

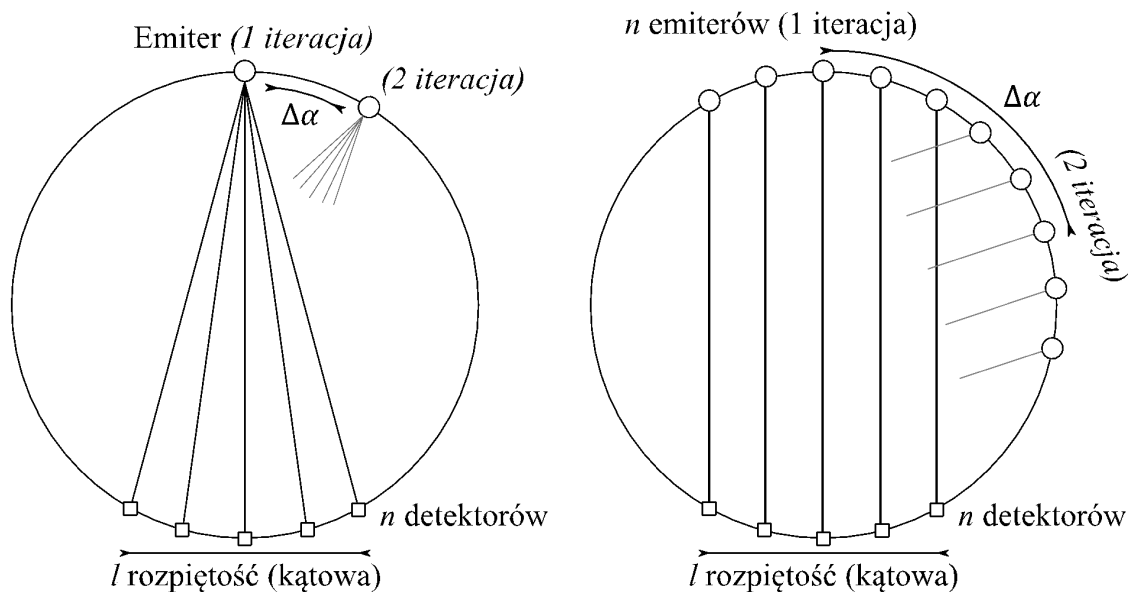
Symulator tomografu komputerowego

Opis

Implementacja aplikacji symulującej działanie tomografu komputerowego (symulacja dwuwymiarowa). Zobacz: <https://www.youtube.com/watch?v=tgNP-n2z3po>

Wymagania obowiązkowe

- Aplikacja okienkowa napisana w dowolnym języku programowania (Java, C#, Python). W przypadku Pythona może być to również (interaktywny) notebook.
- Wejściowy format obrazu: bitmapa, obraz prostokątny, skala szarości.
- Aplikacja musi wykonać transformatę Radona obraz wejściowy \rightarrow sinogram i odwrotną transformatę sinogram \rightarrow obraz wyjściowy. Wymagana jest wizualizacja wyników (obraz wejściowy, sinogram, obraz wyjściowy).
- Niedozwolone jest wykorzystanie gotowych implementacji transformaty Radona -- obliczenia muszą być wykonane samodzielnie. Ponadto nie wolno założyć stałej pozycji emitera(ów)/detektorów i symulować obrót dokonując rotacji obrazu. Ruch emitera(ów) i detektorów należy zamodelować samemu (funkcja kąta).
- Aplikacja powinna umożliwić wygenerowanie sinogramu i obrazu wyjściowego bez pokazania kroków pośrednich oraz z pokazywaniem (iteracyjnie), np. za pomocą suwaka służącego do regulacji postępu obrotu emitera(ów) i detektorów.
- Należy wykorzystać jeden z dwóch modeli emiter/detektor: stożkowy lub równoległy.
- Aplikacja powinna móc pozwolić konfigurować następujące elementy:
 - a) Krok $\Delta\alpha$ układu emiter/detektor.
 - b) Dla jednego układu emiter/detektor liczbę detektorów (n).
 - c) Rozwartość /rozpiętość układu emiter/detektor (l).



- Należy wykorzystać algorytm Bresenhama do liniowego przejścia po kolejnych pikselach obrazu dyskretnego.
- Symulację pochłaniania promieniowania można zasymulować na jeden z wybranych sposobów: addytywny lub subtraktywny. Należy sobie poradzić z normalizacją wyników.

Wymagania na 4.0

- Aplikacja powinna pozwalać na odczyt i zapis uzyskanego obrazu w standardzie DICOM wraz z uwzględnieniem (możliwość wprowadzenia z interfejsu):
 - o podstawowych informacji o pacjencie
 - o daty badania
 - o komentarzy

Poprawność zapisanego pliku należy zweryfikować w dowolnej (darmowej) przeglądarce plików DICOM, np. <https://www.imaio.com/en/Imaios-Dicom-Viewer>.

Uwaga: Należy wykorzystać bibliotekę do obsługi plików DICOM, nie należy implementować obsługi plików DICOM samodzielnie!

Wymagania na 5.0

- Należy zastosować proste filtrowanie (splot), aby zredukować szum powstały przez niedokładną (dyskretną/skończoną) odwrotną transformację (link poniżej) – użycie filtra powinno być opcją w aplikacji.
- Należy dokonać prostej analizy statystycznej w oparciu o jedną zdefiniowaną miarę jakości. Np. mając obraz wejściowy i wyjściowy można policzyć błąd średniokwadratowy (po wszystkich pikselach różnicy obrazu wejściowego i wyjściowego). Taka analiza powinna uwzględnić następujące elementy:

- o zmianę błędu średniokwadratowego podczas wykonywania kolejnych iteracji odwrotnej transformaty Radona,
- o zmianę błędu średniokwadratowego przy zwiększaniu dokładności próbkowania (trzy uprzednio wymienione parametry modelu emiter/detektor),
- o zmianę błędu średniokwadratowy po włączeniu filtrowania.

Raport

Raport powinien zawierać następujące elementy (część z nich dotyczy wymagań na 4.0 i 5.0 - zostało to zaznaczone):

1. Skład grupy
2. Zastosowany model tomografu (stożkowy lub równoległy)
3. Zastosowany język programowania oraz dodatkowe biblioteki
4. Opis głównych funkcji programu (ilustracja za pomocą fragmentów kodu źródłowego)
 - a. pozyskiwanie odczytów dla poszczególnych detektorów
 - b. filtrowanie sinogramu, zastosowany rozmiar maski (wymaganie na 5.0),
 - c. ustalanie jasności poszczególnych punktów obrazu wynikowego oraz jego przetwarzanie końcowe (np. uśrednianie, normalizacja),
 - d. wyznaczanie wartości miary RMSE na podstawie obrazu źródłowego oraz wynikowego (wymaganie na 5.0),
 - e. odczyt i zapis plików DICOM (wymagania na 4.0).
5. 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) (wymagania na 5.0)
 - a. 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):
 - i. liczba detektorów zmienia się od 90 do 720 z krokiem 90,
 - ii. liczba skanów zmienia się od 90 do 720 z krokiem 90,
 - iii. rozpiętość wachlarza zmienia się od 45 do 270 stopni z krokiem 45 stopni.
 - b. 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.
 - c. 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.

Linki

- Iteracyjna symulacja: <https://www.youtube.com/watch?v=tgNP-n2z3po>
- Algorytm Bresenhama: https://pl.wikipedia.org/wiki/Algorytm_Bresenhama
- Filtrowanie: <http://www.dspguide.com/ch25/5.htm>
- Splot: [https://pl.wikipedia.org/wiki/Splot_\(analiza_matematyczna\)](https://pl.wikipedia.org/wiki/Splot_(analiza_matematyczna))
- DICOM: <https://pl.wikipedia.org/wiki/DICOM>
- Wzory na wyznaczanie współrzędnych punktów odpowiadających emiterowi i detektorom:
<http://www.cs.put.poznan.pl/swilk/pmwiki/uploads/Dydaktyka/tomograf-wzory.pdf>
- Przykładowe obrazy do testowania:
<http://www.cs.put.poznan.pl/swilk/pmwiki/uploads/Dydaktyka/tomograf-zdjecia.zip>