NFT
Neuroelectromagnetic Forward Head Modeling Toolbox

Zeynep AKALIN ACAR
15th EEGLAB Workshop, Beijing
June, 2012
A complete framework for accurate forward problem solution.

Easy-to-use MATLAB environment with GUI and command-line functions.

Ability to use available subject information

– T1-weighted 3D MR images
– Digitized sensor (electrode) locations
Comparison with Dipfit

◆ The realistic model in Dipfit is a three-layer MNI head model represented with 3000 vertices.
  – The forward matrices are pre-calculated, so there is no need for FP calculations.

◆ NFT generates subject-specific models.
  – NFT does model generation and forward problem calculations.
  – More accurate.
T1-weighted

http://sccn.ucsd.edu/nft

Digitizer locations

http://sccn.ucsd.edu/nft
Head modeling from MR images

From a magnetic Resonance Image
- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration

MR image

T1-weighted

Segmentation

Electrode Registration

BEM mesh
Preparing the MR Image

- Using FreeSurfer
  - Inhomogeneity correction
  - Convert to 1x1x1 volume
  - Arrange direction of the image
  - Save in analyze format
Image Segmentation

Filter the image
(Curvature Anisotropic Filtering)

From a magnetic Resonance Image

- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration

Classifies four tissues from T1-weighted images
Scalp, Skull, CSF and Brain
Starting NFT

- To start from EEGLAB
  EEGLAB -> Tools -> NFT
- To start as a standalone toolbox
  addpath NFT directory
  Type ‘NFT’ in Matlab
  For demo: go to NFT-2.4_demo folder
Subject Selection

- Select subject folder
- Specify subject name
- Specify session name
Subject Selection

Select current folder as subject folder
Enter “SubjectA” as subject name
Enter “s1” as the session name
Image Segmentation
Image Segmentation

NFT: MR segmentation

1. Anisotropic Filtering
   - Swap LIR
   - Check inhomogeneity
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerbellar low point: 66
   - White matter seed point: 135
   - Fill level: 0.4
   - Threshold: 0.4

4. Outer Skull Segmentation
   - Center of one eye: 110

5. Inner Skull Segmentation

Load image
Segmentation

Select an image in analyze format
Segmentation

Run filtering
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 66
   - White matter seed point: 135
   - Fill level: 0.4
   - Threshold: 0.4

4. Outer Skull Segmentation
   - Center of one eye: 110

5. Inner Skull Segmentation

Display Image:
- MR image
- Filtered image
- Scalp mask
- Brain mask
- Outer skull mask
- Inner skull mask

Save Results:
- Output Folder: /data/projects/seynep/common/home/zevmei/co/deneme
- Filtered Image
- Segmentation

Image is filtered!
Segmentation

View filtered image

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 66
   - White matter seed point: 135
   - Fill level: 0.4
   - Threshold: 0.4

4. Outer Skull Segmentation
   - Center of one eye: 110

5. Inner Skull Segmentation

Save Results
- Output Folder: /data/projects/zeypem.comm/home/zeypem/comdeneme
  - Filtered Image
  - Segmentation

NFT: MR segmentation
Segmentation

Click ‘Next’ for scalp segmentation.
Segmentation

Click ‘Run’ for scalp segmentation
Image Segmentation
Segmentation

View scalp mask

NFT: MR segmentation

(x,y,z) = (128, 128, 128)

Coronal view

Axial view

Sagittal view

Image Segmentation

1. Anisotropic Filtering
   - Check inhomogeneity
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 66
   - White matter seed point: (135, 135)
   - Fill level: 0.4
   - Threshold: 0.4

4. Outer Skull Segmentation
   - Center of one eye: 110

5. Inner Skull Segmentation

Display Image

- MR image
- Filtered image
- Scalp mask
- Brain mask
- Outer skull mask
- Inner skull mask

Save Results

Output Folder: /data/projects/zevneq/common/home/zevneq/ord/cr/segmentation

Run

Scalp segmented!
Segmentation

Click ‘Next’ for brain segmentation
Segmentation

Selection of cerebellar low point
Segmentation

Click 'Set'
Segmentation

Selection of a WM point
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 67, y = 172, z = 158
   - White matter seed point: x = 158, y = 150
   - Fill level: f = 0.4
   - Threshold: T = 0.4

4. Outer Skull Segmentation
   - Center of one eye: z = 110

5. Inner Skull Segmentation

Options:
- Swap L/R
- Check inhomogeneity
- Preview
- Run
- Next
Segmentation

Click ‘Run’ for brain segmentation
Segmentation

Change thresholds if there is need.
Segmentation

View brain mask

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x=172, y=158, z=150
   - White matter seed point: x=158, y=150
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: z=110

5. Inner Skull Segmentation
   - Brain segmented!
Segmentation

Click ‘Next’ for skull segmentation
Segmentation

Select a slice for eyes
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: (x, y, z) = (150, 67, 95)
   - White matter seed point: (172, 158, 150)
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: (95, Set)

5. Inner Skull Segmentation

Save Results

- Output Folder: /data/projects/zyneq/comm_on/home_zeveno/cdeneme
- Filtered Image
- Segmentation

Click ‘Set’
Segmentation

Click ‘Run’ for skull segmentation
Segmentation

Click on the eyes
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Check inhomogeneity
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: 67
   - White matter seed point: 158
   - Fill level: 0.1
   - Threshold: 0.1

4. Outer Skull Segmentation
   - Center of one eye: 95

5. Inner Skull Segmentation
   - Segmenting skull...
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: (x, y, z) = (172, 158, 150)
   - White matter seed point: (x, y, z)
   - Fill level: [0, 1]
   - Threshold: [0, 1]

4. Outer Skull Segmentation
   - Center of one eye: (x, y, z) = (95)

5. Inner Skull Segmentation

Save Results:
- Output Folder
- Filtered Image
- Segmentation

Skull segmented!
Segmentation

View skull segmentation
Segmentation

Click ‘Next’ for CSF segmentation.
Segmentation

Click ‘Run’ for CSF segmentation
Segmentation
Segmentation

View CSF segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations
   - Image diffusion

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point
   - White matter seed point
   - Fill level
   - Threshold

4. Outer Skull Segmentation
   - Center of one eye

5. Inner Skull Segmentation
   - Segmentation complete
Segmentation

Save filtered image
Segmentation
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x = 172, y = 158, z = 150
   - White matter seed point: x = 158, y = 158, z = 150
   - Fill level: 0.1, 0.1, 0.1
   - Threshold: [0, 1]

4. Outer Skull Segmentation
   - Center of one eye: x = 95

5. Inner Skull Segmentation

Save Results
- Output Folder: /data/projects/zyeneplusv/to/dememe
- Segmentation

Saving segmentation as SubjectA_segments.mat
Segmentation

Image Segmentation

1. Anisotropic Filtering
   - Check inhomogeneity
   - Number of iterations: 5
   - Image diffusion: 3

2. Scalp Segmentation

3. Brain Segmentation
   - Cerebellar low point: x=172, y=158, z=150
   - White matter seed point: Fill level = 0.1, Threshold = 0.1

4. Outer Skull Segmentation
   - Center of one eye: x=95

5. Inner Skull Segmentation

Save Results
- Output Folder: /data/projects/zevnek/common/home/zevnek/sk1/deneme
- Filtered Image
- Segmentation

Segmentation saved as SubjectA_segments.mat
Image Segmentation

```
>> dir SubjectA*
SubjectA_mri.mat   SubjectA_segments.mat

>> load SubjectA_mri
>> mri
mri =

    dim: [256 256 256]
xgrid: [1x256 double]
ygrid: [1x256 double]
zgrid: [1x256 double]
anatomy: [256x256x256 double]
transform: [4x4 double]
hdr: []

>> load SubjectA_segments
>> Segm
Segm =

    scalpmask: [256x256x256 logical]
    brainmask: [256x256x256 logical]
    outerskullmask: [256x256x256 logical]
innerskullmask: [256x256x256 logical]
```
Mesh Generation

Generate Mesh for a 3 or 4 layer head model
Generates a simple source space:
Regular Grid inside the brain
With a given spacing and distance to the mesh
Source Space Generation

From a magnetic Resonance Image

Image Segmentation

Mesh Generation

Source Space Generation

Electrode Co-Registration

NFT: Source space generation

Mesh Folder

/data/projects/zeynep/common/home_zeynep/jo/dene/dene_real

Grid spacing (mm) 8

Min. distance from the mesh (mm) 2

Generate Regular Source Space

Source space saved!
Electrode Co-registration

From a magnetic Resonance Image
- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration

NFT: Electrode co-registration

Load sensor locations
Electrode file name
/data/projects/zeynep/common/home_zeynep/jo/deneme/dene_real

Mesh Folder

Initial co-registration
Translation
Rotation

Complete co-registration
Translation
Rotation

Save initial reg.
Save complete reg.
Electrode Co-registration
Electrode Co-registration

From a magnetic Resonance Image
- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration
Electrode Co-registration

From a magnetic Resonance Image

Image Segmentation

Mesh Generation

Source Space Generation

Electrode Co-Registration...
Electrode Co-registration

From a magnetic Resonance Image
- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration
Electrode Co-registration
Electrode Co-registration

From a magnetic Resonance Image

- Image Segmentation
- Mesh Generation
- Source Space Generation
- Electrode Co-Registration
Head Modeling from Electrode Position Data

- Warp a template mesh to electrode positions
  - When no MR images are available
  - Non-rigid thin-plate spline warping
Template Warping

NFT: Template head model warping

Load sensor data

Output Folder: /data/projects/zeypem/common/home/zeypem/jo/deneme/dene_mni

MNI head model

Warped MNI head model

Start warping
Template Warping
Template Warping
Forward Problem Solver

- MATLAB interface to numerical solvers
- Boundary Element Method or Finite Element Method
  - EEG Only (for now)
  - Interfaces to the Matrix generator executable written in C++
- Other computation done in MATLAB
- Generated matrices are stored on disk for future use.
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM

**BEM Mesh Info**
- Mesh Name: SubjectA
- Number of Layers: 4
- Number of Nodes: 13724
- Number of Elements: 27476
- Number of Nodes/Element: 3

**Show Mesh**

**BEM Model**
- Enter conductivity values:
  - Scalp: 0.33
  - Skull: 0.004
  - Brain: 0.33
  - CSF: 1.79
- Modified (Isolated Problem Approach)
- Create Model
- BEM Model Created

**Session**
- Session Name: s1
- Load Sensors
  - Mesh Coordinates
  - Mesh Node List
- Load
- Show Sensors
- Generate transfer matrix
- Value Changed!

**Forward Problem Solution**
- Load Source Space
- Compute Lead Field Matrix
- Plot Potential Distribution
- For Dipole
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with BEM
Forward Problem Solution with FEM

- Tetgen for mesh generation
  - Uses BEM meshes as boundaries
- METU-FEM to generate transfer matrix
  - Compiled from source
  - Requires PETSc for matrix operations
- metufem .mex file for forward solutions in MATLAB
- Instructions available under README.FEM file.
Forward Problem Solution with FEM
Dipole Fitting

- Requires EEGLAB integration to access Component indices.
- Uses FieldTrip in EEGLAB for dipole fitting.
Warping Demo

- If the subject does not have an MRI, a template head model (MNI) can be warped to electrode locations:
  - Go to NFT-2.4_demo_warping folder
  - Start NFT
  - Enter SubjectA as subject name and s1 as session name
  - Select the current folder as the subject folder
  - Click on Warping
  - After warping, continue with Forward and Inverse modeling.
Output

- Dipole source localization is saved in EEG structure, under EEG.etc.nft.
- After source localization with NFT, you can continue using EEGLAB;

```
EEG.dipfit.model = EEG.etc.nft.model;
```
Start EEGLAB and set your parameters:

```matlab
% Start EEGLAB
eeeglab

EEG = pop_loadset('filename', eeg_file, 'filepath', eeg_path);
[ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );

% set 'of' (output folder), subject_name, session_name, and elec_file
subject_name = 'SubjectA';
session_name = 's1';
nl = 4; % number of layers
plotting = 1;
comp_index = 1:20; % component index for source localization
```
NFT Matlab Scripts

- Realistic modeling from MRI

  % Do segmentation using the GUI
  nft_mesh_generation(subject_name, of, nl)
  nft_source_space_generation(subject_name, of)

  % Do co-registration using the GUI
  nft_forward_problem_solution(subject_name, session_name, of);
  dip1 = nft_inverse_problem_solution(subject_name, session_name, of, EEG, comp_index, plotting, elec_file)
NFT Matlab Scripts

- BEM warping mesh
  
  nft_warping_mesh(subject_name, session_name, elec_file, nl, of, 0, 0);
  
nft_forward_problem_solution(subject_name, session_name, of);
  
dip1 = nft_inverse_problem_solution(subject_name, session_name, of, EEG, comp_index, plotting, elec_file)
NFT Matlab Scripts

- FEM warping mesh
  
  ```matlab
  session_name='s1_fem';
  nft_warping_mesh(subject_name, session_name, elec_file, nl, of,0,1);
  nft_fem_forward_problem_solution(subject_name, session_name, of);
  dip1 = nft_inverse_problem_solution(subject_name, session_name, of, EEG, comp_index, plotting, elec_file)
  ```
NFT Matlab Scripts

- Set NFT dipole structure to EEGLAB dipole structure
  
  ```matlab
  eeglab_folder = dirname(which('eeglab'));
  mri_file = [eeglab_folder '/plugins/dipfit2.2/standard_BEM/standard_mri.mat'];
  EEG.dipfit.mrifile = mri_file;
  EEG.dipfit.model = EEG.etc.nft.model;
  ```
NFT download and reference

- http://www.sccn.ucsd.edu/nft