

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI  
I TECHNIK INFORMACYJNYCH



Instytut Mikroelektroniki i Optoelektroniki

# Praca dyplomowa inżynierska

na kierunku Elektronika  
w specjalności Elektronika i Inżynieria Komputerowa

Implementacja algorytmów przetwarzania obrazów dla tomografu  
pozytonowego JPET

Krzysztof Krupiński

Numer albumu 277431

promotor  
dr inż. Arkadiusz Łuczyk

konsultacje  
dr inż. Konrad Klimaszewski

Warszawa 2019



---

## Streszczenie

### Implementacja algorytmów przetwarzania obrazów dla tomografu pozytonowego J-PET

Tematem niniejszej pracy inżynierskiej jest zaimplementowanie oraz przetestowanie oprogramowania służącego do wykrywania obszarów zainteresowania ROI (ang. Region of Interest) na obrazach pochodzących z tomografu pozytonowego emisyjnego J-PET. Tworzenie pracy realizowane było we współpracy z Narodowym Centrum Badań Jądrowych w Świerku, które uczestniczyło w projekcie J-PET mającym na celu stworzenie nowoczesnego i spełniającego określone wymagania tomografu pozytonowego. Oprogramowanie wytworzone w ramach pracy dyplomowej miało posłużyć do wykrywania obszarów o podwyższonej aktywności, a w przyszłości nowotworów oraz innych schorzeń we wczesnych stadiach choroby na skanach pochodzących z tego tomografu. Utworzone oprogramowanie miało być wykorzystywane w dwojaki sposób – jako program uruchamiany z terminala, który segmentowałby obraz oraz jako zestaw narzędzi programistycznych API (ang. Application Programming Interface), które w prosty sposób dałoby się wykorzystać w innych projektach.

W pierwszej kolejności pokrótce omówiono współpracę z NCBJ oraz projektem J-PET. W kolejnym rozdziale przedstawiono teoretyczne zasady działania tomografu PET, a także kroki przetwarzania obrazu, w tym **rekonstrukcję**, czyli odtworzenie trójwymiarowego obrazu pacjenta z punktów zebranych przez tomograf oraz **segmentację** – podział obrazu na obszary ze względu na określone właściwości. Algorytmy segmentacji miały być główną składową tworzonego oprogramowania. Kolejno przedstawiono implementację programistyczną zadania wraz z omówieniem zastosowanych technologii. Na koniec zaprezentowano efekty działania stworzonych programów, w tym algorytmów segmentacji oraz podsumowano pracę nad zadaniem.

W ramach niniejszej pracy powstało oprogramowanie w języku Python, w skład którego wchodzi: implementacje algorytmów wykrywania obszarów ROI oraz program uruchamiany z terminala wykonujący segmentację zrekonstruowanych obrazów tomograficznych. Zaimplementowano oraz przetestowano szereg algorytmów segmentacji, z których wybrane użyte zostały w programie. Metody należały do dwóch kategorii: **algorytmów progujących**, w których punkty przydzielane są do obszarów, jeśli ich wartość jest większa lub mniejsza od pewnego progu oraz **algorytmów bazujących na regionach**, które przyłączają do obszaru kolejne punkty, jeśli te spełniają pewne statystyczne właściwości, np. ich wartość nie odbiega nadmiernie od średniej obszaru.

Testy algorytmów segmentacji przeprowadzono na symulowanych skanach *fantoma* – plastikowego pudełka wielkości ludzkich płuc zawierającego przedmioty emitujące promieniowanie. Najlepsze efekty uzyskano przy pomocy algorytmów progujących. Metody te wykrywały większość punktów emitujących promieniowanie – obszarów ROI oraz były odporne na artefakty, zniekształcenia wprowadzane przez algorytmy rekonstrukcji obrazu.

**Słowa kluczowe:** Algorytmy, segmentacja, tomografia pozytonowa, medycyna nuklearna, programowanie, Python, NCBJ, J-PET

---

## Abstract

### Digital Image processing algorithms for Positron emission tomography J-PET

The aim of this engineering diploma thesis is to implement and verify Region of Interest (ROI) detection software on volumes that originate from positron emission tomography *J-PET* scans. Development is achieved in cooperation with the National Center for Nuclear Research (NCBJ) that participates in the J-PET project which aim is to construct a powerful novel PET scanner. Software designed during the engineering diploma has to detect regions of increased activity and in future diseases in early stages on scans performed by the J-PET tomograph. Created software has to be utilized twofold: as a script run from a terminal which segments tomography images and as a programming library (API – Application Programming Interface) which might be used in external projects.

Firstly, cooperation with NCBJ and J-PET project was explained. Then, basics of PET tomography with image processing steps such as **image reconstruction** – recreation 3D image from points detected by a scanner and **image segmentation** – partition of the image to detect ROIs. Segmentation algorithms had to be included in developed software. In the following sections implementation with used technologies were described. Finally finished software and results of image segmentation were shown.

The software created as an integral part of this thesis is written in Python programming language. It consists of ROI detection algorithms implementation and a command line script which performs segmentation of reconstructed tomographic images. A number of image segmentation algorithms are implemented and verified, some of which are contained in software. Implemented segmentation methods belong to two classes: **thresholding algorithms** that assign points to a region if point's value is higher or lower than a given threshold and **region based algorithms** that join points to a region if they meet certain statistical properties, such as points' values do not vary from the region's average value excessively.

Image segmentation algorithms were verified using simulations of tomographic scans of a phantom – a plastic box containing a few small balls that emit radiation. Best results were achieved for thresholding algorithms. These methods detected the majority of radiating objects, that were in fact ROI regions, and they were not vulnerable to image reconstruction artefacts and distortions.

**Keywords:** Algorithms, image segmentation, positron emission tomography, PET, nuclear medicine, programming, Python, NCBJ, J-PET



„załącznik nr 3 do zarządzenia nr 24/2016 Rektora PW

.....  
miejsowość i data

.....  
imię i nazwisko studenta

.....  
numer albumu

.....  
kierunek studiów

### OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płyce kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....  
czytelny podpis studenta”

---

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>6</b>
1.1	Omówienie tematu pracy inżynierskiej . . . . .	6
1.2	Projekt J-PET . . . . .	6
<b>2</b>	<b>Tomografia PET</b>	<b>8</b>
2.1	Zasada działania . . . . .	8
2.1.1	Model przetwarzania sygnału z tomografu PET . . . . .	9
2.2	Rekonstrukcja obrazu . . . . .	9
2.2.1	Cel rekonstrukcji obrazu . . . . .	9
2.2.2	Algorytmy rekonstrukcji obrazu . . . . .	10
2.3	Segmentacja obrazu . . . . .	11
2.3.1	Problem segmentacji obrazu . . . . .	11
2.3.2	Metody segmentacji obrazu . . . . .	12
2.3.3	Algorytmy segmentacji zaimplementowane w pracy dyplomowej . . . . .	14
<b>3</b>	<b>Implementacja problemu</b>	<b>17</b>
3.1	Narzędzia programistyczne . . . . .	17
3.2	Omówienie kodu źródłowego . . . . .	20
3.2.1	Klasy VolumeData i SegmentVolume . . . . .	20
3.2.2	Moduł SegmentujObrazyJPET . . . . .	22
3.3	Dane wejściowe . . . . .	23
<b>4</b>	<b>Wyniki</b>	<b>28</b>
4.1	Progowanie Yena . . . . .	28
4.2	Progowanie Otsu . . . . .	28
4.2.1	Progowanie Otsu iteracyjne . . . . .	29
4.2.2	Progowanie Otsu wielokrotne . . . . .	31
4.2.3	Region Growing oraz modyfikacja wykorzystująca próg Yena . . . . .	31
4.3	Zestawienie i porównanie algorytmów segmentacji obrazów . . . . .	34
<b>5</b>	<b>Podsumowanie</b>	<b>36</b>
5.1	Podsumowanie pracy nad zadaniem . . . . .	36
5.2	Perspektywy rozwoju oprogramowania . . . . .	38

---

# 1 Wstęp

## 1.1 Omówienie tematu pracy inżynierskiej

Tematem niniejszej pracy jest przygotowanie, zaimplementowanie oraz przetestowanie algorytmów przetwarzania obrazów w celu wykrywania *regionów zainteresowania* ROI (ang. Region of Interest) na trójwymiarowych obrazach rekonstrukcji skanowania emisyjnym tomografem pozytonowym PET (ang. Positron Emission Tomography). Znajdowanie regionów zainteresowania można uogólnić do problemu segmentacji, czyli podziału obrazu na podobszary prezentujące podobne właściwości. Przykładem segmentacji może być podział obrazu pod względem kolorów.

Badacze oraz fizycy wykorzystują obrazy PET do zilustrowania tzw. funkcjonalnej charakterystyki tkanek. Precyzyjne pomiary aktywności biologicznej umożliwiają dostrzeżenie zmian patologicznych (ang. lesion) podczas leczenia oraz po jego zakończeniu. Obrazowanie PET, dzięki dużej rozdzielczości pozwala zobaczyć przepływ krwi w ciele człowieka, a także rozprzestrzenianie się nowotworu w różnych stadiach choroby. Dokładna metoda wykrywania regionów zainteresowania jest zatem kluczowa w procesie diagnozy i przewidywania schorzeń [7].

Na etapie opracowywania zadania sprawdzono szereg algorytmów segmentujących, z których wybrane zostaną zaprezentowane w dalszej części pracy.

Głównym celem projektu jest stworzenie oprogramowania realizującego wspomniane algorytmy, które będzie spełniało następujące założenia:

- oprogramowanie, jak i technologie w nim wykorzystywane powinny być otwartoźródłowe,
- kod programów powinien być przenośny oraz łatwy w obsłudze i przyszłym wykorzystaniu, zapewniając przy tym przejrzyste opisy interfejsu użytkownika.

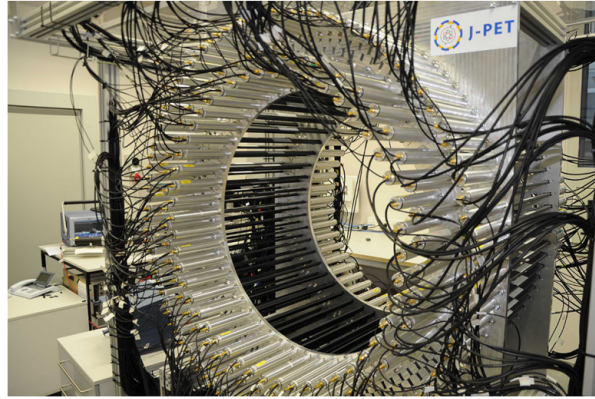
Ponadto wskazuje się, że oprogramowanie będzie mogło być używane w dwojaki sposób, tzn. jako program wywoływany z wiersza poleceń oraz jako interfejs programistyczny użytkownika API (ang. Application Programming Interface) w postaci biblioteki, używany w zewnętrznych programach.

Potencjalnym zastosowaniem pracy dyplomowej jest wykorzystanie oprogramowania w projekcie J-PET, który ma na celu m.in. stworzenie innowacyjnej metody tomografii pozytonowej. Napisana aplikacja posłuży do wykrywania regionów zainteresowania w trójwymiarowym obrazie wygenerowanym przez tomograf J-PET.

## 1.2 Projekt J-PET

Merytoryczna część pracy dyplomowej była wykonywana we współpracy z Narodowym Centrum Badań Jądrowych w Świerku, z ramienia którego osobą odpowiedzialną za opiekę merytoryczną nad projektem był dr inż. Konrad Klimaszewski. NCBJ bierze udział w projekcie naukowym, pod przewodnictwem Uniwersytetu Jagiellońskiego w Krakowie, pod nazwą J-PET – *Jagiellonian PET tomography*. Jest to międzynarodowy projekt mający na celu stworzenie tomografu pozytonowego nowej generacji, który spełniać będzie następujące założenia [1]:

- nisko-kosztowe skanowanie całego ciała,
- możliwość umieszczenia tomografu PET we wnętrzu skanera tomografii komputerowej CT lub rezonansu magnetycznego MR w celu jednoczesnego wykonania badań PET/CT lub PET/MR,



Rysunek 1: Prototyp tomografu J-PET w pełnej skali. Skaner J-PET zbudowany jest z trzech warstw polimerowych scyntylatorów (czarne) oraz 2 rzędów fotopowielaczy (jasne), po jednej z każdej strony scyntylatorów.

- skanowanie o dokładności dopasowanej do wielkości pacjenta,
- urządzenie o budowie modularnej, proste w rozbudowie zapewniając możliwość dopasowania pola widzenia skanera FOV (ang. Field Of View) do badanego obiektu, np. pacjenta.

Warto zauważyć, że obecnie bardzo często badanie PET poprzedzone jest skanem CT lub MR. Skan konwencjonalnymi metodami zapewnia kształt struktur w ciele badanego pacjenta, co pozwala na zoptymalizowanie obrazowania PET pod kątem aktualnie skanowanych obiektów, a także na zmniejszenie błędu dopasowania geometrycznego wywołanego ruchami pacjenta.

Wadą obecnie stosowanych tomografów PET jest ich cena, wynikająca głównie z zastosowanych kryształowych *scyntyatorów* – elementów emitujących błysk światła (scyntyłacje) w wyniku przechodzenia przez nie promieniowania jonizującego. Scyntylatory pełnią funkcję detektorów promieniowania gamma wywołanego pośrednio przez radiofarmaceutyk umieszczony w ciele pacjenta. Skaner J-PET ma być zbudowany ze *scyntyatorów* polimerowych. Poza niską ceną produkcji, plastikowe elementy mają być łatwe w formowaniu, a także wytwarzać relatywnie dużą ilość światła na wyjściu elementu [2]. W efekcie ma to zapewnić możliwość budowania długich, wąskich elementów za niską cenę zamiast kryształów o stosunkowo niewielkiej średnicy. Polimerowe komponenty mają cechować się także krótkim czasem odpowiedzi.

Główną wadą plastikowych scyntylatorów jest niższe prawdopodobieństwo wywołania efektu fotoelektrycznego. Zespół J-PET próbuje zutilizować ten problem poprzez zastosowanie odpowiedniej geometrii scyntylatorów, a także specjalizowanej elektroniki.

Do dnia dzisiejszego stworzony został trójwarstwowy skaner w docelowym rozmiarze, który zbiera dane eksperymentalne. Obecnie montowany jest kolejny fragment urządzenia w formie modułowej. Główne prace skupiają się na kalibracji skanera i pomiarach jego charakterystyk. Rozwijane jest oprogramowanie do przetwarzania danych z tomografu – algorytmy rekonstrukcji oraz segmentacji obrazu [4].



---

## 2 Tomografia PET

W niniejszym rozdziale omówiony zostanie koncept tomografii PET. Na wstępie przedstawione będą jej fizyczne podstawy działania oraz aktualne zastosowania. Następnie zaprezentowane zostaną elementy modelu przetwarzania sygnału z tomografu – poczynając na zbieraniu emitowanych pozytonów przez detektory urządzenia, na przekazaniu lekarzowi zbioru regionów zainteresowania kończąc. Dalej omawiane będą dwa z etapów przetwarzania sygnału, istotnych z punktu widzenia niniejszej pracy: rekonstrukcja oraz segmentacja, prezentując znane ich algorytmy oraz metody. Ostatecznie zaprezentowane zostaną algorytmy segmentacji użyte w zadaniu wykrywania obszarów ROI.

### 2.1 Zasada działania

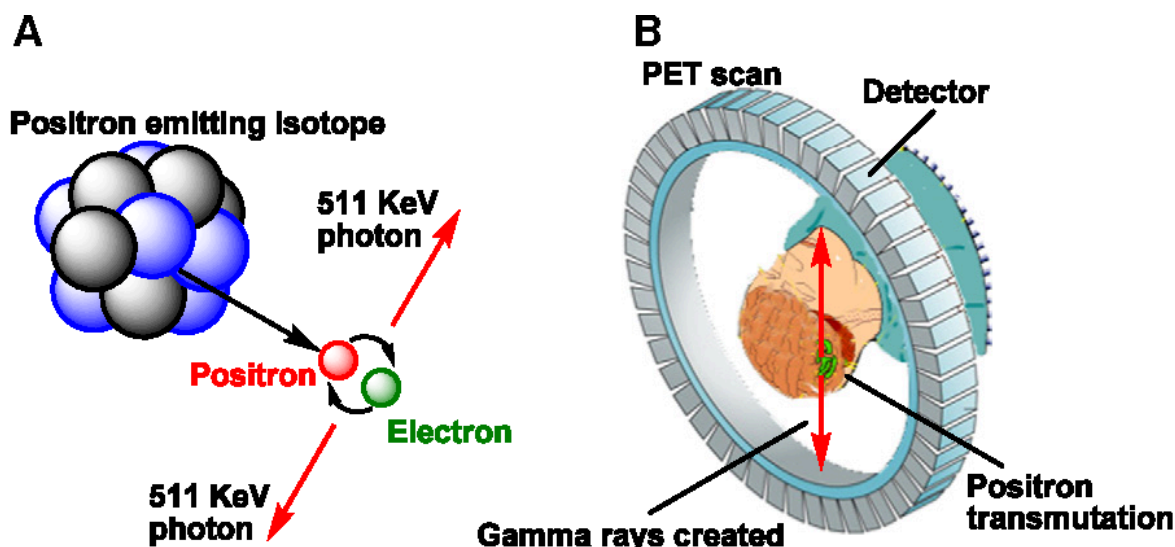
Tomografia pozytonowa emisyjna PET to technika obrazowania, która pozwala na obserwowanie zmian metabolicznych w organizmie, w celu m. in. diagnozowania chorób nowotworowych [5]. W odróżnieniu od konwencjonalnych metod badania tj. tomografii komputerowej (CT) lub rezonansu magnetycznym (MRI), które dobrze się sprawdzają przy badaniu struktur, na przykład narządów, tomografia PET dostarcza informacje na poziomie molekularnym [8]. Obecnie tomografię pozytonową wykorzystuje się w połączeniu z wcześniej wspomnianymi badaniami tomografią komputerową - PET/CT oraz rezonansem magnetycznym - PET/MR. Tomografia komputerowa i rezonans magnetyczny niosą informację o anatomii człowieka, co pozwala na dokładne umiejscowienie informacji o funkcjach metabolicznych uzyskanych z tomografii pozytonowej. Skompilowane obrazy z obu badań pozwalają na wykrywanie zmian w organizmie we wczesnej fazie rozwoju.

**Przebieg badania tomografem PET.** Badanemu pacjentowi podaje się, w formie dożylnego zastrzyku, radiofarmaceutyk – preparat zawierający izotopy\* promieniotwórcze. Obecnie najczęściej stosuje się izotop fluoru  $^{18}\text{F}$  wbudowany w odpowiednik glukozy. Samo badanie wygląda identycznie, jak w przypadku tomografii komputerowej — pacjent leżąc na łóżku przesuwany jest wewnątrz skanera, natomiast czas trwania obrazowania wynosi do 40 min [6]. Lekarz otrzymuje trójwymiarowy obraz pacjenta, który w trakcie analizy może zostać uzupełniony o regiony zainteresowania.

**Opis fizyczny.** Izotop wprowadzony do człowieka ulega rozpadowi beta plus, w wyniku którego emitowany jest pozyton, czyli antycząstka elektronu o przeciwnym ładunku. Pozyton po przebyciu drogi ok. 0.6 mm (dla fluoru  $^{18}\text{F}$ ) zderza się z elektronem, ulegając *anihilacji*, procesowi polegającym na zamianie cząstki i antycząstki w parę kwantów promieniowania elektromagnetycznego, fotonów, poruszających się w przeciwnych kierunkach. Przemierzające się fotony wykrywane są przez scyntylatory zawarte w skanerze PET. Scyntylatory to substancje emitujące błyski światła pod wpływem promieniowania jonizującego. Światło następnie trafia na fotodiody, które zamieniają promieniowanie elektromagnetyczne na wyjściowy sygnał elektryczny [5]. Opisane zjawiska zostały przedstawione na poniższym rysunku.

---

\*odmiany tego samego pierwiastka o różnej liczbie neutronów



Rysunek 2: Opis fizyczny emisji cząstek w tomografie PET

Źródło: <https://www.semanticscholar.org/paper/Radiopharmaceutical-chemistry-for-positron-emission-Li-Conti/f115bcf13a1eab819af72e0023333512bf6a31ff/figure/0>

### 2.1.1 Model przetwarzania sygnału z tomografu PET

Na powyższym rysunku przedstawiony został model przetwarzania sygnału pochodzącego z tomografu. W procesie anihilacji powstaje para fotonów poruszających się w przeciwnych kierunkach. Obie cząstki są wykrywane przez fotodiody, zamieniające światło na sygnały elektryczne. Sygnałem wyjściowym z tomografu jest zatem zbiór par jednoczesnych (z dokładnością do kilku ns) wykryć fotonów przez detektory znajdujące się na linii łączącej je ze źródłem promieniowania. Mówiąc precyzyjniej – dane wyjściowe odbierane z tomografu to tabela zawierająca następujące kolumny: współrzędne 1. detektora, współrzędne 2. detektora, czasy detekcji. Kolejno wykonywana jest rekonstrukcja, czyli algorytmy komputerowe odtwarzające trójwymiarowy rozkład gęstości promieniowania radiofarmaceutyka [9]. Wynik rekonstrukcji poddaje się następnie algorytmom segmentacji, które mają na celu podzielić trójwymiarowy obraz na obszary wg. zadanych parametrów. Efektem segmentacji może być wskazanie regionów zainteresowania będących na przykład obszarami rozprzestrzeniania się nowotworu. Wynik segmentacji z zaznaczonymi ROI przekazywany jest do lekarza badającego pacjenta do dalszej analizy.

W tym rozdziale zostaną omówione dwa istotne etapy przetwarzania obrazu pochodzącego z tomografu PET oraz znane metody ich realizacji. Etapy, o których mowa to **rekonstrukcja** oraz **segmentacja**.

## 2.2 Rekonstrukcja obrazu

### 2.2.1 Cel rekonstrukcji obrazu

Podczas anihilacji pozytonu pochodzącego z radiofarmaceutyka powstają dwa fotony o energii 511 keV\* każdy, poruszające się w przeciwnych kierunkach. Gdy oba fotony zostaną wykryte w tym samym czasie, z dokładnością do kilku ns, na ich podstawie tworzy się *zdarzenie koincydencji* (ang. coincidence event). Linia łącząca dwa detektory, przechodząca przez ciało pacjenta

\*1eV =  $1,6 * 10^{-19} J$

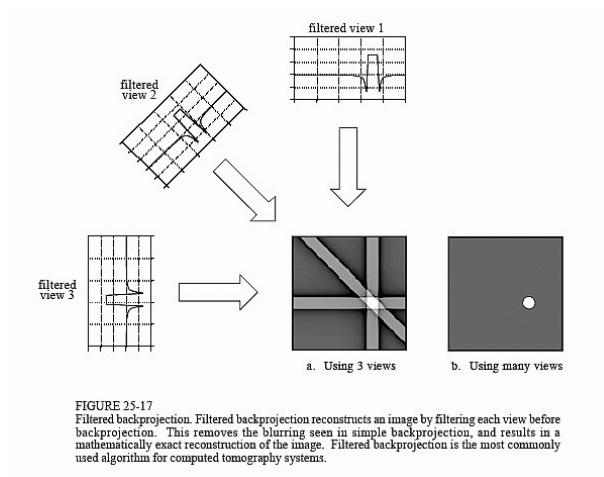
nazywana jest *linią odpowiedzi*, *LOR* (ang. Line of Response). Nie uwzględniając niepożądanych zjawisk, takich jak szumy, maksymalna liczba zdarzeń koincydencji wykryta przez dwa fotodetektory jest proporcjonalna do ilości radiofarmaceutyka znajdującego się na linii odpowiedzi LOR. Cel rekonstrukcji określa się zatem, jako stworzenie trójwymiarowego rozkładu gęstości radiofarmaceutyka na podstawie zebranych zdarzeń koincydencji [9].

### 2.2.2 Algorytmy rekonstrukcji obrazu

Opracowanych zostało wiele algorytmów rekonstrukcji obrazu. Obecnie najczęściej używane są algorytmy: **Filtrowana Projektja Wsteczna**, FBP (ang. Filtered Back Projection) oraz **Estymata maksymalnego prawdopodobieństwa warunkowego**, MLEM (ang. Maximum-likelihood Estimation Method). W ramach tej pracy omówione zostaną natomiast metody: FBP oraz **Projektja Wsteczna z Maksymalną Zmiennością**, BPTV (Back Projection Total Variation), gdyż zostały wykorzystane w projekcie J-PET [10].

**Algorytm Filtrowanej Projektji Wstecznej (FBP).** Jest to metoda analitycznej rekonstrukcji obrazu, szeroko wykorzystywana w medycynie. Załóżmy, że otrzymaliśmy skończoną liczbę projekcji (rzutów) źródła emitującego fale elektromagnetyczne. Rzuty były wykonywane z krokiem  $45^\circ$ . Owe projekcje pochodzą ze zbioru zdarzeń koincydencji ze skanera. Każdy z rzutów poddawany jest filtracji filtrem rampowym, który uwydatnia krawędzie obrazu. Następnie wykonywane jest sumowanie rzutów korespondujących z rozpatrywanym wokselem – projekcja wsteczna. Nie zastosowanie filtru rampowego powodowałoby, że obraz wyjściowy byłby rozmyty. Aby zapobiec wystąpieniu negatywnych skutków tej filtracji, stosuje się dodatkowo filtr dolnoprzepustowy w przypadku zaszumionych danych [11].

Opisany algorytm obrazuje poniższy rysunek.



Rysunek 3: Algorytm FBP

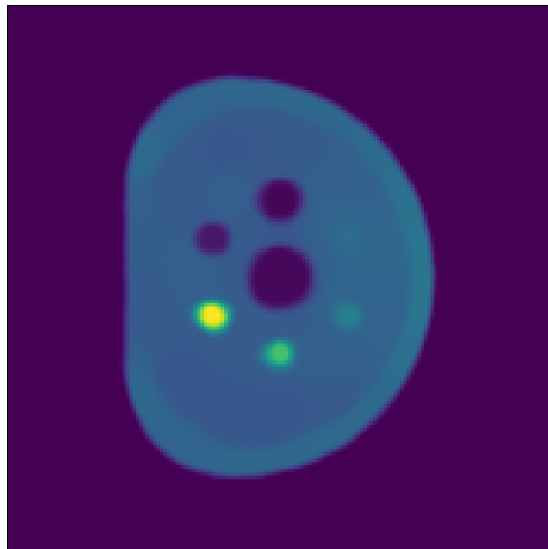
Źródło: <https://www.dspguide.com/ch25/5.htm>

**Algorytm Filtrowanej Projektji wstecznej z Maksymalną Zmiennością (BPTV).** W drugiej metodzie projekcja wsteczna oraz filtracja dolnoprzepustowa występują w odwrotnej kolejności niż w FBP: najpierw odbywa się projekcja wsteczna, a potem filtracja. Ponieważ

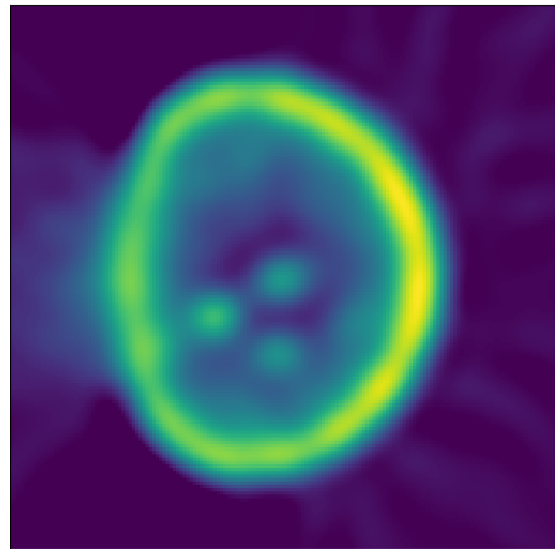
---

obraz ze skanera PET jest nie deterministyczny, tzn. na jego postać wpływa szum gaussowski, nie jest możliwe proste odwrócenie działań. W celu rozwiązania tego problemu stosowana jest regularyzacja, czyli wprowadzenie dodatkowej informacji do modelu, w celu poprawienia jakości rozwiązania. W algorytmie BPTV zadanie regularyzacji wykonuje *Maksymalna Wariancja* (TV) – znana technika odszumiania obrazu. Jak się okazuje, rekonstrukcja metodą BPTV prezentuje znacznie lepsze właściwości i obraz rekonstrukcji tą metodą będzie używany do rozwiązania problemu segmentacji.

Poniżej umieszczono porównanie obu metod.



(a) Rekonstrukcja BPTV.



(b) Rekonstrukcja FBP.

Rysunek 4: Metody BPTV oraz FBP rekonstrukcji obrazu. Algorytm BPTV działa znacznie lepiej – krawędzie elementów są ostre, a kolory łatwo rozróżnialne.

## 2.3 Segmentacja obrazu

### 2.3.1 Problem segmentacji obrazu

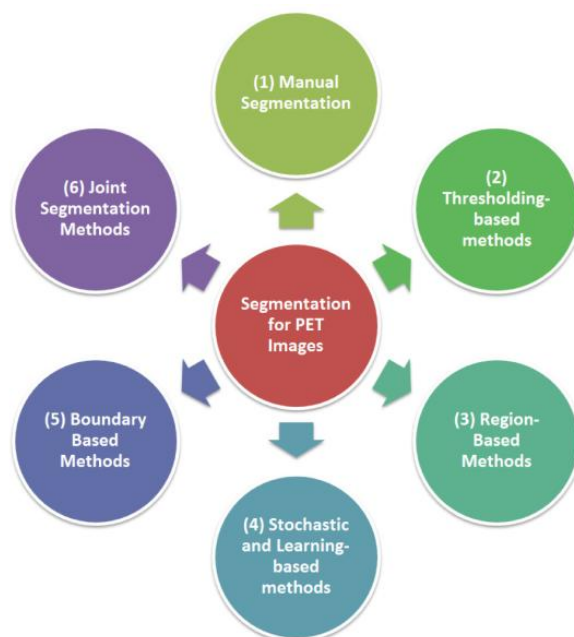
Mając odtworzony obraz 3D za pomocą algorytmów rekonstrukcji, następnym etapem w przetwarzaniu obrazu z tomografu będzie znajdowanie istotnych obszarów z punktu widzenia radiologii. Zadanie to wykonuje segmentacja – proces partycjonowania obrazu cyfrowego na obszary wykonywany w celu otrzymania obrazu łatwiejszego w analizie [12].

Segmentacja jest w ogólności zadaniem trudnym. Trudność ta wynika ze złożoności oraz różnorodności ludzkiego organizmu. Co więcej, dochodzą problemy widoczne na etapie rekonstrukcji, wynikające ze skanowania tomografem – artefakty skanowania, szumy, niski kontrast. To powoduje, że krawędzie struktur są słabo rozpoznawalne, co znacznie utrudnia znajdowanie ROI. Wspomniana różnorodność ciała człowieka szczególnie odznacza się przy badaniu różnych narządów. Segmentacja płuc napotykać będzie inne problemy, niż segmentacja jelita grubego. Wszystko to sprawia, że nie ma obecnie algorytmu segmentacji dostarczającego akceptowalnych wyników dla **każdego** medycznego zbioru danych [13].

W tym rozdziale zostaną przedstawione grupy algorytmów segmentacji. W dalszej części zaprezentowane zostaną metody zaimplementowane w programistycznej części pracy dyplomowej.

### 2.3.2 Metody segmentacji obrazu

W celu lepszego zkontrastowania metod segmentacji, stosowany jest podział na następujące grupy:



Rysunek 5: Rodzaje metod segmentacji obrazów trójwymiarowych

Źródło: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4060809/bin/nihms-590656-f0003.jpg>

- Metody ręczne bazujące na wiedzy i doświadczeniu radiologa
- Metody oparte o progowanie
- Metody bazujące na regionach
- Algorytmy oparte o stochastykę i uczenie maszynowe
- Metody oparte o znajdowanie krawędzi na obrazach
- Metody łączone wykorzystujące skanowanie innymi metodami takimi jak CT lub MRI

Poniżej zostaną opisane wszystkie z przedstawionych rodzajów algorytmów segmentacji.

**Metody ręczne.** Najprostszym podejściem, niewymagającym znajomości algorytmów komputerów i programowania, jest segmentacja manualna. Ze względu na czasochłonność poszukiwane są metody automatyczne które mogłyby ją zastąpić. Metoda ta jest często stosowana przy porównywaniu algorytmów segmentacji.

Grupa radiologów ręcznie (lub przy pomocy komputera) zaznacza na wolumenie obszar, który na podstawie wiedzy oraz doświadczenia sklasyfikuje jako interesujący. Manualne obrysowywanie ROI jest najpowszechniejszym sposobem na otrzymanie *metody odniesienia* (ang. ground truth), która posłuży do porównywania innych, bardziej skomplikowanych metod segmentacji.

---

**Progowanie.** Powyższych problemów nie posiada metoda segmentacji oparta o progowanie, czyli technikę przekształcającą obraz w skali szarości w obraz binarny (binaryzacja). Woksele\* o wartości wyższej od pewnego progu otrzymują wartość 1, a 0 — w przeciwnym przypadku. W ogólności progów może być więcej, a wartości przydzielane wokselom to na przykład liczby równo rozłożone z zakresu 0 - 255. Najważniejszą kwestią w przypadku algorytmów progujących jest dobór odpowiedniego progu. Można wymienić trzy podgrupy metod opartych o progowanie: ze stałym progiem (fixed thresholding), z progiem dobieranym adaptacyjnie (adaptive thresholding) oraz progowanie iteratywne (iterative thresholding).

**Progowanie ze stałym progiem.** W tej metodzie poziom jest stały przez cały etap działania segmentacji. Może być zadany przez eksperta radiologa, nauczony maszynowo na podstawie obrazów podobnego typu lub wyłuskany przez specjalne algorytmy wykorzystujące właściwości statystyczne takie jak histogram (algorytm Minimum), wariancja międzyklasowa (algorytm Otsu), średnia. Wspomniane modyfikacje zostaną szerzej opisane w dalszej części, ponieważ zostały zaimplementowane w niniejszej pracy.

**Progowanie z progiem dobieranym adaptacyjnie.** Ta metoda jest rozwinięciem metody ze stałym progiem. Zadany próg jest optymalizowany ze względu na parametry obrazu takie jak: stosunek pierwszy plan – tło, średnia lub wariancja wszystkich wartości, kontrast. Poza tym próg może być dostosowywany do przedmiotu obrazowania – rozpatrywana intensywność może zależeć od rodzaju badanej tkanki. Na przykład: na obrazie płuc obszary ROI mogą mieć wyższą wartość niż na obrazie mózgu. Progowanie progiem dobieranym adaptacyjnie nie przyniosło oczekiwanych rezultatów przy wykrywaniu ROI.

**Progowanie iteratywne.** W tej metodzie progowanie wybranym progiem powtarza się kilkakrotnie, liczba iteracji podawana jest przez analityka jako hiperparametr. Okazuje się, że takie podejście przynosi efekt. Dużą zaletą progowania iteratywnego jest fakt, że algorytm nie potrzebuje informacji na temat całego obiektu, tak jak w przypadku powyższej metody, w której wymagana była wiedza o parametrach takich jak średnia, stosunek pierwszy plan – tło. Progowanie iteratywne zostało zaimplementowane w tej pracy.

**Metody bazujące na regionach.** Kolejną omawianą grupą algorytmów segmentacji obrazów trójwymiarowych będzie segmentacja oparta o regiony, w których znajduje się obszary jednorodne, o podobnych właściwościach wokseli, np. wartości są zbliżone do średniej obszaru. Metoda nie bazuje jednak na intensywnościach wokseli. Można wyróżnić dwie podgrupy metody bazujących na regionach: przyrastanie regionów (ang. Region Growing) oraz metody oparte o grafy.

**Przyrastanie regionów - Reion Growing.** Region Growing to metoda wykorzystująca statystyczne właściwości obrazów takie jak średnia oraz odchylenie standardowe. Wymaga podania dwóch parametrów: ziarno (ang. seed) – punkt startowy dla algorytmu oraz parametr określający, jak wartość wokseli może różnić się od średniej. Algorytm Region Growing dobrze sprawdzał się przy wykrywaniu ROI. Zostanie opisane szerzej w rozdziale 3 – Implementacja.

---

\*Woksel – najmniejszy element w dyskretnej przestrzeni trójwymiarowej; odpowiednik piksela w 2D

---

**Metody oparte o grafy.** Drugą podgrupą algorytmów bazujących na regionach są metody oparte o grafy. W tych metodach obraz przedstawia się w formie grafów – węzły to woksele obrazu, natomiast w krawędziach zapisywana jest informacja o podobieństwie węzłów. Użycie grafów do opisu obrazu ma spore zalety wiążące się z dobrze rozwiniętą Teorią Grafów. Do grupy metod opartych o grafy należą: algorytm Graph-cut oraz Random walk.

**Stochastyka i uczenie maszynowe.** Kolejną grupą wykorzystywaną w segmentacji obrazów tomograficznych jest stochastyka oraz uczenie maszynowe. Algorytmy oparte o stochastykę wykorzystują różnice statystyczne w wyekstrachowanych regionach i otaczających tkankach. Metody wykorzystujące uczenie maszynowe do segmentacji znajdują i rozpoznają cechy na obrazie. Obie te grupy wymagają pewnej dużej grupy obrazów do stworzenia modelu stochastycznego lub do nauczenia algorytmu uczącego. Tomograf PET tworzony przez zespół J-PET nie posiada jak dotąd wystarczająco dużej bazy danych, aby móc sprostać tym zadaniom. W związku z tym algorytmy wykorzystujące stochastykę i machine learning nie będą omawiane szczegółowiej oraz implementowane w pracy inżynierskiej.

**Metody wykrywania krawędzi.** W tej grupie można wydzielić dwa podstawowe algorytmy: technika aktywnych konturów oraz metoda gradientowa. Obie metody odporne są na szumy w obrazach tomograficznych.

Pierwsza z nich składa się z dwóch etapów: najpierw wyznacza się kontur początkowy – na podstawie wskazania analityka, a następnie iteracyjnie przybliża się kontur optymalny. W kolejnych iteracjach dąży się do osiągnięcia stanu równowagi pomiędzy siłami wewnętrznymi (zależnymi od aktualnego stanu konturu) oraz zewnętrznymi (zależnymi od gradientu całego obrazu) [15].

Natomiast gradientowa metoda wykrywania krawędzi polega na znajdowaniu na obrazie dużych zmian w intensywnościach — wysokiego gradientu [9]. Jak już zostało wspomniane wcześniej, obrazy pochodzące z tomografu PET cechują się niskim kontrastem, wobec czego wykrywanie nagłych zmian intensywności na obrazie jest znacznie utrudnione. Implementacja algorytmów wykrywających krawędzie nie znalazły się w tej pracy.

**Metody łączone.** W literaturze [9] przedstawiony został szereg hybrydowych metod segmentacji wykorzystujących dwie lub więcej metod segmentacji. Używa się w ten sposób zalety różnych algorytmów. Dla przykładu – algorytm Region Growing wymaga podania ziarna – punktu startowego algorytmu. Ziarno można uzyskać używając innej metody, na przykład progując, a następnie pobierając punkt z centroidy powstałego obiektu.

W praktyce proces przetwarzania obrazów jest zadaniem znacznie bardziej skomplikowanym, niż użycie jednej metody segmentacji. Na początku wykonuje się *preprocessing*, który powinien wyselekcjonować ciało człowieka. Do takiego zadania dobrze sprawdzi się progowanie lub algorytm wykrywania krawędzi. Obraz pochodzący z rekonstrukcji często zawierać będzie szum, który powinien zostać wyeliminowany z obrazu przy pomocy filtru dolnoprzepustowego. Następnie, w celu wykrywania ROI, można użyć innych metod segmentacji obrazów.

### 2.3.3 Algorytmy segmentacji zaimplementowane w pracy dyplomowej

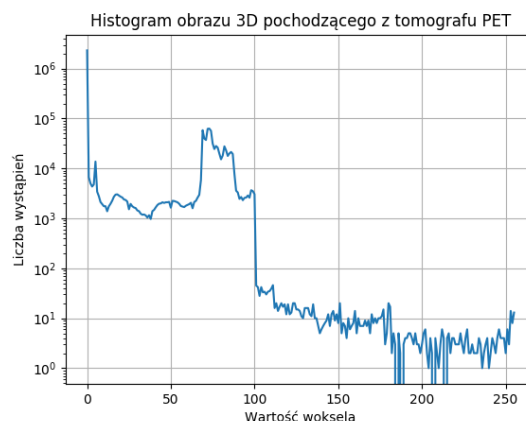
Podczas praktycznej implementacji problemu segmentacji obrazów trójwymiarowych przetestowano znaczną liczbę algorytmów. Większość z nich jednak nie przyniosła oczekiwanych re-

---

zultatów w związku z problemami opisanymi na początku rozdziału. Ostatecznie najbardziej konkurencyjnymi algorytmami okazały się proste metody tj. progowanie oraz Region Growing. W rozwiązywaniu problemu znajdowania regionów zainteresowania ROI zostały zaimplementowane lub wdrożone (istniejące implementacje) następujące algorytmy:

- Progowanie Otsu – implementacja Scikit Image,
- Progowanie Yena – implementacja Scikit Image,
- Region Growing – implementacja własna.

**Algorytm progujący Otsu.** Metoda Otsu to technika progowania globalnego, czyli podział całego wolumenu na pierwszy plan oraz tło na podstawie jednego progu. Metoda Otsu wykorzystuje histogram – rozkład intensywności wokseli, nie bierze natomiast pod uwagę przestrzennego rozłożenia wokseli na obrazie. Można założyć, że histogram zawiera dwa maksima – jedno dla tła, drugiego dla pierwszego planu. Należy stworzyć dwie klasy na obrazie, tak aby intensywności wewnątrz klas były jak najbardziej skupione wokół wartości średniej. Mówiąc ściślej dąży się do znalezienia takiego progu, który podzieli wartości histogramu na dwie klasy i zminimalizuje wariacje wewnątrz obu klas (wariacje wewnątrzklasowe).



Rysunek 6: Przykład histogramu. Na rysunku są widoczne dwa maksima – dla pierwszego planu (wartość ok. 0) oraz dla tła (wartość ok. 80)

Źródło: Własne

W pracy inżynierskiej zaimplementowane zostały dwie modyfikacje algorytmu progującego Otsu. Obie opierały się na poszerzeniu bazowej wersji algorytmu o kolejne progi, które odpowiadać będą obszarom innego typu, dla przykładu: tło, organy obojętne na działanie radiofarmaceutyka, obszary silnie aktywne.

**Algorytm iteracyjny Otsu.** W pierwszej modyfikacji, algorytm wykonywany jest w pętli, za każdym razem zwracając wyższy próg. Liczbę iteracji wskazuje użytkownik programu. Algorytm iteracyjny Otsu pomimo prostej implementacji zwracał bardzo dobre wyniki dla obrazów symulacji skanowania tomografem PET, co zostało przedstawione w rozdziale 4 – Wyniki.



---

**Algorytm Otsu dwuwartościowy.** Kolejno zaimplementowano modyfikację algorytmu Otsu zaprezentowanej w odniesieniu [16]. Znajdowanie odpowiedniego progu nie jest optymalne obliczeniowo, gdyż sprawdzana jest każda wartość progu oraz odpowiednie wariancje. Należy pamiętać, że jak dotąd brano pod uwagę, ze względu na prostszą analizę, wartości progu z przedziału 0 - 255, czyli liczby które można zapisać przy pomocy 8 bitowej liczby całkowitej bez znaku. Skanowanie może mieć znacznie dokładniejszą reprezentację, na przykład za pomocą liczby zmiennoprzecinkowej pojedynczej precyzji (float), liczby z przedziału [0,1). W taki wypadku liczba progów mogłaby wynosić nawet 1 mld.\* W modyfikacji algorytmu Otsu zaproponowano usprawnienie metody *siłowej* (ang. brute-force) poprzez przybliżanie krzywej histogramu. Eksperymenty w artykule [16] wskazały, że złożoność obliczeniowa obniżyła się 19000 razy względem metody brute-force.

Jednoprogową metodą Otsu polega na tym, że wybrany próg minimalizuje wariancje wewnątrzklasowe, co jest równoważne maksymalizowaniu wariancji **międzyklasowych** –  $t = ARGMAX\sigma^2(t)$ , gdzie  $t$  - poszukiwany próg,  $\sigma^2(t)$  - wariancja międzyklasowa dla progu  $t$ . Tę metodę można łatwo uogólnić do wieloprokowej metody Otsu:

$t_1, t_2, \dots, t_n = ARGMAX\sigma^2(t_1, t_2, \dots, t_n)$ , gdzie  $t_1, t_2, \dots, t_n$  - zbiór progów,  $\sigma^2(t_1, t_2, \dots, t_n)$  - suma wariancji międzyklasowych dla progów  $t_1, t_2, \dots, t_n$ .

Segmentacja dwupoziomowa Otsu znalazła zastosowanie w pracy inżynierskiej, a jej działanie zaprezentowane zostało w rozdziale Wyniki.

**Algorytm progujący Yena.** Kolejnym algorytmem segmentacji wykorzystującym progowanie jest algorytm progujący Yena. Metoda Yena bierze pod uwagę dwa czynniki: różnicę między binaryzowanym obrazem a oryginałem oraz ilość pamięci zajmowaną przez segmentowany obraz. Na podstawie tych dwu czynników tworzy się funkcję kosztu. Dowiedzono, że owa funkcja kosztu posiada minimum, które określa optymalny próg segmentacji obrazu trójwymiarowego.

W implementacji pracy inżynierskiej skorzystano z gotowej implementacji algorytmu Yena z pakietu języka programowania Python – Scikit Image.

**Algorytm Region Growing.** Kolejnym, ostatnim już algorytmem opisanym w tej pracy będzie metoda Region Growing. Jest to prosty algorytm wykorzystujący statystyczne miary, takie jak średnia i wariancja. Tworzony obszar rozrasta się dołączając kolejne punkty.

Zaczynając od punktu startowego podanego przez użytkownika, algorytm sprawdza sąsiadujące woksela i dodaje je do regionu jeśli spełniają następujące równanie:

$$f(x, y, z) - avg(R) \leq c * \sigma$$

gdzie:  $f(x,y,z)$  – intensywność woksela,  $avg(\mathbf{R})$  – średnia intensywności wokseli w aktualnym obszarze,  $\sigma$  – odchylenie standardowe intensywności aktualnego obszaru,  $c$  – parametr podawany przez użytkownika, który mówi, jak bardzo intensywności regionu mogą się od siebie różnić.

Istotnym zagadnieniem jest wybór punktu startującego, tzw. ziarna. Jedną z możliwości może być podanie za punkt startowy woksela wyekstrahowanego przy pomocy progowania Yena. Taka hybrydowa metoda skutecznie segmentowała obszary zainteresowania.

---

\*<https://lemire.me/blog/2017/02/28/how-many-floating-point-numbers-are-in-the-interval-01/>

---

## 3 Implementacja problemu

W tym rozdziale opisana zostanie implementacyjna część zadania. Na początku przedstawione będą użyte narzędzia programistyczne wraz z uargumentowaniem ich zastosowania. Dalej omówiony zostanie kod źródłowy programu, skupiając się na zależnościach między modułami. Na końcu zaprezentowane będą dane wejściowe algorytmów segmentacji, czyli obrazy wynikowe rekonstrukcji skanowania tomografem PET.

### 3.1 Narzędzia programistyczne

We wstępie pracy inżynierskiej przedstawione zostały wymagania dotyczące tworzonego oprogramowania. Przede wszystkim kod miał być przenośny, a użyte technologie otwartoźródłowe, aby program mógł stać się integralną częścią projektu J-PET. Z punktu widzenia programisty istotna byłaby obecność bibliotek służących do analizy i przetwarzania obrazu cyfrowego. Równie ważna jest dostępność materiałów i pomocy na temat języków i technologii stosowanych w zadaniu.

Wszystkie te wymagania spełnia język Python.

**Python.** Jest to język programowania ogólnego zastosowania cieszący się ogromną popularnością. Prosta i klarowna składnia, a także niezliczone dostępne pakiety oraz biblioteki sprawiają, że język ten znajduje wiele profesjonalnych zastosowań takich jak Uczenie Maszynowe, przetwarzanie danych, programowanie webowe, przetwarzanie obrazów. Ważną zaletą Pythona jest jego otwartoźródłowość. Tworzenie oraz korzystanie z bibliotek jest darmowe, a kod źródłowy wielu narzędzi udostępniony jest publicznie w Internecie. To sprawia, że z Pythona korzystają nie tylko programiści związani ze światem nauki, ale także firmy tworzące profesjonalne oprogramowanie w celach zarobkowych.

Projekt został napisany w całości w Pythonie, ze względu na wspomniane zalety tego języka. Paradymaty (sposoby tworzenia oprogramowania), jakie towarzyszyły tworzeniu, to programowanie **proceduralne** oraz **obiektywne**. W programowaniu proceduralnym program przedstawia się jako ciąg instrukcji, procedur. Każda z nich wykonuje ściśle określone zadanie, przepływ danych w programie tworzą wartości zwracane przez procedury. Programowanie proceduralne to forma w jakiej najłatwiej przedstawić algorytmy komputerowe. W programowaniu obiektywnym program przedstawia się za pomocą relacji między obiektami różnych klas, dla których definiuje się dane (pola) oraz zachowanie (metody). Klasy mogą między sobą dziedziczyć, dzięki czemu w prosty sposób tworzona jest struktura i hierarchia obiektów i ich użycie w programie. Programowanie obiektywne ma ułatwić ponowne wykorzystywanie fragmentów kodu.

W celu wspomaganie przetwarzania obrazów cyfrowych powstały między innymi pakiety Scikit Image oraz OpenCV (Open Computer Vision).

**Scikit Image.** Pierwszy z nich jest integralną częścią tzw. ekosystemu SciPy – biblioteki naukowej Pythona dostarczającej narzędzi inżynierskich oraz naukowych. Biblioteka Scikit Image posiada narzędzia służące do komputerowej obróbki obrazów cyfrowych. Należą do nich: morfologia, filtry, narzędzia rysujące, konwertery kolorów, narzędzia wykrywające cechy w obrazie.

**Open CV.** Jest to bardzo rozbudowana biblioteka o otwartym kodzie źródłowym posiadająca oprócz narzędzi standardowego przetwarzania obrazów, metody wykorzystujące uczenie maszy-

**Worldwide, Oct 2018 compared to a year ago:**

Rank	Change	Language	Share	Trend
1	↑	Python	24.72 %	+5.4 %
2	↓	Java	22.01 %	-0.7 %
3	↑	Javascript	8.4 %	+0.1 %
4	↑	C#	7.71 %	-0.4 %
5	↓↓	PHP	7.42 %	-1.6 %
6		C/C++	6.32 %	-0.5 %
7		R	4.11 %	-0.1 %
8		Objective-C	3.29 %	-0.9 %
9		Swift	2.69 %	-0.8 %
10		Matlab	2.08 %	-0.3 %
11		Ruby	1.63 %	-0.4 %
12	↑	TypeScript	1.54 %	+0.1 %
13	↓	VBA	1.48 %	-0.0 %
14		Scala	1.22 %	-0.2 %
15		Visual Basic	1.14 %	-0.2 %
16	↑	Kotlin	1.04 %	+0.3 %
17	↑	Go	0.96 %	+0.3 %
18	↓↓	Perl	0.65 %	-0.2 %
19		Lua	0.39 %	+0.0 %
20	↑	Rust	0.38 %	+0.0 %

Rysunek 7: 20 najpopularniejszych języków programowania w październiku 2018. Popularność języka Pythona rośnie z roku na rok. W październiku 2018 programowało w nim ok. 24% programistów na świecie.

Źródło: <https://www.houseofbots.com/news-detail/4504-1-top-20-most-popular-programming-languages-to-learn-in-2019>

nowe. W przeciwieństwie do Scikit Image, Open CV skupia się również na zastosowaniach czasu rzeczywistego. Biblioteka składa się z kilku tysięcy algorytmów zoptymalizowanych pod różne zastosowania, w tym: wykrywanie i rozpoznawanie twarzy, identyfikowanie obiektów, śledzenie obiektów i ruchów kamery, ekstrahowanie modeli 3D obiektów. Narzędzie zostało napisane w języku C++, wobec czego cechuje go szybkość działania. W celu umożliwienia współpracy z innymi językami programowania, w tym Python, zostały utworzone interfejsy [18].

W trakcie implementacji algorytmów wykrywania ROI zostały wykorzystane wyłącznie narzędzia z pakietu Scikit Image. Dostępne moduły okazały się wystarczające na potrzeby zadania, nie było potrzeby sięgania po znacznie bardziej rozbudowane systemy. W oprogramowaniu zostały wykorzystane narzędzia takie jak: morfologia, progowanie, konwersje kolorów.

**Pozostałe pakiety.** Kolejno przedstawione zostaną pozostałe narzędzia i biblioteki użyte w projekcie.

**Numpy.** Podstawową biblioteką niezbędną przy operacjach na macierzach danych (obraz cyfrowy przedstawia się jako macierz 2D lub 3D) jest pakiet **NumPy**. Podobnie jak SciKit Image przynależy do ekosystemu SciPy i jest biblioteką otwartoźródłową. Posiada implementacje macierzy N-wymiarowych oraz wszystkie metody niezbędne przy wykonywaniu operacji na macierzach – proste tworzenie, działania arytmetyczne i logiczne, znajdowanie elementów, indeksowanie, operacje statystyczne na macierzach, takie jak histogram.

**Matplotlib.** Kolejnym powszechnie wykorzystywanym pakietem, pochodzącym również z ekosystemu SciPy jest **Matplotlib**. Ta biblioteka umożliwia proste tworzenie rysunków, wykresów, ich edycję oraz formatowanie, a także tworzy okna do wyświetlania figur. W projekcie Matplotlib został wykorzystany do tworzenia rysunków oraz ich wyświetlania na ekranie.

**Narzędzia z podstawowej biblioteki Pythona 3.7.** Warto przedstawić narzędzia dostarczane przez bibliotekę standardową Pythona. Pakiety wraz z krótkim opisem zobrazowane zostały na poniższej tablicy.

Pakiet	Opis
os	Interfejs do systemu operacyjnego. Pozwala na zarządzanie plikami, procesami i wątkami systemu, właściwościami systemu operacyjnego.
datetime	Pobieranie aktualnej daty oraz czasu, formatowanie ciągów datowych.
pickle	Zapis i odczyt zmiennych oraz struktur danych, np. tablic do specjalnego formatu binarnego – pickle.
argparse	Proste tworzenie konsolowych interfejsów użytkownika. Pakiet pozwala na dodawanie argumentów wywołania oraz wyświetlanie pomocy do programu
json	Wczytywanie/zapis w formacie JSON.

Tablica 1: Pakiety biblioteki standardowej użyte w projekcie

---

## 3.2 Omówienie kodu źródłowego

Kod programów utworzonych w ramach projektu znajduje się w repozytorium w witrynie GitHub [4]. Głównym celem implementacji pracy inżynierskiej miało być stworzenie oprogramowania do znajdowania obszarów zainteresowania ROI na obrazach pochodzących ze skanowania tomografem PET. Żeby to osiągnąć, zostały opracowane oraz zaimplementowane algorytmy segmentacji, których opis znajduje się w podrozdziale 2.3.3. Program miał być wykorzystywany w dwojaki sposób: jako program wywoływany z wiersza poleceń oraz jako zestaw narzędzi z dobrze opisaną dokumentacją, czyli Programistyczny Interfejs Użytkownika API.

Projekt *Wykrywanie obszarów zainteresowania w obrazach tomografu J-PET*, w skrócie J-PET ROI Detection podzielony został na 3 pliki:

1. `VolumeData.py` – plik zawierający implementację klasy `VolumeData`, która odpowiada za operacje na obrazach 3D tj. wczytywanie z pliku, wstępna obróbka (preprocessing), wyświetlanie przekrojów oraz zapis do pliku,
2. `SegmentData.py` – klasa `SegmentData` zawierająca implementację algorytmów segmentacji i wykrywania obszarów zainteresowania,
3. `SegmentujObrazyJPET.py` – skrypt odpowiadający za tekstowy interfejs użytkownika, główny plik projektu, wywoływany z konsoli.

Pokrótkie zostaną omówione kolejne moduły.

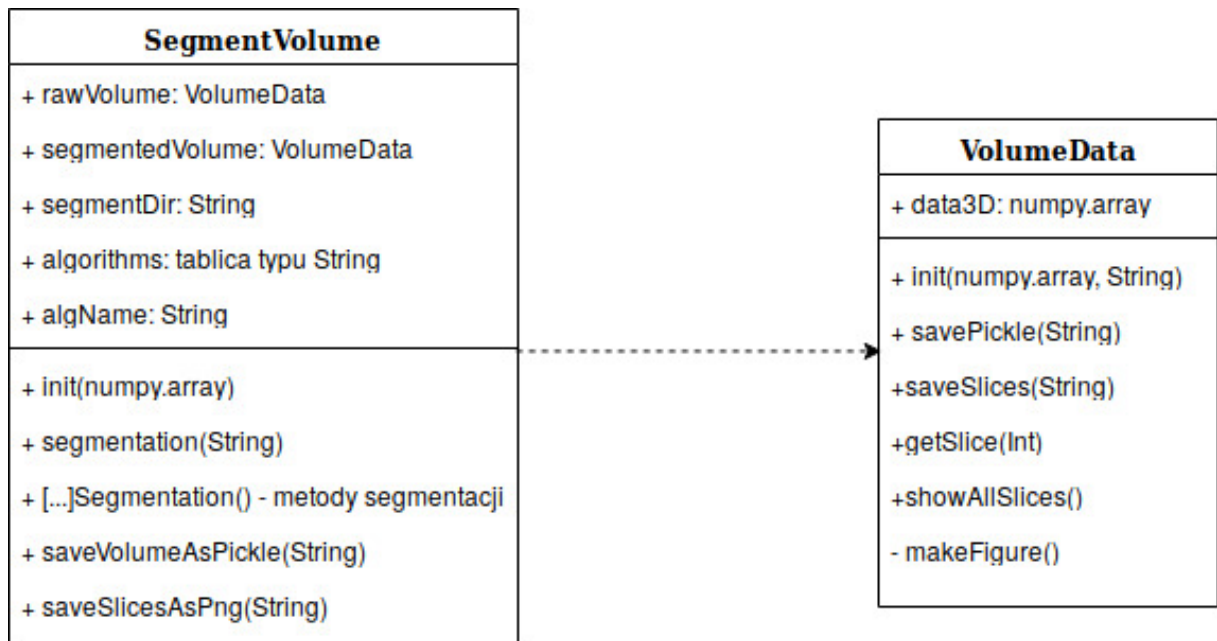
### 3.2.1 Klasy `VolumeData` i `SegmentVolume`

Szkielet projektu tworzą dwie klasy: `VolumeData` oraz `SegmentVolume`. Wykonują główne zadania przewidziane w ramach projektu, czyli segmentują obraz pochodzący z tomografu PET. Razem tworzą interfejs użytkownika API, który w łatwy sposób można zaimportować do innego projektu. Relacja obu klas oraz ich składowe przedstawione zostały na poniższym diagramie UML. Kolejno omówione będą szczegóły implementacyjne obu klas.

**Klasa `VolumeData`.** Zanim wykonane zostaną algorytmy segmentacji, dane pochodzące z rekonstrukcji należy odpowiednio wczytać oraz sformatować do odpowiedniej struktury danych. Zadania te wykonuje klasa `VolumeData`. Dane mogą pochodzić z pliku: tekstowego lub pickle bądź bezpośrednio ze struktury danych typu `numpy.array`. W celu utworzenia obiektu klasy `VolumeData` należy podać jako argument inicjalizatora \* przyjmującego dane z pliku (jako ścieżka typu `String`) lub z macierzy `NumPy`. Jeśli dane wejściowe pochodzą z pliku tekstowego, powinny być ułożone w macierz, w której wiersze to zestaw współrzędnych `x,y,z` oraz odpowiadająca im intensywność promieniowania. Macierz punktów jest następnie konwertowana do macierzy trójwymiarowej, w której tracona jest informacja o bezwzględnych wartościach współrzędnych. We wczytywaniu danych z pliku wykorzystywana jest funkcja pakietu `NumPy` - `loadtxt`. Dane wejściowe mogą być podane w formie macierzy 3D – wtedy, gdy będą podane w postaci zmiennej typu macierz (również z pliku pickle).

---

\*W języku Python konstruktor jest ukryty. Dla programisty jest dostępny inicjalizator, który nie tworzy obiektu, a inicjalizuje go podanymi argumentami, jeżeli takie są.



Rysunek 8: Diagram klas utworzonych na cele zadania. Klasa SegmentVolume wykorzystuje obiekty klasy VolumeData.

Po wczytaniu danych źródłowych wartości konwertowane są do docelowego 8 bitowego typu całkowitego bez znaku *uint8*, o zakresie wartości 0-255. Gotowe dane w formie macierzy 3D zapisywane są w polu *data3D*.

Klasa VolumeData posiada także inne metody takie jak: zapis macierzy do pliku pickle – *savePickle*, pobranie wybranego przekroju obrazu 3D – *getSlice*, wygenerowanie oraz zapis kolarzu zawierającego wszystkie przekroje wolumenu – *saveSlices*.

**Klasa SegmentVolume.** Kluczowym zadaniem z punktu widzenia pracy inżynierskiej miało być zaimplementowanie algorytmów wykrywania obszarów zainteresowania na obrazach rekonstrukcji skanów PET. Za działanie tych algorytmów odpowiada klasa SegmentVolume. Obiekty tej klasy przechowują obiekty typu VolumeData z danymi oryginalnymi – *rawVolume* oraz posegmentowanymi – *segmentedVolume* (powstaje po wykonaniu jednego z algorytmów segmentacji). Metody klasy SegmentVolume zawierają implementacje algorytmów segmentacji (teoretyczny opis algorytmów umieszczony został w podrozdziale 2.3.2 – *Algorytmy segmentacji zaimplementowane w pracy dyplomowej*:

- Progowanie Yena (*yenThreshSegmentation*) – segmentacja poprzez progowanie, próg znajdujący przy pomocy algorytmu Yena w implementacji Scikit Image. Nie pobiera żadnych parametrów.
- Progowanie Otsu iteracyjne (*otsuIterSegmentation*) – iteracyjne progowanie progiem Otsu (implementacja bazująca na Scikit Image), w kolejnych iteracjach próg jest określany na nowo na podstawie obszaru o wartościach intensywności powyżej poprzedniego progu. Parametrem metody jest liczba iteracji.
- Progowanie Otsu wielokrotne (*otsuMultiSegmentation*) – segmentacja przez progowanie wykorzystujące dwa progi Otsu. Metoda Otsu została uogólniona na wielokrotną liczbę

---

progów na podstawie [16]. Metoda nie pobiera parametrów.

- Region Growing (regionGrowingSegmentation) – implementację algorytmu Region Growing można przedstawić następująco (bazując na opisie w Rozdziale 2.3.2):
  1. Obszar zainteresowania zainicjuj punktem startowym.
  2. Sprawdź, czy woksele z każdej strony obszaru spełniają równanie algorytmu (wartość woksela pomniejszona o średnią obszaru ma być niższa od odchylenia standardowego). Jeśli tak, dodaj je do obszaru.
  3. Powtarzaj punkt 2. do czasu aż woksele przestaną spełniać warunek algorytmu.

Parametrami algorytmu są punkty startowe (algorytm mógł być zainicjalizowany równoległe w kilku miejscach) oraz mnożnik odchylenia standardowego ograniczający wielkość tworzonych obszarów.

- Region Growing wykorzystujący próg Yena (yenThreshRegionSegmentation) – dla wolumenu obliczany jest próg Yena, a następnie znajdowane są obszary o wartościach wyższych niż ten próg. Centroidy obszarów podawane są do algorytmu regionGrowingSegmentation, jako parametry punkty startowe. Parametry algorytmu są identyczne jak w przypadku RegionGrowingSegmentation.

Algorytmy segmentacji można wywoływać bezpośrednio lub pośrednio przez metodę segmentation, której podaje się nazwę algorytmu oraz parametry tego algorytmu. Drugi sposób jest zalecany, gdyż bezpośrednie wywołanie metody odpowiadającej algorytmu nie aktualizuje pola *segmentedVolume*, a jedynie zwraca posegmentowany wolumen. Parametry muszą być podane w postaci słownika. W takiej formie program główny SegmentujObrazyJPET wykorzystuje interfejs API. Oprócz metod implementujących algorytmy, obiekty SegmentVolume mogą również zapisać wyniki segmentacji tj. przekroje wolumenu – *saveSlicesAsPng* oraz sam wolumen *saveVolumeAsPickle*.

### 3.2.2 Moduł SegmentujObrazyJPET

Jest to program główny projektu – obsługuje komunikację tekstową z użytkownikiem. Interfejs konsolowy został stworzony przy użyciu pakietu *argparse*, który umożliwia dodawanie argumentów oraz opcji wywołania, a także tekstu pomocy programu. Oprócz definicji *parser*, w pliku znalazły się również stałe <sup>†</sup> typu string, które przechowują napisy oraz informacje wyświetlane w terminalu w komunikacji tekstowej z użytkownikiem.

Struktura wywołania skryptu wygląda następująco:

```
python SegmentujObrazyJPET.py [--savePickle] [--saveSlices]
[--algParams] algName dataPath
```

Listing 1: Składnia wywołania programu SegmentujObrazyJPET.py

Komenda składa się z dwóch argumentów wymaganych tj. nazwa algorytmu *algName* oraz ścieżka do danych *dataPath*. Zostały zdefiniowane także argumenty opcjonalne: flagi zapis

---

<sup>†</sup>Nie jest to do końca prawda. W Pythonie nie zostało zdefiniowane słowo kluczowe typu `const`. W zamian stosuje się konwencję taką, że nazwy zmiennych, które mają pozostać niezmiennione, pisze się wielkimi literami.

danych wyjściowych segmentacji do pliku pickle – *savePickle* oraz zapis przekrojów wolumenu do pliku graficznego png – *saveSlices*, a także parametry przekazywane do algorytmów *algParams*. W poniższej tabeli przedstawione zostały udostępnione algorytmy oraz ich parametry.

Algorytm	Argument wywołania	Parametry
Progowanie Yena	yen-thresh	–
Progowanie Otsu iteratywne	otsu-iter	<i>iterCount</i> - liczba iteracji algorytmu
Progowanie Otsu wieloprogowe	otsu-multi	–
Region Growing	region-growing	<i>startPoints</i> - lista punktów startowych, <i>c</i> - mnożnik odchylenia std
Region Growing wykorzystujący próg Yena	yen-region	<i>c</i> algorytmu Region Growing

Tablica 2: Opis wywołania algorytmów oraz ich parametry.

Parametry przekazywane do programu powinny być podane w formacie JSON <sup>‡</sup>. Dane w takim formacie można w łatwy sposób wczytać do zmiennej typu słownik, a następnie odwoływać się do konkretnych wartości poprzez ich nazwy. Przykładowe użycie programu zostało zaprezentowane poniżej.

```
python SegmentujObrazyJPET.py --savePickle --saveSlices
--algParams='{ "startPoints": [[1,1,1], [10,10,10]], "c": "2" }'
algName='region-growing' dataPath='/home/John/reconstruction.txt'
```

Listing 2: Przykładowe wywołanie programu SegmentujObrazyJPET.py

Wywołanie komendy spowoduje wykonanie segmentacji algorytmem Region Growing o parametrach: punkty startowe: (1,1,1), (10,10,10); mnożnik odchylenia: 2 na danych znajdujących się w pliku */home/John/reconstruction.txt*. Wyniki segmentacji: dane wyjściowe oraz przekroje wolumenu zostaną zapisane na dysku.

### 3.3 Dane wejściowe

W celu zapewnienia jak najlepszej jakości obrazu, urządzenia skanujące ciało człowieka, w tym tomograf J-PET, muszą być przetestowane i zoptymalizowane. Amerykańskie *Narodowe Stowarzyszenie Producentów Aparatury Elektryczne* NEMA (ang. National Electrical Manufacturers Association) wprowadziło standard pozwalający na oszacowanie jakości obrazowania tomografów [21]. Jedną z opracowanych metod badania tomografów PET jest skanowanie **Fantomu NEMA**, czyli plastikowego modelu płuc zaprojektowanego w oparciu o zalecenia NEMA oraz IEC (Międzynarodową Komisję Elektrotechniczną) [22].

**Fantom NEMA.** W zamkniętej obudowie w kształcie zbliżonym do klatki piersiowej, o maksymalnym wymiarze 18 cm, znajdują się: trzon oraz 4 niewielkie kulki pełniące funkcję obszarów aktywnych, o średnicach od 10 mm do 37 mm.

<sup>‡</sup>JavaScript Object Notation – prosty i lekki format przekazywania danych w formie tekstowej [19].

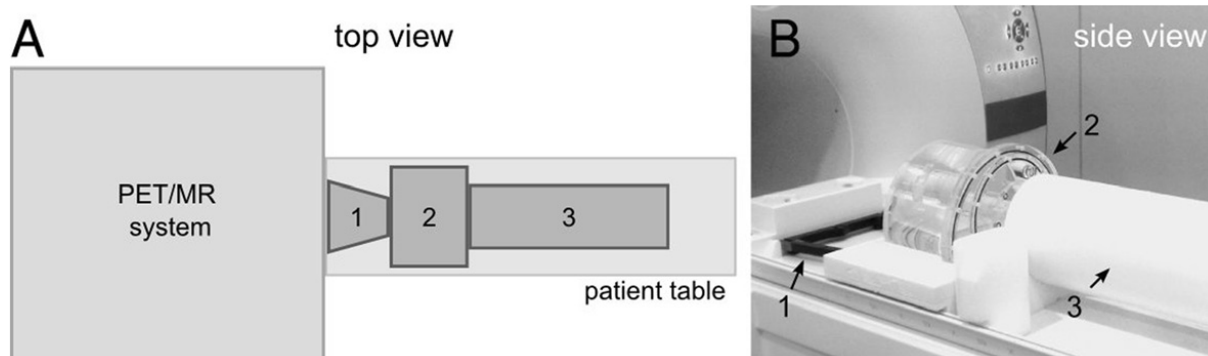




Rysunek 9: Fantom NEMA – plastikowy model płuc służący do badań nad tomografią PET.

Źródło: [http://www.spect.com/pub/NEMA\\_IEC\\_Body\\_Phantom\\_Set.pdf](http://www.spect.com/pub/NEMA_IEC_Body_Phantom_Set.pdf)

Fantom umieszcza się na wysuwanym stole tomografu razem z separatorem oraz innym fantomem symulującym ciało człowieka, które jest neutralne dla skanera [21].



Rysunek 10: Instalacja Fantoma NEMA w celu badania tomografu.

Źródło: [21]

Fantom NEMA znajduje zastosowania m. in. w symulacjach skanów całego ciała w tomografach PET, a także na podstawie otrzymanych obrazów, w optymalizacji algorytmów rekonstrukcji.

**Rekonstrukcje Fantoma NEMA po skanowaniu tomografem PET.** Zgodnie z informacjami w Rozdziale 2.2 – Rekonstrukcja obrazu, zastosowane zostały dwie metody rekonstrukcji obrazów PET: Filtrowana propagacja wsteczna FBP oraz Propagacja Wsteczna z Maksymalną Zmiennością BPTV. Implementacje powyższych algorytmów są aktywnie rozwijane przez zespół z Narodowego Centrum Badań Jądrowych pod kierownictwem dr inż. Lecha Raczyńskiego w ramach projektu J-PET. Przykładowe wyniki rekonstrukcji zostały dostarczone w formie plików

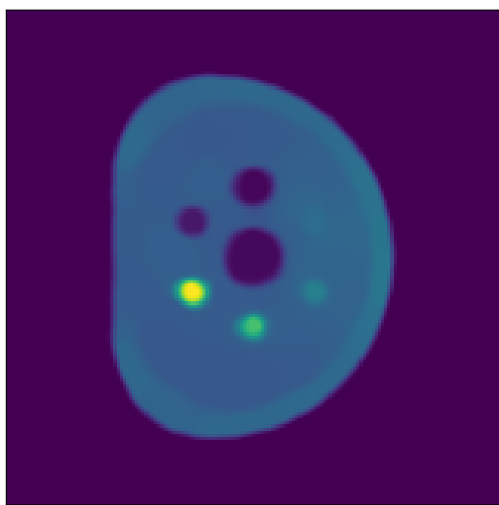
---

tekstowych zawierających tabelę z współrzędnymi oraz odpowiadającymi im wartościami. Do takiego formatu został przystosowany inicjalizator obiektów klasy **VolumeData** wczytującej dane.

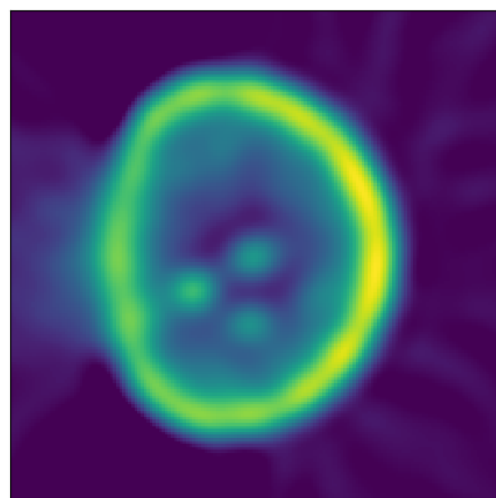
Poniżej przedstawione zostały przykładowe przekroje obrazów 3D po wykonaniu algorytmów rekonstrukcji FBP oraz BPTV. Wczytanie danych z pliku, a następnie wyświetlenie ich zostało wykonane następującymi poleceniami utworzonego API:

```
vol_bptv = VolumeData('rekonstrukcja_bptv.txt')
vol_bptv.showAllSlices()
```

Listing 3: Utworzenie obiektu `VolumeData`. Wczytanie i wyświetlenie danych z rekonstrukcji BPTV



(a) Rekonstrukcja BPTV.



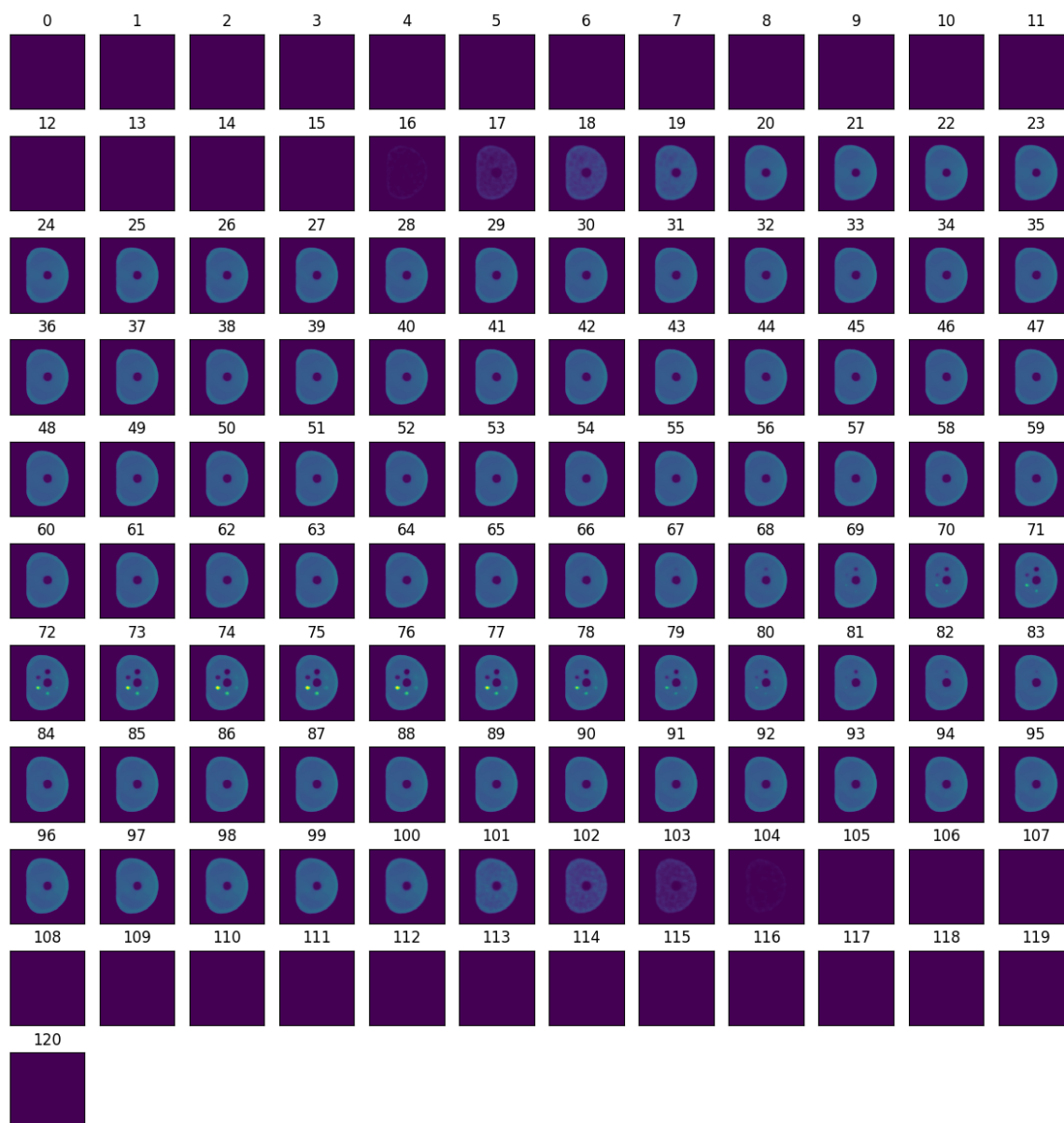
(b) Rekonstrukcja FBP.

Rysunek 11: Metody BPTV oraz FBP rekonstrukcji obrazu. Algorytm BPTV działa znacznie lepiej – krawędzie elementów są ostre, a kolory łatwo rozróżnialne.

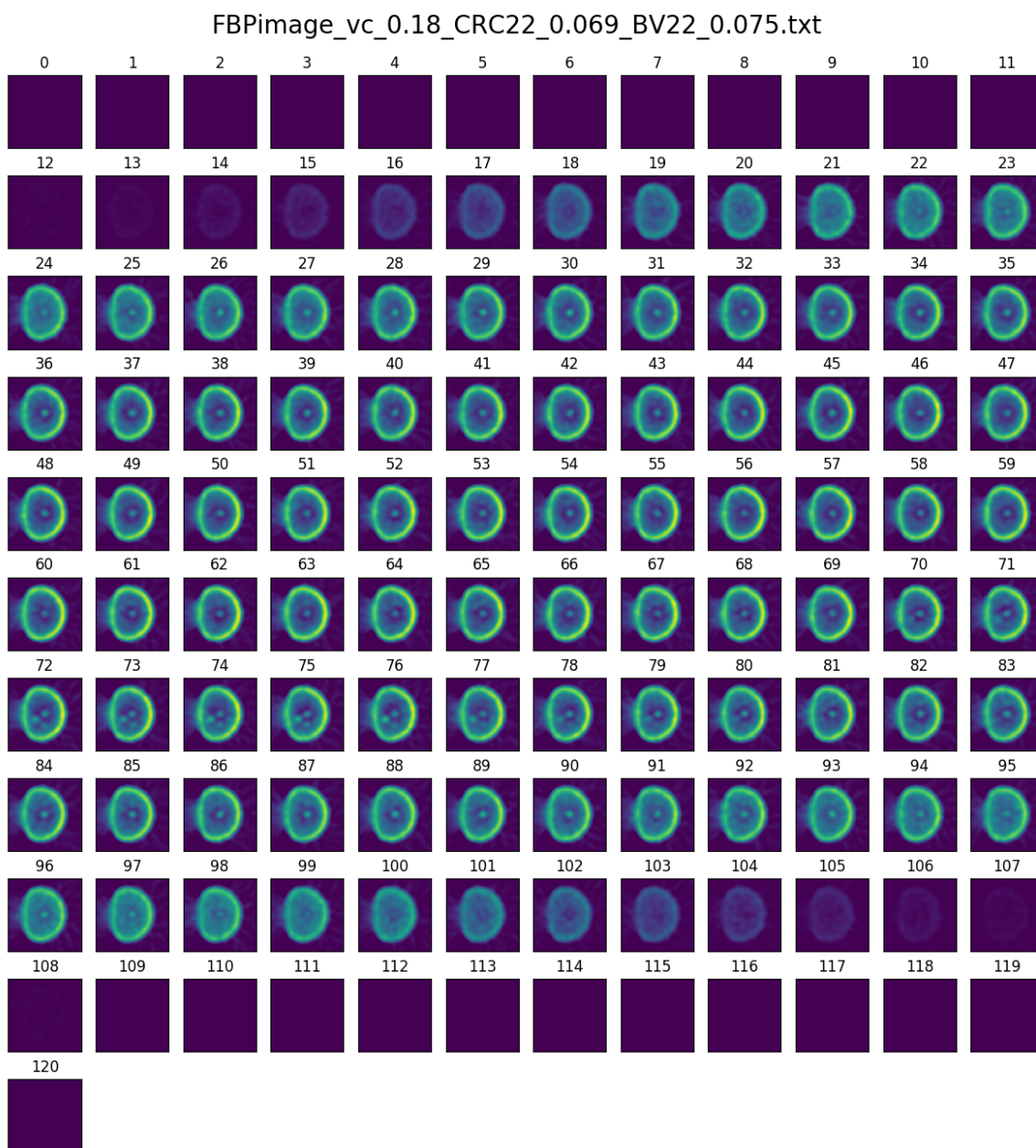
Barwy na powyższych obrazach nie są prawdziwe. Powstały w wyniku zastosowanej skali barw narzędzia do wyświetlania obrazów pakietu Matplotlib. W rzeczywistości obrazy rekonstrukcji są w skali szarości, podobnie jak plik z danymi wejściowymi.

Warto nawdmienić, że w przypadku rekonstrukcji algorytmem FBP nie przeprowadzono poprawki na tłumienie AC (ang. attenuation correction). Powoduje to, że krawędzie fantoma cechują się największą jasnością na obrazie. Utrudnia to zadanie segmentacji, która miałaby wyodrębnić właściwe obszary aktywne. Na obu obrazach rekonstrukcji widoczne są elementy takie jak: tło, trzon, dwie kule – o najniższych jasnościach, bliskich zeru oraz 4 obiekty o większych jasnościach, różniących się także wielkością. Rekonstrukcja BPTV, zgodnie z oczekiwaniami i opisem w Rozdziale 2, daje znacznie lepsze rezultaty od metody FBP. W celu praktyczniejszego porównania algorytmów segmentacji, w którym pomijalne będą negatywne wpływy rekonstrukcji obrazów, zostały wykorzystane obrazy otrzymane w wyniku rekonstrukcji BPTV.

BPTVimage\_CRC22\_0.664\_BV22\_0.049.txt



Rysunek 12: Rekonstrukcja symulacji skanowania tomografem J-PET metodą BPTV.



Rysunek 13: Rekonstrukcja symulacji skanowania tomografem J-PET metodą FBP.

---

## 4 Wyniki

W kolejnym rozdziale przedstawione zostaną wyniki działania poszczególnych algorytmów segmentacji na obrazach pochodzących z rekonstrukcji BPTV. W celu rzetelnego porównania, jako dane wejściowe użyty został ten sam wolumen. Obraz składał się z 161 x 161 x 121 wokseli. Każdy z wokseli był sześcianiem o boku 25 mm. Ich jasność miała zakres 0-255. Algorytmy były wykonywane na całych wolumenach, natomiast w niniejszej pracy zaprezentowane zostaną jedynie pojedyncze przekroje o numerze 75 z 120 – ze względu na zachowanie czytelności pracy.

**Wykorzystane algorytmy.** Implementacje algorytmów segmentacji obrazów pochodzących z tomografu omówione zostały w podrozdziale 3.2 – Klasy VolumeData i SegmentVolume. Prze-testowane zostały następujące algorytmy segmentacji:

- Progowanie Yena (`yenThreshSegmentation`)
- Progowanie Otsu iteracyjne (`otsuIterSegmentation`). W tym algorytmie zmieniana będzie liczba iteracji (parametr `iterCount`): od 1 do 5. Skutkować będzie to tworzeniem kolejnych progów, a w efekcie segmentowane będą obszary o innych właściwościach.
- Progowanie Otsu wielokrotne (`otsuMultiSegmentation`).
- Region Growing (`regionGrowingSegmentation`). Do algorytmu wprowadzane były różne punkty startowe (parametr `startPoints`), mnożnik odchylenia standardowego (parametr `c`).
- Region Growing wykorzystujący próg Yena (`yenThreshRegionSegmentation`). Wykorzystane parametry jak wyżej.

Efekt działania każdego algorytmu (przekrój nr 75 z 121) zestawiony został z obrazem (przekrojem 75) po rekonstrukcji BPTV.

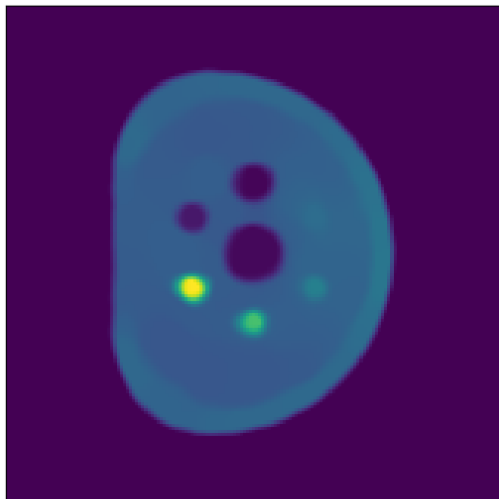
W pierwszej kolejności przedstawione będą algorytmy progujące. Ich istotą jest podział obrazu na dwa lub więcej obszarów ze względu na ich intensywności, w zależności od progów wyznaczonych metodami Yena oraz Otsu.

### 4.1 Progowanie Yena

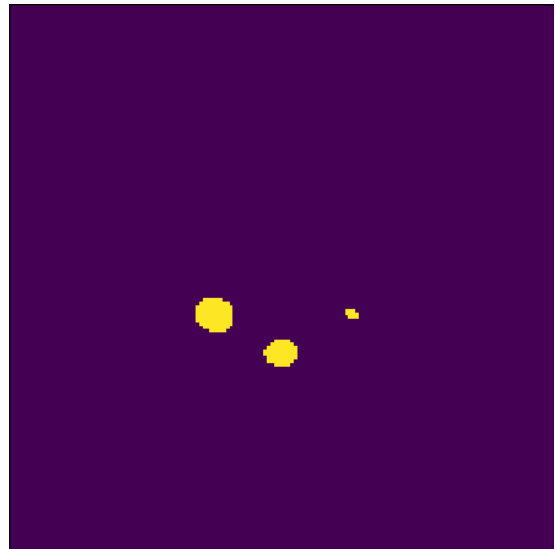
W pierwszej kolejności przetestowany został algorytm progujący Yena. Implementacja programistyczna bezpośrednio wykorzystuje próg zwracany przez funkcję `otsu_yen()` z pakietu Scikit Image. Algorytm nie przyjmuje żadnych parametrów. Wynik poszukiwania obszarów ROI metodą progowania Yena przedstawiony został na Rysunku 14. Przekrój po prawej stronie jest obrazem binarnym, o wartościach 0 oraz 1, gdzie 0 wskazuje tło, natomiast 1 – pierwszy plan, czyli obszar zainteresowania. Można zaobserwować, że algorytm skutecznie wykrył 3 z 4 jasnych punktów na obrazie rekonstrukcji, pozostałe woksele zidentyfikowane zostały jako tło. Działanie prostego algorytmu jakim jest progowanie Yena można uznać za poprawne.

### 4.2 Progowanie Otsu

Kolejno sprawdzono poprawność segmentacji algorytmem progującym Otsu. Zaimplementowane zostały dwie modyfikacje tej metody – iteracyjna oraz wielokrotna. Obie są rozwinięciem podstawowej metody Otsu.



(a) Obraz przed segmentacją.



(b) Obraz po progowaniu Yena.

Rysunek 14: Segmentacja obrazu algorytmem progującym Yena.

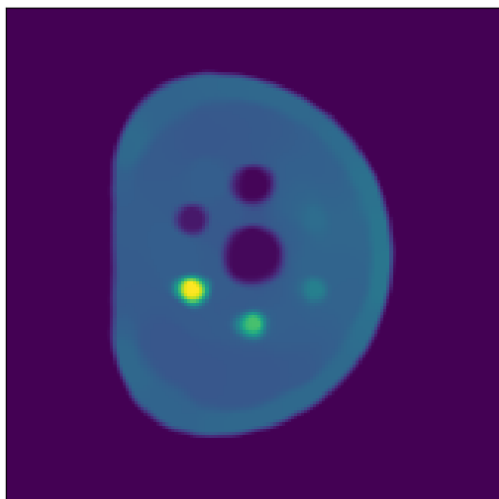
#### 4.2.1 Progowanie Otsu iteracyjne

Implementacja progowania Otsu iteracyjnego przyjmuje za parametr liczbę progów, na podstawie których intensywności obrazu przydzielone zostaną do konkretnych obszarów. Parametr oznaczać będzie liczbę obszarów, na które obraz zostanie podzielony. Dla liczby iteracji równej 1 metoda przedstawia działanie zwykłego progowania Otsu. Wartość parametru musi być liczbą całkowitą większą lub równą od 1, natomiast maksymalna wartość nie jest określona. Doświadczalnie sprawdzono, iż nie powinna być większa od 5-6. Poniżej przedstawione zostały efekty progowania algorytmem Otsu iteracyjnym dla liczby iteracji równej: 1,2,3,5,6.

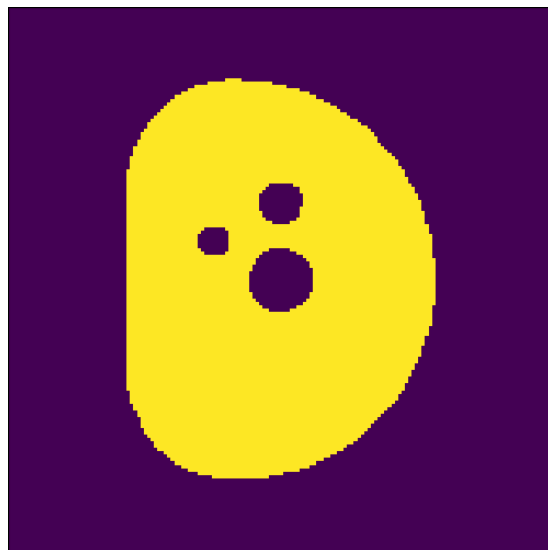
Dla liczby progów:

- 1 – (binaryzacja) algorytm nie znajduje poprawnie obszarów zainteresowania, a jedynie uwypukla krawędź fantoma,
- 2 – interesujące punkty trafiły do jednego obszaru, jednak wraz z dużym obszarem przy krawędzi fantoma,
- 3 – jasne punkty trafiły do tego samego obszaru wraz małym regionem przy krawędzi fantoma,
- 5 i 6 – punkty o wysokich intensywnościach zostały przydzielone do 3 różnych regionów – jest to efekt niepożądany, gdyż nie daje jasnej informacji, czy wydzielony fragment obrazu jest obszarem zainteresowania.

Za rezultat zgodny z oczekiwaniami, tzn. poprawnie segmentujący obraz tomograficzny można uznać algorytm o liczbie iteracji równej 3. Segmentacja Otsu 1 progiem może być wykorzystana jako narzędzie do wykrywania krawędzi fantoma lub pacjenta. Pokazano, że użycie algorytmu o wartości parametru większej od 5 nie zwraca jasnej informacji na temat obecności obszaru zainteresowania.



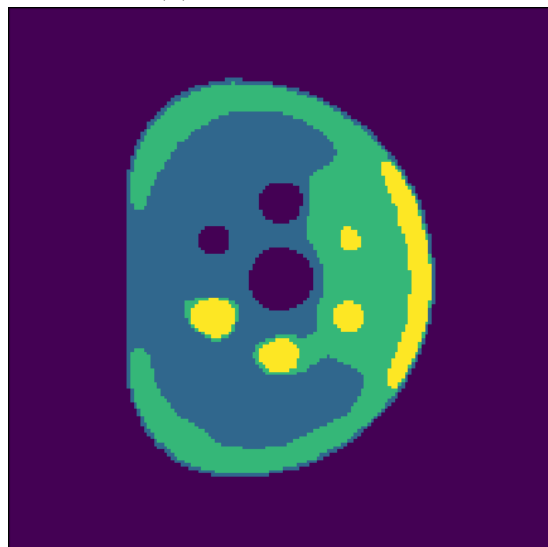
(a) Obraz przed segmentacją.



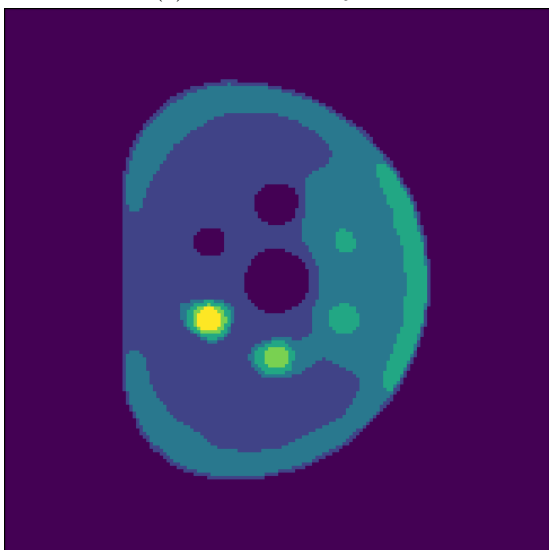
(b) Liczba iteracji: 1.



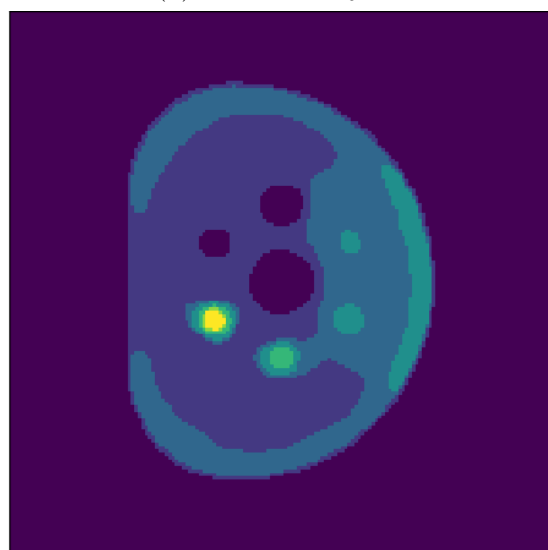
(c) Liczba iteracji: 2.



(d) Liczba iteracji: 3.



(e) Liczba iteracji: 5.



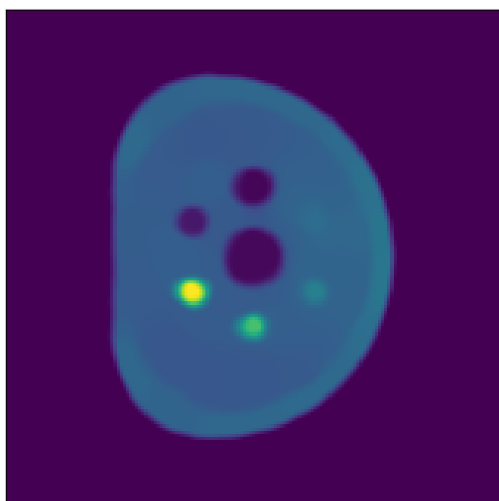
(f) Liczba iteracji: 6.

Rysunek 15: Segmentacja obrazu algorytmem progującym Otsu o różnej liczbie iteracji.

---

#### 4.2.2 Progowanie Otsu wielokrotne

Oprócz iteracyjnej metody Otsu zaimplementowana została również modyfikacja progowania wielokrotnego, którą opisano w podrozdziale 2.2 - Segmentacja. Wielokrotny algorytm Otsu znajduje równocześnie dwa progi dzielące obraz na 3 obszary. Algorytm nie przyjmuje żadnych parametrów sterujących.



(a) Obraz przed segmentacją.



(b) Obraz po progowaniu wielokrotnym Otsu.

Rysunek 16: Segmentacja obrazu algorytmem Otsu wielokrotnym.

Na pierwszy rzut oka działanie wielokrotnego progowania Otsu zwraca identyczne wyniki jak progowanie iteracyjne Otsu dla liczby progów: 2.

W istocie wartości progów dla metody iteratywnej w tym przypadku wynosiły 37 oraz 77, natomiast dla metody wielokrotnej 43 oraz 77<sup>§</sup>. Są to bardzo zbliżone progi, jedynie środkowy obszar metody wielokrotnej jest nieznacznie większy niż dla metody iteratywnej.

#### 4.2.3 Region Growing oraz modyfikacja wykorzystująca próg Yena

Zaimplementowano również algorytm niekorzystający z progowania. Taką metodą jest Region Growing, którego opis przedstawiony został w Rozdziale 2.3. Implementacja algorytmu posiada 3 parametry sterujące: punkty startowe na obrazie oraz mnożnik odchylenia standardowego ( $c$ ), który ogranicza wielkość tworzonych obszarów.

Zastosowany został następujący plan eksperymentów:

1. Sprawdzenie algorytmu ze względu na podany punkt startowy: punkt  $(0,0)$ , punkt w centroidzie obrazu oraz punkt podany z progowania Yena. Wartość parametru  $c$  ustawiona na domyślną,  $c: 2$ .
2. Sprawdzenie algorytmu ze względu na wartość mnożnika  $c$  określającego, jak bardzo punkty wokół punktu startowego mają się od niego różnić. Punkty startowe zostały pobrane z progowania Yena.

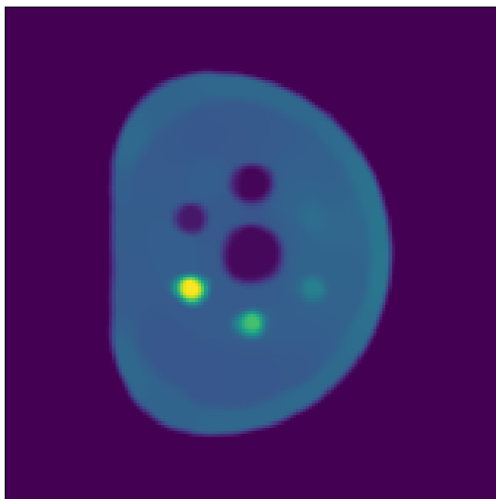
---

<sup>§</sup>Wartość maksymalna wynosi 255.

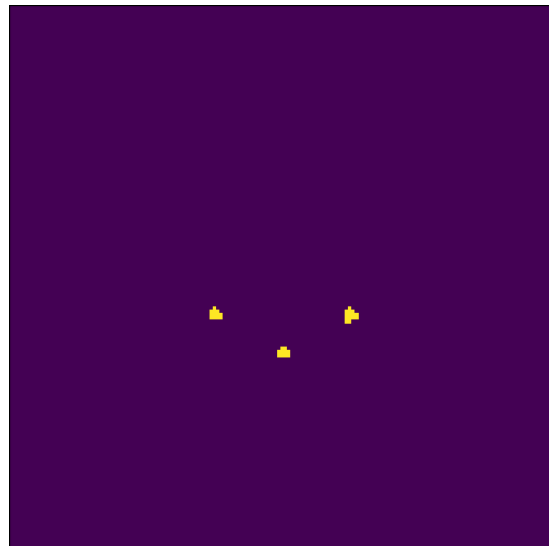


**Test algorytmu ze zmiennym punktem startowym (punktami startowymi).** W przypadku wybrania za punkt startowy punktu (0,0) oraz punktu w centroidzie fantoma, algorytm nie zwrócił oczekiwanych obszarów zainteresowania. Istotą algorytmu Region Growing jest przyłączanie do obszaru, w którym początkowo znajduje się punkt/ -y startowy/ -e, przyległych do tego obszaru punktów, których intensywność nie odbiega szczególnie od średniej intensywności w regionie. Gdy za punkt startowy weźmie się woksel z tła, w którym wartości są bliskie 0, algorytm będzie wprowadzał do obszaru ogromną liczbę punktów z tła. Żeby zapobiec takiemu efektowi, w programie ustawiony został warunek stopu: gdy liczba punktów w obszarze przekroczy 0.02% wszystkich wartości, przerywane jest dodawanie punktów do obszaru, a użytkownik otrzymuje powiadomienie o błędzie.

Dopiero wybranie punktu z obszarów o wyższych intensywnościach pozwoliło na utworzenie obszarów o ograniczonym rozmiarze. W tym celu wykorzystano progowanie Yena, które podzieliło obraz na tło oraz pierwszy plan, czyli jasne punkty. Następnie, przy pomocy metody *regionprops* oraz *centroid* pakietu Scikit Image, otrzymano współrzędne centroid jasnych obszarów fantoma. Uzyskane współrzędne wprowadzono do algorytmu Region Growing jako parametr `startPoints`.



(a) Obraz przed segmentacją.



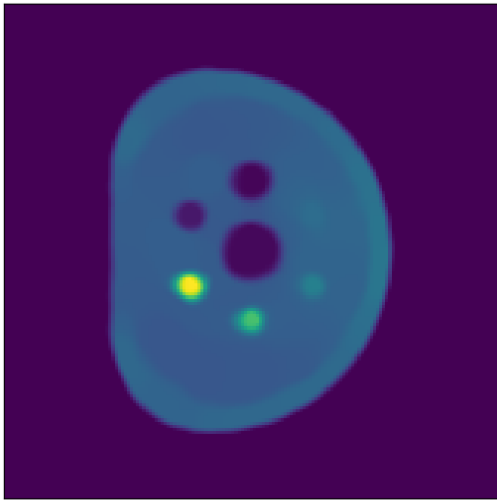
(b) Obraz po segmentacji algorytmem Region Growing wspomaganym progowaniem Yena.

Rysunek 17: Segmentacja obrazu algorytmem Region Growing. Punkty startowe zostały uzyskane poprzez znalezienie środkowych punktów obszarów wysegmentowanych progowaniem Yena.

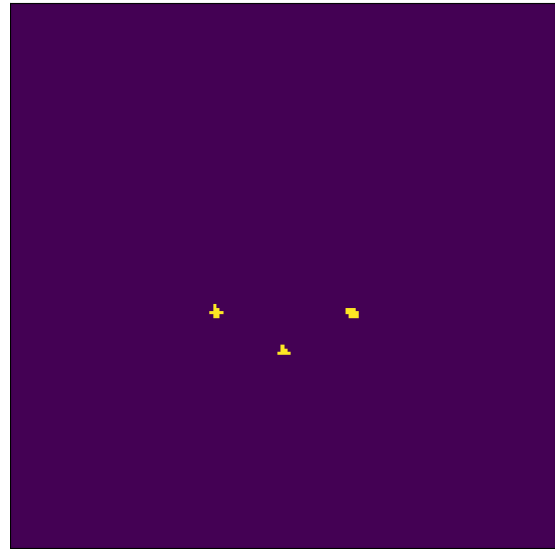
Wskazanie jako punktów początkowych centroid obszarów pochodzących z segmentacji progowaniem Yena sprawiło, że algorytm zadziałał poprawnie i na obrazie znalezione zostały obszary zainteresowania. Wykryto 3 z 4 jasnych obszarów, gdyż tyle punktów początkowych zostało przekazanych do algorytmu.

**Test algorytmu ze zmiennym mnożnikiem c.** W kolejnym teście algorytm będzie również przyjmował punkty startowe z algorytmu Yena, natomiast zmieniany będzie mnożnik  $c$  odpowiadający za liczbę punktów wprowadzanych do obszaru. Sprawdzono zostaną wartości parametru ( $c$  musi być większe od 0): 1-4.

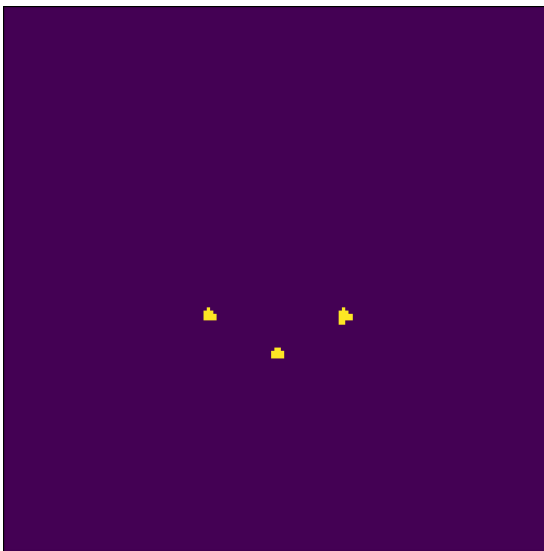
Dla wartości  $c$  powyżej 3, algorytm wprowadzał do obszaru woksele z tła, co spowodowało, że obszary rosły do bardzo dużych rozmiarów, które nie były prawdziwe. Nie mniej jednak, zwiększenie mnożnika powodowało zwiększenie liczby wokseli dołączanych do obszaru. Nie udało się jednak znaleźć optymalnej wartości, dla której algorytm zsegmentował wszystkie woksele jasnych kulek fantoma. W każdym razie algorytm zadziałał poprawnie i tworzył obszary zainteresowania na podstawie właściwości statystycznych obrazu zgodnie z założeniami.



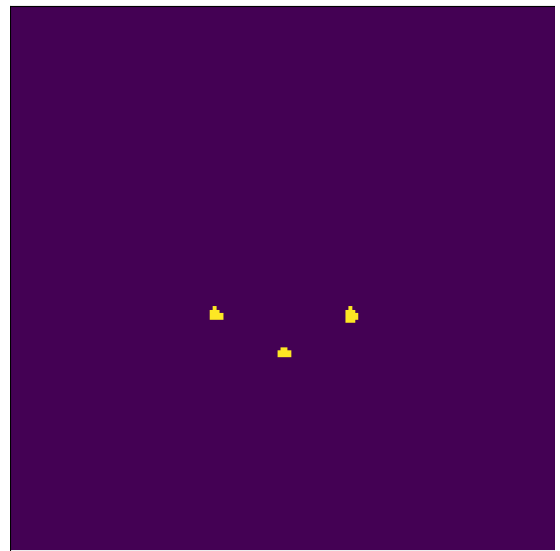
(a) Obraz przed segmentacją.



(b) Algorytm Region Growing, parametr  $c := 1$ . Liczba wokseli w obszarach od lewej: 13, 13, 32.



(c) Algorytm Region Growing, parametr  $c := 2$ . Liczba wokseli w obszarach od lewej: 20, 24, 56.

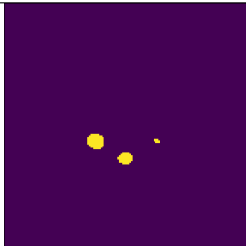
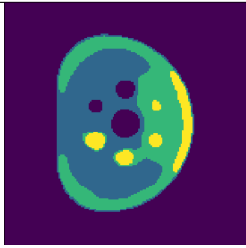
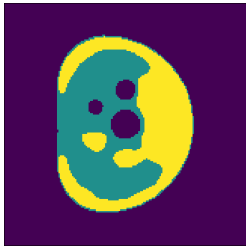
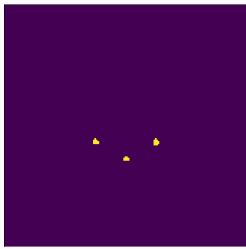


(d) Algorytm Region Growing, parametr  $c := 3$ . Liczba wokseli w obszarach od lewej: 20, 24, 60.

Rysunek 18: Segmentacja obrazu algorytmem Region Growing ze zmiennym mnożnikiem  $c$ . Punkty startowe zostały uzyskane poprzez znalezienie środków punktów obszarów wysegmentowanych progowaniem Yena. Dla wartości parametru  $c$  powyżej 3, niemożliwe było uzyskanie obszarów o skończonym rozmiarze.

### 4.3 Zestawienie i porównanie algorytmów segmentacji obrazów

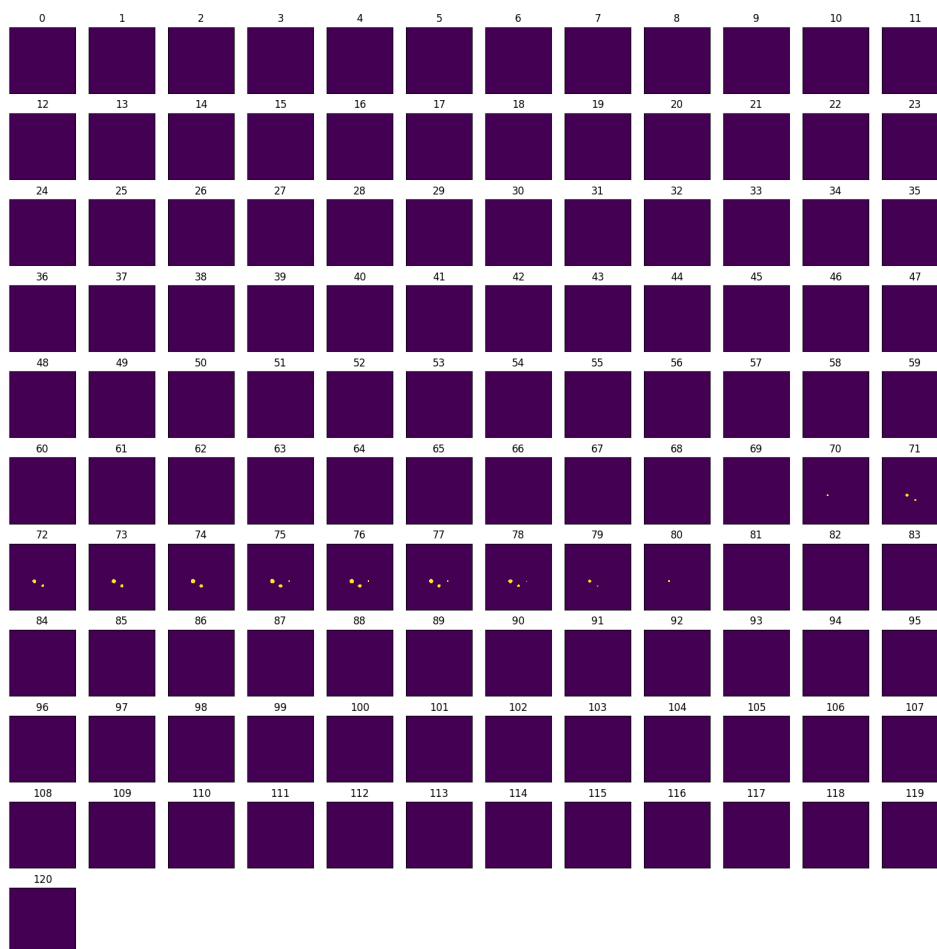
W celu porównania segmentacji obrazów fantoma NEMA, wyniki każdego z nich umieszczone zostały w poniższej tabeli.

Algorytm	Grupa algorytmów segmentacji	Opis działania i skuteczności algorytmu	Przykład działania
Progowanie Yena	Algorytmy progujące	Prosta w implementacji metoda dająca satysfakcjonujące wyniki. Nie wykrywa obszarów ROI o intensywnościach zbliżonych do tła.	
Progowanie Otsu iteratywne	Algorytmy progujące	Również prosta implementacja polegająca na kilkukrotnym wykorzystaniu progowania Otsu. Dla liczby iteracji: 3 algorytm zwrócił zadowalające wyniki - obraz został podzielony na 4 obszary.	
Progowanie Otsu wielokrotne	Algorytmy progujące	Algorytm wielokrotny zaproponowany w [16] zaimplementowany własnoręcznie. Progowanie dwukrotne zwraca wynik podobny do progowania iteracyjnego z dwoma progami. Taka konfiguracja nie sprawdziła się jednak – jasne obszary nie zostały wysegmentowane poprawnie.	
Region Growing	Algorytmy oparte o regiony	W wersji podstawowej, tzn. gdy użytkownik wskazywał punkty startowe algorytmu, metoda nie zwracała wyniku, ponieważ dołączana była duża liczba punktów z tła obrazu. W momencie zastosowania modyfikacji w postaci pobierania punktów startowych z algorytmu progującego Yena, algorytm Region Growing zaczął segmentować poprawnie, jednak tworzone obszary były znacznie mniejsze niż w przypadku samego Progowania Yena.	

Tablica 3: Zestawienie oraz porównanie metod wykrywania obszarów zainteresowania na obrazie skanowania fantoma.

---

Segmentacja algorytmem yen-thresh



Rysunek 19: Pełny wolumen po segmentacji progowaniem Yena.

---

## 5 Podsumowanie

W ostatnim rozdziale przedstawione zostanie podsumowanie prac nad implementacją tematu pracy inżynierskiej. Następnie omówione będą perspektywy rozwoju stworzonego oprogramowania oraz zastosowanie w projekcie J-PET.

### 5.1 Podsumowanie pracy nad zadaniem

Na wstępie należy przypomnieć, że tematem niniejszej pracy miało być przygotowanie, zaimplementowanie oraz przetestowanie algorytmów przetwarzania obrazów w celu wykrywania regionów zainteresowania ROI na trójwymiarowych obrazach rekonstrukcji skanowania emisyjnym tomografem pozytonowym PET. Wynikiem prac miało być oprogramowanie realizujące algorytmy segmentacji obrazów trójwymiarowych. Programy oraz technologie w nim użyte musiały być otwartoźródłowe, natomiast kod źródłowy przenośny i łatwy w ponownym wykorzystaniu.

Zdecydowano, że oprogramowanie napisane będzie w języku Python ze względu otwartoźródłowość tego języka oraz mnogość pakietów stworzonych w tym języku. Na cele zadania stworzono 3 moduły:

- VolumeData.py – Wczytywanie, zapisywanie, wyświetlanie obrazów 3D,
- SegmentVolume.py – Implementacje algorytmów wykrywania ROI,
- SegmentujObrazyJPET.py – Program głównym implementujący komunikację z użytkownikiem w konsoli.

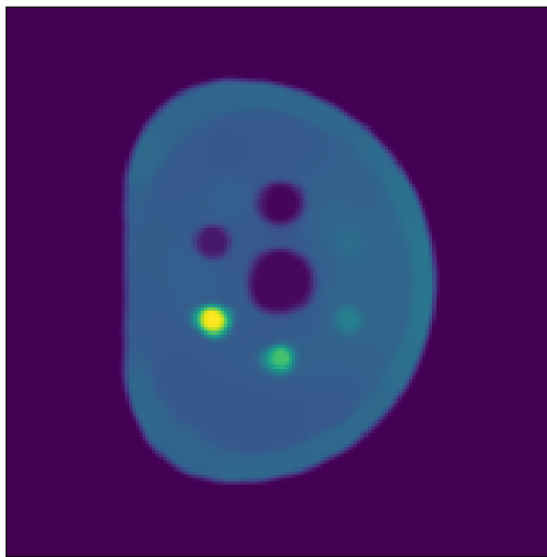
Oprogramowanie napisane zostało w sposób proceduralno – obiektowy. Poszczególne moduły są od siebie niezależne, natomiast utworzone klasy – spójne i luźno powiązane z innymi.

W klasie SegmentVolume zaimplementowano następujące algorytmy:

- Progowanie Yena,
- Progowanie Otsu iteratywne oraz wielokrotne,
- Region Growing – w wersji podstawowej oraz wykorzystującej do działania progowanie Yena.

Progowanie Yena oraz Otsu to algorytmy z grupy algorytmów progujących, natomiast Region Growing należy do grupy algorytmów operujących na regionach. W celu przetestowania skuteczności algorytmów, wykorzystano skan tomografem tzw. fantoma, czyli plastikowego pudełka, gabarytami przypominającego ludzkie płuca, w którym umieszczono 4 elementy emitujące promieniowanie. Owy fantom stworzony został na potrzeby testów skanerów medycznych, takich jak tomograf komputerowy oraz tomograf PET. Skan fantoma poddawany jest rekonstrukcji, czyli odtworzeniu trójwymiarowej postaci rzeczywistego obrazu. Poniżej przedstawiono przykładowy obraz wyjściowy rekonstrukcji BPTV (Propagacja Wsteczna z maksymalną zmiennością).

Zadaniem algorytmów segmentacji miało być podzielenie obrazu tomografu na obszary oraz wydzielenie z niego regionów zainteresowania ROI istotnych z punktu widzenia zdrowia badanego pacjenta. Każdy z algorytmów w pewien sposób dzielił obraz na podobszary, jednak nie wszystkie wysegmentowały obszary ROI – jasne punkty znajdujące się w skanowanym fantomie.



Rysunek 20: Obraz po rekonstrukcji BPTV.

**Omówienie poprawności działania zaimplementowanych algorytmów.** Skuteczniejsze okazały się metody oparte o progowanie, w szczególności metoda Yena, która wykryła 3 z 4 obiektów emitujących promieniowanie, natomiast wydzielone obszary wielkością przypominały świecące punkty na oryginalnym obrazie.

Zadowolające wyniki zwrócił również algorytm iteratywny Otsu, który wielokrotnie wykonuje progowanie standardową metodą Otsu biorąc za obraz wejściowy fragment wysegmentowany poprzednim progowaniem. Najlepiej sprawdzało się 3-krotne progowanie Otsu. Powodowało to podzielenie obrazu na 4 regiony. W jednym z nich znalazły się 4 punkty emitujące promieniowania. Nie sprawdziło się jednak progowanie Otsu wielokrotne, które zwracało segmentację podobną do tej wykonywanej przez progowanie Otsu iteratywne o 2 iteracjach. W wysegmentowanym obszarze znalazło się oprócz jasnych punktów wiele wokseli należących do tła, czego skutkiem jest brak możliwości oceny fragmentu jako obszar ROI.

Natomiast algorytm Region Growing do poprawnej pracy wymaga podania punktów startowych, które będą stosunkowo jasne. Jako punkty startowe użyto środki obszarów wysegmentowanych przez progowanie Yena – w takim przypadku metoda znalazła 3 z 4 jasnych punktów. Jednak wysegmentowane obszary były znacznie mniejsze w porównaniu z obrazem oryginalnym.

Podsumowując najlepiej sprawdziły się proste metody oparte o progowanie – są one odporne na szumy oraz artefakty wprowadzane przez rekonstrukcję obrazu oraz na niewielki kontrast. Nie wymagają również podawania parametrów. Algorytm Region Growing do oczekiwanego działania potrzebował dość dokładnego sprecyzowania punktów startowych. W przeciwnym przypadku algorytm nie był zbieżny.

**Omówienie poprawności implementacji w kontekście wymagań dotyczących oprogramowania.** Na oprogramowanie implementujące algorytmy segmentacji oprócz wymagań, co do zwracanych wyników, były nałożone również wymagania określające wykorzystane narzędzia. Wybrane technologie miały być otwartoźródłowe, natomiast utworzone programy

---

przenośne oraz łatwe w ponownym wykorzystaniu. Oprogramowanie miało być wykorzystywane w dwojaki sposób: jako plik wykonywujący segmentację bezpośrednio oraz jako interfejs programistyczny API w zewnętrznych programach. Wszystkie te założenia zostały spełnione. Oprogramowanie zostało napisane w języku Python, wykorzystując biblioteki Scikit Image oraz bibliotekę standardową Pythona. Wszystkie te technologie udostępnione są bezpłatnie i można je wykorzystywać we własnych celach, szczególnie naukowo - badawczych.

Utworzone oprogramowanie składa się z 3 skryptów języka Python. Dwa z nich przechowują implementacje klas – pierwsza z nich pozwala na operowanie na obrazach trójwymiarowych, druga zawiera implementacje algorytmów segmentacji. Razem tworzą interfejs programistyczny API możliwy do wykorzystania w innych projektach. Natomiast trzeci ze skryptów zawiera implementację komunikacji z użytkownikiem i pozwala na wywoływanie algorytmów segmentacji z terminala.

## 5.2 Perspektywy rozwoju oprogramowania

Utworzone oprogramowanie ma szansę zostać wykorzystane przy projekcie J-PET, którego celem jest stworzenie nowoczesnego tomografu pozytonowego. Algorytmy zaimplementowane w ramach niniejszej pracy mają służyć do wykrywania obszarów zainteresowania na obrazach pochodzących z tomografu J-PET.

W ramach utworzonego oprogramowania zaimplementowano algorytmy bazujące na prostych, jednak skutecznych metodach, takich jak progowanie. Z uwagi na fakt, że tomograf J-PET wciąż jest tworzony, nie otrzymano nadal zbyt wielu obrazów skanowania tym tomografem. Posiadając dużą ilość wyników skanowania, można by do wykrywania obszarów ROI wykorzystać algorytmy bazujące na uczeniu maszynowym, które do poprawnego działania i nauczenia algorytmu wymaga właśnie dużą ilość danych. Takie algorytmy mogłoby w przyszłości poprawić wyniki konwecjonalnych metod takich jak progowanie czy Region Growing.

---

## Literatura

- [1] Witryna projektu JPET  
[koza.if.uj.edu.pl/pet/](http://koza.if.uj.edu.pl/pet/)
- [2] A. Wiczorek et al., *Novel scintillating material 2-(4-styrylphenyl) benzoxazole for the fully digital and MRI compatible J-PET tomograph based on plastic scintillators* PLoS ONE **12(11)**: e0186728 (2017)  
[http://koza.if.uj.edu.pl/files/1ae7be6e8cba28d3a31e19c4f77fb991/AWiczorek\\_POLS2017.p](http://koza.if.uj.edu.pl/files/1ae7be6e8cba28d3a31e19c4f77fb991/AWiczorek_POLS2017.p)
- [3] L. Raczyński et al., *INTRODUCTION OF TOTAL VARIATION REGULARIZATION INTO FILTERED BACKPROJECTION ALGORITHM* Acta Phys. Pol. B **48**, **1611** (2017),  
<http://koza.if.uj.edu.pl/files/0210386cb2fb2ffcd49fee5a1133eb2b/v48p1611.pdf>
- [4] Repozytorium JPETTomography na portalu GitHub  
<https://github.com/JPETTomography>
- [5] Artykuł o tomografii PET na Wikipedii  
[https://en.wikipedia.org/wiki/Positron\\_emission\\_tomography/](https://en.wikipedia.org/wiki/Positron_emission_tomography/)
- [6] Zasada działania oraz wprowadzenie do tomografii PET  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1126321/>
- [7] Przegląd metod segmentacji obrazów PET  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4060809>
- [8] Bailey, D.L; D.W. Townsend; P.E. Valk; M.N. Maisey *Positron-Emission Tomography: Basic Sciences*. **382** (2005)
- [9] Omówienie problemu i metod rekonstrukcji skanów PET/CT  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3039307/>
- [10] Prezentacja z zakresu metod rekonstrukcji implementowanych w J-PET  
<http://koza.if.uj.edu.pl/jpet-bnct-2018/file/talks/Raczyński.pdf>
- [11] Opis algorytmu FBP  
<https://www.dspguide.com/ch25/5.htm>
- [12] Artykuł na Wikipedii na temat segmentacji obrazu  
[https://en.wikipedia.org/wiki/Image\\_segmentation](https://en.wikipedia.org/wiki/Image_segmentation)
- [13] Sarang Lakare *3D Segmentation Techniques for Medical Volumes* State University of New York at Stony Brook (2000)  
<https://pdfs.semanticscholar.org/36d6/b485b1bbf6f0a7ab187d596e944bcf4f9e54.pdf>
- [14] Oliver Wirjadi Survey of 3D Image Segmentation Methods **123** (2007)
- [15] Dr inż. Edward Pórolniczak *Metoda aktywnych konturów w segmentacji znaków na tablicach rejestracyjnych* **58** (2002)  
<http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BSW4-0117-0005>



- 
- [16] Deng-Yuan Huang, Ta-Wei Lin, Wu-Chih Hu *Automatic multilevel thresholding based on two-stage Otsu's method with cluster determination by valley estimation* (2011)  
<http://www.ijicic.org/ijicic-10-05033.pdf>
- [17] Jui-Cheng Yen, Fu-Juay Chang and Shyang Chang *A New Criterion for Automatic Multi-level Thresholding* (1995)  
<https://ieeexplore.ieee.org/document/366472>
- [18] Witryna projektu Open CV  
<https://opencv.org/about/>
- [19] Opis standardu JSON.  
<https://www.json.org/>
- [20] P.Kowalski ,P.Moskal ,W.Wiślicki, L.Raczyński et al. *Multiple Scattering and Accidental Coincidences in the J-PET Detector Simulated Using GATE Package* *APhysPolA* **127** (2015)  
<http://koza.if.uj.edu.pl/files/5747987aab1c03097787a71fe4f88e5b/a127z5p16.pdf>
- [21] Optymalizowanie parametrów tomografów wg. standardu NEMA  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4542864/>
- [22] Opis produktu Fantom NEMA  
[http://www.spect.com/pub/NEMA\\_IEC\\_Body\\_Phantom\\_Set.pdf](http://www.spect.com/pub/NEMA_IEC_Body_Phantom_Set.pdf)

---

## Wykaz symboli i skrótów

**PET** – tomograf pozytonowy emisyjny

**J-PET** – tomograf PET tworzony w ramach projektu o tej samej nazwie

**API** – programistyczny interfejs użytkownika

**ROI** – region zainteresowania

**BPTV** – *Wsteczna propagacja z maksymalną zmiennością* – metoda rekonstrukcji obrazu

**FBP** – *Filtrowana propagacja wsteczna* – metoda rekonstrukcji obrazu

---

## Spis rysunków

1	Prototyp tomografu J-PET w pełnej skali. Skaner J-PET zbudowany jest z trzech warstw polimerowych scyntylatorów (czarne) oraz 2 rzędów fotopowielaczy (jasne), po jednej z każdej strony scyntylatorów. . . . .	7
2	Opis fizyczny emisji cząstek w tomografie PET . . . . .	9
3	Algorytm FBP . . . . .	10
4	Metody BPTV oraz FBP rekonstrukcji obrazu. Algorytm BPTV działa znacznie lepiej – krawędzie elementów są ostre, a kolory łatwo rozróżnialne. . . . .	11
5	Rodzaje metod segmentacji obrazów trójwymiarowych . . . . .	12
6	Przykład histogramu. Na rysunku są widoczne dwa maksima – dla pierwszego planu (wartość ok. 0) oraz dla tła (wartość ok. 80) . . . . .	15
7	20 najpopularniejszych języków programowania w październiku 2018. Popularność języka Pythona rośnie z roku na rok. W październiku 2018 programowało w nim ok. 24% programistów na świecie. . . . .	18
8	Diagram klas utworzonych na cele zadania. Klasa SegmentVolume wykorzystuje obiekty klasy VolumeData. . . . .	21
9	Fantom NEMA – plastikowy model płuc służący do badań nad tomografią PET. . . . .	24
10	Instalacja Fantoma NEMA w celu badania tomografu. . . . .	24
11	Metody BPTV oraz FBP rekonstrukcji obrazu. Algorytm BPTV działa znacznie lepiej – krawędzie elementów są ostre, a kolory łatwo rozróżnialne. . . . .	25
12	Rekonstrukcja symulacji skanowania tomografem J-PET metodą BPTV. . . . .	26
13	Rekonstrukcja symulacji skanowania tomografem J-PET metodą FBP. . . . .	27
14	Segmentacja obrazu algorytmem progującym Yena. . . . .	29
15	Segmentacja obrazu algorytmem progującym Otsu o różnej liczbie iteracji. . . . .	30
16	Segmentacja obrazu algorytmem Otsu wielokrotnym. . . . .	31
17	Segmentacja obrazu algorytmem Region Growing. Punkty startowe zostały uzyskane poprzez znalezienie środkowych punktów obszarów wysegmentowanych progowaniem Yena. . . . .	32
18	Segmentacja obrazu algorytmem Region Growing ze zmiennym mnożnikiem c. Punkty startowe zostały uzyskane poprzez znalezienie środkowych punktów obszarów wysegmentowanych progowaniem Yena. Dla wartości parametru c powyżej 3, niemożliwe było uzyskanie obszarów o skończonym rozmiarze. . . . .	33
19	Pełny wolumen po segmentacji progowaniem Yena. . . . .	35
20	Obraz po rekonstrukcji BPTV. . . . .	37

---

## Spis tablic

1	Pakiety biblioteki standardowej użyte w projekcie . . . . .	19
2	Opis wywołania algorytmów oraz ich parametry. . . . .	23
3	Zestawienie oraz porównanie metod wykrywania obszarów zainteresowania na ob- razie skanowania fantoma. . . . .	34

---

Spis załączników

**Witryna projektu J-PET**

[koza.if.uj.edu.pl/pet/](http://koza.if.uj.edu.pl/pet/)

**Repozytorium projektu J-PET w serwisie GitHub**

<https://github.com/JPETTomography>

**Repozytorium z oprogramowaniem utworzonym w ramach pracy inżynierskiej**

<https://github.com/JPETTomography/j-pet-roi-detector>