

Übersicht aller verwendeten Skripte in der CAEE

Im Folgenden werden alle Python, Grasshopper und Matlab-Skripte in ihrer Funktion mit Inputs, Outputs sowie einer Kurzbeschreibung vorgestellt.

1. Python Skripte

Tabelle A.1: Kurzbeschreibung des GUI-Skripts

Name:	gui.py
Software:	Python
Input:	User-Input: Implantatart, Pfad CT-Scan, Implantatparameter, Output-Pfad
Output:	-
Kurzbeschreibung:	GUI.py erstellt die Benutzeroberfläche und gibt den User-Input an die Hauptfunktionen weiter.

Tabelle A.2: Kurzbeschreibung des main-Skripts

Name:	main.py
Software:	Python
Input:	User-Input: Implantatart, Pfad CT-Scan, Implantatparameter, Output-Pfad
Output:	Oberflächennetz des Implantats im STL-Format
Kurzbeschreibung:	main.py enthält die Hauptfunktionen für die Erstellung der Implantate.

Tabelle A.3: Kurzbeschreibung des functions-Skripts

Name:	functions.py
Software:	Python
Input:	Verschieden
Output:	Verschieden
Kurzbeschreibung:	functions.py enthält alle Funktionen, die in Hauptfunktionen wiederholt aufgerufen werden, darunter z.B. eine Verbindung zum Aufbau mit dem Compute-Server oder eine Funktion um STL-Dateien in ihre Eckpunkte und Flächen zu zerlegen.

2. Grasshopper Skripte

Tabelle A.4: Kurzbeschreibung des mesh_creation-Skripts

Name:	mesh_creation.gh
Software:	Grasshopper
Input:	Punktewolke des Femurs
Output:	Oberflächennetz des Femurs im STL-Format
Kurzbeschreibung:	mesh_creation.gh wandelt die Punktewolke aus der Bildverarbeitung in ein Oberflächennetz im STL-Format um, dass anschließend in der Plausibilisierung und der Landmarkenerkennung verarbeitet werden kann.

Tabelle A.5: Kurzbeschreibung des geo_planes_hip-Skripts

Name:	geo_planes_hip.gh
Software:	Grasshopper
Input:	Punktewolke von Femur und Markraum oder Flächen und Eckpunkte der Oberflächenmodelle
Output:	Ebenen, Ellipsenachsen, Markraumgeometrie, Femurachse, Halsschaftachse
Kurzbeschreibung:	geo_planes_hip.gh verarbeitet die Punktewolke aus der Bildverarbeitung und leitet verschiedene Geometrieinformationen für die Weiterverarbeitung in geo_construction_hip.gh ab. Bezieht sich auf die Erstellung des Hüftimplantats.

Tabelle A.6: Kurzbeschreibung des geo_construction_hip-Skripts

Name:	geo_construction_hip.gh
Software:	Grasshopper
Input:	Ebenen, Ellipsenachsen, Markraumgeometrie, Femurachse, Halsschaftachse
Output:	Oberflächennetz des individualisierten Hüftimplantats
Kurzbeschreibung:	geo_construction_hip.gh verarbeitet die Geometrieinformationen aus geo_planes_hip.gh für die Erstellung des individualisierten Hüftimplantats.

Tabelle A.7: Kurzbeschreibung des geo_planes_endo-Skripts

Name:	geo_planes_endo.gh
Software:	Grasshopper
Input:	Punktwolke von Femur und Markraum oder Flächen und Eckpunkte der Oberflächenmodelle, Landmarken
Output:	Ebenen, Ellipsenachsen, Markraumgeometrie, Femurachse
Kurzbeschreibung:	geo_planes_endo.gh verarbeitet die Punktwolke aus der Bildverarbeitung und leitet verschiedene Geometrieinformationen für die Weiterverarbeitung in geo_construction_endo.gh ab. Bezieht sich auf die Erstellung der Endo-Exo-Prothese,

Tabelle A.8: Kurzbeschreibung des geo_construction_endo-Skripts

Name:	geo_construction_endo.gh
Software:	Grasshopper
Input:	Ebenen, Ellipsenachsen, Markraumgeometrie, Femurachse
Output:	Oberflächennetz der individualisierten Endo-Exo-Prothese
Kurzbeschreibung:	geo_construction_endo.gh verarbeitet die Geometrieinformationen aus geo_planes_hip.gh für die Erstellung der individualisierten Endo-Exo-Prothese.

Tabelle A.9: Kurzbeschreibung des hip_socket-Skripts

Name:	hip_socket.gh
Software:	Grasshopper
Input:	Landmarken, Oberflächennetze beider Femora, beider Beckenhälften und dem Kreuzbein in Form von Flächen und Eckpunkten
Output:	Individualisierte Hüftpfanne und Beinlängendifferenz
Kurzbeschreibung:	hip_socket.gh generiert eine individualisierte Hüftpfanne basierend auf den Oberflächennetzen der Hüfte. Zusätzlich werden die Landmarken genutzt, um die Beinlängendifferenz zu berechnen.

Tabelle A.10: Kurzbeschreibung des lattice_creation_outer-Skripts

Name:	lattice_creation_outer.gh
Software:	Grasshopper
Input:	Dichteverteilung im RPT-Format, Femurachse, Implantatgeometrie
Output:	Gitterstrukturen des Implantatrands

Kurzbeschreibung:	lattice_creation_outer.gh gradiert den Rand des Hüftimplantats basierend auf der Dichteverteilung.
-------------------	--

Tabelle A.11: Kurzbeschreibung des lattice_creation_inner-Skripts

Name:	lattice_creation_inner.gh
Software:	Grasshopper
Input:	Dichteverteilung im RPT-Format, Femurachse, Implantatgeometrie
Output:	Gitterstrukturen innerhalb des Implantats
Kurzbeschreibung:	lattice_creation_inner.gh gradiert das Innere des Hüftimplantats basierend auf der Dichteverteilung.

Tabelle A.12: Kurzbeschreibung des lattice_creation_merge-Skripts

Name:	lattice_creation_merge.gh
Software:	Grasshopper
Input:	Femurachse, Implantatgeometrie, Innere und äußere Gitterstrukturen
Output:	Gradiertes Hüftimplantat
Kurzbeschreibung:	lattice_creation_inner.gh fügt die Outputs aus lattice_creation_outer.gh und lattice_creation_inner.gh zu einem gradierten Implantat zusammen.

3. Matlab Skripte

Tabelle A.13: Kurzbeschreibung der femur_landmark-Funktion

Name:	femur_landmark.m
Software:	Matlab
Input:	Oberflächennetz des Femurs
Output:	Landmarken des Femurs z.B. Rotationszentrum, Radius des Kugelkopfes, Halsschaftachse, usw. Siehe https://github.com/RWTHmediTEC/FemoralCoordinateSystem
Kurzbeschreibung:	femur_landmark.m identifiziert automatisiert Landmarken am Oberflächennetz, durch eine Atlas-Registrierung und gibt diese als Koordinaten aus.

Tabelle A.14: Kurzbeschreibung der pelvic_landmark-Funktion

Name:	pelvic_landmark.m
Software:	Matlab
Input:	Oberflächennetze beider Femora, beider Beckenhälften und dem Kreuzbein im STL-Format
Output:	Landmarken des Femurs und des Beckens, siehe https://github.com/RWTHmediTEC/FemoralCoordinateSystem und https://github.com/RWTHmediTEC/PelvicLandmarkIdentification
Kurzbeschreibung:	landmarks_hip_socket.m enthält Funktionen für die Landmarkenerkennung von Femur und Becken. Ausgegeben werden die Landmarken in Form von Koordinaten.