CFS-Crafter Help Information

CFS-Crafter is an open-source MATLAB application designed for fine control manipulation and analyses of CFS stimuli. As the application does not require prior expertise in image processing and analysis, it provides an accessible platform for improving stimulus control, increasing comprehension of CFS findings and generating more effective CFS animations.

1. Screen Information

Information of the screen at which the user intends to show the stimuli needed to be manually entered. Which would provide the basis for calculating of certain basic parameters. The user can only proceed to the next step after filling in all the required screen information.

1.1. Screen Refresh Rate (Hz)

Screen Refresh Rate should be set to the refresh rate of the screen on which the stimulus will be presented. Incorrect screen Refresh rate will lead to error in frame rate of created stimuli and other temporal features.

1.2. Viewing Distance (cm)

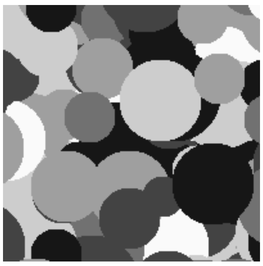
Viewing Distance is the distance between the participants' eyes to the screen during the experiment. Incorrect viewing distance will lead to errors in the calculation of parameters related to the degree of visual angle.

1.3. Screen Resolutions (pixels) & Screen Dimensions (cm)

CFS-Crafter reads the *resolution* and *dimension* of the current screen and input as the default value. If the user intends to use the stimuli on a different screen, these values must be changed manually. Error in this information will result in errors in spatial frequency and other spatial features.

2. Choose Function

2.1. Creation

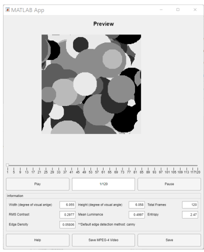


Creation allows the user to create various types of *CFS-Crafter-Mask*, which can be saved as a .mat file with related information or as a .mp4 video file.

2.1.1. Traced Pattern

Details about *traced item patterned masks* in **Creation**.

2.1.2. Stimuli Preview

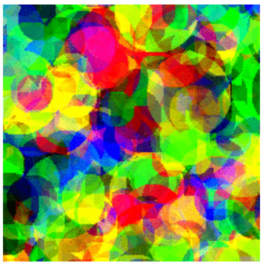


Details about preview & saving *CFS-Crafter-Masks* & *Image*.

2.2. Conversion

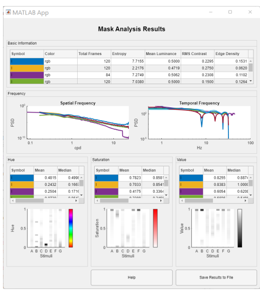
Conversion allows the user to convert a set of images to a *CFS-Crafter-Mask*, such that it can be modified or analysed using **CFS-Crafter**.

2.3. Modification



Modification allows the user to apply spatial/temporal/orientation filtering and phase scramble to *CFS-Crafter-Mask* or *Image*.

2.4. Analysis



Analysis allows the user to analyse and compare the descriptive statistics, spatiotemporal frequencies & colour contents of multiple stimuli (*CFS-Crafter-Mask* or *Image*).

2.4.1. Edge Detection

Details about edge deteciton in **Analysis**.

Homepage

CFS Mask Creation

Creation allows the user to create various types of *CFS-Crafter-Mask*. The created *CFS-Crafter-Mask* and the mask's basic information can be viewed & saved in the preview interface.

1. Mask Selection

CFS-Crafter allows the user to create CFS masks with a variety of different features. When choosing the type of masks, the user should select the options following a left-to-right order.

1.1. Patterned Element:

Patterned Element Mask consists of mondrian patterns with certain type of elements ranging from simple geometrics such as circles or squares, to more complicated elements such as objects traced from an image. Here is the detail of different type of patterns that's available:

1.1.1. Geometric Shape

The first type of pattern is *geometric shape*, which means the modrain pattern is consists of simple geometric shapes of various features, which include:

1.1.1.1. Pattern Shape

Pattern Shape decides the shape of each element in the mondrian masks, which can be *circle, square or diamond.*

The size of the patterns is proportional to the dimension of the stimuli. There are 5 different levels for patch size. For the *circle* patch, the radius is from 8% to 18% of the maximum dimension of the stimuli. The size of *square* and *diamond* patches are set such that they have roughly the same area as the *circle* patches.

Fig.1a - Circle Mondrain Pattern

Fig.1b - Square Mondrain Pattern

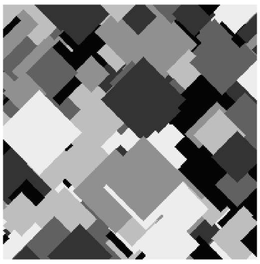


Fig.1c - Diamond Mondrain Pattern

1.1.1.2. Pattern Fill

The fill of the pattern can be either *solid* colours or *noise* patterns.

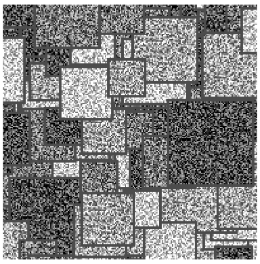


Fig.2a - Circle Mondrain Pattern with Solid Fill

Fig.2b - Square Mondrain Pattern with White Noise Fill

1.1.1.3. Pattern Colour

The *Solid filled pattern* can be either in *grayscale* or in *RGB* colour.

For the *grayscale* filled pattern, there are 6 different luminance levels used (0; 0.2; 0.4; 0.6; 0.8; 1.0) before RMS contrast & mean luminance adjustment.

For *RGB* colour filled pattern, there are 5 differet colours([0 0 0]; [1 0 0]; [0 1 0]; [0 0 1]; [1 1 0]) used before RMS contrast & mean luminance adjustment.

Fig.3a - Grayscale Mondrain Pattern

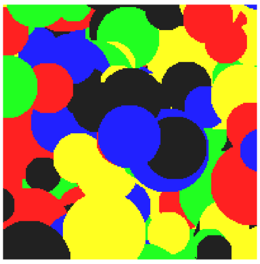


Fig.3b - Rgb Colored Mondrain Pattern

1.1.1.4. Pattern Noise Type

Noise filled pattern will be grayscaled *White Noise* or *Pink Noise.*

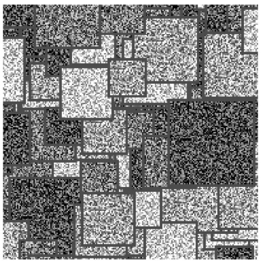


Fig.4a - White Noise Filled Pattern

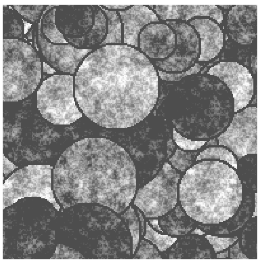


Fig.4b - Pink Noise Filled Pattern

1.1.2: Traced Items

The user can also trace up to 5 different faces/objects and create mondrian patterns from these traced items. **CFS-Crafter** allows users to trace up to 5 different objects/faces to create a stimuli sequence. Click here for help on item tracing.

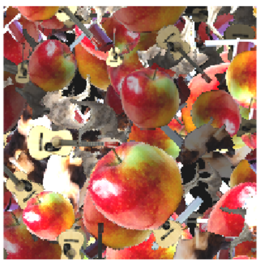


Fig.5a - Traced Face Mondrain Pattern *Fig.5b - Traced Object Mondrain Pattern*

1.2. Noise Masks

Masks consists of dynamic sequence of white noise or pink noise.

1.2.1. White Noise

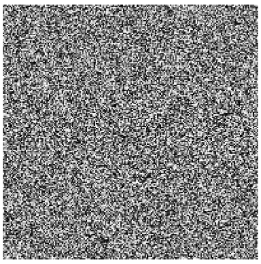


Fig.6a - White Noise Mask

White Noise consists of random intensity between 0 to 1 at all pixels before RMS contrast & mean luminance adjustments, which leads to roughly constant power spectral density (PSD) across all frequencies.

1.2.2. Pink Noise

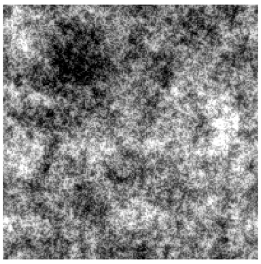


Fig.6b - Pink Noise Mask

Pink Noise has PSD inversely proportional to the frequency. Which closely resemble the natural scene.

2. Mask Creation Parameters

Users need to enter the parameters for the mask they want to create.

2.1. Mask Dimensions:

Width & height in pixels, the size in the degree of visual angle will be calculated & shown in the preview interface.

2.2. Mask Update Rate & Mask Sequence Duration:

The *Mask Update Rate* is the rate at which the images is updated in the stimuli sequence.

Notice *Mask Update Rate* is not the same as *Temporal Frequency*, even through they both in the unit of Hertz. Click here for details on *Temporal Frequency* and means to manipulate it.

Mask Sequence Duration is the duration of the created CFS mask stimulus with unit of second. Notice the duration input here should be relative short (a few seconds at most), since the increase in duration & frames will drastically increase the size of the stimuli array & the processing speed. If the experiment design requires long suppression time, consider looping through one or multiple short sequences created by **CFS-Crafter**.

2.3. RMS Contrast & Mean Luminance:

The <u>*Root Mean Square(RMS) Contrast*</u> of an image is the average standard deviation of the intensity of each pixel. For RGB coloured stimuli each frame is first converted to grayscale and then the RMS contrast is calculated. Here the RMS contrast entered is the targeted average RMS contrast for the entire stimuli sequence.

Similarly, the mean luminance of an image is the average value of intneisty across all the pixels. Here the mean luminance is the targeted average luminance of the created stimuli sequence.

2.2.1. Image Clipping

The figure below shows cross-sections of the same mondrain patterned masks with different levels of RMS contrast and mean luminance setting. As what we see here, when the target RMS contrast is large, or when the target mean luminance is very extreme, or both at the same time, part of the pixels will have intensity fall outside of the [0,1] range, in order to present the stimuli, these pixels must have their intensity clipped back to 0 or 1. This process is called <u>*clipping*</u> of the image. And clipping of the image will result in the actual RMS Contrast & luminance of the stimuli being different from the user's setting.

RMS Contrast Adjustment & Clipping

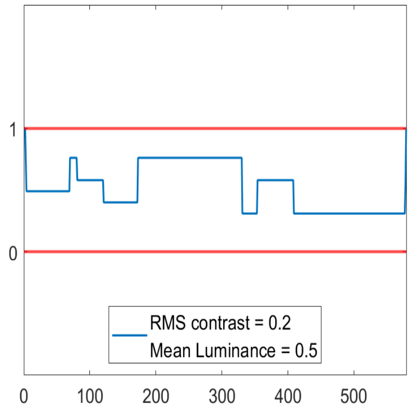


Fig.7a - Original

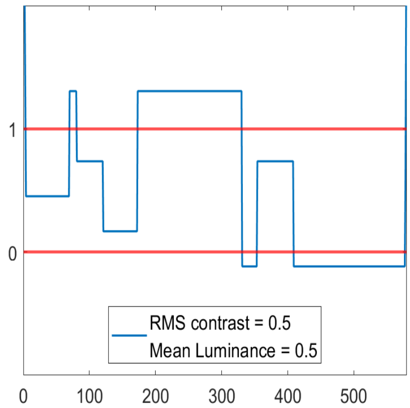


Fig.7b - After RMS contrast adjustment, before clipping

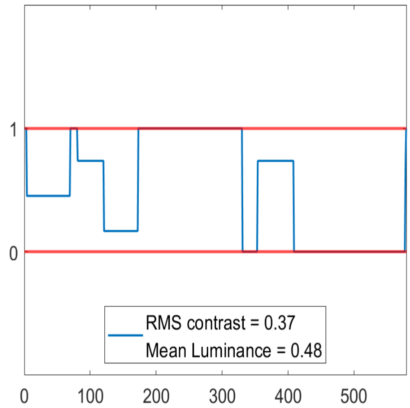


Fig.7c - After RMS contrast adjustment & clipping

Mean Luminance Adjustment & Clipping

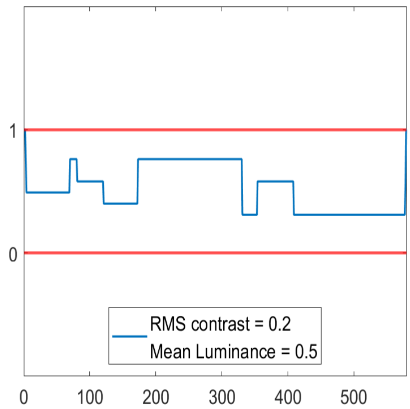


Fig.8a - Original

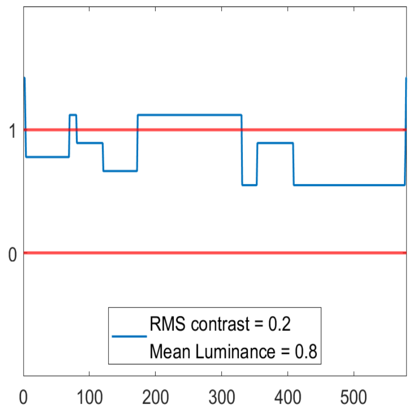


Fig.8b - After mean luminance adjustment, before clipping

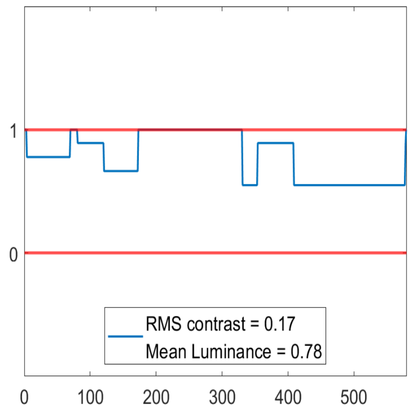


Fig.8c - After mean luminance adjustment & clipping

[Homepage](#)

[Back to Creation](#)

Traced Pattern

CFS-Crafter allows users to trace up to 5 different objects/faces to create a stimuli sequence. The tracing method is based on Matlab's ROI Based processing.

Two types of items were supported here: *Face & Objects.*

1. Face

Face tracing uses an *Elliptical ROI* which allows the users to select the face from the provided source image.

Mean Luminance of all patches traced will be adjusted to 4 different levels (0.2, 0.4, 0.6, 0.8), and the *Maximum Dimension of each patch* will be adjusted to 6 different level (16%, 20%, 24%, 28%, 32%, 36%) of the *Maximum Dimension of the stimuli*.

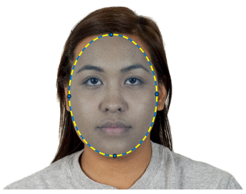


Fig.1a - Elliptical ROI

Fig.1b - Traced face patterns

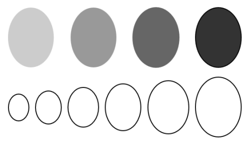


Fig.1c - Mean luminance Levels & Sizes

Fig.1d - Traced Face Patterend Mondrian Mask

2. Object

Obejct Tracing is very similar to face tracing, except a *Assisted freehand ROI* was used to trace the edges of the object. In addition to the previously mentioned adjustment in terms of size and mean luminance level, the patches are also rotated to 8 different *Orientations*(0°, 45°, 90°, 135°,180°, 225°, 270°, 315°).

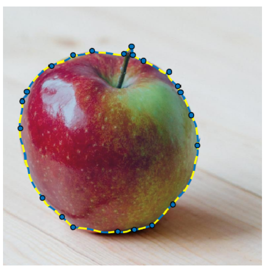


Fig.2a - Assisted-freehand ROI

Fig.2b - Traced object patterns

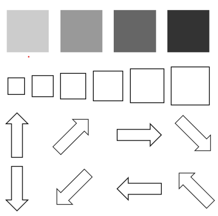


Fig.2c - Mean luminance Levels, Sizes & Orienatations

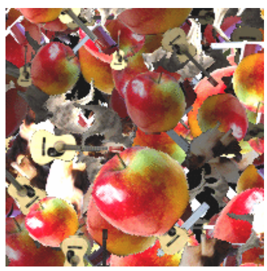


Fig.2d - Traced Object Patterend Mondrian Mask

When drawing the assisted-freehand ROI, the outline might not be exactly as desired. Users can right-click on the ROI edges to add *waypoints*, and then by dragging these waypoints the overall shape as well as details of the ROI can be adjusted.

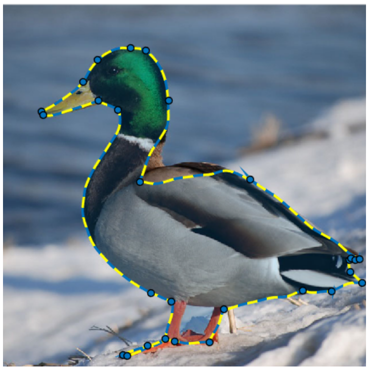


Fig.3a - Assisted-freehand ROI

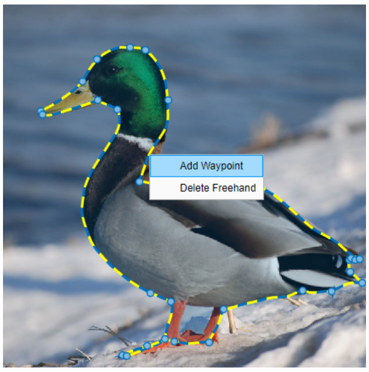


Fig.3b - Right click to add waypoints on the ROI

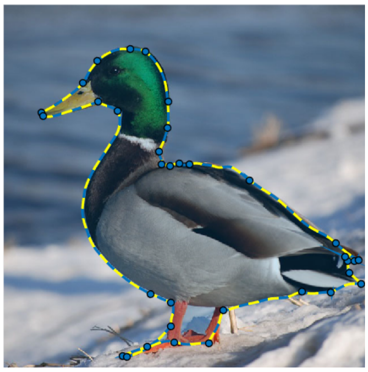


Fig.3c - ROI after adjustment

Homepage

Back to Creation

Stimuli Preview

1. Stimuli Format

In order to be recognized universally by different components of **CFS-Crafter**, mask stimuli created by **Creation**, **Conversion** and **Modification** are all stored in the same format, which is a MATLAB structure *stimuli* with the following fields:

1.1. Label

The *Label* field is used to identify the type of stimuli. All stimuli created by **CFS-Crafter** will have the label *CFS-Crafter-Mask*, except the image file processed by **Modification**, which will be labelled as *CFS-Crafter-Image*. Stimuli labelled as *CFS-Crafter-Image* will not be subject to temporal filtering and temporal frequency analysis.

1.2. Stimuli Array

For *CFS-Crafter-Mask*, the *stimuli_array* field contains a 4-D matrix, with dimensions representing *Rows*, *Columns*, *RGB Color Channels* & *Frames*. Notice that for *Grayscale* stimuli, the *Color Channel* have length 1 instead of 3.

For *CFS-Crafter-Image*, *stimuli_array* will be a 3-D matrix, with dimensions representing *Rows*, *Columns* & *RGB Color Channels*.

1.3. Screen Information

The *screen_info* contains information entered in the main interface of **CFS-Crafter**. This information can help users to implement the stimuli in experiments correctly. Click here for details on the following fields of *screen_info:*

- refresh_rate
- viewing_distance
- screen_width_cm
- screen_height_cm
- screen_width_pixel
- screen_height_pixel

1.4. Stimuli Property

The basic descriptive statistics of the created stimuli is stored in *stimuli_property*, and shown in the **Preview** interface for the user to inspect. For more detailed information the user can use the **Analysis** function of **CFS-Crafter.**

- Click here for detials on the following *stimuli_property* entered in **Creation Interface**
  - update_rate
  - sequence duration
  - stimuli_width_degree
  - stimuli_width_degree
  - stimuli_width_pixel
  - stimuli_height_pixel

- rms_contrast
- mean_luminance
- Click here for detials on the following *stimuli_property* obtained by **CFS-Crafter**
  - total_frames
  - color
  - entropy
- Click here for detials on *stimuli_property* related to **Edge Detection**
  - edge_density
  - edge_detection_method
  - edge_detection_threshold

2. Preview & Save Stimuli

When previewing *CFS-Crafter-Mask*, the **preview** interface will allow the users to play & pause the stimuli sequence, or drag it across frames using the slider. Notice that due to the limitation of MATLAB, the frame rate will NOT be accurate when playing the stimuli in the preview interface. If the user wants to view the stimuli with the correct frame rate, it is better to save the stimuli as a .mp4 file. The user can also save the *CFS-Crafter-Mask* as a .mat file with the fields mentioned above. Which then can be passed on to **modification** or **Analysis**, or to be inspected and used in MATLAB.

For *CFS-Crafter-Image* stimuli, similar options are available, except that it can be saved as a .jpeg image file instead of a .mp4 video file.

Homepage

Image Conversion

The **Conversion** function allows the user to convert a series of images into a *CFS-Crafter-Mask*. After which the user can then proceed to modify this sequence with modification function, or to compare with other *CFS-Crafter-Mask* or *images* using the analysis function

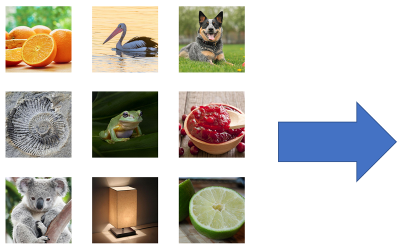


Fig.1a - Images

Fig.1b - Converted Image Sequence Stimuli

1. Select Images

By clicking the add images button, the user can select multiple images to upload to **CFS-Crafter**. The order of the image will be based on the naming of the image file. Moreover, the user can delete any unwanted images by checking the *selection* column and clicking the *delete selection* button.

2. Conversion Parameter

After selecting all the images, the user needs to input the necessary parameters.

2.1. Stimuli Height & Stimuli Width

Firstly, the user needs to set the size of the output stimuli in pixels. All the images input will be resized to these dimensions. **CFS-Crafter** allows images of different sizes to be uploaded, but notice that resizing might lead to changes in aspect ratio. Hence it is recommended to crop the images beforehand to the same aspect ratio.

2.2. Image Update Rate & Stimuli Duration

The *Image Update Rate* is the rate at which the images is updated in the stimuli sequence, similar to *Mask Update Rate* in **Creation**. Click here for more details.

After entering the image update rate, CFS crafter will calculate the duration of the stimuli sequence based on it and the number of images selected, which will then be shown in *stimuli duration*.

2.3. RMS Contrast & Mean Luminance

Similar to creation, converison also allows the user to set the target *RMS contrast* and *mean luminace* value. Click here for details.

Homepage

Modification

The spatiotemporal attributes of CFS masks & targets have a strong influence on the strength of visual suppression. **CFS-Crafter** provides the user with the ability to fine-control these aspects of the masks and targets. Hence, it allows the user to tune the stimuli to best suit their experimental design. The modified stimuli can be save as a *CFS-Crafter-Mask* .mat file, or as a .mp4 video file/.jpg image file. (Depends on the type of the original stimuli) Click here for detials on previewing and saving the modified stimuli.

1. Spatial Fitlering

The change in pixel luminance in the spatial domain can be decomposed into sinusoidal compoents with different *Spatial Frequency*. Higher spatial frequency components can be linked to drastic changes in pixel luminance such as the edges, while lower spatial frequency components are usually linked to solid area or area with gradual luminance change.

In order to manipulate the spatial frequency content of the stimuli, **CFS-Crafter** will firstly apply a 2-D Fast Fourier transform along the two spatial dimensions of the 4D matrix (*CFS-Crafter-Mask*) or 3D matrix(for *Image*). Then by applying a 2-D filter in the frequency domain and back-transform the filtered frequency-domain matrix to its spatial domain, a new stimulus with updated spatial attributes is created.

In **Modification**, the user need to specify the *Type* & *Methods* of the spatial filter they want to apply, as well as the *cutoff frequency* in cycles/degree(cpd). (And for *Butterworth* filter, the *order* also needs to be specified.)

1.1. Filter Type

CFS-Crafter offer 3 different type of spatial filter: *low-pass*, *high-pass* and *band-pass*.

The figures below are example of outcome of each type of spatial filtering. One frame of the original *CFS-Crafter-Mask* is shown in Fig.1a. *Low-pass* spatial filtering will remove the higher frequency component (i.e. the edges) while preserving the lower frequecny component (i.e. the gray blobs). The *High-pass* filtering will preserved the edges and eliminate the lower frequency component, as can be seen in Fig.1c. The *Band-pass* spatial filtering will preserve the frequency component between the lower & higher *cutoff frequency*, and eliminate the rest, as shown in Fig.1d.

Fig.1a - Original Stimuli

Fig.1b - After Low-pass Spatial Filtering

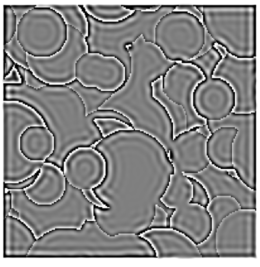


Fig.1c - After High-pass Spatial Filtering

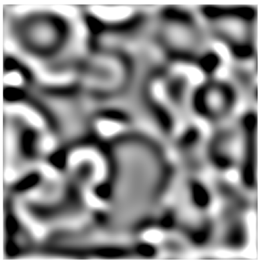


Fig.1d - After Band-pass Spatial Filtering

1.2. Filter Method

Different methods(algorithms) of filtering have different profiles, which will lead to differences in the frequency content of modified stimuli. Fig 2 shows the cross-section of low-pass filters with different filtering methods differs in their profile with the same cutoff frequency.

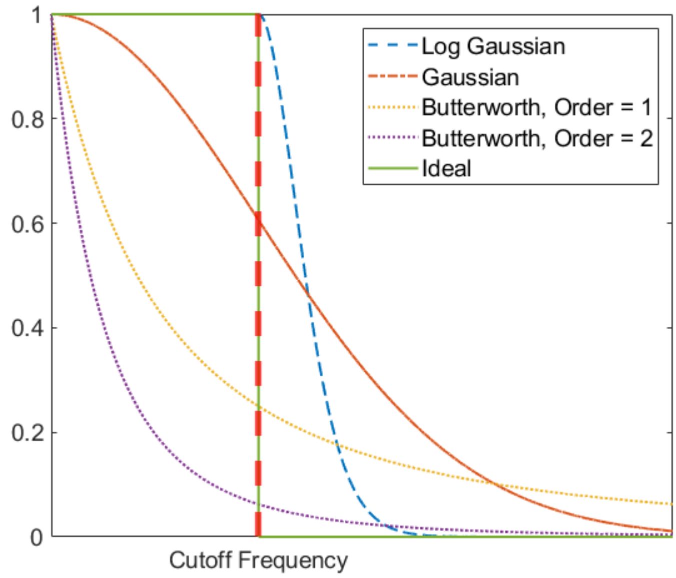


Fig.2 - Different Type of low-pass filter with the same cutoff frequency.

The difference in their profile will lead to different appearances in the filtered image. *Ideal Filter* preserves all frequency components that are lower than the *Cutoff Frequency* and eleiminate all that are above. Due to this discontinuity in the frequency domain, a ringing artefact will appear, as shown in Fig 3e. *Gaussian* and *Butterworth* filter does not have this artefact attribute to their smooth transition, but this smooth transition does lead to loss in lower frequency content(Fig.3c & Fig. 3d). *Log Gaussian Filter* introduce a smooth roll-off at the *Cutoff Frequency*, which mitigates the ringing artefact, and also preserves the lower frequency(Fig. 3b)

Fig.3a - Original Stimuli

Fig.3b - After low-pass Log Gaussian filtering

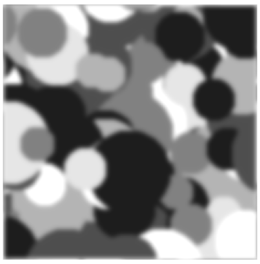


Fig.3c - After low-pass Gaussian filtering

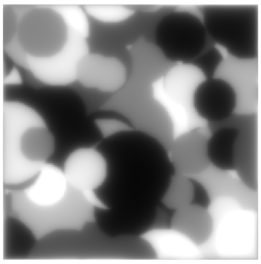


Fig.3d - After low-pass Butterworth filtering

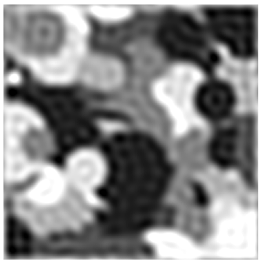


Fig.3e - After low-pass ideal filtering

2. Temporal Filtering

The *Temporal Frequency* of a stimulus refers to the frequency of luminance change in each pixel. Notice this is not equivalent to *Mask Update Rate* in **Creation** or *Image Update Rate* in **Conversion**. The latter two refers to the rate at which a new pattern is present, and can be seen as coarsely to the temporal frequency. However, for each individual pixel, their luminance might stay the same or trend in the same direction for multiple frames(low temporal frequency), or reversing direction from frame to frame (high temporal Frequency). Hence, even though the manipulation of *Update Rate* can shift the overall temporal content of the stimuli ot the desired direction, it cannot guarantee the temporal frequency across all pixels.

To achieve accurate control over the temporal content of the stimuli, **CFS-Crafter** will apply a 1-D Fast Fourier transform along the fourth dimension of the 4D matrix and then filtering out the unwanted frequency components using 1-D filters.

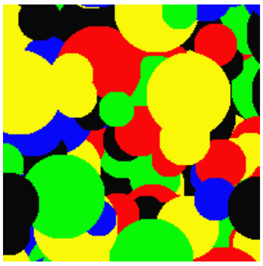


Fig.4a - Original Stimuli

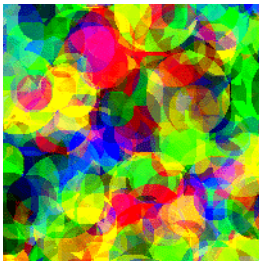


Fig.4b - After low-pass temporal filtering

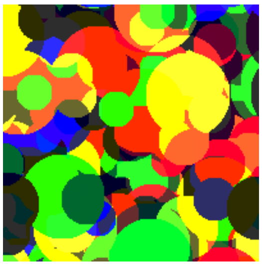


Fig.4c - After high-pass temporal filtering

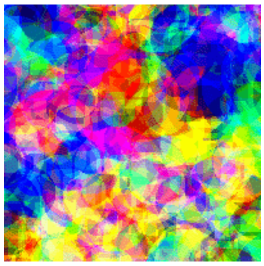


Fig.4d - After band-pass temporal filtering

3. Orientation Filtering

Orientation Filtering is achieved by applying a directional Gaussian filter in the frequency domain. Orientation of the filter decides the direction around which the gaussian will be constructed, and sigma decides the width of the Gaussian. In **CFS Crafter**, orientation indicates the **orientation of the pattern** (which starts at vertical and goes clockwise), and it is perpendicular to the orientaion of the filter, as can be seen in Fig.4b - Fig.4e.

Fig.4a - Original Stimuli

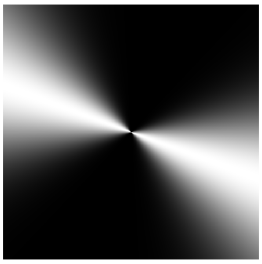


Fig.4b - Oreintation Filter (Orientation = 20°, σ = 20°)

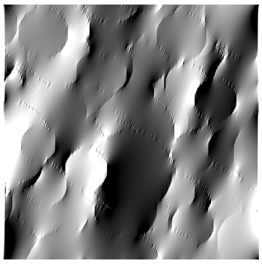


Fig.4c - After Oreintation Filtering (Orientation = 20°, σ = 20°)

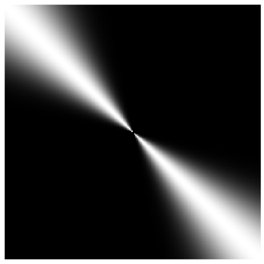


Fig.4d - Orientation Filter (Orientation = 45°, σ = 10°)

Fig.4e - After Oreintation Filtering (Orientation = 45°, σ = 10°)

4. Phase Scramble

If the user wishes to preserve the spatial frequency content(i.e. the low-level spatial features), but destroy the structure of the stimuli, a randomised *phase* can be added in the frequency domain. This is called *Phase Scramble*.

4.1 Phase Scramble Index

Phase Scramble Index should be a number between 0 and 1. The randomised phase matrix (with value ranging from -π to π) will be multiplied by this index before being added to the frequency matrix. The *Phase Scramble Index* decides the proportion of the original phase that is left in the modified stimuli. As can be seen in Fig.5b, when the phase scramble is set to 1, no structure from the original stimuli is left. When the index is set to 0.5, there is some original structure remaining.

Fig.5a - Original Stimuli

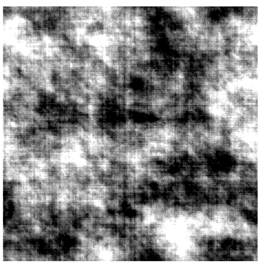


Fig.5b - Phase Scramble Index = 1

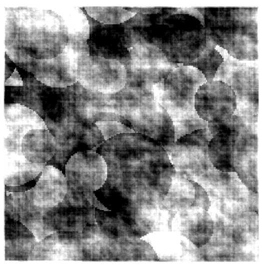


Fig.5c - Phase Scramble Index = 0.5

4.2 Frequency Range

Frequency Range decides the range of frequency that will be subject to the *Phase Scramble process*. When choosing the *Low-pass* option, the phase Scramble will only be applied to frequency components that's lower than the *Cutoff Frequency*, hence as can be seen in Fig.6c, the structure of the higher frequency (i.e. the edges) still remains intact, while the structure of the lower frequency component (i.e. the grayscale blob) is scrambled. The *High-pass* option only passes the higher frequency components to the phase scramble. Hence, the structure of the lower frequency remains unchanged. When the *Band-pass* option is selected, phase scramble only applies to the frequency components between the lower and higher *Cutoff frequencies*, which leaves the original structure in the higher and lower frequency range, as can be observed in Fig.6d, the structure of the edges(i.e. high frequency) and the base colour pattern(i.e. the general shape & location of dark and light blobs) remains the same as the original stimuli, but the frequency in-between lost its structure.

Fig.6a - Original Stimuli

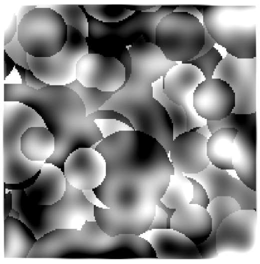


Fig.6b - After Low-pass Phase Scramble

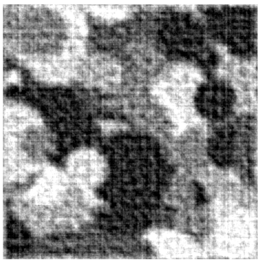


Fig.6c - After High-pass Phase Scramble

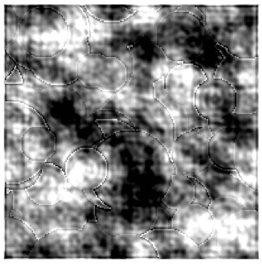


Fig.6d - After Band-pass Phase Scramble

4.3 Phase scrmable, Spatial Filtering & Orientation Filtering

Phase Scramble, *Spatial Filtering* & *Oreintation Filtering* are all modifications made to the spatial content of the sitmuli, which complicates the outcome stimuli when performing together. In **CFS Crafter**, phase scramble and oreintation filtering are not allowed to perform together. And when spatial filtering is applied, phase scrambling can only be applied to all frequency components to avoid scrambling the frequency that was filtered out.

Homepage

Analsysis

1. Select Stimuli

To **Analysis** function allows user to analysing and comparing different stimuli (*CFS-Crafter-Mask*, *CFS-Crafter-Image* or image file in the format of .png, .jpg, .jpeg & .tiff). These stimuli can be added by clicking the **Add Stimuli** button.

Notice that the selected image file will be first converted to *CFS-Crafter-Image* using the *screen_info* entered in the main interface. Hence it is important to ensure the accuracy of the screen information.

2. Select Analysis to be done

The analysis in **CFS-Crafter** is divided into multiple parts. Users can select which analysis is necessary to decrease the processing time. The speed of the analysis will dependent on the number of stimuli, the type of stimuli (sequences or images), and the type of analysis selected.

2.1. Basic Information

Basic Information includes:

- **Colour:**

  This shows whether the selected stimuli are RGB coloured or grayscale. (Images will all be shown as RGB coloured)

- **Total Frames:**

  Number of frames

- **Entropy:**

  The average entropy for grayscale stimuli. Notice for RGB coloured stimuli, the stimuli are firstly converted to grayscale and then the entropy is calculated.

- **RMS Contrast & Mean Luminance**
  RMS(Root-mean-square) Contrast and average luminance (intensity) of the stimuli. For details on RMS Contrast & Mean Luminance click here.

- **Edge Density**

  The edge density is calculated based on the selected edge detection method and threshold. For details on edge detection click here.

2.2. Frequency

Frequency analysis will provide PSD (power spectral density) plot for spatial frequency and temporal frequency.

2.2.1. Spatial Frequency

For spatial frequency analysis, the PSD is calculated based on the radial average power spectral density after 2D fast fourier transform. (Rgb coloured stimuli are first converted to grayscale and then undergo the analysis)

2.2.2. Temporal Frequency

For temporal frequency analysis, similar porcess is performed. Notice here that only *CFS-Crafter-Mask* will be subjected to temporal frequency analysis, *CFS-Crafter-Image* & image file will be ignored.

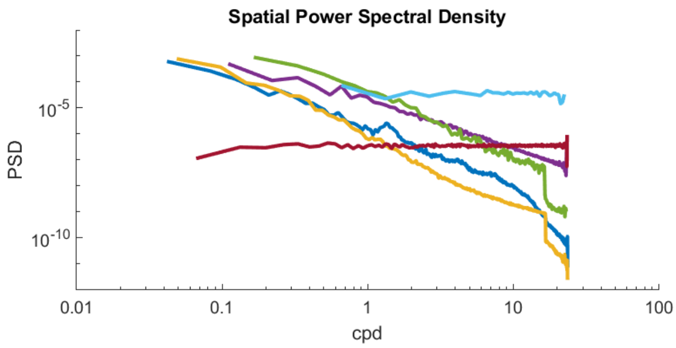


Fig.1a - Spatial Power Spectral Plot

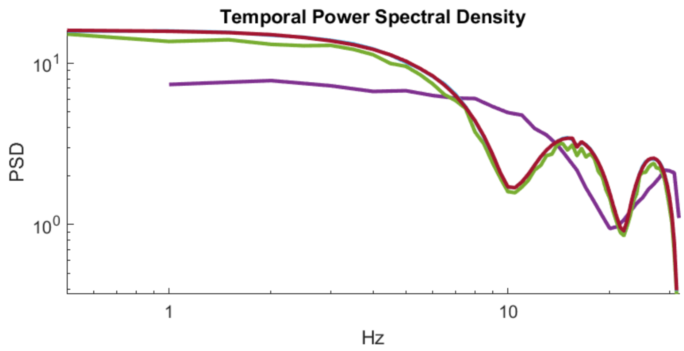


Fig.1b - Temporal Power Spectral Plot (with images ignored)

2.3. Colour

Colour analysis will ignore grayscale stimuli. For rgb stimuli, the analysis consists of two parts: descriptive statics and probability density plot.

As seen in Fig. the x-axis shows each stimulus that was subject to analysis. The y axis shows the value (Hue, Saturation & Value respectively), and the shade of the bar at each y-value respond to the probability density of each stimuli at the corresponding value (the legend bar on the right of each graph shows the feature corresponding to each value), with a darker shade corresponding to a higher density.

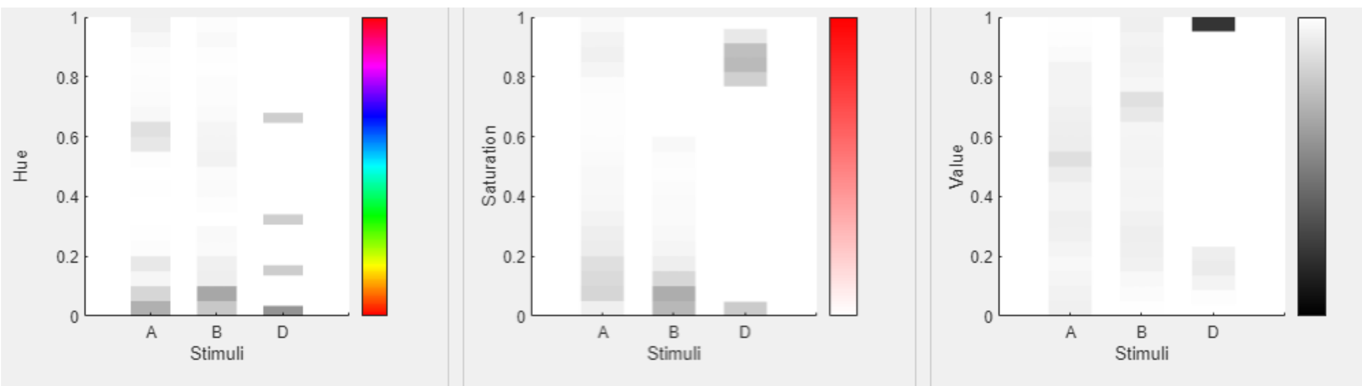


Fig.2 - HSV(Hue, Saturation & Value from left to right) Probability Density Plot

3. Analysis Results

Based on the type of analysis selected by the user, the results will be shown in a seperate interface for the user to view.

The graphs can be saved by using the context menu on the graph. The user can also save the analysis results as a *CFS-Crafter-Analysis-Results* .mat file. Which contains the following information for each stimulus:

- **basic_info:**

  all information included in the **Basic Information** table

- **frequency_results**

  frequency(x value) & PSD (y value) for Spatial & Temporal PSD plot.

- **hsv_results**

  hsv array for each stimuli, bin_counts, bind_edges & descriptive statistics (mean, median, mode & range) for hue, saturation & value respectively.

Edge Detection

CFS Crafter uses MATLAB's Image Processing Toolbox to perform edge detection on selected *CFS-Crafter-Masks*, *CFS-Crafter-Image* & image files. Here we provide the basic procedure of how to use the edge detection in ***CFS Crafter***, for details on the edge detection algorithm please refer to MATLAB's Help Center.

1. Edge Density

Edge Density is the percentage of pixels that are identified as edges with respect to total number of pixels in the image.

When creating a *CFS-Crafter-Masks* or *CFS-Crafter-Image*, ***CFS Crafter*** will perform a default edge detection with *canny* method and default threshold chosen by MATLAB. The average *edge density* over all frames, *edge detection method* and *threshold* will be stored in *stimuli_property*.

2. Edge Detection Method & Threshold

Suppose the user is particularly interested in exploring different edge detection methods & thresholds, and are interested in the binary edge images. In that case, the user can alter the edge detection method & threshold in ***Analysis***, and select ***Preview Edges*** to preview and save the edges for all selected stimuli and updated edge density.

There's 4 methods to choose from in ***CFS Crafter***:

- **canny & approxcanny**

  canny & *approxcanny* use 2 thresholds, edges are found by looking for the local maxima of the gradient. The edges with strength smaller than the lower threshold will be ignored, and those with strength higher than the higher edge will be preserved. And those with edge strength between lower & higher limits will only be preserved if they are connected to the strong edges.

  When specifying the threshold, the user can input either 1 scaler as the *higher threshold* valu. And in that case the lower limit will be set as *0.4*higher threshold*. Otherwise the user can specify both thresholds with a comma separating the two.(i.e. *low,high*)

- **Sobel**

  Find maximum derivative using Sobel approximateion. Edges smaller than the threshold value will be ignored.

- **log**

  Finds edges by looking for zero-crossings after filtering I with a Laplacian of Gaussian (LoG) filter. If the user specify the threshold value 0, then the output image has closed contours because it includes all the zero-crossings in the input image

3. Edge Preview

If the ***Preview Edges*** check box is selected in ***Analysis***, an ***Edge Preview*** interface will be opened after analysis. In this interface, the user is presented with the preivew (fisrt frame) of the binary edge image of each stimulus , edge detection method & threshold used, and average edge density of each stimulus. User can also save the edges as a .mat file, which contains the above-mentioned edge information and the full binary edge array for each stimulus.