

Symulator tomografu komputerowego

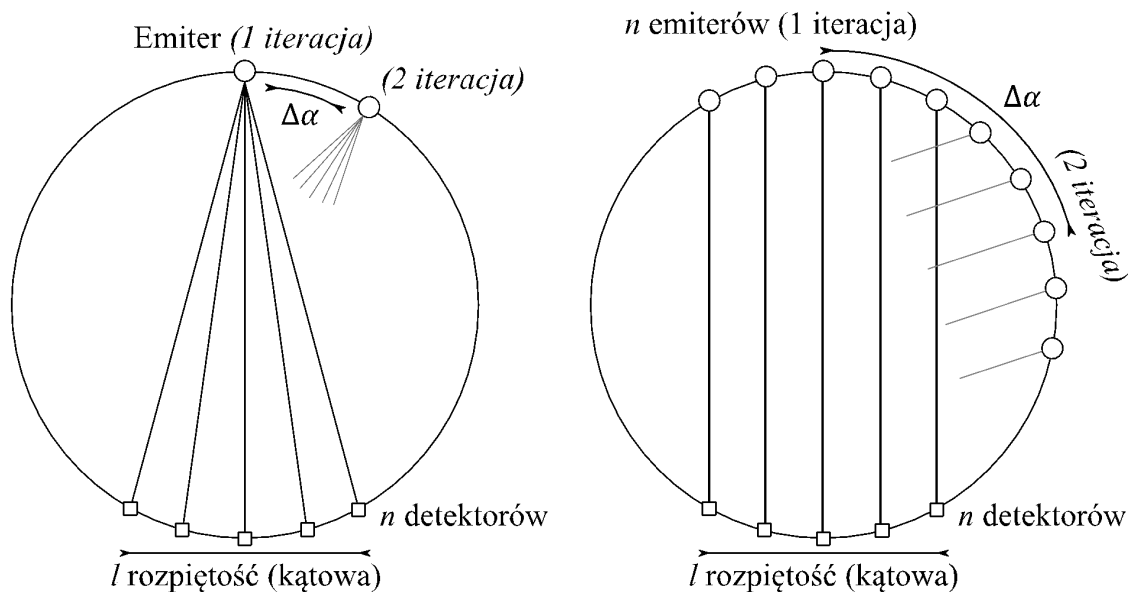
Słowa kluczowe: symulacja, wizualizacja, analiza danych, DICOM

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 (odwrotnej) transformaty Radona oraz filtrowania splotowego -- obliczenia muszą być wykonane samodzielnie. Ponadto nie wolno założyć stałej pozycji emitery(ów)/detektorów i symulować obrót dokonując rotacji obrazu. Ruch emitery(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 emitery(ów) i detektorów.
- Należy wykorzystać jeden z dwóch modeli emitery/detektor: stożkowy lub równoległy.
- Aplikacja powinna móc pozwolić konfigurować następujące elementy:
 - a) Krok $\Delta\alpha$ układu emitery/detektor.
 - b) Dla jednego układu emitery/detektor liczbę detektorów (n).
 - c) Rozwartość /rozpiętość układu emitery/detektor (l).



- Należy wykorzystać algorytm Bresenhama do linowego przejścia po kolejnych pikselach obrazu dyskretnego.
- Symulację pochłaniania promieniowania można zasymulować na jeden ze sposobów wybranych: 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):
 - podstawowych informacji o pacjencie
 - daty badania
 - komentarzy

Poprawność zapisanego pliku należy zweryfikować w dowolnej (darmowej) przeglądarce plików DICOM. **Uwaga:** Należy wykorzystać bibliotekę do obsługi plików DICOM, nie należy implementować obsługi plików DICOM samodzielnie!

Wymagania na 5.0

- Należy zastosować proste filtrowanie (splot) by 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:
 - błąd średniokwadratowy w funkcji iteracji (jego zmianę)
 - zmianę błędu średniokwadratowego przy zwiększaniu dokładności próbkowania (trzy uprzednio wymienione parametry modelu emiter/detektor)
 - zmianę błędu średniokwadratowy po włączeniu filtrowania

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>