

Sprawozdanie z laboratorium: Informatyka w Medycynie

Część I: Symulator tomografu komputerowego

2 kwietnia 2017

Prowadzący: mgr inż. Iwo Błądek

Autorzy: **Sebastian Firlik** inf122485 I2 sebastian.firlik@student.put.poznan.pl
Piotr Hankiewicz inf122565 I2 piotr.hankiewicz@student.put.poznan.pl

Zajęcia piątkowe, 11:45.

Oświadczam/y, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autora/ów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

Udział autorów

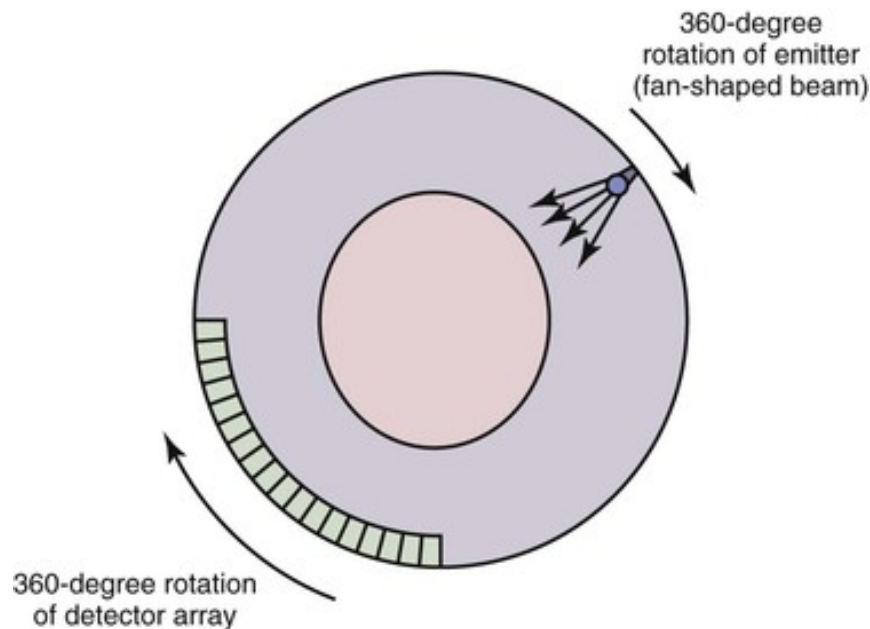
- SF zaimplementował tworzenie sinogramu, a także jego zapis do formatu DICOM, przygotował środowisko graficzne aplikacji
- PH zaimplementował odtwarzanie obrazu wejściowego na podstawie sinogramu wraz z filtrowaniem, przeprowadził eksperyment wyjaśniający wpływ niżej opisanych parametrów na jakość odtwarzanego obrazu.

1 Wstęp

Naszym zadaniem było stworzenie aplikacji desktopowej w wybranej technologii i zaimplementowanie w niej symulacji dwuwymiarowego tomografu komputerowego. Wszystkie wymagania zostały wypunktowane [tutaj](#). Do stworzenia naszej aplikacji użyliśmy:

- języka Python 3,
- środowiska PyQt5 do stworzenia interfejsu okienkowego,
- bibliotek dostępnych w języku Python (matplotlib, numpy, pyDicom...)

Jedynie do obsługi zapisu do formatu DICOM użyliśmy gotowej biblioteki ze względu na trudność manipulacji danymi w tym formacie. Wszystkie obliczenia, zarówno podczas generacji sinogramu, jak i przejściu do obrazu wynikowego, a także obrót emitera i detektorów w funkcji kąta zamodelowaliśmy samodzielnie.



Rysunek 1: Schemat działania tomografu

Na Rysunku 1 widzimy, jak działa nasz tomograf komputerowy - jeden emiter obraca się o 360° , wysyła promieniowanie rentgenowskie przez badany obiekt i każda wiązka trafia na odpowiedni detektor. W naszej symulacji, przy pomocy algorytmu Bresenhama tworzymy dyskretną linię, prowadzącą od emitera do detektora przez obraz i sumuje jasności pikseli

na tej drodze. W ten sposób, na każdej ścieżce emiter-detektor, dla każdego możliwego kąta obrotu emitera, otrzymujemy **sinogram**, czyli pośredni etap wizualizacji badanego obiektu. Nie podlega on żadnej analizie diagnostycznej. Następnie w wyniku operacji odwrotnej (Odwrotna Dyskretna Transformata Radona) otrzymujemy obraz wynikowy.

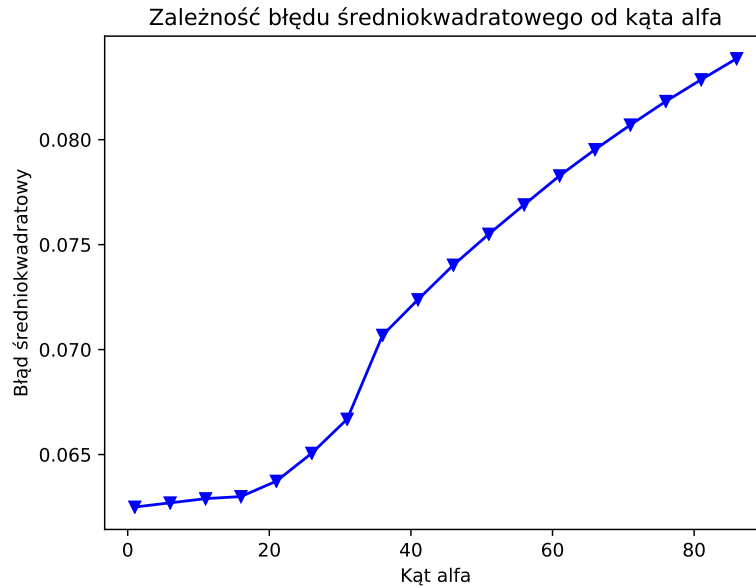
ODTR polega na tym, że z każdym kątem obrotu α i każdą parą emiter-detektor sprzężona jest suma jasności pikseli na ich drodze. Na każdym pikselu, znajdującym się na tej drodze, zostawiamy średnią jas odpowiednienność piksela, wynikającą z sumy z sinogramu. Po iteracyjnym odtworzeniu obrazu, bez nałożonego filtrowania widzimy niedokładny, rozmyty obraz, podobny do początkowego. NORMALIZACJA?

RYSUNEK-PIERWOTNY, SINOGRAM, ODTWORZONY

2 Analiza statystyczna

2.1 Wpływ kąta α na jakość przetwarzania

Pierwszym z parametrów wpływających na jakość przetwarzania jest kąt o jaki obraca się emiter w jednej iteracji.



Rysunek 2: Wpływ kąta alfa na jakość przetwarzania

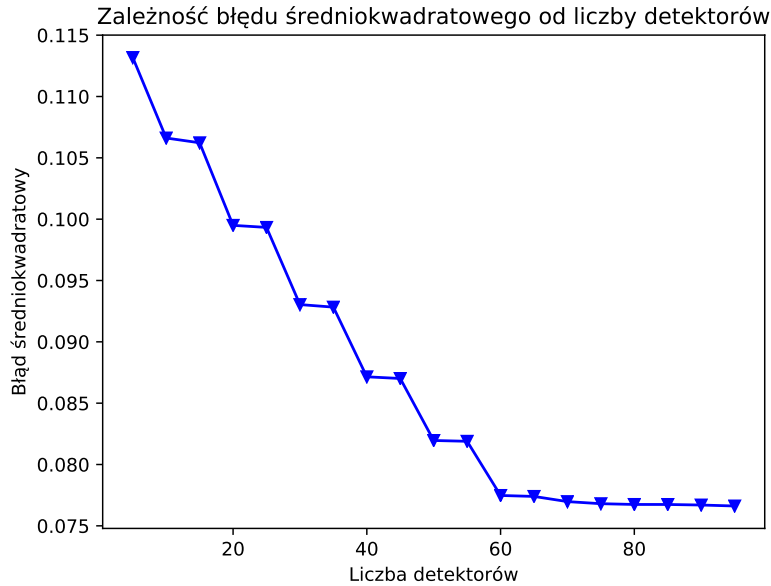
Jak widać na poniższym wykresie wraz ze wzrostem wartości kąta alfa wzrasta też wartość błędu średniokwadratowego. Wynika to z faktu, że im większy omawiany kąt tym mniej razy nastąpi zebranie informacji o obiekcie, a co za tym idzie obraz wyjściowy będzie coraz mniej dokładny. Przeprowadzona analiza wskazuje, że początkowo wartość błędu zwiększa się w nieznacznym stopniu, by dla wartości w przedziale $[35; 40]$ osiągnąć największy skok wartości. Następnie, aż do końca eksperymentu błąd rośnie liniowo względem zmiany wartości kąta alfa.

2.2 Wpływ liczby detektorów na jakość przetwarzania

Kolejnym z czynników jest liczba detektorów. Na osi y przedstawione są wartości błędu średniokwadratowego dla kolejnych wartości liczb detektorów, które znajdują się na osi x.

Wraz ze zwiększeniem liczby detektorów spada błąd średniokwadratowy. Wynika to z faktu, że im więcej detektorów w modelu tomografu tym gęściej wysyłane są wiązki z danego emitera, co skutkuje dokładniejszym obrazem. Spowodowane jest to tym, że im więcej wiązek jest wysyłanych tym mniej niezarejestrowanych przez nie pikseli. W takich sytuacjach, piksele, przez które nie przechodziła wiązka, na obrazie wyjściowym będą miały kolor czarny.

Należy zaznaczyć, że po przekroczeniu granicznej liczby detektorów, która zależna jest od pozostałych parametrów przetwarzania zmiana błędu średniokwadratowego stabilizuje się, zmienia się w bardzo niewielkim stopniu, pomimo początkowego bardzo wyraźnego spadku.



Rysunek 3: Wpływ liczby detektorów na jakość przetwarzania

W naszym eksperymencie, pozostałe parametry czyli kąt alfa i rozpiętość kątowa została ustalona na wartości odpowiednio 5° i 180° . Dla tak przyjętych parametrów omawiana wartość graniczna ustaliła się na wartość 60.

2.3 Wpływ rozpiętości kątowej detektorów na jakość przetwarzania

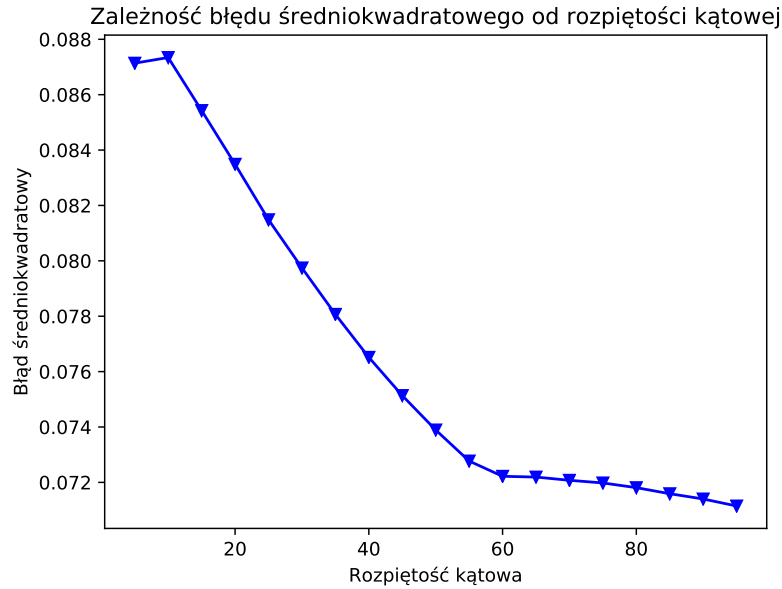
Ostatnim z czynników wpływających na przetwarzanie jest rozpiętość kątowa detektorów. Na osi y przedstawione są wartości błędu średniokwadratowego dla kolejnych wartości liczby detektorów, które znajdują się na osi x.

Wraz ze zwiększeniem rozpiętości kątowej detektorów spada błąd średniokwadratowy. Wynika to z faktu, że im szersze spektrum widzenia tym mniej elementów jest nieobjętych wiązkami w danej iteracji. Pełny obrót tomografu zapewnia zebranie informacji o obiekcie, jednakże im szerszy "kąt widzenia" tomografu, tym więcej zebranych o nim danych.

W początkowej fazie eksperymentu zaobserwować można gwałtowny spadek błędu średniokwadratowego. W odróżnieniu od poprzedniego przypadku, nie ma nagłej zmiany tendencji. Spadek wartości błędu średniokwadratowego zaczyna być powolniejszy, by osiągnąć względną stabilizację. Po osiągnięciu wartości 60° obserwuje się niewielki spadek wartości aż do końca eksperymentu.

3 Filtr

Ostatnim zadaniem omawianego projektu było zastosowanie filtrowania w celu poprawienia jakości obrazu wyjściowego. Na podstawie podanej w zadaniu literatury zaimplementowaliśmy filtr będący częścią techniki filtered backprojection. Filtr ten ma za zadanie po pierwsze „obcięcie” wierzchołków sygnału zawartego w sinogramie uzyskanego przez daną iterację.



Rysunek 4: Wpływ rozpiętości kątowej na jakość przetwarzania

Skutkuje to wyrównaniem koloru danego obiektu na obrazie wyjściowym. Po drugie, filtr ten tworzy ujemne wartości wokół tej części sygnału, która jest interpretacją konkretnego obiektu. Takie zachowanie filtra przeciwdziała rozmyciu obrazu wyjściowego, zarysowuje kontury obiektów.

Omawiany filtr może zostać zaimplementowany w następujący sposób: Początkowo należy dobrać odpowiednią wielkość okna z jakim działać będzie filtr. Następnie dla każdego piksela ustawiamy okno, dla którego przetwarzany piksel jest elementem środkowym o indeksie zero, wszystkie elementy okna znajdujące się po jego prawej stronie indeksowane są liczbami dodatnimi, a wszystkie elementy okna znajdujące się po jego lewej stronie są indeksowane liczbami ujemnymi. Następnie liczymy wartość o jaką zostanie powiększona wartość przetwarzanego piksela. Korzystamy z następujących zależności:

Jeżeli indeks ma wartość 0 to dodajemy do sumy wartość piksela o tym indeksie w oknie

Jeżeli indeks jest nieparzysty odejmujemy od sumy wartość piksela o tym indeksie pomnożonym przez $\frac{4}{\pi^2 k^2}$, gdzie k jest indeksem w przetwarzanym oknie.

Jeżeli indeks jest parzysty nie dodajemy do omawianej sumy żadnej wartości.