

POMwJO

Sprawozdanie z projektu

Temat: Segmentacja śledziona TK

Autorzy: Joanna Wójcik, Anna Janiczek

Spis treści

1. Cel projektu.....	2
2. Analiza i projektowanie.....	3
Podstawy teoretyczne.....	3
Projektowanie.....	3
Struktury danych.....	3
Ograniczenia specyfikacji.....	3
Algorytmy.....	4
Wstępne przetwarzanie obrazów.....	4
Właściwa segmentacja.....	4
Nałożenie obrazów.....	5
3. Specyfikacja zewnętrzna.....	5
Opis programu.....	5
Obsługa programu.....	7
Format danych.....	7
4. Specyfikacja wewnętrzna.....	8
Klasa segmentacja.cs.....	8
Zmienne i obiekty.....	8
Metody.....	9
Klasa obraz.cs.....	9
Zmienne i obiekty.....	9
Zdarzenia.....	10
Klasa Form.cs.....	10
Zmienne i obiekty.....	10
Zdarzenia.....	10
5. Testowanie i wnioski.....	11

1 Cel projektu

Celem projektu było stworzenie aplikacji segmentującej śledzionę w obrazach z tomografii komputerowej. Aplikacja miała korzystać z darmowej biblioteki simpleItk, która udostępnia metody służące do przetwarzania i obróbki obrazu.

2 Analiza i projektowanie

2.1 Podstawy teoretyczne

Segmentacja śledziony to proces, który pozwala na wydzielenie śledziony spośród innych obiektów widocznych na obrazach tomografii komputerowej. Uzyskane obrazy mogą wspomóc diagnostykę różnych chorób związanych z śledzioną. Dzięki segmentacji zmiany patologiczne tego organu mogą być bardziej widoczne i w ten sposób wspomóc lekarzy podczas diagnozy.

W celu wydzielenia śledziony należy posłużyć się automatycznymi metodami, gdyż ręczne metody segmentacji są bardzo czasochłonne i są mało odporne na błędy. Automatyzacja procesu segmentacji śledziony pozwala na dużo szybsze uzyskanie wyników często bardziej odpornych na błędy niż w przypadku ręcznej segmentacji.

2.2 Projektowanie

2.2.1 Struktury danych

Obrazy na których przeprowadza się segmentacje są w formacie DICOM. Powstał on z myślą o normalizacji i ujednoliceniu informacji medycznych. Pliki w tym formacie oprócz samego obrazu zawierają strukturę znaczników tzw. tagów, które zawierają wiele informacji przydatnych podczas analizy obrazu i diagnostyki jak np. imię, nazwisko, wiek, płeć, datę urodzenia pacjenta, rodzaj badania, data badania. Standard DICOM może przechowywać dane jednowymiarowe w formie jednego obrazu bądź dane wielowymiarowe w postaci serii obrazów.

2.2.2 Ograniczenia specyfikacji

Program przeprowadza obliczenia jedynie dla plików w formacie DICOM z rozszerzeniem dcm.

2.3 Algorytmy

Program wczytuje i zapisuje dane w formacie DICOM z rozszerzeniem .dcm. Obrazy poddane procesowi segmentacji mogą być następnie wyświetlane oraz zapisane do folderu WYNIKI. Powyższe operacje są zrealizowane za pomocą biblioteki simpleITK. Proces segmentacji wątroby jest podzielony na trzy etapy:

- wstępne przetwarzanie obrazu
- właściwa segmentacja
- nałożenie obrazów

2.3.1 Wstępne przetwarzanie obrazów

Przed segmentacją obrazu zastosowano filtry mające na celu usunięcie szumów oraz wygładzenie obrazu.

1. Zastosowanie operacji zamknięcia morfologii binarnej w celu usunięcia zakłóceń intensywności w obiektach
2. Wygładzanie obrazu poprzez zastosowanie rekurencyjnego filtru Gaussa

2.3.2 Właściwa segmentacja

W celu wydzielenia z obrazu śledziony zastosowano szereg operacji w określonej kolejności:

1. Dylatacja obrazu za pomocą morfologii w skali szarości.

2. Funkcja, która oblicza dla każdego piksela funkcję sigmoidalną (przekształca zakres dynamiki intensywności obrazu wejściowego) w celu uwydatnienia przedziału intensywności dla śledziony.
3. Progowanie obrazu powyżej progu intensywności równego 180 i poniżej 100. Wyjściowe zakresy wartości obrazu ustawiono w przedziale od zera do jeden
4. Odejmowanie obrazu po progowaniu od od obrazu otrzymanego w punkcie trzecim
5. Progowanie obrazu otrzymanego w punkcie 4.
6. Binarna erozja obrazu za pomocą elementu strukturalnego w kształcie krzyża
7. Etykietowanie obrazu
8. Etykietowanie obrazu według malejącego obrazu, usunięcie obiektów mniejszych niż 5000 pikseli
9. Progowanie binarne obrazu etykiet powyżej i poniżej progu intensywności równego jeden.
10. Morfologia binarna- dylatacja obrazu dla wybranej wartości elementem strukturalnym w kształcie krzyża

2.3.3 Nałożenie obrazów

W tym etapie przetwarzania obrazu nałożono binarny obraz uzyskany w wyniku właściwej segmentacji na oryginalny obraz poprzez szereg operacji:

1. Obliczenie maksymalnej intensywności dla obrazu oryginalnego nie poddanego operacji segmentacji,
2. wymnożenie segmentowanego obrazu z maksymalną wartością intensywności obrazu oryginalnego,
3. implementacja funkcji, która zapisuje dla każdego w wyjściowym obrazie maksymalną wartość każdego piksela dla obrazu oryginalnego i segmentowanego,
4. przeskalowanie obrazu uzyskanego w punkcie trzecim do zakresu intensywności pomiędzy 0-255.

3 Specyfikacja zewnętrzna

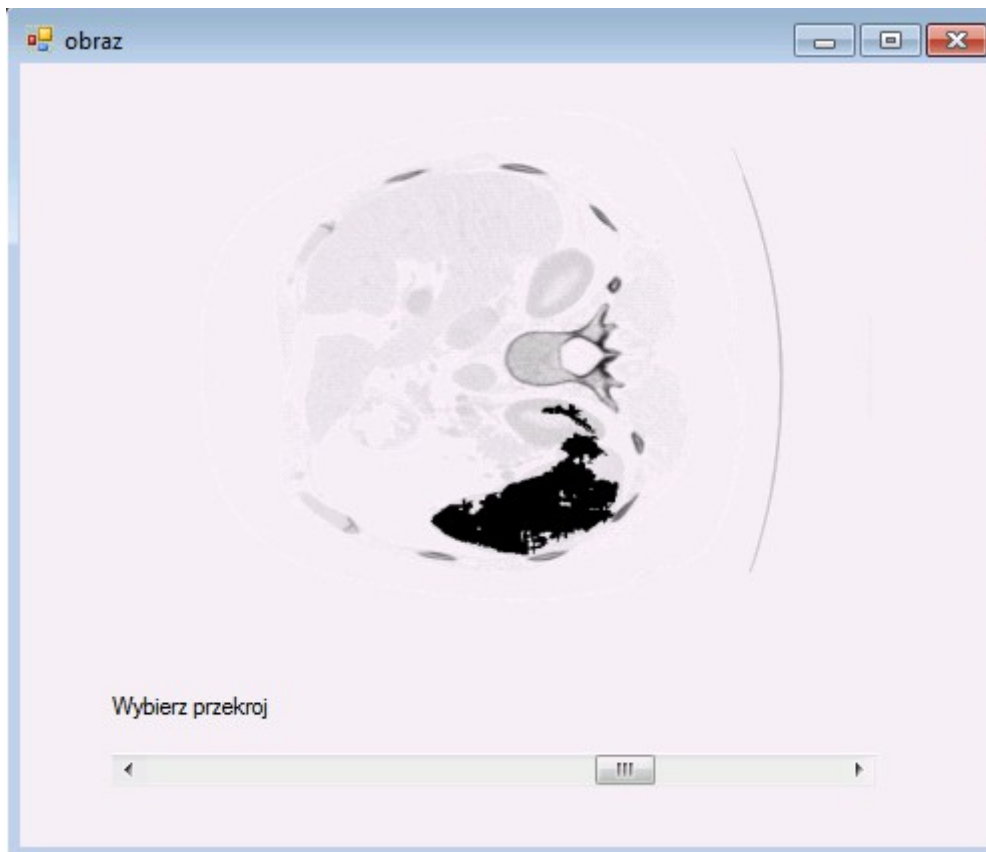
3.1 Opis programu



Rys 1: Okno główne programu

Główne okno aplikacji widoczne zaraz po uruchomieniu programu składa się z czterech przycisków, z których każdy odpowiada innemu procesowi:

- przycisk, za pomocą którego można wczytać dane z wybranego przez siebie folderu
- przycisk, który wywołuje funkcję zapisu danych do folderu WYNIKI znajdującego się w głównym folderze aplikacji
- przycisk, który wywołuje funkcję segmentacji obrazu
- przycisk, za pomocą którego można wyświetlić obrazy poddane segmentacji w nowym oknie



Rys 2: Okno wyświetlające obrazy

Drugie okno aplikacji, które pojawia się po naciśnięciu ostatniego przycisku pozwala na podgląd pliku poddanego segmentacji. Za pomocą slidera można zmieniać aktualnie wyświetlany przekrój obrazu

3.2 Obsługa programu

W celu wydzielenia ślędziona należy uruchomić program i wczytać obraz z tomografii komputerowej, który ma zostać poddany obróbce. Przed wykonaniem tego kroku pozostałe funkcje programu nie są możliwe do wywołania.



Rys 3: Okno główne programu przed wczytaniem pliku

Później należy wywołać segmentację obrazu. Po tym kroku są możliwe opcje podglądu i zapisu obrazu z wydzieloną śledzioną.

3.3 Format danych

Aplikacja odczytuje obrazy z plików DICOM z rozszerzeniem .dcm. Można wczytać pliki zarówno z pojedynczym plikiem jak i całym wolumenem.

Program umożliwia zapis pliku DICOM poddanego segmentacji w standardzie DICOM.

4 Specyfikacja wewnętrzna

4.1 Klasa segmentacja.cs

Klasa ta zawiera metody do segmentacji, zapisu, odczytu obrazu. W ramach tej klasy jest udostępniona funkcja, która konwertuje wybrany przekrój wolumenu pliku do obrazu w formacie Bitmap, który może być wyświetlany w oknie aplikacji. Klasa ta korzysta z biblioteki „simpleItk”.

4.1.1 Zmienne i obiekty

private Image obraz- obraz wejściowy,
private Image obrazWysegmentowany- obraz poddany segmentacji,
private Bitmap obrazBitmap- obraz w formacie bitmap,
private string nazwaObrazu– nazwa wczytanego pliku.

4.1.2 Metody

public Image wczytajObraz(string sciezka)- wczytuje wybrany obraz w formacie dcm,
public void zapiszObraz()- zapisuje plik obraz w formacie dcm,
public void zapiszObraz(Image obraz1)- zapisuje wybrany obraz w formacie dcm,
public Image segmentujObraz(int numerPrzekroj, int numerMaxPrzekroju=0)- zawiera algorytmy służące do wstępnego przetwarzania i segmentacji wybranych przekrojów obrazu,
public itk.simple.Image WstepnePrzetwarzanie(itk.simple.Image image)- zawiera metody służące do wstępnego przetwarzania obrazu,
public itk.simple.Image SegmetacjaSledziony(itk.simple.Image image)- zawiera właściwe algorytmy segmentacji śledziony,
public itk.simple.Image NalozObrazy(itk.simple.Image original, itk.simple.Image segmented)- nakładana oryginalny wczytany plik obraz poddany segmentacji,

public Bitmap konwertujObraz(Image obraz1, int przekroj = 0)- konwertuje wybrany przekrój do obrazu formatu bitmap,

public Image WybierzPrzekroj(Image obraz3d, int numerPrzekroju)

public Image WybierzPrzekroj(Image obraz3d, int przekrojMin, int PrzekrojMax)- zwraca obiekt typu `simple.itk.image` z przekrojami z obrazu `obraz3d` z przedziału pomiędzy `przekrojMin`-`PrzekrojMax`.

4.2 Klasa obraz.cs

Klasa ta służy do obsługi okna, który wyświetla wszystkie przekroje wolumenu wczytanego pliku, które zostały poddane operacji segmentacji. Posiada dwa konstruktory: jeden bezargumentowy, drugi z dwoma argumentami: maksymalną liczbą przekrojów oraz obrazem wejściowym, który ma być wyświetlony

4.2.1 Zmienne i obiekty

int numberOfSlice obiekt przechowujący numer aktualnie wyświetlonego przekroju wolumenu

itk.simple.Image imagedisplay aktualnie wyświetlany obraz

4.2.2 Zdarzenia

void przekroj_Scroll(object sender, ScrollEventArgs e)- wywoływane przy zmianie wartości suwaka. Umożliwia zmianę aktualnie wyświetlanego przekroju

4.3 Klasa Form.cs

Klasa ta służy do obsługi okna głównego aplikacji. Zawiera metody i zdarzenia inicjujące zapis, wczytanie, segmentacje plików.

4.3.1 Zmienne i obiekty

Stream StreamImage- ścieżka do obrazu wejściowego,
itk.simple.Image obrazWys- obraz wyjściowy poddany segmentacji,
public int numberOfSlice- liczba przekrojów pliku,
itk.simple.Image obraz- obraz wejściowy nie poddany segmentacji,
Segmentacja segm- obiekt klasy segmentacja.

4.3.2 Zdarzenia

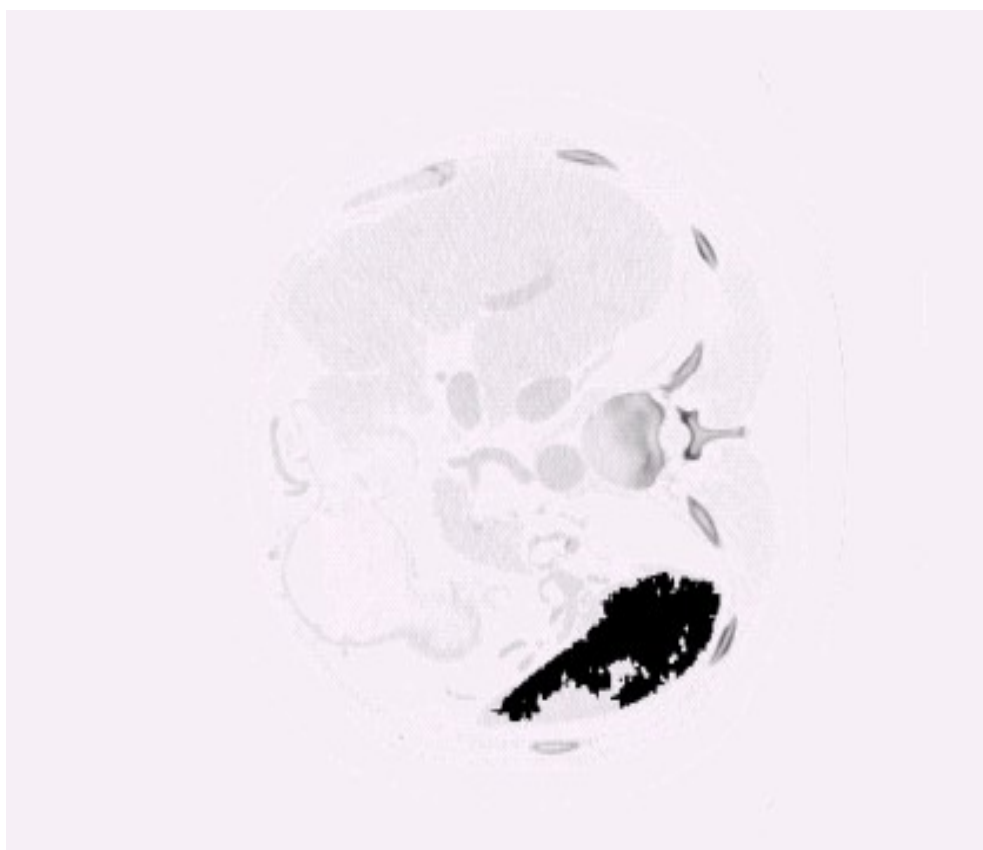
private void Podglad_Click(object sender, EventArgs e)- wywoływane po kliknięciu na przycisk Podgląd wyświetla się nowe okno z podglądem wysegmentowanego obrazu
private void segmentacja_Click(object sender, EventArgs e)- wywoływane po kliknięciu na przycisk segmentacja,
private void Zapisz_Click(object sender, EventArgs e)- wywoływane po kliknięciu na przycisk Zapisz,
private void OtworzPlik_Click(object sender, EventArgs e)- wywoływane po kliknięciu na przycisk Otwórz

5 Testowanie i wnioski

Aplikacja bardzo dobrze radzi sobie z wczytywaniem plików DICOM zawierające zarówno pojedyncze serie jak i wolumeny.

Interfejs aplikacji jest dość przejrzysty i prosty w obsłudze. Zaimplementowane algorytmy działają dość szybko dla pojedynczych przekrojów, niestety segmentacja wszystkich przekrojów wolumenu zajmuje trochę czasu. Aplikacja działa stabilnie, nie zauważono żadnych błędów podczas działania programu.

Program dość dobrze segmentuje śledzionę z wczytanych obrazów dla tomografii komputerowej.



Rys 4: Obraz z wysegmentowaną śledzioną

Dzięki wykorzystaniu biblioteki `simpleITK`, która zawiera wiele metod do przetwarzania i obróbki obrazów, możliwe było proste wygenerowanie wielu metod przetwarzania obrazu oraz metody segmentacji, zapisu i odczytu obrazów. Bez tej biblioteki ich implementacja zajęłaby zdecydowanie więcej czasu.

W celu prawidłowego wykorzystania biblioteki `simpleItk` należy zapoznać się z klasami i metodami udostępnianymi przez nią oraz zdobyć podstawową, niezbędną wiedzę na temat przetwarzania i obróbki obrazów.