

ЛАБОРАТОРНА РОБОТА №6

Сегментація зображень

Тема роботи: методи сегментації зображень.

Мета роботи: дослідити методи сегментації зображень в середовищі MATLAB або Python.

Теоретичні відомості

Виявлення точок

Для виявлення точок на фрагментах зображень, на яких яскравість змінюється повільно доцільно використовувати похідні, зокрема лапласіан. Піксел, в якому є точка (неоднорідність яскравості) можна виявити наступним чином:

$$|R| \geq T$$

де T – ненегативний поріг (зазвичай $T \sim \max(|R|)$), а R – похідні зображення.

Виявлення ліній

Виявлення ліній також здійснюється шляхом застосування до зображення похідних. При цьому, маски похідних повинні мати особливий вигляд: рис. 6.1

<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>2</td><td>2</td><td>2</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	2	2	2	-1	-1	-1	<table><tr><td>-1</td><td>-1</td><td>2</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>2</td><td>-1</td><td>-1</td></tr></table>	-1	-1	2	-1	2	-1	2	-1	-1	<table><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr></table>	-1	2	-1	-1	2	-1	-1	2	-1	<table><tr><td>2</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>2</td></tr></table>	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	-1																																					
2	2	2																																					
-1	-1	-1																																					
-1	-1	2																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	2	-1																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	-1	2																																					
Горизонтальна	+45°	Вертикальна	-45°																																				

Рис. 6.1. Маски для виявлення ліній

При застосування до зображення першої маски, найбільш сильний відгук спостерігатиметься на горизонтальних лініях завтовшки в один піксел. Аналогічно, друга маска дає найбільший відгук на лініях, що

проходять під кутом $+45^\circ$; третя – на вертикальних лініях; четверта – на тих, що проходять під кутом -45° .

Виявлення перепадів

Для виявлення перепадів яскравості на зображенні, як і для виявлення точок та ліній застосовуються похідні першого й другого порядків. Виявлення перепадів в MATLAB можна виконати за допомогою спеціальної функції `edge`. У Python можна для цих цілей можна використати метод Кенні з пакету `skimage.feature`.

Порогова обробка

Порогове перетворення може розглядуватися як операція, що передбачає порівняння елементів зображення з функцією T , яка має вигляд:

$$T = T(x, y, p(x, y), f),$$

де f – зображення, а $p(x, y)$ позначає деяку локальну характеристику точки (x, y) зображення, наприклад, середню яскравість в околиці з центром в цій точці.

Якщо значення T залежить тільки від f , тобто однаково для всіх точок зображення, то такий поріг називається *глобальним*. Якщо поріг T залежить від просторових координат x та y , то він називається *локальним* або *динамічним*.

Якщо T залежить від $p(x, y)$, то такий поріг називається *адаптивним*.

Автоматичний вибір порогу можна виконати наступною ітеративною процедурою:

1. Вибрати деяку початкову оцінку для значення порогу T .
2. Зробити сегментацію за допомогою порогу T (утворюються дві групи пікселів: G_1 и G_2).
3. Обчислити середню яскравість пікселів μ_1 та μ_2 в областях G_1 та G_2 .
4. Обчислити нове значення порогу:

$$T = 0,5(\mu_1 + \mu_2).$$

5. Повторювати кроки з 2-го по 4-й до тих пір, поки різниця порогів T для сусідніх ітерацій не стане менше наперед заданого значення T_0 .

Дана процедура реалізована в стандартній функції MATLAB і називається **graythresh**. Для виконання адаптивної порогової обробки перед операцію **graythresh** можна застосувати перетворення «виступ» (див. [лаб. 6](#)).

Сегментація перетворенням вододілів

Щоб зрозуміти перетворення *вододілу*, необхідно уявити собі півтонове зображення у вигляді поверхні рівнів, на якій величини $f(x, y)$ інтерпретуються як висоти. Якщо уявити собі дощ, що йде над цією поверхнею, то, очевидно, вода збиратиметься в областях, помічених як водозбірні басейни. Краплі дощу, падаючи точно на лінію, помічену як лінія вододілу, з однаковою імовірністю можуть потрапляти як в один, так і в інший басейн.

Перетворення вододілу знаходить водозбірні басейни і будує лінію вододілу на півтоновому зображенні. Якщо використовувати термінологію сегментації, то ключовим моментом є зміна початкового зображення і перетворення його в таке зображення, що водозбірними басейнами на ньому були області, що відповідають об'єктам, які ми хочемо сегментувати.

Для сегментації зображення за допомогою перетворення вододілу в MATLAB передбачена функція **watershed**. У Python пакет **skimage.morphology** має функцію з аналогічним іменем, хоча вона працює дещо інакше, ніж функція MATLAB (див. документацію і приклади).

З метою підвищення якості сегментації двійкових зображень за допомогою перетворення вододілу доцільно використовувати *перетворення відстані* двійкового зображення, яке є досить простою функцією: воно дорівнює відстані від кожного пікселя до найближчого пікселя з ненульовим значенням. Перетворення відстані в MATLAB можна здійснити

за допомогою функції `bwdist`. У Python перетворення відстані може бути виконане функцією `distance_transform_edt` пакету `scipy.ndimage`.

Модуль градієнта часто використовується для попередньої обробки півтонових зображень перед сегментацією по вододілах. Піксели зображення-градієнта з великими значеннями розташовуються поблизу кордонів об'єктів. Таким чином, застосовуючи перетворення вододілу до модуля градієнта, можна отримати лінії вододілів уздовж кордонів об'єктів.

Ще одним підходом, що дозволяє підвищити якість сегментації, зменшуючи надлишкову сегментацією, оснований на *ідеї маркерів*. *Маркер* є зв'язною компонентою, що належить зображенню. Розрізнятимемо *внутрішні маркери*, що належать до об'єктів, які нас цікавлять, і *зовнішні маркери* – ті що відповідають фону зображення. Суть методу полягає в обмеженні за допомогою маркерів розмірів сегментів. Детальніше див. лекцію.

Сегментація шляхом кластеризації за k -середніми

Суть алгоритму кластеризації по k -середніх полягає в оптимізації цільової функції, яка відображає якість розбиття на кластери (сегменти) заданого набору даних (зображення). Цільову функцію отримують із припущення, що існує k кластерів, де k є відомим числом, і вважаючи, що конкретний елемент даних розташовується поблизу центру відповідного йому кластера. Таким чином, у ролі цільової функції виступає деяка міра відстані, наприклад, *евклідова*:

$$\Phi = \sum_{i=1}^k \left[\sum_{j \in \mathbf{I}_i} (x_j - c_i)^T (x_j - c_i) \right],$$

де k – кількість кластерів; \mathbf{I}_i – множина індексів елементів, що належать до i -го кластеру; x_j – j -а інформаційна точка; c_i – центр i -го кластеру.

Алгоритм сегментації по k -середніх можна описати наступною послідовністю дій:

1. Обрати k інформаційних точок у якості центрів кластерів.

2. Віднести кожну інформаційну точку до того кластера, відстань до якого від даної точки є найменшою.
3. Впевнитися, що кожен кластер містить хоча б одну точку. Для цього кожен пустий кластер може бути доповнений довільною інформаційною точкою, розташованою далеко від його центру.
4. Центр кожного кластера замінити середнім значенням елементів, що йому відповідають.
5. Повторювати кроки 2-4, доки не припиниться зміна центрів кластерів.

Для виконання сегментації по k -середніх можна скористатись функцією **kmeans**, яка входить в стандартний набір функцій MATLAB. Для виконання сегментації за k -середніми у Python найпростіше скористатися функцією **kmeans** пакету **cv2** (OpenCV). Особливості використання даної функції описані в документації.

Порядок виконання роботи

1. У відповідність до наведених нижче завдань виконати обробку зображень (зображення та необхідні для їх обробки додаткові файли з функціями надаються окремо).
2. Дослідити як параметри функцій обробки впливають на отримуваний результат.
3. Представити процедури обробки зображень у вигляді файла-скрипта.

Завдання

Виконати обробку заданих зображень:

1. Виявити точки на зображенні (файл – pic.1.tif);
2. Виявити ліній на зображенні (файл – pic.2.tif);
3. Виявити перепади яскравості на зображенні (файл – pic.3.tif);
4. Обробка з глобальним порогом (файл – pic.4.tif);

5. Сегментація по вододілам за допомогою перетворення відстані (файл – pic.5.tif);
6. Сегментація по вододілам за допомогою градієнтів (файл – pic.6.tif);
7. Виконати сегментацію кольорового зображення за допомогою кластеризації по k -середніх (файл – pic.8.jpg).

Запитання для самоконтролю

1. Як здійснюється виділення точок, ліній.
2. Які методи застосовуються для виявлення перепадів яскравості на зображеннях.
3. Що таке порогова обробка (порогове перетворення) зображень, як її формально можна представити.
4. Наведіть процедуру порогової обробки зображень із глобальним порогом.
5. Як можна виконати обробку зображень із адаптивним порогом.
6. В чому полягає суть перетворення вододілу. Як із допомогою цього перетворення можна виконувати сегментацію.
7. Які підходи використовуються для підвищення надійності сегментації за вододілами.
8. Наведіть алгоритм сегментації зображень за k -середніми.