

Opis projektu:

Projekt polegał na zaimplementowaniu aplikacji przeprowadzającej symulację działania tomografu komputerowego. Aplikacja na podstawie zdjęcia w skali szarości symuluje projekcje promieni rentgenowskich i tworzy sinogram następnie na jego podstawie odtwarza pierwotny obraz.

Zdecydowaliśmy się na stożkowy model tomografu (tz. jeden emiter i wiele detektorów ułożonych po przeciwnych stronach obrazu)

Działanie:

Program zaczyna działanie od wczytania zdjęcia, następnie pozwala wybrać parametry symulacji (tj α - kąt kroku emitera i detektorów, l - kąt na jakim rozpięte są detektory, n - ilość detektorów). Kolejnym etapem jest utworzenie sinogramu. W każdym kroku korzystając z równania parametrycznego okręgu wyznaczamy pozycje emitera i detektorów na okręgu opisanym na zdjęciu wejściowym przesuniętych o odpowiedni kąt, następnie korzystając z algorytmu Bresenhama rzutujemy na obraz proste łączące emiter z detektorami. Wykorzystując transformatę Radona, dane z poszczególnych prostych zamieniamy na sinogram. Na osi pionowej sinogramu zaznaczone są kolejne kąty o które zostały przesunięte emiter wraz z detektorami a na osi poziomej uśrednione wartości pikseli na odcinku między emitersm a kolejnymi detektorami.

Na sinogram nakładamy filtr splotowy ram laka. Podczas tej operacji iterujemy po wszystkich pikselach obrazu i dla każdego piksela sumujemy wartość iloczyn sąsiednich pikseli z odpowiadającym elementem jądra. Za wartości pikseli poza zakresem obrazka przyjmujemy wartość 0

$$\begin{aligned} h[0] &= 1 \\ h[k] &= 0 \quad \text{for even values of } k \\ h[k] &= \frac{-4/\pi^2}{k^2} \quad \text{for odd values of } k \end{aligned}$$

gdzie 0 to środek jądra a k to odległość od jądra

Drugim krokiem jest odtworzenie pierwotnego obrazu z sinogramu, korzystając z odwrotnej transformaty Radona. Zaczynamy od czarnego obrazka o wymiarach obrazu początkowego. Odczytujemy kolejne wartości z sinogramu i dla odpowiadających im detektorom i kątom przesunięcia wyznaczamy prostą przechodzącą przez dany detektor i emiter przesunięte o odpowiedni kąt. Następnie odczytaną wartość dodajemy do wszystkich pikseli czarnego obrazka leżących na wcześniej wyznaczonej prostej. Potem wartość każdego piksela normalizujemy na podstawie ilości promieni które przez niego przechodziły.

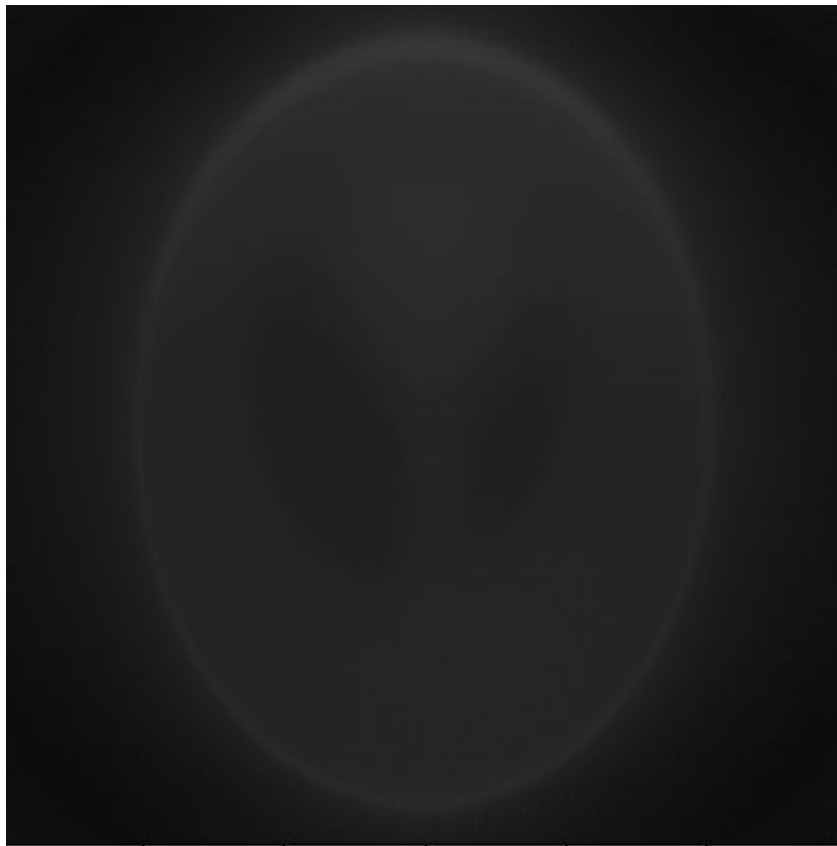
Wyniki działania:



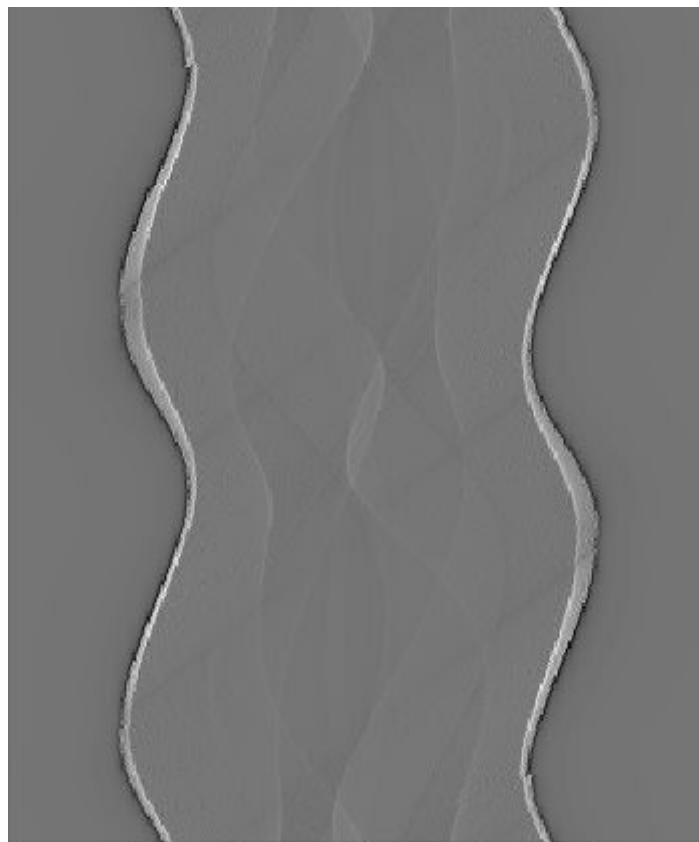
obraz wejściowy



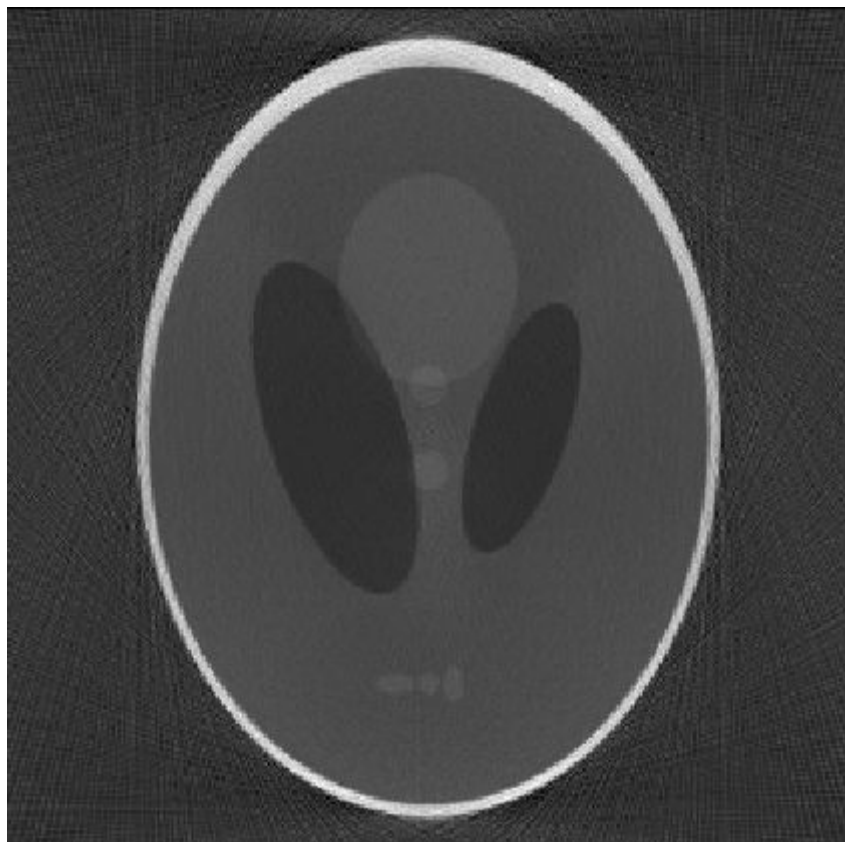
singoram bez filtrem $n = 300$, $l = 240$, $\alpha = 1$



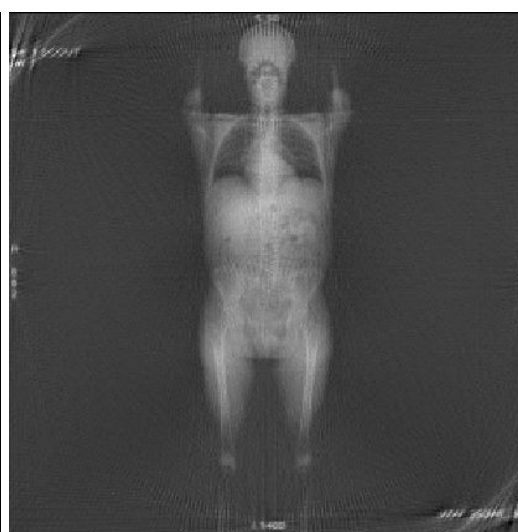
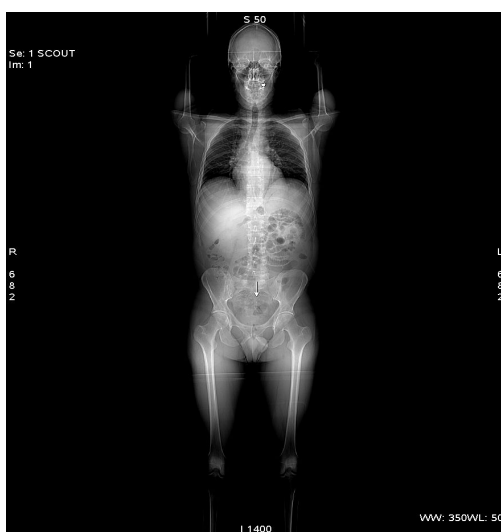
obraz wynikowy bez filtrem



singoram z filtrem $n = 300$, $l = 240$, $\alpha = 1$



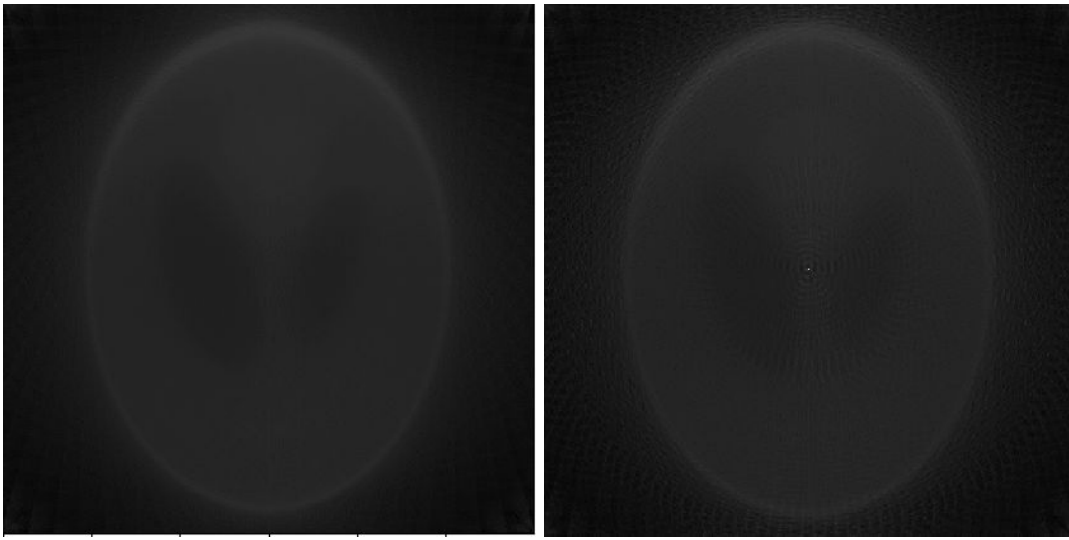
obraz wynikowy z filtrem



po lewej obraz wejściowy po prawej wynikowy

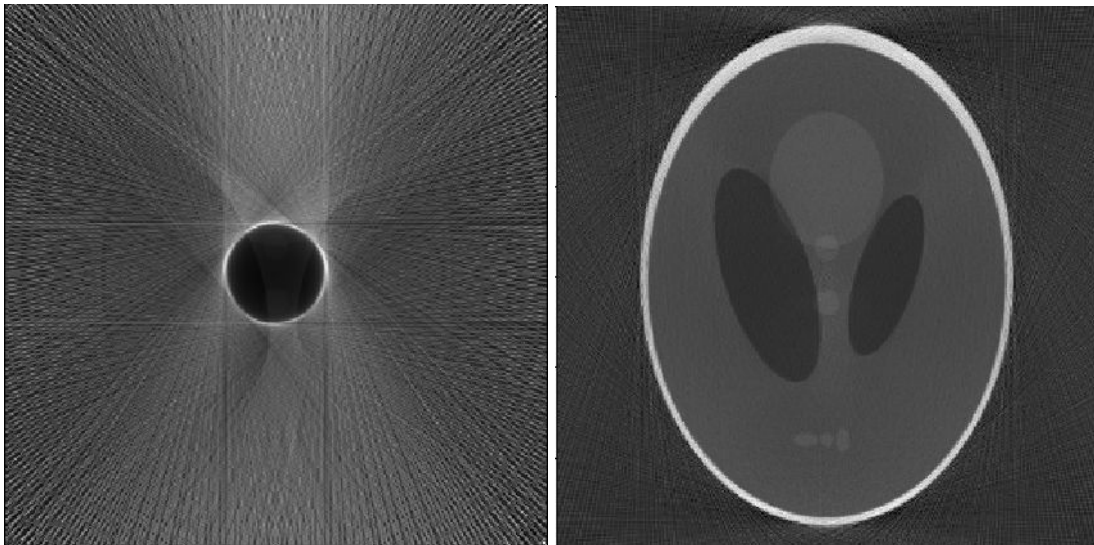
Analiza Błędu Średniokwadratowego:

Analiza błędu dla wyników bez filtra nie miała praktycznie sensu, ponieważ obrazki wychodził na tyle ciemny że różnice w błędach i tak była za mała, przykład:



jedno z tych zdjęć zostało wygenerowane przy parametrach $n=300$ i $\alpha = 1$ stopień drugie natomiast $n = 120$ i $\alpha = 5$. Pomimo tak dużych różnic w parametrach błąd średniokwadratowy różni się minimalnie i dzieje się tak praktycznie niezależnie od doboru parametrów.

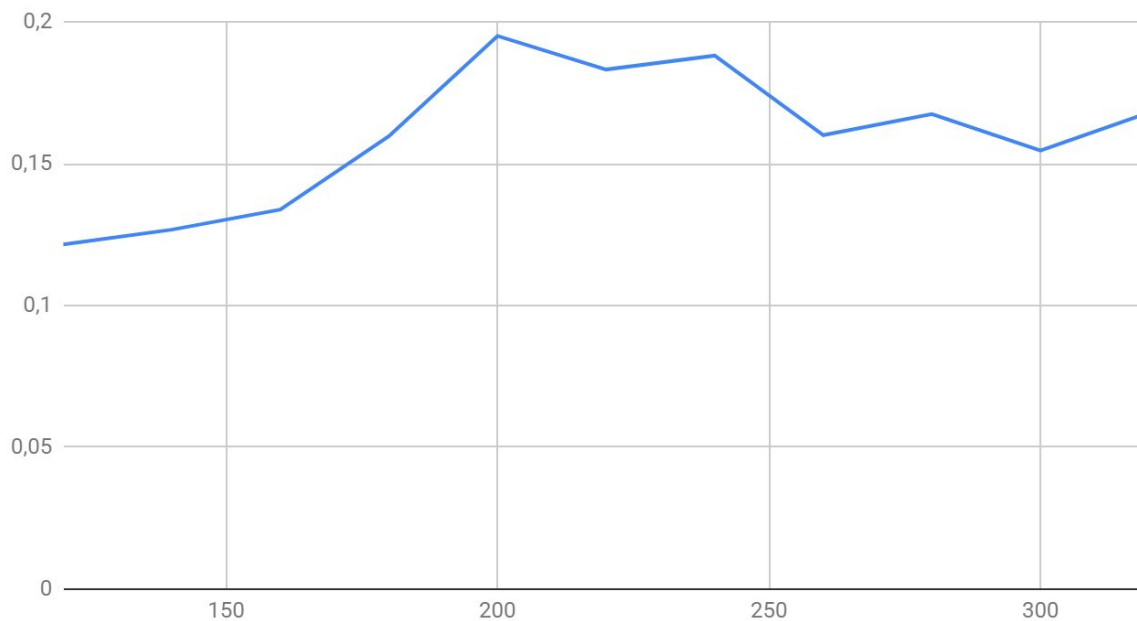
Natomiast dla obrazów z filtrem które są dużo bardziej rozświetlone i pokazują o wiele więcej szczegółów przez co wartość błędu średniokwadratowego praktycznie w ogóle nie odzwierciedla wyniku ponieważ miejsca w których na pierwotnym zdjęciu nic nie ma i są zupełnie czarne w obrazie wynikowym są szare (te miejsca stanowią około 40% obrazka) co za tym idzie np. obrazek zupełnie czarny nie wypada gorzej niż obraz wynikowy właśnie ze względu na te miejsca przykład:



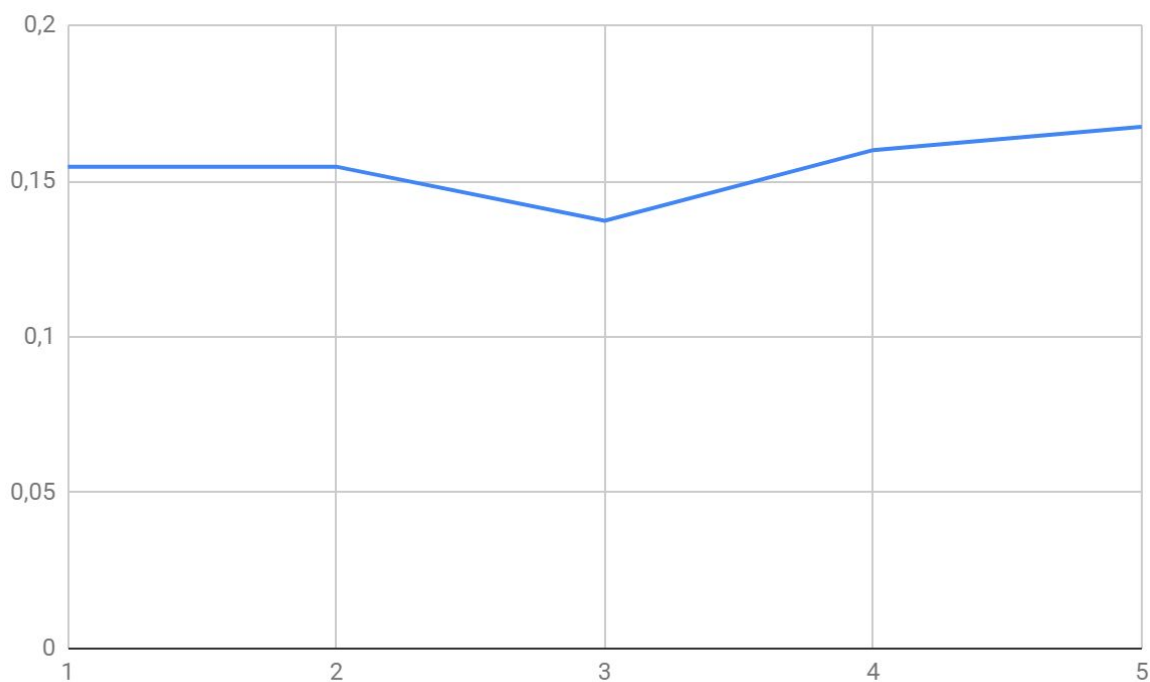
Obraz po lewej ma mniejszy błąd średniokwadratowy ponieważ jego obrzeża mają więcej czarnego .

Dlatego też jeśli chodzi o porównanie obrazów z filtrem i bez to obrazy z filtrem mają o wiele więcej szczegółów ale są dużo jaśniejsze co za tym idzie ich błąd jest dużo mniejszy(problem opisany wyżej).

Kilka przykładów potwierdzających tą tezę:

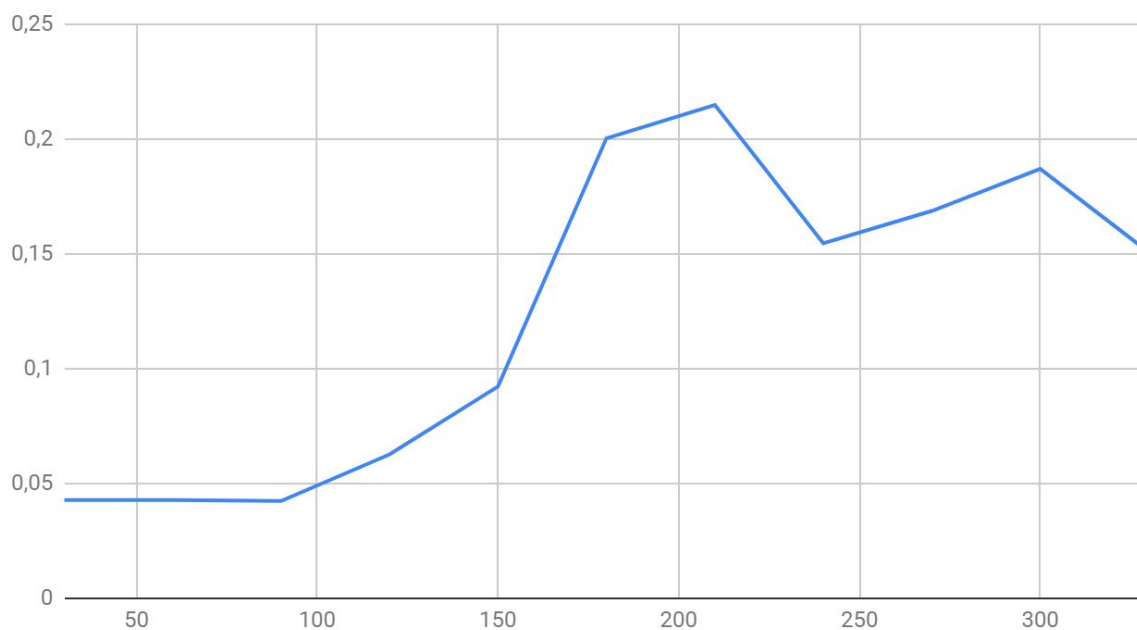


zmiana błędu w zależności od ilości detektorów(alfa = 1, l = 240)

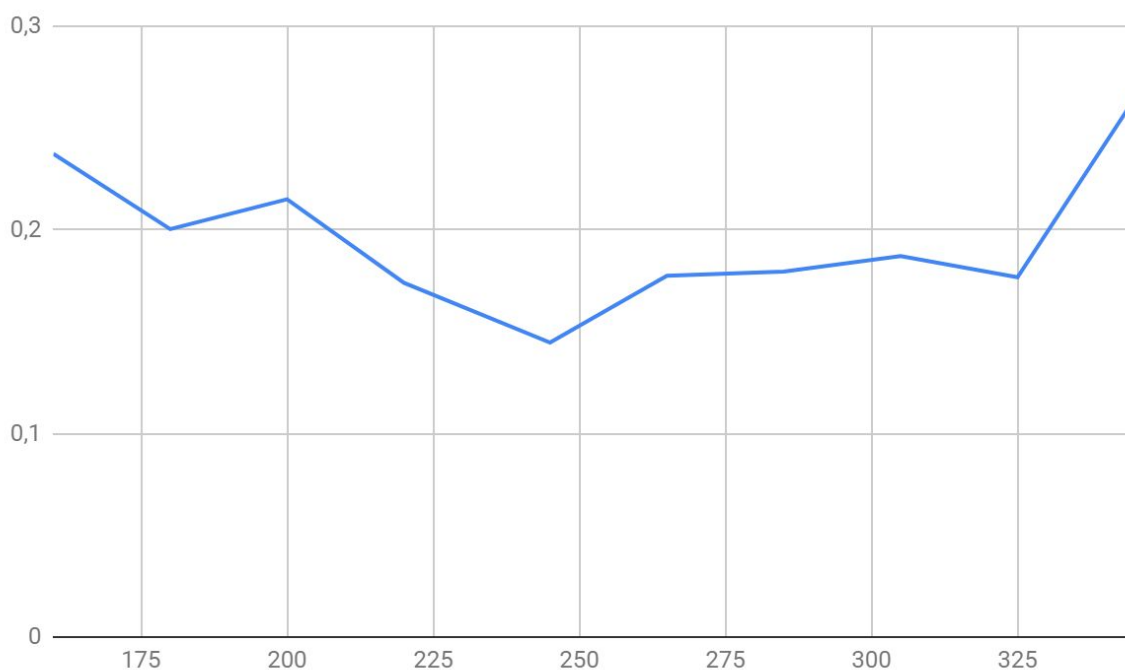


zmiana błędu w zależności od alfy (l = 240, n = 300)

Jedynym wariantem w którym udało się nam zauważyć jakieś zależności był dobór kąta rozwarcia, jednak różnice widać było tylko w dość mocno zawężonym obszarze dla którego obrazy wynikowe były dość podobne ponieważ w większej skali obrazy na których nic nie było widać i tak miały mniejszy błąd.



W całej skali (od 30 - 360 stopni) obrazy z małym kątem rozwarcia i tak miały mały błąd mimo że nic na nich nie było widać.



W mniejszej skali można zauważyć pewną tendencję, obrazu dla kąta rozwarcia ok. 240 stopni rzeczywiście wizualnie wyglądały lepiej.

