

Przetwarzanie obrazów

Zestaw zadań nr 6 i 7 – do rozwiązania zdalnie

---

T-L-6 Przetwarzanie obrazów kolorowych - filtracja, redukcja liczby kolorów – 4h

T-L-7 Histogram obrazu i operacje na histogramie – 2h

---

1. Wyświetlić histogramy obrazów kolorowych i w skali szarości.  
*Należy znaleźć i wykorzystać gotowe funkcje w Matlab. Dla obrazów kolorowych wyświetla się histogram dla każdego kanału w przestrzeni RGB osobno (3 histogramy), należy podpisać który wykres przedstawia jaki histogram.*
2. Wyświetlić histogramy kumulacyjne dla tych samych obrazów.  
*Znaleźć gotową funkcję lub napisać własną, wartość histogramu w każdym kolejnym kroku jest sumą kumulacyjną z aktualną wartością i z poprzedniego poziomu jasności.*
3. Dokonać binaryzacji obrazu z użyciem 3 globalnych progów binaryzacji określonych na sztywno i wyświetlić je obok siebie. Użyć starszej funkcji `im2bw(I,level)`.  
*W tym celu można pokusić się o napisanie własnej funkcji, która przeprowadzi binaryzację na zasadzie, jeśli wartość jasności znajduje się poniżej progu to ściągnąć do 0, jeśli powyżej to 1. Uwaga w Matlabie należy zdefiniować odpowiednio macierz obrazu, do której zostaną wpisane wartości binarne. Można użyć gotowej funkcji w Matlabie, podana wyżej jest przestarzała, ale dobrze sprawdzi się do aktualnego działania. Wartości progów wyznaczyć na zasadzie niski, średni i wysoki próg binaryzacji, w celu zobaczenia różnic i wyciągnięcia wniosków.*
4. Dokonać binaryzacji obrazu wyznaczając próg binaryzacji w dolinie histogramu (należy napisać funkcję znajdującą minimum lokalne wygładzonego histogramu lub realizującą algorytm BHT lub Otsu – do wyboru).  
*Doline wygładzonego histogramu można wyznaczyć „na piechotę”, patrząc na sam histogram. Wyrównanie histogramu powinno być na wykładzie, w razie czego algorytm jest dobrze opisany w sieci. Ewentualnie można skorzystać z gotowej funkcji, ale uwagę, ponieważ algorytm CLAHE jest do porównania w kolejnym zadaniu.*
5. Obraz w odcieniach szarości z wyrównanym histogramem porównać z obrazem poddanym funkcji CLAHE (Contrast-limited adaptive histogram equalization) w Matlabie. Dokonać binaryzacji i porównania wyników binaryzacji dla obrazów przed CLAHE i po użyciu tego algorytmu używając funkcji z Matlabu metodą Otsu oraz binaryzacją adaptacyjną.  
*Dla metody Otsu jest funkcja `graythresh`, dla metody adaptacyjnej jest funkcja `adaptthresh` lub można użyć funkcji `imbinarize` ustawiając odpowiednie parametry w celu realizacji powyższych zadań.*
6. Wykonać redukcję liczby poziomów szarości z 256 do: 128, 64, 32, 16, 8 metodą równomiernego podziału zakresu jasności (zerowanie najmłodszych bitów).  
*Należy zastanowić się, dlaczego zakres jasności w obrazie dla obrazów*

kolorowych lub w odcieniach szarości wynosi od 0 do 255, jaka jest to reprezentacja liczby binarnej. Przypomnieć, jak wygląda reprezentacja liczby w systemie binarnym, jaki wpływ na wartość liczby mają najmłodsze bity (skrajne po prawej stronie).

Zmniejszenie liczby poziomów najprościej można opisać jako usunięcie niektórych wartości poziomu jasności dla pikseli. Przy 128 poziomach w obrazie nie będzie pikseli o wartości nieparzystej, to znaczy piksele będą się zmieniać następująco:

$0 \rightarrow 0$   
 $1 \rightarrow 0$   
 $2 \rightarrow 2$   
 $3 \rightarrow 2$   
 $4 \rightarrow 4$   
 $5 \rightarrow 4$   
 $\vdots$   
 $254 \rightarrow 254$   
 $255 \rightarrow 254$

W celu najprostszej realizacji efekt ten zostanie osiągnięty poprzez zamianę wartości z dziesiętnej na binarną (dec2bin) i wykonanie operacji logicznej AND z liczbą binarną, która „wyzeruje” najmłodszy bit. Dla 128 poziomów należy użyć liczby binarnej w formie  $(1111\ 1110)_2$ , co spowoduje, że przy operacji AND starsze bity pozostaną bez zmian, a najmłodszy będzie miał wynik zawsze 0. Dla pozostałych poziomów należy dobrać odpowiednie zakresy liczby do operacji AND, Matlab pozwala na operację logiczną pomiędzy wartościami dziesiętnymi. Funkcja, którą wygodnie będzie użyć, to bitand.



Uwaga: Matlab umożliwi wykonanie całego zadania bez konieczności wykorzystywania w tym celu pętli, w skrypcie. Należy zastanowić się jak wykonać zadanie najoszczędniej w środkach.

- Wykonać analogicznie redukcję palety barw dla kolorowego obrazu 24-bitowego w taki sposób, aby każdy piksel był reprezentowany przez 16 bitów (7 bitów – kanał G, 5 bitów – kanał R, 4 bity – kanał B).

Zadanie analogiczne do redukcji z poprzedniego, z wyjątkiem tego, że redukcję wykonuje się dla poszczególnych kanałów w obrazie kolorowym, a wynik nie powinien znacząco odbiegać od oryginału na pierwszy rzut oka. Reprezentacja przez 7 bitów oznacza, że jest 7 bitów znaczących – zerowaniu ulega najmłodszy bit, natomiast 5 bitów znaczących powoduje, że zerowaniu ulegają 3 najmłodsze bity. Obraz do zadania jest dowolny (z dużą ilością szczegółów).

- Zaimplementować operacje erozji, dylatacji, otwarcia i zamknięcia dla obrazów binarnych wykorzystując przesuwany element strukturalny o rozmiarze  $3 \times 3$

piksele. Porównać wyniki z funkcjami zaimplementowanymi z Matlabie. Należy zaimplementować własną funkcję do realizacji powyższych zadań. Wyniki porównać do tych zaimplementowanych w Matlabie jak: *imerode*, *imdilate*, *imopen*, *imclose*. Proszę również pamiętać, że do tego zadania niezbędny jest obraz binarny. Przykładowe obrazy można znaleźć w plikach dla zespołu (wysłane zostaną w trakcie zajęć). Podczas pracy należy pamiętać, co jest Państwa elementem, a co tłem. Na obrazach piksele o wartości 1 traktujemy jako obiekt, a te równe 0 jako tło.

Element strukturalny oznacza, że badany obszar (w tym wypadku  $3 \times 3$ ) to piksele wybrane z obrazu oryginalnego. Taką maską należy przesuwając się po całym obrazie binarnym i analizować wartości pikseli w wybranym regionie elementem strukturalnym, przesuwając się co piksel. Uwaga: nie należy modyfikować obrazu oryginalnego, ponieważ będzie to wpływało na kolejne działania, wyniki należy zapisywać do nowego obrazu binarnego (na przykład do kopii oryginału). Dla erozji, ogólnie i upraszczając, jeśli w elemencie strukturalnym  $3 \times 3$  znajduje się jakakolwiek wartość 0, to piksel środkowy powinien mieć również wartość 0, to znaczy, jeśli jest to piksel przy krawędzi to należy zmienić jego wartość na 0. Jeśli w masce są same jedynki, to nie można go modyfikować. W tym celu można użyć analizy samej maski, wykonywać sumowanie elementów oraz robić operacje logiczne wybranych pikseli z maską. Tutaj nie narzucam Państwu sposobu realizacji tych funkcji. Operacje mogą być wykonywane tylko w skrypcie, może to być osobna funkcja napisana przez państwa.

Dla dylatacji - upraszczając - jeśli któryś piksel będzie miał wartość 1 w wybranym regionie to ten środkowy również musi mieć wartość 1. Otwarcie morfologiczne to dylatacja wyniku erozji, a zamknięcie morfologiczne to erozja wyniku dylatacji. Więcej szczegółów znajdą Państwo w książce: "Komputerowa analiza i przetwarzanie obrazów" autorstwa Przemysława Korohody i Ryszarda Tadeusiewicza. Książkę można znaleźć on-line w pdf.

9. Napisać funkcję realizującą morfologiczną detekcję krawędzi w obrazie binarnym. W tym podejściu przesuwając się elementem strukturalnym  $3 \times 3$  należy usunąć (wyzerować) wszystkie piksele, które są "otoczone obiektem" (jedynekami). Wynik należy zapisywać do nowego obrazu lub modyfikować kopię oryginału, tak aby aktualne działania nie wpływały na wynik następnych. Upraszczając, jeśli w wybranym elemencie strukturalnym są same piksele o wartości 1 na obrazie oryginalnym, to w wyniku piksel środkowy elementu strukturalnego powinien mieć wartość 0. W ten sposób na obrazie pozostaną tylko te piksele, które „stykają się” z tłem.
  10. Napisać funkcję realizującą morfologiczną detekcję pojedynczych punktów otoczonych tłem w obrazie binarnym. Postępować podobnie jak w zadaniu powyżej.
-