

1. Utwórz nowy m-plik (*New Script*) lub (*New->Script*). Na początku wykonaj polecenia `close all; clearvars; clc;`
2. Wczytaj obraz *umbrella.png*. Wyświetl go. Załóżmy, że chcemy dokonać segmentacji poszczególnych fragmentów kolorowego parasola. Nasz algorytm opierać się będzie na podejściu podziału i łączenia. Jako kryterium podziału zastosujemy "jednorodność" danego obszaru, którą można opisać poprzez odchylenie standardowe. Przy scalaniu będziemy brać pod uwagę średni ocień koloru tj. łączyć podobszary o zbliżonym odcieniu.
3. W pierwszym etapie należy dokonać konwersji do przestrzeni barw HSV (funkcja `rgb2hsv`). Następnie wybrać tylko składową H (tj. odcień). Proszę nie zapomnieć o konwersji obrazka na typ `double`.
4. Procedurę podziału wygodnie jest zrealizować w formie rekurencji. Utwórz funkcję np. `split`, która jako argumenty przyjmować będzie obraz oraz cztery współrzędne analizowanego podobszaru. W pierwszym kroku należy obliczyć średnią i odchylenie standardowe dla rozważanego fragmentu obrazu (funkcje `mean` i `std`). Uwaga. Wcześniej obraz należy zamienić na wektor np. `I(:)`.
5. Następnie sprawdzamy czy odchylenie jest mniejsze niż zadany przez nas próg (np. 0.05) oraz czy nie osiągnęliśmy limitu podziału (np. bok kwadratu 8 pikseli). Oba potrzebne progi możemy zrealizować za pomocą zmiennych globalnych dostępnych w Matlabie. Deklaruje się je jako `global np, global sTh`. Powyższa linijka powinna się znaleźć w każdym pliku w którym używana jest zmienna. Potrzeby rozmiar kwadratu obliczany na podstawie współrzędnych (przekazanych do funkcji jako argumenty).
6. Jeśli podobszar nie jest jednorodny i nie osiągnęliśmy minimalnego rozmiaru podobszaru to dokonujemy podziału na cztery części. Wyznaczamy rozmiar aktualnego podobszaru (na podstawie jego współrzędnych). Z ich wykorzystaniem otrzymujemy cztery identyczne podobszary (*I1, I2, I3, I4*).
Następnie dla każdego z nich wywołujemy funkcję `split` – rekurencja. Najtrudniejsze jest odpowiednie podanie współrzędnych. Mają to być **rzeczywiste** współrzędne podobszaru we współrzędnych globalnych (obrazu w pełnej rozdzielczości). Podpowiedź. Trzeba "odpowiednio" wykorzystać współrzędne podobszaru przed podziałem (tj. argumenty funkcji) oraz rozmiar podobszaru. Należy zwrócić uwagę na to, aby nie wystąpił błąd przesunięcia o 1 tj. współrzędna była większa/mniejsza o 1 od rzeczywistej.
7. Jeśli podobszar jest jednorodny lub nie możemy już dalej prowadzić podziału to:
 - zapisujemy indeks danego podobszaru: `segRes(yS:yE, xS:xE) = index;`. `segRes` to globalna macierz o rozmiarze takim jak obraz wejściowy, a `yS, yE, xS, xE` to współrzędne podobszaru. W macierzy `segRes` zapisywane będą jednorodne obszary. Zmienna `index` jest globalnym licznikiem podobszarów. Powinna zostać zainicjowana wartością 1 przed pierwszym wywołaniem funkcji `split`. Następnie, każdorazowo po przypisaniu należy ją inkrementować.
 - zapisujemy średnią podobszaru. Wykorzystujemy macierz globalną `MRes` oraz kod podobny do opisanego powyżej. Wartości te wykorzystamy przy etapie scalania.

Uwaga. Podany sposób zapisu wyników podziału jest dość nieefektywny. Następuje powielenie tej samej informacji. Lepszym pomysłem byłoby wykorzystanie podejścia opartego o grafy, jednak jest ono trudniejsze do "szybkiej" implementacji.

8. Drugi etap to łączenie. Idea jest nieco zbliżona do segmentacji przez rozrost. Wybieramy dany podobszar i analizujemy sąsiednie podobszary. Jeśli są one podobne to dołączamy je do aktualnie rozpatrywanego. Za kryterium podobieństwa przyjmujemy niewielką różnicę w uśrednionym odcieniu barwy (składowa H).
9. Procedurę realizujemy wewnątrz pętli `while`. Warunkiem jej stopu jest przekroczenie przez licznik (np. `i`) wartości `index`, co oznacza, że przeanalizowane zostały wszystkie

znalezione w pierwszym etapie podobszary. W pierwszym kroku "wycinamy" maskę pikseli o rozpatrywanym indeksie tj. $IB = \text{segRes} == i$; . Następnie sprawdzamy, czy maska zawiera elementy niezerowe tj. czy nie jest "pusta". Taki przypadek może zajść jeśli pewien podobszar został dołączony do innego i zmienione zostały jego indeksy. Dla "pustej" maski inkrementujemy licznik i i przechodzimy do następnej iteracji – `continue`.

10. Dla rozpatrywanego podobszaru znajdujemy współrzędną lewego górnego rogu. Można to zrobić z wykorzystaniem składni: $[yF, xF] = \text{find}(IB, 1, 'first')$; – znajdowanie współrzędnych pierwszego niezerowego elementu.
11. Następnie należy znaleźć sąsiadów rozpatrywanego obszaru. Można wykorzystać dylatację maski IB (przypomnienie – `imdilate`) z elementem strukturalnym w postaci kwadratu o rozmiarze 3×3 `strel('square', 3)`. Następnie od maski po dylatacji odejmujemy maskę oryginalną. Otrzymujemy "ramkę", którą wykorzystujemy do "wycięcia" (mnożenie punktowe) fragmentu z macierzy `segRes`. Z tego fragmentu wybieramy elementy niezerowe (funkcja `nonzeros`) i unikalne (tj. eliminacja duplikatów) – `unique`.

12. Mając wektor z indeksami sąsiadów realizujemy pętlę w której dla każdego sąsiada:
 - wycinamy opowiadającą mu maskę i znajdujemy jej lewy górny róg – w sposób analogiczny do opisanego powyżej,
 - sprawdzamy, czy moduł z różnicy pomiędzy średnimi odcieniami barwy w dwóch analizowanych podobszarach jest mniejszy od progu (np. $5/255$). Jeśli tak to łączymy obszary tj. sąsiadowi przypisujemy indeks rozpatrywanego $\text{segRes}(IBS) = i$;, gdzie IBS – maska podobszaru sąsiedniego.

Do poprawnego działania konieczna jest jeszcze flaga informująca czy nastąpiło połączenie. Jej wartość ustawiamy na 0 przed pętlą po sąsiednich podobszarach. Jeśli wystąpiło połączenie to wartość ustawiamy na jeden.

Po pętli po sąsiadach sprawdzamy flagę. Jeśli ma ona wartość 0 tj. nie nastąpiło połączenie obszarów to licznik i jest inkrementowany.

13. Uwaga. Obliczenia mogą chwilę trwać. Można wykonać wizualizację działania (wyświetlenie wyników wewnątrz pętli), ale trzeba się liczyć ze spowolnieniem działania aplikacji.

14. Na koniec zastosujemy jeszcze dwie proste metody filtracji wyników. Po pierwsze wyeliminujemy obszary o rozmiarze mniejszym niż zadany (np. 100 pikseli). W tym celu wyznaczamy wektor unikalnych indeksów: $U = \text{unique}(\text{segRes})$; . Następnie implementujemy pętlę `for` po tych indeksach. Wewnątrz wycinamy maskę dla rozpatrywanego indeksu i i obliczamy jej pole (`sum`). Jeśli jest ono mniejsze niż próg to cały podobszar "zerujemy" – $\text{segRes}(C) = 0$;, gdzie C – maska podobszaru.

15. W drugim kroku przeprowadzamy przeindeksowanie indeksów na pierwsze N liczb całkowitych. Ponownie wyznaczamy unikalne indeksy. Następnie w pętli `for` wycinamy maskę o indeksie $U(ii)$, a do wyniku przypisujemy iterator pętli $\text{segRes}(C) = ii$;. Do wizualizacji segmentacji możemy wykorzystać funkcję `label2rgb`.

16. Proszę poeksperymentować z różnymi parametrami. Wyniki proszę zaprezentować prowadzącemu.