

# Laboratorium Informatyki w Medycynie

## 3 punkt kontrolny

Szymon GRAMZA 109785

Przemysław HOFFMANN 109786

23.05.2015r.

Prowadzący: mgr inż. Tomasz Pawlak

Temat zadania: symulator tomografu komputerowego

# 1 Opis problemu

Tematem projektu zaliczeniowego jest symulator działania tomografu komputerowego. Zgodnie z wymaganiami symulator ten powinien pozwala na:

- akwizycję rzutów obrazów 1D z zadanego obrazu 2D
- prezentację użytkownikowi tych rzutów
- rekonstrukcję obrazów 2D z rzutów 1D przy użyciu odwrotnej transformaty Radona
- prezentację zrekonstruowanego obrazu

Dodatkowo aplikacja pozwala w prosty sposób sterować głównymi parametrami symulatora.

## 1.1 Zasada działania

Tomografia komputerowa polega na uzyskaniu przekrojów badanego obiektu, poprzez naświetlenie obiektu wiązką promieniowania.

Urządzenie do tomografii komputerowej to tomograf, a uzyskany jego metodą obraz to tomogram.

Jako narzędzie znalazła zastosowanie w wielu dziedzinach życia, jednak najczęściej kojarzonym jest dziedzina medycyny, gdzie tomografia komputerowa służy do obrazowania ludzkiego ciała. Pozwala ona na wykrywanie wielu chorób już we wczesnych stadiach.

Do najpopularniejszych zastosowań diagnostycznych należą: diagnostyka mózgowia, udaru mózgu, badanie jamy brzusznej, miednicy, kanału kręgosłupa, oczodołów, szyi, zatok obocznych nosa, kości i stawów, klatki piersiowej i śródpiersia, a także wielu innych nie tylko standardowych, ale także bardzo specjalistycznych okolic anatomicznych. Tomografia jest także stosowana w przemyśle do badania jakości produktów, poszukiwania wad materiałów, kontroli spoin.

Zasada działania tomografu komputerowego opiera się na próbie odtworzenia wewnętrznej struktury obiektu. Dokonuje się tego poprzez wielokrotne naświetlanie, równoległe do płaszczyzny obrazowanego przekroju.

Informacje o tkance w trakcie badania pojedynczego przekroju uzyskujemy na podstawie pomiaru osłabienia wiązki.

Osłabienie promieniowania określa wzór:

$$I(S) = I(0)e^{-\int_0^S \mu dx}$$

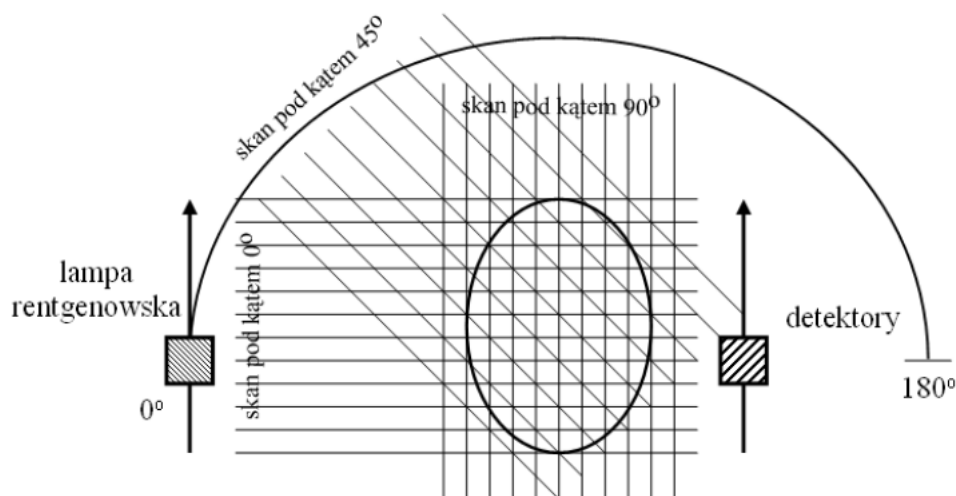
Mała dokładność i niedoskonałość pierwszych tomografów była motywacją do kontynuowania prac nad kolejnymi wersjami. W związku z rozwojem technologii oraz nowymi pomysłami na przestrzeni lat, tomografy sklasyfikowano na tak zwane generacje.

### 1.1.1 Generacje tomografów

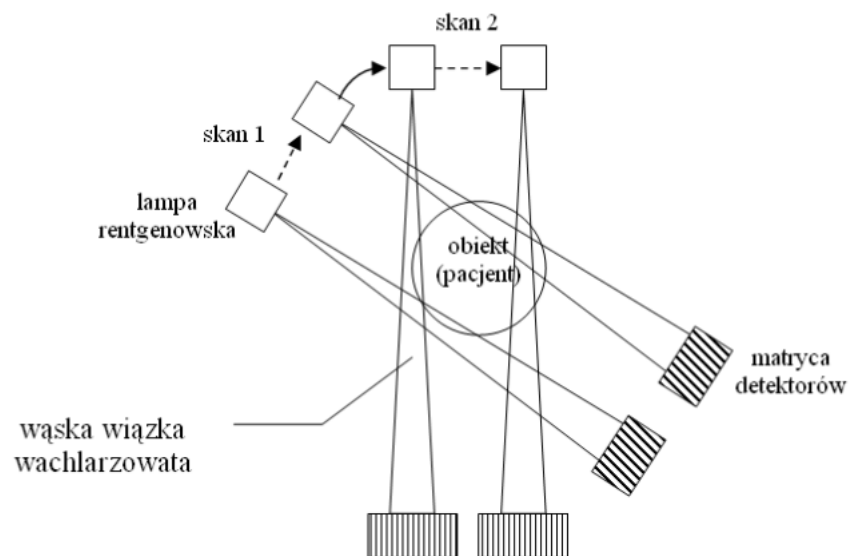
Od czasów wynalezienia tomografu zmieniały się koncepcje i usprawnienia mające na celu zwiększenie ich szybkości i wydajności.

Poniższe zestawienie obrazuje jak zmieniało się podejście do tego zagadnienia:

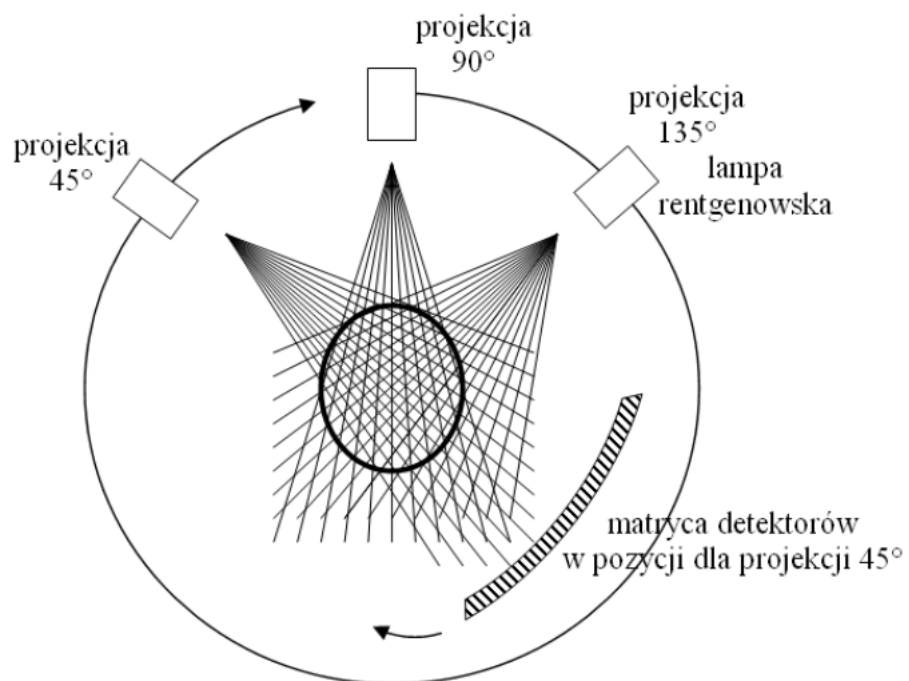
- Generacja I - skaner składał się z pojedynczego emitera i detektora. Lampa i detektor wykonywały ruchy translacyjne i rotacyjne.



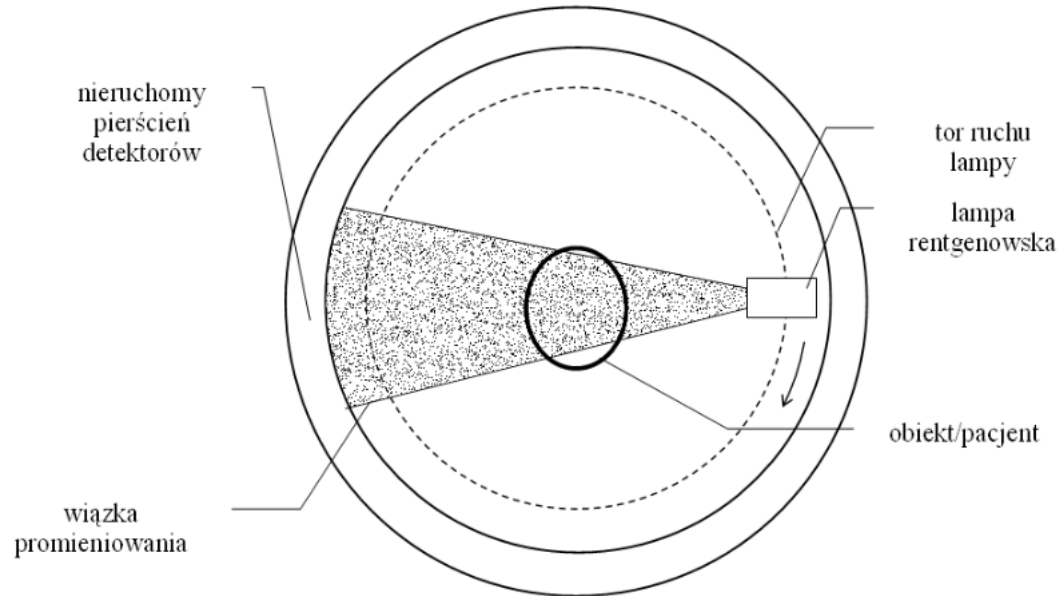
- Generacja II - zwiększono liczbę detektorów co zmniejszyło liczbę ruchów translacyjnych lampy.



- Generacja III - wyeliminowano ruch translacyjny poprzez rozmieszczenie detektorów na łuku pierścienia obracającego się razem z lampą dookoła pacjenta.



- Generacja IV - detektory umieszczone zostały na stałe na pierścieniu, ruch obrotowy wykonuje tylko lampa.



## 2 Opis rozwiązania

Poniżej znajduje się opis użytej technologii oraz lista zastosowanych algorytmów z wyróżnieniem na poszczególne etapy.

### 2.0.2 Technologia

Program zaimplementowany został w języku Python (wersja 2.7) z wykorzystaniem następujących bibliotek:

- tkinter - tworzenie, obsługa oraz zwalnianie interfejsu graficznego. Pozwala na intuicyjne poruszanie się po aplikacji, wczytywanie oraz zapisywanie przetwarzanych obrazów w szczególności wczytanie obrazu do generacji sinogramu, zapis sinogramu, wczytanie sinogramu, zapis zrekonstruowanego obrazu. Dodatkowo biblioteki użyto do wygodnego sterowania najważniejszymi parametrami działania symulatora.
- numpy - działania na macierzach, konwersje

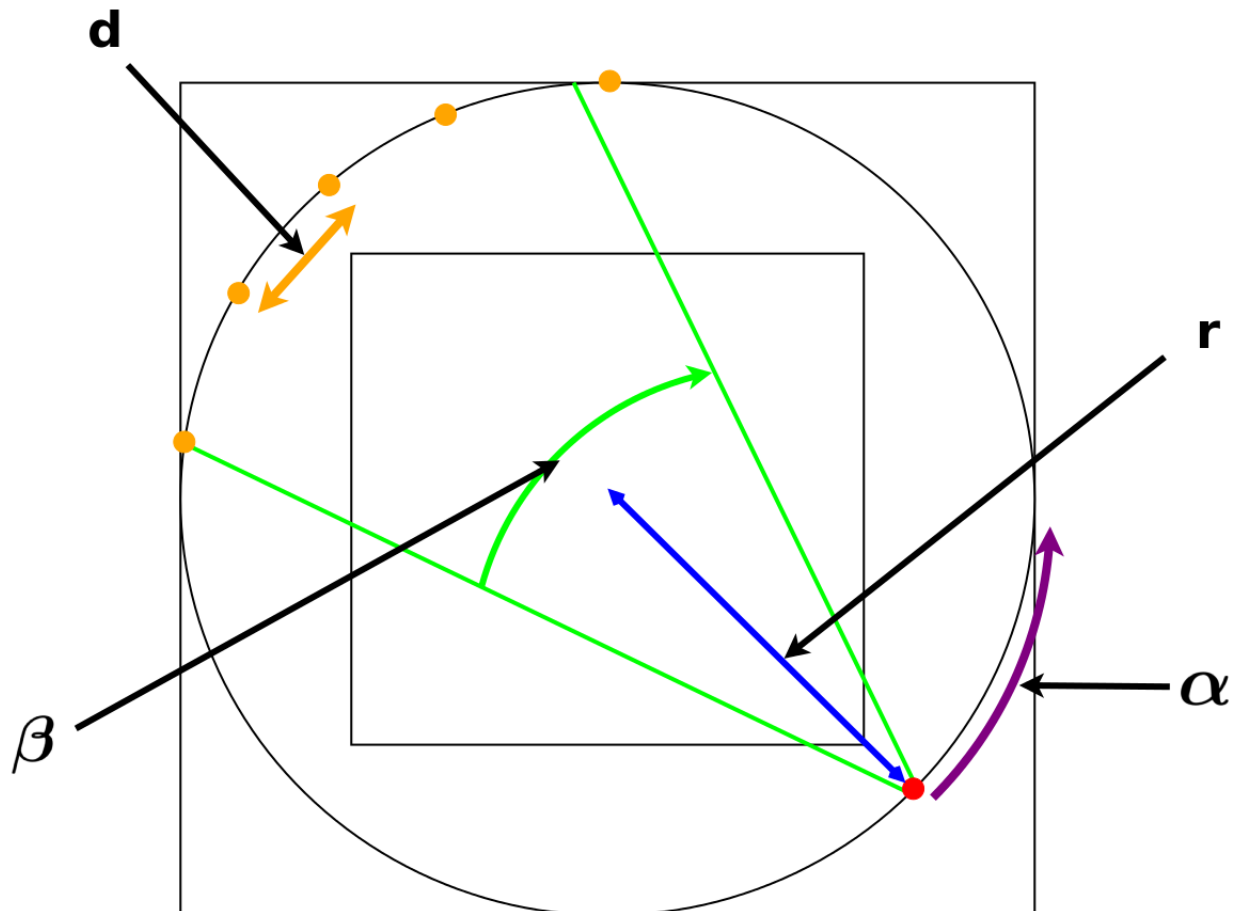
- math - obliczenia trygonometryczne, wartość bezwzględna

Przyjęto model tomografu o następujących parametrach:

- $\alpha$  - kąt o który obracany jest układ emiterów
- $\beta$  - szerokość wiązki
- $r$  - odległość układu emiterów od macierzy obrazu
- $d$  - szerokość detektora - określa liczbę pikseli reprezentujących pojedynczy detektor
- $f$  - szerokość filtra

Na liczbę detektorów wpływają odległość emitera od macierzy detektorów oraz szerokość detektora. liczba detektorów =  $\frac{2\pi r}{d}$

Poniższy schemat obrazuje rozmieszczenie parametrów symulatora:



### 2.0.3 Algorytmy

W tej sekcji znajduje się opis poszczególnych etapów przetwarzania i użytych w tym celu algorytmów. W zakresie działania symulatora wyróżniamy głównie przetwarzanie wczytanego obrazu do postaci sinogramu, oraz rekonstrukcję otrzymanego sinogramu.

- W przypadku generowania sinogramu, wykonywane są następujące kroki:
  - oryginalny obraz zostaje wczytany w skali szarości, a następnie przechowany w pamięci w postaci lokalnej macierzy na której wykonywane są obliczenia
  - jeśli wymiary wczytanego obrazu się różnią (tzn. szerokość  $\neq$  wysokość), wówczas obraz zostaje dopełniony czarnymi pikselami na krótszym wymiarze w celu uzyskania kwadratu, otrzymujemy macierz reprezentującą w trakcie symulacji wczytany obraz (macierz A)
  - tak otrzymana macierz kwadratowa (A) jest następnie rozszerzana do macierzy, której rozmiar zależy ściśle od parametru  $r$  czyli odległości emitera od środka obrazu.

Jak łatwo zauważyć odległość ta nie może być mniejsza od długości przekątnej macierzy uzyskanej w pierwszym kroku.

W uzyskanej macierzy (B) wpisany zostaje okrąg, który odpowiada gantrze tomografu (podłużnemu tunelowi do którego wjeżdża pacjent)

Okrąg ten jest otrzymywany za pomocą równania parametrycznego w kartezjańskim układzie współrzędnych:

$$\begin{cases} x = x_0 + r \cos \alpha \\ y = y_0 + r \sin \alpha \end{cases}$$

gdzie:  $r$  - odległość środka obrazu od emitera

$\alpha$  - kąt obrotu układu emiter - macierz detektorów

$x_0$  oraz  $y_0$  - współrzędne środka przetwarzanego obrazu (macierz A)

- piksele otrzymanego okręgu są grupowane w detektory w zależności od parametru  $d$  (szerokość detektorów), co zgodnie ze wzorem [numer] daje odpowiednią ilość detektorów.

W szczególnym przypadku ( $d = 1$ ), każdy piksel na okręgu stanowi pojedynczy detektor.

- po wyznaczeniu detektorów (w przypadku gdy  $d > 1$  następuje wyznaczenie środka każdego z detektorów (jeden piksel))

- rozpoczyna się praca układu emiter-detektor, emiter startuje z pozycji 0 (obrót wynosi  $0^\circ$ ), z każdym krokiem układ obraca się o kąt  $\alpha$ , aż wykona pełen obrót ( $360^\circ$ )

Warto zwrócić uwagę, że wartość obrotu  $\alpha$  ma bezpośredni wpływ na wysokość uzyskanego sinogramu, która wynosi  $360/\alpha$

- znając w każdej iteracji pozycję emitery wyznaczamy detektory na wycinku okręgu, które w danej iteracji stanowią macierz detektorów.

Zakres tych detektorów obrazuje przedział:

$$< e + 180 - (\beta/2); e + 180 + (\beta/2) >$$

gdzie  $e$  to aktualna pozycja emitery

- kolejne kroki algorytmu korzystają z algorytmu Bresenham'a. Służy on do rasteryzacji krzywych płaskich, czyli do jak najlepszego ich obrazowania na siatce pikseli. W naszym zastosowaniu służy do wyznaczania pikseli obrazu, przez które przechodzi promień (pikseli leżących na odcinku od emitery do środka obrazu) oraz wyznaczenia wartości pochłoniętego promieniowania.

-i opis Bresenham (czy błąd z tej albo z tej to wybiera ten piksel albo ten) -i w każdej iteracji do algorytmu Bresenham podawana jest pozycja emitery oraz środki detektorów należące do aktualnego łuku (iterowanie po liście środków) -i wyznaczamy promień (piksele należące do promienia) -i Transformata Radona -i sumowanie wartości (posumowane/maks pikseli \*255) wartość na detektorze, wartości na wszystkich środkach detektorów to jeden wiersz do sinogramu -i powyższe powtórzone  $360/\alpha$  razy co wpływa na rozmiar sinogramu

- Rekonstrukcja obrazu korzysta z wcześniejszych metod, jednak zmienia się kierunek emisji (emisja następuje od macierzy detektorów w kierunku emitery, który staje się odbiornikiem) W procesie rekonstrukcji bardzo ważnym elementem jest filtrowanie sinogramu, dzięki czemu obraz staje się dokładniejszy i ostrzejszy. Filtr jest wektorem, który tworzony jest zgodnie z poniższym wzorem:

$$h(k) = \begin{cases} 1 & \text{dla } k = 0 \\ 0 & \text{dla } k \% 2 = 0 \\ \frac{-4/\pi^2}{k^2} & \text{dla } k \% 2 = 1 \end{cases}$$

tworzenie filtra (wzorki itp) -i filtr na każdy wiersz -i na macierz detektorów wczytywane są kolejno wiersze sinogramu w każdej iteracji -i wczytanie wiersza na macierz -i wartość ze środka detektora jest dodawana do każdego z



pikseli na drodze - $j$  na końcu wartości we wszystkich komórkach dzielone przez największą wartość w macierzy

//może pozmieniać ale się przyda

#### **2.0.4 Transformata Radona**

W roku 1905 W. Radon udowodnił następujące twierdzenie: „Obraz obiektu dwuwymiarowego można zrekonstruować na podstawie nieskończonej ilości rzutów jednowymiarowych”. Rzutowanie to odpowiada wykonywaniu na obiekcie pewnej transformacji, nazywanej Transformacją Radona. Po dokonaniu transformacji z otrzymanych wyników otrzymuje się sinogram, czyli wykres będący wizualizacją owych wyników. Dokonanie na sinogramie Odwrotnej Transformacji Radona umożliwia zrekonstruowanie obrazu obiektu. Zrekonstruowany obraz zawierał będzie pewne zniekształcenia, lecz wraz ze wzrostem liczby projekcji powinien co raz bardziej odzwierciedlać obraz sprzed transformacji.

### **3 Testy**

Przeprowadzenie testów i jakaś funkcja oceny

### **4 Wnioski**

Na podstawie testów, ocena implementacji, jakie parametry optymalne w funkcji np. rozmiaru obrazu, albo ogólnie

### **5 Ulepszenia**

Na podstawie wniosków co można usprawnić, co było nie tak, co może być lepsze

### **6 Bibliografia**