding the background of the image, we use it to identify the boxes and their areas (three areas per box).
- C. Detect Image Changes: In order to find the position of the rat, we use the weighted mean of the image changes relative to the previous frame and the background image changes.
- D. Detect the position of the rat: Given the boundaries of the boxes and their areas and also the amount of changes in each area, we can determine the position of the mouse (the area where it is located).
- E. Determine the duration of the rat in each area : Given the number of frames which the rat was in a given area and also the number of frames per second (generally 30 frames per second) we can determine the duration of each area we can calculate the time spent in the three zones.

Below we describe each of these subproblems in detail.

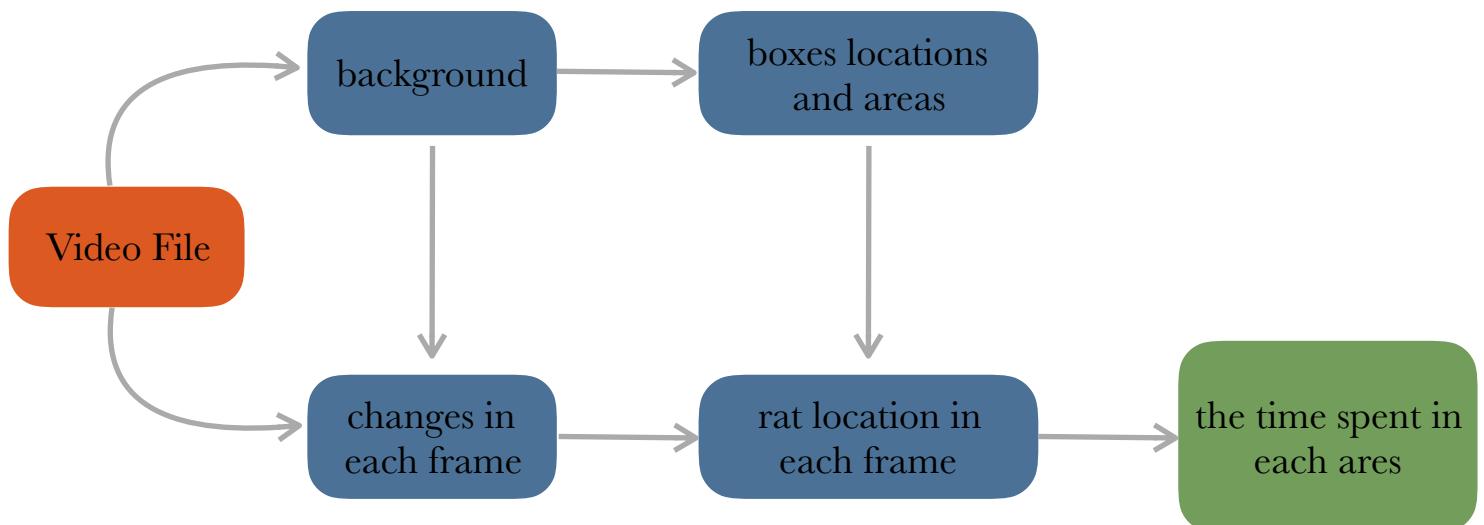


Figure 2: subproblems diagram

## A. Background Extraction

We use a median filter to extract the background. To prevent the runtime from prolonging, we use 1 frame for every 10 frames to calculate the median. So if the movie is 30 frames per second, 3 frames per second is used to calculate the median. We also use a Gaussian filter with a 3-in-3 kernel to eliminate the noise. Figure 3 shows one frame of a video that contains 4 rats and 4 boxes. Figure 4 shows the background image of a film containing approximately 2300 frames.



Figure 3: One of the input video frames

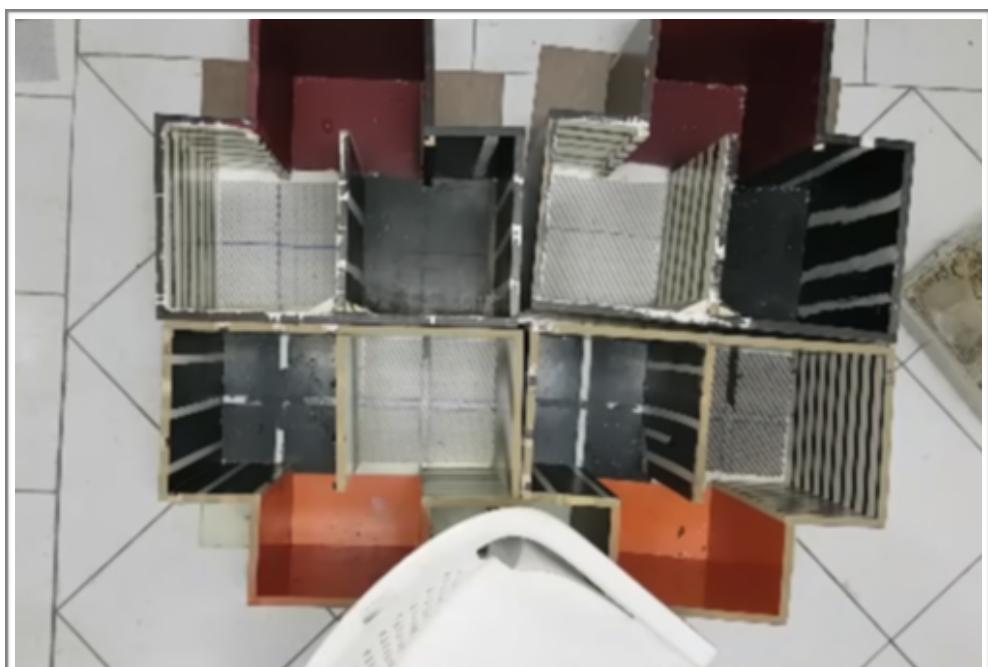


Figure 4: Background obtained using approximately 230 frames (video containing 2300 frames).

## B. Detect the Boxes and Areas

To detect the area in general, we use the different color characteristics of each area. The detection of areas is as follows:

1. Filter by areas colors
2. Run the open and closed operator
3. Convert shape to convex shape

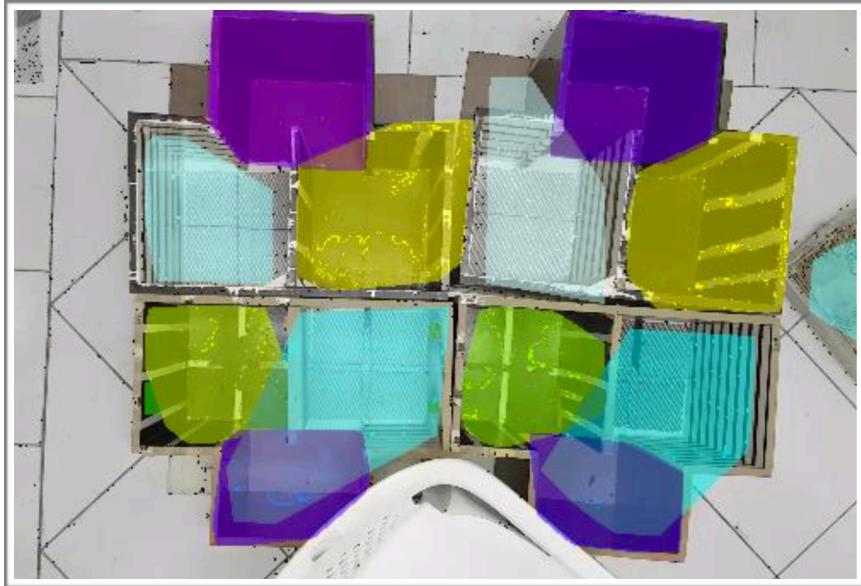
The main challenge here is choosing the right color. Both black and red are easily identified by hsv. But the color of the white area is not recognizable because of the background. So by identifying the background and then subtracting it from the image, the overall shape of the image is as figure 5. Then, by subtracting this image from the area image, we reach the white area shown in figure 6. This way we will have each of the three areas (figure 7).



Figure 5:



Figure 6: Filter by areas colors

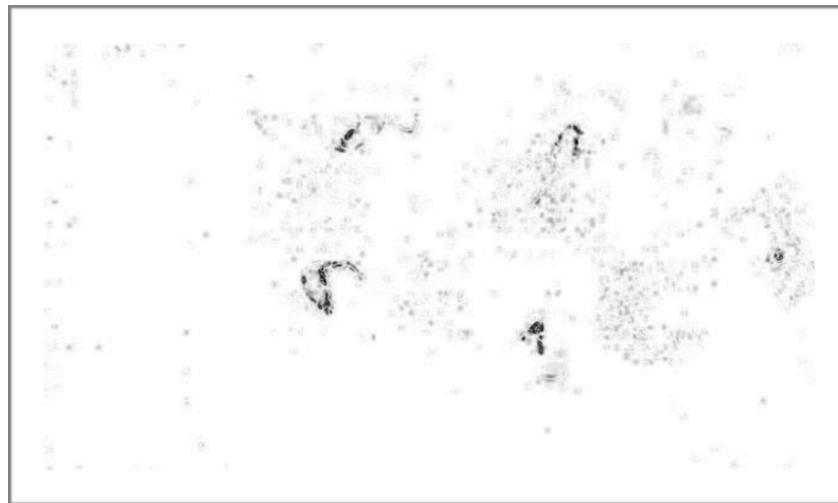


**Figure 7: Detected areas**

### *C. Detect Image Changes*

To find the position of the rat in each frame, we use two images:

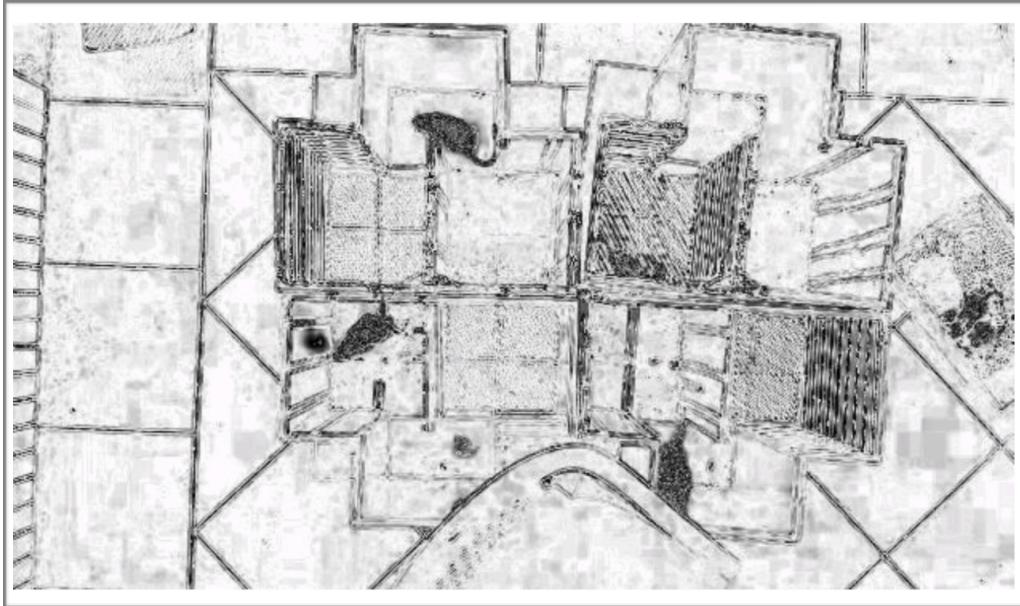
1. The Frame Difference With Previous Frame (Fig. 8): In this image, the movement of the rats is well seen. but if They do not have movement, it is can not be detected with this method.



**Figure 8: The Frame Difference With Previous Frame**

2. Frame difference with background (Fig. 9): In this image the rats are well seen. But because the difference in background image (especially across borders) in the differentiation of mice is challenged by the rest of the arbitrary changes.

We use their weighted averages (coefficients of 0.7 and 0.3, respectively) to take advantage of both the "previous frame difference" and "background difference" images. We see this average image in Figure 9.



**Figure 9: Frame difference with background**

Finally, apply the following steps to the average difference image:

1. Erosion with a rectangular structuring element of size 3x3
2. Threshold number 80
3. Dilation with a rectangular structuring element of size 3x3

The final image is then used to identify the rat location in the next step.



**Figure 10: Weighted Average of Two Images "Frame-by-Frame Difference" with a Factor of 0.7 and "Frame-by-Frame Difference" with a Factor of 0.3**

#### *D. Determine the duration of the rat in each area*

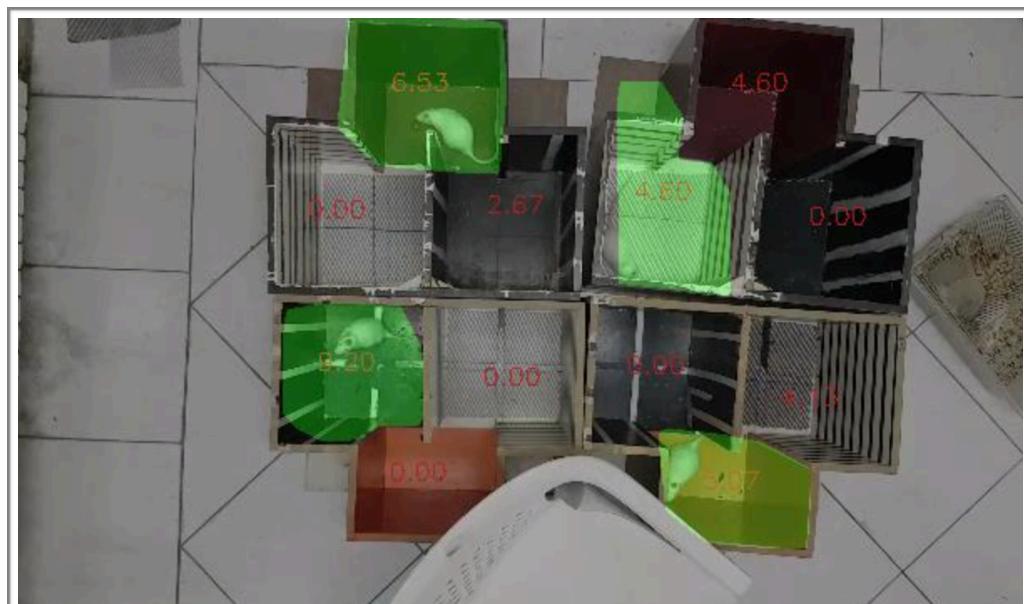
At this point, we have to determine, for each box, its rat is in which of the three regions. For this purpose, we use the extracted areas for each box as well as the image variations (see Section C). The area where has the most changes (the sum of the changes in pixels of the area) is represented as the select area in this frame. However, since this selection area has a jump (erroneous changes), if these conditions are met, declare the previous frame selection as the current frame selection area:

1. If the sum of the changes in the selected region is less than Epsilon (this condition helps to eliminate the effect of noise).
2. If the distance of the current location from the previous location is greater than the speed imagined for rats.

Given these conditions, we come to an acceptable result.

#### *E. Determine the duration of the rat in each area*

To calculate how long each rat was stationed in regions, we divide the number of frames by the number of frames per second (generally 30). Figure 11 shows an example of the processed frame. In this image, the green areas show the current locations of the rats, and in each area, the time that rats have spent in them is shown.



**Figure 11: Current area of each rat (in green) and the time that each rat spent in areas (in red)**