

# Rozpoznawanie Obrazów

Semestr VII, Informatyka

mgr inż. Marcin Skobel

2021

## Laboratorium nr 6: Operacje morfologiczne.

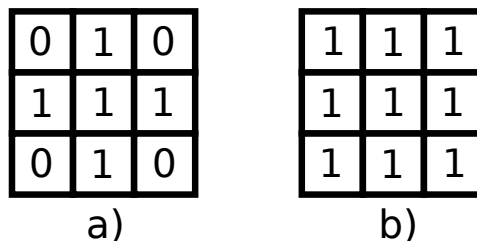
### I. Zagadnienia teoretyczne

#### Wstęp

Operacje morfologiczne są zazwyczaj stosowane do przekształcania obrazów binarnych, jednak można również wykonywać je na obrazach w odcieniach szarości. Na tym laboratorium skupimy się na przetwarzaniu obrazów binarnych, na których wartość zerowa oznacza tło obrazu przetwarzanego, natomiast elementy pierwszoplanowe oznaczone są wartościami 1. W celu ułatwienia sobie procesu wyświetlania obrazu w języku Python można wartość 1 zastąpić wartością 255. W bieżącym dokumencie wszystkie przykładowe rysunki mają odwróconą kolorystykę dla większej czytelności.

#### Elementy strukturalne

Na początku zaczniemy od zdefiniowania podstawowych elementów strukturalnych służących do operacji morfologicznych. Elementy strukturalne mogą powstać na bazie sąsiedztwa. W literaturze wyróżnia się dwa podstawowe typy sąsiedztwa: czterospójne (von Neumanna) oraz ośmiospójne (Moore'a). Dzięki zastosowaniu sąsiedztwa powstają dwa podstawowe elementy strukturalne również czterospójne oraz ośmiospójne. Elementy strukturalne mogą jednak przy-



Rysunek 1: Elementy strukturalne: a) czterospójne, b) ośmiospójne

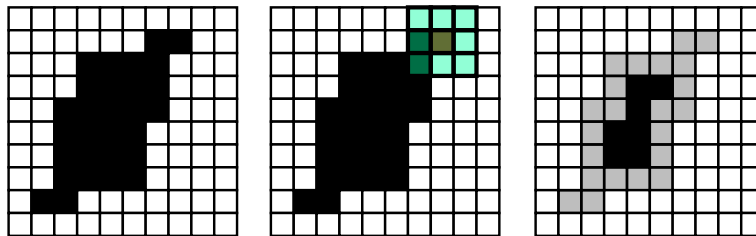
można mieć inne formy, na przykład posiadać jedynie wybrane sąsiedztwa lub mogą przyjmować kształty większych macierzy np.: 5x5, 7x7.

## Erozja

Erozja obrazów może się kojarzyć z pojęciami z zakresu geomorfologii związanymi z procesami powodującymi ubytek powierzchni terenu pod wpływem czynników zewnętrznych. Te skojarzenia są słuszne ponieważ proces erozji w przypadku obrazów cyfrowych polega na redukcji liczby skrajnych pikseli. Operacja erozji polega na przyłożeniu elementu strukturalnego do każdego z pikseli i jeżeli któremukolwiek elementowi o wartości 1 na obiekcie strukturalnym odpowiada wartość 0 na obrazie wówczas piksel obrazu znajdujący się w tej samej pozycji co środek elementu strukturalnego otrzymuje wartość 0. Erozja obrazu  $I$  przy użyciu elementu strukturalnego  $E$  wyraża się wzorem:

$$I \ominus E = x | E'_x \subset I \quad (1)$$

gdzie  $E'_x$  oznacza element przyłożenie elementu strukturalnego  $E'$  do piksela  $x$ .

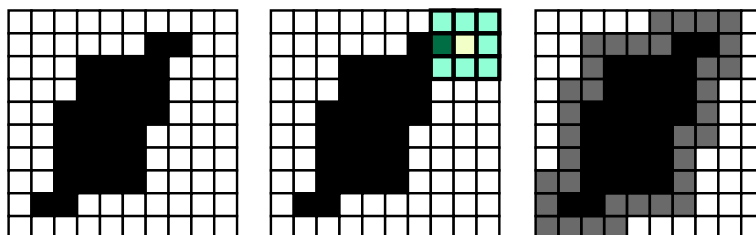


Rysunek 2: Przebieg operacji erozji

## Dylatacja

Operacja dylatacji stanowi przeciwieństwo erozji. Innymi słowy dylatacja powoduje nadbudowanie obiektu w zależności od przyłożonego elementu strukturalnego. Operacja dylatacji polega zatem na przyłożeniu elementu strukturalnego do każdego z pikseli i jeżeli któremukolwiek elementowi o wartości 1 na obiekcie strukturalnym odpowiada wartość 1 na obrazie wówczas piksel obrazu znajdujący się w tej samej pozycji co środek elementu strukturalnego otrzymuje wartość 1. Dylatacja obrazu  $I$  przy użyciu elementu strukturalnego  $E$  wyraża się wzorem:

$$I \oplus E = x | E'_x \subset I \quad (2)$$



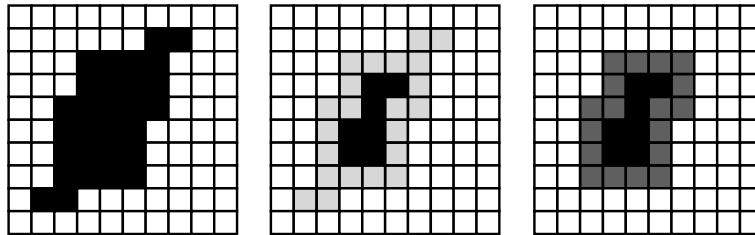
Rysunek 3: Przebieg operacji dylatacji

## Otwarcie

Oprócz podstawowych operacji morfologicznych istnieją również operacje złożone. Jedną z takich operacji jest otwarcie. Operacja ta jest definiowana jako nałożenie operacji dylatacji na obraz po erozji. Otwarcie możemy zatem zdefiniować przy użyciu wzoru:

$$I \circ E = (I \ominus E) \oplus E \quad (3)$$

Operacja otwarcie zachowuje zbliżony do pierwotnego kształt obiektu jednocześnie usuwając „odstające” elementy obiektu.



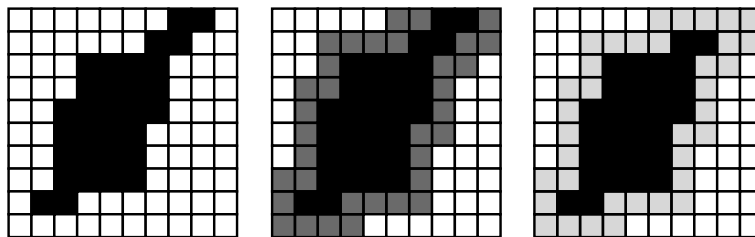
Rysunek 4: Przebieg operacji otwarcia

## Domknięcie

Druga ze złożonych operacji nazywana jest dopełnieniem. Operacja ta jest realizowana przez nałożenie operacji erozji na obraz po dylatacji. Otwarcie możemy zatem zdefiniować przy użyciu wzoru:

$$I \bullet E = (I \oplus E) \ominus E \quad (4)$$

Operacja domknięcia zachowuje zbliżony do pierwotnego kształt obiektu jednocześnie doklejając „odstające” elementy obiektu i reedukując element przy krawędzi obrazu.



Rysunek 5: Przebieg operacji domknięcia

## Znajdowanie konturu

Wyznaczanie konturu może zostać wykonane przy użyciu operacji morfologicznych. Pierwsza metoda polega na wyznaczeniu konturu zewnętrznego z użyciem operacji dylatacji:

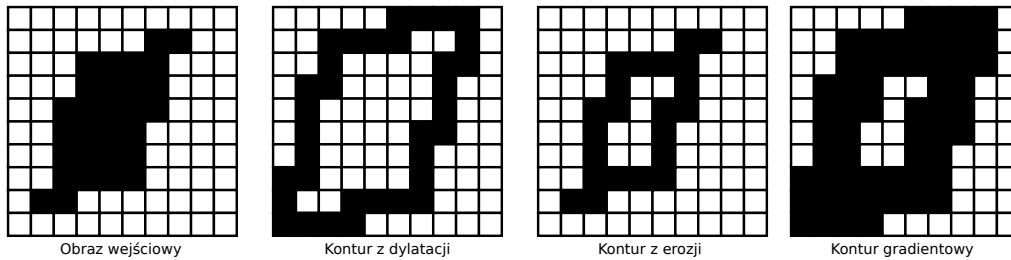
$$K(I) = (I \oplus E) - I \quad (5)$$

Wyznaczenie konturu przy użyciu operacji erozji można przeprowadzić zgodnie ze wzorem:

$$K(I) = I - (I \ominus E) \quad (6)$$

Można również użyć obu podstawowych operacji morfologicznych w celu uzyskania krawędzi metodą gradientu morfologicznego. Metodę można zrealizować przy użyciu wzoru:

$$G(I) = (I \oplus E) - (I \ominus E) \quad (7)$$



Rysunek 6: Kontury

### Szkieletyzacja

Zadanie szkieletyzacji wiąże się z fundamentalną transformacją morfologiczną o nazwie Hit-or-Miss. Transformacja Hit-or-Miss jest stosowana do wykrywania krawędzi oraz narożnych pikseli a także do operacji pogrubiania i pocieniania. Podstawowym elementem strukturalnym transformacji Hit-or-Miss jest macierz rozmiaru  $3 \times 3$  o poniższej konfiguracji zer i jedynek:

$$\begin{bmatrix} 1 \\ 0 & 1 & 1 \\ 0 & 0 \end{bmatrix} \quad (8)$$

W macierzy występują puste miejsca. W zależności od potrzeb miejsca te możemy wypełniać zerami lub jedynekami. Ponadto taki pojedynczy element strukturalny nie wykryje wszystkich punktów narożnych obiektu. Aby poradzić sobie z problemem wykrywania narożników wykonuje się kilkukrotną transformację Hit-or-Miss z elementem strukturalnym w pierwotnej wersji, a następnie obróconym o  $90^\circ$ ,  $180^\circ$  oraz  $270^\circ$ :

$$\begin{bmatrix} 1 \\ 0 & 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 0 & 1 & 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 1 & 1 & 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 & 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (9)$$

Transformacja Hit-or-Miss może również służyć do pogrubiania lub pocieniania. Pogrubienie polega na powiększeniu pikseli obiektu na obrazie i przypomina nieco dylatowanie. Operacja pogrubiania może być zrealizowana na podstawie wzoru:

$$T_{(I,E)} = I \cup T_{Hit-or-Miss (I,E)} \quad (10)$$

Operacja polega na przykładaniu elementu strukturalnego do kolejnych grup pikseli, gdzie analizowany piksel znajduje się w miejscu punktu centralnego obiektu strukturalnego, a następnie jeśli element strukturalny pasuje do otoczenia badanego punktu wartość piksela nie ulega zmianie a jeśli element strukturalny nie pasuje do otoczenia wówczas punkt uzyskuje wartość jeden. Bardzo podobnie wygląda operacja pocieniania:

$$T_{(I,E)} = I - T_{Hit-or-Miss (I,E)} \quad (11)$$

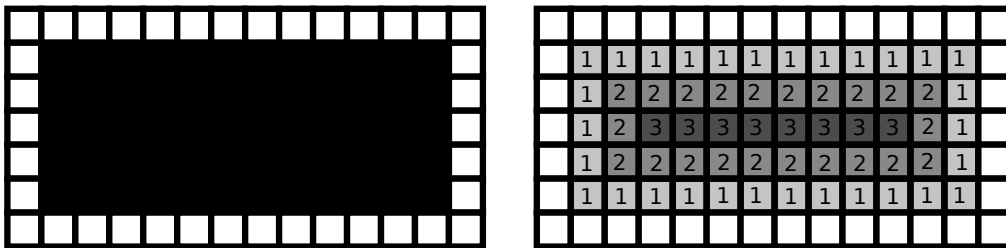
Operacja tym razem polega na przykładaniu elementu strukturalnego do kolejnych grup pikseli, gdzie analizowany piksel znajduje się w miejscu punktu centralnego obiektu strukturalnego, a następnie jeśli element strukturalny pasuje do otoczenia badanego punktu wartość piksela nie ulega zmianie a jeśli element strukturalny nie pasuje do otoczenia wówczas punkt uzyskuje wartość zero. Możemy teraz przejść do sedna sprawy czyli szkieletyzacji obiektów, która polega na wielokrotnym wykonaniu operacji pocieniania przy użyciu 8 elementów strukturalnych. Te 8 elementów uzyskujemy poprzez obrócenie dwóch elementów strukturalnych o  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  i  $270^\circ$ . Dwa podstawowe elementy mają następującą postać:

$$\begin{bmatrix} 1 \\ 0 & 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (12)$$

Druga metoda pozyskiwania szkieletu obiektu polega za zastosowaniu transformacji odległościowej. Przykładem metryki do służącej do obliczania transformacji odległościowej może być metryka euklidesowa, metryka Manhattan, metryka Czebyszewa. Najprostszym przykładem może być metryka euklidesowa określana w przestrzeni dwuwymiarowej wzorem:

$$d_{|AB|} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (13)$$

W wyniku transformacji odległościowej otrzymujemy piksele z wartościami określonymi jako odległość od obszaru tła a zatem przykładowo:



Rysunek 7: Transformacja odległościowa

Najodleglejsze od krawędzi piksele stanowią szkielet obiektu. Transformacje odległościowe mogą być przydatne również w innych obliczeniach na przykład w algorytmie segmentacji wododziałowej.

## II. Przykład praktyczny

### *Wstępna konfiguracja*

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

### *Wczytanie pliku*

```

path = testImagePath = 'drive/My Drive/Colab Notebooks/Images/Test.tif'
#flaga 0 oznacza, ze obrazek na etapie wczytywania jest konwertowany do skali szarosci
img = cv2.imread(path,0)
#do binaryzacji uzyjemy metody Otsu
ret, imgT = cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESHOTSU)
#jesli obiekty wynikowe sa czarne nalezy odwrocic kolory
imgT = 255-imgT
#w efekcie dostajemy taki oto obraz
plt.imshow(imgT, cmap="gray")

```

### ***Element strukturalny***

```

#deklaracja elementu strukturalnego (nie ma znaczenia czy wpisujemy 1 czy 255)
kernel = np.array([[0,1,0],[1,1,1],[0,1,0]],np.uint8)

```

### ***Erozja***

```

erosion = cv2.erode(imgT, kernel, iterations = 1)
plt.imshow(erosion, cmap="gray")

```

### ***Dylatacja***

```

dilation = cv2.dilate(imgT, kernel, iterations = 1)
plt.imshow(dilation, cmap="gray")

```

### ***Otwarcie***

```

opening = cv2.morphologyEx(imgT, cv2.MORPH_OPEN, kernel)
plt.imshow(opening, cmap="gray")

```

### ***Domknięcie***

```

closing = cv2.morphologyEx(imgT, cv2.MORPH_CLOSE, kernel)
plt.imshow(closing, cmap="gray")

```

### ***Znajdowanie konturu - Gradient morfologiczny***

```

gradient = cv2.morphologyEx(imgT, cv2.MORPH_GRADIENT, kernel)
plt.imshow(gradient, cmap="gray")

```

### ***Szkieletyzacja***

```

thinned = cv2.ximgproc.thinning(imgT)
plt.imshow(thinned, cmap="gray")

```

### ***Transformacja odległościowa***

```

distTrans = cv2.distanceTransform(imgT,1,0)
plt.imshow(distTrans, cmap="gray")

```

## **III. Uwagi**

Lista laboratoryjna może być przydatna i kompatybilna z zadaniami projektowymi. Tym razem używamy gotowych funkcji w bibliotece OpenCV. Funkcja `cv2.ximgproc.thinning` jest stosunkowo nowym elementem biblioteki openCV i nazwa może być trochę myląca bo oznacza pocienianie, które jest używane do szkieletyzacji zaś sama funkcja zgodnie z literaturą powinna się raczej nazywać skeletonization. Plik z przykładami można pobrać ze strony: <http://staff.uz.zgora.pl/mskobel/lab6.py>

## IV. Lista zadań

1. Użyj narzędzia erozji w celu utworzenia własnego algorytmu wykonującego erozję ze zmiennym elementem strukturalnym, który ma przybierać formę `np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]], np.uint8)` na zmianę z `np.array([[1, 1, 1], [1, 1, 1], [1, 1, 1]], np.uint8)`
2. Użyj transformacji odległościowej na dowolnym obrazie binarnym. Następnie odszukaj wartość maksymalną na obrazie po transformacji. Zamień wszystkie wartości większe lub równe (maks-20) na 255 a pozostałe wartości na 0. Wyświetl wynik.
3. Wykonaj na dowolnym obrazie wejściowym gradient morfologiczny. Następnie wykonaj kontur przy użyciu erozji oraz drugi kontur przy użyciu dylatacji. Połącz kontury z erozji i dylatacji w jeden kontur i porównaj go z konturem uzyskanym z gradientu morfologicznego. Jak wygląda obraz wynikowy? Czy zgadza się to z teorią?