

Diagnostyka obrazowa

Ćwiczenie siódme Usuwanie tła i segmentacja

1 Cel ćwiczenia

Ćwiczenie ma na celu zapoznanie uczestników kursu „Diagnostyka obrazowa” z metodami usuwania tła z obrazu oraz algorytmami segmentacji obrazu dostępnymi w programie ImageJ.

2 Po co usuwamy tło i przeprowadzamy segmentację obrazu?

Jednym z najważniejszych zagadnień analizy obrazu jest odseparowywanie wybranych części, obiektów od reszty obrazu. W przypadku analizy obrazu dla zagadnień biomedycznych, analiza wybranych komponentów obrazu jest szczególnie istotna. Możemy chcieć na przykład w prosty sposób wyizolować z serii zdjęć komórek w czasie, tylko te komórki, które są w fazie mitozy, ponieważ tylko te są dla nas interesujące.

Przez segmentację obrazu rozumiemy jego podział na pewne rozłączne elementy, które charakteryzują się wybranymi przez nas cechami (np. pewnym kolorem, specyficzną jasnością czy teksturą). Pamiętajmy, że efektem działania procedur segmentacji obrazów jest jedynie podział obrazu na części, a nie ich rozpoznawanie. W instrukcji zostanie przedstawiona idea działania podstawowych rodzajów algorytmów segmentacji, jak i wskazówki do wykorzystywania tych metod w programie *ImageJ*.

Usuwanie tła jest procesem związanym nieodłącznie z segmentacją obrazu, ponieważ aby poprawnie i z jak najlepszym rezultatem przeprowadzić segmentację, musimy doprowadzić do powstania jak najbardziej jednolitego tła. Kilka sposobów na rozwiązanie tych problemów z tłem, uniemożliwiających rozróżnienie obiektów od tła, było już przedstawione na poprzednich laboratoriach. W tej instrukcji, techniki te zostaną przypomniane i uzupełnione.

3 Usuwanie tła

W naszym rozumieniu usunięcie tła będzie krokiem obróbki wstępnej obrazu, przed zastosowaniem pewnego algorytmu segmentacji i ma służyć rozwiązaniu podstawowych problemów związanych z tłem, takich jak:

- różnice w intensywności pikseli tła w różnych częściach obrazu - należy wykonać korekcję jasności tła,
- problemy z saturacją pikseli - należy sprawić, aby obiekty różniły się od tła w sposób znaczący, tj. były reprezentowane przez piksele posiadające pewien stały zakres wartości.

W przypadku niejednorodnego tła, czy też nierównomiernego oświetlenia tła algorytmy segmentacji nie dadzą należytego efektu. Podstawowym sposobem na ujednoczenie tła w programie *ImageJ* jest wykorzystanie opcji `Background Subtraction` (**Rysunek ??**) z menu `Process`. Algorytm usuwania tła bazuje na tzw. algorytmie toczącej się kuli (*rolling ball algorithm*). Jego działanie możemy sobie wyobrazić jako kulę toczącą się pomiędzy obiektami utworzonymi przez wybrane piksele, w taki sposób, że wysokość obiektu odpowiada zaciemnieniu piksela. Obszar, po którym kula jest się w stanie przetoczyć, jest kwalifikowany jako tło obrazu. Parametr opisujący promień kuli: `Radius` powinien być przynajmniej tak duży jak promień największego obiektu na obrazie, który nie należy do tła. Odznaczenie opcji `Light Background` pozwala na analizowanie jasnych obiektów na ciemnym tle, natomiast opcja `Create Background` pozwala na uzyskanie jako obrazu wyjściowego samego tła. Tło jest automatycznie wygładzane w celu usunięcia szumów, jeżeli jednak nie chcemy go wygładzać wystarczy zaznaczyć opcję `Disable Smoothing`.

Kiedy takie usuwanie tła nie przyniesie nam dobrych rezultatów, można jeszcze zastosować technikę **binaryzacji adaptatywnej** (*Adaptive Thresholding*). Opiera się ona na różnicach w poziomach szarości na obrazie. Proces ten składa się z kilku kroków:

1. Wygładzenie obrazu za pomocą filtru Gaussa lub uśredniającego.
2. Odjęcie obrazu przefiltrowanego od oryginalnego.
3. Przeprowadzenie binaryzacji na tak powstałym obrazie.

4 Segmentacja

Segmentacja to jeden z najważniejszych etapów analizy obrazu. Polega ona na rozdzieleniu obrazu na zbiory, będące w ścisłym związku z obiektami przedstawionymi na obrazie. W przypadku segmentacji **kompletnej** obraz jest dzielony na rozłączne obszary odpowiadające pewnym obiektom, natomiast w przypadku segmentacji **częściowej** obraz jest podzielony na fragmenty związane ze zdefiniowanymi klasami obiektów. Zatem dokonanie segmentacji obrazu jest jednoznaczne z określeniem klasyfikatora, który będzie podporządkowywał obiekty do pewnych klas, w zależności od ich jasności, koloru, gradientu, kształtu, powierzchni czy tekstury. Dobór cech odzwierciedlanych przez piksele zależy również od tego czy chcemy segmentować obrazy kolorowe czy monochromatyczne. Efektem, który najczęściej staramy się uzyskać podczas segmentacji, jest wyodrębnienie obiektów z tła, w taki sposób aby tło stanowiło niezależny segment. Techniki segmentacji możemy podzielić na następujące kategorie:

- Przyporządkowanie punktów do kategorii.
 - Progowanie.
 - Klasteryzacja.
- Segmentacja na bazie algorytmów wykrywania krawędzi.
- Segmentacja obszarowa:

- Na podstawie rozrostu obszaru (angl. *region growing*).
 - Na podstawie podziału obszaru (angl. *region splitting*).
 - Techniki mieszane (rozrost i podział).
 - Technika powodziowa.
- Metody hybrydowe segmentacji (łączenie kilku technik segmentacji).

W kolejnych sekcjach bliżej przyjrzymy się algorytmom segmentacji wywodzącym się z wybranych kategorii.

4.1 Przyporządkowanie punktów do klas

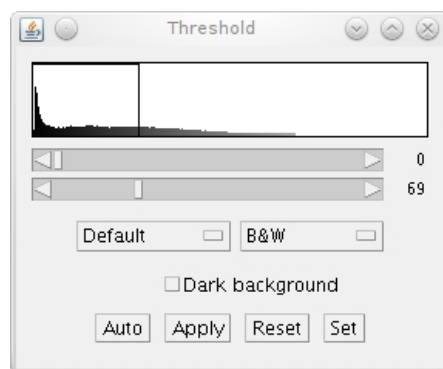
Podejście, w którym przyporządkowujemy punkty obrazu do jednej z wybranych kategorii, jest najprostszym i najbardziej intuicyjnym podejściem. Zakładając, że posiadamy dwie klasy obiektów na obrazie: obiekty klasy *A*, czyli tło oraz obiekty klasy *B*, czyli np. komórki, każdy z punktów obrazu na podstawie określonego kryterium kwalifikujemy do jednej z tych dwóch kategorii. Wadą metod tej klasy jest brak gwarancji spójności wynikowych obszarów. Przykładem metody bazującej na przyporządkowywaniu elementów do kategorii jest poznane już wcześniej *progowanie*.

4.1.1 Progowanie

Podstawową metodą segmentacji jest metoda, z którą mieliśmy już wcześniej do czynienia przy binaryzacji, a mianowicie **progowanie**. Nazwa metody wywodzi się od wyboru pewnego **progu** *P* w skali jasności obrazu, który dzieli punkty obrazu na dwie grupy: $f(n, m) > P$ oraz $f(n, m) \leq P$ gdzie $n \in 1..N$ i $m \in 1..M$ dla *N* i *M* będących wysokością i szerokością obrazu, odpowiednio. W kolejnych krokach algorytmu, piksele spełniające warunki progowania są zmieniane na podstawie wartości progów. Doboru progów dokonuje się zwykle na podstawie histogramu rozkładu jasności punktów na obrazie, w taki sposób, aby jak najlepiej rozdzielić punkty związane z tłem od punktów związanych z obiektami, którymi są np. komórki. Metodę progowania poznaliśmy już jako sposób przekształcania obrazu do postaci binarnej. Progowanie przeprowadzamy za pomocą okna `Threshold` w menu `Image->Adjust` (**Rysunek 1**). Domyślnie, próg jest ustalony w połowie, między największą, a najmniejszą wartością piksela zaobserwowaną na obrazie. Z listy rozwijanej w oknie `Threshold` można również wybrać inne algorytmy ustalania progów, niektóre z nich pozwalają na przykład na lepsze odseparowanie tła od reszty obrazu, w zależności od tego jaki rodzaj obiektów jest umieszczony na obrazie.

W przypadku obrazów kolorowych również możemy dokonać progowania wartości jasności pikseli. W takim przypadku, gdy z każdym punktem jest związany wektor wartości, a nie jak w przypadku obrazów monochromatycznych pojedyncza wartość piksela, stosuje się algorytmy obliczające najbliższe sąsiedztwo piksela w przestrzeni skonstruowanej dla składowych kolorów. W *ImageJ* również możemy przeprowadzać progowanie na bazie przestrzeni koloru. W menu `Image->Adjust` za pomocą polecenia `Threshold Color` możemy wykonać progowanie:

- W przestrzeni koloru RGB (przykład pokazany na **Rysunek 2**) - ustalanie progów dla czerwonego, zielonego i niebieskiego kanału.



Rysunek 1: Okno Threshold z domyślnym progiem binaryzacji.

- W przestrzeni koloru HSB - ustalanie progu dla kanału barwy, kanału określającego saturację pikseli oraz kanału określającego jasność.
- W przestrzeni koloru YUV - gdzie kanał Y odpowiada za jasność obrazu (luminancję), a pod UV reprezentowana jest barwa - dwie chrominancje.

W oknie `Threshold Color`, jeżeli zaznaczone jest polecenie `Pass` to wyświetlane są tylko piksele, których wartości mieszczą się w zakresie określanym przez dolny i górny próg na histogramie. W przeciwnym wypadku wyświetlane są wszystkie wartości poza tymi wewnątrz przedziału. Dodatkowo, mamy możliwość wyboru algorytmu progowania, który będzie najlepiej dostosowany do analizowanego obrazu. Opcję `Dark background` zaznaczamy w momencie jak analizowane przez nas obiekty są jaśniejsze niż tło.

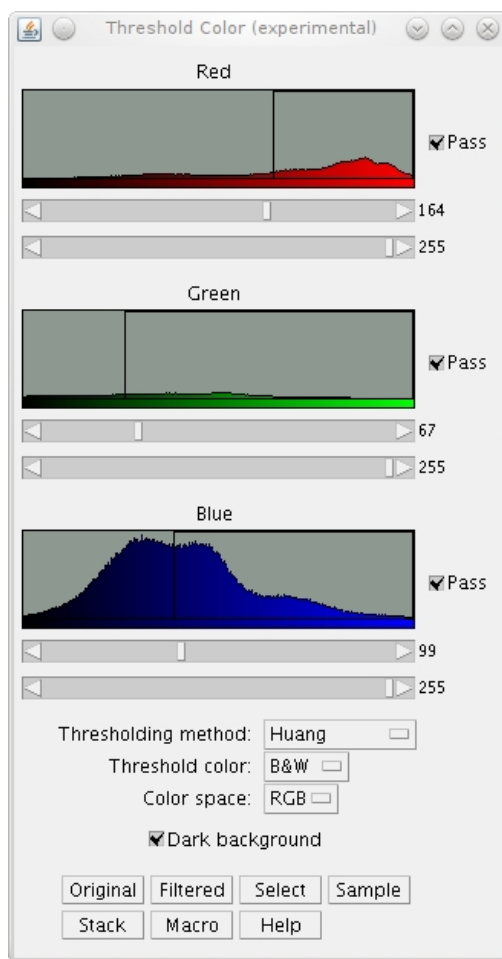
4.1.2 Klasteryzacja

Do zagadnienia problemu podziału obszarów na klasy można również podejść z wykorzystaniem metod statystycznych, wychodząc z jednego z dwóch założeń: że na obrazie istnieje ograniczona liczba klas, albo że na obrazie mamy do czynienia z liczbą klas nieograniczoną. W zależności od podejścia, albo przyporządkowujemy piksele do klas albo łączymy piksele o różnorodnych, ale podobnych właściwościach. W przypadku ograniczonej liczby klas określamy prawdopodobieństwo przynależności piksela do danej klasy, natomiast kiedy liczba klas jest nieznana, algorytm minimalizuje parametr rozmieszczenia pikseli wokół wartości średnich (np. algorytm K-Means).

4.2 Segmentacja na bazie algorytmów wykrywania krawędzi

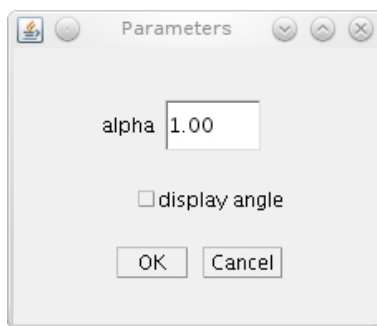
Do tego rodzaju metod zaliczamy również omówione już wcześniej filtry tj. Prewitta i Sobela. Dodatkowo można do tej kategorii zaliczyć filtry Canny'ego-Deriche, Wallisa (analiza jasności punktów w wybranych obszarach) i Kirsha (podobny do maski Sobela).

Filtr Canny'ego-Deriche aktywujemy przez zapisanie w katalogu `Plugins ImageJ` wtyczki z następującej lokalizacji: http://imagejdocu.tudor.lu/lib/exe/fetch.php?media=plugin:filter:edge_detection:image_edge.jar. Po restarcie programu w menu `Plugins->Image Edge` pojawi się opcja `Deriche`. W oknie filtru (Rysunek 3) określamy



Rysunek 2: Okno Threshold Color dla przestrzeni kolorów RGB.

parametr, który odpowiada za stopień wygładzenia wykrytych krawędzi. Im większa wartość parametru, tym mniejsze wygładzenie i bardziej dokładna detekcja.



Rysunek 3: Okno filtru krawędzi Canny'ego-Deriche.

4.3 Segmentacja obszarowa

Techniki segmentacji obszarowej muszą spełniać pewne określone warunki:

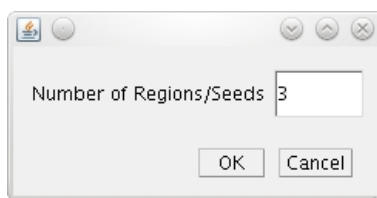
- Każdy punkt obrazu musi być zakwalifikowany do jakiegoś obszaru.
- Obszary będące efektem segmentacji muszą być spójne i rozłączne.
- Wszystkie punkty należące do tego samego obszaru powinny spełniać pewną określoną własność logiczną.
- Predykat logiczny określający przynależność punktu do danego obszaru nie może być prawdziwy dla sumy dwóch różnych, dowolnie wybranych obszarów.

W kolejnych sekcjach przybliżymy bardziej szczegółowo algorytmy należące do tej kategorii.

4.3.1 Segmentacja na podstawie rozrostu obszaru

W przypadku metod segmentacji bazujących na rozroście obszaru, algorytm jest inicjalizowany od punktów na obrazie zwanych ziarnami. Nowe punkty sąsiadujące z ziarnami są dołączane do nich w ramach rozrostu obszaru, jeżeli spełniają pewne z góry określone kryterium. Najtrudniejszym w tej metodzie jest odpowiedni dobór kryterium rozrostu obszaru. Podobieństwo punktów może być określane na zasadzie ich odległości w przestrzeni Euklidesowej czy na podstawie iloczynu skalarnego punktów. To najbardziej podstawowe kryteria.

Aby wykorzystać w *ImageJ* ten rodzaj segmentacji należy w katalogu Plugins zapisać plik znajdujący się pod podanym adresem http://svg.dmi.unict.it/iplab/imagej/Plugins/Segmentation/Region%20Growing/Region_Growing_Segmentation_files/Region_Growing_Segmentation.jar. W wybranym przez nas katalogu w menu Plugins pojawi się polecenie Region Growing Segmentation. Aby dokonać segmentacji obszaru na obrazie monochromatycznym należy jeszcze w oknie (**Rysunek 4**) wybrać punkty startowe (ziarno) algorytmu i wskazać je na obrazie.



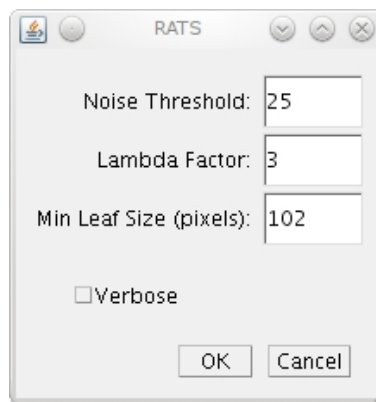
Rysunek 4: Okno algorytmu segmentacji rozrostu obszaru.

4.3.2 Segmentacja na podstawie podziału obszaru

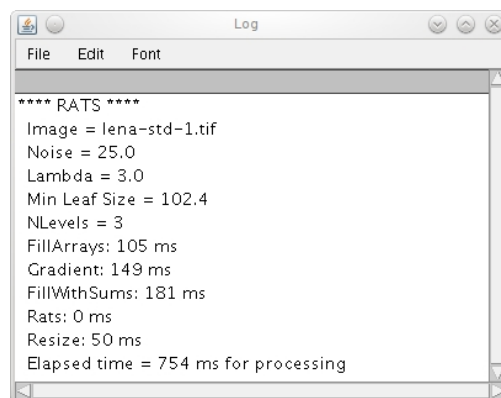
Segmentacja na podstawie podziału obszaru polega na kolejnych podziałach lub łączeniach obszarów obrazu (rekurencyjnych lub iteracyjnych), dla których piksele spełniają pewną własność (np. posiadają taki sam kolor). Przy czym należy pamiętać, że piksele wchodzące w skład już

wyznaczonych obszarów, znacznie różnią się pod względem tej własności od innych pikseli sąsiadujących, należących do innego obszaru.

Przykładem segmentacji na podstawie podziału obszaru może być dołączony jako wtyczka do *ImageJ* algorytm **RATS** (Robust Automatic Threshold Selection). Algorytm działa na zasadzie drzewa czwórkowego (angl. quad-tree decomposition), czyli sekwencyjnych podziałach obszarów kwadratowych na kolejne cztery kwadraty, o ile spełniony jest pewien warunek dla tego podziału. **RATS** można pobrać pod następującym adresem: http://rsbweb.nih.gov/ij/plugins/rats/RATS_.jar i należy go zapisać, jak w przypadku wszystkich wtyczek, w katalogu Plugins. Po uruchomieniu z menu Plugins polecenia RATS, w oknie algorytmu (Rysunek 5) możemy określić: jak duże może być zaszumienie tła obrazu (Noise Threshold), Lambda Factor określa jak duże mogą być dzielone obszary (im większy czynnik, tym większe obszary), natomiast za pomocą wartości parametru Min Leaf Size określamy jaką wielkość (w pikselach) może mieć najmniejszy podzielony obszar. Po aktywowaniu funkcji Verbose, po wykonaniu podziału obrazu na ekranie wyświetli się log z danymi dotyczącymi segmentacji (Rysunek 6), np. jak długo trwało wykonanie algorytmu.



Rysunek 5: Okno algorytmu segmentacji podziału obszaru algorytmem RATS.



Rysunek 6: Log danych dotyczących podziału przeprowadzonego algorytmem RATS.

4.4 Technika powodziowa

Technika *powodziowa* segmentacji, jak sama nazwa wskazuje, powinna się nam kojarzyć z zalewaniem obszarów reprezentowanych na obrazie wodą i wyobrażeniem efektu takiej operacji. Podstawowym i najbardziej znanym algorytmem należącym do tej kategorii jest *Watershed*, czyli po polsku *algorytm działów wodnych*.

4.4.1 Watershed

Do lepszego zrozumienia działania algorytmu, niezbędna jest znajomość pojęcia *gradientu morfologicznego*. Gradientem morfologicznym nazywamy operację będącą wynikiem różnicy obrazu po dylatacji i obrazu po erozji. Operacja gradientu pozwala m.in. na wykrywanie na obrazie krawędzi.

Idea segmentacji watershed polega na wyobrażeniu sobie zalewania wodą zbiornika. Dno zbiornika jest określone przez obrys przekroju obrazu po operacji gradientu morfologicznego. Woda, podnosząc się zalewa kolejne obszary obrazu, do momentu łączenia się niezależnych zbiorników. Gdy dwa zbiorniki mają się połączyć, w miejscu takim jest stawiana zapora i algorytm zatrzymuje się. Końcowym efektem działania algorytmu jest uzyskanie granic pomiędzy sąsiednimi obszarami. Jedną z zalet algorytmu jest możliwość rozdzielania obszarów stykających się, czy też częściowo ze sobą połączonych.

Segmentację watershed przeprowadzamy na obrazie w postaci binarnej. Aby wyznaczyć segmenty obrazu za pomocą programu *ImageJ* należy w menu `Process->Binary` wybrać polecenie `Watershed`.

5 Program ćwiczenia

1. Usuń tło przykładowego obrazu dostarczonego z programem *ImageJ Cell Colony* za pomocą opcji `Subtract Background`. Powtórz ten sam eksperyment z adaptacyjną metodą usuwania tła. Jaki efekty otrzymałeś/aś dla obu metod? Czy są one lepsze niż przy wykonaniu binaryzacji z domyślnym progiem w celu usunięcia tła?
2. Wykorzystując progowanie wartości kolorowych, postaraj się na przykładowym obrazie *ImageJ Fluorecent Cells* usunąć niebieskie jądra komórek. Tak samo spróbuj usunąć z obrazu fragmenty czerwone. Dla jakich wartości progów elementy niebieskie zostały najlepiej odseparowane na pierwszym obrazie? Dla jakich wartości progów elementy czerwone zostały najlepiej odseparowane na drugim obrazie?
3. Dla obrazu z punktu 1 przetestuj dwie wybrane metody segmentacji i Watershed. Jakie dają one rezultaty? Czy efekt jest tak samo dobry jeżeli zastosujemy metody segmentacji do obrazu, na którym tło nie zostało ujednoczone? Pokaż przykład.

Hanna Kamińska