

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Методи та алгоритми створення безшовного

панорамного зображення із набору фотографій

Виконав: студент 6 курсу, групи СНм-61

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Романський С.В.

(підпис) (прізвище та ініціали)

Керівник проф. Марценюк В.П.

(підпис) (прізвище та ініціали)

Нормоконтроль доц. Дуда О.М.

(підпис) (прізвище та ініціали)

Завідувач

кафедри доц. Боднарчук І.О.

(підпис) (прізвище та ініціали)

Рецензент

(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
доц. Боднарчук І.О.
(підпис) (прізвище та ініціали)
« » 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)
за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)
студенту Романському Степану Володимировичу

1. Тема роботи Методи та алгоритми створення безшовного
панорамного зображення із набору фотографій

Керівник роботи Марценюк Василь Петрович, д.т.н, проф. каф. КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від « 29 » листопада 2023 року № 4/7-1139

2. Термін подання студентом роботи 28.05.2025

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналіз предметної області.

2. Теоретична частина. 3. Практична частина.

4. Охорона праці та безпека в надзвичайних ситуаціях

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

1. Тема роботи. 2. Актуальність. 3. Мета, задачі дослідження. 4. Об'єкт, предмет дослідження
наукова новизна, практичне значення роботи. 5. Методи пошуку ключових точок на

зображеннях. 6. Методи для видалення невірних збігів зі зіставлених характеристик та підбору
функції перетворення. 7. Функціональні вимоги до розроблюваного ПЗ.

8. Засоби реалізації програмної системи. 9. Загальна програмна система алгоритмів та функцій
для створення панорамного зображення. 10. Структура розроблюваної програмної системи.

11. Блок-схема алгоритму суміщення зображень та отримання статистичних даних

12. Блок-схема алгоритму зшивання панорами. 13. Результати експериментів

14. Метод опорних векторів. 14. Результати пошуку та зіставлення ключових точок.

15. Результати методів видалення невірних збігів ключових точок. 16. Результати зшивання.

17. Порівняння з іншими програмними системами. 18. Результати зшивання фрагментів.

19. Результати зшивання повної панорами. 20. Основні результати дослідження.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент каф. МТ		
Безпека в надзвичайних ситуаціях	Клепчик В.М., проректор з АГРБ		

7. Дата видачі завдання _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи	29.11.24	Виконано
2	Аналіз літературних джерел	30.11.24–10.12.24	Виконано
3	Обґрунтування актуальності дослідження	11.12.24–20.12.24	Виконано
4	Аналіз предмету дослідження та предметної області	21.12.24–31.12.24	Виконано
5	Проведення дослідження методів та засобів аналітичного опрацювання даних	01.01.25 –20.01.25	Виконано
6	Оформлення розділу «Аналіз предметної області»	21.01.25–10.02.25	Виконано
7	Оформлення розділу «Теоретична частина»	11.02.25–13.03.25	Виконано
8	Оформлення розділу «Практична частина»	14.03.25–15.04.25	Виконано
9	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	16.04.25–22.04.25	Виконано
10	Нормоконтроль	15.04.25–30.04.25	Виконано
11	Перевірка на плагіат	01.05.25–10.05.25	Виконано
12	Попередній захист роботи	14.05.25	Виконано
13	Захист кваліфікаційної роботи	29.05.25	

Студент

_____ (підпис)

Романський С.В.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Марценюк В.П.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Методи та алгоритми створення безшовного панорамного зображення із набору фотографій // Кваліфікаційна робота освітнього рівня «Магістр» // Романський Степан Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2025 // С. – 75, рис. – 29, табл.– 4, слайдів – 20, додат. – 2, бібліогр. – 42.

Ключові слова: панорамне зображення, детектування особливостей, безшовна панорама, перспектива, гомографія, афінні перетворення

У процесі дослідження проводився огляд алгоритмів та методів для створення панорамних зображень, проаналізовано метрики зіставлення ключових особливостей, пошук найкращого перспективного перетворення та форми проекції для веб-реалізації та зшивання панорамних зображень.

В результаті дослідження було запропоновано та розроблено кілька алгоритмів: поєднання зображень з отриманням статистичних даних, сортування зображень, змішування зображень, колірна корекція, зшивка пари зображень, циліндрична проекція, створення повноцінної панорами, а також візуалізація готової панорами з використанням веб-технологій.

Розроблена та реалізована програмна система здатна зшивати панорамні зображення і має значну гнучкість параметрів та налаштувань, а її архітектура може бути досить легко розширена. Це робить її придатною як для використання як основи прикладного програмного забезпечення, так і як основи для подальшої дослідницької та експериментальної роботи, спрямованої безпосередньо на вдосконалення алгоритму створення панорамних зображень.

ANNOTATION

Methods and Algorithms For Creating a Seamless Panoramic Image From a Set of Photos // Romanskyi Stepan // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61 group // Ternopil, 2025 // P. - 75, Fig. - 29, Table – 4, Slide - 20, References - 42.

Keywords: panoramic image, feature detection, seamless panorama, perspective, homography, affine transformations

In the process of research, a review of algorithms and methods for creating panoramic images was conducted, metrics for comparing key features were analyzed, the search for the best perspective transformation and projection form for web implementation and stitching of panoramic images was carried out.

As a result of the research, several algorithms were proposed and developed: combining images with obtaining statistical data, sorting images, mixing images, color correction, stitching a pair of images, cylindrical projection, creating a full-fledged panorama, as well as visualization of the finished panorama using web technologies.

The developed and implemented software system is capable of stitching panoramic images and has significant flexibility of parameters and settings, and its architecture can be quite easily expanded. This makes it suitable for use both as a basis for application software and as a basis for further research and experimental work aimed directly at improving the algorithm for creating panoramic images.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

BFMatcher (Basics of Brute-Force Matcher) - порівняння шляхом грубої сили.

BRIEF - швидкий евристичний дескриптор, що будується з 256 бінарних порівнянь між яскравостями пікселів на розмитом зображенні.

BRISK (Binary Robust Invariant Scalable Keypoints) - бінарні стійкі інваріантні масштабовані ключові точки.

FAST (Features from Accelerated segment test) – ознаки, отримані з прискореної перевірки сегментів.

FlannBasedMatcher (Fast Library for Approximate Nearest Neighbors based Matcher) - швидка бібліотека для найближчих сусідів.

LMEDS (Least-median of Squares) – метод найменших повних квадратів (алгоритм найменшої медіани квадратів).

LSH (Locality-sensitive hashing) - імовірнісний спосіб зниження розмірності багатовимірних даних.

OpenCV (Open Source Computer Vision Library) – бібліотека комп'ютерного зору з відкритим вихідним кодом.

ORB (Oriented FAST and Rotated BRIEF) – орієнтований FAST, отриманий з прискореної перевірки сегментів та дескриптор BRIEF.

PROSAC (Progressive Sample Consensus) – прогресивний метод однорідної вибірки.

RANSAC (Random Sample Consensus) – стабільний метод оцінки параметрів моделі на основі випадкових вибірок.

SIFT (Scale-Invariant Feature Transform) – масштабно-інваріантне перетворення особливостей.

SURF (Speeded-Up Robust Features) – прискорені стійкі перетворення ознак.

БЗЗ – бібліотека зшивання зображень.

ЗМІСТ

Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Отримання якісних картинок та рекомендації зйомки	11
1.2 Огляд алгоритмів та методів для створення панорамних зображень.....	14
1.2.1 Загальна концепція виявлення особливостей	14
1.2.2 Детектування особливостей.....	15
1.2.3 Відповідність особливостей.....	18
1.3 Висновки до першого розділу.....	21
2 Теоретична частина.....	22
2.1 Обчислення матриці гомографії	22
2.1.1 Алгоритм RANSAC	24
2.1.2 Алгоритм LMEDS	24
2.1.3 Алгоритм PROSAC	25
2.2 Проекції панорамних зображень	25
2.3 Процес отримання безшовної панорами.....	29
2.3.1 Корекція кольору та яскравості.....	30
2.3.2 Методи змішування кольорів	31
2.4 Функціональні та нефункціональні вимоги до розроблюваного ПЗ	34
2.5 Засоби реалізації програмної системи	34
2.6 Висновки до другого розділу.....	37
3 Практична частина	38
3.1 Загальна програмна система алгоритмів для створення панорамного зображення.....	38
3.1.1 Розробка та реалізація алгоритму для суміщення зображень та отримання статистичних даних	38
3.1.2 Розробка та реалізація алгоритму зшивання пари зображень.....	40
3.1.3 Розробка та реалізація функцій циліндричної проекції.....	42
3.1.4 Розробка та реалізація функцій колірної корекції.....	42

3.1.5 Розробка та реалізація функцій змішування зображень	43
3.1.6 Розробка та реалізація функцій вирівнювання та змішування країв панорами.....	45
3.1.7 Розробка та реалізація алгоритму сортування зображень	46
3.1.8 Розробка та реалізація алгоритму зшивання панорамного зображення.....	47
3.1.9 Структура програмної системи для зшивання панорамних зображень	48
3.1.10 Розробка та реалізація візуалізації панорамного зображення.....	49
3.2 Результати експериментів	51
3.2.1 Порівняльний аналіз одержаних результатів ефективності методів пошуку ключових точок	51
3.2.2 Порівняльна демонстрація зшивання зображень із використанням корекції кольору та змішування	55
3.3 Порівняння з іншими програмними системами для зшивання панорамних зображень	57
3.4 Висновки до третього розділу	61
4 Охорона праці та безпека в надзвичайних ситуаціях.....	63
4.1 Закордонний досвід організації охорони праці в ІТ-компаніях.....	63
4.2 Оцінка дії електромагнітного імпульсу на елементи комп'ютерної системи.	68
4.3 Висновки до четвертого розділу.....	70
Висновки	71
Перелік джерел	72
Додатки	

ВСТУП

Актуальність теми. Існують кілька підходів до створення панорамних знімків: технічне рішення, засноване на використанні відеопанорамних камер і автоматична зшивка в додатку, та програмне рішення, що полягає в обробці вихідних кадрів з областями, що перекриваються, котрі зшиваються в один панорамний знімок.

Технічне рішення з використанням відеопанорамної камери в комплексі зі штативом, як єдина система, може бути дуже дорогим і потребує обслуговування та калібрування. При використанні програм з автоматичною зшивкою відсутня можливість замінити кадр загального знімка і необхідно перезнімати всю панораму в цілому. При використанні готового програмного рішення користувачі не мають змоги змінювати параметри зшивки.

Для створення панорамних знімків із серії кадрів сцени, що перекриваються, використовується підхід, заснований на пошуку відповідностей між зображеннями зі збором статистичних даних і обчисленні перспективних перетворень циліндричної форми.

Актуальність розробки обумовлена незалежністю від використання дорогих відеосистем та можливістю отримання статистичних даних у процесі зшивки зображень для науково-дослідних робіт.

Мета дослідження: розробка та реалізація алгоритмів для створення панорамних зображень із наборів фотографій, а також дослідження ефективності в рамках цього завдання різних методів та параметрів вилучення та порівняння особливостей (ключових точок) зображення.

Для досягнення мети, в роботі поставлено та розв'язано **такі задачі:**

- провести огляд алгоритмів обробки та аналізу зображень (в т.ч. панорамних);
- проаналізувати метрики співставлення ключових особливостей зображення;
- відшукати найкраще перетворення перспективи та форми проекції для

програмної реалізації і зшивки панорамних зображень;

- розробити алгоритми створення безшовного панорамного зображення;
- спроектувати структуру програмної системи, розробити її основні компоненти;
- провести експериментальні дослідження та порівняти отримані результати із існуючими системами.

Об’єкт дослідження: безшовне панорамне зображення.

Предмет дослідження: алгоритми та методи генерації безшовного панорамного зображення.

Наукова новизна роботи:

- для реалізації зшивання запропоновано використати афінні перетворення і гомографію, а також функції відповідності RANSAC, LMEDS, PROSAC.

Практичне значення одержаних результатів. Використання як основи прикладного ПЗ для реалізації застосунків, заснованих на комп’ютерному зорі в різних областях застосування, наприклад, віртуальні тури, аерофотозйомка і картографування та ін., а також як основи для подальшої дослідницької та експериментальної роботи, спрямованої безпосередньо на вдосконалення алгоритму створення панорамних зображень.

Апробація. Окремі результати роботи були представлені на наукових конференціях як опубліковані тези:

1. Романський С.В. Детектування особливостей на зображеннях // Інформаційні моделі, системи та технології: Праці XII наук.-техн. конф. Тернопіль, 2024. с. 86.
2. Романський С.В. Проекції панорамних зображень // VIII Міжнародна студентська науково-технічна конференція «Природничі та гуманітарні науки. Актуальні питання» Тернопіль, 2025. с. 194.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Отримання якісних картинок та рекомендації зйомки

Панорама - це зображення, яке зібране із серії послідовно знятих фотографій. Панорамні знімки дозволяють досягти великого кута огляду та отримувати зображення із величезною роздільною здатністю.

Як правило, найчастіше процес зйомки панорамного пейзажу є зніманнями кількох кадрів при повороті камери в нове положення. Для отримання якісних, без слідів зшивки, зображень, знімати кадри для панорами краще з використанням штативу та спеціальної панорамної головки, а повороти робити на один і той же градус.

Камера вертикально встановлюється на панорамну головку, що дозволяє обертати камеру навколо безпаралаксної точки (рис. 1.1).

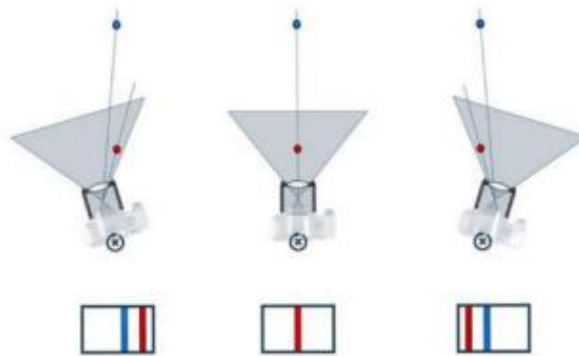


Рисунок 1.1 – Положення камери з використанням штатива

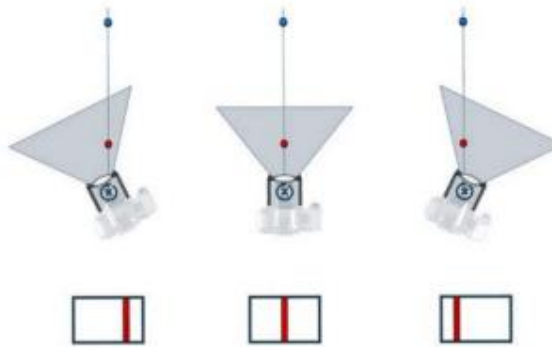
Оскільки часто не вдається зшити зняті з рук знімки в єдину панораму, рекомендується використовувати штатив. Причиною цього є ефект «паралакс», котрий полягає у зміні видимого положення власне об'єкта відносно віддаленого фону у залежності від стану спостерігача.

Для прикладу, на рис. 1.2 видно, що при повертанні камери проходить зміщення об'єктів один щодо одного для ближнього та дальнього планів. Для уникнення цього потрібно повертати знімальний пристрій навколо спеціальної

точки, яка називається нодальною. При обертанні камери навколо такої точки властиво зміщення об'єктів для обох планів не проходитиме [1].



а) усунення об'єктів ближнього та дальнього плану



б) відсутність усунення при обертанні навколо властиво нодальної точки

Рисунок 1.2 – Вплив паралаксу на знімок при повороті камери

Для обертання фотоапарата безпосередньо навколо власне нодальної точки використовуються панорамні головки, які накручуються на штатив і тим самим дозволяють обертати фотоапарат на 360 градусів. Певні моделі таких головок можуть бути пристосовані виключно для зйомки циліндричних панорам (тобто камеру можна обертати тільки по горизонталі). Однак більшість головок дозволяють створювати як сферичні, так і циліндричні панорами. В такому випадку обертання фотоапарата можливе як по горизонталі, так і по вертикалі (рис. 1.3).

Варто також згадати про відсоток перекриття кадрів. Перекриття кадрів є важливим для зшивання панорамних зображень. Воно виконується для того, щоб алгоритм зміг виявити контрольні точки, тобто визначити характерні риси

для сусідніх кадрів. Таким чином, значення перекриття безпосередньо залежить від деталізації та особливостей сцени, що знімається.

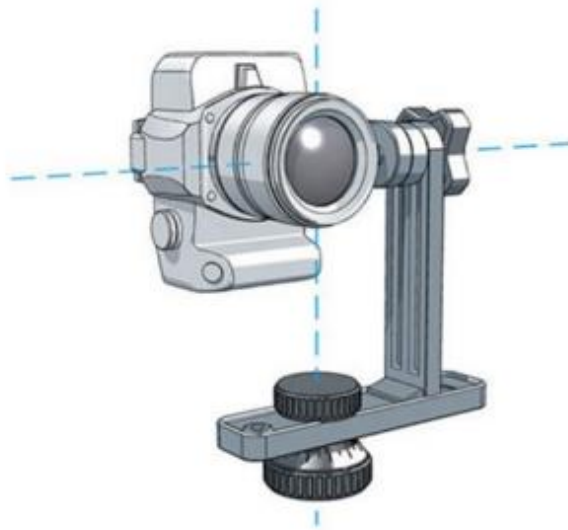


Рисунок 1.3 – Можливість обертання фотоапарата вздовж нодальної точки

Наприклад, якщо знімається єдиний великий об'єкт, то цьому у випадку потрібно зробити більше (до 80%) перекриття, стежачи за тим, щоб найбільш помітні деталі потрапляли в це перекриття, а якщо знімається краєвид, котрий має безліч деталей у кадрі, слід робити відсоток перекриття в межах 10-30%, але зазвичай середньостатистичним відсотком перекриття для впевненої зшивки панорами слід вважати значення в діапазоні від 20 до 40% (рис. 1.4).

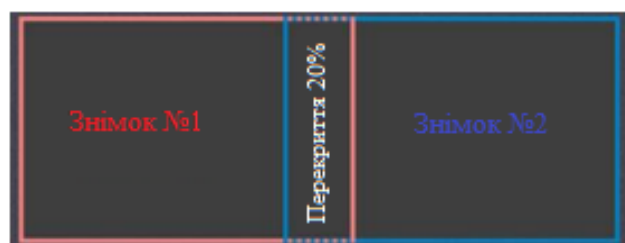


Рисунок 1.4 – Демонстрація варіанта перекриття двох кадрів загальної сцени

Слід зазначити, що відсоток перекриття кадрів визначає їх кількість у круговій панорамі, а сама кількість кадрів для зшивки буде впливати на швидкість роботи алгоритму зшивки та на необхідні обчислювальні потужності.

1.2 Огляд алгоритмів та методів для створення панорамних зображень

1.2.1 Загальна концепція виявлення особливостей

У комп'ютерному зорі існує важливий напрямок - виявлення особливостей (feature detection) на зображеннях, з яких, власне, ключові точки і визначаються як такі. Під цим можна вважати не просто точки, як пару координат, а деякі області на зображеннях, для знаходження яких і існують різні методи та алгоритми.

Власне особливості зображення можуть бути різних видів [2,3]:

- ребра (edges) - межі між двома областями зображень, це можуть бути випадкові контури та стики різних об'єктів; практично часто набір точок, які мають велику величину градієнта;
- кути/точки інтересу, ключові точки (corners / interest points, фактично синоніми). Спочатку продукт аналізу ребер з метою знаходження різких змін у напрямку (тобто кутів), потім були розроблені алгоритми, що не потребують явного обчислення ребер (наприклад, котрі шукають сильне викривлення градієнта зображення). Зазначимо, що знайдені ключові точки зовсім не обов'язково є кутами об'єктів сцен у прямому розумінні, проте вони — особливості, котрі враховуються;
- сфери / ключові області, ключових точок (blobs / regions of interest or interest points). Фактично опис областей зображення. Так само як і кути вважаються ключовими точками (і мають фактичний центр), однак вони мають свої алгоритми розрахунків.

Різні алгоритми знаходження особливостей завжди схожі в одному - як тільки особливість знайдена, відбувається витяг (extraction) невеликої області навколо неї. Результат - дескриптор або векторні особливості. Це стосується і кутів і сфер. Різні алгоритми пошуку особливостей пропонують різні

дескриптори.

Методи зіставлення дескрипторів (matchers) ключових точок на різних зображеннях - наріжний камінь знаходження шаблонів на сцені, можливість комп'ютерного зору для вирішення задачі розпізнавання образів [3].

І ці способи зіставлення можуть бути відмінними для різних алгоритмів пошуку ключових точок та різних дескрипторів.

Існують різні підходи (і їх реалізації) до знаходження ключових точок, це якраз ті методи, котрі розглядаються нижче.

1.2.2 Детектування особливостей

Для виділення ключових точок існує кілька основних методів, які вже реалізовані в різних програмних засобах і широко використовуються на практиці.

Методи, як мінімум, включають алгоритми пошуку, що повертають список ключових точок, а також зазвичай (але не завжди) мають вбудовані алгоритми вилучення дескрипторів. Якщо вбудованих алгоритмів вилучення немає, то існують реалізовані окремо, котрі здатні витягувати із зображення дескриптори певних форматів за знайденими ключовими точками [3].

Основні методи пошуку ключових точок на зображеннях, що мають доступну, готову для використання програмну реалізацію, це SIFT, SURF, ORB, BRISK та AKAZE. Алгоритмічно це різні методи, що тяжіють до різних особливостей зображень.

Ці методи пошуку ключових точок найбільш популярні і поширені (є й інші, зазвичай модифікації вказаних), крім того, важливо те, що їх реалізації є в одній відкритій бібліотеці комп'ютерного зору OpenCV, що робить неактуальним їх реалізацію з нуля (або використання різних джерел/бібліотек) і дає можливість порівнювати їхню роботу в ідентичних умовах.

Відповідно, дана кваліфікаційна робота фокусується на експериментальному порівнянні метрик для зшивки зображень саме цих

методів.

Таким чином, їх порівняння дозволить розмірковувати щодо доцільності застосування методів для реального завдання, їхню конкурентоспроможність один щодо одного.

Очевидно, призначена для експериментів система повинна включати реалізацію даних методів. Далі буде дано короткий опис кожного з них.

SIFT - найстаріший (опублікований в 1999 р.) з обраних методів, але користується неабиякою популярністю. Надає алгоритми пошуку ключових точок за допомогою обчислення сфер (blobs) та вилучення дескрипторів.

Ключові точки тут знаходяться за допомогою обчислення екстремуму масштабу-простору (Scale-space) - зображення згортається фільтрами Гауса в різних масштабах і потім береться різниця послідовних, «розмитих» методом Гауса зображень, а ключові точки беруться як максимум/мінімум різниці гауссіанів, яка відбувається на кількох масштабах. Отриманих таким чином «кандидатів» у ключові точки дуже багато і деякі можуть бути нестабільні, тому потім проводиться локалізація з відсіюванням нестабільних (на цьому етапі також задіяні кути, edges), а вже потім можливе вилучення дескрипторів [4].

SURF – метод, натхненний SIFT. Коротко суть методу можна розписати в такий спосіб. Ключові точки (фактично – ділянки) визначаються максимальною зміною градієнта яскравості. Для цього використовуються матриця Гессе (детермінант - гесіан - досягає екстремуму у тих точках, де є максимальна зміна градієнта яскравості) і інтегральне уявлення зображень (якраз дозволяє швидко рахувати суми яскравостей). Після знаходження для точок витягуються дескриптори, що забезпечують інваріантність масштабу та обертання. Таким чином, потенційно забезпечується знаходження тих самих точок на зображеннях з різним ракурсом/масштабом сцени [5]. Можна сміливо сказати, що це метод знаходячи ключові точки-сфери (blobs) і вмючи отримувати їм дескриптори має архітектуру подібну SIFT.

У ході дослідження було виявлено, що метод SURF деактивовано в актуальній версії бібліотеки OpenCV через те, що він запатентований, тому його використання не є можливим.

ORB - найновіший із вибраних методів (2011р.). Як впливає з розгорнутої назви, це злиття методу FAST і дескриптора BRIEF [6], модифікованих. Спочатку методом FAST знаходяться точки, потім застосовуючи міру кута Харріса знаходяться N верхніх точок серед них. Також використовується піраміда для одержання мультимасштабних особливостей. Далі (на відміну від FAST) тут реалізовано обчислення орієнтації знайдених точок за допомогою обчислення інтенсивності зваженого центроїду від клаптя зі знайденим кутом у центрі, і напрямок від точки кута до центроїду дає орієнтацію. Для поліпшення інваріантності повороту, моменти обчислюються для x і y в середині радіуса r (розмір клаптя). Далі працює дескриптор BRIEF. Через погану роботу дескриптора з орієнтацією, ORB направлятиме його, будуючи матриці повороту для ключових точок.

ORB вважається швидшим, ніж SIFT та SURF, та його дескриптори працюють краще, ніж у SURF [7].

BRISK так само, як і ORB, створений на основі комбінації детектора FAST з оптимізованим дескриптором BRIEF, проте в їхню роботу внесено зміни. Для забезпечення інваріантності методу до масштабування також використовується піраміда Гаусса, в якій проводиться пошук особливих точок з найкращою максимальною інтенсивністю в піраміді за допомогою детектора FAST [8]. Для пошуку кутових точок (corners) по черзі розглядаються околиці 16 пікселів навколо кожного пікселя.

Дескриптор направлений BRIEF представляється як вектор, довжиною 256, котрий містить результати бінарних випробувань навколо властиво особливої точки. Власне в околиці 31×31 піксель порівнюються середні значення яскравостей між x і y де x, y - області 5×5 пікселів.

AKAZE (Accelerated-KAZE) є удосконаленою версією KAZE шляхом використання FED - алгоритму [9] для побудови нелінійної багатомасштабної

піраміди, що працює швидше завдяки відносній простоті реалізації. Застосування нелінійного коефіцієнта масштабування дає змогу збільшити швидкість віднайдення необхідної особливої точки у порівнянні із пірамідою Гауса. Як дескриптор AKAZE використовує дескриптор M-LDB інваріантний до обертання і зміни масштабу. В ньому тести виконуються не для середніх значень усіх пікселів в області, а для заданого числа залежно від розміру, це дає змогу пришвидшити розрахунок дескриптора.

1.2.3 Відповідність особливостей

Після того, як виявили ключові особливості зображень, необхідно вибрати алгоритм для зіставлення їх характеристик.

k-means (метод k -середніх) - метод кластеризації, що робить розбиття деякої заданої вибірки об'єктів (даних) на непересічні підмножини/групи (кластери) на основі близькості їх ознак/значень, який хоче зробити мінімальним сумарне квадратичне відхилення власне точок кластерів від, властиво, центрів таких кластерів [10]:

$$V = \sum_{i=1}^k \sum_{x \in S_i} (x - \mu_i)^2 \quad (1.1)$$

де k – кількість кластерів; S_i – одержані кластери; $i = 1, 2, \dots, k$; μ_i – центри мас усіх векторів x із кластера S_i .

Отже, кожен кластер складається зі схожих між собою об'єктів.

Методу k -середніх часто надається перевага завдяки простоті його реалізації та значній швидкості роботи. Це алгоритм, який ітеративно ділить даний набір пікселів на власне k кластерів точок, котрі наближені максимально до їх центрів, а властиво сама кластеризація проходить шляхом зміщення цих самих центрів. Цей метод є надвичайно чутливим до шуму, котрий здатен значно спотворити одержані результати кластеризації. Є ефективним у випадку, якщо дані поділяються на компактні групи, котрі можна описати сферою.

Далі розглянемо деякі кластерні методи, котрі ґрунтуються на мірі відстані.

Реалізація методу k-середніх з використанням метрики відстані Евкліда порівнює кожен набір дескрипторів з одного зображення з кожним таким набором з іншого зображення і в результаті знаходить найкращу відповідність між особливими точками. Відстань Евкліда обчислюється за класичною формулою (1.2) [11]:

$$D = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1.2)$$

де D – відстань поміж об'єктами, n – кількість характеристик, p і q – порівнювані вектори з i – властивістю.

Метод, заснований на теоремі Піфагора, простий у використанні отриманих результатів, повертає не гарантовано правильне рішення, а лише найкраще із ймовірних. Цей метод може бути витратним щодо обчислювальних ресурсів, оскільки при його використанні може знадобитися повний перебір знайдених ключових особливостей.

Манхеттенська відстань - це відстань, що є середнім для різниць за координатами [12]. Найчастіше, цей така міра призводить до практично таких результатів, як і для раніше розглянутої відстані Евкліда. Тим не менш, особливістю такої міри є зменшення впливу окремих значних різниць (викидів), так як вони не підносяться до степеня два:

$$D = \sum_{i=1}^n (x_i - y_i)^2 \quad (1.3)$$

Відстань Хеммінга є кількістю біт, котрі відрізняються, у двох порівнюваних бінарних векторах (єдина допустима операція - заміна) [11] однакової довжини. Є метрикою відмінності (функцією, котра встановлює відстань в просторі метрик) об'єктів тієї самої розмірності:

$$D_{ij} = \sum_{s=1}^p |x_{is} - y_{js}| \quad (1.4)$$

де p – розмірність, x та y – двійкові послідовності (вектора) довжини n .

Відстань Хеммінга просто реалізується і швидко працює (тобто має високу продуктивність) і зараз достатньо широко застосовується для різних задач, як то відшукування близьких дублікатів або розпізнавання образів, проведення класифікації документів, корегування помилок, детектування вірусів і т.д. Відстань Хеммінга також має помітний недолік, а саме вимогу однакової довжини рядків. Таким чином, ця метрика оптимальна для розрахунку відстані редагування з урахуванням таких спотворень, як заміна та транспозиція, проте не підходить при вставках та видаленнях.

LSH - ймовірнісний спосіб зниження розмірності багатовимірних даних. Його ідеєю є підбір хеш-функцій для деяких вимірювань таким чином, щоб схожі об'єкти з високим ступенем ймовірності потрапляли до одного умовного «кошика» [13]. Даний метод дозволяє будувати структуру для швидкого наближеного (імовірнісного) пошуку n -вимірних векторів, «схожих» на шаблон, котрий шукається. Серед наявних на сьогодні наближених алгоритмів пошуку найближчих сусідів саме LSH є, ймовірно, найпопулярнішим.

KD-Tree (K-вимірне дерево) [14], спеціальна структура даних, яка дозволяє розбити K-вимірний простір на «менші частини», за допомогою перерізу цього самого простору гіперплощинами ($K > 3$), площинами ($K = 3$), прямими ($K = 2$), і у випадку одномірного простору - точкою (виконуючи пошук у такому дереві, буде щось схоже на бінарний пошук). Логічно, що таке розбиття, зазвичай, використовують для звуження діапазону пошуку в K-вимірному просторі. Наприклад, пошук ближнього об'єкта (вершини, сфери, трикутника тощо) до точки, проектування точок на 3D сітку, трасування променів (активно використовується в Ray Tracing) тощо. При цьому об'єкти простору поміщаються в спеціальні паралелепіпеди - bounding box -и (bounding box -ом назвемо такий паралелепіпед, який описує вихідну множину об'єктів

або сам об'єкт, якщо ми будуємо bounding box лише для нього. У точок як bounding box -а береться bounding box з нульовою площею поверхні та нульовим об'ємом), сторони яких паралельні до осей координат.

1.3 Висновки до першого розділу

У цьому розділі виконано аналіз предметної області дослідження. Зокрема наведено особливості одержання якісних картинок і рекомендації для проведення зйомки.

Здійснено огляд алгоритмів і методів для формування панорамних зображень, описано загальний підхід до детектування особливостей та їх відповідності.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Обчислення матриці гомографії

Гомографією називається зворотне перетворення деякої реальної проєктивної площини на проєктивну площину, що переводить прямі лінії в прямі лінії. Може бути матрицею 3 x 3 з вісьмома ступенями свободи:

$$H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}, \tilde{X} \propto HX \quad (2.1)$$

де H - матриця гомографії, $X = [x, y, 1]^T$ - довільний піксель вихідного зображення, $\tilde{X} = [\tilde{x}, \tilde{y}, \tilde{w}]^T$ - координати пікселя після проектування. Після переходу від однорідних координат результатом будуть нові координати $(\hat{x}, \hat{y})$ вихідної точки зображення (x, y) , що у загальному вигляді можна записати так:

$$\begin{cases} \hat{x} = \frac{ax+by+c}{gx+hy+1} \\ \hat{y} = \frac{dx+ey+f}{gx+hy+1} \end{cases} \quad (2.2)$$

Гомографічна матриця H визначається вісьмома елементами, з яких a і e відповідають за масштабування координат по осях OX і OY відповідно, b і d - за зсув по осях OX і OY (разом з a , e впливають на поворот), c і f - за зміщення по осях OX і OY , а g і h - на зміну перспективи.

Матриця гомографії визначає перспективне перетворення другого зображення стосовно першого, еталонного зображення. Далі на основі отриманої функції перетворення проводиться трансформація зображення для вирівнювання другого зображення по відношенню до першого. Вже трансформована версія другого зображення зі свого боку накладається перед

еталонним зображенням, тобто відбувається зшивання зображень.

Перетворення перспективи – це найскладніша форма двовимірних перетворень площини. Щодо зростання складності, перетворення можна розташувати в наступному порядку [15]: перенесення, Евклідове перетворення, перетворення подоби, афінне перетворення та перспективне перетворення (рис. 2.1). Якщо гомографія обчислюється в невеликій області зображення або зображення було отримано з великою фокусною відстанню, то афінна гомографія є більш оптимальною моделлю зшивки зображення

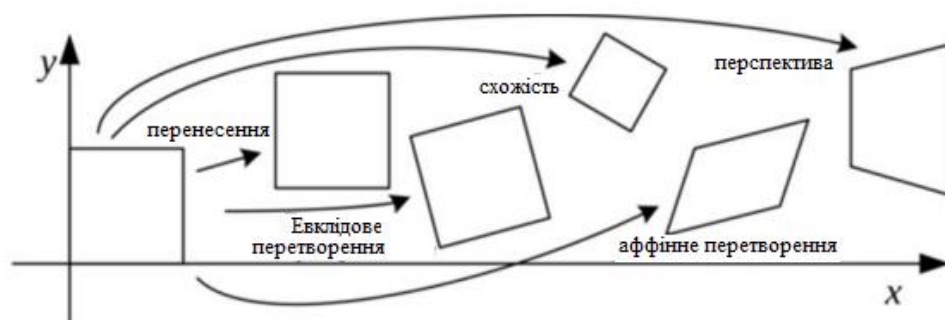


Рисунок 2.1 – Двовимірні перетворення площини

У кваліфікаційній роботі було вирішено реалізувати алгоритм поетапної зшивки зображень із визначенням матриці гомографії та афінними перетвореннями.

Невірних збігів (outliers) неможливо уникнути повністю на етапі зіставлення ознак, таким чином для точного припасування моделі перетворення потрібна ще додаткова фаза виключення невідповідних точок. Після того, як початковий набір відповідностей ознак було обчислено, потрібно знайти підмножину, яка забезпечить вирівнювання з високою точністю.

RANSAC [16], LMEDS [17] та PROSAC [18] - це деякі з надійних імовірнісних методів, котрі можуть бути застосовані (і часто використовуються на практиці) для видалення невірних збігів зі зіставлених характеристик та підбору функції перетворення (у термінах матриці гомографії).

2.1.1 Алгоритм RANSAC

Вважається ітераційним методом формування надійних оцінок атрибутів матмоделі на основі набору даних, які, як відомо, містять викиди.

Цей алгоритм є недетермінованим у тому значенні, що RANSAC видає прийнятне рішення тільки із якоюсь імовірністю, причому сама ця ймовірність зростає зі збільшенням кількості ітерацій.

RANSAC вперше був опублікований у 1981 році Фішлером та Боллесом. Головним його припущенням є те, що в даних містяться «викиди», тобто є дані, розподіл котрих може бути поясний при допомозі якоїсь математичної моделі, і «викиди», котрі є даними, які цій моделі не відповідають. Ключові точки, які є «викидами», можуть розглядатися як результати, викликані шумом, помилковими вимірами або просто невірними даними. У методі також використовується той факт, що для набору атрибутів є певна процедура, котра здатна виконати оцінку параметрів моделі, котра оптимально пояснює чи, властиво, відповідає він цим даним. Алгоритм здатний обробляти практично будь-яке співвідношення викидів, однак необхідний спеціальний параметр, поріг, щоб відрізнити «хороші» ключові точки від викидів.

Алгоритм RANSAC часто застосовується у сфері комп'ютерного зору, зокрема для одночасного вирішення проблеми відповідності та оцінки фундаментальної матриці, що відноситься до пари стереокамер.

2.1.2 Алгоритм LMEDS

Є одним із найбільш ранніх типів стійкої регресії. Його можна назвати «регресія методом найменших квадратів», і його ідея полягає у мінімізації суми квадратів лишків.

У окремому випадку простої регресії це відповідає пошуку найвужчої смуги, що покриває половину спостережень, котра зменшує вплив лишків. Згідно з Венейблсом і Ріплі (1999), цей алгоритм мінімізує медіанне значення упорядкованих квадратів лишків для отримання коефіцієнта регресії.

Для алгоритму LMEDS не потрібне граничне значення, проте він працює правильно лише за наявності більше 50% міток. Метод дуже простий із обчислювальної складової. Оцінювання коефіцієнтів регресії за певних передумов мають низку оптимальних властивостей.

2.1.3 Алгоритм PROSAC

Структура алгоритму PROSAC аналогічна до RANSAC. Спочатку шляхом випадкової вибірки генеруються гіпотези. Вибірки, на відміну від RANSAC, беруться з деякої підмножини даних із найвищою якістю, а не з усіх даних. Розмір набору для створення гіпотез поступово збільшується. Тому на ранньому етапі досліджуються саме ключові особливості, які з більшою ймовірністю та із найменшою кількістю шуму.

Фактично PROSAC призначений для відмальовування тих самих характерних особливостей, що і RANSAC, але у іншому порядку. Гіпотези перевіряються на відповідність усім даним. Як і в RANSAC, алгоритм завершується, коли ймовірність існування рішення, яке на даний момент було б кращим, ніж найкраще, стає низькою (менше 5%).

2.2 Проекції панорамних зображень

Після визначення матриць гомографії всіх вхідних зображень важливим питанням є вибір системи координат простору для розміщення перетворених зображень.

Починаючи роботу з панорамними зображеннями, необхідно отримати деяке уявлення про різні існуючі проекції панорамних зображень і про те, як вони впливають на зображення [19]:

1. Прямолінійна проекція. Визнаючи, що навколишнє середовище навколо вас набуває форми сфери, прямолінійна проекція використовується, коли фотографується лише частина цієї сфери (рис. 2.2). Потім ця частина

сфери проектується на плоску поверхню. Часто використовується для склеювання невеликої кількості зображень: добре адаптована для часткових панорам та мозаїчних зображень.



Рисунок 2.2 – Прямолінійна проекція

2. Циліндрична проекція. Використовуватиметься, коли на панорамі немає даних про зеніт або надир (полюси сфери) - зазвичай це однорядне зображення. Це проекція, котра застосовується для циліндричних панорам і деяких більших часткових панорам, які виглядають надто спотвореними у прямолінійній проекції. Вона проектує частину сфери (циліндр) на плоску поверхню (зображення) (рис. 2.3).



Рисунок 2.3 – Циліндрична проекція

3. Сферична (еквідистантна) проекція. Ця проекція використовується для проектування усієї сфери на плоску поверхню (рис. 2.4). Результатом буде рівнопрямокутне зображення із співвідношенням сторін 2:1. Ширина чітко вдвічі більша за висоту. Це логічно, тому що охоплює 360° по горизонталі та 180° по вертикалі. Можна побачити, що у цій проекції верхня та нижня частини зображення виглядають дуже спотвореними, і це нормально. Ця проекція використовується як джерело для повної сферичної панорами.



Рисунок 2.4 – Сферична проекція

4. Проекція Меркатора. Цю проекцію можна розглядати як похідну від рівнопрямокутної проекції без спотворень у зеніті та надирі (рис. 2.5). Карти світу будуються у цій проекції. Для панорам це може бути перевагою, якщо потрібно, наприклад, надрукувати зображення сферичної панорами або використовувати зображення на веб-сайті. Але цю проекцію не можна використовувати для створення інтерактивних панорам.



Рисунок 2.5 – Проекція Меркатора

5. Кубічна проекція. Ця проекція використовується як розбиття рівнопрямокутного зображення (сферичної панорами) (рис. 2.6). Принцип полягає в тому, що вся сфера перетворюється на 6 граней куба - передню, праву, задню, ліву, zenit та надир. Кожне із 6 квадратних зображень має прямолінійну проекцію.



Рисунок 2.6 – Кубічна проекція

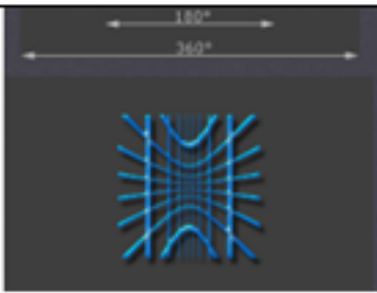
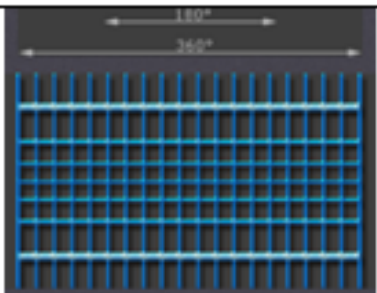
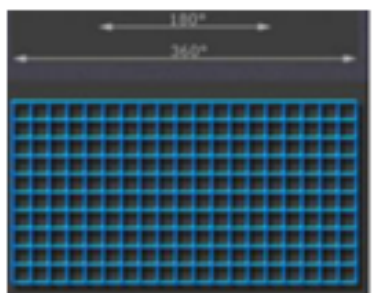
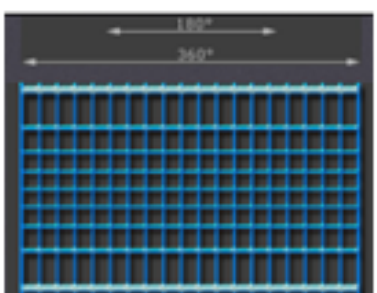
Ця проекція є дуже корисною, коли потрібно відредагувати зображення сферичної панорами. Не завжди легко редагувати деформовані зображення на прямокутному зображенні, а також неможливо редагувати повністю спотворені zenit і надир. Розбивши це спотворене зображення на 6 плоских зображень (прямолінійна проекція - це зображення з меншим спотворенням), можна легко їх відредагувати.

Кожен тип проекції намагається мінімізувати один тип спотворення за рахунок інших.

Кожна з проекцій має свої плюси і мінуси [19], які наведені у табл. 2.1.

Мають місце також і додаткові проекції, наприклад, синусоїдальна стенографічна, але при створенні панорами вони використовуються дуже рідко.

Таблиця 2.1 – Типи проєкцій зображень, їх переваги і недоліки

Тип	Розгортка	Переваги	Недоліки
Прямокутна		Відображає прямі лінії у тривимірному просторі у прямі лінії на плоскій двовимірній сітці	Непридатні для зображень з великим вертикальним кутом зору
Циліндрична		Зберігають більш точні відносні розміри об'єктів	Непридатні для зображень з великим вертикальним кутом зору
Прямолінійна		Відображає прямі лінії у тривимірному просторі у прямі лінії на плоскій двовимірній сітці	Істотно перебільшують перспективу у міру збільшення кута огляду, що призводить до видимого завалу об'єктів до кордонів кадру.
Меркатора		Близько співвідноситься з циліндричною та еквідистантною проєкціями; забезпечує менше розтягування по вертикалі і ширший кут зору, ніж циліндрична проєкція	Більш сильне викривлення ліній по горизонталі

2.3 Процес отримання безшовної панорами

Найчастіше при зшиванні зображень в панорами на підсумковому зображенні можна помітити ефект мозаїки, отриманий в результаті зміни освітленості і відтінків через різне положення камери. Для кращого сприйняття

панорами вдаються до попередньої корекції колірних характеристик вихідних зображень та методів корекції видимого шва між зображеннями.

2.3.1 Корекція кольору та яскравості

Існують кілька алгоритмів корекції кольорів зображень. У кваліфікаційній роботі розглядається метод, що ґрунтується на перетворенні колірної моделі RGB у простір кольорів CIELAB [20]. Після того як у просторі відбулися зміни по каналах кольору, кольори конвертуються назад в RGB.

Колірний простір CIELAB визначено Міжнародною комісією з освітлення (скорочено CIE) у 1976 році.

$L^* a^* b^*$ виражає колір як три значення: L^* - світлість (лежить в діапазоні 0 - 100, від найбільш темного до найбільш світлого), параметри a^* і b^* для чотирьох унікальних кольорів людського зору: червоний, зелений, синій та жовтий.

Простір CIELAB тривимірний та охоплює весь спектр сприйняття людини (рис. 2.7).

У випадку 8-бітових і 16-бітних зображень R, G і B перетворюються на формат з плаваючою комою і масштабуються, щоб відповідати діапазону від 0 до 1.

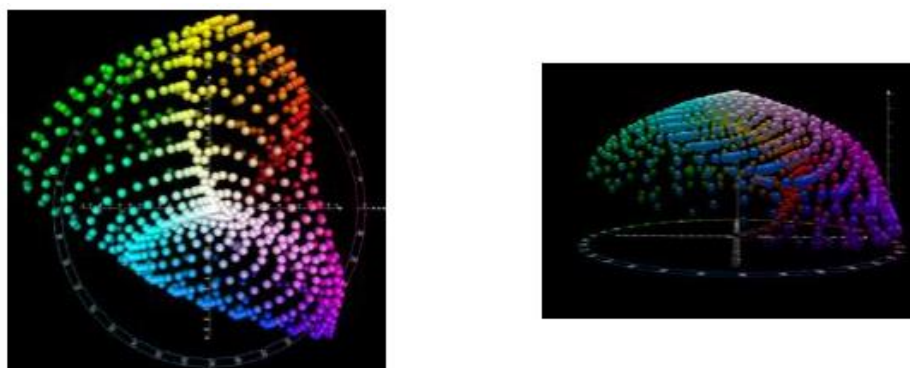


Рисунок 2.7 – Колірний простір CIELAB

Перетворення RGB на $L^*a^*b^*$:

$$\begin{aligned}
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &\leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\
X &\leftarrow X / X_n, \text{ где } X_n = 0.950456 \\
Z &\leftarrow Z / Z_n, \text{ где } Z_n = 1.088754 \\
L &\leftarrow \begin{cases} 116 * Y^{\frac{1}{3}} - 16 & \text{для } Y > 0.008856 \\ 903.3 * Y & \text{для } Y \leq 0.008856 \end{cases} \\
a &\leftarrow 500(f(X) - f(Y)) + \text{delta} \\
b &\leftarrow 200(f(Y) - f(Z)) + \text{delta}
\end{aligned} \tag{2.1}$$

де

$$f(t) = \begin{cases} t^{1/3} & \text{для } t > 0.008856 \\ 7.787t + 16/116 & \text{для } t \leq 0.008856 \end{cases}$$

та

$$\text{delta} = \begin{cases} 128 & \text{для 8 бітних зображень} \\ 0 & \text{для зображень із плаваючою крапкою} \end{cases}$$

Діапазони значень на виході $0 \leq L \leq 100$, $-127 \leq a \leq 127$, $-127 \leq b \leq 127$.

Через механізм встановлення кольору в $L^*a^*b^*$ існує варіант впливати на яскравість зображення, його контраст та колір окремо. Для багатьох випадків це дає змогу значно пришвидшити обробку зображень. Властиво $L^*a^*b^*$ дозволяє вибірково впливати на окремі кольори у об'єкті, посилити колірну суміш, цінною є і такі можливості, які даний колірний простір представляє з метою боротьби із шумами на фото.

2.3.2 Методи змішування кольорів

Метод альфа-змішування. Дуже простий у частині ідеї та реалізації метод [21,22]. Базується на поняттях прозорості зображення (альфа) та розмитті країв (feathering). На рис. 2.8 показано приклад того, як ефект безшовності досягається за рахунок комбінації цих засобів.

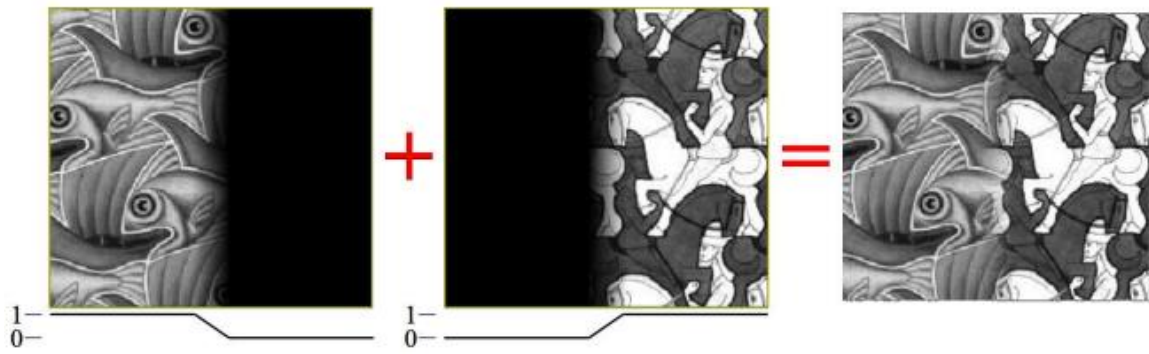


Рисунок 2.8 – Альфа-змішування

Також подібне змішування можна показати через формулу (2.2):

$$IB = a * IL + (1 - a) * IR \quad (2.2)$$

де IB – змішане зображення, IL – ліве зображення, IR – праве, a – прозорість.

Метод багатосмугового змішування з використанням пірамід Гауса та Лапласа. Цей метод ефективніший, хоч і витратніший за продуктивністю, ніж альфа-змішування. Він будується на побудові пірамід Гауса і Лапласа для зображень і подальшого збирання їх підсумкового зображення [23].

Піраміда Гауса – це послідовність масштабованих версій одного зображення (рис. 2.9). Довжина цієї послідовності – кількість рівнів, починаючи від нульового (початкове зображення). Кожен наступний рівень є зображення, зменшеним вдвічі за своїми розмірами щодо попереднього. Тобто, зображення першого рівня має половину роздільної здатності від вихідного, другого – чверть, третього – восьму частину тощо. Сама по собі піраміда Гауса також часто використовується для створення ефекту розмиття.

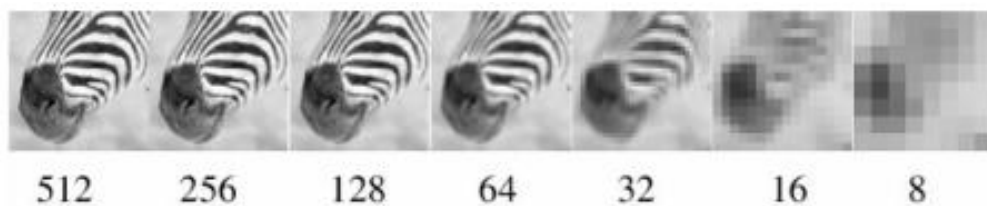


Рисунок 2.9 – Піраміда Гауса

Піраміда Лапласа є похідною від піраміди Гауса і має схожу структуру. Однак, зображення, що містяться в ній, по суті, оброблені високочастотним фільтром і тому відображають лише деталі і грані зображення (рис. 2.10).

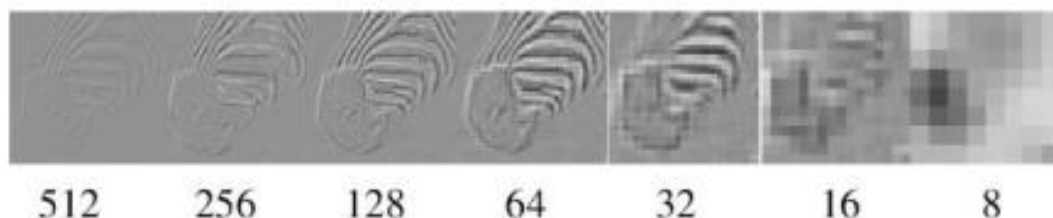


Рисунок 2.10 – Піраміда Лапласа

Для того, щоб змішати два зображення, для обох будуються піраміди Лапласа розміром N рівнів, також будується піраміда Гауса (з тим же числом рівнів) для спеціальної маски, що представляє регіон змішання (рис. 2.11) [24].

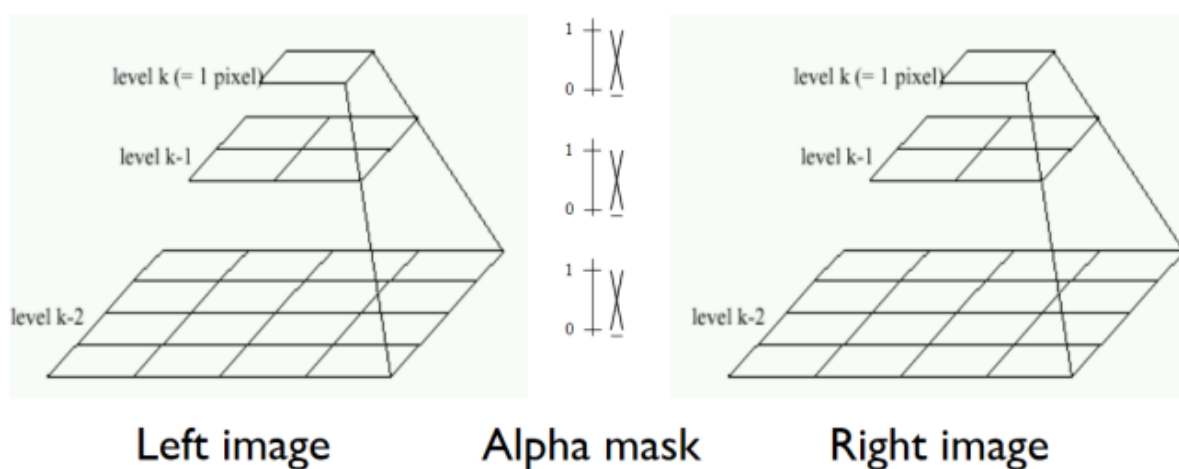


Рисунок 2.11 – Піраміди для змішування

Дві піраміди Лапласа (LA та LB) піддаються змішанню на кожному рівні, використовуючи відповідні рівні з піраміди маски (GR). Результатом буде змішана піраміда, де результуючий рівень, що є комбінацією, обчислюється за формулою (2.3):

$$LS(i,j) = GR(I,j) * LA(I,j) + (1 - GR(I,j)) * LB(I,j) \quad (2.3)$$

Завдяки можливості відновити повноцінне зображення з пірамід Лапласа і Гауса, з піраміди, що утворилася, відновлюється вже змішане зображення, причому якість змішування тим вищою, чим більше задано рівнів.

2.4 Функціональні та нефункціональні вимоги до розроблюваного ПЗ

На основі аналізу галузі генерації панорамних зображень були сформульовані такі функціональні вимоги до ПЗ:

- автоматизована робота з директоріями, що містять серії зображень, що перекриваються в довільній кількості;
- можливість тонкого встановлення із великою кількістю параметрів;
- опціональна можливість застосування циліндричного спотворення до зображень;
- опціональна можливість застосування корекції кольорів;
- опціональна можливість змішування зображень;
- можливість запису підсумкової панорами у файл;
- можливість зберігати статистичні дані для подальшого аналізу;
- можливість тривимірної демонстрації зшитої панорами у циліндричній проекції.

Нефункціональні, але важливі вимоги, такі:

- застосування алгоритмів як командного рядка;
- модульна архітектура з можливістю швидкої побудови високорівневих алгоритмів із більш низькорівневих.

2.5 Засоби реалізації програмної системи

Основою реалізації є OpenCV, в якій є усі необхідні можливості, такі як робота із зображеннями та реалізація методів пошуку особливих точок [25]. Вона написана мовами C і C++, але для багатьох інших мов були розроблені

спеціальні бібліотеки-обгортки для того, щоб можна було користуватися її можливостями.

Як мову розробки необхідних функцій, класів та алгоритмів обрано Python, яка здатна працювати з OpenCV за допомогою пакету OpenCV-Python.

Як середовище розробки вибраний редактор Visual Studio Code з можливостями IDE: спеціальний додаток для мови Python, що включає автодоповнення, статичний аналіз та підсвічування синтаксису, дозволяє ефективно вести розробку цією мовою програмування.

Зокрема, для створення алгоритму використовувалися:

- `opencv_core` – головна функціональність. Містить основні структури і обчислення;
- `opencv_imgproc` - обробка зображень (лінійна та нелінійна фільтрація, геометричні перетворення (масштабування, афінні та перспективні перетворення) тощо);
- `opencv_features2d` - призначений для пошуку ключових точок на зображеннях, зіставлення дескрипторів та виведення особливостей на зображеннях;
- `opencv_highgui` - ввід/вивід фото та відео зображень;
- `opencv_calib3d` – базові геометричні алгоритми стереоскопічної візуалізації, методи калібрування звичайної та стереоскопічної камери, оцінки розташування об'єктів, алгоритми пошуку відповідностей для стереозображень, елементи 3D реконструкції.

У цій бібліотеці є різні алгоритми для обробки зображень та комп'ютерного зору, в т.ч. SIFT, SURF, ORB, BRISK та AKAZE. Вони відносяться до основних детекторів ознак, інваріантних при масштабованості, обертанні та афінних перетвореннях, кожен з яких володіє позначеним дескриптором ознак і має свої переваги і недоліки.

Після виявлення ключових особливостей виконується зіставлення характеристик двома методами зіставлення, що реалізуються через OpenCV: `BFMatcher` та `FlannBasedMatcher`.

BFMatcher приймає дескриптор одного об'єкта в першому наборі і зіставляється з іншими об'єктами в другому наборі з використанням деякого розрахунку відстані, потім повертається найближчий.

FlannBasedMatcher містить набір алгоритмів, оптимізованих для швидкого пошуку найближчого сусіда у великих наборах даних та для багатовимірних об'єктів. Він працює швидше, ніж BFMatcher для великих наборів даних [26]. Іншими словами, FLANN набагато швидше, ніж BFMatcher, але він знаходить лише приблизного найближчого сусіда, що є добрим збігом, але не обов'язково найкращим.

На рис. 2.12 зображено схематично метрики, що оцінюють відповідність між двома аналізованими дескрипторами ключових точок описаними вище методами зіставлення особливостей у бібліотеці OpenCV.

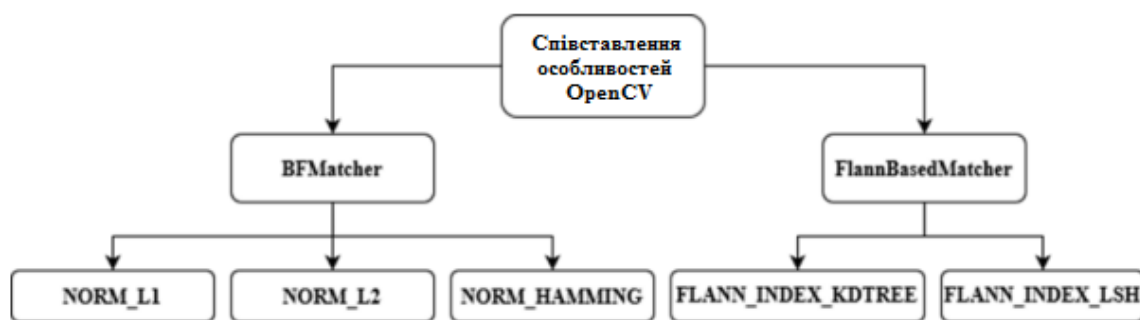


Рисунок 2.12 – Метрики розрахунку відстані між ключовими точками у методах BFMatcher та Flann

BFMatcher використовує функції дистанції L1 -норми (Манхеттенська відстань) або L2-норми (відстань Евкліда) та відстані Хеммінга.

FlannBasedMatcher використовує алгоритм, який застосовує пріоритетний пошук до ієрархічних дерев k -середніх і алгоритм локального хешування (LSH) для пошуку на основі FLANN.

RANSAC, LMEDS та PROSAC, описані раніше, доступні у бібліотеці OpenCV.

Перетворення перспективи обчислюється за матрицею гомографії та афінним перетворенням.

Таким чином, вивчивши всі алгоритми та методи для створення панорамних зображень було вирішено реалізувати систему з кількох алгоритмів:

- алгоритм для суміщення зображень та отримання статистичних даних;
- алгоритм поетапної зшивки пар зображень з визначенням матриці гомографії та афінними перетвореннями з реалізацією функцій циліндричної проекції, корекції кольору та змішування кольорів, функцій вирівнювання та змішування країв панорами;
- алгоритм сортування вхідних зображень.

2.6 Висновки до другого розділу

У розділі розглянуто процес обчислення матриці гомографії із використанням алгоритмів RANSAC, LMEDS та PROSAC. Розглянуто основні проекції панорамних зображень. Досліджено одержання безшовної панорами із застосуванням корекції кольорів і яскравості та методів колірної змішування.

Представлено вимоги до розроблювального програмного продукту (функціональні та нефункціональні), а також програмні засоби розробки.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Загальна програмна система алгоритмів для створення панорамного зображення

Реєстрація зображень - це процес зіставлення, суміщення та накладання двох або більше зображень сцени, знятих з різних точок огляду. Він широко застосовується у різноманітних програмах для машинного зору. Реєстрація зображення складається з таких етапів: виявлення та вилучення особливостей, пошук відповідностей ознак, визначення «хороших» точок та невірних збігів, трансформація зображення, реконструкція та зшивка зображення.

Час і точність суміщення зображень на основі ознак, в основному, залежать від обчислювальної ефективності та надійності обраного методу знаходження ознак, тому початковим завданням роботи є розробка алгоритму для поетапного поєднання зображень, що перекриваються, з різними метриками обраних методів.

3.1.1 Розробка та реалізація алгоритму для суміщення зображень та отримання статистичних даних

Підсумковий алгоритм поєднання зображень та отримання статистичних даних наведений на рис. 3.1.

Варто навести етапи виконання алгоритму:

1. Вказує шляхи до зображень. Створюється директорія для результатів тесту, з назвою поточної дати та часу.

2. Цикл за детекторами ключових точок. За всіма методами пошуку ключових точок – AKAZE, BRISK, SIFT, ORB. На початку тіла циклу зчитуються файли зображень, що перевіряються, створюється піддиректорія за назвою методу, в яку будуть збережені всі результати, а потім створюється об'єкт детектора, за допомогою якого в прочитаних зображеннях відшукуються

ключові точки і витягуються дескриптори. Для загальної статистики методом вважається витрачене на обчислення час і кількість точок.

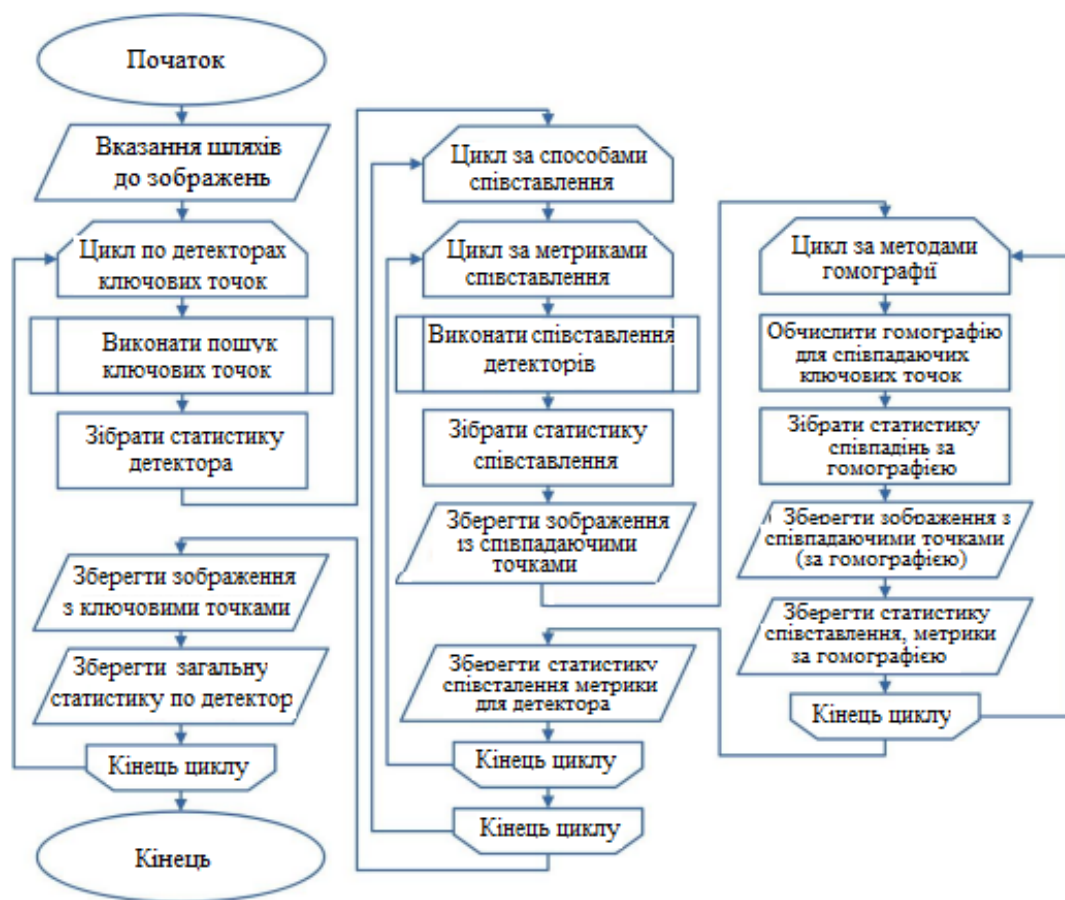


Рисунок 3.1 – Блок-схема алгоритму суміщення зображень та отримання статистичних даних

3. Цикл за способами зіставлення. Цикл проходить за можливими методами зіставлення ключових точок – BFMatcher, FLANN (двох видів). Відразу на початку його тіла реалізований третій цикл - за метриками, таким як NORM_L1, NORM_L2, NORM_HAMMING. У випадку з FLANN метрики не використовуються, тому у третього циклу буде тільки одна ітерація.

Буде виконано створення чергової піддиректорії (вже в рамках методу пошуку ключових точок) порівняння дескрипторів ключових точок обох зображень із поточною комбінацією методу порівняння та метрики (час цієї операції також вимірюється). Для статистики також береться кількість відповідних («хороших») збігів, а директорію зберігається комбіноване

зображення із зіставленими «хорошими» точками.

4. Цикл з методів гомографії. Реалізовано також за допомогою циклів кілька методів обчислення гомографії. Ця операція дозволяє додатково відфільтрувати невідповідні точки. Зберігається нова статистика для кожного методу обчислення гомографії (кількість та відсоток точок, підтверджених гомографією) та нове комбіноване зображення з цими точками. Саме це дозволить порівнювати відносну ефективність різних методів.

Для кожного методу обчислення гомографії зберігається нова статистика (кількість та відсоток точок, підтверджених гомографією) та нове комбіноване зображення з цими точками.

3.1.2 Розробка та реалізація алгоритму зшивання пари зображень

Цей алгоритм реалізований у вигляді спеціальної функції та набору невеликих класів та методів в окремому Python -модулі «panorama.py», призначений для використання у високорівневих алгоритмах зшивання.

Невеликий клас StitchParams не містить методів, лише статичні поля (тобто, не передбачається створення екземплярів даного класу). Він призначений для зміни налаштувань зшивання у межах високорівневих алгоритмів шляхом простого присвоєння інших значень полям.

Поля RATIO і MINDIST призначені для зіставлення наборів ключових точок, будучи критеріями близькості дескрипторів, що порівнюються. DETECT_OPTIMIZE і OPTIMIZE_BY_LAST відповідають за можливість дещо прискорити зшивання великих панорам шляхом пошуку ключових точок тільки на частині зображення (перший параметр – що береться половина зображення, другий – що береться за довжиною останнього пришитого). HOMOGRAPHY_FUNC – обрана функція відповідності (RANSAC, LMEDS, PROSAC). MODE - обраний метод перетворення, афінні чи гомографія. SAVE_PER_STEP - прапор для збереження проміжних зображень на кожному кроці (призначений безпосередньо для алгоритму побудови панорами).

CYL_WARP і CYL_F - параметри циліндричної проекції (перший - використовувати її, другий - радіус віртуального циліндра). CORR_COL і CORR_CHAIN - обраний метод корекції кольору (або його відсутність) і прапор для запам'ятовування кольорів поточного обробленого зображення (щоб кожне наступне коригувалося за попереднім скоригованим).

Безпосередньо функція зшивання крім обліку вищеописаних параметрів приймає кілька своїх - пару зображення для зшивання, об'єкти детектора особливостей, зіставлення особливостей і змішування і опціональну ширину останнього раніше пришитого зображення (для оптимізації при зшиванні масштабних панорам).

На наступному етапі проводиться розширення області зображення (для уникнення в майбутньому втрати ділянок), оптимізація області, в якій буде відбуватися пошук ключових точок на одному із зображень (якщо таке налаштування задане) і безпосередньо пошук ключових точок та витяг дескрипторів.

Далі вилучені особливості порівнюються, фільтруються «хороші» збіги. І проводиться оцінка гомографії (або афінних перетворень), та застосування цієї трансформації до зображення, що зшивається.

Заклучний етап - збереження розмірів трансформованого зображення (для можливої оптимізації) і потім змішування зображень за допомогою переданого об'єкта, який може бути налаштований на певний спосіб, так і просто грубо накласти зображення один на одного, за необхідності. Підсумкове зображення є вихідним значенням функції.

Крім того, з розрахунком на зшивання панорам від центру з рухом ліворуч і праворуч, зроблено функцію для протилежного напрямку руху від центру, яка інвертує зображення необхідним чином, викликає вищеописану функцію, і знову інвертує отриманий від неї результат.

3.1.3 Розробка та реалізація функцій циліндричної проекції

Функції, котрі дозволяють застосувати циліндричне перетворення до переданого зображення, необхідні для зшивання розрахованих на циліндричну проекцію панорам. Реалізована у межах модуля «рапогата.ру».

Перша функція `cylindricalWarp()` в як параметр приймає зображення і матрицю для його перетворення. Використовуючи матрицю, функція перетворює зображення та повертає отриманий результат.

Друга функція `cilWarpAuto()` є «обгорткою» першої і призначена для формування матриці, виходячи з розмірів зображення і заданого в налаштуваннях зшивання радіуса циліндра. Потім вона викликає першу та повертає отримане трансформоване зображення.

3.1.4 Розробка та реалізація функцій колірної корекції

Для реалізації був обраний метод перенесення кольору між зображеннями, і вона була виконана у модулі «color_corrections.py».

Перша функція `image_stats()` як параметр приймає зображення та повертає параметри кольорів у рамках колірного простору LAB, а саме середні та стандартні відхилення для кожного каналу. Вона призначена для використання у складі основної функції, яка буде розглянута останньою у цьому розділі.

Наступні дві функції виконують суто службову роль - обробку вхідних зображень, щоб значення кожного пікселя не вибивалися з діапазону 0-255, або шляхом масштабування, або просто перезаписом значень, що вибиваються, на мінімально або максимально допустимі.

Функція `_min_max_scale()` призначена для масштабування за мінімальними та максимальними значеннями в рамках масиву, що представляє зображення.

Функція `_scale_array()` або викликає попередню залежно від переданого параметра-прапора, або обмежує допустимі значення методом `clip()` бібліотеки Numpy.

Основна функція задіює вищеописані. Вона приймає пару зображень - те, з якого брати кольори (джерело), і те, на яке їх переносити (мета). Параметр `clip` відповідає за використання масштабування, чи просто обмежити значення. Параметр `preserve_paper` - підхід, яким обчислюються середні значення кольорів. Спочатку готується маска цільового зображення, щоб її застосувати в кінці функції - це необхідно, щоб у разі передачі зображення з порожніми областями ми ці області могли відновити на обробленому зображенні. Далі у функції `color_transfer()` обидва зображення конвертуються з RGB у LAB, а потім використовується `image_stats` для отримання параметрів їх кольорів.

Далі цільове зображення розбивається на масиви каналів шляхом `split()` бібліотеки OpenCV. З кожного каналу віднімається середнє відхилення для цього каналу, отримане раніше. Залежно від параметра `preserve_paper` масиви каналів перетворюються.

До змінених каналів додається середнє відхилення зображення-джерела, та проводиться масштабування (чи обмеження) підсумкових масивів.

Фінальний крок - з'єднання каналів, що вийшли, в підсумкове зображення і застосування до нього маски, після цього відбувається повернення отриманого результату.

3.1.5 Розробка та реалізація функцій змішування зображень

Для реалізації було обрано методи альфа-змішування (Alpha Blending, також Feathering) та змішування за допомогою пірамід Гауса/Лапласа (Pyramid Blending, також Multi-band Blending), для них було створено модуль «`blendings.py`».

Крім того, реалізовано просту функцію `blend_primitive()` для накладання зображень одна на одного без змішування.

Функція альфа-змішування `feather_stitching_api()` використовує вбудовану можливість OpenCV, для якої потрібно лише підготувати маски вхідних зображень і послідовно передати (спочатку вихідні зображення, потім маски)

об'єкт `blender`, створений методом `detail_FeatherBlender()`. Шляхом виклику методу `blend()` буде отримано змішане зображення.

Багатосмугове змішання (за допомогою пірамід Гауса/Лапласа) розбите вже на кілька функцій, за аналогією з корекцією кольору. Перша службова функція `preprocess()` призначена для первинної обробки зображень, які потрібно змішати. На вхід приймає обидва зображення, ширину області їх перетину (в межах якої буде проводитися змішання) і прапор для використання лише половини зображень (тому в ряді випадків цього може бути достатньо). Проводяться перевірки на рівність їх розмірів і те, щоб ширина області перетину не перевищувала ширину.

Далі зображення перетворюються – за рахунок порожніх значень ширина змінюється для оптимального накладання зображень друг на друга, також створюється спеціальна маска. При цьому саме тут передбачено варіант, що беруться лише половина зображень. Наприкінці функція повертає зображення та маску.

Далі реалізовано функцію `GaussianPyramid()` для побудови списку зображень, що є пірамідою Гауса. Вона приймає `numpy`-масив, що представляє зображення та необхідну кількість рівнів, а на виході - список модифікованих зображень, від більшого до меншого.

Наступна функція `LaplacianPyramid()` працює подібним чином (і з тими самими вхідними параметрами), але вже генерує піраміду Лапласа.

Далі задається функція `blend_pyramid()` для змішування пірамід, що приймає як параметри дві піраміди Лапласа (одна на кожне зображення, що змішується) і піраміду Гауса.

Чергова функція `reconstruct()` призначена для відновлення вже змішаного зображення зі списку, отриманого попередньої функції.

Оскільки в результаті змішування порожні області вихідних зображень стануть вже не порожніми, то необхідна особлива функція `getBlendMask()`, що створює маску, за якою області поза площею зображень будуть очищені.

Підсумкова функція `multiband()` використовує описані вище. Вона

приймає як параметри зображення, ширину області перетину, число рівнів та прапор для використання половини площі. Першими кроками перевіряється коректність ширини, проводиться первинна обробка функцією `preprocess()`, готується маска, обраховується максимально допустиме число перевірки допустимості переданого в функцію.

Потім створюються піраміда Гауса з маски для змішування та піраміди Лапласа для зображень, ці піраміди змішуються і потім реконструюються зображення. До зображення застосовується маска, результат видається функцією.

Далі реалізується спеціальний клас `BlenderObj()`, щоб можна було швидко використовувати будь-які способи змішування в алгоритмах. У конструкторі він приймає параметри вибору способу змішування та числа рівнів, а методі `blend()` виконує змішання обраним способом. Для багатосмугового змішування є автоматичне обчислення ширини області перетину.

3.1.6 Розробка та реалізація функцій вирівнювання та змішування країв панорами

Після того, як панорама буде зшита, для її придатності для застосування в циліндричній проекції, необхідно зробити змішування її країв. З цією метою створено модуль `align_rotate_ends_blend.py`.

Перед безпосереднім змішуванням країв може знадобитися трохи вирівняти панораму, якщо зйомка була не цілком рівномірною. Спеціальна функція `align_rotate()` здійснює циклічний поворот навколо центру переданого зображення на невеликий градус і перевіряє висоту зображення, котре вийшло.

Якщо продуктивного обертання не відбулося, то робиться аналогічна спроба обертання в протилежному напрямку.

Якщо в результаті знайдено оптимальний градус повороту, то повертається панорама, котра повернена на цей кут, інакше - вихідне зображення, таким чином панорама виявляється підготовленою до змішування

країв.

Функція `blend_ends()` змішування країв є модифікованою версією функції для зшивання двох зображень. Є спрощеною реалізацією методу, запропонованого у статті [27]. Передана панорама поділяється на центральну, праву та ліву частини. Для правої та лівих частин проводиться пошук та зіставлення ключових точок, вибір найкращих збігів.

Наступний етап - обчислення афінних перетворень, у тому числі береться необхідне зміщення по горизонталі одного зображення щодо іншого. Потім області лівого та правого зображення трансформуються по ширині та проводиться їхнє змішання.

Фінальний крок - поділ змішаного зображення навпіл і суміщення половин із центральною частиною вихідного зображення, що залишилася. Таким чином, функція повертає панораму зі змішаними краями.

3.1.7 Розробка та реалізація алгоритму сортування зображень

Цей алгоритм не є вбудованим у зшивання панорамних зображень з міркувань продуктивності та доцільності, але при використанні окремо він дозволить підготувати деякий набір зображень до зшивання. Це може бути актуальним, якщо вихідні зображення не відсортовані. Алгоритм реалізований окремим скриптом, що задіє деякі з описаних раніше модулі.

Перша функція `getNumberOfMatches()` дає змогу порахувати кількість збігів за ключовими точками для пари зображень.

Друга `getLeftmost()` дозволить знайти найлівіше зображення у списку, щонайменше збігів ключових точок з іншими зображеннями списку.

Далі ця функція використовується в циклі за кількістю зображень, на кожному кроці шукається найлівіше з тих, що залишилися і зберігається під наступним індексом. У результаті певної директорії буде відсортований пронумерований набір зображень.

3.1.8 Розробка та реалізація алгоритму зшивання панорамного зображення

Алгоритм призначений для зшивання послідовної серії зображень, що перекриваються (заздалегідь відсортованих) за ключовими точками із застосуванням різних параметрів. Він задіює майже всі описані раніше модулі, крім `align_rotate_ends_blend.py`, оскільки функції даного модуля застосовуються до зшитого зображення. Підсумковий варіант представлений у вигляді блок-схеми на рис. 3.2.

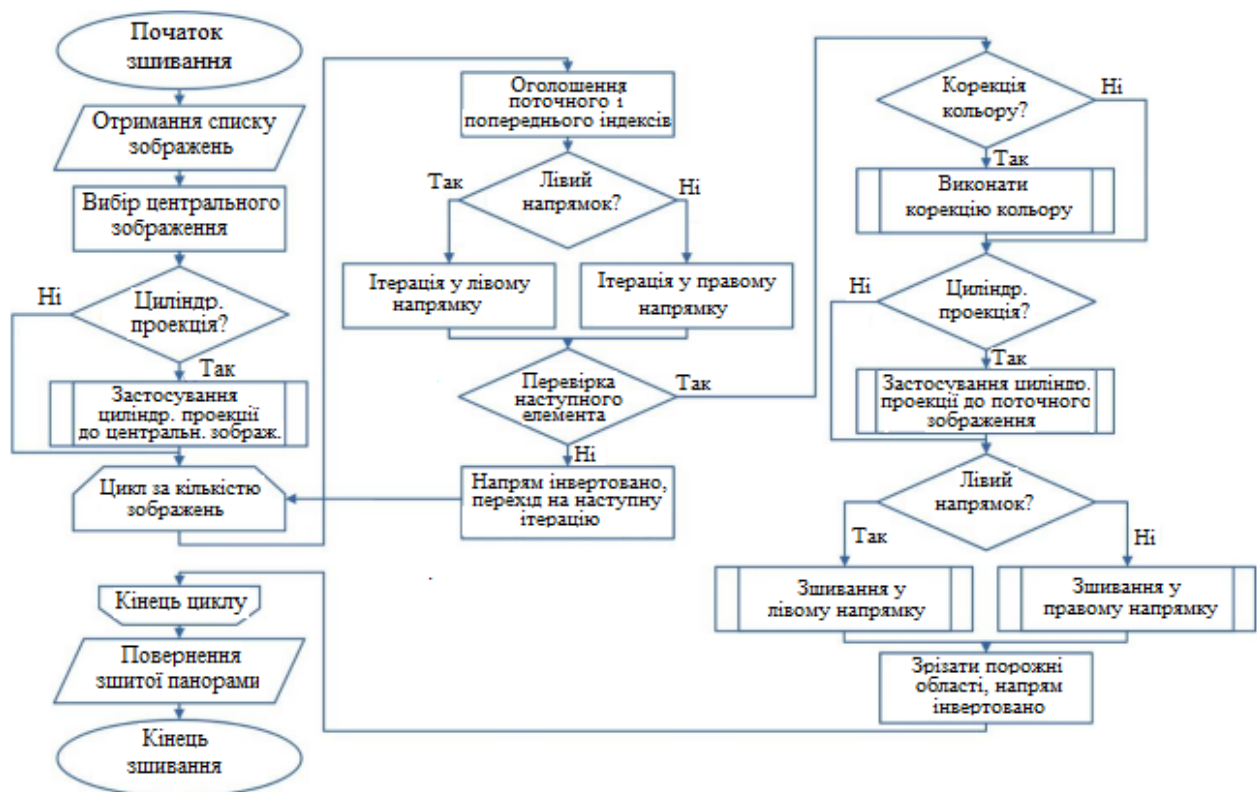


Рисунок 3.2 – Блок-схема алгоритму зшивання панорами

Алгоритм реалізований у модулі `panorama.py` як окремий клас. Конструктор класу приймає шлях до директорії з набором відсортованих зображень, а також об'єкти пошуку та зіставлення ключових точок та об'єкт змішування. Проводиться зчитування зображень з директорії та присвоєння об'єктів внутрішнім полям.

Метод `createPanorama()` з свого боку проводить зшивку панорами із завантажених зображень. Перший його крок - взяти центральне (за індексом) зображення як центральне в панорамі, тобто інші зшиватимуться з ним у правому або в лівому напрямку. За потреби застосовується циліндричне спотворення.

Далі ініціалізується цикл за кількістю зображень, на його початку проводиться визначення індексів поточного та попереднього зображень залежно від того, в якому напрямку зшивка проводиться в даний момент.

Потім проводиться перевірка на те, чи існує чергове зображення, або рух у цю сторону завершено. Якщо воно завершено, то напрямок змінюється і починається нова ітерація циклу, якщо ні - то виконуються (якщо налаштовані) корекція кольору і циліндричне спотворення чергового зображення.

На наступному кроці в залежності від напрямку викликається відповідний метод зшивання двох зображень.

Далі в кінці циклу зі зшитого зображення зрізаються порожні області, змінюється напрямок руху і лічильник зшитих зображень збільшується. Наприкінці метод повертає зображення, що утворилося.

3.1.9 Структура програмної системи для зшивання панорамних зображень

Програмна система розроблялася у вигляді набору модулів Python, кожен із яких спеціалізований. Їх перелік :

- "feature_detection.py" - для роботи з ключовими точками;
- "img_trim.py" - невеликий набір функцій для обрізання зображень;
- "color_corrections.py" - реалізація колірної корекції;
- "blendings.py" - для змішування зображень;
- "panorama.py" - для створення панорамних зображень;
- "align_rotate_ends_blend.py" - змішування країв у створених панорамах.

У результаті, цей набір модулів є невеликою бібліотекою, що базується на основі OpenCV, котра є придатною для використання в прикладних, орієнтованих на практичне застосування, програмах на мові Python (рис. 3.3).

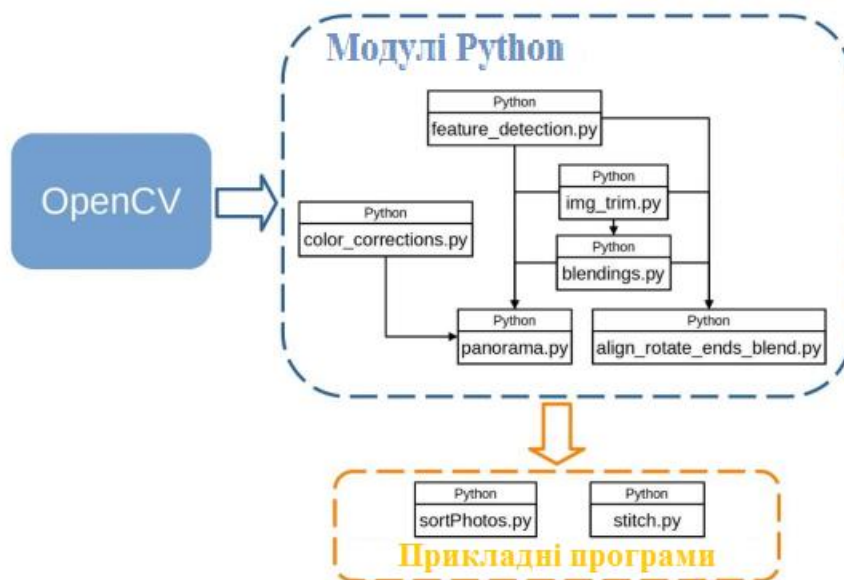


Рисунок 3. 3 – Модулі та прикладні програми

Наприклад, sortPhotos.py - це розглянутий раніше алгоритм сортування зображень, який задіює для цього кілька модулів, а stitcher.py запускає зшивання будь-якого набору зображень, а потім вирівнює отриманий результат і змішує у нього краї.

3.1.10 Розробка та реалізація візуалізації панорамного зображення

Для візуалізації панорами з використанням веб-технологій використано бібліотеку мовою JavaScript panolens.js, яка у свою чергу вимагає графічну бібліотеку threejs. Розгортання демонстраційної візуалізації проведено у рамках репозиторію на сервісі GitHub з використанням опції Pages, що дозволяє репозиторію бути хостингом статичних веб-сторінок.

При цьому, внаслідок особливостей роботи panolens.js і для кращих демонстраційних можливостей, було прийнято рішення зробити дві html-сторінки - одна для безпосереднього відтворення панорами відповідно до

переданих GET-параметрів, а друга - для підключення першої через html-елемент `iframe`. Це дозволяє, за потреби, вивести кілька панорам у межах однієї сторінки без необхідності писати складний JS-код. Схема такого підходу представлена на рис. 3.4.

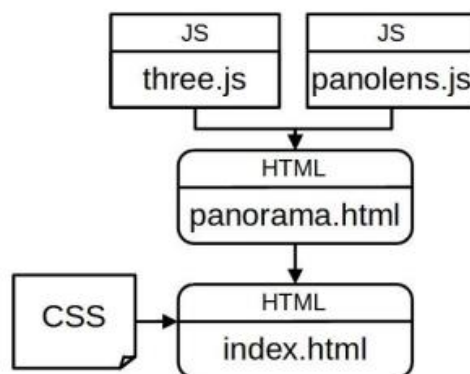


Рисунок 3.4 – Схема веб-візуалізації панорами

До бібліотеки `panolens.js` було внесено доопрацювання для існуючої функції-конструктора. Замість сферичної проекції, що використовується за умовчанням, зроблена циліндрична і додані входні параметри для радіуса і висоти циліндра в пікселях. Підсумкова функція `ImagePanorama()`, використовуючи `three.js`, ініціалізує циліндричний каркас для панорами із заданим радіусом і висотою циліндра. Файл `panorama.html` включає JS-код для обробки входних параметрів та ініціалізації панорами. Для отримання параметрів береться рядок запиту, і з нього витягуються необхідні параметри, такі як радіус і висота циліндра, панорамне цільове зображення і кут огляду.

Далі ініціалізується новий об'єкт панорами із зазначенням зображення, радіусу та висоти.

Наступний крок - завдання об'єкта візуалізації з набором параметрів відображення панорами, таких як кут огляду, відображення елементів керування тощо, а потім і передача об'єкта панорами об'єкт візуалізації.

У файлі `index.html` підключається сторінка `panorama.html` через елемент `iframe`, використовуючи просте `url`-посилання з параметрами.

3.2 Результати експериментів

3.2.1 Порівняльний аналіз одержаних результатів ефективності методів пошуку ключових точок

Для оцінки ефективності обробки двох зображень та зіставлення методами пошуку ключових особливостей SIFT, ORB, BRISK та AKAZE було проведено тестування на різних вхідних даних, а саме зображення «MOUNTAIN», «RIVER», «HOUSE», «UTTOWER» та «ROOFS» при зміні кута огляду та яскравості кадру під час отримання знімка щодо вихідного. Тестування проведено за допомогою розробленого алгоритму поетапного суміщення зображень, що перекриваються, який дозволяє зібрати необхідну статистику і проілюструвати її.

Результати тестування частково представлені на зображеннях MOUNTAIN на рис. 3.5. для методу ORB з вибраними методами зіставлення BFMatcher та FlannBasedMatcher з оцінкою матриці гомографії методом RANSAC.

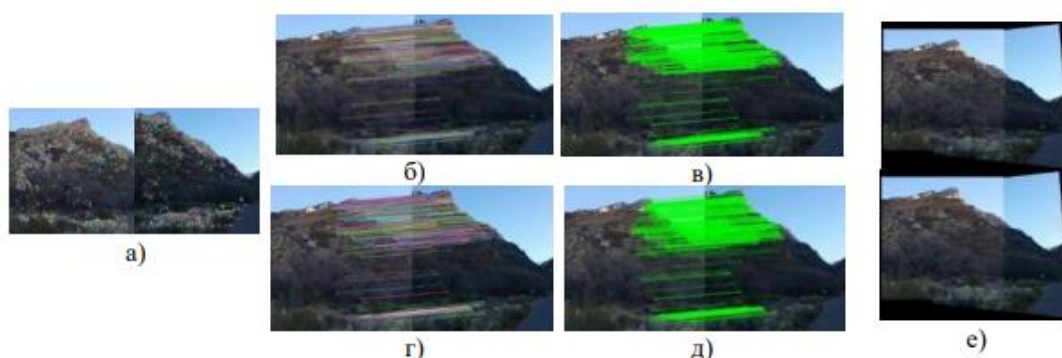


Рисунок 3.5 – Результати тестування методом ORB: а) ключові точки на кадрах MOUNTAIN; зіставлення всіх точок б) BFMatcher_ HAMMING та г) FlannBasedMatcher_ FLANN_LSH; зіставлення RANSAC в) BFMatcher та д) FlannBasedMatcher; зшиті зображення е) гомографія та афінні перетворення

Проведення візуального порівняльного аналізу ефективності обробки методами за підсумковим зображенням є достатньо складним завданням.

Для оцінки ефективності методів виконано порівняння результатів за середніми значеннями як діаграм (рис. 3.6 – 3.8).

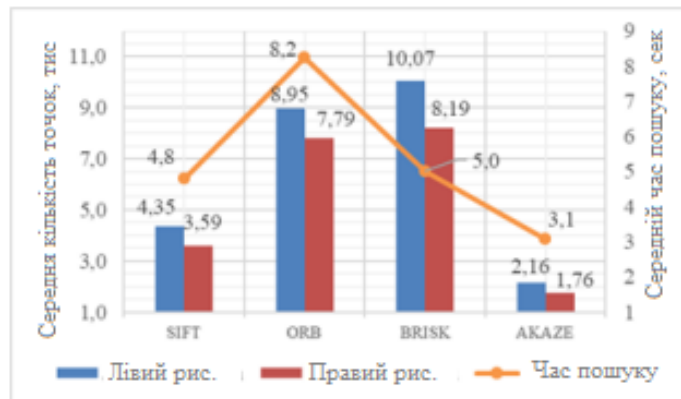


Рисунок 3.6 – Результати пошуку ключових точок

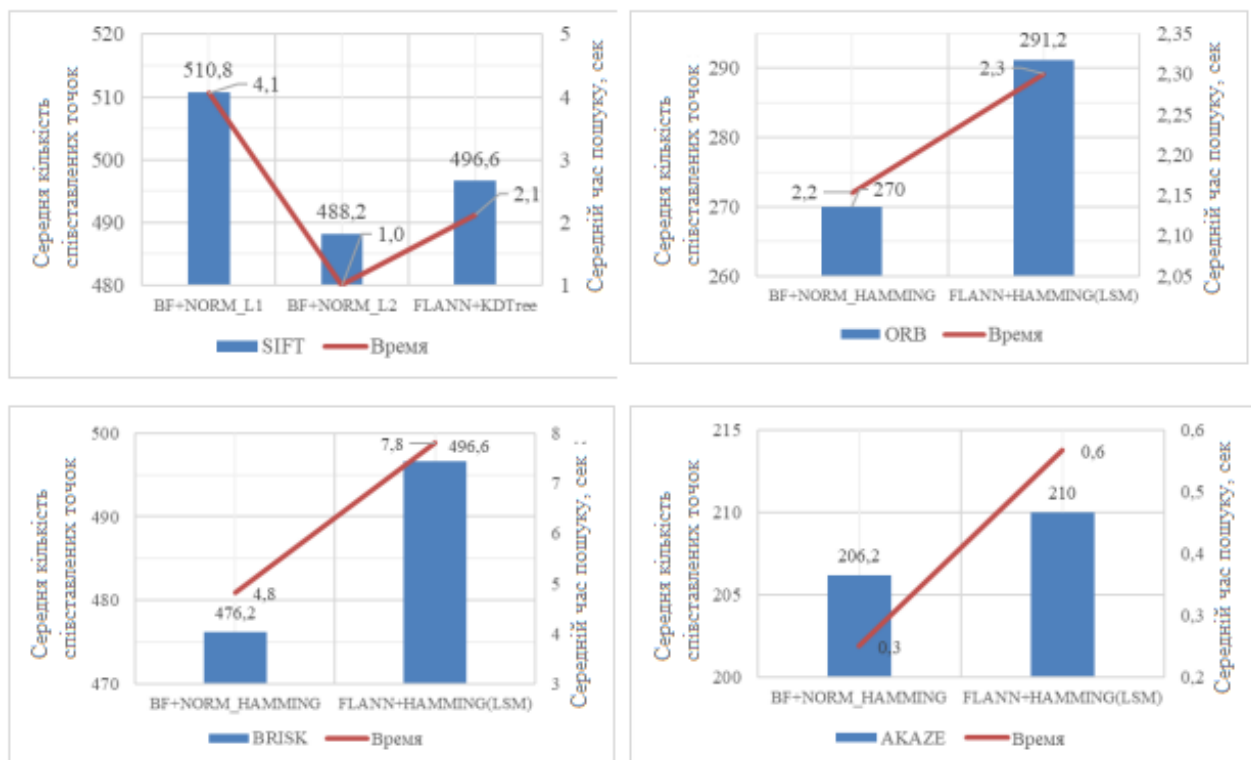


Рисунок 3.7 – Результати зіставлення точок методами BFMatcher та FlannBasedMatcher

З наведеного вище випливає висновок, що метод пошуку особливостей BRISK має найвищу продуктивність - найбільше знайдених ключових точок на тестових зображеннях та найменший середній час пошуку.

Найкращий спосіб зіставлення – BFMatcher з функцією відстані Евкліда та відстані Хеммінга, проте, варто провести тестування метрики FlannBasedMatcher з алгоритмом k-середніх та алгоритмом локального хешування на зшиванні зображень у панорамні знімки.

Також слід зазначити, що на результати досліджень вплинула на якість зображень: більше «хороших» збігів було на зображеннях з однорідним та яскравим фоном, більшою деталізацією та незначним зміною перспективи.

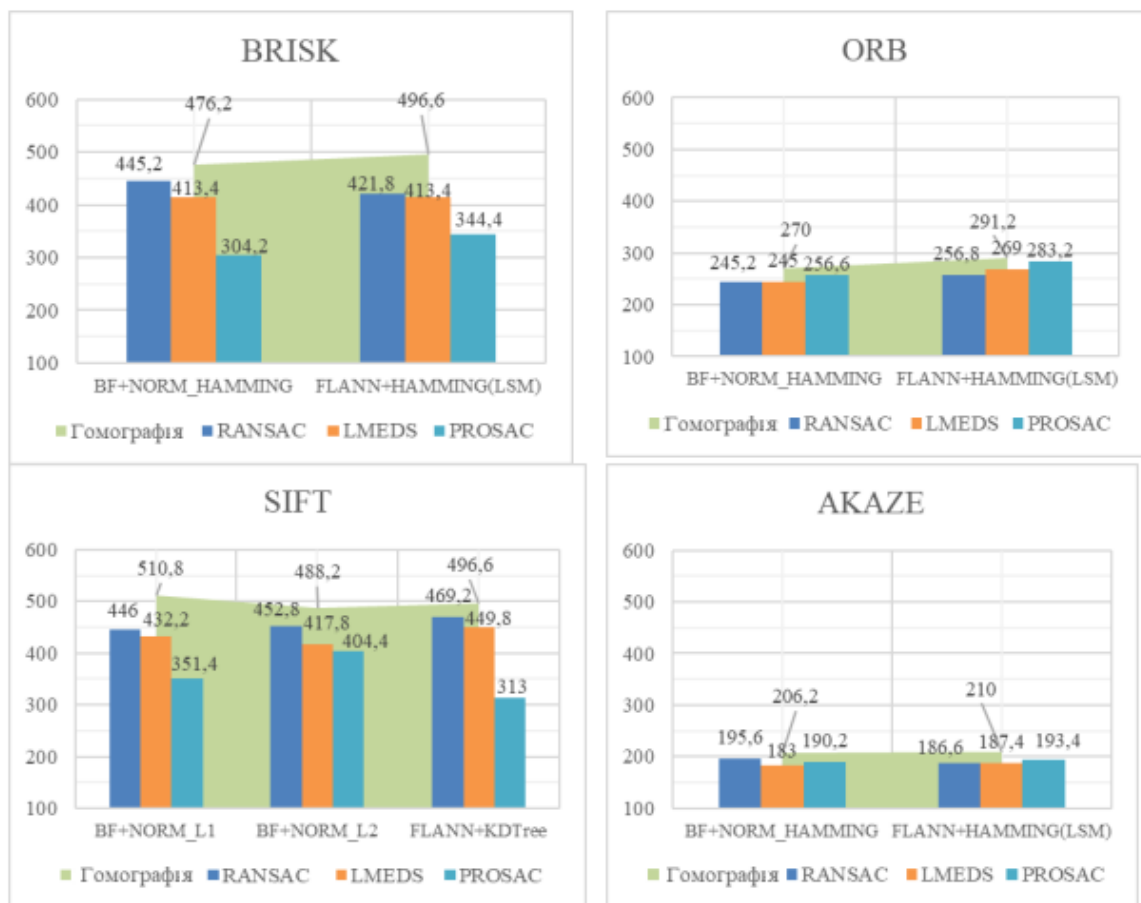


Рисунок 3.8 – Результати методів видалення невірних збігів ключових точок

Проаналізувавши отримані дані, можна говорити, що метод поліпшення гомографії RANSAC як пошук і видалення невірних збігів перевищує інші алгоритми на 3-10 %.

В результаті роботи алгоритму поетапної зшивання зображень MOUNTAIN, RIVER, HOUSE, UT TOWER та ROOFS з визначенням матриці гомографії та афінними перетвореннями утворилося 198 спостережень. Зшити

зображення не вдалося методом SIFT з використанням функції дистанції L1 - норми (Манхеттенська відстань). Раніше була доведена висока продуктивність методу BRISK, найкращий спосіб зіставлення для нього - BFMatcher з функцією відстані Хеммінга та імовірнісним методом RANSAC. Розглянемо докладніше зображення з даними метриками (рис. 3.9).

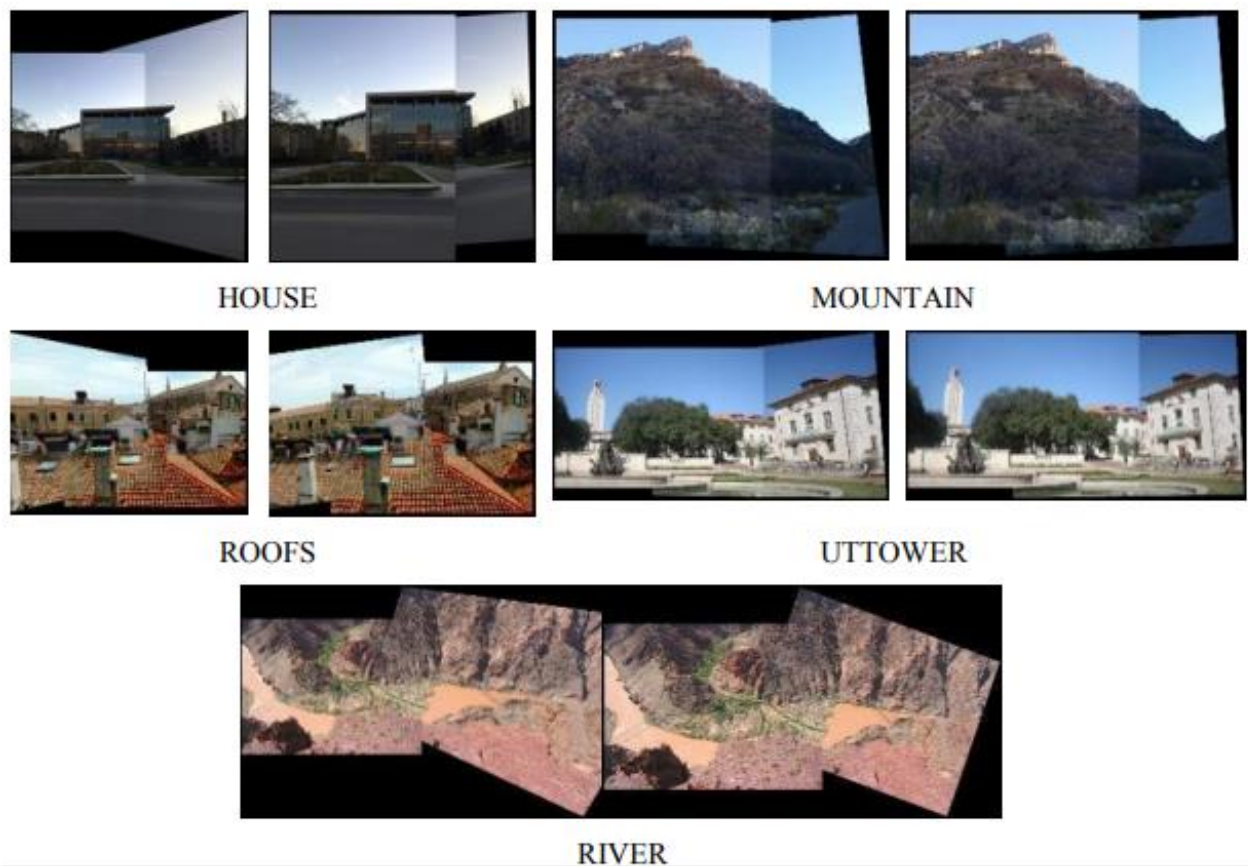


Рисунок 3.9 – Результати зшивання зображень з визначенням матриці гомографії (рисунок зліва) та афінними перетвореннями (рисунок справа)

Можна із упевненістю констатувати, що матриця гомографії ефективніша для зшивання кадрів у панорамні зображення, оскільки перетворення зображень виконуються у тривимірному просторі.

Ці параметри приймаються як параметри за замовчуванням для подальшої роботи, тобто для перевірки можливостей зшивання з використанням можливостей змішування, корекції кольору, циліндричних спотворень і в панорамному режимі із задіянням кількох зображень.

Тестування зі збором статистики проводилося персональному комп'ютері із параметрами конфігурації, приведеними у табл. 3.1.

Таблиця 3.1 – Опис конфігурації комп'ютера

Елемент конфігурації	Його значення
Монітор	LG 23MP68VQ, 1920x1080, 16:9
Процесор	AMD Ryzen 7 1700X (вісім ядер, шістнадцять потоків, 3,4/3,8 ГГц, 16 Мбайт L3)
Оперативна пам'ять	32 Гбайт DDR4-3066
Чіпсет	AMD X570
Відеокарта	AMD Radeon 580X, 8 Гб GDDR5
Жорсткий диск	Seagate Barracuda 1 Тбайт (7200 об/хв, Serial ATA)
Жорсткий жорсткий диск	ADATA 256 Гбайт
Блок живлення	700 Вт
Розміри системного блоку	336x424x320 мм
Операційна система	Windows 10 Pro 64-bit

3.2.2 Порівняльна демонстрація зшивання зображень із використанням корекції кольору та змішування

Для первинної перевірки можливостей та якості зшиванн з використанням корекції кольорів і змішування взяті ті ж невеликі набори зображень - MOUNTAIN, RIVER, HOUSE, UT TOWER і ROOFS і використані найбільш ефективні параметри. При цьому використовується повноцінний алгоритм панорамного зшивання, просто із відключеною опцією циліндричного спотворення. Це дозволить для демонстрації «пошити» повний набір MOUNTAIN, який сам по собі містить 3 зображення, але через особливості алгоритму (напрямок зшивання, визначення центру) набори з пар картинок можуть виглядати трохи інакше, ніж у попередньому розділі. Таким чином, для кожного набору зображень проведено три зшивання: без корекції кольорів і змішування, з корекцією кольорів і альфа-змішенням, з корекцією кольорів і

багатосмуговим змішування (методом пірамід Гауса / Лапласа). Отримані результати на прикладі набору зображень MOUNTAIN наведені на рис. 3.10.



Рисунок 3.10 – Результати зшивання MOUNTAIN; лівий рисунок - без використання корекції кольору та змішування, центральний - з корекцією кольорів і альфа-змішенням, правий - з корекцією кольору і багатосмуговим змішуванням

Можна відзначити, що корекція кольору методом перенесення кольорів і змішання, у загальному, позитивно впливає на зшите зображення. Наявні окремі особливості, та обмеження.

Корекція кольору явно впливає на витягнуті особливості зображення, оскільки застосовується до безпосереднього зшивання, таким чином обчислена гомографія (або афінні координати) і трансформація будуть дещо іншими.

Багатосмугове змішування забезпечує більш плавний перехід у порівнянні із методом альфа-змішування, крім того відсутній ефект паралаксу, проте ті області зображень, які перебуваючи в області перекриття не мають перекриття як такого з областями іншого зображення будуть втрачені. Тим не менш, у разі рівної фотозйомки та гарного горизонтального перекриття зображень цей метод явно кращий і для зшивання великих панорам буде використовуватися саме він.

3.3 Порівняння з іншими програмними системами для зшивання панорамних зображень

Для порівняння з розробленою в рамках роботи програмною системою

було обрано два існуючі аналоги, здатні до безшовного зшивання панорам. При порівнянні з ними, розроблена система буде називатися БЗЗ.

Перший аналог - High level stitching API [28], (далі для стислості Stitching API) високорівневий програмний інтерфейс, що існує в рамках OpenCV і призначений для зшивання панорам на основі методів пошуку ключових точок. Він також має Python- реалізацію та можливість вибору різних параметрів, тому є найпростішим для організації порівняння. По можливості підбиратимуться ті ж параметри, що і для БЗЗ.

Другий - Hugin [29], додаток з графічним інтерфейсом користувача, що реалізує можливості набору утиліт та бібліотек Panorama Tools [30] з відкритим кодом. Цей додаток та пов'язані бібліотеки, на відміну від OpenCV, призначені виключно для зшивання панорам. На відміну від Stitching API та БЗЗ він є більш цілісною та придатною для негайного практичного використання програмною системою, проте менш оптимальною для дослідницької діяльності.

Процес порівняння буде розбито на кілька частин, будуть розглядатися можливості налаштування параметрів зшивання, потім будуть представлені практичні результати на реальних наборах фотографій.

Перша частина - для роботи з ключовими особливостями: можливості вибору методу пошуку ключових точок (і представлені варіанти), методу зіставлення, функції відстані, показників подібності ключових особливостей (табл. 3.2).

Очевидно, що БЗЗ пропонує набагато більш низькорівневий доступ до параметрів, ніж навіть Stitching API, у той час як Hugin таких можливостей зовсім не пропонує. Це потенційна можливість для покращення та оптимізації, а також подальшої дослідницької роботи над алгоритмами зшивання панорам, оскільки немає необхідності зміни вихідного коду бібліотеки OpenCV.

Таблиця 3.2 - Порівняння можливостей роботи з ключовими точками

Система	Вибір методу пошуку особливостей	Вибір способу зіставлення	Вибір функції відстані	Задання показників подібності
Stitching API	SURF (якщо доступний), SIFT, ORB, BRISK, AKAZE	Ні	Ні	Так
Hugin	Ні (SIFT)	Ні	Ні	Ні
БЗЗ	SIFT, ORB, BRISK, AKAZE	Так	Хеммінга, Евкліда, k-середніх та ін.	Так

Подальше порівняння (табл. 3.3) - за наявними особливостями і можливостями налаштування самого зшивання, як вибору методу обчислення перетворень перспективи, ймовірнісного методу для його обчислення, можливості вирівнювання, колірної корекції, змішування зображень і цільової проекції.

Таблиця 3.3 - Порівняння можливостей налаштування з перспективними перетвореннями

Система	Вибір перетворення перспективи	Вибір імовірнісного методу	Вирівнювання зображень	Корекція кольору	Змішування зображень	Проекція
Stitching API	Гомографія, афінні	Ні	Так	Так	Альфа-змішування, багатосмугове змішув.	Різні
Hugin	Ні (гомографія)	Ні	Так	Так	Ні (Власний алгоритм)	Різні
БЗЗ	Гомографія, афінні	RANSAC, PROSAC, LMEDS	Часткове	Так	Альфа-змішування, багатосмугове змішув.	Циліндрична

В даному випадку БЗЗ еквівалентна Stitching API за можливостями перспективних перетворень та змішування, а також є можливість вибрати ймовірнісний метод, але можливості вирівнювання більш обмежені (тільки для

вже зшиті панорами). Алгоритм колірної корекції, що використовується в БЗЗ, в деяких ситуаціях поступається конкурентам, а проекція реалізована тільки циліндрична, у той час як у інших проєктів їх багато. Однак, необхідно акцентувати, що Stitching API та Hugin/PanoTools розвиваються протягом багатьох років, і відповідно можливостей там реалізовано більше. З іншого боку, у БЗЗ необхідні допрацювання можливо проводити швидше, ніж у Stitching API завдяки Python - реалізації, а Hugin/PanoTools є готовим продуктом, а не програмною бібліотекою для використання в інших проєктах.

Порівняння практичних результатів зшивання можна розпочати з набору фотографій MOUNTAIN. Для Stitching API та БЗЗ був обраний метод пошуку ключових особливостей BRISK та багатосмугове змішування. Інші налаштування БЗЗ - згідно з раніше отриманими результатами тестів. Проекції у всіх системи були відключені. Результати зшивання наведені на рис. 3.11.



Рисунок 3.11 – Результати зшивання MOUNTAIN

На цьому наборі зображень можна відзначити, що обчислення гомографії явно налаштовано по-різному в кожній системі. З якихось причин є невеликі артефакти в центрі на панорамі з Hugin. А змішання у Stitching API відпрацювало не найкращим чином - дуже помітна «змазаність», тобто в даному випадку БЗЗ змішано краще навіть з урахуванням зникнення невеликих сегментів, які не мають перекриття з іншими зображеннями. При цьому формально метод змішування у Stitching API та БЗЗ обрані аналогічні (але з різною реалізацією).

Наступний набір фотографій – із торгового центру. Він характерний

низькою яскравістю та контрастністю та великою кількістю деталей. Хоча це і не кругова фотозйомка, було обрано циліндричну проекцію. Результати зшивання показані на рис. 3.12.

Тут БЗЗ та Stitching API демонструють дуже близькі за якістю результати, у той час як Hugin виявився нездатним поєднати зображення рівно.

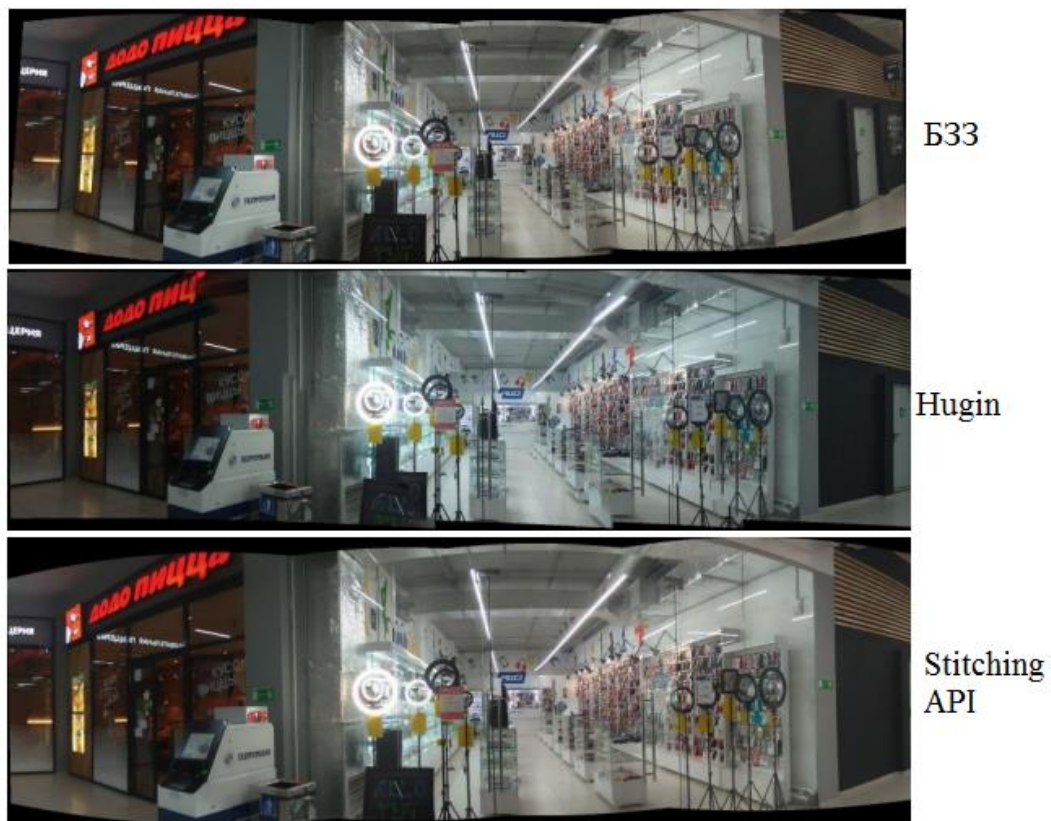


Рисунок 3.12 – Результати зшивання у торговому центрі

Третій набір - це результат повноцінної кругової фотозйомки, придатний для створення циліндричної панорами на 360 градусів. Налаштування БЗЗ та Stitching API залишаються незмінними, але для першого тепер задіюється функція змішування країв. Отримані у результаті повноцінні панорами наведені на рис. 3.13.

На даному наборі найкращий результат – у Hugin. Він зміг компенсувати недоліки зйомки та якісно поєднати краї зображення. У Stitching API явно неправильно перетворені два зображення, а у БЗЗ не дуже акуратно пошиті краї через нестачу вирівнювання.

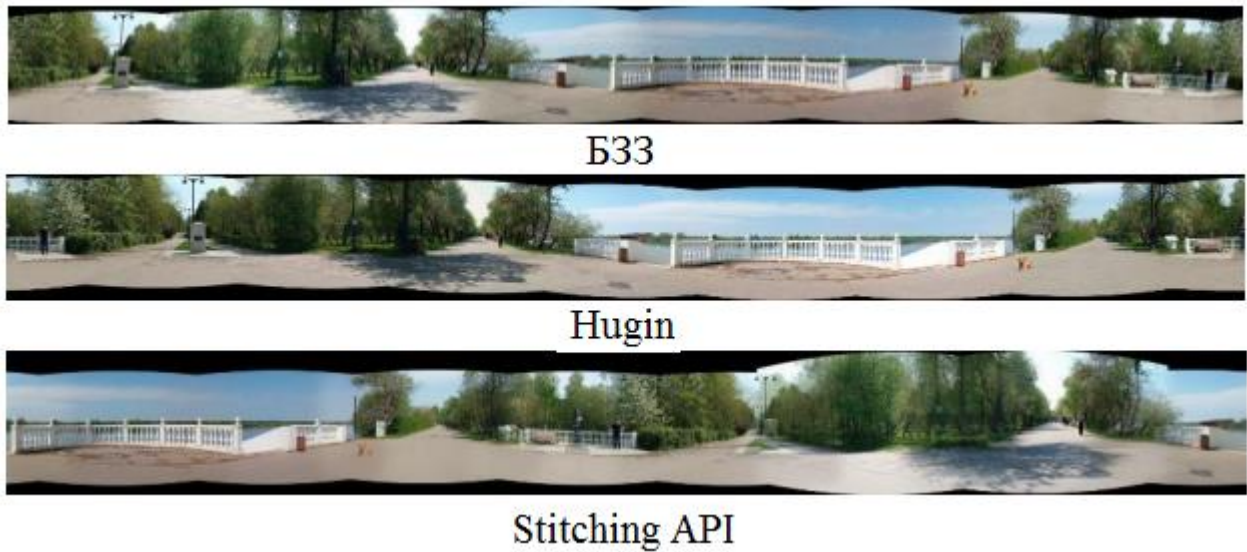


Рисунок 3.13 – Результати зшивання повної панорами

З цих та інших зшивок (повний набір зшитих панорам представлені у додатку Б) можна зробити деякі висновки. Універсального рішення для зшивання із розглянутих систем немає, оскільки у різних систем різні можливості і з одного і того ж набору зображень можуть вийти результати, що відрізняються. При цьому, в одних випадках може краще відпрацювати одна система, а в інших - вже інша.

3.4 Висновки до третього розділу

В цьому розділі докладно описано розробку та реалізацію системи алгоритмів та програмних компонентів для створення панорамного зображення. Наведено блок-схеми окремих розроблених алгоритмів.

Для проведення оцінки ефективності обробки зображень та співставлення методами відшукування ключових особливостей SIFT, ORB, BRISK та AKAZE було виконано експериментальне тестування для різних вхідних даних.

Виконано порівняння з існуючими програмними засобами, котрі застосовуються для зшивання панорам (Hugin та Stitching API). Результати проведених експериментальних досліджень свідчать, що єдиного рішення для

зшивання немає, так як у різних засобів є різні можливості. Варто зауважити, що для одних зображень краще використати одну систему, а для інших - уже іншу.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Закордонний досвід організації охорони праці в ІТ-компаніях

Стан справ з охороною праці у світі стає все більш актуальною проблемою як для профспілок, так і для міждержавних структур, насамперед Міжнародної організації праці (МОП). МОП розглядає цю тему як частину своєї Програми гідної праці. Підвищена увага до проблем безпеки праці пояснюється в першу чергу тим, що з кожним роком, незважаючи на заходи, що вживаються, у різних країнах зростає рівень виробничого травматизму, у тому числі зі смертельними наслідками, і кількість профзахворювань.

На перший погляд, робота за комп'ютером при дослідженні моделей та побудові системи ПФПК здається безпечною, але саме легковажність до неї може призвести до певних проблем у здоров'ї людини. Професія програміста та інших фахівців ІТ-технологій пов'язана з колосальним розумовим напруженням. Розробники – настільки захоплені люди, що навіть відволікаючись від роботи над проектом, продовжують думати про роботу. Нерідко відпочинком вони вважають паралельну заміну основної діяльності, наприклад, читання профільної літератури, верстку сайтів, вивчення нових мов програмування. Однак мозок не може до безкінечності приймати виключно корисну інформацію, яку розробник прагне направляти в русло особистісного та професійного зростання [35].

Зарубіжний досвід охорони праці при використанні новітніх інформаційних технологій та сучасного комп'ютерного обладнання передбачає з метою попередження наслідків монотонної праці, підвищення рівня рухової активності і покращення розумової працездатності фахівців ІТ-індустрії під час технологічних перерв участь у спеціальних облаштованих приміщеннях необхідним спортивним інвентарем та різними тренажерами відповідних фізичних вправ, індивідуальних тренінгових завдань відповідно до віку, статі та категорії зорової роботи. Такий підхід дозволяє зняти надлишкове

психофізіологічне перевантаження, підвищити працездатність центральної нервової системи, попередити перевтому зорового аналізатора. Показана ефективність проведення різноманітних за своєю спрямованістю вправ робітників цієї галузі (приблизно на 5-30%) [35].

Зараз багато ІТ-компаній обладнують свої офіси кімнатами відпочинку та лаунж-зонами, які забезпечують психофізіологічне розвантаження працівників. Адже окремим робочим столом з ноутбуком вже давно нікого не здивуєш. Тому, бажаючи підвищити продуктивність працівників, міжнародні компанії змагаються, перетворюючи нудні одноманітні офіси в креативні простори, де нові ідеї народжуються без титанічних зусиль. Наприклад, винахідники компанії Inventionland трудяться в казкових декораціях. Тут і гігантський гоночний трек, і пряниковий будиночок, і дуже реалістичний піратський корабель, на палубі якого розташувалися комп'ютерні столи, ніжками яких служать винні бочки. Робочі місця співробітників компанії Google в Цюриху нагадують гігантські вулики, а офіс шведського інтернет-провайдера Bahnhof розташувався в бомбосховищі часів холодної війни і походить на підземний притулок землян після глобальної катастрофи. А щоб співробітників не тягнуло додому, роботодавці створюють і можливість релаксувати, не відходячи від робочого місця, обладнавши басейни, ігрові кімнати та спортзали [35].

Корпорація Google піклується і про санітарно-гігієнічні умови праці та використовує систему очищення повітря, яка видаляє з атмосфери всі токсини та важкі домішки [36]. Також незвичайними на території офісу є спеціальні ізолюючі світло й звук капсули, де втомлений співробітник може відпочити, так само як і в спеціалізованих кімнатах у зоні відпочинку з приглушеним світлом та заспокійливою музикою, масажних кабінетах, ігрових кімнатах. Культивуації активного відпочинку та спорту в Google відводиться особлива увага, оскільки одні з найпоширеніших хвороб програмістів пов'язані зі спиною. Саме тому в головному офісі міститься спортзал, в якому можуть одночасно займатися велика частка співробітників, до того ж на території є плавальний басейн з черговим рятувальником, тому, фактично, людина може працювати прямо

звідти (пам'ятаючи про дотримання правил техніки безпеки) [37]. Для переміщення по великій території Гуглмістечка «гуглівці» (так себе називають співробітники цієї компанії) користуються самокатами, які мають моторчики. Звичайно, усі ці зручності безкоштовні.

Намагається не відставати від Google і корпорація Facebook. Так, дана фірма пропонує своїм співробітникам чудові умови праці: якісне різноманітне харчування, зручне апаратне забезпечення, важливою складовою якого є широкі монітори ПК з високою роздільною здатністю [38]. Особливістю Facebook є те, що співробітники не мають ані своїх кабінетів, ані офісних кімнат – усі сидять разом, оскільки це стимулює комунікацію та обмін ідеями, підвищує продуктивність роботи. Загалом у компанії царює атмосфера позитиву. Усі співробітники привітні та посміхаються. На стінах можна побачити карикатури на топменеджерів, а робітники одягають футболки зі смішними написами [38]. Співробітники тут, на відміну від Google, пересуваються по офісу не на самокатах, а на скейтах. У Facebook також наявна велика кількість тематичних та ігрових зон, де кожен може розслабитися та відпочити, і це все також абсолютно безкоштовно.

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. З метою належного правового забезпечення необхідно розширити та доповнити перелік основних професій комп'ютерної галузі у національному класифікаторі [39], а також підготувати відповідний випуск у кваліфікаційному довіднику посад фахівців ІТ-індустрії, що сприятиме вирішенню питань їх соціального захисту, пенсійного забезпечення, атестації робочих місць основних професій за умовами праці на предмет подальших певних видів пільг та компенсацій за важкі шкідливі і небезпечні умови праці. Важливим напрямом стосовно визначення професійної придатності фахівців з ІТ є проведення психофізіологічної експертизи відповідно до 5 статті [40].

ІТ-фахівці, як і будь-які інші працівники, повинні проходити навчання і перевірку знань з охорони праці або в навчальному центрі, або в самій

організації. Якщо в ній є комісія з перевірки знань з охорони праці, атестованих в спеціалізованому навчальному центрі. Навчання охорони праці в організації проводять по самостійно розробленими програмами. Їх складають, спираючись на типові програми, а також з огляду на особливості галузі, в якій працює організація.

Робота з комп'ютерами нового покоління характеризується певним психофізіологічними перенавантаженнями, втому зорового аналізатора, гіпокінезією, відсутність диференційованих норм праці при роботі з новою комп'ютерною технікою в залежності від віку, статі, категорії зорової роботи, режимів праці і відпочинку (протягом робочого дня, тижня, щорічного режиму відпусток). Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Особлива роль з точки зору збереження та відновлення здоров'я працюючих в комп'ютерні галузі належить попереднім та періодичним наглядам з подальшої психофізіологічної експертизи і встановленням професійної придатності при роботі з комп'ютерами нового покоління, який супроводжується виникненням певних факторів професійного ризику електротравматизму при їх ремонті та обслуговуванні. В цьому зв'язку необхідне запровадження експертизи на предмет безпечної експлуатації ПЕОМ, тобто офіційне підтвердження фактичних параметрів електробезпеки, їх відповідності вимогам нормативної документації фахівців, які проводять таку експертизу повинні пройти навчання і перевірку знань відповідно до вимог [41]. За результатами експертизи повинні прийматися рішення про відповідність ПЕОМ нормам безпеки, терміни чергової експертизи, оформлюються протоколи вимірювань і випробувань, проведені у разі потреби розрахунки та експертний висновок. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії.

Заслуговує на увагу зарубіжний досвід створення у приміщеннях та в зоні їх розміщення на територіях підприємств спеціальних візуальних комфортних умов та забезпечення вимог виробничої естетики, дотримання норм рівнів виробничого шуму та акустичної тиші за межами офісу. Також дуже важливим є використання в офісних приміщеннях та кабінетах психофізіологічного розвантаження функціональної музики, яка сприяє попередженню перевтоми і підтриманню необхідного рівня розумової працездатності фахівців комп'ютерної галузі. В цьому напрямі заслуговує на увагу створення при великих центрах інформаційних технологій кімнат (кабінетів) психофізіологічного розвантаження працівників галузі (на 5 місць) [36].

Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Таким чином, використання новітніх технологій вимагає від фахівців ІТ-індустрії дотримання певних правил та вимог з точки зору безпеки праці, її нормування з урахуванням віку працюючих та загального інформаційного навантаження, розробки та впровадження індивідуальних, щотижневих та щорічних режимів праці та відпочинку, які сприятимуть профілактиці перевтомлення і підвищенню розумової працездатності працюючих. Особливу роль в цьому напрямі повинні відігравати ергономічні заходи стосовно створення робочих місць, оптимізації взаємодії людини в рамках системи «оператор-термінал».

4.2 Оцінка дії електромагнітного імпульсу на елементи комп'ютерної системи

У воєнний час при застосуванні ядерної зброї проти України на електронно-обчислювальне обладнання в першу чергу буде впливати електромагнітний

імпульс (ЕМІ) ядерного вибуху у вигляді короткого імпульсу, який вражає головним чином електричну та електронну апаратуру [42].

ЕМІ виникають в основному в результаті взаємодії гамма-випромінювання з атомами навколишнього середовища. На утворення ЕМІ йде невелика кількість ядерної енергії, але він здатен викликати високі імпульси струмів та напруг в кабелях повітряних і підземних ліній зв'язку, сигналізації, управління, електропередачі, в антенах радіостанцій. Вплив ЕМІ може привести до згорання чутливих електронних та електричних елементів, зв'язаних з великими антенами чи відкритими дротами, а також до порушень в обчислювальних пристроях.

Особливістю ЕМІ, як вражаючого фактору є його здатність розповсюджуватись на десятки і сотні кілометрів в оточуючому середовищі. Тому ЕМІ може вплинути своєю дією на об'єкти, там де вибухова хвиля, світлове випромінювання, проникаюча радіація втрачають своє значення, як вражаючі фактори. При наземних та низьких повітряних вибухах в лініях зв'язку та електрозабезпечення виникають напруги, які можуть викликати пробій ізоляції провідників та кабелів відносно землі, пробій ізоляції елементів приладів підключених до повітряних і підземних ліній. Степінь враження залежить від наведеного імпульсу напруги чи струму і також електричної міцності обладнання.

Найбільш піддані впливу ЕМІ системи зв'язку, сигналізації, управління. Використані в цих системах кабелі та апаратура мають обмежену електричну міцність не більше 10кВ імпульсної напруги, тоді як наведені імпульси напруги від ЕМІ можуть перевищувати ці значення. Найбільш піддана впливу ЕМІ апаратура виконана на напівпровідниках та інтегральних схемах, працюючих на малих струмах і напругах, і значить відчутних до впливу зовнішніх електричних і магнітних кіл, в тому числі і елементи програмного засобу для управління процесом міграції віртуальних машин в обчислювальній хмарі. ЕМІ пробиває ізоляцію, спалює елементи електричних схем радіоапаратури, викликає коротке замикання в радіопристроях, іонізацію діелектриків, змінює або повністю стирає магнітний запис. Встановлено, що при дії ЕМІ на апаратуру найбільша напруга

наводиться на вході. В транзисторах відбувається така залежність: чим більший коефіцієнт підсилення транзистора, тим менша його електрична міцність.

ЕМІ пошкоджує також резистори, викликає іскріння в їх міжконтактних з'єднаннях і деяких областях провідної поверхні. Найбільшу небезпеку ЕМІ представляє для апаратури, яка встановлена в особливо міцних спорудах, які витримують великі тиски ударної хвилі. В цих спорудах апаратура не виходить з ладу від механічних пошкоджень, але ЕМІ може вивести з ладу всю незахищену апаратуру системи зв'язку, сигналізації і керування. Найбільших значень досягають напруги, які наводяться між кабелем і землею.

Розглянемо можливі шляхи рішення задачі захисту від ЕМІ сервісу для адміністрування і обліку роботи автомобільної пар. Ідеальним захистом від ЕМІ виявилось б повне укриття приміщення, в якому розміщена радіоелектронна апаратура, металевим екраном. Водночас зрозуміло, що практично забезпечити такий захист у ряді випадків неможливо, тому що для роботи апаратури часто потрібно забезпечити її електричний зв'язок із зовнішніми пристроями. Тому використовуються менш надійні засоби захисту, такі, як струмопровідні сітки, або плівкові покриття для вікон, щільникові металеві конструкції для повітрезабірників і вентиляційних отворів і контактні пружинні прокладки, розміщені по периметру дверей і люків.

Більш складною технічною проблемою рахується захист від проникнення ЕМІ в апаратуру через різноманітні кабельні входи. Радикальним рішенням даної проблеми міг би стати перехід від електричних мереж зв'язку до практично не схильних до впливу ЕМІ волоконно-оптичних. Проте заміна напівпровідникових приладів у всьому спектрі виконуваних ними функцій електронно-оптичними пристроями можлива тільки у віддаленому майбутньому. Тому в даний час в якості засобів захисту кабельних входів найбільші широко використовуються фільтри, у тому числі волоконні, а також іскрові розрядники, металлоокисні варистори та ін. [42].

Металлоокисні варистори є напівпровідниковими приладами, що різко підвищують свою провідність при високій напрузі. Проте, при застосуванні цих

приладів у якості засобів захисту від ЕМІ варто враховувати їх недостатньо високу швидкодію і погіршення характеристик при кількарразовому впливі навантажень. Ці недоліки відсутні у високошвидкісних зенеровських діодах, дія яких заснована на різкій лавиноподібній зміні опору від високого значення практично до нуля, при перевищенні прикладеної до них напруги граничного розміру. Крім того на відміну від варисторів характеристики зенеровських діодів після багатократних впливів високих напруг і переключень режимів не погіршуються.

Найбільш раціональним підходом до проектування засобів захисту від ЕМІ кабельних входів є створення таких роз'ємів у конструкції яких передбачені спеціальні заходи, що забезпечують формування елементів фільтрів і установку вмонтованих зенеровських діодів. Подібне рішення сприяє одержанню дуже малих значень ємності й індуктивності, що необхідно для забезпечення захисту від імпульсів, що мають незначну тривалість і, отже, потужну високочастотну складову.

4.3 Висновки до четвертого розділу

В цьому розділі розглянуто важливі питання охорони праці та безпеки в надзвичайних ситуаціях, зокрема описано закордонний досвід організації охорони праці в ІТ-компаніях та виконано оцінку дії на елементи комп'ютерної системи.

ВИСНОВКИ

В результаті роботи над кваліфікаційною роботою було запропоновано та розроблено декілька алгоритмів: сортування зображень, змішування зображень, колірна корекція, зшивка пари зображень, циліндрична проекція, створення повноцінної панорами, а також візуалізація готової панорами з використанням веб-технологій.

У процесі дослідження та реалізації було розглянуто та досліджено існуючі методи пошуку ключових точок: SIFT, ORB, BRISK, AKAZE, а також методи їх зіставлення (BFMatcher, FLANN) та метрики (відстані Хеммінга, Евкліда, Манхеттена, k-середніх).

Для реалізації самого зшивання було розглянуто афінні перетворення та гомографія, функції відповідності RANSAC, LMEDS, PROSAC.

В результаті проведеного дослідження можна зробити висновок, що метод пошуку особливостей BRISK має найвищу продуктивність - найбільше знайдених ключових точок на тестових зображеннях і найменший середній час пошуку. Найкращий спосіб зіставлення для нього - BFMatcher з функцією відстані Хеммінга і ймовірнісним методом RANSAC, оскільки ця комбінація дає максимальну кількість точок, за якими буде зшивання. Це все дозволяє припустити, що цей метод може бути досить ефективним для зшивання кадрів у панорамні зображення, в тому числі і в реальному часі. У той же час, результати можуть змінюватись на різних парах зображень і не можна виключати, що в якихось ситуаціях будуть кращі інші методи.

У свою чергу, ймовірнісний метод RANSAC показав себе найбільш ефективним в умовах дослідження на тестовій вибірці зображень.

ПЕРЕЛІК ДЖЕРЕЛ

1. How to Take 360° Photos at a Fixed NPP and Manually Stitch Them [Електронний ресурс]. – Режим доступу: <https://medium.com/cupix/how-to-take-360-photos-at-a-fixed-npp-and-manually-stitch-them-6abcb7af2cb7> (дата звертання: 17.12.2024).
2. Szeliski, R., 2011. Computer Vision-Algorithms and Applications. 3, pp: 181- 228.
3. Романський С.В. Детектування особливостей на зображеннях // Інформаційні моделі, системи та технології: Праці XII наук.-техн. конф. Тернопіль, 2024. с. 86.
4. R. Otero, M. Delbracio. Anatomy of the SIFT Method [Електронний ресурс]. – Режим доступу: <https://www.ipol.im/pub/art/2014/82/article.pdf> (дата звертання: 17.12.2024).
5. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. Speeded Up Robust Features [Електронний ресурс]. – Режим доступу: <https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Bay08.pdf> (дата звертання: 20.12.2024).
6. E. Rublee, V. Rabaud, K. Konolige, G. Bradski. ORB: an efficient alternative to SIFT or SURF [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/221111151_ORB_an_efficient_alternative_to_SIFT_or_SURF (дата звертання: 22.12.2024).
7. ORB (Oriented FAST and Rotated BRIEF) [Електронний ресурс]. – Режим доступу: http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_orb/py_orb.html (дата звертання: 27.12.2024).
8. Stefan Leutenegger, Margarita Chli, Roland Siegwart: «BRISK: Binary Robust Invariant Scalable Keypoints». Computer Vision (ICCV), pp. 2548 –2555, 2011.

9. S. Grewenig, J. Weickert, C. Schroers, A. Bruhn: «Cyclic Schemes for PDEBased Image Analysis», In International Journal of Computer Vision, 2013.
10. Стандартні методи кластеризації даних [Електронний ресурс]. – Режим доступу: http://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec2.pdf (дата звертання: 29.01.2024).
11. Гороховатський В.О., Творошенко І.С. Методи інтелектуального аналізу та оброблення даних: навч. посібник. – Харків: ХНУРЕ, 2021. – 92 с.
12. Мангеттенська метрика [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мангеттенська_метрика (дата звертання: 10.01.2025).
13. Mining of Massive Datasets [Електронний ресурс]. – Режим доступу: <http://infolab.stanford.edu/~ullman/mmds.html> (дата звертання: 13.01.2025).
14. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
15. Richard Szeliski. Computer Vision: Algorithms and Applications / Texts in Computer Science. Springer Science & Business Media, 2011. p. 812.
16. M. A. Fischler and R. C. Bolles, «Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,» Communications of the ACM, vol. 24, no. 6, pp. 381-395, 1981.
17. Least Median of Squares Regression. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/242464484_Least_Median_of_Squares_Regression (дата звертання: 17.01.2025).
18. O. Chum and J. Matas, «Matching with PROSAC-progressive sample consensus,» in Computer Vision and Pattern Recognition, San Diego, CVPR, 2005, pp. 220-226.
19. Романський С.В. Проекції панорамних зображень // VIII Міжнародна студентська науково-технічна конференція «Природничі та гуманітарні науки. Актуальні питання». Тернопіль, 2025. с. 194.

20. CIELAB color space [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/CIELAB_color_space (дата звертання: 26.01.2025).
21. Stitching and Blending. Kari Pulli [Електронний ресурс]. – Режим доступу: <https://web.stanford.edu/class/cs231m/lectures/lecture-5-stitching-blending.pdf> (дата звертання: 27.01.2025).
22. ImageBlending [Електронний ресурс]. – Режим доступу: https://piazza.com/class_profile/get_resource/hz5ykuetdmr53k/i0zbj8uvcao7lp (дата звертання: 28.01.2025).
23. VRWorks - 360 Video - Multiband Blending [Електронний ресурс]. – Режим доступу: <https://developer.nvidia.com/vrworks/vrworks-360video/multiband-blending> (дата звертання: 30.01.2025).
24. Image Blending Using Laplacian Pyramids [Електронний ресурс]. – Режим доступу: <https://becominghuman.ai/image-blending-using-laplacian-pyramids-2f8e9982077f> (дата звертання: 10.02.2025).
25. Документація OpenCV [Електронний ресурс]. – Режим доступу: <http://opencv.org/> (дата звертання: 10.02.2025).
26. Feature Matching. OpenCV-Python Tutorials [Електронний ресурс]. – Режим доступу: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html (дата звертання: 12.02.2025).
27. Edge blending using commodity projectors [Електронний ресурс]. – Режим доступу: <http://paulbourke.net/miscellaneous/edgeblend/> (дата звертання: 18.02.2025).
28. High level stitching API (Stitcher class) [Електронний ресурс]. – Режим доступу: https://docs.opencv.org/3.4/d8/d19/tutorial_stitcher.html (дата звертання: 28.02.2025).
29. Panorama photo stitcher «Hugin» [Електронний ресурс]. – Режим доступу: <https://sourceforge.net/projects/hugin/> (дата звертання: 03.03.2025).
30. A panoramic image stitching tool made with OpenCV and Python [Електронний ресурс]. – Режим доступу: <https://github.com/kupiakos/image-stitcher> (дата звертання: 08.03.2025).

31. Zagorodna, N., Skorenkyu, Y., Kunanets, N., Baran, I., Stadnyk, M. Augmented Reality Enhanced Learning Tools Development for Cybersecurity Major. CEUR Workshop Proceedings., 2022, 3309, pp. 25–32. <https://ceur-ws.org/Vol-3309/short1.pdf>
32. Skorenkyu, Yu., Kozak, R., Zagorodna, N., Kramar, O., Baran, I. Use of augmented reality-enabled prototyping of cyber-physical systems for improving cyber-security education. Journal of Physics: Conference Series., 2021, 1840(1), 012026. <https://iopscience.iop.org/article/10.1088/1742-6596/1840/1/012026/pdf>.
33. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі. Книга 2. [навчальний посібник]. Львів : "Магнолія 2006", 2014. 312 с.
34. Vyacheslav Nykytyuk, Vasyl Dozorskyu, Nataliia Kunanets, Volodymyr Pasichnyk, Oleksandr Matsiuk, Ihor Bodnarchuk: Electrical Probe-Signal Processing and Criterion for the Determination of Time Parameters of the Teeth Filling Material Polymerization Process in Dentistry. 4th IDDM 2021: Valencia, Spain. P. 54-63
35. Сьогодні UA [Електронний ресурс] – Режим доступу: <https://www.segodnya.ua/lifestyle/fun/pochti-kak-u-googlechemudivlyayut-ofisy-ukrainskih-it-kompaniy--764025.html> (дата звертання 17.03.2025).
36. Як працюють в Google? Умови, в яких хочеться трудитися. [Електронний ресурс] – Режим доступу: <http://www.clevers.com.ua/articles-cleveradvertising-agency/success-stories/245-google2> (дата звертання 17.03.2025).
37. Офіс мрії: Робота в компанії Google. [Електронний ресурс] – Режим доступу: <http://bigpicture.ua/?p=187580> (дата звертання 19.03.2025).
38. Офіс Facebook: Репортаж із RMA SiliconTrip. [Електронний ресурс] – Режим доступу: <https://habrahabr.ru/company/rma/blog/103800/> (дата звертання 22.03.2025).
39. Класифікатор професій ДК 003:2010. [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/rada/show/va327609-10> (дата звертання 27.03.2025).

40. Закон України «Про охорону праці». [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звертання 29.03.2025).

41. ДНАОП 0.00-8.20-99. Порядок проведення експертизи електроустановок споживачів/ [Електронний ресурс] – Режим доступу: https://dnaop.com/html/43255/doc-%D0%94%D0%9D%D0%90%D0%9E%D0%9F_0.00-8.20-99 - (дата звертання 30.03.2025).

42. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

ДОДАТОК А
Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

МАТЕРІАЛИ

XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



18–19 грудня 2024 року

ТЕРНОПІЛЬ
2024

О. Прокопенко РОЗРОБКА НАВЧАЛЬНОЇ СИСТЕМИ ДЛЯ ЗБОРУ ПЕРСОНАЛЬНИХ ДАНИХ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ ТА СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ O. Prokopenko DEVELOPMENT OF A TRAINING SYSTEM FOR COLLECTING PERSONAL DATA USING AI AND SOCIAL ENGINEERING	77
I. P. Ralík CRM-СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ТОРГІВЛІ НА ПІДПРИЄМСТВАХ I. R. Ralík CRM-SYSTEM FOR RETAIL AUTOMATION AT ENTERPRISES	78
М. Ратншн РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ БЛОКЧЕЙНУ M. Ratyshyn DEVELOPMENT OF A DECENTRALIZED ELECTRONIC VOTING SYSTEM BASED ON BLOCKCHAIN	79
Д. Ревура, Р. Козак ІНТЕРПРЕТАЦІЯ СИСТЕМИ ОЦІНКИ ВРАЗЛИВОСТЕЙ, ЯК МЕТОДУ ДЛЯ ОЦІНКИ РИЗИКІВ D. Revura, R. Kozak INTERPRETATION OF A VULNERABILITY ASSESSMENT SYSTEM AS A METHOD FOR RISK EVALUATION	80
Я. Роран ВИКОРИСТАННЯ МЕТОДУ GRAD-CAM ДЛЯ ДІАГНОСТИКИ ХВОРОБИ АЛЬЦЕЙМЕРА Ya. Rohan USE OF THE GRAD-CAM METHOD FOR DIAGNOSIS OF ALZHEIMER'S DISEASE	81
Я. Роран ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ МЕДИЧНИХ ЗОБРАЖЕНЬ Ya. Rohan USE OF NEURAL NETWORKS FOR MEDICAL IMAGE ANALYSIS	83
М. М. Рокеш ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ КОРПОРАТИВНИХ КОМУНІКАЦІЙ M. Rokosh APPLICATION OF GENERATIVE ARTIFICIAL INTELLIGENCE FOR THE PROTECTION OF CORPORATE COMMUNICATIONS	84
С. В. Романський ДЕТЕКТУВАННЯ ОСОБЛИВОСТЕЙ НА ЗОБРАЖЕННЯХ S. V. Romanskyi FEATURE DETECTION IN IMAGES	86
Б. Слупський, Р. Козак АВТОМАТИЗОВАНІ ІНСТРУМЕНТИ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ТА ЇХ ЕФЕКТИВНІСТЬ B. Slupskyi, R. Kozak AUTOMATED PENETRATION TESTING TOOLS AND THEIR EFFECTIVENESS	87
О. В. Смик МЕТОДИ ТА ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ СИСТЕМИ ОБ'ЄКТУ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ ВІД ВНУТРІШНІХ ЗАГРОЗ O. V. Smyk METHODS AND MEANS OF PROTECTING THE INFORMATION AND COMMUNICATION SYSTEM OF AN INFORMATION ACTIVITY OBJECT FROM INTERNAL THREATS	88

УДК 514.18

С. В. Романський

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДЕТЕКТУВАННЯ ОСОБЛИВОСТЕЙ НА ЗОБРАЖЕННЯХ

UDC 514.18

S. V. Romanskyi

FEATURE DETECTION IN IMAGES

У комп'ютерному зоріснує важливий напрямок - *детекція особливостей (feature detection)* на зображеннях, з яких, власне, ключові точки і визначаються як такі. Під цим можна мати на увазі не просто точки, як пару координат, а деякі області на зображеннях, для знаходження яких і існують різні методи та алгоритми.

Власне особливості зображення може бути різних видів:

- *ребра (edges)* – межі між двома областями зображень, це можуть бути випадкові контури та стики різних об'єктів; практично часто набір точок, які мають велику величину градієнта;

- *кути/точки інтересу, ключові точки (corners / interest points, фактично синоніми)*. Спочатку продукт аналізу ребер з метою знаходження різних змін у напрямку (тобто кутів), потім були розроблені алгоритми, що не потребують явного обчислення ребер (наприклад, шукають сильне викривлення градієнта зображення). Зазначимо, що знайдені ключові точки зовсім не обов'язково є кутами об'єктів сцени у прямому розумінні, проте вони – особливості, які враховуються;

- *сфери / ключові області, ключових точок (blobs / regions of interest or interest points)*. Фактично опис областей зображення. Так само як і кути вважаються ключовими точками (і мають фактичний центр), однак вони мають свої алгоритми розрахунків.

Різні алгоритми знаходження особливостей завжди схожі в одному – як тільки особливість знайдена, відбувається витяг (extraction) невеликої області навколо неї. Результат – дескриптор або векторні особливості. Це стосується і кутів і сфер. Різні алгоритми пошуку особливостей пропонують різні дескриптори.

Методи зіставлення дескрипторів (matchers) ключових точок на різних зображеннях – нарізний камінь знаходження шаблонів на сцені, можливість комп'ютерного зору для вирішення задачі розпізнавання образів.

Ці способи зіставлення можуть бути відмінними для різних алгоритмів пошуку ключових точок та різних дескрипторів.

Для виділення ключових точок існує кілька основних методів, які вже реалізовані в різних програмних засобах і широко використовуються на практиці. Методи як мінімум включають алгоритми пошуку, що повертають список ключових точок, а також зазвичай (але не завжди) мають вбудовані алгоритми витягнення дескрипторів. Якщо вбудованих алгоритмів витягнення немає, то існують реалізовані окремо, здатні витягувати із зображення дескриптори певних форматів за знайденими ключовими точками.

Основні методи пошуку ключових точок на зображеннях, що мають доступну, готову для використання програмну реалізацію, це SIFT, SURF, ORB, BRISK та AKAZE. Алгоритмічно це різні методи, що тяжіють до різних особливостей зображень.

Ці методи пошуку ключових точок найбільш популярні і поширені (є й інші, зазвичай модифікації вказаних), крім того, важливо те, що їх реалізації є в одній відкритій бібліотеці комп'ютерного зору OpenCV, що робить неактуальним їх реалізацію з нуля (або використання різних джерел/бібліотек) і дає можливість порівнювати їхню роботу в ідентичних умовах.

Міністерство освіти і науки України
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія



*Студентське наукове
товариство*



VIII МІЖНАРОДНА
студентська науково - технічна конференція

**"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.**

АКТУАЛЬНІ ПИТАННЯ"

24-25 квітня 2025 р.

(збірник тез конференції)

Тернопіль 2025

Мищак В. ДОЦІЛЬНІСТЬ ВЛАСНОГО АВТОМОБІЛЯ В СУЧАСНИХ МІСТАХ	178
Монастирський М. РОЗРОБКА СЕРВІСУ-ПОРТФОЛІО ДЛЯ РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ FLASK	179
Мордованець А. КОМП'ЮТЕРНА СИСТЕМА БЛОКУВАННЯ РЕКЛАМИ В ЛОКАЛЬНІЙ МЕРЕЖІ НА БАЗІ RASPBERRY PI	180
Мульський С. СУЧАСНІ ПІДХОДИ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ УПРАВЛІННЯ КОМАНДНИМИ ПРОЦЕСАМИ	182
Нішита І. РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ РЕКОМЕНДАЦІЙ ФІЛЬМІВ З ВИКОРИСТАННЯМ REACT NATIVE	183
Олійник І. ТЕХНОЛОГІЇ ВЕБ-РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОГО РІШЕННЯ ДЛЯ ІНТЕРНЕТ-КОМЕРЦІЇ	184
Недошитко Л., Пйонтковський О. НЕЙРОМОРФНІ ЧИПИ	186
Підлипський Р. ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ФАЙЛІВ ЛОГУВАННЯ АНТИВІРУСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	188
Прачук М. ПРОГРАМНА СИСТЕМА ДЛЯ ІДЕНТИФІКАЦІЇ ПАРАМЕТРІВ ДИФУЗІЇ В МІКРОПОРИСТИХ ЗРАЗКАХ	190
Пришляк В. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛ ЛЯ РЕМОНТУ І ПРОДАЖУ АВТОМОБІЛЬНИХ ДЕТАЛЕЙ ЗА ДОПОМОГОЮ PHP	191
Пясецький Д. ПРОЕКТУВАННЯ ТА РОЗРОБКА МАРКЕТПЛЕЙСУ З МЕХАНІЗМОМ РЕКОМЕНДОВАНИХ ТОВАРІВ НА БАЗІ REACT TA C#	192
Боднарчук І., Ралік І. CRM-СИСТЕМА ЯК ВИСОКОПРОДУКТИВНА ОБЧИСЛЮВАЛЬНА СИСТЕМА	193
Романський С. ПРОЕКЦІЇ ПАНОРАМНИХ ЗОБРАЖЕНЬ	194

УДК 514.18

Романський С. – ст. гр. СІІІІІ-61

Тернопільський національний технічний університет імені Івана Пулюя

ПРОЕКЦІЇ ПАНОРАМНИХ ЗОБРАЖЕНЬ

Науковий керівник: професор, доктор технічних наук Марценюк В.П.

Romanskyi S.

Ternopil Ivan Puluj National Technical University

PROJECTIONS OF PANORAMIC IMAGES

Supervisor: professor, Dr. Sci. Vasyl Marceniuk

Ключові слова: проекція, панорамне зображення

Keywords: projection, panoramic image

Починаючи роботу з панорамними зображеннями, необхідно отримати деяке уявлення про різні існуючі проекції панорамних зображень [1] і про те, як вони впливають на зображення:

1. Прямокутний проекція. Визначаючи, що навколишнє середовище навколо нас набуває форми сфери, прямокутний проекція використовується, коли фотографується лише частина цієї сфери. Потім ця частина сфери проектується на плоску поверхню. Часто використовується для склеювання невеликої кількості зображень: добре адаптована для часткових панорам та мозаїчних зображень.

2. Циліндрична проекція. Використовуватиметься, коли на панорамі немає даних про зеніт або надир (полюси сфери) - зазвичай це однорядне зображення. Це проекція, котра застосовується для циліндричних панорам і деяких більших часткових панорам, які виглядають надто спотвореними у прямокутній проекції. Вона проектує частину сфери (циліндр) на плоску поверхню (зображення).

3. Сферична (еквідистантна) проекція. Ця проекція використовується для проектування усієї сфери на плоску поверхню. Результатом буде рівнопрямокутне зображення із співвідношенням сторін 2:1. Ширина чітко вдвічі більша за висоту. Це логічно, тому що охоплює 360 ° по горизонталі та 180 ° по вертикалі. Можна побачити, що у цій проекції верхня та нижня частини зображення виглядають дуже спотвореними, і це нормально. Ця проекція використовується як джерело для повної сферичної панорами.

4. Проекція Меркатора. Її можна розглядати як похідну від рівнопрямокутної проекції без спотворень у зеніті та надирі. Карти світу будуються у цій проекції. Для панорам це може бути перевагою, якщо потрібно, наприклад, надрукувати зображення сферичної панорами або використовувати зображення на веб-сайті. Але цю проекцію не можна використовувати створення інтерактивних панорам.

5. Кубічна проекція. Ця проекція використовується як розбиття рівнопрямокутного зображення (сферичної панорами). Принцип полягає в тому, що вся сфера перетворюється на 6 граней куба - передню, праву, задню, ліву, зеніт та надир. Кожне із 6 квадратних зображень має прямокутний проекцію. Ця проекція є дуже корисною, коли потрібно відредагувати зображення сферичної панорами. Не завжди легко редагувати деформовані зображення на прямокутному зображенні, а також

ДОДАТОК Б

Результати зшивання панорам у різних системах

Панорама набережної



Б33



Hugin



Stitching API

Панорама вулиці



Б33



Hugin



Stitching API