

Übung 6

Lernziele

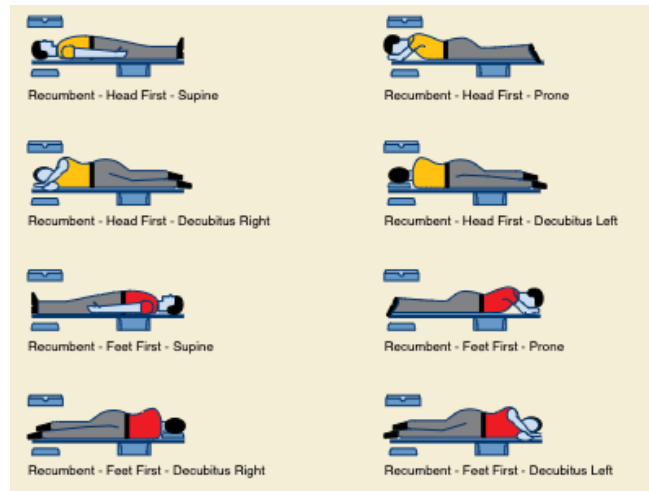
Die Studierenden...

- ... können Meta-Daten wie Patientenname, Datum, Studien-ID und Uhrzeit aus einer DICOM-Datei extrahieren.
- ... können anhand der Patienten-Orientierung vorhersagen, aus welcher Ansicht der Patient betrachtet wird.
- ... können ausführen, wozu die Meta-Information Pixel-Abstand benutzt werden kann.
- ... können den Unterschied zwischen «Slice Thickness» und «Slice-Abstand» erläutern.

Aufgaben

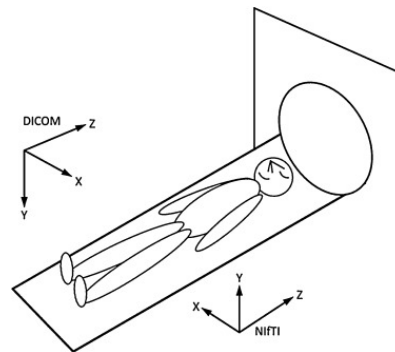
1. Extrahieren Sie folgende Informationen aus den DICOM-Metadaten eines beliebigen Slices und geben Sie sie in der Konsole aus:
 - Name des Patienten
 - Geschlecht des Patienten
 - Patienten ID
 - Studien-ID
 - Bildgebungsverfahren
 - Datum der Bild-Aufnahme
 - Uhrzeit der Bildaufnahme
 - Abstand zwischen Pixeln
 - Koordinaten des Scans (Image Position Patient)
 - Ausrichtung des Scans (Image Orientation Patient)
 - Die Ausrichtung besteht aus einer Liste mit 6 Zahlen:
`[x_row, y_row, z_row, x_column, y_column, z_column]`
Die **ersten drei Zahlen** geben die **Kosinus-Werte** der **Winkel** zwischen der **ersten Zeile** und den Einheits-Koordinaten an.
Die **letzten drei Zahlen** geben die **Kosinus-Werte** der **Winkel** zwischen der **ersten Spalte** und den Einheits-Koordinaten an.

- Ermitteln Sie, wie der Patient beim Scan positioniert wurde. Eine Übersicht der Positionierungs-Arten:



Quelle: <https://dicom.innolitics.com/ciods/ct-image/general-series/00185100>

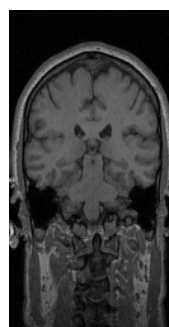
- Mit der **Ausrichtung des Scans** und der **Positionierung des Patienten** kann man vorhersagen, was auf dem Bild zu sehen ist. Überlegen Sie für diesen Scan, aus welcher Perspektive der Patient betrachtet wird (Seitenansicht, Frontalansicht, oder Draufsicht). Folgende Abbildung hilft Ihnen dabei:



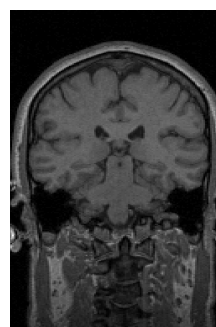
Quelle: <https://www.fieldtriptoolbox.org/faq/coordsys/>

- Visualisieren Sie das Bild (Attribut *pixel_array*). Stimmt es mit der Erwartung überein?
- Was sagt uns das Attribut *SliceThickness*? Kann man es für eine massstabsgetreue 3D Rekonstruktion benutzen?
- Nachfolgend wird ein Slice aus einer anderen Perspektive gezeigt. Rechts ist die anatomisch korrekte Visualisierung. Links ist eine einfache Visualisierung der Slices aus einer anderen Perspektive so wie sie in dem 3D Array gespeichert sind. Wo liegt der Fehler links?

Frontale Visualisierung



Korrekte frontale Visualisierung



Bonus

DICOM Dateien werden in der Regel sortiert und mit einer Nummerierung ausgegeben. Jedoch kann es vorkommen, dass die DICOM Dateien in einer falschen Reihenfolge eingelesen werden (aus diversen Gründen). In solchen Fällen ist es wichtig, dass die Slices anhand der Scan-Koordinaten sortiert werden. Um ein solches Problem zu simulieren und zu lösen, gehen Sie wie folgt vor:

Schritt 1: Dateien falsch einlesen

1. Speichern Sie den Pfad vom Ordner mit den DICOM-Dateien in einer Variable (z.B. *folder_path*)
2. Importieren Sie das package *os* (ist schon installiert). Mit dem Befehl

```
list_file_names = os.listdir(folder_path)
```

können Sie alle Datei-Namen des Ordners in einer Liste speichern. Mit

```
print(liste_file_names)
```

können Sie die Liste mit den Datei-Namen ausgeben lassen.

3. Erstellen Sie ein 3D Array mit *np.zeros()* für die Slices, die Sie einlesen werden. Erstellen Sie ausserdem eine leere Liste, um die Koordinaten der Slices zu speichern.
4. Iterieren Sie mit einer *for*-Schleife durch die Liste *liste_file_names*. Bei jeder Iteration:
 - Bild aus DICOM Datei einlesen und in dem 3D Array an der *i*-ten Stelle einspeichern
 - Koordinate entlang der Scan-Achse einlesen und zur Liste hinzufügen

Am Ende sollten Sie das 3D Array mit Slices und eine 1D Liste mit den Koordinaten entlang der Scan-Achse haben.

Schritt 2: Slices sortieren

Mit dem Befehl

```
order = np.argsort(liste_koordinaten)
```

bekommt man die Reihenfolge, in der man *liste_koordinaten* sortieren müsste, um eine aufsteigende Reihenfolge zu haben. Diese Reihenfolge kann man dann benutzen, um die Slices zu sortieren:

```
sorted_slices = all_slices[:, :, order]
```

wobei *all_slices* das 3D Array ist. Beachten Sie ausserdem die Dimension, in der Sie sortieren möchten. In diesem Beispiel sind die ersten beiden Dimensionen Höhe und Breite der Slices und die dritte Dimension ist die Slice-Nummer, weshalb *order* in der dritten Dimension eingesetzt wird.