# Table of Contents



# Spontaneous and Task-related Activation of Neuronally Correlated Events (STANCE)

Models the 3D activation map from the first fMRI experiment on photic stimulation.

author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU) demo_3D_ex2.m updated 2 APR 2017

```
close all;
clear all;
currentDir = pwd;
if strcmp(currentDir(end-2:end),'GUI')
    % GUI instance of initialization
    cd ../
    STANCEroot = pwd;
    cd(currentDir)
elseif strcmp(currentDir(end-5:end),'STANCE')
    STANCEroot = pwd;
elseif strcmp(currentDir(end-16:end),'scripts_for_demos')
    cd ../
    STANCEroot = pwd;
else
    hSTANCE = msgbox('Please select the STANCE directory');
    uiwait(hSTANCE);
    currPath = fileparts(mfilename('fullpath'));
    STANCEroot = uigetdir(currPath, 'Add STANCE filepath');
end
cd(STANCEroot)
addpath(genpath(pwd));

% Load STANCE globals ...
if ~exist('STANCE.mat','file')
    STANCE_initialize_STANCE;
    load('STANCE.mat');
else
    load('STANCE.mat');
```

```
end
% NOTE: Must add SPM version to filepath prior to usage
addpath(SPMpath);
if exist(spm('Dir'),'dir')
    display('o SPM installation found.')
else
    warning('SPM installation not found. Please add to MATLAB filepath
 or install.')
    warning('SPM8 installation: http://www.fil.ion.ucl.ac.uk/spm/
software/spm8/')
    exit
end
```

```
o SPM installation found.
```

# Turn off warnings ...

... OpenGl warnings

```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
warning('off', 'MATLAB:hg:AutoSoftwareOpenGL');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warnings when loading SPM mat files
warning('off', 'MATLAB:unknownElementsNowStruc');
warning('off', 'MATLAB:dispatcher:ShadowedMEXExtension');
warning('off', 'MATLAB:pfileOlderThanMfile');
% ... removing files from path
warning('off', 'MATLAB:RMDIR:RemovedFromPath');
warning('off', 'MATLAB:DELETE:FileNotFound');
```

# Select subject by index (originally there are 20 subjects to choose from)

```
subject_brain = 1;
Now_sss = [2 1 1]; % study - subject - session ID
STANCE_new_session(2,1,1,true)
filepathOut = STANCE_genpath(Now_sss);
if ~logical(exist(filepathOut,'file'))
    STANCE_new_session(Now_sss);
end
makeFMRI = true;

% for reproducibility
s = 0;
%s = []; % allow MATLAB to spontaneously shuffle
if ~isempty(s)
    rng(s);
end

% show MNI volume conformed to BrainWEB dimensions
```

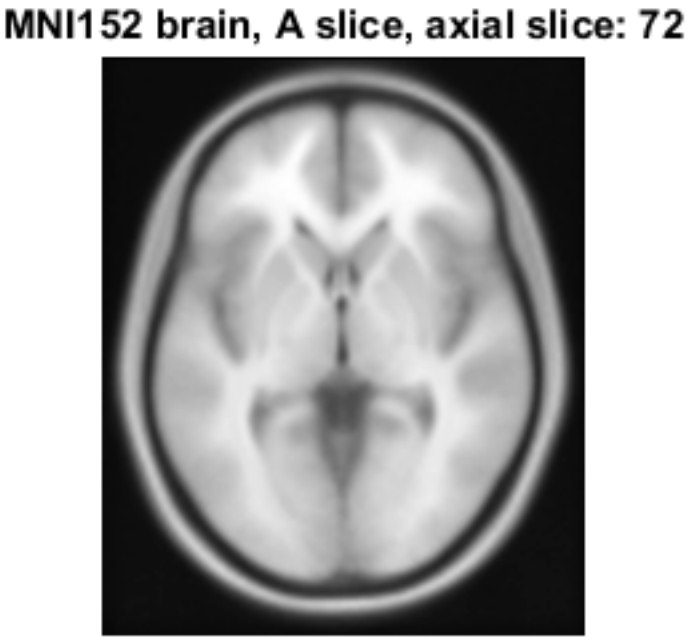
```
[V_MNI,Y_MNI] = STANCE_load_volume(filenameMNI);
display('The matrix dimensions of the MNI152 brain:')
MNI_dim = V_MNI.dim
display('The homographic matrix of the MNI152 brain:')
MNI_mat = V_MNI.mat
display('The origin (AC location) of the MNI152 brain:')
origin  = abs(V_MNI.mat(1:3,4))'

[~,I_max] = max(sum(sum(Y_MNI)));
showSlice = I_max(1);
f0 = figure;
subplot(2,1,2)
imshow(imrotate(Y_MNI(:,:,showSlice),90),[]), drawnow;
TITLE = ['MNI152 brain, A slice, axial slice: ',num2str(showSlice)];
title(TITLE)
truesize
movegui(f0,'northwest');

% load the T1w data for subject, for display purposes
[V_T1w,Y_T1w] = STANCE_choose_subject(subject_brain,'T1');

display('The matrix dimensions of the T1-w image from header:')
T1w_dim = V_T1w.dim  % dimensions of T1-w volume
display('The homographic matrix of the T1-w image from header:')
T1w_mat = V_T1w.mat  % 4x4 homographic matrix relating indeces to
 real-world coordinates
f1 = figure;
subplot(2,1,2)
imshow(imrotate(Y_T1w(:,:,showSlice),90),[]), drawnow;
TITLE = ['Subject T1-w brain, axial slice: ',num2str(showSlice)];
title(TITLE)
truesize
movegui(f1,'north');

% retrieve transformation matrix mapping MNI152 to subjects' native
 spaces
display('The transformation matrix mapping MNI152 to the native spaces
 of the subjects:');
M = M_array(:,:,subject_brain)

[V_MNI_reg,Y_MNI_reg] = STANCE_register_MNI(V_T1w.fname,M);

% figure, imshow(imrotate(Y_MNI_reg(:,:,showSlice),90),[]), drawnow;
% TITLE = ['MNI152 registered to subject brain, A slice:
 ',num2str(showSlice)];
% title(TITLE)

display('The matrix dimensions of the T1-w image:')
dimensions = size(Y_T1w)
display('The origin (AC location) of the T1-w image:')
origin = round(abs(V_T1w.mat(1:3,4)))'


ans =
```

```
     1     1   520

The matrix dimensions of the MNI152 brain:

MNI_dim =

   181   217   181

The homographic matrix of the MNI152 brain:

MNI_mat =

     1     0     0   -91
     0     1     0  -127
     0     0     1   -73
     0     0     0     1

The origin (AC location) of the MNI152 brain:

origin =

    91   127    73

The matrix dimensions of the T1-w image from header:

T1w_dim =

   181   217   181

The homographic matrix of the T1-w image from header:

T1w_mat =

    1.0000         0         0  -90.7500
         0    1.0000         0 -126.7500
         0         0    1.0000  -73.2500
         0         0         0    1.0000

The transformation matrix mapping MNI152 to the native spaces of the
 subjects:

M =

    1.0000   -0.0081    0.0044    0.9122
    0.0081    1.0000   -0.0027    0.0241
   -0.0044    0.0028    1.0000    0.2833
         0         0         0    1.0000

The matrix dimensions of the T1-w image:

dimensions =

   181   217   181
```

*The origin (AC location) of the T1-w image:*

*origin =*

*91 127 73*




# Build activation regions

```
% see Table 1. of "Functional mapping of the human cortex by magnetic
% resonance imaging" in SCIENCE 254(5032):716-9 NOVEMBER 1991

% define Photic Visual Cortex stimulation by optimally pulsed light
task.name = 'PVC photic stim';
task.activation(1).region     = 'L PVC';
task.activation(1).volume     = 6000;        % estimated from reported
 slice of 600 mm^2 -> 18^2 ~ 300 mm^2; 18^3 ~ 6000 mm^3
task.activation(1).center     = [9,-84, 10]; % left side of activation
task.activation(1).rotation   = [30,0,0];    % degrees
task.activation(1).shape      = 'sphere';
task.activation(1).proportion = [3,8,4];     % aspect ratio
task.activation(1).falloff    = 0.005;       % parameterizes
 exponential falloff about center, in [0,1]
task.activation(1).minimum    = 0.2;         % parameterizes
 exponential falloff minimum value in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(2) = task.activation(1);   % right side of activation
task.activation(2).region   = 'R PVC';
task.activation(2).center   = [-10,-80, 10];
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% define signal amplitude
task.amplitude = 0.03; % 3% activation
```

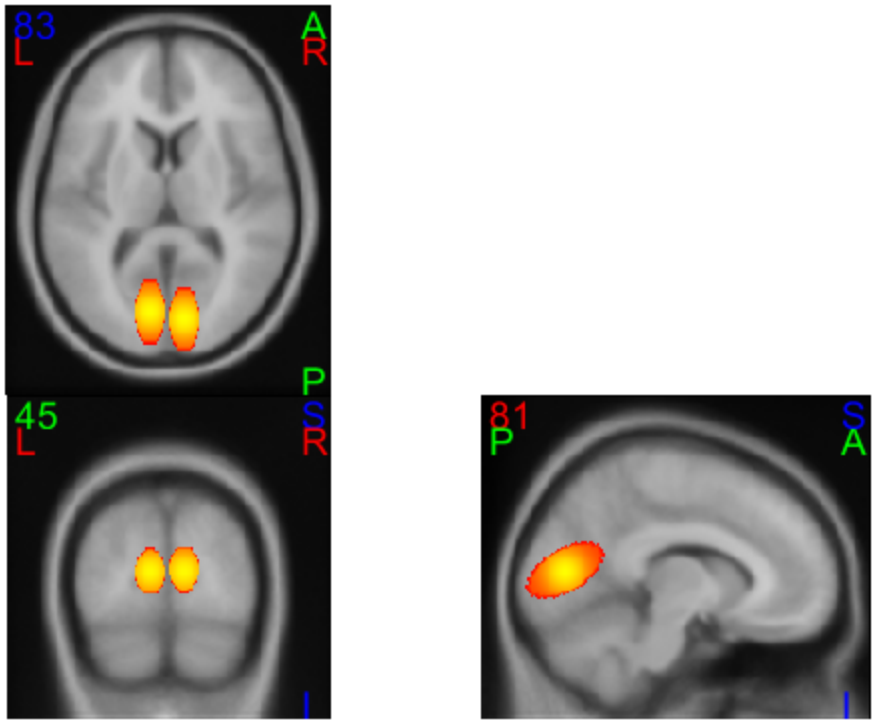
```
if exist('simulations.mat','file')
    load('simulations.mat');
else
    % create simulations struct
end
simulations{Now_sss(1)}.name = 'First fMRI study';
simulations{Now_sss(1)}.task{1} = task;

% left component
disp('Defining left component of Photic stimulation activation
 map...')
task.activation(1).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(1));
% right component
disp('Defining right component of Photic stimulation activation
 map...')
task.activation(2).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(2));
% combine different component with fuzzy logic OR
disp('Combining task components.')
Nactivations = length(task.activation);
% fuzzy logical OR operation:
%task.map = max(task.activation(:).map);
% alternative:
task.map = STANCE_combine_maps('OR',task.activation(:).map);

% free up working memory (optional)
% task.activation(1).map = [];
% task.activation(2).map = [];

% find MNI gray matter volume of activation map
task.GMvolume = STANCE_find_GM_volume(task);

[~,I_max] = max(sum(sum(task.map)));
showSliceTA = I_max(1);

% figure, imshow(imrotate(task.map(:,:,showSliceTA),90),[]), drawnow;
% TITLE = ['Photic Stimulation of the Visual Cortex, A slice:
 ',num2str(showSliceTA)];
% title(TITLE)

TITLE = {'Photic Stimulation of the Visual Cortex'; 'activation
 template in MNI'};
h_task = STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
title(TITLE)
movegui(h_task,'center');

% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
% title('Photic Stimulation of the Visual Cortex: R sagittal')
%
```

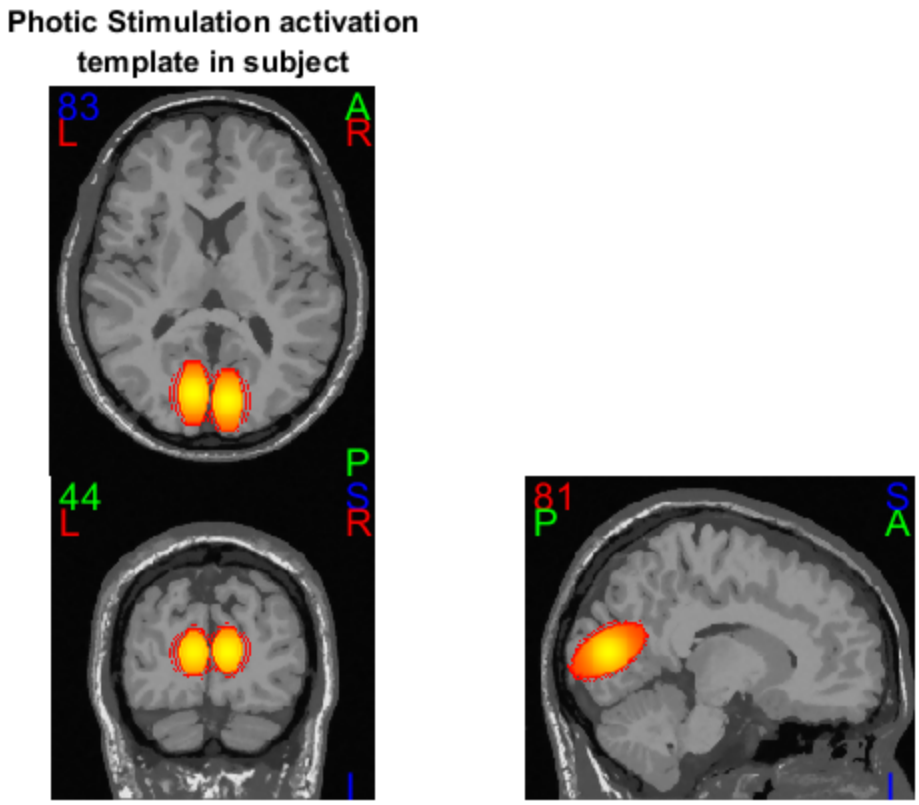
```
% h_task_TS_L = STANCE_display_activation_slice(Y_MNI,task.map,181-
showSliceTS,1);
% title('Photic Stimulation of the Visual Cortex: L sagittal')
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
% title('Photic Stimulation of the Visual Cortex: coronal')

clear('V_MNI','Y_MNI')

[V_PVC_photic_stim_reg,Y_PVC_photic_stim_reg] =
 STANCE_register_activation(V_T1w.fname,task);

% figure, imshow(imrotate(Y_PVC_photic_stim_reg(:,:,showSliceTA),90),
[]), drawnow;
% title('Photic Stimulation of the Visual Cortex registered')

% h_task_reg =
 STANCE_display_activation_slice(Y_MNI_reg,Y_PVC_photic_stim_reg,[]);
% title('Photic Stimulation activation template', 'registered to
 native space')
% title(TITLE)
% movegui(h_task_reg,'center');

% Photic Stimulation of the Visual Cortex of subject
h_task_sub =
 STANCE_display_activation_slice(Y_T1w,Y_PVC_photic_stim_reg,[],[]);
title({'Photic Stimulation activation', 'template in subject'})
movegui(h_task_sub,'center');

% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
%
% h_task_subTS_R =
 STANCE_display_activation_slice(Y_T1w,Y_PVC_photic_stim_reg,showSliceTS,1);
% title('Photic Stimulation of the Visual Cortex: R sagittal')
%
% h_task_subTS_L =
 STANCE_display_activation_slice(Y_T1w,Y_PVC_photic_stim_reg,181-
showSliceTS,1);
% title('Photic Stimulation of the Visual Cortex: L sagittal')
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
%
% h_task_subTC =
 STANCE_display_activation_slice(Y_T1w,Y_PVC_photic_stim_reg,showSliceTC,2);
% title('Photic Stimulation of the Visual Cortex: coronal')

% make room in memory
if ~strcmp(V_T1w.fname(end-1:end),'gz')
    delete(V_T1w.fname);
```

```
end
clear('V_T1w','YT1w');
delete(V_MNI_reg.fname);
clear('V_MNI_reg','Y_MNI_reg')
task.map = [];

% save activation template
task.activation(1).map = int8(255*task.activation(1).map);
task.activation(2).map = int8(255*task.activation(2).map);
cd([STANCEroot,'/activations'])
save([task.name,'.mat'],'task')
cd(STANCEroot)
% free up working memory (optional)
task.activation(1).map = [];
task.activation(2).map = [];
```

*Defining left component of Photic stimulation activation map...*
*o Specifying spherical template*
*o Performing affine transformation (this may take a while).*
*o Building activation map.*
*Defining right component of Photic stimulation activation map...*
*o Specifying spherical template*
*o Performing affine transformation (this may take a while).*
*o Building activation map.*
*Combining task components.*

**Photic Stimulation of the Visual Cortex**
**activation template in MNI**






# Reslice volumes to functional space according to fMRI scan protocol

```
if makeFMRI == true

% use default scan settings:
%     voxelSize    = [3 3 3];
%     new_dims     = [64 64 NaN];  % effectively [64 64 40];
%     tiltAngle    = 15; % degrees
%     voxelSpacing = [0 0 0.6];
%     sumThreshold = 100;
%     TR = 2000 ms; % (whole volume)
%     TE = 30 ms;
% define scan protocol parameters (these are the default values)
scan.voxel.size    = [3 3 3];
scan.voxel.matrix  = [64 64 NaN];  % [64 64 40];
scan.voxel.spacing    = [0 0 0.2*scan.voxel.size(3)]; % assume 20% Z
 spacing
scan.tiltAngle        = 15; % degrees
scan.TR               = 2000; % [ms]
scan.TE               = 30;   % [ms]
scan.ES               = 0.51; % [ms] echo spacing
scan.FA               = 78;   % degrees
scan.BW               = 2232; % [Hz/Px]
scan.order            = 'SD'; % SD = sequential descending order
```

```
scan.KM0              = 2225; % fit to data with max of 909 at 3T and
 FA = 90 degree
scan.noise_method     = 'percent';
scan.noise            = 0;    % percent noise relative to peak
scan.attenuation      = 0;    % coil attenuation factor ~mm^-1

simulations{Now_sss(1)}.scan = scan;
save('simulations.mat','simulations')

% load tissue fuzzy memberships in subject's native space
[V_fuzzy,~] = STANCE_choose_subject(subject_brain,'fuzzy',true);
fn_tissue = [V_fuzzy(1).fname,'.gz'];

% generate the tissue fuzzy memberships in functional space
[V_reslice,Y_reslice] = STANCE_reslice_tissue(fn_tissue,scan,[],
[],false,Now_sss); % change last to 'true' to show figure
sliceLimits =
 [V_reslice(1).sliceLimitLower,V_reslice(1).sliceLimitUpper];
[~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
showSlice2 = I_max(1);

scrsz = get(groot,'ScreenSize');
positionVector2 = [scrsz(3)/2.5 scrsz(4)/2.5 scrsz(3)/5 scrsz(4)/3];
f3 = figure;
imshow(imrotate(Y_reslice(:,:,showSlice2,3),90),[]);
TITLE = {'Gray matter priors,', ['functional axial slice:
 ',num2str(showSlice2)]};
title(TITLE)
set(f3,'OuterPosition',positionVector2);
movegui(f3,'northeast')
fn_fuzzy_reslice = V_reslice(1).fname;

% generate T2* baseline map in functional space
[V_T2star_Map,Y_T2star_Map] =
 STANCE_make_parameter_map(fn_fuzzy_reslice,'T2star');
f4 = figure;
imshow(imrotate(Y_T2star_Map(:,:,showSlice2),90),[]);
TITLE = {'T2* baseline volume,', ['axial slice:
 ',num2str(showSlice2)]};
title(TITLE)
set(f4,'OuterPosition',positionVector2);
movegui(f4,'east')

% project activation map onto functional space
[V_PVC_photic_stim_reslice,Y_PVC_photic_stim_reslice] =
 STANCE_reslice_volume(V_PVC_photic_stim_reg,scan,sliceLimits);
[~,I_max] = max(sum(sum(Y_PVC_photic_stim_reslice)));
showSlice2TA = I_max(1);
TITLE = {'Photic stimulation of PVC,'; ['functional axial:
 ',num2str(showSlice2TA)]};
f5 = figure;
imshow(imrotate(Y_PVC_photic_stim_reslice(:,:,showSlice2TA),90),[]);
title(TITLE)
set(f5,'OuterPosition',positionVector2);
```

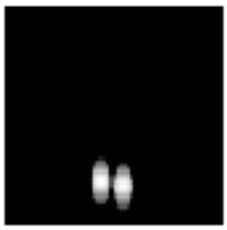
```
movegui(f5,'southeast')

delete(V_PVC_photic_stim_reg.fname);

% mask with gray matter mask
[Y_PVC_photic_stim_reslice,Y_GM] =
 STANCE_GM_mask(Y_PVC_photic_stim_reslice,task.GMvolume,Now_sss);
%figure,
 imshow(imrotate(Y_PVC_photic_stim_reslice(:,:,showSlice2TA),90),[]);

subjectActivation3D(1,:,:,:) = Y_PVC_photic_stim_reslice;

cd(STANCE_genpath([],2))
save('STANCEsubject.mat','Now_sss','subject_brain','fn_tissue','fn_fuzzy_reslice',
cd(STANCEroot)

% add activation to T2* baseline
[V_T2star_Map_Act,Y_T2star_Map_Act] =
 STANCE_add_activation(V_T2star_Map.fname,Y_PVC_photic_stim_reslice,scan.TE,task.a

f6 = figure;
imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90),[]);
TITLE = {'T2* map w/ BOLD activation,', ['axial slice:
 ',num2str(showSlice2TA)]};
title(TITLE)
set(f6,'OuterPosition',positionVector2);
movegui(f6,'south')

f7 = figure;
imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90)-
imrotate(Y_T2star_Map(:,:,showSlice2TA),90),[]);
TITLE = {'(T2* activation - baseline),', ['axial slice:
 ',num2str(showSlice2TA)]};
title(TITLE)
set(f7,'OuterPosition',positionVector2);
movegui(f7,'southwest')

fn_reslice = V_reslice(1).fname;
% exact EPI signal, no noise, no attenuation
[V_EPI0,Y_EPI0] = STANCE_EPI_signal(fn_reslice,Y_T2star_Map,scan);
maxS = max(Y_EPI0(:).*Y_GM(:));
f8 = figure;
imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[0,maxS]);
TITLE = {'Baseline signal volume,', ['axial slice:
 ',num2str(showSlice2TA)]};
title(TITLE)
set(f8,'OuterPosition',positionVector2);
movegui(f8,'west')

Reslicing gray matter fuzzy membership labels...
Reslicing background fuzzy membership labels...
Reslicing CSF fuzzy membership labels...
Reslicing white matter fuzzy membership labels...
Reslicing fat fuzzy membership labels...
```

*Reslicing muscle fuzzy membership labels...*
*Reslicing skin fuzzy membership labels...*
*Reslicing skull fuzzy membership labels...*
*Reslicing blood vessels fuzzy membership labels...*
*Reslicing connective tissue fuzzy membership labels...*
*Reslicing dura matter fuzzy membership labels...*
*Reslicing bone marrow fuzzy membership labels...*
*o Writing C:\spm\STANCE/fMRI/study002/subject0001/session001\EPI_BOLD_0001_001.nii*
*The maximum intensity of the simulated signal: 1087.8398*

**Gray matter priors, functional axial slice: 13**

**T2* baseline volume, axial slice: 13**

**Photic stimulation of PVC, functional axial: 24**



T2* map w/ BOLD activation,
axial slice: 24

(T2* activation - baseline),
axial slice: 24

Baseline signal volume,
axial slice: 24

# Construct pristine EPI signal

exact EPI signal, no noise, no attenuation

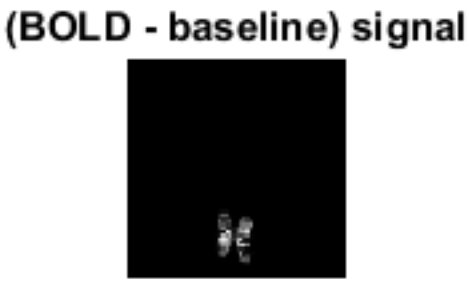
```
[V_EPI,Y_EPI] = STANCE_EPI_signal(fn_reslice,Y_T2star_Map_Act,scan);
maxS = max(Y_EPI(:).*Y_GM(:));
f9 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90),[0,maxS]);
TITLE = {'Exact BOLD signal,',['axial slice:
 ',num2str(showSlice2TA)]};
```

```
title(TITLE)
set(f9,'OuterPosition',positionVector2);
movegui(f9,'northwest')

f10 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90)-
imrotate(Y_EPI0(:,:,showSlice2TA),90),[]);
title('(BOLD - baseline) signal')
set(f10,'OuterPosition',positionVector2);
movegui(f10,'north')

% Photic Stimulation of the Visual Cortex of subject
htasksubfun =
 STANCE_display_activation_slice(Y_EPI,Y_PVC_photic_stim_reslice,[],
[]);
title({'Photic stimulation of PVC', 'masked with gray matter'})
movegui(htasksubfun,'center');

% approximated EPI signal, no noise, no attenuation
[V_EPIapp,Y_EPIapp] =
 STANCE_EPI_signal(fn_reslice,Y_T2star_Map_Act,scan,[],[],[],true);
display('The max intensity of approximated EPI signal:');
maxSapp = max(Y_EPIapp(:).*Y_GM(:))
f11 = figure;
imshow(imrotate(Y_EPIapp(:,:,showSlice2TA),90),[0,maxS]);
title('Approximated BOLD signal')
set(f11,'OuterPosition',positionVector2);
movegui(f11,'northeast')

f12 = figure;
imshow(imrotate(Y_EPIapp(:,:,showSlice2TA),90)-
imrotate(Y_EPI(:,:,showSlice2TA),90),[-50,50]);
title({'(Approximated - exact)',' BOLD signal'})
set(f12,'OuterPosition',positionVector2);
movegui(f12,'east')

display('The MSE between approximated and exact EPI signals:');
EPIerr = immse(Y_EPI,Y_EPIapp)/(maxS*maxSapp*64*64*40)
% no correction:   6.8659e+09
% CSF correction:  6.8760e+09
% WM  correction:  6.8646e+09
% CSF & WM correction: 6.8746e+09

o Writing C:\spm\STANCE/fMRI/study002/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 1090.4507
o Writing C:\spm\STANCE/fMRI/study002/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 2025.3179
The max intensity of approximated EPI signal:

maxSapp =

   1.0588e+03
```

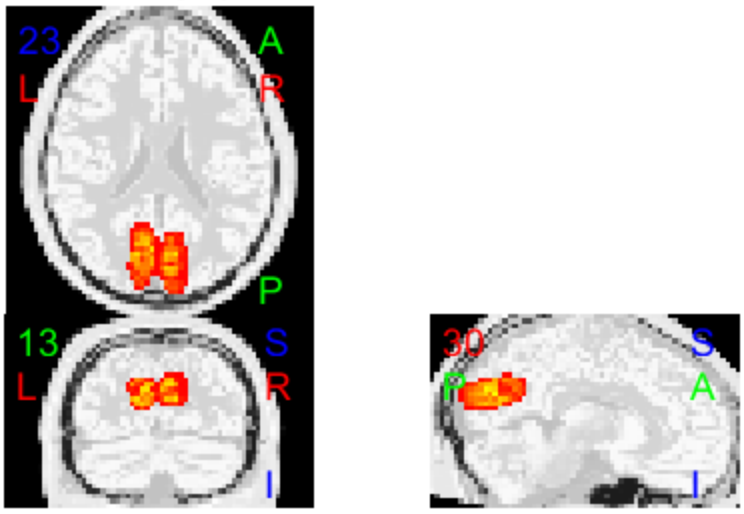
*The MSE between approximated and exact EPI signals:*

*EPIerr =*

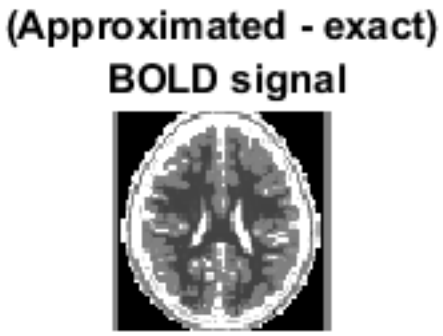
*2.2622e+09*

**Photic stimulation of PVC masked with gray matter**




**Approximated BOLD signal**

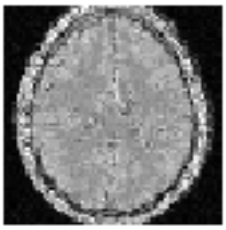



# Construct EPI signal with system noise

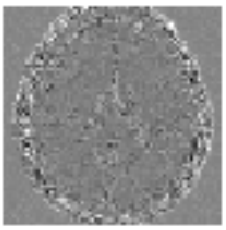
```
scan.noise            = 4;    % percent noise relative to peak
noiseMap = STANCE_make_noise_map(fn_reslice,2,4);
% exact EPI signal, no noise, no attenuation
[V_EPI_p4,Y_EPI_p4] =
 STANCE_EPI_signal(fn_reslice,Y_T2star_Map_Act,scan,noiseMap,[],[],[],
[],s);
maxS = max(Y_EPI_p4(:).*Y_GM(:));
f13 = figure;
imshow(imrotate(Y_EPI_p4(:,:,showSlice2TA),90),[0,maxS]);
title('BOLD signal w/ 4% noise')
set(f13,'OuterPosition',positionVector2);
movegui(f13,'southeast')

f14 = figure;
imshow(imrotate(Y_EPI_p4(:,:,showSlice2TA),90)-
imrotate(Y_EPI(:,:,showSlice2TA),90),[]);
title('(noisy - exact) BOLD signal')
set(f14,'OuterPosition',positionVector2);
movegui(f14,'south')
```

```
o Writing C:\spm\STANCE/fMRI/study002/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 2231.3929
```

BOLD signal w/ 4% noise



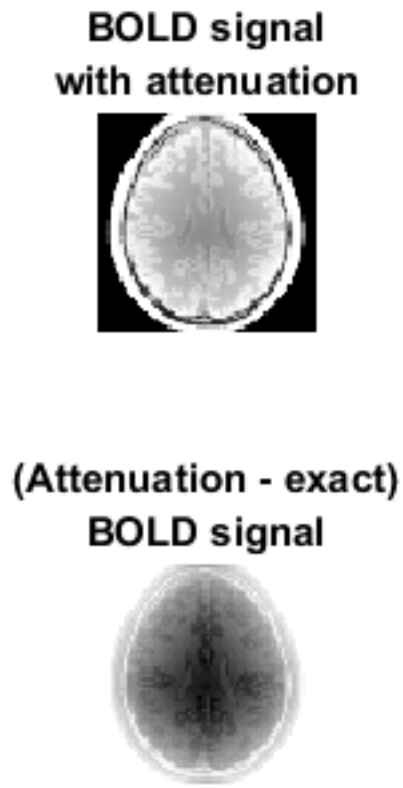
(noisy - exact) BOLD signal



# Construct EPI with attenuation

```
scan.noise            = 0;    % percent noise relative to peak
scan.attenuation      = 25;   % coil attenuation factor ~1/(3mm)
% exact EPI signal, no noise, no attenuation
[V_EPI_att,Y_EPI_att] =
 STANCE_EPI_signal(fn_reslice,Y_T2star_Map_Act,scan);
maxS = max(Y_EPI_att(:).*Y_GM(:));

f15 = figure;
imshow(imrotate(Y_EPI_att(:,:,showSlice2TA),90),[0,maxS]);
title({'BOLD signal', 'with attenuation'})
set(f15,'OuterPosition',positionVector2);
movegui(f15,'southwest')

f16 = figure;
imshow(imrotate(Y_EPI_att(:,:,showSlice2TA),90)-
imrotate(Y_EPI(:,:,showSlice2TA),90),[]);
title({'(Attenuation - exact)','BOLD signal'})
set(f16,'OuterPosition',positionVector2);
movegui(f16,'west')

o Writing C:\spm\STANCE/fMRI/study002/subject0001/
session001\EPI_BOLD_0001_001.nii
```

*The maximum intensity of the simulated signal: 1037.1221*




# Save results, free up memory, and return

save details of scan

```
cd(STANCE_genpath)
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subjectActivation3D
cd(STANCEroot)

else
    save('simulations.mat','simulations')
end
cd(currentDir)
```

*Published with MATLAB® R2016a*