

Autorzy:

Mikołaj Szymczak 136814 grupa L2

Szymon Szczepański 136809 grupa L2

RAPORT INFORMATYKA W MEDYCYNIE: TOMOGRAF

Zastosowany model tomografu:

Model stożkowy

Zastosowany język programowania

Python + Jupyter Notebook

Wszelkie potrzebne biblioteki są wypisane w pliku projektu w zakładce Imports.

Pozyskiwanie odczytów dla poszczególnych detektorów

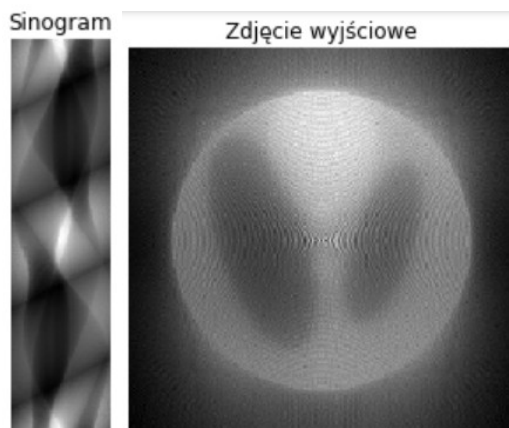
```
def make_sinogram(image):
    emitters, detectors = set_positions()
    sinogram = np.zeros([len(emitters), len(detectors[0])])

    for i, [emitterX, emitterY] in enumerate(emitters):
        for j, [detectorX, detectorY] in enumerate(detectors[i]):
            line = bresenham(emitterX, emitterY, detectorX, detectorY)

            for x, y in line:
                sinogram[i, j] += image[x, y]

    return sinogram
```

Jest to funkcja która umieszcza emiter oraz detektory na odpowiednich miejscach a następnie tworzy pustą macierz sinogramu. W pętli następuje iteracja po wszystkich emiterach umieszczonych w jednym miejscu, a następnie dla każdego emitera iterujemy po wszystkich detektorach i z algorytmu bresenhmana otrzymujemy linię naszego sinogramu i na jej podstawie dodajemy to do tablicy wynikowej.

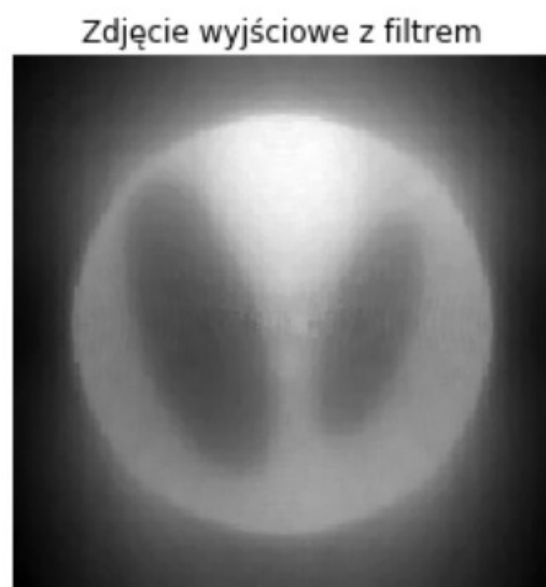


Przykładowy wynik obrazu reprezentującego sinogram oraz zdjęcie wyjściowe po odwrotnej transformacji tego sinogramu.

Filtrowanie sinogramu + zastosowany rozmiar maski

```
median(normalized_reverse_sinogram, selem=np.ones((5, 5)))
```

Zastosowany przez nas rozmiar maski to 5x5



Obraz wyjściowy po zastosowaniu filtru usuwającego szumy.

Ustalanie jasności poszczególnych punktów obrazu wynikowego oraz jego

przetwarzanie końcowe (np. uśrednianie, normalizacja)

```
def normalize(image):
    min_value = np.min(image)
    max_value = np.max(image)

    for i in range(len(image)):
        for j in range(len(image[i])):
            image[i, j] = (image[i, j] - min_value) / (max_value - min_value)

    return image
```

Jest to funkcja służąca do normalizacji otrzymanych wyników sinogramu oraz odwróconej transformaty sinogramu.

```
1[round(len(image) / 4) : 3 * round(len(image) / 4), round(len(image[0]) / 4) : 3 * round(len(image[0]) / 4)]
```

W celu uśrednienia obrazu wykorzystano zamieszczoną powyżej funkcję.

Wyznaczanie wartości miary RMSE na podstawie obrazu źródłowego oraz wynikowego

```
def mse(x, y):  
    err = np.sum((x.astype("float") - y.astype("float")) ** 2)  
    err /= float(len(x) * len(x[0]))  
    return err
```

Funkcja obliczająca średni błąd kwadratowy poprzez sumowanie potęgi różnic, a następnie podzielenie otrzymanej wartości przez całkowitą wielkość tablic.

Błąd średniokwadratowy: 0.2126804070273892

Przykładowy wynik działania na jednym obrazie.

Odczyt i zapis plików DICOM

```
def create_dicom(image):
    image *= 256
    image = image.astype(np.uint16)

    file_meta = Dataset()
    file_meta.MediaStorageSOPClassUID = '1.2.840.10008.5.1.4.1.1.2'
    file_meta.MediaStorageSOPInstanceUID = "1.2.3"
    file_meta.ImplementationClassUID = "1.2.3.4"
    file_meta.TransferSyntaxUID = ExplicitVRLittleEndian

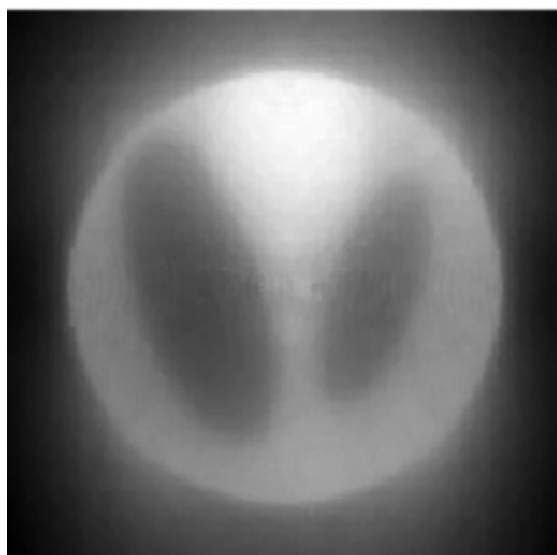
    ds = FileDataset("output.dcm", {}, file_meta=file_meta, preamble=b"\0" * 128)
    ds.is_little_endian = True
    ds.is_implicit_VR = False
    ds.PatientName = PATIENT_NAME
    ds.ContentDate = DATE.strftime('%Y%m%d')
    ds.ContentTime = DATE.strftime('%H%M%S.%f')
    ds.StudyDescription = DESCRIPTION
    ds.is_little_endian = True
    ds.is_implicit_VR = False

    ds.PixelRepresentation = 1
    ds.SamplesPerPixel = 1
    ds.PhotometricInterpretation = "MONOCHROME2"
    ds.HighBit = 15
    ds.BitsStored = 16
    ds.BitsAllocated = 16
    ds.SmallestImagePixelValue = str.encode('\x00\x00')
    ds.LargestImagePixelValue = str.encode('\xff\xff')
    ds.SOPClassUID = MRImageStorage
    ds.Columns = len(image)
    ds.Rows = len(image[0])

    ds.PixelData = image.tobytes()

    ds.save_as("output.dcm")
```

DICOM



Funkcja odpowiedzialna za obraz DICOM oraz przykładowy wynik jej działania.

Wynik eksperymentu sprawdzającego wpływ poszczególnych parametrów (liczba detektorów, liczba skanów, rozpiętość stożka/wachlarza z detektorami) na jakość obrazu wynikowego wyrażoną za pomocą miary RMSE. Jako wartości domyślne proszę przyjąć 180 detektorów, 180 skanów oraz rozpiętość wachlarza równą 180 stopni (PI)

Eksperyment proszę przeprowadzić na tym samym obrazie (nie może być to jednak obraz trywialny typu kółko lub dwa kwadraty). Proszę rozważyć jego następujące warianty (parametry, które nie są modyfikowane, powinny mieć wartości domyślne)

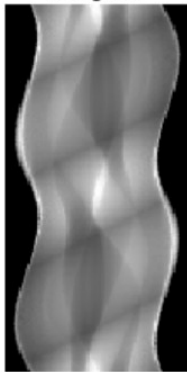
1. Liczba detektorów zmienia się od 90 do 720 z krokiem 90:

Tabela błędów średniokwadratowych:

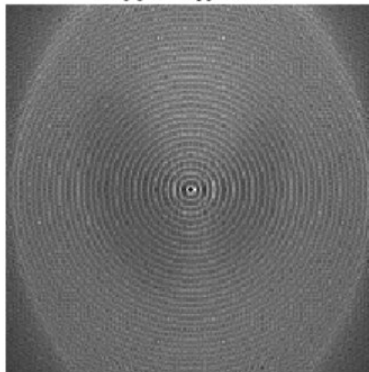
Liczba detektorów	Błąd średniokwadratowy
90	0,17363184311092342
180	0,17363184311092342
270	0,30806746738874735
360	0,331152007525297
450	0,3353641270973506
540	0,3580343356316612
630	0,3524163489147708
720	0,35149359807045394

Wyniki w postaci obrazów dla kolejnych iteracji:

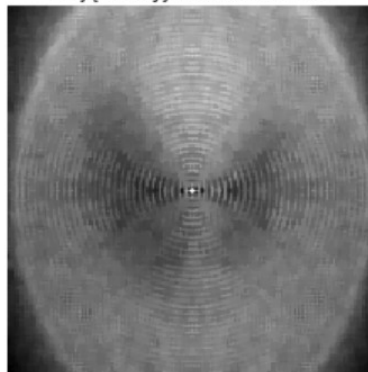
Sinogram



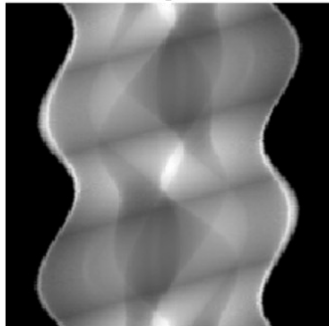
Zdjęcie wyjściowe



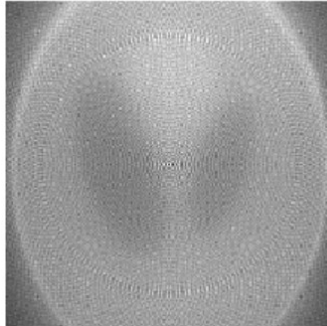
Zdjęcie wyjściowe z filtrem



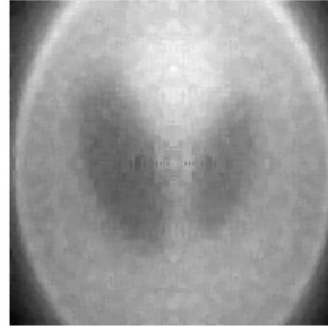
Sinogram



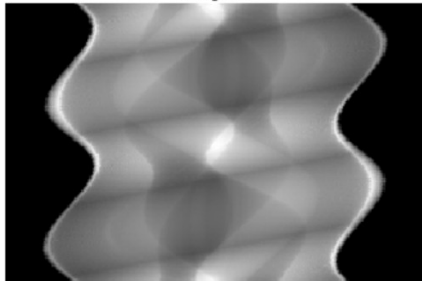
Zdjęcie wyjściowe



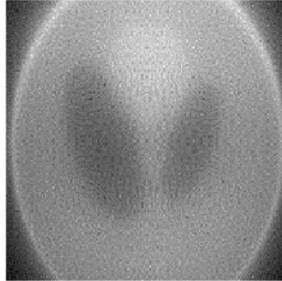
Zdjęcie wyjściowe z filtrem



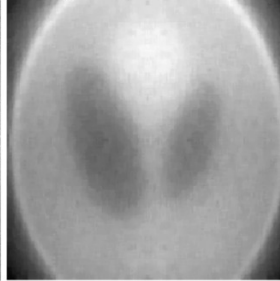
Sinogram



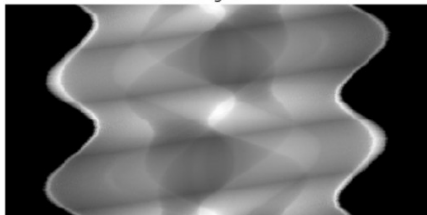
Zdjęcie wyjściowe



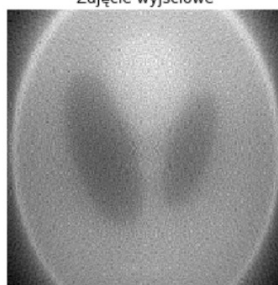
Zdjęcie wyjściowe z filtrem



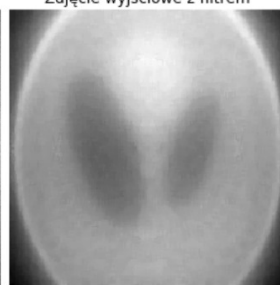
Sinogram



Zdjęcie wyjściowe



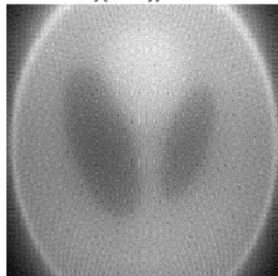
Zdjęcie wyjściowe z filtrem



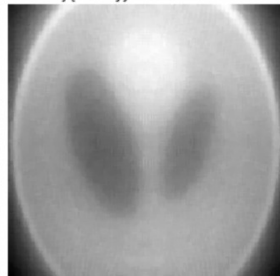
Sinogram

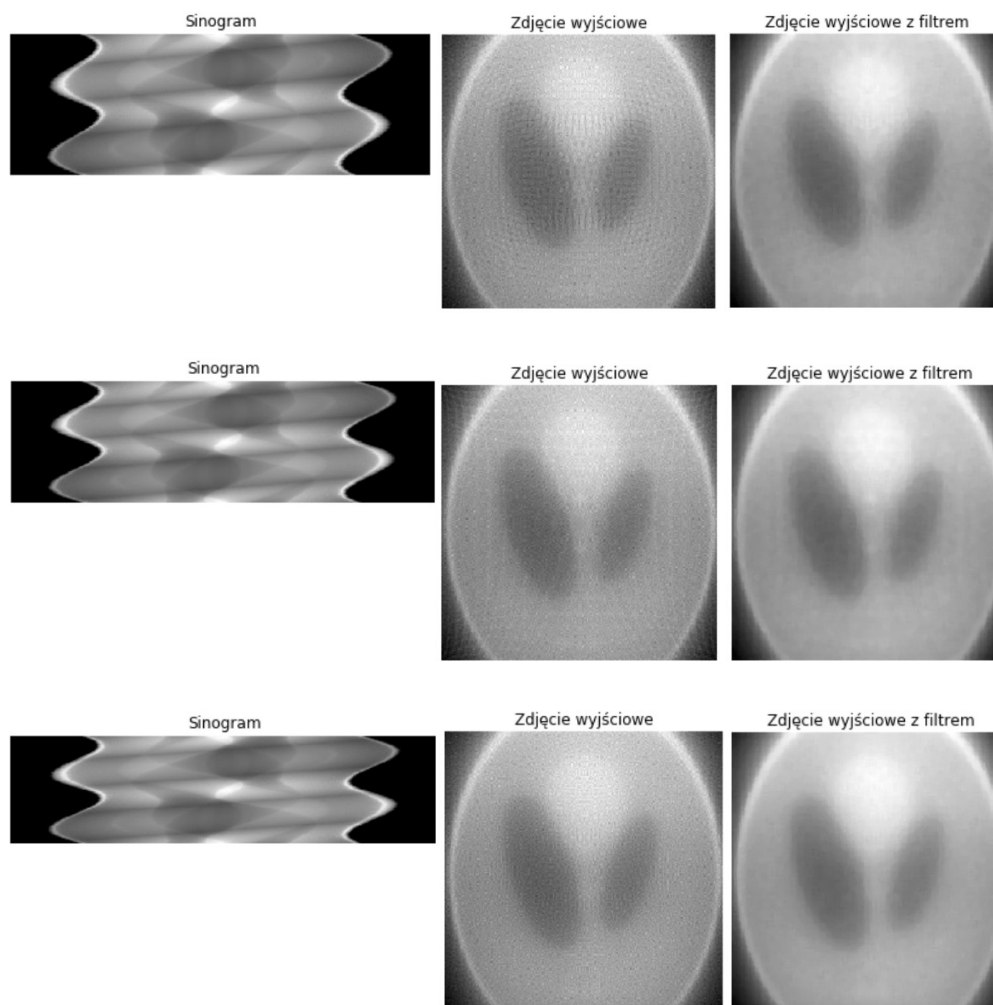


Zdjęcie wyjściowe



Zdjęcie wyjściowe z filtrem





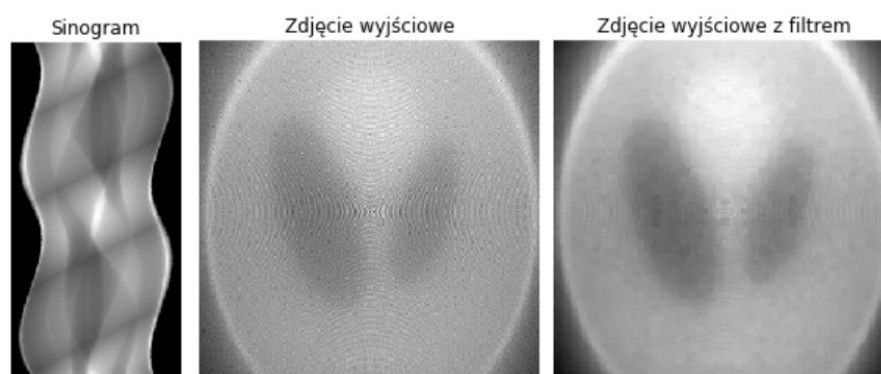
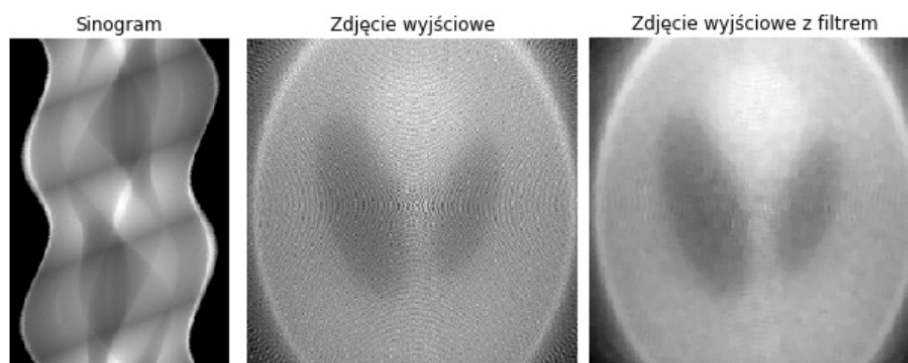
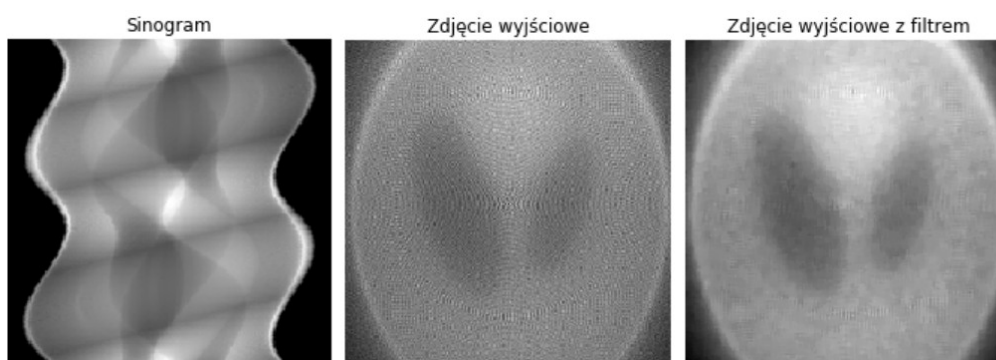
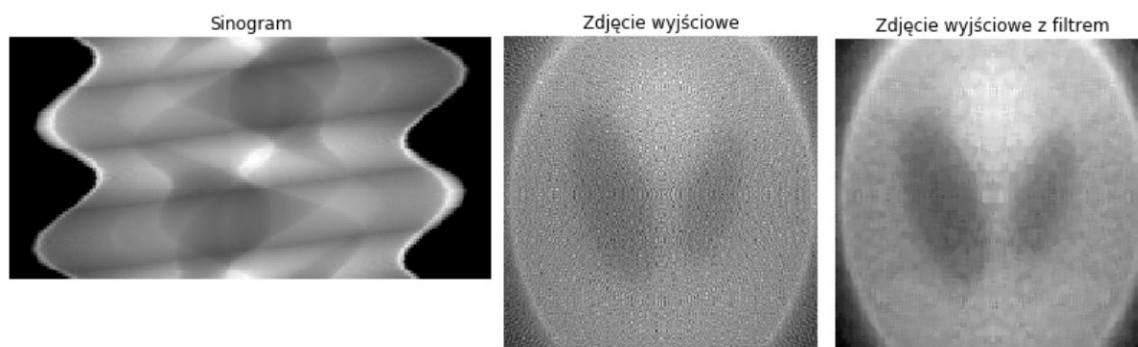
Z każdą iteracją, a co za tym idzie zwiększoną liczbą detektorów otrzymuje coraz szerszy obraz naszego sinogramu, co przekłada się na jakość końcową uzyskanego zdjęcia, szczególnie przed odfiltrowaniem.

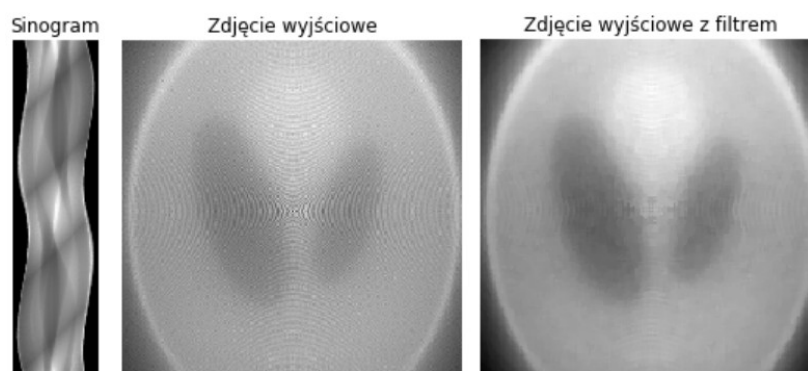
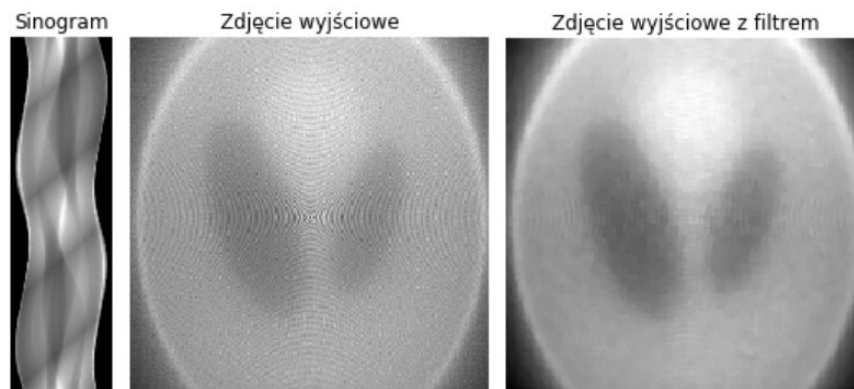
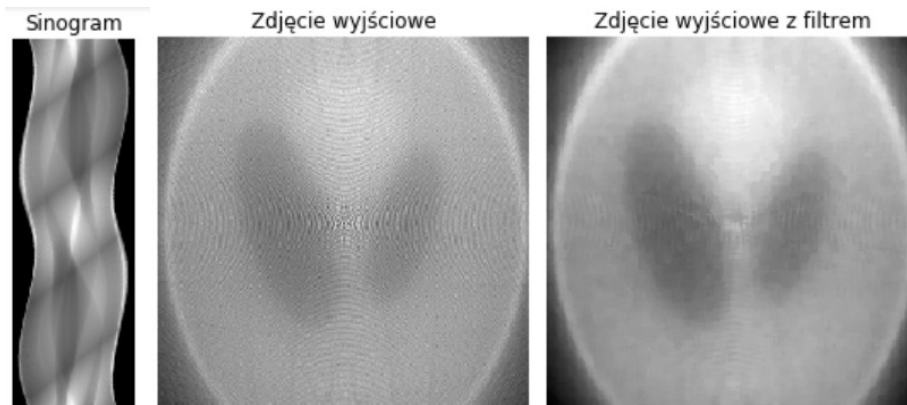
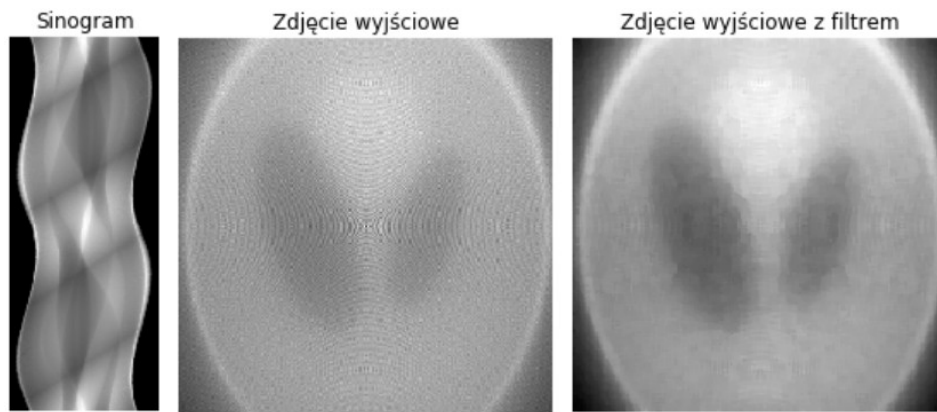
2. Liczba skanów zmienia się od 90 do 720 z krokiem 90:

Tabela błędów średniokwadratowych:

Liczba skanów	Błąd średniokwadratowy
90 (x4)	0,29830203025063967
180 (x2)	0,25418988590136243
270 (x1.33)	0,3389849417389246
360 (x1)	0,37721812979121166
450 (x0.8)	0,38004032228529216
540 (x0.66)	0,36798068461800526
630 (x0.57)	0,38635025960817165
720 (x0.5)	0,383257486789202

Wyniki w postaci obrazów dla kolejnych iteracji:





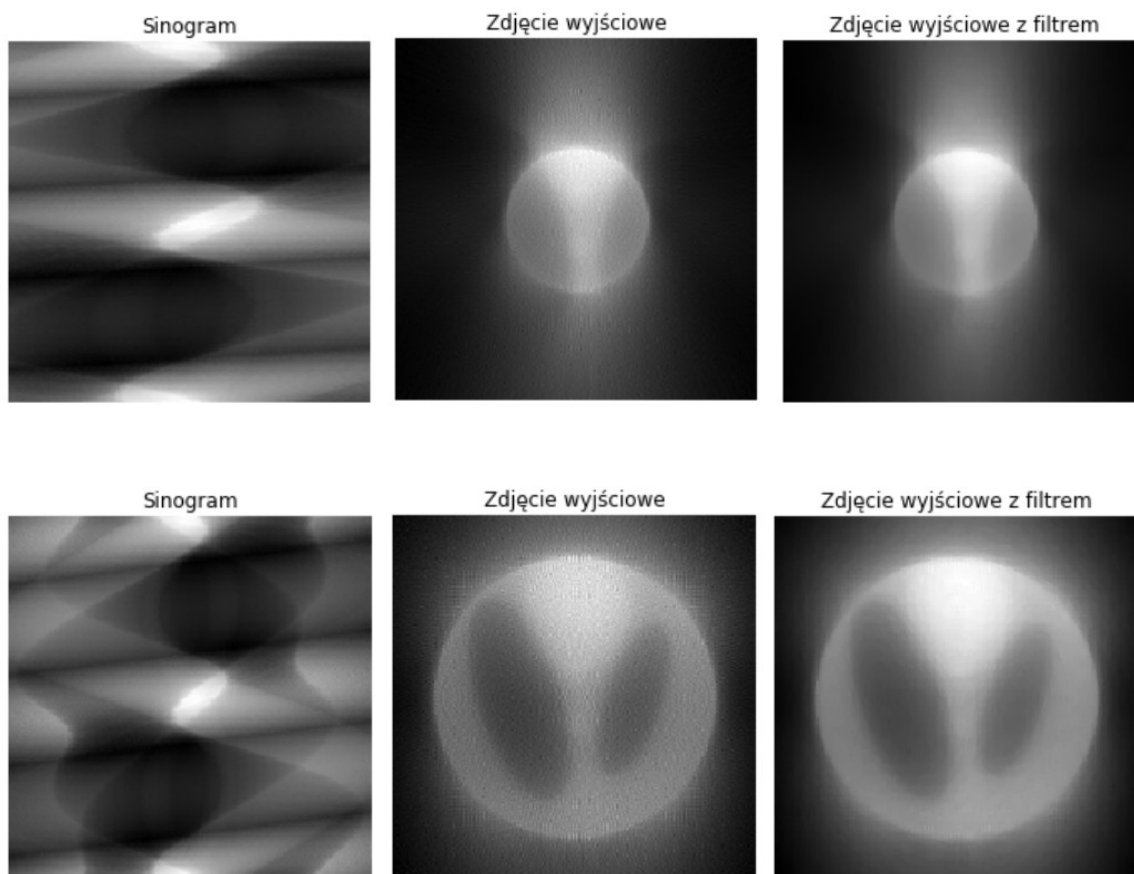
Z każdą iteracją długość sinogramu się wydłuża, za sprawą większej ilości próbek wykonanych przez pary emiter-detektor. Dzięki większej ilości próbek jakość obrazu się stosunkowo poprawia.

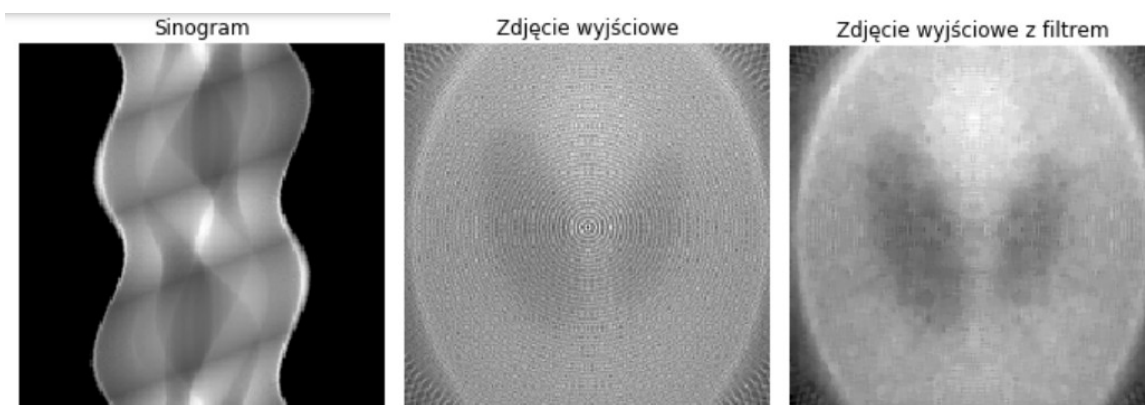
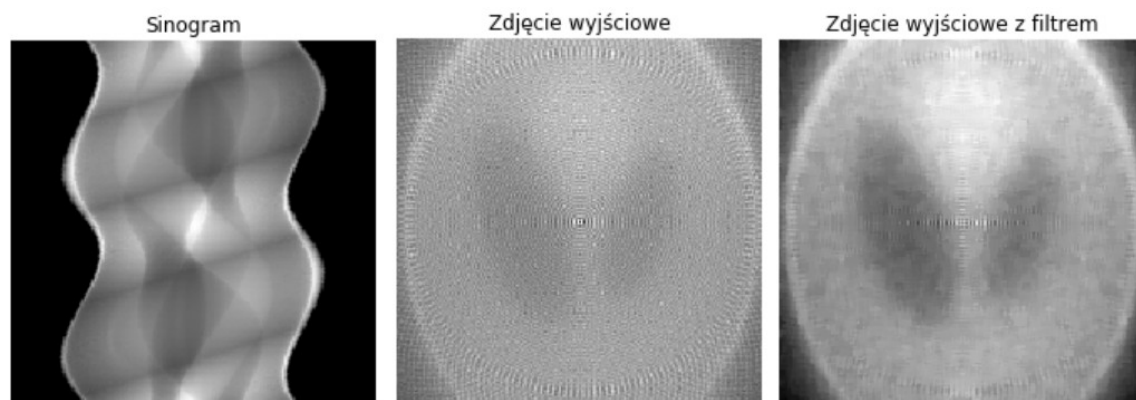
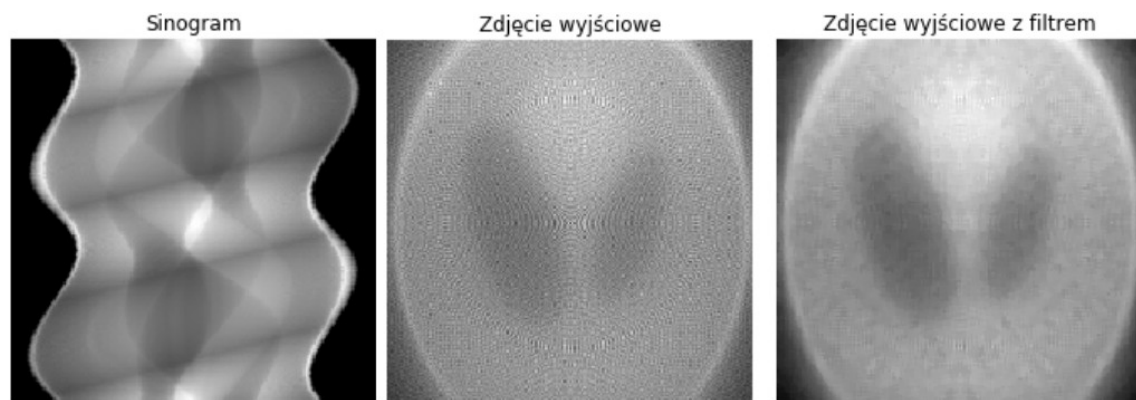
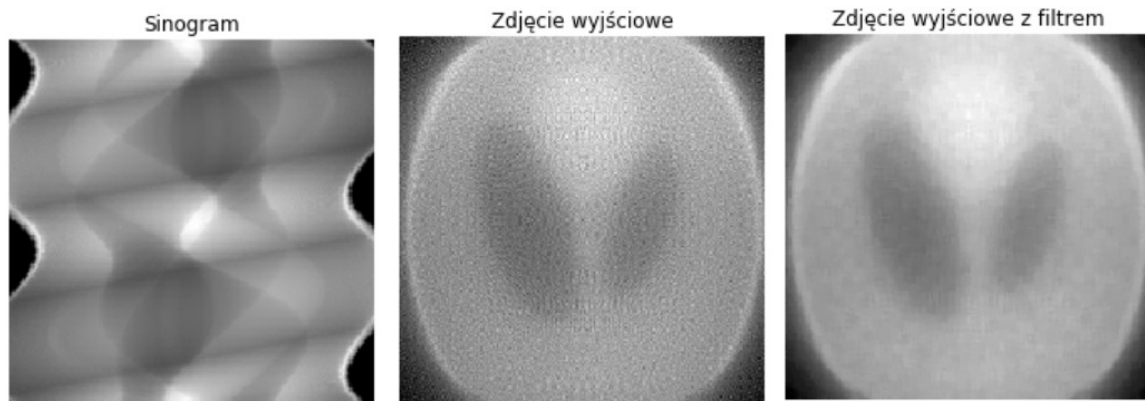
3. Rozpiętość wachlarza zmienia się od 45 do 270 stopni z krokiem 45 stopni:

Tabela błędów średniokwadratowych:

Rozpiętość wachlarza	Błąd średniokwadratowy
45	0,14474119259663606
90	0,21268040702738916
135	0,3371477552002176
180	0,3216264667999515
225	0,36338530612782177
270	0,34496575567097454

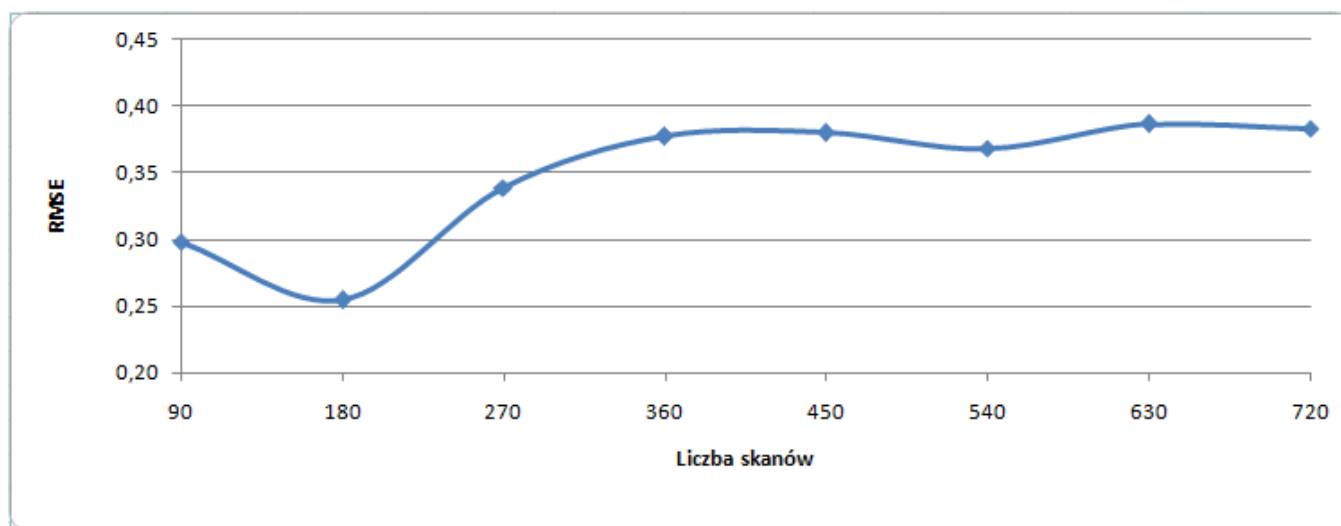
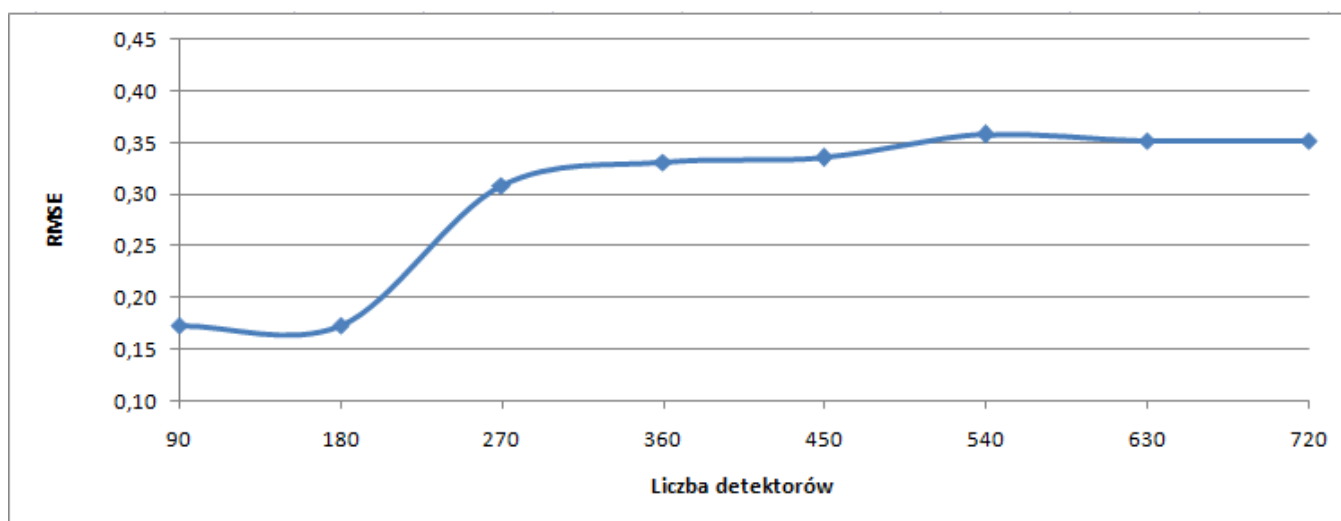
Wyniki w postaci obrazów dla kolejnych iteracji:





Wraz z każdą iteracją gdy rozpiętość wachlarza się zwiększa dochodzi do coraz mniejszego skupienia na elementach obrazu przez co po osiągnięciu pewnego progu obraz pogarsza się zyskując coraz większą ziarnistość.

Dla każdego wariantu proszę przedstawić wykres pokazujący zależność RMSE (oś Y) od aktualnej wartości zmienianego parametru (oś X) oraz krótko skomentować zaobserwowany przebieg -- zwłaszcza, czy wnioski wynikające z tego przebiegu są zgodne z oceną subiektywną jakości obrazu





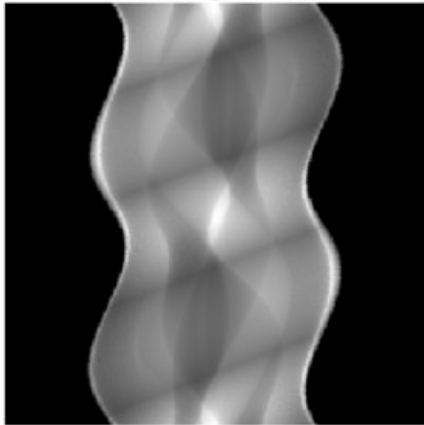
We wszystkich 3 przebiegach możemy zaobserwować tendencję do wzrostu wartości RMSE na samym początku, jednak po krótkiej chwili stabilizuje się uzyskując wartości oscylujące w granicach 0,35. Nie oddaje jednak ona charakteru uzyskanych obrazów na których można zobaczyć ewidentną różnicę w postaci poprawy lub pogorszenia obrazu. Zdecydowanie lepiej w tym wypadku sprawdzają się w ocenie subiektywne metody polegające na ocenie uzyskanych wyników kolokwialnie mówiąc „na oko”.

Dla dwóch wybranych obrazów oraz następujących parametrów: liczba detektorów = 360, liczba skanów = 360, rozpiętość wachlarza = 270 stopni, proszę wykonać dwa warianty obliczeń -- z włączonym i wyłączonym filtrowaniem sinogramu. Dla każdego obrazu proszę zaprezentować RMSE dla obrazu bez filtrowania i z filtrowaniem oraz krótko skomentować różnice w jakości między obrazami

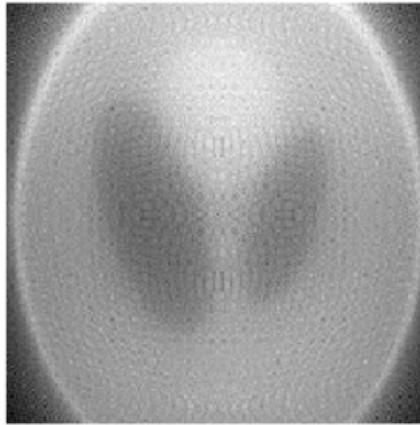
Liczba detektorów	Liczba skanów	Rozpiętość wachlarza	RMSE bez filtra	RMSE z filtrem
360	360	270	0,37971754256473894	0,3778036024273121
360	360	45	0,1467152037962746	0,14635749025411274
180	180	180	0,32478841985182855	0,3216264667939948

Zastosowanie filtrowania ma widoczny wpływ na jakość obrazu co można gołym okiem zobaczyć na zamieszczonych zrzutach ekranu przedstawiających wyniki operacji. Jeśli chodzi o średnie odchylenie standardowe to podobnie jak było to opisane w poprzednim punkcie, nie jest satysfakcjonującym odzwierciedleniem jakości konkretnych obrazów.

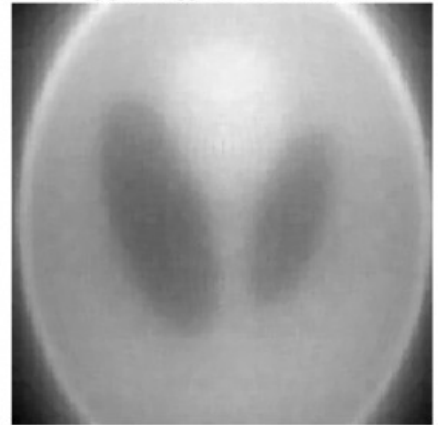
Sinogram



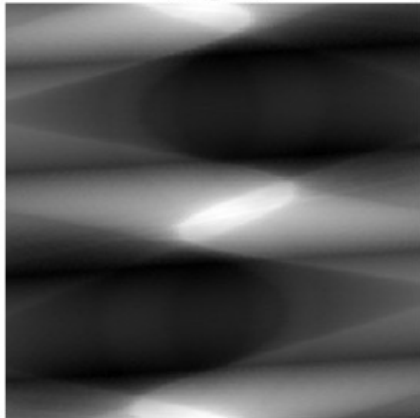
Zdjęcie wyjściowe



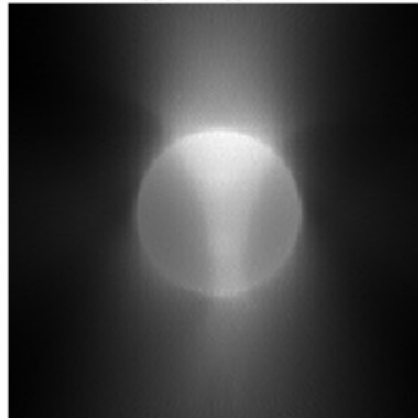
Zdjęcie wyjściowe z filtrem



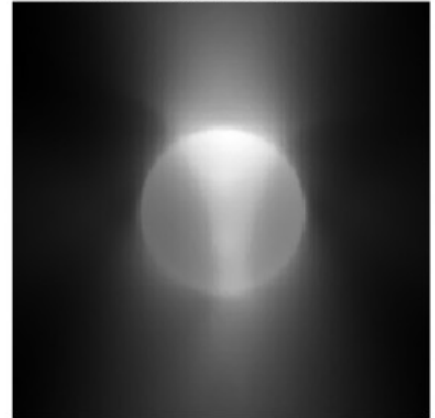
Sinogram



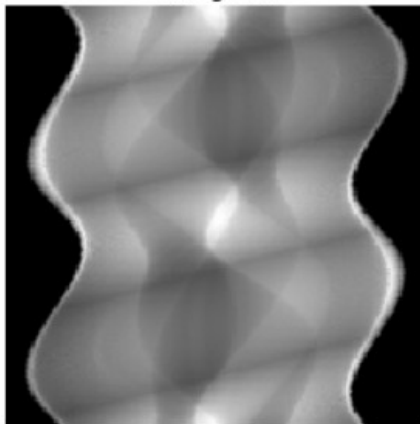
Zdjęcie wyjściowe



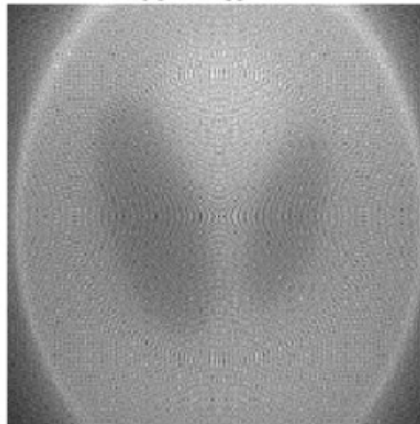
Zdjęcie wyjściowe z filtrem



Sinogram



Zdjęcie wyjściowe



Zdjęcie wyjściowe z filtrem

