

[upel.agh.edu.pl](https://upel.agh.edu.pl)

# SW: Instrukcja - Segmentacja przez rozrost

6 — 8 minut

---

**Piotr Pawlik, Tomasz Kryjak**

## **Cel:**

- zapoznanie z segmentacją poprzez rozrost

Proces segmentacji możemy określić jako operację, która powoduje, że:

- każdy piksel obrazu zostaje przypisany do jednego ze zbiorów wynikowych
- powstają rejony stykających się ze sobą (sąsiadujących wg sąsiedztwa 4 lub 8-punktowego) pikseli
- każde dwa regiony są rozłączne (nie mają wspólnych pikseli)
- piksele w ramach jednego regionu są do siebie podobne względem jakiejś własności (jasność, kolor, tekstura)
- piksele sąsiadujących regionów różnią się względem własności z poprzedniego punktu (jasność, kolor, tekstura)

## **Segmentacja przez rozrost**

Algorytm segmentacji przez rozrost jest następujący:

1. wybieramy piksel startowy, oznaczamy go jako należący do

nowej klasy

2. Sprawdzamy, czy sąsiedzi, nie przypisani do żadnej klasy, są podobni do piksela startowego (wg wybranej własności/kryterium)
3. Sąsiedzi podobni są oznaczani jako należący do tej samej klasy co piksel startowy
4. Sąsiedzi podobni stają się nowymi pikselami startowymi
5. Jeżeli w obrazie pozostały piksele nieprzypisane do żadnej klasy to zaczynamy od punktu 1.

Powyższy algorytm ma dwa niejasno sprecyzowane miejsca. Po pierwsze - jak wybrać punkt startowy?, po drugie co to znaczy podobne?

Teoretyczna odpowiedź na pierwsze pytanie - dowolny piksel może stać się startowym (można go np. wylosować). Oznacza to, że metoda przez rozrost dla tego samego obrazu może dać różne wyniki w zależności od wyboru/wylosowania pikseli startowych. I tak już musi zostać :)

Natomiast odpowiedź drugie pytanie jest bardziej złożona. Można wybrać pewną stałą wartość dla danej klasy (np. wziętą z piksela startowego) i dołączać piksele różniące się od niej o z góry zadaną wartość. Można też liczyć nie względem piksela startowego, ale względem aktualnie rozpatrywanego. Wtedy wartość dla klasy nie jest stała - zmienia się z kolejno przyłączanymi pikselami. Można też jako wartość odniesienia przyjąć średnią z otoczenia piksela (podobnie jak robiliśmy przy binaryzacji lokalnej).

## **Implementacja segmentacji przez rozrost**

W tym ćwiczeniu nie dokonamy całkowitej segmentacji, lecz wysegmentujemy jeden region - podobny do piksela wybranego przez nas 'ręcznie'

1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na własny katalog. Utwórz nowy m-plik. Na początku wykonaj polecenia `close all; clear all;`
2. Wczytaj obraz "[knee.png](#)" (jest to obraz z rezonansu stawu kolanowego). Wyświetl go. Przyjmijmy, że chcemy wysegmentować górną kość. Jako piksel startowy będziemy przyjmować punkt 'kliknięty' na tym obrazie. Będzie do tego potrzebna funkcja `ginput`, której podaje się liczbę punktów do pobrania (u nas 1) a która zwraca współrzędne 'klikniętych' punktów. Uzyskane współrzędne należy zaokrąglić (`floor` lub `round`). UWAGA: przetwarzany obraz powinien być typu `double`.
3. W implementacji wykorzystamy stos. Jeżeli chcą Państwo zaimplementować stos wg własnego algorytmu - nie ma problemu (nawet zachęcam do tego). Jeżeli nie mają Państwo pomysłu jak to zrobić to ... stos można zaimplementować jako dużą tablicę (np. 1000 wierszy) o dwóch kolumnach - będziemy do niej wpisywać współrzędne odwiedzanych pikseli. Oprócz tego będziemy potrzebować wskazania na szczyt stosu. Poza stosem algorytm wymaga jeszcze dwóch tablic (obrazów o wymiarach równych obrazowi segmentowanemu) do zapamiętania odwiedzonych już pikseli (nazwijmy go `visited`) oraz do zapamiętania wyników segmentacji (nazwijmy go `segmented`). Wyzerujmy wszystkie trzy macierze (`zeros`)
4. Pierwszy krok to odłożenie na stos współrzędnych piksela startowego i wpisanie np. wartości 1 do `visited` i `segmented` pod te

współrzędne.

5. Następnie w pętli, dopóki stos się nie opróżni, należy :

1. pobrać współrzędne ze stosu,
2. dla wszystkich 8-miu sąsiadów sprawdzić, którzy jeszcze nie byli odwiedzeni (macierz visited) i dla takich policzyć miarę podobieństwa (niech to będzie moduł z różnicy jasności). W wypadku gdy miara dla sąsiada jest mniejsza od przyjętego progu (na początku niech wynosi on 4) oznaczamy tego sąsiada jako 'należący do obiektu' (1 w macierzy segmented) i jego współrzędne odkładamy na stos. Ponadto, niezależnie od wyniku sprawdzenia miary, sąsiad powinien zostać wpisany do macierzy visited.
6. Po pętli proszę wyświetlić obraz wynikowy (segmented).  
Poeksperymentuj z innymi niż 4 wartościami progu.
7. Przed przystąpieniem do dalszych działań dotychczasowe wyniki zaprezentuj prowadzącemu.
8. Powyższy algorytm reprezentuje podejście 'globalne' - stały próg dla całego regionu, co powoduje zbyt ograniczoną lub zbyt rozległą segmentację.
9. Teraz zmienimy kryterium podobieństwa, na podobne do stosowanego w binaryzacji - będziemy liczyć średnią kroczącą (ruchomą), ale tym razem z wartości wszystkich dotychczas dołączonych pikseli regionu - wg wzoru:
 
$$m_n = m_{n-1} \cdot (n-1) + I_n = \frac{m_{n-1} \cdot (n-1) + I_n}{n}$$
 gdzie **m** oznacza aktualną wartość średniej, **n** ilość punktów z których wylicza się średnią, a **I** - jasność w analizowanym punkcie. **n** zwiększa się o jeden z każdym punktem pobranym ze stosu. W

miarze podobieństwa dotychczasową wartość punktu centralnego zastępuje się przez wyliczoną średnią. Należy liczyć się z koniecznością znacznego zwiększenia progu celem uzyskania 'przyzwoitych' wyników.

10. Spróbuj poprawić efekt segmentacji poprzez zastosowanie filtracji usредniającej Gaussa (*imfilter + fspecial*)
11. Ciekawy efekt można osiągnąć wyświetlając 'na bieżąco' jak zmienia się wysegmentowany obraz. W tym celu stwórz kopię segmentowanego obrazu (np. nazwaną *lcopy*) i w pętli segmentacji co pewien czas (co kilka iteracji) wyświetl wynik operacji *lcopy* . \* *segmented* poleceniami ***imshow*** i ***drawnow***
12. Wynik zaprezentuj prowadzącemu.