

Leadfield-Interpolation für EEG

Christof Pieloth

12.06.2014

1 Problemstellung

- Source Imaging benötigt eine Leadfield-Matrix
- Leadfield-Matrix ist abhängig von:
 - Probanden bzw. deren Kopfmodell
 - konkreter Messaufbau, EEG-Sensorpositionen
- Berechnung der Leadfield für ein 60-Kanal-EEG und ca. 240.000 Quellen benötigt mehrere Minuten
- neben dem Zeitaufwand ist auch der zusätzliche Arbeitsschritt beim Messaufbau hinderlich

2 Idee

- Berechnung virtueller EEG-Sensoren, welche auf der Kopfoberfläche verteilt sind
- Anzahl virtueller EEG-Sensoren sollte um ein Vielfaches höher sein, als die Anzahl tatsächlichen EEG-Sensoren (genauere Untersuchungen des Einflusses)
- Leadfield der virtueller EEG-Sensoren wird hochaufgelöste Leadfield oder kurz HD-Leadfield genannt
- HD-Leadfield
 - kann in Rahmen der Vorbereitungen, Quellenmodell, Segmentierung usw., berechnet werden
 - eine pro Kopf-/Quellenmodell
- Generierung der Leadfield für tatsächliche EEG-Sensorpositionen aus den HD-Leadfield
 - aus der Menge der virtuellen EEG-Sensoren wird für jeden tatsächlichen EEG-Sensor der naheliegendste Sensor gesucht
 - für diesen Sensor wird der entsprechende Vektor aus der HD-Leadfield in die „normale“ Leadfield übertragen
 - bei Verwendung von k Nachbarn/Sensoren findet eine Interpolation statt
- Problem:
 - Modell der Kopfoberfläche bzw. entsprechender BEM-Layer ist im AC-PC-Koordinatensystem
 - EEG-Sensorpositionen sind im Kopfkoordinatensystem (LPA, RPA, Nasion)

3 Arbeitsschritte

1. HD-Leadfield:
 - a) Generierung der virtuellen EEG-Sensorpositionen
 - b) Berechnung der HD-Leadfield für die virtuellen EEG-Sensorpositionen
2. Alignment: Transformationsmatrix zwischen Kopfkoordinaten und AC-PC-Koordinaten
 - a) Schätzung einer initialen Transformationsmatrix mit Hilfe von Fiducials
 - b) Optimierung der initialen Transformation durch Minimierung des Abstands zwischen EEG-Haube und Kopfoberfläche mittels ICP
3. Leadfield-Interpolation:
 - a) Transformation der EEG-Sensorpositionen in AC-PC-Koordinaten
 - b) Suche von 4 virtuellen EEG-Sensoren mit geringsten Abstand zum EEG-Sensor
 - c) Interpolation aus den 4 HD-Leadfield-Vektoren bildet Leadfield-Vektor für diesen EEG-Sensor

4 HD-Leadfield

- Generierung der virtuellen EEG-Sensorpositionen mit dem Tool `eeg_sensor_generator`
- Beispiel:

```
eeg_sensor_generator \  
-i exp01a-5120-5120-5120-bem.fif \  
-o exp01a-eeg-virtual-sensors.fif \  
-s 1 \  
-c 0.33  
# -i: FIFF-Datei mit dem BEM-Layer Head/Scalp Surface  
# -o: Ausgabedatei in FIFF-Format  
# -s: Nimmt jeden Punkt (1), jeden zweiten (2), usw. vom BEM-Layer  
# -c: Faktor zur Entfernung die unteren Punkte des BEM-Layers
```

- Ausgabedatei enthält nur virtuelle EEG-Sensorpositionen in AC-PC Koordinaten!
- Skip-Parameter beeinflusst die Anzahl der virtuellen EEG-Sensorpositionen, abhängig von Anzahl der Punkte auf dem BEM-Layer
- Cut-Parameter beschneidet das 3D-Modell
 - häufig enthält der Head-BEM-Layer noch Teile vom Hals
 - dieser kann mit dem Parameter abgeschnitten werden
 - Bestimmung des niedrigsten und höchsten Punktes auf der z-Achse aus dem BEM-Layer:

$$z_{min}, z_{max}$$

- Bildung eines Schwellwertes:

$$z_{cut} = z_{min} + (z_{max} - z_{min}) \cdot cutFactor$$

- Entfernung aller Punkte mit:

$$P_z < z_{cut}$$

- Berechnung der HD-Leadfield mit Hilfe der MNE Toolbox, zum Beispiel:

```

mne_forward_solution --eeg --fixed \
--src <source file> \
--trans <ASCII file containing a 4x4 identity matrix>
--meas exp01a-eeg-virtual-sensors.fif \
--bem exp01a-5120-5120-5120-bem.fif \
--fwd exp01a-fwd-eeg-virtual-sensors.fif

```

5 Alignment

- Fiducials: $F = \{LPA, RPA, N\}$
 - linkes Ohr: lpa, left pre-auricular LPA
 - rechtes Ohr: rpa, right pre-auricular RPA
 - Nasenwurzel: nasion N
 - dienen zur Bestimmung der initialen Transformationsmatrix, da Registrierung vorhanden
- Gegeben:
 - $F_{head} = \{LPA_{head}, RPA_{head}, N_{head}\}$... Fiducials in Kopfkoordinaten aus Digitization-/Isotrak-Punkte
 - P_{eeg} ... EEG-Sensorpositionen in Kopfkoordinaten
 - P_{bem} ... Punkte des BEM-Layer (Head/Scalp Surface)
- Gesucht:
 - $F_{acpc} = \{LPA_{acpc}, RPA_{acpc}, N_{acpc}\}$... Fiducials in AC-PC-Koordinaten
 - T_I ... initiale Transformationsmatrix
 - T_F ... Transformation von Kopfkoordinaten nach AC-PC-Koordinaten

1. Fiducials in AC-PC-Koordinaten manuell bestimmen, z.B. BEM-Layer (Head/Scalp Surface)

2. initiale Transformationsmatrix T_I zur Lösung von $T_I \cdot F_{head} = F_{acpc}$

- Registrierung der Punkte im Kopf- und AC-PC-Koordinatensystem:

$$LPA_{head} \iff LPA_{acpc}$$

$$RPA_{head} \iff RPA_{acpc}$$

$$N_{head} \iff N_{acpc}$$

- Lösung mittels Singulärwertzerlegung
- Implementierung der Point Cloud Library:

```
pcl::registration::TransformationEstimationSVD
```

- T_I optimal für Fiducials, jedoch nicht für EEG-Haube und Kopfoberfläche, weil:
 - F_{acpc} manuell mit anderer Software bestimmt als F_{head}
 - keine Optimierung über alle Punkte bzw. des Abstands zwischen EEG-Haube und Kopfoberfläche

3. finale Transformationsmatrix T_F

- EEG-Sensorpositionen und Punkte im BEM-Layer sind unabhängig, d.h. keine eindeutige Registrierung

- Lösung mittels Iterative Closest Point (ICP)
- Implementierung der Point Cloud Library:

```
pcl::IterativeClosestPoint
```

- Start des ICP mit T_I , P_{eeg} , $cut(P_{bem})$
- $cut(P_{bem})$... BEM-Layer wird wie bei der Generierung der virtuellen EEG-Sensoren beschnitten, s.o.

6 Leadfield-Interpolation

- Gegeben:

- L_{hd} ... HD-Leadfield
- P_v ... Positionen der virtuellen EEG-Sensoren (AC-PC), $p_v \in P_v$
- P_{eeg} ... Positionen der „tatsächlichen“ EEG-Sensoren (Head)

- Gesucht:

- P'_{eeg} ... Positionen der „tatsächlichen“ EEG-Sensoren (AC-PC), $p_{eeg} \in P'_{eeg}$
- L_{eeg} ... Leadfield für EEG

1. Transformation der EEG-Sensorpositionen von Head-Koordinaten in AC-PC-Koordinaten:

$$P_{eeg} \rightarrow P'_{eeg}$$

2. Bestimmung der 4 Nachbarn mit geringsten Abstand

- Implementierung der Point Cloud Library:

```
pcl::KdTreeFLANN::nearestKSearch()
```

- für jeden Punkt $p_i \in P'_{eeg}$ suche 4 Punkte $p_{i,k} \in P_v$ mit geringsten Abstand:

$$(p_{i,1}, p_{i,2}, p_{i,3}, p_{i,4}) = \text{nearestKSearch}(p_i, 4, P_v)$$

3. Interpolation

- Bestimme 4 Wichtungsfaktoren $w_{i,k}$ und w_i :

$$d(p_1, p_2) \dots \text{Abstand zw. } p_1 \text{ und } p_2$$

$$w_{i,k} = \frac{1}{d(p_i, p_{i,k})}$$

$$w_i = \sum_{k=1}^4 w_{i,k}$$

$$w_i = w_{i,1} + w_{i,2} + w_{i,3} + w_{i,4}$$

- für jeden Punkt $p_i \in P'_{eeg}$ und Dipol s berechne Interpolation der 4 HD-Leadfield-Zeilen:

$$r_{i,k} \dots \text{Zeilenindex von } p_{i,k} \text{ in } L_{hd}$$

$$L_{eeg}(i, s) = \sum_{k=1}^4 L_{hd}(r_{i,k}, s) \cdot \frac{w_{i,k}}{w_i}$$

Literatur

- [1] A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, (0):–, 2013.
- [2] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145 –152, 2001.
- [3] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [4] Blaise Yvert, Anne Crouzeix-Cheylus, and Jacques Pernier. Fast realistic modeling in bioelectromagnetism using lead-field interpolation. *Human Brain Mapping*, 14:48–63, 2001.