

令和 6 年度卒業研究報告書
画像処理技術を用いた DICOM 医療画像データ
の活用法の研究

情報技術科 小原 弘暉

金野 朔也

佐々木 亮太

多田 光希

指導教員 菅野 研一

目次

第 1 章	はじめに	3
第 2 章	研究概要	4
2.1	研究の目的	4
2.2	研究の進め方	5
2.3	DICOM データの基礎	6
2.3.1	DICOM とは	6
2.3.2	DICOM の主な特性	7
2.4	3D モデルの構築方法	8
2.4.1	医療画像の 3D 再構成技術	8
2.4.2	HU 値	8
2.4.3	ボクセルとは	10
2.4.4	DICOM データの活用	11
2.4.5	完成イメージ	12
2.5	開発環境	14
2.5.1	使用ライブラリの説明	15
第 3 章	医療画像の解析と 3D 再構成	17
3.1	医療画像における基準面の名称	17
3.1.1	主な基準面と特徴	17
3.1.2	使用する DICOM データ	18
3.2	3D 再構成	20
3.2.1	コード説明	20
3.2.2	3D モデルについて	24

3.3 3D再構成（骨）	25
3.3.1 コード説明.....	25
3.3.2 3Dモデルについて.....	33
3.4 完全に骨のみを取り出せない原因	36
3.4.1 固定閾値処理における課題と適応的閾値の可能性.....	36
3.4.2 適応的な閾値処理の可能性	37
第4章 適応的閾値処理による課題.....	38
4.1 技術的な考察.....	38
第5章 結論と解決策	39
5.1 グローバル閾値を利用.....	39
5.2 マルチクラス分類（クラスタリング）の適用.....	40
5.3 ボリュームデータ全体の輝度ヒストグラムの解析と閾値の設定	40
5.4 より一貫性のある閾値決定方法	41
第6章 研究成果.....	42
第7章 おわりに	45
参考文献.....	46

第1章 はじめに

近年,医療分野におけるデジタル化の進展に伴い,医療画像データの解析・活用がますます重要になっている.特に,CT や MRI などの医用画像データの標準フォーマットである DICOM (Digital Imaging and Communications in Medicine) は,診断・治療の質の向上に大きく貢献している.しかし,DICOM データは大量の情報を含むため,効果的な処理・可視化が求められる.

本研究では,画像処理技術を用いて DICOM データの活用方法を検討し,特に 3D モデルの構築・表示に焦点を当てる.具体的には,Python を用いた画像処理技術を駆使し,DICOM データから骨や臓器を抽出し,直感的に扱える 3D モデルを作成することを目指す.

本研究の動機は,授業で画像処理技術に触れた際に,その応用範囲の広さに興味を抱いたことにある.特に,医療分野における DICOM データの活用は,診断支援や手術計画などにおいて極めて重要であり,その可能性を探究したいと考えた.

本卒業研究では,まず DICOM データの概要とその活用例について述べ,次に画像処理技術を用いた 3D モデル化の手法を紹介する.その後,研究過程で直面した課題や問題点を整理し,最終的に得られた成果について考察する.

第2章 研究概要

2.1 研究の目的

本研究の目的は,DICOM (Digital Imaging and Communications in Medicine) データの解析と 3D 可視化を行うことで,医療分野における画像処理技術の可能性を探ることである.DICOM は CT (Computed Tomography) や MRI (Magnetic Resonance Imaging) などの医用画像データを保存・管理するための国際標準規格であり,診断支援や手術計画において重要な役割を果たしている.しかし,DICOM データは一般的な画像フォーマットと異なり,スライス画像の集合として保存されており,そのままでは直感的な理解が難しい.

本研究では,Python を用いた画像処理技術を活用し,DICOM データの前処理,3D モデルの構築,閾値処理による骨・臓器の分離,および 3D 可視化の実装を行う.特に,固定閾値法と適応的閾値法の比較を通じて,より高精度な医療画像の解析手法を検討する.

2.2 研究の進め方

本研究では,DICOM データの取得,前処理,3D モデルの構築,可視化,評価の各プロセスを体系的に進めることで,安定して高精度な医療画像解析を実現することを目指した.

このプロセスは以下の 4 つのフェーズから構成される.

1. データの取得と解析

- DICOM 形式の医療画像データを収集し,Python のライブラリ (Pydicom) を用いてファイルを読み込む.
- メタデータの解析を行い,患者情報,撮影条件,HU 値 (Hounsfield Unit) などの基本特性を把握.

2. 画像の前処理

- ノイズ除去,輝度補正,ヒストグラム平坦化などの手法を適用し,後続の 3D 再構成の精度を向上させる.

3. 3D モデルの構築

- 各スライス画像を統合し,ボクセルデータを生成.
- 固定閾値法と適応的閾値法を比較し,骨や臓器の抽出精度を評価.

4. 3D 可視化の実装

- 直感的に操作可能なインタラクティブな 3D ビューワーを作成し,モデルの回転・ズーム・断面表示を実現

2.3 DICOM データの基礎

2.3.1 DICOM とは

- 医療データ通信の国際標準規格 DICOM（ダイコム）

DICOM（Digital Imaging and Communications in Medicine ダイコム）は、CT や MRI、CR の医用画像フォーマット、医用画像機器間で用いる通信プロトコルについて定義する、医用画像の共通規格である。医用画像共通フォーマット、通信フォーマットである DICOM によって、異なるメーカー間における医用画像機器の共有化を実現している[1]。

DICOM 画像は、ビットマップ RAW フォーマットなど、劣化しない可逆フォーマットで撮影される。医療用デジタル画像として高精細が求められるため、民生用の画像フォーマットに比べて色深度（1 ピクセル当りに割り当てるデータ量）が高く、リトルエンディアン式（最下位ビットの属するバイトを低位のアドレスへ格納する方式）のバイトオーダーで保存される。そのためサーバからビューアへの配信時は、通信処理の便宜の目的で、JPEG など圧縮率の高い不可逆フォーマットが採用される場合がある。

医療画像、すなわちレントゲン、CT、MRI の画像は、病院から持ち出す際に DICOM（ダイコム）データとして保存される。

CD データの中身は（図 2-1）のようになっている。

データを表示するときに見えるレントゲン画像は白黒のデータのみ（図の紫の枠）だが、DICOM データ形式のファイルには、それぞれさまざまな情報が含まれている（黄緑の枠）。

そのため、ファイルを 1 つだけ見ても、患者のプライバシー情報にアクセスできてしまう[2]。

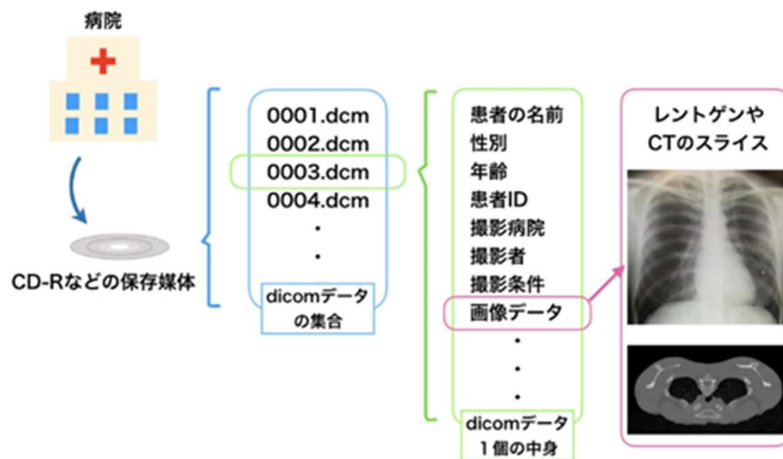


図 2-1 DICOM データについて

2.3.2 DICOM の主な特性

- データの統一性と相互運用

異なるベンダーの機器間でのシームレスなデータ交換が可能

- 階層構造

患者情報, 検査情報, 画像スライス情報が一元管理される.

2.4 3D モデルの構築方法

2.4.1 医療画像の3D 再構成技術

医療画像データを 3D モデルに再構成するためには,いくつかのステップが必要である.具体的には,スライス画像を統合してボクセルデータを生成し,閾値処理を通じて対象領域を抽出し,さらに 3D メッシュ化を行う.

2.4.2 HU 値

HU 値 (Hounsfield Unit,ハウズフィールド単位) は,CT (Computed Tomography) 画像において各ピクセル (ボクセル) の X 線吸収率を示す指標である[17].

(1) HU 値の定義

U 値は,水の X 線吸収率を基準 (0HU) とし,空気 (-1000HU) や骨 (1000HU 以上) など,異なる組織の密度に応じて数値が設定される.

表 2-1

組織・物質	HU 値
空気	-1000
脂肪	-100 ~ -50
水	0
筋肉・軟部組織	30 ~ 60
海綿骨	150 ~ 300
皮質骨	700 ~ 3000

(2) HU 値の利用例

- 骨と軟部組織の識別：一定の閾値を設定することで、骨や内臓を抽出可能
- 病変の診断：腫瘍や出血など、HU 値の異常から病変を検出。
- 3D 再構成：HU 値を基に組織を分類し、3D モデルを作成。

2.4.3 ボクセルとは

ボクセル (Voxel: Volume + Pixel の造語) とは、3D 空間における最小単位のデータであり、2D 画像におけるピクセル (Pixel) の 3D 版に相当する。DICOM データにおいては、CT や MRI のスライス画像を積み重ねることでボクセルデータが構築される (図 2-2)。各ボクセルには、HU 値 (Hounsfield Unit) などの密度情報が割り当てられており、骨や軟部組織の区別が可能となる [3]。

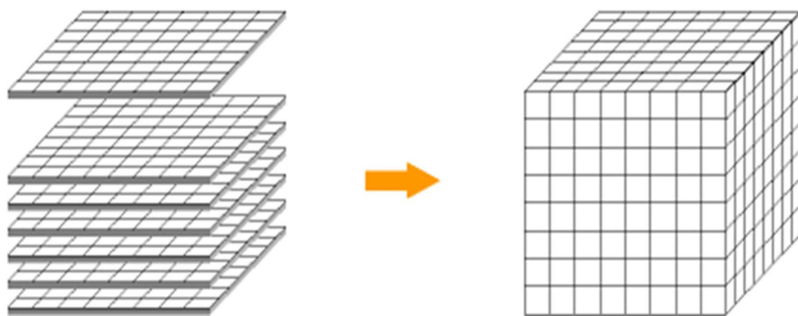


図 2-2 ボクセルについて

コメントの追加 [KK1]: この前か後ろの節で HU 値とは何かの説明が必要ですね。検索かけた範囲では説明を見つけられませんでした。

使用する手法と技術

1. 固定閾値法 (Thresholding) [7]

- 閾値を一律に設定し,指定した HU 値以上のボクセルを抽出.
- 骨の抽出に有効だが,密度に近い軟部組織との分離が難しい.

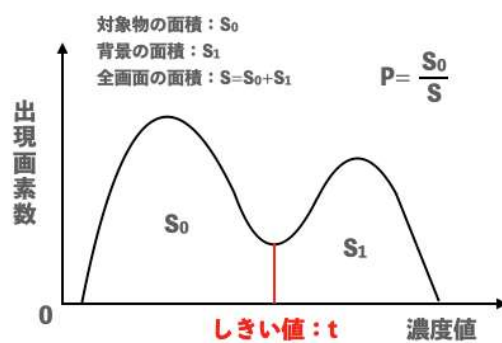


図 2-3 閾値について

2. 適応的閾値法 (Adaptive Thresholding) [7]

- 局所領域ごとに異なる閾値を計算し,輝度変動の影響を軽減
- 3D 再構成におけるノイズ耐性が向上.

2.4.4 DICOM データの活用

本研究では,DICOM データを活用し,医療画像から 3D モデルを構築する.特に以下の手法を用いて,より正確なモデルを実現する.

1. スライス画像の統合

- 各 DICOM スライス画像を一定間隔で積み重ね,3D ボクセルデータとして再構成.
- 撮影時のスライス間隔を考慮し,適切なスケーリングを実施.

2. 画像処理の適用

- ガウスフィルタを用いたノイズ除去.
- 閾値処理による特定組織の抽出 (骨や臓器など) .

3. メッシュ化とレンダリング

- Marching Cubes 法を用いた 3D メッシュ化.
- 表面レンダリングおよびボリュームレンダリングを適用.

2.4.5 完成イメージ

本研究では,以下のようなインターフェースと機能を持つ 3D モデルを開発することを目指す.

インタラクティブな操作性

1. 3D モデルの基本操作

- マウスやスライダを用いて回転,ズーム,移動が可能 (図 2-4) .

2. スライスごとの表示

- 任意の位置でスライスを可視化し,内部構造を確認できる.

3. 透明度の調整:

- 骨・臓器の表示濃度を変更し,特定の組織を強調可能.

断面表示機能

- パーツを切り離し,内部構造の観察が可能 (図 2-5) .
- 特定部位のみを強調し,局所的な詳細解析を実施.

レンダリングモードの切り替え

1. 表面レンダリング (Surface Rendering)
 - 骨や臓器の形状をリアルに再現
2. ボリュームレンダリング (Volume Rendering)
 - 密度情報を考慮した半透明な描画を行い,深部構造を可視化

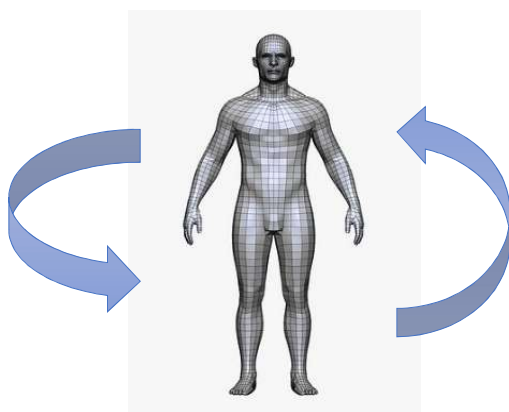


図 2-4 完成イメージ

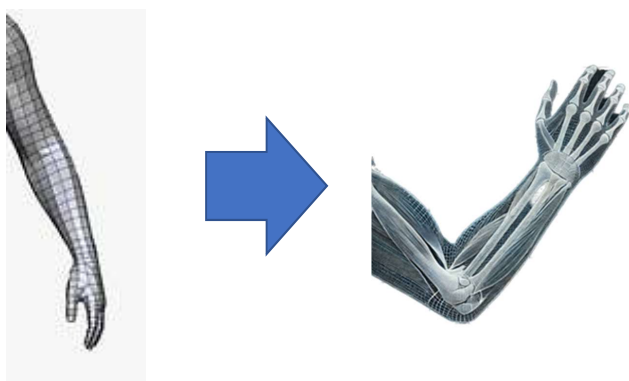


図 2-5 完成イメージ

2.5 開発環境

表 2-2 開発環境一覧

項目	内容
開発環境	Visual Studio Code (VSCode)
言語	Python
使用ライブラリ	Os Pycicom NumPy MPI Skimage Matplotlib SciPy

本研究では,Python を用いて DICOM データの解析および 3D 可視化を行う.開発環境としては,コードの記述とデバッグを容易にするために**Visual Studio Code (VSCode) **を採用した.Python は,数値計算や配列操作に強く,医療用途に使えるライブラリが豊富であるため,本研究の実装に適している.

2.5.1 使用ライブラリの説明

本研究で使用するライブラリについて,以下に説明する.

1. OS (標準モジュール)

Python に標準で含まれるモジュールであり,オペレーティングシステムとのやり取りを行う.主にファイルやディレクトリの操作に使用され,DICOM データのフォルダ管理やファイルの自動読み込み処理に役立つ.

2. Pydicom

医療用画像データ (DICOM ファイル) の読み書きや解析を行うためのライブラリ.DICOM フォーマットに特化しており,患者情報や撮影条件などのメタデータの取得や画像データの抽出が可能である.本研究では,DICOM ファイルの読み取りと解析に使用する.

3. NumPy

高速な数値計算と多次元配列操作を行うためのライブラリ.DICOM スライス画像を等間隔で並べ,立体的なデータ構造を作成するために使用する.特に 3D ボリュームデータの生成や配列操作の効率化に貢献する.

4. MPI (Message Passing Interface)

データの並列計算を行うライブラリ.本研究では,大量の DICOM スライスデータを効率よく処理し,計算速度の向上を図るために使用する.特に,3D モデルの再構成処理やフィルタリングなどの計算負荷の高い処理を分散する目的で活用する.

5. Scikit-image (Skimage)

画像処理に特化したライブラリであり,フィルタリングや輪郭抽出,画像変換などをサポートする.DICOM 画像のサイズ統一やノイズ除去の前処理に活用する.

6. Matplotlib

データの可視化を行うライブラリであり,2D および3D のグラフ描画が可能.本研究では,DICOM スライスの表示,スライダーやボタンを含む UI の構築さらには3D モデルの可視化に使用する.

7. SciPy

NumPy と連携し,数値計算や信号処理を行うためのライブラリ.DICOM 画像の読み取り,ノイズ除去,補間処理などを行うために利用する.

この開発環境とライブラリを活用することで,効率的に DICOM データの解析・可視化を行い,医療分野での応用可能性を探求する.

第3章 医療画像の解析と3D再構成

3.1 医療画像における基準面の名称

医療画像の解析や3D再構成を行う際、解剖学的な基準面の理解は重要である。特に、CTやMRI画像は特定の基準面で撮影されるため、適切に識別する必要がある。

3.1.1 主な基準面と特徴

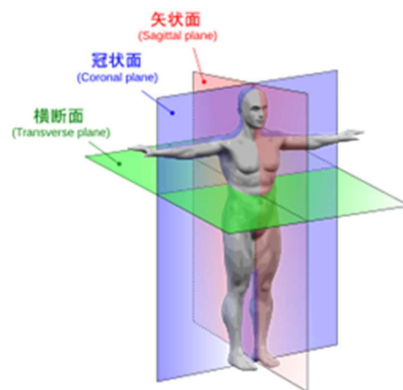


図 3-1 基準面の名称

- 矢状面（Sagittal Plane）[4]
 - ・ 体を 左右 に分ける面。
 - ・ 正中矢状面（Median Sagittal Plane）は 体の中央を通る矢状面 を指す。
 - ・ 脳の左右半球や脊柱の歪みを評価する際に使用される。
- 冠状面（Coronal Plane）[5]
 - ・ 体を 前後（腹側・背側） に分ける面。
 - ・ 心臓や肺などの正面からの構造を観察する際に用いられる。
 - ・ MRI では脳の前後関係を把握するためによく使用される。

- 横断面 (Axial/Transverse Plane) [6]
 - ・ 体を 水平に切り,上半身と下半身に分ける面.
 - ・ CT スキャンでは一般的にこの断面で撮影され,脳,胸部,腹部などの診断に利用される.
 - ・ 3D 再構成の際の基礎データとなるスライス画像がこの面で取得される.

3.1.2 使用する DICOM データ

以下のデータは,指導教員から提供されたもので,全体で899 枚の画像から構成されている.

以下の画像は,卒業研究で使用している DICOM データの一部を示している.取り扱っている横断面画像は320 枚存在し (図 3-2) ,骨や内部構造の詳細な解析を行う上で重要な役割を果たしており,これらのデータを用いて研究を進めている.

全体のデータセットの中で,冠状面で撮影された画像は92 枚存在しており (図 3-3) ,これらの画像は体の前後方向の断面を詳しく観察するために利用される.これらのデータを用いることで,骨や組織の位置関係を明確にでき,研究の重要な基盤となっている.

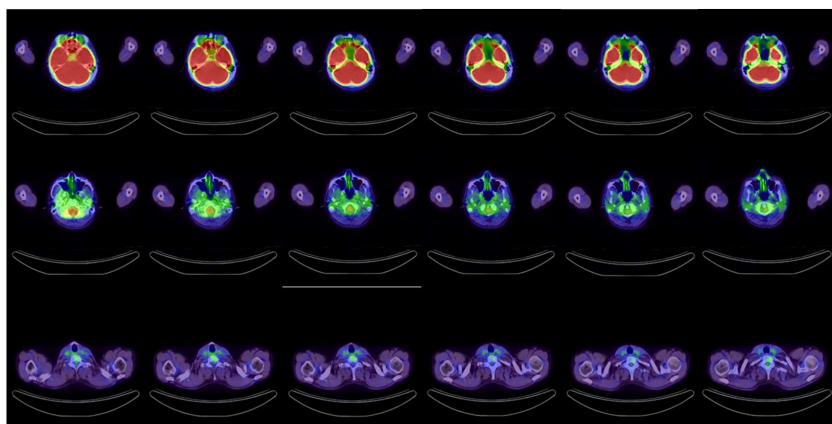


図 3-2 RGB 形式の横断面画像

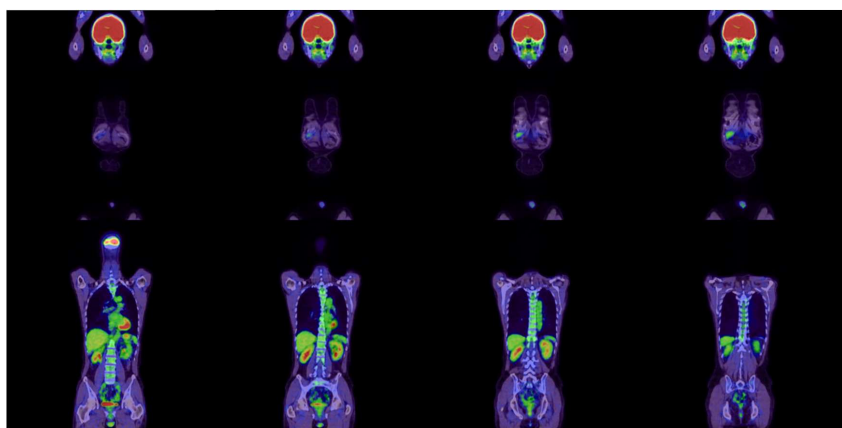


図 3-3 RGB 形式の冠状面画像

3.2 3D 再構成

3.2.1 コード説明

1. DICOM ファイルの読み込み

まず,Python の pydicom ライブラリを用いて DICOM ファイルを読み込み,画像データを取得する.DICOM ファイルはフォルダ内に複数存在するため,それらをリスト化し,ファイル名に含まれる数値でソートする.

```
import os

import pydicom

# DICOM ファイルがあるディレクトリのパスを指定
directory_path = r'P:\sotuken¥Coronal_RBG_datae' # 冠状面
#directory_path = r'P:\sotuken¥Axial_GrayScale_date' # 横断面

# ディレクトリ内の DICOM ファイルを取得し,ファイル名の数字でソート
dicom_files = [f for f in os.listdir(directory_path) if not f.startswith('.')]
dicom_files.sort(key=lambda x: int("".join(filter(str.isdigit, x))))
```

- os.listdir() を使用してディレクトリ内の DICOM ファイルをリスト化
- sort(key=lambda x: int("".join(filter(str.isdigit, x)))) により,ファイル名の数値部分でソートすることで,スライス順を正しく保つ.

2. 画像の前処理

DICOM 画像のピクセルデータを取得し,統一したサイズ (512×512) にリサイズする.また,RGB 画像である場合はグレースケールに変換する.

```
import numpy as np

from skimage.transform import resize

from skimage.color import rgb2gray

# 画像データを格納するリスト
images = []

target_shape = (512, 512) # 画像サイズを統一

# DICOM ファイルを読み込んでスタック
for file in dicom_files:

    file_path = os.path.join(directory_path, file)

    try:

        dicom_data = pydicom.dcmread(file_path) # DICOM ファイルを読み込む

        img_array = dicom_data.pixel_array # ピクセルデータを取得

        # 画像サイズを統一

        if img_array.shape != target_shape:

            img_array = resize(img_array, target_shape, anti_aliasing=True)

        # RGB 画像の場合はグレースケールに変換
```

```

if img_array.ndim == 3 and img_array.shape[2] == 3:

    img_array = rgb2gray(img_array)

    img_array = (img_array * 255).astype(np.uint8) # 8 ビット画像に変換

images.append(img_array) # 画像リストに追加

except Exception as e:

    print(f'ファイル '{file}' の読み込みエラー: {e}')

```

- `resize()` を用いてすべての画像を (512, 512) にリサイズし,形状を統一.
- `rgb2gray()` を用いて RGB 画像をグレースケールに変換し,データの一貫性を保つ.

3. 3D ボリュームの作成

前処理した画像を 3D 配列に変換し,スタックする.これにより,CT 画像のボリュームデータを作成する.

```

# 3D ボリュームデータを作成

if images:

    volume = np.array(images)

    print(f'ボリュームの形状: {volume.shape}') # (スライス数, 画像の高さ, 画像の幅)

```

- `numpy.array()` を使用して,2D 画像のリストを 3D ボリュームデータに変換.

4. 3D 可視化

3D ボリュームデータを matplotlib を用いて可視化する。スライスごとに `contourf` を用いて描画し、対象領域を 3D で表示する。

```
import matplotlib.pyplot as plt

# 3D 表示の関数
def show_3d(volume_data):

    fig_3d = plt.figure()

    ax_3d = fig_3d.add_subplot(111, projection='3d')

    z, y, x = volume_data.shape

    X, Y = np.meshgrid(np.arange(x), np.arange(y))

    for i in range(z):

        slice_data = volume_data[i]

        if np.isnan(slice_data).all(): # スライス全体が NaN ならスキップ

            continue

        ax_3d.contourf(X, Y, slice_data, zdir='z', offset=i, alpha=0.5, cmap='gray')

    ax_3d.set_xticks([]) # X 軸の目盛りを非表示
    ax_3d.set_yticks([]) # Y 軸の目盛りを非表示
    ax_3d.set_zticks([]) # Z 軸の目盛りを非表示

    plt.show()
```



```
# フィルタ済みのデータで 3D 表示
```

```
show_3d(volume_filtered)
```

- `contourf()` を用いてスライスごとに描画し,3D 視覚化を実現
- `NaN` 値を透明化することで,対象領域のみが強調される.

3.2.2 3D モデルについて

図 3-4,図 3-5 は,冠状面および横断面で撮影された DICOM データを基に作成した 3D モデルである.

図 3-5 は,冠状面データを基に構築した 3D モデルで,身体を前後に分けた断面を用いている.

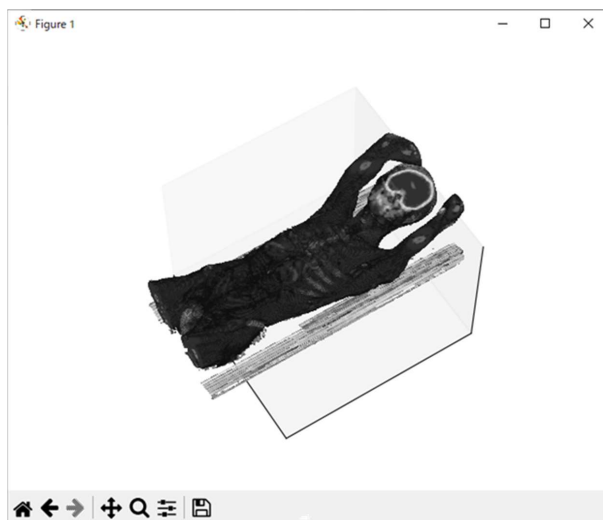


図 3-4 冠状面スライスから構築した 3D ボリューム

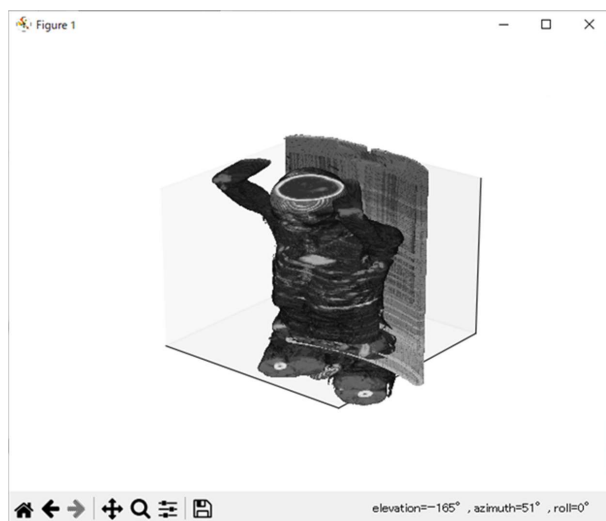


図 3-5 横断面スライスから構築した 3D ボリューム

3.3 3D 再構成（骨）

3.3.1 コード説明

本研究では、医療画像フォーマットである DICOM (Digital Imaging and Communications in Medicine) ファイルを用いて、CT 画像の前処理および 3D 可視化を行う手法を検討した。DICOM データは医用画像解析に広く用いられるが、そのままでは視覚化や解析が難しいため、適切な前処理が必要である。本研究では、DICOM 画像を読み込み、平滑化処理や閾値処理を行い、骨領域を強調した 3D 可視化を行うことを目的とする。

1. DICOM ファイルの読み込み

まず,Python の pydicom ライブラリを用いて DICOM ファイルを読み込み,画像データを取得する.

本研究では,指定したフォルダ内の DICOM ファイルをリスト化し,順番に処理を行う.

```
import os

import pydicom

# DICOM ファイルがあるディレクトリのパスを指定
directory_path = r'P:\sotuken\Coronal_RGB_datae' # データセットのパス

# ディレクトリ内の DICOM ファイルを取得
dicom_files = [f for f in os.listdir(directory_path) if not f.startswith('.')]

dicom_files.sort() # ファイル名をソート
```

2. 画像の前処理

DICOM 画像のピクセルデータを取得し,統一したサイズ (512×512) にリサイズする.また,RGB 画像である場合はグレースケールに変換する.

```
import numpy as np

from skimage.transform import resize

from skimage.color import rgb2gray

# 画像データを格納するリスト
images = []

target_shape = (512, 512) # 画像サイズを統一
```

```

# DICOM ファイルを読み込んでスタック
for file in dicom_files:

    file_path = os.path.join(directory_path, file)

    try:

        dicom_data = pydicom.dcmread(file_path) # DICOM ファイルを読み込む

        img_array = dicom_data.pixel_array # ピクセルデータを取得

        # 画像サイズを統一

        if img_array.shape != target_shape:

            print(f'{file}: 画像のリサイズ実行.元の形状: {img_array.shape}')

            img_array = resize(img_array, target_shape, anti_aliasing=True)

        # RGB 画像の場合はグレースケールに変換

        if img_array.ndim == 3 and img_array.shape[2] == 3:

            print(f'{file}: RGB 画像をグレースケールに変換")

            img_array = rgb2gray(img_array)

            img_array = (img_array * 255).astype(np.uint8) # 8 ビット画像に変換

        images.append(img_array) # 画像リストに追加

    except Exception as e:

        print(f'ファイル '{file}' の読み込みエラー: {e}')

```

3. 3D ボリュームの作成

前処理した画像を3D 配列に変換し,スタックする.これにより,CT 画像のボリュームデータを作成する.

```
# 3D ボリュームデータを作成

if images:

    volume = np.array(images)

    print(f"ボリュームの形状: {volume.shape}") # (スライス数, 画像の高さ, 画像の幅)
```

4. ガウスフィルタによる平滑化[18][19]

ガウスフィルタとは？

ガウスフィルタ (Gaussian Filter) は,画像のノイズを軽減し,滑らかな表現を得るためのフィルタ処理の一種である.特に医療画像では,不要なノイズを抑えつつ,重要な構造を維持するために有用である.

コメントの追加 [KK2]: ガウスフィルタは他のところで説明がないようなので,ここで解説してからコードを提示するように,ガウスフィルタの説明を加えてください。 sigma 値の意味も。

平均化フィルター

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3×3 画素

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

5×5 画素

ガウシアンフィルター

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

3×3 画素

1/256	4/256	6/256	4/256	1/256
4/256	16/256	24/256	16/256	4/256
6/256	24/256	36/256	24/256	6/256
4/256	16/256	24/256	16/256	4/256
1/256	4/256	6/256	4/256	1/256

5×5 画素

図 3-6 平均・ガウシアンフィルタ

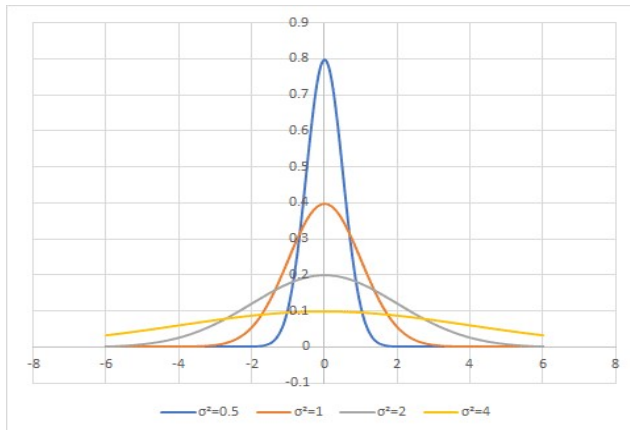


図 3-7 正規分布の分散による変化

図 3-6 の通り平均化フィルタとの大きな違いは処理対象となる真ん中の注目画素が最も重みが大きくなり、外側の画素ほど重みが小さくなる点。この特性により、画素付近の情報をより残したまま画像をぼかすことが可能になる。

CT 画像にはノイズが含まれることが多いため、`scipy.ndimage.gaussian_filter` を用いてガウスフィルタを適用し、ノイズを低減する。

```
from scipy.ndimage import gaussian_filter # ガウスフィルタをインポート

# ガウスフィルタを適用してノイズを軽減
volume_smoothed = gaussian_filter(volume, sigma=1.5) # sigma 値は 1.5 に設定
```

sigma の意味

- 小さい値（例：sigma=0.5） → ぼかしが弱く、細かいディテールが保持される

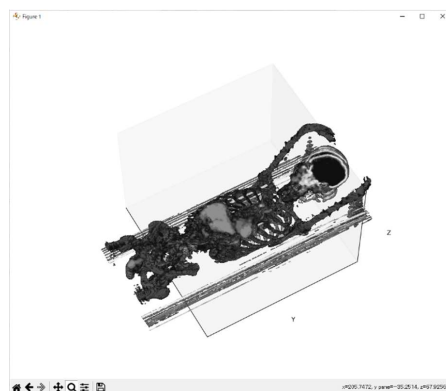


図 3-8 sigma=0.5

- 大きい値（例：sigma=3.0） → ぼかしが強くなり、ノイズは減るが細かい情報も失われる

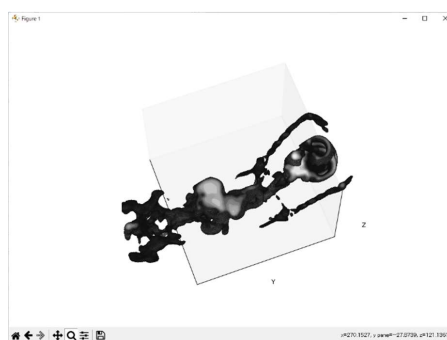


図 3-9 sigma=3.0

本プログラムでは,sigma=1.5 を設定しており,適度なノイズ除去を行いながら骨の輪郭を保持することを目的としている.

5. 骨組織の強調

CT 画像では HU 値 (Hounsfield Unit) を用いて組織を識別できる.本研究では,HU 値が 80 以上の領域を骨とみなし,閾値処理を行った.

```
# 骨組織のマスクを作成

threshold = 80 # 骨の閾値

bone_mask = (volume_smoothed > threshold) # 閾値を超える部分を骨とする


# 骨領域のみを抽出し,その他の部分を NaN に設定

volume_bone = np.where(bone_mask, volume_smoothed, np.nan)
```

6. 3D 可視化

3D ボリュームデータ matplotlib を用いて可視化する.contourf を用いてスライスごとに 2D コンターを描画し,それを積み重ねることで 3D 構造を表現する.

```
import matplotlib.pyplot as plt


# 3D ボリュームの可視化

fig = plt.figure(figsize=(10, 10))

ax = fig.add_subplot(111, projection='3d')


z, y, x = volume_bone.shape

X, Y = np.meshgrid(np.arange(x), np.arange(y))


# スライスごとに表示
```



```
slice_gap = 0.5 # スライス間隔を調整

for i in range(z):

    slice_data = volume_bone[i]

    ax.contourf(X, Y, slice_data, zdir='z', offset=i * slice_gap, alpha=0.7, cmap='gray')

# 軸のラベル設定
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_xticks([])
ax.set_yticks([])
ax.set_zticks([])

# グラフの表示
plt.show()
```

3.3.2 3D モデルについて

以下の図は骨のみを表示させたかったものである。図 3-10 のモデルは閾値を 70 にしたもので、骨だけでなく皮膚のような部分までうっすらと見えてしまっている。図 3-11 のモデルは閾値を 80 に設定したもので、一見すると骨だけがうまく取り出せているように見えるが、よく見ると胃や腸などの臓器が完全には除去されていない。しかし、3 つの中では一番バランスよく骨を取り出せている。

最後に図 3-12 のモデルは閾値を 90 にすると、一部の骨が消えかかっているのが確認できる。

こうしたことから、閾値を 80 より高くすると、軟部組織や低密度の骨が除外され、高密度の骨だけが残る。その結果、細かい部分やスカスカな骨の部分が消えてしまうことがある。

逆に、閾値を 80 より低くすると、軟部組織や脂肪まで含まれてしまい、ノイズが増えて骨の抽出が曖昧になる。

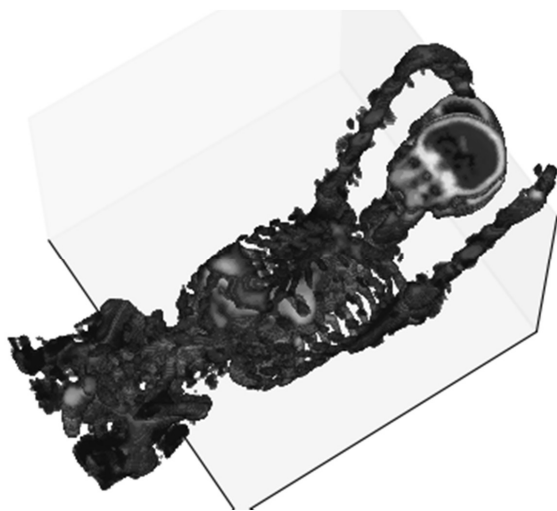


図 3-10 閾値 70 の閾値処理による 3D 骨モデル



図 3-11 閾値80の閾値処理による3D骨モデル

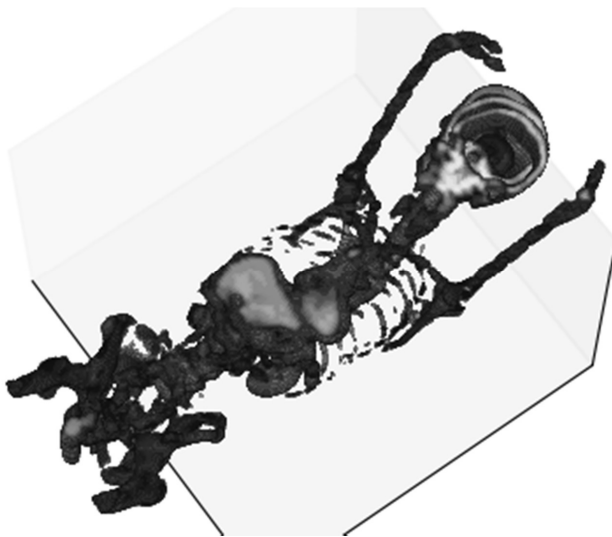


図 3-12 閾値90の閾値処理による3D骨モデル

また,中間発表の際に閾値を変更できるスライダーを付けた方がリアルタイムで確認できるとご指摘いただいたので,実際にスライダー機能を付けたのが下の図である.

実際にスライダー機能を付けたが,スライダーを動かすたびに処理が始まるので,とても重く思うように動かせないのが現実である(図 3-13).



図 3-13 スライダーによる調整可能な 3D モデル

3.4 完全に骨のみを取り出せない原因

以下は閾値を80に設定し、臓器付近をアップにして撮影したものである。骨のみを表示する3Dモデルであるが、胃や小腸などの臓器が完全には除去されていないことがわかる(図 3-14)。

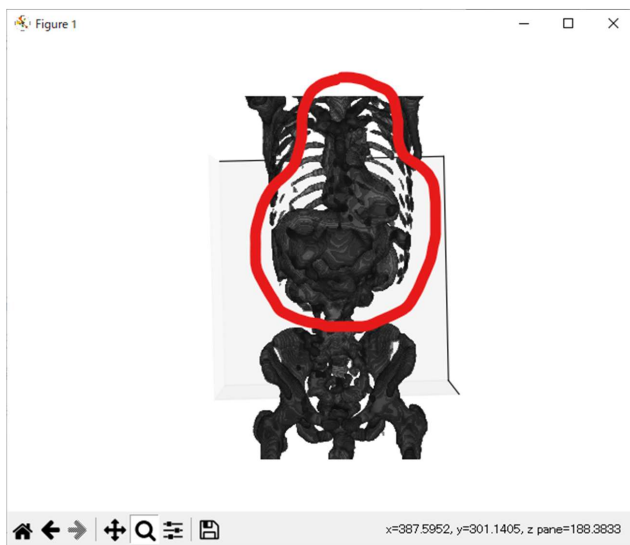


図 3-14 閾値80で抽出した臓器3Dモデルの臓器付近

3.4.1 固定閾値処理における課題と適応的閾値の可能性

胃や腸に食物や水分が含まれている場合、それらの領域は比較的高いHU値(80~300度)を持つことと海綿骨(150~300)と近い値。単純な閾値処理では骨と明確に区別することが難しい。

現在、固定閾値法では閾値を80に設定しているが、骨の範囲だけでなく、密度がやや高い軟部組織や臓器の内容物も含まれてしまう。固定閾値処理はHU値のみを基準に判定するため、骨の形状や位置などの追加情報を考慮できない結果として、胃や腸などの高密度領域が誤って骨と認識されることがある。

3.4.2 適応的な閾値処理の可能性

適応的な閾値処理を置くことで,以下のような改善の可能性はある.

1. 局所的な HU 値の分布を考慮し,適応的に閾値を設定できる
 - 骨のある領域では高い閾値を設定し,軟部組織の領域では低い閾値を設定することで,密度の近い組織をより正確に分類できる.
2. 組織の局所的なコントラストを考慮し,より正確な分類が可能
 - 単純な固定閾値ではなく,適応的な閾値処理を導入することで,骨と軟部組織をより適切に分離できる可能性がある.
3. 組み合わせ手法による精度向上
 - 適応的な閾値に加え,形状情報 (モルフォロジー処理) や機械学習を活用することで,誤分類のリスクを軽減する.

固定閾値法のみでは,骨と胃・腸の内容物を明確に分離することが難しい.適応的な暫定値処理を利用することで,局所的な密度変動を考慮し,より精度の高い骨抽出が可能となる可能性があります.適応的な閾値と形状情報を組み合わせた手法の検討が必要である.

第4章 適応的閾値処理による課題

適応的閾値処理を用いたプログラムで骨と内臓を正確に分離し、3D 表示できない主な原因は、「局所的な輝度変動による閾値の不安定性」である。

4.1 技術的な考察

1. DICOM データの輝度分布が均一でない

- DICOM 画像は撮影条件や患者ごとにコントラストや輝度のばらつきが大きい。
- 骨と内臓の輝度差は撮影条件や患者の状態によって異なるため、適応的閾値処理を適用しても、骨と内臓を一貫して正しく分離することが難しくなる。

2. 適応的な閾値の影響範囲が局所的すぎる

- 適応的な閾値は「局所領域（小さな画像ブロック）」で計算されるため、大きな組織（骨や臓器全体）を集中して分類できない。
- CT スキャンのようにコントラストが低いデータでは、適応的閾値処理を適用しても、領域間で閾値が大きく変動し、安定した分類が難しくなる。
- 特に、内部組織の緻密な構造がある場合、閾値の変動によって意図的に部分が抽出される可能性が高い。

3. 3D データへの適用が必要

適応的閾値処理は、通常 2D 画像（1 枚のスライス）に対して適用される処理である。

そのため、3D 空間ではスライスごとに異なる閾値が適用され、骨や内部構造が不連続になり、途切れてしまうことがある。

結果として、3D 再構成を行う際に骨や臓器の形状が崩れ、正確なモデルを作成することが難しくなる。

第5章 結論と解決策

適応的閾値は局所的なコントラストを強調するが,全体的な構造の分類（骨と内臓の識別）には不向き.この場合,グローバルな閾値設定が重要となる.

5.1 グローバル閾値を利用

1. 大津の二値化 [15]

- 画像のヒストグラムに基づき,クラス間の分散が最大となる閾値を自動的に決定する手法.画像全体を対象として,複数のクラス（例：背景,骨,内臓）を分割するのに有効.
- 大津の方法では,画像のヒストグラムにおける明るさの分布に基づき,最適な分割点を見つけて出す.これにより,適応的閾値では難しい,より一貫性のある分類が可能になる.

2. ヒストグラムの分析

- 画像の輝度分布（ヒストグラム）を用いて,特定の範囲に基づいた閾値を設定する方法.骨や内臓はそれぞれ異なる輝度値を持つことが多いため,全体的なヒストグラムを解析し,各構造に対応する輝度範囲を特定する.
- ヒストグラム分析をすることで,画像全体の特徴に基づいたより堅牢な閾値を設定することができる.

5.2 マルチクラス分類（クラスタリング）の適用

画像データが複数の異なるクラス（骨、内臓、背景など）を含んでいる場合、単純な二値化ではなく、マルチクラス分類を使うことで、より精度の高い分類が可能。

- クラスタリング手法（例：K-means 法や k-NN 法など）を用いることで、複数のクラスに基づいた特徴量の分布を捉えることができる。
- クラスタリングを適用すると、画像内の異なる領域を個別に分類することが可能となり、グローバルな構造（骨や内臓など）の識別がより正確になる。

5.3 ボリュームデータ全体の輝度ヒストグラムの解析と閾値の設定

ボリュームデータ（CT スキャンや MRI などの 3D 画像データ）の場合、画像内の各スライスに対して個別に閾値を設定するのではなく、全体的な輝度ヒストグラムを解析して、適切な閾値を決定する方法が有効。

- ボリュームデータ全体にわたって輝度の分布を調べ、全体的な傾向を理解することで、より安定した閾値を設定できる。これにより、各スライス間で一貫性のある分類が可能になる。
- 特に、骨や内臓のように、特定の輝度範囲を持つ構造を識別する際に、全体のヒストグラムを考慮することで、局所的なノイズに影響されずに適切な分類が可能。

5.4 より一貫性のある閾値決定方法

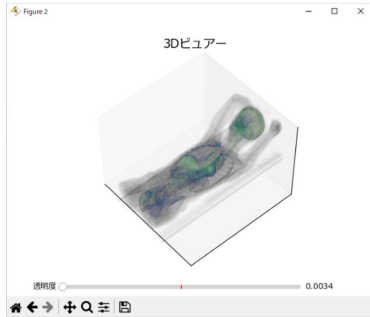
適応的閾値では,局所的なコントラストの変化に影響されるため,グローバルな構造分類には不向き.これに対して,全体的なヒストグラム分析や大津の二値化など,より一貫性のある方法を使うことで,分類結果が安定する.

- グローバル閾値を用いることで,全体の輝度範囲を考慮し,全体的な画像の性質に基づいた分類が可能になります.これにより,適応的閾値が局所的な変動に敏感であるという問題を解決できる.

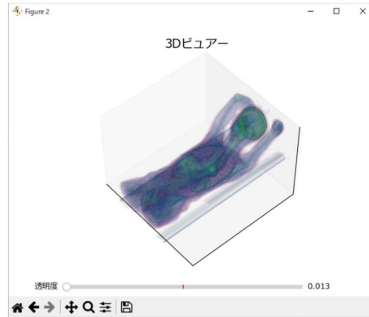
グローバルな構造の分類(例:骨と内臓の識別)には,適応的閾値ではなく,全体的な輝度分析に基づいた方法を使用することが有効.大津の二値化やヒストグラム分析を用いて,ボリウムデータ全体に適した閾値を決定し,さらにマルチクラス分類を組み合わせることで,より精度高く,安定した分類が実現できる.

第6章 研究成果

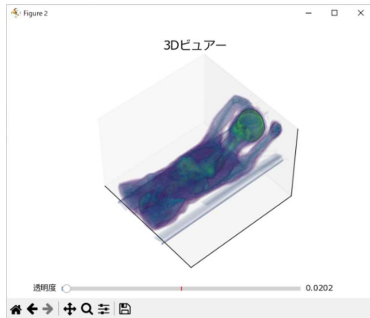
- 横断面データは 320 枚の処理により,データサイズが大きくなりプログラムの実行が遅くなった.
- 縦断面データは 92 枚であり,処理が軽快で実際の作業が可能であった.
- 縦断面画像を主に使用し,3D 再構成の検討を進めた.
- データサイズの制約が処理効率や実行可能性に与える影響を認識した.
- 適応的閾値を利用することで骨と内臓を分けることができる可能性がある.
- 解像度の不均一やノイズが 3D モデルに影響.
- 大量のスライスデータにより処理が重くなる.
- 高性能な GPU やクラウドの活用が必要な場合がある.
- DICOM データを取得し,3D 表示する際に,利便性を向上するために 3D ビュアーの透明度を変更するスライダー機能を追加した(図 6-1).
- DICOM ファイルを軽量化させると,データに劣化が生じ,3D 時にノイズが入ってしまうため,データの軽量化はできなかった.
- 適応的閾値を利用することはできなかった.



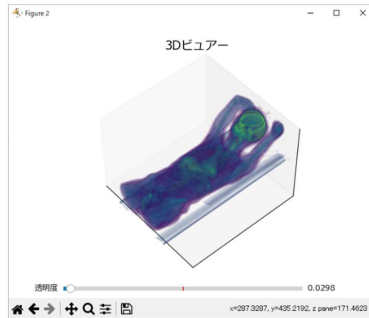
(1)



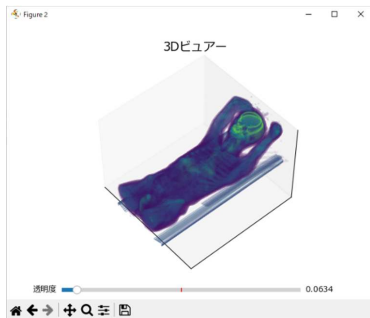
(2)



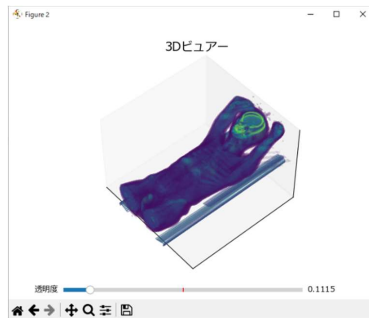
(3)



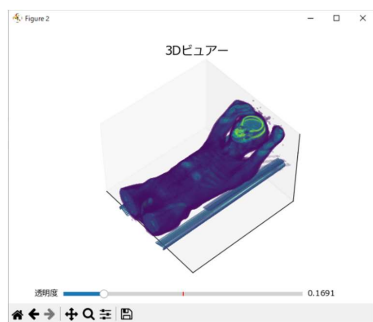
(4)



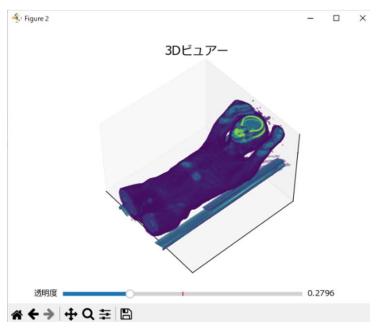
(5)



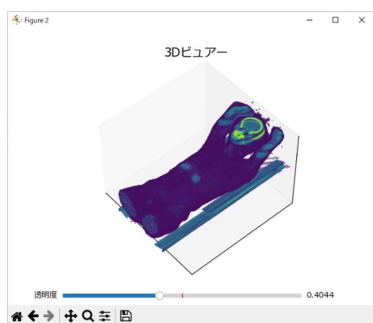
(6)



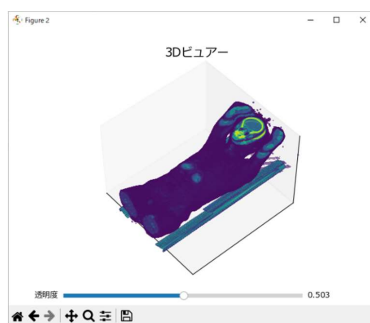
(7)



(8)



(9)



(10)

図 6-1 3D ビューアーの色付けと透明化

第7章 おわりに

本研究では,DICOM 画像の取り扱いや Python を用いたプログラム作成技術,さらに Os や Pydicom などのライブラリを活用する方法について学び,医療画像データの処理に関する知識と画像処理技術を深く理解することができた.

しかし,研究開始時に計画していた画像圧縮ツールや 3D モデルのパーツ見極めなどの高度な処理については,パソコンのスペックや処理速度に起因する問題から実現できなかった.特に,3D モデルの詳細な調整や,大規模なデータセットを扱う際の計算リソースの不足が技術的な壁となり,作業の効率を向上させるためにハードウェアのアップグレードやプログラムの最適化が重要であることを認識した.

また,プログラム作成時に発生した動作の遅延や,DICOM データの種類によって異なる挙動についても,実際のデータを用いて試行錯誤を繰り返した.これにより,フォーマットの違いが処理速度や結果にどのような影響を与えるのかを検証し,最適な設定を見つけるために多くの時間を費やした.

さらに,研究を進める中で,DICOM データを用いた画像解析の精度向上や,より効率的なデータ処理手法の模索にも取り組み,今後の研究において,得られた成果を活かし,さらに課題解決に向けた改善を進めていく予定である.

参考文献

- [1] DICOM とは - 医療データ通信の国際標準規格 DICOM (ダイコム)
https://www.liberworks.co.jp/know/know_dicom.html
- [2] DICOM データとはなにか - 医療のための Python プログラミング
<https://programming-surgeon.com/imageanalysis/what-is-dicom/>
- [3] ピクセルとボクセル - 空間分解能・スライス厚
<https://www.ct-tekiyuku.net/basic/size/size005.html>
- [4] Sagittal
<https://www.zio.co.jp/revoras/wiki/sagittal/>
- [5] Coronal
<https://www.zio.co.jp/revoras/wiki/coronal/>
- [6] Axial
<https://www.zio.co.jp/revoras/wiki/axial/>
- [7] 二値化入門: 画像を鮮明にする技術を解説
<https://www.phoxter.co.jp/archives/2036>
- [8] CT とは, どのような仕組み? | CT に関する疑問にお答えします | キヤノンメディカル
https://jp.medical.canon/general/CT_construction
- [9] 放射線画像診断技術はどこまで進化したのか? CT, MRI 研究の第一人者が語る最前線
<https://www.juntendo.ac.jp/branding/report/151>
- [10] 日本の誇る医療技術 (2) CT と MRI | メディアスホールディングス株式会社
https://www.medius.co.jp/asourcenavi/ct_and_mri/
- [11] 【Python 入門】絶対に知っとくべきライブラリ scipy の基本的な使い方
<https://qiita.com/Morio/items/5bf23db5ae5109ab1882>

[12] 【Python】 3D モデルを描画できるライブラリ 3 選

<https://trends.codecamp.jp/blogs/media/column17>

[13] 二値化処理とは？仕組みや流れとしきい値の設定に必要な基礎知識 - 株式会社 FA プロダクツ JSS 事業部 | 関東最大級のロボット Sler

https://jss1.jp/column/column_246/

[14] 【Python・OpenCV】ソーベルフィルタによるエッジ検出 - codevace

https://www.codevace.com/py-opencv-sobel/#google_vignette

[15] 二値化入門：画像を鮮明にする技術を解説 - Phoxter Corporation

<https://www.phoxter.co.jp/archives/2036>

[16] [OpenCV]画像の二値化処理を初心者にも分かりやすく徹底解説！ | すらぶろ

<https://tomtom-stock.com/2022/03/20/opencv-cv2-threshold/>

[17] CT 値の単位は HU

<https://www.ct-tekiyuku.net/basic/density/density005.html>

[18] ガウシアンフィルタ - 画像処理におけるノイズの除去

<https://www.mitani-visual.jp/mivlog/imageprocessing/gf3r89.php>

[19] 空間フィルタによる平滑化とエッジ検出

<https://htsuda.net/archives/2064>