

9 — Segmentacja obrazów

9.1 Cel

- zapoznanie z metodami segmentacji obrazów:
 - segmentacją przez rozrost,
 - segmentacją przez podział i scalanie,

9.2 Wstęp

W ramach dotychczas wykonanych ćwiczeń poznaliśmy segmentację z wykorzystaniem binaryzacji (progowania) – tj. na podstawie jasności (koloru) poszczególnych pikseli. Wykonaliśmy dwa warianty metody: globalny i lokalny oraz przetestowaliśmy różne podejścia do automatycznego wyznaczania progu bianryzacji: Otsu, Yen, Sauvola.

Ponadto poznaliśmy możliwość segmentacji na podstawie krawędzi z wykorzystaniem transformaty Hough'a.

W tym ćwiczeniu poznamy dwie inne metody podziału obrazu na fragmenty:

- segmentację przez rozrost obszaru (ang. *region growing*),
- segmentację przez podział i scalanie (ang. *split and merge*).

9.3 Podstawy

Niech R oznacza obszar równy całemu analizowanemu obrazowi. Segmentację możemy uznać za proces podziału R na n podobszarów R_1, R_2, \dots, R_n takich że:

- a. $\bigcup_{i=1}^n R_i = R$
- b. R_i – składa się z połączonych ze sobą pikseli,
- c. $R_i \cap R_j = \emptyset$ dla wszystkich i i $j, i \neq j$,
- d. $Q(R_i) = \text{TRUE}$ dla $i = 1, 2, \dots, n$
- e. $Q(R_i \cup R_j) = \text{FALSE}$ dla każdych sąsiednich R_i i R_j .

gdzie: symbole \cup i \cap oznaczają odpowiednio sumę i iloczyn zbiorów, a Q jest pewnym predykatem.

Punkt a. oznacza, że segmentacja musi być kompletna tj. każdy piksel powinien zostać przyporządkowany do jakiegoś zbioru. Punkt b. oznacza, że piksele w ramach jednego podobszaru muszą być ze sobą połączone (na zasadzie sąsiedztwa 4 lub 8 punktowego). Punkt c. oznacza, że dowolne różne podobszary muszą być rozłączne. Punkt d. oznacza, że wszystkie piksele będące w ramach jednego podobszaru muszą spełniać pewną własność. Przykładowo może to być ten sam lub podobny odcień szarości. Punkt e. oznacza, że dwa sąsiednie podobszary muszą być różne w sensie predykatu Q (inaczej powinny zostać uznane za ten sam podobszar).

9.4 Segmentacja przez rozrost obszaru

Pomysł jest następujący. Wybieramy (jak? – o tym później) piksele startowe (ang. *seed*) i od nich zaczynamy segmentację. Odbywa się ona na zasadzie sprawdzania czy sąsiednie piksele (sąsiedztwo 4 lub 8 punktowe) są podobne do centralnego pod względem jakieś cechy (predykatu

Q). Jeśli tak to oznaczane są jako należące do tej samej klasy co piksel centralny. Ponadto stają się one kolejnymi punktami startowymi metody. Zatem procedura ma charakter rekurencyjny.

Wybór punktów startowych może być podyktowany charakterem problemu (przykładowo wiemy gdzie na pewno zaczynają się obiekty). W ogólnym przypadku trzeba założyć, że pikselem startowym może być każdy piksel, co oczywiście wpływa na złożoność metody.

Kolejnym problemem jest wybór kryterium stopu tj. kiedy nasza procedura rekurencyjna ma się zakończyć. Dla danego podobszaru będzie to moment, kiedy nie istnieją już piksele, które można do niego dołączyć.

Warto w tym miejscu zwrócić uwagę, że stosowanie "sztywnego" warunku – np. różnica jasności pomiędzy pikselem centralnym, a analizowanym jest mniejsza niż 5 – może często dać niepożądane wyniki, gdyż nie uwzględnia pewnych globalnych właściwości. Przykładowo, może się okazać, że jeśli na obrazie występuje niewielki gradient to za należące do tego samego obszaru uznane zostaną piksele o zupełnie różnych jasnościach. Możliwa jest też sytuacja odwrotna. Duże zróżnicowanie wartości na obrazie spowoduje zbyt duże "poszarpanie" wykrytych obszarów.

Jednym z możliwych rozwiązań jest uzależnienie kryterium podobieństwa (predykatu Q) od własności obrazu np. średniej jasności w obrębie danego obszaru. Można również dodać inne kryteria np. kształt podobszaru itp.

Uwaga. Pojęcie segmentacja przez rozrost to pewna koncepcja podejścia do segmentacji, a nie konkretna metoda. Na etapie projektowania algorytmu należy skupić się na konstrukcji kryterium podobieństwa (tj. co i jak ma być ze sobą porównywane) oraz ewentualnym uzupełnianiu metody o dodatkowe kryteria.

Zadanie: zaprojektować system segmentacji wybranej struktury na obrazie MRI (np. stawu kolanowego). Punkt startowy wyznaczany będzie "ręcznie" (poprzez kliknięcie na obrazie).

1. Otwórz program Matlab. Ustal ścieżkę *Current Directory* na swój własny katalog roboczy.
Utwórz nowy m-plik (*New Script*) lub (*New->Script*). Na początku wykonaj polecenia
`close all; clearvars; clc;`
2. Wczytaj obraz *knee.png* – obraz MRI stawu kolanowego. Wyświetl go. Założmy, że chcemy dokonać segmentacji górnej kości. Przyjęliśmy, że punkt startowy metody wyznaczany będzie w sposób ręczny. Do pobrania położenia kurSORA myszy na ekranie służy funkcja *giinput*. Jej argumentem jest liczba punktów do pobrania. My potrzebujemy 1. Uwaga 1. Pobrane współrzędne należy zaokrąglić (*floor* lub *round*). Uwaga 2. Dla potrzeb testów dobrze jest wpisać punkt startowy na "sztywno". Pozwoli to uniknąć dość irytującego klikania przy każdym uruchomieniu skryptu. Uwaga 3. Proszę nie zapomnieć o konwersji obrazu na typ *double*.
3. Metodę zaimplementujemy z wykorzystaniem stosu. Uwaga. Podany poniżej opis jest tylko jedną z możliwych realizacji (niekoniecznie najlepszą). Na początek tworzymy dwie macierze o rozmiarach takich jak analizowany obraz. W jednej będziemy zapisywać odwiedzone lokalizacje (*visited*), a w drugiej rezultaty segmentacji (*segmented*). Tworzymy też stos tj. tablicę o np. 10000 wierszy i dwóch kolumnach (współrzędne) – *stack*. Do stosu dodajemy wskaźnik na element na górze (*iStack*). Obie macierze i stos tworzymy wypełnione zerami (funkcja *zeros*).
4. W pierwszym kroku metody na stos odkładamy współrzędne wybranego przez użytkownika piksela. Oznaczamy go również jako odwiedzony (macierz *visited*) i zaliczony do obiektu (macierz *segmented*).
5. Pozostałe działanie odbywać się będzie w pętli *while*, której warunkiem stopu jest obecność elementów na stosie. W iteracji należy pobrać współrzędne piksela ze stosu. Następnie sprawdzamy, czy dla tego piksela można określić kontekst o rozmiarze 3×3 tj. czy ma wszystkich sąsiadów. Uwaga. Przyjmujemy tutaj uproszczenie – nie segmentujemy brzegu obrazka (ramki o szerokości 1 piksela).

9.4 Segmentacja przez rozrost obszaru

55

6. W kolejnym kroku rozpisyjemy pętlę po otoczeniu 3×3 (`x2 for`). Wewnątrz obliczamy odległość pomiędzy pikselem centralnym, a każdym z kontekstu. Przyjmijmy, że będzie to moduł z różnicy jasności. Jeśli wartość modułu będzie mniejsza od zdefiniowanego progu (proszę przyjąć jako początkową wartość 4) oraz rozpatrywany piksel nie był wcześniej odwiedzany to oznaczamy go jako należący do obiektu oraz jego współrzędne "odkładamy" na stosie. Uwaga. Pierwsza część warunku logicznego to nasz predykat Q. Za piksele podobne uznajemy takie, których różnica w jasności jest mniejsza niż zadany próg. Niezależnie od wyniku testu oznaczamy piksel jako odwiedzony (żeby wielokrotnie nie analizować tych samych lokalizacji).
7. Poza pętlą `while` proszę wyświetlić rezultat segmentacji. Czy wyniki są poprawne? Proszę poeksperymentować z wartością progu.
8. Przykład ukazuje wspomniany wcześniej problem z "globalnym" podejściem do predykatu Q. Jeśli próg będzie mały, to wyznaczymy jedynie niewielki fragment kości. Natomiast zwiększenie progu skutkuje segmentacją nadmiarową. Mówiąc kolokwialnie, na obrazie znajdzie się "ścieżka" po której możliwe jest przejście od obszaru jasnego do ciemnego nie "łamiąc" progu odległości pomiędzy sąsiednimi pikselami.
9. Aby zaradzić powyższemu problemowi, można za kryterium podobieństwa przyjąć, nie różnicę jasności względem piksela centralnego, a od globalnie wyznaczonego i aktualizowanego progu. W najprostszym przypadku może to być średnia jasność w wyznaczonym obszarze. W celu implementacji mechanizmu wystarczy dodać dwie zmienne: średnią (mV) oraz licznik pikseli uznanych za należące do obiektu (nS). Przy każdym zdjęciu ze stosu licznik jest zwiększany o 1. Aktualizacja średniej następuje na podstawie równania:

$$mV_n = \frac{mV_{nS-1}(nS - 1) + I}{nS} \quad (9.1)$$

Następnie wystarczy tylko zmienić sposób obliczania odległości – zamienić piksel centralny na wartość średnią. Proszę spróbować jak działa metodą z taką modyfikacją. Proszę się liczyć z koniecznością zwiększenia progu (nawet dość znaczną).

10. Poprawić działanie metody może również dodanie filtracji uśredniającej np. filtrem Gaussa. Realizacja: `imfilter` i `fspecial`.
11. Możliwa jest również wizualizacja działania algorytmu "na bieżąco". W tym celu wynik należy wyświetlić na obrazie w pętli `while` np. `Icopy(segmented>0) = 255;` i wykorzystać polecenie `drawnow`. `Icopy` to kopia obrazu wejściowego. Oczywiście należy się liczyć ze znacznym spowolnieniem działania.