

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

Манзик Сергій Віталійович

Алгоритми скелетонізації зображень в метричних просторах / Skeletonization images algorithms in metric spaces

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

Випускна кваліфікаційна робота

Виконав студент групи КІзм-21
Сергій Віталійович Манзик

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

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Випускна кваліфікаційна робота на тему «Алгоритми скелетонізації зображень в метричних просторах» із спеціальності 123 «Комп'ютерна інженерія» написана обсягом 102 сторінки і містить 26 ілюстрацій, 6 таблиць, 3 додатки та 46 джерел за переліком посилань.

Метою роботи є розроблення алгоритмів скелетонізації зображень для систем автоматизованої мікроскопії.

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

Результати дослідження: алгоритми скелетонізації зображень, програмний модуль скелетонізації 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, СКЕЛЕТОНІЗАЦІЯ, БІОМЕДИЧНІ ЗОБРАЖЕННЯ, МЕТРИЧНІ ПРОСТОРИ, СИСТЕМИ АВТОМАТИЗОВАНОЇ МІКРОСКОПІЇ.

RESUME

The qualification graduation thesis on "Skeletonization algorithms for metric spaces" from specialty 123 "Computer Engineering" is written on a volume of 102 pages and contains 26 illustrations, 6 tables, 3 appendices and 46 references in the list of links.

The purpose of the work is to develop algorithms for skeletonization of images for automated microscopy systems.

Research Methods: Digital Imaging Techniques, Computer Modeling Techniques.

Results of the study: image skeletonization algorithms, software module of image skeletonization.

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.

KEY WORDS: IMAGES, SKELETONIZATION, BIOMEDICAL IMAGES, METRIC SPACES, SYSTEMS OF AUTOMATED MICROSCOPY.

ЗМІСТ

Вступ.....	7
1 Аналіз алгоритмів аналізу зображень	9
1.1 Поняття бінарного зображення.....	9
1.2 Аналіз границь зображень.....	13
1.3 Аналіз систем автоматизованої мікроскопії.....	18
1.4 Аналіз біомедичних зображень.....	25
1.5 Аналіз завдання на випускню кваліфікаційну роботу та постановка задач	28
2 Алгоритми скелетонізації зображень	30
2.1 Поняття метричного простору та метрик.	30
2.2 Поняття скелетону зображення	33
2.3 Алгоритми скелетонізації полігонів.....	38
3 Програмна система скелетизації об'єктів на цифрових зображеннях ...	53
3.1 Структура програмної системи скелетизації об'єктів.....	53
3.2 Програмні модулі системи скелетизації об'єктів на цифрових зображеннях	61
3.3 Тестування та аналіз реалізованої системи	69
Висновки.....	74
Список використаних джерел.....	75
Додаток А Лістинг коду програми.....	80
Додаток Б Світлокопії виданих публікацій	94
Додаток В Довідка про використання результатів випускної кваліфікаційної роботи.....	102

ВСТУП

Актуальність теми. Рак молочної залози в Україні, так само, як і в більшості країн світу, є одним з найпоширеніших пухлинних процесів у жінок і посідає перше місце в структурі захворюваності та смертності від злоякісних новоутворень (25% від всіх випадків раку). Кількість жінок, які хворіють на рак молочної залози в Україні в абсолютних цифрах зростає. Кожна жінка, яка захворіла на рак молочної залози, у середньому втрачає 17 - 18 років життя, що становить 53% від всіх втрат жіночої популяції в Україні. Тому проблема вдосконалення інформаційних технологій, що використовуються при діагностуванні злоякісних новоутворень залишається однією з найбільш актуальних [1].

Об'єкт дослідження – системи автоматизованої мікроскопії.

Предмет дослідження – алгоритми скелетонізації зображень.

Метою роботи є розроблення алгоритмів скелетонізації зображень для систем автоматизованої мікроскопії.

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

Наукова новизна одержаних результатів.

Розроблено алгоритми скелетонізації зображень в різних метричних просторах.

Практичне значення отриманих результатів. Розроблено програмний модуль скелетонізації зображень.

Публікації та апробація магістерської роботи [2, 3].

1. Вакуліч О.В. Опрацювання біомедичних зображень на основі алгоритмів комп'ютерного зору / О.В. Вакуліч, В.І. Димінський, І.М. Макогон, С. В. Манзик // Матеріали III всеукраїнської науково-практичної конференції «Прикладна геометрія та інформаційні технології в моделюванні

об'єктів, явищ і процесів» (AGIT-2018), м. Миколаїв, 17–19 жовтня 2018 р. – Миколаїв: МНУ імені В.О. Сухомлинського, 2018. – С. 174-175.

2. Дериш Б. Б., Манзик С. В., Попов В.О., Кравець В. Б. Аналіз хеш – алгоритмів порівняння зображень: II Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 10.

Випускна кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків [4, 5].

У першому розділі проведено аналіз алгоритмів аналізу зображень, систем автоматизованої мікроскопії, біомедичних зображень, здійснено аналіз завдання на магістерську роботу та зроблено постановку задач.

У другому розділі розглянуті метричні простори, алгоритми скелетонізації зображень вметричних просторах.

У третьому розділі розроблена структура програмної системи скелетизації об'єктів, програмні модулі системи скелетизації об'єктів на цифрових зображеннях і проведено тестування та аналіз реалізованої системи.

У додатках представлена довідка про впровадження, світлокопії публікацій.

1 АНАЛІЗ АЛГОРИТМІВ АНАЛІЗУ ЗОБРАЖЕНЬ

1.1 Поняття бінарного зображення

Бінарне зображення – це двоколірна картинка, на якій представлені один або кілька об'єктів одного кольору на тлі, що має інший колір. Такі зображення представляються в комп'ютері у вигляді матриці точок, кожна з яких може мати лише одне з двох значень кольору, умовно позначаються через 0 або 1. Ці точки називаються пікселями. Не порушуючи загальності, будемо вважати бінарне зображення «чорно-білим»: точки об'єкта є чорними (значення кольору дорівнює 1), а точки фону - білими (значення кольору дорівнює 0). Таке розфарбування відповідає звичним способом рисунням чорною тушшю на білому папері. Бінарні зображення вимагають мінімальних обсягів пам'яті для зберігання (досить одного біта на піксель) і дозволяють мінімізувати витрату інших обчислювальних ресурсів для вирішення завдань обробки та аналізу.

Бінарні зображення в системах машинного зору формуються, як правило, в результаті сегментації більш складних – напівтонових монохромних, або кольорових зображень, які, в свою чергу, виходять в результаті сканування картин, документів, фотографій, а також введення відеоінформації безпосередньо з цифрових фото- і відеокамер.

У сучасних системах машинного зору бінарні зображення відіграють дуже важливу роль. Вони дозволяють виділити об'єкт інтересу в зображенні і є основою для аналізу і розпізнавання форми об'єктів в зображенні [6-9].

Процес перетворення кольорових і напівтонових зображень в бінарні називається бінаризація. Множина перетворень, використаних при бінаризації, є численною і різноманітною [6, 9]. До них відносяться методи точкових перетворень, згортки, посилення країв, виділення низькочастотних і високочастотних компонент зображення і т. д. Результатом таких перетворень, зокрема, є визначення властивостей точок зображення, за якими можна судити про те, чи належать чи точки об'єкту або їх слід віднести до фону. На основі

перевірки виконання цих властивостей для всіх точок зображення і створюється, в кінцевому рахунку, бінарне двокольорове растрове зображення, в якому чорні точки відповідають об'єкту, а білі - фону. Замітимо, що в явному вигляді побудова такого бінарного зображення може і не здійснюватися, але неявно воно обов'язково присутнє й, в принципі, виділений таким чином набір властивостей об'єкта може бути візуалізований у вигляді чорно-білого зображення.

Незважаючи на те, що зображення, представлене у вигляді матриці точок, є надзвичайно зручним для введення, запам'ятовування, обробки в комп'ютері, людині звичніше і простіше при описі форми об'єктів оперувати геометричними поняттями. Прості геометричні фігури, такі як багатокутники, кола, а також більш складні фігури, наприклад, обмежені сплайнів кривими, однозначно розуміються усіма, мають чітке математичний опис і можуть бути легко використані для подання та перетворення форми об'єктів. У той же час, описати яку-небудь форму, наприклад, коло або багатокутник, в термінах матриці точок представляється набагато більш важким завданням. Тому використання неперервних моделей істотно спрощує створення алгоритмів аналізу, класифікацією, перетворення форми об'єктів. Неперервний опис форми відповідає інтуїтивному уявленню людини про «суцільному» об'єкті, наприклад, нарисованому фарбою на папері за допомогою пензля. Зображення може бути при цьому досить складним в тому сенсі, що воно складається з великого числа окремих плям, кожне з яких може мати, в свою чергу, внутрішні отвори.

Така відмінність у людському і машинному описі форми ставить розробника алгоритмів аналізу зображень перед дилемою. Можна використовувати дискретний опис форми об'єкта у вигляді матриці точок, розраховуючи на простоту комп'ютерної обробки таких даних. Але при цьому потрібно витратити великі зусилля на те, щоб переформулювати людські геометричні правила і алгоритми опису, порівняння, перетворення форми об'єктів в операції над такими матрицями. Складність такої алгоритмізації

може виявитися значною і може перекрити виграш від простоти операцій над матричними зображеннями.

Інший шлях полягає в перетворенні форми об'єктів з дискретного уявлення у вигляді матриці точок в будь-яку неперервну модель з тим, щоб виконувати операції з формами в термінах, адекватних розуміння і досвіду людини. У цьому випадку витрати на побудову неперервної моделі досить високі, але зате істотно спрощується створення алгоритмів аналізу, порівняння та перетворення форми об'єктів, представлених на зображенні.

Ці два походу, умовно названі нами дискретним і неперервним, являють собою крайні позиції, між якими можуть існувати змішані варіанти: та чи інша частина обробки виконується над матричних зображенням, а потім можна здійснити перехід до неперервних форм.

Розглянемо підхід, що базується на дискретному «матричному» рівні обробки, який повинен бути вирішений на етапі бінаризації – виділення об'єкта інтересу на зображенні. Результатом такого виділення є бінарне зображення, в якому точки об'єкта мають один колір, а точки фону - інший (для визначення чорний і білий відповідно). Після цього пропонується за бінарним зображенням побудувати неперервний опис форми виділених об'єктів. Такий перехід від матричної форми до неперервної моделі будемо називати дискретно-неперервним перетворенням форми зображення.

Дискретно-неперервне перетворення будує для зображення, представленого матрицею точок двох кольорів деяку множину «неперервних» примітивів, наближено описують це зображення. Неперервні примітиви – це геометричні фігури, які допускають простий аналітичний опис на площині – в двовимірному евклідовому просторі. Таке «неперервне зображення» має бути підібрано досить тонко, щоб адекватно відобразити особливості вихідного дискретного зображення. Таким чином, в рамках цього підходу дискретно-безперервне перетворення є головним етапом обробки зображення, який покликаний перевести зображення з формату, зручного для комп'ютерного кодування і зберігання, в формат, зручний для людського розуміння.

З точки зору математики дискретно-неперервне перетворення є апроксимація деякого об'єкта одного класу (заданого чорно-білої матрицею) об'єктом іншого класу, володіючи властивістю «неперервності». Іноді подібну апроксимацію називають згладжуванням. Завдання згладжування функції, що має низку ступінь регулярності (наприклад, розривної або не диференційованої), полягає в апроксимації її іншою, більш регулярної функцією. При бажанні можна розуміти в підборі неперервного аналога для дискретного зображення саме завдання згладжування деякої функції.

Важливу роль в реалізації неперервного підходу відіграє вибір класу неперервних примітивів для апроксимації форми дискретних об'єктів. В якості універсальної математичної моделі для опису форми об'єктів в зображенні ми будемо використовувати поняття фігури – замкнутої обмеженої області, межа якої складається з кінцевого числа жорданових кривих, які не перетинаються. Такий опис цілком відповідає вербальному визначенню поняття форми зі словника: «зовнішні обриси, зовнішній вигляд предмета». Дійсно, замкнута обмежена область має чітко виражені зовнішні обриси у вигляді границі, має внутрішню частину, що складається з усіх внутрішніх точок області. Ця внутрішня частина відокремлена границею від зовнішнього оточуючого простору. І при цьому «зовнішні обриси» можуть мати як дуже простий, так і самий складний і складний вид.

Здатність комп'ютера швидко оперувати з об'єктами, що описує замкнуті області, визначається тим, наскільки зручно представляти ці області в числовому вигляді. Саме від формату представлення об'єктів залежить швидкість обробки інформації в комп'ютері. А від швидкості обробки, в свою чергу, залежить, чи можна вирішувати завдання в реальному часі, обробляючи, наприклад, в секунду 5-10 кадрів, знятих відеокамерою, або на обробку одного кадру знадобиться кілька хвилин. Відповідно, область використання алгоритмів в системах аналізу зображень принципово змінюється. Підхід полягає у використанні трьох різних способів неперервного представлення фігур: граничного, скелетного і циркулярного.

Граничне представлення фігури полягає в описі границі у вигляді замкнутих кривих ліній кінцевої довжини. Границя однозначно визначає замкнуту область і, разом з тим, добре характеризує її зовнішні контури, тобто форму. Важливу роль в такому представленні відіграє вибір способу чисельного опису меж області.

Для цього ми використовуємо багатокутники, а також замкнуті сплайнні лінії – криві Безьє. Скелет замкнутої області, що становить основу скелетного представлення фігури, – це множина точок, які є центрами максимальних кіл, вписаних в цю область. Скелет фігури є собою плоский граф, структура якого дуже добре відображає особливості форми об'єкта. Додаткову інформацію для аналізу форми дає радіальна функція скелета, яка ставить у відповідність кожній його точці радіус максимально вписаного кола з центром в цій точці. Опис скелета і радіальної функції становить скелетне представлення фігури.

Циркулярний представлення фігури ґрунтується на використанні однопараметричних сімейств кіл. При цьому центри кіл заповнюють гладкі лінії кінцевої довжини, а радіус кола є гладкою функцією параметра сімейства. Циркулярне представлення замкнутої області є розвитком поняття скелета і дає можливість не тільки аналізувати форму об'єкта, а й здійснювати її перетворення. Потреба в таких перетвореннях виникає в задачах порівняння форми гнучких об'єктів і в задачах синтезу нових форм на основі перетворення наявних еталонів.

1.2 Аналіз границь зображень

Поняття границі і скелета об'єкта давно і широко використовуються в якості базових інструментів аналізу форми зображень.

Границя, як основний дескриптор форми, є найбільш зрозумілим і природним інструментом для її аналізу. Обчислення меж в бінарному зображенні ніяких особливих труднощів не викликає. Зазвичай це множина точок об'єкта, що мають сусідні точки, які не належать об'єкту. Поєднавши ці граничні точки, отримуємо багатокутники, що описують контури границі. Знаючи границю, можна обчислити периметр, площа, опуклу оболонку фігури, її топологічну структуру (кількість дірок), визначити характер краю (сильно або слабо звивистий) і т.п. Скелет є більш складною конструкцією, отримання та використання якої вимагає рішення набагато складніших завдань. Протягом сорока років з моменту виходу в світ першої роботи зі скелета [10], яка б показала широкі можливості їх використання при аналізі та розпізнаванні зображень, опубліковані сотні статей по методам побудови скелетів і їх застосуванням в алгоритмах обробки зображень. Однак сам факт такого великого числа публікацій свідчить про те, що отримати досконалі алгоритми скелетизації, придатні для будь-яких реальних цифрових зображень, поки не вдається.

Спочатку визначення скелета було сформульовано в неперервному вигляді для замкнутих областей з неперервною границею на евклідовій площині [10]. Природно, що відразу ж виникло питання про адаптацію цього поняття до растрових зображень. Однак така адаптація зіткнулася з серйозними труднощами.

Перша фундаментальна проблема полягає в тому, що для безпосереднього використання понять неперервної геометрії в аналізі дискретних зображень потрібно коректно вирішити задачу апроксимації границі дискретного об'єкта неперервними лініями. Однак, незважаючи на вдавану простоту, зробити це не завжди вдається. Зокрема, згаданий вище традиційний «наївний» спосіб представлення границі бінарного зображення багатокутниками з вершинами у всіх граничних точках не є коректним, оскільки отримана замкнута лінія не обов'язково буде простим багатокутником. Виявляється, взагалі неможливо математично коректно

визначити поняття межі зображення в термінах дискретної матриці таким чином, щоб зберегти адекватність з неперервним аналогом. Причина цього полягає в невиконанні теореми Жордана для ліній на растрі. Якщо вважати, що в дискретній матриці, яка описує зображення, лінія представляють собою ланцюжок суміжних пікселів, то виявляється, що замкнута растрова лінія, взагалі кажучи, не завжди поділяє дискретну площину на два зв'язані множини, для яких вона є границею. Тому наївний підхід до безперервної апроксимації ганиці зображень не дає доброго геометричного опису форми. Зокрема, отримана таким чином межа може мати самоперетини. Поява самоперетинів сильно ускладнює інтерпретацію моделей форми, свідчить про їхню неадекватність.

Друга проблема полягає в тому, що не для всяких замкнутих областей, обмежених Жорданова кривими, існують ефективні алгоритми побудови скелетів. Обчислювальна геометрія представляє ефективні алгоритми побудови скелетів тільки для фігур, обмежених простими багатокутниками [11, 12]. Скелетизація фігур з нелінійною границею вимагає занадто великих витрат часу. Виникає необхідність зведення складних багатозв'язних об'єктів з криволінійними межами до більш простих з кусочно-лінійними межами, для яких існують ефективні алгоритми скелетизації. Необхідно побудувати таку полігональну апроксимацію об'єктів бінарного зображення, щоб все багатокутники були простими і не перетиналися один з одним. Виконати цю вимогу в складних зображеннях не просто. Слід також зазначити, що, незважаючи на доведене існування ефективних алгоритмів побудови скелета складної багатозв'язної фігури, практична розробка таких алгоритмів також є непростим завданням.

Нарешті, існує ще одна проблема, пов'язана з інтерпретацією отриманого скелета апроксимуючої багатокутної фігури. Чим точніше апроксимує багатокутник межу фігури, тим більше у нього вершин і, відповідно, тим більше в його скелетоні шумових ребер, що не несуть ніякої інформації про форму об'єкта.

Мабуть, ці труднощі привели до того, що більшість відомих рішень задачі скелетизації бінарних зображень не використовує потужних і елегантних методів обчислювальної геометрії. Незважаючи на велику кількість робіт по скелетизації, до теперішнього часу нам не відомі інші приклади успішної реалізації чисто неперервного підходу, що є найбільш наближеним до гео-метричного рішення (неперервна межа + неперервний скелет).

Невирішені проблеми неперервного підходу привели до того, що найбільшого поширення набули прагматичні методи, що базуються на чисто дискретному підході (дискретна межа + дискретний скелет) [6, 9, 13, 14]. Дискретні методи не претендують на повну адекватність з неперервним аналогом, хоча і прагнуть цю аналогію підтримати. В рамках дискретного підходу, як саме собою зрозуміле, межа зображення визначається як множина точок об'єкта, що мають сусідні точки фону. А для скелета строгих визначень взагалі не пропонується. Зазвичай під скелетом розуміється результат деякого процесу перетворення зображення, наприклад, так званого стоншення за рахунок перефарбування граничних пікселів.

Всякий дискретний аналог коректної неперервної моделі добре працює «в межі», тобто в тих випадках, коли крок дискретизації дуже малий. Стосовно до зображень крок дискретизації обчислюється в розмірах пікселів. У зображеннях з високою роздільною здатністю, де розміри об'єктів в сотні разів перевершують розмір пікселя, дискретні межі і скелети часто виявляються цілком придатними для аналізу форми. Але в зображення низької роздільної здатності їх використання виявляється неприйнятним.

Недоліки дискретного підходу, що є постійним мотивом для його розвитку, відзначаються в багатьох роботах. Дискретна межа в ряді випадків не дозволяє визначити топологічну структуру об'єкта – кількість і вкладеність граничних контурів, особливо для зображень з низькою роздільною здатністю. Дискретні скелети, отримані методом стоншення, не є серединими осями в евклідовій метриці, а скелети, отримані за допомогою так званої дистанційній

карти, виявляються не гомотопними вихідному зображенню, оскільки мають іншу зв'язність.

Крім цих недоліків, відзначимо ще одну важливу особливість. Дискретні границі і скелети описуються в формі нового дискретного зображення, що є результатом перетворення вихідного зображення. Для того щоб вирішувати задачі аналізу, розпізнавання, порівняння і перетворення форми, потрібне створення нових, часто нетривіальних алгоритмів обробки отриманих зображень границь і скелетів. Наприклад, для дискретних границь потрібно «витягнути» граничні пікселі в ланцюжки, тобто впорядкувати пікселі і утворити з них растрові лінії. Потрібно визначити, які лінії задають зовнішні, а які внутрішні границі об'єктів. В скелеті потрібно виділити графову структуру, тобто знайти пікселі - вершини графа і знайти растрові лінії, що описують його ребра і т.п.

Крім «чистого» дискретного підходу відомі також варіанти комбінованого проміжного рішення (дискретна межа + неперервний скелет) [7, 15, 16]. Метод [16] полягає в побудові діаграми Вороного для множини граничних точок зображення. Безпосередньо як скелет розглядається деякий підграф діаграми Вороного, одержаний на основі процедури регуляризації. Перевагою методу є те, що отриманий скелет є неперервним, відповідає евклідової метриці, має радіальну функцію, що визначає його корисність для аналізу форми зображень. Друга важлива перевага цього методу полягає в його високій швидкості за рахунок використання ефективних алгоритмів обчислювальної геометрії для побудови діаграм Вороного великої розмірності. Недоліками методу є відсутність строгого визначення поняття скелета, а також нерегулярність, зигзагоподібних гілок скелета. У методі [15] виходить скелет в неевклідової метриці і з порушеннями зв'язності.

Слід зазначити ще одну проблему, яку потрібно вирішувати при аналізі форми зображень. Вона пов'язана з високою чутливістю границь і скелетів до шумових ефектів в зображенні. Невеликі зміни границі області можуть привести до істотних змін топологічної структури скелета. Це приводить до

того, що для об'єктів, що мають на погляд людини дуже високий ступінь подібності, комп'ютерні алгоритми будують дуже різні скелети. Для того щоб виділити в формі стійкі властивості, інваріантними до шумових ефектів, розробляються різні методи регуляризації границь [17, 18] і скелетів [19] для поліпшення їх властивостей. Потрібно сказати, що регуляризація необхідна для всіх скелетів, як неперервних, так і дискретних. Питання полягає в тому, наскільки коректно ставиться завдання регуляризації і наскільки зручно регуляризують скелети. Аналіз робіт по регуляризації дискретних скелетів показує, що застосовувані тут методи, як правило, використовують евристичні прийоми і не мають строгого математичного обґрунтування.

1.3 Аналіз систем автоматизованої мікроскопії

Методи та алгоритми для аналізу біомедичних зображень надають спеціалізовані програмні системи. Система автоматизованої мікроскопії (САМ) – це програмно-апаратна система, що дозволяє автоматизувати аналіз біомедичних зображень в умовах клінічної практики. Основне завдання програмної частини САМ полягає у обробці вхідного зображення, ідентифікації мікрооб'єктів і визначенні ознак для подальшої постановки діагнозу лікарем-фахівцем або у автоматичному режимі [20-23]. Більшість існуючих САМ володіють лише стандартним набором алгоритмів та методів комп'ютерного зору і не мають інтерфейсів для різних груп користувачів [24]. Тому існує проблема розробки систем автоматизованої мікроскопії з елементами штучного інтелекту, які спрощували б взаємодію різних користувачів із системою та давали консультації при постановці діагнозу. Інструменти аналізу САМ BioImageXD [25] охоплюють як об'єктно-орієнтовані, так і воксельно-обґрунтовані підходи. Програмне забезпечення

може бути використане як для простої візуалізації багатоканальних тимчасових стеків, так і для складної 3D обробки.

Малов А. М. [26] наводить методику комп'ютерної обробки багатоспектральних зображень препаратів, які використовуються в медико-біологічних дослідженнях. Методика дозволяє в інтерактивному режимі підготувати до проведення автоматичного вимірювання морфометричних параметрів мікрооб'єктів в складних випадках при наявності на зображенні декількох груп різних мікрооб'єктів. Методика відрізняється цілеспрямованістю візуалізації, що дозволяє швидко виділити мікрооб'єкти інтересу. Методика впроваджена в програмне забезпечення апаратно-програмного комплексу «ДіаМорф». Yokoyma Yuhki [27] наводить результати досліджень раку молочної залози з використанням апаратно – програмного забезпечення MetaMorph 7.1.

Автори [28] аналізують систему OncoDoc для підвищення якості діагностування раку молочної залози. Система підтримки прийняття рішень OncoDoc призначена для забезпечення терапевтичних рекомендацій для хворих на рак молочної залози. Розроблено інструмент перегляду бази знань, який виконаний у вигляді дерева рішень. Система OncoDoc містить БД, що значно вирізняє її серед інших систем.

Dercksen V. [22] розглядає та аналізує інтерактивне програмне середовище для візуалізації, коректування та аналізу 3D-морфології.

Vrekoussis T [23] проводить аналіз досліджень в області імуногістохімії раку молочної залози з використанням програмного забезпечення ImageJ на основі RGB - моделі гістологічних зображень.

Медовий В.С. у [29] наводить структуру, переваги та недоліки інформаційної автоматизованої системи МЕКОС-ЦГ. Дана система володіє БД, що надалі дозволить об'єднувати зусилля лабораторій для якісної діагностики.

Комплекс Имаджер-ЦГ складається з стандартної техніки - мікроскопа, відеокамери, комп'ютера з платою відеозахопленням та прикладного

програмного забезпечення. Призначенням комплексу є візуалізація даних та розпізнавання образів, оперативне отримання кількісної інформації про властивості мікрооб'єктів, оцінку їх характеристик з метою підвищення якості діагностики в різних областях медицини.

Серед найпопулярніших САМ виділимо такі: МЕКОС-Ц2, TissueFAXS, AnalySIS Five, BioVision, ВидеоТест Морфо 5.2, BioImageXD, Ariol, ImageJ, analySIS FIVE, Motic Images Advanced 3.2, ДиаМорф, Motic ВидеоТест Морфо 5.2, Cell D.

Основні критерії оцінювання САМ такі: режими роботи алгоритмів сегментації зображень, обчислення числових характеристик мікрооб'єктів, можливість роботи із зовнішніми програмами, можливість виводу результатів у вигляді графіків та гістограм.

Усі з розглянутих САМ володіють функцією попередньої обробки зображення та визначення характерних ознак. Одним із недоліків ImageTool, ИМАДЖЕР-ЦГ та ScreenMeter є відсутність можливості використання плагінів, що значно обмежує можливості САМ. Важливим елементом усіх систем автоматизованої мікроскопії є можливість формування звітів після дослідження та їх подальший вивід на екран або друк.

Такі системи, як BioVision, TissueFAXS, МЕКОС-Ц2, Ariol є комбінованими і надають комплекси для роботи у різних напрямках. Наприклад, комплекс для дослідження тканин, комплекс автоматичного аналізу мазків крові, комплекс для роботи з флуоресценцією і імуногістохімічними зразками та ін. Ще однією перевагою таких систем над їхніми аналогами є наявність великої кількості фільтрів та методик для обробки зображень, що дозволяє більш точно обчислити характеристичні ознаки.

Програмні системи BioVision, ВидеоТест, МЕКОС-Ц2, TissueFAXS PLUS, Ariol володіють БД для зберігання зображень та результатів дослідження. До того ж МЕКОС-Ц2 надає механізм роботи з віддаленою БД. Використання БД у подібних системах значно спрощує роботу медичних

працівників та дозволяє формувати звіти та висновки в процесі роботи. Перевагою BioVision, Cell D є використання фазового аналізу (сегментація мікрооб'єктів на основі гістограм).

Image-Pro Plus надає більше п'ятдесяти інструментів вимірювання мікрооб'єктів. За допомогою Image-Pro Plus і додатковим 3D доповненням можна досліджувати глибину зображень, візуалізацію та взаємодію в трьох вимірах.

Відмінністю програмної системи МЕКОС-Ц2 є можливість сортування та візуалізації мікрооб'єктів аналізу, автоматичне формування компактної представницької цифрової моделі препарату для візуального контролю, засоби колективного локального та дистанційного обслуговування потоків біоматеріалів та телеморфологія.

Перевагами програмної системи TissueFAXS є автоматичне багатоканальне захоплення зображення, виділення окремих клітин за запатентованими алгоритмами, автоматичне виділення і розділення ядер, робота із замороженими і парафіновими зрізами, підвищена швидкість аналізу за рахунок використання багатоядерних комп'ютерних робочих станцій, створення шаблонів і профілів для повторного використання перевірених параметрів. Використання технології розподілених обчислень дозволяє значно пришвидшити процес опрацювання зображення або групи зображень, збільшуючи таким чином продуктивність.

Програмна система САМ Ariol спеціалізується на виявленні та діагностиці раку молочної залози. Її особливістю у порівнянні з іншими системами є можливість підключення до основного обладнання до дев'яти робочих станцій. У підсумку кожна лабораторія або дослідницька група отримують доступ до централізовано відсканованих препаратів, проводять спеціалізований аналіз і зберігають результати в загальній інформаційній базі. Спеціалізовані програмні модулі автоматизованої платформи Ariol призначені для кількісного імуногістохімічного дослідження, FISH аналізу, морфометричного аналізу гістологічних препаратів в світлому полі і

дослідження за методом імунофлуоресценції, автоматизованого пошуку окремих ракових клітин і мікрометастазів.

Більшість САМ реалізують у своєму складі відомі алгоритми обробки зображень (наприклад, алгоритм крос-кореляції, алгоритм найближчого сусіда, алгоритм локалізації границь на основі вейвлет перетворення, оптичне розрізнення звичайної флуоресценції, алгоритм зшивання, відображення достовірності, довірче відображення алгоритм водоподілу), однак деякі з них, наприклад TissueFAXS чи МЕКОС-Ц2 працюють з оригінальними запатентованими алгоритмами.

Однією з переваг Ariol та BioVision є можливість використання Grid технології для швидшої обробки інформації. Система Ariol дозволяє ефективно розподіляти спеціалізовані дослідження і отримувати комплексні результати аналізів за рахунок підключення до основного обладнання до робочих станцій.

Такі САМ, як Ariol, МЕКОС-Ц2, BioVision, TissueFAXS, Image-Pro Plus надають спеціалізоване апаратне забезпечення (мікроскоп, камера, адаптер для камери). Так, Ariol - автоматизована платформа на базі моторизованого дослідного мікроскопа Leica DM6000, а TissueFAXS – на базі спеціалізованих мікроскопів фірми Nikon. Інші ж дозволяють використовувати універсальні апаратні засоби, які впливають на ціну.

Недоліками таких систем, як Image-Pro Plus, Cell D, QCapture PRO 6.0 у порівнянні з іншими, є відсутність можливості передачі даних в зовнішнє програмне середовище.

Порівняльну характеристику систем автоматизованої мікроскопії наведено у таблиці 1.1. В таблиці наявність позначено «+», відсутність «-», наявність додаткового модуля «+/-»).

Порівняльну характеристику алгоритмів середнього рівня обробки зображень наведено у таблиці 1.2.

В результаті порівняння можна зробити висновок, що більшість систем мають у своєму складі подібний набір реалізованих алгоритмів сегментації. Серед наведених САМ потрібно виділити BioImageXD та AxioVisison, які мають у своєму складі найбільший набір реалізованих алгоритмів сегментації.

Програмна система ImageJ у своєму складі має лише декілька алгоритмів, однак надає можливість інсталяції додаткових модулів.

Таблиця 1.1 – Порівняльна характеристика САМ на низькому рівні обробки зображень

Критерії	Image j	Axio Visison	BioImage XD	motisc	QCapture PRO	Icy	Image Pro Plus	Micro Manager	analySIS FIVE
Контрастування	+	+	+	+	+	+	+	+	+
Зміна рівня яскравості	+	+	+	+	+	+	+	+	+
Алгоритми виділення порогів: Лапласа, Собеля	- +	+ +	- +	+ -	+ +	- +	+ +	+ +/-	+ +
Фільтри: Гаусовий Медіанний Усереднюючий	+ + +	+ + +	+ + -	+ + -	+ + -	+ - -	+ + -	+ +/- +/-	+/- +/- +/-
Швидке перетворення Фур'є	+	+	+	+	+	-	+	+	+

Ключовим етапом аналізу є етап виділення та розпізнання мікрооб'єктів на зображенні, наприклад ядра, цитоплазми. Порівняльні характеристики систем автоматизованої мікроскопії на високому рівні опрацювання зображень зображено у таблиці 1.3.

Проведений аналіз показав, що сучасні САМ мають у своєму складі основні алгоритми опрацювання зображень. Проте, у них відсутній адаптивний інтерфейс і можливість проведення діагностування на основі експертних знань.

Таблиця 1.2 – Порівняльна характеристика алгоритмів середнього рівня обробки зображень

CAM	Метод K – means	Розум ні ножиц і	Snak es	Метод водо- розподіл у	Алгори тм Круска ла	Grab Cut	Mea n shift	Кодува ння контури в	RAN SAC	Перетво рення Хафа
ImageJ	+	+/-	+	+	+/-	+/-	+	-	+	+/-
Axio Vision	+	-	+	+	+	+	-	+/-	-	+
BioImage	+	+	+	+	+	-	-	+/-	+	+/-
Motic	+	-	+	-	+	-	-	-	+	-
QCapture	+	+	-	+	-	-	-	+/-	+	+
Image Pro	+	-	+	-	+	+	-	-	-	+
Icy	+	-	-	+	-	-	+	-	-	-
Micro Manager	+	+/-	+	+/-	-	+/-	-	-	+	+

Тому актуальним є розробка інтелектуальної САМ для діагностування передракових станів молочної залози із спеціалізованими програмними модулями для різних типів користувачів для полегшення роботи медичного персоналу.

Таблиця 1.3 – Порівняльна характеристика САМ на високому рівні опрацювання зображень

Критерії	Image J	Axio Visio n	BioImag e XD	moti c	QCaptu re PRO	Icy	Imag e Pro Plus	Micro Manage r	analySI S FIVE
Автоматична адаптація до зображення	-	-	+	-	-	-	+	-	-
Детектування об'єктів	+/-	+/-	-	-	-	-	+/-	+/-	-
Порівняння зображень	-	+	+	+	+	+	+	-	-
Класифікатори нейронні мережі SVM	- +/-	+	-	-	-	-	- +	- +/-	- -

1.4 Аналіз біомедичних зображень

Сучасна медична діагностика нерозривно пов'язана з візуалізацією внутрішніх структур біомедичних об'єктів.

Медична візуалізація (медична інтроскопія) – це методика і процес створення візуальних зображень внутрішніх органів з метою проведення клінічного аналізу і медичного втручання.

У наукових дослідженнях медична візуалізація є субдисципліною в області біомедичної інженерії, медичної фізики або медицини в залежності від контексту: отримання зображень (наприклад, рентгенографія), біомедична інженерія, медична фізика та інформатика тощо.

Зображення служать для подання інформації у візуальному вигляді. Вони є однією з найзручніших форм представлення інформації при діагностуванні органів людини в медицині.

Медичне зображення – це структурно-функціональний образ органів людини, призначений для діагностики захворювань і вивчення анатомофізіологічної картини організму.

Біомедичне зображення – це структурно-функціональний образ органів людини і тварин, призначений для діагностики захворювань і вивчення анатомофізіологічної картини організму.

Біомедичні зображення не завжди забезпечують необхідну інформацію для однозначної ідентифікації захворювань [30]. Поділ біомедичних зображень за критеріями є умовним [31].

За критерії класифікації біомедичних зображень візьмемо такі: 1) спосіб отримання зображень, 2) тип зображення, 3) розмірність зображення. Тоді класифікацію біомедичних зображень подамо у такому вигляді (таблиця 1.4).

Таблиця 1.4 – Класифікація біомедичних зображень

Критерій класифікації		
За способом отримання зображення	За типом зображення	За розмірністю
Цифрова радіологія. Комп'ютерна томографія. Ядерно–магнітний резонанс. Ультразвук. Мікроскопія.	RGB зображення (мікроскопія, еластографія, термографія). Зображення в градаціях сірого (рентгенографія, томографія). Чорно – біле (ультразвукове дослідження)	2D (всі зображення незалежно від способу отримання). 3D (послідовність радіологічних зображень, томографічне зображення динамічного об'єкта)

Рентгенографічні зображення – це 2D зображення в форматі градацій сірого і за способом отримання відносяться до класу цифрової радіології. Такий механізм візуалізації використовує широкий промінь рентгенівського випромінювання для отримання зображення. Рентгеноскопія зображує внутрішні структури організму подібно радіографії, але використовує рентгенівські промені меншої потужності дози опромінення [32].

Результатом магнітно-резонансної томографії (МРТ) є 2D або 3D зображення в градаціях сірого (МРТ зображення). Перевагою використання 2D МРТ зображень є вища швидкість формування зображень у порівнянні з 3D. Також 2D зображення краще підходять для дослідження довгих судин. Перевагою використання 3D зображень є висока просторова роздільна здатність у порівнянні з 2D і відображення об'єкту у тривимірному просторі. Принцип роботи пристрою такий: машина МРТ випромінює радіочастотний

(РЧ) імпульс на резонансній частоті атомів водню молекул води. Радіочастотні антени ("РЧ-котушки") відправляють імпульс в область дослідження тіла. РЧ-імпульс поглинається протонами, в результаті чого їх напрямок змінюється відносно напрямку основного магнітного поля. Коли радіочастотний імпульс вимикається, протони "розслабляються" і назад вирівнюються первинним магнітом і випромінюють при цьому радіохвилі. Це радіовипромінювання від атомів водню виявляється і реконструюється в зображення [33]).

Позитронно-емісійна томографія (ПЕТ) використовує розпізнавання збігів для зображень функціональних процесів. Короткоживучий позитрон, який випромінює ізотоп, об'єднаний з органічною речовиною (глюкозою) створює F18-фтордезоксиглюкозу, що може бути використана в якості маркера метаболічної утилізації. Позитронно-емісійно-томографічне зображення розподілу активності по всьому тілу може швидко показати пухлину або інфекцію.

Результатом ультразвукового дослідження є 2D або 3D ультразвукове зображення. 2D зображення – це двовимірне чорно-біле зображення, а 3D зображення є об'ємним RGB зображенням. Ультразвукове дослідження (УЗД) використовує високочастотні звукові хвилі, які по-різному відбиваються від тканини і утворюють зображення. Звукові хвилі високої частоти надсилаються в тканини і в залежності від складу різних тканин сигнал буде ослаблений і повернеться через певні проміжки часу. Даний підхід є безпечним і не викликає побічних ефектів.

Ехокардіографія – метод УЗД, спрямований на дослідження морфологічних і функціональних змін серця і його клапанного апарату. Результатом ехокардіографії є чорно-біле зображення. Метод базується на реєстрації відбитих від структур серця ультразвукових сигналів. Принцип дії методу ґрунтується на здатності ультразвуку відбиватися при взаємодії із середовищами різної акустичної щільності. Даний метод дозволяє встановити стан м'яких тканин, визначити товщину стінок серця, стан клапанного апарату та обсяг порожнин серця [34].

Цитологічними називають RGB мікроскопічні зображення, в яких клітини знаходяться поза межами тканини. Виділяють такі два типи розташування клітин на зображенні: з окремо розташованими клітинами та з клітинами, об'єднаними в конгломерати. Цитологічний метод дозволяє виявляти різні патології в розвитку клітин.

Гістологічні зображення (ГЗ) – це RGB зображення, що формуються в результаті мікроскопічного дослідження. Біоматеріалом для гістологічного дослідження можуть бути зразки різних тканин: шкіри, слизових оболонок, м'язів. Іноді для гістологічного дослідження надається мазок – клітини епітелію, які зняті за допомогою спеціальної м'якої щітки [35]. В основі дослідження цитологічного і гістологічного досліджень лежить морфологічний принцип аналізу.

Імуногістохімія (ІГХ) – це аналітичний метод визначення протеїнів (антигенів) у клітинах біологічних тканин на основі реакції антиген – антитіло. Результатом імуногістохімічного дослідження є двовимірне імуногістохімічне RGB зображення. ІГХ можна проводити на свіжозаморожених зразках або, найчастіше, на фіксованих в формаліні та залитих у парафінові блоки тканинах. Залиту в парафіні тканину мікротом нарізають у тонкий шар (4-5 мкм), який зафіксують на скельці для проведення подальших етапів ІГХ. Даний метод широко застосовують у базовій науково-дослідній роботі для визначення протеїнів або для підтвердження результатів інших методів дослідження білків.

1.5 Аналіз завдання на випускню кваліфікаційну роботу та постановка задач

Для реалізації задач випускної кваліфікаційної роботи, які впливають з мети, об'єкту та предмету необхідно:

- проаналізувати біомедичні зображення;
- здійснити аналіз алгоритмів виділення границь;
- провести аналіз систем автоматизованої мікроскопії;
- розробити алгоритми скелетонізації полігонів;
- розробити структуру програмної системи скелетизації мікрооб'єктів;
- розробити програмні модулі системи скелетизації об'єктів на цифрових зображеннях і провести тестування.

2 АЛГОРИТМИ СКЕЛЕТОНІЗАЦІЇ ЗОБРАЖЕНЬ

2.1 Поняття метричного простору та метрик

Метричним простором називається пара (X, d) , яка складається з деякої множини елементів X і відстані $d: X \times X \rightarrow R$, а саме однозначної, невід'ємної, дійсної функції $d(x, y)$, визначеної для $\forall x, y \in X$, яка задовольняє наступні 3 аксіоми:

- 1) $d(x, y) = 0 \Leftrightarrow x = y$ (аксіома тотожності);
- 2) $d(x, y) = d(y, x)$ (аксіома симетрії);
- 3) $d(x, z) \leq d(x, y) + d(y, z)$ (нерівність трикутника).

Невід'ємність доводиться за допомогою наступних міркувань:

$$0 = d(x, x) \leq d(x, y) + d(y, x) = 2d(x, y).$$

Евклідів простір – скінченновимірний дійсний векторний простір E із скалярним добутком. Названий на честь давньогрецького математика Евкліда із Александрії. Розширює двовимірну евклідову площину, до тривимірного простору і є поняттям Евклідової геометрії. Термін "евклідовий" дозволяє відрізнити ці простори від інших типів просторів, що можуть розглядатися в сучасній геометрії. Евклідів простір також узагальнюють і до більшої кількості вимірів.

В класичній давньогрецькій геометрії існує визначення евклідової площини і тривимірного евклідового простору, що ґрунтується на певних постулатах, в той час як інші властивості цих просторів виведені як теореми. Також використовувалися геометричні побудови для визначення раціональних чисел, що є відношеннями співмірних довжин. Коли алгебра і математичний аналіз набули достатнього розвитку, цей зв'язок зберігся і тепер більш загальним стало визначення Евклідового простору на основі векторних

просторів, що дозволяють використовувати декартові координати і методи алгебри та диференціального та інтегрального числення. Це означає, що точки визначають за допомогою трійок дійсних чисел, які називаються координатними векторами, а геометричні фігури описують рівняннями і нерівностями, що визначають співвідношення цих координат. Цей підхід також дозволяє легко узагальнити в. геометрію до евклідових просторів до просторів більшої розмірності.

Евклідов простір визначено за допомогою аксіом, які не вказують як саме мають бути представлені точки цього простору. Евклідов простір може бути побудований за допомогою декартової системи координат, як один із можливих способів його представлення. В такому випадку, Евклідов простір моделюють застосовуючи дійсний простір координат (\mathbf{R}^n), що має таку ж розмірність. Для одного виміру це була б шкала дійсних чисел; для двох вимірів, він представляється декартовою системою координат на площині; і для більшої кількості вимірів, це є координатний простір із трьома або більше координатами, що представлені дійсними числами. Математики позначають n -вимірний Евклідов простір як \mathbf{E}^n , якщо вони хочуть підкреслити його природу та властивості, але також використовують позначення \mathbf{R}^n , оскільки ці дві структури мають подібні властивості і їх як правило не розрізняють. Евклідові простори мають скінченну кількість вимірів.

Метрика, l_∞ -метрика, норма. Нехай X – деяка множина [36]. Метрикою на X називаємо функцію $d : X \times X \rightarrow R$, що задовольняє умови для всіх $x, y, z \in X$:

- 1) $d(x, y) \geq 0$, $d(x, y) = 0 \Leftrightarrow x = y$;
- 2) $d(x, y) = d(y, x)$ (симетрія);
- 3) $d(x, y) \leq d(x, z) + d(z, y)$ (нерівність трикутника).

Прикладами метрик на евклідовій площині R^2 є такі:

- a) $d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ (евклідова метрика);

б) $d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|$ (відома також під назвою l_1 – метрика);

в) $d((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$ (l_∞ – метрика).

Число $d(x, y)$ називають відстанню між $x, y \in X$. Якщо $r > 0$, то множину

$$B_r(x) = \{y \in X \mid d(x, y) < r\}$$

називають кулею радіуса r з центром у точці $x \in X$ (рисунок 2.1).

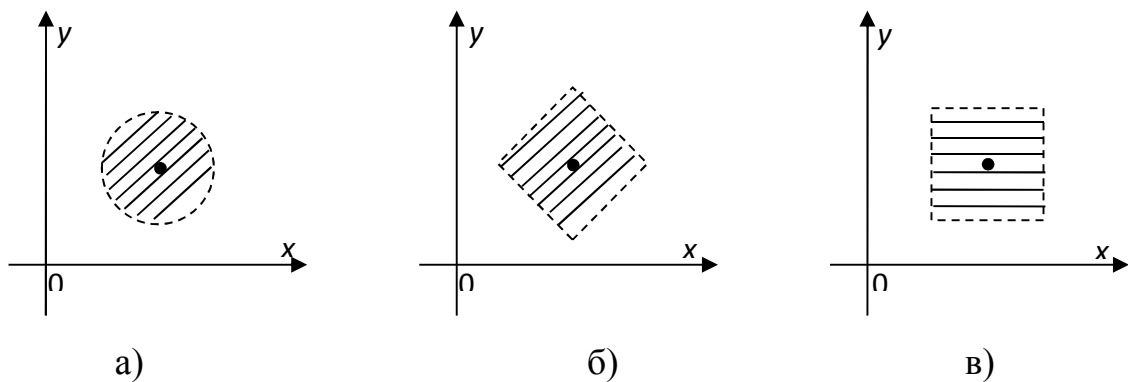


Рисунок 2.1 – Кулі для: а) евклідової метрики, б) l_1 – метрики, в) l_∞ – метрики

Підмножину $U \subset X$ у метричному просторі (X, d) називають відкритою, якщо для кожного $x \in U$ існує $r > 0$ таке, що $B_r(x) \subset U$. Сім'я \mathfrak{T} відкритих множин в X задовольняє умови:

- 1) $\emptyset \in \mathfrak{T}, X \in \mathfrak{T}$;
- 2) якщо $U, V \in \mathfrak{T}$, то $U \cap V \in \mathfrak{T}$;
- 3) якщо $U_\alpha \in \mathfrak{T}$ для кожного $\alpha \in A$ (тут A – деяка множина індексів), то

$$\bigcup_{\alpha \in A} U_\alpha \in \mathfrak{T}.$$

2.2 Поняття скелетону зображення

Нехай V – обмежена замкнена область зображення в площині R^2 . Межу області позначимо через $\partial V = C$. В даному випадку ми використовуємо не евклідову метрику на площині R^2 , а l_∞ – метрику, тобто метрику задану формулою $|A - B| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$, де (x_1, y_1) – координати т. A , (x_2, y_2) – координати т. B .

Нам знадобляться поняття скелета області, які запроваджені в роботі [37-38]. Позначимо для кожного $C \in V$, $pr(C) = \{C' \in \partial V \mid |C' - C| = \inf\{|C'' - C|\}, C'' \in \partial V$. Цю множину називають метричною проекцією т. C .

Для евклідової метрики використовується таке означення скелета:

$$sk(V) = \{C' \in \partial V \mid \text{множина } pr(C) \text{ складається більше, ніж з однієї точки}\}.$$

Воно еквівалентне іншому означенню: точка належить скелетові області, якщо вона є центром максимального вписаного в область круга.

Оскільки кулями в l_∞ -метриці є квадрати зі сторонами паралельними до координатних осей, приходимо до такого означення скелета: точка належить скелетові області, якщо вона є центром максимального вписаного в область квадрата зі сторонами, паралельними координатним осям.

Наведемо деякі приклади скелетів для евклідової та l_∞ -метрики (рисунок 2.2).

Порівняння цих прикладів показує, що складність скелетів може бути різною для цих двох метрик, і є області, для яких доцільно застосувати саме l_∞ -метрику.

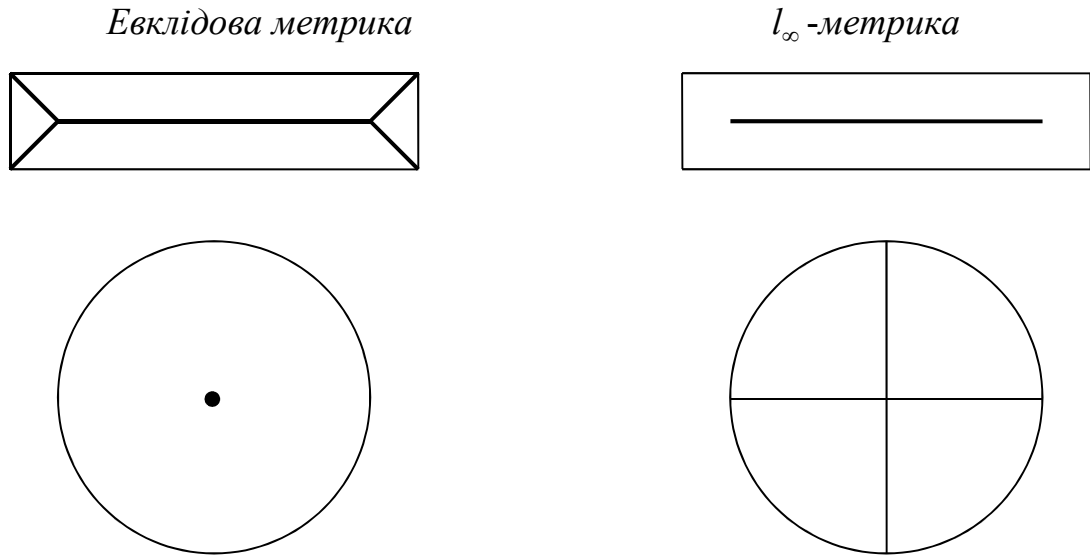


Рисунок 2.2 – Приклади скелетів

Приклад цілого класу таких областей задається такою теоремою.

Теорема 2.1. Нехай V – область на площині з кусково-лінійною межею. Тоді скелет $sk(V)$ є кусково-лінійним.

Отже, траєкторією центрів вписаних квадратів є прями. Оскільки множина вписаних квадратів, що лежать в області задається лінійними нерівностями, одержуємо, що скелет задається скінченним об'єднанням відрізків.

Для формалізації того факту, що форма скелета відображає форму області, нам знадобляться деякі поняття алгебраїчної топології [39]:

1. Нехай A – підмножина метричного простору X . Кажемо, що A є ретрактом простору X , якщо існує таке неперервне відображення $r : X \rightarrow A$, що $r(a) = a$ для кожного $a \in A$. Відображення r при цьому називається ретракцією.

2. Нехай X і Y – метричні простори. Сім'я неперервних відображень $f_t : X \rightarrow Y$ $t \in [0;1]$ називається гомотопією відображення f у відображення g , якщо

а) $f_0 = f$;

б) $f_1 = g$;

в) відображення $f_i(x)$ неперервне за сукупністю змінних.

3. Ретракція $r: X \rightarrow A$ називається деформаційною ретракцією (а A називається при цьому деформаційним ретрактом простору X), якщо тотожне відображення 1_X і r є гомотопні, тобто існує гомотопія між ними.

Аналогічно до скелета, який базується на евклідовій метриці, для кожного скелета в l_∞ -метриці справедлива наступна теорема.

Теорема 2.2. Скелет є деформаційним ретрактом області [40].

Нагадаємо, що область V називається однозв'язною, якщо кожна замкнену криву в V можна неперервно продеформувати в точку. Іншими словами, кожне відображення кола в V гомотопне сталому відображенню. Розглянемо множину всеможливих квадратів зі сторонами паралельними координатним осям. Позначимо її $Q(R^2)$. Відповідно для області $V \subseteq R^2$ позначимо через $Q(V)$ множину $\{A \in Q(R^2) \mid A \subset V\}$.

Оскільки кожен квадрат з $Q(R^2)$ однозначно визначається своїм центром і довжиною сторони, то $Q(R^2) = \{(x, y, a) \mid x, y \in R, a > 0\}$.

Маємо: $(x, y, a) \in Q(V) \Leftrightarrow (x, y) \in V$ і виконано умову: $|x - x'| \leq \frac{a}{2}$,

$|y - y'| \leq \frac{a}{2} \Rightarrow (x', y') \in V$. Звідси безпосередньо випливає, що межа області $Q(V)$ в R^2 є об'єднанням скінченного числа частин, що задаються лінійними нерівностями. Звідси безпосередньо випливає, що скелет області V є скінченим об'єднанням відрізків.

Підмножина $MQ(V) = \{A \in Q(V) \mid A \text{ максимальний}\}$ є аналогом множини $MAT(V)$ (medial axis transform), що досліджувалося в роботі [37].

З теореми 2.2 і приведених вище міркувань випливає такий наслідок:

Наслідок. Якщо область є однозв'язна, то $sk(V)$ є графом без циклів (дерево).

Згідно теореми Жордана [36, 39] для замкнутих областей існує перетворення, яке перетворює одну область в іншу, одну підобласть в іншу підобласть. Тобто для областей O_1 і O_2 маємо $O_2 = Q_{12}(O_1)$, де Q_{12} – перетворення області O_1 в O_2 . Дано визначення скелета. Нехай для кожного $C \in O$, $pr(C) = \{C' \in \partial O \mid |C' - C| = \inf\{|C'' - C|\}, \text{ де } C'' \in \partial O$. Цю множину називають метричною проекцією точки C .

Для евклідової метрики є таке визначення скелету:

$$sk(O) = \{C' \in \partial O \mid \text{множина } pr(C) \text{ складається більш ніж із однієї точки}\}.$$

Використаємо l_∞ -метрику. В ній означення скелету наступне: точка належить скелету області, якщо вона є центром максимального вписаного в область квадрата із сторонами, паралельними координатним осям.

Отже, для області O_1 будемо мати скелет $sk_1(O_1)$, а для O_2 – $sk_2(O_2)$.

В роботі [41] показано, що для кусково-лінійних апроксимованих контурів C_1 і C_2 скелети областей O_1 і O_2 є кусково-лінійними. Крім цього $sk_1(O_1)$ і $sk_2(O_2)$ є графами без циклів (деревами). Отже, скелети $sk_1(V_1, E_1)$ і $sk_2(V_2, E_2)$ є деревами, де V_1, V_2 – множини вершин, а E_1, E_2 – множини ребер (гілок) [42-44].

Для існування перетворення однієї області в іншу необхідно і достатньо, щоб виділені скелети були ізоморфно вкладеними графами. Два графи G_1 і $G_2 \subset R^2$ називаються ізоморфно вкладеними, якщо існує гомеоморфізм $u : R^2 \rightarrow R^2$ такий, що $u(G_1) = G_2$. Скелети $sk_1(V_1, E_1)$ і $sk_2(V_2, E_2)$ будуть ізоморфними, якщо існує перетворення $f : V_1 \rightarrow V_2$, для якого виконується така умова: $v, w \in V_1$ з'єднані ребром, то $f(v), f(w) \in V_2$ також з'єднані ребром.

Нехай задано два зображення Im_1 та Im_2 . Представимо кожне зображення у вигляді $Im_1 = C_1 \cup O_1$, $Im_2 = C_2 \cup O_2$ контурів і внутрішніх областей. Лінійно апроксимуємо контури зображень

$$C_i = \bigcup_{j=1}^l \{(x, a_j x + b_j | x \in [c_j; d_j])\}, \text{ де } a_j, b_j, c_j, d_j \in R, i = \overline{1,2}.$$

Знайдемо скелети двох зображень $sk_1(O_1)$ і $sk_2(O_2)$.

Дані скелети $sk_1(O_1)$ і $sk_2(O_2)$ в загальному випадку є неізоморфними деревами.

Введемо поняття ε -скелету.

Означення. ε -скелетом називається скелет, для якого виконується умова:

$$sk_\varepsilon(O) = \{x \in \text{Int}O \mid \text{diam } pr(x) \geq \varepsilon\},$$

$$pr(x) = \{y \in \partial O \mid |x - y| = \min_{y' \in O} |x - y'|\}.$$

Теорема 2.3. Скелет зображення складається з його ε -скелетів.

Дане твердження впливає із означення ε -скелету (рисунок 2.3).

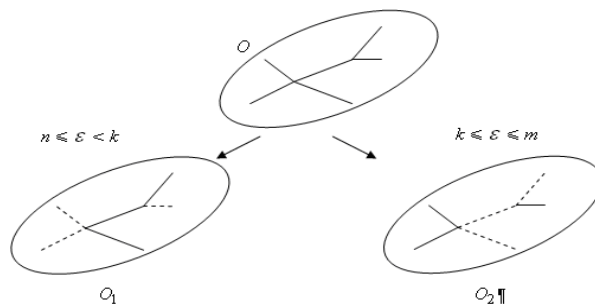


Рисунок 2.3 – Розклад скелету зображення на ε -скелети

Теорема 2.4. Існує $\varepsilon > 0$ таке, що ε -скелет є деформаційним ретрактом полігональної області.

2.3 Алгоритми скелетонізації полігонів

Розглянемо задачу скелетонізації неперервних фігур в практичному плані, маючи на увазі створення відповідних ефективних алгоритмів. З цієї точки зору можна виділити важливий клас фігур – багатокутні фігури, для яких розробка ефективних алгоритмів скелетонізації цілком можлива, тому що гілки скелета багатокутної фігури мають простий аналітичний опис - це або відрізки прямих ліній, або сегменти квадратичних парабол.

Нехай C – багатокутна фігура з границею ∂C , а S – її скелет. Загальна кількість точок, що складають множину S , є скінченою. Проте, виявляється, що всі ці точки лежать на кінцевій множині відрізків прямих ліній і квадратичних парабол. Нехай $s \in S$ – точка скелета, а g_1 і g_2 – дві найближчі до неї граничні точки фігури, $g_1, g_2 \in \partial C$. Точки g_1 і g_2 можуть займати різне положення на границі фігури. Кожна з них може бути належати багатокутнику або просто лежати на одній з його сторін. Будемо називати такі точки відповідно кутовими або простими. Можливі три варіанти поєднання типів точок g_1 і g_2 : пара кутових, пара простих або пара з кутових і простий точок.

У разі, коли обидві точки g_1 і g_2 кутові, скелетна точка s , рівновіддалена від g_1 і g_2 , обов'язково повинна лежати на прямій, яка є серединним перпендикуляром для відрізка $[g_1, g_2]$ (рисунок 2.4).

Якщо обидві точки g_1 і g_2 прості, то точка s рівновіддалена від сторін багатокутників, що містять g_1 і g_2 . Це означає що, точка s лежить на бісектрисі кута, утвореного цими сторонами. Якщо ж сторони паралельні, то s лежить на прямій лінії, рівновіддаленою від цих сторін. Для спільності можна вважати,

що ця пряма також є бісектрисою кута, утвореного сторонами багатокутної фігури.

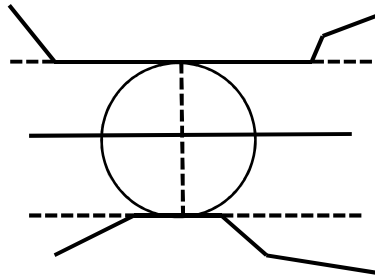


Рисунок 2.4 – Елементи скелетона багатокутної фігури

Нарешті, в разі, коли пара g_1, g_2 складається з кутової і простої точок, скелетна точка s рівновіддалена від кутової точки і від сторони багатокутника, що містить просту точку. Це означає, що s лежить на параболі, для якої фокусом є кутова точка, а директрисою – сторона багатокутника, на якій лежить точка g_2 .

Таким чином, всі скелетні точки повинні лежати або на бісектрисах кутів, утворених сторонами багатокутної фігури, або на прямих, рівновіддалених від пар вершин багатокутної фігури, або на параболі, що визначаються парами «вершина-сторона» багатокутної фігури. Оскільки число можливих пар «вершина-вершина», «сторона-сторона» і «вершина-сторона» кінцево, то і число ліній, що визначають гілки скелета, теж кінцево. З цього випливає що, завдання обчислення скелета для багатокутної фігури зводиться до комбінаторному перебору різних сполучень сторін і вершин фігури і обчисленню відповідних сегментів прямих ліній і парабол, що утворюють серединні осі.

«Жадібний» алгоритм побудови діаграми Вороного. Якщо багатокутна фігура має n вершин, то вона має n сторін і з них утворюється $2n$ сайтів. Виходить $4n^2$ пар сайтів, для яких принципово можуть існувати бісектори. Але серед цієї кількості пар сайтів, насправді, суміжні пари складають лише

невелику частину $O(n)$. Тому виникає задача відбору суміжних пар сайтів з множини всіх можливих пар. Вирішивши завдання відбору таких пар і обчисливши їх бісектори, можна побудувати діаграму Вороного для множини сайтів багатокутної фігури.

Для побудови скелетонів бінарних зображень в практичних додатках потрібні більш ефективні алгоритми. Розглянемо такий алгоритм, заснований на послідовному знаходженні всіх вершин і ребер діаграми Вороного. Оскільки діаграма Вороного багатокутної фігури є зв'язним графом, цей процес може бути реалізований шляхом послідовного приєднання до вже побудованої частини діаграми нових її елементів – ребер і вершин.

Починається цей процес з виявлення всіх термінальних вершин діаграми Вороного, що мають степінь 1. Такими є всі опуклі вершини багатокутної фігури. Дійсно, з кжною опуклою вершиною пов'язана пара сайтів-сегментів, утворених сторонами багатокутної фігури, інцидентними до цієї вершини. А бісектриса кута, утвореного такою парою сегментів, завжди задає бісектор, для якого вершина фігури є кінцевою точкою. Цей бісектор є відрізком прямої. Таким чином, для всіх вершин діаграми першого степеня існують інцидентні їм бісектори, які ініціалізують процес побудови діаграми (рисунк). Будемо називати такі бісектори термінальними.

Вершини бісекторів будемо шукати послідовно: спочатку знаходимо першу кінцеву точку бісектора, а потім другу. Бісектор, для якого обчислена лише одна кінцева точка, будемо називати активним. А ті бісектори, у яких відомі обидві кінцеві точки, будемо називати готовими. В процесі побудови діаграми Вороного створюються два списки: список активних бісекторів і список готових бісекторів.

Робота алгоритму починається із знаходження першої кінцевої крапки одного з термінальних бісекторів. Такою точкою є будь-яка з опуклих вершин багатокутної фігури (рисунк 2.5). Очевидно, що така вершина завжди існує (і не одна). Перший знайдений бісектор є активним. Він міститься в списку активних бісекторів, після чого починає працювати основний цикл алгоритму.

Основний цикл включає в себе обробку чергового активного бісектора, в результаті якої для нього знаходиться друга кінцева точка, після чого він переводиться в список готових бісекторів.

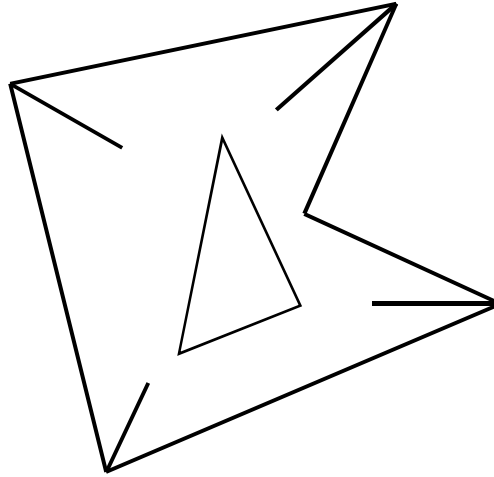


Рисунок 2.5 – Термінальні бісектори

Знайдена друга кінцева точка бісектора є або термінальною вершиною діаграми Вороного, або початковою точкою для інших бісекторів. В цьому останньому випадку породжуються нові бісектори, які поповнюють список активних. Цикл завершується, коли список активних бісекторів порожній.

Таким чином, основний алгоритм побудови діаграми Вороного полягає в послідовному породженні і обробці бісекторів і включає в себе такі кроки.

1. Вибрати черговий бісектор зі списку активних бісекторів і перейти до кроку 2. Якщо цей список порожній, то завершити роботу.

2. Побудова другої кінцевої точки бісектора. Для обраного активного бісектора знайти другу кінцеву точку. Знайдений бісектор розглядається в якості кандидата для включення в діаграму Вороного. Перейти до кроку 3.

3. Поповнення діаграми новим бісекторів. Отриманий новий бісектор потрібно перевірити на збіг з раніше побудованими бісекторами, що знаходяться в списку готових. Якщо збіги не знайдено, то новий бісектор поміщається в список готових, і далі перехід до кроку 4. Якщо ж бісектор-

кандидат збігся з якимось з побудованих раніше готових бісекторів, то він просто знищується і далі потрібно перейти до кроку 1.

4. Породження активних бісекторів. Знайдена друга кінцева точка бісектора оголошується вершиною діаграми Вороного. Якщо вона не збігається з вершиною багатокутної фігури, тобто не є термінальною вершиною діаграми Вороного, то з неї повинні виходити інші ребра діаграми. Це означає, що потрібно створити нові активні бісектори, для яких ця вершина є першою кінцевою точкою, і помістити їх в список активних бісекторів. Далі перейти до кроку 1.

Розглянемо кроки алгоритму більш детально.

1. Побудова другої кінцевої точки бісектора. Активний бісектор описується парою своїх визначальних сайтів і першою кінцевою точкою. Крім того, відома орієнтація бісектора щодо сайтів, тобто відомо, який із визначальних сайтів лежить зліва від бісектора, а який праворуч. Шукана друга кінцева точка є центром порожнього кола, дотичні до двох визначальним сайтам і деякого третього невідомого сайту. Отже, побудова цієї другої кінцевої точки зводиться до пошуку відповідного третього сайту. Такий пошук можна здійснити послідовним переглядом всіх сайтів і вибором серед них потрібного. Для кожного пробного сайту і пари визначають сайтів робиться спроба побудови загального дотичного кола. Причому шукається таке коло, яке стосується трьох сайтів у визначеному порядку. Точки дотику лівого, правого визначених сайтів і пробного сайту повинні утворювати праву трійку. Якщо таке коло побудувати вдалося, то її центр може розглядатися як потенційна друга кінцева точка бісектора.

Приклад на рисунку 2.6 ілюструє деякі окремі випадки, які можуть виникнути при вирішенні цього завдання. Тут точки дотику сайтів сегментів позначені тими ж буквами, що й самі сайти, але мають нижні індекси. У прикладах (а, б) для визначальних сайтів L (лівий) і R (правий) і пробного сайту T існують дві дотичні кола. Але орієнтація точок дотику для цих кіл різна. У прикладі (а) трійка точок (L, R_1, T) є правою, а трійка (L, R_2, T) –

лівою. Тому в якості дотичного кола вибираємо C_1 . А у випадку (б) трійка (L_1, R_1, T) є правою, а трійка (L_2, R_2, T) – лівою. Тут теж шуканої дотичної кола буде C_1 . Для випадку (в) існує єдина каса-точок кола C_1 в трійці сайтів L, R і T . При цьому трійка точок дотику (L, R_1, T_1) є правою. Отже, коло C_1 задовольняє необхідним умовам. А в разі (г) єдина дотична кола до сайтів L, R і T , яку можна побудувати, не може бути обрана, оскільки трійка точок дотику (L_1, R_1, T_1) не є правою.

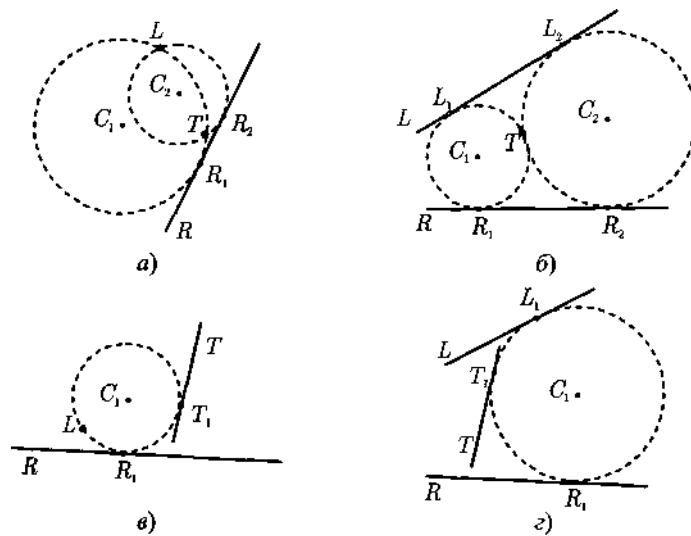


Рисунок 2.6 – Побудова дотичного кола

Бісектор будується для пари сайтів. А його кінцева точка виділяється як центр порожнього кола, дотичного до цієї пари сайтів і деякого третього сайту, який потрібно знайти. Однак в ході перегляду всіх сайтів може бути виявлено кілька таких кандидатів в «треті» сайти, для яких вдається побудувати порожні дотичні кола (рисунки 2.7, 2.8).

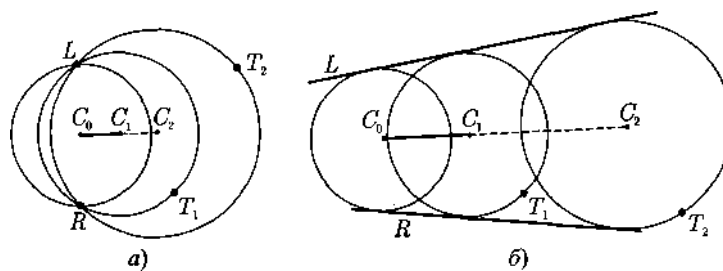


Рисунок 2.7 – Вибір кінцевої точки для лінійного бісектора

Такий вибір повинен здійснюватися з таких міркувань. Всі центри побудованих кіл лежать на лінії, рівновіддаленою від двох визначальних сайтів. Тому в якості другої кінцевої точки бісектора слід взяти центр того кола, який знаходяться ближче уздовж цієї лінії до першої кінцевої точки. У разі, коли пара визначених сайтів складається з двох сайтів-точок або двох сайтів-сегментів, вибирається просто та точка, яка ближче до першої кінцевої. На рисунку 2.8 перша кінцева точка бісектора позначена C_0 . Поміж двох точок C_1 і C_2 , що є кандидатами для вибору другої кінцевої точки бісектора, потрібно вибрати C_1 оскільки вона лежить ближче до C_0 , ніж C_2 . В результаті бісектором стає відрізок прямої C_0C_1 .

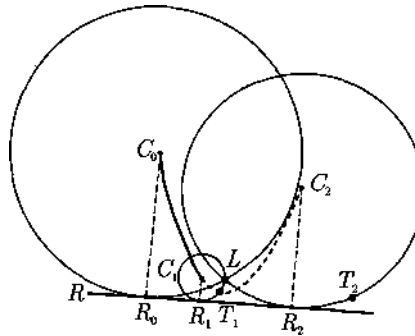


Рисунок 2.8 – Вибір кінцевої точки для параболічного бісектора

Більш складним є випадок, коли пара визначають сайтів складається з сайту-точки та сайту-сегмента. У цьому випадку в якості другої кінцевої крапки вибирається та, у якої проекція на сайт-сегмент лежить ближче інших до проекції першої кінцевої точки на цей сайт-сегмент. У прикладі на рисунку в якості пробних сайтів виступають сайти T_1 і T_2 . Центри відповідних дотичних кіл C_1 і C_2 лежать на параболі. При цьому відстань від першої кінцевої крапки бісектора C_0 до C_2 може бути меншою, ніж до C_1 . Однак, якщо розглянути проекції точок C_0 , C_1 і C_2 на сайт-сегмент R (точки R_0 , R_1 і R_2 відповідно), то з'ясується, що довжина відрізка R_0R_1 менша, ніж R_0R_2 . Тому в якості другої кінцевої точки бісектора вибирається C_1 і сам бісектор буде є сегментом параболи C_0C_1 .

Поповнення діаграми новим бісектором. Отриманий новий бісектор потрібно включити в список готових. Однак існує небезпека, що такий бісектор був уже побудований раніше і міститься в списку готових. Механізм можливої повторної побудови бісектора пояснюється наступним чином. Оскільки бісектор має дві кінцеві точки, які є вершинами діаграми Вороного, він може бути побудований з двох сторін. Все залежить від того, яка з його кінцевих точок було знайдено раніше. Тому цілком можливий випадок, коли в ході побудови діаграми можуть виникнути незалежно два активних бісектора, у яких першими кінцевими точками є дві суміжні вершини діаграми Вороного.

Для того щоб уникнути повторного включення в діаграму одного і того ж готового бісектора, необхідно здійснити перевірку знову отриманого бісектора на збіг його з яким-небудь готовим бісектором. Така перевірка здійснюється простим перебором списку готових бісекторів і порівняння їх з знову отриманими бісектором-кандидатом. У разі, коли збігу не відбулося, новий бісектор поповнює список готових. Якщо ж стався збіг, то бісектор-кандидат просто знищується.

Породження активних бісекторів. У діаграмі Вороного всі вершини мають степінь або 1, або 3 і більше. У разі, якщо знайдена друга кінцева точка бісектора має ступінь 1, це означає, що вона є термінальною вершиною діаграми. Термінальна вершина збігається з опуклою вершиною багатокутної фігури. Якщо ж точка не є термінальною, то в ній сходяться кілька бісекторів. Отже, знайдена друга кінцева точка бісектора може служити початковою точкою для породження нових активних бісекторів.

Породження нових бісекторів здійснюється за наступним правилом. Нехай побудований готовий бісектор має кінцеву точку C , що є центром кола, дотичній до сайтів L , R , T . Сайти L і R є відповідно лівим і правим визначаючими сайтами бісектора, а сайт T знайдений в результаті перегляду всіх пробних сайтів так, як це описано вище. Тоді здійснюється наступна перевірка. Якщо сайти L і R є сусідніми, то вершина C оголошується

термінальною. В іншому випадку породжуються два нових активних бісектора. Перший з них має визначальні сайти L і T , а другий – T і R . Лівим визначальним сайтом першого бісектора є L , а правим – T . Для другого бісектора лівим є T , а правим – R . Обидва бісектора мають одну і ту ж першу кінцеву точку C .

При пошуку другої кінцевої точки може виникнути особливий випадок, коли знайдеться кілька пробних сайтів $T_1, \dots, T_m, m > 1$, у яких центри дотичних кіл співпадуть. Це означає, що існує максимально порожнє коло, що стосується двох визначальних сайтів і всіх цих пробних сайтів. Отже, знайдена вершина діаграми Вороного є точкою, в якій сходяться осередку Вороного всіх цих сайтів. Ця ситуація є цілком допустимою і не порушує роботи алгоритму. Впорядкуємо сайти T_1, \dots, T_m в послідовності торкання їх порожнього кола від L до R . А потім утворюємо $m + 1$ нових бісекторів-кандидатів для наступних пар сайтів: L і T_1, T_1 і T_2, \dots, T_{m-1} і T_m, T_m і R .

Для ілюстрації роботи описаного алгоритму розглянемо в якості прикладу побудова діаграми Вороного багатокутної фігури, зображеної на рисунку 2.9, а.

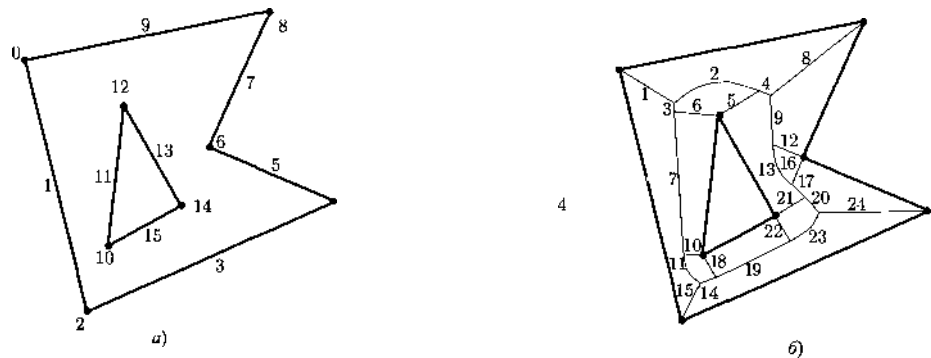


Рисунок 2.9 – Приклад роботи алгоритму побудови діаграми Вороного

Використовуються такі позначення. Запис (J, r) позначає бісектор з лівим визначальним сайтом J і правим r . Запис $\{i, j, k\}$ позначає вершину діаграми Вороного, що є центром дотичного кола сайтів i, j, k . При цьому

точки дотику утворюють праву трійку. Запис $[m]$ позначає термінальну вершину діаграми Вороного, збігається з сайтом-точкою m .

Якщо багатокутна фігура має n вершин, то загальне число сайтів, утворених з її вершин і сторін, становить $2n$. Тоді число осередків Вороного, що входять в розбиття Вороного, також не перевищує $2n$. Оскільки діаграма Вороного є плоский граф, для нього виконується теорема Ейлера, що зв'язує кількість вершин v , ребер e і граней f плоского графа наступним чином

$$v - e + f = 2.$$

Наслідком теореми Ейлера є асимптотична оцінка $O(n)$ для числа граней і ребер плоского графа [45]. Тоді число ребер в діаграмі Вороного є $O(n)$, а кожне ребро діаграми складається з одного бісектора, отже, загальна кількість бісекторів в діаграмі складе величину $O(n)$.

Відповідно до запропонованого алгоритму кожен бісектор в якийсь момент потрапляє в список активних, Причому завдяки контролю на збіг бісекторів повторне потрапляння бісектора в цей список неможливо. Таким чином, загальне число кроків в основному циклі алгоритму, яке визначається числом породжених активних бісекторів, є $O(n)$.

Оцінимо трудомісткість одного кроку основного циклу. Побудова бісектора включає в себе перебір всіх сайтів-кандидатів для побудови другої кінцевої точки бісектора, вимагає $O(n)$ перевірок. Кожна перевірка полягає в побудові дотичного кола для трьох сайтів, що займає час $O(1)$. Отже, побудова бісектора вимагає час $O(n)$. Далі побудований бісектор повинен бути перевірений збіг зі списком готових бісекторів. Перевірка здійснюється переглядом всього списку готових бісекторів, довжина якого становить $O(n)$, і порівняння з кожним бісектором з цього списку, яке вимагає часу $O(1)$. Таким чином, перевірка на збіг займає також часу $O(n)$. В результаті загальна оцінка часу виконання одного кроку основного циклу є $O(n)$. Враховуючи, що число кроків є теж $O(n)$, отримуємо загальну оцінку часу роботи алгоритму $O(n^2)$.

Описаний алгоритм будує послідовно всі ребра і вершини діаграми Вороного. Причому побудова кожного чергового елемента здійснюється на основі локально оптимального вибору. Такі алгоритми називаються жадібними. Перевагою алгоритму є його відносна простота з точки зору програмної реалізації. Однак «квадратична» обчислювальна складність $O(n^2)$ дозволяє застосовувати алгоритм тільки при вирішенні завдань невеликої розмірності. У тих практичних завданнях, в яких число-вершин багатокутної фігури становить $10^5 \div 10^6$, алгоритми квадратичної складності, як правило, неприйнятні. Для цих завдань потрібні більш ефективні алгоритми, що мають «субквадратичну» складність.

Рекурсивна декомпозиція діаграми Вороного. Для підвищення обчислювальної ефективності алгоритмів побудови діаграми Вороного зазвичай використовується алгоритмічна парадигма рекурсивної декомпозиції. Суть її полягає в тому, щоб розбити вихідну задачу з розмірністю входу n на дві підзадачі, які мають приблизно однакову розмірність близьку до $\frac{n}{2}$ отримати рішення цих підзадач, а потім з цих рішень сконструювати власне рішення вихідної задачі. При цьому кожна з двох підзадач, в свою чергу, вирішується точно таким же чином, шляхом розбиття на дві підзадачі. Таке рекурсивне розбиття підзадач триває до тих пір, поки розмірність підзадачі не стане настільки малою, що її рішення можна буде отримати безпосередньо, без подальшого розбиття. Якщо вдається за час $O(n)$ розбити задачу n розмірності на дві підзадачі з розмірностями $\frac{n}{2}$, а потім з рішень підзадач зібрати рішення вихідної завдання теж за час $O(n)$, то загальний час, який потрібен на весь процес рекурсивного рішення, складе величину $O(n \log n)$ [45].

Таким чином, основу ефективних алгоритмів рекурсивної декомпозиції для діаграм Вороного складають алгоритми розбиття вихідної задачі на підзадачі і об'єднання рішень підзадач за лінійний час. Розглянемо спочатку

застосування цього підходу до побудови звичайної діаграми Вороного кінцевого безлічі точок, а потім узагальнимо його на випадок діаграми Вороного багатокутної фігури.

Нехай на площині задано n сайтів-точок, для яких потрібно побудувати діаграму Вороного. Процес рекурсивного побудови включає в себе розбиття задачі на підзадачі, рекурсивне рішення підзадач і побудова спільного рішення на основі рішень підзадач. Цей останній крок будемо називати для простоти об'єднанням рішень або злиттям двох діаграм Вороного.

Розбиття вихідної множини точок зазвичай здійснюється за допомогою прямої лінії, що розділяє площину на дві півплощини, так щоб в одній півплощині перебувала половина всіх сайтів, а в іншій півплощині – ще половина (рисунок 2.10). Якщо загальна кількість точок непарна, то їх число в різних півплощинах має відрізнятись на 1. При такій розбивці опуклі оболонки підмножин (зображені в прикладі пунктиром) не перетинаються. Таким чином, вихідна множина сайтів розбивається на дві підмножини, званих розділеними. Як розділяє прямий найпростіше використовувати вертикальну лінію.

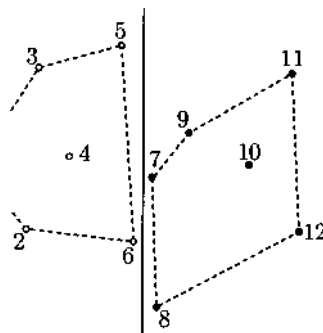


Рисунок 2.10 – Розбиття вихідної задачі на дві підзадачі

Нехай $V = \{v_1, \dots, v_n\}$ – вихідна множина з n сайтів-точок. Впорядкуємо лексикографічно множину V по зростанню і поміняємо індексацію сайтів так, щоб $v_1 < v_2 < \dots < v_n$. При цьому вважаємо, що для сайтів $v_i = (x_i, y_i)$ і $v_j = (x_j, y_j)$ має місце відношення порядку $v_i < v_j$, якщо їх координати задовольняють

умові $x_i < x_j$ або $x_i = x_j$ і $y_i < y_j$. Впорядковану таким чином множину V ділимо на дві частини із $\mu = \left\lfloor \frac{n}{2} \right\rfloor$ і $(n - \mu)$ сайтів: ліву $V_L = \{v_1, \dots, v_\mu\}$ і праву $V_R = \{v_{\mu+1}, \dots, v_n\}$. Рекурсивне розбиття підмножин точок триває доти, поки число точок в підмножині не стане рівним 1 або 2.

Діаграма Вороного для однієї або двох точок має простий вигляд і будується за фіксований час $O(1)$. Таким чином, кількість завдань нижнього рівня, що є основою рекурсії, тобто $O(n)$. В результаті вирішення цих завдань утворюється найнижчий «шар» діаграм Вороного. Далі, об'єднуючи попарно діаграми цього шару, ми отримуємо нові діаграми Вороного підмножин з 3-4 точок, які складають наступний шар діаграм. Потім будемо наступний шар об'єднаних діаграм. Процес об'єднання триває до тих пір, поки не утвориться одна діаграма для всього безлічі сайтів. Кількість шарів одержуваних діаграм складе в результаті величину $O(\log n)$.

Центральну роль в цьому підході відіграє алгоритм злиття двох побудованих рекурсивно діаграм Вороного $DV(V_L)$ і $DV(V_R)$ розділених підмножин $V_L = \{v_1, \dots, v_\mu\}$ і $V_R = \{v_{\mu+1}, \dots, v_n\}$ (рисунок 2.11). Злиття діаграм здійснюється шляхом побудови ламаної лінії, що складається з точок, рівновіддалених від підмножин сайтів V_L і V_R :

$$L = \{s: d(s, V_L) = d(s, V_R)\}.$$

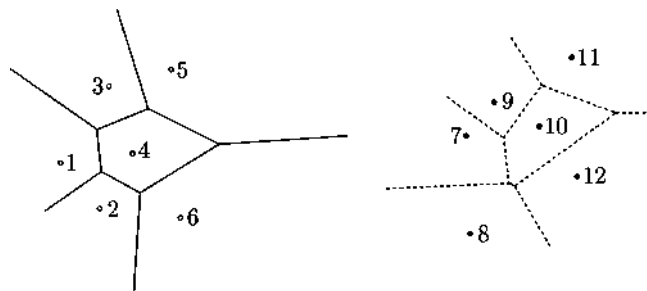


Рисунок 2.11 – Діаграми Вороного двох розділених підмножин сайтів

Ця лінія розділяє площину на два півпростору F_L і F_R , які умовно будемо називати лівим і правим:

$$F_L = \{s: d(s, V_L) \leq d(s, V_R)\},$$

$$F_R = \{s: d(s, V_L) \geq d(s, V_R)\}.$$

У лівому напівпросторі лежить підмножина сайтів V_L , в правому – V_R . Тоді об'єднана діаграма Вороного (рисунок 2.12) складається з самої розділової лінії L , а також з частин діаграм Вороного $DV(V_L)$ і $DV(V_R)$, що лежать відповідно в півпросторі F_L і F_R .

$$DV(V_L \cup V_R) = [DV(V_L) \cap F_L] \cup [DV(V_R) \cap F_R] \cup L.$$

Таким чином, завдання злиття двох діаграм Вороного $DV(V_L)$ і $DV(V_R)$ зводиться до побудови лінії L і видалення «зайвих» частин цих діаграм. Видаляються при цьому ребра, або фрагменти ребер діаграм, які лежать в «чужому» півпросторі: частина діаграми $DV(V_R)$, яка виявилася зліва від лінії L , і частина діаграми $DV(V_L)$, що лежить праворуч від цієї лінії (рисунок 2.12б).

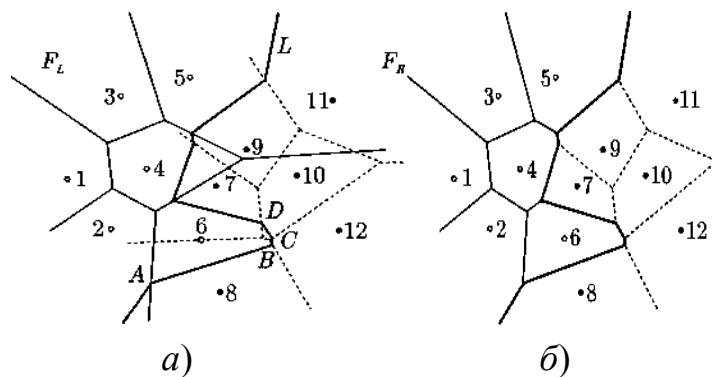


Рисунок 2.12 – Розділова лінія діаграм Вороного підмножин (а), об'єднана діаграма Вороного підмножин (б)

Розділяюча лінія є ламаною, складеною з двох кінцевих променів і кінцевого числа лінійних сегментів. Побудова розділяє лінії здійснюється наступним чином. Спочатку потрібно знайти один з променів. Для цього будуються опуклі оболонки $CH(V_L)$ і $CH(V_R)$ для підмножин V_L і V_R , що представляють собою непересічні опуклі багатокутники (рисунок 2.10), а потім шукаються спільні опорні прямі для цих багатокутників. Всього таких опорних прямих дві.

Відрізки опорних прямих є ребрами опуклої оболонки $CH(V)$. А шукані бісектори-промені розділяє лінії L – це серединні перпендикуляри цих ребер. Згідно розташуванню точок перетину опорних прямих з розділяє вертикальної прямої одна з опорних прямих називається нижній, а інша - верхньої.

Після цього, починаючи з одного зі знайдених променів, зазвичай з нижнього, здійснюється послідовний пошук всіх відрізків, що складає ламану лінію, до тих пір, поки не буде приєднаний другий (верхній) промінь. Початком кожного чергового відрізка вважається точка перетину попереднього відрізка або променя з ребрами діаграм $DV(V_L)$ і $DV(V_R)$. А напрямком відрізка вибирається як ортогональне до парі сайтів, в чиїх осередках Вороного лежить відрізок. Напрямок цього відрізка вибирається перпендикулярно до прямої, що проходить через сайти 6 і 8, оскільки він лежить в їх осередках Вороного. Точка B - це перша точка перетину з ребрами діаграм $DV(V_L)$ і $DV(V_R)$. Таким виявляється ребро осередку сайту 12. Тому наступний відрізок BC розділяє ламаної L перпендикулярний прямій, що з'єднує сайти 6 і 12. По завершенні цього процесу ми отримуємо розділену ламану лінію, після чого здійснюється відсікання частин діаграм $DV(V_L)$ і $DV(V_R)$, які не ввійдуть в об'єднану діаграму $DV(V)$.

Описаний алгоритм був запропонований в роботі [46] і став основою для розробки цілого сімейства ефективних алгоритмів в обчислювальній геометрії.

3 ПРОГРАМНА СИСТЕМА СКЕЛЕТИЗАЦІЇ ОБ'ЄКТІВ НА ЦИФРОВИХ ЗОБРАЖЕННЯХ

3.1 Структура програмної системи скелетизації об'єктів

Компонент Image дає відображення на формі графічного зображення. Властивість Picture типу TPicture містить відображається графічна складова у якій тип – бітова матриця, піктограми, метафайлу або певного користувачем типу. Властивість Canvas дозволяє створювати і редагувати зображення.

Під час проектування необхідно завантажити в властивість Picture графічний файл клацнувши на кнопці з трьома крапками близько властивості Picture у вікні Інспектора Об'єктів або при натисканні двічі на Image. Повинно відкритися вікно Picture Editor, яке дозволить завантажити в властивість Picture деякий графічний файл (через кнопку Load), і зберегти відкритий файл, давши йому нове ім'я або зберегти в новому каталозі.

Коли в процесі створення проекту зробити завантаження зображення з файлу в компонент Image, він не просто відобразить його, але і збереже завантажене зображення в додатку. Переваги становить в тому, що проект в цілому буде поставлятися без окремого графічного файлу.

При установці властивості AutoSize рівному true, то розмір компонента Image буде автоматом підігнаний під розмір поміщеної в image картинки. Якщо властивість AutoSize встановлено в значення false, то зображення може і не уміщатися в компонент або ж, навпаки, площа компонента може виявитися набагато більше площі зображення.

Інша властивість - Stretch дозволяє реалізувати підгонку HE компонента під розмір малюнка, а сам малюнок під параметр компонента. Але оскільки навряд чи реально насправді встановити розміри Image дуже точно пропорційними величині малюнка, то зображення буде спотворено. При установці Stretch рівне true може мати сенс тільки для якихось візерунків, але не для картинок. Властивість Stretch не діє на зображення піктограм, які не

змінюють своїх розмірів. Властивість - Center, який встановлений в true, робить по центру зображення на площі Image, якщо параметри розміру компонента більше параметра розмірів малюнка. Властивість - Transparent (прозорість). Якщо значення Transparent одно true, то зображення в Image буде прозорим. Це використовується при накладення зображень один на одного. Слід пам'ятати, що властивість Transparent поширюється тільки на бітові матриці. При цьому прозорим (тобто замінним на колір розташованого під ним зображення) робиться за замовчуванням колір лівого нижнього пікселя бітової матриці.

Властивість Picture дозволяє легко організувати обмін з графічними файлами будь-яких типів в процесі виконання програми. Це властивість - об'єкт, який має в свою чергу подсвойства, які вказують на зберігається графічний об'єкт. Якщо в Picture зберігається бітова матриця, на неї вказує властивість Picture.Bitmap. Якщо зберігається піктограма, на неї вказує властивість Picture.Icon. На зберігається метафайл вказує властивість Picture.Metafile. Нарешті, на графічний об'єкт довільного типу вказує властивість Picture.Graphic.

Об'єкт Picture і його властивості Bitmap, Icon, Metafile і Graphic мають методи файлового читання і запису LoadFromFile і SaveToFile. Для властивостей Picture.Bitmap, Picture.Icon і Picture.Metafile формат файлу повинен відповідати класу об'єкта: бітової матриці, піктограми, метафайлу. При читанні файлу в властивість Picture.Graphic файл повинен мати формат метафайлу. А для самого об'єкта Picture методи читання і запису автоматично підлаштовуються під тип файлу.

У цьому випадку користувач отримає можливість зберегти зображення будь-якого формату під новим ім'ям. Тільки при цьому, щоб не виникало в подальшому плутанини, розширення файлу все-таки має відповідати формату зберігається зображення. Абсолютно ідентична для зображень будь-якого формату буде працювати програма.

Для всіх розглянутих об'єктів Picture, Picture.Bitmap, Picture.Icon, і Picture.Metafile визначають методи привласнення значень об'єктів Assign. Однак для BitMap, Icon і Metafile привласнювати можна тільки значення однорідних об'єктів: відповідно бітових матриць, піктограм, метафайлів.

При спробі привласнення значень різнорідних об'єктів буде генеруватися виняток EConvertError. Об'єкт Picture - є універсальним, йому можна присвоїти значення об'єктів з інших трьох класів. А значення Picture можна привласнити тільки тому об'єкту, тип якого співпадає з типом об'єкта, що зберігається в ньому.

Завдяки наявності канви Canvas компонент Image широко використовується не тільки для зберігання готових зображень, але і для побудови різних графічних редакторів.

Менеджер TWAIN (DSM) - діє як координатор між додатками і Джерелом Даних (Data Source). DSM має мінімальний користувальницький інтерфейс - тільки вибір DS. Всі взаємодія з користувачем поза прикладної програми здійснюється за коштами DS. Кожен джерело даних розробляється безпосередньо виробником відповідних пристроїв. І їх підтримка стандарту TWAIN здійснюється на добровільних засадах.

DSM і DS це DLLs завантажуються в адресний простір програми і працюють як підпрограми додатки. DSM використовує межпроцессну зв'язок, що б координувати дії зі своїми копіями, коли більше ніж одна програма використовує TWAIN.

Спрощена схема роботи з технологією TWAIN:

- Відкрити діалог настройки відповідного пристрою (діалог відображає DS) і задати відповідні налаштування. Додаток очікує повідомлення від DS, що зображення готове. Під час очікування всі зареєстровані повідомлення будуть направлятися через TWAIN. Якщо це не буде виконуватися, то програма не отримає повідомлення про готовність зображення.

- Додаток приймає зображення від DS.

– TWAIN визначає три типи передачі зображення: Native - в Windows це DIB в пам'яті; Memory - як блоки пікселів в буферах пам'яті; File - DS записує зображення безпосередньо в файл (не обов'язково підтримується);

– Додаток закриває DS.

Після проведеного аналізу поставлених задач, та розробки алгоритмів обробки даних або спроектовано структуру програмної системи аналізу та опису цифрових зображень (рисунок 3.1).

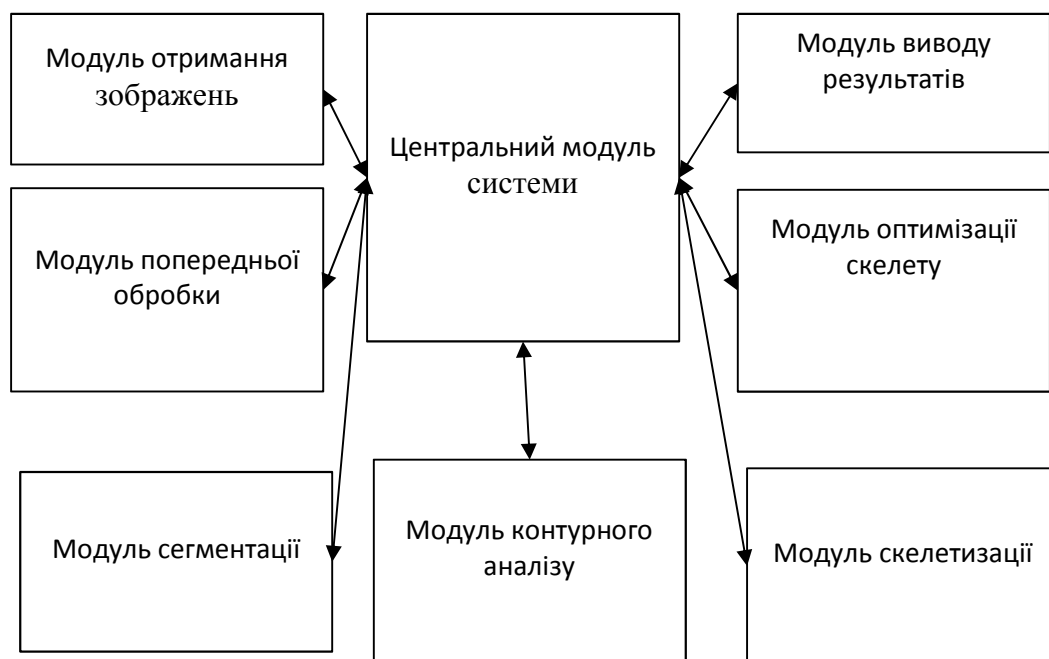


Рисунок 3.1 – Структура програмної системи скелетизації зображень

Розроблена структура програмної системи дозволяє в повній мірі забезпечити виконання поставлених завдань. Серед яких провести додаткову обробку вхідного зображення, процеси розбиття та виділення областей, опис форм даних областей та їх опис за допомогою скелетів.

Розглянемо основні характеристики найбільш важливих функціональних модулів системи.

«Центральний модуль системи» – даний модуль призначений для коректної організації взаємодії між окремими модулями системи. В ньому містяться засоби перевірки цілісності програмної системи, доступність

окремих модулів, актуальність версій та забезпечення передачі даних між окремими модулями під час роботи системи. Даний модуль є основним в системі.

«Модуль отримання зображень» – функції даного модуля забезпечують завантаження вхідних зображень та переконвертація їх у формати, що підтримуються програмною системою. Оскільки наперед не відомо який саме формат буде у вхідного зображення, то для запобігання критичних помилок в роботі системи необхідна система перевірок вхідного зображення. Якщо користувач має невеликий досвід роботи з програмами даного типу, то на вхід може бути подано зображення формат якого не підтримується, після цього програма спробує провести перекодування у зручний для неї формат, а при неможливості виконати дану дію, запропонує користувачеві змінити вхідне зображення.

«Модуль попередньої обробки» – для підвищення точності виконання процесу сегментації необхідно максимально зменшити вплив дефектів вхідного зображення. Саме для цієї мети в програмі передбачено модуль попередньої обробки, в який входять функції корекції яскравості та контрасту, функції виділення обремних областей на зображенні, для пришвидшення процесу аналізу, функції перекодування зображення з кольорових в бінарні або в градації сірого, для зменшення об'ємів пам'яті при зберіганні зображень.

«Модуль сегментації» – процес розбиття вхідного зображення на окремі одноріжні за деяким критерієм області наивається сегментація. Оскільки для опису зображення необхідно виділити окремі об'єкти, то для цього використовуються функції саме цього модуля. Серед алгоритмів, що реалізовані в даному модулі є: алгоритм порогової сегментації, водоподілу та нарощування областей. Дані алгоритми в повній мірі забезпечують можливість обробки та виокремлення областей для подальшого опису.

«Модуль контурного аналізу» – для проведення процедури скелетизації необхідно описати та закодувати форму області інтересу. Для цього використовуються засоби контурного аналізу, серед яких окремою групою

реалізовані алгоритми опису контуру шляхом його проходження. Вибір алгоритмів саме даної групи зводиться до того, що при своїй простоті та високій швидкодії вони можуть описувати форми овільної складності. Кодування контуру відбувається шляхом занесення виділених точок в структуру, що описує точку звязного контура як пару чисел, що відповідають координатам даної точки по осях абсцис та ординат на вхідному зображенні. Кодування в даному форматі оптимально підходить для подальшої обробки зображення запропонованим алгоритмом.

«Модуль скелетизації» – один з головних модулів системив якому реалізовані функції скелетизації. Для виділення скелету проводиться послідовний пошук точок, що є рівновіддалені від хоча б двох прямих за допомогою яких описано контур. Дана умова забезпечується якщо в вибрану область вписквати квадрати, оскільки в них сторони є однаковими, а отже якщо дві протилежні вершини квадрата лежить на контурі області, то точка претину діагоналей квадрата буде рівновіддаленою, а отже може бути визнана як точка скелету. Результатом роботи даного модуля є масив точок, що є рівновіддаленими від контуру області та формують скелет.

«Модуль оптимізації скелету» – Оскільки після завершення роботи алгоритмів скелетизації в програмі зберігаєть набір незв'язних точок, що відповідають скелету області, то необхідні додаткові операції для отримання моноскелета. Для цього проводиться пошук та з'єднання точок скелету, які розташовані поруч. Після чого проводиться спроба об'єднати окремі частини скелету в однорідну структуру. Після отримання моно скелету на ньому може бути присутя деяка кількість малоінформативних ребер. Ребра даного типу утворюються в результати похибок дискретизації, зашумленості контуру тощо. Для їх відсікання та подальшої роботи з більш інформативним скелетом відбувається процедура відновлення області на основі скелету. Причому, якщо відновлена без деякого ребара область відповідає з деякою похибкою хідній області, то таке ребро визнається малоінформативним та видвляється з структури скелету. Перевірку послідовно проводять для всіх кінцевих ребер скелету. В результаті рооти даного модуля отримується моноскелет, у якого відсічені малоінформативні ребра, а отже в подальшому це дозволить

зменшити об'єми пам'яті, що необхідні для зберігання опису скелету, а також підвищить час обробки даного скелету.

«Модуль виводу результатів» – для візуалізації отриманих результатів було спроектовано та реалізовано ряд функцій для формування звітів та виводу візуальних підтверджень роботоздатності програмної системи. Даний модуль є допоміжним в структурі програмного додатку, та виконує завдання в залежності від команд користувача та параметрів налаштування роти системою.

Для отримання комплексної оцінки про розроблену структуру програмної системи було проведено моделювання даного архітектури за допомогою UML діаграм, а саме послідовності та прецедентів.

Для дослідження системи взаємодії між окремими елементами системи та користувачем було проведено моделювання на основі діаграм прецедентів, яку наведено на рисунку 3.2.



Рисунок 3.2 – Діаграма прецедентів системи скелетизації зображень

Як видно з наведеної діаграми, група «Користувач» отримає доступ до різних функціональних параметрів роботи програмної системи, що в достатній мірі забезпечить якісну роботу системи та дозволить користувачам отримати потрібні результати при вирішенні поставлених задач.

Іншим важливим кроком в перевірці якості спроектованої програмної системи слід пропелювати послідовності кроків, які здійснить система при роботі з користувачем і чи там не відбудеться зациклення роботи системи. Результат моделювання наведено на рисунку 3.3.

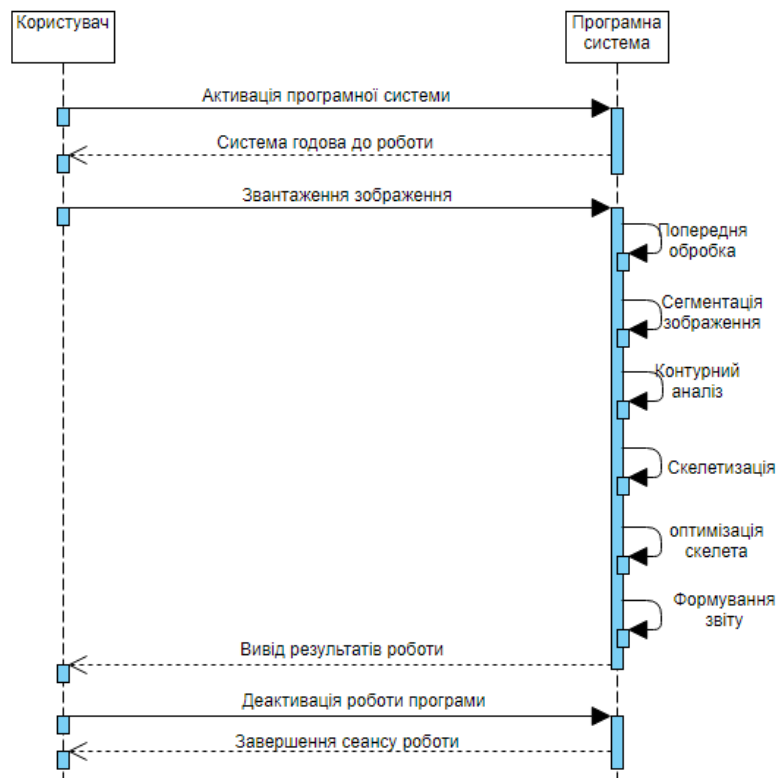


Рисунок 3.3 – Діаграма послідовності системи скелетизації зображень

Як показали результати моделювання програмної системи, всі процеси відбуваються послідовно, закінчення одного з процесів, автоматично активує початок іншого, а отже зависання роботи програмної системи неможливе.

Даний зовнішній вигляд програмної системи достатньо зручний та швидкий в опануванні, що є безперечною перевагою під час роботи з програмним продуктом.

3.2 Програмні модулі системи скелетизації об'єктів на цифрових зображеннях

Використовуючи класи з інтерфейсом, можна накласти фільтр на зображення або застосувати один з rgb ефектів, як збільшення яскравості. Успіх обробки також залежить від типу зображення. Для роботи програми було реалізовано декілька функцій, що забезпечать попередню обробку та підвищать якість вхідних зображень. Серед таких, що були реалізовані є ряд алгоритмів для проведення афінних перетворень, приклади програмного коду, для їх реалізації в розроблюній програмній системі наведено нище:

Функція роздіення вхідного кольорового зображення на окремі кольорові канали, при цьому на виході отримується зображення на якому присутній тільки червоний або синій або зелений колір (рисунок 3.4).

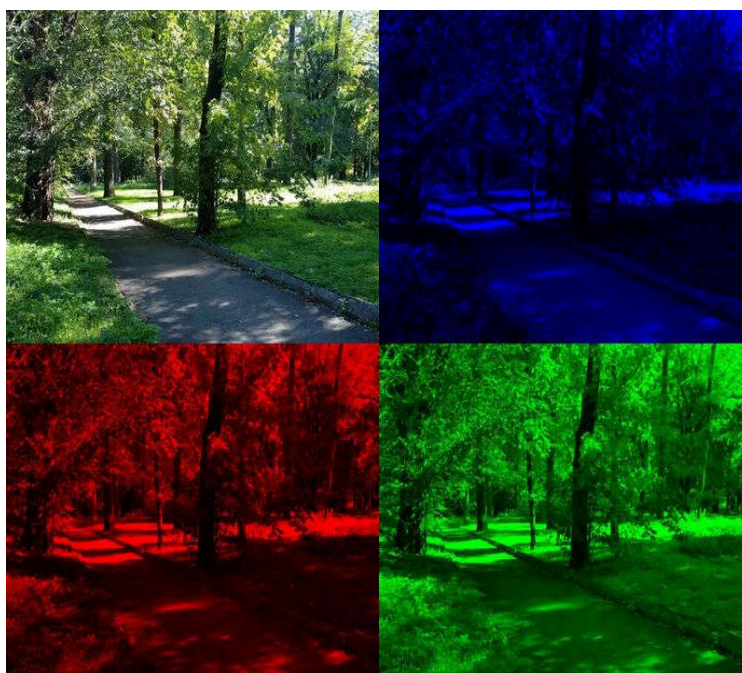


Рисунок 3.4 – Приклад розбиття вхідного зображення на окремі кольорові канали

Даний алгоритм використовується для спрощення процедури бінаризації та подальшого виділення об'єкту на зображенні, та зменшення часу на знаходження скелету зображення.

```
// Red
for (let i = 0; i < dst.length; i++) {
  let r = src[i] & 0xFF;
  dst[i] = 0xFF000000 | r;
}
// Green
for (let i = 0; i < dst.length; i++) {
  let g = (src[i] >> 8) & 0xFF;
  dst[i] = 0xFF000000 | (g << 8);
}
// Blue
for (let i = 0; i < dst.length; i++) {
  let b = (src[i] >> 16) & 0xFF;
  dst[i] = 0xFF000000 | (b << 16);
}
```

При цьому слід враховувати, що прозорість зберігається в старших бітах, тому будемо їх враховувати як 255.

Іншою додатковою функцією є реалізація алгоритма інвертування зображення. Для отримання даного ефекту необхідно інвертувати кожен з каналі, при цьому слід враховувати, що чорний колір (0,0,0) повинен завжди претворюватись в білий (255,255,255) і навпаки. Програмний код для реалізації даної функції наведено нижче:

```
for (let i = 0; i < dst.length; i++) {
  let r = src[i] & 0xFF;
  let g = (src[i] >> 8) & 0xFF;
  let b =
(src[i] >> 16) & 0xFF;
  let a = (src[i] >> 24) & 0xFF;
```

```

r = 255 - r;
g = 255 - g;
b = 255 - b;

dst[i] = (src[i] & 0xFF000000) | (b << 16) | (g << 8) | r;
}

```

RGB-корекція застосовується в разі, коли потрібно злегка підправити фотографію, якщо на ній занадто багато небажаного відтінку. Наприклад, в результаті неправильно виставленого балансу білого фотографія може мати синій або помаранчевий відтінок. Фільтр RGB-корекції реалізується дуже просто. Є три змінні з діапазоном від -255 до 255 включно, по одному на кожен колірну компоненту. Далі просто проходимо по масиву пікселів, розбиваємо піксель на три компоненти і додаємо відповідне значення. Важливо не забути обмежити отриманий результат діапазоном 0..255.

```

for (let i = 0; i < dst.length; i++) {
let r = src [i] & 0xFF;
let g = (src [i] >> 8) & 0xFF;
let b = (src [i] >> 16) & 0xFF;

r + = deltaR;
g + = deltaG;
b + = deltaB;
if (r > 255) r = 255;
else if (r < 0) r = 0;
if (g > 255) g = 255;
else if (g < 0) g = 0;
if (b > 255) b = 255;
else if (b < 0) b = 0;

dst [i] = (src [i] & 0xFF000000) | (B << 16) | (G << 8) | r;}

```

В результаті виконання даної функції можна відкоректувати вхідне зображення та привести його до більш зручного виду. Даний підхід оптимально підійде для зображень які при відсіканні деякого каналу

згладжують границь областей. Приклад корекції зображення наведено на рисунку 3.5.



Рисунок 3.5 – Приклад RGB-корекція вхідного зображення

Також додатковою функцією при роботі з вхідними зображеннями є алгоритми корекції яскравості зображення. Приклад реалізації даного лагоритму наведено нижче:

```
function processImage(inImg) {
  const width = inImg.width;
  const height = inImg.height;
  const src = new Uint32Array(inImg.data.buffer);

  // Brightness
  processCanvas('canvasBrightness', width, height, function(dst)
  {
    let delta = parseInt($("#rangeBrightness").val());
    for (let i = 0; i < dst.length; i++) {
      let r = src[i] & 0xFF;
      let g = (src[i] >> 8) & 0xFF;
      let b = (src[i] >> 16) & 0xFF;

      r += delta;
      g += delta;
      b += delta;
      if (r > 255) r = 255;
      else if (r < 0) r = 0;
    }
  });
}
```

```

    if (g > 255) g = 255;
    else if (g < 0) g = 0;
    if (b > 255) b = 255;
    else if (b < 0) b = 0;

    dst[i] = (src[i] & 0xFF000000) | (b << 16) | (g << 8) | r;
  }
});

```

Приклад використання даних функцій наведено на рисунку 3.6



Рисунок 3.6 – Приклад корекції яскравості вхідного зображення

Під час роботи програмного додатку, на вхід можуть поступати зображення в різних кольорових гаммах, через те в програмі реалізовано функцію корекції зображення в формат СМУК. На вхід подають білий папір накладаючи блакитний, пурпурний і жовтий, отримує необхідний колір. Для економії кольорової фарби використовується четвертий компонент К – інтенсивність чорного. Лістинг програмного коду даної функції наведено нижче:

```

function processImage(inImg) {
  const width = inImg.width;
  const height = inImg.height;
  const src = new Uint32Array(inImg.data.buffer);
  processCanvas('canvasCmyk', width, height, function(dst) {
    for (let i = 0; i < dst.length; i++) {
      let r = src[i] & 0xFF;

```

```

let g = (src[i] >> 8) & 0xFF;
let b = (src[i] >> 16) & 0xFF;
// RGB to CMYK
let k = 1 - (Math.max(r, g, b) / 255.0);
let c = (1 - (r / 255.0) - k) / (1 - k);
let m = (1 - (g / 255.0) - k) / (1 - k);
let y = (1 - (b / 255.0) - k) / (1 - k);
c += deltaC;
m += deltaM;
y += deltaY;
k += deltaK;
if (c > 1) c = 1;
else if (c < 0) c = 0;
if (m > 1) m = 1;
else if (m < 0) m = 0;
if (y > 1) y = 1;
else if (y < 0) y = 0;
if (k > 1) k = 1;
else if (k < 0) k = 0;
// CMYK to RGB
r = Math.floor(255 * (1 - c) * (1 - k));
g = Math.floor(255 * (1 - m) * (1 - k));
b = Math.floor(255 * (1 - y) * (1 - k));
dst[i] = (src[i] & 0xFF000000) | (b << 16) | (g << 8) | r;
} });
}

```

Приклад корекції зображення заданого да допомогою даної кольорової гама наведено на рисунку 3.7.



Рисунок 3.7 – Приклад корекції вхідного зображення

Якщо всі компоненти скинути до значення -100, то зображення перетвориться в бінарне. Приклад такого перекодування вхідного зображення наведено на рисунку 3.8

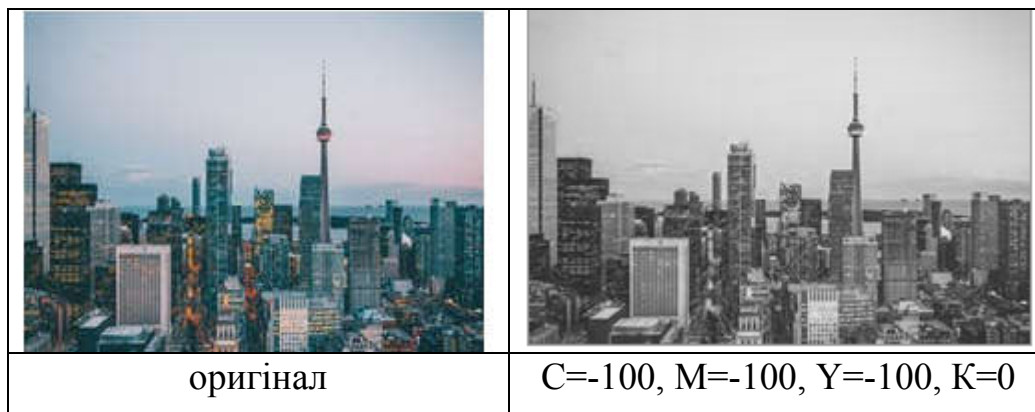


Рисунок 3.8 – Приклад корекції вхідного зображення

Іншою кольоровою системою, що часто використовується в програмах для обробки зображень є модель HSL. Модель HSL побудована на більш зрозумілому людині сприйняття кольорів і використовує такі складові: hue - відтінок, saturation - насиченість і lightness - світлота. Приклад реалізації алгоритму для перекодування зображення з моделі RGB в модель HSL наведено ниши. При цьому слід зазначити, що подібні перетворення будуть справедливими і для моделей HSV (або HSB) - hue, saturation, value (або brightness) - яскравість. Відрізняються вони тим, що максимальне значення третьої компоненти у HSL дає білий колір, а у HSB / HSV - яскравий колір

```
function rgbToHsl (r, g, b) {  
  let max = Math.max (r, g, b);  
  let min = Math.min (r, g, b);  
  let l = (max + min) / 2;  
  if (max == min) {  
    return [0, 0, l];  
  }  
  let h, s;  
  let d = max - min;  
  if (l > 0.5) {  
    s = d / (2 - max - min);  
  } Else {
```

```

s = d / (max + min);}
switch (max) {
case r:
h = (g - b) / d + (g <b? 6: 0);
break;
case g:
h = (b - r) / d + 2;
break;
case b:
h = (r - g) / d + 4;
break;}
h /= 6;
return [h, s, l];}
function hue2rgb (p, q, t) {
if (t <0) t += 1;
else if (t > 1) t -= 1;
if (t <1/6) return p + (q - p) * 6 * t;
if (t <1/2) return q;
if (t <2/3) return p + (q - p) * (2/3 - t) * 6;
return p;}
function hslToRgb (h, s, l) {
if (s == 0) {
return [l, l, l];}
let q = l <0.5? l * (1 + s): l + s - l * s;
let p = 2 * l - q;
let r = hue2rgb (p, q, h + 1/3);
let g = hue2rgb (p, q, h);
let b = hue2rgb (p, q, h - 1/3);
return [r, g, b];}

```

Приклади корекції вхідного цифрового зображення в моделі HSL наведено на рисунку 3.9. При цьому слід зазначити, що дане кольорова модел найбіль зрозуміла для людини, адже просто зрозуміти, коли необхідно зробити зображення світлішим або темнішим.

Дана реалізація окремих функції а також цілого класу в цілому дозволяли реалізувати на базі розроленого класу програмну систему обробки цифрових зображень та провести її тестування.

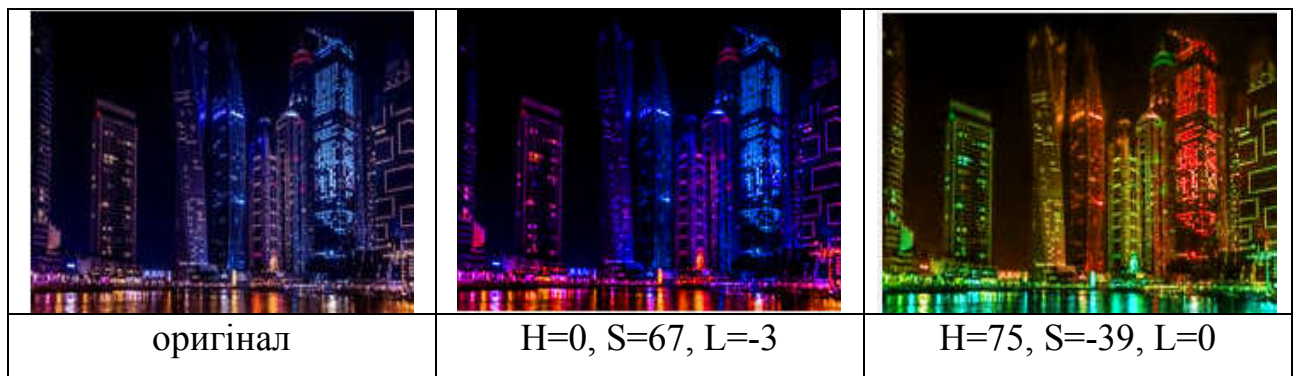


Рисунок 3.9 – Приклад корекції вхідного зображення

3.3 Тестування та аналіз реалізованої системи

Для тестування програмної системи обробки цифрових зображень було використано робочу станцію з такими характеристиками як таблиця 3.1:

Таблиця 3.1 – Технічні характеристики робочої станції

Параметр	Значення
корпус	Zaerwerlman Z1 Black + блок живлення Chieftec APS-43SB;
HDD	WD 500GB;
Відеокарта	Asus PH-GTX1060-3G;
ОЗП	4Gb 2400GHz ;
Процесор	AMD Ryzen 3 2200G BOX 120;
материнська плата	Asus B350M-E 90.

Характеристики монітора:

- дисплея 19";
- роздільна здатність 1920 x 1080;
- матриця TN;
- частота оновлення 65 Гц;
- інтерфейси HDMI;
- відношення сторін 16: 9.

Технічні характеристики обраної робочої станції є цілком достатніми для проведення процесу тестування розробленої програмної системи обробки цифрових зображень та отримання аналітичних даних про коректність роботи, швидкість аналізу та опису вхідних зображень.

При проведенні тестування було проведено ряд тестів, на основі трьох різних груп областей:

- правильні геометричні фігури – дані області мають форму близьку до простих геометричних фігур, а отже їх контурні лінії вже наближені до апроксимованих, що значно спроще процес отримання їх скелету;

- прості фігури – до областей даного типу було віднесено області, що мають в своїй контурній лінії елементи природніх об'єктів (деяку варіативність), що може повпливати на процес апроксимації та подальшої обробки зображення;

- складні фігури – області, що мають складну контурну лінію з великою кількістю змін напрямку та знаку її приросту. При апроксимації важливу роль відіграє коефіцієнт точності, чим він ближче до 0, тим точнішим буде опис контуру, проте зросте кількість апроксимуючих прямих.

Для початку роботи програмної системи необхідно встановити параметри роботи та параметри обробки зображень.

Після встановлення параметрів роботи системи, необхідно завантажити вхідне зображення, для цього необхідно активувати відповідний пункт меню, та обрати зображення, що потребує обробки (рисунок 3.10).

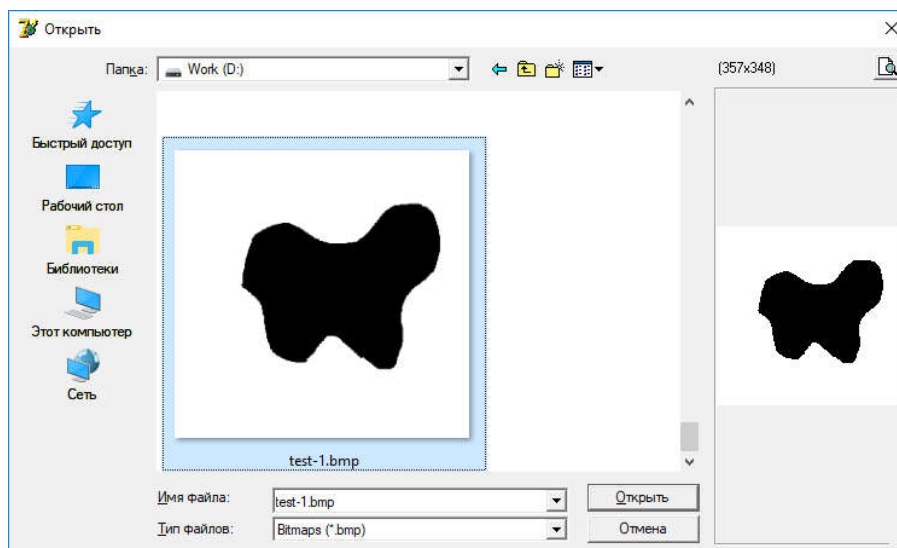


Рисунок 3.10 – Приклад обробки зображення

Після отримання зображення програмна система виконає перший етап обробки зображення, а саме проведе процедуру сегментації, виділення контуру та апроксимації виділеної області (рисунок 3.11).

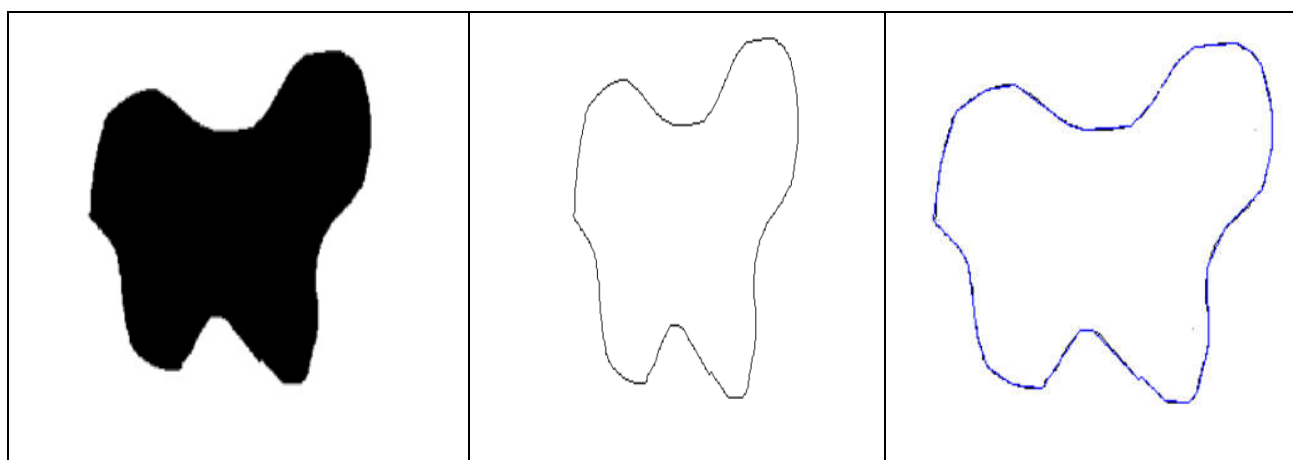


Рисунок 3.11 – Приклад виділення контуру вхідної області

На наступному етапі відбувається процес скелетизації виділеної області, проте на даному етапі скелет схожий на набір пікселів, тому для отримання якісного скелета необхідна додаткова обробка зображення (рисунок 3.12).

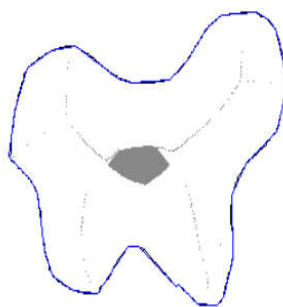


Рисунок 3.12 – Область з виділеними скелетними точками

На останньому етапі обробки вхідного зображення, програмна система проводить формування моно скелета та відсікання малоінформативних ребер скелету, в результаті роботи програмної ситеми було отримано осьтакий результат, що представлено на рисунку 3.13.

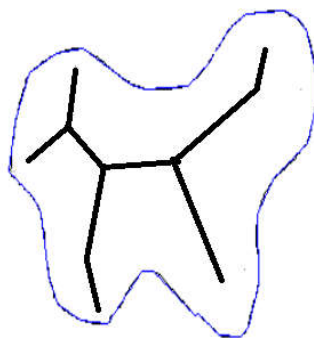


Рисунок 3.13 – Виділений скелет на вхідному зображенні

Також на результати скелетизації сильний вплив має поріг апроксимації, на рисунку 3.14 проілюстровано ситуації з некоректним порогом.

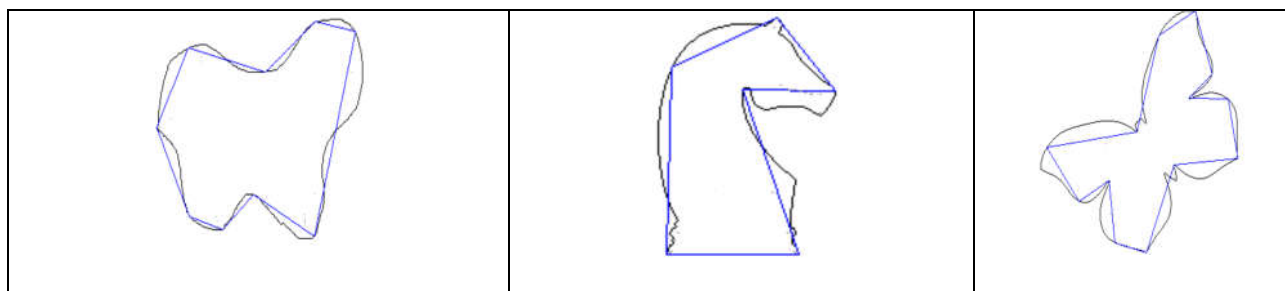


Рисунок 3.14 – Некоректно проведена апроксимація

Після проведення тестування, результати роботи програмної системи були погруповані в групи в залежності від того, який тип зображення було передано на вхід. Під час роботи оцінювався відсоток формування моноскелетів та можливість відновлення області на основі отриманого скелета. Результати тестування наведено в таблиці 3.2.

Таблиця 3.2 – Узагальнені результати роботи програмної системи

	Правильні геометричні фігури	Прості фігури	Складні фігури
Бінарні зображення	100%	>98%	>89%
Напівтонові зображення	100%	>90%	>78%
Кольорові зображення	>93%	>84%	>54%

Після отримання даних під час проведеного тестування можна зробити висновок, що на результат роботи значною мірою впливає тип вхідного зображення, оскільки від цього залежить результат процедури сегментації, що в свою чергу має вплив на подальші результати роботи програмної системи. При умові, якщо отримані якісно виділено область з чітким контуром, то результати скелетизації будуть на високому рівні.

ВИСНОВКИ

В результаті виконання магістерської роботи отримано такі результати:

1. Проаналізовано біомедичні зображення, виділено їх особливості.
2. Проведений аналіз сучасних систем автоматизованої мікроскопії показав, що ці системи мають у своєму складі основні алгоритми опрацювання зображень.
3. Розроблено алгоритми скелетонізації полігонів, що дозволило ефективно описувати області зображень в різних метричних просторах;
4. Спроектовано структуру програмної системи скелетизації мікрооб'єктів, що дозволило розробити програмні модулі системи скелетизації об'єктів на цифрових зображеннях.
5. Проведено тестування розроблених алгоритмів скеле тонізації на прикладі біомедичних зображень і зроблено оцінку похибок скеле тонізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Національний Канцер Реєстр [Electronic resource]: <http://unci.org.ua/spetsialistam/nacionalnij-kancer-reyestr/>.
2. Вакуліч О.В., Димінський В.І., Макогон І.М., Манзик С. В. Опрацювання біомедичних зображень на основі алгоритмів комп'ютерного зору. Матеріали III всеукраїнської науково-практичної конференції «Прикладна геометрія та інформаційні технології в моделюванні об'єктів, явищ і процесів» (AGIT-2018), м. Миколаїв, 17–19 жовтня 2018 р. – Миколаїв: МНУ імені В.О. Сухомлинського, 2018. – С. 174-175.
3. Дериш Б. Б., Манзик С. В., Попов В.О., Кравець В. Б. Аналіз хеш – алгоритмів порівняння зображень: II Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Частина 2. 14 листопада 2019 р. Тернопіль, 2019. С. 10-11.
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. Гонзалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2006.
7. Местецкий Л. М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. М.: ФИЗМАТЛИТ, 2009. 288 с.
8. Фурман Я. А., Юрьев А. Н., Яншин В. В. Цифровые методы обработки и распознавания бинарных изображений. Красноярск: Изд-во Красноярского ун-та, 1992.

9. Russ J. C. The image processing handbook. CRC Press, 1999.
10. Blum H. A transformation for extracting new descriptors of shape. In Proc. Symposium Models for the perception of speech and visual form. MIT Press, Cambridge, MA, 1967.
11. Klein R., Lingas A. Fast skeleton construction. In Proc. 3rd Europ. Symposium on Alg. (ESA'95), 1995.
12. Lee D. Medial axis transformation of a planar shape. IEEE Trans. Pat. Anal. Mach. Int. PAMI-4(4). 1982. P.363-369.
13. Castelman K. R. Digital image processing. Prentice hall, 1996.
14. Costa L., Cesar R. Shape analysis and classification. CRC Press, 2001.
15. Иванов Д., Кузьмин Е. Эффективный алгоритм построения остова растрового изображения. Тр. межд. конф. «Графикон-98». М.: МГУ, 1998. С. 65-70.
16. Ogniewicz R., Kubler O. Hierarchic Voronoi Skeletons. Pattern Recognition. 1995. Vol. 28, №3. P. 343-359.
17. Montanari U. Continuous skeletons from digitalized image. Journal ACM. 1969. Vol. 16, №4. P. 534-549.
18. Sklansky J., Chazin R. L., Hansen B. J. Minimal-perimeter polygons of digitized silhouettes. IEEE transactions on computers. 1972. Vol. C-21, №3. P. 260-268.
19. Bai X., Latecki L. J., Liu W. Y. Skeleton pruning by contour partitioning with discrete curve evolution. IEEE transactions on pattern analysis and machine intelligence. March 2007. Vol. 29, №3.
20. Kankaanpää P. BioImageXD: an open, general-purpose and high-throughput image-processing platform / P. Kankaanpää, L. Paavolainen, S. Tiitta / Nat Methods. – 2012. – Vol. 9(7). – P. 683-692 .
21. Малов А. Компьютерная обработка биомедицинских многоканальных изображений с использованием визуализации меры сходства с эталоном (ДиаМорф) / А. М. Малов, Т. А. Шереметьева, Г. Н. Филиппов // Известия ВУЗов. Сер, Приборостроение. -СПб.:Издание Санкт-

Петербургского государственного института точной механики. - № 52(8). – 2009. – С. 74-49.

22. Séroussi B. ONCODOC: a successful experiment of computer-supported guideline development and implementation in the treatment of breast cancer / B. Séroussi, J. Bouaud, E. Antoine // *Artif Intell Med.* – 2001. – Vol. 22(1). – P. 43-64.

23. Vrekoussis T. Image analysis of breast cancer immunohistochemistry-stained sections using ImageJ: an RGB-based model / T. Vrekoussis, V. Chaniotis, I. Navrozoglou, V. Dousias // *Anticancer Res.* Vol. 29(12). – 2009. – P. 4995-5003.

24. Березький О.М. Системи автоматизованої мікроскопії: стан та перспективи розвитку / О.М. Березький, С.О. Вербовий, О.Й. Піцун // *Вісник Хмельницького національного університету.* – 2016. – №2 (235). – С. 61-68.

25. Bronikov A.V. Kombinirovaniy algoritmy filtratsii zashymlenych signalov I izobrazheniy / A.V. Bronikov, Yu.E. Voskoboynikov // *Avtometriya.* – 1990. – Vol 1.

26. Wang X.-Y. A pixel-based color image segmentation using support vector machine and fuzzy C-means / X.-Y. Wang, X.-J. Zhang, X.-Y. Yang, J. Bu // *Neural Networks.* – 2012. – Vol. 33. – P. 148-159.

27. Rahman Z.U. Multi-scale retinex for color image enhancement / Z.U. Rahman, D.J. Jobson, G. A. Woodell // *Proc. IEEE Int. Conf. on Image Processing.* – Vol. 3. – 1996. – p. 1003-1006.

28. Yang X.S. Cuckoo search via Levy flights. / X.S. Yang, S. Deb // *Proceedings of the world congress on nature and biologically inspired computing (NaBIC).* – 2009. – P. 210-214.

29. Медовый В. С. Информационные автоматизированные системы микроскопии для анализа биоматериалов // *Врач и информационные технологии.* – 2004. – № 6. – С.32-37.

30. Кузнецова Л.А. Анализ проблематики использования изображений в цифровом формате в медицинской практике / Л.А. Кузнецова, М.А. Мельник, К.А. Катилова // *Системи обробки інформації.* – 2016. – № 4. – С. 169–171.

31. Королюк И.П. Медицинская информатика: Учебник / И.П. Королюк. – 2 изд., перераб. и доп. – Самара: ООО «Офорт»: ГБОУ ВПО «СамГМУ». 2012. – 244 с; ил.
32. Дороничева А.В. Методы Распознавания Медицинских Изображений Для Задач Компьютерной Автоматизированной Диагностики / А.В. Дороничева, С.З. Савин // Современные проблемы науки и образования. – 2014. – № 4.
33. Федьків С. В. Магнітно–резонансна томографія в кардіології: інформаційно–методичний посібник / С. В. Федьків ; під ред. : В. М. Коваленка, В. М. Корнацького. – К., 2013. – 60 с.
34. Рыбакова М. К. Практическое руководство по ультразвуковой диагностике, Эхокардиография / М.К. Рыбакова, М. Н. Алехин, В. В. Митьков, // Издательский дом Видар, Москва, 2008, 512 с.
35. Барінов Е. Ф. Гістологія, цитологія та ембріологія. Кн.1: Цитологія і загальна ембріологія: навч. посіб. для студ. вищ. мед. навч. закл. IV рівня акредитації / Е. Ф. Барінов, Ю. Б. Чайковський. 2010. – 216 с.
36. Мищенко А.С., Фоменко А.Т. Краткий курс дифференциальной геометрии и топологии. М.: Физматлит, 2004. 304 с.
37. Blum H. A Transformation for Extracting New Descriptors of Shape / H. Blum // Models for the Perception of Speech and Visual Form. – USA: MIT Press, 1967. – P. 362–380.
38. 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., Николіук В.Д., Піцун О.Й. Тернопіль: Економічна думка, ТНЕУ, 2017. 330 с.
39. Косневски Ч. Начальный курс алгебраической топологии. – М.: Мир, 1983. 304 с.
40. Дуда Р., Харт П. Распознавание образов и анализ сцен. М.: Мир, 1976. 511 с.

41. Березский О. Н. Топологические методы и алгоритмы преобразования контуров и областей плоских изображений. Проблемы управления и информатики. 2010. № 5. С. 123–131.

42. Березький О.М. Аналіз метрик знаходження відстані між областями зображень для кількісної оцінки результатів сегментації / О.М. Березький, Г.М. Мельник, Ю.М. Батько, О.Й. Піцун // ISDMCI'2016. XII міжнародна наукова конференція. Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту 24-28 травня 2016р. – Херсон: Видавництво ПП Вишемирський В.С., 2016. – С. 252-253

43. Березький О.М. Segmentation algorithms of biomedical images: development and quantitative evaluation / О.М. Березький, Ю.М. Батько, Г.М. Мельник, С.О. Вербовий, О.Й. Піцун // Штучний інтелект. – Київ. – 2016. – №3 (73). – С. 104-116

44. Berezsky O. M. Regions Matching Algorithms Analysis to Quantify the Image Segmentation Results / О.М. Berezsky, G. M. Melnyk, Y. M. Batko, O.Y. Pitsun // Sensors & Transducers. – 2017. – Vol. 208(1). – pp. 44-49.

45. Препарата Ф., Шеймос М. Вычислительная геометрия: введение. М.: Мир, 1989.

46. Shamos M. I., Hoey D. Closest point problem. Proc. 16th Annual IEEE symposium on foundations of computer science. 1975. P. 151-162.