CFS MATLAB toolbox: An experiment builder for continuous flash suppression (CFS) task

Mikko Nuutinen¹ · Terhi Mustonen¹ · Jukka Häkkinen¹

Published online: 15 September 2017
© Psychonomic Society, Inc. 2017

Abstract CFS toolbox is an open-source collection of MATLAB functions that utilizes PsychToolbox-3 (PTB-3). It is designed to allow a researcher to create and run continuous flash suppression experiments using a variety of experimental parameters (i.e., stimulus types and locations, noise characteristics, and experiment window settings). In a CFS experiment, one of the eyes at a time is presented with a dynamically changing noise pattern, while the other eye is concurrently presented with a static target stimulus, such as a Gabor patch. Due to the strong interocular suppression created by the dominant noise pattern mask, the target stimulus is rendered invisible for an extended duration. Very little knowledge of MATLAB is required for using the toolbox; experiments are generated by modifying csv files with the required parameters, and result data are output to text files for further analysis. The open-source code is available on the project page under a Creative Commons License (http://www.mikkonuutinen.arkku.net/CFS_toolbox/ and https://bitbucket.org/mikkonuutinen/cfs_toolbox).

Keywords Image rating · Image quality · MATLAB · Computer software

This project has been supported by the Finnish Doctoral Program in User-Centered Information Technology.

✉ Mikko Nuutinen
mikko.s.nuutinen@gmail.com

¹ Department of Psychology and Logopedics, Faculty of Medicine, University of Helsinki, PO Box 9 (Siltavuorenpenger 1 A), Helsinki, Finland

Introduction

Continuous flash suppression (CFS) (Tsuchiya & Koch, 2005) is an experimental method that enables the investigation of visual processing outside conscious perception. The method can be considered as an evolved form of the binocular rivalry paradigm. In this paradigm, two dissimilar stimuli are shown to the two eyes of an observer, causing that the observer's conscious percept alternates between the competing stimuli. While traditional binocular rivalry involves two roughly equally strong stimuli, one presented to each eye, CFS involves a more unbalanced design. In CFS a series of high-contrast masks (noise patterns) are flashed to one eye, typically at a rate around 10 Hz, to temporally prevent the conscious perception of a stimulus shown to the other eye. Due to the strong interocular suppression that the dynamically changing noise pattern creates, the target stimulus (i.e., a Gabor patch) is rendered invisible for an extended duration. For instance, contrast thresholds may increase up to over tenfold as compared to thresholds during binocular rivalry (Tsuchiya & Koch, 2005; Tsuchiya, Koch, Gilroy, & Blake, 2006). Furthermore, as the transition from suppression to dominance stage can be more precisely defined under CFS than binocular rivalry, CFS reduces reporting variance that is a common concern in binocular rivalry designs. One typical approach to determine the transition during CFS is to measure the time needed for the detection of targets presented to the suppressed eye (i.e., breaking-CFS; Jiang, Costello, & He, 2007).

Because CFS provides a means to control the rivalry alteration during dichoptic viewing, it is a great tool for studying visual targets' ability to overcome suppression; that is, to be detected. For example, research has shown that target detectability during CFS strongly depends on low-level visual features, such as the contrast (Tsuchiya

et al., 2006; Yang & Blake, 2012), spatial frequency, and orientation of the stimulus (Yang & Blake, 2012). Some scholars have claimed that even higher-order cognitive factors, such as familiarity or emotional content, may affect the speed with which the suppressed targets break into awareness (Jiang et al., 2007; Yang & Yeh, 2011). However, it should be mentioned that recent findings show contradictory results (Moors, Wagemans, van Ee, & de Wit, 2016c; Moors, Wagemans, & de Wit, 2016a; Gelbard-Sagiv, Faivre, Mudrik, & Koch, 2016; Gayet, Van der Stigchel, & Paffen, 2014).

As far as we know, no prior scholars have published easy-to-use software that enables building and executing CFS experiments with different parameters and stimuli. Most researchers have used MATLAB (Tsuchiya et al., 2006; Yang, & Blake, 2012; Hesselmann, Darcy, Ludwig, & Sterzer, 2016; Zhu, Drewes, Melcher, 2016; Gelbard-Sagiv et al., 2016; Hong, 2015) or Python (Moors, Wagemans, van Ee, & de Wit, 2016b, c) environments and Psychtoolbox-3 (PTB-3) (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997) or PsychoPy packages (Peirce, 2006, 2009) for preparing and conducting CFS experiments. A few scholars (Stein, Thoma, & Sterzer, 2015) have also used Cogent 2000 toolbox.¹ Furthermore, software^{2,3} for creating CFS masks are available and these can be utilized in developing CFS-related software further. However, no links or repositories for a ready-to-use software has been previously published.

Because preparing a new CFS experiment from scratch is not a trivial task, and because programming is not necessarily the primary skill of scholars in this research area, we believe that a publicly available, easily modifiable, and versatile CFS experiment builder would be highly valuable. With the present off-the-shelf CFS toolbox, researchers are able to build CFS environments with various parameters and stimuli, and execute experiments without the need of software design and implementation.

The present article is an introduction to CFS toolbox, a platform that we developed for building and conducting continuous flash suppression experiments. The toolbox is not restricted to the specific types of stimuli or experimental settings. Instead, the user can modify the types, sizes, locations, and timings of visual stimuli and employ different noise patterns as CFS masks. The present version of the toolbox supports research designs that utilize accuracy and time-to-detection as the dependent measures of observers' performance (cf. breaking-CFS) (Jiang et al., 2007). Furthermore, the toolbox enables the utilization of auditory stimuli. Because all settings and experimental

phases are built through csv files, CFS toolbox provides tools for forming continuous flash suppression experiments without the need for programming. The toolbox source code and setup manual are available from the CFS toolbox project page (http://www.mikkonuutinen.arkku.net/CFS_toolbox/) and Bitbucket repository (https://bitbucket.org/mikkonuutinen/cfs_toolbox).

Program description

Software and hardware requirements

CFS toolbox requires the installation of MATLAB or GNU/Octave and PTB-3 (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997). PTB-3 is a set of functions for experimental psychology research that runs on multiple platforms (Windows, Mac, Linux) and that makes the presentation of accurately controlled visual and auditory stimuli easy.

CFS toolbox has been tested under Windows 7 with MATLAB R2014a and Linux Ubuntu 15.04 with GNU/Octave 4.0. The toolbox versions for both environments are available from the project page. CFS toolbox draws and plays visual and auditory stimuli using the functions of Screen⁴ and PsychPortAudio⁵ in PTB-3. Screen is a function for the precise control of video display. PsychPortAudio contains a set of parameters for working with sounds. The functionality requires a suitable graphic and audio cards and computer. It is recommended that the user refers to the PTB-3 documentation⁶ for up-to-date details.

In our laboratory setup, visual stimuli are viewed through a mirror stereoscope that presents the stimuli in the right half of the display exclusively to the right eye and the stimuli in the left half of the display exclusively to the left eye (see Fig. 1). Some scholars (Hesselmann et al., 2016; Moors et al., 2016b) have used a setup in which two displays that are set opposite to each other, project stimulus to the left and right eye respectively via two mirrors. More details of our mirror stereoscope setup can be found from the toolbox project page.

Parameter files

The program architecture is described in Fig. 2. The primary interface to CFS toolbox are stimulus parameters ("stimulus_parameters.csv") and trial parameters ("trial_parameters.csv") files. Toolbox reads the parameter files and runs the experiment according to the parameters.

¹<http://www.vislab.ucl.ac.uk/cogent.2000.php>

²http://martin-hebart.de/code/make_mondrian_masks.m

³<https://doi.org/10.6084/m9.figshare.3102730.v1>

⁴<http://docs.psychtoolbox.org/Screen>

⁵<http://docs.psychtoolbox.org/PsychPortAudio>

⁶<http://psychtoolbox.org/requirements/>

Fig. 1 Mirror stereoscope presents the stimulus in the left half of the display exclusively to the left eye and the stimulus in the right half of the display exclusively to the right eye. A head-and-chin rest is used to keep the head stable

Stimulus parameters file contains parameters for the whole experimental setup (see Table 1). Trial parameters file contains a sorted list of definitions for each individual trial (see Table 2). The parameter values in both files can be changed, although the structure of the files must be preserved when saving.

Furthermore, the structure of the CFS toolbox director and the file names must be preserved (see Fig. 3). For example, there are directors for visual and auditory stimuli and borders and infographs files. The MATLAB m-files and csv parameter files are located in the root of the toolbox ('/CFS/'). When the all trials defined in trial parameters file have been run, the result file is saved to the result director ('/CFS/Results'). The structure of result file is presented in Section "Result file".

Fig. 2 Structogram of the CFS toolbox architecture

Stimulus parameters file

The stimulus parameters define the location and size of the experiment window and the options for all elements of the experiment. All elements, such as border frames, noise pattern, and visual target, are drawn inside the experiment window. That is, the coordinate space of the experiment window is used for defining the locations and sizes of the elements. The location and size of the experiment window are defined using the coordinate space of the display device.

Figure 4 shows an example experiment window containing the elements of visual target, noise pattern, fixation crosses, and border frames. In the actual experiment, mirror stereoscope projects the left half of the experiment window (i.e., visual target or noise pattern) to the left eye and the right half to the right eye. Figure 5 shows a simulated perception of Fig. 4 assuming that the experiment window is watched through a mirror stereoscope. That is, the figure visualizes perception after the visual target located in the upper left half of the left side (i.e., presented to the left eye) of the experiment window is detected.

Table 1 lists the parameters of stimulus parameters file that can be changed in order to modify the experiment setup. For example, the experiment window's location and size are set by the parameter #2a. The values of parameter #2a are the coordinates for the left-up (x1,y1) and right-down (x2,y2) corners of the experiment window. It is recommended that the user sets the coordinates to fulfill the whole screen area of display. In this way, distracting light sources from the borders of display screen can be minimized.

The coordinate options for visual targets and the coordinate values for noise patterns and border frames are set by the parameters #3c, #4i, and #5a, respectively. CFS toolbox enables the user to set eight different options for a visual target stimulus. For instance, the targets can be displayed in the four corners of the two frames (eight options), or up or down in one frame only (two options). The principle is that the eight coordinate options are recorded in stimulus parameters file from which the trial specific values are then selected for each trial according to trial parameters file.

The border frames are used for framing the left and right active area of the experiment window; that is, the areas for the left eye and for the right eye. The border frame image ('border.png') is read from the director '/CFS/Borders_and_info'. The type of border frames can be changed by changing the image file from the director while preserving the naming of that file. Different noise types and noise presentation parameters are defined in more detail in Section "Noise".

After the user has set or changed the coordinate values of visual target, noise, or border frames, it is recommended to run the function "test_settings.m". The function draws a mock-up of the experiment window that can be used

Table 1 The variables of the stimulus parameters file

#	Parameter	Description
1	Number of warmup trials	No training session before actual test → set 0; or set 1 – n warmup trials
2a	EW location and size	Left-up corner (x1,y1) and right-down corner (x2,y2) coordinates
2b	EW background intensity	Gray value (0–255)
2c	Fixation cross color	RGB values (0–255) [R G B]
2d	Fixation cross disparity	The distance (in pixels) between the left and right fixation crosses
2e	Noise and info image locations and sizes	Coordinate values on the experiment window
2f	Border frames locations and sizes	Coordinate values on the experiment window
3a	Refresh rate of visual target	Waiting time for visual target update (s)
3b	Fade rate of visual target	The intensity amount of visual target strengthen on every update (0–1)
3c	Visual target locations and sizes	Visual target locations on the experiment window (eight options: S1–S8)
4a	Noise type	Grayscale, pepper, color, dead leaves disk, or dead leaves square
4b	Dead leaves sigma ^a	If $\sigma \rightarrow 0$ elements are uniform; $\sigma = 3$ means scale invariant image
4c	Noise temporal frequency	How many noise images are presented per second
4d	Noise variation	Number of different noise images presented
4e	Noise intensity range ^b	The intensity range from which noise blocks are randomly selected (0–255)
4f	Noise block size ^b	The block size of noise image
4g	Noise start timing	The delay between noise and visual target (i.e., 2 s + random interval)
4h	Random interval	Random additive delay for noise start interval (i.e., 0–2 s → set 2)
5	Random trial order	1 = yes, 0 = no; randomizing the trial order defined in the trial parameters file
6	Audio stimulus timing	Audio stimulus duration relative to visual stimulus (six options: A1–A6)
7	Time out	Timeout of trial without interaction (s)
8	Disable audio stimuli	1 = yes, 0 = no; If disabled, only visual stimuli are used in the experiment
9	Response keys	1 = Up, 2 = Down, 3 = Left, 4 = Right; Two response keys can be selected
10	Response delay	Time delay after the response keys are active after visual target onset

EW experiment window

^aParameter for dead leaves noise type

^bParameter for grayscale, pepper, and color noise types

for checking that all experimental elements are correctly located.

A new property of CFS toolbox, compared to many previous CFS research settings, is the possibility to include timed audio stimulus for experimentation; an audio stimulus can be displayed before, at the same time, or after the visual target onset. We see that the possibility to utilize audio stimuli can create new CFS research branches. For example, audio can be utilized in triggering some novel multimodal effects.

The timing options for audio stimulus are set by the parameter #6. The timing of visual and audio stimuli is defined in more detail in Section “Stimulus timing”.

Trial parameters file

Each row of trial parameters file defines the presentation of visual and auditory stimuli for one trial. The number of rows defines the number of trials. That is, if 20 rows have been

Table 2 The row-specific values of the trial parameters file: the types of visual (1) and audio (3) targets, the location of visual target (2) and the timing of audio target (4)

#	Parameter	Description
1	Visual stimulus type	Select visual stimulus file (1 = “visual_stimulus.1.png”, 2 = “visual_stimulus.2.png”)
2	Visual stimulus location	1 = S1, 2 = S2, 3 = S3, 4 = S4, 5 = S5, 6 = S6, 7 = S7, 8 = S8 (see stimulus parameter #3c)
3	Audio stimulus type	Select audio file (1 = “auditory_stimulus.1.wav”, 2 = “auditory_stimulus.2.wav”)
4	Audio stimulus timing	1 = A1, 2 = A2, 3 = A3, 4 = A4, 5 = A5, 6 = A6 (see stimulus parameter #6)

Fig. 3 The folder structure of the CFS toolbox

filled, then the experiment will run 20 trials. The order of trials is randomized for each observer by the parameter #5 of stimulus parameters file.

Fig. 5 A simulated perception of the experiment window after the visual stimulus has been detected

In summary, each trial is defined by four parameters: (1) the location of visual target, (2) the type of visual target (image file), (3) the timing of audio stimulus, and (4) the type of audio stimulus (audio file). The parameters are described in Table 2. The location of visual target is set by integer value from 1 to 8. Actual coordinate locations for different integer values are coded in the parameter #3c of stimulus parameters file. For example, value 1 can be coded to mean the top-left location of the right-side frame in the experiment window, value 2 the top-right location of the left-side frame, and so on.

In this version of the toolbox, there is an option to use two different visual target types in experimentation. The toolbox reads the visual targets from the director ‘/CFS/visual_stimuli’. The value 1 of the parameter #2 means that the visual target is the image file named as ‘visual_stimulus_1.png’, and the value 2 means that the target is the image file ‘visual_stimulus_2.png’. Visual target images can be changed just by changing the image files from the director while preserving the naming of images.

Audio stimulus timing and type are set by the parameters #3 and #4. The timing of the audio stimulus in each

Fig. 4 The experiment window containing the elements of noise mask, border frames, fixation cross, and visual target

trial relative to the visual target is set by integer values from 1 to 4. Actual timing values for audio stimulus relative to visual targets are coded in the parameter #6 of stimulus parameters file. For example, value 1 can be coded to mean that the audio stimulus is presented 0.5 s before the visual target, value 2 can mean that audio and visual targets are presented at the same time, and so on. The timing principle of audio and visual targets is explained in more detail in section “Stimulus timing”.

The audio stimulus type for a trial is set by the parameter #4. In this version of the toolbox, there is an option to use two different audio stimulus types in experimentation. Audio stimulus files are read from the director ‘/CFS/audio_stimuli’. The value 1 of the parameter #4 means that the audio file named as ‘auditory_stimulus_1.wav’ is selected for the trial, and value 2 means that ‘auditory_stimulus_2.wav’ is selected. As with the visual target files, audio stimuli can be changed by changing the audio files from the director while preserving naming of the files.

Stimulus timing

The timing settings for noise and target stimuli should be carefully designed when building a new CFS experiment. Figure 6 shows an example CFS trial as a function of time line. In the figure, the timings of target stimuli (right eye location in this trial and audio) and noise (left eye location in this trial) have been visualized. For example, the interval of $t_2 - t_4$ describes the time between the noise stimulus onset (left eye) and the consequent target stimulus onset (right eye) that follows. This discrepancy between the stimulus timings is used to ensure the dominance of the noise mask at target initiation.

The length of the interval of $t_2 - t_4$ is controlled by the parameters of noise start timing (#4g) and random interval (#4h). The noise start timing is a constant delay time for all trials of the experiment. That is, what is the minimum

time before visual target forming starts after the noise onset. The random interval is a random additive delay added to the noise start timing. For example, if the value of random interval is 1, then additive delay is something between 0 and 1 s, and the total delay between the noise and visual target consists of the constant noise start timing plus the random delay of 0–1 s that varies from trial to trial during the experiment.

The interval of $t_4 - t_5$ is the duration of a visual target reaching its full contrast state. CFS toolbox enables ramping up the targets so that the contrast increases slowly from low to full. This feature is used to avoid the abrupt onset of target stimuli. The duration of transition from low to full contrast is controlled by the parameters of stimulus refresh rate (#3a) and stimulus fade rate (#3b). The stimulus refresh rate defines how many times the stimulus contrast is updated per second. The stimulus fade rate defines how much the contrast increases at each update. For example, if the stimulus refresh rate, $rr = 75 \text{ frames/seconds}$, and the stimulus fade rate, $fr = 0.0133$, then the interval $t_4 - t_5$ is 1.0 s. That is, it takes 1 s to form a full-contrast visual target after the target triggering. If $fr = 1$, then the full contrast state is reached immediately after the triggering.

In the example trial presented in Fig. 6, the audio stimulus onset takes place at the time point of t_3 . That is, the audio file is played before the visual target onset with a duration defined as $t_3 - t_4$. For each trial, the duration of $t_3 - t_4$ is defined by the parameter #4 of trial parameters file. The principle is that CFS toolbox enables displaying audio files at different time points (as compared to the visual target) for different trials of the experiment.

A trial ends when the observer presses the arrow key in the keyboard or when the value given for the time out is reached. The value of the time out is set by the parameter #7 in stimulus parameters file. Two arrow keys (up, down, left or right) that observers use to respond to the stimuli should be set by the parameter #9. The next trial starts when the observer presses the space key.

Fig. 6 An example trial: Timing of stimuli and noise as a function of time ($t_1 - t_5$)

Fig. 7 Grayscale, pepper, and color noise frame examples

Noise

The properties of the dynamically changing noise pattern, presented to the other eye than the visual target, are selected by the parameters #4a–#4h of stimulus parameters file. Noise type is selected by the parameter #4a. This version of toolbox contains the functions to form the noise types “grayscale”, “color”, “pepper”, and “dead leaves”. Figure 7 presents the grayscale, pepper, and color noise. The parameters of these noise types are the block size and intensity range. In Fig. 7, the block size of the noise images is 15 pixels. The native size of the noise images produced by CFS toolbox is 500×500 pixels. The spatial frequency of the example noise images is 33 units per image width ($500 \text{ pixels} / 15 \text{ pixels} = 33$). That is, the spatial frequency content of the noise images is controlled by the block size parameter. The intensity range is the range from which the intensity values of single blocks are randomly selected. The intensity ranges for the grayscale and pepper noise examples in Fig. 7 are 0–220 and 0–255, respectively.

Figure 8 presents dead leaves noise examples. CFS toolbox uses the code of⁷ to form the dead leaves noise images. The spatial structure of the dead leaves noise differs from the grayscale, pepper, and color noise, as the dead leaves noise consists of a series of overlapping elements. In this version of toolbox, the form of the elements can be a disk or square. The dead leaves noise with square elements corresponds to the so-called Mondrian patterns that are typically used in CFS experiments (Tsuchiya et al., 2006). The second parameter of the dead leaves noise is the sigma, defined by the parameter #4b. Sigma controls the repartition of the size of the basic shape: $\sigma \rightarrow 0$ gives a more uniform repartition of shape and $\sigma = 3$ gives a nearly scale invariant image. In Fig. 8, the elements of noise targets are square (up) and disk (down). The sigma value is 1 in the left-side examples and 3 in the right-side examples.

⁷https://se.mathworks.com/matlabcentral/fileexchange/16201-toolbox-image/content/toolbox_image/compute_dead_leaves_image.m

In addition to noise type and noise type-specific parameters, the number of different noise images (n) and presentation frequency (f) should be defined. The dynamically changing noise mask (video stream) is produced by preparing n random noise images that are streamed in consecutive order. For example, when $n = 60$ and $f = 30$, CFS toolbox displays a sequence of 60 noise images with a frequency of 30 images per second. That is, the same noise video sequence is re-played every 2 s. The value n is set by the parameter #4d and f by the parameter #4c of stimulus parameters file.

Result file

After all trials defined in the trial parameters file have been run, an experiment-specific result file is saved to the

Fig. 8 Dead leaves noise frame examples: square, $\sigma = 1$ (up-left); square, $\sigma = 3$ (up-right); disk, $\sigma = 1$ (down-left); disk, $\sigma = 3$ (down-right)

trial	trial_rand	visual_stimulus_file	visual_stimulus_pos	stimulus_start	audio_file	audio_start	answer	ansdur
1	9	visual_stimulus_1	OPTION 1	1.5177	auditory_stimulus_1	0.0000	DOWN	0.5798
2	22	visual_stimulus_2	OPTION 2	2.2238	auditory_stimulus_1	-0.2000	DOWN	0.8480
3	30	visual_stimulus_2	OPTION 2	2.4191	auditory_stimulus_1	0.2000	DOWN	0.6568
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
44	35	visual_stimulus_1	OPTION 3	2.2118	auditory_stimulus_1	0.0000	UP	0.6098
45	16	visual_stimulus_2	OPTION 6	2.3008	auditory_stimulus_1	0.2000	DOWN	0.6591

Fig. 9 Example rows of result file

result director ('CFS/Results'). Result files are named as 'seed-ID-date-dd.mm.yyyy-hh.mm.ss.txt', in which 'ID' is the participant's identification number, 'dd.mm.yyyy' is the date, and 'hh.mm.ss' is the exact time when the experiment was finished.

Figure 9 shows example rows from a result file. The first row of the file shows that the trial parameters were selected from the row 9 of trial parameters file (column: trial_rand). That is, the order of the trials was randomized. The visual target of the first trial was the image file 'visual_stimulus_1.png' (column: visual_stimulus_file), which was triggered 1.5177 s after the noise onset (column: stimulus_start). The value of stimulus_start equals to the interval of $t_2 - t_4$ in Fig. 6. The position of the visual target was the integer value 1 coded in stimulus parameters file.

The audio file of the first trial was 'auditory_stimulus_1.wav' (column: audio_file). In this trial, the audio was initiated at the same time with the visual target (column: audio_start = 0.0000). The answer column shows that the observer pressed the "DOWN" button of the keyboard. The response time was 0.5798 second (column: ansdur). The response time is the duration from the visual stimulus onset to the response button press. It thus tells how long did it take for the observer to detect the visual target.

Key functions

test_settings.m The function displays the location and size of the experiment window and all elements inside it (Fig. 4). This enables the user to ensure that all targets are displayed correctly (e.g., in the correct locations on the monitor relative to the mirror stereoscope). It is recommended to run this function before running a new experiment.

run_experiment.m Once the experiment has been specified in the parameter files, it can be run from MATLAB command window by calling the function *run_experiment*. This function reads the parameter files and runs the experiment. The experiment window and visual and audio targets are presented using the function *draw_stimulus*. The noise mask is created using the function *create_noise_basic* (grayscale, pepper or color noise) or *create_noise_dead_leaves* (dead leaves noise).

save_answer After the experiment is finished, result data are output to a text file and saved to the folder "Results" using the participant's ID code and the time stamp of the experiment in the file name.

Example experiment

To demonstrate an experiment conducted with CFS toolbox, we present a simple experiment in which the effects of visual target position (above or below the fixation cross) and audio tone pitch (high or low pitch) were investigated. Subjects responded to visual target occurrence (i.e., arrow-shaped square) based on the target's pointing direction (i.e., left or right). Performance was defined as the time required for target detection (reaction time, RT) according to the b-CFS procedure (Jiang et al., 2007).

Subjects

The subjects were 20 students from the University of Helsinki (five males; mean age = 24.8 ± 3.4 years). All subjects were right-handed and 11 of them were right-eye dominant. All subjects reported normal hearing and normal or corrected-to-normal vision, and wore normal corrective lenses during the experiment. The subjects were screened for normal stereopsis, heterophoria, and near distance visual acuity at three contrast levels (100%, 10%, and 2.5%).

Fig. 10 Effects of Location and Sound. Mean RTs for visual targets above and below the fixation associated with high pitch and low pitch tones. Error bars are standard errors of the mean

Procedure

Visual target and noise mask, both surrounded by a black-and-white border frame, were presented on the experiment window 13° apart from each other. The arrow-shaped visual targets appeared directly above or below the fixation cross at 2.2° eccentricity. The visual targets were ramped up to a target contrast ($C_{Michelson} = 0.10$) during the initial 400 ms, and were then kept constant until the end of the trial. Each target was accompanied with a high or low pitch tone.

Subjects were instructed to respond to target occurrence based on the arrow's pointing direction (i.e., left vs. right). They were asked to respond immediately upon detecting the target by pressing the corresponding key (i.e., left or right arrow) on the keyboard with the right hand. The experiment consisted a total of 128 trials. Target location and pointing direction were counterbalanced within the stimuli. High and low pitch tones appeared equally often with each of these combinations.

Results

Only correct response trials (98.8%) were included in the RT analysis. Outliers ($> 3SD$) were identified and removed from the dataset. In total, 3.28% of the original data was excluded. The dataset was log-transformed before statistical testing. Descriptive statistics are based on the geometric means of RT distribution.

Figure 10 presents the results. Data were analyzed using a mixed-design ANOVA with Location (up, down), Sound (low pitch, high pitch), and Pointing direction (left, right) as within subject factors. Results revealed a significant main effect of Location, $F(1, 19) = 5.20$, $p = .034$, $\eta_p^2 = .22$. This indicates that visual targets presented above the fixation cross ($M = 1054$ ms, 95% $CI[972, 1143]$) were detected faster than targets below the fixation cross ($M = 1124$ ms, 95% $CI[1032, 1223]$). No significant effects were found for Sound, $F(1, 19) = .40$, $p = .54$, $\eta_p^2 = .021$, or Pointing direction, $F(1, 19) = 1.24$, $p = .28$, $\eta_p^2 = .061$. Neither Sound*Location interaction, $F(1, 19) = .70$, $p = .41$, $\eta_p^2 = .036$, nor any other interaction, reached statistical significance, all $F \leq 1.69$, $p \geq .21$, $\eta_p^2 \leq .081$.

Conclusions

CFS toolbox is currently in use in research laboratories at the University of Helsinki. It has proven to be useful in the experimental psychology studies of visual attention and cognition. The easy handling of experiment building and launching through the parameter files provide researchers a set of easy-to-use tools for forming complex continuous flash suppression experiments.

CFS toolbox is being actively maintained and developed. New functionalities (e.g., stimulus generator with the parameters of frequency, direction and phase) and CFS masks (e.g., pink noise images; see Gayet, Paffen, & der Stigchel, 2013; Gayet et al., 2014), an option to present stimuli with anaglyph glasses (instead of mirror stereoscope), and a graphical user interface are expected in future versions. In addition, support for Mac operating system and Python implementation will be considered.

Furthermore, researchers are welcome to add functionalities to the code themselves. We hope that the research community finds this toolbox a useful tool for their laboratories.

References

- Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision*, 10(4), 433–436. <https://doi.org/10.1163/156856897x00357>
- Gayet, S., Paffen, C. L. E., & Van der Stigchel, S. (2013). Information matching the content of visual working memory is prioritized for conscious access. *Psychological Science*, 24(12), 2472–2480. <https://doi.org/10.1177/0956797613495882>
- Gayet, S., Van der Stigchel, S., & Paffen, C. L. E. (2014). Seeing is believing: Utilization of subliminal symbols requires a visible relevant context. *Attention, Perception, & Psychophysics*, 76(2), 489–507. <https://doi.org/10.3758/s13414-013-0580-4>
- Gelbard-Sagiv, H., Faivre, N., Mudrik, L., & Koch, C. (2016). Low-level awareness accompanies “unconscious” high-level processing during continuous flash suppression. *Journal of Vision*, 16(1), 3. <https://doi.org/10.1167/16.1.3>
- Hesselmann, G., Darcy, N., Ludwig, K., & Sterzer, P. (2016). Priming in a shape task but not in a category task under continuous flash suppression. *Journal of Vision*, 16(3), 17. <https://doi.org/10.1167/16.3.17>
- Hong, S. W. (2015). Radial bias for orientation and direction of motion modulates access to visual awareness during continuous flash suppression. *Journal of Vision*, 15(1), 3. <https://doi.org/10.1167/15.1.3>
- Jiang, Y., Costello, P., & He, S. (2007). Processing of invisible stimuli: Advantage of upright faces and recognizable words in overcoming interocular suppression. *Psychological Science*, 18(4), 349–355. <https://doi.org/10.1111/j.1467-9280.2007.01902.x>
- Kleiner, M., Brainard, D., Pelli, D., Ingling, A., Murray, R., & Broussard, C. (2007). What's new in psychtoolbox-3. *Perception*, 36(14), 1–16. <https://doi.org/10.1068/v070821>
- Moors, P., Wagemans, J., & de Wit, L. (2016a). Faces in commonly experienced configurations enter awareness faster due to their curvature relative to fixation. *PeerJ*, 4, e1565. <https://doi.org/10.7717/peerj.1565>
- Moors, P., Wagemans, J., van Ee, R., & de Wit, L. (2016b). No evidence for surface organization in Kanizsa configurations during continuous flash suppression. *Attention, Perception, & Psychophysics*, 78(3), 902–914. <https://doi.org/10.3758/s13414-015-1043-x>
- Moors, P., Wagemans, J., van Ee, R., & de Wit, L. (2016c). Scene integration without awareness: No conclusive evidence for processing scene congruency during continuous flash suppression. *Psychological Science*, 27(7), 45–56. <https://doi.org/10.1177/0956797616642525>
- Peirce, J. (2006). Psychopy—psychophysics software in python. *Journal of Neuroscience Methods*, 162, 8–13. <https://doi.org/10.1016/j.jneumeth.2006.11.017>

- Peirce, J. (2009). Generating stimuli for neuroscience using psychopy. *Frontiers in Neuroinformatics*, 2, 10. <https://doi.org/10.3389/neuro.11.010.2008>
- Pelli, D. (1997). The Video Toolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, 10(4), 437–442. <https://doi.org/10.1163/156856897X00366>
- Stein, T., Thoma, V., & Sterzer, P. (2015). Priming of object detection under continuous flash suppression depends on attention but not on part-whole configuration. *Journal of Vision*, 15(3), 15. <https://doi.org/10.1167/15.3.15>
- Tsuchiya, N., & Koch, C. (2005). Continuous flash suppression reduces negative afterimages. *Nature Neuroscience*, 8(8), 1096–1101. <https://doi.org/10.1038/nn1500>
- Tsuchiya, N., Koch, C., Gilroy, L. A., & Blake, R. (2006). Depth of interocular suppression associated with continuous flash suppression, flash suppression, and binocular rivalry. *Journal of Vision*, 6(10), 6. <https://doi.org/10.1167/6.10.6>
- Yang, E., & Blake, R. (2012). Deconstructing continuous flash suppression. *Journal of Vision*, 12(3), 8. <https://doi.org/10.1167/12.3.8>
- Yang, Y. H., & Yeh, S. L. (2011). Accessing the meaning of invisible words. *Consciousness and Cognition*, 20(2), 223–233. <https://doi.org/10.1016/j.concog.2010.07.005>
- Zhu, W., Drewes, J., & Melcher, D. (2016). Time for awareness: The influence of temporal properties of the mask on continuous flash suppression effectiveness. *PLoS ONE*, 11(7), 1–15. <https://doi.org/10.1371/journal.pone.0159206>