

Rozpoznawanie Obrazów

Semestr VII, Informatyka

mgr inż. Marcin Skobel

2021

Laboratorium nr 8: Detekcja punktów środkowych.

I. Zagadnienia teoretyczne

Wstęp

Wyznaczanie punktów środkowych obiektów w oparciu o operacje morfologiczne stanowi spore wyzwanie. Niemniej efekty tego typu działań mogą być niezwykle przydatne. Szczególnie jako wsparcie złożonych metod segmentacji obrazów cyfrowych.

Erozja warunkowa

Erozja warunkowa została zaproponowana do rozwiązania problemu rozdzielania połączonych jąder komórkowych na obrazach cyfrowych. Idea rozwiązania polega na utworzeniu dwóch typów elementów strukturalnych. Pierwszy typ elementu strukturalnego to duży element (np. wielkości 7x7 pikseli). Drugi typ elementu strukturalnego stanowi mniejszy obiekt o wielkości 3x3 piksele. Mniejsze elementy zwane drobnymi przyjmują następujące kształty:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

Natomiast elementy gruboziarniste przybierają taką formę:

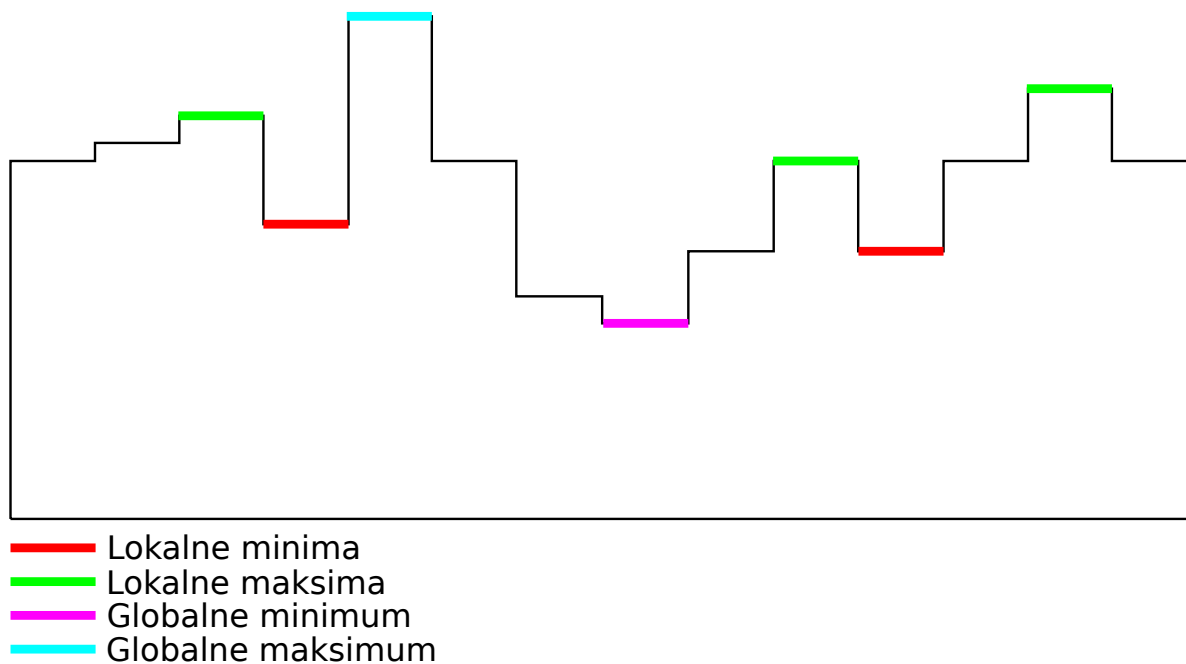
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

Podstawę do wykonania erozji warunkowej stanowią obrazy binarne. Oprócz danych w postaci binarnych obrazów wejściowych oraz elementów strukturalnych potrzebne będą jeszcze dwa progi erozji. Algorytm działania erozji warunkowej można zapisać słownie w kilku punktach:

1. Wczytanie obrazu binarnego, który należy poddać usuwaniu szumu (odrzuć małych obiektów).
2. Zdefiniowanie dwóch progów erozji $T1$ i $T2$. Pierwszy próg jest stanowi punkt zatrzymania operacji erozji przy użyciu gruboziarnistego elementu. Natomiast próg $T2$ stanowi informację przy jakiej wielkości elementów na obrazie ma się zakończyć erozja elementami drobnymi.
3. Erozja iteracyjna na obrazie binarnym z elementem gruboziarnistym, dopóki rozmiar obiektów będzie większy niż $T1$.
4. Erozja iteracyjna z drobnymi strukturami do wyniku uzyskany w kroku 2. Proces jest taki sam jak w kroku 2 z wyjątkiem, że próg wynosi $T2$ zamiast $T1$.

Transformacja H-Maxima

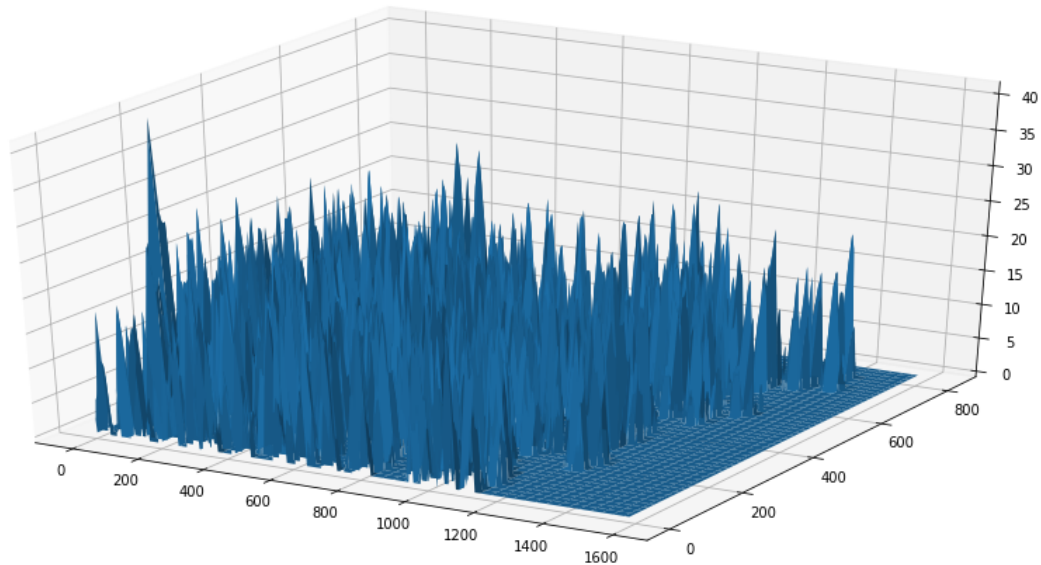
Transformacja H-Maxima wiąże się z pojęciem ekstremów globalnych i lokalnych. Ekstrema globalne występują pojedynczo, czyli jest jedno minimum globalne i jedno maksimum globalne. Inaczej wygląda sprawa minimum lokalnego, które może wystąpić wielokrotnie podobnie jak maksimum lokalne. Przykładowe ekstrema zostały zaprezentowane na poniższym rysunku:



Rysunek 1: Ekstrema lokalne i globalne

Na załączonym rysunku przedstawiono schemat występowania ekstremów w przypadku 1D. W przypadku obrazu ekstrema występują przestrzennie. Ekstrema obrazu cyfrowego możemy zatem porównywać do mapy topograficznej Rys. 2 składającej się z dolin oraz wzniesień. Transformacja H-maxima polega na znalezieniu obszarów najwyższej położonych na mapie, ale nie mniejszych niż ustalona wartość h . Obrazowo rzecz ujmując transformacja ścina najwyższe wzniesienia do poziomu h . To z kolei pozwala na znalezienie obszarów najwyższej położonych

jednocześnie czyli wyrównania poziomów maksimum lokalnych.



Rysunek 2: Mapa topograficzna obrazu testowego po transformacji Euklidesowej

Transformacja H-maxima występuje także jako H-minima, jedyną różnicą jest fakt, że w metodzie H-minima bazę do transformacji stanowią zagłębienia mapy. Proces transformacji H-maxima można zapisać w postaci kilku punktów:

1. Pracę rozpoczynamy od pozyskania obrazu binarnego.
2. W kolejnym kroku wykonujemy transformację odległościową, która spowoduje powstanie mapy topograficznej obrazu. Im piksel obiektu znajduje się dalej od tła tym wyższą wartość uzyska.
3. W kolejnym kroku wyznaczamy wartość h , która zostanie użyta do detekcji lokalnych maksimumów.
4. Na koniec wartości obszarów, które zostały sklasyfikowane jako lokalne ekstrema uzyskują wartość 255 lub jeden a pozostałe obszary uzyskują wartość 0.

II. Przykład praktyczny

Wstępna konfiguracja

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
import skimage.morphology
from google.colab import drive
drive.mount('/content/drive')
```

Wczytanie pliku

```
path = testImagePath = 'drive/My Drive/Colab Notebooks/Images/Test03.tif'
img = cv2.imread(path,0)
```

Obróbka wstępna obrazu

```
#progowanie (binaryzacja)
ret2, th2 = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
binary = 255-th2
#USUWANIE MALYCH OBIEKTOW
#odszukanie wszystkich pojedynczych elementow na obrazie
nb_comp, output, stats, ctr = cv2.connectedComponentsWithStats(binary, connectivity=8)
#connectedComponentsWithStats zwraca kazdy oddzielny komponent z informacjami
#o kazdym z nich, takimi jak rozmiar. Nastepna czesc po prostu usuwa tlo, ktore
#rowniez jest uwazane za komponent, ale w wiekszosci przypadkow tego nie chcemy.
sizes = stats[1:, -1]; nb_comp = nb_comp - 1
#elementy mniejsze od tego progu beda usuwane (wielkosc w pikselach)
min_size = 150
# deklaracja obrazu po usunięciu malych obiektow
DeleteSmall = np.zeros((output.shape))
#przegląd obiektow z pozostawieniem tylko najwiekszych
for i in range(0, nb_comp):
    if sizes[i] >= min_size:
        DeleteSmall[output == i + 1] = 255

#wyniki
plt.figure(figsize=(20, 15))
plt.subplot(211), plt.imshow(binary, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(212), plt.imshow(DeleteSmall, cmap = 'gray')
plt.title('Image after delete small objects'), plt.xticks([]), plt.yticks([])
plt.show()
```

Erozja warunkowa

```
#EROZJA WARUNKOWA
#elementy strukturalne
StructFineElement1 = np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]], np.uint8)
StructElementCoarse1 = np.array([[0, 0, 0, 1, 0, 0, 0], [0, 0, 1, 1, 1, 0, 0], [0, 1, 1, 1, 1, 1, 0],
                                  [0, 1, 1, 1, 1, 1, 0], [0, 0, 1, 1, 1, 0, 0],
                                  [0, 0, 0, 1, 0, 0, 0]], np.uint8)

#progi
T1 = 5000
T2 = 1500
ImageE = np.asarray(DeleteSmall, dtype="uint8")
ret, ImageE = cv2.threshold(ImageE, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
Erode = ImageE
maxsize = T1+100
while True:
    Erode = cv2.erode(Erode, StructElementCoarse1, iterations = 1)
    nb_comp, output, stats, ctr = cv2.connectedComponentsWithStats(Erode, connectivity=8)
    sizes = stats[1:, -1]; nb_comp = nb_comp - 1
    maxsize = max(sizes)
    if maxsize <= T1:
        break
while True:
```

```

Erode = cv2.erode(Erode, StructFineElement1, iterations = 1)
nb_comp, output, stats, ctr = cv2.connectedComponentsWithStats(Erode, connectivity=8)
sizes = stats[1:, -1]; nb_comp = nb_comp - 1
maxsize = max(sizes)
if maxsize <= T2:
    break
plt.figure(figsize=(20, 15))
plt.imshow(Erode, cmap = 'gray')

```

Transformacja H-maxima - krok po kroku

```

#TRANSFORMACJA H-MAXIMA I
#transformacja Euklidesowa
edt, inds = ndimage.distance_transform_edt(DeleteSmall, return_indices=True)
# prog dla lokalnych maksimow
LocalMaxThresHold = 20
LocalMaxCenters = edt
LocalMaxCenters[LocalMaxCenters>=LocalMaxThresHold] = 255
LocalMaxCenters[LocalMaxCenters<LocalMaxThresHold] = 0
plt.figure(figsize=(20, 15))
plt.imshow(LocalMaxCenters, cmap = 'gray')

```

Transformacja H-maxima - z gotowej funkcji

```

#TRANSFORMACJA H-MAXIMA II
#transformacja Euklidesowa
edt, inds = ndimage.distance_transform_edt(DeleteSmall, return_indices=True)
Img = skimage.morphology.h_maxima(edt, 20, selem=None)
plt.figure(figsize=(20, 15))
plt.imshow(Img, cmap = 'gray')

```

III. Uwagi

Do wykonania dzisiejszego przykładu warto pobrać plik testowy dostępny pod adresem:

<http://staff.uz.zgora.pl/mskobel/Test03.tif>

Plik z przykładami można pobrać ze strony: <http://staff.uz.zgora.pl/mskobel/lab8.py>

IV. Lista zadań

1. Na bazie skryptu z erozji warunkowej dokonaj modyfikacji która umożliwi wykonywanie erozji wszystkimi elementami strukturalnymi, które zostały przedstawione w części teoretycznej niniejszej listy.
2. Wykonaj transformację H-maxima z użyciem dystansu Czebyszewa (najlepiej z użyciem gotowej funkcji).
- 3.* Na bazie transformacji H-Maxima oraz metody odszukiwania elementów z zadania z erozją warunkową wykonaj algorytm automatycznie wyznaczający próg h w zależności od największej liczby obiektów na obrazie.