**Cone activation**

The main code in this folder is “mainAuto.m”. At the beginning you will be asked to enter some videos and data matrixes, but then the program will continue working automatically. For each 1 second video it takes between 20 to 30 seconds to complete analysis. So, if you have a large number of videos to analyze, maybe a good option is to leave the program to run all of them and do sth else meanwhile.

Remember that this program uses cone locations and CDCs of the master sumNorm (obtained from PRL videos). However, we need them for individual sumNorms (obtained from each video). So, we have to register individual sumNorms to master sumNorms. To obtain a good result, it is better to use Niklas’s code for this, because his code considers both shift and rotation and is more accurate than other candidates that I tested. I put these codes in folder entitled “Niklas-Registration”. It is a very user-friendly code. Once you run it you will see all the required instructions in the Matlab workspace.

After registration, you can start with “mainAuto”. This code calls three other codes, that I will explain here:

1. QualityCheck\_July14
2. FineStimMaker\_July16
3. SVA\_oneStrip\_traceAD

# QualityCheck\_July14

This program was first made to check if the registration of the individual sumNorm to Master sumNorm works well or not, and if the cross is successfully found in all the frames containing cross. Because previously we used to find cross with “template-matching” (see the previous versions of “QualityCheck”). In the last version, as I explained, we use Niklas’s code for registration and we can check the quality of registration there. Also, we don’t use template matching for finding cross anymore; we now use the shifts obtained from offline stabilization for finding cross (FindPRL\_July14).

Warning! “FindPRL\_July14” only works for gain=0 videos.

# FineStimMaker\_July16

We need to rebuild the stimuli that we used in the experiments (in case you haven’t saved them). For this, before starting “mainAuto” we need to run “MyStim” available in “E\_Maker” folder. Remember that it only works for tumbling E, for other stimuli we need to change the pattern of stimulus. In line 4, we need to copy rawData or if we want to do the analysis for synthetic data, we need to make up some synthetic raw data and put them here. The output of “E\_Maker” is a mat file called “Es.mat”. Copy it in the folder containing all the videos from one session of experiment.

Warning! Since rawData is different for each session of experiment, “Es.mat” is also different in each session.

Now, what does “ FineStimMaker\_July16” do? It simply pads the stimulus image to a fixed size (maximum size in the current session), and makes an array so that the pipeline can work properly.

# SVA\_oneStrip\_traceAD

This code is the heart of cone activation pipeline. Having the stimulus for the current video (from “FineStimMaker\_July16”) and the location of the cones, cross (or the stimulus), this code can find the cone activations and record them in a structure called “ConeLoc\_actLogs.mat”. If you open this structure you will find the cone activation of all videos of one session here.

Warning! This structure only contains the sum of the activation in each cone, in other words the sum of Gaussian function representing the cone probability distribution on each cone (Figure 1). If you are interested in the cone probability distribution map, uncomment line 250 of the code.

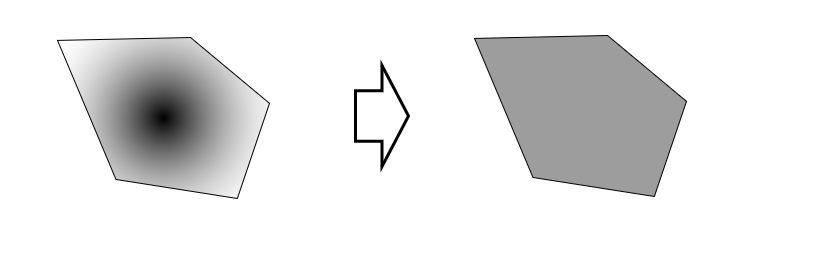


Figure - left: a schematic of the cone probability distribution map on a Voronoi cell. Right: the summation of the cone probability distribution through the corresponding Voronoi cell.

Now, we have the cone activation in all the frames. We can consider the time-course of the cone activations. For this, run “timeCourse.m”.

Warning! This code is not finalized, because I couldn’t find enough examples to test if it works correct or not. Please see the reference papers that I put in this folder. These are all the related papers that I could find.