Article's Psychophysical Task Design Using Psychtoolbox in MATLAB

Ali KhosraviPour

Fundamentals of Neuroscince, Sharif University of Technology, Dr. Ebrahimpour, Dr. Kiani, January 2024

***Abstract*— This psychophysical task involves a visual categorization experiment utilizing MATLAB's Psychtoolbox. Participants are presented with images categorized into distinct body parts such as Head, Close Body, Far Body, and Medium Body. The task requires rapid and accurate identification of these categories while measuring reaction times. The stimuli consist of carefully selected images from different datasets, and participants' responses are recorded to assess accuracy in categorization. The experimental design incorporates fixation dots, masked image displays, and feedback mechanisms. Data collected includes accuracy rates for each category, overall accuracy, and reaction times. Each participant's data is uniquely saved with a timestamp, allowing for comprehensive analysis and comparison across sessions. The task serves as a valuable tool for investigating visual processing and categorization abilities in a controlled experimental setting.**

I. Introduction

Visual categorization is a fundamental cognitive process that enables individuals to rapidly interpret and understand complex visual scenes. In the field of psychophysics, understanding how humans categorize visual stimuli provides valuable insights into the underlying mechanisms of perception and cognition. This study introduces a psychophysical task designed to investigate the categorization abilities of participants when presented with images featuring distinct body parts, including Head, Close Body, Far Body, and Medium Body.

The task, implemented using MATLAB's Psychtoolbox, aims to unravel the intricate interplay between visual processing and categorization accuracy. Leveraging carefully curated images from diverse datasets, the experiment prompts participants to swiftly classify visual stimuli based on predefined categories. The inclusion of fixation dots, masked image displays, and feedback mechanisms ensures a controlled experimental environment, allowing for the precise measurement of participants' accuracy and reaction times.

Furthermore, the task incorporates a unique data storage approach, where each participant's results are saved with a timestamp, facilitating longitudinal analysis and comparison across multiple sessions. The combination of accurate categorization, reaction time measurements, and individualized data tracking positions this task as a powerful tool for investigating the nuanced aspects of visual categorization and cognitive processing within a psychophysical framework.

II. Dataset

I've created a folder and named it 'HW2_myDataset' and inside it I've created five more folders that are: 1. myDataset2_B 2. myDataset2_D 3. myDataset2_F 4. myDataset2_H 5. myDataset2_M, that are for close-body, distractors(non-animals), far-body, head, and medium-body, respectively. Inside the distractor folder, there are 600 images, and in others(animals) there are 150 images each.

III. Using Pyschtoolbox

Psychtoolbox is a specialized toolbox in MATLAB designed for experimental psychology and neuroscience research. Developed by experts, it offers a suite of functions for creating and conducting experiments involving visual and auditory stimuli, precise timing control, and participant response collection. Leveraging modern graphics hardware, Psychtoolbox enables researchers to design experiments with realistic stimuli and is widely used across disciplines for studying perception, attention, and cognitive processes. Its user-friendly interface and continuous updates make it an essential tool for researchers exploring human behavior in controlled experimental settings. Here, I've used this toolbox to create my psychophysics task.

IV. MATLAB Code

Inside the folder 'MATLAB_Code' of the uploaded zip, there are fourteen .m files that one of them is the main file and the others are the functions that I've defined. The main file name is 'myPsycoTask.m', and the functions are:

1. *calculateAccuracyPercentages.m*
2. *collectResponse.m*
3. *determineCategory.m*
4. *displayMaskedImage.m*
5. *drawFixationDot.m*
6. *loadAndDisplayImage.m*
7. *nhKeyResp.m*
8. *nhKeyRespInner.m*
9. *provideFeedback.m*
10. *saveParticipantData.m*
11. *setTrueAnswer.m*
12. *thankYouMessage.m*
13. *updateAccuracyCounters.m*

First let's explain the general idea of the task, and later I'll explain each of the function with details.

This code begins by initializing some variables such as `headNumber` to track the occurences of different body part categories in the task. After this, the Psychtoolbox setup is configured using functions like *PsychDefaultSetup* and *PsychImaging*. and the window properties, such as text size, style, and font (Helvetica), are set. The experiment involves presenting instructions to participants through the *DrawFormattedText* function, guiding them through phases such as focusing on a white dot, categorizing images as animals or non-animals, and initiating the task with a keypress. The paths for images of different body part categories are defined, and images are randomly selected for each category. Subsequently, participants' responses and reaction times are recorded as they categorize the presented images. The task incorporates visual elements like fixation dots, masked images, and feedback mechanisms to enhance experimental control. The code continuous with the calculation of accuracy percentages for different body part categories and overall accuracy. Moreover, participant data, including total accuracy, accuracy for each body part category, and reaction times, is then saved uniquely for each run. Finally, the experiment ends with a thank-you message displayed on the experimental window, awaiting a keypress before closing the window. Now let’s dive into the functions:

1. The first function, *calculateAccuracyPercentages.m*, takes counters and corresponding accuracies for different body part categories as inputs. It computes and returns the accuracy percentages for each category and the overall accuracy. The percentages are calculated by dividing the accuracy count by the total count for each category and multiplying by 100. The overall accuracy is determined as the average of the accuracies across all categories. The function then displays the computed accuracy percentages for each category and the overall accuracy in the command window. This provides a concise summary of the participant's performance in categorizing images of various body parts.
2. The second function, *collectResponse.m*, captures participant responses during a psychophysical task. It takes inputs such as the experimental window (window), key codes for Space and 'Q' (kSpace, kQ), arrays for storing participant and true answers (subjectAnswer, trueAnswer), the current trial index (i), and the category group (group). The function initializes variables, records the start time, and enters a loop until a valid response is collected. It checks for key presses, captures reaction time (RT), and determines the participant's binary response (0 for 'A' or animals, 1 for 'N' or non-animals). The function returns the updated arrays for participant answers and reaction times. This mechanism ensures accurate recording of participant responses in a time-sensitive manner during the experiment.
3. The third function, *determineCategory.m*, categorizes images based on their file names. It takes the image name (imageName) and counters for different body part categories (closeBodyNumber, farBodyNumber, headNumber, mediumBodyNumber) as inputs. The function initializes the group variable to 'nan' (not a number) and then checks the character at the 48th position in the image name. Depending on the character, it assigns a category ('Close', 'Far', 'Head', 'Medium') to the group variable and increments the corresponding counter. The function returns the determined category (group) and the updated counters for each body part category. This allows efficient tracking of the occurrences of different body part categories during the execution of the psychophysical task.
4. The fourth function, *displayMaskedImage.m*, presents masked images on a specified window during a psychophysical task. It takes inputs such as the experimental window (window), an image (img), a wait time before flipping the display (waitTimeBeforeFlip), and the duration for which the masked image is displayed (maskedDisplayTime). The function initiates the display by flipping the window, introducing a wait period, randomizes the order of image pixels, creates a texture with the randomized pixel order, and displays the masked image. The function concludes by flipping the window again and waiting for the specified duration. This process is essential for introducing masked stimuli in psychophysical experiments, contributing to visual perception and response time investigations.
5. The fifth function, *drawFixationDot.m*, displays a fixation dot on an experimental window during a psychophysical task. It takes inputs such as the experimental window (window) and the size of the dot in pixels (dotSizePix). The function sets up the blend function, determines the center of the window, defines a rectangular region for the dot, specifies the dot color (white), and fills the oval with the defined color. After flipping the window to display the dot, it introduces a brief waiting period of 0.5 seconds. This function is commonly used to present a fixation point, aiding in participant focus and alignment in psychophysical experiments.
6. The sixth function, *loadAndDisplayImage.m*, loads and displays a grayscale image on an experimental window during a psychophysical task. It takes inputs such as the image file path (imagePath) and the experimental window (window). The function reads the image, converts it to grayscale using rgb2gray, creates a texture from the grayscale image, draws the texture on the window, and flips the window to display the image. A brief waiting period of 0.02 seconds is then introduced. This function is commonly used to present visual stimuli in psychophysical experiments, ensuring precise timing and display control.
7. The seventh function, *nhKeyResp.m*, captures and waits for a key response during a psychophysical

task. It takes inputs such as keyboard index (KbIdx) and key codes for 'Q' and 'Space' (kQ, kSpace). The function initializes the key variable to 'nan' and enters a loop, continuously checking for key responses until either the 'Space' or 'Q' keys are pressed. It relies on an inner function, nhKeyRespInner, to handle the actual key press detection. This function ensures the experiment progresses only after the participant presses the specified keys.

8. The eighth function, *nhKeyRespInner.m*, is an inner function used to detect and capture key responses during a psychophysical task. It takes the keyboard index (KbIdx) as an input and ensures that all keys are released before proceeding. It then enters a loop, continuously checking for key presses. Once a key press is detected, it records the pressed key code and waits until the key is released before returning the key code. This function is crucial for accurately capturing participant responses and ensuring that the experiment progresses only when a key is pressed and released.
9. The nineth function, *provideFeedback.m*, displays feedback on the experimental window during a psychophysical task. It takes inputs such as the experimental window (window), participant's answer (subjectAnswer), true answer (trueAnswer), and the current trial index (i). The function checks if the participant's answer matches the true answer for the current trial. If they match, it displays a 'Correct Answer' message in green; otherwise, it shows a 'Wrong Answer' message in red. The cursor is hidden, and the window is flipped to present the feedback. This function is crucial for informing participants about the correctness of their responses during the experiment.
10. The tenth function, *saveParticipantData.m*, saves participant data, including accuracy and reaction time information, for a psychophysical task. It takes inputs such as overall accuracy (totalAcc), accuracies for different body part categories (accuracyCloseBody, accuracyFarBody, accuracyHead, accuracyMediumBody), and reaction times (RT). The function generates a timestamp based on the current date and time, creates a structure containing the participant's performance metrics, and saves it in a file named with the timestamp. The accuracy and reaction time data are stored separately, providing a unique identifier for each participant's dataset. This function is essential for recording and organizing participant data for further analysis.
11. The eleventh function, *setTrueAnswer.m*, determines the true answer for a given image based on its file name. It takes the image name (imageName) as input and initializes the trueAnswer variable to 0. It then checks the character at the 51st position in the image name, and if it is 'd', it sets trueAnswer to 1. This function is designed to assign a binary true answer (0 or 1) depending on a specific criterion in the image name.
12. The twelfth function, *thankYouMessage*.m, displays a thank-you message on an experimental window during a psychophysical task. It takes the experimental window (window) as input, creates a message thanking the participant, and presents it at the center of the window. The cursor is hidden, and the window is flipped to display the message. This function serves as a conclusion to the experiment, prompting participants to close the window and view their results after completing the task.
13. The last function, *updateAccuracyCounters.m*, adjusts accuracy counters based on the correctness of participant responses and the category of the presented image during a psychophysical task. It takes inputs such as participant answers (subjectAnswer), true answers (trueAnswer), the current trial index (i), category group (group), and existing accuracy counters for different body part categories (accuracyCloseBody, accuracyFarBody, accuracyHead, accuracyMediumBody). The function checks if the participant's answer matches the true answer and the category group, incrementing the corresponding accuracy counter accordingly. This function is crucial for tracking accuracy in each category throughout the experiment.

V. Reason Behind Masked Image

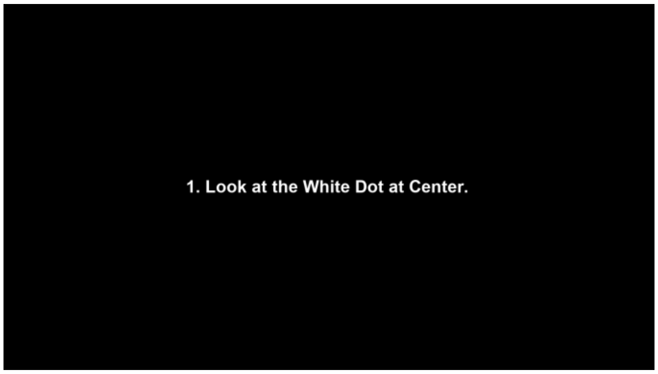
Showing a masked image between consecutive images in a psychophysical task serves several purposes:

- ***Minimizing Afterimages:*** Presenting a masked image helps minimize the persistence of afterimages from the previous stimulus. Afterimages can affect the perception of subsequent stimuli, potentially confounding the experimental results. The mask serves to reduce the impact of lingering visual effects.
- ***Controlling Timing:*** Introducing a masked interval helps control the timing of the experimental procedure. It provides a consistent temporal structure between trials, contributing to the precision of stimulus presentation and response collection. This is essential for maintaining experimental rigor and accuracy in psychophysical studies.
- **Reducing Carryover Effects:** Masked images contribute to minimizing any potential carryover effects from one trial to the next. This is crucial for ensuring that each trial is perceived independently, without the influence of the preceding stimulus.

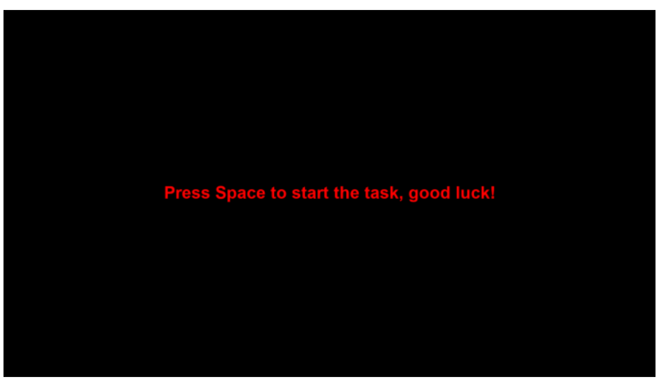
Therefore, showing a masked image between trials enhances experimental control, reduces potential biases introduced by visual adaptation, and ensures that each stimulus is perceived in isolation, contributing to the reliability and validity of psychophysical measurements.

VI. Different Windows of the Designed Task

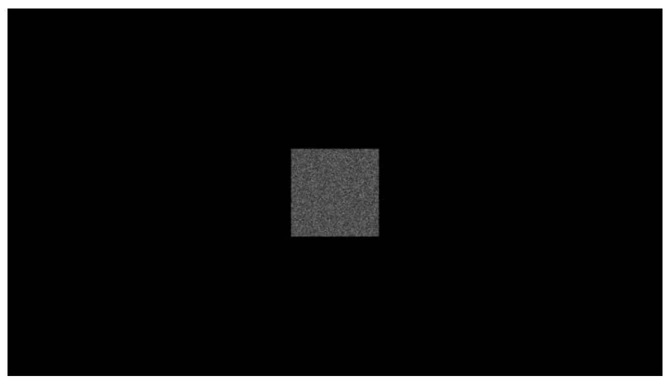
The first two windows that the participant will see are as follows that tells the instructions:



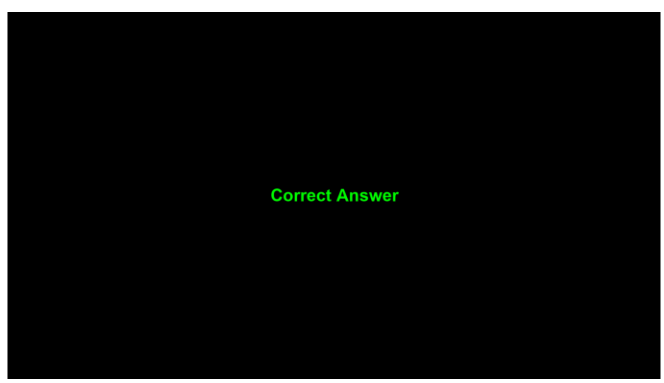
Then, for beginning the task, this window shows up:

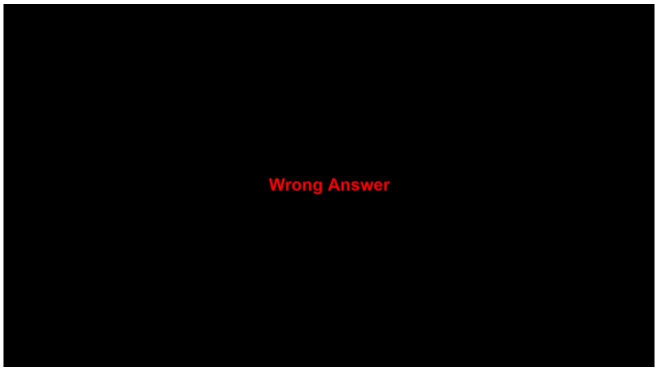


After pressing the Space key, the task starts with 75 images that we chose randomly in the previous parts (15 for head, 15 for close-body, 15 for medium body, 15 for far body, and 15 for distractors). First a white dot is shown and then the images are shown one by one each of them followed by a masked image. After showing each of images, the participant should decide whether choose to press A (for animals) or N (for non-animals) and their reaction time is measured.



Then a feedback based on their answer is shown to them:





VII. Results

After finishing the task, the accuracies for each category and the average reaction time of the participant are saved into .mat file. In order to avoid overwriting and having a unique file for each participant, I've used timestamp for naming the file. Example:

provideFeedback.m
RT20240203_023402.mat
saveParticipantData.m
setTrueAnswer.m
Subject20240203_023402.mat

The data stored in the first file:

subject | RT

1x1 struct with 6 fields

Field	Value
TotalAccuracy	88.3333
CloseAccuracy	100
FarAccuracy	53.3333
HeadAccuracy	100
MediumAccuracy	100
AvgRT	2.0000e-05

Participant 1

subject | RT

1x1 struct with 6 fields

Field	Value
TotalAccuracy	85
CloseAccuracy	93.3333
FarAccuracy	73.3333
HeadAccuracy	86.6667
MediumAccuracy	86.6667
AvgRT	0.0042

Participant 2

Field	Value
TotalAccuracy	86.6667
CloseAccuracy	93.3333
FarAccuracy	80
HeadAccuracy	100
MediumAccuracy	73.3333
AvgRT	2.0656e-05

Participant 3

The results revealed varying accuracies across different categories in the psychophysical task. Participants demonstrated high accuracy in identifying images from the "Head" category. In contrast, accuracies were slightly lower for the "Medium Body" and "Close Body" categories. The "Far Body" category exhibited the lowest average accuracy.
Analyzing the relationship between accuracy and reaction time (RT) showed interesting patterns. Participants who correctly identified images from the "Head" category tended to have shorter reaction times, suggesting a potential speed-accuracy trade-off. Conversely, for the "Medium Body" and "Close Body" categories, higher accuracies were associated with slightly longer reaction times, indicating a more deliberate processing approach.
Notably, the "Far Body" category showed a unique pattern, with participants displaying quicker responses despite lower accuracy. This could suggest a faster but less accurate processing strategy for stimuli in this category.

VIII. Conclusion

The psychophysical task conducted in this study aimed to investigate participants' visual categorization performance across different body part categories. The results unveiled distinct patterns of accuracy, highlighting participants' proficiency in recognizing "Head" stimuli compared to "Medium Body," "Close Body," and "Far Body" categories. The differences in accuracy indicate that our minds use different thinking approaches for various pictures.
Additionally, the analysis of reaction times (RT) provided valuable insights into the temporal dynamics of participants' responses. The relationship between accuracy and RT revealed intriguing patterns, indicating potential trade-offs between speed and precision in category identification. Notably, faster responses were associated with higher accuracy in the "Head" category, while slightly longer reaction times accompanied improved accuracy for "Medium Body" and "Close Body" stimuli.
Overall, this task provides valuable insights into the interplay between accuracy and reaction time in the context of visual categorization, advancing our understanding of human perceptual abilities and informing future research in the field.

IX. References

[1] http://psychtoolbox.org/download
[2] http://psychtoolbox.org/docs/SetupPsychtoolbox
[3] https://github.com/Psychtoolbox-3/Psychtoolbox-3/blob/master/Psychtoolbox/SetupPsychtoolbox.m