

Практична робота №5

Тема: Бінаризація зображень з використанням таблиць сумарної області

Мета роботи: Вивчення алгоритму побудови інтегральних зображень та бінаризація з їх використанням.

5.1 Теоретична інформація

У випадку коли у якості зчитувального пристрою використовується світлочутлива кольорова матриця, отримане з її допомогою зображення підлягає попередній обробці. Одним із етапів попередньої обробки зображень є бінаризація, яка найбільш широко використовується в системах оптичного розпізнавання символів (ОРС). В процесі бінаризації вхідне зображення в градаціях сірого перетворюється в чорно-біле зображення, яскравості пікселів якого приймають два значення – 0 або 1

Розглянемо основні алгоритми та методи бінаризації півтонових зображень. Історично першим методом бінаризації є метод глобального порогу. Якщо яскравість пікселу менша порогу то йому привласнюється значення 0 (чорний колір), а якщо більше, то 1 (білий колір). Порог обирається на підґрунті аналізу гістограми яскравості зображення. Для побудови гістограми обчислюється або кількість пікселів, що мають однакову яскравість, або частоти

$$p_i = \frac{n_i}{N}, \quad (5.1)$$

де n_i - це кількість пікселів з рівнем яскравості Y , N - загальна кількість пікселів у зображенні. На практиці ці частоти можуть визначатися не тільки для рівнів яскравості з проміжку 0...255, але і для обраних діапазонів яскравості. На рис. 5.1 наведено приклад гістограми яскравості для деякого півтонового зображення.

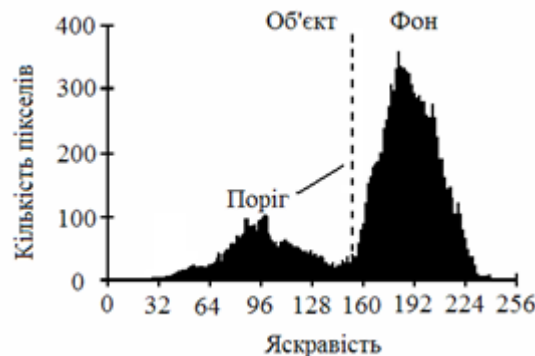


Рис.5.1 – Бімодальна гістограма.

Характерною особливістю цієї гістограми є її бімодальність (тобто вона має два максимуми). При чому один з максимумів відповідає об'єкту, а другий фону. Слід зазначити, що бінаризація з глобальним порогом дає якісні результати лише у випадку коли зображення має чітко виражену бімодальну гістограму. Значення глобального порогу Y_T підбирається, на підґрунті аналізу певної множини схожих за контекстом зображень, так щоб воно знаходилось між максимумами гістограми. Після цього бінаризація зображення здійснюється шляхом по елементної заміни значень яскравості на бінарні в результаті порівняння яскравості $Y(x, y)$ пікселя пороговим значенням Y_T :

$$Y'(x, y) = \begin{cases} 0, & Y(x, y) \leq Y_T; \\ 1, & Y(x, y) > Y_T, \end{cases} \quad (5.2)$$

де $Y'(x, y)$ – бінарне значення пікселів вихідного зображення. Піксель зображення для якого виконується умова $Y(x, y) > Y_T$ називатиметься пікселем об'єкту, а в іншому випадку – пікселем фону.

Відзначимо, що коли об'єкт та фон на зображенні займають приблизно однакову площу, то у якості порогу Y_T можна обрати середнє значення яскравості зображення. У тому випадку, коли площа, яку займає об'єкт, менша ніж площа фону у якості порогу краще обрати пів суму яскравостей, що відповідають максимумам бімодальної гістограми.

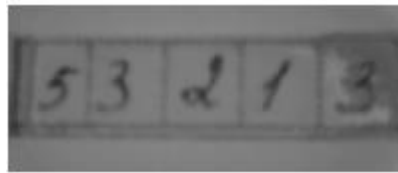


Рис.5.2 - Зображення у градаціях сірого.

У якості глобального порогу оберемо середнє значення яскравості.

$$Y_T = \frac{1}{n \cdot m} \sum_{x,y} Y(x, y), \quad (5.3)$$

де n та m – розмір зображення у пікселях. У результаті отримаємо бінаризоване зображення поштового індексу



Рис.5.3 – Бінаризація за допомогою глобального порогу.

Для кращого розпізнавання потрібно зменшувати значення глобального порогу, щоб цифра «3» стала видимою на зображенні. Більш точними методами є адаптивні методи визначення локального порогового значення інтенсивності.

Локальне порогове значення усуває обмеження глобального порогового значення, враховуючи менші регіони зображення. Він розраховує порогове значення для кожного регіону на основі його місцевих характеристик, таких як середня або середня інтенсивність. Цей підхід забезпечує кращу адаптацію до різних умов освітлення та складного розподілу інтенсивності, що призводить до більш точної сегментації об'єктів або областей із значеннями інтенсивності, що перекриваються. Однак локальне порогове значення може вимагати більше обчислювальних ресурсів і може бути чутливим до шуму або нерівномірного освітлення всередині зображення, що може вплинути на загальну продуктивність алгоритму сегментації.

Адаптивні порогові значення для різних областей зображення потрібні, щоб подолати проблеми, пов'язані з варіаціями умов освітлення та контрастності зображення. Ці адаптивні пороги допомагають підвищити точність і чіткість виявлення об'єкта або області. Цей підхід

передбачає поділ зображення на менші підрегіони та обчислення порогового значення для кожного підрегіону на основі його місцевих характеристик. Завдяки цьому алгоритм може краще врахувати ці варіації та пом'якшити вплив шуму чи нерівномірного освітлення, оскільки кожна підобласть обробляється незалежно.

Найпростішим методом сегментації зображення є порогове значення. За допомогою методу порогового значення сегментація зображення виконується шляхом фіксації всіх пікселів, значення інтенсивності яких перевищують порогове значення, до значення переднього плану.

Середнє та адаптивне порогове значення Гауса

Двома методами, які зазвичай використовуються в обробці зображень, є середнє значення та адаптивне порогове значення Гауса. Середнє адаптивне порогове значення обчислює порогове значення для кожної підобласті, беручи середню інтенсивність усіх пікселів у цій області. З іншого боку, адаптивне порогове визначення Гауса використовує середньозважену інтенсивність пікселів, надаючи більшого значення пікселям, розташованим ближче до центру підобласті. Ці методи ефективні для підвищення якості зображення та підвищення точності в таких завданнях, як виявлення об'єктів або сегментація.



Рис.5.3 – Бінаризація за допомогою адаптивного порогу.

Використовуються в:

- Виявлення об'єктів: встановивши порогове значення, об'єкти можна відокремити від фону, що забезпечує точніше та ефективніше виявлення об'єктів.
- Медичні зображення: порогове значення зображення можна використовувати для сегментації різних структур або аномалій для діагностики та аналізу в медичній візуалізації.
- Контроль якості: порогове значення зображення відіграє вирішальну роль у процесах контролю якості, наприклад, перевірка виготовлених продуктів на наявність дефектів або забезпечення узгодженості кольору та текстури кольорового зображення.
- Сегментація об'єктів: порогове значення зображення також часто використовується в задачах комп'ютерного зору, таких як сегментація об'єктів, де це допомагає відокремити об'єкти переднього плану від фону. Це дозволяє точніше та ефективніше виявляти об'єкти на зображенні.
- Зменшення шуму: порогове значення можна використовувати для зменшення шуму, оскільки це може допомогти усунути небажані артефакти або перешкоди на зображенні.
- Виявлення країв: порогове значення зображення допомагає ідентифікувати та виділяти межі між різними об'єктами або областями в межах зображення за допомогою алгоритмів виявлення країв.

Обчислення адаптивного порогу є доволі важкою операцією для обрахунку тому рекомендовано використовувати інтегральні зображення.

Таблиця сумарної області — це структура даних і алгоритм для швидкого й ефективного генерування суми значень у прямокутній підмножині сітки. У області обробки зображень це також відоме як інтегральне зображення. Він був представлений у комп'ютерній графіці в 1984 році Френком Кроу для використання з МІР-картами. У комп'ютерному зорі він був популяризований Льюїсом, а потім отримав назву «інтегральне зображення» та широко використовувався в структурі виявлення об'єктів Віоли–Джонса в 2001 році. Історично цей принцип дуже добре відомий у вивченні багатовимірних функцій розподілу ймовірностей, а саме в обчисленні 2D (або ND) ймовірностей (площі під розподілом ймовірностей) з відповідних кумулятивних функцій розподілу.

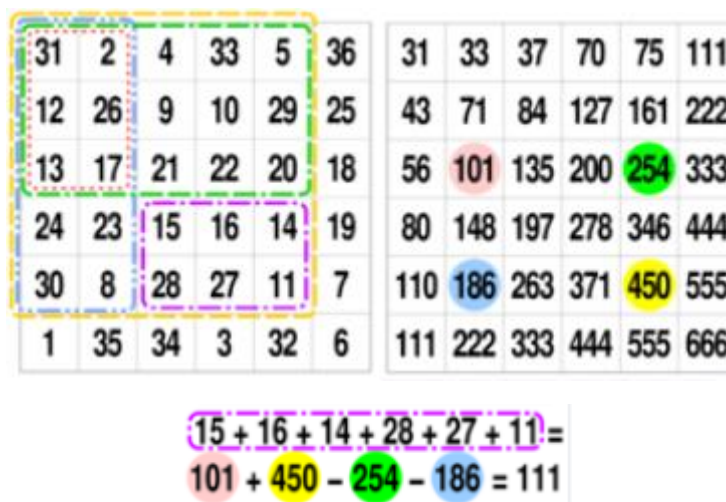


Рис.5.4 – Визначення суми інтенсивності прямокутника за допомогою інтегрального зображення.

Як випливає з назви, значення в будь-якій точці (x, y) таблиці сумарної області є сумою всіх пікселів вище та ліворуч від (x, y) , включно:

$$I(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y'), \quad (5.4)$$

де $i(x, y)$ це значення пікселя в (x, y) .

Таблицю сумарної площі можна ефективно обчислити за один прохід по зображенню, оскільки значення в таблиці сумарної площі в (x, y) дорівнює лише: $I(x, y) = i(x, y) + I(x, y - 1) + I(x - 1, y) - I(x - 1, y - 1)$ (Зверніть увагу, що підсумкова матриця обчислюється з верхнього лівого кута)

Після обчислення таблиці сумарної площі для оцінки суми інтенсивностей будь-якої прямокутної області потрібні чотири посилання на масив незалежно від розміру області. Тобто позначення на рисунку 5.1, де $A = (x_0, y_0)$, $B = (x_1, y_0)$, $C = (x_0, y_1)$ і $D = (x_1, y_1)$, сума $i(x, y)$ у прямокутнику, охопленому A , B , C і D , є:

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} i(x, y) = I(D) + I(A) - I(B) - I(C), \quad (5.5)$$

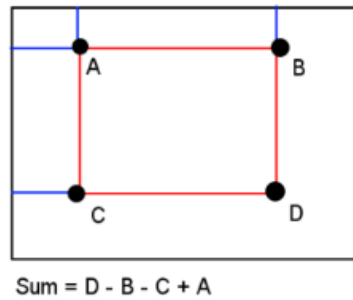


Рис.5.5 – Визначення інтенсивності прямокутної області.

5.2 Порядок виконання роботи

Цей метод природно поширюється на неперервні області.

- 5.2.1. Завантажити зображення;
- 5.2.2. Експериментальним шляхом визначити найоптимальніше значення вікна для адаптивного пошуку порогу на зображенні;
- 5.2.3. Порівняти методи бінаризації (встановлення глобального порогу та адаптивного порогу);
- 5.2.4. Провести повторну бінаризацію для 10 випадків розміру вікна (3, 9, 15, 25, 35, 51, 73, 79, 81, 99) для адаптивного пошуку порогу на зображенні та зафіксувати час виконання алгоритму для класичного пошуку порогового значення(без інтегрального зображення) та з використанням інтегральних зображень;
- 5.2.5. На основі даних про час виконання алгоритму створити графік;
- 5.2.6. Відповідно до свого варіанту обрати зображення 5x5 та провести розрахунок таблиці сумарної області для обраного зображення.

5.3 Зміст звіту

- 5.3.1. Тема лабораторної;
- 5.3.2. Мета лабораторної роботи;
- 5.3.3. Оригінальне зображення;
- 5.3.4. Бінаризоване зображення з визначеним глобальним порогом;
- 5.3.5. Бінаризоване зображення з визначеним адаптивним порогом;
- 5.3.6. Графік залежності часу від розміру вікна(для класичного алгоритму та з використанням інтегральних зображень);
- 5.3.7. Обраховане відповідно до варіанту інтегральне зображення;
- 5.3.8. Висновки.

5.4 Контрольні запитання

- 5.4.1. Що таке глобальна порогова бінаризація і які її недоліки?
- 5.4.2. Які основні відмінності між глобальною та адаптивною бінаризацією?
- 5.4.3. Чому інтегральне зображення дозволяє швидко обчислювати суму пікселів у прямокутній області?
- 5.4.4. Які підходи використовуються для обчислення локального порогу (наприклад, середнє значення, метод Бернсена)?
- 5.4.5. Як обчислюється інтегральне зображення та яку складність має цей алгоритм?
- 5.4.6. Як впливає рівень шуму в зображенні на результати бінаризації?

Варіант 1

1	31	19	13	33
9	7	8	25	4
12	31	4	7	36
0	27	22	13	34
3	20	26	6	26

Варіант 6

6	14	1	29	28
1	12	17	5	3
35	9	1	8	34
19	26	19	0	23
38	5	33	7	24

Варіант 2

2	14	8	39	0
26	36	12	29	30
5	32	15	8	22
10	23	5	0	9
26	25	1	1	3

Варіант 7

7	17	22	39	29
18	1	22	29	38
2	36	20	33	3
18	13	19	6	21
30	19	35	38	37

Варіант 3

3	22	16	17	2
37	18	33	25	29
22	1	26	5	0
8	13	22	25	18
1	17	38	21	23

Варіант 8

8	31	9	33	6
11	39	6	2	30
3	25	36	10	31
14	37	27	25	2
28	28	21	38	36

Варіант 4

4	31	4	36	29
0	37	2	37	39
10	15	4	24	14
11	33	18	4	28
36	28	8	14	29

Варіант 9

9	8	7	15	19
16	37	14	1	10
30	12	24	31	24
5	29	9	7	23
15	11	4	23	32

Варіант 5

5	24	3	0	24
8	36	2	17	31
21	37	36	18	17
10	11	38	0	15
26	25	13	34	13

Варіант 10

10	3	17	29	13
17	19	5	4	34
9	3	8	20	18
7	16	36	27	8
25	23	20	31	14

```

clc; clear; close all;

% Завантаження зображення у відтинках сірого
img = im2gray(imread('number.png'));
[h, w] = size(img);

sigma = 2; % Стандартне відхилення Гаусового ядра
window_size = 21; % Розмір ядра
gaussian_filter = fspecial('gaussian', [window_size window_size], sigma);
img = imfilter(img, gaussian_filter, 'symmetric');

% ===== 1. Глобальний поріг (Otsu's method) =====
global_thresh = graythresh(uint8(img)) * 255;
binary_global = img > global_thresh;

% Визначаємо розмір вікна для адаптивного порога
window_size = 15; % Вікно
half_w = floor(window_size / 2);

% ===== 1. Класичний метод =====
tic;
thresholded_classic = zeros(h, w);

for i = 1:h
    for j = 1:w
        % Визначення меж вікна
        x1 = max(i - half_w, 1);
        x2 = min(i + half_w, h);
        y1 = max(j - half_w, 1);
        y2 = min(j + half_w, w);

        % Обчислення локального середнього значення
        local_mean = mean2(img(x1:x2, y1:y2));

        % Порогова обробка
        if img(i, j) > local_mean
            thresholded_classic(i, j) = 255;
        else
            thresholded_classic(i, j) = 0;
        end
    end
end
time_classic = toc;

% ===== 2. Метод з інтегральним зображенням =====
tic;
% Обчислення інтегрального зображення
integral_img = cumsum(cumsum(double(img), 1), 2);

thresholded_integral = zeros(h, w);

for i = 1:h
    for j = 1:w

```

```

% Визначення меж вікна
x1 = max(i - half_w, 1);
x2 = min(i + half_w, h);
y1 = max(j - half_w, 1);
y2 = min(j + half_w, w);

% Використання інтегрального зображення для швидкого підрахунку суми
S = integral_img(x2, y2);
if x1 > 1
    S = S - integral_img(x1 - 1, y2);
end
if y1 > 1
    S = S - integral_img(x2, y1 - 1);
end
if x1 > 1 && y1 > 1
    S = S + integral_img(x1 - 1, y1 - 1);
end

% Визначення середнього значення
area = (x2 - x1 + 1) * (y2 - y1 + 1);
local_mean = S / area;

% Порогова обробка
if img(i, j) > local_mean
    thresholded_integral(i, j) = 255;
else
    thresholded_integral(i, j) = 0;
end
end
end
time_integral = toc;

% ===== Відображення результатів =====
figure;
subplot(2,2,1); imshow(img); title('Оригінальне зображення');
subplot(2,2,2); imshow(thresholded_classic, []); title('Класичний метод');
subplot(2,2,3); imshow(thresholded_integral, []); title('Метод з інтегральним зображенням');
subplot(2,2,4); imshow(binary_global); title('Глобальний поріг (Otsu)');
% Вивід часу виконання
fprintf('Час класичного методу: %.4f c\n', time_classic);
fprintf('Час методу з інтегральним зображенням: %.4f c\n', time_integral);
fprintf('Прискорення: %.2f разів\n', time_classic / time_integral);

```