Report code MRI_segmentation_meshgeneration

%%% inputs %%%

- **MRI**
- **elec25.m and elec_67.m**
  reconstruted 25 channels and 67 channels electrodes in matrix form with size 25*3 and 67*3
  (3 last locations in each electrode file are fiducials: nasion, T9(left-preauricular) and T10(right preauricular))

%%% output variables %%%

- **segmented MRI, named tpm**, i=1:3 for 'gray','skull','scalp' segments
- **3D surface mesh bnd(i).pos, bnd(i).tri**, i=1:3 for 'scalp', 'skull', 'gray'
- **elec_new** achieved after registration of electrodes on scalp mesh by ICP,
- **rms_ER is RMS error after registration and ER is distance error after registration**

- **elec_realigned** achieved after registration of electrodes on scalp mesh by fieldtrip which applies ICP and has more options for better aligning (forexample electrode shaping (warping),
- **rms_ER1 is rms error after registration and ER is distance error after registration**
-

%% initial parameters decision for showing plots (1 is for showing or applying a method)

show_elec25=0; plot 10-20 25 electrodes
show_elec67=1; plot 10-10 67 electrodes
showmri=1; show anatomical MRI
showsegmri=1;show segmented MRI
consequtive_2D_MRI=1; show consequtive MRI slices (i.e 2D MRI slices in axial view)
MRIvolume=0; show whole MRI in 3D
MRIscalpvolume=0; show scalp MRI in 3D view
ICPmethod=1; use ICP method for electrode registration
electrodealign_fieldtrip=1; use fieldtrip for electrode registration

The process step by step is:

1. Build Electrode file locations
   Codes: elec_25.m and elec_67.m
   - build electrodes on sphere according to standards 10-20 and 10-10
   - output matrices: (N*3 locations) (1*N labels) for electrodes

   ft_plot_sens.m (Figure. 1)→ shows electrodes and corresponding labels

2. Read MRI
   - Command: ft_read_mri.m →reads anatomical MRI
   - Output :
     MRI. Anatomy (intensity of voxels)

MRI. Coordsys (coordinate system in which head allocates and MRI is achieved)
MRI. Transform (transformation matrix for transferring from voxel space to real head positions according to coordinate system)
ft_volumereslice.m→ ft_volumereslice interpolates and reslices a volume along the principal axes of the coordinate system (for example CTF) . If the input MRI has a coordsys-field, the centre of the volume will be shifted (with respect to the origin of the coordinate system).
ft_sourceplot.m (Figure. 2) shows MRI 2D slices in 3 orthogonal view

3. Segment MRI
    - Command: ft_volumesegment.m→ segments MRI in to subvolumes (scalp, skull, gray matter)
    - **Method**:
      Layers of head (scalp, skull, brain) are reconstructed by their intensities, morphologies, and MRI atlas.
      MRI atlas or templates (made by registration of different head MRIs) has information of segments of head compartments or tissues (scalp, skull, brain). It contains tissue probability maps (TPM). TPM assigns probability of different head compartments to each voxel.
      For a particular tissue, TPM assigns number one to maximum and zero to minimum probability for each voxel. So, the area with light color TPM shows a particular tissue. The current MRI is segmented first by comparing with TPM from atlases. That is allocating TPM to current MRI is done by comparing with atlas (prior TPMs).
      Bayes rule[1] is applied next step, which uses image intensity and prior TPMs (by registering on atlas) for calculating the posterior TPM which is described in details in [1][2].
    - Output: tissue probability maps (TPM) for MRI different volumes (scalp, skull, gray)
    - Ft_sourceplot.m (Figure. 3) shows segmented MRI 2D slices (TPM) in 3 orthogonal view for scalp

4. mesh generation (3D surface triangular) from the segmented MRI
    - command: ft_prepare_mesh and ft_plot_mesh.m→ reconstructs surface 3D mesh from each segmented MRI part and plots it (Figure. 4)
    - method:
      input for mesh generation is TPM from segmented MRI.
      method is named marching cube [10], This method uses a smoothly varying level set function, $\emptyset(i,j,k)$ for each intensity or TPM and each location of segmeneted MRI that changes sign at the object boundary, hance the surface is achieved. As the achieved positions by thresholding are not a smooth surface some mesh cosmetics are applied after initial mesh point detection by thresholding[11].

[1] In probability theory and statistics, **Bayes' theorem** (alternatively **Bayes' law** or **Bayes' rule**, also written as **Bayes's theorem**) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

5. consequtive_2D_MRI
    - As MRI contains stacked slices (here 256 slices of each 256*256) we can show them if consequtive_2D_MRI=1 (Figure. 5)
6. Showing MRIvolume
    - If MRIvolume=1 it shows 3D MRI
    - Method:
      MRI voxels have locations in voxel space
      Ft_warp_apply.m converts each voxel from voxel space to real location according to head coordinate.
      MRI also contains intensity for each voxel (MRI.anatomy)
      So, a 3D MRI is plotted after knowing locations and intensities for each voxel figure (6); ft_plot_mesh.m for showing 3D MRI

7. MRIscalpvolume
   → like the last step, it shows just scalp volume
   If MRIscalpvolume=1; it shows scalp volume
   figure (7); ft_plot_mesh.m for showing 3D MRI

8. Register electrodes on scalp mesh
    - Command: ICP_electrode_realign.m applies ICP method for electrode registration
    - Inputs are electrodes and mesh nodes
    - If ICPmethod=1, registration is done by ICP (iterative closest point) [5][6]
    - ICP: this method is described in document electrode_MRI_coregistration.docx.
    - Summary of ICP method

Mathematically, the objective of the ICP algorithm is to iteratively find a rotation matrix R and a translation vector T that moves electrodes to align with mesh nodes. The purpose is changing electrode locations (S_i) in order to achieve the minimum distance between each electrode location and the nearest mesh node locations (M_i). Mesh node locations is also called reference set. In other words, the purpose is to minimize the sum of squared Euclidean distance of electrode locations and their corresponding nearest mesh nodes in 5 iterations [5][6] as following:

- Calculate the corresponding mesh (reference) points, $M_{i,k}$ at iteration k, so that, $\|(S_{i,k} - M_{i,k})\|$=min;
- Calculate the rotation matrix R_k and the translation vector T_k at iteration k so that, $\sum_{i=1}^{N}\|R_k\, S_{i,k} + T_k - M_{i,k}\|^2$=min;

$\|\ \ \|$ is euqlidean distance (The Euclidean distance between points p and q is the length of the line connecting them)
N: number of electrodes
R_k : rotation matrix at iteration k
$S_{i,k}$: electrode locations at iteration k
T_k : translation vector at iteration k
$M_{i,k}$: nearest mesh node locations at iteration k

- Calculate $S_{k+1} = \{S_{i,k+1} | S_{i,k+1} = R_k\, S_{i,k} + T_k\}$;
- Calculate the error, $d_{k+1} = \sum_{i=1}^{N}\|(s_{i,k+1} - M_{i,k})\|^2$
- If $d_{k+1} < \tau\ and\ \tau$ is a preset threshold. The initial value for rotation matrix can be $R_0 = I$, the unitary matrix and for translation vector is $T_0 = (0,0,0)$.

- As a conclusion, ICP needs the prior information of fiducial points for aligning the coordinate system. The algorithm works quite effectively when given a good initial estimate.

- figure (8); ft_plot_mesh.m for showing 3D surface mesh and ft_plot_sens.m plots 3D electrode locations after registration

9. Register electrodes on scalp mesh by fieldtrip

   - fieldtrip_electrode_realign.m

   - If electrodealign_fieldtrip=1, registration is applied by fieldtrip
   - Method: ICP and electrode warping by different methods
   - figure (9); ft_plot_mesh.m for showing 3D surface mesh and ft_plot_sens.m plots 3D electrode locations after registration
   -

References

1. Ashburner, J., Friston, K.J., 1997. Multimodal image coregistration and partitioning—a unified framework. NeuroImage 6 (3), 209– 217.
2. Ashburner J, Friston KJ. Unified segmentation. Neuroimage. 2005 Jul 1;26(3):839-51.
3. https://en.wikipedia.org/wiki/Marching_cubes#cite_note-1
4. Sazonov I, Nithiarasu P. Semi-automatic surface and volume mesh generation for subject-specific biomedical geometries. International Journal for Numerical Methods in Biomedical Engineering. 2012 Jan 1;28(1):133-57.
5. He Y, Liang B, Yang J, Li S, He J. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. Sensors. 2017 Aug 11;17(8):1862.
6. D. J. Kroon, "Iterative Closest Point using finite difference optimization to register 3D point clouds affine," http://www.mathworks.com/matlabcentral/fileexchange/24301-finite-iterative-closest-point.
7.