Simultaneous Estimation of Population Receptive Field and Hemodynamic Parameters from Single Point BOLD Responses using Metropolis-Hastings Sampling

Stanisław Adaszewski[1,2,*], David Slater[1], Lester Melie-Garcia[1], Bogdan Draganski[1], Piotr Bogorodzki[2]

[1] Laboratoire de Recherche en Neuroimagerie, DNC, CHUV, Lausanne, Switzerland

[2] Instytut Radioelektroniki i Technik Multimedialnych, WEITI, PW, Warsaw, Poland

[*] Corresponding author

Corresponding Author's Address:

Stanisław Adaszewski

Route de Carroz-Derriere 9

1844 Villeneuve

s.adaszewski@gmail.com

Abstract

We introduce a new approach to Bayesian pRF model estimation using Markov Chain Monte Carlo (MCMC) sampling for simultaneous estimation of pRF and hemodynamic parameters. To obtain high performance on commonly accessible hardware we present a novel heuristic consisting of interpolation between precomputed responses for predetermined stimuli and a large cross-section of receptive field parameters. We investigate the validity of the proposed approach with respect to MCMC

convergence, tuning and biases. We compare different combinations of pRF - Compressive Spatial Summation (CSS), Dumoulin-Wandell (DW) and hemodynamic (5-parameter and 3-parameter Balloon-Windkessel) models within our framework with and without the usage of the new heuristic. We evaluate estimation consistency and log probability across models. We perform as well a comparison of one model with and without lookup table within the RStan framework using its No-U-Turn Sampler. We present accelerated computation of whole-ROI parameters for one subject. Finally, we discuss risks and limitations associated with the usage of the new heuristic as well as the means of resolving them. We found that the new algorithm is a valid sampling approach to joint pRF/hemodynamic parameter estimation and that it exhibits very high performance.

Introduction

Modelling is an important domain of science in general and a recurring topic in population receptive field (pRF) research in particular, where functional magnetic resonance imaging (fMRI) serves as evidence acquisition method. Classical approaches such as those by (Dumoulin and Wandell, 2008) and (Kay et al., 2013) focus on point estimates of parameters in predefined models motivated by physiology and empirical evidence. In the recent work of (Zeidman et al., 2016) authors introduce the formalism of Bayesian Model Selection in order to root pRF model choices in an objective quantitative measure such as Variational Free Energy. Furthermore, the proposed formulation employs a Balloon-Windkessel model for joint estimation of pRF and hemodynamic parameters. The main limitation of this approach lies in the assumption about the form of the posterior distribution of parameters, in this case Gaussian. The method is characterized as well by high computational requirements – the reference implementation of the algorithm is reported to require about 100 seconds per voxel to converge which renders its use problematic on modern PCs with exception of high-end multi-core cluster setups (the authors give an example of 192-core cluster used to estimate 14395 voxels) or small regions of interest. A Bayesian approach using slice-sampling Monte Carlo method with fixed Hemodynamic Response

Function (HRF) was recently described in (Quax et al., 2016). Similarly to the variational method the sampling approach quantifies how variable the underlying receptive field is by using the uncertainty of the posterior estimate except with the added advantage of not imposing any particular form on the posterior probability distribution. The authors underline the importance of their method's capability to estimate variability – rendered particularly relevant by the fact that receptive fields are not rigid over time, but can change due to attention effects or task demands (Klein et al., 2014). The main contribution of our work is a new approach to Bayesian pRF model estimation combining the best characteristics of the above methods – inclusion of the Balloon-Windkessel hemodynamic model, Dumoulin-Wandell and compressive spatial summation (CSS) pRF models and using sampling for model inversion therefore not imposing any form on the posterior. Furthermore in order to address potential obstacles due to high performance requirements we introduce a novel heuristic for solving the Dumoulin-Wandell pRF model by using interpolation across a lookup table containing precomputed responses for given stimuli and a large number of predefined receptive field parameters. This enables us to massively parallelize the algorithm using a graphics processing unit (GPU) implementation of the Markov Chain Monte Carlo (MCMC) scheme. Our algorithm offers choice between existing pRF models – Dumoulin-Wandell model (Dumoulin and Wandell, 2008) and compressive spatial summation (CSS) model of pRF introduced in (Kay et al., 2013) and for BOLD generation between well-established Balloon-Windkessel model (Buxton et al., 1998; Friston et al., 2000; Irikura et al., 1994; Mayhew et al., 1998), its 3-parameter version (Stephan et al., 2007) used in the recent implementation of Dynamic Causal Modelling (DCM) in Statistical Parametric Mapping (SPM) toolbox as well as a fixed user-provided HRF. Our algorithm is presented and discussed along with introduction of QPrf – its freely available implementation in the form of a standalone toolbox (https://github.com/sadaszewski/qprf) available with source code under the terms of GNU GPLv3 license. We demonstrate CSS-pRF/Balloon-Windkessel model inversion using the new heuristic and compare it to a classical two-stage method. Furthermore, we compare different combinations of pRF (CSS, classical Dumoulin-Wandell) and hemodynamic (5-parameter and 3-parameter Balloon) models within QPrf and against existing Bayesian inversion software (BayespRF).

Finally, we discuss risks and limitations associated with usage of the new heuristic as well as means of resolving them.

Visual field mapping consists of measuring responses to rings and wedges stimuli presented at varying visual field locations. Within each voxel the experimenter estimates the visual field position that produces the largest fMRI response. However, in reality the population of neurons in such voxel responds (with varying intensity) to a whole range of visual field locations. The region of visual space that stimulates the voxel is called the population receptive field (pRF) (Victor et al., 1994). The pRF method can provide estimates for receptive field location, size, orientation, laterality and surround suppression ((Kay et al., 2013; Zeidman et al., 2016)). To this end a series of stimuli is specifically designed to differentiate between the above parameters. Temporal responses are then used to fit model values with best support from the observed data (evidence).

In Dumoulin and Wandell (2008) the authors propose a quantitative approach for estimating population receptive field (pRF) parameters using a model-based coarse-to-fine optimization scheme. The pRF model is defined as two-dimensional Gaussian with means corresponding to pRF position in the visual field and a scalar covariance matrix with diagonal values equal to (pRF size)^2. Subsequently, model parameters are varied in order to match functional magnetic resonance imaging (fMRI) time series obtained using wedges, rings and lines stimuli displayed in a series of animations. In order to do so - Frobenius inner product of stimuli and pRF Gaussian is convolved with a space-invariant hemodynamic response function (HRF) and the residual sum of squares (RSS) between the simulation and the data is iteratively minimized starting from a seed point determined by exhaustive search on predefined parameter grid.

This model is the base for further elaboration in (Kay et al., 2013) leading to the compressive spatial summation (CSS) approach. While measuring BOLD responses to a set of contrast patterns, the

authors discover systematic deviation from linearity. The data are more accurately explained by a model in which a compressive static nonlinearity is applied after linear spatial summation. The authors conclude that the nonlinearity is present in early visual areas (e.g., V1, V2) and increases in anterior extrastriate areas (e.g., LO-2, VO-2). The effect of compressive spatial summation has been analyzed in terms of changes in the position and size of a viewed object. It is stated that compressive spatial summation is consistent with tolerance to changes in position and size, an important characteristic of object representation. A similar grid-based fitting approach is used for estimating parameters of the CSS-extended pRF model.

The CSS-pRF approach is characterized by simplicity and relatively good speed/accuracy of fit in most cases. Some of its shortcomings however are that it: i. provides only point estimates of the parameters; ii. does not account for spatial HRF variation (which, as acknowledged by the authors, may introduce systematic errors in pRF size estimates); iii. uses an explicit HRF model based on two gamma functions which do not allow for robust estimation of more informative hemodynamic parameters introduced by the Balloon-Windkessel model (Buxton et al., 1998).

The advancement proposed by this work is a Bayesian approach to joint estimation of pRF and hemodynamic parameters full posterior distributions using a forward signal generation model and Markov Chain Monte Carlo (MCMC) sampling. Furthermore, due to the computational costs incurred by MCMC, an optimized implementation using OpenCL is presented which allows one to take advantage of modern Graphics Processing Units (GPUs) in order to keep the processing time within the same order of magnitude as previous method while providing richer and more robust results.

Materials and Methods

PRF Model

A population receptive field (pRF) is the region of the visual field within which stimuli evoke responses from a local population of neurons. In (Dumoulin and Wandell, 2008) the authors proposed a model [Figure 7] of neuronal population receptive field defined by a two-dimensional Gaussian function:

$$g(x,y) = e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}$$

[1]

where $(x0, y0)$ is the receptive field center and σ is the Gaussian standard deviation. Subsequently, the predicted pRF response $r(t)$ is defined as sum of cells in element-wise (Hadamard) product of effective stimulus $s(x, y, t)$ and the Gaussian $g(x, y)$:

$$r(t) = \sum_{x,y} s(x,y,t)g(x,y)$$

[2]

The BOLD signal time series prediction $p(t)$ is then obtained by convolving $r(t)$ with a model hemodynamic response function (HRF) $h(t)$:

$$p(t) = r(t) * h(t)$$

[3]

This model is further elaborated in (Kay et al., 2013) leading to compressive spatial summation (CSS) approach (Figure 1), which defines $r(t)$ as:

$$r(t) = g \left(\sum_{x,y} s(x, y, t) g(x, y) \right)^{n}$$

[4]

where g is a gain parameter and n is an exponent parameter. This additional compressive static nonlinearity has been proven to better explain experimental data.

In contrast to previous studies, we use the CSS component for modeling the pRF response but instead of using convolution with a spatially invariant canonical HRF to obtain the predicted BOLD time series in [Equation 3], we employ the Balloon-Windkessel model described in the following section. We do this to account for per-voxel variability of parameters determining hemodynamic response.

Balloon-Windkessel Model

The hemodynamic model (Figure 2) used in this study is a combination of the Balloon model and regional cerebral blood flow (rCBF) model as introduced in (Friston et al., 2000) and used for dynamic causal modelling (Friston et al., 2003). The remainder of this section contains a brief summary of the model [Figure 6].

The Balloon component is responsible for coupling rCBF to the BOLD signal (Buxton et al., 1998). The BOLD signal is modelled as the volume-weighted sum of intrinsic extravascular signal (S_e) and the intravascular signal (S_i):

$$S = (1 - V) * S_e + V * S_i \qquad [5]$$

Relative change in this signal can further be expressed as a nonlinear function of normalized venous volume (v), normalized total deoxyhemoglobin content (q) and oxygen extraction fraction (E_0):

$$y = \frac{\Delta S}{S} = V_0 * \left(k_1 * (1 - q) + k_2 * \left(1 - \frac{q}{v}\right) + k_3 * (1 - v)\right) \qquad [6]$$

where the first term describes the intrinsic extravascular signal, the second term describes the intravascular signal, and the third term describes the effect of changing the balance of the sum in Eq. 5. V_0 indicates resting blood volume fraction and the k factors are modeled as follows:

$$k_1 = 7 * E_0$$
$$k_2 = 2 \qquad [7]$$
$$k_3 = 2 * E_0 - 0.2$$

The rate of volume change $v\dot{}$ is expressed as:

$$\tau_0 * \dot{v} = f_{in} - f_{out}(v) \qquad [8]$$

The constant τ_0 represents the average time it takes to traverse the venous compartment or for that compartment to be replenished (mean transit time) and is equal to V_0/F_0 where F_0 is the resting flow. The outflow (f_{out}) is modelled with single parameter α:

$$f_{out(v)} = v^{1/\alpha} \tag{9}$$

Steady-state values of α have been estimated by (Grubb et al., 1974) $\alpha \approx 0.38$ and (Mandeville et al., 1999) $\alpha \approx 0.36$.

Deoxyhemoglobin content change $\dot{q}$ is modelled as the difference:

$$\tau_0 \dot{q} = f_{in} \frac{E(f_{in}, E_0)}{E_0} - f_{out(v)} \frac{q}{v} \tag{10}$$

where the first term expresses the delivery of deoxyhemoglobin into the venous compartment, the second term its expulsion and $E(f_{in}, E_0)$ is the fraction of oxygen extracted from the inflowing blood.

$$E(f_{in}, E_0) = 1 - (1 - E_0)^{1/f_{in}} \tag{11}$$

Therefore, the Balloon component contains three parameters – E_0, τ_0 and α specifying respectively resting oxygen extraction fraction, mean transit time and stiffness exponent. Together they determine the flow-volume relationship of the venous balloon.

The rCBF component serves to specify the missing inflow state. To this end, following results from (Miller, K. L., Luh, W. M., Liu, T. T., Martinez, A., Obata, T., Wong, E. C., ... & Buxton, 2000) and (Friston et al., 2000) linear relationship between blood flow and synaptic activity is assumed resulting in a parsimonious model of inflow:

$$\dot{f}_{in} = s,$$

[12]

where s is a flow inducing signal generated by neuronal activity $u(t)$ subject to:

$$\dot{s} = \varepsilon u(t) - \frac{s}{\tau_s} - \frac{(f_{in} - 1)}{\tau_f}.$$

[13]

ε, τ_s and τ_f are the three unknown parameters of this component and represent respectively: the efficacy with which the neural activity causes increase in signal, the time constant for signal decay and time constant for autoregulatory feedback from blood flow (Irikura et al., 1994; Mayhew et al., 1998).

For the remainder of this work, values of $\varepsilon=1$ and $V_0=0.02$ are assumed in line with the way this model is applied in Dynamic Causal Modelling (Friston et al., 2003) software package.

Metropolis-Hastings Sampling

The Metropolis-Hastings (MH) scheme consists of an evolving state variable $\Theta = \{\mu_x, \mu_y, \sigma, \kappa, \gamma, \tau, \alpha, \rho, a, g\}$ and a list of all previously accepted values of Θ. Initial value of $\Theta(t=0)$ is provided by using existing point-estimate method for pRF parameters (Kay et al., 2013) and prior means for hemodynamic parameters (Empirical Bayes approach). Subsequently a proposal $\Theta(t+1)$ is picked in the neighborhood of

$\Theta(t)$ using normal random distributions with small variances configurable separately for each parameter. The new value of Θ replaces the current one (and is added to the list) with probability $p=min(1, p(y \mid \Theta(t+1)/p(y \mid \Theta(t)))$ which depends only on the last value of $\Theta(t)$ therefore satisfying the Markov criterion. As $t \rightarrow \infty$ distribution of accepted samples approaches the desired posterior distribution $p(\Theta \mid y)$ (Hastings, 1970). The initial amount of samples m is skipped (this process is called burn-in period) to account for the convergence of Θ to the desired distribution. Furthermore, only every k-th remaining accepted sample is taken into account in the final posterior sample (this process is called thinning) to compensate for the correlation between successive random samples and ensure unbiased sampling.

The model inversion procedure relies on a stationary noise estimate obtained by a procedure loosely inspired by (Dumoulin and Wandell, 2008) and can be summarized in the following steps: 1) Principal Components Analysis (PCA) of the stimuli; 2) General Linear Model (GLM) using a subset of n=128 first principal components as independent variables and acquired Blood Oxygenation Level Dependent (BOLD) response as dependent variable; 3) 1st order autoregressive (AR) model for the residuals of the GLM model; 4) noise statistic estimation based on AR residuals; 5) response whitening by removing AR predictions; 6) a Metropolis-Hastings (MH) scheme with 3 parameters for pRF model: horizontal position (μ_x), vertical position (μ_y) and size (σ), 5 for Balloon-Windkessel model: rate of signal decay (κ), rate of flow-dependent elimination (γ), hemodynamic transit time (τ), Grubb's exponent (α), resting oxygen extraction fraction (ρ) and 2 for compressive spatial summation which has to be applied after the Balloon-Windkessel model evaluation: exponent (a) and gain (g).

Priors (see [Table 1]) were established using an Empirical Bayes approach where applicable. The prior for the σ parameter was obtained by fitting a Gaussian distribution to single subject results from an existing point-estimation method (Kay et al., 2013). Non-informative priors with means at (0, 0) and large variances have been established for μ_x and μ_y. For hemodynamic parameters $\kappa, \gamma, \tau, \alpha, \rho$ priors reported in (Friston et al., 2003) were used. Gaussian priors with large variances and means calculated from maximum likelihood fit to the whitened detrended BOLD data were used for a and g respectively.

Table 1. MCMC prior and proposal distributions. Note 1: the parameter space is bounded so that μ_x, μ_y and σ are in the range [0, 100], a and g are positive. It is achieved by rejecting samples where any of the parameters is outside of the allowed range. Note 2: Receptive field size depends on eccentricity and visual area (Amano et al., 2009; Kay et al., 2008; Larsson and Heeger, 2006; Smith et al., 2001; Winawer et al., 2010) therefore usage of an informative prior is disputable; our method readily supports using a non-informative prior instead.

Parameter	Name	Prior (Gaussian)		Proposal (Gaussian with mean at current point estimate)
		Mean	Standard Deviation	Standard Deviation
μ_x	Receptive field horizontal center (in pixels of stimulus mask)	0	Infinity	10
μ_y	Receptive field vertical center (in pixels of stimulus mask)	0	Infinity	10
σ	Receptive field size (in pixels of stimulus mask)	33.3	Infinity	1
a	CSS Exponent	0.5	0.5	0.05
g	CSS Gain	10	10	1
κ	Vasodilatory signal decay	0.65	0.015	0.0015
γ	Rate constant for autoregulatory feedback	0.41	0.002	0.0002

	by blood flow			
τ	Transit time	0.98	0.0568	0.00568
α	Grubb's stiffness exponent	0.32	0.0015	0.00015
ρ	Deoxyhemoglobin content	0.34	0.0024	0.00024

The summary of MCMC evaluation is contained in [Table 2].

Table 2. MCMC scheme summary

Name	Expression
CSS Gaussian	$g(x, y) \sim \exp(-((x - \mu_x)^2 + (y - \mu_y)^2) / 2\sigma^2)$
CSS Summation	$r(t) = \sum_{x,y} s(x, y, t)g(x, y)$
Activity-Dependent Signal	$ds = r - \kappa s - \gamma(f-1)$
Flow Induction	$df = s$
Changes in Volume	$\tau dv = f - v^{(1-\alpha)}$
Changes in deoxyhemoglobin content	$\tau dq = f*E(f, \rho)/\rho - v^{(1-\alpha)}q/v$
BOLD Signal	$y = k_1*(1-q) + k_2*(1-q/v) + k_3*(1-v)$
CSS Exponentatiation	$y_1 = y^a$
CSS Gain	$y_2 = g*y_1$

Residuals	$res = y_{fMRI} - y_2$
Likelihood	$$p(y \mid \Theta) = p(\mu_x)p(\mu_y)p(\sigma)$$ $$p(a)p(g)$$ $$p(\kappa)p(\gamma)p(\tau)p(\alpha)p(\rho)$$ $$\frac{1}{\sigma\sqrt{2\pi}} \prod_t \exp\left(\frac{-res(t)}{\sigma_{noise}^2}\right)$$

Algorithm Listing Conventions

The following two sections describe the main CPU and GPU algorithms employed by our implementation of the estimation procedure. We assume the following conventions in the listings below. Variables are written in *italics*, constants are CAPITALIZED, pseudo-code keywords are written in **bold**.

Two distinct notations are used for describing ranges of values. The bracket notation [start, end) signifies a set of values {*start, start + 1, ... , end – 2, end – 1*}. The colon notation [*start:step:end*) signifies the following set of values: {*start, start + step, start + 2 * step, ... , start + (k - 1) * step, start + k * step*} where *k* is the biggest integer such that *start + k * step < end*. Sets are treated as row vectors where relevant.

The pseudocode refers to a number of built-in functions. bsxfun(*fun*, *A*, *B*) is inspired by a MATLAB routine with the same name. It takes a lambda expression *fun* and applies it to corresponding pairs of elements from *A* and *B*. Implicit expansion is used in case the shapes of *A* and *B* don't match

exactly. The term "implicit expansion" signifies that the smaller array is repeated across singleton dimensions in order to match the shape of the bigger one.

Routine pinv(*A)* computes a Moore-Penrose pseudoinverse of matrix *A*. The computation is based on singular value decomposition (SVD) algorithm with automatically adjusted tolerance *tol* = max(*ncols*, *nrows*) * abs(*sv*[0]) where *ncols* and *nrows* signify respectively number of columns and rows in the matrix and *sv* is an array of singular values in descending order of absolute magnitude.

Function *repmat(A, {m, n})* repeats matrix *A m* times vertically and *n* times horizontally.

Furthermore, for reason of brevity some of the convenient MATLAB syntax is employed in listing of Algorithm 2. Expression *"@power"* signifies lambda function raising its first argument to the power expressed by the second argument. Expression *A'* signifies transpose of a matrix.

Standard trigonometric functions *sin(a)* and *cos(a)* compute respectively sine and cosine of an angle *a* expressed in radians. The function *min(a, b)* returns the smaller of two scalar values *a* and *b*.

Notation *N(A)* is a highly abbreviated notation used to signify calculation of element-wise log-probabilities using Gaussian distribution with mean and standard deviation corresponding to prior distribution characteristic for quantity in array *A*.

Notation *randn(…)* is used to signify generating random values according to normal distribution with zero mean and standard deviance corresponding to step size selected for respective components of MH state variable.

Function *uniform_random(a, b)* returns a random real number from range [*a, b*].

CPU Algorithms

The Metropolis-Hastings sampling (Algorithm 1) constitutes core of the implementation. It is divided into Initialization, Data Preprocessing and Sampling phases. The Sampling phase is further

divided into GPU part and CPU part depending on the device performing bulk of the computation involved and Stepping part responsible for performing the random walk.

Initialization phase loads acceleration lookup table (*stage2*) and initial pRF parameter estimates (*xs, ys, rfsizes, expts, gains*), prepares polynomial detrending matrix (*pmatrix*) and bootstraps remaining working variables.

In Data Preprocessing phase, MRI data are divided into batches and processed using preprocess_data() GPU routine (Kernel 5).

Subsequent Sampling phase runs a predefined number of iterations (*NumberOfSteps*) of a loop consisting of GPU and CPU computation. The GPU part is responsible for executing the signal generation model in three steps: reset() routine (Kernel 2) restores initial states of the Balloon-Windkessel model variables, fwd_model() kernel (Kernel 3) performs a forward run of combined pRF and Balloon-Windkessel MRI signal generation model, whereas eval_proposal() function (Kernel 6) computes the logprobability of resulting signal given registered MRI data (evidence). The CPU part adds logprobabilities resulting from priors and for each voxel decides whether to accept state variable values or revert to old ones. The latter is done in stochastic manner by taking ratio of current state probability to previous state probability and comparing against a realization of uniformly random variable from range [0, 1]. If the random value is less, new sample is accepted, otherwise it is rejected. If new state has higher probability than the old state, it will always be accepted because ratio is larger than one. Accepted samples are added to respective per-voxel sets of samples for individual state variables. Finally, the next proposal state is generated by adding normal random variable with zero mean and predefined standard deviation to current values of each state variable.

Algorithm 1. *Metropolis-Hastings sampling* – Perform posterior sampling on joint pRF / Balloon-Windkessel model
Input: *data* – recorded fMRI signal data; *starting_ang, starting_ecc, starting_rfsize, starting_expt, starting_gain* – initial estimates for respectively: angle, eccentricity, RF size, exponent, gain; *stage2* –

stimulus premultiplied with different Gaussians and summed; *x_range, y_range, rfsize_range* – ranges of pRF parameters used for premultiplying stimulus; *W, H* – stimulus width/height; *TOT_DUR* – duration of the stimulus, *polydeg* – degree of polynomial used for detrending; *nvoxels* – number of voxels.

Output: *x_samples, y_samples, rfsize_samples, expt_samples, gain_samples, kappa_samples, gamma_samples, tau_samples, grubb_samples, rho_samples* – Values of respectively pRF center's x and y coordinates, RF size, exponent, gain and voxel hemodynamic parameters: kappa, gamma, tau, grubb, rho – sampled from the posterior distribution.

Note: Variables prefixed with g_ indicate GPU-side counterparts of variables designated by part of the name following the prefix. Assignment to such variables entails a CPU-to-GPU buffer transfer and vice versa if their values are assigned to CPU variables.

Initialization

g_stage2 ← *stage2*

pmatrix ← compute_pmatrix(*polydeg*, *TOT_DUR*)

g_pmatrix ← *pmatrix*

xs ← *starting_ecc* * cos(*starting_ang*) + *W*/2

ys ← -*starting_ecc* * sin(*starting_ang*) + *H*/2

(rfsizes, expts, gains) ← *(starting_rfsizes, starting_expts, starting_gains)*

(kappas[:]*, gammas*[:]*, taus*[:]*)* ← *(0.65*, 0.41, 0.98)

(grubbs[:]*, rhos*[:]*)* ← (0.32, 0.34)

noise_stdevs[:] ← ∞

(old_xs, *old_ys*, *old_rfsizes)* ← *(xs*, *ys*, *rfsizes)*

(old_expts, *old_gains)* ← *(expts*, *gains)*

(old_kappas, *old_gammas*, *old_taus)* ← *(kappas*, *gammas*, *taus)*

(old_grubbs, *old_rhos)* ← *(grubbs*, *rhos)*

old_logprobs[:] ← -∞

Data Preprocessing

for t ∈ [0:ntasks:TOT_DUR) **do**

$n \leftarrow$ min(*ntasks*, *nvoxels - i*)

g_data ← data[i * TOT_DUR:(i + n) * TOT_DUR]

prepare_data*(g_data, TOT_DUR, nvoxels)*

data[i * TOT_DUR:(i + n) * TOT_DUR] ← g_data

end for

Sampling

for $i \in [0, \textit{NumberOfSteps})$ **do**

    GPU Part

    for $k \in$ 0:ntasks:*nvoxels* **do**

      $n \leftarrow$ min(*ntasks*, *nvoxels - i*)

      g_data ← data[k * TOT_DUR:(k+n)*TOT_DUR]

      g_xs ← xs[k:(k+n)]

      g_ys ← ys[k:(k+n)]

      g_rfsizes ← rfsizes[k:(k+n)]

      g_expts ← expts[k:(k+n)]

      g_gains ← gains[k:(k+n)]

      g_kappas ← kappas[k:(k+n)]

      g_gammas ← gammas[k:(k+n)]

      g_taus ← taus[k:(k+n)]

      g_grubbs ← grubbs[k:(k+n)]

      g_rhos ← rhos[k:(k+n)]

      g_noise_stdevs ← noise_stdevs[k:(k+n)]

      reset()

      fwd_model()

      eval_proposal()

      Ys[k * TOT_DUR:(k+n)*TOT_DUR] ← g_Ys

      logprobs[k:(k+n)] ← g_logprobs

      noise_stdevs[k:(k+n)] ← g_noise_stdevs

    end for

    CPU Part

logprobs ← *logprobs* + N(xs) + N(ys) + N(rfsizes) +

   N(expts) + N(gains) + N(kappas) + N(gammas) +

   N(taus) + N(grubbs) + N(rhos)

ratio ← min(1, exp(*logprobs* – *old_logprobs*))

for $k \in [0, nvoxels)$ **do**

  rnd ← uniform_random(0, 1)

  if *rnd* < *ratio* **then**

    x_samples[*k*] ← *x_samples*[*k*] ∪ {xs[k]}

    y_samples[*k*] ← *y_samples*[*k*] ∪ {ys[k]}

    rfsize_samples[*k*] ← *rfsize_samples*[*k*] ∪ {rfsizes[k]}

    expt_samples[*k*] ← *expt_samples*[*k*] ∪ {expts[k]}

    gain_samples[*k*] ← *gain_samples*[*k*] ∪ {gains[k]}

    kappa_samples[*k*] ← *kappa_samples*[*k*] ∪ {kappas[k]}

    gamma_samples[*k*] ← *gamma_samples*[*k*] ∪ {gammas[k]}

    tau_samples[*k*] ← *tau_samples*[*k*] ∪ {taus[k]}

    grubb_samples[*k*] ← *grubb_samples*[*k*] ∪ {grubbs[k]}

    rho_samples[*k*] ← *rho_samples*[*k*] ∪ {rhos[k]}

  else

    xs[*k*] ← *old_xs*[*k*]

    ys[*k*] ← *old_ys*[*k*]

    rfsizes[*k*] ← *old_rfsizes*[*k*]

    expts[*k*] ← *old_expts*[*k*]

    gains[*k*] ← *old_gains*[*k*]

    kappas[*k*] ← *old_kappas*[*k*]

    gamma*s*[*k*] ← *old_gammas*[*k*]

    tau*s*[*k*] ← *old_taus*[*k*]

    grubb*s*[*k*] ← *old_grubbs*[*k*]

    rho*s*[*k*] ← *old_rhos*[*k*]

  end if

end for

Prepare for next iteration

$(old\_xs, old\_ys, old\_rfsizes) \leftarrow (xs, ys, rfsizes)$

$(old\_expts, old\_gains) \leftarrow (expts, gains)$

$(old\_kappas, old\_gammas, old\_taus) \leftarrow (kappas, gammas, taus)$

$(old\_grubbs, old\_rhos) \leftarrow (grubbs, rhos)$

$old\_logprobs \leftarrow logprobs$

Make a step

$(xs, ys, rfsizes, expts, gains) \leftarrow (xs, ys, rfsizes, expts, gains) + randn(\ldots)$

$(kappas, gammas, taus, grubbs, rhos) \leftarrow$

$(kappas, gammas, taus, grubbs, rhos) + randn(\ldots)$

end for

The compute_pmatrix() routine (Algorithm 2) computes polynomial detrending matrix. To this end, a matrix of the form:

$$A = \begin{bmatrix} 1 & 1^2 & \ldots & 1^p \\ 2 & 2^2 & \ldots & 2^p \\ \ldots & \ldots & \ldots & \ldots \\ n & n^2 & \ldots & n^p \end{bmatrix} \qquad [14]$$

is created where n is the duration of the MRI stimuli/signal and p is the chosen polynomial degree. Then, the expression:

$$B = AA^+ \qquad [15]$$

is the polynomial detrending matrix where A^+ signifies Moore-Penrose pseudoinverse of matrix A.

Algorithm 2. *compute_pmatrix()* – Compute polynomial detrending matrix
Input: *polydeg* – polynomial degree, *TOT_DUR* – stimulus duration

Output: *pmatrix* – polynomial detrending matrix

Note: Assignment to *polymat* uses MATLAB syntax but the implementation is native C++

---

polymat ← bsxfun(@power, repmat([1, *TOT_DUR* + 1), {*polydeg*+1, 1}), [0, *polydeg*]')'

polyinv ← pinv(*polymat*)

pmatrix ← *polymat* * *polyinv*

GPU Algorithms

The key observation that allowed acceleration of the sampling procedure was that the 2D visual stimuli is uniform across all voxels and can be premultiplied and summed over a wide range of Gaussians to create a lookup structure (Figure 3) which can then be used to approximate results of such operation for *any* Gaussian we encounter during the random walk by taking eight closest precomputed signals and performing a trilinear interpolation. This function is performed by the interp_gaussian() GPU routine (Kernel 1).

Kernel 1. *interp_gaussian()* - Interpolate premultiplied stimuli according to specified pRF parameters

Input: *x* – specifies μ_x of the Gaussian, *y* – specifies μx of the Gaussian, rfsize – specifies σ of the Gaussian, *gaussians* – array of premultiplied stimuli for certain range of *x*, *y* and *rfsize* parameters, *RFSIZE_RANGE={RFSIZE_MIN, RFSIZE_MAX, RFSIZE_STEP}, X_RANGE={X_MIN, X_MAX, X_STEP}, Y_RANGE={Y_MIN, Y_MAX, Y_STEP}* – specification of ranges used for premultiplied signal generation, *TOT_DUR* – stimuli duration

Output: *interp* – Interpolated signal

x ← (x – *X_MIN*) / *X_STEP*

y ← (y – Y_MIN) / Y_STEP

rfsize ← (*rfsize* – *RFSIZE_MIN*) / *RFSIZE_STEP*

x1 ← floor(*x*)

x2 ← ceil(*x*)

y1 ← floor(*x*)

y2 ← ceil(*y*)

rfsize1 ← floor(*rfsize*)

rfsize2 ← ceil(*rfsize*)

x_frac ←fract(*x*)

y_frac ←fract(*y*)

rfsize_frac ← fract(*rfsize*)

for t ∈ [0, TOT_DUR) **do**

   interp[*t*] ← interp3(*gaussians*[{*x1*, *x2*} x {*y1*, *y2*} x {*rfsize1*, *rfsize2*}], *x_frac*, *y_frac*, *rfsize_frac*)

end for

The joint pRF / Balloon-Windkessel model is handled by the following three GPU procedures. The reset() call (Kernel 2) restores initial values of Balloon-Windkessel state variables.

Kernel 2. *reset( )* – Reset simulation data to baseline values

Input: *id – global kernel instance identifier*

Input/Output: *signals* – vector of signal values, *flows* - vector of flows, *volumes* – vector of cerebral blood volume values, *deoxys* – vector of deoxyhemoglobin content values

signals[*id*] ← 0

flows[*id*] ← 1

volumes[*id*] ← 1

deoxys[*id*] ← 1

The fwd_model() function (Kernel 3) initializes variables pertaining to the joint model and uses interp_gaussian() to obtain stimuli multipled with Gaussian with parameters for current random walk state. Such signal is used as the neural activity input for the Balloon-Windkessel model. Subsequently evaluation of the latter is performed over the time corresponding to one stimulus frame. The output of Balloon-Windkessel model is then subjected to remaining part of the compressive spatial summation model, namely exponentiation and multiplication by the gain factor to account for unknown MRI signal units. Results are detrended using polynomial detrending matrix (*pmatrix*). Finally, Balloon-Windkessel

model state is updated for current voxel and output copied from local array to global buffer accessible from CPU.

Kernel 3. *fwd_model()* – Perform joint pRF / Balloon-Windkessel model simulation
Input: *id* – global kernel instance identifier, *stage2* – precomputed product of Gaussian by stimulus multiplication, used by *interp_gaussian()* call, *pmatrix* – polynomial detrending matrix, *xs* – array of horizontal positions of 2D Gaussian centers, *ys* – array of vertical positions of 2D Gaussian centers, *rfsizes* – array of standard deviations of 2D gaussians, *exponents* – array of exponent parameter values in pRF model, *gains* – array of gain values in pRF model, *kappas* – array of kappa parameter values in B-W model, *gammas* – array of gamma parameter values in B-W model, *taus* – array of tau parameter values in B-W model, *grubbs* – array of Grubb's coefficient values in B-W model, *rhos* – array of rho parameter values in B-W model, *signals* – vector of signal values, *flows* - vector of flows, *volumes* – vector of cerebral blood volume values, *deoxys* – vector of deoxyhemoglobin content values   **Input/Output:** *Ys* – vector of signal values simulated by joint pRF / B-W model
Helper variables initialization u_x ← xs[id] u_y ← ys[id] rfsize ← rfsizes[id] kappa ← kappas[id] gamma ← gammas[id] tau ← taus[id] grubb ← grubbs[id] rho ← rhos[id] exponent ← exponents[id] gain ← gains[id] s ← signals[id]

f ← flows[id]

v ← volumes[id]

q ← deoxys[id]

k1 ← 7 * rho[id]

k2 ← 2

k3 ← 2 * rho[id] – 0.2

V0 ← 0.02

Approximation of 2D Gaussian PDF times stimuli product (1st step of pRF model)

z ← *interp_gaussian*(u_x, u_y, rfsize, stage2)

Balloon-Windkessel model

Y ← array(NSTEPS)

for n ∈ [0, NSTEPS) **do**

  t ← n / SUBDIV

  Y[n] ← V0 * (k1 * (1 - q) + k2 * (1 - q / v) + k3 * (1 - v))

  ds ← z[t] - kappa * s - gamma * (f - 1)

  df ← s

  dv ← (f - pow(v, (1 / grubb))) / tau

  dq ← (f * (1 - pow((1 - rho), (1 / f))) / rho - pow(v, (1 / grubb)) * q / v) / tau

  s ← s + ds * TSTEP

  f ← f + df * TSTEP

  v ← v + dv * TSTEP

  q ← q + dq * TSTEP

end for

Second step of pRF model (exponent and gain)

Y_1 ← array(TOT_DUR)

for t ∈ [0, TOT_DUR) **do**

  Y_1[t] ← copysign(pow(fabs(Y[t * SUBDIV]), exponent) * gain, Y[t * SUBDIV])

end for

Polynomial detrending (3rd step of pRF model)

Y_2 ← pmatrix * Y_1

for t ∈ [0, TOT_DUR) **do**

    Y_1[t] ← Y_1[t] - Y_2[t]

end for

Update Balloon-Windkessel state

signals[id] ← s

flows[id] ← f

volumes[id] ← v

deoxys[id] ← q

Output

for t ∈ [0, TOT_DUR) **do**

    Ys[id * TOT_DUR + t] ← Y_1[t]

end for

Kernel 4. *mean_squared_error ()* – Compute mean squared error of the model

Input: *id – global kernel instance identifier, Ys –* vector of signal values simulated by joint pRF / B-W model, *data –* vector of acquired experimental data

Input/Output: *mse –* mean squared error of the model

accum ← 0

ofs ← id * TOT_DUR

for t ∈ [0, TOT_DUR) **do**

    accum ← accum + pow(data[ofs] - Ys[ofs], 2.0)

    ofs ← ofs + 1

end for

mse[id] ← accum / TOT_DUR

Kernel 5. *prepare_data ()* – Perform polynomial detrending on evidence data
Input: *id* – global kernel instance identifier, *pmatrix* – polynomial detrending matrix
Input/Output: *data* – vector of acquired experimental data

ofs ← id * TOT_DUR
poly_fit ← matmul(*pmatrix*, *data*)
for $t \in$ [0, TOT_DUR) **do**
   data[*ofs* + *t*] ← *data*[*ofs* + *t*] – *poly_fit*[*t*]
end for

Kernel 6. *eval_proposal ()* – Compute logprobability of simulated signal given evidence
Input: *id* – global kernel instance identifier, *Ys* – vector of signal values simulated by joint pRF / B-W model, *data* – vector of acquired experimental data, *noise_stdevs* – estimated noise amplitude
Output: *logprobs* – logprobability of simulated signal being equal to real MRI signal minus noise, given evidence (*data*)

mean ← 0
stdev ← 0
ofs ← id * TOT_DUR
for $t \in$ [0, TOT_DUR) **do**
   diff ← Ys[*ofs* + *t*] – data[*ofs* + *t*]
   mean ← *mean* + *diff*
   stdev ← *stdev* + *diff*^{2}
end for
mean ← *mean* / TOT_DUR
stdev ← sqrt(*stdev* / TOT_DUR – *mean*^{2})
noise_stdevs[*id*] ← min(*noise_stdevs*[*id*], stdev)
stdev ← *noise_stdevs*[*id*]
logp ← 0
for $t \in$ [0, TOT_DUR) **do**
   diff ← Ys[*ofs* + *t*] – data[*ofs* + *t*]

$logp \leftarrow logp + \text{normal\_logpdf}(0, stdev, diff)$

end for

$logprobs[id] \leftarrow \text{logp}$

Stimulus

Visual stimuli were created using extensions from the PsychToolbox (Brainard, 1997; Pelli, 1997) within the Matlab programming environment. An LCD projector was used to image the stimuli onto a back projection screen within the bore of the magnet. Subjects viewed the display through an angled mirror. The maximal stimulus radius was 9.15° of visual angle. Each stimulus image was at a resolution of 768 x 768 pixels. Binary stimuli masks were defined at time points which coincided with the slice-time corrected fMRI data and down sampled to a resolution of 100x100 pixels to improve efficiency of the MCMC sampling routines. For all experiments a small dot at the centre of the image served as a fixation point (4 pixel diameter) which changed color at randomized intervals between 1 and 5 seconds. Subjects were asked to fixate on the dot and indicate each time the dot changed color using an MRI compatible button box. The stimuli consisted of achromatic contrast patterns (spatially pink noise) overlaid with randomly positioned and scaled visual object stimuli from Kriegeskorte et al. (2008). The full stimuli patterns can be found at http://cvnlab.net/analyzePRF/.

In the first set of experiments the stimulus consisted of a combination of rotating wedge, contracting ring and sweeping bar stimuli. The stimuli can be broken down into the following temporal structure:

- Wedge stimuli covering a 90° angle swept 8 full counter-clockwise rotations. The wedge took 32 seconds to complete one full 360° rotation.

- Bars with a width equal to 12% of the stimulus diameter were swept across 8 unique (cardinal and oblique) directions. Each bar sweep took 32-seconds to complete (28-second sweep followed by a 4-second rest).

- Contracting ring stimuli were swept across the visual field over the course of 32-seconds (28-second sweep followed by a 4-second rest). This was repeated 5 times. The widths of the rings were scaled linearly with eccentricity to compensate for cortical magnification in visual cortex (Cowey and Rolls, 1974; Rovamo and Virsu, 1979).

- 16 second rest periods with mean luminance (zero contrast) images were placed at the start, between stimuli types and at the end of the stimulus (taking into account the additional 4 second rest periods at the end of bar and ring sweeps).

The stimulus took a total of 12 minutes 20 seconds to complete (16 + 8*32 + 16 + 4*32 + 12 + 4*32 + 12 + 5*32 + 12 s).

In-Vivo Datasets

All MRI data was acquired with a 3T Siemens Prisma MR system using a 64-channel head coil. One male subject with no history of eye disease (aged 28) underwent all of the fMRI acquisition protocols. Foam padding and audio headphones minimized head motion. The functional MRI data were acquired using a 2D-EPI sequence (TR/TE = 1518/30 ms, 23 slices, 3x3x2.5 mm 3 resolution, 20% distance factor, FOV = 192x192 mm 2 , 12% phase oversampling). The slices were oriented parallel to the calcarine sulcus, covering the majority of occipital cortex. A whole brain 2D-EPI sequence with 49 slices and 5 volumes was acquired to aid with structural registration. All other parameters of the whole brain sequence were the same as in the fMRI acquisition. A T1-weighted anatomical image (TR/TE = 2000/2.93 ms, 1x1x1 mm 3 resolution) was acquired and used to reconstruct the cortical surfaces. In order to estimate local field distortions we acquired a B0 field map (TR/TE = 1020/10 ms, 3x3x3 mm 3 resolution). Functional scans for our dataset used the combined rotating wedge, contracting ring and sweeping bar stimuli and were acquired over 488 volumes. A total of 3 fMRI scans were performed in the session and later used to investigate scan-rescan reproducibility of the estimated pRF parameters.

Image Preprocessing

The first 5 volumes of each fMRI run were discarded to allow magnetization to reach a steady state. Preprocessing of the fMRI data was performed within SPM12 using the following procedure. Slice-time correction was applied to adjust for differences in slice acquisition times. The fMRI data was aligned to correct for movement artifacts and co-registered with the whole brain reference data. The fMRI and reference data were spatially unwarped (Andersson et al., 2001; Jezzard and Balaban, 1995) to improve coregistration between the EPI and structural data (Hutton et al., 2002). The reference data was then coregistered with the T1-weighted structural image and the same rigid-body transformation applied to the fMRI data. Binary stimuli masks were defined at time points which coincided with the slice-time corrected fMRI data and down sampled to a resolution of 100x100 pixels to improve efficiency of the nonlinear optimization routines.

The T1-weighted anatomical images were processed with FreeSurfer (Dale et al., 1999) for white and grey matter segmentation and cortical surface reconstruction. FreeSurfer's Desikan-Killiany atlas (Desikan et al., 2006) was used to select cortical regions within the occipital lobe and to reduce to the amount of cortex over which we fit the pRF models. The preprocessed fMRI data were sampled at distance corresponding to 20% of cortical thickness along normals of FreeSurfer's fsaverage surface. All CSS-pRF model estimations were performed on the surface projected fMRI responses. Visual field delineation was performed manually using the polar angle maps projected onto inflated cortical surfaces (Wandell et al., 2007).

Experiments

To fully evaluate the impact of introducing the lookup table heuristic we have performed an exhaustive range of tests on simulated data and concluded with a demonstration of the new algorithm as applied to a whole-ROI in-vivo dataset.

We performed a test of the lookup table signal reconstruction accuracy compared to full pRF model evaluation. To this end we computed the root mean squared error (RMSE) of the sum of the product of the stimulus and the Gaussians with different positions and sizes. We generated 1000 Gaussians with randomized parameters and computed two signals – i. by using the full pRF model evaluation (the reference signal) and ii. by using the lookup table. We compared the RMSE between the two relative to the reference signal amplitude.

We performed a series of MCMC runs on the simulated data in order to reliably compare estimated means with the known ground truth. To this end, we generated 90 time series with pRF parameters spread uniformly across the available space and hemodynamic parameters picked randomly according to their empirical prior distributions. We added noise to the resulting signals using contrast-to-noise ratio (CNR) typical for fMRI studies – 5:1. We used randomized starting points in the MCMC scheme and used a time limit of 3 hours of sampling. We didn't apply any burn-in or thinning, instead we report mESS measures along with the parameter estimates.

In order to add weight to our findings we have executed the above MCMC scheme for different combinations of hemodynamic (5-parameter Balloon and 3-parameter Balloon) and pRF models (CSS and classical Dumoulin-Wandell). Furthermore, we have performed the above using implementations with and without the usage of the lookup table. Therefore in total we have evaluated 8 variants of the above MCMC scheme – QPrf with LUT using CSS-pRF and 5-parameter Balloon model HRF (Q_CSS_5 and Q_CSS_5_bis, two runs to assess convergence), QPrf with LUT using Dumoulin-Wandell pRF and 5-parameter Balloon model HRF (Q_DW_5), QPrf with LUT using CSS-pRF and 3-parameter Balloon model HRF (Q_CSS_3), QPrf with LUT using Dumoulin-Wandell pRF and 3-parameter Balloon model HRF (Q_DW_3), QPrf without LUT using CSS-pRF and 5-parameter Balloon model HRF

(noLUT_CSS_5 and Q_CSS_5_bis, two runs to assess convergence), QPrf without LUT using Dumoulin-Wandell pRF and 5-parameter Balloon model HRF (noLUT_DW_5), QPrf without LUT using CSS-pRF and 3-parameter Balloon model HRF (noLUT_CSS_3), QPrf without LUT using Dumoulin-Wandell pRF and 3-parameter Balloon model HRF (noLUT_DW_3).

In order to put the LUT approach to test within another sampling framework we decided to use RStan (Carpenter et al., 2017). Due to excessive run time we performed a test on a single voxel of simulated data using RStan's NUTS sampler without using the LUT and using the LUT. NUTS settings included 1000 leapfrog iterations of warmup and 2000 leapfrog iterations of sampling. We used the same Bayesian model and the same priors as in QPrf and compared the estimation results to one another and the ground truth. We report as well differences in run time.

Finally, we applied accelerated QPrf model with CSS-pRF component and 5-parameter Balloon model to estimate all 50000 voxels in the visual ROI of the brain of one subject. We present parameter maps for pRF parameters and report our findings with respect to covariance between parameters.

Results and Discussion

Simulated Data Results

Figure 4 illustrates the relative RMSE error between the signal obtained from the lookup table and the signal generated using full evaluation of the model for 1000 Gaussians with randomly picked parameters. Mean RMSE relative to amplitude was equal to 2%. There were 22 points (2%) exceeding relative error of 3% with the maximum error below 5%.

Across the parameter estimates for simulated data we observed differing degree of correlation between RF Size, Exponent and Gain. The max(p-value) (Friston et al., 1999) approach revealed significant correlation between gain and exponent. While this is not in line with the "ground truth" of the

generation method were both values were picked independently - it is not a surprising result. Both exponent and gain affect the scaling of the signal. In the original compressive pRF model which included bringing simulated signal to unit length they were intended to influence the "speed" of signal edges. In our simulation and MCMC method we rely on model of physical fMRI signal in which the influence of gain and exponent seems to be largely interchangeable. The role of RF-Size might be the following: as the standard deviation of correctly centered Gaussian increases, a growing portion of the probability mass is placed outside of the considered stimulus area (100x100 in scaled stimulus space). This results in the need to increase estimates of gain and/or exponent to maintain the observed signal amplitude. This highlights the intrinsic imperfections of the model. We found no other correlations in the estimates.

Model Comparison Results

We report the most strict (choice of batch size giving the smallest result) multivariate ESS (Vats et al., 2015) measures for all runs (Table 3). In a single plot (Figure 5) we present estimated mean comparisons between noLUT_CSS_5, noLUT_CSS_3, noLUT_DW_5, noLUT_DW_3, Q_CSS_5, Q_CSS_3, Q_DW_5 and Q_DW_3 and the ground truth. In (Figure 6) we show estimated means and ground truth for the exponent which is present only in noLUT_CSS_5, noLUT_CSS_3, Q_CSS_5 and Q_CSS_3 models. Furthermore we report log probability estimates for all methods (Table 4) and present a per-voxel plot (Figure 7).

Table 3. Multivariate Effective Sample Size (mESS) measures for runs of different pRF/HRF models

HRF Model	pRF Model	With Lookup Table		Without Lookup Table	
		Mean	Stdev	Mean	Stdev
3-param Balloon	CSS	2703	556	2239	355
3-param Balloon	DW	2196	391	2570	418
5-param	CSS	1847	482	2756	354

Balloon					
5-param Balloon	DW	2265	210	2942	216

Table 4. Log probability comparison between different pRF/HRF models within QPrf with and without the usage of the lookup table.

HRF Model	pRF Model	With Lookup Table	Without Lookup Table
3-param Balloon	CSS	665	360
3-param Balloon	DW	370	242
5-param Balloon	CSS	1122	1207
5-param Balloon	DW	1140	1152

The mESS measures obtained (Table 3) were satisfactorily large. Convergence between two runs of Q_CSS_5 was almost perfect. PRF parameter estimates common to all models (Figure 5) were largely consistent across models. We observed bigger discrepancies in pRF size and gain estimates. Position estimates were identical up to a few pixels (in rescaled – 100x100 stimulus coordinate system). Exponent estimates were also largely consistent (Figure 6) with no evidence of systematic over-/underestimation in any particular model. Hemodynamic estimates were not deviating significantly from their prior means and no systematic bias was observed. Log probabilities (Table 4, Figure 7) were higher for 5-parameter Balloon model both with and without the lookup table which is unsurprising since this was the model used to generate the data. Log probability was of similar magnitude with and without the lookup table.

fMRI Results

We have estimated means, variances and confidence intervals for each parameter i. directly from parameter samples and ii. by fitting multivariate Gaussian distribution as the posterior density model p(Θ |

y) for whole-state samples. Fitted Gaussian means were within 1SD (computed directly from samples) of the sample-based mean for 90% of the voxels, supporting Gaussian character of the distributions.

Furthermore we computed pairwise correlation coefficients using direct samples approach for each pair of model parameters. Patterns of correlation covering the whole [-1, 1] range have been observed with most points presenting close to zero correlation for all pairs of parameters. Spatial distribution of correlation coefficients didn't exhibit any meaningful pattern suggesting that higher absolute values of correlation coefficient should be attributed to estimation uncertainty (within our model the shape of time series matching observed time series might be achieved by varying either pRF or hemodynamic parameters) rather than any systematic change of pRF / hemodynamic configuration depending on region. This stays in line with (Dumoulin and Wandell, 2008) who quote hemodynamic parameters as sources of additional uncertainty in pRF parameter estimation.

In (Figure 8) we present a whole-ROI map of pRF parameters for a single subject. The spatial distribution of angle and eccentricity is in line with literature (Dumoulin and Wandell, 2008). PRF size exhibits significant local variations superposing the expected pattern of pRF size increase with increased eccentricity. Gain and exponent estimates appear largely homogenous with local variations without any systematic pattern. Hemodynamic parameter estimates were largely homogenous close to the prior means.

Performance

As opposed to (Adaszewski et al., 2016) where we obtained samples from the posterior distribution for a limited number of points (~1000), in the present study using a new sampling software implementation all ~50000 points in the fMRI dataset were sampled. The runtime of the MH scheme was 48 hours when executed on an NVidia Quadro K2000 GPU (384 computation cores) card which allowed us to perform MH schemes for 384 observation points in parallel. It is a marked improvement over the

previously estimated time of ~16 days required to perform MH sampling on all 50000 voxels . The computational efficiency improvements can be fully attributed to the Gaussian premultiplication and interpolation approach. Usage of a lookup table (LUT) eliminates the overhead of computing the exact product of the 2D Gaussian and the visual stimulus mask. This overhead is significant and would dominate the total run time of the GPU code. The main advantage of a LUT is the reduction of dimensionality of data involved in computing the "pRF Gaussian"-related component of the generative model from 3 (stimulus/Gaussian in 2D+time) to 1 dimension (time). In the case of this study - due to usage of the LUT - computing the Gaussian-related component of the model involves just T=734 tri-linear interpolations (where T = duration of the stimulus) for each voxel in each MCMC iteration. Without a LUT this computation would involve 734 evaluations of 2D Gaussian density and multiplications with corresponding stimulus value per each pixel of the stimulus mask. Therefore, with a stimulus size of 100x100 the estimated performance drop without the LUT would be on the order of 10000. It is therefore evident that the LUT brings tremendous performance advantages and could considerably accelerate not only GPU- but also CPU-based samplers. The runtime of RStan's NUTS sampling scheme for one voxel was 5 hours using the LUT method and 20 days using full Gaussian evaluation, while giving approximately the same results (Table 5).

Table 5. Comparison of ground truth, RStan with LUT and RStan without LUT parameter values for one voxel.

	X	Y	RFSize	Expt	Gain	Kappa	Gamma	Tau	Grubb	Rho
Ground Truth	19.55	13.63	35.59	0.18	23.05	0.65	0.41	0.98	0.32	0.34
With LUT	19.79	13.77	23.53	0.45	21.7	0.65	0.41	1.02	0.32	0.34
Without LUT	19.25	13.77	21.01	0.38	20.08	0.66	0.41	1.01	0.32	0.34

Tradeoffs of Using a Lookup Table

Regarding the tradeoffs associated with the usage of a LUT heuristic we recognize two types: i. impact on specific behaviors of the MCMC sampler; ii. impact on aggregated metrics such as model log-evidence. General limitations of the method are the following: i. parameter space is bounded by the lookup table; ii. resolution of the lookup table influences accuracy of approximation of the pRF signal; iii. doubling the LUT resolution causes 8-fold increase in lookup table size (frequently seen computation/memory tradeoff – the burden of cubic increase in computational complexity is replaced by cubic increase of LUT size); iv. the assumption about local linearity of the pRF signal form as function of Gaussian parameters is problematic if we want to sample large parameter space because the LUT has to be coarse (due to memory limitations). We believe that the impact of these limitations on specific sampler functions has been observed within this study as abnormalities such as slightly inaccurate estimates of exponent, gain and RF size. Impact of change in both polar angle, radius and RF size parameters is smaller when RF size is big. However it has increasing impact when RF size is small (Figure 4). This is particularly visible when polar angle and radius of the pRF are not perfectly aligned with the ground truth, i.e. with big RF size the signal change is slighter than with small RF size where the peaks will be visibly shifted. Smaller RF size requires the polar angle and radius to match the ground truth much more accurately in order to have good log-evidence and accept a sample. We therefore hypothesize that due to inherent inaccuracy of LUT interpolation RF sizes are positively biased which entails (as compensation) positive bias for gain and exponent as well - since enlarging the Gaussian usually decreases signal amplitude – this is compensated by increased exponent and gain. A few potential solutions come to mind with regards to the above limitations: i. the sampler could be allowed to wander away from parameter bounds covered by the LUT and perform full pRF model computation as long as it stays outside of the LUT; presumably the sampler would spend most of the time within the most probable region (covered by the LUT) therefore this extension would not be significantly detrimental to the performance; ii. quality of interpolation could be improved by using a signal "warping" scheme rather than tri-linear approach; this

would imply determining how the signal is "warped" between neighboring cells of the LUT and applying such transformation proportionately to the position of desired parameter point between the LUT cells; to this end we could adapt existing schemes such as Dynamic Time Warping (DWT) (Gupta et al., 1996) or develop new approaches; one potential novelty we considered was usage of a "1.5D" diffeomorphic transform which could be computed from signal change as we follow sub-steps between two LUT cells.

Tuning the MCMC Sampler

As commonly observed in Monte-Carlo approaches - we believe that recalibration might be necessary depending on the scanning protocol, visual stimulus, as well as chosen hemodynamic (3-/5-parameter Balloon or fixed HRF) and pRF (Dumoulin-Wandell or CSS) components of the model. In the present study we have relied on multivariate effective sample size (mESS) measure (Vats et al., 2015) in order to establish the thinning factor. For the fMRI data, knowing the ESS (ranging from 8 to 50) of a run without thinning we kept increasing the thinning factor as long as the ESS of any thinned series remained within +/- 1 sample. This resulted in the choice of 4 as the thinning factor. We have used the Heidelberg-Welch measure (Heidelberger and Welch, 1983) in a mass-univariate scheme to assess stationarity of the posterior distribution. As burn-in we have arbitrarily chosen to discard half of the samples. We have since reflected upon our thinning approach and our method of choice in the future would be not to use thinning at all (Link et al. 2012; MacEachern et al., 1994; Geyer, 1992) and report ESS measures instead like we do for the simulated data.

Ambiguities Between pRF and HRF Parameter Estimation

Joint estimation of pRF-HRF parameters is susceptible to ambiguity due to interdependencies between the former and the latter. Bayesian model inversion (either by means of free energy optimization or MCMC sampling) allows to quantify the degree of uncertainty implicated in simultaneous estimation

of pRF and HRF parameters by means of the covariance matrix which is absent in point-estimate methods (such as analyzePRF - (Kay et al., 2013)). Given the stochastic nature of MCMC sampling a systematic bias in mean estimates shouldn't be present unless stemming from or substantiated by further causes. We discuss examples of lookup-table related biases in the "Tradeoffs of Using a Lookup Table" subsection in the Discussion. On the other hand any covariance between pRF and HRF parameters should be reflected in the estimated covariance matrix. Bias in the mean value of gathered samples will be present if a predetermined HRF response is used as the hemodynamic model as opposed to one where HRF parameters are simultaneously sampled and response expressed as a mathematical function. Other sources of estimation uncertainty may include observation noise, subjects' movement or signal changes not accounted for by used neuronal and hemodynamic models. (Zeidman et al., 2016) include a short discussion of estimation uncertainty and underline that it is important to take into account when making inferences. Using a specifically designed stimulus (e.g. alternating sweep directions - (Harvey and Dumoulin, 2011)) can help to disambiguate between pRF and HRF parameters by reducing the possibility of compensating wrong pRF position with parameters influencing HRF lag. Noteworthy the stimulus used in the present study uses such counterbalanced bar sweeps. The width of pRF remains the most troublesome parameter to disambiguate against HRF width. At this point we believe that this issue remains open and model inversion methods allowing to estimate full covariance matrix allow us to make the most informed inferences by taking into account the estimated covariance between pRF and HRF parameters.

Future Development: Performance

Current implementation of the MH scheme exhibited satisfactory robustness in testing scenarios. As expected with MCMC approaches, computational load necessary to perform a full sampling run remains the biggest limitation to gathering very large numbers of samples for multiple subjects. Thanks to

the advancements presented in this article however, the extension to large datasets becomes more of an inconvenience rather than an impossibility.

To further shorten the execution time, a number of optimizations remain to be tested. Evaluation of hemodynamic state equations is currently performed using full numerical integration and the time step used is that of the scan repetition time (TR) divided by 20; simple improvement in this part of the code involves decreasing the subdivision factor as long as equations remain stable; reduction to 10 would offer 2-fold speed improvement, whereas reduction to 5 would accelerate the process 4-fold. We haven't thoroughly optimized this parameter and assumed 20 as an arbitrary value.

A more elaborate improvement would be to re-parametrize state equations using Volterra kernels as suggested in (Friston et al., 2003). This approach would allow to eliminate costly integration altogether, however as before a balance between precision and number of Volterra coefficients would have to be carefully evaluated to maintain performance improvement without degrading simulation fidelity.

As the two-step optimization method described by (Dumoulin and Wandell, 2008) with compressive spatial summation extension by (Kay et al., 2013) takes on average 24 hours using CPU implementation we were aiming to stay within the same order of magnitude and with the ensemble of aforementioned techniques we have achieved this goal. We haven't compared the performance against pure Dumoulin-Wandell classical model.

Furthermore, addressing the shortcomings of the sampler itself could make the algorithm orders of magnitude faster. A comprehensive overview of gradient-free and gradient-based MCMC samplers applied to Dynamic Causal Modelling (DCM) is presented in (Sengupta et al., 2016, 2015). The authors conclude that gradient-based Hamiltonian Monte-Carlo and Langevin Monte-Carlo both offer superior computational performance than Metropolis-Hastings algorithm. Among the gradient-free samplers, single chain adaptive MCMC is highlighted as possessing better efficiency and requiring no tuning. All of

the above schemes are good candidates for replacing our current Metropolis-Hastings approach which was chosen primarily due to its simplicity and in order to avoid unexpected interactions with the new heuristic.

Finally, once the validity of chosen pRF model and its relationships with hemodynamic parameters is established using sampling methods, GPU implementation of Variational Bayes approaches such as (Zeidman et al., 2016) should offer much better estimation performance than an MCMC scheme.

Future Development: Population Receptive Fields

In the study under consideration, visual stimuli comprised wedges, rings and bar patterns similar to (Dumoulin and Wandell, 2008), however the algorithm permits the use of any sequence of images allowing for completely arbitrary stimuli.

A circular pRF model without inhibitory surround was used, however with a Gaussian premultiplication lookup table approach an attempt could be made to model elliptical pRF with inhibitory surround. This would require to perform n-linear interpolation depending on the sophistication of target pRF model and would significantly increase the size of lookup table which might prove to be the limiting factor to this solution. Decomposing different "parts" (e.g. horizontal, vertical, surround, center, etc.) of the pRF Gaussian into separate lookup tables could prove to be sufficient remedy to this instance of the "curse of dimensionality".

Sampling approaches offer a powerful tool to investigate the ensemble of pRF, hemodynamic as well as other parameters such as anatomical T1-weighted images, Diffusion-Weighted Images (DWI), Voxel Based Quantification (VBQ) results (Draganski et al., 2011), etc. as long as they're coupled by a common signal generation model. Longitudinal changes as well as function-behavior relationships (Rigoux and Daunizeau, 2015) could also be investigated along the same lines.

Conclusion

The method presented in this work combines a population receptive field (pRF) model with neuronal hemodynamic state equations and a Markov Chain Monte Carlo (MCMC) sampling approach in order to estimate full Bayesian posterior distributions of visual and hemodynamic parameters based on a sequence of visual stimuli. Thanks to the novel lookup table heuristic we were able to perform large sampling runs on whole-ROI data within a short allotted time. We showed that the proposed heuristic doesn't introduce significant error to the signal generated in the forward model. We further showed that estimates obtained with and without using the lookup table are consistent. We inspected the log probability with and without the heuristic across many different models and demonstrated that they are consistent. Using in-vivo dataset we verified the correlation of pRF and hemodynamic parameters and concluded that no spatial pattern can be observed which suggests that using a canonical hemodynamic response function for the entire set of observation points as employed by other methods might be justified.

We discussed a number of future directions for the development of our methodology including adaptive equation integration, application of Volterra kernels, elliptical Gaussian field with inhibitory surround using lookup table decomposition, integration of other modalities (anatomical, functional and behavioral) as well as longitudinal component and group studies.

We also proposed that the pRF/hemodynamic sampling approach could be adapted to serve as validation method for Variational Bayes approaches.

Finally, hoping that members of the neuroimaging and vision communities will find our approach useful we provide a stand-alone toolbox for MCMC sampling of pRF/hemodynamic parameter

distributions called QPrf. The software is provided along with its source code under the terms of GNU General Public License version 3 (GNU GPLv3) and is available in the following GitHub repository: https://github.com/sadaszewski/qprf.

Funding

References

Adaszewski, S., Slater, D., Melie-García, L., Draganski, B., 2016. Posterior distribution of receptive field and hemodynamic parameters using Markov chain Monte Carlo, in: 22ND ANNUAL MEETING OF THE ORGANIZATION FOR HUMAN BRAIN MAPPING. p. 2354.

Amano, K., Wandell, B.A., Dumoulin, S.O., 2009. Visual field maps, population receptive field sizes, and visual field coverage in the human MT+ comple. J Neurophysiol. 102, 2704–18.

Andersson, J.L., Hutton, C., Ashburner, J., Turner, R., Friston, K., 2001. Modeling geometric deformations in EPI time series. Neuroimage 13, 903–919. doi:10.1006/nimg.2001.0746

Brainard, D.H., 1997. The Psychophysics Toolbox. Spat. Vis. 10, 433–6.

Buxton, R.B., Wong, E.C., Frank, L.R., 1998. Dynamics of blood flow and oxygenation changes during

brain activation: The balloon model. Magn. Reson. Med. 39, 855–864. doi:10.1002/mrm.1910390602

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M.A., Li, P., Riddell, A., 2017. Stan: A Probabilistic Programming Language. J. Stat. Softw. 76, 1–32. doi:10.18637/jss.v076.i01

Cowey, a, Rolls, E.T., 1974. Human cortical magnification factor and its relation to visual acuity. Exp. Brain Res. 21, 447–454. doi:10.1007/BF00237163

Dale, A.M., Fischl, B., Sereno, M.I., 1999. Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction. Neuroimage 9, 179–194. doi:10.1006/nimg.1998.0395

Desikan, R.S., Ségonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maguire, R.P., Hyman, B.T., Albert, M.S., Killiany, R.J., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage 31, 968–980. doi:10.1016/j.neuroimage.2006.01.021

Draganski, B., Ashburner, J., Hutton, C., Kherif, F., Frackowiak, R.S.J., Helms, G., Weiskopf, N., 2011. Regional specificity of MRI contrast parameter changes in normal ageing revealed by voxel-based quantification (VBQ). Neuroimage 55, 1423–1434. doi:10.1016/j.neuroimage.2011.01.052

Dumoulin, S.O., Wandell, B.A., 2008. Population receptive field estimates in human visual cortex. Neuroimage 39, 647–660. doi:10.1016/j.neuroimage.2007.09.034

Friston, K.J., Harrison, L., Penny, W., 2003. Dynamic causal modelling. Neuroimage 19, 1273–1302. doi:10.1016/S1053-8119(03)00202-7

Friston, K.J., Holmes, a P., Price, C.J., Büchel, C., Worsley, K.J., 1999. Multisubject fMRI studies and conjunction analyses. Neuroimage 10, 385–396. doi:10.1006/nimg.1999.0484

Friston, K.J., Mechelli, A., Turner, R., Price, C.J., 2000. Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics. Neuroimage 12, 466–77. doi:10.1006/nimg.2000.0630

Grubb, R.L., Raichle, M.E., Eichling, J.O., Ter-Pogossian, M.M., 1974. The Effects of Changes in PaCO2 Cerebral Blood Volume, Blood Flow, and Vascular Mean Transit Time. Stroke 5, 630–639. doi:10.1161/01.STR.5.5.630

Gupta, L., Molfese, D.L., Tammana, R., Simos, P.G., 1996. Nonlinear alignment and averaging for estimating the evoked potential. IEEE Trans. Biomed. Eng. 43, 348–356.

Harvey, B.M., Dumoulin, S.O., 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. doi:10.1523/JNEUROSCI.2572-11.2011

Hastings, W.K., 1970. Monte carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109. doi:10.1093/biomet/57.1.97

Heidelberger, P., Welch, P.D., 1983. Simulation run length control in the presence of an initial transient. Oper. Res. 31, 1109–1144.

Hutton, C., Bork, A., Josephs, O., Deichmann, R., Ashburner, J., Turner, R., 2002. Image distortion correction in fMRI: A quantitative evaluation. Neuroimage 16, 217–240. doi:10.1006/nimg.2001.1054

Irikura, K., Maynard, K.I., Moskowitz, M. a, 1994. Importance of nitric oxide synthase inhibition to the attenuated vascular responses induced by topical L-nitroarginine during vibrissal stimulation. J. Cereb. Blood Flow Metab. 14, 45–8. doi:10.1038/jcbfm.1994.7

Jezzard, P., Balaban, R.S., 1995. Correction for geometric distortion in echo planar images from B0 field variations. Magn. Reson. Med. 34, 65–73. doi:10.1002/mrm.1910340111

Kay, K.N., Naselaris, T., Prenger, R.J., Gallant, J.L., 2008. Identifying natural images from human brain activity. Nature 452, 352–5.

Kay, K.N., Winawer, J., Mezer, A., Wandell, B.A., 2013. Compressive spatial summation in human visual cortex. J. Neurophysiol. 110, 481–94. doi:10.1152/jn.00105.2013

Klein, B.P., Harvey B. M., Dumoulin, S.O., 2014. Attraction of position preference by spatial attention throughout human visual cortex. Neuron 84, 227–237.

Kriegeskorte, N., Mur, M., Ruff, D. a., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., Bandettini, P. a., 2008. Matching Categorical Object Representations in Inferior Temporal Cortex of Man and Monkey. Neuron 60, 1126–1141. doi:10.1016/j.neuron.2008.10.043

Larsson, J., Heeger, D.J., 2006. Two retinotopic visual areas in human lateral occipital cortex. J Neurosci. 26, 13128–42.

Mandeville, J.B., Marota, J.J., Ayata, C., Zaharchuk, G., Moskowitz, M. a, Rosen, B.R., Weisskoff, R.M., 1999. Evidence of a cerebrovascular postarteriole windkessel with delayed compliance. J. Cereb. Blood Flow Metab. 19, 679–689. doi:10.1097/00004647-199906000-00012

Mayhew, J., Hu, D., Zheng, Y., Askew, S., Hou, Y., Berwick, J., Coffey, P.J., Brown, N., 1998. An evaluation of linear model analysis techniques for processing images of microcirculation activity. Neuroimage 7, 49–71. doi:10.1006/nimg.1997.0311

Miller, K. L., Luh, W. M., Liu, T. T., Martinez, A., Obata, T., Wong, E. C., ... & Buxton, R.B., 2000. Characterizing the dynamic perfusion response to stimuli of short duration, in: ISRM. p. Vol. 8, 580.

Pelli, D.G., 1997. The VideoToolbox software for visual psychophysics: transforming numbers into movies. Spat. Vis. 10, 437–42.

Quax, S.C., van Koppen, T.C., Jylänki, P., Dumoulin, S.O., van Gerven, M.A., 2016. Slice-sampled Bayesian PRF mapping. bioRxiv.

Rigoux, L., Daunizeau, J., 2015. Dynamic causal modelling of brain-behaviour relationships. Neuroimage 117, 202–221. doi:10.1016/j.neuroimage.2015.05.041

Rovamo, J., Virsu, V., 1979. An estimation and application of the human cortical magnification factor. Exp. Brain Res. 37. doi:10.1007/BF00236819

Sengupta, B., Friston, K.J., Penny, W.D., 2016. Gradient-based MCMC samplers for dynamic causal modelling. Neuroimage 125, 1107–1118.

Sengupta, B., Friston, K.J., Penny, W.D., 2015. Gradient-free MCMC methods for dynamic causal modelling. Neuroimage 112, 375–381.

Smith, A.T., Singh, K.D., Williams, A.L., Greenlee, M.W., 2001. Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. Cereb Cortex. 11, 1182–90.

Stephan, K.E., Weiskopf, N., Drysdale, P.M., Robinson, P.A., Friston, K.J., 2007. Comparing hemodynamic models with DCM. Neuroimage 38, 387–401.

Vats, D., Flegal, J.M., Jones, G.L., 2015. Multivariate Output Analysis for Markov chain Monte Carlo. arXiv Prepr. 1512.07713.

Victor, J.D., Purpura, ., Katz, E., Mao, B., 1994. Population encoding of spatial frequency, orientation, and color in macaque V1. J. Neurophysiol. 72, 2151–66.

Wandell, B.A., Dumoulin, S.O., Brewer, A.A., 2007. Visual Field Maps in Human Cortex. Neuron 56, 366–383. doi:10.1016/j.neuron.2007.10.012

Winawer, J., Horiguchi, H., Sayres, R.A., Amano, K., Wandell, B.A., 2010. Mapping hV4 and ventral occipital cortex: the venous eclipse. J Vis. 10, 1.

Zeidman, P., Silson, E.H., Schwarzkopf, D.S., Baker, C.I., Penny, W., 2016. Bayesian Population Receptive Field Modelling. arXiv Prepr.
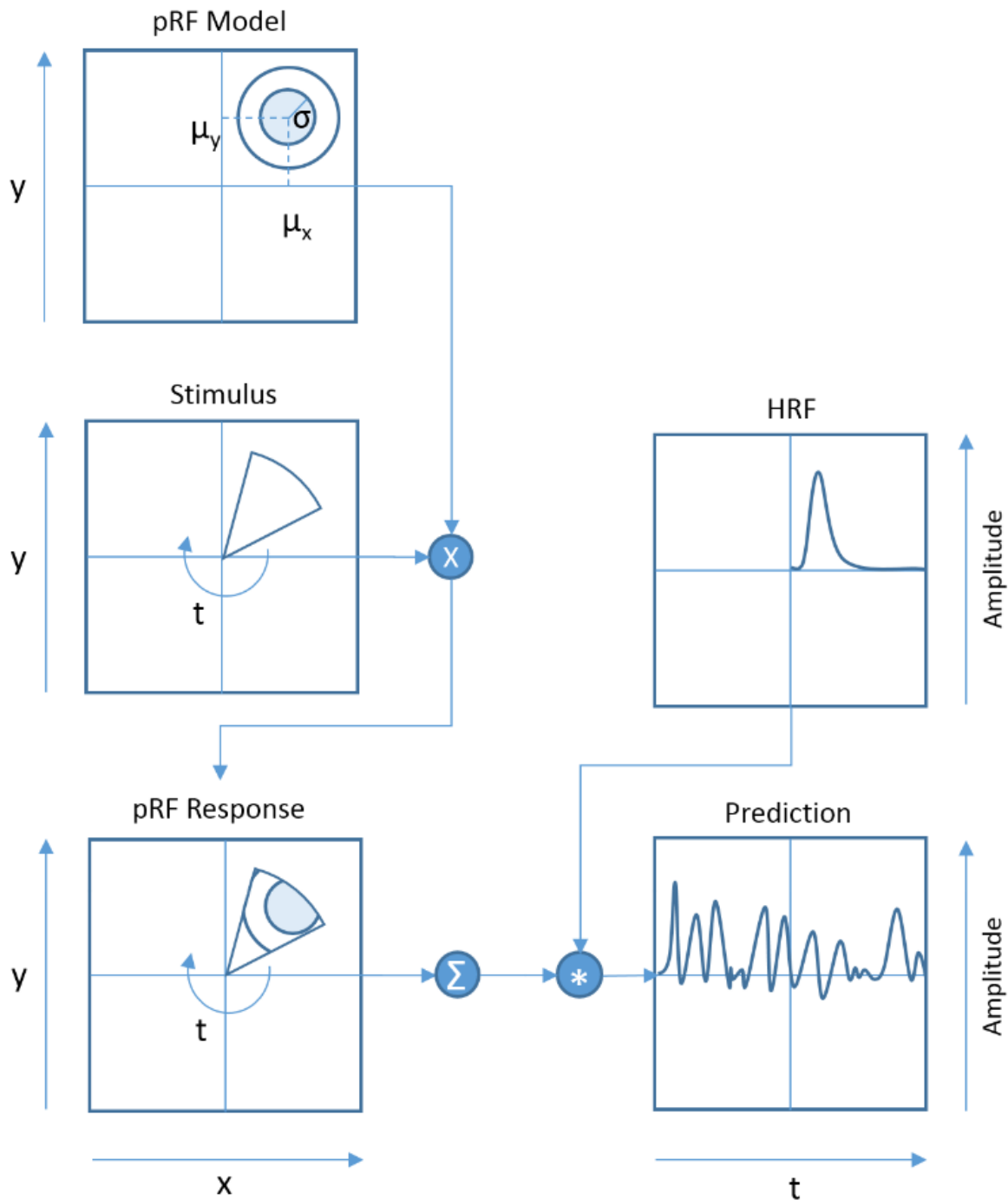
Figures

Figure 1. Dumoulin-Wandell pRF Model (1-column figure). Neuronal population receptive field is modelled as a two-dimensional Gaussian function (1st row) where (μ_x, μ_y) is the receptive field center and σ is the Gaussian standard deviation. Two-dimensional visual stimulus (2nd row, left) is multiplied

element-wise with the Gaussian (3rd row, left). Sum of all cells of the resulting matrix gives the pRF response. Convolution of pRF response with canonical HRF response (2nd row, right) gives the final output of the model (3rd row, right).
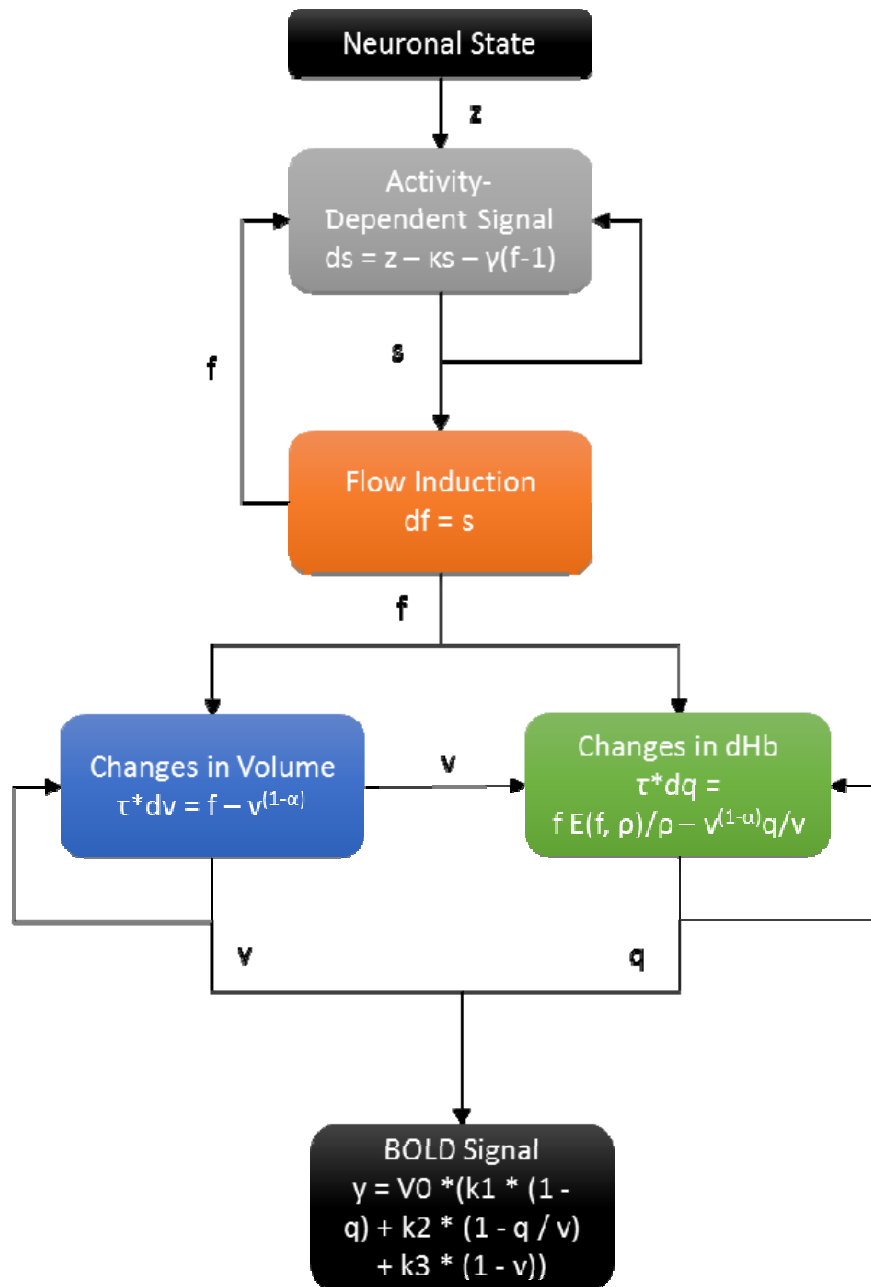
Figure 2. Balloon-Windkessel model (1-column figure). Relative change in BOLD signal (5th row) can be expressed as a nonlinear function of normalized venous volume (v), normalized total deoxyhemoglobin

content (q) and oxygen extraction fraction (E_0). Changes in volume and deoxyhemoglobin content are described with coupled differential equations (4th row) driven by cerebral blood flow (f). Flow induction is described by regional cerebral blood flow (rCBF) model (2nd and 3rd rows) triggered by neuronal activity (1st row).
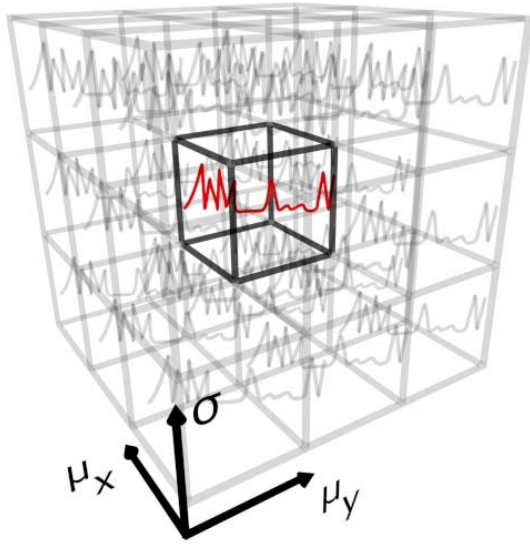
Figure 3. Premultiplied signal illustration (1-column figure). 2D visual stimuli is the same for all voxels so it can be premultiplied and summed over a wide range of Gaussians to create a lookup structure valid across the entire dataset. For arbitrary values of μ_x, μ_y, σ eight surrounding cells from the lookup table are picked and the resulting signal is approximated using trilinear interpolation. This scheme implicitly bounds the pRF model parameter space as the Gaussian parameters cannot exceed the bounds of the lookup structure.
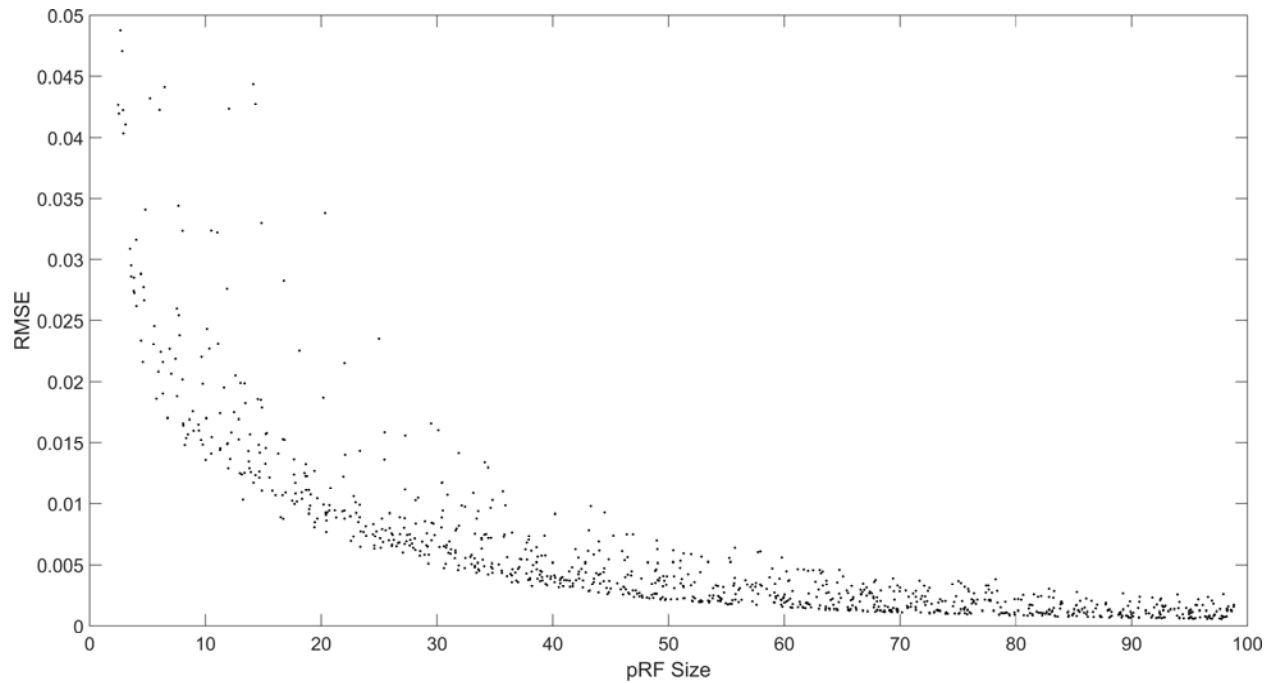
Figure 4 (1-column). Root mean squared error (RMSE) of the sum of the product of the stimulus and the Gaussians with different positions and sizes relative to maximum signal amplitude. The figure illustrates the relative RMSE error between the signal obtained from the lookup table and the signal generated using full evaluation of the model for 1000 Gaussians with randomly picked parameters. Mean RMSE relative to amplitude was equal to 2%. There were 22 points (2%) exceeding relative error of 3% with the maximum error below 5%.
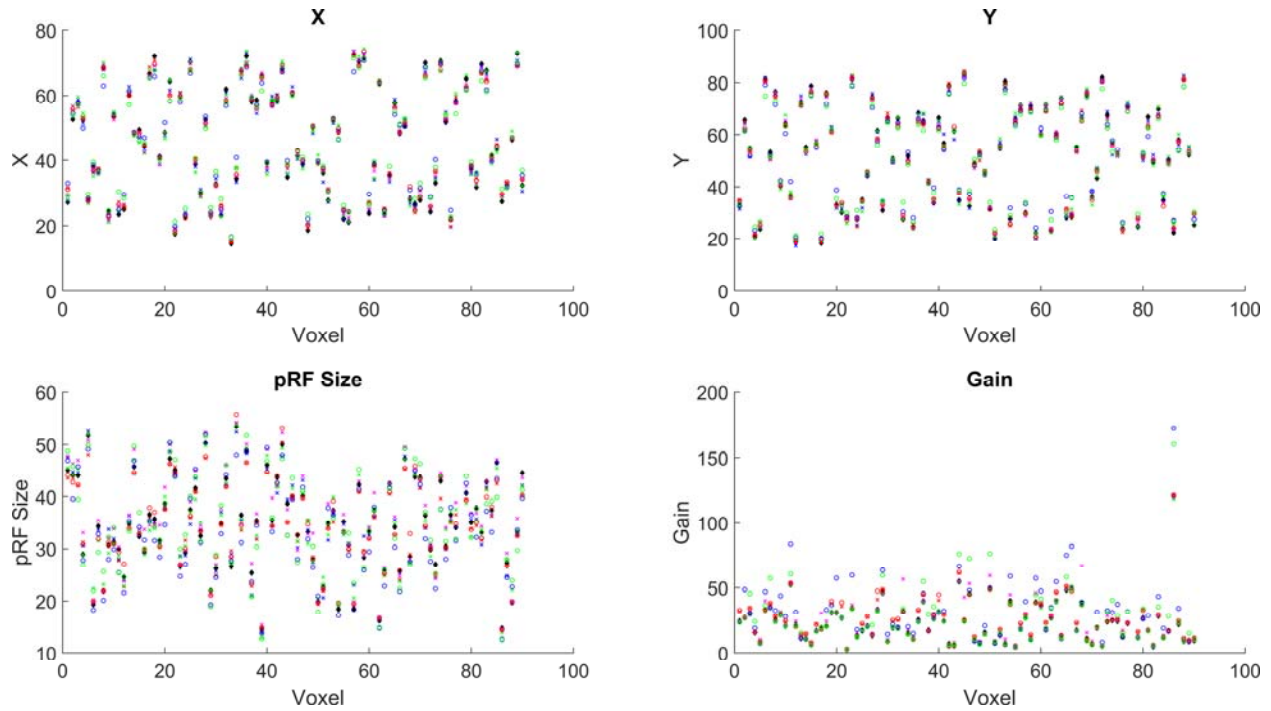
Figure 5 (2-column). Ground truth and parameter estimates using different models for simulated data. Black diamond – ground truth. Blue X – noLUT_CSS_5, red X – noLUT_CSS_3, green X – noLUT_DW_5, magenta X – noLUT_DW_3, blue circle – Q_CSS_5, red circle Q_CSS_3, green circle – Q_DW_5, magenta circle – Q_DW_3.
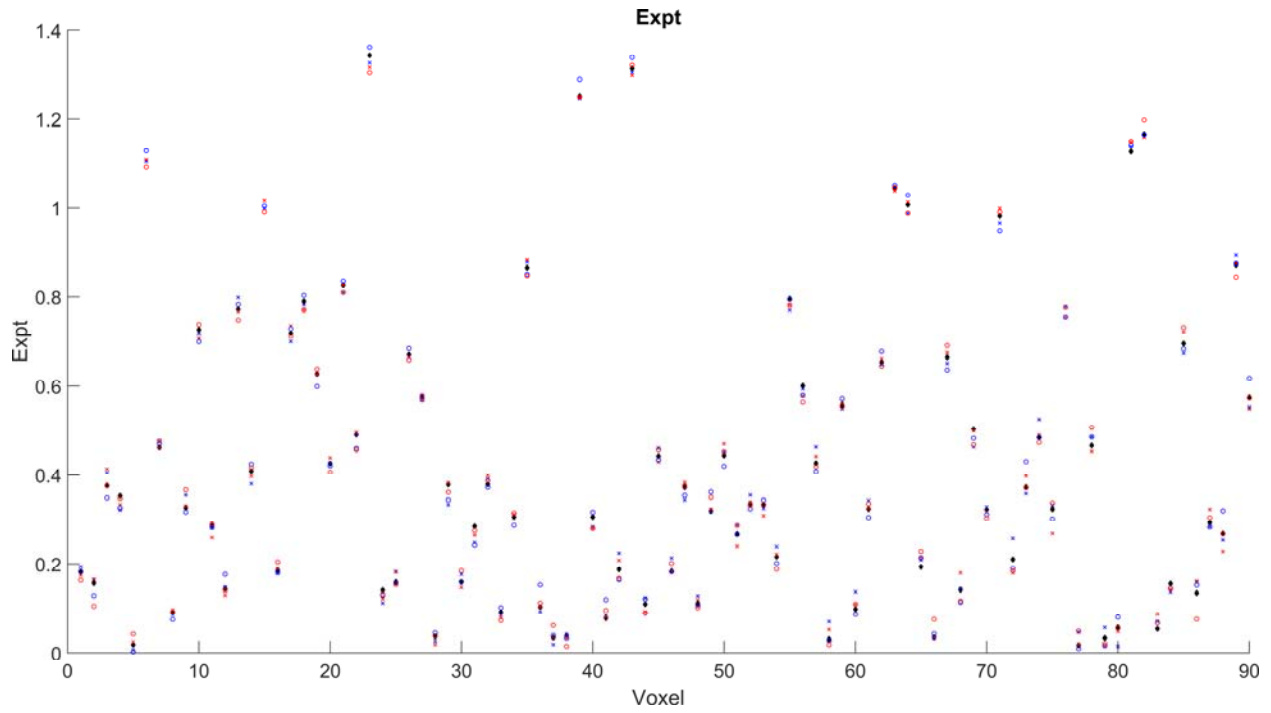
Figure 6 (1-column). Ground truth and exponent estimates using different models for simulated data. Black diamond – ground truth. Blue X – noLUT_CSS_5, red X – noLUT_CSS_3, blue circle – Q_CSS_5, red circle Q_CSS_3.
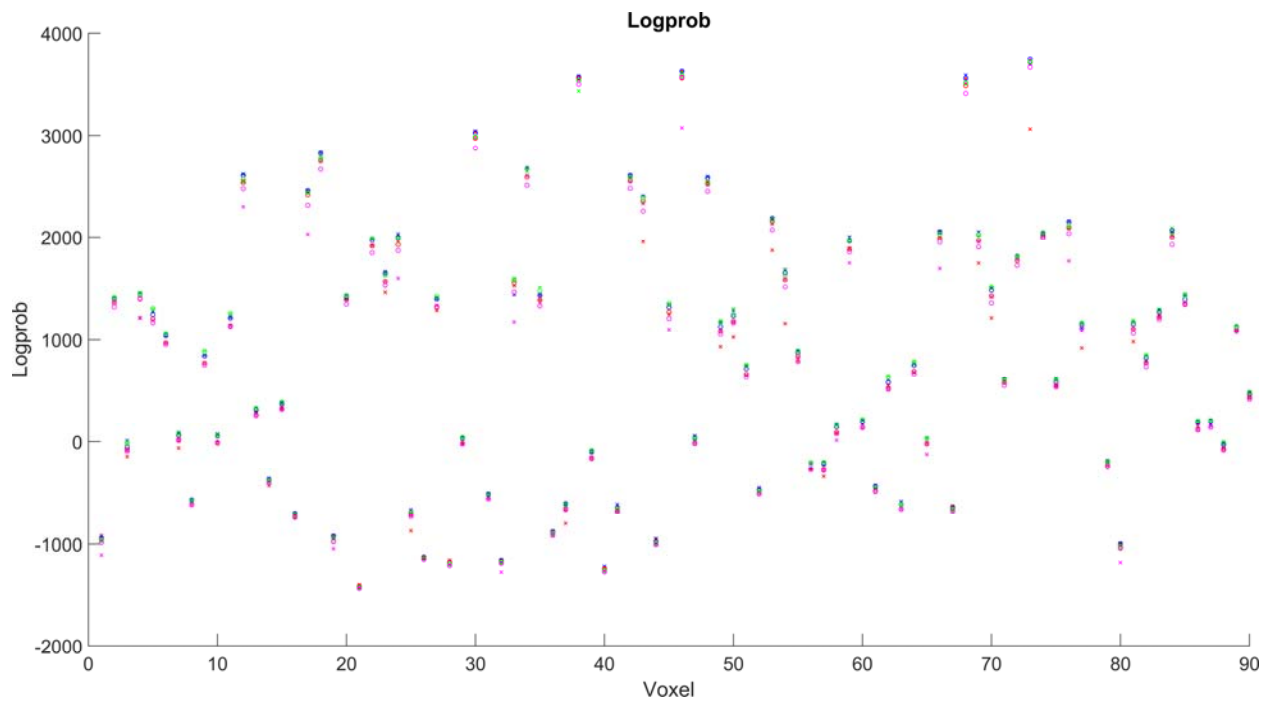
Figure 7 (1-column). Log probability comparison between models. Blue X – noLUT_CSS_5, red X – noLUT_CSS_3, green X – noLUT_DW_5, magenta X – noLUT_DW_3, blue circle – Q_CSS_5, red circle Q_CSS_3, green circle – Q_DW_5, magenta circle – Q_DW_3.
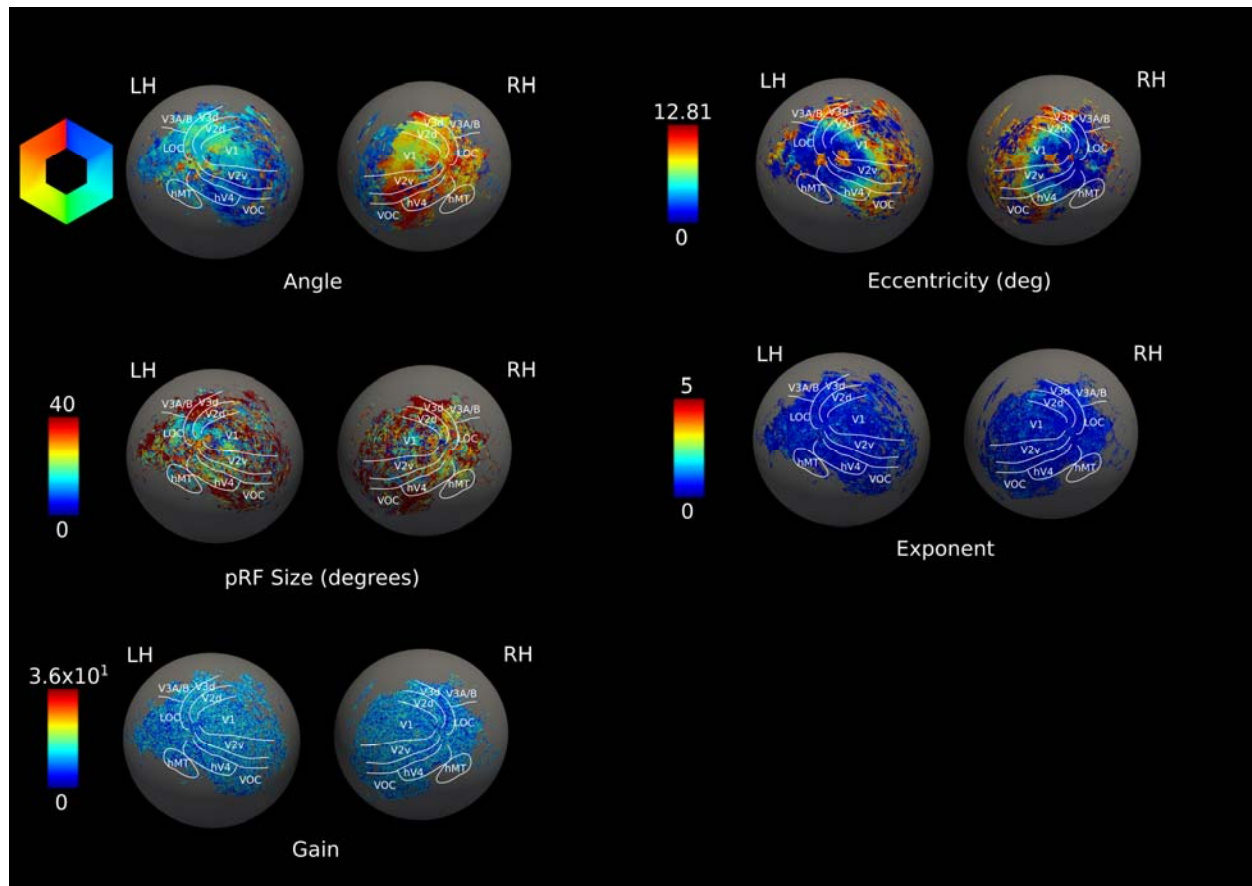

Figure 8 (2-column). MCMC Parameter estimates (two-column figure). The CSS-pRF parameters Θ, r, σ, expt and gain estimated using MCMC are reported on the inflated left/right hemispheres of a single subject – Subject #1. Two leftmost columns show the parameter means whereas the two rightmost columns show the standard deviations of respective parameters.