



High-Performance
Computing Center
Stuttgart

Berechnung des scheinbaren Elastizitätsmoduls von trabekulärem Knochen mittels Finite-Elemente Simulation

Simulierte Welten

1. Treffen

2. Treffen

Literaturverzeichnis

Substantia spongiosa

H L R I S

- Spongiosa ist ein im Innenraum des Knochens schwammartig aufgebautes System
- Besteht aus feinen Knochenbälkchen (Trabekeln)

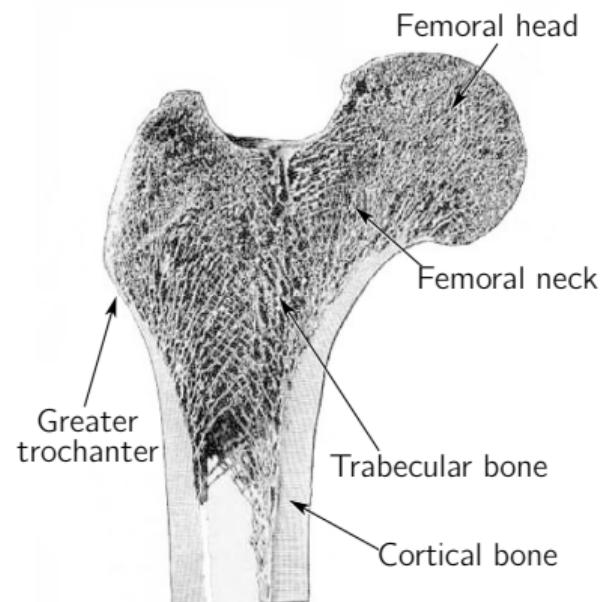


Abbildung: Anatomie des menschlichen Oberschenkels

[1]

- Scheinbares Elastizitätsmodul: Beschreibt das elastische Verhalten der makroskopischen Struktur des trabukulären Knochens
- Das scheinbare Elastizitätsmodul wird zum Beispiel über den Druckversuch bestimmt
- Das scheinbare Elastizitätsmodul berechnet sich mittels Gleichung (1)

$$E_{\text{App}} = \frac{\sigma}{\varepsilon} \quad (1)$$

Die Dehnung ε berechnet sich mittels Gleichung (2), wobei Δl die Längenänderung und l_0 die ursprüngliche Länge ist.

$$\varepsilon = \frac{\Delta l}{l_0} \quad (2)$$

Die Spannung σ berechnet sich mittels Gleichung (3), wobei F_{tot} die Reaktionskraft und A die Schnittfläche ist.

$$\sigma = \frac{F_{\text{tot}}}{A} \quad (3)$$



Abbildung: FE Pipeline

Funktionsweise Computertomographie

H L R I S



Abbildung: Computertomograph

[2]

Funktionsweise Mikrocomputertomographie

H L R I S

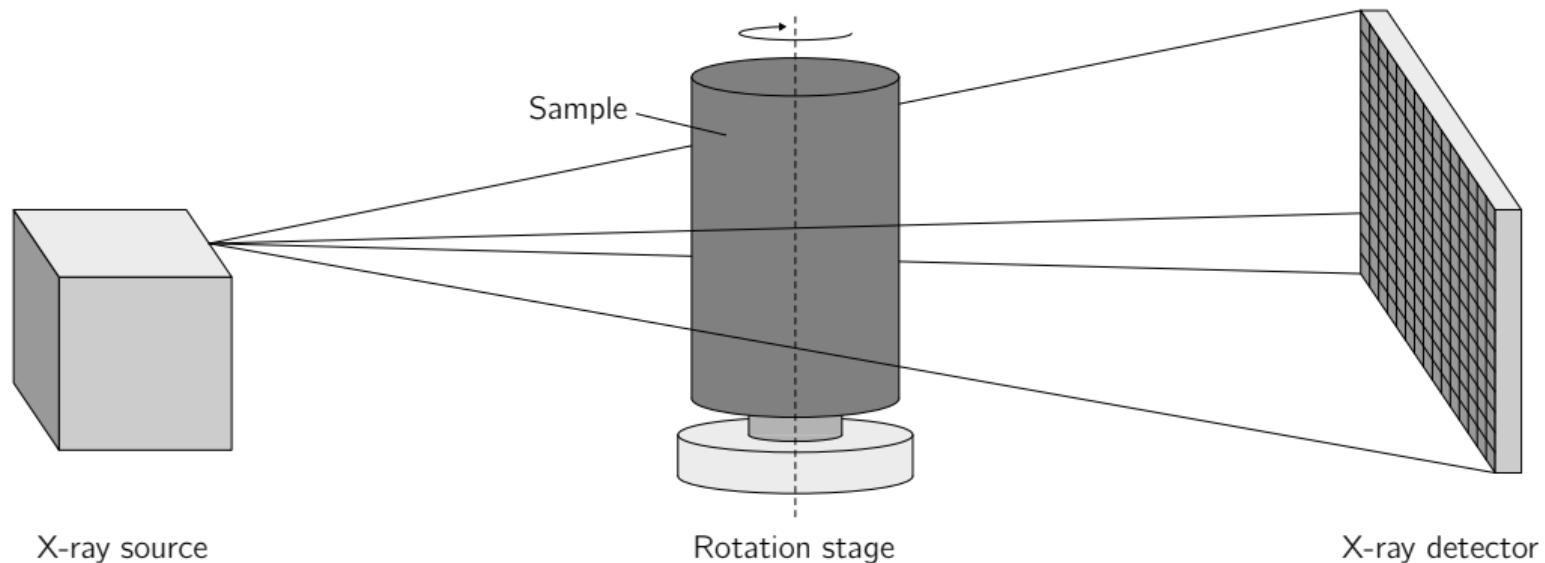


Abbildung: Schematische Darstellung eines Mikro CT-Systems

[3]

Datensatz

H L R I S

Lab	LTM@IOR Bologna
Sample	Human trabecular bone
Voxel size	19,5 µm
Number of voxels per edge	200
Edge length	4 mm

Tabelle: Scan-Parameter des Datensatzes

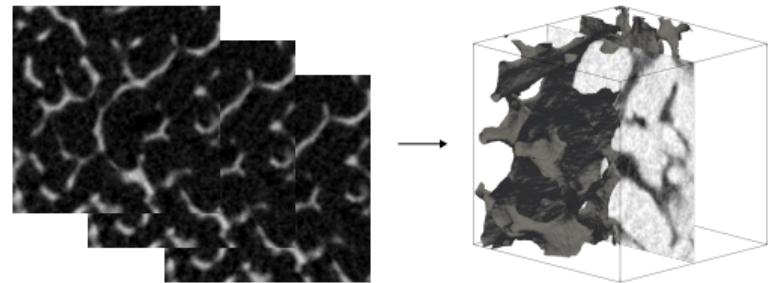


Abbildung: 3D-Rekonstruktion aus mehreren Bildern

1. Treffen

Arbeitsauftrag

- Installation von Visual Studio Code (Entwicklungsumgebung)
<https://code.visualstudio.com/docs/setup/windows>
- Installation von PrePoMax (FEM-Softwarepaket)
<https://prepomax.fs.um.si/>
- Installation von ParaView (Wissenschaftliche Visualisierungen)
<https://www.paraview.org/download/>
- Installation von Python und Anaconda (Programmiersprache)
<https://www.elab2go.de/demo-py1/installation-python-anaconda.php>
- Conda Umgebung erstellen (siehe Abschnitt 6 in der Installationsanleitung)
`conda create --name py3818 python==3.8.18`
- Conda Umgebung aktivieren
`conda activate py3818`

Installation der notwendigen Module und Pakete:

- Installation von NumPy (array processing for numbers, strings, records, and objects)
`conda install -c conda-forge numpy`
- Installation von SciPy (scientific library for Python)
`conda install -c conda-forge scipy`
- Installation von scikit-image (image processing in Python)
`conda install -c conda-forge scikit-image`
- Installation von PyPNG (Python PNG image encoder and decoder)
`conda install -c conda-forge pypng`
- Installation von tqdm (progress meter)
`conda install -c conda-forge tqdm`
- Installation von Matplotlib (publication quality figures in Python)
`conda install -c conda-forge matplotlib`
- Installation von tifffile (read and write image data from and to TIFF files.)
`conda install -c conda-forge tifffile`

- Installation von tifffile (multidimensional image analysis)
`conda install -c conda-forge itk`
- Installation von vtk (process images and create 3D computer graphics)
`conda install -c conda-forge vtk`
- Installation von ccx2paraview (CalculiX to Paraview converter)
`conda install -c conda-forge ccx2paraview`
- Installation von meshio (I/O for various mesh formats)
`conda install -c conda-forge meshio==5.0.0`
- Installation von PyMCubes (marching cubes for Python)
`pip install PyMCubes`
- Installation von pygalmesh (Python frontend to CGAL's 3D mesh generators)
`conda install -c conda-forge pygalmesh`
- Installation von DXchange (data I/O for tomography)
`conda install -c conda-forge dxchange`
- Installation von ciclope (computed tomography to finite elements)
`pip install ciclope[all]`

- Bourne und Meulen: „Finite element models predict cancellous apparent modulus when tissue modulus is scaled from specimen CT-attenuation“
- Cox, Smith, Meulen und Cole: „Heterogeneous Tissue Modulus Improved Prediction of Mechanical Behavior in Osteoporotic Vertebral Cancellous Bone“

1. Einlesen des Datensatzes (*read tif image stack, numpy*)
2. Filtern des Datensatzes (*Gauß-Glättung, skimage, gaussian*)
3. Binarisieren des Datensatzes
(*Schwellenwertverfahren, skimage, thresholding, threshold_otsu*)
4. Plotten des Histogramms des Datensatzes
(*matplotlib*)

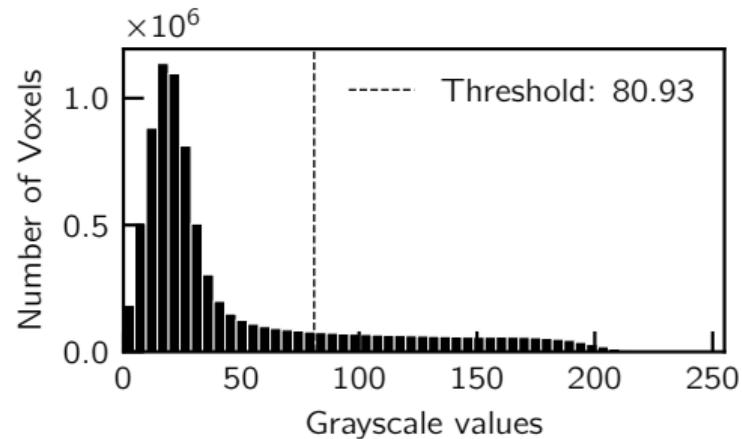


Abbildung: Histogramm als Ziel

Hinweis - Einlesen des Datensatzes

H L R I S

```
1 import os
2 import numpy as np
3 import tifffile as tiff
4 import matplotlib
5 import matplotlib.pyplot as plt
6
7 input_file = '../dataset/3155_D_4_bc_0000.tif'
8
9 input_dir = os.path.dirname(input_file)
10 stack_files = []
11
12 for f in os.listdir(input_dir):
13     if os.path.isfile:
14         stack_files.append(os.path.join(input_dir, f))
15
16 stack_files.sort()
17 data_3D = tiff.imread(stack_files)
18
19 plt.figure()
20 plt.imshow(data_3D[0, :, :], cmap="bone")
21 plt.show()
```

Listing: *importing_the_data_set*

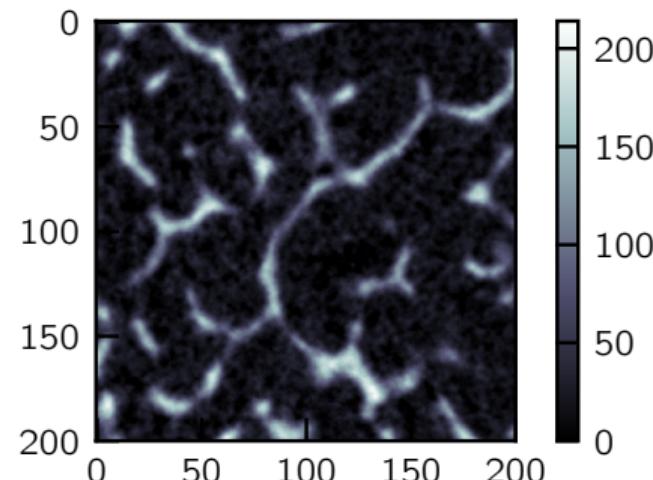


Abbildung: Datensatz

Hinweis - Filtern des Datensatzes

H L R I S

```
data_3D = gaussian(data_3D, sigma=1, preserve_range=True)
```

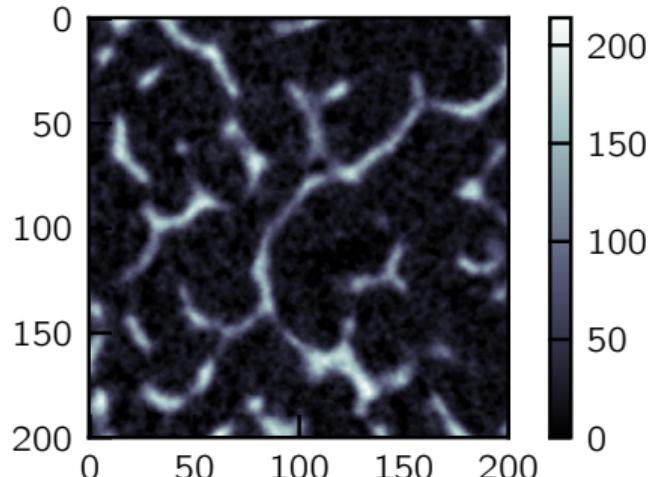


Abbildung: Datensatz

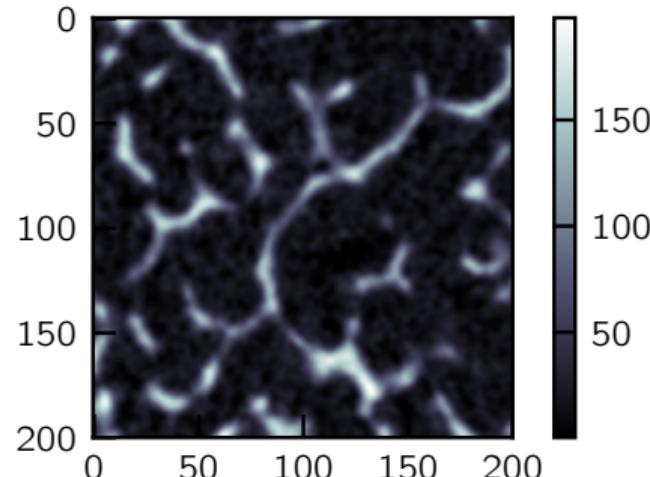


Abbildung: Datensatz nach der Glättung



Abbildung: FE Pipeline

Definition

Segmentierung bezeichnet die Erzeugung inhaltlich zusammenhängender Regionen durch Zusammenfassung benachbarter Pixel oder Voxel entsprechend einem bestimmten Homogenitätskriterium.

Anmerkung: Die Wahl der Segmentierungstechnik und der Parameter wirkt sich auf die Genauigkeit der mikroarchitektonischen Darstellung des trabekulären Knochens aus, was wiederum biomechanische Simulationen und Analysen beeinflusst. So wirkt sich die Segmentierung beispielsweise auf die Quantifizierung des Knochenvolumens und der Strukturparameter aus, die für die Bewertung der Knochensteifigkeit und des Potenzials für mechanisches Versagen entscheidend sind [8].

[7, 8]

Der Datensatz wird durch Spline-Interpolation vergrößert oder verkleinert.

```
1 import numpy as np
2 import scipy.ndimage
3 from matplotlib import pyplot as plt
4
5 ndarray = np.array([
6 [11, 12, 13, 14],
7 [21, 22, 23, 24],
8 [31, 32, 33, 34],
9 [41, 42, 43, 44]])
10
11 result = scipy.ndimage.zoom(ndarray, 0.5,
12     output=None, order=2)
13
14 fig = plt.figure()
15 ax1 = fig.add_subplot(121)
16 ax1.imshow(ndarray)
17 ax2 = fig.add_subplot(122)
18 ax2.imshow(result)
19 plt.show()
```

Listing: *scipy_ndimage_zoom.py*

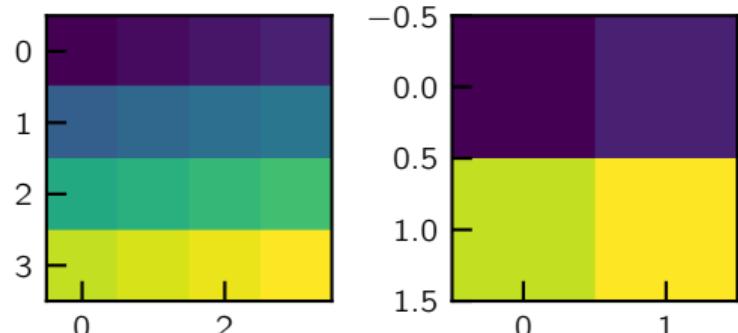


Abbildung: Verkleinern des Arrays durch Spline-Interpolation 2. Ordnung

Ergebnis der Skalierung

H L R I S

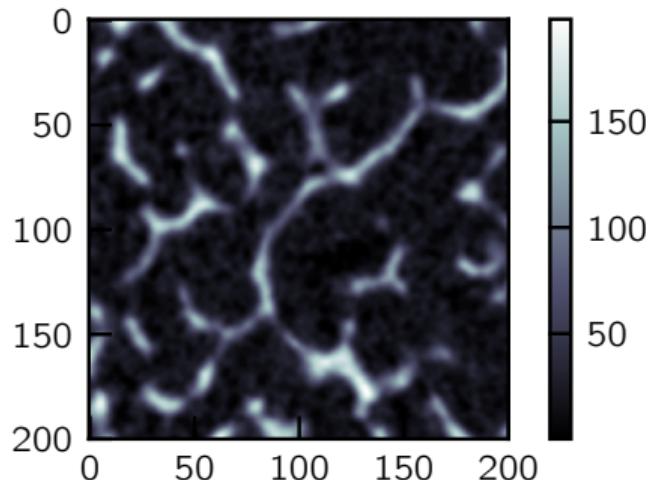


Abbildung: Datensatz nach der Glättung

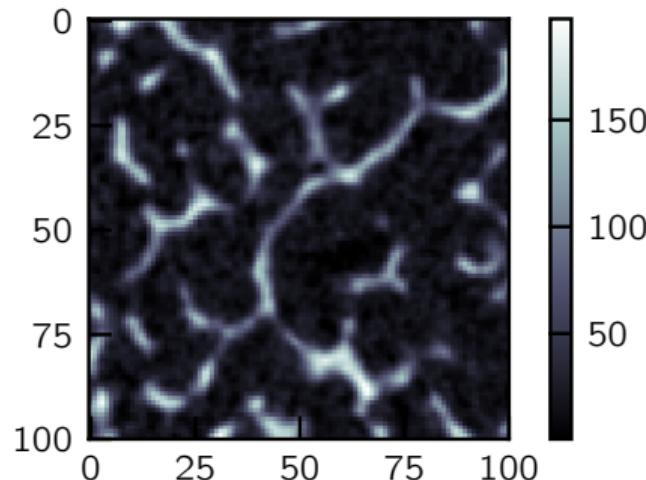


Abbildung: Datensatz nach der Skalierung

Ergebnis der Binarisierung

H L R I S

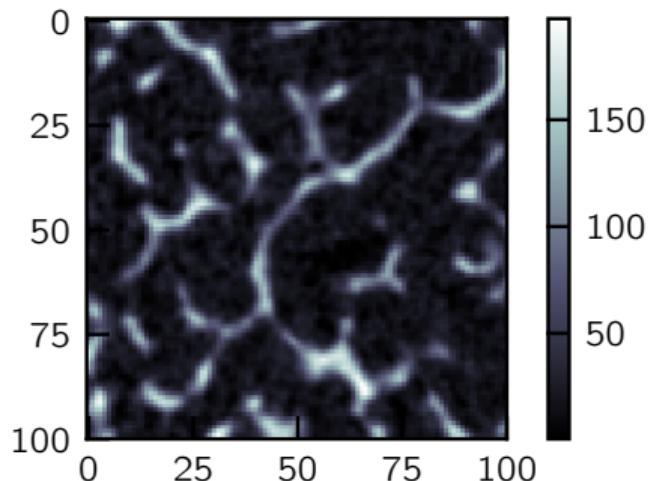


Abbildung: Datensatz nach der Skalierung

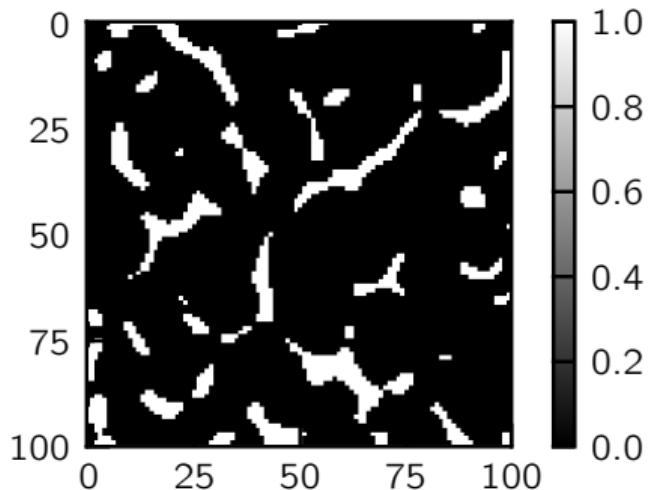


Abbildung: Datensatz nach der Binarisierung

- Morphologisch: die äußere **Gestalt** betreffend
- morphologische Operationen
 - Operationen auf der Gestalt von Objekten
 - setzt die Extraktion einer Gestalt voraus
 - in erster Linie Operation auf Segmenten (d.h., auf Binärbildern)
- Ziel von morphologischen Operationen
 - Veränderung der Gestalt, um Störungen nach einer Segmentierung zu beseitigen
 - Berechnung von Formmerkmalen
 - Suche nach bestimmten Formen (also: Analyse)

Definition Closing (Schließen)

Kombination von Dilatation gefolgt von einer Erosion mit einem am Ankerpunkt gespiegelten Strukturelement.

Ziel:

- **Dilatation:** Schließen von kleinen Löchern (kleiner als das Strukturelement)
- **Erosion:** Wiederherstellung der ursprünglichen Größe des Objekts

[10]

Beispiel Closing

H L R I S

```
1 import matplotlib.pyplot as plt
2 from skimage import data, morphology
3 from skimage.util import img_as_ubyte
4
5 phantom = img_as_ubyte(data.shepp_logan_phantom())
6
7 footprint = morphology.disk(6)
8 footprint = morphology.ball(6)
9
10 phantom[10:30, 200:210] = 0
11
12 closed = morphology.closing(phantom,
13     footprint=morphology.disk(6))
14 fig = plt.figure()
15 ax1 = fig.add_subplot(121)
16 ax2 = fig.add_subplot(122)
17 ax1.imshow(phantom, cmap="gray")
18 ax2.imshow(closed, cmap="gray")
19 plt.show()
```

Listing: *skimage_morphology_closing.py*

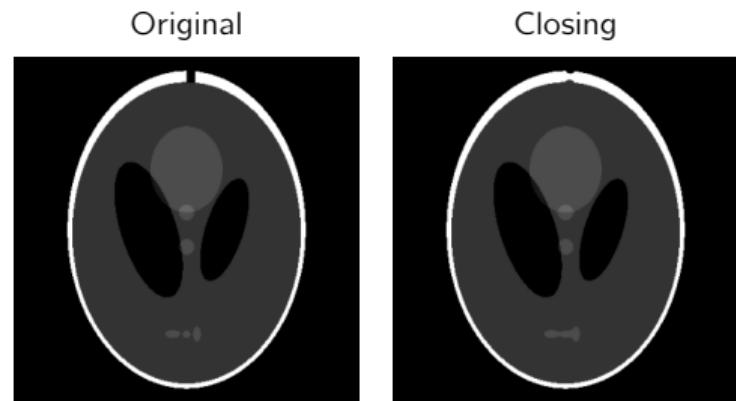


Abbildung: Weiße Ellipsen am unteren Rand, Riss am oberen Rand

Ergebnis des Closing

H L R I S

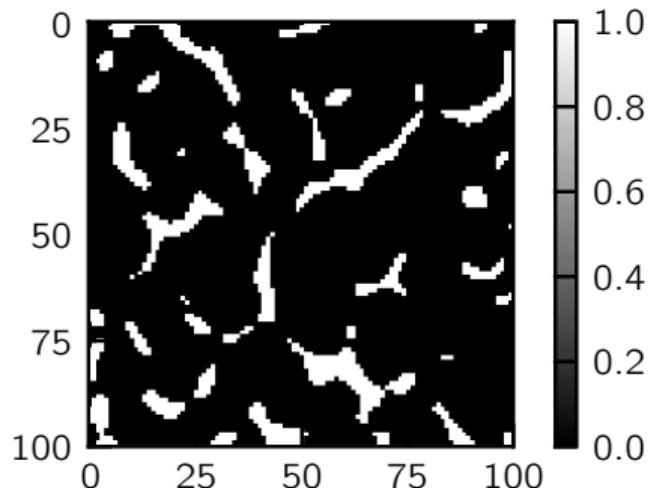


Abbildung: Datensatz nach der Binarisierung

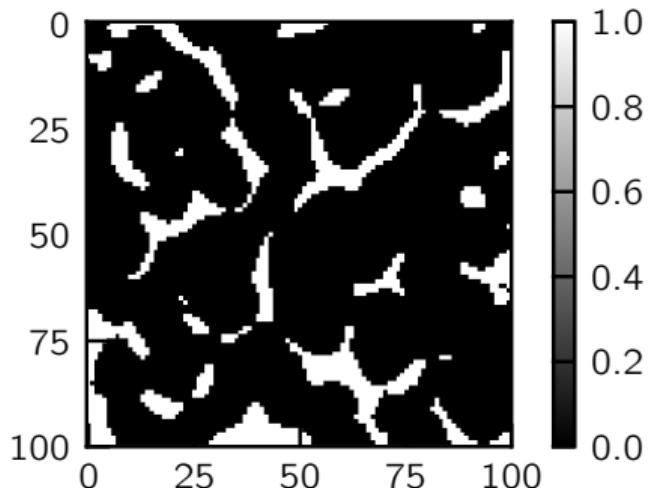


Abbildung: Datensatz nach dem Closing

1. Kennzeichnung zusammenhängender Bereiche (Labels) eines Integer-Arrays (`skimage.measure.label`)
2. Zählen der Häufigkeit der einzelnen Bereiche (Labels) (`numpy.bincount`)
3. Finden des größten zusammenhängenden Bereiches

[13]

```
1 from skimage import measure
2 import numpy as np
3
4 binary_image = np.array([
5 [0, 0, 1, 1, 0],
6 [0, 1, 1, 0, 0],
7 [0, 0, 0, 1, 1],
8 [1, 1, 0, 0, 0]]).astype(int)
9
10 labels, num = measure.label(binary_image, background=None, return_num=True, connectivity=1)
11
12 print(binary_image)
13 print(labels)
14 print(num)
```

Listing: *skimage_measure_label.py*

[13, 14]

Ergebnis des Clustern

H L R I S

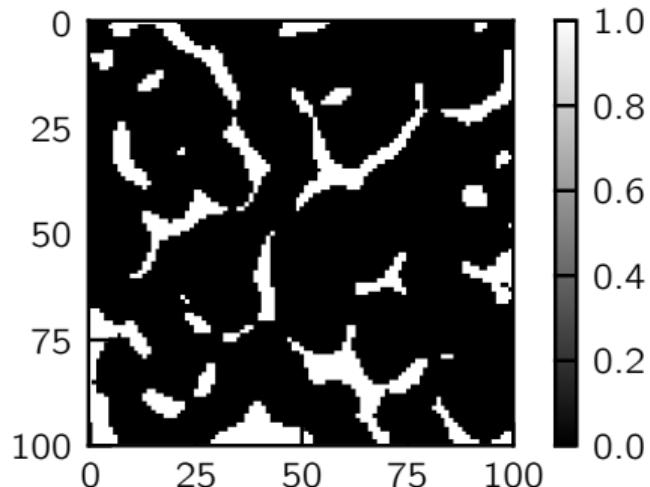


Abbildung: Datensatz nach dem Closing

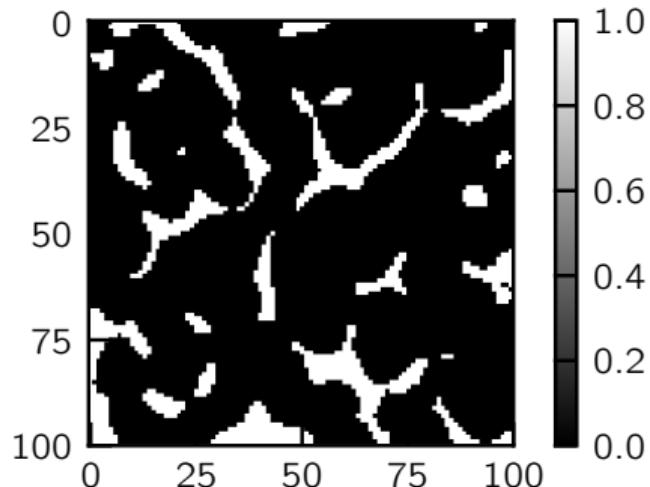


Abbildung: Datensatz nach dem Clustern

2. Treffen

Arbeitsauftrag

1. Skalierung des Datensatzes (`scipy.ndimage.zoom`)
2. Binarisieren des Datensatzes (`Vergleichsoperator >`)
3. Morphologisches Closing
(`skimage.morphology.closing`,
`skimage.morphology.ball`)
4. Bestimmen des größten isolierten Voxel-Clusters

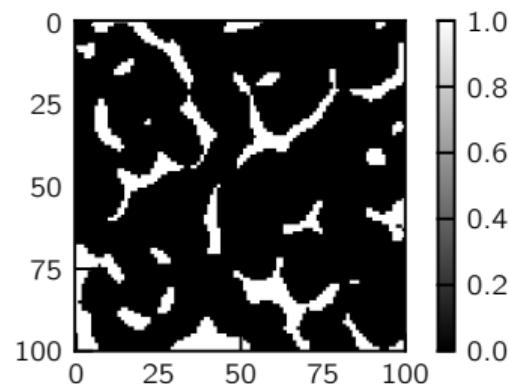


Abbildung: Binarisierter und zusammenhängender Datensatz

- Lernen Sie die Grundlagen von GitHub <https://docs.github.com/de/get-started/start-your-journey>
- Introduction to GitHub <https://github.com/skills/introduction-to-github>
- Vorlesung MSO I

Benjamin Schnabel, M.Sc.
High-Performance Computing Center Stuttgart (HLRS)

-  Mail benjamin.schnabel@hlrs.de
-  Phone +49 711 685-87288
-  GitHub <https://github.com/bennyschnabel>
-  ORCID 0000-0002-9882-0263

- [1] L Voo, M Armand und M Kleinberger. „Stress Fracture Risk Analysis of the Human Femur Based on Computational Biomechanics“. In: *Johns Hopkins APL Technical Digest* 25.3 (2004), S. 223–230. URL: <https://secwww.jhuapl.edu/techdigest/content/techdigest/pdf/V25-N03/25-03-Voo.pdf> (besucht am 04. 11. 2024).
- [2] T Vendiš. *CT-Scanner Siemens Naeotom Alpha*. 2022. URL: [https://commons.wikimedia.org/wiki/File:CT_Naeotom_Alpha_Pilsen_2022_\(cropped\).jpg](https://commons.wikimedia.org/wiki/File:CT_Naeotom_Alpha_Pilsen_2022_(cropped).jpg) (besucht am 05. 11. 2024).
- [3] SC Lee, HK Kim, IK Chun, MH Cho, SY Lee und MH Cho. „A flat-panel detector based micro-CT system: performance evaluation for small-animal imaging“. In: *Physics in Medicine and Biology* 48.24 (Dez. 2003), S. 4173–4185. DOI: 10.1088/0031-9155/48/24/014.
- [4] F Milena. *Medical Technology Laboratory*. Istituto Ortopedico Rizzoli. 24. Nov. 2022. URL: <https://www.ior.it/en/laboratori/lab-di-tecnolog-medica/medical-technology-laboratory> (besucht am 11. 11. 2024).
- [5] BC Bourne und MCH van der Meulen. „Finite element models predict cancellous apparent modulus when tissue modulus is scaled from specimen CT-attenuation“. In: *Journal of Biomechanics* 37.5 (Mai 2004), S. 613–621. DOI: 10.1016/j.jbiomech.2003.10.002.

- [6] JM Cox, JD Smith, MCH van der Meulen und JH Cole. „Heterogeneous Tissue Modulus Improved Prediction of Mechanical Behavior in Osteoporotic Vertebral Cancellous Bone“. In: (Dez. 2021). DOI: 10.1101/2021.11.30.470675.
- [7] Wikipedia. *Segmentierung (Bildverarbeitung)*. 13. Nov. 2024. URL: [https://de.wikipedia.org/wiki/Segmentierung_\(Bildverarbeitung\)](https://de.wikipedia.org/wiki/Segmentierung_(Bildverarbeitung)) (besucht am 05.12.2024).
- [8] Y Wu, S Adeeb und MR Doschak. „Using Micro-CT Derived Bone Microarchitecture to Analyze Bone Stiffness – A Case Study on Osteoporosis Rat Bone“. In: *Frontiers in Endocrinology* 6 (Mai 2015). DOI: 10.3389/fendo.2015.00080.
- [9] SciPy community. *scipy.ndimage.zoom*. Version 1.14.1. 2024. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.zoom.html> (besucht am 05.12.2024).
- [10] M Rohs und S Kratz. *Computergrafik 2. Morphologische Operationen*. Ludwig-Maximilians-Universität München, 2012. URL: <https://www.medien.ifi.lmu.de/lehre/ss12/cg2/vorlesung/CG2-S12-09-MorphologicalOperations.pdf> (besucht am 05.12.2024).

- [11] scikit-image team. *skimage.morphology*. Version 0.24. 2024. URL:
<https://scikit-image.org/docs/0.24.x/api/skimage.morphology.html> (besucht am 04.12.2024).
- [12] scikit-image team. *Morphological Filtering*. Version 0.24. 2024. URL:
https://scikit-image.org/docs/0.24.x/auto_examples/applications/plot_morphology.html (besucht am 05.12.2024).
- [13] scikit-image team. *skimage.measure*. Version 0.24. 2024. URL:
<https://scikit-image.org/docs/0.24.x/api/skimage.measure.html> (besucht am 05.12.2024).
- [14] scikit-image team. *Segment human cells (in mitosis)*. Version 0.24. 2024. URL:
https://scikit-image.org/docs/0.24.x/auto_examples/applications/plot_human_mitosis.html (besucht am 05.12.2024).