# Introduction to pRF data-analysis using MATLAB and VISTASOFT

Version 1.1

Spinoza Centre for Neuroimaging

# Table of Contents

# Before you begin

This assignment is designed as a practical introduction to functional magnetic resonance imaging (fMRI) analysis of retinotopy and pRF model-based data using MatLab (version 7.x or later) and Vistasoft (mrVista). More information can be found on their respective websites www.mathworks.com and white.stanford.edu/software. Parts of this assignment are based on the Vistasoft website (wiki) instructions. The assignments will likely work for MacIntosh, Windows and Linux operating systems, but have only been tested extensively on Linux.

The aim of this assignment is to provide a general introduction to the practical data-analysis steps. It deals with various idiosyncrasies of the software implementation and assumes a general knowledge of the underlying data-analyses principles.

For this particular assignment no special toolboxes are necessary, but for full functionality additional tools need to be installed: Vistasoft occasionally uses SPM functions (SPM5 or later), and the full pRF model-based analysis requires *Signal Processing* and *Optimization* Toolboxes.

To add the necessary toolbox on the server type:

```
run /packages/matlab/toolbox/startup.m
```

and

```
addpath(genpath('/packages/matlab/toolbox/vistasoft'))
```

When working on a different machine, you can download the vistasoft code here:
https://github.com/vistalab/vistasoft
Then you need to add the software folder *vistasoft* to your MatLab path (addpath(genpath())), or simply go to that directory using MatLab and add it to your path:

```
addpath(genpath(pwd));
```

Because of the many functions, this may take a while. Throughout this course indented `courier` font indicates commands to be executed in MATLAB (as shown above).

Enjoy!

# 1. pRF model-based analysis

## 1.1. Background

Visual field mapping techniques summarize the most effective visual location to drive neuronal responses at a particular cortical location as a point in visual space. Yet every neuron does not process a single location but a region of visual space known as its receptive field. Moreover, given estimates of neuronal packing density (Rockel et al., 1980; Leuba and Garey, 1989) and typical fMRI resolutions (~2.5mm isotropic), each recording location contains about a million neurons.

Using fMRI we can measure the aggregate receptive fields of these large neuronal populations. To acknowledge that these fMRI receptive fields are different from single neuron receptive fields, the region of visual space that stimulates the recording site is typically referred to as the population receptive field (pRF) (Victor et al., 1994; Jancke et al., 2004; Dumoulin and Wandell, 2008).

PRF sizes can be modeled by fitting two-dimensional models to the fMRI signals. These pRF models were either Gaussians (Dumoulin and Wandell, 2008) or Gabor wavelet pyramids (Kay et al., 2008). This type of analysis is independent of the exact stimulus layout, though the insertion of proper baseline is crucial to estimate the exact pRF sizes (Dumoulin and Wandell, 2008). Without a baseline, the analysis cannot distinguish a small pRF responding only to certain visual field locations from a large pRF responding to all visual field locations but with a preference to certain visual field locations.

The neural model-based method is more than just a technique to estimate neuronal receptive field sizes. Compared to older approaches such as the traveling wave paradigm they have several advantages. First, these approaches do not depend on a particular stimulus paradigm. Second – and most important – these approaches are poised to model many other properties of the underlying neuronal population, such as quantitative estimates of surround suppression and the relative amount to which neuronal populations process the contra or ipsi-lateral visual field (Dumoulin and Wandell, 2008, Zuiderbaan et al.,2012).

## 1.2 Data-analysis

After organizing your acquired neuroimaging data in a BIDS-format, we can start with the data analysis.
The first step is to initialize the session.

### 1.2.1 Initalizing the session

The fMRI data for a session has to be initialized into the directory and file format used by mrVista. The procedure for this requires:

1) A series of functional image files

2) An anatomy image

3) An Inplane anatomical file. If this is not acquired during scanning, it can be created by saving the first slice from one of the functional data scans. To split the functional scan per slice you can use the function fslsplit (*fslsplit name_of_the_first_functional_run.nii.gz*). The first file can then be used as your inplane anatomical file.

They all should be stored in a NIfTI format. As a convention you can make a main folder depending on the subject name (sub_001) from where the session will be initialized. In this folder make a subfolder called 'Raw', 'Anatomy' and 'Stimulus'. In 'Raw' you store the functional scans. In Anatomy you store the Anatomy image and the inplane anatomical file.

Move to the main folder from where you initialize the session (sub_001). In MATLAB type the following command to perform the initialization of the session:

```
mrInit
```

This opens a GUI dialog box that has the options to upload:

1) Inplane anatomy

2) Functional data

3) Anatomy image

In case there are prescans that should be skipped/clipped, the dialog box ("Clip frames from time series) should be selected.

Upon clicking OK, a new dialog box opens wherein the session descriptions can be provided. This includes session name, name of the subject, and any other comments that are relevant.

If the clipping prescan box was checked it will then be opened as another dialog box where you can add the number of frames to skip.

Click 'OK' in the Assign Analysis Parameters window.

This results in a mrSESSION.mat file and inplane subdirectory.

To open the mrVista interface go to the subject folder. You should stay in this directory when using mrVista.

The "inplane" view can be opened by typing:

```
mrVista
```

The data associated with this view is found in the Inplane subdirectory. It represents data in the format acquired during scanning.

## 1.2.2 From "inplane" to "gray" view

To analyse the data on a cortical surface, we need to transform our data from the "inplane" view to the "gray" view (also called the "Volume" view). For this we have to align the inplane

data with a high quality anatomical MRI and segment the gray and white matter.

#### 1.2.2.1 Alignment

The goal of the alignment is to make the interpolated anatomy slices as much as possible like the reference inplane slices. rxAlign allows to alter the translation and rotation settings (and, when necessary, flipping the volume along one axis or another).

Type

```
rxAlign
```

Four figures will open on the screen:

1. CONTROL figure (at top)- contains controls for setting a 4x4 affine transformation which can be applied to the volume anatomy (vAnatomy) to produce the inplane prescription.
2. PRESCRIPTION (Rx) figure (to the middle-left of the screen) - The Rx figure shows where on the vAnatomy the inplane slices lies.
3. PRESCRIBED SLICE figure (in the center) - The Prescribed Slice figure shows an interpolated slice from the vAnatomy, after the transformation has been applied.
4. REFERENCE SLICE figure (to the middle-right) - The Reference Slice figure shows the equivalent slice from the inplane anatomies.

In the GUI navigate to Window | Open Rx/Ref comparison window to create an overlay image of your alignment. Use the three 'Translate' sliders and 'Rotate axials CW' slider to get your images aligned.
When you're satisfied with the alignment save these settings: from the drop down menu File | Save | Xform Settings and File | Save | mrVista Alignment and close the mrRx window.

#### 1.2.2.2 Segmentation

The BIDS data structure provides files for the segmentation of the white and gray matter. This file can be found in the mri-folder of freesurfer. To be able to use them in mrVista, we need to do some further processing, both in freesurfer and in Matlab.

**FreeSurfer part:**

The first step is to process the anatomical image through freesurfer pipeline. You can use the freesurfer commands in the terminal, or in the matlab command line. If you use it in the matlab command line, put an ! in front of the command.

1. Type the following command (Note that if you use the BIDS-format, which for this course you do, you don't have to do this step, but step 2 and 3 you do):

```
recon-all -s NameOfSubj -i NameOfSubj-T1image.nii -all
```

2. Freesurfer creates segmented maps of the cortex, but we need to align them back to the native space of the subject in order to use them in the mrVista:

   Make sure you are in the folder of the subject (-s from the previous step) and type:

```
mri_label2vol --seg mri/aseg.mgz --temp mri/rawavg.mgz --o aseg-in-
raw.nii --regheader mri/aseg.mgz
```

aseg-in-raw.nii is the output-file of this command.

3. Now we created a file named aseg-in-raw.nii which has many different labels marking different cortical regions. From this map we will create 4 different nifti images of the left and right white and gray matter:

type in the terminal:

```
mri_binarize --i aseg-in-raw.nii --o LeftWhiteMatter.nii --match 2
mri_binarize --i aseg-in-raw.nii --o LeftGrayMatter.nii --match 3
mri_binarize --i aseg-in-raw.nii --o RightWhiteMatter.nii --match 41
mri_binarize --i aseg-in-raw.nii --o RightGrayMatter.nii --match 42
```

These binary images now need to be combined in order to create the matched mrVista segmented image:

**MATLAB part :**

In the folder of the subject where the previous nifti images were created (LeftWhiteMatter.nii etc') type the following commands:

```
LG=niftiRead('LeftGrayMatter.nii');
RG=niftiRead('RightGrayMatter.nii');
LW=niftiRead('LeftWhiteMatter.nii');
RW=niftiRead('RightWhiteMatter.nii');

New=LG;
New.data(LG.data==1)=0;
New.data(LG.data==0)=1;
New.data(RG.data==1)=0;
New.data(LW.data==1)=3;
New.data(RW.data==1)=4;

New.fname= ['Segmentation_subj001.nii'];
writeFileNifti(New);
```

### 1.2.3 Transformation of the functional data to the "gray" view

The further analysis is performed in the "gray" view (also called Volume view). To initialize the "gray" view, type:

```
mrVista 3
```

For number of gray layers choose 4 or 5. These are the number of voxels that will be grown on top of the white matter, and define the gray matter.

A gray 3-view window will be opened.

In the window 'Select class file', you select the segmentation-file you made in the previous step.

To transform the time-series from the Inplane view to the Volume ("gray") view, select from the menu

"Xform | Inplane -> Volume | tSeries (all scans) | Trilinear Interpolation".

In this dataset the original data contains several scans with the same stimulus protocol. The stimulus consisted of bar-shaped apertures sweeping across eight different directions through the visual field. Four mean-luminance (baseline) blocks were interspersed. The entire stimulation protocol sequence was repeated during the different scans. To obtain a high signal-to-noise ratio, we average the collected scans of the bars.

To average all 'retinotopic bar'-scans, click "Analysis | Time Series | Average tSeries". The "Average Time Series" window pops up. Select all retinotopic bar runs. By default these averaged data will be stored in a new data type called "Averages". You can give a different name, such as "Averages_Bars". Last, give an annotation, a name you prefer to describe your averaged data.

Select the data type as "Averages_Bars" (previously it was "original") on the upper-right corner of mrVista main interface.

Next, we can proceed to perform the analysis on the averaged scans.

### 1.2.4 Define the stimulus

Now we need to define the stimulus parameters.

For this dataset we need to make two stimulus files, the *image-file* and the *params-file*. The information you need to save in these files can be found in the stimulus mat-files that were output during the scanning.
Load this mat-file.
In the *image-file* you need to save the presented stimuli. These can be found in the parameter stimulus.images. Rename stimulus.images to images and save this as your image-file. Save both the image-file and the params-file in the subfolder called *Stimuli*.

```
images = stimulus.images;
save images_bars images;
```

In the *params-file* you again save the presented stimuli, rename stimulus.images to original_stimulus.images{1}. You also need to save the params and stimulus parameters from the stimulus matfile.

```
original_stimulus.images{1} = images;
save params_bars original_stimulus params stimulus
```

In the window, select the menu "Analysis | Retinotopic Model | Set Parameters". You will get a dialog allowing you to define what stimulus was used for each scan in this data type.
(*In case you get an error: Make sure you always return to the main subject folder when you use the menu of the gray-view*)

Note: The pRF analysis always fits all scans in a data type! So it is important to make sure

you set every scan in the data type. If there are several scans there will be a slider at the top of the GUI, which takes you across the scans. In this case, we run the model on only one (1) average scan. Select the datatype you made when averaging the scans.

The parameters for the sample data set should already be set appropriately. But it is useful to examine them:
Spatial parameters:

- **Stimulus Class**: Defines the type of stimulus. Nowadays we mainly use "StimFromScan". This allows you to load the stimulus files that were output by the stimulus software when the scan was run.
- **Radius (deg)**: Stimulus size in radius and degrees of visual angle.
- **Detrending fMax (#)**: The DCT Frequency Max may be set to zero; this is fine, and turns off all detrending except for the constant-offset DC component. We normally usually use detrending up to 3 cycles/scan. This removes low-frequency noise from the time-series.
- When using "StimFromScan" the other parameters are set from the stimulus files – these parameters are now grayed-out.

Image parameters (these are only used if the stimulus class is "StimFromScan":

- **Image file**: Stimulus files. Select the images-file you made previously.
- **Params file**: Stimulus parameter files. Select the params-file you made previously.
- **Jitter**: "None" (option to jitter the stimulus to simulate the effects of eyemovements.
- **Image filter**: "Binary", convert the stimulus files to binary, i.e. "1" for stimulus present, "0" for no stimulus or mean-luminance.

Timing parameters:

- **HRF model**: Defines HRF model, we normally use "two gammas (SPM style)".
- Other parameters are read from the predefined session parameters and are shown as a sanity check.

Save:

- **Save to dataTYPES**: Select to remember these parameters (even when you close MatLab and/or log out of your machine). The parameters are already saved so there is no need to resave them. But should you want to change them, it is a good idea to save them so the software will remember the settings.

Select "Done" to confirm these parameters. The stimuli are now recreated. You can view the stimulus aperture using "Plots | Retinotopic Model | View stimulus aperture". It should show the stimulus energy – in this case eight bars traversing across the visual field interspersed with mean luminance blocks (no stimulus).

**Important**: If your stimulus definition is incorrect, your pRF analysis will be wrong as well!

We are now ready to estimate the pRF parameters for the data. Solving the pRF model is generally the longest step in the analysis. On our modern process servers, solving the model can take around 1-5 hours. It can take longer on slower machines.

You have many choices to solve the model. We'll use the default choice, which is to do a full coarse-to-fine fitting for each voxel, and run it within our current instance of MATLAB. The default model we will solve is a circular 2D Gaussian for each voxel. To do this, select from the menu "Analysis | Retinotopy Model | Run".

Important: if there is an ROI loaded the model will only run on the data of the ROI. This saves time and is a good idea for testing purposes!

*Due to different versions of vistasoft, you might get an error that matlab is unable to find the tSeries1.mat file. If you get this error, rename the tSeries.mat in the folder 'Gray/Averages_bars/TSeries to tSeries1.mat*

When the model finishes, it will produce several **.mat** files containing three iterations of the model solution. These files will all be located in the data directory ('Gray/Averages_bars/'), will start with the text 'retModel-' followed by a time stamp indicating when they were solved, and will have the following suffixes at the end:

- **retModel-*-gFit.mat** represents the grid-fit stage. This stage take a discrete "grid" of pre-set pRF parameters (x0, y0, sigma) and fits them to each voxel. There are approximately 100,000 different sets of parameters it generates. It selects the best-fitting set of parameters for each voxel. Because we're using a coarse-to-fine fit (the default mode in the Gray/Volume views), this fitting is actually on a version of the time series that is spatially-smoothed along the cortical surface. The motivation is to have voxels which are nearby in cortical space to have similar initial pRF estimates. In the refinement stage, we will not use this spatial smoothing, and will fit the original data.
- **retModel-*-sFit.mat** represents a search fit. This is a constrained optimization that is run for each voxel, where the set of pRF parameters (x0, y0, sigma) for each voxel can assume any value to best fit the data (within certain constraints of the search space--it avoids solutions which are wildly different from the coarse fit).
- **retModel-*-fFit.mat** represents the final fit of the model. It contains the pRF estimaes from the search-fit, but recomputes the measurement of proportion variance explained for each voxel. Whereas the search fit file estimated this based on a refinement of the grid fit, the **-fFit** file uses the final pRF estimate for each voxel to explain the original data.

It is the **-fFit** file which we usually analyze, and which we will look at in subsequent sections.

## 1.3 Visualization

Now we can load a solved model. Select the menu item "File | Retinotopy Model | Select and Load Model". You will get a file dialog.

Select the file that ends with "-fFit.mat**"**. This is the final-fit retinotopy model for a coarse-to-fine pRF fit for this data set.

As the code loads the retinotopy model, it loads four data fields from the model into the four data slots used by mrVista. The naming convention is inherited from conventional retinotopy data slot and unfortunately not as applicable to the retinotopy model. Anyway, by default, the following data are loaded into the following fields:

- The **co** slot has the *variance explained* ($r^2$) for each voxel. This is the proportion of that voxel's variance in the time series explained by the best-fit pRF model.
- The **ph** slot has the *polar-angle* for each voxel. Because the expected phases are well defined for the pRF analysis, there are predefined polar-angle maps that can be loaded using left and right hemisphere using "Color Map | Phase Mode | Wedge map for pRF (left)" or "Color Map | Phase Mode | Wedge map for pRF (right)", respectively.
- The **amp** slot has the *pRF size* for each voxel. This is expressed as the standard deviation of the 2D circular Gaussian function used for this pRF model.
- The **map** slot has the *eccentricity* of each voxel. This is the distance, in visual degrees, of the pRF center from the fovea.

The pRF has many data fields, and you can load these in whichever of the **co**, **ph**, **amp** or **map** slots you like. To do this, select the menu **File** | **Retinotopy Model** | **Load Model Parameter**. You will get a dialog allowing you to choose a data field, and select which data field to load it in.

You can also inspect the time-series of small ROIs. Inspecting the time series is important to check the visual quality of your data. You can create ROIs in several ways, but the most easy one is “ROI | Create | Create Rectangle ROI”. Click two points. You can also restrict the ROI to current view thresholds by: “ROI | restrict | restrict selected ROI”.

In order to visualize the ROI time series click “Plots | Current scan | mean time series”. In the same manner you can view the average single cycle and the powerspectra.

Once you have defined an ROI you view the model fits using “Plots | Retinotopic Model | Receptive field and model fit (all time points). A separate window will open up. There are several parts to this interface:

- **Voxel Slider** (top right): selects the current voxel whose data you want to view (not in any specific order – but you can order them under “Sort / Select voxels”).
- **Overlay Stimulus** (under voxel slider): This option toggles the stimulus overlay on and off. The stimulus overlay will also show the stimulus window in the pRF profile, and the corresponding time point a slider underneat and in the Time Series.
- **pRF profile** (top left): Shows the best-fit pRF for this voxel, as saved in the currently-loaded retinotopy model file.
- **Time Series** (bottom left): Shows three time series traces: (1) The observed time series for the voxel (black), after trend removal and conversion to percent signal change; (2) the predicted response produced by the estimated pRF (blue); (3) the residuals (observed - prediction). Each of these traces can be toggled using the checkboxes to the right of the plot.
- **pRF stats** (middle top): Shows the parameters for the selected voxel's pRF, and the coordinates of the selected voxel. The coordinates are specified (for this Gray/Volume ROI) as [axi cor sag], specifying the axial (S -> I), coronal (A -> P), and sagittal (L -> R) slice of the selected voxel.

You can also create a graph plotting pRF size versus eccentricity by selecting “Plots | Retinotopic Model | Plot pRF Size vs. Eccentricity (current ROI)”. The first figure shows all individual voxels. The second figure shows the same data above the thresholds defined in the view and across predefined eccentricity bins.

## 1.4 Surfaces

The surfaces are displayed using a stand-alone program called mrMeshServer (mrMeshSrv). This may take a little while. **On the server, it should work without starting it manually**, but if you get several errors “Can’t establish connection to mrMeshSrv” and no windows pop-up, you need to start up the server manually. You can do that using the command:

```
mrmStart
```

Or launch the cortical surface rendering software mrMeshSrv, from vistasoft/mrMesh/meshserver/mrMeshSrv. To learn a bit about the mrMesh and common

known issues check vista lab wiki (https://web.stanford.edu/group/vista/cgi-bin/wiki/index.php/Visualization).

With mrMeshSrv running, navigate to Gray | Surface Mesh | Build New Mesh | Left Hemisphere (or Right Hemisphere) in the VOLUME view.
In the next window, change the number of Gray Layers to 4 or 5 and select the corresponding hemisphere. Click 'Yes' in the next window. Saving the mesh is of little use. So, click cancel when asked.
To inflate the mesh, navigate to Gray | Inflate | Inflate (Set Params). This opens the Set Mesh Build Parameters. Set the Smooth Iterations to 128 and Smooth Relaxation to 1.

With the model loaded, navigate to Gray | Update Mesh to display the model on the surface. By default, it loads the eccentricity map.
To load the polar angle maps, select Color Map | Phase Mode | Wedge map for pRF (left - for left Hemisphere). Update the mesh always when you make any changes in the VOLUME view.
Different other parameters can also be visualized on the surface from the drop-down menu 'View'. From top to bottom, it has 'Anatomy and ROIs (no overlay)' which displays just the anatomy. 'Coherence Map' view contains goodness of fit of the estimate. 'Amplitude Map' view contains the pRF size estimates. 'Phase Map' contains phase information which is similar to the step above. 'Parameter Map' view contains the eccentricity estimate. Set 'mapWinMax' on the right side of the “gray” view window to your stimulus radius to exclude voxels where the preferred pRF location is outside the stimulus area.

To map the data that is shown on the “gray” view onto the surface click “Gray | Update Mesh” or if you have more than one surface mesh open “Gray | Update All Meshes”. The current surface mesh is identified under “Gray | Selected Mesh”. You need to redo do this after changing different views, colormaps etc, to make sure you are viewing the current data.

To unfold the meshes select “Gray | Inflate | Inflate (Set Params)”. The amount of unfolding can be determined by the “Smooth Iterations” and “Smooth Relaxation”: 128 and 1, respectively are good choices.

Click left mouse button (and hold) on the mesh to rotate. Click middle button to translate. Middle mouse wheel will zoom.

### 1.4.1 Define Region of Interest (ROI)

You can also draw on the surface meshes.

- “**d**” to toggle drawing mode on or off.
- **[del]** to delete everything
- Mouse click to draw lines connecting each mouse click
- “**u**" undo last action in drawing mode
- “**c**” for close your points
- “**f**” + click to fill in the ROI.

After drawing the ROI on the surface, navigate to VOLUME view and select Gray | Mesh ROIs | Get ROI From Mesh (drawn with "d" key, All Layers)
Your ROI will now appear in the ROI drop down menu in the VOLUME view. Update the mesh to see it on the surface.

A Region of Interest can be defined based on the visual map displayed on the cortical surface rendering. In early visual areas, visual map borders are defined by a reversal in polar angle representations at the horizontal and vertical meridians.

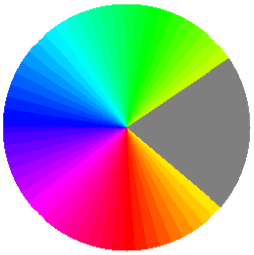

**Figure 1** Colorbar of the left polar angle map (right hemisphere). Based upon the reversals of polar angle, visual field maps can be defined.

V1 is defined as a full hemifield representation in and around the Calcarine sulcus.
Go ahead and define V1. See the paper of Wandell et al., 2007 for a description and location of visual field map borders.

It is best to define the ROI until the stimulus extend. Although the model pRF fits are not limited to this eccentricity range, parameters outside this range cannot reliably be estimated. For this you need to restrict the eccentricity range. To define V1, you will need to define both left and right V1 separately and combine them using the ROI submenus:
ROI | Select/Edit/Combine | Combine Multiple ROIs Into One ROI, or
ROI | Select/Edit/Combine | Combine ROIs

# 2 Programming and scripting notes

All information in the different views is also available within the MatLab workspace. The inplane and gray view is stored in the view-structures INPLANE{} and VOLUME{}, respectively.

Each menu item calls a .m-function that modifies the view-structure. You can visualize these .m-function calls by selecting “Help | Identify Callback for a Menu Item”. After selecting this option, the view-window will turn yellow, and a message appears asking you to select the callback item.

For example, selecting “Analysis | Retinotopic Model | Run (pRF)” will output in the matlab command window:

```
Selected Menu Item:
 Label: Run (pRF)
 Handle: 107.001099
 Callback: VOLUME{1}=rmMain(VOLUME{1},[],3);VOLUME{1}=refreshScreen(VOLUME{1});
```

It reveals rmMain.m as the main function to run the retinotopic model.

The Difference-of-Gaussian (DoG) model (Zuiderbaan et al., 2012) can be called using:

```
VOLUME{1}=rmMain(VOLUME{1},[],3,'model',{'dog'},'dc',1);
```

Note here that in the modeling there is a dc-component added to the fitting, this is an estimation of the baseline (0% contrast). This is necessary to accurately interpret the parameters of the model. However, it does require that you run the pRF-mapping stimulus using long blank periods (mean-luminance).

The view-structures can also be opened without the GUI interface by using the functions initHiddenInplane() and initHiddenGray(). In this fashion, the entire processing steps can be easily put into a matlab script.

In addition of documenting your processing steps in an .m-script, a script also allows you to call the functions using non-standard options, and to call functions that are not part of the GUI.

# References

DeYoe EA, Carman GJ, Bandettini P, Glickman S, Wieser J, Cox R, Miller D, Neitz J (1996) Mapping striate and extrastriate visual areas in human cerebral cortex. Proceedings of the National Academy of Sciences of the United States of America 93:2382-2386.

Dumoulin SO, Wandell BA (2008) Population receptive field estimates in human visual cortex. NeuroImage 39:647–660.

Dumoulin SO, Hoge RD, Baker CL, Jr., Hess RF, Achtman RL, Evans AC (2003) Automatic volumetric segmentation of human visual retinotopic cortex. NeuroImage 18:576-587.

Engel SA, Glover GH, Wandell BA (1997) Retinotopic organization in human visual cortex and the spatial precision of functional MRI. Cereb Cortex 7:181-192.

Engel SA, Rumelhart DE, Wandell BA, Lee AT, Glover GH, Chichilnisky EJ, Shadlen MN (1994) fMRI of human visual cortex. Nature 369:525.

Harvey BM, Dumoulin SO (2011) The Relationship between Cortical Magnification Factor and Population Receptive Field Size in Human Visual Cortex: Constancies in Cortical Architecture. J Neurosci 31:13604-13612.

Jancke D, Erlhagen W, Schoner G, Dinse HR (2004) Shorter latencies for motion trajectories than for flashes in population responses of cat primary visual cortex. The Journal of physiology 556:971-982.

Kay KN, Naselaris T, Prenger RJ, Gallant JL (2008) Identifying natural images from human brain activity. Nature 452:352-355.

Leuba G, Garey LJ (1989) Comparison of neuronal and glial numerical density in primary and secondary visual cortex of man. Experimental brain research Experimentelle Hirnforschung 77:31-38.

Rockel AJ, Hiorns RW, Powell TP (1980) The basic uniformity in structure of the neocortex. Brain 103:221-244.

Sereno MI, Dale AM, Reppas JB, Kwong KK, Belliveau JW, Brady TJ, Rosen BR, Tootell RB (1995) Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. Science 268:889-893.

Victor JD, Purpura K, Katz E, Mao B (1994) Population encoding of spatial frequency, orientation, and color in macaque V1. Journal of neurophysiology 72:2151-2166.

Wandell, B. A., Dumoulin, S. O., & Brewer, A. A. (2007). Visual field maps in human cortex. *Neuron*, *56*(2), 366–383. https://doi.org/10.1016/j.neuron.2007.10.012

Warnking J, Dojat M, Guerin-Dugue A, Delon-Martin C, Olympieff S, Richard N, Chehikian A, Segebarth C (2002) fMRI retinotopic mapping--step by step. NeuroImage 17:1665-1683.

Zuiderbaan, W., Harvey, B. M., & Dumoulin, S. O. (2012). Modeling center-surround configurations in population receptive fields using fMRI. *Journal of Vision*, *12*(3), 10.