

Sprawozdanie z laboratorium: Informatyka w medycynie

Sprawozdanie I: Symulator tomografu komputerowego

Autorzy:	Anna Maziejuk	122566
	Kamil Jankowski	122558

Wstęp

Celem projektu było zaimplementowanie symulatora tomografu komputerowego.

Do tego celu zastosowaliśmy transformację Radona oraz stożkowy model układu emiter-detektor.

Dodatkowo zastosowaliśmy splot i zapis obrazu w formacie DICOM.

Aplikacja została zaimplementowana w języku Java.

Opis metody

Na samym początku wczytujemy bitmapę o równej wysokości i szerokości.

Następnie na podstawie oryginalnego obrazu generujemy sinogram. Odbywa się to w kilku krokach :

- Obliczenie pozycji emitera

```
double help0 = i * Math.PI / 180;  
Double EmiterX = Math.cos(help0) * radius + radius;  
Double EmiterY = Math.sin(help0) * (-radius) + radius;
```

- Obliczenie pozycji kolejnych detektorów w pętli

```
double help = help0 + Math.PI - (angle * Math.PI) / 360 + (angle * Math.PI * j)  
/(180 * (detectorsAmount - 1));  
Double DetektorX = Math.cos(help) * radius + radius;  
Double DetektorY = Math.sin(help) * (-radius) + radius;
```

- Obliczenie wartości piksela w sinogramie przy pomocy algorytmu Bresenhama

Wszystkie wartości kolorów poszczególnych pikseli na linii są sumowane, a następnie suma ta jest dzielona przez liczbę tych pikseli.

- Zastosowanie splotu dla pojedynczego wiersza sinogramu

```
if(j%2==1){  
double factor = -4 / ((Math.PI * Math.PI) * (j * j));  
if (i - j >= 0) {  
newColor += pix[row][i - j] * factor;  
}  
if (i + j < detectorsAmount) {  
newColor += pix[row][i + j] * factor;  
}}  
}}
```

- Normalizacja całości sinogramu (konieczna, gdyż ze splotu otrzymujemy również ujemne wartości niemożliwe do odtworzenia jako obraz)

```
int kol = (int) ((pix[i][j] - min) * (255.0 / (max - min)));
```

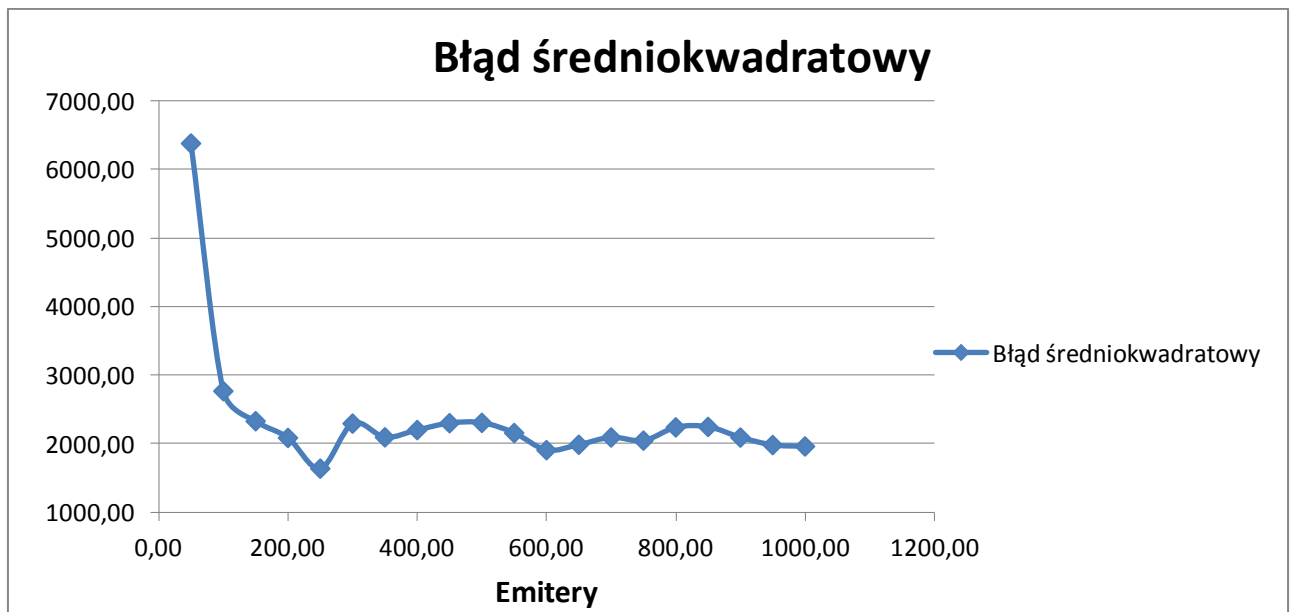
Kolejnym etapem jest odtworzenie obrazu z sinogramu, które również odbywa się w kilku podobnych krokach: obliczenie pozycji emiterów i detektorów, odtworzenie obrazu przy pomocy algorytmu Bresenhama, normalizacja kolorów pikseli.

Eksperyment

Eksperymenty przeprowadzano na obrazie „logan phantom” o wymiarach 200x200 pikseli. Przyjętą funkcją oceny działania algorytmu jest błąd średniokwadratowy obliczony na podstawie zestawienia obrazu wynikowego z wejściowym obrazem.

Badaliśmy wyniki pod kątem liczby wykonanych iteracji, detektorów, emiterów, wartości kąta rozwarcia stożka oraz zastosowania splotu.

1. Funkcja liczby emiterów:
Zmienność wielkości w zakresie 50-1000.



Wybrane wyniki:

f(50)=	6383
f(250)=	1641
f(550)=	2160
f(600)=	1911
f(800)=	2243
f(1000)=	1965

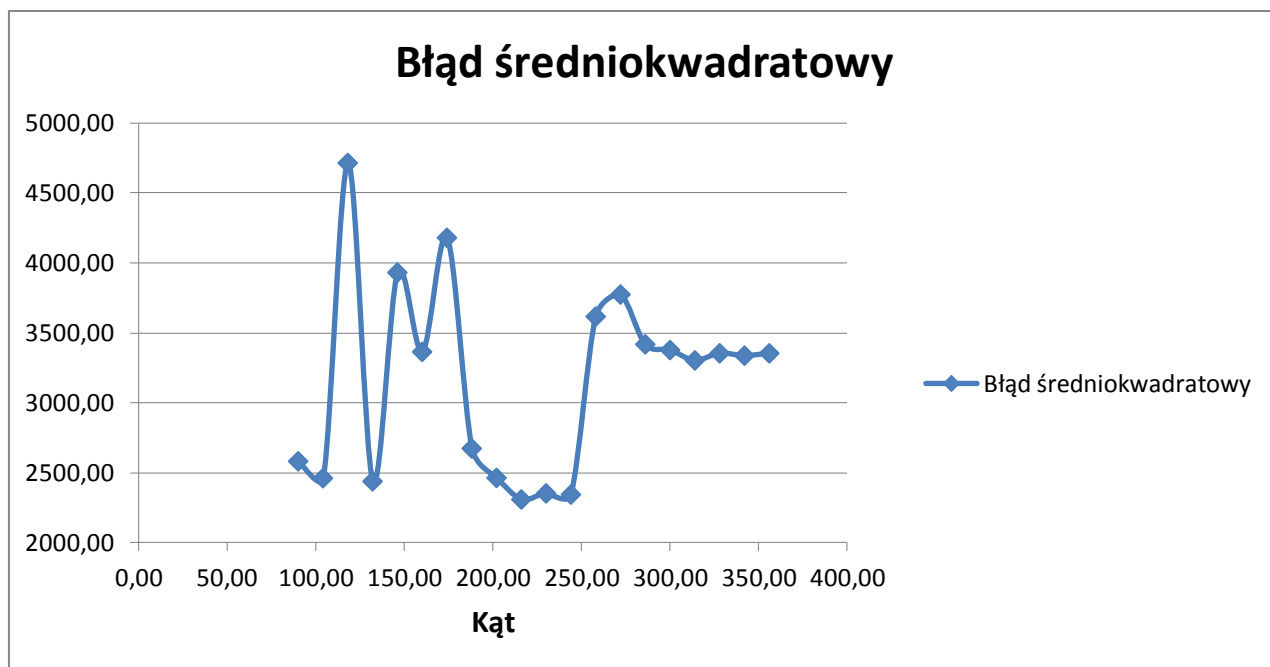
2. Funkcja liczby detektorów:
Zmienność wielkości w zakresie 50-1000.



Wybrane wyniki:

$f(50)=$ 3597
 $f(300)=$ 2377
 $f(450)=$ 3518
 $f(650)=$ 2469
 $f(1000)=$ 2363

3. Funkcja wartości kąta rozwarcia:
Zmienność wielkości w zakresie 90-356.



Wybrane wyniki:

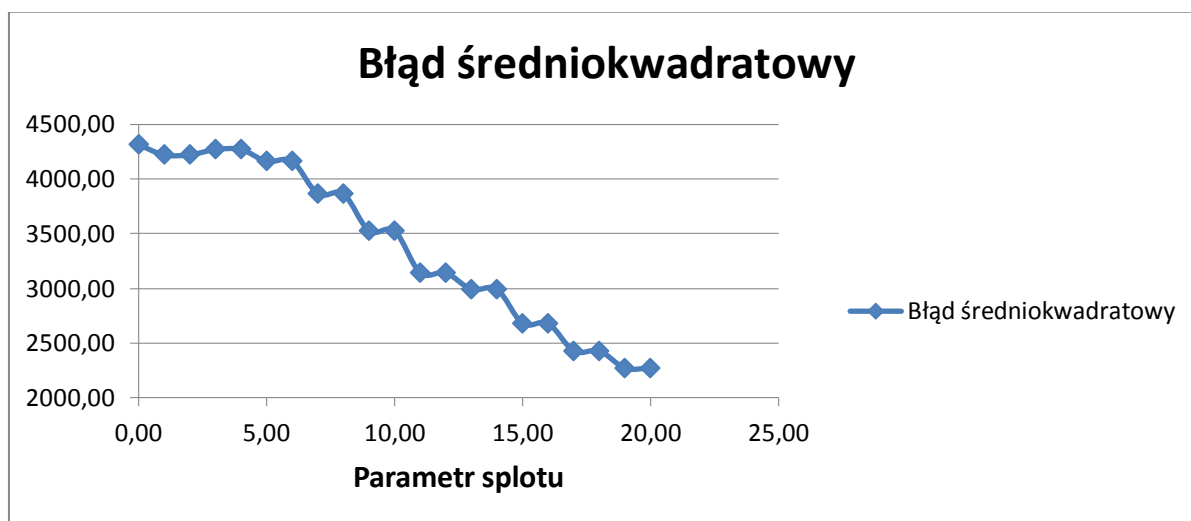
$f(90)=$ 2584

$f(216)=$ 2311

$f(356)=$ 3355

4. Zastosowanie splotu i liczby sąsiednich elementów wpływających na wartość danego piksela:

Zmienność wielkości w zakresie 0-20.



Wybrane wyniki:

$f(0)=$	4314	-brak splotu
$f(1)=$	4222	
$f(10)=$	3526	
$f(20)=$	2272	

5. Funkcja liczby iteracji:
Zmienność wielkości w zakresie 0-513.



Wybrane wyniki:

$f(0)=$ 55549

$f(270)=$ 1538

$f(513)=$ 2292

Wnioski

Wraz ze wzrostem liczby emiterów i detektorów maleje błąd średniokwadratowy, odpowiada to naszej wiedzy teoretycznej, gdyż dokładne odwzorowanie można w teorii uzyskać dla nieskończonej liczby emiterów.

Bardzo dziwne wyniki otrzymaliśmy przy badaniu wpływu rozwartości kąta na dokładność odwzorowania. W zakresie od 90-300 stopni wyniki są bardzo niestabilne, powyżej tego progu wyniki stabilizują się i praktycznie nie zmienia się już ocena naszego algorytmu. Wynika to prawdopodobnie z charakterystyki naszej funkcji oceny oraz tego, że oryginalny obraz jest w większości czarny.

Zgodnie z przewidywaniami wzrost ilości iteracji zmniejsza błąd średniokwadratowy. Okazuje się jednak, że wartość błędu po 100 iteracji praktycznie przestaje się zmniejszać, jednak z naszego punktu widzenia obraz wynikowy coraz dokładniej odwzorowuje oryginał. Znowu wynika to prawdopodobnie z charakterystyki samej funkcji oceny, a nie rzeczywistych zależności.

Największy wpływ na efektywność algorytmu okazał się mieć zastosowany splot. Na dodatek eksperymentowaliśmy z wielkością maski nakładanej na sinogram i z naszych obserwacji wynika, że jej wielkość ma drastyczny wpływ na efektywność algorytmu.