

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА  
АКАДЕМІЯ»  
Кафедра мультимедійних систем

**АНАЛІЗ КОНТУРІВ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ  
КЛІТИННИХ АВТОМАТІВ**  
**Текстова частина до курсової роботи**  
**за спеціальністю „Інженерія програмного забезпечення” 121**

Керівник курсової роботи  
к.ф.-м.н., доц. Жежерун О.П.  
\_\_\_\_\_  
(підпис)

“ \_\_\_\_ “ \_\_\_\_\_ 2021 р.

Виконав студент Кривошея М.І.  
“ \_\_\_\_ “ \_\_\_\_\_ 2021 р.

Київ 2021

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ  
Зав.кафедри мультимедійних  
систем, к.ф.-м.н., доц.  
\_\_\_\_\_ О. П. Жежерун  
„\_\_\_\_\_” \_\_\_\_\_ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ  
на курсову роботу

студенту Кривошеї Михайлу Ігоровичу факультету інформатики 4-го курсу  
ТЕМА Аналіз контурів зображень за допомогою клітинних автоматів

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

1. Огляд методів аналізу контурів зображень
2. Розробка алгоритму на основі КА з використанням ДВ для вдосконалення методів аналізу контурів зображень та подальшої його реалізації за допомогою графічних карт
3. Розробка програми реалізації алгоритму побудови ДВ на основі КА

Висновки

Список літератури

Додатки (за необхідністю)

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2020 р. Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

## Календарний план виконання курсової роботи

Тема: Аналіз контурів зображень за допомогою клітинних автоматів

### Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	01.11.2020	
2.	Огляд технічної літератури за темою роботи	До 15.11.2020	
3.	Виконання огляду методів аналізу контурів зображень	До 25.11.2020	
4.	Виконання огляду теоретичних основ понять КА та ДВ	До 01.12.2020	
5.	З'ясування суті методу КА для аналізу контурів зображень	До 05.12.2020	
6.	Виконання порівняльного аналізу виявлення контурів на основі КА та традиційних методів	До 10.12.2020	
7.	Розробка алгоритму для побудови ДВ на основі	До 27.12.2020	
8.	Програмування розробленого алгоритму	До 05.02.2021	
9.	Застосування розробленого алгоритму	До 20.02.2021	
10.	Виконання порівняльного аналізу отримуваних результатів	До 25.02.2021	
11.	Написання пояснювальної роботи	До 17.03.2021	
12.	Створення слайдів для доповіді та написання доповіді	До 19.03.2021	
13.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист курсової роботи	До 23.03.2021	
14.	Корегування роботи за результатами попереднього захисту	До 30.03.2021	
15.	Остаточне оформлення пояснювальної роботи та слайдів	До 02.04.2021	
16.	Захист курсової роботи	Квітень 2021	

Студент \_\_\_\_\_ М. І.Кривошея

Керівник \_\_\_\_\_ О.П.Жежерун

“ \_\_\_\_\_ ” \_\_\_\_\_

## ЗМІСТ

<b>Анотація .....</b>	<b>4</b>
<b>Перелік прийнятих скорочень, умовних позначень та термінів .....</b>	<b>5</b>
<b>Вступ .....</b>	<b>6</b>
<b>РОЗДІЛ 1 Огляд методів аналізу контурів зображень .....</b>	<b>8</b>
1.1 Основна задача аналізу зображень та методи розпізнавання .....	8
1.2 Виявлення контурів зображень .....	8
1.2.1 Відслідкування меж областей .....	9
1.2.2 Виявлення країв Кені .....	9
1.2.3 Виявлення прямих і відрізків, кіл і їх дуг .....	10
1.2.4 Підбір моделей сегментів та групування контурних фрагментів .....	11
1.3 Алгоритми виділення й проходження контуру зображення та їх порівняльна характеристика .....	12
1.4 Постановка задачі курсової роботи .....	17
1.5 Висновки до розділу 1 .....	17
<b>РОЗДІЛ 2 Дефінітивна характеристика понять «клітинний автомат» та «діаграма Вороного». Метод КА .....</b>	<b>18</b>
2.1 КА як дискретна динамічна система .....	18
2.1.1 Визначення КА та основних понять, пов'язаних з ним ...	18
2.1.2 Класифікація КА .....	25
2.2 ДВ як універсальна структура .....	29
2.2.1 Визначення ДВ та основних понять, пов'язаних з нею ...	29
2.2.2 Властивості ДВ .....	32
2.2.3 Алгоритми побудови ДВ .....	34
2.2.4 Метод ЦДОВ як приклад застосування ДВ .....	36
2.3 Теоретичні основи написання алгоритмів на базі КА .....	38
2.4 Метод КА, його застосування та переваги .....	39
2.5 Висновки до розділу 2 .....	41

<b>РОЗДІЛ 3 Розробка ПП на основі КА з використанням ДВ .....</b>	<b>42</b>
3.1 Виявлення контурів зображень на основі КА .....	42
3.2 Порівняння результатів виявлення контурів на основі КА та традиційних методів .....	43
3.3 Розробка на основі КА алгоритму побудови ДВ .....	44
3.4 Програмна реалізація алгоритму на основі КА для побудови ДВ	47
3.4.1 Мова та середовище програмування .....	47
3.4.2 Етапи розробки програми .....	48
3.4.3 Тестування програми .....	49
3.4.4 Порівняння одержуваних результатів побудови ДВ ....	53
3.5 Висновки до розділу 3 .....	54
<b>Висновки по роботі .....</b>	<b>55</b>
<b>Список літератури .....</b>	<b>56</b>

## АНОТАЦІЯ

курсової роботи Кривошеї Михайла Ігоровича  
на тему «Аналіз контурів зображень за допомогою клітинних автоматів»

Створення програмного продукту на основі клітинних автоматів з використанням діаграм Вороного для вдосконалення існуючих методів аналізу контурів зображень та для подальшої його реалізації за допомогою графічних карт – мета даної роботи.

В ході виконання роботи було здійснено огляд алгоритмів аналізу контурів зображень; здійснено дефінітивну характеристику понять «клітинний автомат» та «діаграма Вороного»; розглянуто метод клітинних автоматів та його застосування, з'ясовано переваги цього методу; розглянуто властивості діаграм Вороного, алгоритми їх побудови та приклади застосування; опрацьовано теоретичні основи виявлення контурів на основі клітинних автоматів і здійснено порівняння результатів роботи алгоритмів на основі клітинних автоматів та традиційних методів щодо виявлення контурів, вказано можливі шляхи вдосконалення алгоритмів на основі КА для виявлення контурів.

На цьому підґрунті було розроблено алгоритм побудови діаграм Вороного на основі клітинних автоматів та створено програмний продукт, що реалізує цей алгоритм.

Ключові слова: клітинний автомат, аналіз контурів зображень, діаграма Вороного.

Розмір текстової частини до курсової роботи 54 аркуші, робота містить рисунків – 23, таблиць – 1.

## ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

ДВ – діаграма Вороного

КА - клітинний автомат

ПП - програмний продукт

ЦДОВ – метод центральних дискретних осей Вороного

API (Application Programming Interface) – інтерфейс програмування

C + + – мова програмування

OpenCV (Open Source Computer Vision Library) – бібліотека програмного забезпечення

Visual Studio – інтегроване середовище розробки програмного забезпечення

## ВСТУП

Задачі аналізу і обробки зображень постають все в більш численних галузях науки. Цього потребують великі обсяги даних, що надходять з найрізноманітніших джерел. У сучасному світі розвиток засобів аналізу й обробки інформації характеризується широким впровадженням різних алгоритмів. Методи аналізу та обробки зображень є дуже різноманітними. Вони залежать від завдання дослідження [1].

Клітинні автомати поширені в багатьох сферах, у тому числі в соціології, військовій справі, економіці, фізиці, математиці. Їх застосовують як у моделюванні біологічної системи, так і в створенні віртуальної соціальної мережі [2]. Немає чітких критеріїв, у якому випадку краще спрацює той чи інший метод, а в якому необхідно шукати нові підходи. Аналіз зображень, зокрема і їх контурів, за допомогою КА – цікава задача. Водночас вона є перспективною, оскільки не існує досліджень, які б не можна було змодельовати за допомогою КА [3]. З огляду на вищесказане, тема цієї роботи є актуальною.

**Метою** даної роботи є створення програмного продукту на основі КА з використанням діаграм Вороного для вдосконалення існуючих методів аналізу контурів зображень та подальшої його реалізації за допомогою графічних карт.

**Об'єкт дослідження** – алгоритми аналізу контурів зображень.

**Предмет дослідження** – метод КА для аналізу контурів зображень.

**Завдання дослідження:**

1. Здійснити огляд методів аналізу контурів зображень.
2. Дати дефінітивну характеристику понять «клітинний автомат» та «діаграма Вороного».
3. На основі КА розробити алгоритм побудови ДВ з метою його використання для вдосконалення методів аналізу контурів зображень і подальшої реалізації за допомогою графічних карт.



4. Виконати програмну реалізацію розробленого алгоритму.

**Практичне значення одержаних результатів** – розроблений алгоритм може слугувати вдосконаленню методів аналізу контурів зображень на основі КА та має перспективу подальшої реалізації за допомогою графічних карт.

**Використане програмне забезпечення** – Visual studio, C++, бібліотека OpenCV для роботи із зображеннями.

**Структура роботи.** Робота складається зі вступу, трьох розділів, висновків, списку літератури.

Перший розділ присвячений огляду методів аналізу контурів зображень. У цьому розділі розглянуто основну задачу аналізу зображень і методи розпізнавання; здійснено огляд алгоритмів виявлення, проходження та опису контурів зображень, у тому числі оглянуто відповідно алгоритм змії та алгоритм Кені; алгоритми, що базуються на операторах Собеля, Лапласа, Превіта; алгоритм жука й алгоритми «Moore-Neighbor Tracing», «Redial Sweep», «Theo Pavlidi's Algorithm»; дано порівняльну характеристику переваг та недоліків цих алгоритмів та здійснено постановку задачі курсової роботи.

У другому розділі дано дефінітивну характеристику понять «Клітинний автомат» та «Діаграма Вороного», розглянуто класифікацію КА, вказано властивості ДВ; наведено алгоритми побудови ДВ та приклад застосування ДВ – метод автоматичної сегментації зображень; описано метод КА та розглянуто теоретичні основи написання алгоритмів на базі КА, проаналізовано переваги КА перед іншими методами.

Третій розділ присвячений опису реалізації ПП, зокрема в ньому розглянуто виявлення контурів зображень на основі КА, порівняно одержувані результати з традиційними методами, вказано можливі шляхи вдосконалення алгоритмів виявлення контурів зображень на основі КА, розроблено алгоритм побудови ДВ на основі КА та здійснено програмну реалізацію цього алгоритму.

## РОЗДІЛ 1. ОГЛЯД МЕТОДІВ АНАЛІЗУ КОНТУРІВ ЗОБРАЖЕНЬ

### *1.1 Основна задача аналізу зображень та методи розпізнавання*

Основна задача аналізу зображень – це комп’ютерне розпізнавання та класифікація зорових образів. В роботі [4] виділено наступні найбільш поширені методи розпізнавання образів:

- кореляційні (найбільш трудомісткі щодо обчислювальної складності);
- статистичні (провідна проблема – знаходження функції умовної густини розподілу ймовірності значень ознак для кожного класу);
- методи, що базуються на штучних нейронних мережах (складність полягає у виборі архітектури мережі та алгоритмів їх навчання);
- структурні та синтетичні (важко формулюється задача визначення граматик на множині висловлювань, що продовжують мову).

Немає спільного універсального теоретичного підходу щодо аналізу зображень в цілому та їх контурів зокрема. Існує ряд класів зображень, які характеризуються повторюваністю елементів. За умови, що вони будуються на законах симетрії, підґрунтям для їх опису є теорія груп. Розробленим є метод аналізу симетричних зображень щодо мінімальних структурних частин. Частіше мають місце асиметричні зображення. Суть методу їх аналізу полягає у зведенні асиметричних зображень до відомих симетричних структур. Для асиметричних зображень також використовують методи, що базуються на застосуванні автокореляційної функції та перетворення Фур’є [4].

### *1.2. Виявлення контурів зображень*

Контур – це межа між двома різними областями зображення, які відрізняються або інтенсивністю, або кольором, або текстурою.

Існує багато випадків, коли в задачах аналізу зображень потрібна робота не з виділеними областями, а лише з їх контурами (межами). Відрізки

прямих, дуг кіл, еліпсів та інших кривих можуть бути складовими межі. Існують алгоритми виявлення та виділення контурів.

### ***1.2.1 Відслідкування меж областей***

Відслідкування контурів здійснюється за допомогою алгоритму, що працює з тими зображеннями, в яких вже виконана сегментація областей або виконано маркування зв'язних компонент.

Для невеликих зображень суть розв'язання задачі полягає в переборі всіх точок зображення та формуванні списку, що містить перші граничні точки кожної з усіх областей. На наступному етапі відбувається обробка кожної області шляхом пошуку сусідніх пікселів контуру. Здійснюють пошук за годинниковою стрілкою до повернення в початкову точку. У випадку великих зображень використовується інший алгоритм, який працює з попередньо маркованими зображеннями. Суть роботи цього алгоритму полягає у перегляді всіх пікселів зліва – направо та згори – донизу. На виході після обробки зображення будуть межі областей, кожна з яких подана списком пікселів (порядок обходу межі за годинниковою стрілкою). При цьому вважають, що фонові області є єдині. Тим областям, контури яких не відслідковуються, присвоюється мітка контуру. В цьому процесі при послідовній обробці пікселів зображення шуканий піксель межі області відноситься до наступної точки відповідної області [5].

В ході процесу обробки зображення наявні межі областей, які вже оброблені, часткові контури областей, які знаходяться в обробці та межі областей, обробка яких ще має відбутися. Зауважимо, що у вигляді структури даних зберігаються результати роботи алгоритму.

### ***1.2.2 Виявлення країв Кені***

Для виділення контурів на напівтонових зображеннях використовують детектор країв Кені, який генерує тонкі фрагменти контурів зображень.

В роботі цього алгоритму задіяно оператор згладжування  $\sigma$  та два порогових значення інтенсивності. Алгоритм виявлення країв Кені подано в роботі [5]. Після згладжування зображення Гаусовим фільтром, розмах якого  $\sigma$ , обчислюють величину та напрям градієнта. Напрямок градієнта необхідний для уточнення контурів, а саме: видаляють пікселі, величина градієнта для яких не більша, ніж у двох сусідніх пікселях із обох боків пікселя, що обробляють, вздовж напрямку градієнта. Цей підхід забезпечує одержання тонких контурів. На наступному етапі виявляють неперервні сегменти контурів. Починають із крайових пікселів, значення градієнта в яких більше, ніж верхнє порогове значення. Потім виявлення контуру відбувається у всіх тих пікселях, у яких значення градієнта є не меншим від нижнього порогового значення. Нижнє порогове значення, як правило, в два рази менше від величини верхнього порога.

### ***1.2.3 Виявлення прямих і їх відрізків, кіл і їх дуг***

На півтонових та кольорових зображеннях виявлення прямих, кіл та інших параметричних кривих здійснюється за допомогою перетворення Хафа. В процесі перевірки кожного пікселя та його сусідства (околу) з'ясовується, чи є в даному пікселі чітко виражений край. Якщо так, то знаходять параметри прямої  $y = ax + b$ , яка проходить через даний піксель. Алгоритм визначення цих параметрів використовує масив, цей алгоритм подано в роботі [5]. Кількість невідомих параметрів у рівняннях шуканих прямих (аналогічно й інших кривих) визначають розмірність масиву. Отримані параметри аналізують. В масиві знаходять локальні максимуми. Вони і є параметрами найбільш ймовірних прямих на зображенні.

За умови, що на зображення є вертикальні відрізки та прямі, для їх виявлення використовують наступний вигляд рівняння:

$$d = x \cos \theta - y \sin \theta, \quad (1.1)$$

де  $d$  – довжина перпендикуляра, опущеного з початку координат на дану пряму;

$\theta$  – кут між цим перпендикуляром і горизонтальною віссю.

Знак « – » відображає специфіку екранної системи координат – початок координат у лівому верхньому куті, напрям горизонтальної осі зліва – направо, а вертикальної – зверху донизу. Алгоритми О`Гормана і Кловса для знаходження параметрів  $d$  і  $\theta$  подано в роботі [6].

За умови екранної системи координат, координати точок кола обчислюються за формулами:

$$r = r_0 + d \sin \theta; c = c_0 + d \cos \theta, \quad (1.2)$$

де  $r, c$  – координати точок кола;

$r_0, c_0$  – координати центра кола;

$d$  – радіус кола;

$\theta$  – кут між радіусом, що сполучає точку кола з центром цього кола, та горизонтальною віссю.

#### ***1.2.4 Підбір моделі сегментів та групування контурних фрагментів***

При виділенні контурів зображення потрібно розв'язувати задачу щодо його апроксимації відповідною моделлю (заміни виділеного контуру прямою, дугою кола, еліпса, іншої кривої). Для підбору такої моделі може бути використаний метод найменших квадратів, який описано в роботі [5]. В цій же роботі для бінарного контурного зображення подано алгоритм виявлення контурів.

За умови, що дане зображення містить марковані контурні зображення, пікселі яких мають бінарні значення, контурні лінії не завжди є замкненими, а тому можуть утворювати сегменти з початковими та кінцевими пікселями. Сегменти можуть мати вигляд кутів і зіткнень. Реалізація алгоритму виявлення контурних фрагментів має забезпечувати виявлення нового сегменту, приєднання нового пікселя до сегменту, виявлення кінця сегменту й виявлення кутів і зіткнень. Суть реалізації алгоритму полягає в аналізі сусідства (околу) кожного наступного пікселя. З'ясовується, даний піксель є

або ізольованою точкою, або початковою точкою нового сегмента, кутовою точкою або точкою зіткнення. За умови, що піксель є кінцевою або внутрішньою точкою існуючого сегменту, алгоритм визначає його ідентифікатор. У випадку, коли це кутова точка або точка зіткнення, визначаються списки пікселів, які належать цим сегментам, а також визначають ідентифікатори сегментів.

### ***1.3 Алгоритми виділення й проходження контуру, їх порівняльна характеристика***

Важливими етапами аналізу зображення є виділення та проходження контуру цього зображення. Контур об'єкту є такою частиною об'єкту, що містить велику кількість інформації про форму об'єкту та мало залежить від кольору й текстури зображення [7].

Структура контуру може бути наступною:

- проста (у формі мікрооб'єкта відсутні малоінформативні відгалуження, а також мікрооб'єкт знаходиться окремо від інших);
- з однопіксельними відгалуженнями (на контурі мікрооб'єкта мають місце малоінформативні відростки);
- складна (мікрооб'єкти дотикаються або накладаються, при цьому утворюються хибні мікрооб'єкти, їх доцільно роз'єднати).

Проходження контуру (contour tracing) є етапом отримання дискретного сигналу, який описує межі оцифрованого мікрооб'єкту. Алгоритм проходження контуру повинен задовольняти наступним вимогам:

- зменшення обсягів пам'яті для зберігання;
- зменшення часу та складності подальшої обробки;
- отримання інформативних ознак об'єкт, – всі ці вимоги вказані в роботі [8].

Виділення контуру можна здійснювати наступними способами:

- підкреслення меж мікрооб'єкта шляхом фільтрації вхідного зображення;
- проходження внутрішнім контуром однорідної області.

Алгоритм зм'ї, алгоритм Кенні, фільтрація, що ґрунтується на операторах Собеля, Лапласа, Превіта – це основні алгоритми виділення меж об'єкта [9 – 13]. Суть їх у підкресленні різних перепадів яскравості. Ці перепади характерні межах об'єктів. При застосуванні цих алгоритмів отримують набір незв'язних областей, а тому з метою отримання зв'язного контуру потрібно здійснити додаткову обробку.

Зауважимо, що контур є частиною об'єкту. Порогова сегментація, кластеризація, нарощування областей, алгоритм водоподілу, блочна сегментація – це алгоритми виділення областей [9 – 13]. Підґрунтям для цих алгоритмів є об'єднання пікселів у однорідні області з урахуванням деякого критерію однорідності. В результаті матимемо набір однорідних областей.

Для розв'язання задачі щодо визначення зовнішнього контуру (межі) зображення в роботі [14] використано метод, який не потребує аналізу ділянок, колір яких ідентичний до кольору зображення об'єкту або наближений до нього з похибкою  $\Delta$ . При такому підході визначають пікселі фону. В роботі [15] запропоновано використання динамічного значення кольору фону, що слугує покращенню роботи запропонованого алгоритму, а саме знижує вплив шумів та спотворень. Проте за умови, що фон впродовж значного проміжку є схожим із кольором зображення об'єкту або за умови, що спотворення та шуми перекривають зображення об'єкту, знижується ефективність роботи. В такому випадку зображення потребує додаткової обробки перед початком роботи програми.

З метою відсікання малоінформативних точок і усунення деяких недоліків (поодинокі пікселі, що відрізняються від фону, але не є частиною об'єкту) застосовують перевірку сусідніх пікселів на належність до фону. Для цього вводять зображення рисунка зі значеннями 0 та 1, де 0 – точка належить фону, 1 – точка не належить фону. Правила виконання алгоритму фільтрації та алгоритм визначення точок контуру зображення подано в роботі [15] і він є наступним:

1. Обирається стартова точка.

2. Обирається наступний піксель (сусідній при русі за годинниковою стрілкою), що не належить фону, але межує з ним.
3. Перевіряється наявність наступної (сусідньої) такої точки, яка б задовольняла умови належності точки контуру. В противному випадку переходять до пункту 5.
4. Якщо координати активної точки не змінилися, то їй присвоюється статус фону й активною обирається попередня точка і здійснюється перехід до пункту 2.
5. Перевіряється на завершення визначення контуру (його замкнення), за умови, що активна точка повертається до початкового стану застосовують критерій Джакобса. Якщо умова не виконується – переходять до пункту 2.
6. Після виділення програмою деякої цілісної області зображення, відмінної від фону (це означає, що отримано замкнений контур), кожній точці зображення, яка обмежена контуром, присвоюється ідентифікатор клітини. Це здійснюється з метою унеможливлення повторного опрацювання цих пікселів. За умови завершення присвоєння здійснюється перехід до пункту 1.

Щоб мати опис контуру об'єкту, потрібно застосувати алгоритми проходження контуром. Для проходження та опису контуру широко використовуються наступні алгоритми:

- Алгоритм «жука» (Square Tracing Algorithm) описаний в роботі [16]. Базовими є два правила: якщо значення активного пікселя дорівнює одиниці (активний піксель знаходиться в точці, що належить об'єкту), то здійснюється поворот ліворуч і навпаки, за умови нульового значення пікселя (активний піксель знаходиться в точці, яка не належить об'єкту) здійснюється поворот праворуч. Робота алгоритму завершується за умови повернення в стартову точку. Простота алгоритму «жука» та його швидкодія є основними його перевагами в порівнянні з іншими алгоритмами, а недоліком – неефективне опрацювання тих зображень, які



мають діагональні сторони або «відгалуження», товщиною в один піксель, що унеможливорює використання цього алгоритму для визначення контуру складної фігури;

- Алгоритм «Moore-Neighbor Tracing» практично є удосконаленням алгоритму «Square Tracing Algorithm», описаний в роботі [8]. Суть полягає у покроковій перевірці всіх сусідніх точок для того, щоб знайти наступну контурну точку. Робота алгоритму завершується за умови повернення в стартову точку. Внаслідок збільшення кількості точок, які проходять процес перевірки, цей алгоритм є більш ефективним у порівнянні з попереднім за умови проходження зображень, що мають довільні сторони. Серед переваг цього алгоритму – можливість опрацювання складніших зображень та підвищення точності оброблення. Його недоліками є ускладнення обчислень, зменшення швидкодії та можливість некоректного завершення роботи;
- Алгоритм «Redial Sweep» - це модифікація попереднього алгоритму. Вдосконалення полягають у тому, що кожна наступна точка контуру прив'язується до попередньої. Точка початку обходу активного пікселя - це точка, яка була визнана контурною на попередньому кроці алгоритму (на відміну від попереднього алгоритму, де точкою початку обходу була точка, з якої відбувався перехід на активний піксель). Крім того, з'явився ще один критерій для зупинки алгоритму – у випадку потрапляння в точку, яка вже належить контуру. Серед переваг цього алгоритму можливість працювати зі складними контурними лініями, наявність додаткового критерію зупинки – це дає можливість підвищувати ефективність щодо уникнення зациклення роботи алгоритму й коректного завершення роботи ;
- «Theo Pavlidi's Algorithm» описаний в роботі [17]. Суть полягає у використанні групи, до складу якої входить три пікселя, для визначення наступного пікселя. Відбувається послідовна перевірка сусідніх точок зі строго визначеною черговістю. Висока швидкодія, висока якість роботи з

нескладною контурною лінією, можливість використання довільної точки контуру як стартової точки – переваги цього алгоритму. Складність реалізації, складність розв’язання проблеми щодо обробки зображень з контурами, які мають виступи товщиною в один піксель, недосконалість критеріїв завершення алгоритму – його недоліки.

Практично для кожної мови програмування є бібліотеки з реалізованими алгоритмами [18].

Залежність від складності контуру й критерію зупинки – недолік вказаних алгоритмів. Чутливість до мікрооб’єктів, контур яких містить відгалуження товщиною в один піксель, характерна для цих алгоритмів. Це може зумовити помилкове завершення роботи алгоритмів, а також некоректне виділення контуру. У випадку, коли мікрооб’єкт складається з кількох частин, які з’єднані між собою тільки одиничними пікселями, виникає аналогічна проблема. Некоректні результати роботи алгоритмів зумовлюються недосконалими критеріями зупинки, зокрема, повернення у стартову точку, проходження деякої точки декілька разів.

Контур характеризується формою, структурою, типом зв’язності. Зауважимо, що тип зв’язності задається при кодуванні контуру. За умови, що контур 4-зв’язний, то координати двох сусідніх точок, які належать контуру об’єкту, можуть відрізнятися лише по осі абсцис або лише по осі ординат. У випадку, що контур 8-зв’язний, то координати двох сусідніх точок, які належать контуру об’єкту, можуть відрізнятися по осі абсцис і/або по осі ординат.

Підвищенню ефективності та працездатності алгоритмів проходження контуром зображення слугують критерії зупинки їх роботи. Вони є наступними:

- зупинка у випадку відвідування алгоритмом точки  $n$  разів (ефективно, якщо наперед відома кількість повернень алгоритму до стартової точки);
- критерій зупинки Джакобса (зупинка після відвідування стартової точки алгоритмом два рази) – частинний випадок попереднього, його перевага у

простоті реалізації та високій ефективності при роботі з простими контурами фігур, а недолік пов'язаний з вибором стартової точки;

- якщо має місце потрапляння на точку, яку вже було визнано точкою контуру, то це є критерієм зупинки. Довільність вибору стартової точки – перевага, необхідність у маркуванні пройдених контурних точок, неефективність щодо визначення виступів на контурі, товщина яких один піксель – недоліки критерію, – всі ці критерії подано в роботах [15, 16].

#### ***1.4 Постановка задачі***

Для виявлення країв у спосіб, що базується на градієнтних операторах, а також на операторі Лапласа необхідним є значний обчислювальний час, тому для розв'язання задачі швидкісної обробки зображень у реальному часі потрібно реалізовувати паралельні алгоритми замість послідовних. Клітинні автомати, як паралельна обчислювальна модель, можуть слугувати успішному розв'язанню цієї проблеми. Ось чому постає задача створення алгоритму на базі КА, який слугував би вдосконаленню методів аналізу контурів зображень. Доцільно, щоб цей метод застосовувався в органічному поєднанні з іншими, які б посилювали переваги та щоб мав перспективи подальшої реалізації за допомогою графічних карт.

#### ***1.5 Висновки до розділу 1***

У цьому розділі розглянуто основну задачу аналізу зображень і методи розпізнавання; здійснено огляд алгоритмів виявлення, проходження та опису контурів зображень, у тому числі оглянуто відповідно алгоритм змії та алгоритм Кені; алгоритми, що базуються на операторах Собеля, Лапласа, Превіта; алгоритм жука й алгоритми «Moore-Neighbor Tracing», «Redial Sweep», «Theo Pavlidi's Algorithm»; дано порівняльну характеристику переваг та недоліків цих алгоритмів і на цій основі здійснено постановку задачі курсової роботи.

## РОЗДІЛ 2. ДЕФІНІТИВНА ХАРАКТЕРИСТИКА ПОНЯТЬ «КЛІТИННИЙ АВТОМАТ» ТА «ДІАГРАМА ВОРОНОГО». МЕТОД КА

### *2.1 КА як дискретна динамічна система*

КА є дискретними динамічними системами, поведінка яких визначається в термінах локальних взаємодій і є прикладом розподілених систем, які засновані на простих правилах, що дозволяють реалізувати складну поведінку [3].

Вперше термін клітинний автомат був введений наприкінці 1940-х років Джоном фон Нейманом за пропозицією Станіслава Улама, проте КА стали популярними лише тоді, коли Джон Хортон Конвей розробив гру «Життя». Це був кінець 1960-х років [19].

Машина клітинного автомата є синтезатором світів. Подібно до органу, він має клавіші та реєстри, за допомогою яких можливості інструменту можна приводити до дії, комбінувати та перекомпоновувати, а його кольоровий екран є вікном, крізь яке можна спостерігати за світом, який весь час «грає» [20].

#### *2.1.1 Визначення КА та основних понять, пов'язаних з ним*

КА можна розглядати як деяку сітку регулярних комірок. Їх називають клітинами автомату. Здебільшого стан клітини подається цілим значенням. Кожній комірці (клітині) притаманний один з кінцевих станів. Зауважимо, що клітини змінюються синхронно за допомогою певної функції (ця функція керує станом самої клітини). В свою чергу сітка регулярних комірок (клітин) КА може мати скінчену кількість вимірів. На рисунку 2.1 зображено сусіди клітини в одновимірній сітці. Найближче оточення сірої клітини зображено зеленим кольором; жовтим та червоним кольорами відповідно зображено розширене сусідство в радіусі 2 і 3 від сірої клітини.



Рисунок 2.1 – Сусіди сірої клітинки в одновимірній сітці [21]

Клітинний автомат можна визначати, як множину кінцевих автоматів, кожен з яких може знаходитися в одному з можливих станів  $\sigma \in Z$ , де  $Z$  - множина станів кожної клітини [20, 22].

Зміна стану автоматів відбувається згідно правила переходу  $f$ :

$$Z \times Z^{\mathbb{N}} \rightarrow Z$$

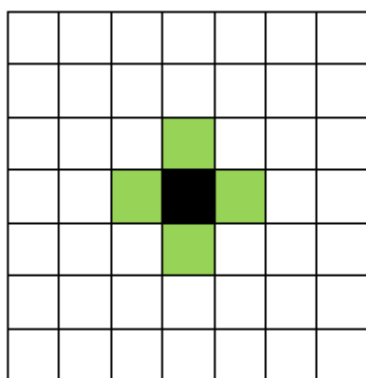
наступним чином:

$$\sigma_{i,j}(t+1) = f(\sigma_{k,l}(t), \sigma_{k,l}(t) \in N) \quad , (2.1)$$

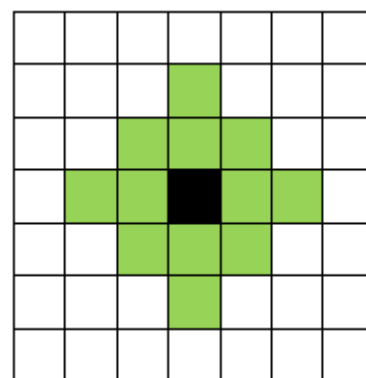
де  $N$  – множина автоматів, які складають отвір.

Отвір клітини складається з клітини ядра (центральної клітини) та оточуючих клітин, стани яких визначають її наступний стан. Радіус околу визначається як максимальна відстань від центральної клітини, горизонтально або вертикально, до сусідніх.

Для КА найчастіше використовують сусідство (отвір) Неймана та сусідство (отвір) Мура (див. рисунок 2.2 та рисунок 2.3)



а)



б)

Рисунок 2.2 - Приклади двовимірного околу фон Неймана: а) першого порядку; б) другого порядку [22]

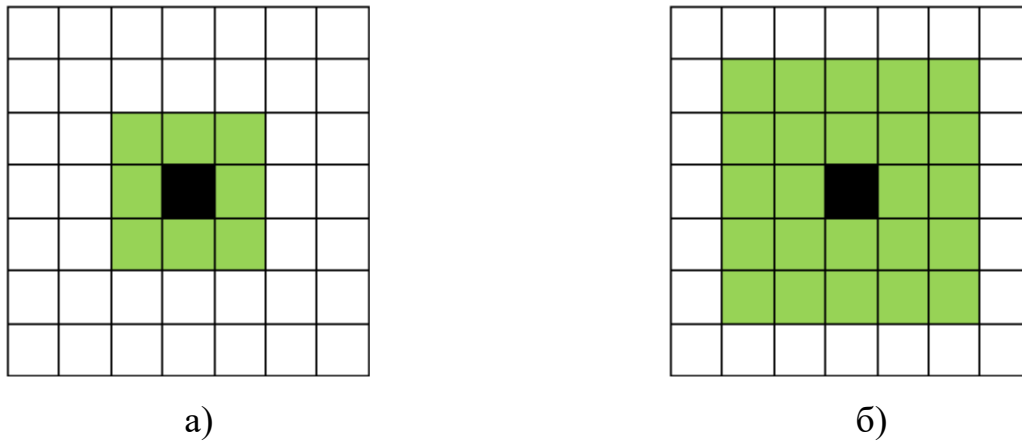


Рисунок 2.3 – Приклади двовимірного околу Мура: (а) – першого порядку; б) другого порядку [22]

За умови розширення запропонованої концепції до трьох вимірів, матимемо більше можливостей для вибору сусідства.

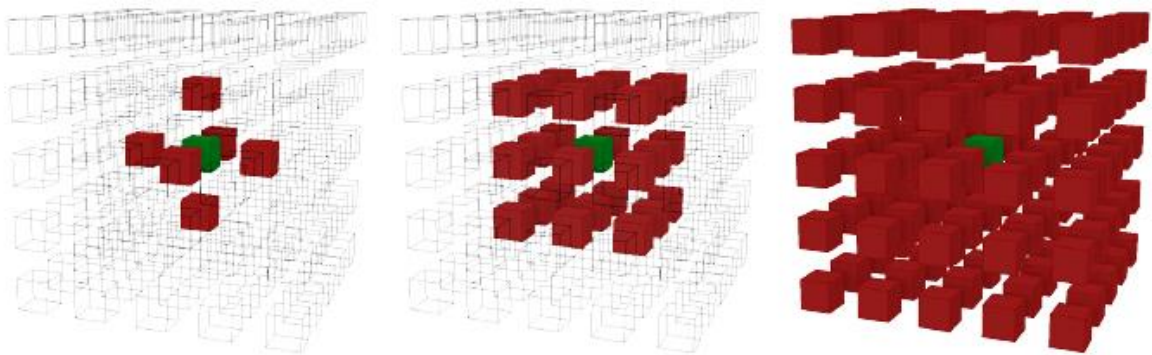


Рисунок 2.4 – Приклади сусідства Неймана-Мура для тривимірного КА [23]

Найбільш дослідженими досі залишаються двовимірні КА, зокрема гра «Життя» та мураха Ленгтона, а тривимірні КА є малодослідженими, попри те, що вони мають величезний потенціал для застосування. Причина у складності дослідження КА, зумовленій відсутністю належних відповідних програмних засобів.

Цифрове зображення розглядається як двовимірний масив пікселів  $M \times N$  (див. рисунок 2.5). Кожен піксель може бути охарактеризований впорядкованою трійкою  $(i, j, k)$ , де впорядкована пара  $(i, j)$  представляє його положення в масиві,  $k$  являє собою колір, пов'язаний з цим пікселем.

Зображення потім може розглядатися як особлива конфігурація КА, що займає клітинний простір масиву  $M \times N$ , який визначається зображенням. Кожен піксель зображення являє собою комірку КА і стан клітинки визначається величиною пікселя у зображенні. Архітектура КА залежить від розміру моделі, кількості кінцевих станів клітини, кількості сусідніх клітин та їх радіусу, граничних умов і перехідних функцій або правил [19].

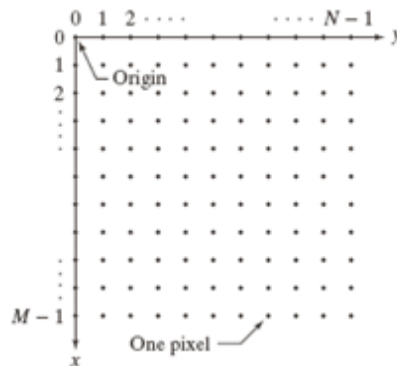


Рисунок 2.5 – Піксельне зображення цифрового зображення [19]

Граничними правилами називають значення, якими будуть заповнені клітини, котрі мають неповний набір сусідів.

Всього існує чотири наступних типи граничних умов:

- фіксована (крайні клітини з'єднанні з логічним станом нуль/один;
- періодична (крайні клітини прилягають одна до одної);
- адіабатична (крайні клітини реплікують цей стан);
- рефлексивна (дзеркальні стани замінюють крайніми клітинами), –

всі ці правила визначені в роботі [24].

Граничним правилом КА ще називають функцію, яка задовольняє цим вимогам. В КА необхідно синхронно оцінити кожну клітину в кожній ітерації КА й відповідно змінити сітку.

На рисунку 2.6 зображено поведінку простого правила.

Однією сірою клітинкою посередині (стан 1) ініціалізується кінцева одномірною сіткою. Для кожної ітерації клітинка стає сірою за умови, в її сусідстві існує одна або дві сірі клітинки, у противному випадку вона перетворюється, або залишається білою (стан 0).

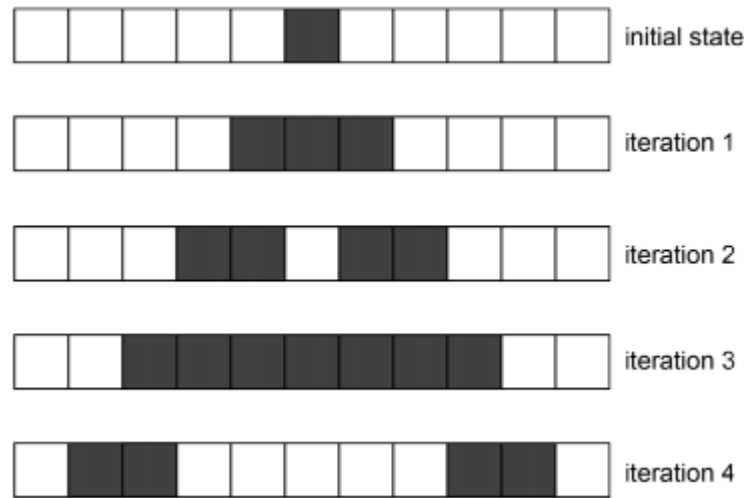


Рисунок 2.6 – Чотири ітерації правила ітерації елементарного КА на одновимірній сітці [21]

Розглянемо одновимірний КА, довжина якого п'ять клітин з наступними станами:  $x_1, x_2, x_3, x_4, x_5$ . Нехай сірі клітини – це крайні клітини сусідніх станів. Типи граничних умов будуть наступними (див. рисунок 2.7):

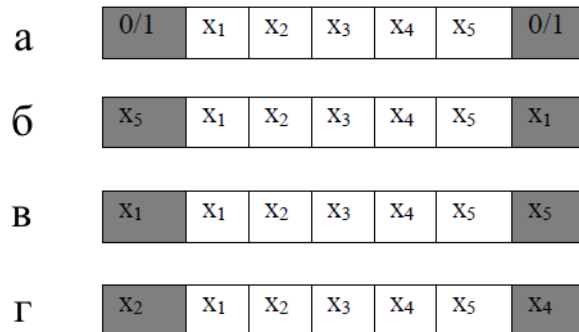


Рисунок 2.7 – Типи граничних умов:

а – фіксована; б – періодична; в – адіабатична; г – рефлексивна [24]

Кількість усіх можливих правил переходу визначається числом станів  $\sigma$  та кількістю сусідів  $m$ :

$$n = \sigma^{\sigma^m} \quad (2.2)$$

Зрозуміло, що загальна кількість правил переходу різко зростає навіть при досить малих  $\sigma$  та  $m$  (див. таблицю 2.1).

Таблиця 2.1 – Залежність кількості правил переходу від числа станів  $\sigma$  та кількості сусідів  $m$  [22]

Кількість станів	Кількість сусідів	$\sigma^{\sigma^m}$	Кількість правил
------------------	-------------------	---------------------	------------------



$\sigma$	$m$		$n$
2	2	$2^{2^2}$	16
2	3	$2^{2^3}$	256
2	5	$2^{2^5}$	4 296 967 296
2	10	$2^{2^{10}}$	$1,797 \cdot 10^{308}$
5	2	$5^{5^2}$	$2,98 \cdot 10^{17}$
5	3	$5^{5^3}$	$2,35 \cdot 10^{87}$
5	5	$5^{5^5}$	$1,91 \cdot 10^{2184}$
10	2	$10^{10^2}$	$10^{100}$
10	3	$10^{10^3}$	$10^{1000}$
10	5	$10^{10^5}$	$10^{100000}$

Для двовимірного КА з двома станами й сусідством Мура існує  $2^{512}$  правил і лише  $2^9$  серед них є лінійними, а віднявши базові та нульове правило, матимемо 502 лінійних правил [25].

Для класичних КА характеристичною властивістю є прогресування часу з дискретним кроком, однаковим у всіх точках простору; щодо кожної точки, то вона має динамічні зміни стану, яких є кінцеве число значень; більш детально критерії класичних КА мають наступний вигляд:

1. По завершенню обчислення нового стану кожної клітинки решітки відбувається одночасна зміна значень всіх клітин (у протилежному випадку в ході генерації структури наступного стану обчислення значно впливали б на час отримання результату).
2. Однорідність сітки (зауважимо, що на практиці сітка зазвичай є набором клітин, що зумовлює виникнення прикордонних ефектів, зокрема, клітини, які знаходяться на межі сітки, відрізнятимуться кількістю сусідів, цьому можна запобігти за допомогою періодичних граничних умов).
3. Наявність лише місцевих взаємодій (впливають на клітину виключно клітини сусідства).
4. Кількість станів комірки може бути порівняно великою, але скінченною

(це не суперечить тому, що клітина може зберігати числа з плаваючою точкою для розв'язання прикладних задач), – всі ці критерії визначено в роботі [26].

За умови, що з будь – якого початкового стану КА можна привести до довільної заданої конфігурації шляхом зміни значення загального вхідного параметра, його називають повним КА. У будь – який довільний, але фіксований момент часу, кожен елемент КА набуває одного якогось стану з усього можливого набору кінцевих станів. У залежності від цих станів, згодом набір елементів може набутися іншого стану. За умови, що елементи відрізняються кількістю можливих станів, КА називають полігенними.

Найширше практичне застосування здобули так звані лінійні КА, для яких характеристичною властивістю є еквівалентність наборів можливих станів для усіх клітин. Геометрично елементи КА можуть розташовуватися у різний спосіб, розмір простору, загалом кажучи, є довільним, а кількість елементів може бути як нескінченною, так і скінченною. Для останнього випадку вводиться додаткова міра свободи в граничних умовах. Практично використовують сталі у часі (найчастіше нульові) або періодичні граничні умови. [26].

Якщо для кожного елемента КА кількість входів і виходів однакова, то КА називають збалансованим. Для будь – якого довільного, але фіксованого КА, визначаються певні локальні правила та згідно цих правил змінюється у часі стан кожного елемента КА. Ті КА, для яких мають місце різні локальні правила для різних елементів, називаються неоднорідними. З часом, локальне правило може змінюватися або набувати випадкового характеру, проте воно встановлює взаємозв'язок між поточним станом кожного елемента (разом із сусідством) і станом цього елемента на майбутньому наступному етапі.

На рисунку 2.8 зображено типи КА та показано взаємозв'язки між ними.

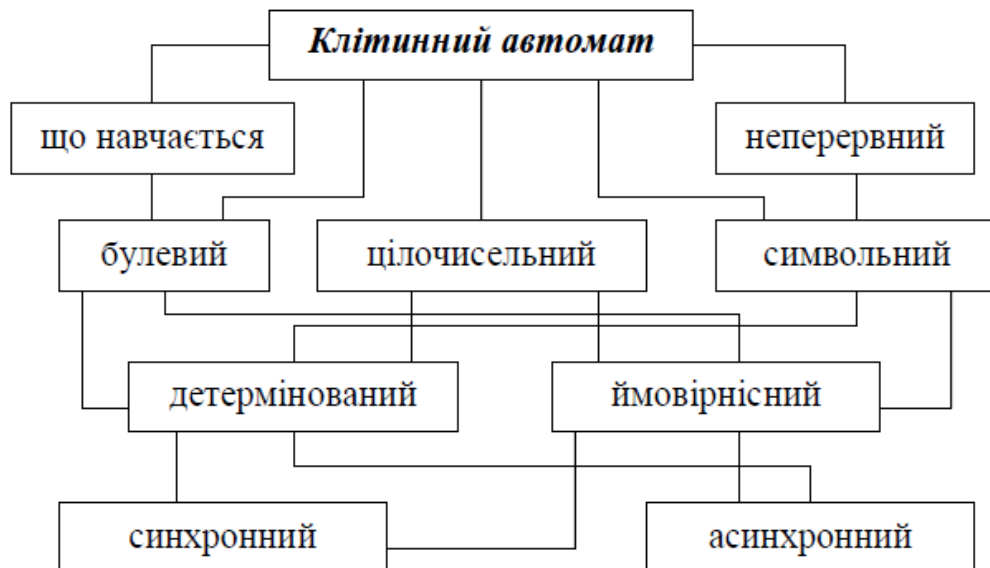


Рисунок 2.8 – Типи КА та взаємозв'язки між ними [26]

Переосмислення ідеї КА зумовило появу нових модифікацій КА. Зокрема, суть методу рухомих КА полягає у використанні пари взаємодіючих автоматів, що є підґрунтям для переходу від статичної сіткової концепції до концепції сусідів. Наступна ідея модифікації КА базується на представленні всіх дій, як випадкових процесів (дійсно, для них у природних умовах відсутня спеціальна синхронізація. За умов синхронного режиму роботи має місце послідовна детермінована схема взаємодії між усіма клітинами поля, впродовж одного ітераційного циклу. Дослідження асинхронних КА та композиції КА узагальнюють досвід побудови складних КА – моделей. Моделювання методом неперервних асинхронних КА ґрунтується на виборі певної клітини у випадковий спосіб. Зауважимо, що при цьому клітини вважаються рівноймовірними щодо їх вибору. Наступним кроком є випадковий рівноймовірний вибір сусідньої клітини (відповідно до схеми сусідства) та взаємодія між двома комірками КА [26].

### 2.1.2 Класифікація КА

З огляду на те, що задача класифікації КА залишається відкритою, є різні підходи щодо їх класифікації, зокрема мають місце наступні:

- у залежності від розмірності решітки, розрізняють одно-, дво-, тривимірні і т. д. КА (комірки поля мають форму трикутників, чотирикутників або шестикутників; в одновимірному (лінійному) КА ґрадкою є ланцюжок клітинок, кожна з яких окрім крайніх має по два сусіди, тому для усунення крайових ефектів цей одновимірний масив «згортають» у тор; для двовимірного масиву – аналогічно);
- у залежності від кількості станів КА бувають бінарні, трійкові і т. д.;
- у залежності від конфігурації КА розрізняють мозаїчні КА (в локальному правилі кожного елемента використовується стан самого елемента та стан його сусідства), ітеративні КА (значення вхідного параметру використовується для зміни стану лише одного елемента), односторонні КА (має місце лише одностороння взаємодія, – залежність стану лише від лівого або лише від правого сусіда), КА типу Л – системи (один елемент може бути видаленим із системи або заміненим на декілька елементів);
- у залежності від синхронізації оновлення клітин КА бувають синхронні та асинхронні (в першому випадку клітини оновлюються одночасно, у другому – кожна клітина робить це незалежно від інших).

Серед найбільш поширених класифікацій КА є наступні:

- за основу класифікації КА беруть використання автоматизованого знаходження локальних структур, виділяють три області: впорядковані системи, хаотичні системи, системи зі складною поведінкою [27];
- застосовують ймовірнісний метод класифікації, – виділяють відповідно наступні класи: строго сприятливі, сприятливі в середньому, нейтральні в середньому, несприятливі в середньому, строго несприятливі (класи виділяють на основі ймовірнісного переходу випадкової клітини у стан 1 на першому кроці еволюції при випадкових початкових умовах) [28];

- класифікація КА за типами їх еволюції, - Стівеном Вольфрамом виділено наступні чотири класи [3]:

- до першого класу віднесено КА, де еволюція системи закінчується переходом всіх клітин поля в однаковий стан. Результатом еволюції майже всіх початкових станів є швидка стабілізація стану та його гомогенність, при цьому будь-які випадкові конструкції в таких правилах швидко зникають;

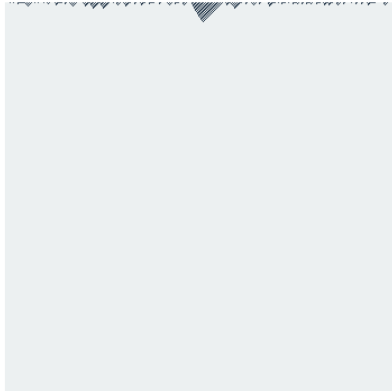


Рисунок 2.9 – Результат еволюції 1 класу КА на прикладі правила 40 [3]

- до другого класу віднесено КА, де існує багато кінцевих станів, проте всі вони складаються з набору простих структур, які залишаються незмінними або повторюються через певну невелику кількість кроків. Більшість випадкових структур у початкових умовах швидко зникає і залишаються лише деякі. Локальні зміни в початкових умовах мають локальний характер на подальший хід еволюції системи;

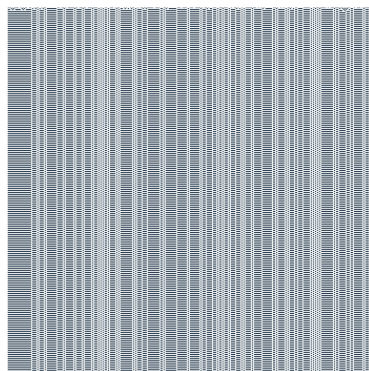


Рисунок 2.10 – Результат еволюції другого класу КА на прикладі правила 33 [3]

– до третього класу віднесено КА, де поведінка виглядає хаотичною і є складною. Структури змінюються з певним періодом і залежать від початкової конфігурації. Будь-які стабільні структури, що виникають, майже одразу знищуються шумом, що їх оточує. Локальні зміни в початкових умовах надають широкий, невизначений вплив на хід еволюції усієї системи;

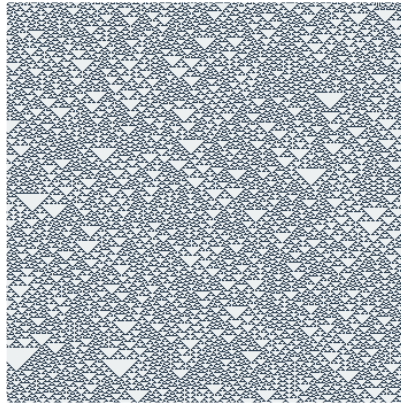


Рисунок 2.11 – Результат еволюції третього класу КА на прикладі правила 22 [3]

– до четвертого класу віднесено КА, де суміш хаосу і порядку: породжуються локальні структури, що переміщуються та взаємодіють між собою у дуже складний спосіб. Результатом еволюції майже всіх правил є структури, наслідком взаємодії яких є формування локальних стійких структур, які здатні виживати впродовж тривалого часу. В результаті еволюції правил цього класу можуть з'являтися деякі послідовності першого класу. Локальні зміни в початкових умовах мають широкий, невизначений вплив на хід усієї еволюції системи.

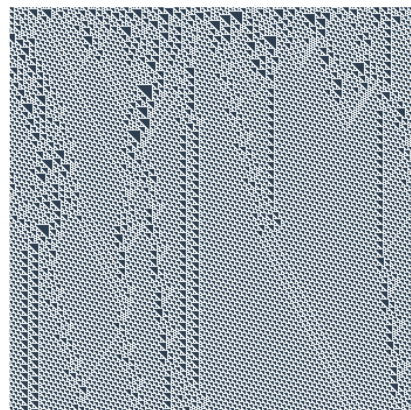


Рисунок 2.12 – Результат еволюції четвертого класу КА на прикладі правила 193 [3]

Досить складним є віднесення КА до якогось із класів, оскільки не вказано, за яких початкових умов очікується вказана вище поведінка. Здійснити класифікацію клітинного автомата за оптимальний проміжок часу практично неможливо у зв'язку з необхідністю виконання великого обсягу обчислень та за рахунок складного аналізу.

## 2.2 Діаграми Вороного як універсальна структура

Діаграма Вороного – це розбиття простору на області близькості – багатокутники Вороного [29], які ще називають багатокутниками або клітинками Діріхле, Тіссена, Вігнера – Гейтса, «багатокутниками близькості». Ці діаграми використав український математик Г. Вороний у своїй роботі [30] з квадратичних форм. Основою діаграм Вороного є опуклі багатокутники, оскільки кожен із них можна розглядати як переріз півплощин.

### 2.2.1 Визначення поняття діаграми Вороного

Нехай  $P = \{ p_1, p_2, \dots, p_N \}$  – множина  $N$  точок площини. Вважатимемо, що:

- 1) якщо  $p_i = p_j$ , то  $i = j$ ;
- 2) жодні три точки, які належать множині  $P$ , не належать одній прямій;
- 3) будь яке порожнє коло може проходити лише через три точки множини  $P$ .

Діаграма Вороного для множини  $P = \{ p_1, p_2, \dots, p_N \}$  – це розбиття площини на області близькості (клітинки або багатокутники Вороного) при



якому кожній точці  $p_i \in P$  ставиться у відповідність деякий унікальний багатокутник Вороного, що визначається для цієї точки  $p_i$ , яку називають генератором багатокутника Вороного. Цей багатокутник є підмножиною точок площини, відстань від яких до  $p_i$  не більша від відстані до будь якої іншої точки множини  $P$  [31].

За умови, що деяка точка площини є рівновіддаленою від двох генераторів  $p_i, p_j \in P$ , вважають, що вона належить до обох багатокутників Вороного, що відповідають цим точкам.

Отже, множина багатокутників Вороного є покриттям площини й кожна точка площини належить принаймні одному багатокутнику Вороного.

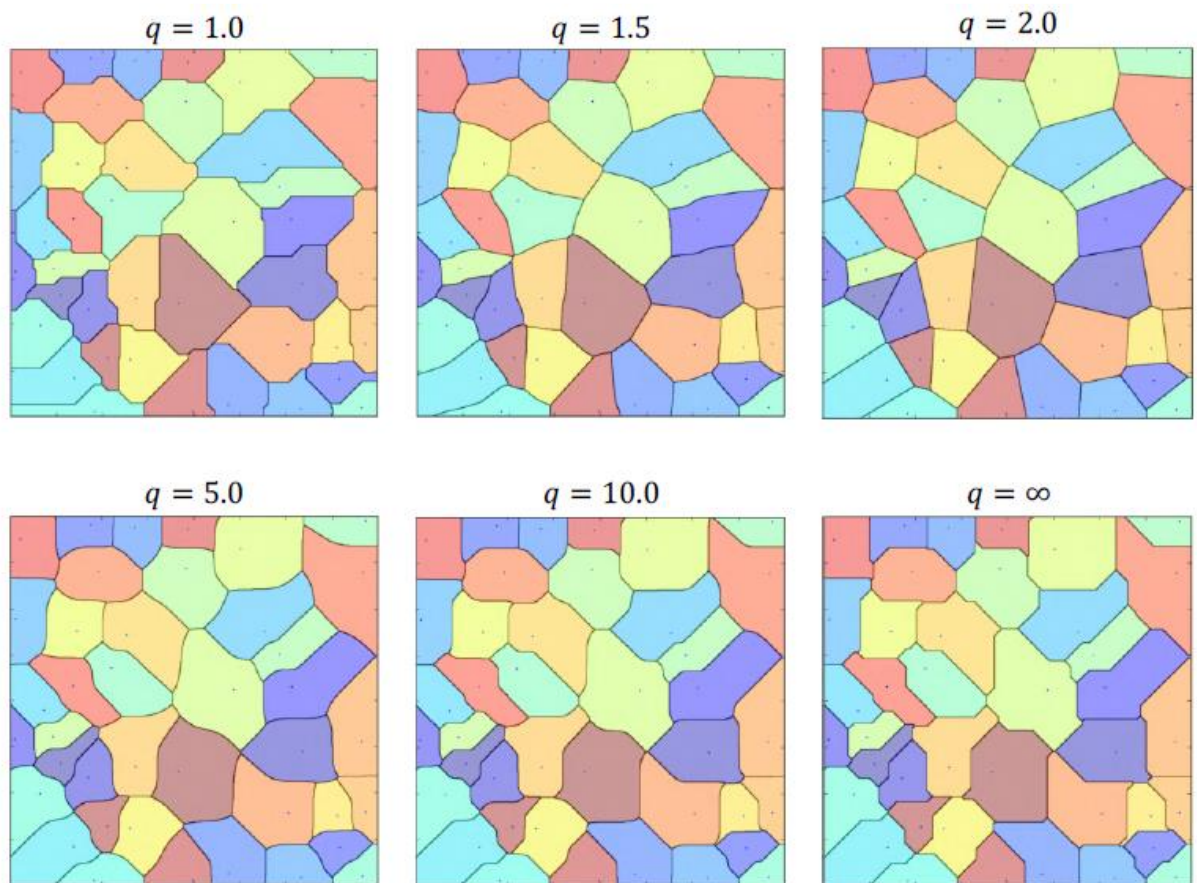


Рисунок 2.13 – Приклади діаграм Вороного у різних топологічних просторах з метрикою на основі Гельдера робота [32]

Згідно означення ДВ, багатокутник Вороного – це множина, яка містить усі точки своєї границі, тобто замкнена множина. Відрізки, промені,



прямі лінії, з яких складається границя багатокутника Вороного, називають ребрами діаграми Вороного.

Зауважимо, що іноді діаграмою Вороного називають об'єднання множини ребер Вороного. В свою чергу, ребро діаграми Вороного є перетином двох суміжних багатокутників Вороного, а кінці ребра Вороного, за умови, що вони існують, називаються вершинами Вороного. Як наслідок, діаграму Вороного може розглядатися як граф, що визначає границі багатокутників Вороного. В роботі [31] визначено критерій суміжності, а саме: для того, щоб багатокутники Вороного були суміжними, необхідно і достатньо, щоб їх перетин був непорожньою множиною.

Бісектором між двома точками  $p_i, p_j \in P$ , де  $p_i \neq p_j$  називають множину таких точок площини, які є рівновіддаленими від точок  $p_i$  та  $p_j$ . Отже, бісектор розділяє площину на деякі дві півплощини.

Областю домінування генератора  $p_i \in P$  над генератором  $p_j \in P$  називається така множина точок площини, відстань від яких до точки  $p_i$  не перевищує відстань до точки  $p_j$ .

Для випадку, коли розглядають діаграми Вороного для множини точок, бісектором є серединний перпендикуляр до відрізка, кінцями якого є точки  $p_i, p_j \in P$ . Цей серединний перпендикуляр розділяє площину на дві півплощини і областю домінування генератора  $p_i$  над генератором  $p_j$  є півплощина, яка містить точку  $p_i$ . Як наслідок, багатокутник (або клітинку) Вороного, що відповідає деякому генератору, можна розглядати як перетин областей його домінування. З огляду на те, що півплощина є необмеженою областю, то і багатокутник Вороного може бути необмеженим.

В роботі [31] доведено факт, що за умови розгляду множини точок площини, границя багатокутника Вороного може складатися з прямих ліній, променів або відрізків. Якщо ж розглядаються діаграми Вороного для множини відрізків на площині, то множина генераторів діаграми Вороного розширюється до множини  $P$  точок і множини  $L$  відрізків. Для цього випадку істинними вважаються наступні твердження:

- відрізки множини  $L$  не можуть перетинатися та дотикатися, виняток складають лише кінці відрізків;
- жодна точка множини  $P$  не належить відрізку множини  $L$ .

Зауважимо, що бісектор між відрізками може складатися з променів, параболічних дуг та відрізків.

Поняття багатокутника Вороного, бісектора та діаграми Вороного для множини кривих визначається аналогічно. Їх частинним випадком є діаграма Вороного для кіл, які характеризуються координатами відповідних центрів кіл  $P_c = \{ p_1, p_2, \dots, p_N \}$ , а також їх відповідними радіусами  $R_c = \{ r_1, r_2, \dots, r_N \}$ . Для цього випадку приймають за істину той факт, що будь-які два кола мають або різні центри, або різні радіуси.

### 2.2.2 Властивості діаграм Вороного

Розглянемо множину  $P = \{ p_1, p_2, \dots, p_N \}$  точок площини. В роботі [31] виділено наступні властивості діаграм Вороного:

- Багатокутники Вороного є непорожніми опуклими підмножинами площини і їх об'єднання повністю покриває площину;
- Багатокутник Вороного є необмеженим лише коли  $p_i$  належить границі опуклої оболонки множини  $P$  (це є необхідною та достатньою умовою).  
(Ця властивість встановлює взаємозв'язок між опуклою оболонкою та діаграмою Вороного для множини  $P$ . Границю опуклої оболонки будують за допомогою сполучення відрізками точок-генераторів, які відповідають необмеженим багатокутникам Вороного, за умови, що останні мають спільне ребро.);
- Необхідною та достатньою умовою того, що ребра діаграми Вороного будуть необмеженими прямими лініями є належність усіх точок множини  $P$  одній прямій;
- Ребро діаграми Вороного буде променем тоді і лише тоді, коли точки множини  $P$  не належать одній прямій та  $p_i, p_j \in P$  є сусідніми точками – генераторами, які належать границі опуклої оболонки;

- Ребро, яке породжується точками  $p_i, p_j \in P$ , буде відрізком тоді і лише тоді, коли відрізок з кінцями в цих точках, не належить границі опуклої оболонки.  
(З вищевказаних трьох властивостей випливає, що ребра діаграми Вороного складаються з частин бісекторів, але бісектори в свою чергу не завжди відповідають ребрам діаграм Вороного);
- Найближча точка до  $p_i \in P$  точка-генератор діаграми Вороного утворює ребро багатокутника Вороного (це достатня умова для утворення ребра діаграми Вороного);
- Найближча точка до  $p_i \in P$  точка-генератор діаграми Вороного належить такій підмножині точок, для яких відповідні їм багатокутники Вороного мають спільні ребра з багатокутником Вороного, що відповідає генератору  $p_i$ ;
- Генератор  $p_i \in P$  є найближчим генератором для деякої точки  $p$  площини тоді і лише тоді, коли багатокутник Вороного, що породжений генератором  $p_i$  містить точку  $p$ ;
- Кожній вершині  $q_i \in Q$  діаграми Вороного для множини  $P$  відповідає порожнє коло  $C_i$  з центром у точці  $q_i$ , яке проходить принаймні через три точки множини  $P$  та не містить будь-якої іншої внутрішньої точки цієї множини  $P$ . (Коло  $C_i$  є найбільшим порожнім колом серед усіх порожніх кіл з центром у точці  $q_i \in Q$ );
- Позначимо відповідно  $N, N_e, N_v, N_c$  число точок-генераторів, число ребер, число вершин та число необмежених ребер діаграми Вороного для точок множини  $P$  (зауважимо, що  $3 \leq N < \infty$ ). Тоді мають місце наступні рівності та співвідношення:

$$\begin{aligned}
 N_v - N_e + N &= 1, \\
 N_e &\leq 3N - 6, \\
 N_v &\leq 2N - 5, \\
 N_v &\geq \frac{1}{2}(N - N_c) + 1, \\
 N_e &\leq 3N_v - N_c - 3.
 \end{aligned}$$

З огляду на те, що кожне ребро діаграми Вороного належить точно двом багатокутникам Вороного, то середнє число ребер діаграми Вороного не перевищує  $2(3N - 6) / N$ , а середнє число ребер багатокутника Вороного не перевищує шести [31].

### 2.2.3 Алгоритми побудови ДВ

Серед основних алгоритмів побудови діаграм Вороного для точок, відрізків, кіл та дуг в роботі [31] (з відповідними посиланнями на джерела) вказано наступні:

- для множини точок (перетин півплощин, послідовна вставка точок, «Розділяй та володарюй»);
- для множини відрізків (алгоритм Фортуна, кінематичний (точки послідовно замінюються на відрізки), алгоритм Бає (відрізки можуть перетинатися), алгоритм Каравеласа (відрізки теж можуть перетинатися));
- для відрізків та дуг кола (алгоритм Япа);
- для множини кіл (алгоритм Шаріра (кола можуть перетинатися), алгоритм плоского замітання (кола теж можуть перетинатися)).

Для створення алгоритму щодо побудови ДВ необхідно ввести таку структуру даних, яка б забезпечувала швидкий обхід ребра грані, що відповідає багатокутнику Вороного (за або проти годинникової стрілки), а також слугувала обходу ребер ДВ навколо заданої вершини цієї діаграми (звісно також за або проти годинникової стрілки).

Реберний список з подвійними зв'язками, що містить інформацію про вершини, ребра та грані діаграми Вороного, а також зв'язки між ними у вигляді вказівників на відповідні елементи структури даних дає можливість успішної реалізації вищевказаних завдань [33].

Детальніше розглянемо алгоритм «Розділяй та володарюй». Нехай на площині дано множину точок  $P = \{ p_1, p_2, \dots, p_N \}$  ( $0 < N < \infty$ ). Оскільки вона є лінійно роздільною, то для побудови ДВ можна використати принцип «розділяй та володарюй» [34]. Суть полягає у реалізації наступних кроків:

1. Розділити множину  $P$  на дві підмножини  $P_1$  та  $P_2$ , які були б приблизно однакового розміру та задовольняли наступні умови:  
 $P_1 \cap P_2 = \emptyset, ||P_1| - |P_2|| = 1.$
2. Рекурсивно побудувати діаграми Вороного для цих підмножин  $P_1$  та  $P_2$ .
3. Для отримання діаграми Вороного для множини  $P$  виконати об'єднання одержаних у попередньому кроці діаграм.

Якщо припустити, що точки множини  $P$  відсортовані в якийсь спосіб (наприклад, за  $x$  – координатою), то перший крок алгоритму може бути виконаний за лінійний час. Нехай  $T(N)$  – загальний час виконання алгоритму, тоді реалізація другого кроку алгоритму займе  $2T(N/2)$  часу.

Нехай дано розбиття  $P_1, P_2$  множини  $P = P_1 \cup P_2$  таке, що  $P_1 \cap P_2 = \emptyset$ . Множина  $\sigma(P_1, P_2)$  складається з ребер діаграми Вороного, які є спільними для багатокутників Вороного, породжених відповідно  $p_i$  та  $p_j$  діаграми Вороного множини  $P$ , де  $p_i \in P_1, p_j \in P_2$ .

Ця множина  $\sigma(P_1, P_2)$  є множиною ребер деякого підграфа ДВ множини  $P$  і роботі [31] виділено наступні її властивості:

- $\sigma(P_1, P_2)$  складається з циклів і ланцюгів, причому вони не мають спільних ребер. За умови, що деякий ланцюг складається з одного ребра, це ребро є прямою лінією, в протилежному випадку два крайні ребра ланцюга є променями;
- За умови, що множини  $P_1, P_2$  є лінійно роздільними,  $\sigma(P_1, P_2)$  складається з єдиного ланцюга, який є монотонним.

Нехай маємо випадок, коли множина  $P$  розділена вертикальною прямою на підмножини  $P_1, P_2$ , тобто площина поділена на ліву і праву півплощини. Розширений алгоритм побудови ДВ складається з наступних кроків [31]:

0. Відсортувати точки множини  $P$  за  $x$  – координатою (попередня обробка).

1. Розподілити точки множини  $P$  за допомогою вертикальної прямої лінії на підмножини  $P_1$  і  $P_2$  приблизно однакового розміру, які задовольняють умовам:  $P_1 \cap P_2 = \emptyset, ||P_1| - |P_2|| = 1$ .
2. Рекурсивно побудувати діаграми Вороного для підмножин  $P_1$  і  $P_2$ .
3. Побудувати ланцюг  $\sigma(P_1, P_2)$ , що розділяє підмножини  $P_1$  і  $P_2$ .
4. Видалити всі ті ребра діаграми Вороного підмножини  $P_2$ , що знаходяться зліва від  $\sigma(P_1, P_2)$  та видалити всі ребра діаграми Вороного для підмножини  $P_1$ , які знаходяться справа від  $\sigma(P_1, P_2)$ . Як результат, отримуємо діаграму Вороного для множини точок  $P$ .

#### 2.2.4 Метод ЦДОВ як приклад застосування ДВ

Розглянемо як здійснюється автоматично сегментації напівтонових (чорно-білих) зображень за допомогою методу центральних дискретних діаграм (осей) Вороного (ЦДОВ).

Для побудови розділяючого ланцюга  $\sigma(P_1, P_2)$  необхідно буде визначити його початковий  $r_1$  та кінцевий  $r_2$  промені. Кожен з них є серединним перпендикуляром для деякого опорного відрізка. Задача виділення скелетного компонента структуру досліджуваного об'єкта складна. Основна проблема полягає в неоднозначності або навіть неможливості вибору найбільш доцільного рівня дискримінації, у відповідності до якого структурні елементи можна виділити на напівтоновому зображенні так, щоб спотворення форм і розмірів були мінімальними [35].

Використання оригінальних алгоритмів [36-38] дозволяє підвищити ефективність розв'язання цієї задачі. Суть цього алгоритму полягає у застосуванні центральних дискретних осей Вороного, зокрема: розглядають деяку множину  $S$ , яка складається з  $n$  маркованих точок площини, тобто  $S = (p_1, p_2, \dots, p_n)$ . У площині з кожною точкою  $p_i$  цієї множини визначені всі локалізації  $x$ , дуга якої ближче до  $p_i$ , ніж до будь-якої іншої точки  $p_j$  ( $i \neq j$ ) [35]. За умови, що такий поділ здійснено для всіх точок, то як наслідок отримаємо множину багатокутників  $P_1, P_2, \dots, P_n$  - діаграму Вороного -

Делоне. Для загального випадку в роботі [35] розглядається підмножина  $S_k$ , яка містить  $k$  точок (замість розгляду індивідуальних точок  $S$ ) і створюється така область  $P(S_k)$ , в якій усі локалізації в  $P(S_k)$  є ближчими до всіх точок  $d S_k$ , ніж до будь-якої точки, що не належить  $S_k$ ; зрозуміло, що при  $k = 1$ , отримуємо основну модель діаграми Делоне.

Суть алгоритму Делоне - Вороного полягає у використанні дискретних осей Вороного для граничних точок поділу на структурні елементи (див. рисунок 2.14).

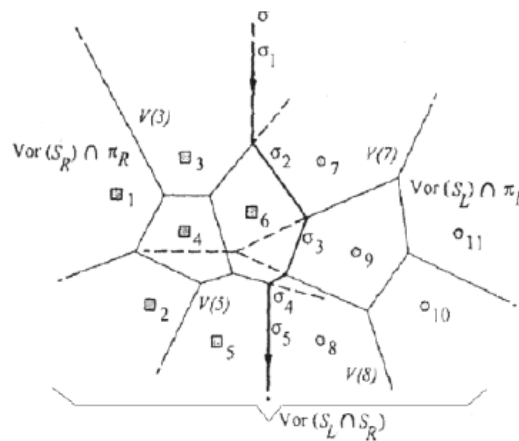


Рисунок 2.14 – Послідовність побудови діаграми Делоне – Вороного [35]

На рисунку зображено поділ дванадцяти точок зображення на лівий піднабір (це точки від 7 до 11) та правий (це точки від 1 до 6). Поділ здійснено відносно лінії розмежування та узгоджено з піддіаграмами Вороного  $Vor(S_L)$  і  $Vor(S_R)$ , які потім у свою чергу потрібно скласти і провести оцінку тих частин кожної з діаграм, які зазнали змін від прилеглих точок. За умови, що можливо розділити  $S_L$  і  $S_R$  прямою, то границя між піддіаграмами Вороного  $Vor(S_L)$  і  $Vor(S_R)$  складається з єдиного монотонного розділового ланцюга  $\sigma$  прямих лінійних сегментів. Кожен сегмент  $\sigma$  є підмножиною бісектора, яка містить дві вузлові точки, одна з яких належить  $Vor(S_L)$ , а інша належить  $Vor(S_R)$ .

В статті [59] описано алгоритм знаходження початкового та заключного елементів  $\sigma$ , алгоритм розв'язання задачі щодо виявлення «вхідних» та

«вихідних» сегментів  $\sigma$ , а також алгоритм пошуку перетинів  $\sigma$  із будь-яким  $Vor(S_L)$  і  $Vor(S_R)$ , в цій же статті описана процедура регуляризації, яка застосовується для зменшення впливу дрібних деталей границі розділюваного об'єкта на форму ЦДОВ.

### 2.3 Теоретичні основи написання алгоритмів на базі КА

Теоретичним підґрунтям для написання алгоритмів на основі КА є граничні правила.

Граничними правилами для КА називають значення, якими будуть заповнені клітини, котрі мають неповний набір сусідів. Всього є чотири типи граничних умов, а саме:

- а) фіксована (крайні клітини з'єднанні з логічним станом нуль/один);
- б) періодична (крайні клітини прилягають одна до одної);
- в) адіабатична (крайні клітини реплікують цей стан);
- г) рефлексивна (дзеркальні стани замінюють крайніми клітинами), – всі вони вказані в роботі [19].

За умови проведення дослідження, що ґрунтується на простих правилах з'являється дуже багато корисних конфігурацій. В роботі [21] вказано умови, яким повинна задовольняти структура відповідного КА, серед них наступні:

- можливість керування генерацією КА;
- зберігання стану КА;
- генерація наступної ітерації;
- відображення КА;
- керування станом КА.

На практиці КА розглядаються як інструмент, що відображає поведінку динамічної системи. Ця система в свою чергу визначається локальними взаємозв'язками. За умови розгляду КА, який підтримує ймовірнісну складову процесу розвитку, цю складову визначають так: клітина може зародитись лише за умови, що кількість її сусідів дорівнює випадково визначеному числу з певного діапазону. Для кожного окремого випадку,



виходячи із заданих меж значень (відповідно самотність і перенаселення), випадково визначаються границі загибелі клітин. В роботі [39] з метою обрахунку кожної наступної ітерації доцільним визначено використання наступних умов:

- зміни, що зумовлені переходом, не мають впливу на кінцевий результат;
- наявна можливість візуалізації поточного стану КА.

Якщо реалізовувати алгоритми, що задовольняють ці умови, то можуть виникати наступні ускладнення:

- необхідність візуалізації як поточного стану структури, так і будь – якого стану її розвитку;
- ускладнення розуміння структури поточної ітерації та її зміни при переході до наступної ітерації, що зумовлено великою кількістю клітин з декількома можливими станами;
- проблематичність у розпізнаванні кінцевого стану КА та ускладнення, щодо критеріїв, за якими стан вважається кінцевим, – всі ці ускладнення вказано в роботі [39].

#### ***2.4 Метод КА, його застосування та переваги***

Незважаючи на високий рівень розвитку сучасної комп'ютерної техніки, на теперішній час залишається багато практичних задач, вирішення яких виявляється дуже проблемним. До них належать задачі автоматизованого аналізу зображень. Це зумовлено перш за все складністю формалізації процесу сприйняття образів і зображень. Тому все ще немає «універсального» математичного чи технологічного підходу, який дозволяв би конструктивно розробляти методи та алгоритми, які б ефективно здійснювали процес. Проте, для деяких частинних випадків, коли математичні моделі виявляються відповідними для тієї чи іншої практичної задачі, вдається отримати гідні результати [40].

Застосування КА ще й досі залишається лише для вирішення конкретизованих задач. Навіть з огляду на тривалий період досліджень,

наразі загальної теорії КА не сформовано. Основним напрямком дослідження КА є алгоритмічна розв'язність окремих задач [19]. У наш час сліди КА можна знайти в різних галузях науки, а саме – від моделювання біологічної системи до створення віртуальної соціальної мережі. Основною проблемою при використанні КА для моделювання різних процесів є побудова локальних правил мікровзаємодії, які б слугували породженню відповідної макродинаміки [36]. Загальних методів розв'язання цієї проблеми не існує. КА досягли значних результатів у багатьох областях аналізу та обробки зображень, зокрема відновлення та розпізнавання образів та текстів, згладжування, контурування [19], фільтрація шумів [41,42]. Серед актуальних та цікавих конкретизованих задач для вирішення яких застосовують КА, є наступні: аналіз рентгенівських знімків, аналіз супутникових зображень, ідентифікація мінних полів [43,41]. КА можна розглядати як універсальну математичну модель, на використанні якої ґрунтується вирішення різноманітних задач. Зокрема, двовимірні КА здобули широке застосування для моделювання поведінки дорожнього трафіку, для симуляції гідродинамічних та газодинамічних течій [26]. Серед нерозв'язаних актуальних проблем слід вказати теорію про самовідновлення електричного кола [21].

Надзвичайно високий потенціал мають тривимірні КА, проте оскільки їх властивості ще недостатньо досліджені, а також відсутні зручні інструменти для дослідження й генерації таких КА, то реалізація цього потенціалу гальмується.

Перевагами КА перед традиційними методами обчислень є наступні:

- простота реалізації (встановлено, що системи, основою яких є КА, можуть бути легко впроваджені, оскільки кожна клітина зазвичай працює за декількома простими правилами);
- КА є одночасно паралельними та просто обчислювальними;
- КА є розширюваними (можемо розширити прості правила, використовуючи деякі нові методи обчислення);

- підтримує категорії  $n$ -розмірів та  $m$ -міток, де кількість міток не збільшує час або складність обчислень;
- КА є більш інтерактивними (користувачі можуть вносити виправлення та модифікації в будь-коли під час операцій), – всі ці переваги визначено в роботі [44].

## ***2.5 Висновки до розділу 2***

У цьому розділі дано дефінітивну характеристику понять «клітинний автомат» та «діаграма Вороного», розглянуто класифікацію КА, вказано властивості ДВ; наведено приклади алгоритмів побудови ДВ та застосування ДВ; описано метод КА, його застосування та переваги; розглянуто теоретичні основи написання алгоритмів на базі КА.

## РОЗДІЛ 3. РОЗРОБКА ПП НА ОСНОВІ КА З ВИКОРИСТАННЯМ ДВ

### 3.1 Виявлення контурів зображень на основі КА

Аналіз та обробка зображень і їх контурів за допомогою КА є перспективним напрямком. Його особливість полягає в розгляді зображень, як системи, що складається з простих компонентів, а поведінка кожного компонента змінюється згідно поведінки сусідів і їх попередньої поведінки. Завдяки заданню певних правил поведінки компонентів КА досягли значних результатів у різних областях аналізу й обробки зображень та їх контурів, проте застосування цих досліджень й досі залишається лише для конкретизованих задач [19].

Для розв'язання задачі щодо швидкісного аналізу зображень та їх контурів потрібна реалізація паралельних алгоритмів. КА, як модель паралельних обчислень, використовуються з цією метою. Для виявлення контурів метод на основі КА є більш ефективним у порівнянні з традиційними методами.. Виявлення контурів – це основний засіб при вирішенні задачі виявлення ознак. Суть ідеї виявлення контуру бінарного зображення на основі КА полягає у використанні найбільш придатних лінійних правил.

У випадку сусідства Мура оптимальними правилами, результативність яких співставна зі стандартними алгоритмами виявлення контурів, є правила №29, №113, №263 і №449 [19]. Ці правила називаються правилами граней (див. рисунок 3.1).

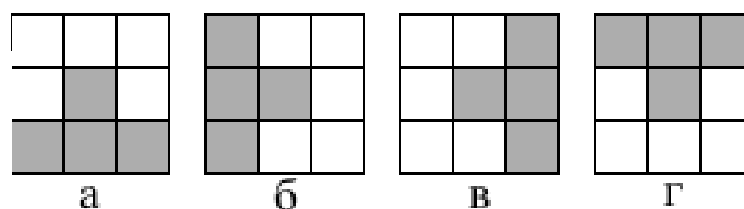


Рисунок 3.1 – Лінійні правила сусідства Мура для виявлення контурів:

а - №29, б - №113, в - №263, г - №449 [19]

У випадку розширеного сусідства Мура для виявлення контурів використовують правила №1040, №1088, №262176 [19] (див. рисунок 2.15).

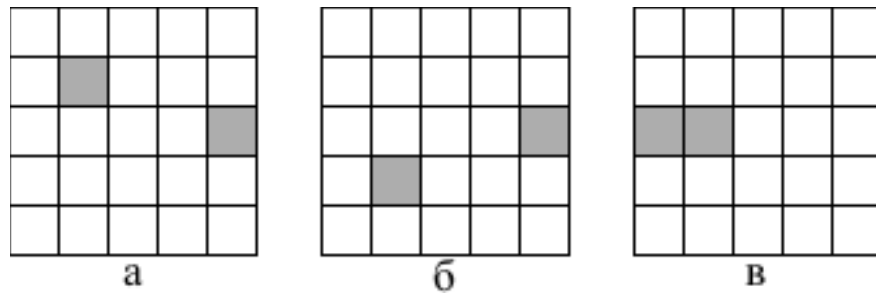


Рисунок 3.2 – Лінійні правила розширеного сусідства Мура для виявлення контурів а - №1088, б - №1040, в - №262176 [19]

### ***3.2 Порівняння результатів виявлення контурів на основі КА та традиційних методів***

При порівнянні результатів виявлення контурів на основі КА та за допомогою традиційних методів необхідно враховувати, що для традиційних методів виявлення контурів підґрунтям здебільшого є обчислення дискретного градієнта. Значення модуля цього градієнта обчислюється за допомогою операторів Робертса, Собеля, Лапласа. Назви вказаних операторів зумовили назви відповідних методів виявлення контурів. Ці методи різняться між собою, перш за все, своєю стійкості щодо шуму, зокрема, оператор Робертса та оператор Собеля більш стійкі до впливу шуму.

Рисунок 3.3 ілюструє порівняння результатів роботи виявлення контурів на основі КА та традиційних методів Робертса, Собеля-Фельдмана, Лапласа.



Рисунок 3.3 – Виявлення контурів зображення Lena алгоритмами:

а – КА правило №449, б – КА правило №1040, в – оператор Робертса, г – оператор Собеля - Фельдмана, д – оператор Лапласа [19]

Відзначаємо, що з огляду на контрастне підсилення, використання КА є кращим для визначення контурів зображень і як наслідок, результати є більш придатними для подальшого аналізу. Зауважимо, що одним зі шляхів покращення результатів є таке використання КА, за якого для різних областей зображення використовують одразу декілька правил. Перспективнішим бачиться шлях, суть якого полягає у використанні КА в поєднанні з іншими підходами, до прикладу з ДВ. Ось чому завершальним завданням курсової роботи є розробка та програмна реалізація алгоритму побудови ДВ на основі КА, з метою подальшого його використання для аналізу контурів зображень і реалізації за допомогою графічних карт.

### 3.3 Розробка на основі КА алгоритму побудови ДВ

Серед існуючих способів опису алгоритму, скористаємось словесно – формульним.

Нехай дано деяке зображення, що має відповідно розміри  $h$  – висота,  $w$  – ширина. Кожен піксель цього зображення розглядатимемо як клітинку (комірку) КА. Створимо клас клітинок, для якого визначимо наступні параметри:

- статус клітинки (може набувати значення 0, якщо клітинка звичайна та значення 1, якщо ця клітинка є центром сайту);
- відстань від цього пікселя до найближчого з центрів сайтів;

- колір пікселя;
- координата по  $x$  ;
- координата по  $y$ .

Запишемо початкові параметри кожної клітинки.

1. Початкове значення статусу буде 0, оскільки всі клітинки, за замовчуванням, є звичайними.
2. Відстань обчислюватимемо за формулою

$$d = \sqrt{h^2 + w^2} + 1 \quad (3.1)$$

Доцільність введення саме такої формули зумовлена наступним: ця відстань гарантовано має бути більшою від відстані між найбільш віддаленими клітинками на зображенні, якими є кінці найбільшої діагоналі.

3. Колір, за замовчуванням, приймемо рівним 0.
4. Координати відповідно по  $x$  та по  $y$ , за замовчуванням приймемо  $x_0, y_0$ .

Зауважимо, що при роботі з зображеннями використовується екранна система координат, початком якої є лівий верхній кут зображення, додатній напрям осі  $x$  – напрям вправо від початку координат, додатній напрям осі  $y$  – напрям донизу від початку координат.

Створимо двовимірний масив клітинок. Розмірами масиву відповідно є  $[h]$  та  $[w]$ . Назвемо цей масив `table`.

Забезпечуємо проходження по всьому масиву. На кожному етапі проходження значенню координати по  $x$  клітинки  $[i, j]$  присвоюється значення  $j$ , а значенню координати по  $y$  присвоюється значення  $i$ .

Визначаємося з кількістю центрів сайтів, їх координатами та кольором. При визначенні центру сайту, значення відповідної клітинки в масиві набуває значення центру сайту, а її параметр статусу змінюється з 0 на 1, тобто вона стає центром сайту.

Як наслідок, матимемо масив із зернистими вкрапленнями у вигляді клітинок, що є центрами відповідних сайтів.

Наступне – переходимо до поділу зображення на сайти.

Суть алгоритму побудови ДВ на основі КА полягає у наступному: при проходженні масиву беремо клітинку з координатами  $[k, m]$ , де  $k$  набуває всіх значень від  $0, 1, \dots$  до  $h - 1$  та  $m$  набуває всіх значень від  $0, 1, \dots$  до  $w - 1$  і здійснюємо перевірку кожної такої клітинки на залежність від кожної точки розглядуваного масиву.

При взаємодії розглядуваної клітинки з клітинкою масиву для зміни значень параметрів клітинки повинні виконуватись наступні умови:

- значення статусу клітинки  $[k, m]$  не дорівнює 1 (це зумовлене тим, що 1 є значенням статусу центру сайту, а також тим, що лише звичайні клітинки залежать від інших клітинок);
- значення статусу клітинки  $[i, j]$  має дорівнювати 1 (клітинка  $[i, j]$  – це та клітинка, з якою відбувається перевірка на взаємодію; щоб деяка клітинка залежала від іншої певної клітинки, ця певна клітинка має бути центром сайту);
- значення відстані між клітинками  $[k, m]$  та  $[i, j]$  має бути меншим від значення параметру відстані клітинки  $[k, m]$ .

Зрозуміло, що якщо хоч одна з цих трьох умов не виконується, то зміни значень параметрів клітинки не відбувається.

Отже, щоб клітинка змінювала своє значення в масиві, необхідним є виконання наступних умов:

- ця клітинка має бути звичайною (її статус має дорівнювати 0);
- клітинка, з якою ми перевіряємо на взаємодію, повинна бути центром сайту (її статус має дорівнювати 1);
- відстань між цими двома клітинками повинна бути меншою від параметра відстані цієї звичайної клітинки.

За умови, що всі ці необхідні умови виконуються, значення параметру кольору клітинки  $[k, m]$  присвоюється значення параметра кольору клітинки  $[i, j]$ .

Відстані до найближчого центру сайту клітинки  $[k, m]$  присвоїмо значення, що є відстанню між клітинками  $[k, m]$  та  $[i, j]$ .



### ***3.4 Програмна реалізація алгоритму на основі КА побудови ДВ***

#### ***3.4.1 Мова та середовище програмування***

З огляду на досить стрімкий розвиток обчислювальної техніки та її компонентів, для реалізації поставленої задачі курсової роботи доступною була велика кількість засобів. Найбільш доцільними для розробки ПП було обрано:

- об'єктно – орієнтовану мову програмування C ++ ;
- середовище Visual Studio 2019.

Мова програмування, як формальна знакова система, є способом передачі команд і використовується для визначення та маніпулювання структурами даних і управління процесом обчислень. Обрана мною мова для програмної реалізації створеного алгоритму – це мова C ++ , яка є універсальною мовою програмування високого рівня з підтримкою наступних засад програмування: об'єктно – орієнтованої, узагальненої та процедурної. Мова програмування C ++ надає можливість створювати узагальнені алгоритми для різних типів даних, а також є плідним підґрунтям для швидкого та ефективного створення програм. Серед переваг мови C ++ вкажемо наступні:

- можливість працювати з пам'яттю на низькому рівні;
- відсутність фреймворків;
- швидке виконання коду;
- повний доступ до API операційної системи.

Найбільш вагомими чиниками у виборі мови програмування C ++ для реалізації розробленого мною алгоритму є два наступних:

- 1) швидкісні характеристики мови, щодо виконання коду;
- 2) закладання основ для майбутнього дослідження, оскільки в подальшому планується реалізація алгоритму на графічних процесорах, де також використовується мова програмування C ++.

Інтегроване середовище розробки Visual Studio є основою для написання, редагування та збору коду, а також для наступної публікації застосувань. Окрім стандартного набору, Visual Studio містить компілятори, засоби автозавершення коду, графічні конструктори та багато інших функцій, які дають можливість спростити процес розробки. Застосування середовища Visual Studio 2019 має наступні зручності та переваги:

- є можливість переглядати файли коду, рухатися по них та керувати ними, а також є можливість впорядковувати код шляхом об'єднання файлів у розв'язки та проекти;
- у вікні редактору відображається вміст файлу (тут можна редагувати код);
- є можливість відслідковувати робочі елементи та використовувати код паралельно з іншими користувачами.

Вищевказані переваги стали переконливими аргументами щодо мого вибору середовища Visual Studio 2019.

Для реалізації візуалізації результатів роботи створеного алгоритму було обрано бібліотеку програмного забезпечення OpenCV для роботи із зображеннями. Ця бібліотека регулярно оновлюється та з огляду на те, що вона містить весь необхідний функціонал для створеного алгоритму, стосовно поточної його реалізації, бібліотека програмного забезпечення

OpenCV є такою, що ідеально підходить. Як аргумент зауважимо, що OpenCV , а також її основний інтерфейс написані мовою C ++; всі алгоритми та нові розробки OpenCV також у C ++ інтерфейсі.

### ***3.4.2 Етапи розробки програми***

Розподілимо опис розробки програми на декілька наступних кроків. Спочатку було складено та узгоджено вищеописаний алгоритм. Потім визначалася мова програмування та середовище програмування. Було проведено дослідження на предмет існування бібліотеки для створення програми і серед знайдених обрана оптимальна. Наступним кроком була

реалізація розробленого алгоритму з деякими спрощеннями, зокрема консольне виведення матриці значень після застосування до неї діаграм Вороного.

Завершальним етапом стала реалізація алгоритму за допомогою обраних технологій.

### 3.4.3 Тестування програми

Тестування програми відбувалося в режимі роботи із зображеннями у відтінках сірого. Обиралися наступні дані: розмір зображення, кількість центрів сайтів, координати центрів сайтів, кольори сайтів.

Здійснення перевірки на предмет того, чи потрібно змінювати колір пікселя:

```
for (int k = 0; k < rows ; k++)
{
    for (int m = 0; m < cols; m++)
    {
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols ; j++)
            {
                if (table[k][m]._status != 1)
                {
                    if (table[i][j]._status == 1)
                    {
                        localDistance = sqrt(((i - k) * (i - k) + (j - m) * (j - m)));

                        if (table[k][m]._distance > localDistance)
                        {
                            table[k][m]._color = table[i][j]._color;
                            table[k][m]._distance = localDistance;
                        }
                    }
                }
            }
        }
        newImage.data[k*cols+m] = static_cast<int>(table[k][m]._color);
    }
}
```

Тест 1. Визначаємося з розмірами зображення, кількістю центрів сайтів, їх координатами та кольором сайтів, а саме: розмір зображення – 400 x 400 пікселів, кількість сайтів – 3, відповідно їх координати (20, 10), (90, 80),

(150, 100) (зауважимо, що на першому місці координати по у, а на другому – по х), кольори сайтів наступні:

```
table[20][10]._status = 1;
table[20][10]._color = 60;
table[90][80]._status = 1;
table[90][80]._color = 90;
table[150][100]._status = 1;
table[150][100]._color = 190;
```

Результатом тестування стало зображення, поділене на 3 сайти (див. рисунок 3.4).

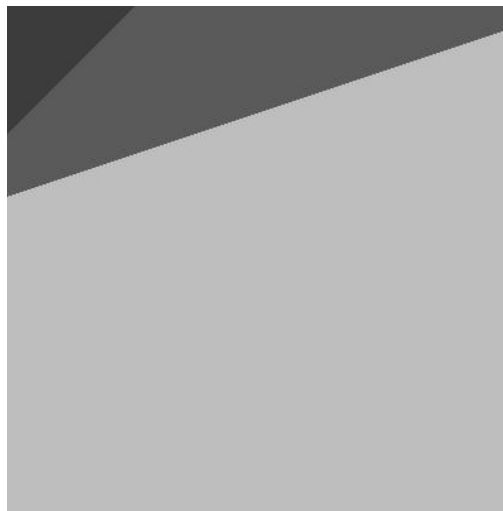


Рисунок 3.4 – Результати тестування програми на зображенні у відтінках сірого за умови трьох центрів сайтів

Тест 2. Визначаємося з розмірами зображення, кількістю центрів сайтів, їх координатами та кольором сайтів, а саме: розмір зображення – 500 х 500 пікселів, центрів сайтів – 4, відповідно їх координати (20, 10), (90, 80), (150, 100), (40, 150) (на першому місці координати по у, а на другому – по х), кольори сайтів наступні:

```
table[20][10]._status = 1;
table[20][10]._color = 60;
table[90][80]._status = 1;
table[90][80]._color = 90;
table[150][100]._status = 1;
table[150][100]._color = 190;
table[40][150]._status = 1;
table[40][150]._color = 230;
```

Результатом тестування 2 стало зображення, поділене на 4 сайти (див. рисунок 3.5).



Рисунок 3.5 – Результати тестування програми на зображенні у відтінках сірого за умови чотирьох центрів сайтів

Тест 3. Визначаємося з розмірами зображення, кількістю центрів сайтів, їх координатами та кольором сайтів, а саме: розмір зображення – 500 x 600 пікселів, центрів сайтів – 5, відповідно їх координати: (20, 10), (90, 80), (150, 100), (40, 150), (300, 450) (на першому місці координати по у, а на другому – по х), кольори сайтів наступні:

```
table[20][10]._status = 1;
table[20][10]._color = 60;
table[90][80]._status = 1;
table[90][80]._color = 90;
table[150][100]._status = 1;
table[150][100]._color = 190;
table[40][150]._status = 1;
table[40][150]._color = 230;
table[300][450]._status = 1;
table[300][450]._color = 160;
```

Результатом тестування 3 стало зображення, поділене на 5 сайтів (див. рисунок 3.6).



Рисунок 3.6 – Результати тестування програми на зображенні у відтінках сірого за умови п'яти центрів сайтів

Тест 4. Визначаємося з розмірами зображення, кількістю центрів сайтів, їх координатами та кольором сайтів, а саме: розмір зображення – 600 x 500 пікселів, кількість центрів сайтів – 6, відповідно їх координати (20, 10), (90, 80), (150, 100), (40, 150), (300, 450), (400, 200) (на першому місці координати по y, а на другому – по x), кольори сайтів наступні:

```
table[20][10]._status = 1;
table[20][10]._color = 60;
table[90][80]._status = 1;
table[90][80]._color = 90;
table[150][100]._status = 1;
table[150][100]._color = 190;
table[40][150]._status = 1;
table[40][150]._color = 230;
table[300][450]._status = 1;
table[300][450]._color = 160;
table[400][200]._status = 1;
table[400][200]._color = 120;
```

Результатом тестування 3 стало зображення, поділене на 6 сайтів (див. рисунок 3.7).

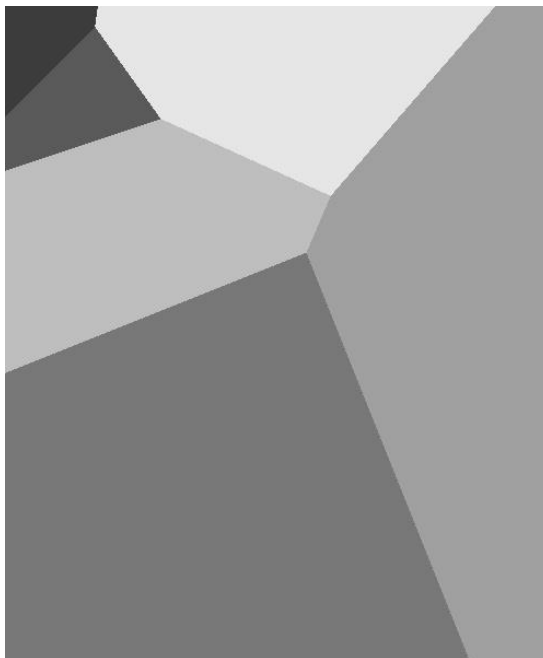


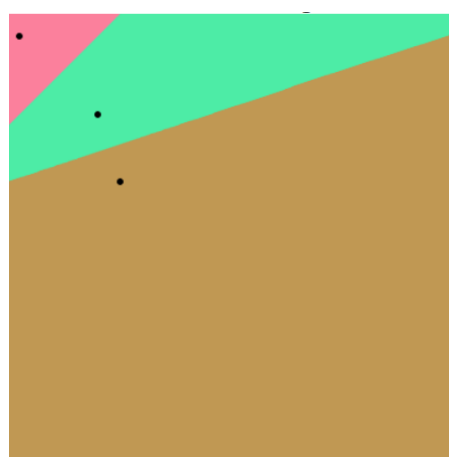
Рисунок 3.7 – Результати тестування програми на зображенні у відтінках сірого за умови шести центрів сайтів

#### ***3.4.4 Порівняння одержуваних результатів побудови ДВ***

Для порівняння результатів побудови ДВ на основі КА було обрано результати, що отримуються за допомогою традиційних (класичних) методів [45]. Рисунок 3.8, 3.9 ілюструють це порівняння.



а



б

Рисунок 3.8 – Побудовані ДВ для трьох центрів алгоритмами:

а – КА, б – традиційні [45]

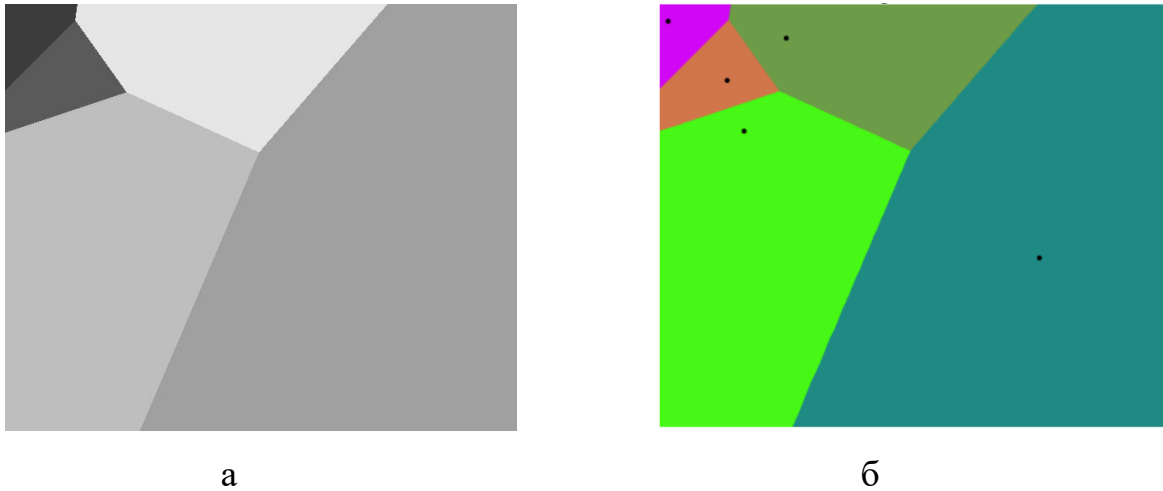


Рисунок 3.9 – Побудовані ДВ для п'яти центрів алгоритмами:

а – КА, б – традиційні [45]

Одержані результати є свідченням того, що розроблений алгоритм та його програмна реалізація є вірними. Відзначимо, що як видно з порівняння, якість одержуваних результатів є співставною. Перевага ДВ, побудованих на основі КА, полягає в можливості такого подальшого їх застосування для виявлення контурів зображень, яке слугує поліпшенню результатів, а також у тому, що вони можуть слугувати плідним підґрунтям для подальшої реалізації за допомогою графічних карт.

Отже, завершальне завдання курсової роботи виконано.

### ***3.5 Висновки до розділу 3***

У цьому розділі розглянуто виявлення контурів зображення на основі КА, порівняно результати виявлення контурів на основі КА й традиційних методів. На основі цього було вказано можливі шляхи покращення одержуваних результатів виявлення контурів зображень на основі КА. В розділі розроблено алгоритм побудови ДВ на основі КА та здійснено програмну реалізацію цього алгоритму.



## ВИСНОВКИ ПО РОБОТІ

В роботі розглянуто основну задачу аналізу зображень і методи розпізнавання, здійснено огляд алгоритмів виявлення, проходження та опису контурів зображень, проведено порівняльну характеристику переваг та недоліків цих алгоритмів; дано дефінітивну характеристику понять «клітинний автомат» та «діаграма Вороного», розглянуто класифікацію КА, вказано властивості ДВ, розглянуто алгоритми побудови ДВ, наведено приклад їх застосування. Також в роботі розглянуто метод КА, його застосування та вказано переваги методу КА перед традиційними методами дослідження. З огляду на існуючі теоретичні основи написання алгоритмів на базі КА розглянуто виявлення контурів та співставлено отримувані результати, вказано шляхи удосконалення алгоритмів на основі КА для виявлення контурів зображень.

Розроблено алгоритм побудови ДВ на основі КА та здійснено програмну реалізацію цього алгоритму.

## СПИСОК ЛІТЕРАТУРИ

1. Чичварин Н. В. Предварительная обработка изображений [Электронный ресурс]. – Режим доступа : <http://ru.bmstu.wiki/>
2. Rosin, P. L., Adamatzky, A., & Xianfang, Sun. (2014). *Cellular Automata in Image Processing and Geometry*. Springer International Publishing, 2014. – 304 p.
3. Wolfram S. A New Kind of Science. / S. Wolfram. – Wolfram Media. Inc., 2002. – 1280 p.
4. Березький О. М. Аналіз і синтез зображень на основі теорії алгебро-топологічних структур : автореф. дис. на здобуття наук. ступеня докт. техн. наук : спец. 05.13.23 «Системи та засоби штучного інтелекту» / О. М. Березький. – Львів, 2012. – 38 с.
5. Селянкин В.В., Анализ и обработка изображений в задачах компьютерного зрения: учебное пособие / В. В. Селянкин, С. В. Скороход. – Таганрог : Изд-во ЮФУ, 2015. – 82 с.
6. Шапиро Л. Компьютерное зрение: Пер. с англ. / Л. Шапиро, Дж. Стокман. – М. : Бином. Лаборатория знаний, 2006. – 752 с.
7. Фурман Я. А. Введение в контурный анализ. Приложения к обработке изображений и сигналов / Я. А. Фурман, А. В. Кревецкий, А. К. Передреев, А. А. Роженцов, Р. Г. Хафизов, И. Л. Егошина, А. Н. Леухин; Под ред. Я. А. Фурмана. – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2003. – 592 с.
8. Pratt W. K. Digital Image Processing: PIKS Inside, Third Edition. William K. Pratt / John Wiley and Sons, Inc., New York, 2001. – 736 p.
9. Hlavac V. Image Processing, Analysis, and Machine Vision / V. Hlavac, M. Sonka, R. Boyle // International Student Edition. – Thomson Learning, part of the Thomson Corporation, 2008. – 829 p.
10. Jian Wu. Skeleton Extraction of Cerebrovascular Image Based on Topological Nodes // Jian Wu, Guangming Zh., Jie X. – Proceedings of the 2009

International Symposium on Information Processing (ISIP'09) – Huangshan, P. R. China, – August 21–23. – 2009. – pp. 159–162.

11. Kiyko V. M. Recognition of objects in images of paper based line drawings / V. M. Kiyko // Third Intern. Conf. on Document Analysis and Recognition. – Monreal, 1995. – pp. 970–973.

12. Ramesh B. V. Crossing Edge Detection and Contour Tracing for Segmentation of Cervical Cell Nucleus // B. V. Ramesh, S. Raghunandan, K. R. Ramakrishnan // Defense Science Journal. – 1993. – Vol 43. - No 3. July. – pp. 223 – 228.

13. Батько Ю.М. Метод і алгоритми сегментації біомедичних зображень на основі попередніх розміток / Ю. М. Батько // Штучний інтелект. – 2010. – №4. – С. 140–149.

14. Попова Г. М. Анализ и обработка изображений медико-биологических объектов / Г. М. Попова, В. Н. Степанов // Автоматика и телемеханика. – 2004. - №1. – С. 131-142.

15. Березький О., Аналіз алгоритмів виділення контурів зображень / О. Березький, Ю. Батько [Електронний ресурс]. – Режим доступу : [http://ena.lp.edu.ua:8080/bitstream/ntb/36139/1/34\\_212-216.pdf](http://ena.lp.edu.ua:8080/bitstream/ntb/36139/1/34_212-216.pdf)

16. Contour Tracing Algorithms [Електронний ресурс]. – Режим доступу : [http:// www.cs.mcgill.ca/~aghnei/alg.html](http://www.cs.mcgill.ca/~aghnei/alg.html)

17. Theo Pavlidis, Algorithms for Graphics and Image Processing. – 1982, chapter 7, section 5.

18. Батько Ю. М. Аналіз цитологічних зображень в системах автоматизованої мікроскопії : дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.23 «Системи та засоби штучного інтелекту» / Ю. М. Батько. – Тернопіль, 2013. – 186 с.

19. Жежерун О. П. Використання клітинних автоматів для вирішення задач фільтрації шумів та виявлення контурів зображень / О. П. Жежерун, Б. П. Калітовський // Наукові записки НаУКМА. Комп'ютерні науки. – 2019. – Том 2. – С. 66 – 72.

20. Тоффоли Т. Машины клеточных автоматов / Т. Тоффоли, Н. Марголюс / Перевод с англ. – М. : Мир, 1991. – 280 с.
21. Ванін В. В. Структура тривимірного клітинного автомату для побудови зображення динамічних систем / В. В. Ванін, О. В. Залевська, В. О. Чередниченко [Електронний ресурс] // Modern Problems of Modeling, 2019. - №15. – с. 43-50. – Режим доступу: <https://doi.org/10.33842/2313-125X/2019/15/43/150>
22. Hoekstra A.G., Kroc J., Sloot P. (2010). Simulating complex systems by cellular automata. London: Springer, 235 p.
23. Panagiotis A. Mikedakis. Two and Three-Dimensional Cellular Automata for the Generation of Objects in Computer Graphics. Bachelor Thesis. Athens, 2017. – 70 p.
24. Mohammed, J., & Deepak, R. N. (2010). An Efficient Edge Detection Technique by Two Dimensional Rectangular Cellular Automata [Електронний ресурс]. – Режим доступу: <https://arxiv.org/ftp/arxiv/papers/1312/1312.6370.pdf>
25. Choudhury, P. P., Nayak, B. K., Sahoo, S., & Rath S. P. (2008). Theory and Applications of Two-dimensional, Null boundary, Nine neighborhood, Cellular Automata Linear rules [Електронний ресурс]. – Режим доступу: <https://arxiv.org/ftp/arxiv/papers/0804/0804.2346.pdf>
26. Шумиляк Л. М. Моделювання процесів кристалізації сплавів методом неперервних клітинних автоматів : дис. на здобуття наук. ступеня канд. техн. наук : спец. 01.05.02 «Математичне моделювання та обчислювальні методи. Технічні науки» / Л. М. Шумиляк. – Чернівці, 2018. – 164 с.
27. Wuensche A. Classifying Cellular Automata Automatically / Wuensche A. COMPLEXITY. – 1999. – №4. – pp. 47–66.
28. Лебедев А. В. Вероятностные методы классификации клеточных автоматов / А. В. Лебедев // Фундаментальная и прикладная математика. Открытые системы. – 2002. – № 2. – С. 621–626.

29. Voronoi G. Recherches sur les paralleloèdres primitifs. Journal für die Reine und Angewandte Mathematik, Berlin, 1908. – Vol. 134. – P. 198-287.

30. Voronoi G. Nouvelles applications des paramètres continus à théorie des formes quadratiques. Deuxième Mémoire. Journal für die reine und angewandte Mathematik, 1909. – Vol. 133. – P. 67-182.

31. Коцур Д. В. Побудова моделі єдиного алгоритмічного середовища на основі діаграми Вороного для розв'язування комплексу задач обробки зображень: дис. на здобуття наук. ступеня канд. фіз.-мат. наук : спец. 01.05.01 «Теоретичні основи інформатики та кібернетики» / Д. В. Коцур, Київ. нац. ун-т ім. Т. Шевченка. – Київ, 2019. – 200 с.

32. Коцур Д. В. Алгоритм побудови діаграми Вороного на двовимірній сітці з  $L_p$  нормою / Д. В. Коцур // Вісник Київського національного університету імені Тараса Шевченка. – Серія: фізико-математичні науки. – 2017. – Вип. 4. – С. 95-98.

33. Preparata F., Shamos M. Computational Geometry: An Introduction. Berlin, 1985. – 390 p.

34. Shamos M., Hoey D. Closest-point problems. Foundations of Computer Science: Proceedings of 16th Annual Symposium, Berkley, CA, USA, 13 October – 15 October, 1975. Washington, DC, USA: IEEE, 1975. - P. 131-162.

35. Мельник В.М. Автоматична сегментація напівтонових зображень методом ЦДОВ / В. М. Мельник, Н. В. Муляр// Science and Education a New Dimension. Natural and Technical Sciences, VI (19), Issue: 171, 2018. Режим доступу: <https://seanewdim.com/uploads/3/4/5/1/34511564/httpsdoi.org10.31174seand-nt2018-171vi19-09.pdf>

36. Ogniewicz R. Discrete Voronoi Skeletons / R. Ogniewicz. – Zürich, 1992. – P. 208.

37. Соколов В. Н. Автоматизированная система морфологического анализа скелетного компонента микроструктуры по РЭМ-изображениям / В. Н. Соколов, Д. И. Юрковец, О. В. Разгулина, В. Н. Мельник //

Поверхность. Рентгеновские, синхротронные и нейтронные исследования. – 2002. – № 10. – С. 66–69.

38. Соколов В. Н. Программно-аппаратный комплекс для исследования микроморфологии поверхности твердых тел по РЭМ-изображениям / В. Н. Соколов, Д. И. Юрковец, О. В. Разгулина, В. Н. Мельник // Поверхность. Рентгеновские, синхротронные и нейтронные исследования. – 1998. – № 1. – С. 33–41.

39. Ванін В. В. Візуалізація розвитку динамічної системи за допомогою тривимірних клітинних автоматів / В. В. Ванін, О. В. Залевська, В. О. Чередніченко // Сучасні проблеми моделювання. – 2019. – Вип. 14. - С. 48-53. - Режим доступу: [http://nbuv.gov.ua/UJRN/cpm\\_2019\\_14\\_7](http://nbuv.gov.ua/UJRN/cpm_2019_14_7)

40. Методы распознавания образов и анализа изображений: учеб. пособие / [Глумов Н. И., Коломиец Э. И., Мясников В. В., Сергеев В. В.]. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2013. – 142 с.

41. Peer, M. A., Fasel, Qadir, & Khan, K. A. (2012). Investigations of Cellular Automata Game of Life Rules for Noise Filtering and Edge Detection. *I. J. Information Engineering and Electronic Business*, 2, 22–28.

42. Selvapeter, P. J., & Wim, Hordijk (2009). Cellular Automata for Image Noise Filtering. In *World Congress on Nature Biologically Inspired Computing*, 193– 197.

43. Большаков А. А. Использование клеточных автоматов для обработки изображений минных полей / А. А. Большаков, Н. С. Булдаков // Вестник Саратовского государственного технического университета. – 2010. – №2. – С. 120–124. – Режим доступу: <http://cyberleninka.ru/article/n/ispolzovanie-kletochnyh-avtomatov-dlyaobrabotki-izobrazheniy-minnyh-poley>

44. Shukla, A. P. (2016). Training Cellular Automata for Image Edge Detection. *Romanian Journal of Information Science and Technology*, 19, 4, P. 338–359.

45. Beutel Alex. Interactive Voronoi Diagram Generator with WebGL. -

Режим доступа: <http://alexbeutel.com/webgl/voronoi.html>