

# Алгоритми та структури даних для ефективної морфологічної обробки зображень

*Микита Козлов*

Кафедра моделювання складних систем  
Київський національний університет  
імені Тараса Шевченка  
nikita.kozlov@gmail.com

## Анотація

В роботі розглянуто спосіб узагальненої реалізації алгоритмів морфологічної обробки зображень. Використано архітектурну та алгоритмічну ідеологію клітинних автоматів для побудови швидкодійних модульних реалізацій. Проведено обчислювальний експеримент на прикладі алгоритму скелетизації.

## 1. Вступ

Математична морфологія є сучасним засобом обробки цифрових зображень. Апарат морфологічних операцій широко досліджений теоретично, але безпосередні програмні реалізації не є швидкодійними.

Для широкого спектру практичних задач можна встановити зв'язок між математичною морфологією та теорією клітинних автоматів. Такий підхід дозволяє розробити нові програмні та апаратні рішення, отримати швидкодійні, добре пристосовані до машинного виконання реалізації.

В цій роботі запропоновано структури даних та алгоритми, що дозволяють виконувати широкоскопичні операції над бітональними зображеннями: скелетизація, морфологічна фільтрація, гранулометрія.

## 2. Математична морфологія і клітинні автомати

Апарат математичної морфології застосовується для розв'язання задач фільтрації, геометричного аналізу зображень за допомогою структурних елементів. Як наука, математична морфо-

логія бере початок з ранніх робіт Мінковського та Хадвігера про геометричну теорію міри, інтегральної геометрії; сучасні постулати математичної морфології сформульовані в монографії Серри [1].

Клітинні автомати є стилізованими, синтетичними світами, що визначаються простими правилами, подібними до правил настільної гри. Вони мають власний вид матерії, що циркулює в своєму просторі та часі. Вперше ідея клітинних автоматів була запропонована Конрадом Цузе й Станіславом Уламом і втілена практично Джоном фон Нейманом з метою відтворити поведінку складних просторово-простірних систем [2, 3]. Клітинні автомати є дискретними динамічними системами, поведінка яких повністю визначається в термінах локальних залежностей. Клітинні автомати в інформатиці є аналогом фізичного поняття «поле». Простір являє собою рівномірну сітку, кожна комірка якої, або клітинка, займає декілька бітів даних. Час йде вперед дискретними кроками, а закони світу виражені єдиним набором правил, спрощено – невеликою довідковою таблицею, за допомогою якої будь-яка комірка на кожному кроці обчислює свій новий стан за станом її близьких сусідів. Таким чином, закони системи є локальними й всюди однаковими.

Клітинні автомати дають корисні моделі для багатьох досліджень в природничих науках та комбінаторній математиці; вони, частково, являють собою природній шлях вивчення еволюції великих фізичних систем. До того ж, клітинні автомати утворюють загальну парадигму паралельних обчислень, подібно до машин Тюрінга для послідовних обчислень.

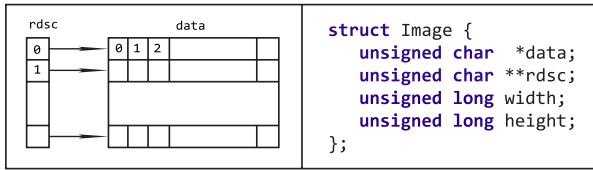


Рис. 1. Схема та реалізація мовою програмування C++ структури даних зображення.

Зв'язок між клітинними автоматами та морфологічною обробкою існує на рівні концепцій. Наприклад, морфологічний структурний елемент відповідає поняттю сусідства для клітинних автоматів; сигнал, що обробляється — простору де діють правила клітинного автомата.

### 3. Структури даних

Не обмежуючи загальності, в роботі розглядаються алгоритми для сусідства в розумінні фон Неймана та Мура, два основних типи простору: площина, тор.

Для представлення зображення використовуємо структуру даних що містить C-масив байт, де кожен елемент відповідає пікселю зображення, додатковий масив-дескриптор стрічок, де  $i$ -й елемент містить вказівник на  $i$ -ту стрічку зображення

Таке представлення є інтуїтивно зрозумілим, простим, гарно пристосованим для реалізації багатьох алгоритмів обробки зображень. Наявність дескриптора стрічок спрощує процес реалізації, є ефективним з точки зору швидкодії.

Безпосередня реалізація морфологічних алгоритмів є складною з багатьох причин. На рисунку 2 зображено підзадачі, що виникають при реалізації алгоритмів морфологічної обробки зображень. В даній роботі запропоновано параметризовані алгоритми відносно типу зображення, представлення об'єктів, сусідства, методів обробки країв зображення. Таким чином, маючи невелику кількість функцій бібліотека алгоритмів морфологічної обробки бітональних зображень є напорчуд гнучкою в використанні. Керуючи лише декількома параметрами користувач отримує понад  $2^{512}$  різноманітних алгоритмів. Далі розглянемо більш детально аспекти реалізації морфологічних алгоритмів.

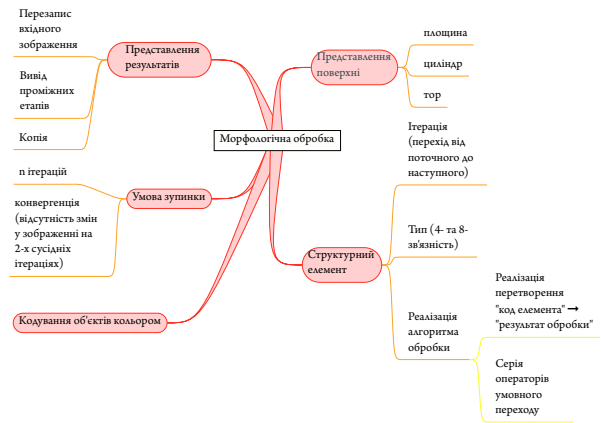


Рис. 2. Аспекти реалізації морфологічних алгоритмів.

#### 3.1. Структурний елемент, представлення поверхні

Представимо структурний елемент у вигляді картки сусідства (тобто фонові пікселі кодуються 0-м, а пікселі об'єкта — 1-ми). Для такої імплементації достатньо 9 біт інформації.

Припустимо, що процес проходження по зображенню виконується в порядку читання книжки (зверзу-вниз зліва-направо). Така схема вимагає двох операцій доступу до зображення, для коректної обробки правої та нижньої границі структурного елемента. Для зберігання всіх можливих варіантів обробки заданого користувачем структурного елемента достатньо попередньо підрахованої таблиці з  $2^9 = 512$  елементів. Підрахунок поточного зміщення в цій таблиці за попередніми значеннями можна реалізувати мовою C++ наступним чином [4]

```

p = ((p<<1) & 0666)
  | ((g<<3) & 0110)
  | (rdsc[y+1][x+1] != 0);

```

Тут  $p$  — кодування структурного елемента поточного пікселя,  $g$  — кодування структурного елемента пікселя, що знаходиться над поточним,  $x$  та  $y$  — координати поточного пікселя, константи представлено в системі числення з основою 8 (octal).

Декомпозиція параметра алгоритма «тип поверхні» необхідна для забезпечення гнучкості бібліотеки, врахування більшої кількості типових задач морфологічної обробки [5]. При реалізації така параметризація впливає лише на

спосіб формування масивів тимчасових значень  $\sigma$ , обробку лівих і нижніх країв зображення. Якщо для простого представлення «площина» достатньо вважати пікселі, що не належать зображенню фоновими, то для поверхонь типу «тор», «циліндр» потрібно відповідним чином замкнути ліву та праву (верхню і нижню) границі зображення відповідно. Це реалізується під час ініціалізації масиву попередніх значень (він стає копією нижньої стрічки зображення), під час обробки правого краю — замість виразу  $(rdsc[y+1][x+1] \neq 0)$  підраховується  $(rdsc[y+1][x+1] \neq rdsc[y+1][0])$ .

### 3.2. Умова зупинки

З всієї множини морфологічних алгоритмів за типом ітерацій можна виділити два: з фіксованою кількістю проходів, з обмеженням на відносні зміни (алгоритм припиняє свою роботу, коли не відбувається змін в зображенні на двох сусідніх ітераціях). Прикладом останнього типу алгоритмів є сімейство алгоритмів скелетизації. Параметризацію за умовою зупинки ввести досить просто, за допомогою картки сусідства та реалізації відповідної функції, що змінює кількість перетворених пікселів. Якщо кількість незмінна — алгоритм зупиняє свою роботу.

### 3.3. Представлення результатів

Виходячи з потреб застосування конкретної реалізації можна виділити такі типи представлення результатів роботи морфологічного алгоритму:

- Вивід результатів безпосередньо в вихідне зображення.
- Вивід результатів в зображення-копію.
- Вивід результату та попередніх, закодованих кольором етапів роботи.

### 3.4. Кодування об'єктів кольором

Як правило, крім інформації про форму об'єкта, вхідне зображення аналізується з метою визначити кількість та різноманітні властивості об'єктів. Такі завдання виникають в задачах сегментації зображень, де природним є кодування різних регіонів різними кольорами (мітками). Не виняток і бітональні зображення. Для людини



Рис. 3. Результати роботи алгоритму скелетизації, з кольоровим кодуванням кожної ітерації для двох типів представлення фону.

природно уявляти фон білим, а об'єкти аналізу (наприклад зображення друкованих символів) — чорним. Для задачі підрахунку інтегральних ознак (інваріантів моментів) зручно фон представити чорним кольором, а об'єкт — білим. Чорний колір в комп'ютерному представленні зазвичай має значення 0, тоді пікселі, належні до об'єкта, нестимуть додатній сигнал, в звичній для задач обробки сигналів дискретизованій формі.

В цій роботі введено параметризацію алгоритмів відносно кольору фону. Тобто можливі два представлення: чорні об'єкти або фон. Семантика структурного елемента при цьому не змінюється. Достатньо ввести необхідні зміни в алгоритм обробки сусідства.

На прикладі використання скелетного представлення в задачах розпізнавання може виникнути потреба зберігати на одному зображенні скелет разом з вхідним об'єктом. В нагоді тут стане використання кодування кольором результатів роботи алгоритму. На рисунку 3 показано результат роботи паралельного алгоритму скелетизації Робертса [4] для різних типів вхідних зображень. Використано вказану властивість виводу попередніх ітерацій (видалених пікселів).

## 4. Застосування

Даний підхід до аналізу та реалізації морфологічних алгоритмів використано при розробці бібліотеки оптичного розпізнавання тексту. Відповідні алгоритми застосовуються в модулях попередньої обробки, фільтрації, пошуку текстових регіонів [3], пошуку ліній та таблиць (алгоритм

скелетизації).

Listing 1. Шаблонна функція морфологічної обробки

```
template<class TrFuncT>
void Image::CAIter(unsigned long qb[], TrFuncT transformFunc )
{
    int p = rdsc[0][0] != 0, q;
    unsigned long x, y;
    for ( x = 0 ; x < width-1 ; ++x )
        qb[x] = p = ((p<<1)&0006) | (rdsc[0][x+1] != 0);

    for ( y = 0 ; y < height-1 ; ++y ) {
        q = qb[0];
        p = ((q<<3)&0110) | (rdsc[y+1][0] != 0);
        for ( x = 0 ; x < width-1 ; ++x ) {
            q = qb[x];
            p = ((p<<1)&0666) | ((q<<3)&0110) |
                (rdsc[y+1][x+1] != 0);
            qb[x] = p;
            transformFunc(rdsc[y][x],p);
        }

        p = (p<<1)&0666;
        transformFunc(rdsc[y][width-1],p);
    }

    for ( x = 0 ; x < width ; ++x ) {
        q = qb[x];
        p = ((p<<1)&0666) | ((q<<3)&0110);
        transformFunc(rdsc[height-1][x],p);
    }
}
```

Listing 2. Клас, відповідний алгоритму скелетизації

```
class RobertsThinTransform
{
    int count;
    unsigned char markcolor;
    unsigned char mask[512];
    RobertsThinTransform(unsigned char markcolor_) {
        // ...
        markcolor = markcolor_;
        count = 0;
    }
public:
    void operator()(unsigned char* dest, int code) {
        if ( mask[code] ) {
            ++count;
            *dest = markcolor;
        }
    }
    int GetCount() {
        return count;
    }
};
```

Listing 3. Функція скелетизації

```
void Image::Thin()
{
    RobertsThinTransform rtobj(127);
    unsigned char* qb = new unsigned char[width];
    do {
        CAIter(qb, &rtobj::operator());
    } while (rtobj.GetCount() != 0);
    delete [] qb;
}
```

Розробки проводилися в рамках проектів Студентської Лабораторії з інформатики компанії International Land Systems (ILS), Inc. (USA), що створена при кафедрі моделювання складних систем факультету кібернетики Київського національного університету імені Тараса Шевченка.

## 5. Висновки

В роботі наведено приклад поєднання засобів клітинних автоматів для побудови ефективних реалізацій алгоритмів морфологічної обробки цифрових зображень. Виділено основні модулі та підзадачі, що виникають під час реалізації. Показано спосіб узагальнення, що може бути використаний для імплементації гнучкої бібліотеки морфологічних алгоритмів [5]. Описані ідеї втілені на прикладі алгоритму скелетизації Робертса [4]. Вказано можливі місця застосування алгоритмів.

В подальшому планується об'єднати бібліотеку морфологічних алгоритмів з бібліотекою генетичних алгоритмів. Таким чином, описана вище ідеологія (набір функцій перетворення) може мати семантику гена. Запустивши відповідним чином алгоритм, в множині всіх можливих переворень проводиться пошук послідовності, яка переводить зображення текстової сторінки з формулами в зображення, що містить лише формули, тобто проводить пошук певних типів структурних елементів на зображенні.

## 6. Література

- [1] Ritter, G.X. and Wilson, J.N., Handbook of Computer Vision Algorithms in Image Algebra, second edition, CRC Press, 2001.
- [2] Schroeder, M., Fractals, chaos, power laws: Minutes from an infinite paradise, W.H. Freeman, New York, 1991.
- [3] Weber, D., Detecting text regions using cellular automata. C Users J. Vol. 11, 8:35–52, 1993.
- [4] Cychosz, J. M., Efficient binary image thinning using neighborhood maps. In Graphics Gems IV, P. S. Heckbert, Ed. Academic Press Professional, San Diego, CA, 1994, pp. 465–473.
- [5] Alexandrescu, A., Modern C++ Design: Generic Programming and Design Patterns Applied. Addison-Wesley, 2001.