

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**  
**Тернопільський національний економічний університет**  
**Навчально-науковий інститут інноваційних освітніх технологій**  
Кафедра комп'ютерної інженерії

Вихиль Юрій Олегович

**Алгоритми порівняння зображень в метриці Фреше  
для незважених дерев / Algorithms for the images  
comparing in the Frechet metric for unweighted trees**

спеціальність:123 – Комп'ютерна інженерія  
освітньо-професійна програма – Комп'ютерна інженерія

Випускна кваліфікаційна робота

Виконав студент групи КІзм-21  
Юрій Олегович Вихиль

---

Науковий керівник:  
д.т.н., професор О.М. Березький

---

**ТЕРНОПІЛЬ - 2019**

## РЕЗЮМЕ

Випускна кваліфікаційна робота на тему «Алгоритми порівняння зображень в метриці Фреше для незважених дерев» зі спеціальності 123 «Комп'ютерна інженерія» написана обсягом 86 сторінок і містить 9 рисунків, 9 таблиць, 3 додатки та 51 джерело за переліком посилань.

Метою роботи є розроблення алгоритмів порівняння зображень в метриці Фреше для незважених дерев.

Методи досліджень: методи цифрової обробки зображень, методи комп'ютерного моделювання.

Результати дослідження: алгоритми порівняння зображень в метриці Фреше для незважених дерев, програмний модуль порівняння зображень.

Результати роботи можуть бути використані у системах автоматизованої мікроскопії для порівняння зображень.

Орієнтовні напрямки розвитку досліджень: розроблення інтелектуальних систем для діагностики онкологічних захворювань.

**КЛЮЧОВІ СЛОВА:** ЗОБРАЖЕННЯ, МЕТРИКА ФРЕШЕ, МЕТРИЧНІ ПРОСТОРИ, СИСТЕМИ АВТОМАТИЗОВАНОЇ МІКРОСКОПІЇ.

## RESUME

The qualification graduation thesis on «Algorithms for the images comparing in the Frechet metric for unweighted trees» in the specialty 123 «Computer engineering» consists of 86 pages and contains 9 illustrations, 9 tables, 3 appendices and 51 references in the list of links.

The purpose of the work is to develop algorithms for comparing images in the Frechet metric  $c$  for unweighted trees.

Research Methods: Digital Imaging Techniques, Computer Modeling Techniques.

Results of the research: algorithms for comparison of images in the Frechet metric for unweighted trees, software module for comparison of images.

The results can be used in automated microscopy systems to compare images.

Approximate directions of research development: development of intelligent systems for the diagnosis of cancer.

KEYWORDS: IMAGES, FRECHET METRIC, METRIC SPACES, AUTOMATED MICROSCOPY SYSTEMS.

## ЗМІСТ

Вступ.....	7
1 Аналіз метрик порівняння зображень .....	9
1.1 Сегментація зображень та методи оцінки її якості .....	9
1.2 Аналіз моделей опису кольорових зображень .....	11
1.3 Метрики порівняння зображень .....	14
1.4 Відстань між параметризованими кривими .....	22
1.5 Експериментальні дослідження алгоритмів сегментації.....	30
2 Алгоритми визначення відстані між контурами в метриці Фреше .....	38
2.1 Алгоритми визначення відстані між ланцюжками .....	39
2.2 Алгоритми визначення відстані між циклами .....	40
2.3 Алгоритми визначення відстані між ламаними лініями .....	42
2.4 Алгоритми визначення відстані між багатокутниками .....	45
2.5 Алгоритми визначення дискретної відстані між контурами .....	53
3 Програмний модуль обчислення відстані між зображеннями.. ..	60
3.1 Системні вимоги та структура програмного модуля.....	60
3.2 Вибірка полігональних об'єктів.....	67
3.3 Порівняльний аналіз результатів .....	71
Висновки.....	75
Список використаних джерел.....	76
Додаток А. Світлокопії публікацій.....	81
Додаток Б. Довідка про використання результатів дипломної роботи.....	86

## ВСТУП

Актуальність теми. При проведенні сегментації основною процедурою є порівняння результатів сегментації певним алгоритмом із еталонним розбиттям. Основна ознака для його проведення – це яскравість для монохромного зображення та кольорова компонента для кольорового зображення. Крім того, для проведення сегментації використовуються границі зображення та текстура.

Сегментація особливо актуальна при обробці біомедичних зображень [1] наприклад, для виявлення пухлин та інших патологій, визначення об'ємів тканин, різних діагностик, планування лікування. Тому тема дослідження є безумовно актуальною.

Об'єкт дослідження – системи автоматизованої мікроскопії.

Предмет дослідження – алгоритми порівняння зображень.

Метою роботи є розроблення алгоритмів порівняння зображень в метриці Фреше для незважених дерев для систем автоматизованої мікроскопії.

Методи досліджень: методи цифрової обробки зображень, методи математичного аналізу, методи комп'ютерного моделювання.

Наукова новизна одержаних результатів.

Розроблено алгоритми порівняння зображень в метриці Фреше для незважених дерев.

Практичне значення отриманих результатів. Розроблено програмний модуль порівняння зображень .

Публікації та апробація магістерської роботи.

1. Дериш Б. Б., Вихиль Ю.О., Юглічек О. В., Герасимчук С. Ю. Алгоритми порівняння зображень на основі контурного аналізу: II Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 11-12 [2].

2. Дериш Б. Б., Вихиль Ю.О., Юглічек О. В. Порівняння зображень на основі колірних і текстурних ознак: II Наук.-практ. конф. молодих вчених і

студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 12-13 [3].

У першому розділі проведено аналіз методів сегментації, моделей опису кольорових зображень та здійснено порівняння метрик зображень, здійснено аналіз завдання на магістерську роботу та зроблено постановку задач [4,5].

У другому розділі розроблено алгоритми визначення відстані між ланцюжками, циклами, ламаними лініями, багатокутниками та визначення дискретної відстані між контурами.

У третьому розділі розроблений програмний модуль обчислення відстані між зображеннями, приведені системні вимоги та структура програмного модуля та здійснено порівняльний аналіз результатів алгоритмів оцінки відстані між зображеннями.

В додатках представлені довідка про впровадження та світлокопії публікацій.

# 1 АНАЛІЗ МЕТРИК ПОРІВНЯННЯ ЗОБРАЖЕНЬ

## 1.1 Сегментація зображень та методи оцінки її якості

Порівняння зображень використовується при аналізі похибок сегментації. Тому розглянемо алгоритми сегментації зображень та методи її оцінки. Сегментація кольорового зображення – це процес виділення на зображенні однієї або декількох областей, що задовольняють деякому критерію однорідності [6].

Іншими словами, сегментація – це поділ зображення на області з подібними характеристиками. Основна ознака для його проведення – це яскравість для монохромного зображення та кольорова компонента для кольорового зображення. Крім того, для проведення сегментації використовуються границі зображення та текстура.

Розглянемо основні алгоритми сегментації.

Алгоритми порогової сегментації використовуються для контрастних зображень і застосовується разом з морфологічними операціями дилатації й ерозії, а вибір порога, як правило, ґрунтується на апріорній інформації про об'єкти, що виділяються.

Алгоритми морфологічної сегментації зводяться до пошуку контура об'єкта, що найкраще відповідав би її границі. Це досягається за допомогою морфологічного градієнта. Проте, наявність великої кількості помилкових гілок і внутрішніх контурів не дозволяє ефективно розв'язати задачу. Тому, очевидна висока складність таких алгоритмів, і, як наслідок, низька швидкодія і висока трудомісткість.

Третя група – це алгоритми сегментації нарощування областей. Як правило, вибір стартових точок сегментації відбувається або випадково, або за допомогою людини-оператора на підставі визначеної апріорної інформації.

Ще одна група алгоритмів базується на кластерному аналізі. Під час сегментації на зображенні вибираються центри кластерів, а далі послідовно

проводиться перевірка всіх точок на відстань до центрів у визначеній метриці. Недоліком даної групи алгоритмів є необхідність попереднього задання кількості кластерів.

На сьогодні не існує єдиного універсального алгоритму сегментації зображення. Найчастіше використовують набір спеціалізованих алгоритмів для певного класу задач. Оскільки алгоритми є спеціалізованими, то необхідно мати критерії оцінювання їх продуктивності. Р. Харлак та Л. Шапіро [7] запропонували такий якісний критерій для сегментації зображення: області сегментованого зображення повинні бути однорідними щодо деякої характеристики; сусідні області сегментації повинні суттєво відрізнитись одна від одної відносно обраної характеристики; границі кожного сегменту повинні бути простими, неперервними та просторово точними.

У більшості практичних задач якість сегментації розглядається як міра подібності двох зображень: сегментованого експертом та сегментованого певними алгоритмами. Якість сегментації зображення може визначатися як на суб'єктивному (якісному) так і на об'єктивному (кількісному) рівнях [8].

Суб'єктивні критерії – це критерії візуального сприйняття, отримані в процесі аналізу деякою групою експертів. Найбільшого поширення набув метод оцінок, при якому користувач оцінює якість зображення в балах за певною шкалою, вважаючи, що ідеальне зображення (просегментоване експертом в ручному режимі) має максимальний бал. Цей метод дозволяє оцінити такі характеристики зображення як правильність виділення контурів, областей, кольорів тощо.

Об'єктивні критерії – це критерії, що отримані внаслідок порівняння (знаходження різниці) кількісних ознак сегментованого та еталонного (сегментованого експертом) зображень.

Основними при цьому критеріями є:

- критерій кількості однорідних областей (визначається як абсолютне значення різниці кількості однорідних областей сегментованого та еталонного зображень);



- відхилення значення кольорової ознаки (для кольорового зображення) або рівня яскравості (для монохромного зображення) піксела від середнього значення кольорової ознаки (яскравості) однорідної області;

- середньоквадратичний критерій;
- критерій максимальної похибки;
- критерій абсолютної похибки.

Недоліком цих критеріїв є отримання значення якості сегментації на основі характерних ознак відповідних пікселів, проте без врахування зв'язків між самими пікселами.

Всі відомі критерії кількісної оцінки можна розділити на дві групи [9,10]:

- несупервізорні критерії;
- супервізорні критерії.

Несупервізорні критерії базуються на обчисленні різних статистик і застосовуються при відсутності апріорної інформації про сегментовані зображення. Супервізорні критерії побудовані на обчисленні міри відстані результатів сегментованої і еталонної форм об'єктів. Еталонна форма об'єктів задається експертом.

До супервізорних критеріїв відносяться такі: FOM критерій, критерій Хаусдорфа, RMS критерій, D<sub>Ku</sub> критерій, M критерій, V критерій, Баделлі критерій, RI-критерій, критерій GCE, VI-критерій.

Застосуванню метрик для порівняння зображень присвячено ряд публікацій [11-18].

## 1.2 Аналіз моделей опису кольорових зображень

Обробка образів (зображень) має діло з такими як фотографії, відеодані або томографічні зображення. Частково, комп'ютерна графіка представляє собою процес синтезування образів з абстрактних моделей, тоді як машинне

розпізнавання образів – це витягування деякої абстрактної інформації: н-д, 3D опису тієї чи іншої сцени, використовуючи її відеозйомку. Починаючи з 2000 р. аналогова обробка зображень (оптичними засобами) уступає місце цифровій обробці і, частково, цифровому редагуванню (наприклад, обробці зображень, отриманих з допомогою звичайних цифрових фотоапаратів).

Комп'ютерна графіка (і мозок людини) має справу з образами векторної графіки, тобто такими, які представлені геометричними кривими, багатокутниками і т.п. Зображення растрової графіки (або цифрове зображення, побітове відображення) в  $2D$  є представлення  $2D$  зображення скінченної множини дискретних величин, що називаються пікселями, розміщених на квадратній ґриді  $Z^2$  або шестикутній сітці. Як правило, растр – це квадратна  $2^k \times 2^k$  сітка з  $k=8,9$  або  $10$ . Відеозображення і томографічні (отримані як серія поперечних січень окремими частинами) зображення є  $3D$  зображеннями ( $2D$  плюс час); їх дискретні величини називаються вокселями (елементами об'єму).

Дискретне двійкове зображення використовує тільки два значення: 0 і 1; 1 інтерпретується як логічна «істина» і відображається чорним кольором; таким чином, само зображення ототожнюється з множиною чорних пікселів. Елементи бінарного  $2D$  зображення можна розглядати як комплексні числа  $x = iy$ , де  $(x, y)$  – координата точки на дійсній площині  $R^2$ . Неперервне бінарне зображення є (зазвичай компактною) підмножиною локально компактного метричного простору (зазвичай евклідового простору  $E^n$  з  $n = 2,3$ ).

Напівтонові зображення можуть розглядатися як точково-зважені бінарні зображення. В загальному випадку нечітка множина є точково-зваженою множиною з вагами (значеннями належності) [19, 20]. Для напівтонових зображень  $x_{ui}$ -представлення застосовується у випадку, коли площинні координати  $(x, y)$  позначають форму, в той час як вага  $i$  – текстуру (розподіл інтенсивності). Іноді використовується також матриця  $((i_{xy}))$  напівтонів. Гістограма яскравості напівтонового зображення показує частоту кожного значення яскравості, що є в даному зображенні. Якщо зображення має  $m$  рівнів

яскравості (стовпчиків гістограми напівтонів), то існують  $2^m$  різних можливих інтенсивностей. Зазвичай  $m=8$  і числа  $0, 1, \dots, 255$  представляють діапазон інтенсивності від білого до чорного; інші типові значення  $m=10, 12, 14, 16$ . Око людини відрізняє порядку 350 тис. різних кольорів, але тільки 30 різних напівтонів; таким чином, колір володіє набагато більш високою роздільною здатністю.

Для кольорових зображень найбільш відомим є *RGB* представлення, де просторові координати *R, G, B* означають рівні червоної, зеленої і синьої колірної складової: *3D* гістограма показує яскравість в кожній точці. Серед багатьох інших *3D* моделей (просторів) кольорів розрізняють: (*CMY*) куб (кольори голубий, пурпурний, жовтий), (*HSL*) конус (тип кольору *H*, заданий як кут, насиченість *S*, задана у %, освітленість *L*, задана у %) і (*YUV*), (*YIQ*), що використовуються відповідно в телевізійних системах PAL і NTSC. Згідно затвердженої Міжнародною комісією по освітленості (*MKO*) методиці перерахунок (*RGB*) в міру яскравості (освітленості) напівтону здійснюється як  $0,299R+0,587G+0,114B$ . Колірна гістограма є вектором ознак довжини *n* (зазвичай  $n=64$  або  $256$ ) з компонентами, що представляють або загальну кількість пікселів, або відсоток пікселів даного кольору в зображенні.

Зображення найчастіше представлені векторами ознак, що включають колірні гістограми, колірну насиченість текстури, дескриптори форми і т.п. прикладами просторів ознак є: вихідна інтенсивність (значення пікселів), краї (границі, контури, поверхні) відмінні характеристики (кутові точки, перетини ліній, точки високої кривизни) і статистичні ознаки (моментні інваріанти, центроїди). До типових відеоознак відносяться перекриття кадрів, переміщення. Відновлення зображення (пошук подібностей) полягає (так як і для інших даних, таких як аудіо-записи, послідовності ДНК, текстові документи, часові ряди і т.п.) в пошуку зображень, ознаки яких або близькі між собою, або близькі до конкретного запиту, або знаходяться в заданому діапазоні.

### 1.3 Метрики порівняння зображень

Є два методи безпосереднього порівняння зображень: за інтенсивністю (кольором і текстурою гістограми) і за геометрією (опис форми з допомогою серединної осі, скелету і т.п.) Нечіткий термін форма застосовується для опису зовнішнього вигляду (силуету) об'єкта, його локальної геометрії або загального геометричного рисунку (геометричних особливостей, точок, кривих і т.п.) або для такого рисунку з точністю до деякої групи перетворень подібності (переносів, поворотів і масштабування). Нечіткий термін текстура означає все, що залишається після обробки даних про колір і форму.

Подібність між векторними представленнями зображень вимірюється за допомогою звичайних відстаней:  $l_p$ -метрик, метрик зваженого редагування, відстані Танімото, відстані косинуса, відстані Махаланобіса і його узагальнень, відстані бульдозера. З ймовірнісних відстаней найбільш часто використовуються: відстань Бхаттачар'я 2, відстань Хеллінджера, відстань Кулбака-Лейблера, відстань Джеффрі і (особливо для гістограм)  $\chi^2$ -відстань, відстань Колмогорова-Смірнова, відстань Купера.

Основними відстанями, що застосовуються для компактних підмножин  $X$  і  $Y$  множини  $R^n$  (зазвичай  $n = 2,3$ ) або для дискретних варіантів, є: метрика Асплунда, метрика Шепарда, напівметрика симетричної різниці  $Vol(X\Delta Y)$ , метрика Никодима, відхилення площі, метрика цифрового об'єму і їх нормалізації, а також варіанти хаусдорфової відстані.

Для цілей обробки зображень перераховані нижче відстані є відстанями між «істинним» і наближеним цифровими зображеннями з тим, щоб оцінити якість алгоритмів. Для цілей відновлення зображень відстані вимірюються між векторами ознак запиту і посилань.

Колірні відстані.

Колірний простір – це 3-параметричний опис колірності. Необхідність саме цих трьох параметрів обумовлена існуванням в людському оці трьох видів рецепторів, які сприймають короткохвильові, середньохвильові і довгохвильові випромінювання, що відповідають синьому, зеленому і червоному кольору.

Міжнародна комісія по освітленості визначила в 1931 р. параметри колірного простору ( $XYZ$ ) на основі ( $RGB$ )-моделі і вимірювань людського зору. Згідно стандарту комісії в колірному просторі ( $XYZ$ ) величини  $X$ ,  $Y$  і  $Z$  приблизно відповідають червоному, зеленому і синьому кольорам. Головним припущенням колориметричного аналізу, який є експериментально обґрунтованим Індоу (1991), є те, що сприйманий колірний простір допускає існування метрики істинної колірної відстані. Припускається, що дана метрика буде локально евклідовою, тобто рімановою метрикою. Другим припущенням є існування неперервного відображення з метричного простору колірних стимулів в цей метричний простір.

Такої рівно контрастної колірної шкали, де рівні відстані в колірному просторі відповідають рівним відстаням в кольорах, поки ще не отримано і існуючі колірні відстані є різними її апроксимаціями. Першим кроком в цьому напрямку є еліпси МакАдама, тобто області  $(x, y)$  на діаграмі хроматичності, всі кольори, що містяться в якій виглядають нерозрізненними для нормального людського ока. Ці 25 еліпсів визначають метрику в колірному просторі. Тут  $x = \frac{X}{X+Y+Z}$  і  $y = \frac{Y}{X+Y+Z}$  є проєктивними координатами і кольори діаграми хроматичності займають деяку область дійсної проєктивної площини. Простір CIE ( $L^*a^*b^*$ ) є адаптацією колірного простору комісії МКО (від 1931 р.); він забезпечує часткову лінеаризацію метрики, що закладена в еліпсах МакАдама. Параметри  $L^*$ ,  $a^*$ ,  $b^*$  найбільш повної моделі – похідні від  $L$ ,  $a$ ,  $b$ , які є характеристикою яскравості  $L$  кольору від чорного  $L=0$  до білого  $L=100$ , при цьому  $a$  знаходиться між зеленим  $a < 0$  і червоним  $a > 0$ ,  $b$  – між зеленим  $a < 0$  і жовтим  $b > 0$ .

Середня колірна відстань.

Для даного  $3D$  колірному простору і переліку  $n$  кольорів нехай  $(c_{i1}, c_{i2}, c_{i3})$  – представлення  $i$ -го кольору з переліку в даному просторі. Для колірної гістограми  $x = (x_1, \dots, x_n)$  її середнім кольором є вектор  $(x_{(1)}, x_{(2)}, x_{(3)})$ , де  $x_{(j)} = \sum_{i=1}^n x_i c_{ij}$  (наприклад, середні значення червоного, синього і зеленого в  $(RGB)$ ).

Середня колірна відстань [21] між двома колірними гістограмами є евклідовою відстанню їх середніх кольорів.

Відстані колірних компонентів.

Нехай дано зображення (як підмножина множини  $R^2$ ); нехай  $p_i$  означає (в відсотках) область даного зображення колір  $c_i$ . Колірною складовою зображення є пара  $(c_i, p_i)$ . Відстань Ма-Денга-Манжуната між колірними складовими  $(c_i, p_i)$  і  $(c_j, p_j)$  визначається як

$$|p_i - p_j| d(c_i, c_j)$$

де  $d(c_i, c_j)$  – відстань між кольорами  $c_i$  і  $c_j$  в даному колірному просторі. Мойсилевич і ін. ввели модифікацію даної відстані, подібну відстані бульдозера.

Квазівідстань перетинів гістограм.

Візьмемо дві колірні гістограми  $x = (x_1, \dots, x_n)$  і  $y = (y_1, \dots, y_n)$  (з  $x_i, y_i$ , що представляють кількість пікселів в стовпчику  $i$ ). Квазівідстань перетинів гістограм Свейна-Балларда між ними визначається як

$$1 - \frac{\sum_{i=1}^n \min(x_i, y_i)}{\sum_{i=1}^n x_i}.$$

Для нормалізованих гістограм (загальна сума дорівнює 1) вищеприведена квазівідстань стає звичайною  $l_1$ -метрикою  $\sum_{i=1}^n |x_i - y_i|$ . Нормалізована взаємна

кореляція Розенфельда-Кака між  $x$  і  $y$  є подібністю, визначеною як  $\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$ .

Квадратична відстань гістограми.

Для двох гістограм  $x = (x_1, \dots, x_n)$  і  $y = (y_1, \dots, y_n)$  (зазвичай  $n = 256$  або  $n = 64$ ), що представляють колірність (в відсотках) двох зображень, їх квадратична відстань гістограми є відстанню Махаланобіса, визначеною як

$$\sqrt{(x - y)^T A (x - y)},$$

де  $A = ((a_{ij}))$  – симетрична додатньо визначена матриця;

вага  $a_{ij}$  – деяка підтверджена безпосереднім сприйняттям подібності між кольорами  $i$  і  $j$  [21].

Наприклад,  $a_{ij} = 1 - \frac{d_{ij}}{\max_{1 \leq p, q \leq n} d_{pq}}$ , де  $d_{ij}$  є евклідовою відстанню між 3

векторами, що представляють  $i$  і  $j$  в деякому колірному просторі. Інше визначення задається як

$$a_{ij} = 1 - \frac{1}{\sqrt{5}} \left( (v_i - v_j)^2 + (s_i \cosh_i - s_j \cosh_j)^2 + (s_i \sinh_i - s_j \sinh_j)^2 \right)^{\frac{1}{2}},$$

де  $(h_i, s_i, v_i)$  і  $(h_j, s_j, v_j)$  – представлення кольорів  $i$  і  $j$  в колірному просторі (HSV).

Відстань напівтонового зображення.

Нехай  $f(x)$  і  $g(x)$  – значення яскравості двох цифрових напівтонових зображень  $f$  і  $g$  для пікселя  $x \in X$ , де  $X$  є растром пікселів. Будь-яка відстань

між точно зваженими множинами  $(X, f)$  і  $(X, g)$  (наприклад, відстань бульдозера) може бути застосоване для вимірювання відстані між  $f$  і  $g$ . Проте основними використовуваними відстанями (помилками) між зображеннями  $f$  і  $g$  є:

$$1) \text{ середньоквадратична помилка } RMS(f, g) = \left( \frac{1}{|X|} \sum_{x \in X} (f(x) - g(x))^2 \right)^{1/2}$$

(як варіант допускається використання  $l_1$ -норми  $|f(x) - g(x)|$  замість  $l_2$ -норми;

$$2) \text{ відношення сигнал-шум } SNR(f, g) = \left( \frac{\sum_{x \in X} g(x)^2}{\sum_{x \in X} (f(x) - g(x))^2} \right)^{1/2};$$

3) коефіцієнт помилок неправильної класифікації пікселів  $\frac{1}{|X|} |\{x \in X : f(x) \neq g(x)\}|$  (нормалізована хемінгова відстань);

$$4) \text{ середньоквадратична частотна помилка } \left( \frac{1}{|U|^2} \sum_{u \in U} (F(u) - G(u))^2 \right)^{1/2}, \text{ де}$$

$F$  і  $G$  – дискретні перетворення Фур'є для  $f$  і  $g$  відповідно і  $U$  – частотна область;

$$5) \text{ помилка порядку } \delta \text{ в номер Соболева } \left( \frac{1}{|U|^2} \sum_{u \in U} (1 + |\eta_u|^2)^\delta (F(u) - G(u))^2 \right)^{1/2}, \text{ де } 0 < \delta < 1 \text{ фіксоване (зазвичай } \frac{1}{2}) \text{ і } \eta_u$$

є частотний вектор, що асоціюється з позицією  $u$  в частотній області  $U$ .

$L_p$ -метрика стиснення зображення.

Візьмемо число  $r$ ,  $0 \leq r < 1$ .  $L_p$ -метрика стиснення зображення є звичайною  $L_p$ -метрикою на  $R_{\geq 0}^{n^2}$  (множині напівтонових зображень, що

розглядаються як  $n \times n$  матриці), де  $p$  – розв'язок рівняння  $r = \frac{p-1}{2p-1} \cdot e^{\frac{p}{2p-1}}$ . Так,



$p = 1, 2$  або  $\infty$  для  $r = 0$ ,  $r = \frac{1}{3}e^{2/3} \approx 0,65$  або  $r \geq \frac{\sqrt{e}}{2} \approx 0,82$ . Тут  $r$  оцінює інформативну (тобто наповнену ненульовими значеннями) частину зображення. Ця метрика є найкращою за якістю метрикою для вибору схеми стиснення з втратами [22].

Відстані округлення.

Відстанями округлення називаються відстані, що апроксимують евклідову відстань як зважену відстань шляху в графі  $G = (Z^2, E)$ , де два пікселі вважаються сусідніми, якщо один може бути отриманий з другого однокроковим ходом по  $Z^2$ . При цьому даються перелік дозволених ходів і проста відстань, тобто додатна вага поставлена у відповідність кожному типу такого ходу.

Метрика  $(\alpha, \beta)$ -округлення відповідає двом дозволеним ходам – з  $l_1$ -відстанню і  $l_\infty$ -відстанню 1 (тільки діагональні переміщення) – зважені числами  $\alpha$  і  $\beta$  відповідно. Основними випадками застосування є  $(\alpha, \beta) = (1, 0)$  (метрика міського квартала або 4-метрика), метрика шахової дошки, або 8-метрика),  $(1, \sqrt{2})$  (метрика Монтанарі),  $((3, 4)$ -метрика), (метрика Хілдича-Рутовіца),  $(5, 7)$  (метрика Вервера).

Метрика Боргефоре відповідає трьом дозволеним ходам – з  $l_1$ -відстанню, з  $l_\infty$ -відстанню 1 (тільки діагональні переміщення) і ходом коня – з вагами 5, 7 і 11 відповідно.

Метрика  $3D$ -округлення (або метрика  $(\alpha, \beta, \gamma)$ -округлення) є метрикою зваженого шляху нескінченного графа з множиною вершин  $Z^3$ , в якому дві вершини є сусідніми, якщо їх  $l_\infty$ -відстань дорівнює 1, а ваги  $\alpha$ ,  $\beta$  і  $\gamma$  зв'язані з 6 сусідніми гранями, 12 сусідніми ребрами і 8 сусідніми вершинами відповідно. Якщо  $\alpha = \beta = \gamma = 1$ , то ми маємо  $l_\infty$ -метрику. Метрики  $(3, 4, 5)$ - і  $(1, 2, 3)$ -округлення є найбільш часто застосовними для роботи з  $3D$  зображеннями.

Метрика Чаудхури-Мурти-Чаудхури між послідовностями  $x = (x_1, \dots, x_m)$  і  $y = (y_1, \dots, y_m)$  визначається як

$$|x_{i(x,y)} - y_{i(x,y)}| + \frac{1}{1 + \left\lfloor \frac{n}{2} \right\rfloor} \sum_{1 \leq i \leq n, i \neq i(x,y)} |x_i - y_i|,$$

де максимальне значення  $x_i - y_i$  отримується для  $i = i(x, y)$ . Для  $n = 2$  це метрика  $\left(1, \frac{3}{2}\right)$ -округлення.

Відстань бульдозера є дискретною формою відстані Монжа-Канторовича. Грубо говорячи, це мінімальний обсяг роботи, яка необхідна для переміщення ґрунту з одного місця (відповідним чином розміщеного в просторі) на інше (сукупність ям). Для будь-яких скінчених послідовностей  $x = (x_1, \dots, x_m)$  і  $y = (y_1, \dots, y_m)$  метричного простору  $(X, d)$  розглянемо сигнатури, тобто точково зважені множини  $P_1 = (p_1(x_1), \dots, p_1(x_m))$  і  $P_2 = (p_2(x_1), \dots, p_2(x_n))$ . Наприклад [23], сигнатури можуть представляти кластери кольорів або текстурного змісту зображень: елементи  $X$  є центроїдами кластерів, а  $p_1(x_i)$ ,  $p_2(y_j)$  – розмірами відповідних кластерів. Вихідна відстань  $d$  є деякою колірною відстанню, скажемо, евклідовою відстанню в  $3D (L^* a^* b^*)$  колірному просторі.

Нехай  $W_1 = \sum_i p_1(x_i)$  і  $W_2 = \sum_j p_2(y_j)$  є сумарними вагами  $P_1$  і  $P_2$  відповідно. Тоді відстань бульдозера (або відстань транспортування) між сигнатурами  $P_1$  і  $P_2$  визначається як функція

$$\frac{\sum_{i,j} f_{ij}^* d(x_i, y_j)}{\sum_{i,j} f_{ij}^*},$$

де  $m \times n$  матриця  $S^* = ((f_{ij}^*))$  є оптимальним, тобто мінімізуючим  $\sum_{i,j} f_{ij}^* d(x_i, y_j)$

поток. Потік (ваги ґрунту) – це  $m \times n$  матриця  $S = ((f_{ij}))$ , що задовольняє наступні обмеження:

- 1) всі  $f_{ij} \geq 0$ ;
- 2)  $\sum_{i,j} f_{ij} = \min\{W_1, W_2\}$ ;
- 3)  $\sum_i f_{ij} \leq p_2(y_j)$  і  $\sum_i f_{ij} \leq p_1(x_i)$ .

Отже, дана відстань є усередненням вихідної відстані  $d$ , на яку вантажі переміщуються оптимальним потоком.

У випадку  $W_1 = W_2 = 1$  вищеприведені дві нерівності стають рівностями. Нормалізація сигнатур до  $W_1 = W_2 = 1$  (що не змінює відстані) дозволяє розглядати  $P_1$  і  $P_2$  як розподіли ймовірностей випадкових величин  $X$  і  $Y$ . Тоді  $\sum_{i,j} f_{ij}^* d(x_i, y_j)$  є просто  $E_S[d(X, Y)]$ , тобто відстань бульдозера співпадає в цьому випадку з метрикою Канторовича-Меллоуза-Монжа-Васермана. А для випадку  $W_1 < W_2$  вона в загальному випадку не є метрикою. Проте заміна в вищеприведеному визначенні нерівності 3) рівностями

$$3') \sum_i f_{ij} = p_2(y_j) \text{ і } \sum_i f_{ij} = \frac{p_1(x_1)W_1}{W_2}$$

дає напівметрику пропорційного переносу Жіаннополоса-Вельткампа.

## 1.4 Відстань між параметризованими кривими

Відстань функції обертання. Для плоского багатокутника  $P$  його функцією обертання  $Tr(s)$  називається кут (проти часової стрілки) між дотичною і  $x$ -віссю як функція довжини дуги  $s$ . Ця функція зростає при кожному повороті наліво і спадає при повороті направо.

Для двох багатокутників з рівними периметрами їх відстанню функції обертання є  $L_p$ -метрика між їх функціями обертання.

Відстань функції розміру. Для плоского графа  $G=(V,E)$  і функції  $f$  на його множині вершин  $V$  функція розміру  $S_G(x,y)$  визначається на точках  $(x,y) \in \mathbb{R}^2$  як число зв'язних компонент звуження  $G$  на вершини  $\{v \in V : f(v) \leq y\}$ , що містять точку  $v'$  з  $f(v') \leq x$ .

Для двох плоских графів з множинами вершин, що належать растру  $R \subset \mathbb{Z}^2$ , їх відстанню функції розміру Ураза-Веррі є нормалізована  $l_1$ -відстань між їх функціями відстані над растрами пікселів.

Відстань відображення. Для скінченного об'єднання  $A$  плоских фігур і кожної точки  $x \in \mathbb{R}^2$  нехай  $V_A^x$  означає об'єднання інтервалів  $]x,a[$   $a \in A$ , які видно з  $x$ , тобто  $]x,a[ \cap A = \emptyset$ .

Нехай  $p_A^x$  – площа множини  $\{x+v \in V_A^x : x-v \in V_A^x\}$ .

Відстанню відображення Хагедорна-Велькампа між скінченими об'єднаннями  $A$  і  $B$  плоских кривих є нормалізована  $l_1$ -відстань між відповідними функціями  $p_A^x$  і  $p_B^x$ , визначена як

$$\frac{\int_{\mathbb{R}^2} |p_A^x - p_B^x| dx}{\int_{\mathbb{R}^2} \max\{p_A^x, p_B^x\} dx}.$$

Перетворення між відстанями. Візьмемо метричний простір  $(X = Z^2, d)$  і двійкове цифрове зображення  $M \subset X$ . Перетворення між відстанями називається функція  $f_M : X \rightarrow R_{\geq 0}$ , де  $f_M(x) = \inf_{u \in M} d(x, u)$  є відстанню між точкою і множиною  $d(x, M)$ . Отже, відстаннє перетворення можна розглядати як напівтонове цифрове зображення, в якому кожному пікселю присвоюється мітка (рівень напівтону), що відповідає відстані до найближчого пікселя фону. Відстаннє перетворення в процесах обробки зображень також називаються відстанневими полями і, головним чином, відстанневими картами. Відстаннє перетворення форми – відстаннє перетворення, в якому  $M$  – межа зображення. Для  $X = R^2$  граф  $\{(x, f(x)) : x \in X\}$  для  $d(x, M)$  називається поверхнею Вороного для  $M$ .

Серединна скелетна вісь. Нехай  $(X, d)$  – метричний простір і  $M$  – підмножина  $X$ . Серединна вісь  $X$  – множина  $MA(X) = \{x \in X : |\{m \in M : d(x, m) = d(x, M)\}| \geq 2\}$ , тобто всі точки  $X$ , які мають в  $M$  не менше двох елементів найкращого наближення.  $MA(X)$  складається з всіх точок границь областей Вороного для точок з  $M$ . Скелет  $Skel(X)$  множини  $X$  є множина центрів всіх куль (відносно відстані  $d$ ), які вписані в  $X$  і є максимальними, тобто не належать ніякій іншій такій кулі. Геометричне місце розрізів множини  $X$  – це замкнення  $\overline{MA(X)}$  серединної осі. В загальному випадку  $MA(X) \subset Skel(X) \subset \overline{MA(X)}$ . Перетворення серединної осі, скелета і геометричного місця розрізів – це точково-зважені множиною  $MA(X)$ ,  $Skel(X)$  і  $\overline{MA(X)}$  (звуження відстаннєвого перетворення на ці множини) з  $d(x, M)$ , що розглядається як вага точки  $x \in X$ .

Зазвичай  $X \subset E^n$  і  $M$  – границя  $X$ . У випадку коли  $M$  є неперервною границею, серединна вісь може вважатися межею діаграми Вороного по мірі того як число породжуючих точок стає нескінченим. Для 2D бінарних зображень  $X$  скелет є кривою товщиною в один піксель в цифровому випадку.

Екзоскелет множини  $X$  – скелет доповнення множини  $X$ , тобто фону зображення, для якого  $X$  є переднім планом.

Прокрустова відстань. Обрис форми (конфігурація точок в  $R^2$ ), який розглядається як вираження інваріантних властивостей форми відносно переносу, повороту і масштабу, можна представити як послідовність орієнтирів, тобто специфічних точок на формі, вибраних за певним правилом. Кожен орієнтир  $a$  може розглядатися як елемент  $(a', a'') \in R^2$  або елемент  $a' + a''i \in C$ .

Розглянемо дві форми  $x$  і  $y$ , що представлені їх орієнтирними векторами  $(x_1, \dots, x_n)$  і  $(y_1, \dots, y_n)$  з  $C^n$ . Припустимо, що  $x$  і  $y$  корегуються для переносу умовою  $\sum_t x_t = \sum_t y_t = 0$ . Тоді їх прокрустова відстань визначається як

$$\sqrt{\sum_{t=1}^n |x_t - y_t|^2},$$

де дві форми є оптимально (за критерієм найменших квадратів) розміщеними по доній лінії для корегування масштабу і їх відстань обрису Кендала визначається як

$$\arccos \sqrt{\frac{\left(\sum_t x_t \bar{y}_t\right)\left(\sum_t y_t \bar{x}_t\right)}{\left(\sum_t x_t \bar{x}_t\right)\left(\sum_t y_t \bar{y}_t\right)}},$$

де  $\bar{\alpha} = a' - a''i$  є комплексно спряженим числа  $\alpha = a' + a''i$ .

Дотична відстань. Для будь якого  $x \in R^n$  і сімейства перетворень  $t(x, \alpha)$  де  $\alpha \in R^k$  – вектор  $k$  параметрів (наприклад, коефіцієнт масштабування і кут повороту), множина  $M_x = \{t(x, \sigma) : \alpha \in R^k\} \subset R^n$  є різноманіттям розмірності не більше ніж  $k$ . Це крива, якщо  $k=1$ . Мінімальна евклідова відстань між

різноманіттями  $M_x$  і  $M_y$  є корисною відстанню, оскільки вона інваріантна відносно перетворень  $t(x, \alpha)$ . Проте розрахувати таку відстань в загальному випадку дуже важко; тому  $M_x$  апроксимують як її дотичний підпростір в точці  $x$ :  $\{x + \sum_{i=1}^k \alpha_k x^i : \alpha \in R^k\} \subset R^n$ , де породжуючі його дотичні вектори  $x^i$ ,  $1 \leq i \leq k$ , є частинними похідними  $t(x, \alpha)$  відносно  $\alpha$ . Одностороння (або орієнтована) дотична відстань між елементами  $x$  і  $y$  з  $R^n$  є квазівідстань  $d$ , яка визначена як

$$\sqrt{\min_{\alpha} \left\| x + \sum_{i=1}^k \alpha_k x^i - y \right\|^2}.$$

Дотична відстань Симара-Ле Кана-Денкера визначається як  $\min\{d(x, y), d(y, x)\}$ .

В загальному випадку дотична множина метричного простору  $X$  в точці  $x$  визначається (за Громовим) як будь-яка гранична точка сімейства його розтягів з коефіцієнтом розтягу, що прямує в нескінченність, яка береться в точковій топології Громова-Хаусдорфа.

Відстань пікселя. Візьмемо два цифрових образи, як розглядаються як бінарні  $m \times n$  матриці  $x = ((x_{ij}))$  і  $y = ((y_{ij}))$ , де піксель  $x_{ij}$  є чорним або білим, якщо він дорівнює 1 або 0 відповідно. Для кожного пікселя  $x_{ij}$  обрамлене відстаннєве відображення до найближчого пікселя протилежного кольору  $D_{BW}(x_{ij})$  є число обрамлень (де кожне обрамлення складається з пікселів, що рівновіддалені  $(i, j)$ ), які протягнулися від  $(i, j)$  до зустрічі з першим обрамленням, яке містить піксель протилежного кольору.

Відстані пікселів задаються як

$$\sum_{1 \leq i \leq m} \sum_{1 \leq j \leq n} |x_{ij} - y_{ij}| (D_{BW}(x_{ij}) + D_{BW}(y_{ij})).$$

Квазівідстань коефіцієнта якості.

Візьмемо два бінарних зображення, що розглядаються як непорожні скінчені підмножини  $A$  і  $B$  скінченного метричного простору  $(X, d)$ . Для них квазівідстань коефіцієнта якості Пратта визначається як

$$\left( \max\{|A|, |B|\} \sum_{x \in B} \frac{1}{1 + \alpha d(x, A)^2} \right)^{-1},$$

де  $\alpha$  – константа масштабування (зазвичай  $\frac{1}{9}$ ) і  $d(x, A) = \min_{y \in A} d(x, y)$  – відстань між точкою і множиною.

Прикладами подібних квазівідстаней є відстань середньої похибки Пелі-Малаха  $\frac{1}{|B|} \sum_{x \in B} d(x, A)$  і відстань середньоквадратичної похибки  $\frac{1}{|B|} \sum_{x \in B} d(x, A)^2$ .

Середня хаусдорфова відстань  $p$ -го порядку. Візьмемо два бінарних зображення, що розглядаються як непорожні скінчені підмножини  $A$  і  $B$  скінченного метричного простору  $(X, d)$ . Їх середня хаусдорфова відстань  $p$ -го порядку є нормалізована  $L_p$ -відстань Хаусдорфа, визначена як

$$\left( \frac{1}{|X|} \sum_{x \in B} |d(x, A) - d(x, B)|^p \right)^{\frac{1}{p}},$$

де  $d(x, A) = \min_{y \in A} d(x, y)$  – відстань між точкою і множиною. Звичайна хаусдорфова метрика пропорційна середній хаусдорфовій відстані  $\infty$ -го порядку.

$\Sigma$ -хаусдорфова відстань Венкатасубрамініана  $d_{d\text{Haus}}(A, B) + d_{d\text{Haus}}(B, A)$  дорівнює  $\sum_{x \in A \cup B} |d(x, A) - d(x, B)|$ , тобто є варіантом  $L_1$ -відстані Хаусдорфа.



Іншим варіантом середньої хаусдорфової відстані 1-го порядку є середня геометрична похибка Ліндстрьома-Турка між двома зображеннями, що розглядаються як поверхні  $A$  і  $B$ . Вона визначається як

$$\frac{1}{\text{Area}(A) + \text{Area}(B)} \left( \int_{x \in A} d(x, B) dS + \int_{x \in B} d(x, A) dS \right),$$

де  $\text{Area}(A)$  – площа поверхні  $A$ . Якщо розглядати зображення як скінчені множини  $A$  і  $B$ , то їх середня геометрична похибка визначається як

$$\frac{1}{|A| + |B|} \left( \sum_{x \in A} d(x, B) + \sum_{x \in B} d(x, A) \right).$$

Модифікована хаусдорфова відстань. Візьмемо два бінарних зображення, що розглядаються як непорожні скінчені підмножини  $A$  і  $B$  скінченного метричного простору  $(X, d)$ . Їх модифікована хаусдорфова відстань по Дюбюсону-Джейну визначається як максимум відстаней між точкою і множиною, усереднених по  $A$  і  $B$ :

$$\max \left\{ \frac{1}{|A|} \sum_{x \in A} d(x, B), \frac{1}{|B|} \sum_{x \in B} d(x, A) \right\}.$$

Часткова хаусдорфова квазівідстань. Візьмемо два бінарних зображення, що розглядаються як непорожні скінчені підмножини  $A$  і  $B$  скінченного метричного простору  $(X, d)$ , і цілі числа  $k, l$ , такі що  $1 \leq k \leq |A|$ ,  $1 \leq l \leq |B|$ . Їх часткова  $(k, l)$ -хаусдорфова квазівідстань за Хаттенлокера-Рукліджу визначається як

$$\max \left\{ k^{th}_{x \in A} d(x, B), l^{th}_{x \in B} d(x, A) \right\},$$

де  $k_{x \in A}^{th} d(x, B)$  означає  $k$ -у (замість, найбільшої  $A$ -ої, що розміщена першою) серед  $|A|$  відстаней  $d(x, B)$ , які розташовані в зростаючому порядку. Випадок  $k = \left\lfloor \frac{|A|}{2} \right\rfloor$ ,  $l = \left\lfloor \frac{|B|}{2} \right\rfloor$  відповідає середній модифікованій хаусдорфовій квазівідстані.

Відстань пляшкового горлечка. Візьмемо два бінарних зображення, що розглядаються як непорожні скінчені підмножини  $A$  і  $B$  з  $|A|=|B|=m$  скінченного метричного простору  $(X, d)$ . Їх відстань пляшкового горлечка визначається як

$$\min_f \max_{x \in A} d(x, f(x)),$$

де  $f$  – будь яке бієктивне відображення між  $A$  і  $B$ .

Варіантами вище приведеної відстані є:

- 1) відповідність мінімальної ваги:  $\min_f \sum_{x \in A} d(x, f(x))$ ;
- 2) рівномірна відповідність  $\{\max_{x \in A} d(x, f(x)) - \min_{x \in A} d(x, f(x))\}$ ;
- 3) відповідність найменшого відхилення

$$\min_f \left\{ \max_{x \in A} d(x, f(x)) - \frac{1}{|A|} \sum_{x \in A} d(x, f(x)) \right\}.$$

Для цілого числа  $t$ ,  $1 \leq t \leq |A|$ , відстань  $t$ -пляшкового горлечка між  $A$  і  $B$  рівна вище згаданому мінімуму, якщо  $f$  – будь яке відображення з  $A$  в  $B$ , таке що  $|\{x \in A : f(x) = e\}| \leq t$ . Випадки  $t=1$  і  $t=|A|$  аналогічні відповідно відстані пляшкового горлечка і орієнтованій хаусдорфовій відстані  $d_{d\text{Haus}}(A, B) = \max_{x \in A} \min_{y \in B} d(x, y)$ .

Хаусдорфова відстань з точністю до  $G$ .

Для групи  $(G, ;, id)$ , яка діє на евклідовому просторі  $E^n$ , хаусдорфова відстань з точністю до  $G$  між двома компактними підмножинами  $A$  і  $B$  (використовуване при обробці зображень) є узагальнена  $G$  - хаусдорфова відстань між ними, тобто мінімум  $d_{\text{Haus}}(A, g(B))$  по всіх  $g \in G$ . Зазвичай  $G$  – множина всіх ізометрій або всіх переносів простору  $E^n$ .

Гіперболічна хаусдорфова відстань.

Для будь-якої компактної підмножини  $A$  множини  $R^n$  позначимо через  $MAT(A)$  її перетворення серединної осі за Блюмом, тобто підмножину  $X = R^n \times R_{\geq 0}$ , всі елементи якої є парами  $x = (x', r_x)$  центрів  $x'$  і радіусів  $r_x$  максимальних вписаних в  $A$  куль застосовно до евклідової відстані  $d_E$  в  $R^n$ .

Гіперболічна хаусдорфова відстань [19] – хаусдорфова метрика на непустих компактних підмножинах  $MAT(A)$  метричного простору  $(X, d)$ , де гіперболічна відстань  $d$  на  $X$  визначається для його елементів  $x = (x', r_x)$  і  $y = (y', r_y)$  як

$$\max\{0, d_E(x', y') - (r_y - r_x)\}.$$

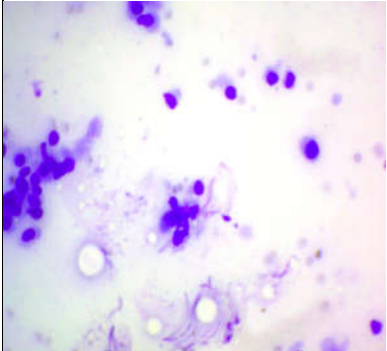
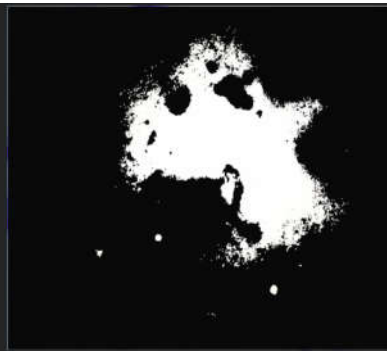

Нелінійна хаусдорфова метрика.

Для двох компактних підмножин  $A$  і  $B$  метричного простору  $(X, d)$  їх нелінійною хаусдорфовою метрикою (або хвильовою відстанню Затмарі-Рекечкі-Роска) називається хаусдорфова відстань  $d_{\text{Haus}}(A \cap B, (A \cup B)^*)$ , де  $(A \cup B)^*$  є підмножина  $A \cap B$ , що утворює замкнену неперервну область з  $A \cap B$  і відстані між точками можуть вимірюватися тільки вздовж шляхів, що повністю належать  $A \cup B$ .

## 1.5 Експериментальні дослідження алгоритмів сегментації

Якість сегментації зображення напряму залежить від етапу попередньої обробки. Особливо це стосується гістологічних та цитологічних зображень [25-26]. Приклад порогової сегментації зображення до і після попередньої обробки наведено у таблиці 1.1.

Таблиця 1.1 – Якість порогової сегментації

Вхідне зображення	Сегментація (без попередньої обробки)	Сегментація (після попередньої обробки)
		

Для приведення зображення до потрібного вигляду застосовувались такі етапи попередньої обробки:

- контрастність;
- яскравість;
- Dilate;
- Erode.

Окремої уваги заслуговує етап фільтрації. До усіх зображень застосовувались такі види фільтрів:

- гаусовий;
- білатеральний;
- адаптивний білатеральний;
- медіанний.

Для сегментації зображення застосовувались такі методи:

- порогова сегментація;
- метод водорозподілу;
- k-means.

Для цитологічних зображень, які відносяться до класу «Цитологія\_рак» було обрано такий діапазон параметрів попередньої обробки, що задовольняють всі зображення даного класу:

- контраст: 1.2 – 1.7;
- яскравість: 11-19;
- Dilate: 1;
- Erode: 1-5;
- фільтр;
- Blur:3;
- медіанний: 27;
- середнє знач. Гістограми: 95.

Метою сегментації цитологічного зображення є виділення ядер. Проблема виникає у тому, деякі ядра зливаються між собою, тож їх подальше виділення стає неможливим. У цьому випадку доцільно змінювати рівні контрастності, яскравості та ерозії. Дані операції дозволять відділити темні області (ядра) від фону.

Фільтрування дозволяє видалити шуми, що негативно вплинуть надалі на етапі сегментації. В результаті експериментів було встановлено, що доцільно використовувати етап фільтрації декілька разів різними фільтрами.

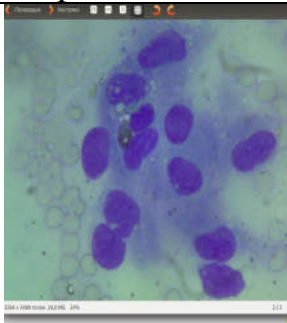
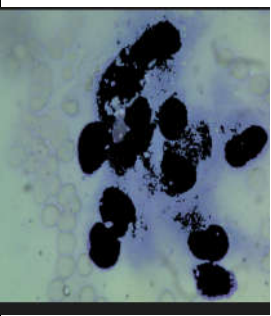
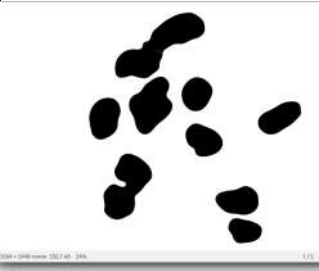

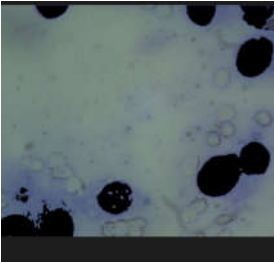
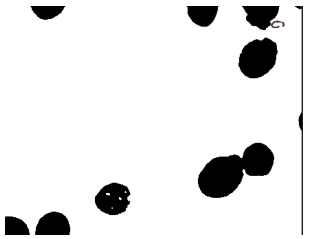

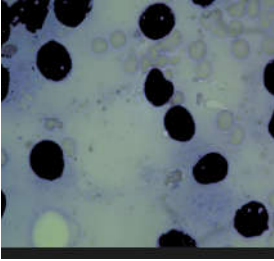
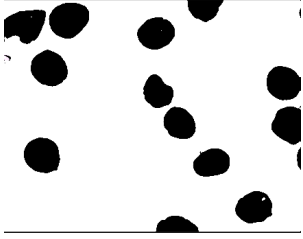
Приклад результатів обробки зображення з подальшою сегментацією наведено у таблиці 1.2.

У таблиці 1.2 наведено вхідне зображення, результат сегментації без попередньої обробки, параметри попередньої обробки та сегментація після обробки.

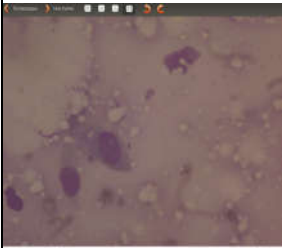
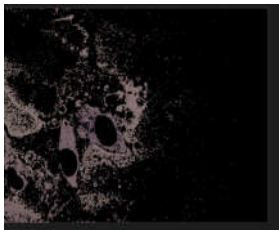
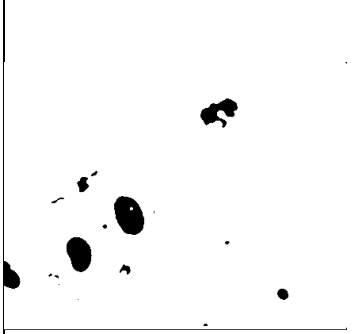

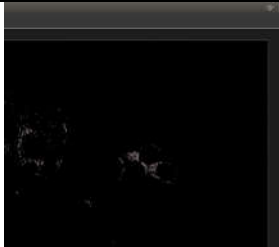
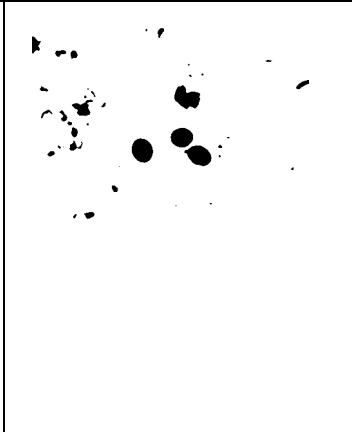



Виходячи із експериментів, можна зробити висновок, що досягнуто базові параметри основних етапів попередньої обробки, що в тій чи іншій мірі задовольняють усі зображення даного класу раку.

Параметри обробки зображення для класу «Цитологія\_ фіброаденома» не суттєво відрізняються від класу «Цитологія\_рак», однак потребують деякої корекції. Приклад порівняння зображень до і після попередньої обробки для класу зображень «Цитологія\_ фіброаденома» наведено у таблиці 1.3.

Таблиця 1.2 – Порівняння зображень

Вхідне зображення	Сегментація (без попер. обробки)	Параметри налаштування	Сегментація (після попер. обробки)
		Контраст: 1.2 Яскравість : 11 Dilate: 1 Erode: 1  Фільтр: Blur:13 Медіанний:27 Середнє знач. Гістограми: 86	
		Контраст: 1.2 Яскравість : 9 Dilate: 1 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: 81	
		Контраст: 1.2 Яскравість : 11 Dilate: 1 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: 102	

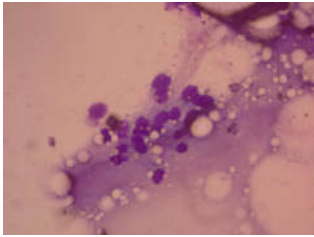
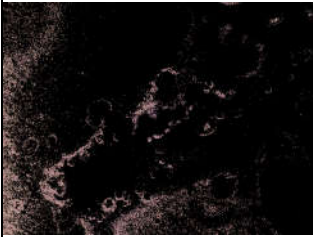
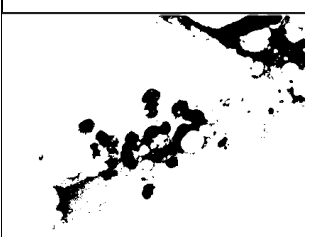
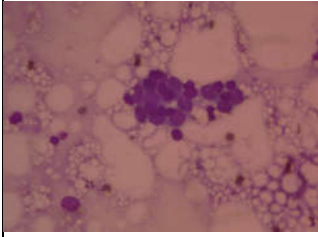


Таблиця 1.3 – Порівняння зображень для класу «Цитологія\_ фібroadенома»

Вхідне зображення	Сегментація(без попер. обробки)	Параметри попер. обробки	Сегментація(післ попер. обробки)
		Контраст: 1.5 Яскравість : 11 Dilate: 1 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: 19	
		Контраст: 1.5 Яскравість : 15 Dilate: 1 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: 9	
		Контраст: 1.5 Яскравість : 16 Dilate: 1 Erode: 1 Фільтр:Медіанний:27 Середнє знач. Гістограми: 15	

Виходячи із вищенаведених дослідів, можна зробити висновок, що зображення класу «Цитологія\_ фібroadенома» потребують більшої корекції, ніж попередній клас. Особливо це стосується таких параметрів, як контрастність та яскравість (потрібно підвищити значення) для подальшого чіткішого виділення ядер на зображенні. Застосування параметрів фільтрів не відрізняється від попереднього класу.

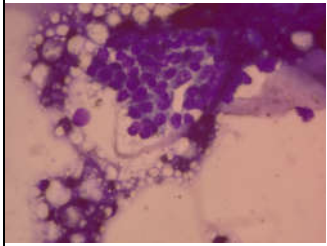
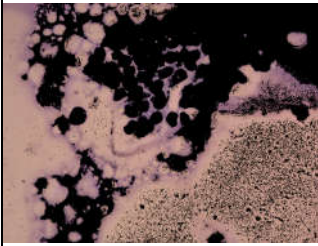

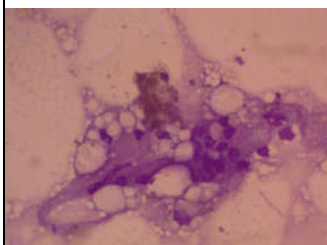


Зображення, що відносяться до класу «Цитологія\_непроліферативна\_мастопатія» характеризуються своєю складністю обробки та подальшого виділення ядер. Параметри попередньої обробки для класу зображень «Цитологія\_непроліферативна\_мастопатія» відрізняються від двох попередніх класів. Для даного класу зображень актуальним є застосування таких операцій, як Dilate та Erode, тобто збільшення області світлих та темних областей. Приклад порівняння зображень для класу «Цитологія\_непроліферативна\_мастопатія» наведено у таблиці 1.4.

Таблиця 1.4 – Порівняння зображень для класу «Цитологія\_непроліферативна\_мастопатія»

Вхідне зображення	Сегментація (без попер. обробки)	Параметри	Сегментація (після попер. обробки)
		Контраст: 1.5 Яскравість : 15 Dilate: 1 Erode: 3  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 65 55	
		Контраст: 2.1 Яскравість : 19 Dilate: 9 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 68 36	



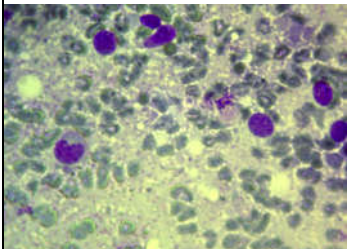
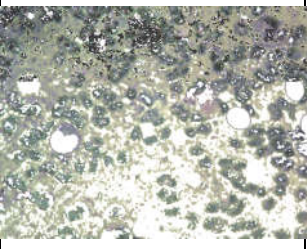
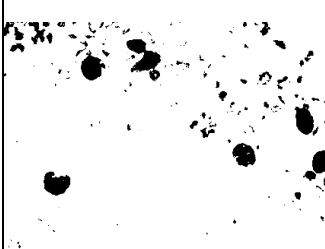
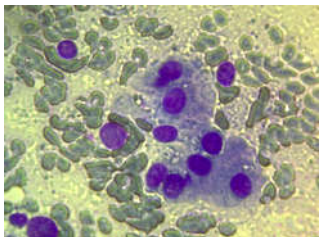
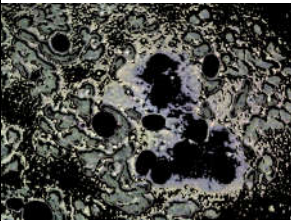
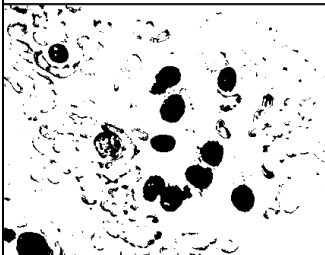
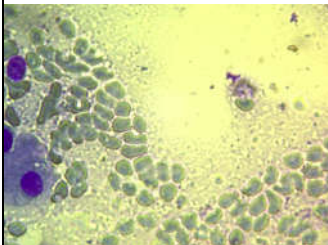
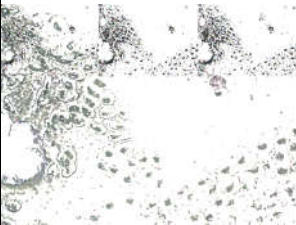

Продовження таблиці 1.4

		<p>Контраст: 1.3 Яскравість : 21 Dilate: 1 Erode: 3</p> <p>Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 75 71</p>	
		<p>Контраст: 2.1 Яскравість : 19 Dilate: 3 Erode: 1</p> <p>Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 59 47</p>	

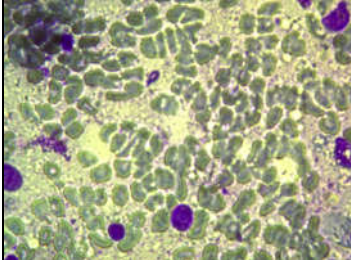

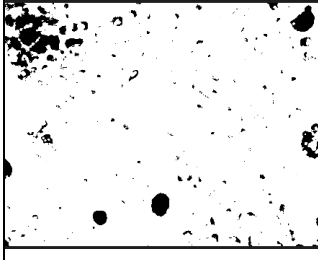
Виходячи, із вищенаведених експериментів, можна зробити висновок, що зображення даного класу потребують обробки, відмінної від двох попередніх класів. Загалом, даний клас зображень характеризується складністю виділення ядер. Особливо увагу потрібно зробити на контрастності, яскравості та ерозії. Лише при збільшенні даних параметрів можна досягнути кращого результату, тобто чітко виділити ядра. Параметри фільтрів не надто відрізняється від попередніх класів. Зображення класу «Цитологія\_проліферативна\_мастопатія» простіше у обробці, порівнюючи і з класом «Цитологія\_непроліферативна\_мастопатія», однак характеризується появою зайвих(непотрібних) елементів на зображенні, що може якщо не знизити якість розпізнавання, то значно сповільнити цей процес.

Приклад порівняння зображень класу «Цитологія\_проліферативна\_мастопатія» наведено у таблиці 1.5.

Таблиця 1.5 – Порівняння зображень для класу «Цитологія\_проліферативна\_мастопатія»

Вхідне зображення	Сегментація (без попер.обр)	Параметри	Сегментація (з попер. Обр.)
		Контраст: 2.1 Яскравість : 19 Dilate: 3 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 55 40	
		Контраст: 2.3 Яскравість : 13 Dilate: 3 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До Після 53 39	
		Контраст: 2.3 Яскравість : 15 Dilate: 7 Erode: 1  Фільтр: Blur:3 Медіанний:27 Середнє знач. Гістограми: До 23 Після 20	

Продовження таблиці 1.5

		<p>Контраст: 2.3  Яскравість : 15  Dilate: 3  Erode: 1</p> <p>Фільтр:  Blur:3  Медіанний:27  Середнє знач.  Гістограми:  До 44  Після 25</p>	
---	---	--	---

В результаті експериментів, можна зробити висновок, що для відносно якісного виділення ядер на даному зображенні потрібно значно підвищити рівень таких параметрів, як контрастність, яскравість та Dilate для збільшення світлих областей на зображенні.

Виходячи із мети в магістерській роботі необхідно :

- проаналізувати метрики порівняння зображень;
- проаналізувати моделі опису кольорових зображень;
- розробити алгоритми визначення відстані між циклами, між ламаними лініями та між багатокутниками;
- розробити програмний модуль обчислення відстані між зображеннями;
- здійснити тестування програмного модуля.

## 2 АЛГОРИТМИ ВИЗНАЧЕННЯ ВІДСТАНІ МІЖ КОНТУРАМИ В МЕТРИЦІ ФРЕШЕ

Форма зображення може бути представлена параметризованими кривими на площині [24]. Зазвичай така крива є простою, тобто не має самоперетинів. Нехай  $X = X(x(t))$  і  $Y = Y(y(t))$  – дві параметризовані криві, в яких (неперервні) функції параметризації  $x(t)$  і  $y(t)$  на  $[0,1]$  задовольняють умовам  $x(0) = y(0) = 0$  і  $x(1) = y(1) = 1$ .

Найбільш використовуваною відстанню параметризованих кривих є мінімум (який береться по всіх монотонно зростаючим параметризаціям  $x(t)$  і  $y(t)$  максимальної евклідової відстані  $d_E(X(x(t)), Y(y(t)))$ ). Це – спеціальний евклідовий випадок відстані собаководів, яка, в свою чергу, є метрикою Фреше для випадку кривих. Варіантами цієї відстані є відкидання умови монотонності параметризації або знаходження частини кривої від якої друга її частина знаходиться на мінімальній такій відстані.

Відстані нелінійного гнучкого узгодження.

Розглянемо дискретне представлення кривих. Нехай  $r \geq 1$  – константа і  $A = \{a_1, \dots, a_m\}$ ,  $B = \{b_1, \dots, b_n\}$  – скінчені впорядковані множини послідовностей точок на замкнутих кривих. Для будь-якої відповідності  $f$ , яка зберігає порядок, між всіма точками  $A$  і всіма точками  $B$  ділянка  $s(a_i, b_j)$  для  $(a_i, f(a_i) = b_j)$  дорівнює  $r$ , якщо  $f(a_{i-1}) = b_j$  або  $f(a_i) = b_{j-1}$ , і дорівнює 0, інакше.

Ослаблена відстань нелінійного гнучкого узгодження є мінімумом по всіх таким  $f$  величини  $\sum (s(a_i, b_j) + d(a_i, b_j))$ , де  $d(a_i, b_j)$  – різниця між дотичними кутами  $a_i$  і  $b_j$ . Вона є майже метрикою для деякого  $r$ . Для  $r=1$  вона називається відстанню нелінійного гнучкого узгодження.

В загальному випадку контури можуть замкнуті та розімкнуті. Замкнутий

контур – це крива перша і остання точки якої співпадають. відповідно розімкнутий контур – це крива перша і остання точки якої не співпадають. Тому розглянемо окремо алгоритми знаходження відстані між замкнутими та розімкнутими ламаними лініями, які представляють відповідно контури. Ламана лінія представляється лінійно апроксимованими відрізками.

Розглянемо чотири випадки, контури представлені ланцюжками, циклами, ломаними лініями і багатокутниками.

## 2.1 Алгоритми визначення відстані між ланцюжками

Розпізнавання  $\varepsilon$ -подібності ланцюжків і обчислення відстані між ними [27,28]. Вихідними для розпізнавання  $\varepsilon$ -подібності ланцюжків є множини  $I = \{0, 1, \dots, m\}$ ,  $J = \{0, 1, \dots, n\}$  і ланцюжки  $\bar{x} = (x_i | i \in I)$ ,  $\bar{y} = (y_j | j \in J)$ . Ці дані визначають прямокутник  $\Pi = I \times J$  і функцію  $q: \Pi \rightarrow \{0, 1\}$  таку, що  $q(i, j) = 1$ , якщо  $d(x_i, y_j) \leq \varepsilon$ ,  $q(i, j) = 0$  і в іншому випадку. Точки  $\pi_1 \in \Pi$ ,  $\pi_2 \in \Pi$  назвемо взаємно досяжними, якщо існує монотонна крива  $\gamma$  така, що  $\pi_1 \in \gamma$ ,  $\pi_2 \in \gamma$  і  $q(\pi) = 1$  для всіх  $\pi \in \gamma$ . Відповідно до твердження 1  $\varepsilon$ -подібність ланцюжків  $\bar{x}$  і  $\bar{y}$  означає, що точка  $(m, n)$  досяжна з точки  $(0, 0)$ . Ця умова перевіряється відомими методами динамічного програмування. Наведемо алгоритм цієї перевірки для повноти викладу і як базу для подальших алгоритмів.

Нехай  $g: \Pi \rightarrow \{0, 1\}$  – функція, значення  $g(i, j)$  якої означає, чи досяжна точка  $(i, j) \in \Pi$  з точки  $(0, 0) \in \Pi$ . Значення функції  $g: \Pi \rightarrow \{0, 1\}$  обчислюються з допомогою алгоритму 1:

- 1)  $g(0, 0) = q(0, 0)$ ;
- 2)  $g(i, 0) = q(i, 0) \wedge g(i-1, 0)$ ,  $i = 1, 2, \dots, m$ ;
- 3)  $g(0, j) = q(0, j) \wedge g(0, j-1)$ ,  $j = 1, 2, \dots, n$ ;
- 4)  $g(i, j) = q(i, j) \wedge [g(i-1, j) \vee g(i, j-1) \vee g(i-1, j-1)]$ ,  $i = 1, \dots, m$ ;  $j = 1, \dots, n$ .

Результат розпізнавання  $\varepsilon$ -подібності представлений числом  $g(m, n)$ .  
Складність алгоритму має порядок  $m \times n$ .

Відстань між ланцюжками обчислюється подібним чином. Алгоритм формує значення  $q(i, j) = d(x_i, y_j)$  для всіх точок  $(i, j) \in \Pi$ , а потім – значення

$$g(i, j) = \min_{\gamma \in \Gamma(i, j)} \max_{(i, j) \in \gamma} d(x_i, y_j),$$

де  $g(i, j)$  – множина всіх монотонних кривих, що починаються в  $(0, 0)$  і закінчуються в  $(i, j)$ . Значення  $g(i, j)$  обчислюються за допомогою алгоритму 2.

- 1)  $g(0, 0) = q(0, 0)$ ;
- 2)  $g(i, 0) = \max\{q(i, 0), g(i-1, 0)\}$ ,  $i = 1, 2, \dots, m$ ;
- 3)  $g(0, j) = \max\{q(0, j), g(0, j-1)\}$ ,  $j = 1, 2, \dots, n$ ;
- 4)  $g'(i, j) = \min\{g(i-1, j), g(i, j-1), g(i-1, j-1)\}$ ,
- 5)  $g(i, j) = \max\{q(i, j), g'(i, j)\}$ ,  $i = 1, \dots, m$ ;  $j = 1, \dots, n$ .

Відстань між ланцюжками є число  $g(m, n)$ , обчислення якого має складність порядку  $m \times n$ .

## 2.2 Алгоритми визначення відстані між циклами

Розпізнавання  $\varepsilon$ -подібності циклів і обчислення відстані між ними. Вихідними даними при розпізнаванні  $\varepsilon$ -подібності циклів є множини  $I = \{0, 1, \dots, m\}$ ,  $J = \{0, 1, \dots, n\}$  і цикли  $\bar{x} = (x_i | i \in I)$ ,  $\bar{y} = (y_j | j \in J)$ . Відповідно до твердження 2  $\varepsilon$ -подібність циклів означає існування певної бімонотонної кривої в прямокутнику  $\Pi = I \times J$ . Дану умову можна модифікувати так, щоб замість пошуку монотонної кривої в прямокутнику  $\Pi = I \times J$  перевіряти існування монотонної кривої в прямокутнику іншого виду. Для множин

$I = \{0, 1, \dots, m\}$  і  $J = \{0, 1, \dots, n\}$  визначимо прямокутник  $\Pi_2$  як декартовий добуток  $\{0, 1, \dots, m\} \times \{0, 1, \dots, 2 \times n\}$ . Циклічний обхід множин  $I$  і  $J$  можна представити як монотонний обхід множин  $\{0, 1, \dots, m\}$  і  $\{j^*, j^* + 1, \dots, j^* + n\}$ , де точка  $(0, j^*)$  – початок обходу. Графік циклічного обходу, в свою чергу, можна представити монотонною кривою в  $\Pi_2$ , що починається в точці  $(0, j^*)$  і що закінчується в точці  $(m, n + j^*)$ .

Визначимо функцію  $q: \Pi_2 \rightarrow \{0, 1\}$  так, що для  $j \leq n$  функція  $q$  приймає значення  $q(i, j) = 1$ , якщо  $d(x_i, y_j) \leq \varepsilon$ , і  $q(i, j) = 0$ , якщо  $d(x_i, y_j) > \varepsilon$ , а для  $j \geq n$  – значення  $q(i, j) = q(i, j - n)$ . В цих термінах  $\varepsilon$ -подібність циклів означає існування такого  $j^*$  і такої монотонної кривої, що починається в  $(0, j^*)$  і закінчується в  $(m, n + j^*)$ , що  $q(i, j) = 1$  в усіх точках цієї кривої. Іншими словами,  $\varepsilon$ -подібність циклів означає існування такого  $j^*$ , що точка  $(m, n + j^*)$  досяжна з точки  $(0, j^*)$ .

Для кожної точки  $(i, j) \in \Pi_2$  визначимо число  $up(i, j)$ , що дорівнює максимальному значенню  $j'$ , при якому  $(i, j)$  досяжна з  $(0, j')$ . Якщо такого  $j'$  нема, то  $up(i, j) = (-1)$ . Подібним чином кожній точці  $(i, j) \in \Pi_2$  поставимо у відповідність число  $up'(i, j)$ , рівне максимальному значенню  $j'$ , при якому  $(m, j')$  досяжна з  $(i, j)$ , и рівне  $(-1)$ , якщо такого  $j'$  не існує. Очевидно, що якщо точка  $(m, n + j^*)$  досяжна з  $(0, j^*)$ , то  $up(m, n + j^*) \geq j^*$  і  $up'(0, j^*) \geq n + j^*$ . Менш очевидно, проте вірно, що якщо  $up(m, n + j^*) \geq j^*$  і  $up'(0, j^*) \geq n + j^*$ , то  $(m, n + j^*)$  досяжна з  $(0, j^*)$ . Тому перевірка  $\varepsilon$ -подібності циклів зводиться до відшукування  $j^*$ , що задовольняє нерівності  $up(m, n + j^*) \geq j^*$  і  $up'(0, j^*) \geq n + j^*$ . При відомих числах  $up(i, j)$  і  $up'(i, j)$  складність цього пошуку має порядок  $n$ . Числа  $up(i, j)$  обчислюються за час, що лінійно залежить від  $m \times n$ , з допомогою наступного алгоритму 3.

- 1) якщо  $q(0, 0) = 0$ , то  $up(0, 0) = (-1)$ ;  
в протилежному випадку  $up(0, 0) = 0$ ;
- 2) для  $i = 1, 2, \dots, m$   
якщо  $q(i, 0) = 0$ , то  $up(i, 0) = (-1)$ ;

в протилежному випадку  $up(i, 0) = up(i-1, 0)$ ;

3) для  $j = 1, 2, \dots, 2 \cdot n$

якщо  $q(0, j) = 0$ , то  $up(0, j) = (-1)$ ;

в протилежному випадку  $up(0, j) = j$ ;

4) для  $i = 1, 2, \dots, m$  и  $j = 1, 2, \dots, 2 \cdot n$

якщо  $q(i, j) = 0$ , то  $up(i, j) = (-1)$ ;

в протилежному випадку  $up(i, j) = \max \{up(i-1, j), up(i, j-1), up(i-1, j-1)\}$ .

Аналогічно обчислюються і числа  $up'(i, j)$ , і таким чином, складність розпізнавання  $\varepsilon$ -подібності циклів має порядок  $m \times n$ . Обчислення відстані між циклами зводиться до багаторазового розпізнання  $\varepsilon$ -подібності, яке виконується для певних тестових значень  $\varepsilon$ . Шукане значення відстані належить множині  $\{d(x_i, y_j) \mid i \in I, j \in J\}$ , що складається не більше ніж з  $m \times n$  елементів. Цю множину слід упорядкувати за час порядку  $(m \times n) \log(m \times n)$ , а потім знайти шукану відстань, визначаючи  $\varepsilon$ -подібність не більше ніж для  $\log(m \times n)$  тестових значень.

Таким чином, обчислення відстані між циклами має порядок складності не гірший, ніж  $(m \times n) \log(m \times n)$ .

### 2.3 Алгоритми визначення відстані між ламаними лініями

Розпізнавання  $\varepsilon$ -подібності ламаних ліній. Вихідні дані для розпізнавання  $\varepsilon$ -подібності ламаних ліній  $X$  і  $Y$  представлені множинами  $I = \{0, 1, 2, \dots, m\}$  і  $J = \{0, 1, 2, \dots, n\}$  номерів вершин і їх координатами  $(x_i \mid i \in I)$ ,  $(y_j \mid j \in J)$ . Ці дані визначають довжини  $U$  і  $V$  ламаних ліній, прямокутник  $\Pi = \{u \mid 0 \leq u \leq U\} \times \{v \mid 0 \leq v \leq V\}$  і його підмножину  $\Pi(\varepsilon) = \{(u, v) \in \Pi \mid d(x(u), y(v)) \leq \varepsilon\}$ . Число  $u$  – горизонтальна координата точки  $(u, v) \in \Pi$ , а число  $v$  – її вертикальна



координата. У відповідності з твердженням 3  $\varepsilon$ -подібність ламаних ліній означає, що при цьому значенні  $\varepsilon$  точка  $(U, V)$  досяжна з точки  $(0, 0)$ . Як і при розпізнаванні подібності ланцюжків (див. алгоритм 1), основна ідея розпізнавання ламаних ліній полягає в послідовному нарощуванні множини точок, досяжних з  $(0, 0)$ . Однак в даному випадку  $\Pi$ ,  $\Pi(\varepsilon)$  і області досяжності - це не кінцеві, а нескінченні множини. Покажемо, що і в цьому випадку розпізнавання  $\varepsilon$ -подібності зводиться до кінцевих обчислень.

Представимо прямокутник  $\Pi$  як об'єднання  $m \times n$  підмножин  $\Pi(i, j) = \{(u, v) \mid u(x_{i-1}) \leq u \leq u(x_i), v(y_{j-1}) \leq v \leq v(y_j)\}$ ,  $i \in \{1, 2, \dots, m\}$ ,  $j \in \{1, 2, \dots, n\}$ , які назвемо комірками. Нехай  $\gamma(i, j) = \gamma \cap \Pi(i, j)$  - ділянка цієї кривої в комірці  $\Pi(i, j)$ . В силу опуклості відстані  $d: \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$  множина  $\Pi(\varepsilon) \cap \Pi(i, j)$  опукла для кожної пари  $(i, j)$ . Нехай  $\gamma$  - монотонна крива, що починається в  $(0, 0)$ , що закінчується в  $(U, V)$  і повністю лежить в  $\Pi(\varepsilon)$ . Нехай  $\gamma(i, j) = \gamma \cap \Pi(i, j)$  - ділянка цієї кривої в комірці  $\Pi(i, j)$ . Визначимо іншу монотонну криву  $\gamma'$  так, що якщо  $\gamma(i, j) = \emptyset$ , то  $\gamma'(i, j) = \emptyset$ , інакше  $\gamma'(i, j)$  - це прямолінійний відрізок, кінці якого збігаються з кінцями  $\gamma(i, j)$ . Оскільки  $\gamma(i, j) \subset \Pi(\varepsilon) \cap \Pi(i, j)$ , а множина  $\Pi(\varepsilon) \cap \Pi(i, j)$  - опукла, то  $\gamma'(i, j) \subset \Pi(\varepsilon) \cap \Pi(i, j)$ . Отже,  $\gamma'$  - теж монотонна крива, що починається в  $(0, 0)$ , що закінчується в  $(U, V)$  і повністю лежить в  $\Pi(\varepsilon)$ . Таким чином, пошук підходящої монотонної кривої зводиться до пошуку кінцевої послідовності точок в яких шукана крива перетинає границі комірок. Щоб знайти цю послідовність слід шукати не множину  $g$  всіх точок, досяжних з  $(0, 0)$ , а тільки її перетин з межами комірок.

Введемо позначення

$$qhor(i, j) = \{u \mid (u, v(y_j)) \in \Pi(\varepsilon), u(x_{i-1}) \leq u \leq u(x_i)\},$$

$$qver(i, j) = \{v \mid (u(x_i), v) \in \Pi(\varepsilon), v(y_{j-1}) \leq v \leq v(y_j)\}.$$

Підмножини  $qhor(i, j)$  і  $qver(i, j)$  – це замкнуті інтервали, і їх можна представити кінцевою сукупністю даних, обсяг якої має порядок  $m \times n$ . Кожна підмножина представляється бінарною міткою, що позначає, чи порожнє воно, і двома його граничними точками, якщо воно не порожнє. Складність обчислення цих даних має порядок  $m \times n$ .

Перетини множини  $g$  точок, досяжних з  $(0, 0)$ , з межами комірок позначимо

$$ghor(i, j) \subset qhor(i, j) \text{ і } gver(i, j) \subset qver(i, j),$$

$$ghor(i, j) = \{u \mid (u, v(y_j)) \in g, u(x_{i-1}) \leq u \leq u(x_i)\},$$

$$gver(i, j) = \{v \mid (u(x_i), v) \in g, v(y_{j-1}) \leq v \leq v(y_j)\}.$$

Це теж замкнуті інтервали, і їх теж можна представити сукупністю чисел, обсяг якої має порядок  $m \times n$ . Підмножини  $gver(i, j)$  і  $ghor(i, j)$  будуються на підставі підмножин  $qver(i, j)$  і  $qhor(i, j)$  за допомогою наступного алгоритму 4.

1) якщо  $d(x_0, y_0) > \varepsilon$ , то  $ghor(1, 0) = gver(0, 1) = \emptyset$ ;

в протилежному випадку  $ghor(1, 0) = qhor(1, 0)$ ,  $gver(0, 1) = qver(0, 1)$ ;

2) для  $i \in \{2, 3, \dots, m\}$ ,

якщо  $ghor(i-1, 0) \cap qhor(i, 0) = \emptyset$ , то  $ghor(i, 0) = \emptyset$ ;

в протилежному випадку  $ghor(i, 0) = qhor(i, 0)$ ;

3) для  $j \in \{2, 3, \dots, n\}$ ,

якщо  $gver(0, j-1) \cap qver(0, j) = \emptyset$ , то  $gver(0, j) = \emptyset$ ;

в протилежному випадку  $gver(0, j) = qver(0, j)$ ;

4) для  $i \in \{1, 2, \dots, m\}$  и  $j \in \{1, 2, \dots, n\}$ ,

якщо  $ghor(i, j-1) \neq \emptyset$ , то  $gver(i, j) = qver(i, j)$ ;

в протилежному випадку  $gver(i, j) = \{v \in qver(i, j) \mid v \geq \min gver(i-1, j)\}$ ;

якщо  $gver(i-1, j) \neq \emptyset$ , то  $ghor(i, j) = qhor(i, j)$ ;

в протилежному випадку  $ghor(i, j) = \{u \in qhor(i, j) \mid u \geq \min qhor(i, j-1)\}$ .

Умови виду  $x \geq \min P$  в п.4 алгоритму 4 перевіряються і для  $P = \emptyset$ . У цьому випадку передбачається, що нерівність  $x \geq \min P$  помилкова для будь-якого  $x$  і, отже,  $\{x \mid x \geq \min P\} = \emptyset$ . Результат розпізнавання  $\varepsilon$ -подібності ламаних ліній представлений підмножинами  $ghor(m, n)$  і  $gver(m, n)$ . Точка  $(U, V)$  або міститься в обох підмножинах, або не міститься ні в одній з них. У першому випадку  $H(X, Y) \leq \varepsilon$ , і  $H(X, Y) > \varepsilon$  – в іншому. Обчислювальна складність алгоритму має порядок  $m \times n$ .

В [11] показано, що обчислення відстані між ламаними лініями можна звести до багаторазового, але не більше ніж  $\log(m \times n)$ -кратного розпізнавання  $\varepsilon$ -подібності.

## 2.4 Алгоритми визначення відстані між багатокутниками

Розпізнавання  $\varepsilon$ -подібності багатокутників. Нехай  $X$  і  $Y$  – багатокутники з периметрами  $U$  і  $V$ ,  $I = \{0, 1, \dots, m\}$  і  $J = \{0, 1, \dots, n\}$  – множини номерів їх вершин,  $(x_i \in R^k \mid i \in I)$  і  $(y_j \in R^k \mid j \in J)$  – координати цих вершин. Відповідно до твердження 4  $\varepsilon$ -подібність цих багатокутників означає існування бімонотонної кривої в прямокутнику  $\Pi = \{u \mid 0 \leq u \leq U\} \times \{v \mid 0 \leq v \leq V\}$ , для кожної точки  $(u, v)$  якої виконується нерівність  $d(x(u), y(v)) \leq \varepsilon$ . Як і при розпізнаванні  $\varepsilon$ -подібності циклів, пошук бімонотонної кривої в прямокутнику  $\Pi$  можна замінити пошуком монотонної кривої в прямокутнику  $\Pi_2 = \{u \mid 0 \leq u \leq U\} \times \{v \mid 0 \leq v \leq 2V\}$ . А саме, два багатокутники  $\varepsilon$ -подібні, якщо існує таке число  $v^*$ , що точка  $(U, V + v^*)$  досяжна з точки  $(0, v^*)$ . Для аналізу цієї умови сформулюємо допоміжні поняття.

Будь-якій парі  $x \in X$ ,  $y \in Y$  відповідають дві точки в прямокутнику  $\Pi_2$ :

$(u(x), v(y))$  і  $(u(x), V+v(y))$ , де  $u(x)$  – довжина ламаної лінії від  $x_0$  до  $x$ , а  $v(y)$  – довжина ламаної лінії від  $y_0$  до  $y$ . Кожній точці  $(u, v) \in \Pi_2$  відповідає пара  $x(u) \in X$ ,  $y(v) \in Y$  така, що довжина ламаної лінії від  $x_0$  до  $x(u)$  дорівнює  $u$ , а довжина ламаної лінії від  $y_0$  до  $y(v)$  дорівнює  $v$ , якщо  $v \leq V$ , і дорівнює  $v-V$ , якщо  $v \geq V$ .

Для  $i \in \{1, 2, \dots, m\}$  і  $j \in \{0, 1, \dots, n\}$  визначимо підмножини

$$\text{Phor}(i, j) = \{(u, v(y_j)) \mid u(x_{i-1}) \leq u \leq u(x_i)\} \text{ і } \text{Phor}(i, j+n) = \{(u, v+V) \mid (u, v) \in \text{Phor}(i, j)\}.$$

Подібні підмножини визначимо для  $i \in \{0, 1, \dots, m\}$  і  $j \in \{1, 2, \dots, n\}$  так, що

$$\text{Pver}(i, j) = \{(u(x_i), v) \mid v(y_{j-1}) \leq v \leq v(y_j)\} \text{ і } \text{Pver}(i, j+n) = \{(u, v+V) \mid (u, v) \in \text{Pver}(i, j)\}.$$

Точку  $(u^*, v^*)$  назвемо досяжною зліва, якщо існує  $v$ , таке що  $(u^*, v^*)$  досяжна з  $(0, v)$ , і досяжною справа, якщо існує  $v$ , таке що  $(U, v)$  досяжна з  $(u^*, v^*)$ . Позначимо  $g$  множину точок, досяжних зліва, і  $g'$  – досяжних справа. Для кожної точки  $(u^*, v^*) \in g$  визначимо число  $up(u^*, v^*)$  – максимальне значення  $v$ , при якому  $(u^*, v^*)$  досяжна з  $(0, v)$ . Для кожної точки  $(u^*, v^*) \in g'$  визначимо число  $up'(u^*, v^*)$  – максимальне значення  $v$ , при якому  $(U, v)$  досяжна з  $(u^*, v^*) \in g'$ .

Значення  $up$  і  $up'$  аналогічні числам  $up(i, j)$  і  $up'(i, j)$  при розпізнаванні  $\varepsilon$ -подібності циклів. Очевидно, що якщо точка  $(U, V+v^*)$  досяжна з  $(0, v^*)$ , то  $up(U, V+v^*) \geq v^*$  і  $up'(0, v^*) \geq V+v^*$ . Менш очевидно, проте вірно, що якщо  $up(U, V+v^*) \geq v^*$  і  $up'(0, v^*) \geq V+v^*$ , то  $(U, V+v^*)$  досяжна з  $(0, v^*)$ . Тому  $\varepsilon$ -подібність багатокутників рівнозначна існуванню такого числа  $v^*$ , що

$$(0, v^*) \in g', up'(0, v^*) \geq V+v^*, (U, V+v^*) \in g, up(U, V+v^*) \geq v^*. \quad (2.1)$$

Як і для розпізнавання  $\varepsilon$ -подібності ламаних ліній, для перевірки умови (2.1) не треба знати повністю множини  $g$  і  $g'$ , а тільки їх підмножини

$$gver(i, j) = g \cap \text{Pver}(i, j), ghor(i, j) = g \cap \text{Phor}(i, j), \quad (2.2)$$

$$gver'(i, j) = g' \cap \text{П}ver(i, j), ghor'(i, j) = g' \cap \text{П}hor(i, j). \quad (2.3)$$

Ті підмножини (2.2), (2.3), які не порожні, є вертикальними або горизонтальними відрізками в прямокутнику П2. Кожна з підмножин (2.2), (2.3) задається бінарною міткою, що позначає, чи є вона порожньою, і координатами крайніх точок. Сумарний обсяг цих даних має порядок  $m \times n$ . Підмножини (2.2), (2.3) будуються подібно до того, як будуються підмножини  $gver(i, j)$ ,  $ghor(i, j)$  с допомогою алгоритму 4 при розпізнаванні  $\varepsilon$ -подібності ламаних ліній. Трудомісткість цієї побудови має порядок  $m \times n$ .

Функції  $up$  і  $up'$  також не потрібно знати повністю. Для перевірки умови (5) досить знати звуження  $up$  на підмножини  $gver(i, j)$ ,  $ghor(i, j)$  і звуження  $up'$  на підмножини  $gver'(i, j)$ ,  $ghor'(i, j)$ . Їх в свою чергу можна представити кінцевою сукупністю чисел. Функція  $up'$  на будь-якій підмножині  $gver'(i, j)$  приймає постійне значення, яке позначимо  $UPver'(i, j)$ . На множинах  $ghor(i, j)$  функція  $up$  приймає постійне значення, яке позначимо  $UPhor(i, j)$ . Звуження функції  $up$  на підмножини  $gver(i, j)$  і функції  $up'$  на підмножини  $ghor'(i, j)$  мають більш складний вид. Покажемо, як можна представити функцію  $up$  на підмножинах  $gver(i, j)$ .

Виберемо деяку пару  $(i^*, j^*)$  і зафіксуємо її для подальшого розгляду. Для будь-якої точки  $(u(x_{i^*}), v) \in gver(i^*, j^*)$  точка  $(0, up(u(x_{i^*}), v))$ , з якої досяжна  $(u(x_{i^*}), v)$ , належить або підмножині  $\text{П}ver(0, j^*)$ , або підмножині  $\text{П}ver(0, j')$ ,  $j' < j^*$ . Розглянемо ці ситуації.

Нехай  $(0, up(u(x_{i^*}), v)) \in \text{П}ver(0, j^*)$ . Множина значень  $v$ , що задовольняють цій умові, утворює інтервал, межі якого позначимо  $begin$  і  $end$ . Множина значень функції  $up$  на цьому інтервалі – це теж інтервал, верхню межу якого позначимо  $value$ . Величина  $up(u(x_{i^*}), v)$  дорівнює  $value$  на підмножині  $\{v \mid value \leq v \leq end\}$  і дорівнює  $v$  на підмножині  $\{v \mid begin \leq v \leq value\}$ . Таким чином,

на інтервалі  $\{v \mid \text{begin} \leq v \leq \text{end}\}$  величина  $up(u(x_{i^*}), v)$  повністю визначається значенням  $value$  так, що  $up(u(x_{i^*}), v) = \min\{v, value\}$ .

Нехай  $(0, up(u(x_{i^*}), v)) \in Pver(0, j')$ ,  $j' < j^*$ . На множині значень  $v$ , які відповідають цій умові, функція  $up$  є кусково-постійною неспадною функцією від  $v$ , яка приймає не більше, ніж  $i^*$  значень. Дійсно, монотонна крива, що з'єднує точки  $(u(x_{i^*}), v)$  і  $(0, up(u(x_{i^*}), v)) \in Pver(0, j')$ , перетинає одну з підмножин  $ghor(i, j^* - 1)$ ,  $i = 1, 2, \dots, i^*$ . На кожному з них функція  $up$  приймає постійне значення  $UPhor(i, j^* - 1)$ . Якщо монотонна крива, що з'єднує точки  $(u(x_{i^*}), v)$  і  $(0, up(u(x_{i^*}), v))$ , перетинає  $ghor(i, j^* - 1)$ , то  $up(u(x_{i^*}), v) = UPhor(i, j^* - 1)$ .

Таким чином, звуження функції  $up$  на підмножину  $gver(i, j)$  задається множиною трійок вигляду  $(\text{begin}, \text{end}, \text{value})$ , яку позначимо  $Intver(i, j)$ . Множина  $Intver(i, j)$  трійок визначає розбиття вертикального відрізка  $gver(i, j)$  на непересічні інтервали, кількість яких не перевищує  $i+1$ . Множина інтервалів упорядкована в звичайному сенсі цього слова. А саме, з двох різних інтервалів один вище, а інший нижче, в непорожній множині обов'язково є найвищий і найнижчий інтервал і т.п. Множина  $Intver(i, j)$  визначає величину  $up(u, v)$  для будь-якої точки  $(u(x_i), v) \in gver(i, j)$ . Для цієї точки слід знайти трійку  $(\text{begin}, \text{end}, \text{value}) \in Intver(i, j)$ , таку що:

- 1) або  $\text{begin} \leq v \leq \text{end}$ ,  $value \geq v(y_{j-1})$ ;
- 2) або  $\text{begin} \leq v < \text{end}$ ,  $value < v(y_{j-1})$ .

Величина  $(up(u(x_i), v))$  дорівнює  $\min\{value, v\}$  в першому випадку і дорівнює  $value$  в другому.

Якщо для кожного  $j \in \{1, 2, \dots, n\}$  відоме число  $UPver'(0, j)$  і множина  $Intver(m, j+n)$ , то перевірку умови (5) виконує наступний алгоритм 5.

- 1) знайти  $j \in \{1, 2, \dots, n\}$ , таке що  $UPver'(0, j) \geq \max\{v \mid (U, v) \in gver(m, j+n)\}$ ;

якщо таке  $j$  не існує, то представлення багатокутники не є  $\varepsilon$ -подібними;

2) для всіх тих  $j$ , які задовольняють умову п.1, і всіх трійок

$(begin, end, value) \in Intver(m, j+n)$  перевірити умови

2.1)  $(0, value) \in Pver(0, j+n)$ ;

2.2) існує  $v$ , таке що  $begin \leq v < end$  і  $value \geq v - V$ ;

3) якщо знайдено  $j$ , для якого виконалося по крайній мірі одна з умов 2.1 або 2.2,

то умова (5) виконана і представлені багатокутники  $\varepsilon$ -подібні;

якщо таке  $j$  не знайдено, представлені багатокутники не є  $\varepsilon$ -подібними.

Кількість значень  $j \in \{1, 2, \dots, n\}$ , для яких перевіряються умови 2.1 і 2.2, не перевищує  $n$ . Кількість трійок  $(begin, end, value) \in Intver(m, j+n)$ , що перевіряються для кожного перевіреного значення  $j$ , не перевищує  $m$ . Для кожного  $j$ , що перевіряється і кожної такої трійки  $(begin, end, value)$  умови 2.1 і 2.2 перевіряються за обмежений час, що не залежить від  $m$  і  $n$ . Таким чином, при відомих числах  $UPver'(0, j)$  і множинах  $Intver(m, j+n)$ ,  $j \in \{1, 2, \dots, n\}$ , складність алгоритму 5 має порядок  $m \times n$ .

Покажемо, як слід будувати множини  $Intver(i, j)$ ,  $i \in \{0, 1, 2, \dots, m\}$ ,  $j \in \{1, 2, \dots, 2 \cdot n\}$ , частина яких, а саме, множини  $Intver(m, j)$ ,  $j \in \{n+1, n+2, \dots, 2 \cdot n\}$ , необхідна для виконання алгоритму 5. Для побудови цих множин необхідні відомості про множини  $gver(i, j)$  і числа  $UPhor(i, j)$ . Ми припускаємо, що множини  $gver(i, j)$  вже побудовані за допомогою алгоритму, що аналогічний алгоритму 4 для розпізнавання  $\varepsilon$ -подібності ламаних ліній. Обчислення чисел  $UPhor(i, j)$  буде показане одночасно з побудовою множин  $Intver(i, j)$ .

Ключова ідея описаного нижче алгоритму полягає в тому, що в ньому не передбачена індивідуальна пам'ять для кожної з множин  $Intver(i, j)$ ,  $i \in \{0, 1, 2, \dots, m\}$ ,  $j \in \{1, 2, \dots, 2 \cdot n\}$ . Для кожного  $j^* \in \{1, 2, \dots, 2 \cdot n\}$  передбачена пам'ять, загальна для всіх множин  $Intver(i, j^*)$ ,  $i \in \{0, 1, 2, \dots, m\}$ . На якомусь одному кроці роботи алгоритму пам'ять  $Intver^*(j^*)$  представляє множину  $Intver(i, j^*)$  для якогось одного  $i$ , а на якомусь іншому кроці – для іншого. Саме така

організація пам'яті дозволяє побудувати множини  $Intver(m, j)$ ,  $j \in \{n+1, n+2, \dots, 2 \cdot n\}$ , за час, що лінійно залежить від  $m \times n$ , і в кінцевому підсумку розпізнати  $\varepsilon$ -подібність за час того ж порядку. Пам'ять  $Intver^*(j)$  являє собою чергу магазинного типу, в якій записана множина трійок вигляду  $(begin, end, value)$ . Черга допускає виконання наступних операцій:

- перевірити, чи є черга порожньою чи ні;
- очистити чергу; в результаті цієї операції черга стає порожньою;
- прочитати саму верхню або саму нижню трійку з черги;
- виключити з черги найвищу або найнижчу трійку;
- записати трійку виду  $(begin, end, value)$  в самий верх або самий низ

черги.

В силу впорядкованості множини трійок, що знаходяться в черзі, кожна операція виконується за постійний час, який не залежить від вмісту черги. Ініціалізація алгоритму полягає у визначенні значень  $UPhor(i, 0)$ ,  $i \in \{1, 2, \dots, m\}$ , і початкового стану черг  $Intver^*(j)$ ,  $j \in \{1, 2, \dots, 2 \cdot n\}$ . Значення  $UPhor(i, 0)$  дорівнює 0, якщо  $ghor(i, 0) \neq \emptyset$ , і не визначене в іншому випадку. Черга  $Intver^*(j)$  в своєму початковому стані порожня, якщо  $gver(0, j) = \emptyset$ , і містить єдину трійку  $(begin = bottom(0, j), end = top(0, j), value = top(0, j))$ , якщо  $gver(0, j) \neq \emptyset$ . Тут і далі прийняті позначення

$$top(i, j) = \max\{v \mid (u, v) \in gver(i, j)\}, bottom(i, j) = \min\{v \mid (u, v) \in gver(i, j)\}.$$

Алгоритм виконує роботу за кроками. Кожен крок відповідає певній парі  $(i, j)$ ,  $i \in \{1, 2, \dots, m\}$ ,  $j \in \{1, 2, \dots, 2 \cdot n\}$ . Крок з номером  $(i, j)$  виконується після того, як обчислено значення  $UPhor(i, j-1)$ , а в черзі  $Intver^*(j)$  записана впорядкована множина  $Intver(i-1, j)$ . На підставі цих даних величина  $UPhor(i, j)$  визначається за наступним алгоритмом 6.

Якщо  $ghor(i, j) = \emptyset$ , то  $UPhor(i, j)$  не визначено;

в протилежному випадку



{ якщо  $gver(i-1, j) \neq \emptyset$ ,  
 то { прочитати саму верхню трійку  $(begin, end, value)$  з черги  $Intver^*(j)$ ;  
      $UPhor(i, j) = value$ ;  
 }  
 в протилежному випадку  $UPhor(i, j) = UPhor(i, j-1)$ .  
 }

Сумарна по всім  $(i, j)$  складність цих операцій має порядок  $m \times n$ . Нехай до початку  $(i, j)$ -го кроку отримано значення  $UPhor(i, j-1)$ , а черга  $Intver^*(j)$  представляє множину  $Intver(i-1, j)$ . В результаті  $(i, j)$ -го кроку черга  $Intver^*(j)$  повинна модифікуватися так, щоб представляти множину  $Intver(i, j)$ . Очевидно, що якщо  $gver(i, j) = \emptyset$ , то в результаті  $(i, j)$ -го кроку черга  $Intver^*(j)$  повинна стати пустою. Якщо  $gver(i, j) \neq \emptyset$ , а  $gver(i-1, j) = \emptyset$ , то в результаті  $(i, j)$ -го кроку черга  $Intver^*(j)$  повинна складатися з єдиною трійки  $(begin = bottom(i, j), end = top(i, j), value = UPhor(i, j-1))$ , яка стає одночасно і самою верхньою, і найнижчою в черзі. Якщо  $gver(i, j) \neq \emptyset$  і  $gver(i-1, j) \neq \emptyset$ , то з верхньої та нижньої частини черги  $Intver^*(j)$  слід виключити деяку, можливо, порожню підмножину трійок, записати одну трійку в самий верх черги і записати одну або дві трійки в самий низ черги. Це перетворення виконує алгоритм 7:

- 1) Якщо черга  $Intver^*(j)$  непуста,
- то { прочитати з  $Intver^*(j)$  саму верхню трійку  $(begin, end, value)$ ;
- виключити з  $Intver^*(j)$  саму верхню трійку;
- якщо  $begin \leq top(i, j)$ ,  
 то { записати  $(begin, top(i, j), value)$  в самий верх черги;  
     перейти на команду 2; }  
 перейти на команду 1;
- }
- 2) якщо черга  $Intver^*(j)$  непуста,
- то { прочитати з  $Intver^*(j)$  саму нижню трійку  $(begin, end, value)$ ;

- виключити з  $Intver^*(j)$  саму нижню трійку;
  - якщо  $end \geq bottom(i, j)$ ,  
то { записати  $(\max\{begin, bottom(i, j)\}, end, value)$  в самий низ черги;  
перейти на команду 3; }
  - перейти на команду 2;
- }
- 3) якщо  $ghor(i, j-1) \neq \emptyset$ ,
- то { якщо  $Intver^*(j) = \emptyset$ ,
- то записати трійку  $(bottom(i, j), top(i, j), UPhor(i, j-1))$  в чергу;
- в протилежному випадку
- { прочитати саму нижню трійку  $(begin, end, value)$  з черги;
- якщо  $begin > bottom(i, j)$ ,
- то записати трійку  $(bottom(i, j), begin, UPhor(i, j-1))$  в самий низ черги.
- }
- }

Деякі команди в наведеному алгоритмі відзначені міткою •, а інші не зазначені. Кожна невідзначена команда виконується не більше одного разу на кожному  $(i, j)$ -му кроці. Тому загальний час виконання всіх  $m \times n$  кроків складається з часу порядку  $m \times n$  і загального часу виконання зазначених команд. Запис трійок в черзі виконують тільки невідмічені команди, тому загальна кількість записів має порядок  $m \times n$ . Виконання кожної зазначеної команди супроводжується винятком деякої трійки з черги. Тому сумарний час роботи всіх зазначених команд має порядок, що не перевищує  $m \times n$ , тому що кількість винятків трійок з черги не перевищує кількість записів. Таким чином, сумарний час перетворення черг має порядок  $m \times n$ .

На кожному  $(i, j)$ -му кроці спільної роботи алгоритмів 6 і 7 обчислюється число  $UPhor(i, j)$  і будується множина  $Intver(i, j)$ . Результатом всіх  $m \times n$  кроків цієї спільної роботи є множини  $Intver(m, j+n)$ ,  $j \in \{1, 2, \dots, n\}$ , що утворюють частину вихідних даних для алгоритму 5. Решта вихідні для алгоритму 5 дані -

це числа  $UPver'(0, j)$ ,  $j \in \{1, 2, \dots, n\}$ . Ці числа формуються в процесі спільної роботи двох інших алгоритмів. На кожному  $(i, j)$ -му кроці перший з них обчислює значення  $UPver'(i, j)$  на підставі раніше обчислених значення  $UPver'(i+1, j)$  і множини  $Inthor'(i, j+1)$ , що визначає звуження функції  $up'$  на підмножину  $g'(i, j+1)$ . Другий алгоритм на підставі значення  $UPver'(i+1, j)$  і множини  $Inthor'(i, j+1)$  формує множину  $Inthor'(i, j)$ . Перший алгоритм працює аналогічно алгоритму 6, а другий – алгоритму 7.

У статті [27] показано, яким чином обчислення відстані між багатокутниками зводиться до багаторазового, але не більше, ніж  $\log(m \times n)$ -кратного розпізнавання  $\varepsilon$ -подібності.

## 2.5 Алгоритми визначення дискретної відстані між контурами

Постановка задачі. Нехай задано вхідне зображення  $Im$ , над яким необхідно провести процедуру сегментації [28-30]. Позначимо процедуру сегментації через  $S$ ; вона є скінченною множиною алгоритмів сегментації:  $S = \{A_1, A_2, \dots, A_k\}$ , де  $A_i$  – алгоритм сегментації. Процедура сегментації в кожному випадку є певною конкретною комбінацією алгоритмів сегментації або алгоритму з різними параметрами  $S_j, j = \overline{1, l}$ , де  $l$  – кількість комбінацій.

Після сегментування початкового зображення експертом отримуємо зображення  $Im_e$ , тобто  $S_e(Im) = Im_e$ . Представимо дане зображення у вигляді  $Im_e = C_{ei} \cup O_{ei}, i = \overline{1, s}$ , де  $C_{ei}$  –  $i$ -ий контур області, виділеної експертом,  $O_{ei}$  –  $i$ -а область зображення.

Після кожної процедури сегментації вхідного зображення  $Im$  отримаємо відповідно такі зображення (рисунок 2.1):

$$S_1(Im) = Im_1, S_2(Im) = Im_2, \dots, S_k(Im) = Im_k, k = \overline{1, z}$$

$$i \quad Im_1 = C_{1i} \cup O_{1i}, i = \overline{1, m}, Im_2 = C_{2i} \cup O_{2i}, i = \overline{1, p}, \dots, Im_k = C_{ki} \cup O_{ki}, i = \overline{1, r}.$$

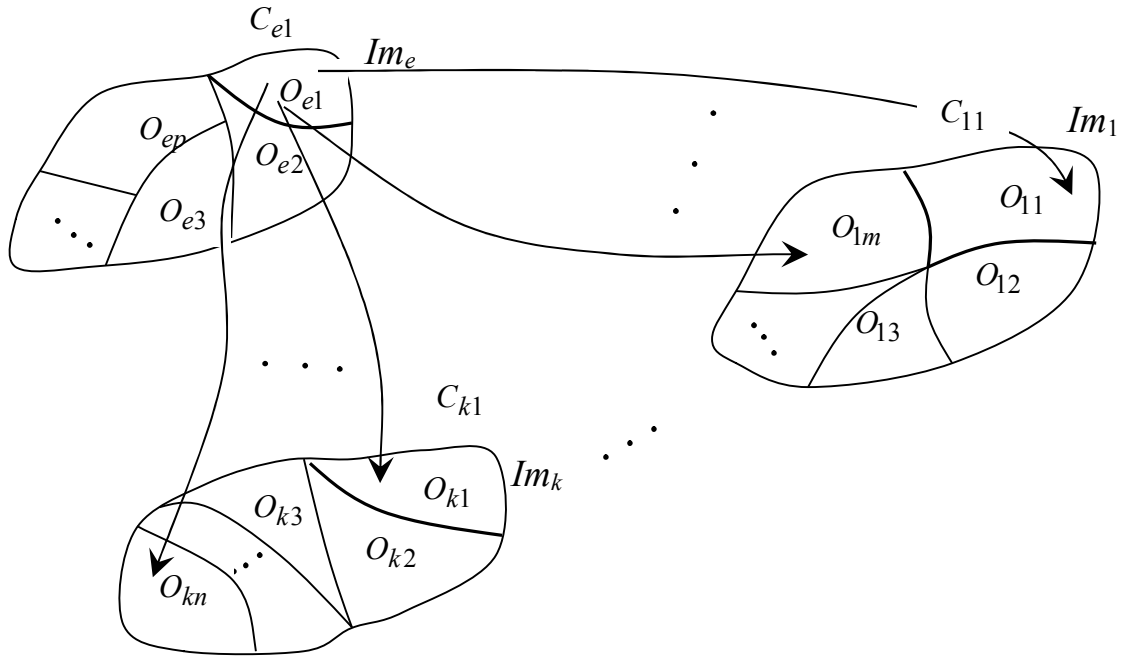


Рисунок 2.1 – Загальна постановка задачі

Для порівняння контурів сегментованих зображень використаємо метрику Фреше. Для функції (параметричного задання)  $f_1$  контура  $C_{ei}$  сегментованого зображення експертом і функції  $f_2$  контура  $C_{li}$ , сегментованого певним алгоритмом (алгоритмами), відстань Фреше рівна:

$$d_F(f_1, f_2) = \inf_{\alpha, \beta} \max_t d(f_1(\alpha(t)), f_2(\beta(t))),$$

де  $\alpha, \beta$  – неперервні неспадні функції з проміжку  $[0,1]$  на  $[0,1]$ .

Після розбиття вхідного зображення експертом і алгоритмом (алгоритмами) сегментації отримуємо області

$$O_e = O_{ei} \cup \dots \cup O_{en} | P_1 ,$$

$$O_1 = O_{1i} \cup \dots \cup O_{1m} | P_2 .$$

Тоді  $d(P_1, P_2) = \inf\{c > 0 \mid \forall i = \overline{1, n}, \exists j = \overline{1, m}$  таке, що  $d_H(O_{ei}, O_{1j}) \leq c$ ; і навпаки  $\forall j = \overline{1, m}, \exists i = \overline{1, n}$  таке, що  $d_H(O_{1j}, O_{ei}) \leq c\}$ , де  $d_H$  – відстань Хаусдорфа [31-33].

Тоді відстань між зображеннями  $Im_e$  та  $Im_i$  після кожної розбивки буде відповідно рівна:

$$D_i = \alpha d_{F_i} + \beta d_{H_i}, \alpha + \beta = 1, \text{ де } \alpha, \beta = \text{const} \in [0, 1].$$

Коефіцієнти  $\alpha$ ,  $\beta$  вибираються апріорно з деяких практичних міркувань. Необхідно знайти таке  $Im_i$ ,  $i = \overline{1, k}$ , щоб  $D_i = D_{\min}$ .

У роботі розглядаються алгоритми для контурів, які задані полігональними кривими, та опуклих областей.

Допоміжні поняття і результати. Отже, для заданого зображення  $Im$  проводимо сегментацію набором алгоритмів за допомогою процедури  $S_j$ :

$S_j = \{A_1, \dots, A_n\}$ . Після кожної процедури сегментації отримуємо такі сегментовані зображення  $S_1(Im) = Im_1$ , ...,  $S_k(Im) = Im_k$ . Еталонну сегментацію проводить експерт  $S_e(Im) = Im_e$ .

Представимо сегментовані зображення у вигляді сукупності контурів (зовнішніх границь області) і областей (внутрішня множина точок)

$$Im_1 = \bigcup_{i=1}^{m_1} C_{1i} \cup O_{1i}, \quad Im_2 = \bigcup_{i=1}^{m_2} C_{2i} \cup O_{2i}, \quad \dots, \quad Im_k = \bigcup_{i=1}^{m_k} C_{ki} \cup O_{ki}, \quad \text{для деяких } m_1, \dots, m_k.$$

Отже, нам необхідно порівняти контури і області еталонного

просегментованого зображення  $Im_e = \bigcup_{i=1}^{m_e} C_{ei} \cup O_{ei}$  і відповідно просегментованих зображень  $Im_1, Im_2, \dots, Im_k$ .

Загальна постановка задачі, таким чином, зводиться до порівняння двох сегментів  $Im_{e1} = C_{e1} \cup O_{e1}$  та  $Im_{11} = C_{11} \cup O_{11}$  зображень  $Im_e$  та  $Im_1$ .

Метрика Фреше. Для оцінки близькості контурів зображень використаємо метрику Фреше [19].

Для двох кривих  $f:[a,b] \rightarrow X$  і  $g:[a',b'] \rightarrow X$  відстань Фреше між ними рівна [11]:

$$d_F = \inf_{\alpha, \beta} \sup_{t \in [0,1]} d(f(\alpha(t)), g(\beta(t))),$$

де  $d(x,y)$  – евклідова відстань між точками  $x$  і  $y$ ,  $\alpha$  та  $\beta$  – довільні неперервні неспадні функції з проміжку  $[0,1]$  на проміжки  $[a,b]$  та  $[a',b']$  відповідно.

Значення функції  $\alpha(0) = 0$  і  $\alpha(1) = 1$ , і аналогічно для функції  $\beta$ .

Для обчислення дискретної відстані Фреше задані криві апроксимують полігональними кривими [34-39].

Полігональна крива – це крива  $C:[0,n] \rightarrow X$ , де  $n$  – ціле число і для кожного  $i \in [0,1, \dots, n-1]$  на інтервалі  $[i, i+1]$  функція  $C$  афінна,

$$C(i + \lambda) = (1 - \lambda)C(i) + \lambda C(i + 1).$$

Полігональна крива  $C:[0,n] \rightarrow X$  однозначно задається послідовністю  $\sigma(C) = (C(0), C(1), \dots, C(n))$  кінців лінійних сегментів.

Тоді для двох полігональних кривих  $C$  і  $R$ :

$$\sigma(C) = (v_1, \dots, v_m) \text{ і } \sigma(R) = (w_1, \dots, w_s).$$

Позначимо через  $L$  послідовність пар зв'язку між кривими  $C$  і  $R$ , яка рівна  $(v_{a_1}, w_{b_1})$ ,  $(v_{a_2}, w_{b_2})$ , ...,  $(v_{a_m}, w_{b_m})$ . Ці окремі пари є елементами декартового добутку  $\sigma(C) \times \sigma(R)$ , причому  $a_1 = 1$ ,  $b_1 = 1$ ,  $a_m = r$ ,  $b_m = s$  і для  $i = 1 \dots r$ ,  $a_{i+1} = a_i$  або  $a_{i+1} = a_i + 1$  і для  $i = 1 \dots s$ ,  $b_{i+1} = b_i$  або  $b_{i+1} = b_i + 1$ .

Довжина послідовності  $\|L\|$  визначається як найдовший зв'язок в  $L$ , тобто:  $\|L\| = \max_{i=1 \dots m} d(v_{a_i}, w_{b_i})$ . Складність даного алгоритму рівна  $O(mn)$ , де  $m$  і  $n$  – кількість лінійних сегментів двох полігональних кривих. Аналогічна складність отримана у роботі [27]. Деталізована постановка задачі приведена на рисунку 2.2.

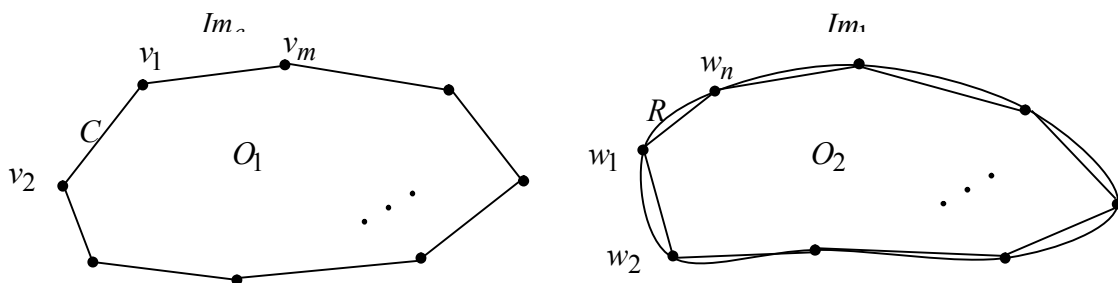


Рисунок 2.2 – Деталізована постановка задачі

Отже, дискретна відстань Фреше  $d_{dF}$  визначається формулою:

$$d_{dF}(C, R) = \min \{ \|L\| \}.$$

Алгоритм визначення дискретної відстані Фреше

Алгоритм визначення дискретної відстані Фреше розглянемо для випадку двох згаданих вище контурів.

1. Контур кожного сегмента  $C$  і  $R$  представимо у вигляді полігональних кривих:  $\sigma(C) = (v_1, \dots, v_r)$ ,  $\sigma(R) = (w_1, \dots, w_s)$ , де  $r$ ,  $s$  – кількість лінійно апроксимованих сегментів.

2. Формуємо послідовність  $L$  між кривими  $C$  і  $R$

$$L = (v_{a_1}, w_{b_1}), (v_{a_2}, w_{b_2}), \dots, (v_{a_m}, w_{b_m}), \quad a_1 = 1, \quad b_1 = 1, \quad a_m = r, \quad b_m = s.$$

3. Знаходимо евклідову норму послідовності  $\|L\| = \max_{i,j=1,\dots,m} d(v_{a_i}, w_{b_j})$ ,

використовуючи кроки:

3.1) якщо  $i=1$  та  $j=1$ , то дана відстань знаходиться, як евклідова відстань між точками:

$$d = \sqrt{(w_{b_j} - v_{a_i})^2};$$

3.2) якщо  $i>1$  та  $j=1$ , тоді відстань знаходиться за формулою:

$$\max\{d(v_{a_{i-1}}, w_{b_1}), d(v_{a_i}, w_{b_1})\};$$

3.3) якщо  $i=1$  та  $j>1$ , тоді відстань знаходиться так:

$$\max\{d(v_{a_1}, w_{b_{j-1}}), d(v_{a_1}, w_{b_j})\};$$

3.4) якщо  $i>1$  та  $j>1$ , тоді відстань знаходиться за формулою:

$$\max\{\min(d(v_{a_{i-1}}, w_{b_j}), d(v_{a_{i-1}}, w_{b_{j-1}}), d(v_{a_i}, w_{b_{j-1}})), d(v_{a_1}, w_{b_j})\}.$$

Отже,

1) ми розглянули геометричні об'єкти чотирьох типів: ланцюжки, цикли, ламані лінії і багатокутники. Це об'єкти різної складності. Ланцюжки і цикли – це кінцеві множини, а ламані лінії і багатокутники нескінченні. Для ланцюжків і ламаних ліній вказані точки, в яких може починатися їх обхід, а обхід циклів і багатокутників може починатися в будь-якій точці;

2) незважаючи на різну складність розглянутих об'єктів, розпізнавання їх



$\varepsilon$ -подібності має одну і ту ж складність порядку  $m \times n$ , де  $m$  і  $n$  – кількості точок в ланцюжках і циклах або кількості вершин у ламаних лініях і багатокутниках. Це той же порядок складності, що і при розпізнаванні  $\varepsilon$ -подібності в звичайній хаусдорфовій метриці, що не є посиленою;

3) різниця зазначених чотирьох типів об'єктів полягає в різній складності переходу від розпізнавання  $\varepsilon$ -подібності об'єктів до обчислення відстані між ними. Обчислення відстані між ланцюжками і розпізнавання їх  $\varepsilon$ -подібності не тільки мають одну і ту ж складність порядку  $m \times n$ , але виконуються дуже подібними алгоритмами 1 і 2. Обчислення відстані між циклами трохи складніше.

## 3 ПРОГРАМНИЙ МОДУЛЬ ОБЧИСЛЕННЯ ВІДСТАНИ МІЖ ЗОБРАЖЕННЯМИ

### 3.1 Системні вимоги та структура програмного модуля

Даний програмний модуль оцінює відстань між зображеннями.

Сформуємо основні вимоги до системної підтримки програмного модуля:

- робота із зображеннями форматів JPEG, PNG, та GIF;
- система повинна приймати на вхід два зображення та проводити їхню паралельну обробку;

- реалізація порівняння контурів зображень в метриці Фреше;
- результат роботи програми повинен відображатися у консоль;
- введення шляхів до зображень вводяться через консоль.

Вимоги до надійності:

- програма повинна працювати із великими за розміром зображеннями, без збоїв;

- запобігання помилкам при переповненні пам'яті;
- передбачити виведення помилок при їхньому виникненні.

Вимоги до робочої станції:

- система повинна працювати на IBM-сумісних робочих станціях;
- мінімальні вимоги до робочих станцій: процесор із частотою від 1.8 ГГц із кількістю ядер більше 1, клавіатура, монітор;

- оперативна пам'ять від 1 Гб;
- наявність дискретної відео карти nVidia з підтримкою технології

CUDA.

Вимоги до програмного забезпечення:

- операційна система – система Unix, Windows;
- наявність Java;
- розмір програми до 10 Мб;

- код програмних модулів повинен містити необхідні для його розуміння коментарі.

Для розробки програмного модулю алгоритму порівняння зображень в метриці Фреше для незважених дерев обрано шаблон проектування MVC. Модель–вигляд–контролер (MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності. Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Перелік програмного забезпечення для реалізацію алгоритму порівняння зображень в метриці Фреше для незважених дерев наведено у таблиці 3.1.

Таблиця 3.1 – Перелік програмного забезпечення для реалізацію алгоритму порівняння зображень в метриці Фреше для незважених дерев.

№	Назва
1	Java 1.8
2	OpenCV 3.0
3	Intellig IDEA
4	Windows, linux

Структуру шаблону MVC наведено на рисунку 3.1.

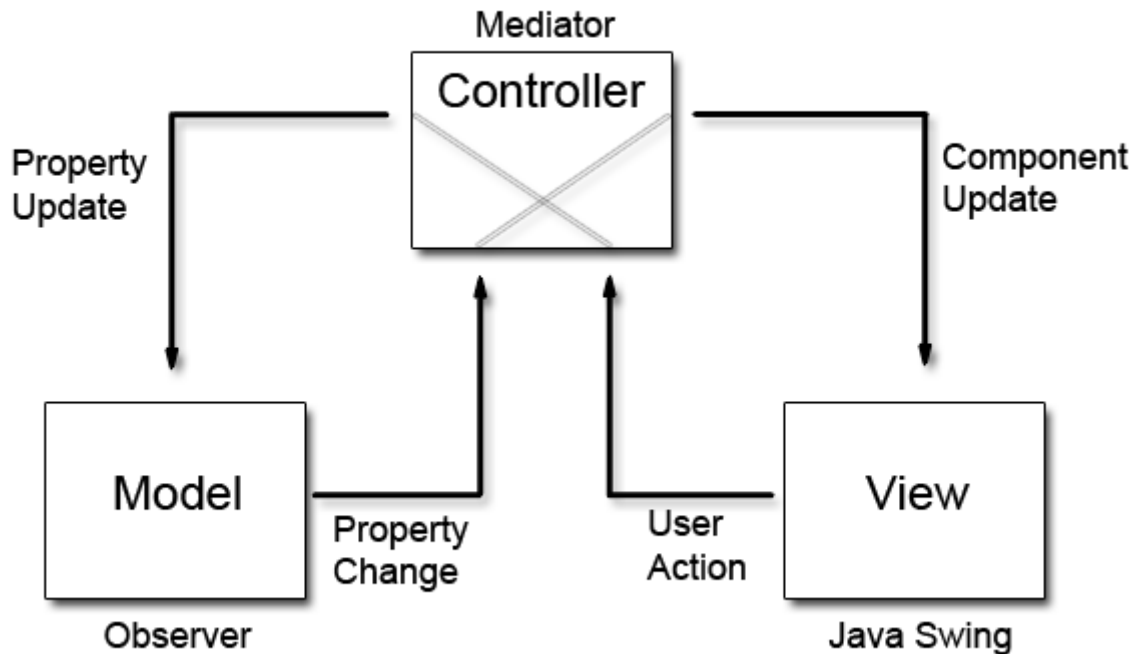


Рисунок 3.1 – Структура шаблону MVC

Графічний інтерфейс програми розроблено з використанням технології JavaFX. Приклад графічного інтерфейсу вікна Scene Builder наведено на рисунку 3.2.

JavaFX Scene Builder - це візуальний інструмент дизайну дозволяє користувачеві швидко зробити дизайн інтерфейсу для користувача додатки JavaFX, і не потрібно писати код. Ви можете перетягувати компоненти інтерфейсу в область роботи, змінювати їх атрибути, застосовувати стилі, і FXML для Layout автоматично створені нижче. Отриманий результат це набір FXML який можна комбінувати з проектом Java, поєднуючи інтерфейс з логікою програми.

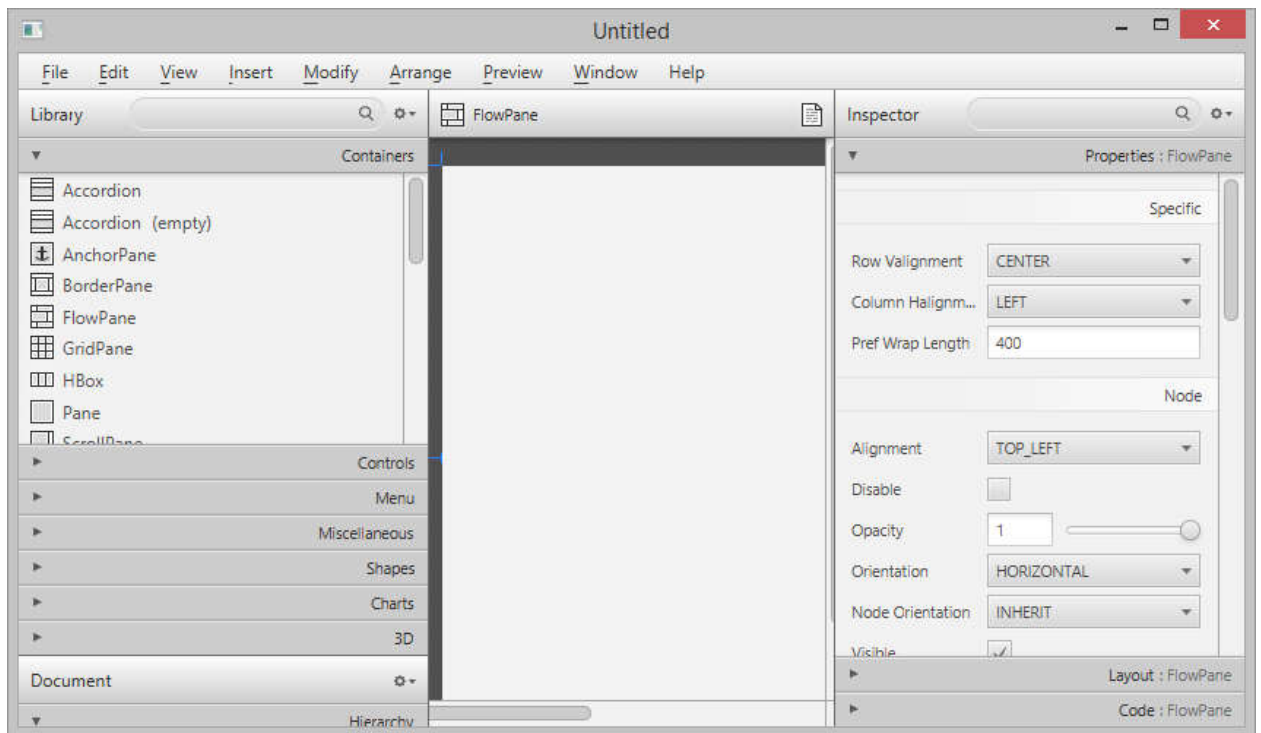


Рисунок 3.2 – Приклад графічного інтерфейсу вікна Scene Builder

JavaFX Scene Builder може бути інтегрований в IDE, наприклад Eclipse і Netbean, в даній статті я покажу вам як встановити JavaFX Scene Builder, і інтегрувати його Eclipse.

Щоб вбудувати Scene Builder в Eclipse, для початку потрібно встановити е (fx) clipse, це набір необхідних інструментів і бібліотек для програмування JavaFX, вам потрібно встановити як Plugin для eclipse.

Клас «Isometric transformation» вміщує такі методи:

- distBy350Rotation;
- distBySecantMethod;
- distByLongestChord;
- distByLongestChordAndPerpendiculare;
- distByListOfChords.

Діаграму класів для пакету «qualityEstimator» наведено на рисунку 3.3. «Package Chorde» вміщує класи для реалізації функціоналу пошуку хорд. «Package Contour» вміщує класи та методи для формування контуру полігонів, визначення вершин тощо.

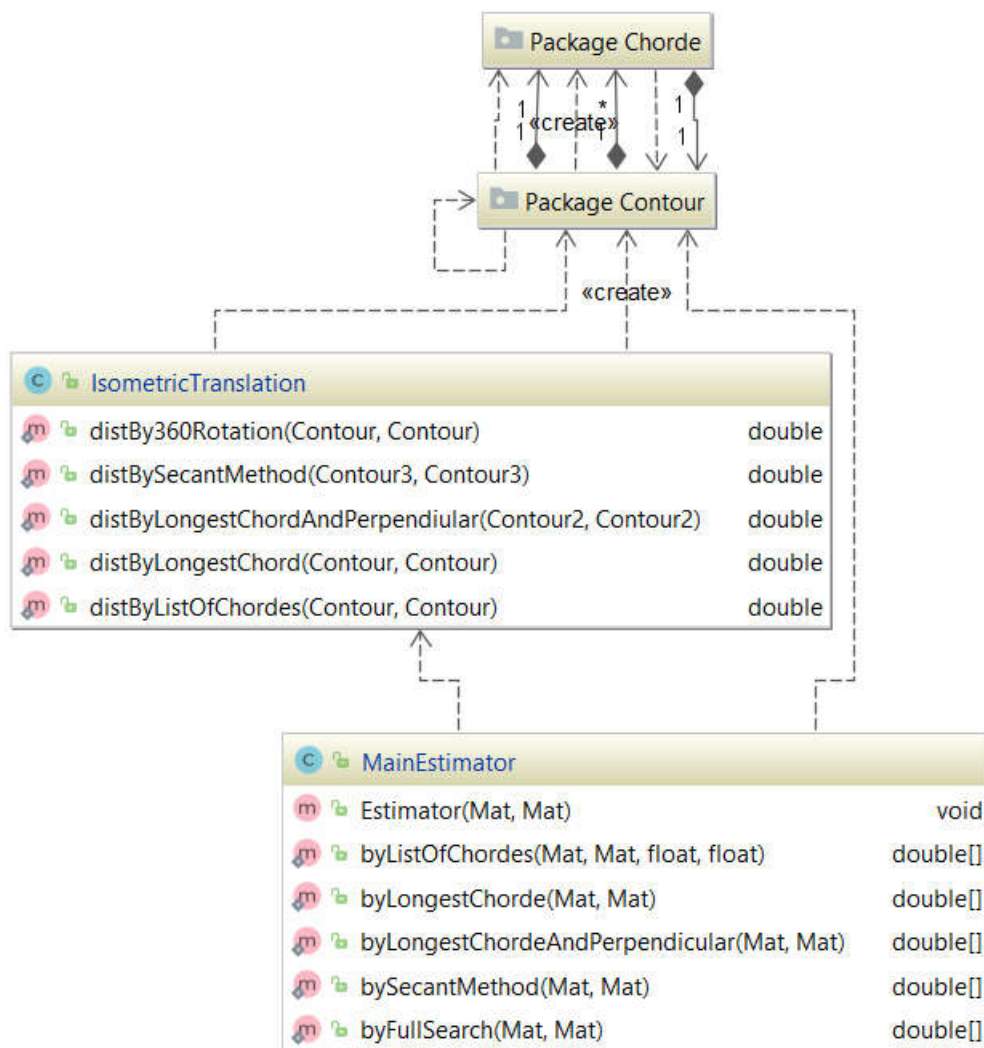


Рисунок 3.3 – Діаграма класів для пакету «qualityEstimator»

Класи – це конструкції спеціального виду, які дозволяють об’єднати ряд змінних різних типів в одне ціле. Крім власне даних, класи зазвичай включають підпрограми (в термінології java - методи) і можуть включати блоки (сукупність інструкцій між фігурними дужками { }) та інші класи (внутрішні класи). Таким чином утворюються нові типи даних.

Роз’яснення принципів побудови і роботи з класами доволі громіздка тема, тому тут розглянемо лише спрощені основи їх роботи і створення. Детальніше дивіться розділ присвячений класам. Розглянути зараз необхідно через те, що Java повністю об’єктно-орієнтована мова. З класами та пов’язаними з ними методами ви зіштовхуватиметеся на кожному кроці.

Зокрема, навіть виведення даних `System.out.println()`; - здійснюється за допомогою класу `System` та методу `println()`.

Методи в `java` – це аналог підпрограм, функцій, процедур в інших мовах програмування. За допомогою методів ми виносимо текст повторюваного коду програми окремо в тіло методу, після чого можна викликати даний метод з будь-якого місця програми, безліч разів.

Конструктор - це спеціальний метод, який викликається при створенні нового об'єкта. Не завжди зручно форматовувати всі змінні класу при створенні його примірника. Іноді простіше, щоб якісь значення були б створені за замовчуванням при створенні об'єкта. По суті конструктор потрібен для автоматичної ініціалізації змінних.

Клас «`MainEstimator`» включає такі методи:

- `Estimator`;
- `byListOfChordes`;
- `byLongestChorde`;
- `byLongestChordeAndPerpendicular`;
- `bySecantMethod`;
- `byFullSearch`.

Лістинг коду класу «`Skeleton`»:

```
public class Skeleton {
    private int pointNum_1;
    private int pointNum_2;
    public double maxDistance;
    public Skeleton(double[][] branch1, double[][] branch2){
        pointNum_1 = branch1.length;
        pointNum_2 = branch2.length;
        Compare compare = new Compare();
    }
}
```

Лістинг коду виклику методу для поірівняння гілок скелетону у форматі «1 => 1»:

```
/** Порівняння 1 гілки до 1 гілки
 * якщо к-сть точок дорівнює 2*/
if( pointNum_1 == 2 && pointNum_1 == pointNum_2)
```

```

        {
            maxDistance = compare.OneToOne(branch1, branch2, 1);
            //System.out.println("Відстань 1-1 : " +
Branch.getBranchName() + " " + maxDistance);
        }

```

Лістинг коду виклику методу для поірвняння гілок скелетону у форматі «1 => N»:

```

/** Порівняння 1 гілки до N гілки
 * якщо к-сть точок дорівнює 2 та к-сть точок 1 гілки
менше 2 */
else if(pointNum_1 == 2 && pointNum_1 < pointNum_2)
{
    maxDistance = compare.OneToMany(branch1, branch2,
pointNum_2);
    //System.out.println("Відстань 1-N : " +
Branch.getBranchName() + " " + maxDistance);
}

```

Лістинг коду виклику методу для поірвняння гілок скелетону у форматі «N => 1»:

```

/** Порівняння N гілок до 1 гілки
 * якщо к-сть точок 1 більше ніж у 2 та к-сть точок другої
гілки = 2 */
else if(pointNum_1 > pointNum_2 && pointNum_2 == 2)
{
    maxDistance = compare.ManyToOne(branch1, branch2,
pointNum_1);
    //System.out.println("Відстань N-1: " +
Branch.getBranchName() + " " + maxDistance);
}

```

Лістинг коду виклику методу для поірвняння гілок скелетону у форматі «N => N»:

```

/** Порівняння N гілок до N гілок */
else if(pointNum_1 > 2 && pointNum_2 > 2)
{
    maxDistance = compare.ManyToMany(branch1, branch2,
pointNum_1, pointNum_2);
    //System.out.println("Відстань N-N: " +
Branch.getBranchName() + " " + maxDistance);
}

```



```
        else {System.out.println("Значення не задовільняють жодну  
умову");maxDistance = 0;}  
    }
```








### 3.2 Вибірка полігональних об'єктів

Для тестування роботи алгоритму було штучно згенеровано полігональні об'єкти, зображені у таблиці 3.2. Вибірка — це множина об'єктів, подій, зразків або сукупність вимірів, за допомогою визначеної процедури вибраних з генеральної сукупності для участі в дослідженні. Зазвичай, обсяг генеральної сукупності дуже великий, що робить прийняття до уваги всіх членів непрактичним або неможливим. Вибірка являє собою множину або сукупність певного обсягу, члени якої збираються і статистичні характеристики обчислюються таким чином, що в результаті можна зробити висновки або екстраполяцію із вибірки на всю генеральну сукупність. У математичних термінах, для даної статистичної змінної  $X$  з вибірковим розподілом  $F$ , вибірка обсягу  $n = 4,5\dots$  являє собою сукупність  $n$  незалежних вимірів, показникам якої приписуються статистичні ймовірності.

Вибірка представляє  $n$  експериментів, в яких можна виміряти ту ж змінну. Наприклад, якщо  $X$  представляє висоту індивідуума і ми вимірюємо  $n$  індивідуумів, буде висотою  $i$ -того індивідуума. Відзначте, що вибірка випадкових змінних (тобто набір вимірних функцій) не повинна бути змішаною з реалізаціями цих змінних (які є значеннями, що приймають ці випадкові змінні).

Характеристикою положення для вибіркового розподілу є вибіркове середнє, а характеристиками розсіювання вибіркового розподілу — вибіркові дисперсії. Числовою характеристикою вибірки є також розмах її значень.

Таблиця 3.2 – Полігональні об'єкти

№	Приклад
1	
2	
3	
4	
5	
6	
7	

Приклад формування скелетону із полігональних зображень наведено на рисунку 3.4.



Рисунок 3.4 – Приклад формування скелетону із полігональних зображень

Результат формування скелетонів із полігональних зображень наведено у таблиці 3.3.

Таблиця 3.3 - Результат формування скелетонів із полігональних зображень

№	Полігон	Скелетон
1		
2		
3		
4		

Медична візуалізація — це методика і процес створення візуальних зображень внутрішніх органів з метою проведення клінічного аналізу і медичного втручання. Медична візуалізація використовується для огляду внутрішніх структур тіла людини, а також для діагностики і лікування хвороб. Крім того, за допомогою цієї методики створюють базу даних нормальної

анатомії і фізіології, яка дозволить виявляти аномалії. У медицині також виконують візуалізацію видалених органів і тканин, однак такі процедури зазвичай відносять до патології, а не до медичної візуалізації. Як дисципліна в найширшому сенсі, медична візуалізація є розділом біологічної візуалізації, й до неї належать радіологія, у якій зображення формується за допомогою рентгенівських променів (радіографія), МРТ, медична акустика, чи ультразвукова діагностика, ендоскопія, еластографія, тактильна візуалізація, термографія, медична фотографія й такі методики функціональної візуалізації ядерної медицини, як позитрон-емісійна томографія.

Приклади сегментації цитологічних зображень наведено на рисунку 3.5.

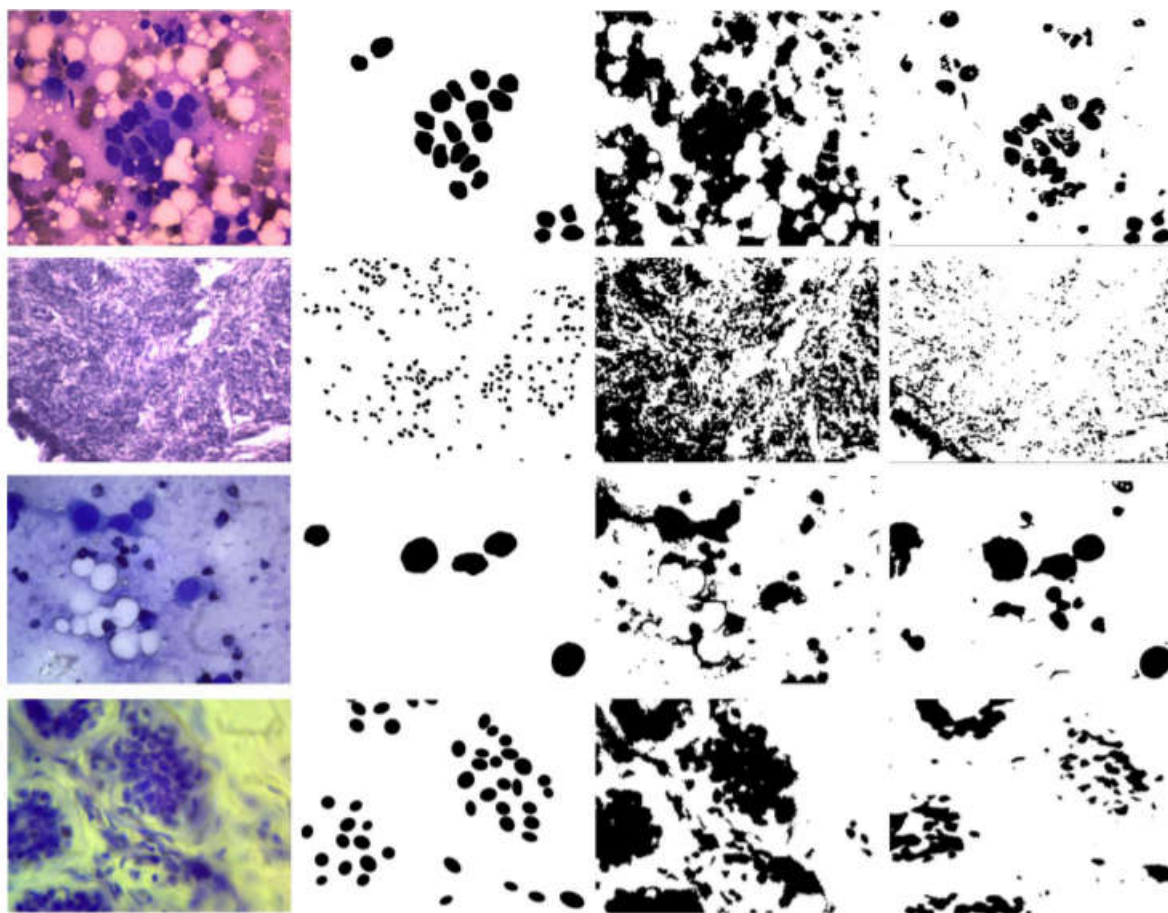


Рисунок 3.5 – Приклади сегментації цитологічних зображень















У комп'ютерному зорі, сегментація – це процес поділу цифрового зображення на декілька сегментів (множина пікселів, названих суперпікселями).

Мета сегментації полягає в спрощенні і / або зміні представлення зображення, щоб його було простіше і легше аналізувати. Сегментація зображень зазвичай використовується для того, щоб виділити об'єкти і кордони на зображеннях. Більш точно, сегментація зображень - це процес присвоєння таких міток кожному пікселю зображення, що пікселі з однаковими мітками мають загальні візуальні характеристики.

### 3.3 Порівняльний аналіз результатів

Полігональні зображення для тестування алгоритмів згенеровані випадковим чином, щоб врахувати усі можливі варіанти розташування їх вершин. Порівняльний аналіз алгоритмів знаходження мінімальної відстані між полігонами наведено у таблиці 3.4.

Таблиця 3.4 – Порівняльний аналіз алгоритмів знаходження мінімальної відстані між полігонами

Еталонний полігон							
Згенерований полігон							
Відстань Фреше	37.48	43.01	38.35	46.6	36.9	48.85	65.28

Час роботи алгоритму Фреше та алгоритму Хаусдорфа наведено на рисунку 3.6.

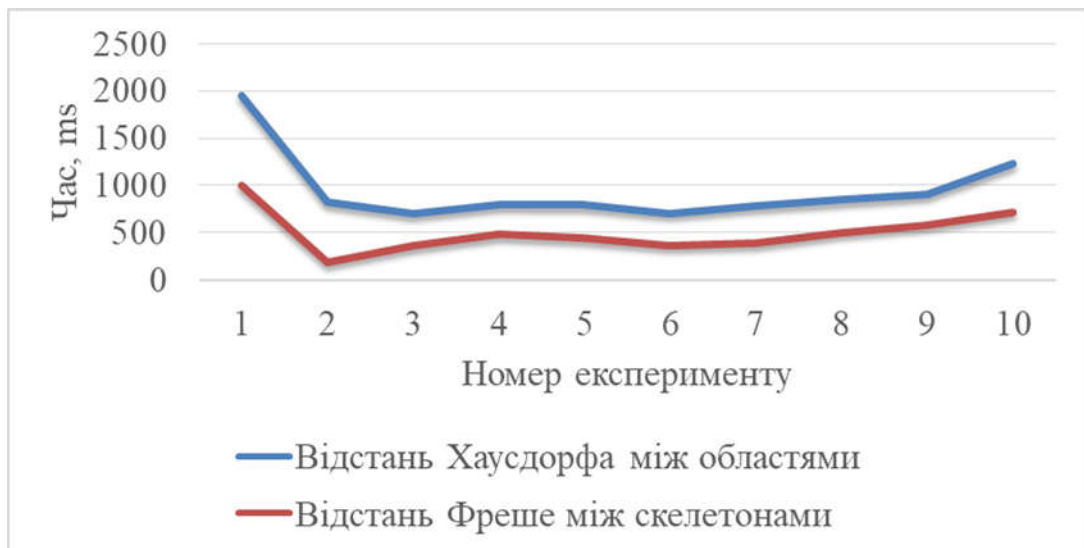


Рисунок 3.6 – Порівняння роботи алгоритмів Хаусдорфа та Фреше

З результатів, наведених на рисунку 3.3, можна зробити висновок, що час обчислення відстані Фреше між скелетонами менший ніж час обчислення відстані Хаусдорфа між областями.

Графічний інтерфейс головного вікна програми для пошуку відстані між мікрооб'єктами на основі метрики Фреше між деревами наведено на рисунку 3.7.

Вікно складається з таких елементів:

- шапка. У даному блоці знаходиться меню програми, зокрема в даному випадку посилання на головне вікно;
- блок завантаження зображень. У даному блоці знаходиться кнопка, яка дозволяє вибрати зображення у форматах «png, jpg, bmp» з комп'ютера користувача;
- блок відображення зображень. У даному блоці відображаються обрані два зображення між якими буде проводитись порівняння. Візуально зображення підігнані до певного розміру для зручності їх відображення, однак порівняння відбувається між оригінальними зображенням;
- блок обчислення відстані Фреше та виводу результатів. Після натиснення кнопки «ОК» відбувається обчислення відстані Фреше між обраними зображеннями.

Програму розроблено з використанням мови програмування Java та бібліотеки opencv 3.0.

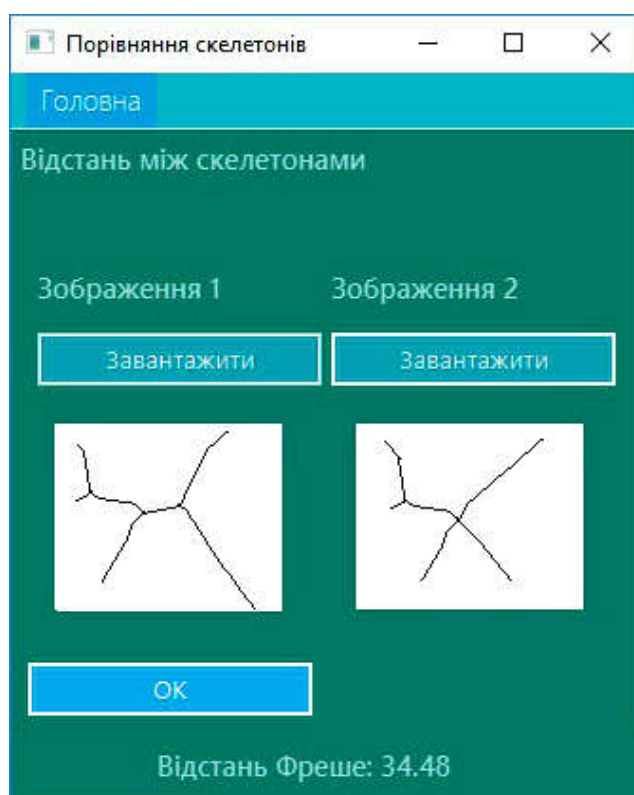


Рисунок 3.7 – Графічний інтерфейс головного вікна програми для пошуку відстані між мікроб'єктами на основі метрики Фреше між деревами

Графічний інтерфейс користувача — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

Виконання дій в ГІК — це безпосередня маніпуляція з графічними елементами. Окрім комп'ютерів, GUI використовується в мобільних пристроях, таких, як мобільні телефони, планшети, електронні книги, портативні медіаплеєри тощо. Термін ГІК зазвичай не вживають стосовно інтерфейсів з низькою роздільною здатністю. Наприклад, в відеоіграх використовують інтерфейс HUD.

JavaFX — платформа та набір інструментів для створення насичених інтернет-застосунків з можливістю підвантаження медіа та змісту. Вперше продемонстровано Sun Microsystems на Міжнародній конференції Java-розробників JavaOne у травні 2007. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та надавати розвинуті інтернет-застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

На момент випуску JavaFX 1.0 доступна тільки для ОС Windows і Mac OS, але розробники обіцяють незабаром забезпечити підтримку й інших платформ. Sun дещо спізнюється на ринку потокового медіа: якщо у 1995 році Java була одноосібним лідером і законодавцем, то сьогодні її серйозно підтискають конкуренти, такі як Microsoft Silverlight, Adobe Flash і AIR. Щоб підвищити інтерес до своєї платформи, Sun також планує розкрити сирцеві коди JavaFX.

Головним нововведенням JavaFX 2.0, що вийшов вже під орудою Oracle у жовтні 2011, стала відмова від використання спеціальної скриптової мови JavaFX Script на користь стандартного Java API, що дозволяє створювати JavaFX-застосунки мовами, що базуються на JVM — JRuby, Clojure, Scala і Groovy. Для розробки застосунків доступний багатий графічний і мультимедійний API, що спрощує створення візуальних програм.

Мінімальні вимоги до середовища виконання модуля оцінки сегментації:

- ОС: Windows (XP, 7,8,10), Linux-подібні системи з підтримкою JAVA (Ubuntu 16.04), Mac OS X 10.7.3.
- RAM 512 MB.
- Вільний простір на HDD: 50 MB.
- Процесор 800 MHz Intel Pentium III або 1000 MHz AMD Athlon.



## ВИСНОВКИ

1. Проаналізовано основні алгоритми порівняння зображень і проведено комп'ютерні експерименти сегментації цитологічних . Проаналізовані метрики роботи із зображеннями, що дало можливість вибрати метрику Фреше для оцінки відстані між контурами зображень.

2. Розглянуті геометричні об'єкти чотирьох типів: ланцюжки, цикли, ламані лінії і багатокутники. Це об'єкти різної складності. Ланцюжки і цикли – це кінцеві множини, а ламані лінії і багатокутники нескінченні. Для ланцюжків і ламаних ліній вказані точки, в яких може починатися їх обхід, а обхід циклів і багатокутників може починатися в будь-якій точці.

3. Незважаючи на різну складність розглянутих об'єктів, розпізнавання їх  $\varepsilon$ -подібності має одну і ту ж складність порядку  $m \times n$ , де  $m$  і  $n$  – кількості точок в ланцюжках і циклах або кількості вершин у ламаних лініях і багатокутниках.

4. Різниця зазначених чотирьох типів об'єктів полягає в різній складності переходу від розпізнавання  $\varepsilon$ -подібності об'єктів до обчислення відстані між ними. Обчислення відстані між ланцюжками і розпізнавання їх  $\varepsilon$ -подібності не тільки мають одну і ту ж складність порядку  $m \times n$ , але виконуються дуже подібними Обчислення відстані між циклами трохи складніше.

5. Розроблено алгоритм порівняння контурів зображень в метриці - Фреше на основі алгоритму визначення дискретної відстані Фреше і, що дало можливість оцінити відстань між двома контурами.

6. Сформовано системні вимоги до програмного модуля, розроблено його архітектуру, що дало можливість спроектувати та програмно реалізувати модуль порівняння контурів зображень в метриці Фреше.

7. Проведено тестування програмного модуля та комп'ютерні експерименти на прикладі цитологічних зображень, що дало можливість оцінювати відстань між контурами зображень.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ76

1. Національний Канцер Реєстр: веб-сайт. URL: <http://unci.org.ua/spetsialistam/nacionalnij-kancer-reyestr/>.
2. Дериш Б. Б., Вихиль Ю.О., Юглічек О. В., Герасимчук С. Ю. Алгоритми порівняння зображень на основі контурного аналізу: II Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 11-12.
3. Дериш Б. Б., Вихиль Ю.О., Юглічек О. В. Порівняння зображень на основі колірних і текстурних ознак: II Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 12-13.
4. Методичні рекомендації до виконання випускної кваліфікаційної роботи з освітнього ступеня «Магістр» спеціальності 123 «Комп'ютерна інженерія» / О.М. Березький, Л.О. Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 47 с.
5. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.
6. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс – М.: Техносфера. – 2005.–1072 с.
7. Шапиро Л. Компьютерное зрение: Пер. с англ. / Л. Шапиро, Дж. Стокман – М.:Бином. Лаборатория знаний. – 2006. – 752 с.
8. Вежневек А.П. Методы классификации с обучением по прецедентам в задаче распознавания объектов на изображениях / Вежневек А.П. – Новосибирск, Труды конференции «Graphicon-2006» – 2006. – С.166–173.

9. Кольцов П. П. Использование метрик при сравнительном исследовании качества работы алгоритмов сегментации изображений / П. П. Кольцов // Информатика и её применение. – 2011. – Т. 5, Вып. 3. – С. 53–63.
10. Левашкина А. О. Исследование супервизорных критериев оценки качества сегментации изображений / А. О. Левашкина, С. В. Поршнева // Известия Томского политехнического университета. – 2008. – Т. 313, №5. – С. 28–33.
11. Helmut Alt and Michael Godau. Computing the Frèchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.
12. Laurence Boxer. On Hausdorff-like metrics for fuzzy sets. *Pattern Recognition Letters*, 18(2):115–118, 1997.
13. Jingying Chen, Maylor K. H. Leung, and Yongsheng Gao. Noisy logo recognition using line segment Hausdorff distance. *Pattern Recognition*, 36(4):943–955, 2003.
14. H.J.A.M. Heijmans and J.B.T.M. Roerdink. *Mathematical Morphology and Its Applications to Image and Signal Processing*. Computational Imaging and Vision. Springer, 1998.
15. R.T. Rockafellar, R.J.B. Wets, and M. Wets. *Variational Analysis*. Grundlehren Der Mathematischen Wissenschaften. Springer, 2011.
16. William Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
17. Pierre Soille, Martino Pesaresi, and Georgios K. Ouzounis, editors. *Mathematical Morphology and Its Applications to Image and Signal Processing - 10th International Symposium, ISMM 2011, Verbania-Intra, Italy, July 6-8, 2011*. Proceedings, volume 6671 of Lecture Notes in Computer Science. Springer, 2011.
18. Файнзильберг Л. С. Восстановление эталона циклических сигналов на основе использования хаусдорфовой метрики в фазовом пространстве. *Кибернетика и системный анализ*, (3):20–28, 2003.
19. Деза Елена И. *Энциклопедический словарь расстояний: пер. с англ.* / Елена И. Деза, Мишель Мари Деза. – М. : Наука, 2008. – 444 с.

20. Bloch I. On fuzzy distances and their use in image processing under unprecision, *Pattern Recognition*, Vol. 32 pp. 1873-1895, 1999.

21. Hafner J., Sawhney H.S., Equitz W., Flickner M. and Niblack W. Efficient Color Histogram Indexing for Quadratic Form Distance Functions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, Nr. 7, pp. 729-736, 1995.

22. Kosheleva O., Kreinovich V. and Nguyen H.T. On the Optimal Choice of Quality Metric in Image Compression, *Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, 7-9 April 2002, Santa Fe. IEEE Comp. Soc. Digital Library, Electronic Edition, pp. 116-120, 2002.

23. Rubner Y., Tomasi C. and Guibas L.J. The Earth Mover's Distance as a Metric for Image Retrieval, *Int. J. of Comp. Vision*, Vol. 40, Nr. 2, pp. 99-121, 2000.

24. Choi S.W. and Seidel H.-P. Hyperbolic Hausdorff Distance for Medial Axis Transform, *Research Report MPI-I-2000-4-003 of Max-Planck-Institute fur Informatik*, 2000.

25. Березький О. М. База даних цитологічних та гістологічних зображень ауто- та ксеногенних тканин / О. М. Березький, Г.М. Мельник, Т.В. Дацко, С. О. Вербовий // *Науковий вісник НЛТУ України: зб. наук.-техн. праць*. – Львів: РВВ НЛТУ України. – 2014. – Вип. 24.10. – С. 338-345.

26. Методи, алгоритми і програмні засоби опрацювання біомедичних зображень / Березький О. М., Батько Ю.М., Березька К.М., Вербовий С.О., Дацко Т.В., Дубчак Л.О., Ігнатєв І.В., Мельник Г.М., Николюк В.Д., Піцун О.Й. Тернопіль: Економічна думка, ТНЕУ, 2017. 330 с.

27. Шлезингер М. И., Водолазский Е.В., Яковенко В.М. Распознавание сходства многоугольников в усиленной хаусдорфовой метрике // *Кибернетика и системный анализ*. – 2014. - №3. – С.174-187.

28. Водолазский Е.В., Крюков В.И. Распознавание бесконфликтности двух кусочно-линейных траекторий // *Управляющие системы и машины*. – 2015. - №3. – С.42-46.

- 29.Березький О. Методи кількісної оцінки якості сегментації зображень / Олег Березький // Матеріали дванадцятої всеукраїнської міжнародної конференції «Оброблення сигналів і зображень та розпізнавання образів» (УкрОБРАЗ'2014), Київ, 3-7 листопада 2014 р. – К., 2014. – С. 51–54.
- 30.Thomas Eiter and Heikki Mannila. Computing discrete Frechet distance. Technical Report CD-TR 94/64, Information Systems Department, Technical University of Vienna, 1994. - P. 7.
- 31.Gromov, Metric Structures for Riemannian and Non-Riemannian Spaces, Progress in Mathematics 152, Birkhäuser Boston, Inc., Boston, MA, 1999.
- 32.Atallah M.J. A Linear time Algorithm for the Hausdorff Distance between Convex Polygons, Inform. Process. Lett., 1983, 17, - P. 2017-209.
- 33.Atallah M.J., Ribeiro C.C., Lifschits S. A Linear time Algorithm for the Computation of some Distance between Convex Polygons. Operations Research. Vol. 25, №4, 1991, - P. 413-424.
- 34.Thomas Eiter and Heikki Mannila. Computing discrete Frechet distance. Technical Report CD-TR 94/64, Information Systems Department, Technical University of Vienna, 1994. - P. 7.
- 35.Berezsky, O. Fréchet distance between weighted rooted trees / O. Berezsky, M. Zarichnyi // Matematychni Studii. – 2017. – Vol. 48, No.2. – P. 165-170.
- 36.Berezsky O. Gromov-Fréchet distance between curves / O. Berezsky, M. Zarichnyi // Matematychni Studii. – 2018. – Vol. 50, No.1.
- 37.Berezsky O. Frechet Metric for Trees / Oleh Berezsky // Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, August 23-27, 2016. – Lviv, 2016. – P. 213-217.
- 38.Березький О.М. Адаптивний метод сегментації зображень на основі метрик / О.М. Березький, О.Й. Піцун // Науковий вісник НЛТУ України: збірник науково-технічних праць. Львів: РВВ НЛТУ України. – 2018. – №. 28(3). – С.122-126.
- 39.Berezsky O. Image Segmentation Metric-Based Adaptive Method / Oleh Berezsky, Oleh Pitsun, Natalia Batryn, Kateryna Berezska, Nadiya Savka, Taras

Dolyniuk // Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, August 21-25, 2018. – Lviv, 2018. – P. 554-557.

40.JMicroVision: веб-сайт. URL: <http://www.jmicrovision.com>.

41.NEXSYS ImageExpert Pro 3: веб-сайт. URL: <http://www.nexsys.ru/iepro3x.htm>.

42.ImageJ Features: веб-сайт. URL: <http://rsb.info.nih.gov/ij/features.html>.

43.ImageJ: веб-сайт. URL: <https://ru.wikipedia.org/wiki/ImageJ>.

44.Javascript implementation of the Ramer Douglas Peucker Algorithm: веб-сайт. URL: [http://karthaus.nl/rdp/Javascript implementation of the Ramer Douglas Peucker Algorithm](http://karthaus.nl/rdp/Javascript%20implementation%20of%20the%20Ramer%20Douglas%20Peucker%20Algorithm).

45.Цифрове зображення: веб-сайт. URL: [http://uk.wikipedia.org/wiki/Цифрове зображення](http://uk.wikipedia.org/wiki/Цифрове_зображення).

46.Biomedical Imaging: веб-сайт. URL: <http://seas.yale.edu/faculty-research/research-areas/biomedical-imaging>

47.Гістологія людини / О. Д. Луцик, А. Й. Іванова, К. С. Кабак, Ю. Б. Чайковський / За ред. О. Л. Босецького. – К.: «Книга плюс», 2003. – 592 с.

48.Гістологія, цитологія і ембріологія: веб-сайт. URL: [http://www.morphology.dp.ua/\\_mp3/intro.php](http://www.morphology.dp.ua/_mp3/intro.php).

49.Методи гістологічних досліджень: веб-сайт. URL: [http://intranet.tdmu.edu.ua/data/kafedra/internal/histolog/classes\\_stud/uk/stomat/ptn/1/01 гістологічна техніка. методи гістологічних досліджень. основи цитології.загальна організація клітини. поверхневий комплекс.htm](http://intranet.tdmu.edu.ua/data/kafedra/internal/histolog/classes_stud/uk/stomat/ptn/1/01_gistologichna_tekhnika_metodi_gistologichnih_doslidzenj_osnovi_cytologii_zagalna_organizacija_klitini_poverkhniviy_kompleks.htm).

50.Новак В. П. Цитологія, гістологія, ембріологія. /В. П. Новак, А. П. Мельниченко. Біла Церква, 2005. – 256 с.

51.Автандилов Г.Г. Основы количественной паталлогической анатомии / Г.Г. Автандилов. – М. : Изд-во "Медицина", 2002. – 238 с.