

ЛАБОРАТОРНА РОБОТА №5

Морфологічна обробка зображень

Тема роботи: морфологічна обробка зображень.

Мета роботи: дослідити та застосувати основні методи морфологічної обробки зображень в середовищі MATLAB або Python.

Теоретичні відомості

Базові поняття теорії множин, що застосовуються в морфологічній обробці зображень

Базові поняття теорії множин, які застосовуються в морфологічній обробці зображень проілюстровано нижче на рис. 5.1

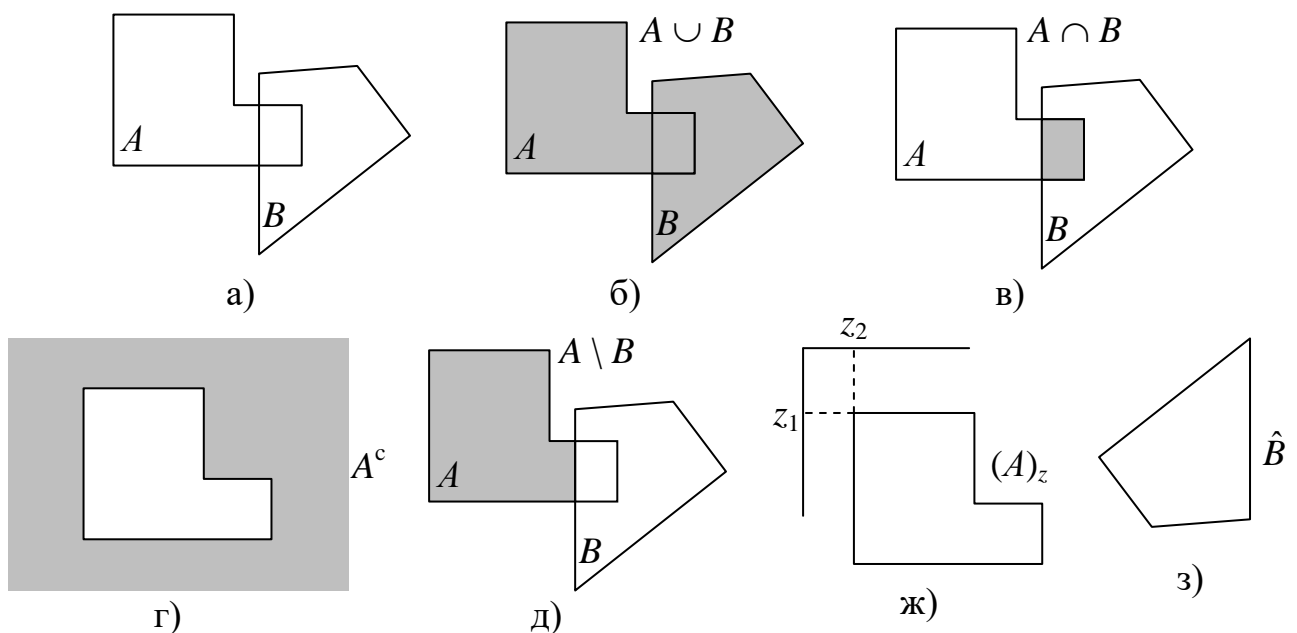


Рис. 5.1. Базові поняття теорії множин:

- а) дві множини A та B ;
- б) об'єднання множин A та B : $C = A \cup B$;
- в) перетин множин A та B : $C = A \cap B$;
- г) доповнення множини A : $A^c = \{w \mid w \notin A\}$;
- д) різниця множин A та B : $A \setminus B = \{w \mid w \in A, w \notin B\}$;
- ж) зсув множини A : $(A)_z = \{c \mid c = a + z, a \in A\}$;
- з) центральне віддзеркалення множини B : $\hat{B} = \{w \mid w = -b, b \in B\}$.

Дилатація та ерозія

Нехай A та B – множини з простору Z^2 .

Дилатація множини A по множині B (або відносно B) позначається $A \oplus B$ і визначається як:

$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset\}.$$

Дилатація множини A по B – це множина всіх таких зсувів z , за яких множини B і A збігаються щонайменше в одному елементі.

Ерозія A по B , що позначається $A \ominus B$, визначається як

$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\}.$$

Інакше кажучи, ерозія множини A по примітиву B – це множина всіх таких точок z , при зсуві в які множина B цілком міститься в A .

Півтонова дилатація зображення f по структуроутворюючому елементу b позначається $f \oplus b$ і визначається формулою:

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\},$$

де D_b позначає область визначення b , і передбачається, що $f(x, y)$ дорівнює $-\infty$ поза областю визначення f . Це рівняння реалізує процес, аналогічний процедурі знаходження просторової згортки.

Півтонова ерозія зображення f по структуроутворюючому елементу b позначається $f \ominus b$ і визначається як:

$$(f \ominus b)(x, y) = \min\{f(x - x', y - y') - b(x', y') \mid (x', y') \in D_b\},$$

де D_b позначає область визначення b , і передбачається, що $f(x, y)$ дорівнює $+\infty$ поза областю визначення f .

Процедури ерозії та дилатації в системі МАТЛАВ реалізовані функціями `imerode` та `imdilate` відповідно. Вони виконують звичайні чи півтонові ерозію та дилатацію, в залежності від класу (типу даних) аргументу.

У Python для виконання морфологічних операцій рекомендовано використовувати пакет `skimage.morphology`. В даному пакеті ерозія та

дилатація виконуються функціями `binary_erosion`, `binary_dilation` для двійкових зображень та `erosion` і `dilation` для півтонових зображень.

Замикання та розмикання

Розмикання множини A по примітиву B позначається $A \circ B$ і визначається рівністю:

$$A \circ B = (A \ominus B) \oplus B.$$

Таким чином, розмикання множини A по примітиву B будується як ерозія A по B , результат якої потім піддається дилатації по тому ж примітиву B .

Замикання множини A по примітиву B позначається як $A \bullet B$ і визначається наступним чином:

$$A \bullet B = (A \oplus B) \ominus B,$$

тобто, як дилатація A по B , результат якої потім піддається ерозії по тому ж примітиву B .

Півтонові операції замикання та розмикання аналогічні звичайним відповідним їм операціям, за виключенням того, що тут застосовуються півтонові ерозія та дилатація.

Процедури розмикання та замикання в системі MATLAB реалізовані функціями `imopen` та `imclose` відповідно. Вони виконують звичайні чи півтонові розмикання та замикання, в залежності від класу (типу даних) аргументу. У Python розмикання та замикання зображень реалізовані функціями `binary_opening` і `opening` та `binary_closing` і `closing` відповідно.

Перетворення успіх невдача

Перетворення успіх/невдача множини A по множині B позначається через $A \otimes B$. Тут B позначає структуротворну пару елементів $B = (B_1, B_2)$, а не один елемент, як це було раніше. За визначенням, перетворенням множини A по B називається множина:

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2).$$

Аналіз компонент зв'язності

Піксел p з координатами (x, y) може мати 4 або 8 сусідніх пікселів, які позначаються як $N_4(p)$, $N_D(p)$ та $N_8(p)$ і показані сірим кольором на рис. 5.2.

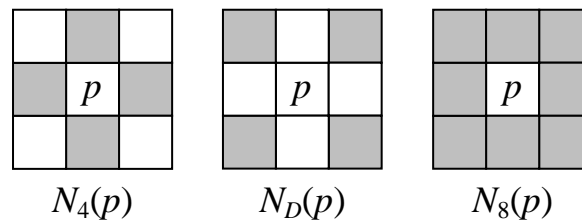


Рис. 5.2. Сусіди пікселя p та їх позначення

Шляхом між пікселями p_1 і p_n називається послідовність пікселів $p_1, p_2, \dots, p_{n-1}, p_n$, така, що p_k є суміжним для p_{k+1} при $k = 1, 2, \dots, n - 1$. Шлях може бути *4-зв'язним* або *8-зв'язним*, залежно від використовуваного визначення суміжності пікселів.

Нехай S – деяка підмножина елементів зображення. Два його елементи p та q називаються зв'язаними в S , якщо між ними існує шлях, що цілком складається з елементів підмножини S . Для будь-якого пікселя p із S множина всіх пікселів, зв'язаних з ним в S називається *компонентою зв'язності* S . Якщо множина S містить тільки одну компоненту зв'язності, то вона називається *зв'язною множиною*.

Для визначення компонент зв'язності в системі MATLAB використовуються функції `bwconncomp`, `bwlabel`. Для аналізу параметрів компонент зв'язності існує функція `regionprops`. Пошук компонент зв'язності у Python можна провести за допомогою функції `label` пакету `skimage.morphology`, а їх аналіз – за допомогою однойменної функції `regionprops` пакету `skimage.measure`.

Морфологічна реконструкція

Реконструкція є морфологічним перетворенням, в якому беруть участь два зображення і один структуроутворюючий елемент. Одне зображення, яке називається *маркером*, є вихідною точкою перетворення. Інше

зображення, *маска*, накладає певні обмеження (зв'язки) на відображення. Використовуваний структуроутворюючий елемент визначає зв'язність.

Якщо g – це маска, а f – маркер, то реконструкція g по f , яка позначається $R_g(f)$, визначається наступною ітеративною процедурою.

1. Ініціалізація: привласнити h_1 маркерне зображення f .
2. Побудувати структуроутворюючий елемент $B = \text{ones}(3)$.
3. Повторювати: $h_{k+1} = (h_k \oplus B) \cap g$ до тих пір, поки не стане $h_{k+1} = h_k$.

Маркер f має бути підмножиною g , тобто $f \subseteq g$.

Процедура морфологічної реконструкції в системі MATLAB реалізована функцією `imreconstruct`. У Python аналогічна операція може бути виконана функцією `reconstruction` пакету `skimage.morphology`.

Інші поширені морфологічні операції

Серед інших поширених морфологічних операцій можна виділити:

- Потоншення
- Потовщення
- Побудову остова (скелетонізація)
- Заповнення порожнистих зв'язних областей
- Сполучення близько розташованих зв'язних областей.
- Операція «виступ» (`tophat`).
- Операція «впадина» (`bottomhat`).
- Отримання контуру зв'язної області.

Для виконання зазначених операцій у MATLAB використовується функція `bwmorph`. У Python ці операції виконуються окремими функціями наявними у пакеті `skimage.morphology`.

Порядок виконання роботи

1. У відповідність до наведених нижче завдань виконати обробку зображень (зображення та необхідні для їх обробки додаткові процедури надаються окремо).

2. Провести експериментальні дослідження впливу параметрів функцій обробки на якість результуючих зображень.
3. Представити процедури обробки зображень у вигляді файла-скрипта.

Завдання

Виконати обробку заданих зображень:

1. Дилатація (файл – ріс.1.jpg).
2. Ерозія (файл – ріс.2.jpg).
3. Розмикання та замикання (файл – ріс.3.jpg).
4. Потоншення (файл – ріс.3.jpg).
5. Побудова остова (файл – ріс.4.jpg).
6. Виділення компонент зв'язності (файл – ріс.5.jpg).
7. Морфологічна реконструкція (слайд – 14, файли – ріс.6a.tif та ріс.6b.tif).
8. Морфологічна реконструкція (слайд – 15, файл – ріс.7.tif).
9. Півтонова дилатація та ерозія (файл – ріс.8.tif).
- 10.Півтонове розмикання та замикання (файл – ріс.9.tif).
- 11.Морфологічний градієнт (файл – cameraman.tif).
- 12.Перетворення «виступ» (файл – ріс.png).
- 13.Морфологічна півтонова реконструкція (файл – ріс.10.tif).

Запитання для самоконтролю

1. В чому полягає суть операцій ерозії та дилатації для бінарних зображень. Як їх можна застосовувати для аналізу зображень.
2. В чому полягає відмінність операцій ерозії та дилатації для півтонових зображень. Для чого застосовуються ці операції.
3. Що таке операції розмикання та замикання. Які дії із зображеннями дозволяють виконувати дані операції.
4. Що таке перетворення успіх/невдача, для чого це перетворення може застосовуватись.
5. Що таке зв'язність, якою вона буває.

6. Дайте визначення компоненту зв'язності.
7. Для чого застосовується морфологічна реконструкція.
8. Наведіть алгоритм морфологічної реконструкції. Який аналіз зображень дозволяє даний алгоритм виконувати.
9. На прикладі функцій **regionprops** та **bwmorph**, наведіть які ще види аналізу зображень можна виконувати за допомогою морфологічних операцій.