Psychtoolbox

Mahdi Tabatabaei

Neuroscience Lab

Javad Khodadoost

March 20, 2024

Contents

Visual Search

For most animals, survival depends on rapid detection of rewarding objects, but search for an object surrounded by many others is known to be difficult and time consuming. However, there is neuronal evidence for robust and rapid differentiation of objects based on their reward history in primates (Hikosaka et al., 2014).We hypothesized that such robust coding should support efficient search for high-value objects, similar to a pop-out mechanism. To test this hypothesis, we let subjects view a large number of complex objects with consistently biased rewards with variable training durations (1, 5, or 30 days). Following training, subjects searched for a high-value object (Good) among a variable number of low-value objects (Bad) (Ghazizadeh et al., 2016).

Search task consisted of target present and target absent trials that were intermixed with equal probability. A single Good object was present in target present trials while all objects were Bad in target absent trials.

Experiment Settings

For each group 48 fractals in total consist of:

- 24 for value group:
  - 12 Good
  - 12 Bad
- 24 for perceptual group:
  - 12 Good
  - 12 Bad

Search phase

Subjects have to find the sole high-value objects among low-value distractors in each target present (TP) trial and reject the target absent (TA) trials.

- 4 sessions for each subject
- Equal TP and TA trials for each good fractal
- Each session last about 12 minutes:
  - Half of the trials are TA and half TP
  - Half of the trials are from perceptual group and the other half from value group
  - 6 trials on average for each good fractal in each session (total of 144 trials per session)

Search Steps

1. Fixation at the center
2. Fixation point disappears, objects onset
3. Select an object or reject the trial
   - Select: fixating on the object and press space button

- Reject the trial: press X button

4. Display off
5. Delivering the corresponding reward
   - value group: 1 unit for bad, 3 unit for good
   - perceptual group: 1 unit for bad, 3 unit for good

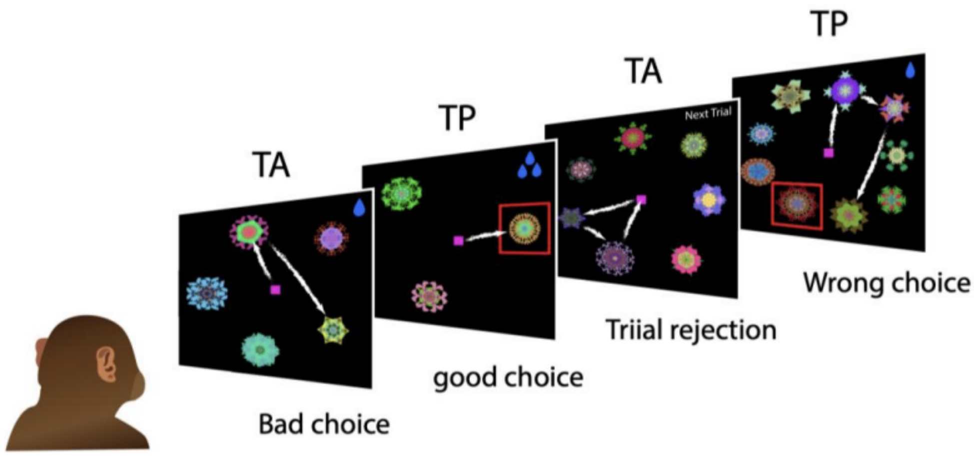

Figure 1: Task Paradigm

6. Go to the next trail (step 1)

Display Size (DS)

3, 5, 7, 9 random assignment

Rotation

1-360 deg random assignment

Each Trial

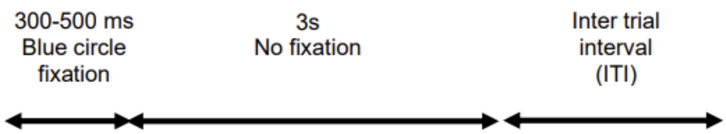

Figure 2: State Diagram

1. No press: error beep, 1.5s ITI
2. Reject press (TA): low reward delivery after each 2-4 random correct rejection
   - ITI for reward delivery: 1.5s
   - ITI for no reward delivery: 200ms
3. Reject press (TP): same as above(2) without reward

4. Accept press:
   - No fractal selection: error beep, 1.5s ITI
   - Fractal selection: associated reward delivery, 1.5s ITI

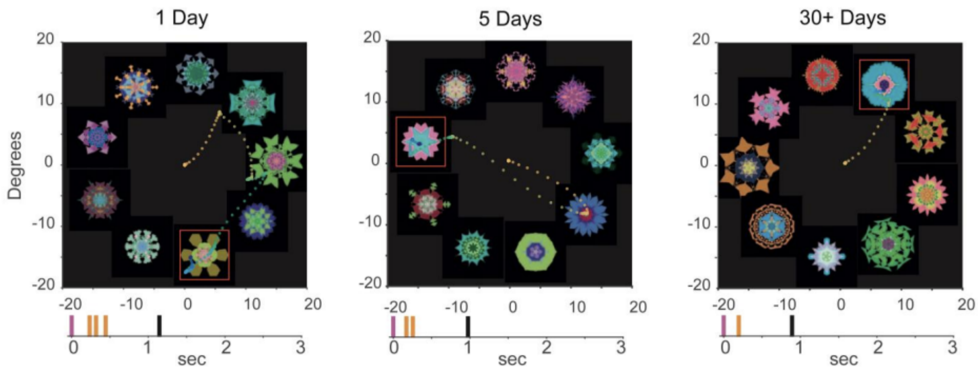

Figure 3: Example search performance of monkey after different training amounts. Eye position is shown by time-dependent color-coded dots (2/ms dot, from orange to blue). Red square indicates Good object (not shown to the monkey). Tick marks at bottom show the timings of saccades (orange) and reward (black) relative to display onset (purple).

Tasks

1. Draw state diagram for every state and condition.

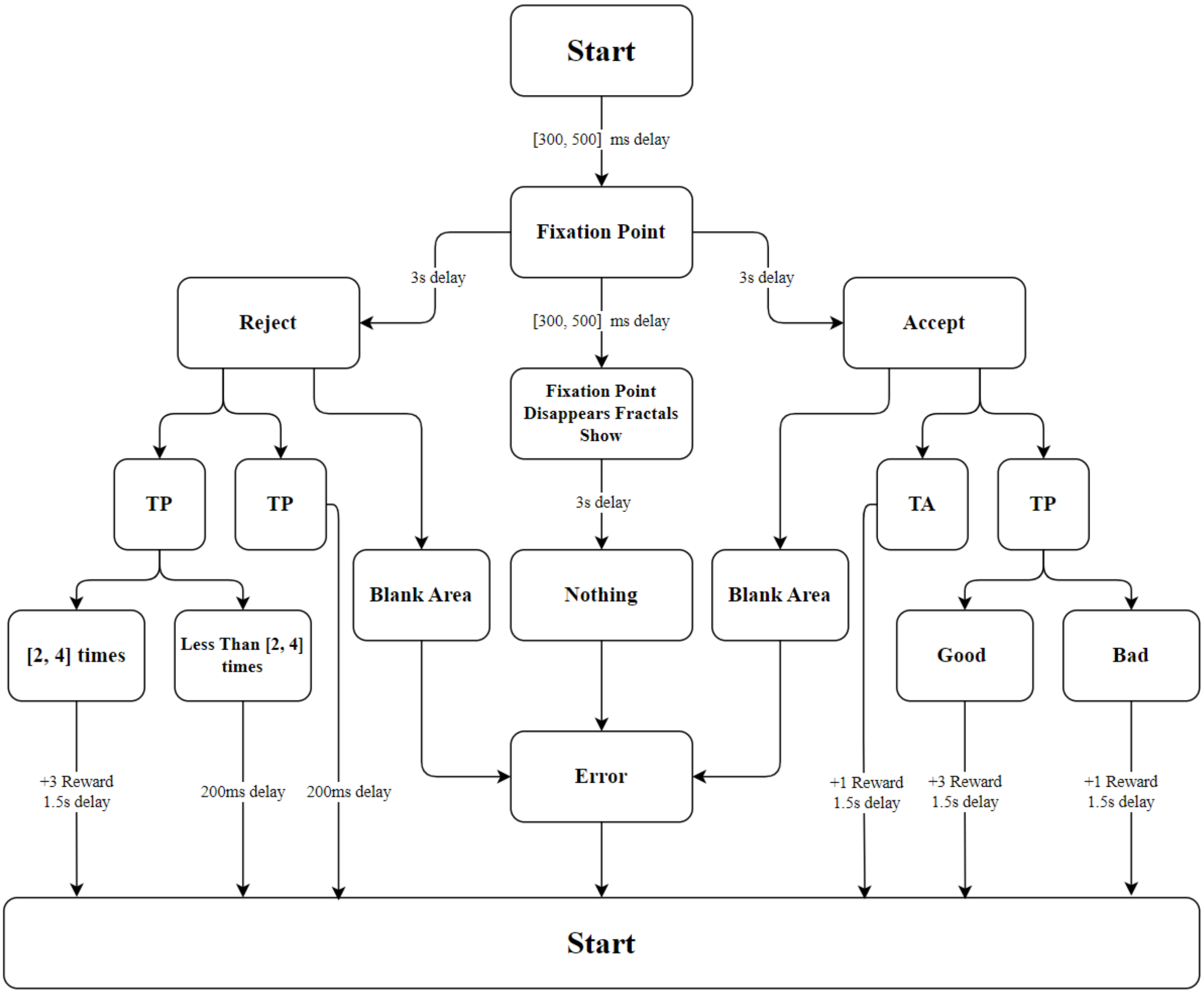

Figure 4: State Diagram

2. We want to use different combinations of fractals for each subject in a way that each fractal appears equally in each of the 4 aforementioned categories (Value/Perceptual, Good/Bad) in order to control the bias of low-level guiding features and recognition difficulties among fractal objects. Implement random assignment as described above. How many subjects do we need for equal appearance?

Soloution

As we have 2 categories (Good/Bad and Value/Perceptual), we need $2 \times 2 = 4$ subjects for each fractal. There are 48 fractals and we need $48 \times 4 = 192$ subjects to have all fractals in all 4 categories.

3. Fractals should appear equally at each conditions in all 144 Trials. Half of the trials are TA and half TP. Half of the trials are from perceptual group and the other half from value group. Fractals placement in each Trial should be equal and random according to several

conditions(TA/TP, Value/Perceptual and DS). Implement random assignment as described above.

Fractals Categorization

We can categorize the fractals in 4 groups:

Category	Vector Element
Perceptual - Good	1
Perceptual - Bad	2
Value - Good	3
Value - Bad	4

Now we need to put 12 numbers of each $1, 2, 3, 4$ in a vector and through permutation we can have a 48 elements vector which has a random order of these numbers.

```
% Permuting fractals
fractals_permutation = [ones(12,1); 2.*ones(12,1); 3.*ones(12,1); 4*ones(12,1)];
fractals_permutation = fractals_permutation(randperm(48));

% 4 fractal groups (Perceptual-Good=1, Perceptual-Bad=2, Value-Good=3, Value-Bad=4)
Perceptual_Good = find(fractals_permutation == 1);
Perceptual_Bad = find(fractals_permutation == 2);
Value_Good = find(fractals_permutation == 3);
Value_Bad = find(fractals_permutation == 4);
```

Source Code 1: Fractals Categorization

Trials Categorization

Now we need to find all groups which we can categorize trials to them.

- Target Present/ Target absent *
- Value / Perceptual *
- DS = 3, 5, 7, 9
- Initial Angle $= 0, 1, ..., 359$

By this categorization we have $2 \times 2 \times 4 \times 360 = 5760$ conditions for our trials. But we want to categorize our trials to 4 groups of 36 trials which are Perceptuals-TA, Percptual-TP, Value-TA, and Value-TP. We can have 4 matrix with $\frac{5760}{4} = 1440$ rows which are like below. After that we can permute matrices and then we choose first 36 rows of each matrix.

Category	Vector Element
Perceptual - TP	1
Perceptual - TA	2
Value - TP	3
Value - TA	4

rows	DS	Initial Angle
1	3	0
2	3	1
⋮	⋮	⋮
1440	9	359

We name metrices as VTP-Permutation, VTA-Permutaion, PTP-Permutation, and PTA-Permutation.

Assigning Fractals to different Trials

We know that in each TP trial which its equivalent matrices are VTP-Permutation and PTP-Permutation, has to exist one Good fractal. In another hand, each Good fractal has to exist in 3 TP trilas. To do this, we consider all 12 Value-Good and 12 Percptual-Good fractals and we permute a matrix which we had all 1 to 12 numbers in it. Then, we attribute the fractals corresponding to those numbers to the corresponding trials of their own group.
We do same work for Bad fractals but here we choosw $DS-1$ Bad fractals for TP trials and DS Bad fractals for TA trials.
Now we ad a colum to VTP-Permutation, VTA-Permutaion, PTP-Permutation, and PTA-Permutation which is fractal numbers. We permute order of fractal columns to have Good fractals in different places.
After all, we put all matrices which show our trials status in a Trial matrix and then we permute it.

rows	Value/Perceptual - TP/TA	DS	Initial Angle	Fractal Numbers
1	3	5	175	3,2,4,6,5,0,0,0,0
2	4	3	300	37,5,22,0,0,0,0,0,0
⋮	⋮	⋮	⋮	⋮
144	1	9	255	17,13,23,6,2,8,9,1,5

By having this matrix, we can use a for loop to show all trials. Zeros in fractal numbers exist because of size of matrices.

```
% Permuting trials
init_angle = 0:359;
DS = [3,5,7,9];

% 4 groups (Perceptual-TP=1, Perceptual-TA=2, Value-TP=3, Value-TA=4)
PTP_Permutation = zeros(1440, 2);
PTA_Permutation = zeros(1440, 2);
VTA_Permutation = zeros(1440, 2);
VTP_Permutation = zeros(1440, 2);

% Putting initial angle & DS in permutation
for i = 1:4
    PTP_Permutation(360*(i-1)+1:360*i,1) = DS(i).*ones(360,1);
    PTP_Permutation(360*(i-1)+1:360*i,2) = init_angle';
    PTA_Permutation(360*(i-1)+1:360*i,1) = DS(i).*ones(360,1);
    PTA_Permutation(360*(i-1)+1:360*i,2) = init_angle';
```

```
        VTA_Permutation(360*(i-1)+1:360*i,1) = DS(i).*ones(360,1);
        VTA_Permutation(360*(i-1)+1:360*i,2) = init_angle';
        VTP_Permutation(360*(i-1)+1:360*i,1) = DS(i).*ones(360,1);
        VTP_Permutation(360*(i-1)+1:360*i,2) = init_angle';
    end

    % Randomize the rows
    PTP_Permutation = PTP_Permutation(randperm(1440, 36),:);
    PTA_Permutation = PTA_Permutation(randperm(1440, 36),:);
    VTA_Permutation = VTA_Permutation(randperm(1440, 36),:);
    VTP_Permutation = VTP_Permutation(randperm(1440, 36),:);

    % To have a 36 elemenets of vector with 3 of each number among 1 to 12
    temp = zeros(36,1);
    for i = 1:12
        temp(3*(i-1)+1:3*i,1) = i;
    end

    % Add Good fractal to trial
    PTP_Permutation(:,3) = Perceptual_Good(temp(randperm(36)));
    VTP_Permutation(:,3) = Value_Good(temp(randperm(36)));

    % Add Bad fractal to trial
    for i = 1:36
        index = Perceptual_Bad(randperm(12))';
        PTP_Permutation(i,4:(PTP_Permutation(i,1)+2)) = index(1,1:(PTP_Permutation(i,1)-1));
        index = Value_Bad(randperm(12))';
        VTP_Permutation(i,4:(VTP_Permutation(i,1)+2)) = index(1,1:(VTP_Permutation(i,1)-1));
        index = Perceptual_Bad(randperm(12))';
        PTA_Permutation(i,3:(PTA_Permutation(i,1)+2)) = index(1,1:(PTA_Permutation(i,1)));
        index = Value_Bad(randperm(12))';
        VTA_Permutation(i,3:(VTA_Permutation(i,1)+2)) = index(1,1:(VTA_Permutation(i,1)));
    end

    % Define Trials matrix
    trials = zeros(144,12);
    for i = 1:4
        trials(36*(i-1)+1:36*i,1) = i;
    end

    % Add column to Trials matrix
    trials(0*36+1:1*36,2:12) = PTP_Permutation(1:36,:);
    trials(1*36+1:2*36,2:12) = PTA_Permutation(1:36,:);
    trials(2*36+1:3*36,2:12) = VTP_Permutation(1:36,:);
    trials(3*36+1:4*36,2:12) = VTA_Permutation(1:36,:);

    % Randomize the rows
    trials = trials(randperm(144),:);
```

Source Code 2: Trials Categorization

4. Implement Search Phase as described in Experiment structure. In the beginning of script we should collect subject ID and session number. Use mouse position to simulate eyes movement. Use keyboard for key pressing (Accept: Space, Reject: X). Draw green and red

rectangle for good and bad fractals as shown in Assignment2.mov. Prepare Recorded video of your final search task in the same way as `Assignment2.mov`. Recorded video should include occurrence of different situation (No press beeping, Reject press (TA), Reject press (TP), Accept press, No fractal selection).

Task Explaination

At first we have to inout the `Subject ID, Session Number, Distance from Monitor`. after that we run part1 in MATLAB code to save `fractals` and `trials`. After that we run part2 and the task will start.
In each trial, we can Accept the fractal with `space button` or `left click` and reject the fractal with `X button` or `right click`. Information box will appear after first trial.

5. Draw information box in some corner of recorded video for each trial. information box should contain key pressed, value/perceptual group and reward amount.

Video

The recorded video of the task is provided in this link.

6. Proper output data at the end of script as will describe below. Save it to a single .mat file.

- Subject ID, session number, fractal size in degree, peripheral circle degree, screen size
- At each trial: button pressed, fractals name, fractals position on display, DS, value/perceptual group, TA/TP, mouse position in time.

Results

The results will be saved after finishing the task in to the folder.

7. Explain the reason for selection different ITI in TA and TP? Looking at below paper would be helpful.
Ghazizadeh, A., Griggs, W., & Hikosaka, O. (2016). Object-finding skill created by repeated reward experience. Journal of Vision, 16(10):17, 1-13, doi:10.1167/16.10.17.

Soloution

Less ITI in Reject trials can help the monkey to acheive reward sooner and because of this, we can see this difference.

Bonus

Find proper tools (Matlab script or other tools) for real-time detection eye gaze position on the display using webcam. Replace mouse simulation with real eyes movement in the experiment and prepare Question 4 with real eyes data. Draw eye position as described in Fig3 in recorded video.