UNIVERSITÀ DEGLI STUDI DI GENOVA
DIBRIS

A population of neuromorphic operators: V1-MT pathway

Fulvio Missoni
August 31, 2021

This software module implements the architecture described in Supplementary Material (see Technical Report). The architecture consists in two population, organized hierarchically, that model the V1-MT pathway of the dorsal visual stream:

1. Disparity detectors that mimic the V1 complex cells behavior.
2. Motion-in-depth (MID) detectors that represent a population of MT-like cells.

The main functionalities implemented are two: the analysis of stereo video by two population (`pop_flowV1MT()` method) and the tuning analysis of the population with typically used stimuli in psychophysical experiments (`main_pop_tuning_curves()` method).

The code is organized as follows. The *main* scripts are in the homepage. The "FUNCTIONS" directory contains all the functions necessary for the operation of the simulator. The stereo input are stored in the "IMAGES" folder while the "FILTERS" directory contains the .mat files in which the 1D Gabor components of different Gabor filters are stored. These components are needed to make a 2D oriented Gabor filter (see `filter_gabor_space2D()` method for more details). Finally, the tuning data of the population are available ("SIMULATION" folder). In all methods the population activity is defined as multidimensional matrix. For instance, a population of `nr`×`nc`×`No`×`Np`×`Nv` MT cells—where, `nr`×`nc` are the dimensions of the image frame, `No` is the number of the orientation channels, `Np` is the number of the phase-shifts, `Nv` is the number of velocity channels—is represented by a matrix of the that size. The parameters of the population is provided by a struct variable (`param`) in according of the following table:

Struct field	Description
`param.spatFreq`	spatial frequency of spatial Gabor filter in [cycle/pix]]
`param.ocDom`	ocular dominance factor
`param.phShift`	phase shifts for each orientation channel
`param.nOrient`	number of orientation channels
`param.prefVel`	vector of preferred velocity in [pix/frame]
`param.tempFilt`	model of temporal component of the spatio-temporal RFs ('gabor', 'exp_decay', 'adelson_bergen')
`param.spatialFilt`	name of .mat file that contains 1D gabor filter components that are necessary to construct the 2D oriented Gabor filters (see Supplementary Materials for more detail)
`param.sample`	number of samples of the spatial filter
`param.normParam`	normalization factors used in the two stages (layer 2 and layer 3)

Table 1: Description of struct fields

1 Distributed representation of stereo-input

```
[EMT, EC21, EC22, varargout] = pop_flowV1MT(II,param,tun_flag)
```

Compute the population activity of disparity (`EC21`, `EC22`) and MID (`EMT`) detectors in response of a visual stimulus `II`. Moreover, it is possible to evaluate the activity of the populations of the lower layers (es. `varargout{1}` = C1-layer response).

Ouput arguments	Description
`EMT`	MT neurons population activity
`EC21`	C21 neurons population activity
`EC22`	C22 neurons population activity
`vargout`	activity of the populations of the lower layers (es. `varargout{1}` = C1-layer response)
Input arguments	**Description**
`II`	is a stereo clip organized as a stack of stereo frames (`[sy, sx, nframes, 2] = size(II)`)
`param`	see Tab 1
`tun_flag`	if set equal to '1' then the function returns the excitation value of the cell positioned on the center of the frame with preferred orientation equal to 0 degrees, for each channel (phase shift and preferred velocity)

1.1 Sub-methods

First step, compute the 3D convolution between dynamic stereo input `II` and bank of separable spatio-temporal filters.

- `F_new = filt_gabor_space2D(II,fname,flag,DC_thr)`

Monocular multi-orientation representation of the stereo input (`II`). `F_new` is 1×2 cell array where the first- and second-element are, respectively, the real and the imaginary part of the convolution. More in detail, each cell contains a matrix of size `sy`×`sx`×`No`×`n_frame`×`side` where, `nr`×`nc` are the dimensions of the image frame, `No` is the number of the orientation channels (8, that span from 0 to π), `n_frame` is the number of the frames of the stimulus and `side` identifies, respectively, the left and the right side.

Argument	Description
`II`	stereo image (input) of size `nr`×`nc`×`n_frame`×`side`
`fname`	name of .mat file that contains the 1D Gabor components needed to obtain 8 orientation channels ($\theta = 0 : \pi/8 : \pi - \pi/8$)
`flag`	is set to "1" then is considered only the vertical orientation ($\theta = 0$)
`DC_thr`	energy threshold value (default equal to "0")

- `F_new = filt_time(F,conv_type,filter_type,v,k0)`

Convolution with temporal component of the separable spatio-temporal filter. `F_new` is 1×4 cell array that contains the real and the imaginary part of the response (first- and second-element) and their temporal derivatives (third- and fourth-element). Moreover, this method introduces the velocity channels. Each cell contains a multidimensional matrix of size: `sy`×`sx`×`No`×`n_frame`×`side`×`n_vel`.

Argument	Description
`F`	1×2 cell array that contains the monocular spatial component response of `n_orient` orientation channels
`conv_type`	type of convolution ('full', 'same', 'valid'). (see `conv2()` documentation).
`filter_type`	temporal filter choice ('gabor', 'exp_decay', 'adelson-bergen') (see Supplementary Material, Technical Report)
`v`	vector of preferred velocities
`k_0`	gabor spatial frequency in [cycle/pix]

Second step, computing distributed multi-channel monocular representation relative to right eye.

- `[G_even, G_odd] = shift_in_phase(F_real,F_imag,ph_shift)`

Computes of neurons' response with different phase-shift values.

Argument	Description
`F_real,F_imag`	real and imaginary part of the response of a population of monocular separable spatio-temporale filters obtained on the previous steps.
`ph_shift`	No×Np matrix where No is the number of the orientation channels and Np is the number of phase-shift channels. Each row contains the phase-shift values repeated for each orientation channel.

1.2 EXAMPLE

An example script is provided (`MY_EXAMPLE.m`). In this example is shown how to define a population of disparity and MID detectors. The stimulus consists in the left and right view of a slanted 3D plane.

2 Analysis of population tuning

In this section is described how to make tuning analysis of a neural population. The simulator provides: disparity (both horizontal and vector disparity) motion in depth (MID) and monocular motion tuning analysis.

- `main_pop_tuning_curves()`

This method is the *main* of tuning analysis. Parameters of the population are initialized and the tuning analysis are chosen with if condition. For each analysis an appropriate method is provided to display data.

- `RDS_simulation(parameters,choice)`

The following method performs disparity tuning analysis with static Random Dot Stereograms (RDS). Two options are available: horizontal disparity tuning or full disparity vector analysis. Resulting population activity is stored in the `SIMULATIONS` folder. Such data can be displayed with the appropriate plot methods (`plot_disp_tun()` and `surf_disp_tun()`, see section 2.2 for more detail). RDS stimuli are generated during the simulation with `myRDS()` function (see section 2.1 for more detail).

Argument	Description
`param`	see Tab 1
`choice`	choice of the disparity analysis to do. '2D_disp' for the full vector disparity tuning analysis of the entire population of neurons. '1D_disp' for the horizontal disparity tuning analysis of the sub-population of cells with orientation fixed ($\theta = 0$).

- `MID_simulation(param)`

Execute MID tuning analysis. MID is simulated by means of couple of monocular gratings with different velocity stored in `IMAGES/Input`. Resulting activity map of each cells is plotted using `surf_MID_tun()`. Another grating stimulus can be constructed via `sinGrating()` method (see section 2.1 for more detail).

- `[varargout] = motion_popV1MT(param,stim)`

Operate monocular motion tuning analysis with two types of motion stimulus: sinusoidal gratings or plaid.

Output arguments	**Description**
`varargout`	optionally population activity matrix can be obtained. By default is collected in SIMULATIONS folder.
Input arguments	**Description**
`param`	see Tab.1
`stim`	see Tab.2

Stimulus properties are defined with a struct variable named as **stim**. Stimulus size and its spatial frequency are set in according to the RFs properties. Moreover luminance (channel of transparency) and contrast of two gratings are set by default at 0.5 value. All set properties are shown in the following table.

Field name	**Description**
`stim.type`	type of stimulus. Two option: moving sinusoidal gratings ('grat') or plaid ('plaid').
`stim.dur`	period of stimulation.
`stim.vgrat`	velocity of moving gratings in [pix/frame]. If a plaid stimuli is used two velocity must be set.
`stim.theta_g`	orientation of gratings in [rad]. If a plaid stimuli is used two orientation must be set.
`stim.vpld`	velocity of plaid stimuli in [pix/frame].
`stim.truetheta`	plaid's true orientation. Used only for plaid stimuli.

Table 2: Struct stim properties

2.1 Visual stimuli

- `[L, R] = myRDS(dc,dr,flag,myseed,varargin)`

Generate a Random Dot Stereogram (RDS) input. RDS is a stereo image where each pixel has (randomly) -1 or 1 as value. If the RDS is correlated then the two images (right and left) have the same seed but the right image is shifted of a certain value respect to the left image. The output of the function is the stereo image where `L` and `R` are respectively the left and the right image. The inputs of the function have the following meaning:

Argument	Description
`[dc, dr]`	displacement along columns and rows, respectively.
`flag`	integer variable that can be three values: '1' to generate correlated RDS, '2' to generate correlated RDS (background + foreground) and '3' to generate uncorrelated RDS.
`myseed`	seed of random number generator
`varargin{1}`	number of columns of frames (256 by default)
`varargin{2}`	number of rows of frames (256 by default)

- `II = sinGrating(sx,sy,v,N,K,k0,theta)`

Function to generate sinusoidal grating signal moving in temporal domain. The output of the function is a cell array of Nv×No where Nv and No are respectively of the number of different velocities and orientation of the stimulus. The inputs of the function have the following meaning:

Argument	Description
`[sx, sy]`	x and y dimension of frames of the stimulus.
`v`	motion velocity of the stimulus.
`N`	number of frames.
`K`	multiple for which multiply period-value to define total duration of the stimulus
`k_0`	spatial frequency of the stimulus (default value k0=0.0625 cycle/pixels).
`theta`	orientation of the stimulus (respect to image frame of reference).

- **@plaid**

Plaid stimuli are defined as a class. An object `plaid` is characterized by several attributes and methods listed in the following table:

Attribute	Description
`apert_rad`	aperture radius of stimulus in pixel.
`dur`	period of stimulus in frame.
`truetheta`	true orientation of plaid stimulus, in radiant.
`vpld`	plaid velocity, in pixel/frame.
`k`	spatial frequency of two gratings.
`vgrat`	velocity of two gratings in pixel/frame.
`theta_g`	orientation of two gratings in radiant.
Method	**Description**
`get`	get specific value of one attribute of the object.
`set`	set specific value of one attribute of the object.
`plaid`	constructor of `plaid` object.
`generate_plaid`	made 3D matrix from plaid object (see below).
`simulate`	orientation of the stimulus (respect to image frame of reference).

`II = generate_plaid(pld,varargin)`

Made 3D matrix (`II`) from plaid object (`pld`). Two different algorithms are implemented. First one uses the analytic expression of plaid object and rotational matrix to rotate two gratings (`analytic_expression`). The second one, instead, rotate two gratings applying `imrotate()` function. Optional input parameters accepted (varargin) are two: disp ('0' or '1') and algorithm ('1' or '2'). By default disp is set to 0 and second algorithm is used.

2.2 Plot functions

- `plot_disp_tun(varargin)`

Horizontal disparity tuning curves plot of a sub-population of complex- and MT-like neurons varying the phase-shift channel. The orientation channel is fixed by the method ($\theta = 0°$) while the velocity channel is fixed by the user (only one value!). The function plot only the data stored in SIMULATIONS folder. By default the `'disp_tuning_curve_norm.mat'` file is plotted but it is possible to specify another file name(`varargin{1}`). This file must be contain a multidimensional matrix of name "e" and of size `ncell`×`No`×`Nph`×`Nd`×`Nseed` where `ncell` is the number of cells, `No` is the number of the orientation channel (equal to 1), `Nph` is the number of the phase shift channels, `Nd` is the number of samples of the disparity values vector and `Nseed` is the number of seeds of the pseudo-random number generator.

- `surf_disp_tun(varargin)`

Surface plot of disparity tuning of a sub-population of disparity detectors (C21, C22). In the tuning analysis provided the number of orientation channels are eight (span from 0 to π) and the number of phase shift values are seven. It can use another population activity file in .mat extension identified by its file name. In this file must be provided several information:

Argument	Description
e	multidimensional matrix that contains the population activity of size: ncell×No×Nph×d_samples×d_samples×Nseed where ncell is the number of cells, No is the number of the orientation channels, Nph is the number of the phase shift channels, d_samples is the number of samples of the disparity vector and Nseed is the number of seeds of the pseudo-random generator.
d	disparity values vector
c_or	string that contains the labels of each orientation channel
c_ph	string that contains the labels of each phase-shift channel

- surf_MID_tun(varargin)

Surface plot of motion-in-depth of a sub-population of C21,C22 and MT cells. In the tuning analysis provided the number of orientation channel, phase shift channel and v_pref is one (see parameters). It can use another population activity file in .mat extension identified by its file name. In this file must be provided several information:

Argument	Description
e	multidimensional matrix that contains the population activity of size: ncell×No×Nph×v_pref×vL_samples×vR_samples where ncell is the number of cells, No is the number of the orientation channels, Nph is the number of the phase shift channels, v_pref is the number of the velocity channels, and vL_samples and vR_samples are the number of samples of the velocity values of the moving sinusoidal gratings.
v	velocity values vector
parameters	see Tab.1.

- surf_motion_pop(varargin)

- polar_motion_tun(varargin)

Contents