

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

**Дослідження методів векторизації зображень**

---

---

---

Освітній ступінь “Магістр”  
Спеціальність 122 “Комп’ютерні науки”

Науковий керівник роботи:

\_\_\_\_\_

(підпис)

О.І.Рязанцев

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

Я.О.Критська

\_\_\_\_\_

(ініціали, прізвище)

Студент:

\_\_\_\_\_

(підпис)

О.Є. Відіборова

\_\_\_\_\_

(ініціали, прізвище)

Група:

КН-17зм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній ступінь магістр

Напрямок підготовки \_\_\_\_\_

(шифр і назва)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_

І.С. Скарга-Бандурова

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Відіборовій Оксані Євгенівні

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів векторизації зображень

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.

(прізвище, м. 'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 221/48

2. Строк подання студентом роботи 10.01.2018

3. Вихідні дані до роботи Матеріали науково-дослідної практики, тестові зображення, теоретичні відомості про способи векторизації

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Сучасний стан питання векторизації багатокольорового растру та постановка задачі, математичні методи, які застосовуються при вирішенні задачі векторизації багатокольорового растру, алгоритми та програмна реалізація хмарного сервісу векторизації зображень, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2018

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Розробка технічного завдання	10.09.2018-15.09.2018	
2	Підбір та вивчення технічної літератури за темою роботи	16.09.2018-22.09.2018	
3	Вибір та розробка методу	26.09.2018-06.10.2018	
4	Проведення аналітичних досліджень та розрахунків	07.10.2018-25.11.2018	
5	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2018-1.12.2018	
6	Оформлення пояснювальної записки, автореферату та презентації	2.12.2018-09.01.2018	
7			

Студент

\_\_\_\_\_ ( підпис )

Відіборова О.С.

\_\_\_\_\_ (прізвище та ініціали)

Науковий керівник

\_\_\_\_\_ ( підпис )

Рязанцев О.І.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Відіборова О.Є. Дослідження методів векторизації зображень.

Робота присвячена вирішенню проблеми розробки методу векторизації багатокольорового растру для хмарного сервісу векторизації зображень. Розроблено алгоритм векторизації з використанням триангуляції Делоне та математичної морфології. Дипломний проект виконаний із застосуванням мови програмування С# у середовищі програмування Visual Studio .NET. Метою даної дипломної роботи є набуття практичних навичок векторизації зображень.

**Ключові слова:** розпізнавання образів, векторизація, триангуляція делоне, математична морфологія, хмарний сервіс

## ABSTRACT

Vidioborova O.E. Research of methods of image vectorization.

This work is devoted to solving the problem of developing a method of color images vectorization that used in cloud image vectorization service. The vectorization method bases on Delaunay triangulation and mathematical morphology. The degree project is executed with the C# programming language, in programming environment Visual Studio. NET. The aim of this thesis is the acquisition of practical skills development of color image vectorization.

**Keywords:** recognition, vectorization, delaunay triangulation, mathematical morphology, cloud services.

## АННОТАЦИЯ

Видиборова А.Е. Исследование методов векторизации изображений.

Работа посвящена решению проблемы разработки метода векторизации цветного растра для облачного сервиса векторизации изображений. Разработано алгоритм векторизации с использованием триангуляции Делоне и математической морфологии. Дипломный проект выполнен с применением языка программирования С# в среде программирования Visual Studio .NET. Целью данной дипломной работы является получение практических навыков векторизации изображений.

**Ключевые слова:** распознавание образов, векторизация, триангуляция делоне, математическая морфология, облачный сервис.

## ЗМІСТ

ВСТУП.....	6
1 СУЧАСНИЙ СТАН ПИТАННЯ ВЕКТОРИЗАЦІЇ БАГАТОКОЛЬОРОВОГО РАСТРУ ТА ПОСТАНОВКА ЗАДАЧІ .....	8
1.1 Існуючі системи векторизації багатокольорового растру .....	8
1.1.1 Векторизація за допомогою графічного пакету Adobe Illustrator.....	9
1.1.2 Векторизація за допомогою графічного пакету Corel Draw .....	11
1.1.3 Векторизація для геоінформаційних систем.....	13
1.2 Постановка завдання.....	14
2 МАТЕМАТИЧНІ МЕТОДИ, ЯКІ ЗАСТОСОВУЮТЬСЯ ПРИ ВИРІШЕННІ ЗАДАЧІ ВЕКТОРИЗАЦІЇ БАГАТОКОЛЬОРОВОГО РАСТРУ .....	16
2.1 Тріангуляція .....	16
2.1.1 Жадібний алгоритм побудування тріангуляції .....	17
2.1.2 Тріангуляція Делоне .....	18
2.2 Математична морфологія .....	24
2.2.1 НМ – перетворення (Hit or Miss), операції ерозії та делатації .....	25
2.2.2 Розкриття та закриття .....	29
2.2.3 Тоншення та товщення .....	29
3 АЛГОРИТМИ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ХМАРНОГО СЕРВІСУ ВЕКТОРИЗАЦІЇ ЗОБРАЖЕНЬ .....	32
3.1 Архітектура системи .....	32
3.1.1 Сервісно-орієнтована архітектура .....	32
3.1.2 Модель SaaS .....	33
3.1.3 Концепція MVC та шаблони проектування .....	34
3.2 Обґрунтування вибору середовища розробки та зовнішніх бібліотек.....	38
3.3 Алгоритм роботи програмного засобу .....	40
3.4 Результати роботи програмного засобу .....	40
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	43
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об'єкту, що мають вплив на персонал .....	43
4.2 Заходи щодо техніки безпеки .....	44
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці .....	47
4.4 Рекомендації по пожежній безпеці.....	50

	5
4.5 Вплив на навколишнє середовища.....	52
ВИСНОВКИ .....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	56
ДОДАТОК А. Електронні плакати.....	58

## ВСТУП

Зазвичай у повсякденному житті ми маємо справу з растровими зображеннями: фотографії, зроблені на цифровий фотоапарат або зіскановані, зображення, що завантажені з мережі Інтернет – усі вони складаються з пікселів. Принципова відмінність векторних зображень – це те, що вони складаються з об'єктів, що задані математичними формулами та є стійкими до перетворень.

Векторна графіка застосовується у різноманітних галузях людської діяльності: від графічного дизайну до космічних проєктів. Можливість точно задавати координати та виконувати точні геометричні побудовання зробило векторну графіку дуже зручною для виконання креслень, проєктування мікросхем, використання у ландшафтному дизайні, архітектурі та для конструкторській документації узагалі.

Можливість вільної трансформації – це найбільша перевага векторних зображень. Такі зображення можна обертати і масштабувати практично без втрати його якості. Тому векторна графіка незамінна, коли важливо зберегти ясність і чіткість контурів, наприклад при створенні графічних заголовків, логотипів, малюнків, технічних ілюстрацій, схем, діаграм та якщо зображення планується обробляти на специфічному обладнанні (плоттери та ін.). Також трасування застосовується для отримання деяких спеціальних ефектів на зображеннях. Таким чином можна створити ефект масляного живопису, гуаші, акварелі або рисунку, що виконаний олівцем. Але у той самий час векторна графіка не може передати переходи тонів та полу тонів, тому векторні зображення виглядають нереалістичними.

Також важливою перевагою є розмір файлів. Векторна графіка дуже економна в плані використання обсягів дискового простору, необхідного для зберігання зображень. Це пов'язано з тим, що зберігаються тільки деякі основні дані, зокрема, координати опорних і керуючих точок, використовуючи які програма щоразу заново відтворює зображення. Крім того, опис колірних характеристик не сильно збільшує розмір файлу, оскільки дані про колір ідентичні для всього об'єкта, на відміну від растрового формату, коли ми зберігаємо колірні характеристики кожного пікселя.

Основною задачею векторизації є створення серії векторних контурів, які в цілому збігаються за формою з реальними об'єктами вхідного растрового зображення, з якими надалі можна виконувати будь-які перетворення, що застосовуються для векторних контурів. Векторний графічний об'єкт складається з контуру та його внутрішньої області, яка може бути або пустою, або мати заливку у вигляді одного кольору чи градієнту.

Контур може бути як замкнутим, так і розімкненим, саме він задає форму графічного об'єкта.

Незважаючи на те, що в даний час графічні пакети включають в себе функцію векторизації, або трасування, найбільш якісною залишається ручна векторизація. До цих пір в світі не існує єдиного алгоритму для трасування кольорових зображень. Це пов'язано, в основному з індивідуальними особливостями різних форматів. До того ж, векторизація багатобарвних зображень – це досить-таки трудомісткий процес, складність якого безпосередньо залежить від кількості кольорів растра.

Для векторизації мап використовуються окремі програми і алгоритми, що перетворюють знімки місцевості, що зроблені за допомогою супутників або з літака, на цифрові мапи. Такі мапи потім можна завантажувати у GPS-пристрої.

В рамках даної роботи розробляється хмарний web-сервіс, який буде надавати кінцевому користувачеві функціональність по векторизації растрових зображень. Система організована за принципом SaaS (Software as a service). У цій бізнес-моделі продажу й використання програмного забезпечення доступ до сервісів здійснюється через мережу Інтернет. Архітектура системи побудована з використанням SOA (Service-oriented architecture). Таким чином, система буде доступна для користування великій кількості клієнтів. У розпорядження користувача будуть надані великі обчислювальні потужності, незалежно від технічних характеристик пристрою, яким він користується, також і від характеристик операційної системи, встановленої на пристрої.

Користувач матиме змогу самостійно корегувати деякі параметри векторизації, які неможливо обчислити абсолютно автоматично. Завдяки цьому система може забезпечити високу якість результуючих зображень.



# 1 СУЧАСНИЙ СТАН ПИТАННЯ ВЕКТОРИЗАЦІЇ БАГАТОКОЛЬОРОВОГО РАСТРУ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Існуючі системи векторизації багатокольорового растру

Серед найбільш поширених у наш час систем, що надають можливість векторизації багатокольорового растру найбільш відомими є графічні пакети Adobe Illustrator, Corel Draw. Найпоширенішою програмою для креслень є AutoCAD.

На сьогоднішній день існують наступні формати векторних графічних файлів:

.ai – це формат графічного пакету Adobe Illustrator, відрізняється найбільшою стабільністю і сумісністю з PostScript пристроями.

.cdr – це формат графічного редактору Corel Draw, не дуже сумісний з іншими графічними редакторами; у форматі застосовується компресія для векторів і растра окремо, можуть впроваджуватися шрифти; файли мають величезне робоче поле 45x45 метрів (цей параметр важливий для зовнішньої реклами).

.ctm (Corel Presentation Exchange) – це формат, призначений для передавання зображень між різними програмами, розроблений корпорацією Corel.

.eps – це універсальний формат, який підтримує більшість графічних редакторів (але з обмеженнями щодо версії формату); забезпечує досить високу якість зображень, але застосовувати його має сенс лише тоді, коли вивід буде здійснюватися на PostScript пристрої.

.fla, .fh – формат редактору Adobe Flash.

.svg (Scalable Vector Graphics) – це формат, що базується на XML розмітці та підтримується багатьма браузерами; не призначений для складної графіки та має обмеження у використанні.

.swf – Flash формат, який підтримується Flash Player.

.wmf (Windows Media Format) – універсальний графічний формат файлу в системі Microsoft Windows, який підтримує більшість векторних редакторів; не забезпечує високу якість для складних малюнків; підтримується рядом веб-браузерів і може бути використаний при оформленні веб-сторінок.

### 1.1.1 Векторизація за допомогою графічного пакету Adobe Illustrator

Векторизувати можна будь-яке зображення, але від якості вхідних даних буде залежати і результат трасування. Тому зазвичай треба виконати попередню обробку зображення у програмі Adobe Photoshop: підвищити різкість та контрастність, освітлити занадто темні фрагменти, прибрати шум, підвищити чіткість. Також особливу увагу треба звернути на зображення, що містять текст. Найкраще буде видалити текст, а потім вже заново увести після векторизації.

У графічному пакеті Adobe Illustrator існують три способи векторизації зображень: ручна векторизація, вбудоване автоматичне трасування та автоматичне трасування за допомогою спеціалізованих програм. Вибір способу трасування залежить від складності вхідного зображення, вимог до якості результату трасування та інших факторів.

Зазвичай, автоматична векторизація застосовується до зображень зі суцільною заливкою та дуже чіткими контурами. Якщо на вхідному зображенні присутні градієнти або інші складні заливки, то буде сформована дуже велика кількість контурів – кожний контур буде відповідати однорідно залитим ділянкам на зображенні. Такий результат не може нас задовольнити, тому що головна ідея трасування – це отримання векторних контурів самих зображених об'єктів, а не лише однорідно забарвлених їх частин. Також, файл з такою кількістю контурів може мати набагато більший розмір, ніж файл відповідного растрового зображення. Таким чином, найбільш вдалим зазвичай є трасування зображень, які схожі на векторні, а найгірше виходить трасування фотографій.

На рисунку 1.1 зображено результат автоматичного трасування фотографії. У вікні Layers відображається велика кількість контурів, що була сформована.

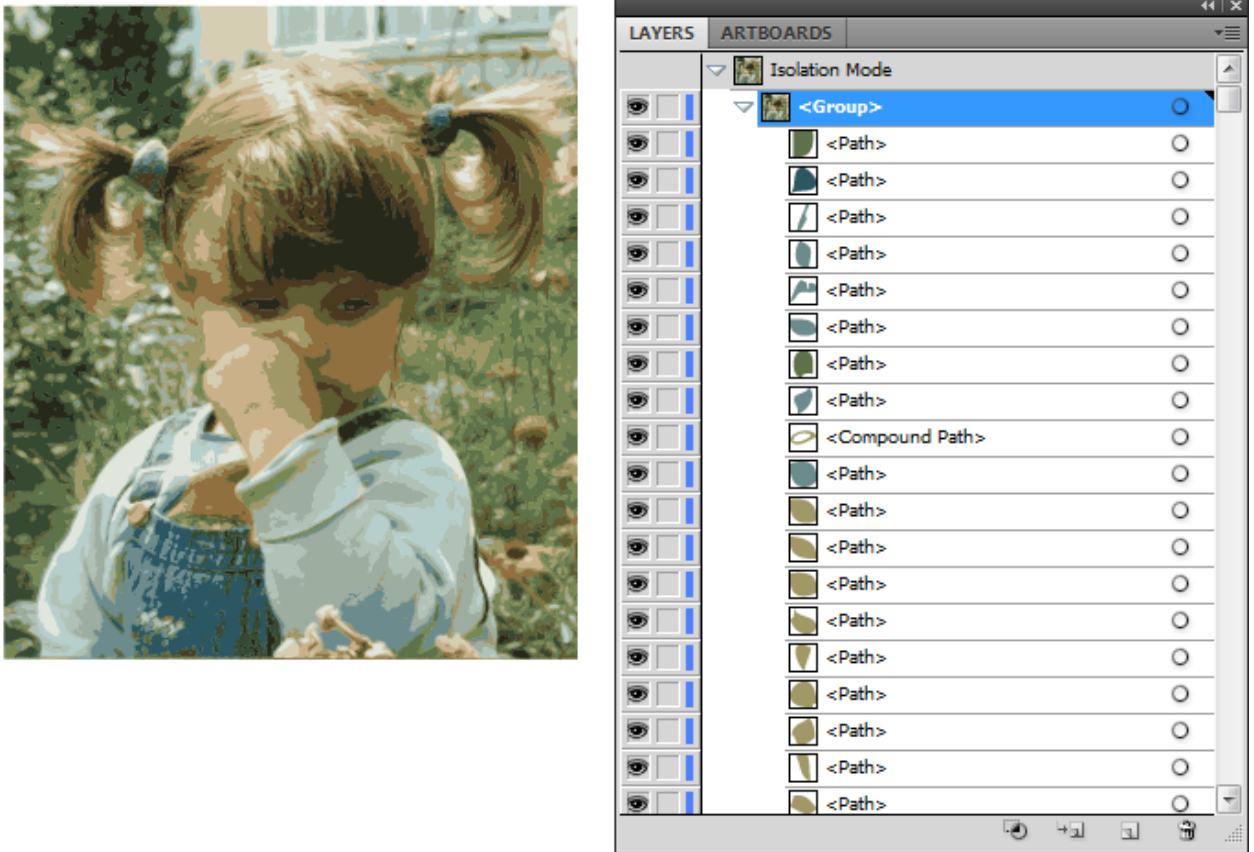


Рисунок 1.1 – Результат застосування функції автоматичного трасування до фотографії

Ручне трасування – досить складна процедура, тому що, фактично, треба перемалювати растровий малюнок за допомогою інструментів Pen (Перо) та Pensil (Олівець). Використовуючи растрове зображення, як шаблон, потрібно створити нові векторні контури. Зазвичай, щоб підготувати зображення до ручної векторизації, спочатку його обробляють за допомогою графічного пакету Adobe Photoshop. Окрім світокоррекції та видалення шумів, застосовують функцію Posterize, тобто виконують пастеризацію зображення. Пастеризація – це зменшення кількості відтінків одного кольору. Чим вище рівень, тим складніше буде виконувати ручне трасування, тому у загальному випадку рекомендований рівень 15-20. На рисунку 1.2а подано вхідне зображення, до якого була застосована функція Posterize а на рисунку 1.2б – результат ручного трасування з використанням пастеризованого зображення.



а



б

Рисунок 1.2 – Результат ручного трасування у Adobe Illustrator: а – результат застосування функції пастеризації, б – результат ручного трасування

### 1.1.2 Векторизація за допомогою графічного пакету Corel Draw

В CorelDRAW можна виконувати трасування растрових зображень, тобто перетворювати їх у векторну графіку з можливістю попереднього перегляду і налаштування результатів трасування.

При трасуванні користувач має можливість вибору стилю заготовки. Стиль заготовки являє собою набір параметрів, відповідних певному типу растрового зображення, трасування якого необхідно виконати: штриховий рисунок, кольорове зображення, фотографія.

У результатах трасування можна виконати згладжування кривих ліній і настройку кількості деталей. Від згладжування залежить кількість вузлів в результатах трасування, а від деталей - кількість об'єктів і квітів. Вхідна в CorelDRAW утиліта PowerTRACE надає відомості про кількість вузлів, об'єктах і квітах в результатах трасування. Ця інформація оновлюється при кожній зміні налаштування параметрів.

За замовчуванням після трасування вихідне растрове зображення зберігається, а об'єкти в результатах трасування автоматично групуються. Можна також вказати, щоб вихідне растрове зображення автоматично віддалялося після завершення трасування.

У процесі векторизації растрових зображень можна виконувати повторну трасування растрових зображень стільки разів, скільки необхідно для досягнення бажаного результату. Якщо була допущена помилка, дія можна скасувати або повернути, а також відновити перші результати трасування.

Для автоматичної векторизації растрових зображень у складі CorelDRAW включається існує утиліта PowerTRACE, яка дозволяє за вихідними зображень у форматі точкової графіки автоматично будувати векторні зображення у форматі CMX, який в сімействі графічних програм фірми Corel використовується для обміну графічною інформацією. Для переходу до автоматичної векторизації імпортованого точкового зображення слід виділити його інструментом Pick (Вибір) і вибрати команду Bitmaps (Точкові зображення) | Trace Bitmap (Трасувати точкове зображення) | Quick Trace (Швидка векторизація).

При трасуванні найбільш часто вживаються два алгоритми векторизації: трасування по контурах і трасування по осьовій лінії.

При векторизації по контурах векторизованное зображення складається з об'єктів областей з приблизно однаковою забарвленням - цей колір стає заливкою об'єкта.

Векторизація по осьовій лінії виконується дещо іншим способом. Програма векторизації для кожного знайденого нею ділянки точкового зображення з однаковою заливкою намагається за задане число ітерацій знайти осьову лінію. Сукупність таких осьових ліній товщиною в один піксель і становить результат векторизації. Цей метод практично придатний тільки для штрихових зображень.



Рисунок 1.3 – Вхідне зображення і результат його автоматичного трасування за допомогою Corel Draw

### 1.1.3 Векторизація для геоінформаційних систем

Геоінформаційна система (географічна інформаційна система, ГІС) - система збору, зберігання, аналізу та графічної візуалізації просторових (географічних) даних і пов'язаної з ними інформації про необхідні об'єктах.

Поняття геоінформаційної системи також використовується в більш вузькому сенсі - як інструмента (програмного продукту), що дозволяє користувачам шукати, аналізувати і редагувати як цифрову карту місцевості, так і додаткову інформацію про об'єкти .

Геоінформаційна система може включати до свого складу просторові бази даних (у тому числі, під керуванням універсальних СУБД), редактори растрової та векторної графіки, різні засоби просторового аналізу даних. Застосовуються в картографії, геології, метеорології, землеустрій, екології, муніципальному управлінні, транспорті, економіці, обороні та багатьох інших областях. Наукові, технічні, технологічні та прикладні аспекти проектування, створення та використання геоінформаційних систем вивчаються геоінформатики.

За територіальним охопленням геоінформаційні системи підрозділяють на глобальні (англ. Global), субконтинентальна, національні, часто мають статус державних,

регіональні (regional), субрегіональні, локальні, або місцеві (local). У деяких випадках, такі територіальні ГІС можуть бути розміщені у відкритому доступі в мережі Інтернет і називаються геопорталів.

За предметної області інформаційного моделювання виділяються міські (муніципальні) (urban GIS), недропользовательські, гірничо-геологічні, природоохоронні (environmental) і т. п.; серед них особливе найменування, як особливо широко поширені, отримали земельні інформаційні системи.

Також геоінформаційні системи можуть бути класифіковані за проблемної орієнтації - вирішуванним науковим і прикладним завданням. Такими завданнями можуть бути інвентаризація ресурсів (у тому числі кадастр), аналіз, оцінка, моніторинг, управління і планування, підтримка прийняття рішень. Крім того, інтегровані геоінформаційні системи поєднують функціональні можливості і систем цифрової обробки зображень (даних дистанційного зондування) в єдиному інтегрованому середовищі.

Розрізняють також:

- полімасштабні, або масштабно-незалежні геоінформаційні системи (multiscale), засновані на множинних, або полімасштабних уявленнях просторових об'єктів, забезпечуючи графічне або картографічне відтворення даних на будь-якому з обраних рівнів масштабного ряду на основі єдиного набору даних з найбільшим просторовим дозволом;

- просторово-часові геоінформаційні системи (spatio-temporal), які оперують просторово-часовими даними.

## **1.2 Постановка завдання**

В основу векторної графіки покладено математичне подання властивостей геометричних фігур. Відображення векторної графіки на екрані – це процес вираховування на екрані точок, що належать до контуру зображення, тобто зображення побудоване координатним способом.

Базовими поняттями у векторній комп'ютерній графіці є лінія та об'єкт. Для математичного подання лінії використовуються точка, пряма, парабола, криві другого та третього порядків, криві Без'є. Об'єктом називають будь-який графічний елемент, що

належить векторному зображенню та містить відрізок прямої або кривої лінії або замкнутого контуру.

Головна проблема сучасних векторизаторів – отримання зображення з великою кількістю контурів у випадку застосування алгоритму векторизації до зображень фотографічної якості.

Для вирішення задачі векторизації був вибраний метод векторизації за допомогою триангуляції Делоне. Заради досягнення кращого розпізнання об'єктів на зображенні спершу до зображення буде застосовано операції математичної морфології, що дозволить видалити занадто маленькі області. Для прибирання шумів на зображенні було використано медіанну фільтрацію.

Щоб уникнути утворення великої кількості контурів, спочатку аналізується кількість кольорів на зображенні, та, якщо кількість відтінків перевищує певний поріг, то до зображення застосовується пастеризація з метою зменшення кількості відтінків одного кольору.



## 2 МАТЕМАТИЧНІ МЕТОДИ, ЯКІ ЗАСТОСОВУЮТЬСЯ ПРИ ВИРІШЕННІ ЗАДАЧІ ВЕКТОРИЗАЦІЇ БАГАТОКОЛЬОРОВОГО РАСТРУ

Задача векторизації зазвичай вирішується для випадку двокольорового (бінарного) растру. При застосуванні відомих методів для багатокольорового зображення значно росте трудомісткість, тому для векторизації кольорового растрового зображення необхідно побудувати комплексну векторну модель. Для цього потрібно виділити граничні лінії об'єктів та виконати апроксимацію знайдених ліній прямими відрізками та кривими Без'є. Виділення об'єктів на зображенні можна реалізувати за допомогою тріангуляції та математичної морфології.

### 2.1 Тріангуляція

Тріангуляція – це розбиття геометричного об'єкта на симплекси, тобто на геометричні фігури, які є  $n$ -мірним узагальненням трикутника.

Як відомо, через будь-які  $n$  точок можна провести  $(n-1)$ -мірну площину. Також існують множини з  $n+1$  точок, через які  $(n-1)$ -площину провести не можна. Таким чином  $n+1$  — мінімальна кількість точок в  $n$ -просторі, які не лежать в одній  $(n-1)$ -площині, і можуть бути вершинами  $n$ -багатогранника, тобто,  $n$ -симплекс являє собою з'єднання з  $n+1$  точок [3]. Отже, симплекс – це простий  $n$ -багатогранник з кількістю вершин  $n+1$ .

Приклади симплексів у просторах найменших розмірностей:

- точка (0-симплекс);
- відрізок (1-симплекс);
- трикутник (2-симплекс);
- тетраedr (3-симплекс).

Загальні властивості симплексів:

- кількість вершин симплекса на одиницю більша розмірності простору;
- існує загальне правило перетворення фігур нижчої розмірності у фігури вищої розмірності. Воно полягає в тому, що з геометричного центра фігури

будується перпендикуляр в наступний вимір, на цьому перпендикулярі будується нова вершина і з'єднується ребрами зі всіма вершинами початкового симплексу [3];

- будь-яка вершина симплекса сполучена ребрами зі всіма іншими вершинами.

Триангуляція  $T$  простору  $R^{n+1}$  – це розбиття  $R^{n+1}$  на  $(n + 1)$ -мірні симплекси, такі що:

- будь-які два симплекса в  $T$  або мають загальне ребро чи вершину, або взагалі не перетинаються;

- будь-яка обмежена множина у  $R^{n+1}$  перетинає кінцеву кількість симплексів з  $T$ .

Триангуляція множини точок, тобто, триангуляція дискретної множини точок  $P \in R^{n+1}$  – це розбиття опуклої оболонки точок на симплекси так, що виконується перша умова з попереднього визначення, і безліч точок, які є вершинами симплексів розбиття, збігається з  $P$ .

Опуклою триангуляцією називається така триангуляція, для якої мінімальний багатокутник, що охоплює всі трикутники, буде опуклим.

Задача побудови триангуляції по заданому набору двовимірних точок – це задача з'єднання заданих точок непересічними відрізками так, щоб утворилася триангуляція. Трудомісткість цієї задачі становить  $O(N * \log N)$ . Існують алгоритми, що досягають цієї оцінки в середньому і найгіршому випадках. Крім того, відомі алгоритми дозволяють в деяких випадках досягти в середньому  $O(N)$  [4].

Завдання побудови триангуляції по вихідному набору точок є неоднозначною. Триангуляція називається оптимальною, якщо сума довжин всіх ребер мінімальна серед усіх можливих триангуляцій, побудованих на тих самих вихідних точках. Ця задача є NP-повною. Тому для більшості реальних задач існуючі алгоритми побудови оптимальної триангуляції неприйнятні через занадто високу трудомісткість. При необхідності на практиці застосовують наближені алгоритми.

### **2.1.1 Жадібний алгоритм побудування триангуляції**

Жадібний алгоритм – це алгоритм, що полягає у прийнятті локально оптимальних рішень на кожному етапі, припускаючи, що остаточний розв'язок теж буде оптимальним. Жадібний алгоритм побудування триангуляції був запропонований одним з перших. Це такий метод, у якому ніколи не відмінюються вже зроблені кроки.

В алгоритмі послідовно виконуються наступні кроки:

1) На першому кроці генерується список усіх можливих відрізків, що з'єднують пари вихідних точок.

2) Сформований список сортується по довжинах відрізків.

3) Послідовно виконується вставка відрізків в триангуляцію, починаючи з самого короткого. Якщо відрізок не перетинається з іншими, раніше вставленими відрізками, то він вставляється, інакше він відкидається. Якщо всі можливі відрізки мають різну довжину, то результат роботи цього алгоритму однозначний, інакше він залежить від порядку вставки відрізків однакової довжини.

У жадібному алгоритмі ніколи не відміняються вже зроблені кроки, бо полягає у прийнятті локально оптимальних рішень на кожному етапі, припускаючи, що остаточний розв'язок теж буде оптимальним. Трудомісткість роботи жадібного алгоритму при деяких його поліпшення становить  $O(N^2 * \log N)$ . У зв'язку з настільки великою трудомісткістю на практиці він майже не застосовується [4].

### 2.1.2 Триангуляція Делоне

Триангуляція Делоне є найбільш відомим видом триангуляції множини точок. Вперше задача побудови триангуляції Делоне була поставлена в 1934 в роботі радянського математика Б.Н. Делоне.

Триангуляція Делоне для множини точок  $P$  на площині – це така опукла триангуляція  $DT(P)$ , що жодна точка множини  $P$  не знаходиться всередині описаних довкола трикутників кіл в множині  $DT(P)$ . Триангуляція Делоне дозволяє якомога зменшити кількість малих кутів.

Базуючись на визначенні Делоне, описане коло трикутника утворене трьома точками з вихідної множини точок називається пустим, якщо воно не містить вершин трикутника інших ніж ті три, що його задають (інші точки допускаються тільки на периметрі кола, але не всередині). Умова Делоне стверджує, що мережа трикутників є триангуляцією Делоне, якщо всі описані кола трикутників пусті. Це є початкове визначення для двовимірного простору. Його можна використовувати для тривимірного простору, якщо використовувати описані сфери замість описаних кіл. На рисунку 2.1 продемонстрований приклад триангуляції Делоне.

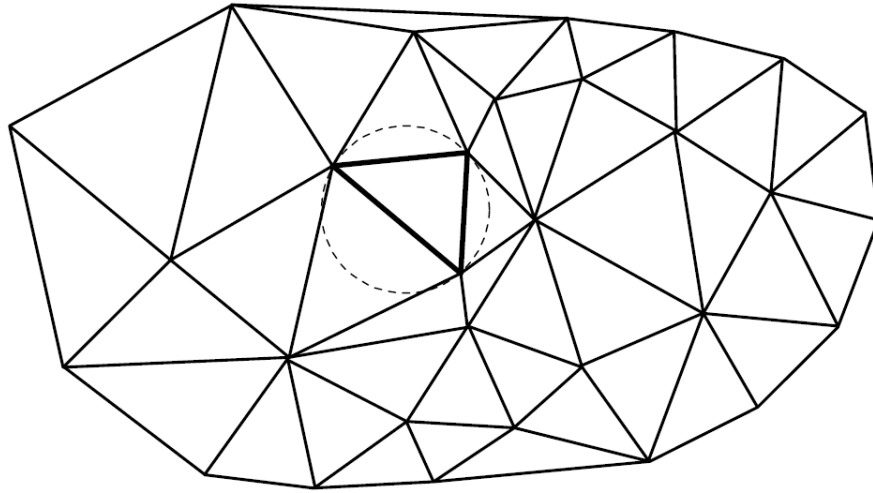


Рисунок 2.1 – Приклад триангуляції Делоне

Триангуляція Делоне вперше з'явилася в науковому світі як граф, двоїстий діаграмі Вороного – однієї з базових структур обчислювальної геометрії.

Діаграма Вороного скінченої множини точок  $S$  на площині являє собою таке розбиття площини, при якому кожна область цього розбиття формує множину точок, які найбільш близькі до одного з елементів множини  $S$  ніж до будь-якого іншого елемента множини. Названа у честь російського вченого Георгія Вороного, також відома як мозаїка Вороного, розбиття Вороного, розбиття Дирихле та розбиття Тиссена. Двоїстість діаграми Вороного триангуляції Делоне виявляється, якщо з'єднати відрізками ті вихідні точки, багатокутники яких дотикаються кутами. Відрізки у такому випадку утворюють триангуляцію Делоне, як показано на рисунку 2.2.

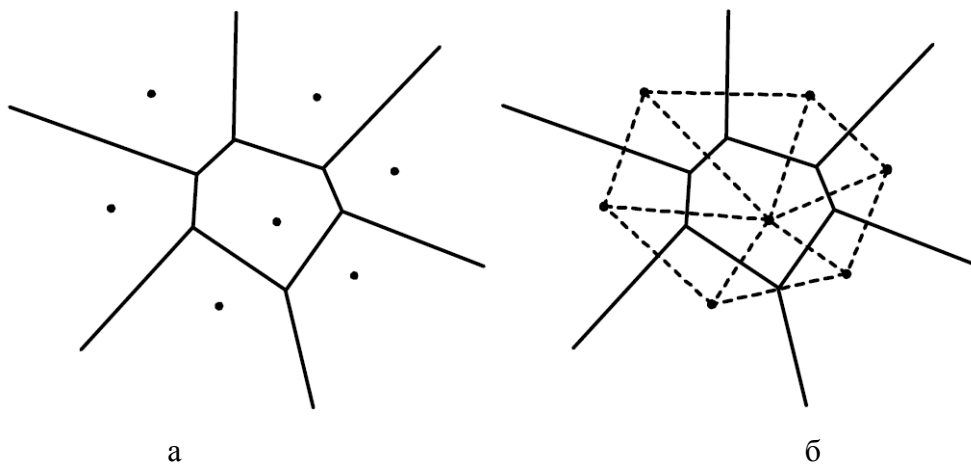


Рисунок 2.2 – Двоїстість діаграми Вороного триангуляції Делоне: а – діаграма Вороного для множини точок на площині, б – відповідна триангуляція Делоне

Розбиття Вороного використовується для створювання синтетичних полікристалічних агрегатів у обчислювальному матеріалознавстві, а також у комп'ютерній графіці для випадкового розбиття поверхонь.

Нехай  $n$  – кількість точок, а  $d$  – розмірність триангуляції Делоне. Тоді можна визначити наступні властивості триангуляції:

- 1) Об'єднання всіх симплексів в триангуляції – це опукла оболонка точок.
- 2) Триангуляція Делоне може містити щонайбільше  $O(n^{d/2})$  симплексів.
- 3) Якщо на площині ( $d = 2$ ),  $b$  вершин входять до опуклої оболонки, то будь-яка триангуляція точок має щонайбільше  $2n - 2 - b$  трикутників, і ще одну зовнішню «грань».
- 4) На площині, в середньому кожна вершина має шість інцидентних трикутників.
- 5) На площині, триангуляція Делоне максимізує найменший кут. Найменший кут в триангуляції Делоне, буде не меншим ніж в будь-якій іншій триангуляції. Правда триангуляція Делоне не обов'язково мінімізує максимальний кут.
- 6) Коло описане навколо довільного трикутника триангуляції не містить всередині жодних інших вхідних вершин.
- 7) Якщо коло що проходить через дві вхідні точки не містить всередині жодних інших, тоді сегмент що з'єднує ці дві точки є ребром триангуляції Делоне цих точок.
- 8) Триангуляція Делоне множини точок в  $d$ -вимірному просторі є проекцією точок опуклої оболонки на  $(d+1)$ -мірний параболоїд.
- 9) Найближчий сусід  $v$  довільної точки  $p$  лежить на ребрі  $vp$  триангуляції Делоне, бо граф найближчих сусідів – це підграф триангуляції Делоне.

Триангуляцію Делоне можна отримати з будь-якої іншої триангуляції з тою же системою точок, послідовно перебудовуючи пари сусідніх трикутників  $\triangle ABC$  і  $\triangle BCD$ , які не задовольняють умові Делоне, у пари трикутників  $\triangle ABD$  і  $\triangle ACD$  [4]. Така операція часто називається фліпом. Приклад такого перетворення зображено на рисунку 2.3.

Однією з найважливіших операцій є перевірка умови Делоне для заданих пар трикутників. На практиці зазвичай застосовують декілька способів перевірки:

- перевірка через рівняння описаної окружності;
- перевірка з задалегідь обчисленою описаною окружністю;
- перевірка суми протилежних кутів;
- модифікована перевірка суми протилежних кутів.

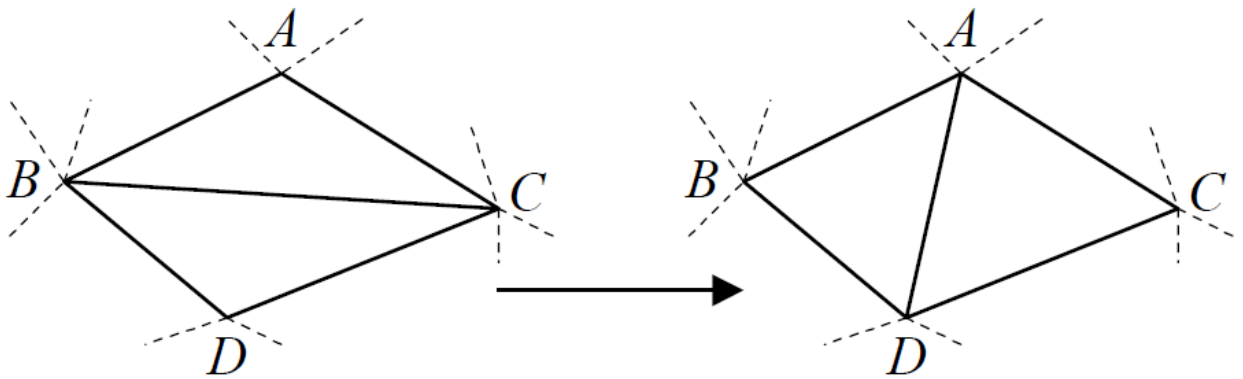


Рисунок 2.3 – Приклад перетворення триангуляції, яка не задовольняє умові Делоне.

Безпосередня реалізація перевірки через рівняння описаної окружності потребує 29 операцій множення та возведення у квадрат, а також 24 операції додавання та віднімання [4]. Рівняння кола, що проходить через точки  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  можна виразити формулою 2.1. У такому випадку умова Делоне для будь якого заданого трикутника буде виконуватися тільки тоді, коли  $(x_0, y_0)$  не буде належати до кола, описаного довкола трикутника  $\Delta((x_1, y_1), (x_2, y_2), (x_3, y_3))$ .

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0. \quad (2.1)$$

Зазначений вище спосіб перевірки потребує великої кількості арифметичних операцій. У більшості випадків операція перевірки умови Делоне виконується багато разів, отже цей варіант є не дуже ефективним.

Перевірка з заздалегідь обчисленою описаною окружністю потребує для кожного трикутника 36 операції множення, возведення у квадрат та ділення, а також 22 операції додавання та віднімання. На етапі безпосереднього виконання перевірок потрібно лише 2 операції возведення у квадрат, 2 віднімання та 1 порівняння [4].

Основна ідея алгоритму перевірки із заздалегідь обчисленою окружністю полягає у попередньому обчисленні для кожного трикутника центру та радіусу описаної довкола окружності, після чого перевірка умови Делоне буде зводитися до обчислення відстані до центру цієї окружності та порівнянню результату з радіусом. Таким чином центр  $(x_c, y_c)$  та радіус  $r$  окружності, що описана довкола трикутника  $\Delta((x_1, y_1), (x_2, y_2), (x_3, y_3))$  можна знайти як  $x_c = b / 2a$ ,  $y_c = -c / 2a$ ,  $r^2 = (b^2 + c^2 - 4ad) / 4a^2$ , де

$$a = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, \quad b = \begin{vmatrix} x_1^2 + y_1^2 & y_1 & 1 \\ x_2^2 + y_2^2 & y_2 & 1 \\ x_3^2 + y_3^2 & y_3 & 1 \end{vmatrix},$$

$$c = \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix}, \quad d = \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix}.$$

Також, необов'язково знати параметри описаної окружності для кожного трикутника. Перевірка умови Делоне завжди виконується для деякої пари трикутників, тому достатньо знати окружність тільки для одного з цих трикутників. Тоді обчислювати треба лише тоді, коли у парі аналізованих трикутників ще не обчислена жодна окружність. Тоді умова Делоне для трикутника  $\Delta((x_1, y_1), (x_2, y_2), (x_3, y_3))$  буде виконуватися лише тоді, коли для будь-якої іншої точки  $(x_0, y_0)$  триангуляції  $(x_0 - x_c)^2 + (y_0 - y_c)^2 \geq r^2$ .

Реалізація такої процедури перевірки потребує для кожного трикутника 36 операцій множення, возведення у квадрат та ділення, а також 22 операції додавання та віднімання. На етапі безпосередньої перевірки потрібно лише 2 возведення у квадрат, 2 віднімання, 1 додавання та 1 порівняння [4].

Перевірка суми протилежних кутів полягає у тому, що умова Делоне для заданого трикутника  $\Delta((x_1, y_1), (x_2, y_2), (x_3, y_3))$  буде виконуватися тоді, коли для будь-якої точки  $(x_0, y_0)$  триангуляції  $\alpha + \beta \leq \pi$ . Ця умова еквівалентна  $\sin(\alpha + \beta) > 0$ , тобто

$$\sin \alpha * \cos \beta + \cos \alpha * \sin \beta \geq 0. \quad (2.2)$$

Якщо вирахувати значення синусів та косинусів через скалярні добутки векторів та підставити у формулу 2.2, то отримаємо наступну формулу перевірки:

$$\begin{aligned} & ((x_0 - x_1) * (y_0 - y_3) - (x_0 - x_3) * (y_0 - y_1)) \\ & * ((x_2 - x_1) * (x_2 - x_3) + (y_2 - y_1) * (y_2 - y_3)) \\ & + ((x_0 - x_1) * (x_0 - x_3) + (y_0 - y_1) * (y_0 - y_3)) \\ & * ((x_2 - x_1) * (y_2 - y_3) - (x_2 - x_3) * (y_2 - y_1)) \geq 0. \end{aligned} \quad (2.3)$$

Безпосередня реалізація процедури такої перевірки потребує 10 операцій множення, а також 13 операцій додавання та віднімання [4]. Виконання даної умови показано на рисунку 2.4.

А якщо не вираховувати одразу всі скалярні та векторні добутки, а вираховувати лише ті, що відповідають  $\cos \alpha$  та  $\cos \beta$

$$S_{\alpha} = (x_0 - x_1) * (x_0 - x_3) + (y_0 - y_1) * (y_0 - y_3),$$

$$S_{\beta} = (x_2 - x_1) * (x_2 - x_3) + (y_2 - y_1) * (y_2 - y_3),$$

тоді, якщо  $S_{\alpha} < 0$  та  $S_{\beta} < 0$ , тоді  $\alpha > 90$ ,  $\beta > 90$  і тому умова Делоне не виконується. Якщо  $S_{\alpha} \geq 0$  та  $S_{\beta} \geq 0$ , тоді  $\alpha \leq 90$ ,  $\beta \leq 90$ , і тому умова Делоне виконується. Інакше потрібні повні обчислення формули 2.3.

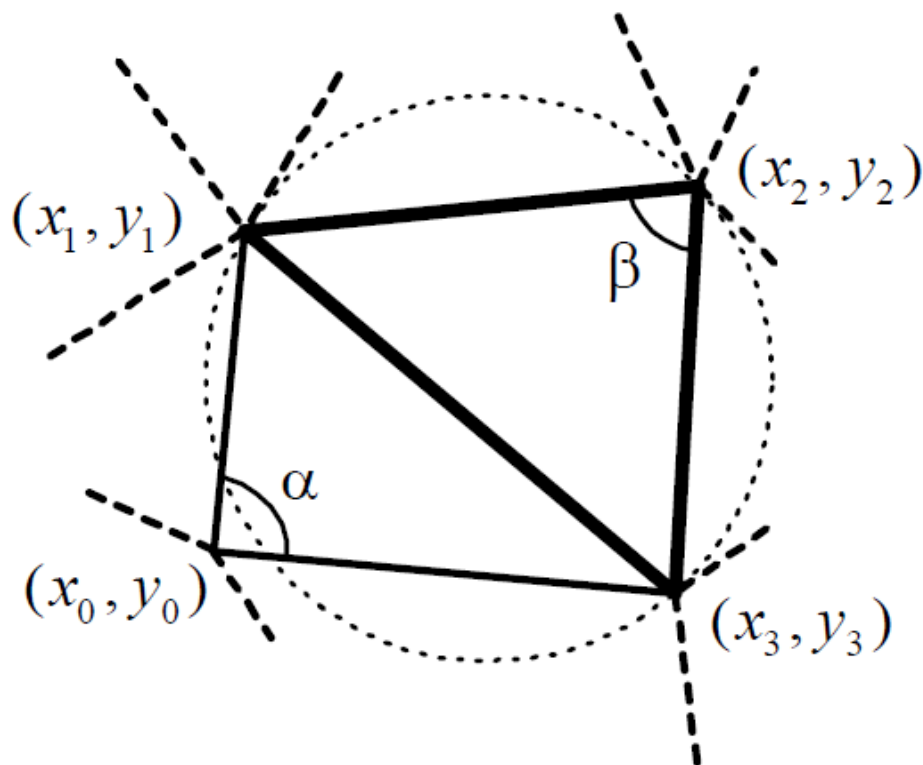


Рисунок 2.4 – Перевірка суми протилежних кутів

Таке удосконалення дозволяє у середньому зменшити кількість арифметичних операцій на 20-40%. Також ця перевірка більш стійка до втрати точності у проміжних



обчисленнях з використанням дійсних чисел з плаваючою крапкою. Середня кількість арифметичних операцій зведена и подана у таблиці 2.1.

Алгоритмів реалізації триангуляції Делоне зараз є дуже багато. Вони розподіляються на 4 групи:

- ітеративні;
- алгоритми злиття;
- алгоритми прямого будівництва;
- двохрохідні алгоритми.

Таблиця 2.1 – Середня кількість арифметичних операцій для усіх способів перевірки умови Делоне.

Спосіб перевірки	Кількість операцій множення та ділення	Кількість операцій додавання та віднімання
Через рівняння описаної окружності	29	24
Із задалегідь обчисленою окружністю	~ 4-7	~ 4-6
Сума протилежних кутів	10	13
Модифікована сума протилежних кутів	~ 7	~ 9

## 2.2 Математична морфологія

Математична морфологія використовує поняття теорії множин та інтегральної геометрії.

Математична морфологія була започаткована французькими дослідниками Матероном и Серром, що вивчали проблеми у мінералогії та петрографії. Вони намагалися визначити фізичні та механічні властивості матеріалів за допомогою аналізу геометричної структури об'єкта.

Нехай  $E^n = R^n$  та  $E^n = Z^n$ , де  $R^n$  – це n-мірний Евклідов простір, а  $Z^n$  – це n-мірний дискретний простір. При роботі з зображеннями зазвичай використовують двомірні простори. На  $E^n$  визначимо індикаторну функцію:

$$\varphi_X(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases} \quad (2.4)$$

Також визначимо міри для множини  $X$ :

$$Mes(X) = \int_{\mathbb{R}^n} \varphi_X(x) dx, \quad (2.5)$$

$$Mes(X) = \sum_{x_i \in X} \varphi_X(x_i). \quad (2.6)$$

Формула 2.5 зображує міру множини  $X$  для неперервного простору  $\mathbb{R}^n$ , у нашому випадку двовимірного зображення – площу множини, а формула 2.6 – міру для дискретного простору  $Z^n$ , що є кількістю вузлів у решітці.

В математичній морфології одним з основних понять є поняття структурного елемента. Основний структурний елемент  $B$  – це множина, яка складається з двох підмножин  $B^1$  та  $B^2$ , що не перетинаються, та для яких є визначений спільний початок.

### 2.2.1 НМ – перетворення (Hit or Miss), операції ерозії та делатації

Назву «НМ-перетворення» запропоновано у [5]. НМ-перетворення є базовим у математичній морфології. Воно дозволяє будувати різноманітні операції на множині. Результат НМ-перетворення для множини  $X = \{x \in E^n\}$  та структурного елемента  $B$  виглядає наступним чином:

$$Y = X \otimes B = \{x: B_x^1 \subset X, B_x^2 \subset X^c\}, \quad (2.7)$$

де  $X^c$  – це доповнення множини  $X$ .

Застосовуючи НМ-перетворення з різноманітними структурними елементами, можна виділяти специфічні геометричні особливості зображень. У результаті виконання

НМ-перетворення на вхідному зображенні будуть виділені елементи, околиця яких співпадає із структурним елементом. Форма околиці визначається формою структурного елемента.

На рисунку 2.2 зображено приклад НМ-перетворення, який наведений у книзі [5]. Умова 2.4 виконується для тих елементів, які знаходяться на нижній границі множини  $X$  (1-4 позиції структурного елемента). На 5 позиції  $B_x^1 \subset X$ , але  $B_x^2 \not\subset X^c$ , на позиції 6 навпаки,  $B_x^1 \not\subset X$  та  $B_x^2 \subset X^c$ , а на 7 позиції не виконуються обидві умови.

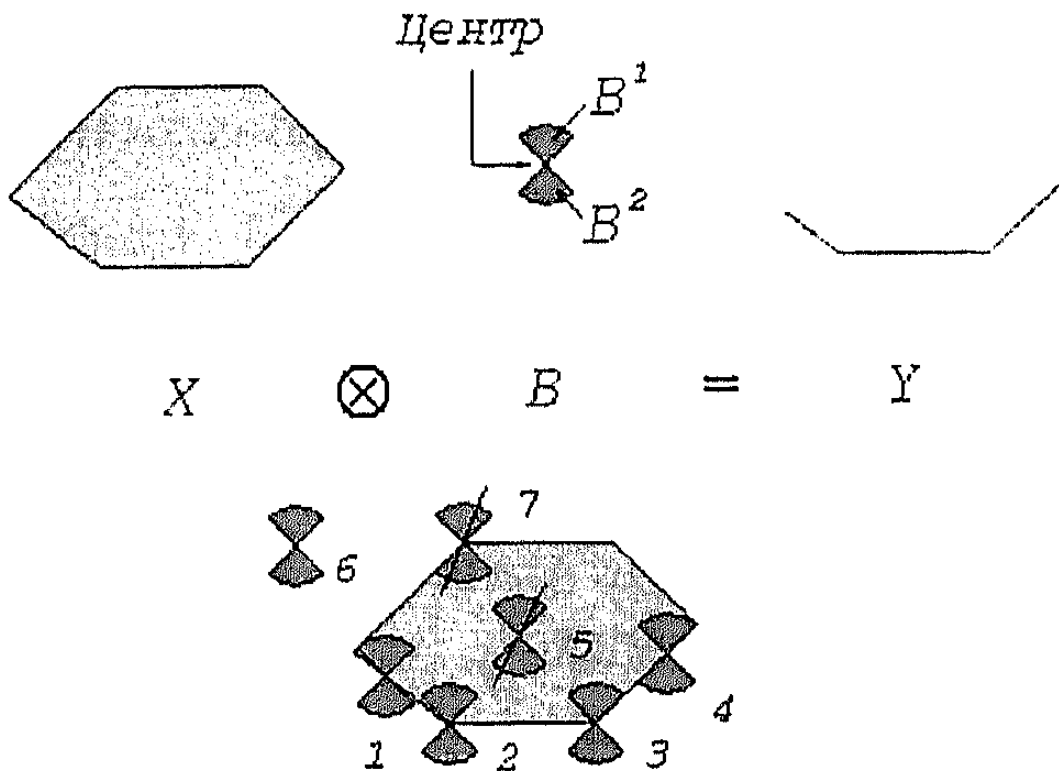


Рисунок 2.2 – Приклад НМ-перетворення

Окремим випадком НМ-перетворення є операція ерозії. Нехай в структурному елементі  $B$  підмножина  $B^2$  є пустою,  $B^2 = \emptyset$ . Тоді умова  $B_x^2 \subset X^c$  завжди виконується. У такому випадку множина  $Y$  буде містити лише ті елементи вхідної множини  $X$ , для яких виконується умова  $B_x \subset X$ :

$$Y = X \ominus B = \{x: B_x \subset X\}.$$

Тобто, якщо  $x \in X$ , а  $b \in B$ , то у множині  $Y$  будуть такі елементи, для яких виконується умова  $x + b \in X$ . На рисунку 2.3 зображено приклад ерозії.

Якщо  $b$  приймає усі можливі положення у  $B$ , то умова  $x+b \in X$  виконується тоді і лише тоді, коли  $x$  належить до зміщеної множини  $X_{-b}$ . У такому випадку можна надати еквівалентне подання ерозії:

$$Y = X \ominus B = \bigcap_{b \in B} X_{-b} = \bigcap_{-b \in B} X_b = \bigcap_{b \in \check{B}} X_b,$$

де  $\check{B} = \{b: -b \in B\}$  – це множина, симетрична  $B$  відносно його початку. Таке подання може виявитися корисним при чисельній реалізації операції ерозії.

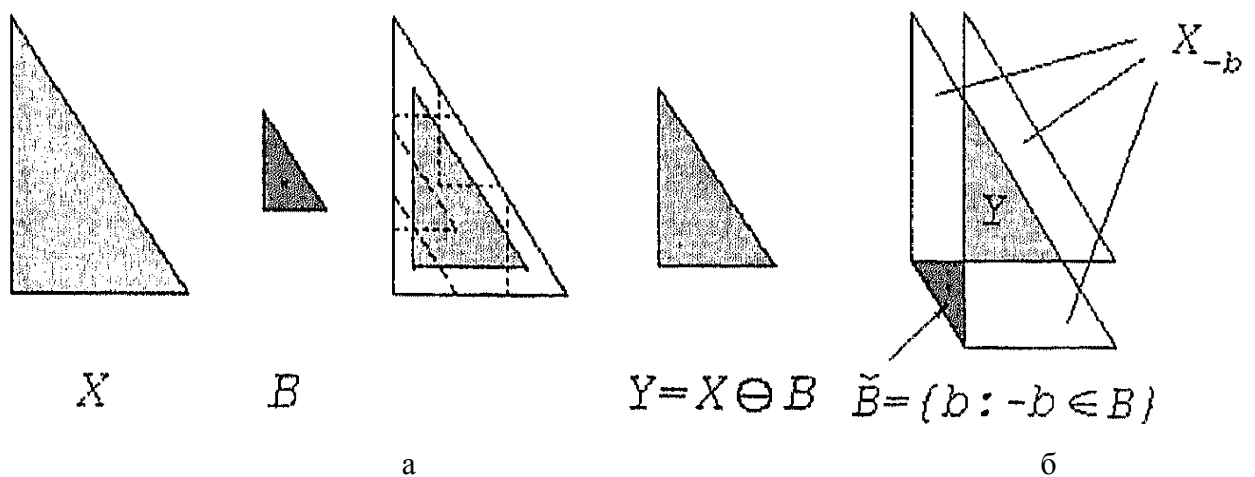


Рисунок 2.3 – Приклад ерозії: а – звичайна ерозія, б – ерозія, як перетин зміщених множин

Двоїстою операцією до ерозії є дилатація, яка визначається за допомогою формул 2.8 або 2.9.

$$Y = X \oplus B = \{x: B_x \cap X = \emptyset\}, \quad (2.8)$$

$$Y = X \oplus B = \bigcup_{b \in \check{B}} X_b. \quad (2.9)$$

Якщо розглядати множину  $X$  як об'єкт, а  $X^c$  як фон зображення, то дилатацію об'єкта можна інтерпретувати як ерозію фону:

$$X \oplus B = (X^c \ominus B)^c.$$

На рисунку 2.4 зображено приклад дилатації.

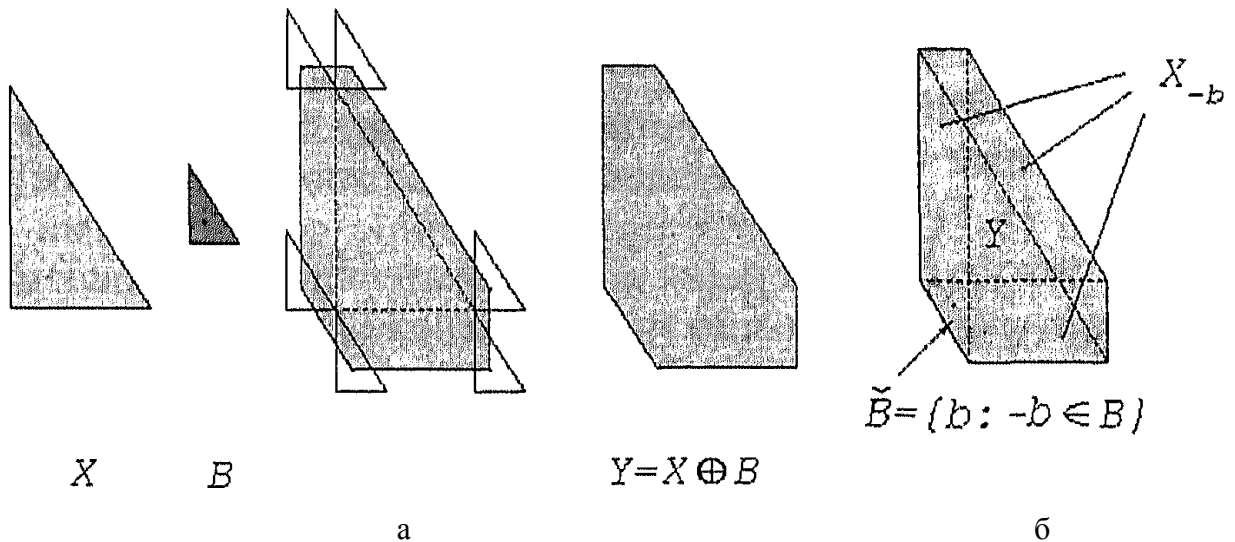


Рисунок 2.4 – Дилатація: а – звичайна дилатація, б – дилатація, як об'єднання зміщених множин.

Ерозія та дилатація призначені у першу чергу для виявлення різноманітних морфологічних особливостей зображення з використанням різних структурних елементів.

Якщо за структурний елемент взяти дві крапки, зсув між якими визначається вектором  $h$ , то за допомогою ерозії можна вирахувати автокореляцію зображення. Автокореляція – це статичний зв'язок між випадковими величинами одного ряду, що взяті зі зсувом, наприклад, у часі. У аналізі зображень при автокореляції шаблон співпадає із зображенням, а зсув здійснюється за напрямком осей  $XU$ . Автокореляція зображення, яке задано індикаторною функцією  $\varphi_X(x)$ , визначається як

$$K_X(h) = \int_{R^n} \varphi_X(x) \varphi_X(x+h) dx = \int_{R^n} \psi_Y(x; h) dx,$$

де  $\psi_Y(x; h)$  – це індикаторна функція множини  $Y = X \cap X_{-h}$ .

У загальному випадку неможливо точно відновити вихідну множину  $X$  після ерозії за допомогою однієї лише дилатації. Дилатація відновлює тільки частину множини  $X$ , яка

має менше деталей, але є найбільш суттєвою с точки зору характеристик форми та розміру.

### 2.2.2 Розкриття та закриття

Визначимо розкриття (opening) множини  $X$  за допомогою структурного елемента  $B$  як

$$Y = X_B = (X \ominus B) \oplus \check{B}.$$

Аналогічно визначимо операцію закриття (closing) множини  $X$  з використанням структурного елемента  $B$ :

$$Y = X^B = (X \oplus B) \ominus \check{B}.$$

Можна вивести такі співвідношення:  $(X^c)_B = (X^B)^c$  та  $(X_B)^c = (X^c)^B$ . Для зображень ці співвідношення означають, що відкриття та закриття є еквівалентними операціями. Також вони є ідемпотентними відносно одна одної.

Відкриття і закриття використовуються для фільтрації зображень, згладжування границь, видалення дрібних об'єктів та вузьких «хвостів» (відкриття) чи видалення дрібних порожнин та вузьких «каналів» (поповнення). Ступінь згладжування та розміри видалених об'єктів залежать від розмірів структурного елемента. Під час фільтрації однаковими структурними елементами рівень викривлень у корисних деталях зображення значно менший при використуванні відкриття чи закриття, ніж при використуванні ерозії або дилатації.

### 2.2.3 Тоншення та товщення

Визначимо операцію тоншення формулою 2.10, а операцію товщення – формулою 2.11:

$$Y = X \odot T = X \setminus (X \otimes T), \quad (2.10)$$

$$Y = X \odot T = X \cup (X \otimes T), \quad (2.11)$$

де  $T = (T_1, T_2)$  – це структурний елемент, який містить дві множини  $T_1$  та  $T_2$ , що не перетинаються.

Визначимо послідовні тоншення і товщення формулами 2.12 та 2.13 відповідно.

$$X \odot \{T^i\} = (\dots ((X \odot T^1) \odot T^2) \dots) \odot T^l, \quad (2.12)$$

$$X \odot \{T^l\} = (\dots ((X \odot T^1) \odot T^2) \dots) \odot T^l, \quad (2.13)$$

де  $\{T^i\}$  – це послідовність структурних елементів  $T$ .

На рисунку 2.5 зображено результат послідовного тоншення множини  $X$  за допомогою послідовності структурних елементів  $\{L^i\}$ , де  $L^i$  відрізняються одна від одної поворотом навколо центральної точки. Результат послідовного тоншення не змінюється при наступному застосуванні до нього операції тоншення.

Отриманий результат послідовного тоншення – це скелетон (або скелет) множини  $X$ . Отримання скелету має наглядне пояснення, яке називають «пожежею у степу». Множина уявляється степом, що вкритий сухою травою. Припустимо, що одночасно вздовж усієї границі степу спалахує вогонь, що поширюється у всіх напрямках с однаковою швидкістю. У першу мить фронт розповсюдження вогню співпадає з границею. Під час поширення різні ділянки фронту зустрічаються один з одним, і у місцях зустрічі вогонь буде згасати. Ці ділянки самозгасання утворюють «скелетон» масиву. Якщо множина  $X$  – зв'язна, то скелетон теж є зв'язною множиною.

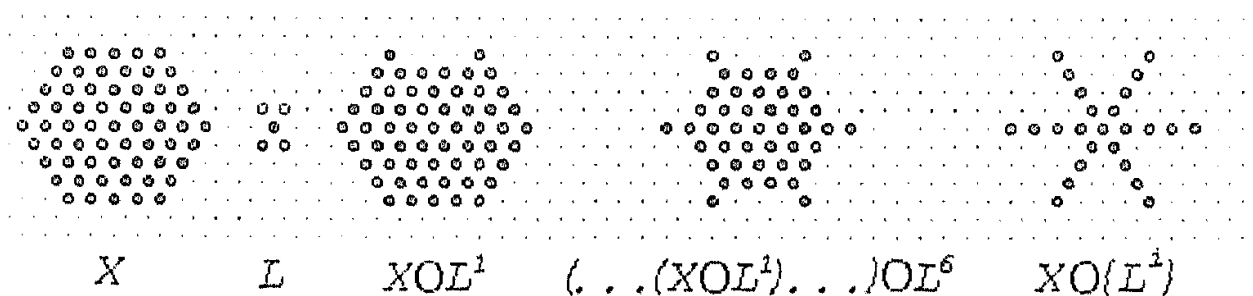


Рисунок 2.5 – Послідовне тоншення

Для неперервного двомірного простору визначені наступні властивості точок скелетону множини  $X$ :

– якщо точка  $x$  належить скелетону, та  $B_x$  – найбільше коло з центром у точці  $x$ , що належить  $X$ , то неможливо знайти більше коло, що належить  $X$  (не обов'язково з центром у  $x$ ), що містить  $B_x$ .

– коло  $B_x$  торкається границі множини  $X$  у двох чи більше точках.

Також скелетон визначений, як множина максимальних кіл, що належать  $X$ . Під максимальним колом мається на увазі коло, що доторкається до границь  $X$  у двох чи більше точках. Рисунок 2.6 ілюструє це визначення.

З цього визначення випливає ще одна властивість скелетона: якщо кожній точці скелетона зіставити значення радіуса максимального кола, центром якого вона є, то можна відновити множину  $X$ , що породила скелетон:

$$X = \bigcup_{x \in S(X)} (\rho(x)B)_x,$$

де  $\rho(x)$  – це радіус максимального кола для точки  $x$  скелетону,  $B$  – коло одиничного радіусу.

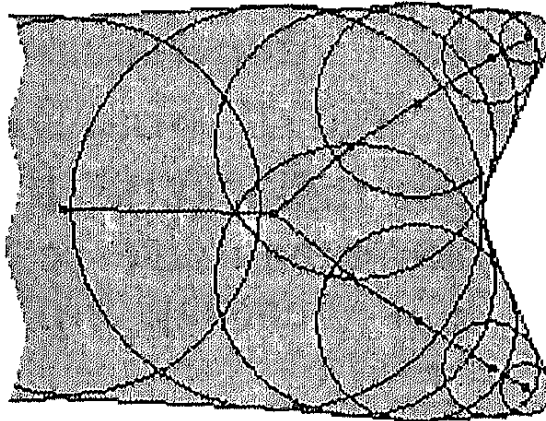


Рисунок 2.6 – Визначення скелетона через максимальні кола

Скелетонізацію використовують під час обробки креслень та розпізнаванні символів для зведення ліній до одиничної ширини. Побудування скелетону фонові компоненти зображення, що містить деяку кількість об'єктів, дозволяє сегментувати його на ділянки, кожна з яких може бути інтерпретована як зона впливу (життєвий простір) об'єкта.



## **3 АЛГОРИТМИ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ХМАРНОГО СЕРВІСУ ВЕКТОРИЗАЦІЇ ЗОБРАЖЕНЬ**

Система побудована за принципами моделі SaaS з використанням сервісно-орієнтованої архітектури. Доступ до додатка буде здійснюватись через web-інтерфейс у браузері. Web-клієнт реалізує шаблон проектування MVC. У якості мови програмування було обрано мову високого рівня C#. Середовищем для створення програмного продукту було обрано Visual Studio 2010. Використовувалася бібліотека комп'ютерного зору з відкритим кодом OpenCV.

### **3.1 Архітектура системи**

#### **3.1.1 Сервісно-орієнтована архітектура**

Сервісно-орієнтована архітектура (Service Oriented Architecture, SOA) – це різновид архітектури програмного забезпечення, що характеризується модульним підходом до розробки програмного забезпечення. Він заснований на використанні розподілених, слабко пов'язаних замінних компонентів, оснащених стандартизованими інтерфейсами для взаємодії по стандартизованим протоколам. Джерімі Уестерман дає наступне визначення SOA: це парадигма, призначена для проектування, розробки та управління дискретних одиниць логіки (сервісів) в обчислювальному середовищі.

Для використання сервісу необхідно дотримуватися угоди про інтерфейс для звернення до сервісу — інтерфейс повинен не залежати від платформи. SOA реалізує масштабованість сервісів — можливість додавання сервісів, а також їх модернізацію. Постачальник сервісу і його споживач виявляються непов'язаними — вони спілкуються за допомогою повідомлень. Оскільки інтерфейс повинен не залежати від платформи, то і технологія, використовувана для визначення повідомлень, також повинна не залежати від платформи. Тому, як правило, повідомлення є XML-документами, які відповідають XML-схемі.

На рисунку 3.1 проілюстрована взаємодія з сервісом.

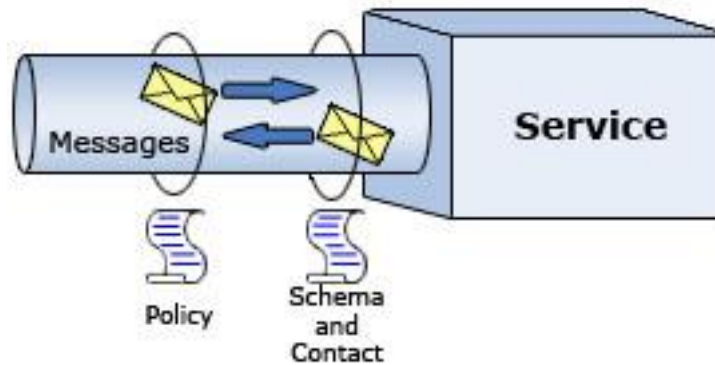


Рисунок 3.1 – Схема взаємодії з сервісом

### 3.1.2 Модель SaaS

SaaS (Software as a Service, укр. «програмне забезпечення, як послуга») – бізнес-модель використання програмного забезпечення. Постачальник розроблює web-додаток та самостійно керує їм, а замовник має доступ до функціоналу через мережу Інтернет.

Найбільша перевага цієї моделі – це відсутність витрат на встановлення, оновлення та підтримку програмного забезпечення на стороні користувача. Також до переваг можна віднести мультплатформеність, відсутність необхідності установки ПЗ на робочих місцях користувачів та швидкість впровадження, обумовлена відсутністю витрат часу на розгортання системи.

Основні принципи моделі SaaS:

- 1) застосунок пристосований до використання через мережу Інтернет;
- 2) одним застосунком можуть користуватися багато клієнтів одночасно;
- 3) модернізація та оновлення відбувається непомітно та прозоро для користувачів, та зручно для розробників.

Ця модель дозволяє розробникам програмного забезпечення ефективніше боротися з неліцензійним використанням програмного забезпечення. Так як користувачі мають доступ лише до однієї копії за стосунку, то розробникам легше контролювати доступ до програми та відстежувати спроби несанкціонованого доступу.

### 3.1.3 Концепція MVC та шаблони проектування

MVC (Model-View-Controller) – це схема використання шаблонів проектування за допомогою яких модель застосунку, інтерфейс користувача та взаємодія з користувачем розділені на три окремі компоненти. Модифікація одного з компонентів мінімально впливає на інші.

Термін Model-View-Controller використовується з кінця 1970х років. Уперше його було застосовано у проекті Smalltalk компанії Xerox PARC Трюгве Реенсаугом. Розрізняють дві модифікації цього підходу:

- пасивна, коли модель не може впливати на вигляд або контролер та використовується ними лише для відображення даних.

- активна, коли модель сповіщає вигляд про те, що у ній сталися зміни, а вигляди, які заінтересовані у оповіщенні, підписуються на ці повідомлення, що дозволяє зберегти незалежність моделі як від контролера, так і від вигляду.

Класичною реалізацією концепції MVC зазвичай вважають реалізацію активної моделі. Основна ціль застосування даної концепції – це відділення бізнес-логіки (моделі, model) від її візуалізації (вигляду, view). За рахунок цього підвищується шанс повторного використання.

Концепція MVC дозволяє розділити дані, вигляд та обробку дій користувача на три основних елемента:

- 1) Модель (Model) – це дані та методи роботи з цими даними; модель реагує на запити, змінюючи свій стан; не містить інформації про візуалізацію даних.

- 2) Вигляд (View) – це модуль, який відповідає за візуалізацію інформації; найчастіше у ролі вигляду використовуються форма або вікно з графічними елементами.

- 3) Контролер (Controller) – це модуль, який забезпечує зв'язок між користувачем та системою: контролює ввід даних користувачем та використовує модель та вигляд для реалізації необхідної реакції.

Вигляд та контролер залежні від моделі, у той час коли модель незалежна від вигляду та контролеру. Це і є призначення даної концепції: можна будувати модель незалежно від візуального вигляду, а також створювати декілька різних виглядів для кожної моделі.

Для реалізації концепції Model-View-Controller зазвичай застосовується велика кількість шаблонів проектування. Найчастіше використовуються шаблони «Спостерігач» (Observer), «Стратегія» (Strategy) та «Компонувальник» (Composite).

Шаблон «Спостерігач», також відомий під назвами «Підлегли» та «Видавець-передплатник», є поведінковим шаблоном проектування. Ключовими об'єктами у цьому шаблоні є суб'єкт та спостерігач. У суб'єкта може бути скільки завгодно залежних від нього спостерігачів. Усі спостерігачі сповіщаються про зміни у стані суб'єкта. Отримавши повідомлення, спостерігач опитує суб'єкта з метою синхронізації стану [3]. UML діаграма класів, що демонструє реалізацію цього шаблону проектування зображена на рисунку 3.1.

Така взаємодія часто називають відносинами «Видавець-передплатник», тому що суб'єкт видає чи публікує повідомлення та розсилає їх, навіть не маючи інформації про те, які об'єкти є передплатниками. На отримання повідомлень може підписатися необмежена кількість спостерігачів.

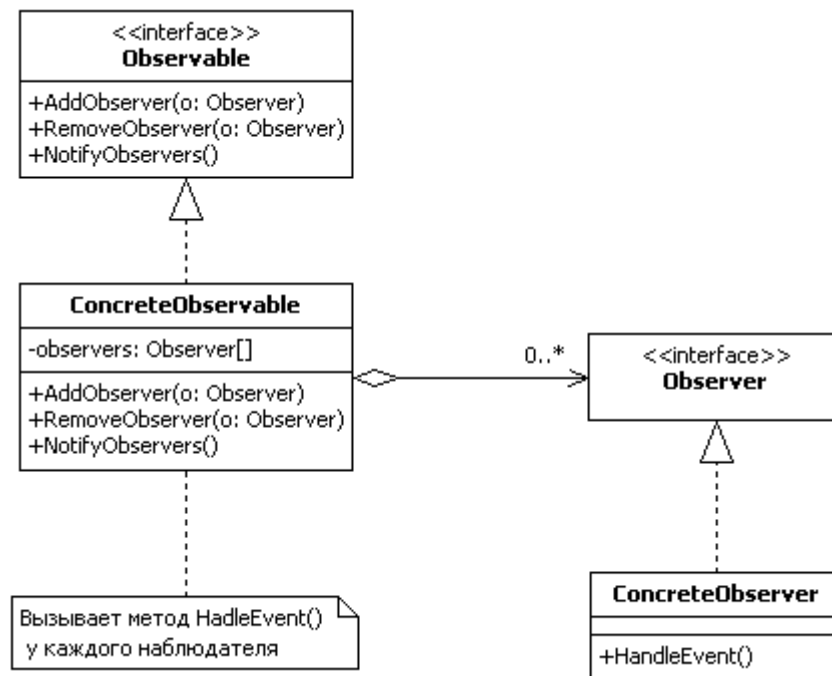


Рисунок 3.1 – UML діаграма класів для шаблону проектування «Спостерігач»

Цей шаблон рекомендовано використовувати, коли при модифікації одного об'єкта потрібно змінити інші і неможливо знати скільки саме об'єктів потрібно буде оновити. Також такий підхід допомагає уникнути дуже тісних зв'язків між об'єктами, що дає можливість створювати та видаляти спостерігачів для об'єкта у будь-який час [3]. Об'єкт має інформацію лише про те, що у нього є спостерігачі, кожний з них реалізує визначений інтерфейс, про реальні класи спостерігачів йому нічого невідомо. Таким чином зв'язки між об'єктами та спостерігачами мають абстрактний характер і зведені до мінімуму.

Оскільки об'єкт та спостерігач не є тісно зв'язаними, вони можуть знаходитися на різних рівнях абстракції системи. Об'єкт більш низького рівня може сповіщати спостерігачів, що знаходяться на верхніх рівнях, не порушуючи ієрархію системи.

У цієї концепції існує недолік, який пов'язаний з неочікуваними оновленнями. Оскільки спостерігачі не мають інформацію один про одного і про те, наскільки дорого обходиться зміна об'єктів. Звичайна операція над об'єктом може викликати цілий ряд оновлень спостерігаючи за ним класів. Також нечітко визначені чи погано визначені критерії залежності можуть стати причиною непередбачених оновлень, які дуже важко буде простежити.

Шаблон «Стратегія» (Strategy), відомий також під назвою «Політика» (Policy), є поведінковим шаблоном. Він визначає групу алгоритмів, інкапсулює кожний з них и робить їх взаємозамінними. «Стратегія» дозволяє змінювати алгоритми незалежно від клієнтів, які їх використовують. На рисунку 3.2 зображена UML діаграма класів для данного шаблону проектування.

Цей шаблон потрібно використовувати, коли є багато споріднених класів, які відрізняються лише поведінкою. «Стратегія» дозволяє сконфігурувати клас, визначивши одну з можливих поведінок. Також цей шаблон дуже зручний, коли потрібно мати декілька реалізацій різних варіантів алгоритму, або у класі визначено багато поведінок, що подано у вигляді умовних операторів. У такому випадку краще перенести код з гілок в окремі класи стратегій. Цей підхід також дозволяє ховати специфічні структури даних, які користувач не повинен знати.

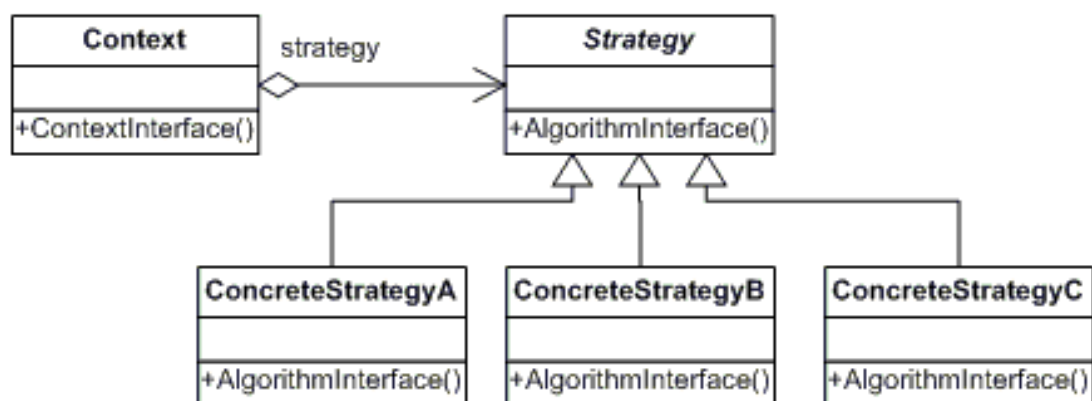


Рисунок 3.2 – UML діаграма класів для шаблону проектування «Стратегія»

Шаблон «Стратегія» визначає групу алгоритмів або поведінок, які можна використовувати повторно у різних контекстах. Наслідування дозволяє виділити загальну для усіх алгоритмів функціональність. Потенціальний недолік цього шаблону полягає у

тому, що клієнти мають «знати» про усі можливі стратегії, та розуміти чим одна відрізняється від одної. Тому доведеться розкрити клієнту деякі особливості реалізації усіх стратегій. Отже, цей шаблон має сенс застосовувати лише тоді, коли відмінності між стратегіями мають значення для клієнта. Також використання стратегій збільшує кількість об'єктів у за стосунку. Іноді ці витрати можна скоротити, якщо реалізувати стратегії у вигляді об'єктів без стану, які будуть розділятися декількома контекстами. Розподілені стратегії не повинні зберігати стан між викликами [3].

Шаблон «Компонувальник» (Composite) – це шаблон, який структурує об'єкти, компонуючи їх у деревовидні структури для подання ієрархій частина-ціле. Дозволяє клієнтам одноманітно трактувати індивідуальні та зіставні об'єкти. На рисунку 3.3 зображена UML діаграма класів для цього шаблону проектування.

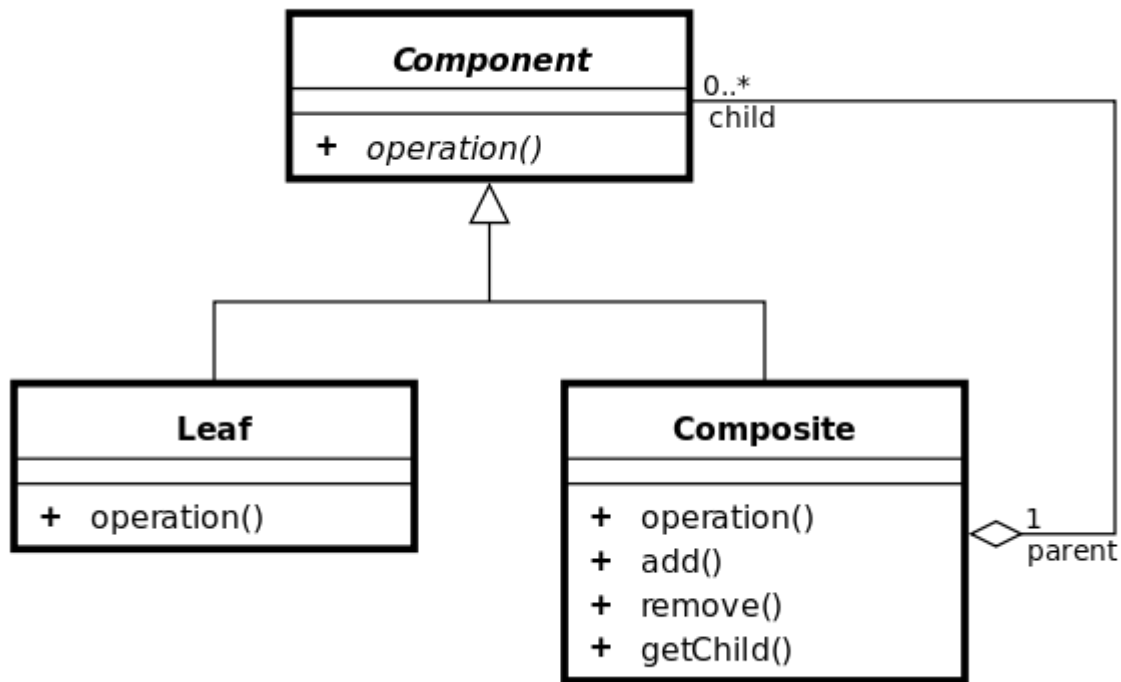


Рисунок 3.3 – UML діаграма класів для шаблону проектування «Компонувальник»

Цей шаблон використовується, коли потрібно подати ієрархію об'єктів у вигляді частина-ціле, а також коли потрібно щоб клієнти одноманітно трактували зіставні та індивідуальні об'єкти [3]. З примітивних об'єктів можна зіставляти більш складні, які у свою чергу, приймають участь у більш складних композиціях.

Такий підхід спрощує архітектуру клієнта, тому що клієнт зазвичай не знає з чим він працює – із зіставним об'єктом чи з примітивом. При додаванні нових компонентів не

потрібно змінювати архітектуру клієнту. Але ця простота додавання компонентів має і негативні сторони: стає дуже важко накладати обмеження на те, які об'єкти можуть бути у складі композиції. Іноді потрібно, щоб зіставний об'єкт міг складатися з чітко визначених компонентів. Шаблон «Компонувальник» не дозволяє використовувати для реалізації таких обмежень статичною системою типів. Замість цього потрібно проводити перевірки під час виконання за стосунку.

### **3.2 Обґрунтування вибору середовища розробки та зовнішніх бібліотек**

Середовищем для створення програмного продукту було обрано Microsoft Visual Studio 2010. Воно є найпотужнішим засобом розробки надійних багаторівневих програм для Windows, мережі Інтернет, мобільних пристроїв та додатків Microsoft Office. Розробка програми проводилася із допомогою мови програмування високого рівня C#.

Використовувалася бібліотека комп'ютерного зору з відкритим кодом OpenCV. Ця бібліотека містить більше 500 функцій, оптимізованих для виконання в реальному часі. До першої версії розробкою займалася російська команда Intel в Нижньому Новгороді. Бібліотека містить алгоритми для обробки, реконструкції і очищення зображень, розпізнавання образів, захоплення відео, стеження за об'єктами, калібрування камер та ін.

Спочатку перед OpenCV ставилися наступні цілі:

- дослідження машинного зору, розробка і оптимізація коду;
- поширення інформації про машинний зір, розробка загальної інфраструктури на якій могли б ґрунтуватися розробники;
- програми мають бути переносимими, оптимізованими, код яких не обов'язково має бути відкритим. Додатки також можуть створюватися в комерційних цілях.

Для завантаження доступні версії для Windows, Mac, Linux. Підтримується безліч мов, включаючи C/C++, Python, Java, C#.

OpenCV складається з декількох модулів:

1) CXCORE – ядро, містить:

- базові структури;
- матричну алгебру;
- алгоритми роботи з пам'яттю;

- алгоритми перетворення типів;
  - алгоритми для обробки помилок;
  - функції для запису/читання XML файлів;
  - функції для роботи з 2D графіками.
- 2) CV – модуль обробки зображень, робота з комп'ютерним зором містить:
- функції для роботи із зображеннями (перетворення, фільтрація і так далі);
  - функції для аналізу зображень (пошук контурів, гістограми і так далі);
  - алгоритми аналізу рухів, стеження за об'єктами;
  - алгоритми розпізнавання об'єктів (облич, предметів);
  - алгоритми для калібрування камер.
- 3) ML – навчання машин:
- функції для класифікації;
  - функції для аналізу даних.
- 4) HighGUI – модуль для створення призначеного для користувача інтерфейсу, відповідає за:
- створення вікон;
  - виведення зображень;
  - захоплення відео з файлів і камер;
  - читання/запис зображень.
- 5) CVCAM – захоплення відео з цифрових камер;
- 6) CVAUX – застарілі функції:
- просторовий зір;
  - знаходження і опис рис обличчя;
  - пошук стерео відповідностей;
  - опис текстур.

Підтримувані компілятори:

- Windows: Microsoft Visual C++, Borland C++, Intel Compiler, MinGW
- Linux: GCC, Intel Compiler
- Mac: Intel Compiler, Carbon та ін. [6].

У розробленій системі комп'ютерного зору бібліотека OpenCV застосовується для вирішення задачі триангуляції. Таке рішення дозволяє уникнути ускладнень при обробці даних і допомагає прискорити процес виділення об'єктів. Реалізований в бібліотеці алгоритм побудування триангуляції показує стабільну роботу з мінімумом неточностей, а так само високу швидкодію.



### 3.3 Алгоритм роботи програмного засобу

Поставлена задача може бути підрозділена на наступний ряд підзадач:

- завантаження зображення на сервер;
- нормалізація та позбавлення шумів;
- обчислення кількості кольорів та застосування пастеризації;
- застосування операцій математичної морфології до отриманого зображення;
- побудування тріангуляції Делоне з обмеженнями;
- апроксимація об'єктів відрізками та кривими Без'є;
- створення графічного файлу та відправлення його клієнтському за стосунку.

### 3.4 Результати роботи програмного засобу

Залежність розміру результуючого файлу зображення від кількості кольорів на зображенні подано у таблиці 3.1. При зменшенні кількості кольорів до 256 розмір файлу зменшився у 3,5 рази відносно вихідного розміру.

Таблиця 3.1 – Залежність розміру результуючого файлу зображення від кількості кольорів

Кількість кольорів	Розмір, байт
42450 (вихідне )	175430
4096	148456
1024	104496
256	53228

Діаграму залежності розміру файлу від кількості кольорів на зображенні можна побачити на рисунку 3.4.

Залежність кількості контурів на результуючому зображенні від кількості кольорів подано у таблиці 3.2. Зменшення кількості кольорів може проводити тільки людина. Тому адміністратор повинен самостійно задати базові значення, а юзер потім буде мати змогу їх корегувати залежно від бажаного результату. Стандартні розміри палітер не можна

змінити: 2, 4, 8, 16, 32, 64, 128, 256. Але якщо кількість кольорів не співпадає з палітрою – то це означає, що деякі кольори просто не використовуються.

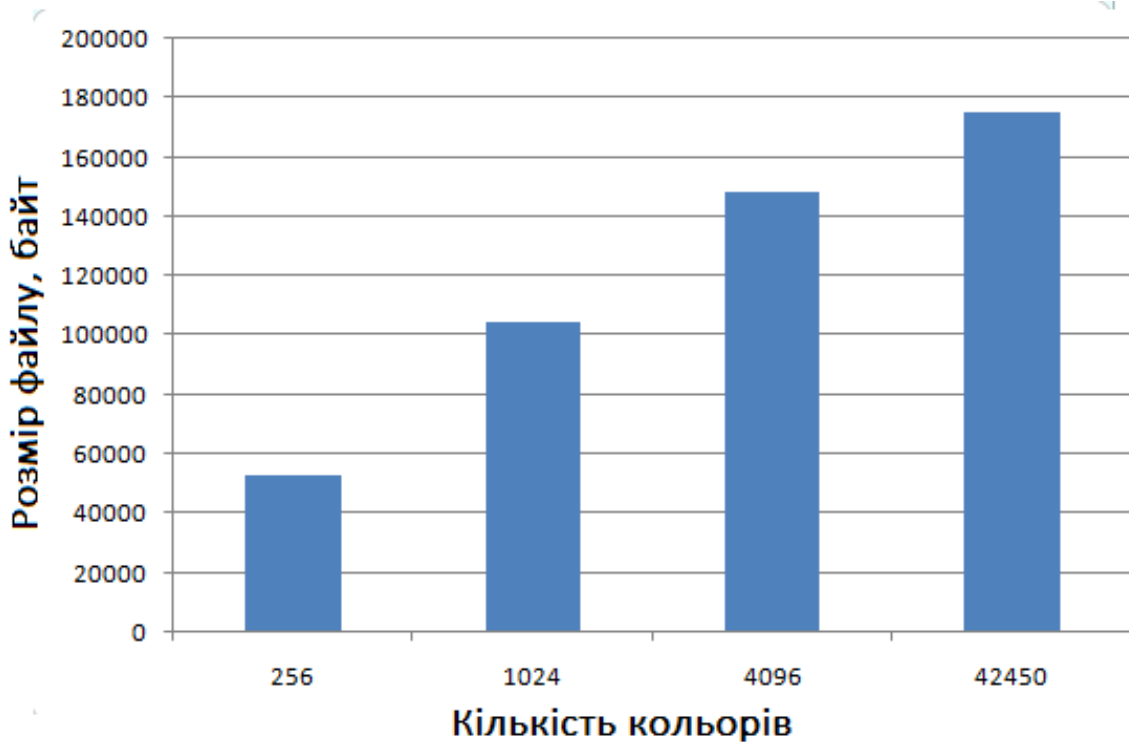


Рисунок 3.4 – Діаграма залежності розміру файлу від кількості кольорів

Таблиця 3.2 – Залежність кількості контурів на результуючому зображенні від кількості кольорів

Кількість кольорів	Кількість контурів
42450 (вихідне )	90430
4096	54845
1024	13711
256	1228

Але залежність кількості контурів від кількості кольорів не є прямою, тому що зображення може мати велику кількість невеликих кластерів, забарвлених двома чи трьома кольорами, як, наприклад, шахова дошка. У деякий момент кількість контурів різко зменшується, тому що області близьких відтінків починають зливатися.

Діаграма залежності кількості контурів на результуючому зображенні від кількості кольорів показана на рисунку 3.5.



Рисунок 3.5 – Діаграма залежності кількості контурів на результуючому зображенні від кількості кольорів

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення. Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [10] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу

слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [11], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

## 4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;

- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де  $U_\phi$  - номінальна фазна напруга мережі, В;

$Z_\Pi$  - повний опір петлі, створене фазними і нульовими дротами, Ом;

$Z_T$  - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [12]:  $Z_T / 3 = 0,1$  Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де  $R_\Pi = R_\phi + R_0$  - сумарний активний опір фазного  $R_\phi$  і нульового  $R_0$  дротів, Ом;

$X_\Pi$  - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту  $S = 2.5$  мм, тоді згідно таблицям 5 і 6 [12], має такий опір:

$X_\Pi = 0,11$  Ом;

$$R_{\phi} = 7,55 \text{ Ом};$$

$$R_o = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,1^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де  $I_n$  - номінальний струм спрацьовування плавкої вставки, А;

$k$  - коефіцієнт кратності нелінійного струму  $I_n$ , А.

Коефіцієнт кратності нелінійного струму  $I_n$  розраховується по формулі (4.4) :

$$I_n = P / U, \quad (4.4)$$

де  $P = 220 \text{ Вт}$  - споживана потужність;

$U = 220 \text{ В}$  - робоча напруга;

$k = 3 \text{ А}$  - для плавких вставок.

$$\text{Отже, } I_n = 220 / 220 = 1 \text{ А}.$$

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

### 4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [10], ДБН, відповідними ГОСТами і ОСТами з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [10] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл.4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в ДБН (30 м<sup>3</sup> в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для



оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибрати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплєну).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [13].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [17] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛПО2П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки  $a=12$  м, шириною  $b=10$  м, заввишки  $z=4$  м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де  $F$  - світловий потік = 3120 лм;

$E$  - максимально допустима освітленість робочих поверхонь = 200 лк;

$S$  - площа підлоги = 120 м<sup>2</sup>;

$Z$  - поправочний коефіцієнт світильника = 1,2;

$k$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

$n$  - кількість світильників;

$U$  - коефіцієнт використання освітлювальної установки = 0,6;

$M$  - кількість ламп у світильнику = 2.

Отже,  $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$ .

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

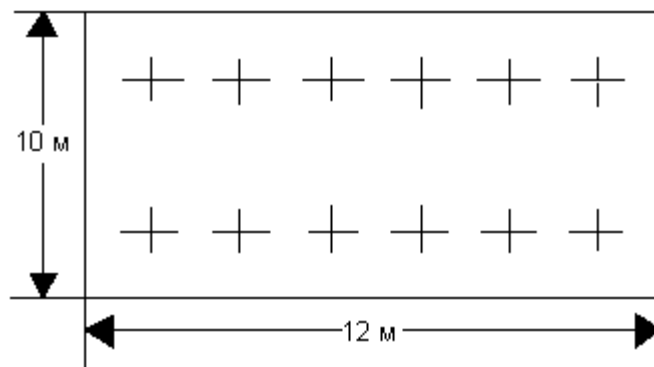


Рисунок 4.1 - Схема розташування світильників

#### 4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно [19] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогненебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [14] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1 (димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, лом, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

#### **4.5 Вплив на навколишнє середовище**

В даний час зростає кількість комп'ютерної техніки в усіх галузях діяльності людини. Багато користувачів і виробників помиляються, вважаючи, що зі зменшенням і удосконаленням комп'ютерів, зменшиться їх негативний вплив на навколишнє середовище.

На даний момент найбільш суворим з існуючих світових стандартів екологічності для комп'ютерної техніки є стандарт ТСО-99. У порівнянні з попередніми він містить додаткові обмеження по частині екології, ергономіки, енергоспоживання і емісії пристроїв.

Організація по захисту навколишнього середовища Greenpeace з 2006 року оцінює виробників електроніки за кількістю важких металів і отруйних речовин, наприклад інгібіторів горіння, використовуваних ними при виробництві (інгібітор - речовина, присутність якого в невеликих кількостях призводить до запобігання або уповільнення процесів горіння або корозії; інгібітори знижують швидкість хімічних реакцій або пригнічують їх). Однак навіть оцінки такої організації, як Greenpeace, не можуть претендувати на об'єктивність. Адже в одних випадках вона використовує перевірену інформацію, що стосується, наприклад, заходів щодо утилізації відходів, а в інших спирається тільки на дані виробника. А якщо компанія не повідомляє ніяких відомостей, то автоматично опиняється на нижніх рядках рейтингу. Крім того, енергетичні витрати на виробництво і перевезення продукції також необхідно враховувати при оцінці екологічної ефективності. Адже часи, коли техніка виготовлялася тільки на одному заводі, давно пройшли. Сьогодні окремі комплектуючі закупаються на різних підприємствах по всьому світу, після чого здійснюється складання пристроїв. Тому найчастіше навіть самі компанії не можуть знати, які шкідливі речовини потрапляють в атмосферу при виготовленні їх продукції і які саме метали або токсини в ній містяться.

ЖК-екрани - один з джерел парникових газів, які набагато шкідливіше діоксиду вуглецю. Рідкокристалічні монітори швидко знайшли популярність, прийшовши на зміну громіздким ЕПТ-моделям. І це не дивно, адже вони мають тонкі корпуси і споживають значно менше електроенергії. За іншим аспектам екологічної безпеки дисплеї на основі рідких кристалів також вважалися проривом, тому що в них не використовувався газ, що містить свинець. Досить довго ніхто не звертав уваги на застосовуваний для чищення РК-панелей тріфтористий азот (NF<sub>3</sub>), і тільки в середині 2008 року вченими було доведено наявність даної хімічної речовини в атмосфері. Відкриття було вражаючим: порівняно з діоксидом вуглецю (CO<sub>2</sub>) NF<sub>3</sub> має в 17 000 разів більше активного парникового газу, а його атмосферний час напіврозпаду може складати від 550 до 740 світлових років (у CO<sub>2</sub> - від 30 до 40 років). Закону, який обмежував би рівень викиду NF<sub>3</sub>, поки не існує.

Виявлення енерговитрат є таким же проблематичним процесом, як і визначення кількості матеріалів, придатних для вторинної переробки, і важких металів, що містяться в пристроях. Таким чином, надійним показником екологічності залишається тільки рівень енергоспоживання.

Полівінілхлорид, що позначається зазвичай аббревіатурою ПВХ, - це різновид пластику, що застосовується в самих різних цілях. З нього зроблена зовнішня оболонка кабелів, якими з'єднуються пристрої, він оточує електричний провід портативного комп'ютера. Це дешевий, міцний і вельми поширений матеріал. Разом з тим, за словами ІТ-аналітика «Грінпіс» Кейсі Харрелл, «ПВХ - найгірший з пластиків». Він є причиною виникнення гормонального дисбалансу, проблем в репродуктивній сфері та різних форм раку. Полівінілхлорид практично неможливо правильно утилізувати. Внаслідок старий матеріал виявляється зазвичай на звалищі з відходами або, того гірше, спалюється з метою вилучення мідних жил і інших цінних компонентів. При його згорянні утворюється вкрай шкідливий канцерогенний діоксин. Звалища і хімічні поховання забруднюють джерела води. Єдиний спосіб правильно утилізувати ПВХ полягає в тому, щоб відправити його в центр небезпечних відходів.

Залишається лише сподіватися, що настане час, коли технології будуть допомагати людині, не завдаючи незворотної шкоди здоров'ю навколишнього середовища.

## ВИСНОВКИ

Векторна графіка застосовується у різноманітних галузях людської діяльності. Найбільші її переваги – це можливість точно задавати координати та виконувати точні геометричні побудовання, можливість вільної трансформації та невеликий розмір файлів. Векторна графіка незамінна, коли важливо зберегти ясність і чіткість контурів, наприклад при створенні графічних заголовків, логотипів, технічних ілюстрацій, схем та якщо зображення планується обробляти на специфічному обладнанні.

В ході роботи розроблений сервіс приймає зображення, яке користувач завантажив через клієнтський додаток та виконує векторизацію цього зображення, будуючи комплексну векторну модель. Спочатку аналізується кількість кольорів та користувачеві пропонується зменшити їх кількість, якщо завантажене зображення містить дуже велику кількість відтінків. Потім до зображення застосовуються операції математичної морфології за для того, щоб підготувати його до тріангуляції. Після побудування тріангуляції та корегування точок, сервіс відправляє результуюче зображення до клієнтського застосунку.

Ефективність системи залежить від кількості кольорів на зображенні. Ця процедура дуже складна, тому її не можна повністю автоматизувати. Базові значення вираховуються автоматично, але потім користувач має змогу редагувати їх за для досягнення бажаного результату.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці, розглянутий можливий вплив на навколишнє середовище.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Розенфельд, А. Распознавание и обработка изображений с помощью вычислительных машин [Текст]: пер. с англ./А. Розенфельд. – М.:Мир, 1972. – 230 с.
- 2) Семенов, О.И. Обработка и отображение информации в растровых графических системах [Текст] / О.И. Семенов, С.В. Абламейко, В.И. Берейшик, В.В. Старовойтов. – Минск: Наука и техника, 1989. —174 с.
- 3) Гамма, Э. Приёмы объектно-ориентированного программирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. – Спб: Питер, 2001. – 388 с.
- 4) Цифровая обработка изображений в информационных системах [Текст] : учебник / И.С. Грузман, В.С. Киричук, В.П. Косых и др. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.
- 5) Bradsky, G. Learning OpenCV[Text] / G. Bradsky, A. Kaehler. – O'Reilly, 2008. – 576р.
- 6) Форсайт, Дэвид А. Компьютерное зрение. Современный поход [Текст] : пер. с англ. / Д.А. Форсайт, Ж. Понс. – М: Издательский дом «Вильямс», 2004. – 928 с.
- 7) Новейшие методы обработки зображений [Текст] / А.А. Потапов, Ю.В. Гуляев, С.А. Никитов и др.; под ред. А.А. Потапова; – М: ФИЗМАТЛИТ, 2008. – 496 с.
- 8) Розенфельд, А. Распознавание и обработка изображений с помощью вычислительных машин [Текст] : пер. с англ. / А. Розенфельд; под ред. Д.С. Лебедева. – М: Мир, 1972, – 453 с.
- 9) Сандерсон, С. ASP.Net MVC Framework с примерами на С# для профкссионалов [Текст] : пер. с англ. / С. Сандерсон. – М: Издательский дом «Вильямс», 2010. – 560 с.
- 10) ГОСТ 12.1.005-88. Міждержавний стандарт. Система стандартів безпеки праці. Загальні санітарно-гігієнічні вимоги до повітря робочої зони
- 11) ГОСТ 12.0.003-74 Небезпечні і шкідливі виробничі фактори. Класифікація
- 12) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту
- 13) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
- 14) ГОСТ 12.1.004-91. Пожежна безпека. Загальні вимоги .

- 15) ДБН В.2.5-67. Опалення вентиляція та кондиціонування.
- 16) ГОСТ 12.1.006-84. Електромагнітні поля радіочастот. Допустимі рівні на робочих місцях і вимоги до проведення контролю
- 17) ДБН В.2.5-28-2006. Природне і штучне освітлення.
- 18) ГОСТ 12.4.009-83. Пожежна техніка для захисту об'єктів. Основні види. Розміщення і обслуговування.
- 19) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою.
- 20) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів
- 21) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.
- 22) Симметрон. Электронные компоненты. Каталог 2004, 2004г. – 192с.

## ДОДАТОК А. Електронні плакати

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТУ ІМЕНІ  
ВОЛОДИМИРА ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

### Магістерська робота Дослідження методів векторизації зображень

Виконав  
Ст.гр. КН-17зм  
Відіборова Оксана Євгенівна

Керівник: проф.Рязанцев О.І.

#### Актуальність проблеми

---

Сфери застосування векторної графіки:

- дизайн, ландшафтний дизайн,
  - архітектура
  - проектування креслень, мікросхем
  - ГІС, карти для GPS-приладів
-

Приклад автоматичної векторизації в  
Adobe Illustrator



Приклад ручної векторизації фотографії



## Особливості векторної графіки

---

### Переваги:

- можливість точно задавати координати
- можливість виконувати геометричні побудови
- невеликий розмір файлів

### Недоліки:

неможливо передати плавні переходи відтінків

---

## Постановка задачі

---

Метою даного дипломного проекту є написання хмарного web-сервісу, який буде надавати функціонал для векторизації кольорових растрових зображень.

Щоб забезпечити високу якість розпізнавання і задовольнити потреби кінцевого користувача сервіс повинен відповідати таким вимогам:

- стійкість до спотворень зображення;
  - можливість коригування параметрів векторизації;
  - можливість зберігати розпізнане зображення.
-

## Первинна обробка зображення

---

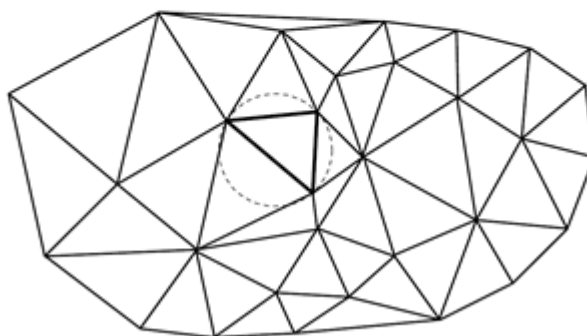
- усунення шумів (медіанна фільтрація)
- зменшення кількості кольорів на зображенні

---

## Триангуляція Делоне

---

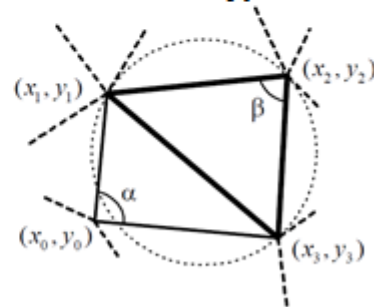
Триангуляція Делоне для безлічі точок  $P$  на площині - це така опукла триангуляція  $DT(P)$ , що жодна точка безлічі  $P$  не всередині описаних навколо трикутників кіл в безлічі  $DT(P)$ . Триангуляція Делоне максимізує мінімальний кут, але не мінімізує максимальний.



## Перевірка умови Делоне

Перевірку умови Делоне можна виконувати наступними способами:

- перевірка через рівняння описаного кола
- перевірка з заздалегідь обчисленої описаної окружності
- перевірка суми протилежних кутів
- модифікована перевірка суми протилежних кутів

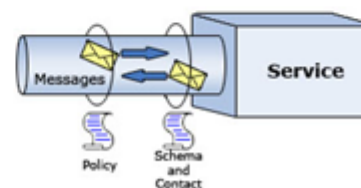


## Реалізація хмарного сервісу

Сервіс-орієнтована архітектура (Service Oriented Architecture, SOA)

Основні принципи:

- модульний підхід;
- всі компоненти розподілені і слабо пов'язані
- компоненти взаємодіють між собою по стандартних протоколах за допомогою інтерфейсів;
- мультиплатформеність (інтерфейс не повинен залежати від платформи)
- масштабованість



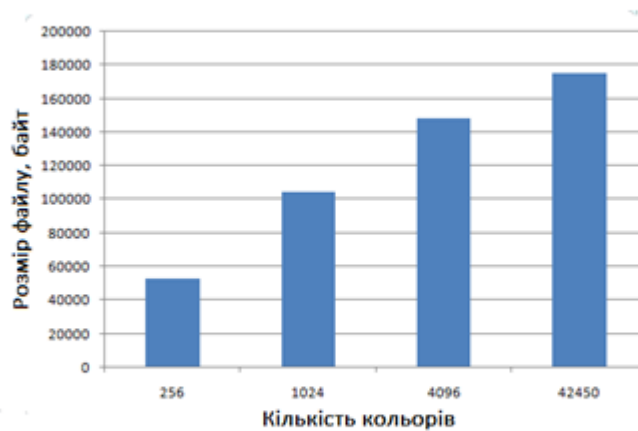
## Використовувані технології

---

- Мова програмування високого рівня C #
- MVC .Net
- Бібліотека Open CV
- Середовище розробки Microsoft Visual Studio 2010

## Результати роботи програми

---



Залежність розміру результуючого файлу від кількості квітів на оригінальному документі

---



## Результати роботи програми



Залежність кількості контурів на результуючому зображенні від кількості квітів на вхідному зображенні

## Висновки

В результаті виконання даної роботи був розроблений хмарний сервіс, який надає функціонал для векторизації кольорових растрових зображень.

Були вирішені наступні завдання:

- проаналізовані сучасні системи векторизації і алгоритми, які вони використовують;
- вивчені особливості архітектури хмарних сервісів (SOA)
- програмно реалізований алгоритм векторизації з використанням триангуляції Делоне.