(1/2) Wprowadzenie do przetwarzania obrazów DICOMowych w środowisku Matlab

I. Reprezentacja podstawowych formatów obrazów w Matlabie

1. Intensity image (grayscale image) - obraz w odcieniach szarości

Obraz jest reprezentowany jako macierz o rozmiarze `MxN`, gdzie każdy jej element poprzez swą wartość liczbową określa jasność piksela znajdującego się na danej pozycji. Istnieją dwa sposoby reprezentacji wartości liczbowej odpowiadającej jasności piksela: typ `double`, który przypisuje do każdego piksela zmiennoprzecinkową liczbę z przedziału `[0,1]`, przy czym `0` odpowiada kolorowi czarnemu, a `1` kolorowi białemu. W przypadku typów `uint8[0,255]/uint16[0,65535]/ int16[-32768,32767]` jasność piksela reprezentuje liczba całkowita z przedziału zależnego od typu danych. Wartość `intmin(class(I))` reprezentuje kolor czarny, natomiast wartość `intmax(class(I))` odpowiada kolorowi białemu (`I` - macierz obrazu). Klasa `uint8` potrzebuje ok. 1/8 pamięci w porównaniu do klasy `double`. Z drugiej strony, wiele funkcji matematycznych może być wywoływane tylko dla klasy `double`. Można jednak dokonywać konwersji, np.:

```
I=im2double(I);  % konwersja obrazu I do double
I=im2uint8(I);   % konwersja obrazu I do uint8
```

2. Binary image – obraz binarny

W formacie tym obraz również jest zapisany jako macierz `MxN` lecz poszczególne piksele mogą przyjmować jedną z dwóch wartości: `1` lub `0`. Wartość `0` przypisywana jest do koloru czarnego, natomiast `1` do koloru białego.

3. Indexed image – obraz indeksowany

W formacie tym obraz zapisywany jest w postaci dwóch macierzy. Pierwsza o rozmiarze `MxN` posiada taki sam rozmiar jak obraz, a poszczególne jej elementy reprezentują indeksy do drugiej macierzy (numery kolorów kolejnych pikseli obrazu). Druga macierz nazywana `colormap` (jej rozmiar przeważnie jest inny od rozmiaru obrazu) posiada trzy kolumny odpowiadające odcieniom trzech barw podstawowych i tyle wierszy *m*, ile kolorów znajduje się w palecie (rozmiar: *mx3*).

4. RGB image – obraz barwny RGB

Format ten reprezentuje obraz w postaci trzech macierzy i rozmiarach dopasowanych do formatu obrazu (macierz trójwymiarowa `MxNx3`). Każda z macierzy odpowiada jednej z barw podstawowych (**R**ed, **G**reen, **B**lue) i reprezentuje jasność składowych barw dla kolejnych pikseli obrazu.

5. Multiframe image

W niektórych zastosowaniach chcemy badać sekwencję obrazów. Jest to bardzo częste w przypadku obrazów medycznych oraz biologicznych, gdzie wizualizujemy np. kolejne przekroje badanej

struktury. W takich przypadkach format `multiframe` jest bardzo dogodnym sposobem pracy z sekwencją obrazów.

II. Struktura pliku obrazu zapisanego w formacie DICOM i podstawowe operacje na tego typu plikach w środowisku Matlab

W dużym uproszczeniu można przyjąć, że struktura pliku obrazu zapisanego w formacie *DICOM* składa się z dwóch części: `nagłówka` oraz `wartości pikseli obrazu`. Wszystkie informacje odnośnie danych obrazowych, danych pacjenta, badania, itd. zawarte są w nagłówku pliku, dlatego przed wyświetleniem pliku obrazu zapisanego w formacie *DICOM* należy zapoznać się z tymi danymi.

1. Odczytywanie meta danych

W *Matlabie* do odczytania meta danych plików *DICOM* służy funkcja `dicominfo()`, np.:

```
info = dicominfo('CT-MONO2-16-ankle.dcm')
```

Dostęp do poszczególnych pól struktury otrzymujemy za pomocą operatora kropki, np.:

```
info.Filename
```

Można również użyć funkcji `imageinfo()`, co spowoduje wyświetlenie osobnego okna z meta danymi wywoływanego pliku *DICOM*, np.:

```
imageinfo('CT-MONO2-16-ankle.dcm')
```

2. Wczytywanie obrazu

Do wczytania obrazu w formacie *DICOM* służy funkcja `dicomread()`, np.:

```
X = dicomread('CT-MONO2-16-ankle.dcm');
```

W przypadku obrazów indeksowanych (sprawdź w meta danych pliku pole `ColorType`) należy również wczytać mapę kolorów, np.:

```
[Y,map] = dicomread('US-PAL-8-10x-echo.dcm');
```

Jeśli pracujemy z wielostronicowym plikiem typu `multiframe`, wówczas w meta danych tego pliku pojawi się pole `NumberOfFrames` z informacją o ilości stron (obrazów) wyświetlanego pliku.

W przypadku pliku `US-PAL-8-10x-echo.dcm` po odczytaniu meta danych (sprawdź!) `NumberOfFrames=10`, zatem wczytany został plik typu `multiframe`, zawierający 10 stron (10 obrazów).

3. Wyświetlanie obrazu

Do wyświetlenia na ekranie pliku obrazu w formacie *DICOM* można użyć funkcji `imshow()` lub w przypadku obrazów typu `multiframe` funkcji `montage()`, np.:

```
imshow(X);
```

Sprawdź również działanie funkcji `imshow()` z parametrem []

Dlaczego znacząco zmienił się wyświetlany obraz? W jaki sposób działa dodatkowy parametr?

Wyświetl w jednym oknie (funkcja `subplot()`) dwa obrazy, korzystając z funkcji `imshow(X)` oraz `imshow(X,[])`. Przy pomocy narzędzia `imdisplayrange` sprawdź zakres wyświetlanych pikseli.

Odczytaj najmniejszą i największą wartość pikseli w tym obrazie przy pomocy funkcji `min()` i `max()`. Dane te również mogą być zawarte w meta danych pliku - pola: `SmallestImagePixelValue` oraz `LargestImagePixelValue`.

Zapoznaj się z funkcją `getrangefromclass()`

Sprawdź działanie narzędzia do zmiany kontrastu obrazu `imcontrast()`

Do wyświetlenia zawartości pliku `US-PAL-8-10x-echo.dcm` (plik typu `multiframe`) użyj funkcji `montage()`, np.:

```
montage(Y,map);
```

Sprawdź przy użyciu funkcji `montage()` w jaki sposób wyświetlić kolejne strony (obrazy) danego pliku w 2 kolumnach, a w jaki sposób wyświetlić kilka początkowych stron, np. o numerach 2-7?

Za pomocą funkcji `imshow()` wyświetl 5 obraz tego pliku.

Z pliku typu `multiframe` można również utworzyć klip filmowy za pomocą funkcji `immovie()` i uruchomić w odtwarzaczu `Movie Player`, np.:

```
mov = immovie(Y,map);
implay(mov);
```

Sprawdź również działanie funkcji `movie()` i spowoduj, aby klip filmowy wyświetlił się 2x.

Aby wyświetlić za pomocą funkcji `montage()` pliki dicomowe `brain_001.dcm – brain_020.dcm` należy je odpowiednio wczytać do workspace'a Matlaba, np. tak jak poniżej:

```
brain_info = dicominfo('brain_001.dcm')
brain_info.ColorType   % = grayscale – nie wczytujemy mapy kolorów
for i=1:20;
    filename = sprintf('brain_%03d.dcm', i);
    X(:,:,:,i) = dicomread(filename);
end
```

Bardziej dogodnym sposobem jest użycie funkcji `dir()` z odpowiednią maską (chcemy wyświetlić tylko pliki `brain_*.dcm` z danego folderu). Zapoznaj się z działaniem funkcji `dir()` i wczytaj odpowiednio pliki `brain_001.dcm – brain_020.dcm`, a następnie za pomocą funkcji `montage()` wyświetl je na ekranie w dwóch rzędach.

4. Zapisywanie obrazu do pliku DICOM

W odróżnieniu od specyfikacji formatu *DICOM* zawierającej definicję wielu obiektów informacyjnych (*IOD*), które mogą zostać utworzone (*CT*, *MR*, *USG*, *RTG*, itd.), *Matlab* przy użyciu funkcji `dicomwrite()` pozwala na „zweryfikowane" zapisanie obrazu wraz z niezbędnymi meta danymi do pliku w formacie *DICOM* dla trzech następujących obiektów informacyjnych (*IODs*):

1. *Secondary Capture Image Storage* (domyślnie)
2. *CT Image Storage*

3. *MR Image Storage*

Dla każdego z tych typów został określony zbiór niezbędnych meta danych wraz z dopuszczalnymi wartościami jakie mogą przyjmować pozostałe atrybuty. Z tego też względu, wykorzystując funkcję `dicomwrite()` do obiektów informacyjnych „wspieranych" przez *Matlaba* mamy pewność, że utworzony za pomocą tej funkcji plik będzie posiadał wszystkie niezbędne atrybuty, a ewentualnie brakujące zostaną utworzone. Dodatkowo funkcja `dicomwrite()` przypisuje domyślne wartości tam gdzie jest to możliwe. W ten sposób mamy gwarancję, że utworzony plik będzie zgodny ze specyfikacją *DICOM*.

Korzystając z helpa *Matlaba* zapoznaj się z funkcją `dicomwrite()` i zapisz kilka przykładowych obrazów do formatu *DICOM*. Jaka jest różnica w działaniu funkcji `dicomwrite()` na poniższych dwóch przykładach?

```
X = dicomread('CT-MONO2-16-ankle.dcm');
dicomwrite(X, 'SecondaryCapture.dcm');
```

```
metadata = dicominfo('CT-MONO2-16-ankle.dcm');
dicomwrite(X, 'CT.dcm', metadata, 'CreateMode', 'copy');
```

Czym różnią się pliki wynikowe `SecondaryCapture.dcm` oraz `CT.dcm` ?

5. Anonimizacja plików DICOMowych

W nagłówku pliku *DICOM* mogą znajdować się poufne informacje, takie jak dane pacjenta, opis badania, itp. Do usunięcia wszystkich wrażliwych danych można wykorzystać funkcję `dicomanon()`, np.:

```
dicomanon('CT-MONO2-16-ankle.dcm', 'anonymized.dcm')
```

Jaką postać przyjmie funkcja `dicomanon()` jeśli chcemy usunąć wybrane poufne dane?

6. Tworzenie nowej serii

Nową serię pliku *DICOM* tworzymy z reguły wtedy, gdy wprowadzamy pewne zmiany na obrazie. Zmodyfikuj zatem plik `CT-MONO2-16-ankle.dcm`, usuwając wszystkie etykiety tekstowe. Ponieważ tekst pisany jest kolorem białym, znajdź na obrazie maksymalną wartość pikseli, która odpowiadać będzie za kolor biały i minimalną wartość odpowiadającą za kolor czarny, a następnie podmień białe piksele z czarnymi (oczywiście operacja ta może mieć wpływ na wizualizowaną strukturę medyczną, ale na potrzeby tego ćwiczenia przyjmujemy, że pomijalnie mały). Następnie wygeneruj nowy unikalny numer serii za pomocą funkcji `dicomuid`:

```
uid = dicomuid
```

i wpisz tę wartość do pola `SeriesInstanceUID` w meta danych pliku `CT-MONO2-16-ankle.dcm`:

```
info_org.SeriesInstanceUID = uid;
```

następnie zapisz zmodyfikowany plik *DICOM* do nowego pliku, podając jako argument zmodyfikowaną strukturę meta danych:

```
dicomwrite(X,'ankle_newseries.dcm',info_org);
```

W ten sposób plik `ankle_newseries.dcm` stał się częścią nowej serii.

WSZYSTKIE ĆWICZENIA ZAPISZ W M-PLIKU - WYNIKI ZAPREZENTUJ PROWADZĄCEMU