# Continuous Psychophysics with Eye Tracking: A MATLAB-Based Program for Estimation of Contrast Sensitivity and Assessing Refractive Errors

**Ethan Pirso**

## Overview

The Continuous Psychophysics with Eye Tracking (CPET) program is a MATLAB-based software tool that enables the conduct of eye-tracking experiments within the field of continuous psychophysics. The program interfaces with the EyeLink eye tracking system and uses the integrated PsychToolbox-3 toolbox in MATLAB for controlling visual stimuli.

CPET's primary function is to monitor and evaluate a subject's ability to track a moving visual stimulus. It determines a contrast sensitivity threshold based on the tracking errors, which is used to estimate the subject's refractive error. Through repeated trials using varied spherical lenses, a refractive series can be established from the contrast threshold measurements, enabling the estimation of the subject's refractive errors. The CPET program thus stands as a tool for vision researchers interested in understanding the relationship between refractive errors, visual acuity, and gaze tracking performance.

While initially deployed with human subjects, CPET can be extended to non-human primate populations like the common marmoset, frequently used in vision research. Laboratory conditions often lead these primates to develop myopia, highlighting the necessity for refractive error measurements. This presents a valid use case for the CPET program.

# Hardware

The preliminary testing of the Continuous Psychophysics with Eye Tracking (CPET) program utilized a dedicated lab computer, running on Ubuntu 20.04.6 LTS[1] as the host machine. The program is designed to be compatible with other operating systems, including MacOS and Windows, as long as all program dependencies are met. This host machine executes the experimental program, which is written in MATLAB.

The host machine is connected to two monitors: one for the experimenter and a CRT monitor for the subject. The CRT monitor used in the setup is a Sony Trinitron Multiscan G400, 18", which operates at a refresh rate of 120 Hz. The monitor, powered by an NVIDIA GeForce GTX 960 graphics card, offers a resolution of 1,024 x 768 pixels. It is positioned 80 cm from the subject. This high framerate for stimulus presentation and the displacement of the monitor further than nearsighted vision are important considerations when measuring refractive errors in myopia.

Eye-tracking data is recorded using an EyeLink 1000 system from SR Research, Ottawa, Canada[2]. This system is used to examine the relationship between the subject's gaze position and the position of the stimulus. All eye-tracking has previously been performed in Desktop Remote mode, which allows the subject's head to move freely, however the program can be used in other EyeLink modes. To record eye-tracking data during the experiment, an EyeLink PC running on ROM-DOS is set up and connected to the host machine via an Ethernet connection. For further details pertaining to the EyeLink, refer to manual provided by SR Research.

---

[1] Download Ubuntu Desktop

[2] EyeLink 1000 User Manual

# Installation and setup

To begin the setup and installation process for the Continuous Psychophysics with Eye Tracking (CPET) program, it's critical to verify that MATLAB[3] and PsychToolbox-3[4] are correctly installed on your machine. Both of these are necessary for the proper functioning of the CPET program.

Next, the EyeLink Software Development Kit (SDK)[5] needs to be installed on the machine. This is required to interface with the EyeLink eye tracking system, a crucial component of the program.

Following these installations, move a copy of the CPET program folder into your chosen working directory. The location of this directory may vary based on your specific system configuration and personal preferences.

As a final step, it is advisable to verify the functionality of your setup. You can do this by running the demo script, gaborWalk.m, which is located in the DEMOS folder within the CPET program folder. This demo will allow you to confirm that PsychToolbox is functioning as expected.

Additionally, make sure to verify the EyeLink connection before initiating any experiments with the CPET program. Ensuring the EyeLink system is properly connected and functioning is crucial to collect accurate eye-tracking data during experiments.

# Contents

The CPET software is organized into various folders each with a specific role. Within the CODE folder, various MATLAB scripts perform different functions crucial to the functioning of the CPET software.

---

[3] Installation of MATLAB and Simulink

[4] Psychtoolbox-3

[5] EyeLink Developers Kit / API

Within each script, a docstring provides a detailed documentation of its functionality which should be reviewed, although here is a brief overview of the code;

1. calc_freq_cpd.m: Calculates the required 'freq' parameter for a Gabor patch in Psychtoolbox based on the desired frequency in cycles per degree (CPD), screen size in centimeters, screen size in pixels, and viewing distance in centimeters.

2. calc_ppd.m: Calculates 'pixels_per_degree' parameter used when setting the velocity of a curvilinear trajectory stimulus in visual degrees per second.

3. closeEyelink.m: Stops EyeLink data acquisition, closes the EDF file, downloads the output file, and shuts down the EyeLink system.

4. collectData.m: Collects eye gaze data from the Eyelink and updates stimulus position data.

5. drawStim.m: Draws the stimulus in the Psychtoolbox window and flips screen.

6. estimateError.m: Calculates position error in real-time and uses MAD as an early trial stopping criterion.

7. generateRefractiveSeries.m: Generates a refractive series for a subject based on a range of trials, plots the series, and identifies the 'best' diopter. Saves the plot and all data to files.

8. initEyelink.m: Initializes the Eyelink eye tracker and configures the calibration type.

9. initStim.m: Initializes the position and properties of the stimulus.

10. initWin.m: Sets up the Psychtoolbox (PTB) window and defines screen parameters.

11. params.m: Initializes various parameters for the experiment, such as stimulus type, stimulus size, viewing distance, spatial frequency, step size, dwell time, and contrast step size.

12. plotData.m: Generates plots of the target and subject gaze positions, normalized position error, and contrast.

13. prefs.m: Sets preferred stimulus parameters.

14. runCPET.m: Main script to run the tracking experiment, collect data, and save results.

15. updateStim.m: Updates the position of the stimulus based on the specified distribution and movement type.

16. wpolyfit.m: Function for fitting a weighted polynomial to data.

The relation between these scripts can be viewed in the dependency tree (figure 1).

**CPET dependency tree**

```
runCPET
        params
                prefs
                calc_freq_cpd
        initWin
                Screen (PTB)
        initEyeLink
                EyeLink (PTB)
                EyelinkInitDefaults  (PTB)
                EyelinkDoTrackerSetup (PTB)
                Screen (PTB)
        initStim
                CreateProceduralGabor (PTB)
                Screen (PTB)
                calc_ppd
        drawStim
                Screen (PTB)
        updateStim
                collectData
                Screen (PTB)
        estimateError
        closeEyeLink
                EyeLink (PTB)
                Screen (PTB)
        plotData


generateRefractiveSeries
        wpolyfit
```

Figure 1: CPET dependency tree.

Data generated during experiments is stored in the DATA folder. After running the program, EyeLink Data Files (.EDF) are stored in the EDF subfolder, MATLAB plots (.fig) generated by `plotData.m` are stored in the FIG subfolder, and the MATLAB data file (.mat) containing subject information, trial settings, and tracking data are stored in the MAT subfolder.

The DEMOS folder contains demonstration scripts such as `gaborWalk.m`, which displays a Gabor patch following a random walk trajectory, and `trackGaze.m`, a script that demonstrates gaze tracking using the EyeLink eye tracker.

Lastly, the STIMULI folder holds the images for the Aukland Optotypes and Marmoset stimuli used in the experiments.

# Parameters

This section highlights some of the key parameters defined in the `params.m` script of the Continuous Psychophysics with Eye Tracking (CPET) program and their potential impact on your experiment. If you're adapting the program for a new setup, it's crucial to review these parameters to ensure they align with your specific hardware setup and experimental design.

Several parameters related to the hardware setup should be of interest. For example, `screenWidthCm` and `screenWidthPx` define the width of the screen in centimeters and pixels, respectively, while `viewingDistanceCm` indicates the distance from the observer to the screen in centimeters. These parameters can significantly affect how stimuli are displayed and interpreted, and thus should be carefully adjusted to match the actual physical setup of the experiment.

The chosen stimulus type (`stim`) can be Gabor patch, Letter, Optotype, or Marmoset Face, and various attributes of these stimuli can be configured. For instance, `stimSize` denotes the size of the stimulus in pixels, and should be adjusted according to the resolution of your screen and the requirements of your experiment.

For the Gabor patch stimulus, several other parameters might be adjusted. `freqCpdDesired` specifies the desired spatial frequency in cycles per degree (CPD), and `res` denotes the resolution of the Gabor patch stimulus in pixels. Furthermore, `contrast` indicates the initial contrast of the stimulus and `conStep` represents the step size for adjusting the stimulus contrast, represented as a percent decrease.

In terms of motion, two types are available: jitter and curvilinear trajectory, determined by the `movement` parameter. For jitter motion, `stepSize` and `dwellTime` define the size of the stimulus movement in pixels and the time interval between each stimulus update in milliseconds, respectively. For curvilinear trajectory, `dwellFrames` specifies the number of frames between each stimulus update.

The `stimFreq` parameter is especially important as it indicates the frequency of stimulus position updates. This parameter will vary based on the `stepSize`, `dwellTime`, and `dwellFrames`, and determines how often the stimulus position or contrast is updated, as well as how often data is collected.

Please note this description doesn't cover all parameters included in `params.m`. For a full understanding of each parameter, please refer to the comments and documentation provided in the script itself. Bear in mind that adjusting these parameters appropriately for your specific setup and experimental needs is essential for the successful execution and interpretation of your experiment.

# Stimuli and motion

The CPET program offers a choice of four different stimuli: Gabor, letters, Marmoset face, and Aukland Optotypes. The selection of the stimulus is made at runtime and is dependent on the settings specified in the scripts `params.m` and `initStim.m`.

Regarding motion, the program offers two options: jitter motion and curvilinear trajectory. In the case of jitter motion, displacement for each position update is determined by a chosen step size in pixels. The frequency of these position updates is controlled by a specified dwell time. When the experiment is running, the position updates are sampled from either a uniform or normal distribution, an aspect that can be modified by the user. Data collection occurs at each position update, so the frequency of data sampling is dictated by the `stepSize` and `dwellTime` parameters.

On the other hand, in the curvilinear trajectory option, a predetermined number of random points on the screen are generated. These points are then connected by a smooth spline, creating a trajectory that the stimulus follows throughout a trial. As position updates are continuous in this motion type, a variable `dwellFrames` is set to determine the number of frames during which data is sampled. This variable serves to determine the sample rate in this type of motion. Thus, the CPET program provides flexible options for stimuli and their motion paths, catering to a range of experimental requirements.

# Experiment Procedure

Before starting, first ensure that the EyeLink PC has been turned on and is connected to the lab computer. To conduct an experiment, the user executes the `runCPET.m` script which starts by initializing various experimental parameters with calls to `params.m` and `prefs.m` scripts. `params.m` prompts the experimenter to set general parameters such as type of stimulus (Gabor, letter, Aukland Optotype, or marmoset face), contrast, tracking movement type (jitter or curvilinear trajectory), distribution of position updates (normal or uniform), and EyeLink settings. `prefs.m` is an optional script that allows the user to set their preferred parameters for the stimulus, such as size, viewing distance, spatial frequency, step size for movement and contrast adjustment, dwell time, and others.

The `runCPET.m` script then initiates the experiment by initializing the PsychToolbox window (via `initWin.m`), the EyeLink tracker (via `initEyelink.m`), and the visual stimulus (via `initStim.m`).

Following the execution of `initEyelink.m`, a EyeLink calibration window appears on the CRT monitor and the experimenter can carry out the calibration and validation of eye positions on the EyeLink PC.

Once the calibration step is complete, the experimenter can exit the EyeLink setup, and the experiment will continue. After a brief instruction period, the stimulus is displayed, and the gaze tracking begins.

The primary experiment loop in `runCPET.m` operates continuously to draw the stimulus, update its position and contrast, collect gaze data, and estimate positional errors in real-time. The loop executes until either an 'escape' key is pressed, the contrast falls below a certain threshold, or an early stopping criterion based on the estimated positional error (assessed every loop after the first 30 seconds of a trial via `estimateError.m`) is met.

## Data Collection and Analysis

The `collectData.m` script gathers down sampled gaze data from the EyeLink tracker and information about the stimulus for each position update. After the experiment loop, data recording is halted and the EyeLink connection is closed by `closeEyelink.m`. EyeLink Data Files containing the raw data sampled at either 250 or 500 Hz are transferred from the EyeLink PC to the Host machine via Ethernet and saved as .EDF format in the ../DATA/EDF directory. The script `plotData.m` is then called to visualize the target and subject gaze positions, and the normalized position error (figure 2). This visualization helps to better understand the subject's tracking performance and the impact of gradually decreasing the stimulus contrast. The figures generated are saved to a .fig file and stored in the ../DATA/FIG directory.

Finally, all collected data are saved to a .mat file via the `runCPET.m` script. The saved data includes

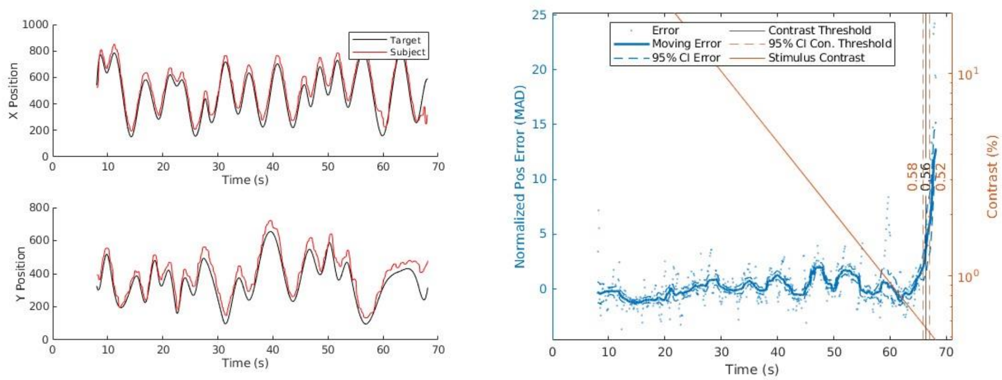


Figure 2: Examples of plots generated from each trial. (Left) Plot of the target and subject positions. (Right) Normalized position error and contrast threshold estimate.

subject identification, trial number, diopter value, stimulus selection, stimulus data, gaze data, contrast threshold, contrast confidence interval, and lag. These data files are stored in the ../DATA/MAT directory, allowing for subsequent analysis.

# Estimation of Refraction

The ultimate aim of the CPET program is to estimate the refractive error of the subject based on the collected gaze tracking data. This is achieved by determining a contrast threshold at which the subject can no longer accurately track the stimulus. As the contrast of the stimulus decreases, the subject's ability to accurately track the stimulus diminishes. The point at which tracking errors exceed three units of median absolute deviation is identified as the contrast threshold.

By conducting the tracking experiment on a subject multiple times while utilizing a range of spherical concave and convex trial lenses, a refractive series can be plotted via the `generateRefractiveSeries.m` script that highlights the subjects best lens power based on the contrast thresholds obtained throughout trials (figure 3). The estimated refractive error expressed in diopters indicates the degree of nearsightedness (negative value) or farsightedness (positive value) of the subject's eye.

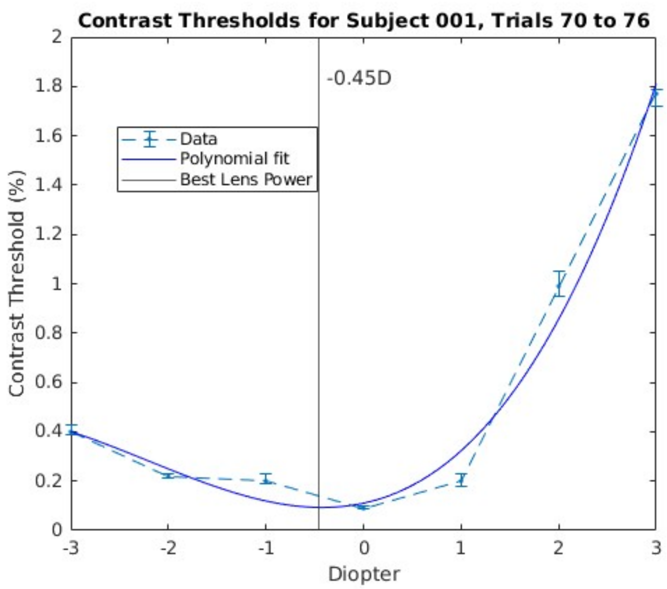


Figure 3: Example of the refractive series plot generated from seven trials with varying trial lens powers.

# Limitations and Future Directions

This psychophysical paradigm with eye tracking could be useful for visual research in young children or non-human primates, wherever assessing refractive errors is necessary and standard methods cannot be employed. However, there are certain limitations to consider.

The CPET program has initially been deployed with human subjects, successfully demonstrating its ability to generate a refractive series and estimate refractive errors. However, the potential applications of this program extend beyond human subjects. Non-human primates, such as the common marmoset, are widely used animal models in vision research. Due to laboratory conditions they often exhibit myopia, which underscores the importance of measuring refractive errors. This presents a compelling use case for the CPET program, positioning it as a vital tool in the ongoing advancement of vision research within non-human primate populations.

In terms of eye tracking methods, the CPET program currently interfaces with the EyeLink eye tracking system. Expanding compatibility to other eye tracking devices and technologies would enhance its versatility and applicability to a wider range of research settings.

Overall, the CPET program provides a valuable tool for conducting continuous psychophysics experiments with eye tracking. By monitoring gaze behavior and adjusting stimulus parameters, researchers can estimate refractive errors. With further development and refinement, this program has the potential to contribute to advancements in visual perception research and clinical applications related to assessing refractive errors.