

# Sprawozdanie z laboratorium

## Informatyka w medycynie

### Symulator Tomografa Komputerowego

Prowadzący: mgr inż. Tomasz Sternal

Autorzy:      Piotr Górny              nr indeksu: 127248  
                 Mateusz Bamberski      nr indeksu: 109893

#### 1. Podział pracy:

Piotr Górny – rozmycie obrazu wynikowego, normalizacja, tworzenie obrazu interaktywnego

Mateusz Bamberski – splot, błąd średnio-kwadratowy w funkcji iteracji

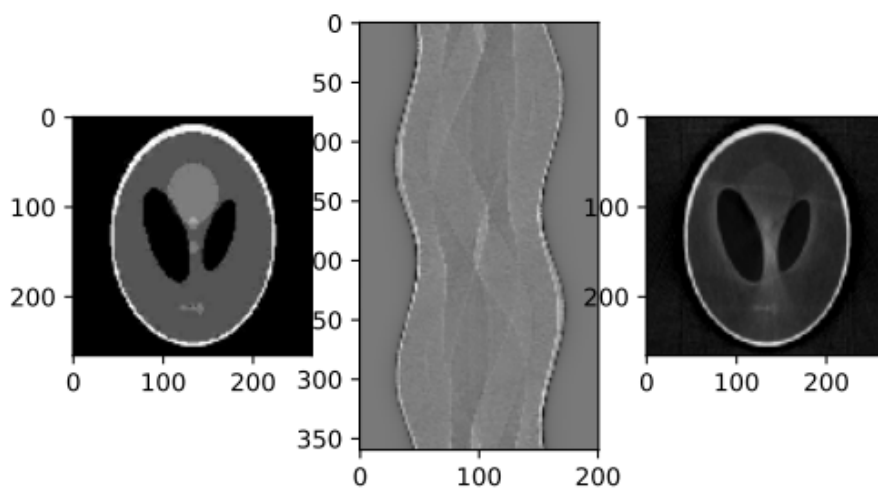
Wspólnie: algorytm Besenhama, funkcje konwertujące położenie emitera na układ współrzędnych kartezjańskich, błąd średnio-kwadratowy, wykresy

#### 2. Opis projektu

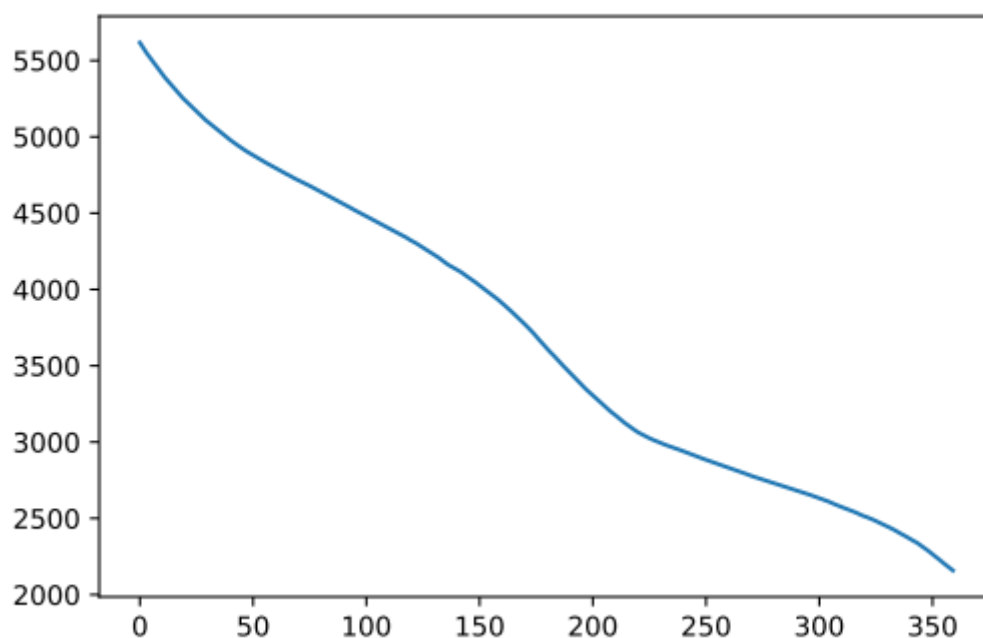
Aplikacja zaimplementowana w języku Python z użyciem Jupyter Notebook. Wczytany obrazek jest bitmapą, gdzie każdy piksel może mieć dowolny kolor szarości (0-255). Dokonano transformacji Radona, a aplikacja wyświetla obrazek wejściowy, powstały sinogram oraz obrazek wyjściowy po odtworzeniu (transformacja odwrotna). Wykorzystano model emitera stożkowego. Do liniowego przejścia po kolejnych pikselach wykorzystano algorytm Besenhama. Zasymlowano pochłanianie promieniowania w sposób addytywny.

#### 2. Wyniki pracy symulatora

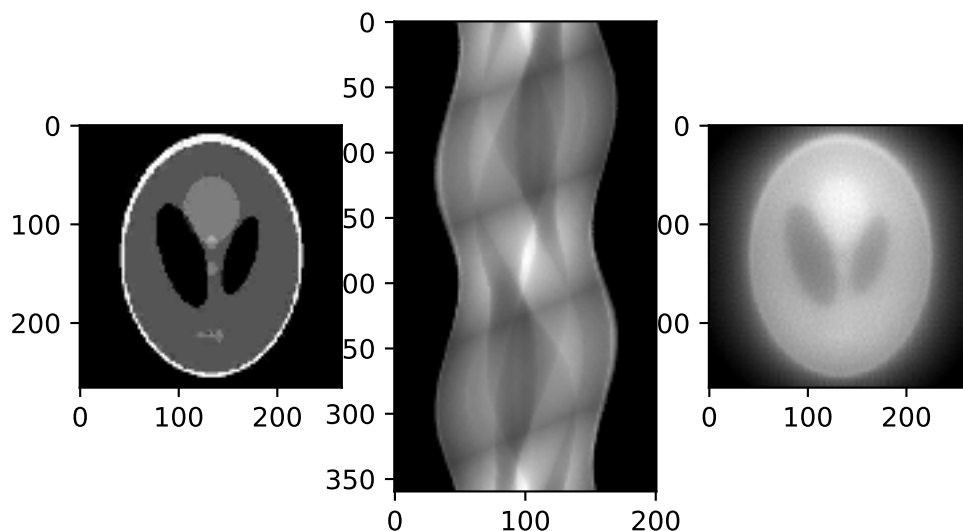
Zdjęcie wejściowe, sinogram i zdjęcie wyjściowe (parametry: liczba detektorów: 201, kąt z jakim emiter wysyła promieniowanie: 80, zmiana położenia emitera: co 1 stopień, dodatkowa filtracja (splot)):



Wykres błędu średnio-kwadratowego w funkcji iteracji dla tych samych parametrów:

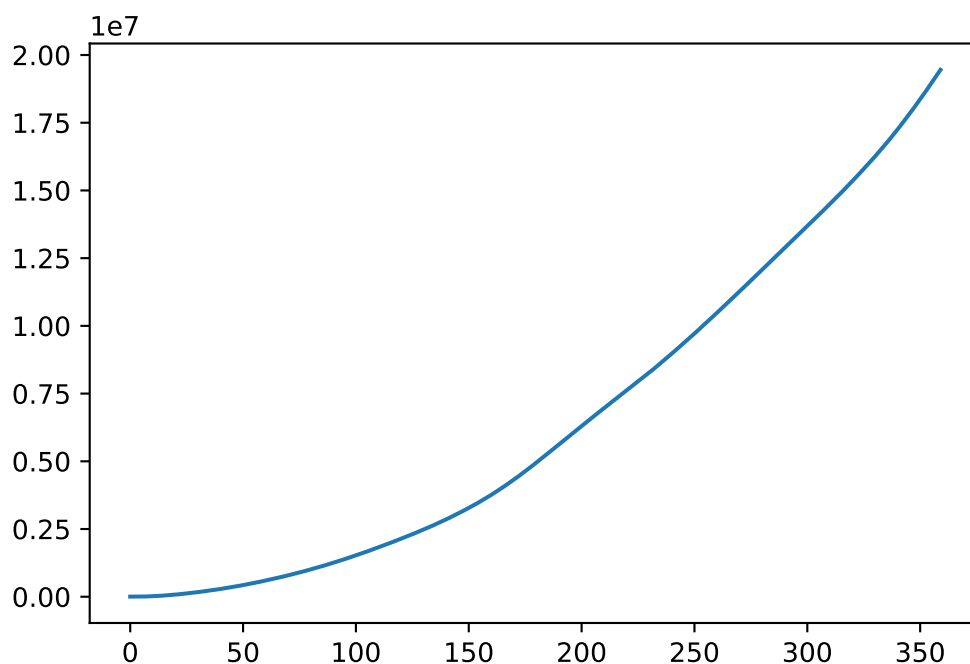


Na podstawie wykresu stwierdzono, że wraz ze zwiększającą się liczbą iteracji błąd średnio-kwadratowy zmniejsza się, co było celem badania. Sinogram w programie wraz z włączoną opcją filtrowania jest szary, co jest efektem ubocznym działania filtru. Poniżej przedstawiono wyniki działania symulatora dla tych samych parametrów, lecz bez włączonej filtracji.



Sinogram jest zdecydowanie bardziej czytelny, natomiast obraz wyjściowy jest dużo jaśniejszy. Więcej szczegółów w obrazie wyjściowym zobaczymy przy zastosowaniu filtru.

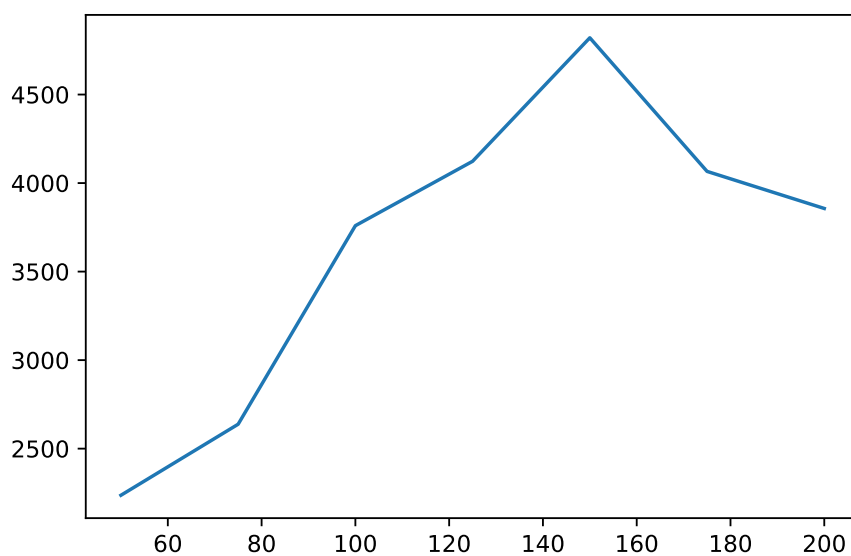
Wykres błędu średnio-kwadratowego w zależności od liczby iteracji dla tej symulacji:



Zauważamy, że błąd rośnie wraz z iteracjami co jest wynikiem odwrotnym niż oczekiwany. Jest to spowodowane dużą jasnością obrazu wyjściowego (dużą część błędu stanowi porównanie pixeli czarnych z obrazka wejściowego wraz z tymi co są jasne w otoczeniu elementu który jest poddany badaniu).

Dodatkowo sprawdzono, czy liczba emiterów ma znaczenie w działaniu symulatora. Poniżej przedstawiono wykres zależności błędu średnio-kwadratowego od liczby detektorów dla pozostałych parametrów stałych: liczba detektorów: 51, 76, 101, 126, 151, 201, kąt z jakim emiter wysła promieniowanie: 70, zmiana położenia emitera: co 1 stopień, bez i z dodatkową filtracją.

Bez filtracji:



Z filtracją:

