
Maschinelles Lernen in der Medizinphysik

Alexander Ratke

26. März 2021

Bachelor-Programmierkurs

Tagesablauf

Grundlagen zur Bildgebung und -bearbeitung

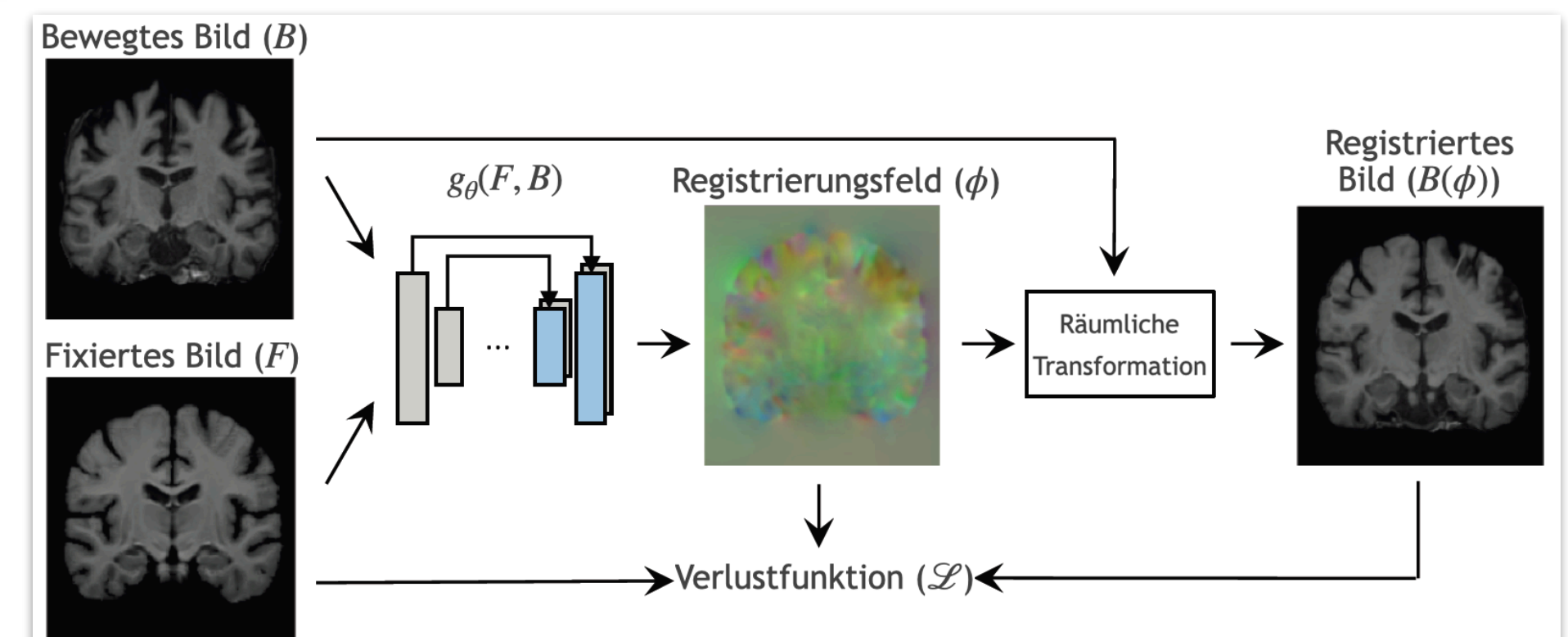
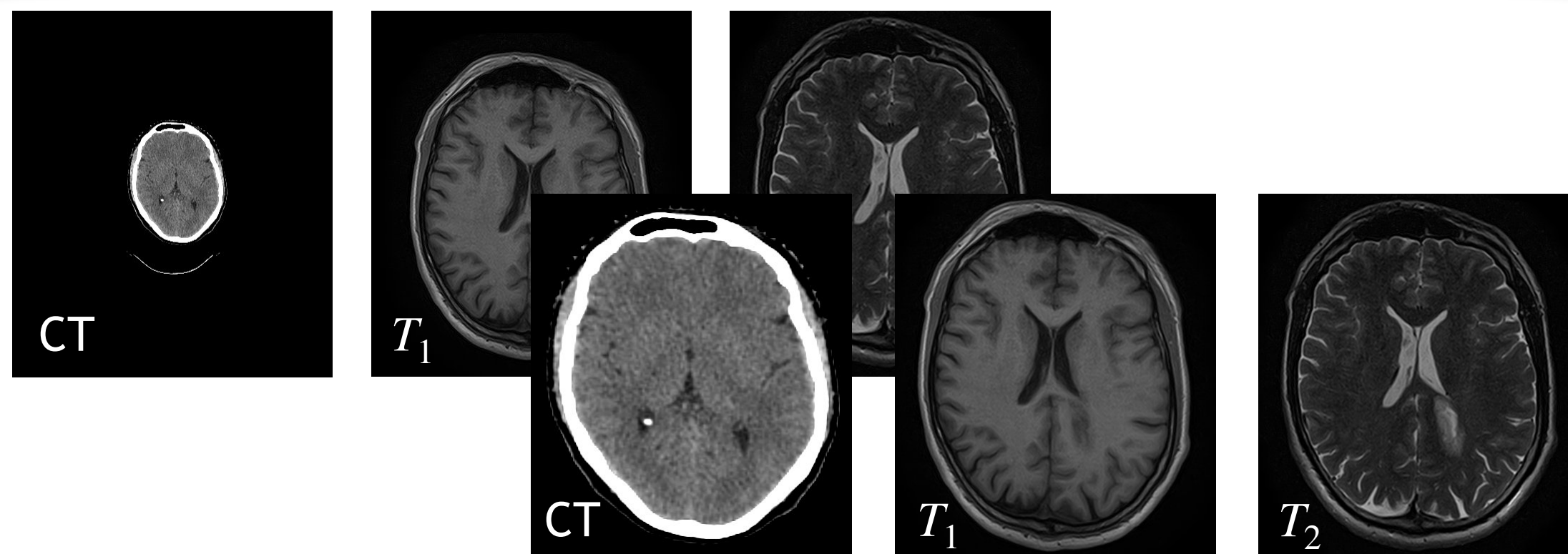
08:30 - 11:00

- Check von Hardware und Software
- Einführung in die Bildgebung
- Arbeiten mit Aufnahmen im DICOM-Format
 - ➔ 1. Aufgabe
- Einführung in die Bildbearbeitung
 - ➔ 2. Aufgabe

Grundlagen zur Bildregistrierung mit ML

13:00 - 16:00

- Einrichtung und Einweisung in Git
- Installation notwendiger Python-Pakete
- Einführung in die Bildregistrierung
 - ➔ 3. Aufgabe
 - ➔ 4. Aufgabe

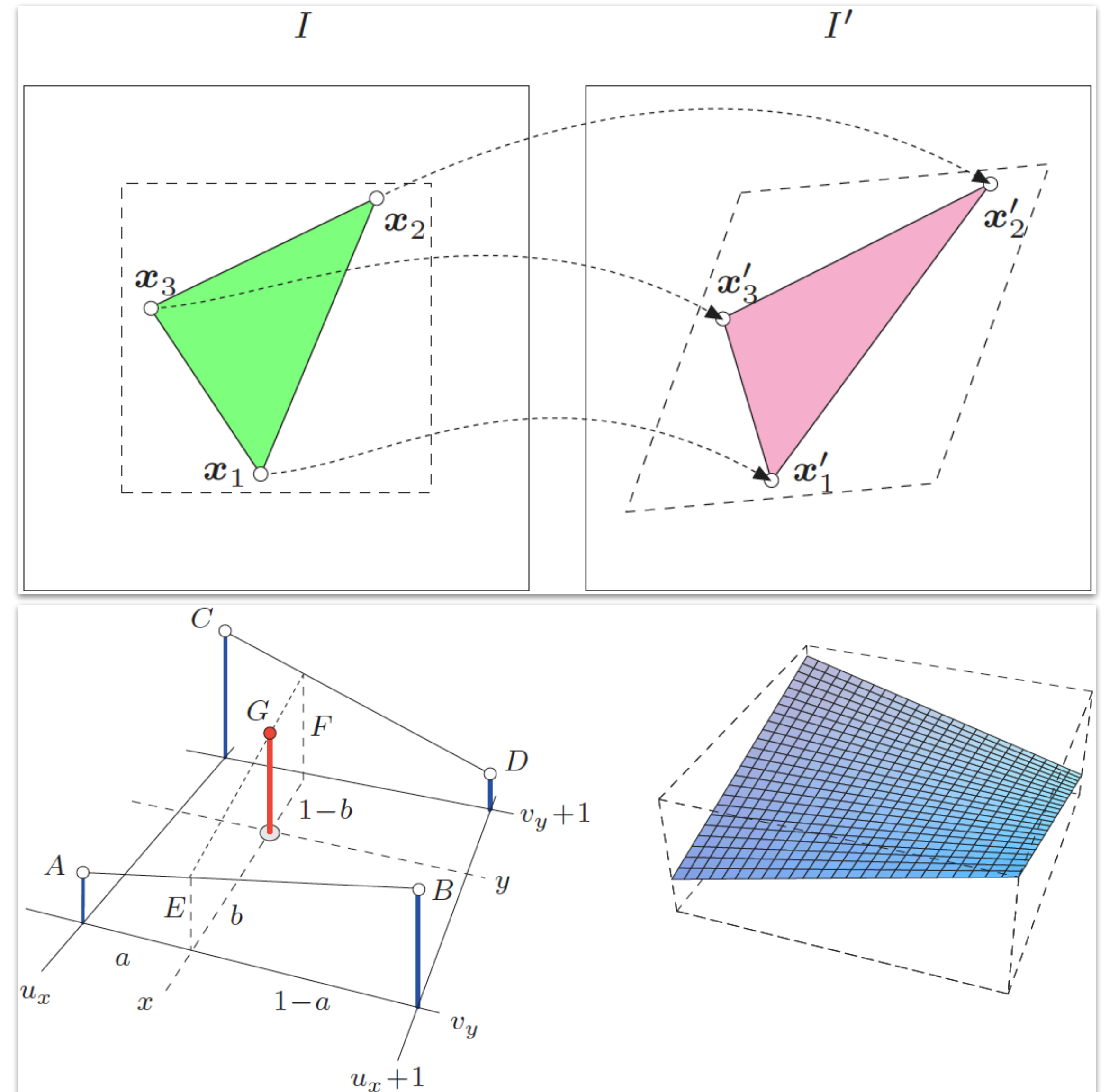
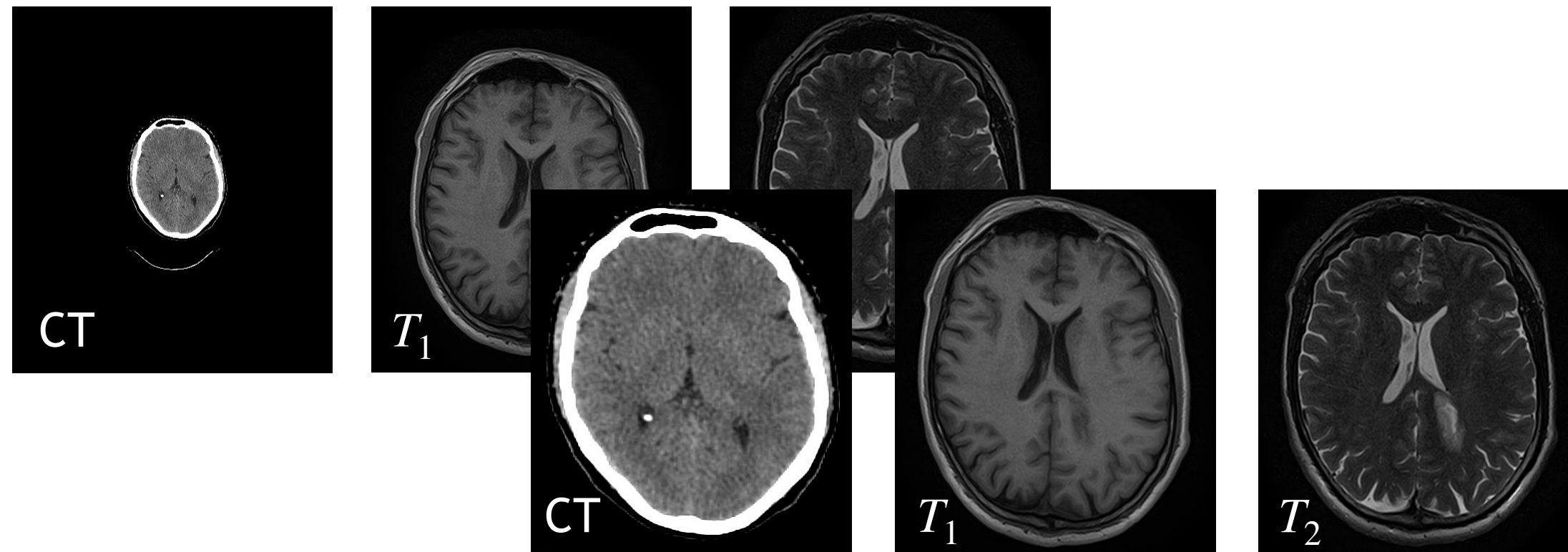


Teil 1

Grundlagen zur Bildgebung und -bearbeitung

08:30 - 11:00

- Check von Hardware und Software
- Einführung in die Bildgebung
- Arbeiten mit Aufnahmen im DICOM-Format
 - ➔ 1. Aufgabe
- Einführung in die Bildbearbeitung
 - ➔ 2. Aufgabe



Check-Up

- Login via ssh:

- Zugang auf die interaktive Maschine bernhard
- Aktivierung von Anaconda zum Verwenden von python durch

```
source /usr/scripts/set_conda.sh
```

```
source activate root_forge
```

- Befehle in `.bashrc` schreiben zur automatischen Aktivierung bei jedem login

- Mounting via sshfs:

- home-Verzeichnis

```
sshfs USER@login.e5.physik.tu-dortmund.de:/net/nfshome/home/USER /PFAD/MOUNTORDNER -o volname=MOUNTORDNER
```

- ceph-Verzeichnis

```
sshfs USER@login.e5.physik.tu-dortmund.de:/ceph/users/USER /PFAD/MOUNTORDNER -o volname=MOUNTORDNER
```

- Notwendige python-Pakete mit `pip install --user {package_name}`

- *Image Processing in Python*: `scikit-image`

Einleitung

Fusionierte Bildgebung mittels maschinellen Lernens

- Verschmelzung von mindestens zwei Objekten
- Hier: anatomische Bilder
- Entscheidung über die Kombinationsparameter

- Computer- und Magnetresonanztomographie
- Verschiedene Modalitäten bieten unterschiedliche Vorteile

- „Deep Learning“-Architektur zur Bildregistrierung
- Hier: *Convolutional Neural Network*
- Anpassung von Strukturen

Strahlentherapie

Ziel: Präzise Lokalisation von Tumorgewebe in der Protonentherapie und Schonung von gesundem Gewebe

Methode:

- Separate Betrachtung von CT und MRT
- Verwendung von CT bei Therapieplanung wegen gewebsspezifischen Absorptionskoeffizienten

Herausforderung:

- Kombination: Absorptionskoeffizienten der CT-Aufnahme und Strukturinformationen von MRT
- Anpassung der Merkmale von CT und MRT durch Bildregistrierung und anschließende Fusion
- Fusionierte CT-Aufnahme zur genaueren Therapieplanung

Computertomographie

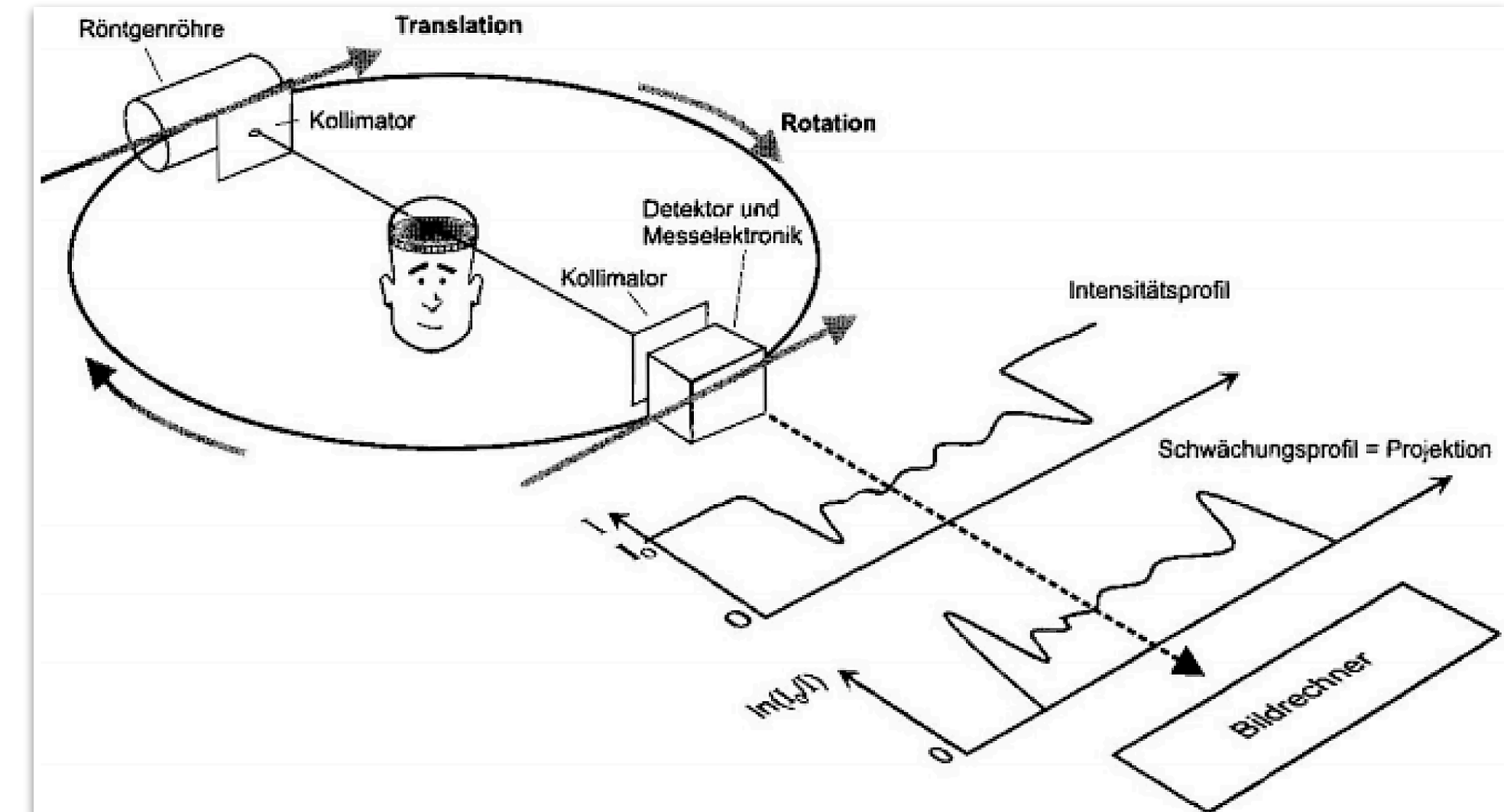
Grundlagen

- Verwendung von Röntgenstrahlung zur Visualisierung der menschlichen Anatomie
- Beschreibung der abgeschwächten Röntgenstrahlen durch

$$I(u, \gamma) = I_0 \cdot \exp \left(- \int \int \mu(\vec{x}) \delta(\vec{x} \cdot \vec{e}_\gamma - u) d\vec{x} \right)$$

- Darstellung von Knochen und Weichteilgewebe
- Rotation der Röntgenröhre erzeugt Querschnittsbilder

Abschwächungs-
koeffizient



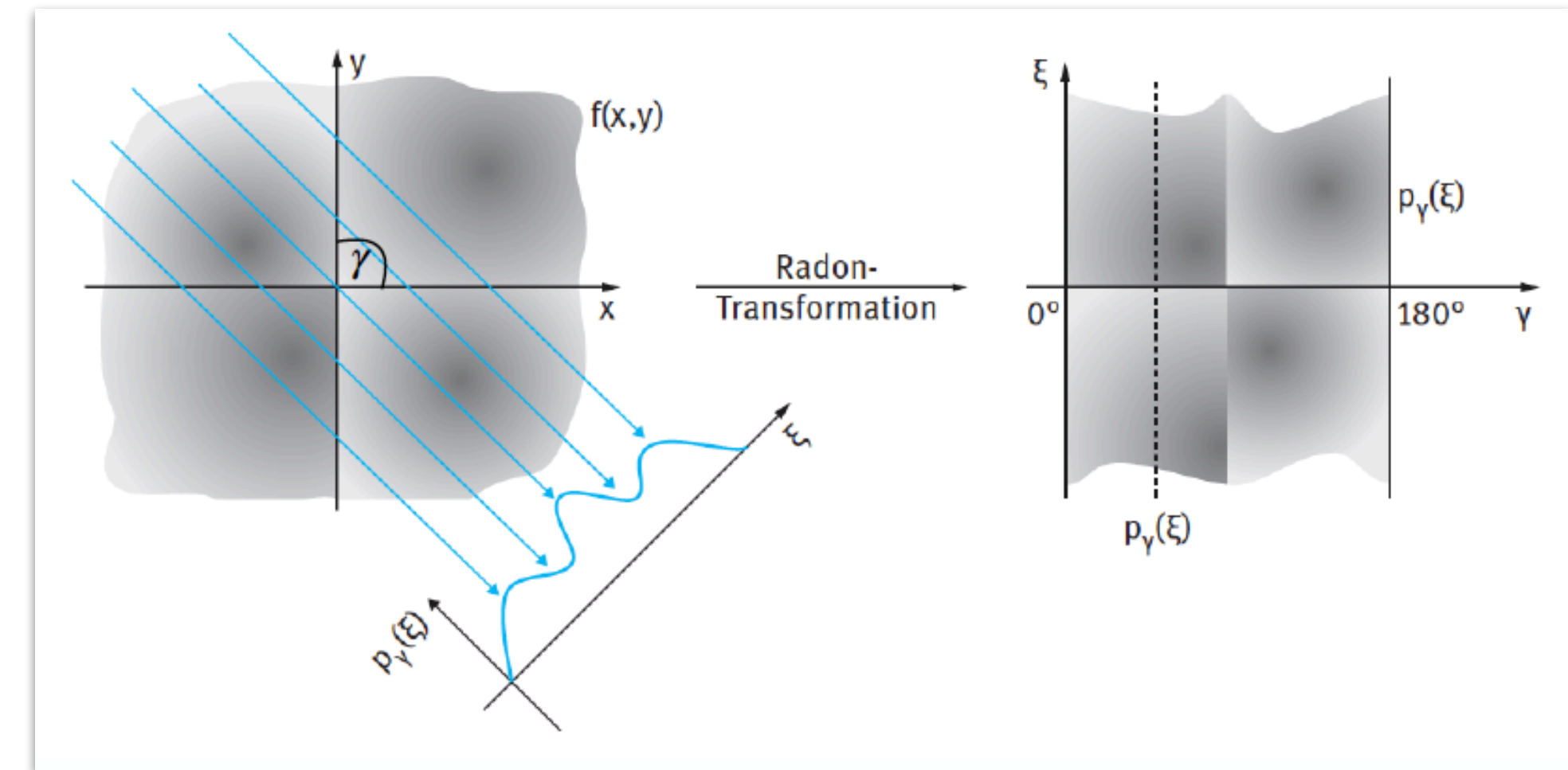
A. Kalender: „Computertomographie: Grundlagen, Gerätetechnologie, Bildqualität, Anwendungen.“

Radontransformation

- Darstellung der Projektion durch Radontransformierte

$$p_\gamma(\xi) = - \ln \left(\frac{I(u, \gamma)}{I_0} \right) = \int \int \mu(\vec{x}) \delta(\vec{x} \cdot \vec{e}_\gamma - u) d\vec{x}$$

- Sinogramm: Darstellung aller Projektionen in Abhängigkeit vom Winkel γ



T. M. Buzug et al.: „Computertomographie“. In: Medizinische Bildgebung

Computertomographie

Gefilterte Rückprojektion zur Bildrekonstruktion

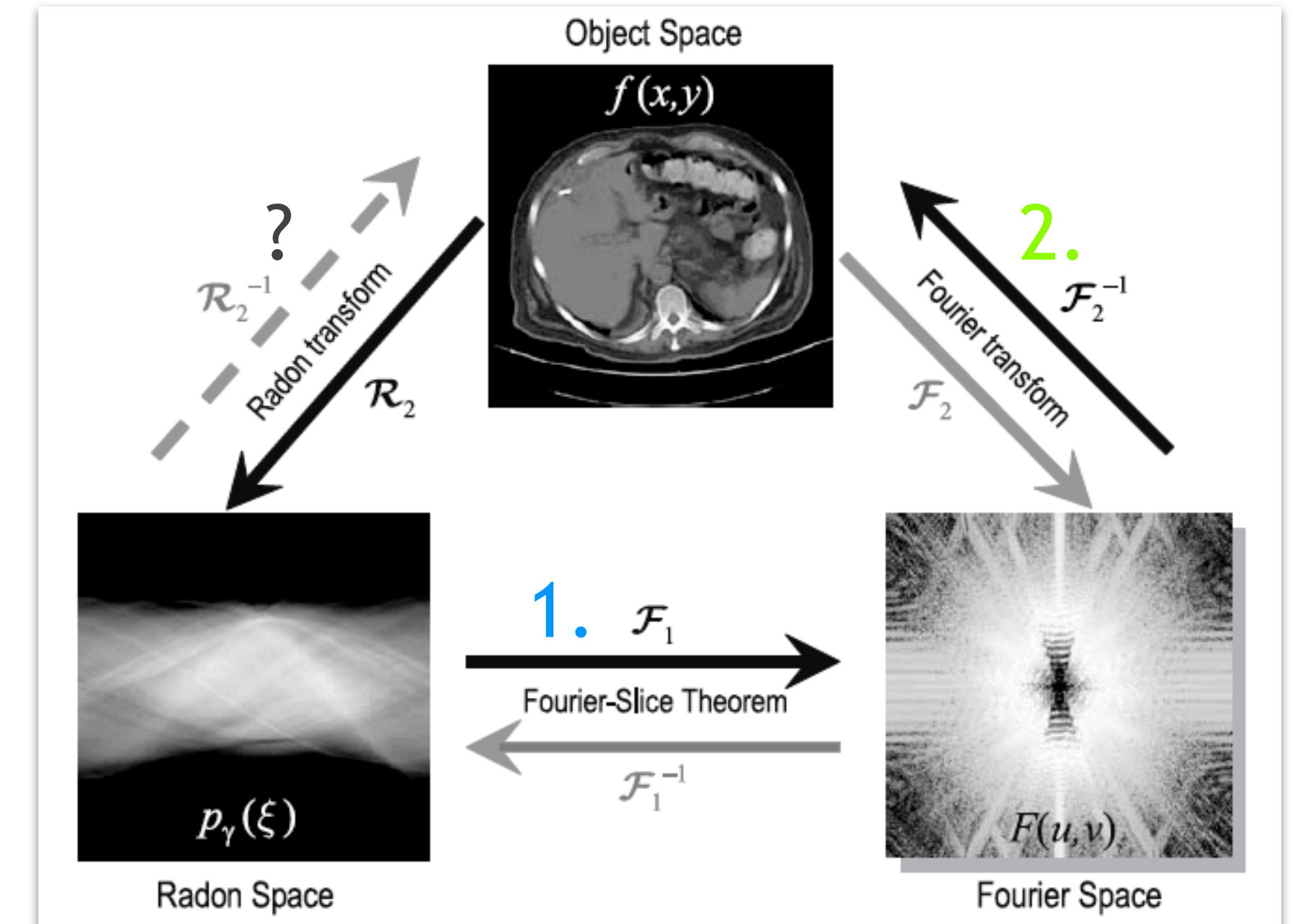
1. Fourier-Scheiben-Theorem
 - ▶ Überführung jeder Projektion in den Fourier-Raum
 - ▶ 2D-Schicht $F(u, v)$ wird nach und nach gefüllt
2. Inverse Fourier-Transformation
 - ▶ Rekonstruktion von $f(x, y) \rightarrow \mu(x, y)$
 - ▶ Filterung im Fourier-Raum für Variation der Bildschärfe

Hounsfield-Skala

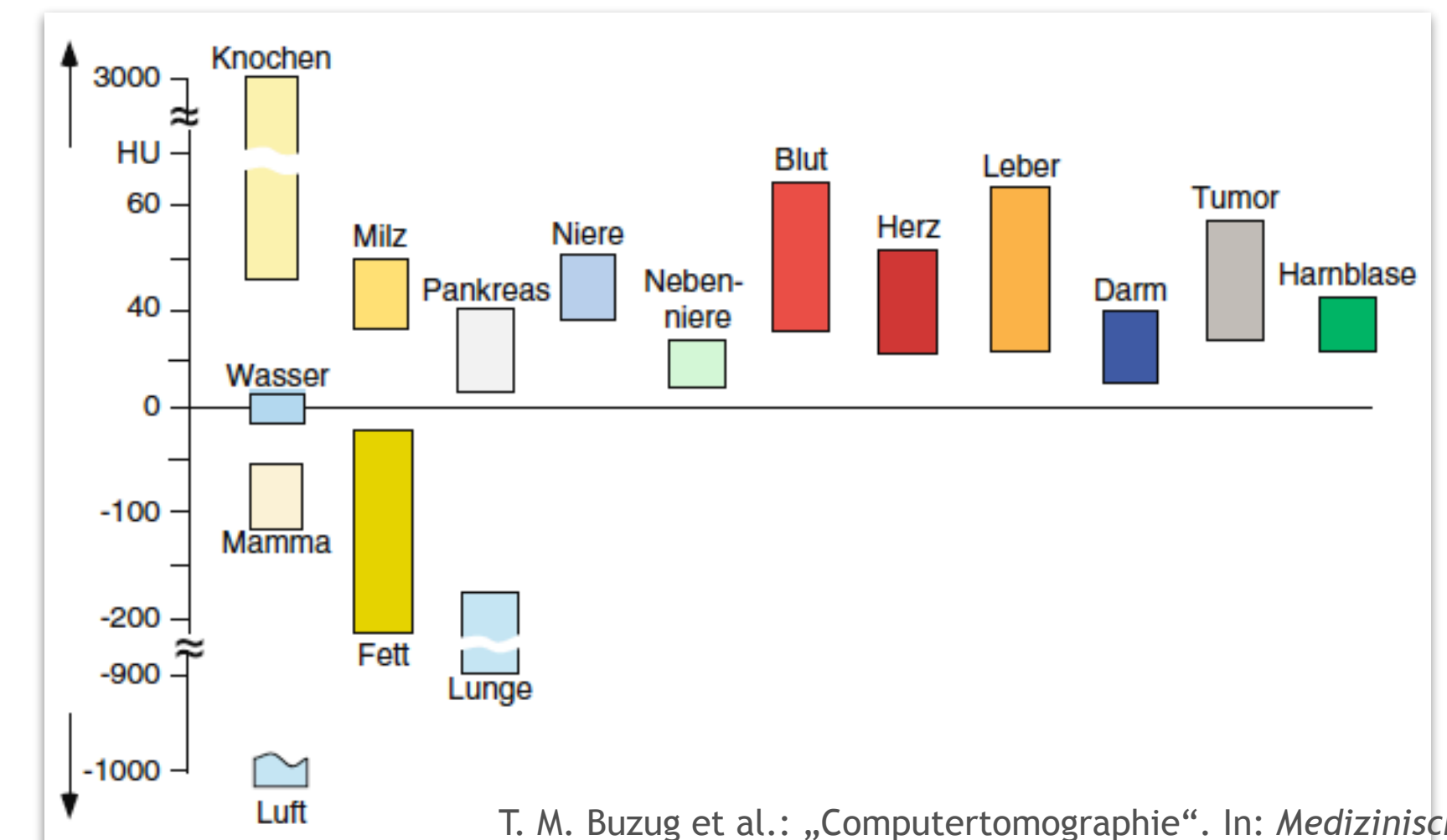
- CT-Werte durch Abschwächungskoeffizienten von Gewebe (G) normiert auf Wasser (W)

$$\text{CT-Zahl} = \frac{\mu_G - \mu_W}{\mu_W} \cdot 1000\text{HU}$$

- Stärkste Intensitätsabschwächung im Knochenfenster
- Geringste Abschwächung in Luft

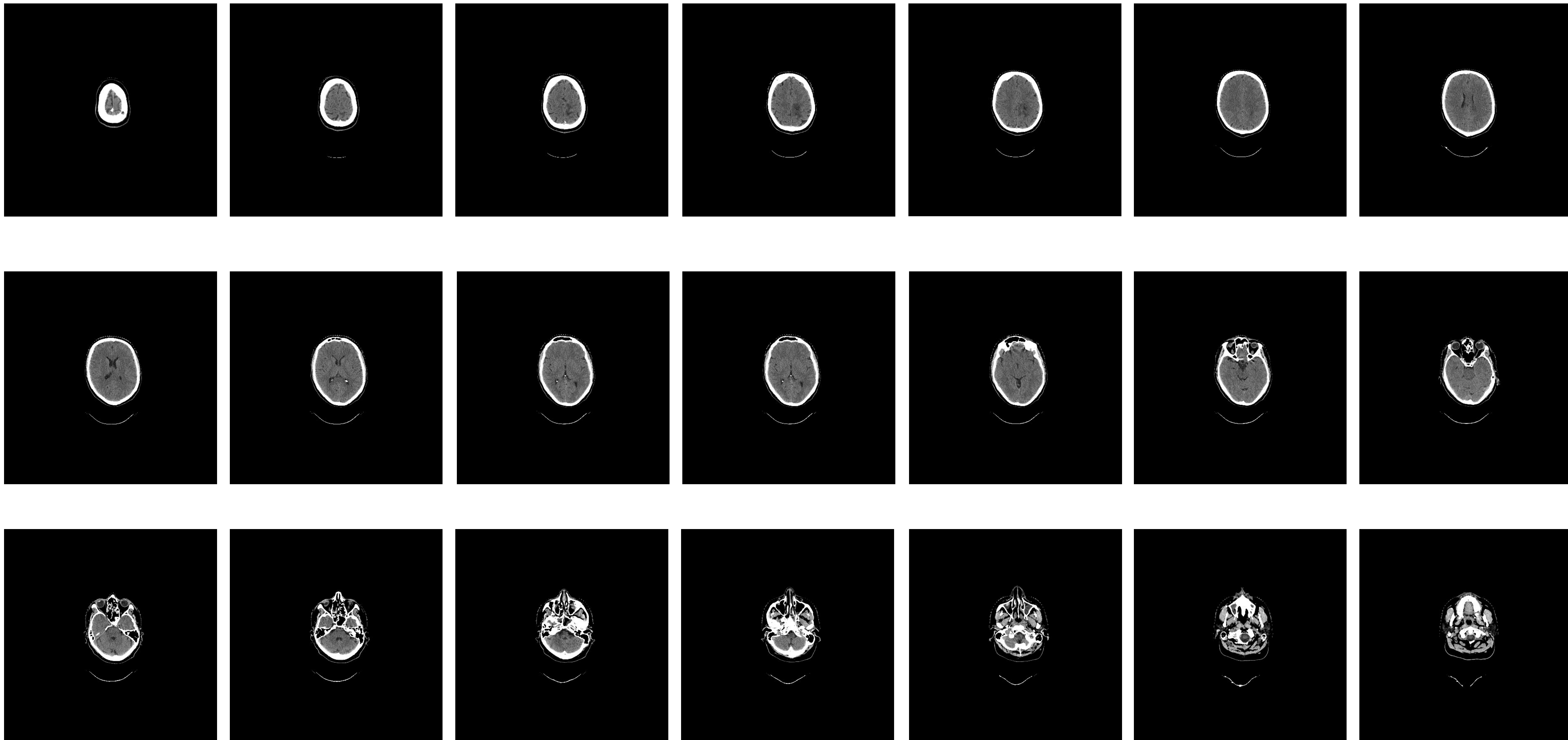


T. M. Buzug: „Computed Tomography“. In: Two-Dimensional Fourier-Based Reconstruction Methods



T. M. Buzug et al.: „Computertomographie“. In: Medizinische Bildgebung

CT-Aufnahmen



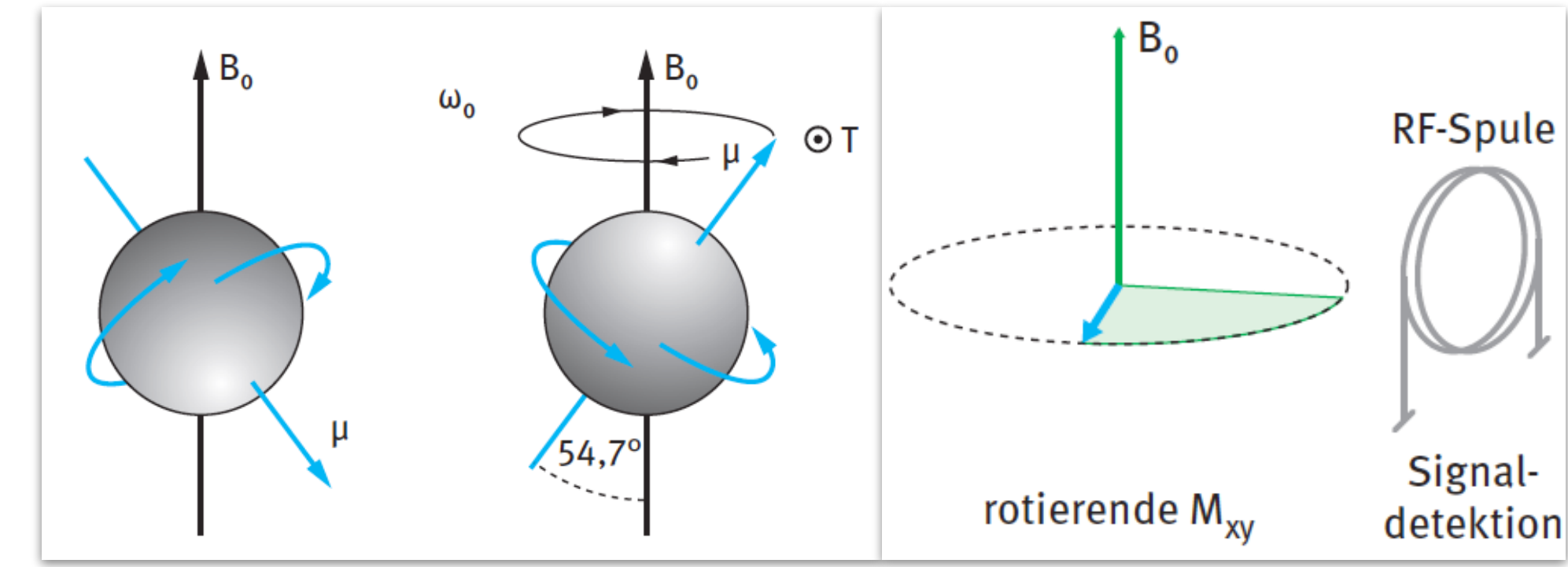
Magnetresonanztomographie

Grundlagen

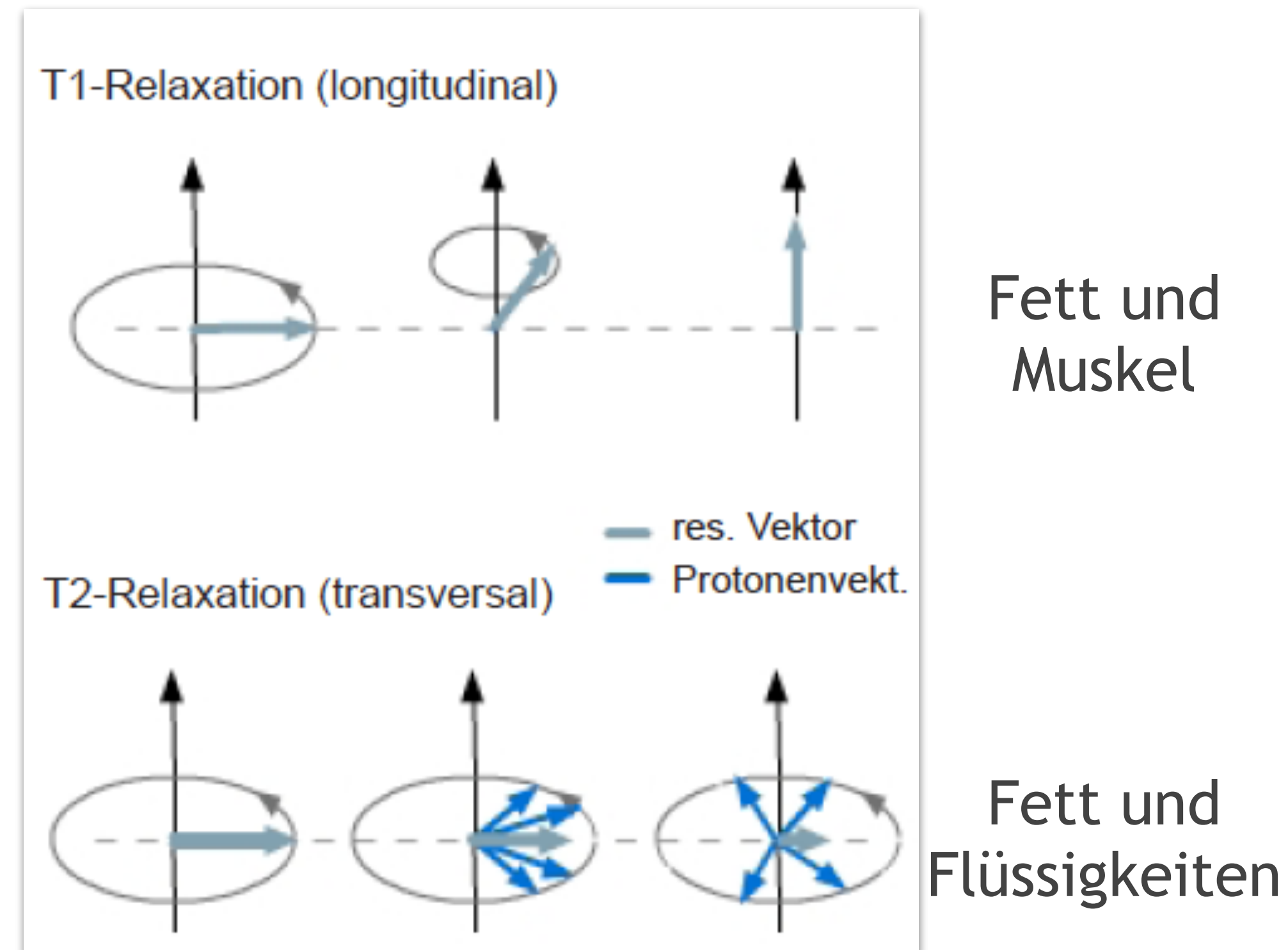
- Beeinflussung des Kernspins von Protonen in Wasserstoff durch starke Magnetfelder ($B_0 = 1,5 - 3T$)
 - Ausrichtung magnetischer Momente entlang und entgegengesetzt zu B_0
 - Larmorpräzession mit der Frequenz $\omega_0 = \gamma \cdot B_0$
 - Überschuss für Signalmessung relevant
- Hohe Auflösung für Weichteilgewebe

Bildkontraste

- Spin-Spin-Relaxation (T_1): Longitudinale Magnetisierung
- Spin-Gitter-Relaxation (T_2): Transversale Magnetisierung
- Kontrasteinstellung mit
 - Repetitionszeit (T_R)
 - Echozeit (T_E)



T. Schaeffter et al.: „Magnetische Resonanztomographie“. In: Medizinische Bildgebung

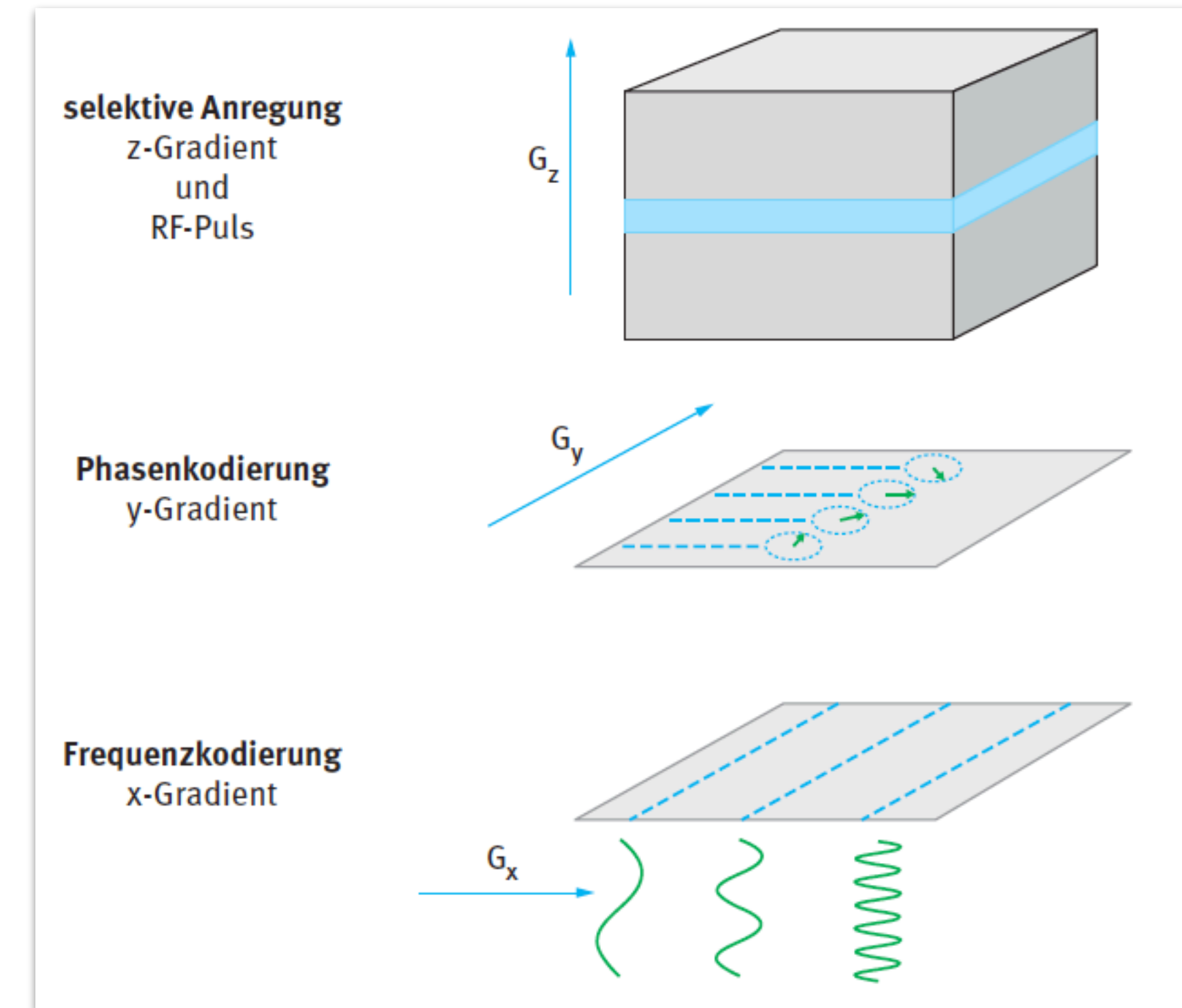


Dr. med. C. Pabst: „Grundlagen der Magnetresonanztomographie“. In: Magnetresonanztomographie

Magnetresonanztomographie

Bildrekonstruktion

- Ortskodierung:
 - Auswahl einer Schicht mit zusätzlichem z-Gradienten
 - Einstellung der Schichtdicke mit Gradientensteigung
- Phasenkodierung:
 - Verwendung von y-Gradienten
 - Zeilenweise Phasenverschiebung der Spins
- Frequenzkodierung:
 - Verwendung von x-Gradienten
 - Reihenweise Frequenzänderung für Signalmessung
- Bezeichnung der Rohdatenmatrix als k-Raum

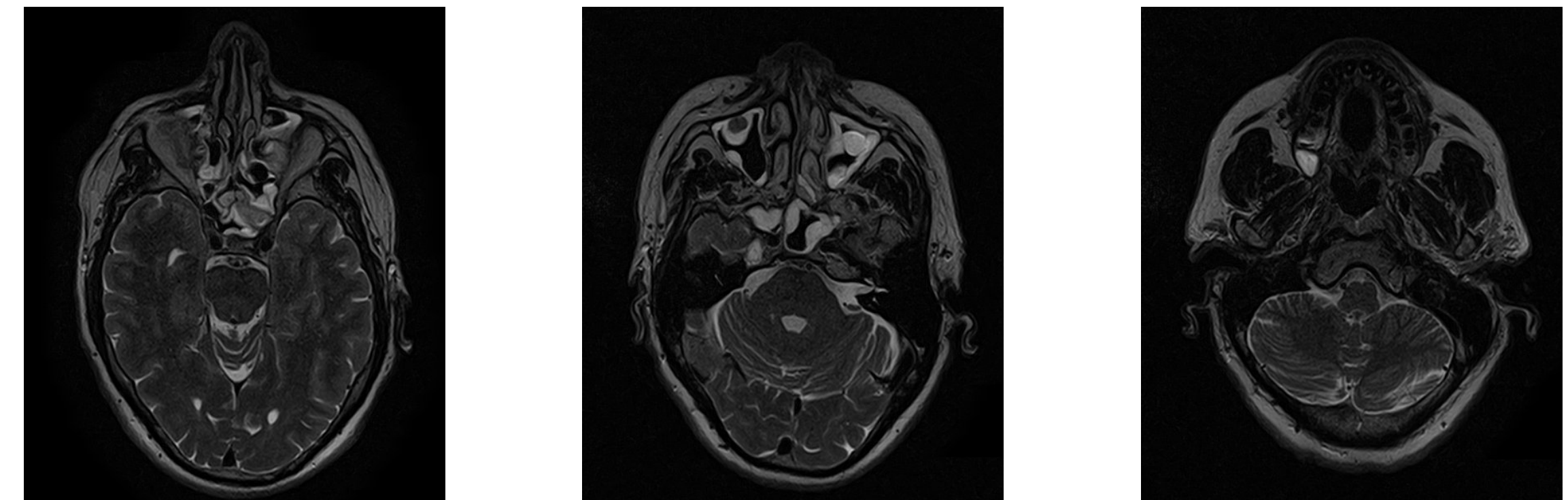
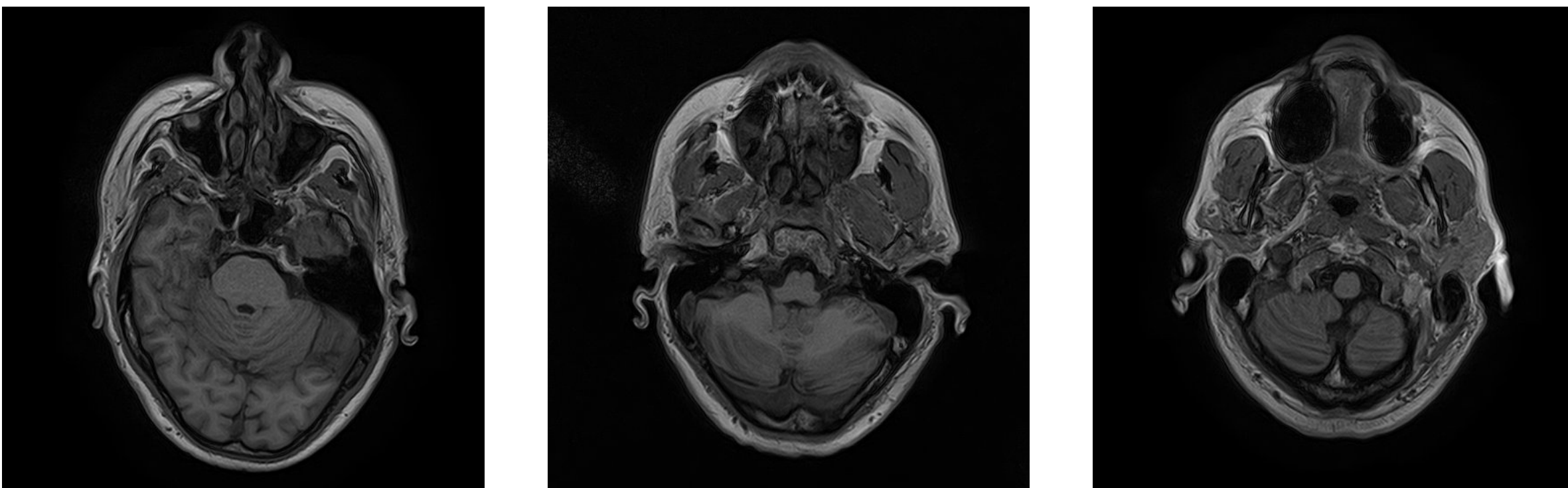
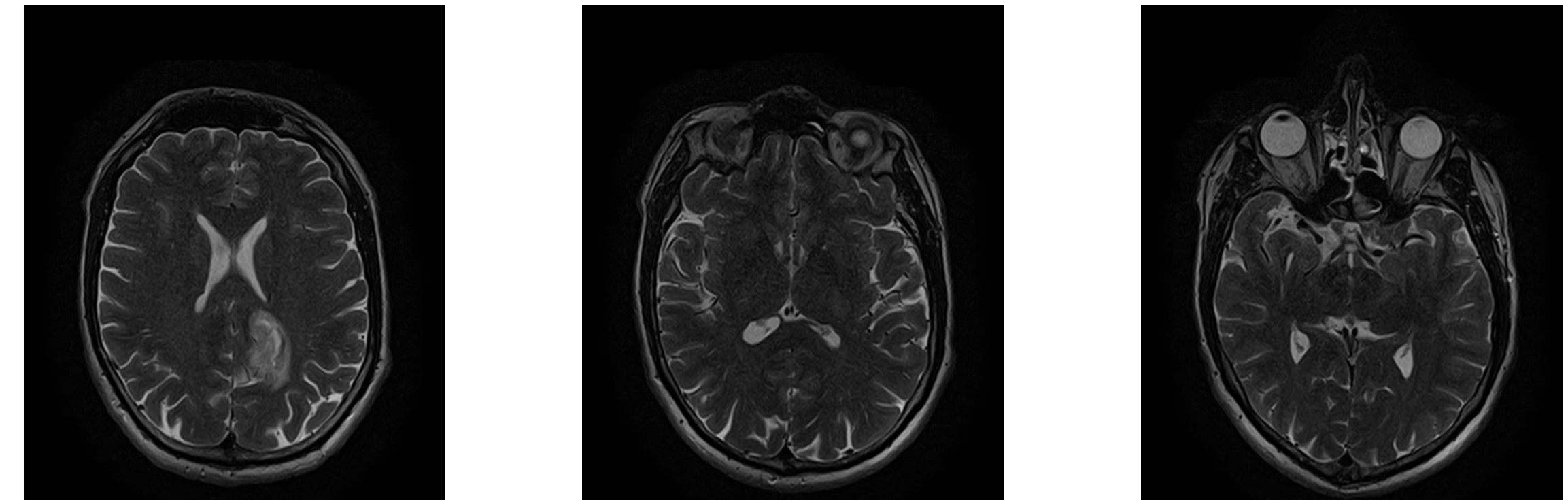
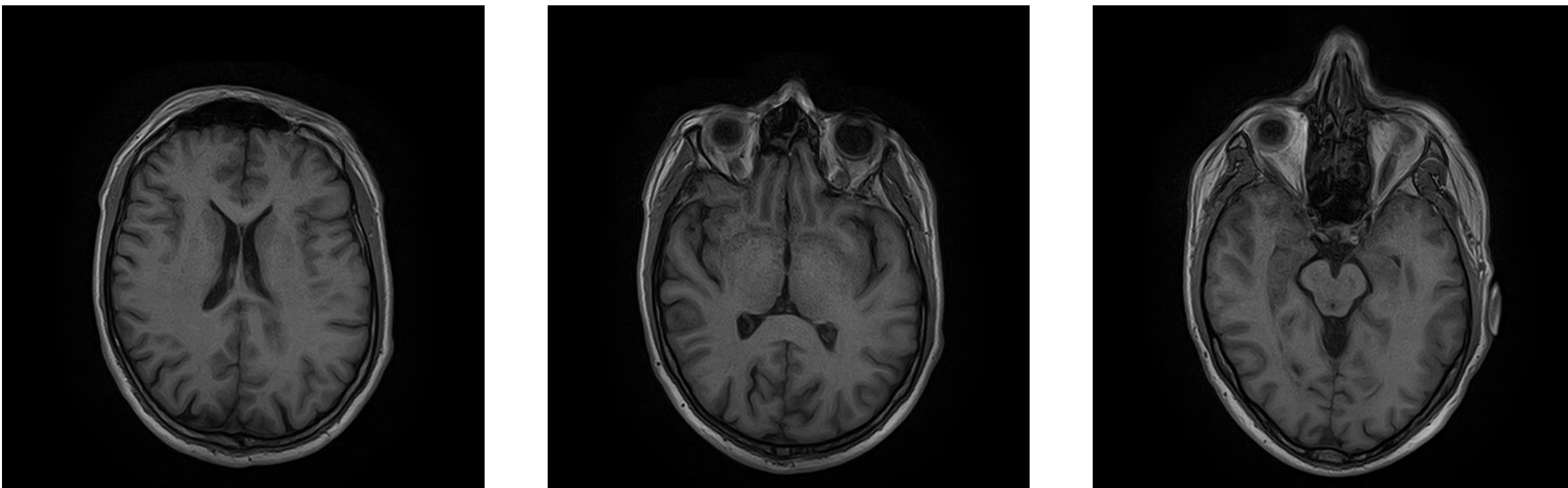
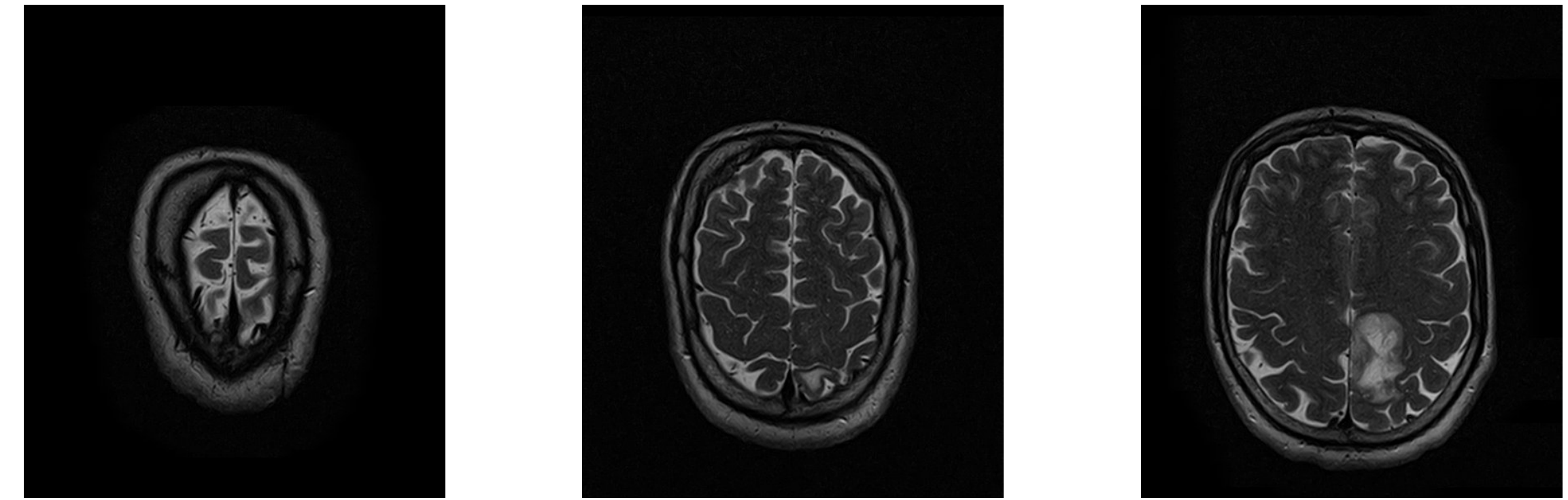
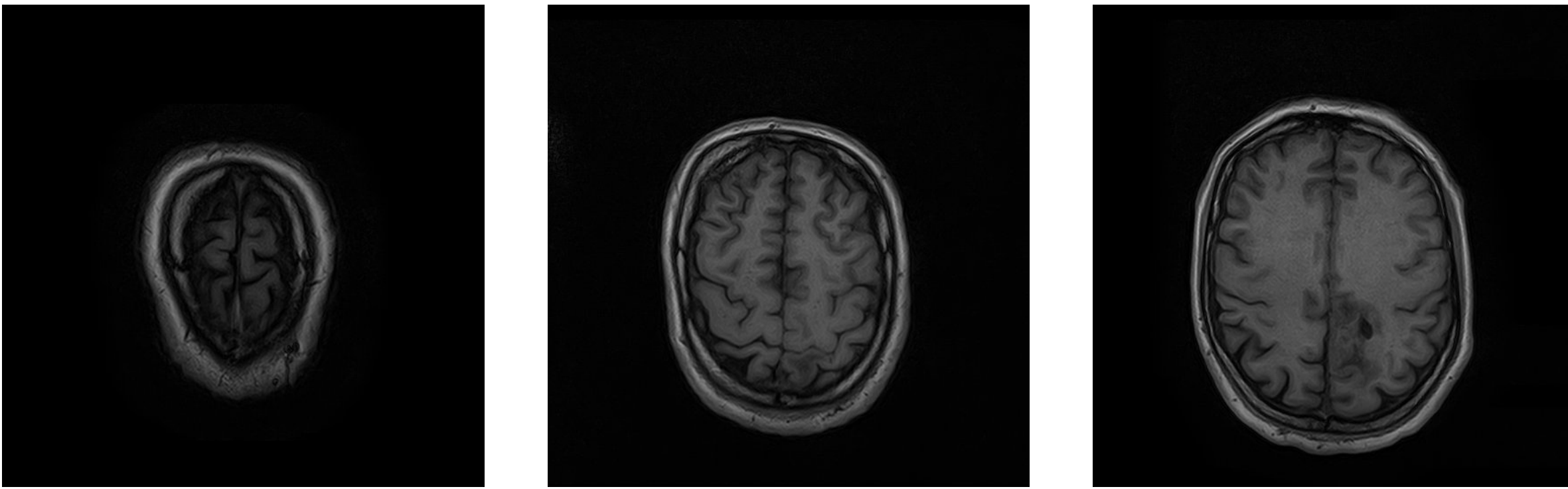


A. Hendrix: „Magnete, Spins und Resonanzen“.

Messverfahren pro Schicht:
Bild mit 512x512 Pixeln benötigt 512 gemessene Frequenzreihen

T_1 -Aufnahmen

T_2 -Aufnahmen



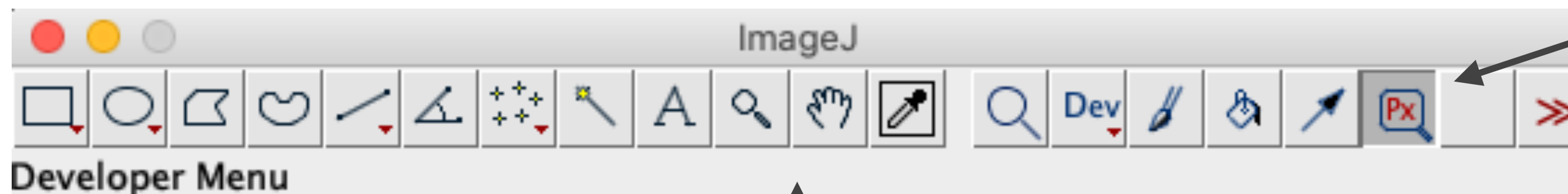
DICOM-Format

Allgemein:

- Standardisiertes Format: *Digital Imaging and Communications in Medicine*
- Speicherung von Informationen über Patienten, Aufnahmeeinstellungen, Geräteparameter etc.

ImageJ

- Visuelle Betrachtung von .dcm-Dateien
- Hauptfenster zur Steuerung:



Pixel Inspection Tool

- Öffnen eines Pixelfensters
- Größe des Fensters variabel bei *Prefs*

Per Drag-and-Drop öffnen

- Einzelne Dateien
- Ganzer Ordner für gestapelte Ansicht

A screenshot of the 'Pixel Values' window in ImageJ. It displays a grid of numerical values. The first column is labeled 'Prefs' and the subsequent columns are numbered 235 through 243. The value 239 in the first row and 4062 in the second row are highlighted in red.

	235	236	237	238	239	240	241	242	243	
Prefs	255	3704	3637	3405	3096	2881	3063	3503	3808	3705
	256	4451	3989	3091	2523	2604	3191	3810	4118	3979
	257	4695	3932	2693	2228	2842	3748	4274	4393	4253
	258	4183	3482	2592	2568	3476	4324	4568	4441	4221
	259	3487	3223	3096	3409	4062	4473	4462	4184	3831
	260	3274	3460	3841	4122	4198	4109	3999	3763	3416
	261	3571	3844	4190	4222	3912	3572	3486	3456	3298
	262	3829	3787	3846	3825	3605	3287	3201	3347	3451
	263	3654	3328	3269	3477	3605	3392	3170	3282	3531

- Hilfreiche Informationen abrufbar durch [Menü -> Image -> Show Info]
 - DICOM Standard Browser ([Link](#)) als Hilfe für die Bezeichnungen

Verwendung von DICOM in python

```
import pydicom as pyd          ## Importieren von pydicom

img = pyd.dcmread('datei.dcm') ## Öffnen/Laden der DICOM-Datei
arr = img.pixel_array         ## Darstellung der Datei als numpy array

new_arr = arr + 2             ## Bild/Array wird modifiziert/verändert zu new_arr

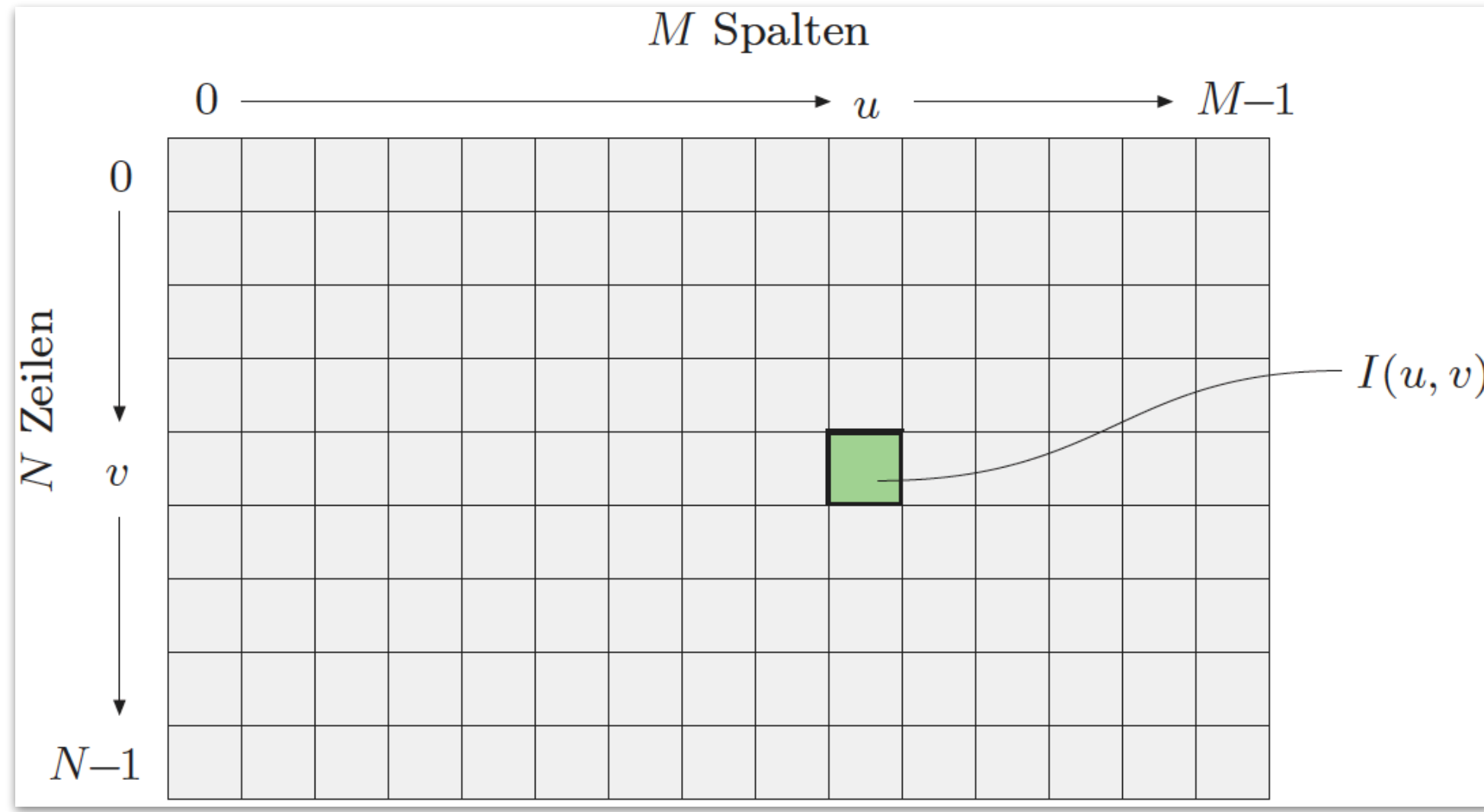
img.PixelData = new_arr.tobytes() ## Neues array wird in bytes umschreiben
img.Rows, img.Columns = new_arr.shape ## Reihen und Spalten durch die Form des neuen Bildes definieren
img.save_as('neue_datei.dcm') ## Speichern der neuen Aufnahme
```

1. Aufgabe:

- Download und Installation von ImageJ ([Link zur Website](#))
- Betrachte einzelne und mehrere DICOM-Dateien mit ImageJ durch eine Mount-Verknüpfung zu `/ceph/users/aratke/programmierkurs/ImageProcessing/`
- Fragen:
 - Suche nach den drei Raumkoordinaten der 15ten T_2 -Schicht!
 - Wie groß sind T_R und T_E in T_2 ? Wie ist die Bestrahlungsdauer im CT pro Schicht und insgesamt?
 - Finde die Schichtdicke der CT- und T_2 -Aufnahme heraus!

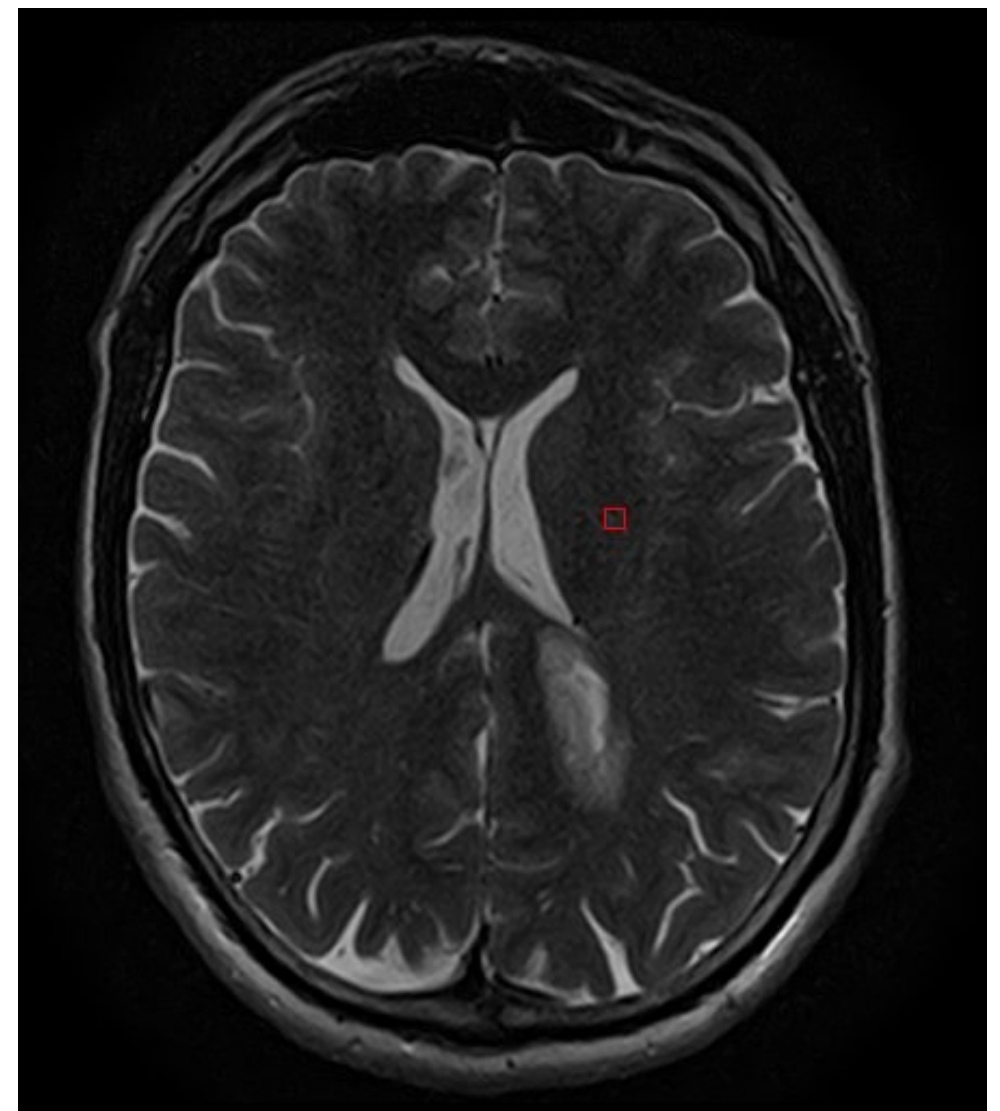
Grundlagen zu Bildmatrizen

- Bestehend aus vielen Bildelementen (Pixel) mit diskreten Werten
 - Farbbilder: RGB-Kanal
 - Grauwertbilder: 1-Kanal-Intensitäten
- Größe definiert durch
 - Breite (Anzahl der Spalten, M)
 - Höhe (Anzahl der Zeilen, N)
- Auflösung wird angegeben durch
 - Anzahl der Pixel pro Längeneinheit
 - *dots per inch* (dpi)



W. Burger and M. J. Burge: „Bildverarbeitung“, 2015

Prefs	271	272	273	274	275	276	277	278	279
231	1176	1072	976	924	1000	1202	1438	1499	1235
232	1280	1173	1042	872	800	975	1357	1626	1442
233	1395	1461	1358	1046	763	833	1267	1627	1509
234	1383	1684	1705	1427	1141	1094	1305	1526	1453
235	1194	1644	1862	1787	1593	1428	1364	1397	1414
236	1011	1448	1805	1904	1773	1537	1354	1314	1393
237	1012	1337	1667	1768	1636	1462	1377	1364	1360
238	1252	1433	1603	1573	1413	1364	1455	1471	1280
239	1513	1562	1613	1507	1342	1327	1425	1401	1173



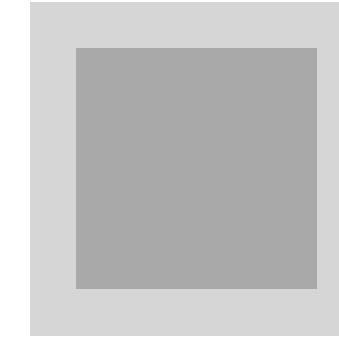
Grundlagen zu Bildmatrizen

- Geometrische Operationen:

Verschiebung



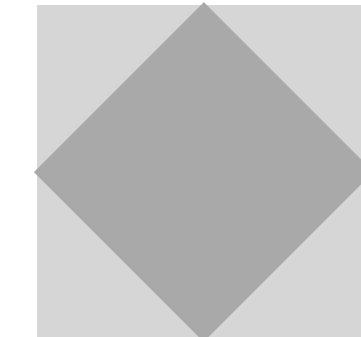
Skalierung



Scherung



Rotation



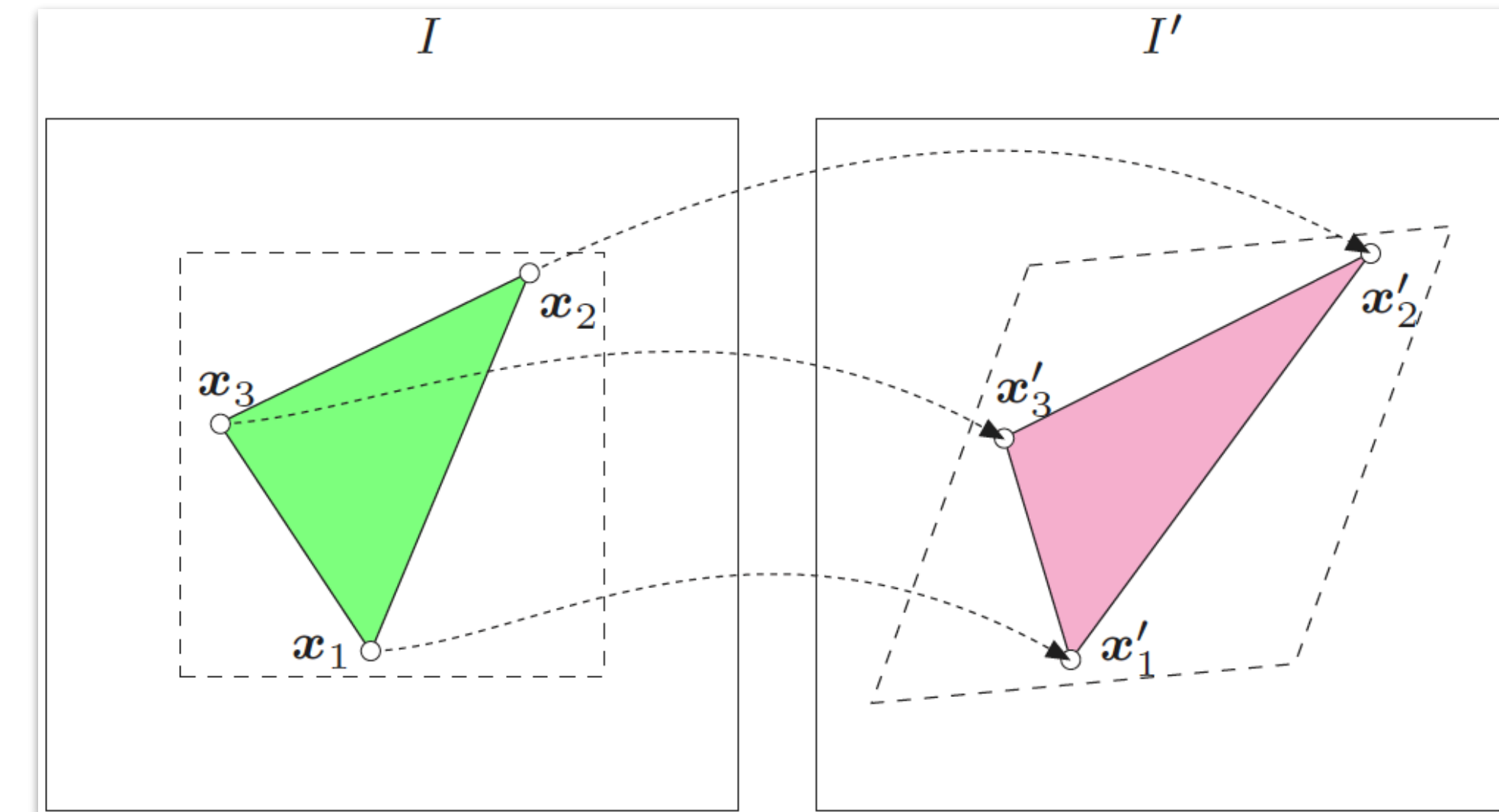
- Affine Abbildung:

$$T = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

- Koordinatentransformation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- ▶ Berechnung neuer Koordinaten \vec{x}' für jeden Pixel in I
- ▶ Mindestens drei Punkte \vec{x}_1 , \vec{x}_2 und \vec{x}_3 notwendig

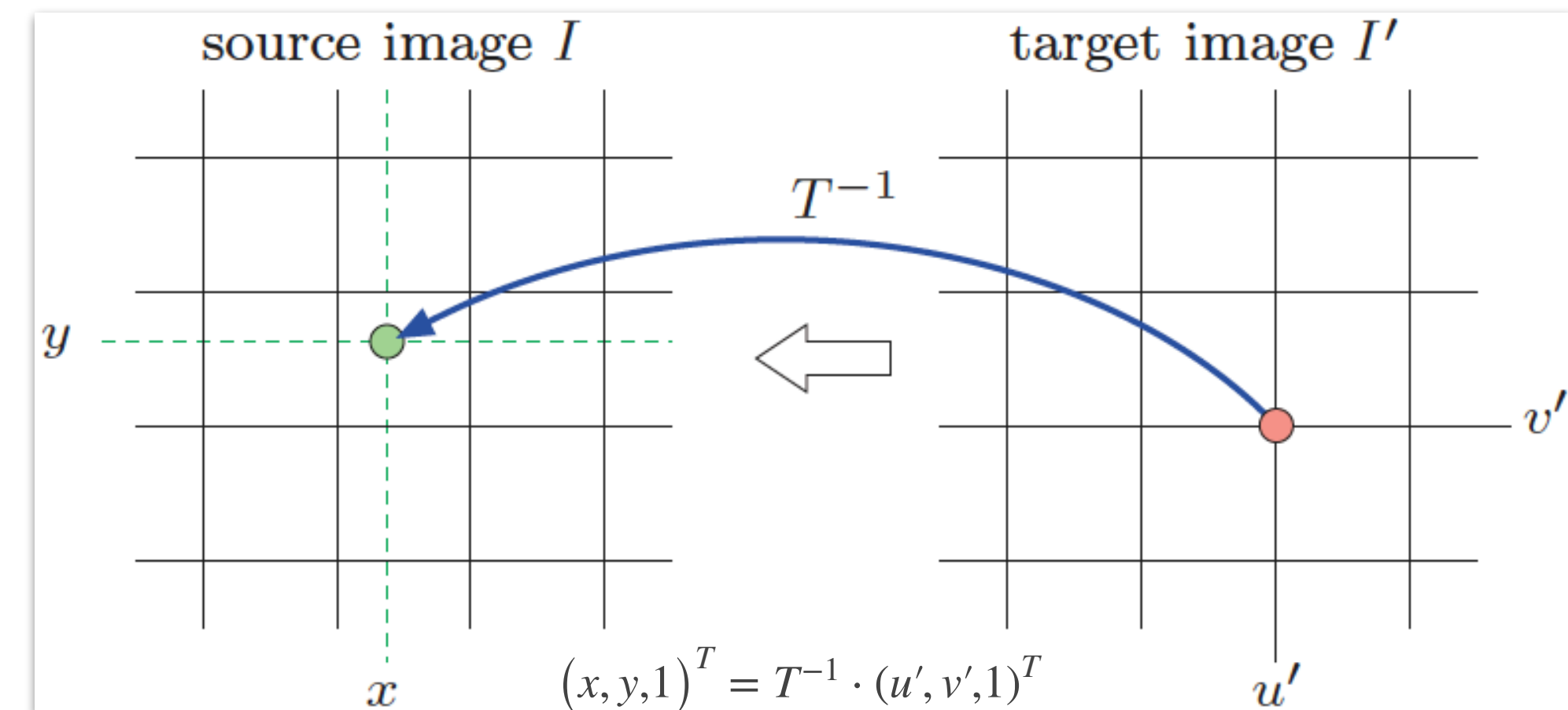
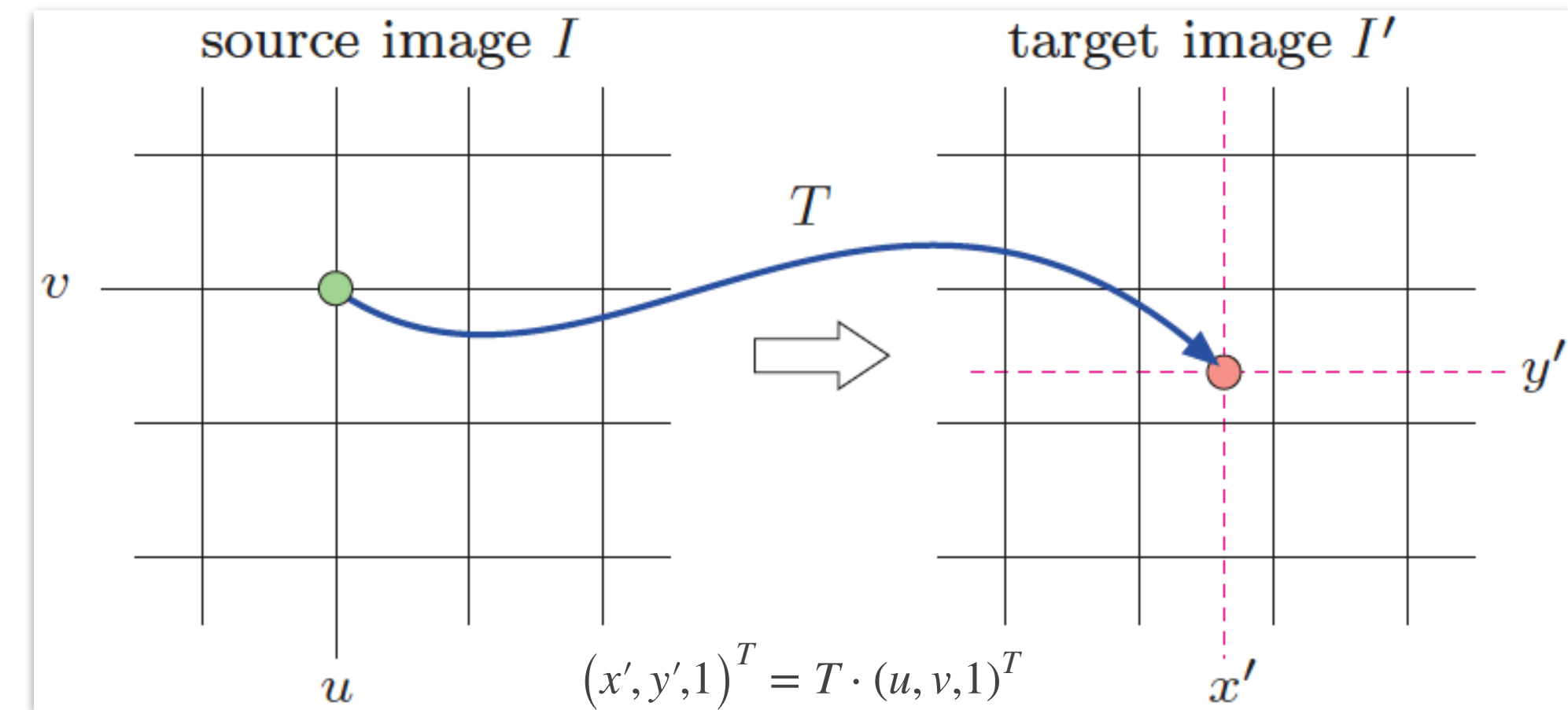


Quelle: „Bildverarbeitung“, W. Burger and M. J. Burge, 2015

Welche Grauwerte werden im neuen Bild I' verwendet?

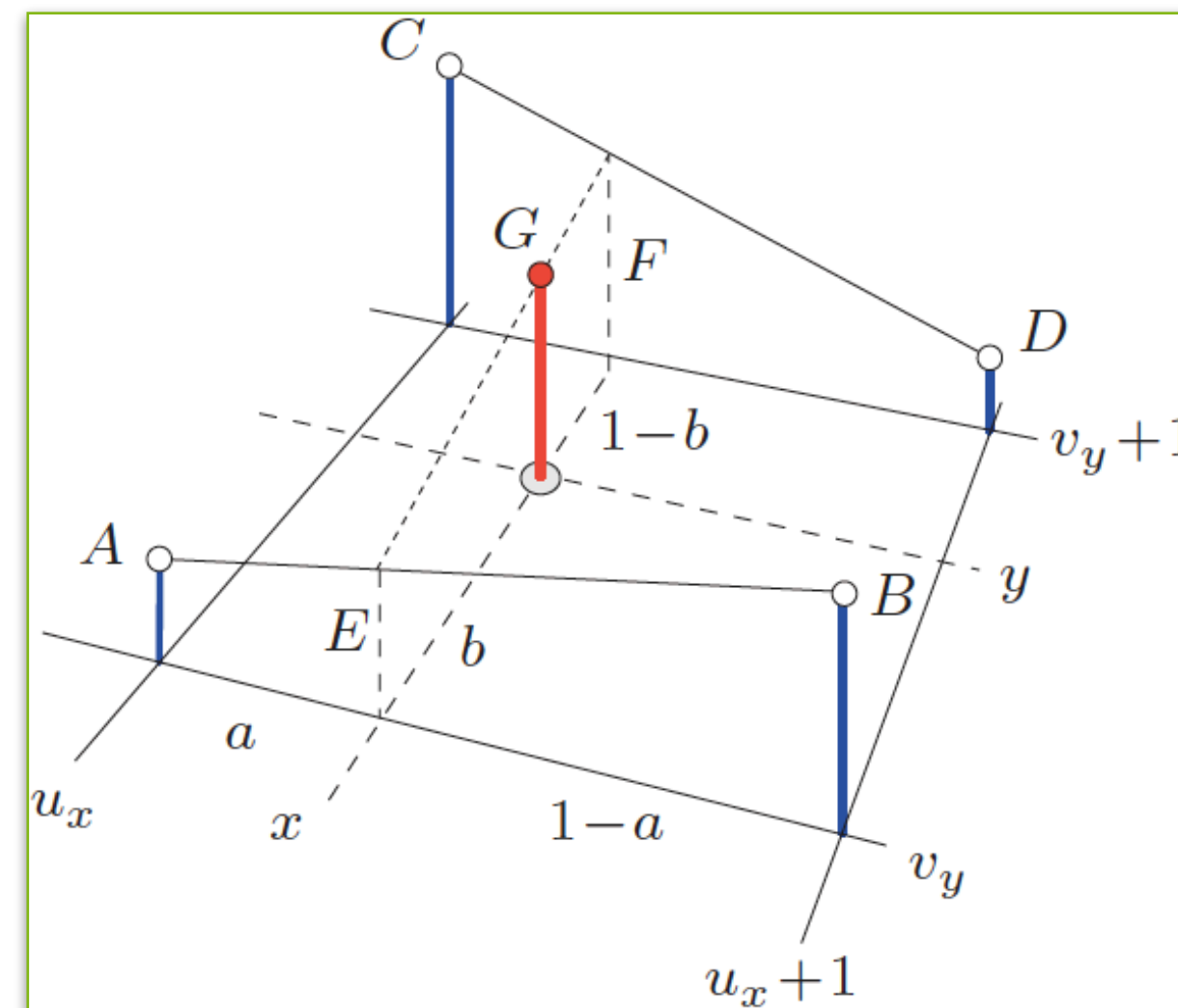
Grundlagen zu Bildmatrizen

- Source-to-Target Mapping
 - Zuweisung der Intensitätswerte aus I
 - Entstehung von Leerstellen ohne Intensität
- Target-to-Source Mapping
 - Inverse Koordinatenberechnung für jeden Pixel in I'
 - Intensitätswerte aus Interpolation von umliegenden Pixeln



Quelle: „Bildverarbeitung“, W. Burger and M. J. Burge, 2015

- Interpolationsmethoden:
 - Nächster Nachbar
 - **Bilinear**
 - Bikubisch
- Auswahl durch Berücksichtigung des Zeitaufwands und Ergebnisses



Grundlagen zu Bildmatrizen

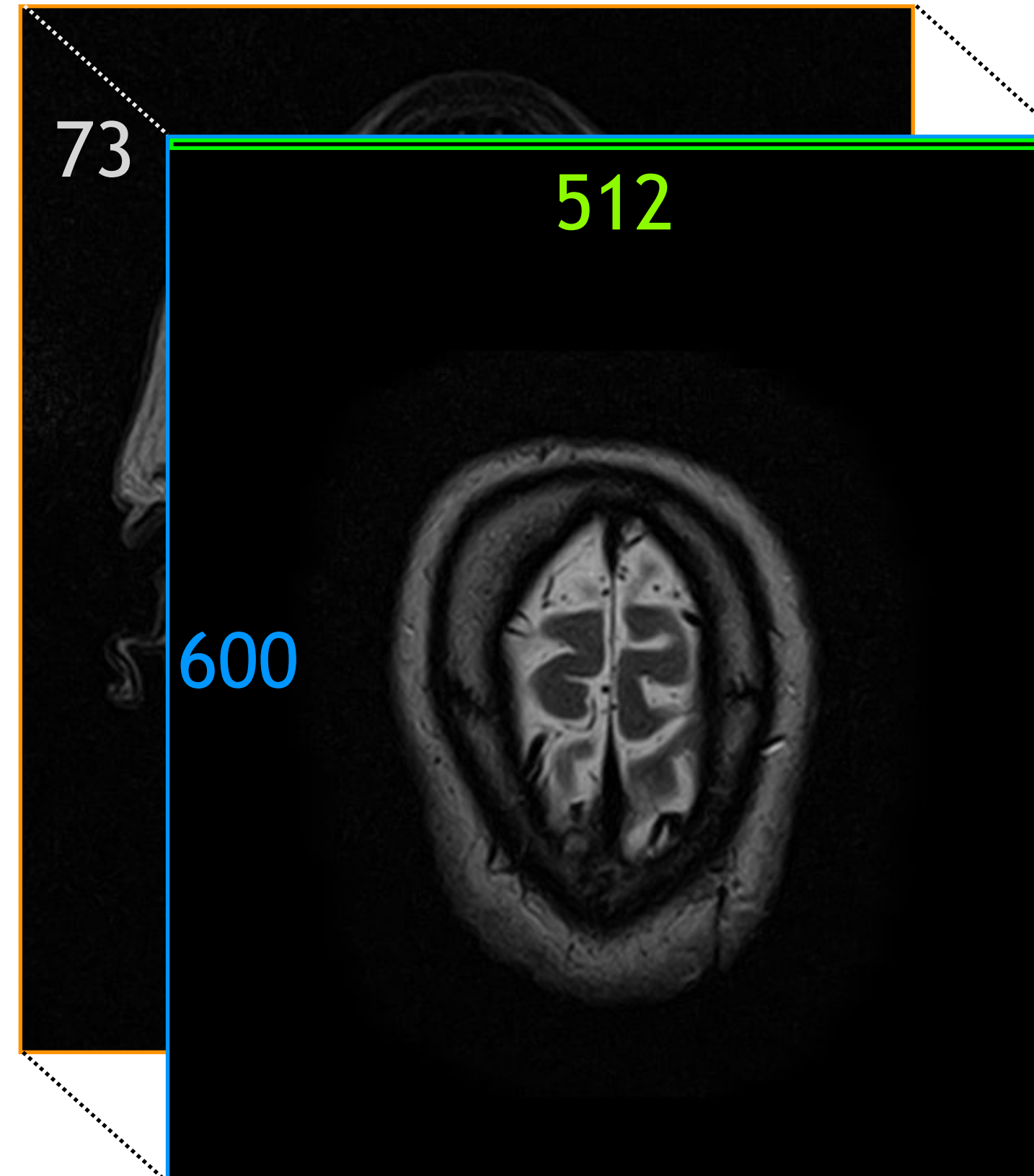
- Darstellung von 2D und 3D Aufnahmen als Numpy-Array in python

```
import numpy as np          ## Importieren von numpy

image = np.load("datei.npz") ## Öffnen/Laden einer .npz-Datei
array = image["vol"]        ## Definition des arrays mit dem Label "vol"
## Mehrere Elemente wie arrays, listen, dictionaries oder tuples können in einer
## .npz-Datei gespeichert werden. Jedem Element muss ein Label zugeordnet werden
## mit dem das Element wieder aufgerufen werden kann. Hier ist das Label für das
## array "vol". Mit list(image) wird die Bezeichnung aller Labels aufgerufen.
z, y, x = np.shape(array)   ## Form des arrays
```

- Form des Arrays durch `np.shape()`
 - $(z, y, x) \rightarrow (73, 600, 512)$
- Intensitätswerte abhängig vom Datentyp
 - int16: 2^{16} Möglichkeiten (-32768 bis +32767)
 - uint8: 2^8 ganze Zahlen (0 - 255)
 - float: Fließkommazahl zwischen -1 und 1

([Weitere Informationen](#))



```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int16)
```

Manipulation von Bildmatrizen

- Zahlreiche Möglichkeiten zur Bildverarbeitung in python durch *scikit-image* ([Link](#)):
 - Modul: transform
 - Verschiedene Transformationen (`AffineTransform`, `EuclideanTransform`, ...)
 - Rotation von Bildern (`rotate`)
 - Größenanpassungen (`warp`, `resize`, `rescale`)
 - Modul: filters
 - Bildfilter für Schärfe, Kontrast und Rauschen (`gaussian`, `sobel`, `unsharp_mask`, ...)
 - Segmentierung von Bildern durch Schwellenwerte (`threshold_mean`, `threshold_otsu`, ...)
- Visualisierung durch *matplotlib.pyplot*:

```
import matplotlib.pyplot as plt          ## Importieren von matplotlib.pyplot
# ... Erzeugung eines arrays
plt.imshow(array[50,:,:], cmap="gray")  ## Speichern von Schicht 50 des Bildes
```

- Vorsicht: Nur ein 2D-Array visuell darstellbar, bei allen Schichten muss getrickst werden
- Nachdem ein Array bearbeitet wurde, folgt zum Schluss das Speichern:
 - `np.savez_compressed(name, label1, label2, ...)`

Bildbearbeitung in python

2. Aufgabe:

- Nutze im Ordner *programmierkurs/ImageProcessing* die Dateien CT.npz und T2.npz!
- Schreibe ein python-Skript, in dem du jeweils
 - das Original-Bild um 20° rotierst,
 - zwei Segmente mit der Mittelwert- und Yen-Schwellenwertmethode erzeugst,
 - ein Segment um den Faktor 0.5 verkleinerst! (Hinweis)

Speichere immer die 30te Schicht als pdf-Bild ab!

Pause