

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 124 Системний аналіз

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____

(підпис)

“ _____ ” _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Педану Радиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Математичні моделі і методи узгодження зображень на основі
ключових точок та перцептивного хешу

затверджена наказом по університету від 05 листопада 2021 р. № 1642 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2021 р.

3. Вихідні дані до роботи методи узгодження зображень; перцептивний хеш;
ключові точки; математичні моделі узгодження зображень

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Системний аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

РЕФЕРАТ

Пояснювальна записка: 56 с., 29 рис., 1 дод., 21 джерело.

КЛЮЧОВА ТОЧКА, УЗГОДЖЕННЯ ЗОБРАЖЕНЬ, ПЕРЦЕПТИВНИЙ ХЕШ, ОЗНАКИ ЗОБРАЖЕННЯ, ДЕСКРИПТОРИ, ГАУСОВЕ РОЗМИТТЯ, ORB, BRISK, SIFT

Об'єкт дослідження – пара цифрових зображень .

Мета роботи – застосування методів узгодження цифрових зображень.

Методи дослідження – ORB, BRISK, SIFT, Hamming distance, k-means.

Проведено дослідження методів узгодження зображень за допомогою ключових точок та використанням перцептивного хешу. Було проведено системний аналіз, промодельовано роботу особливих точок зображення та їх детекторів на зображенні. В методах узгодження зображень імплементовано застосування перцептивного хешу. Проведено порівняльний аналіз роботи алгоритмів.

На ряду з прийнятними підсумками виконаного дослідження було приведено такі приклади, в яких викладені у роботі алгоритми показали слабкі результати. Ці дані було проаналізовано. Наведена програмна реалізація алгоритмів могла б бути використана для вирішення задач у багатьох аспектах життя: автомобілебудуванні, баз даних з продажу нерухомості, пошуку фото і відео тощо.

ABSTRACT

Explanatory note: 56 p., 29 fig., 1 appendix, 21 source.

KEY POINT, IMAGE MATCHING, PERCEPTIVE HASH, IMAGE SIGNS, DESCRIPTORS, GAUSIC DIFFERENCE, ORB, BRISK, SIFT

The object of study is a pair of digital images.

The purpose of the work – the use of methods of matching digital images.

Research methods – ORB, BRISK, SIFT, Hamming distance, k-means.

A study of methods of image matching using key points and the use of perceptual hashes. The system analysis was carried out, the work of special points of the image and their detectors on the image was modeled. The use of perceptual hash has been implemented in image matching methods. A comparative analysis of the algorithms.

Along with the acceptable results of the study, the following examples were given in which the algorithms presented in the paper showed poor results. These data were analyzed. The given software implementation of algorithms could be used to solve problems in many aspects of life: automotive, real estate databases, photo and video search, etc.

ЗМІСТ

	С.
Вступ	7
1 Системний аналіз предметної області та постановка задач дослідження	8
1.1 Системний аналіз проблем узгодження зображень на основі ключових точок та перцептивного хешу	8
1.1.1 Вербальна модель системи.....	8
1.1.2 Морфологічний опис системи	9
1.1.3 Функціональна модель системи	9
1.1.4 Інформаційна модель системи	11
1.2 Аналіз сценаріїв вирішення задачі узгодження ключових точок	12
1.3 Формальна та змістовна постановка задачі	14
1.4 Постановка задач дослідження	15
2 Вибір та обґрунтування методу розв’язання	16
2.1 Комп’ютерне бачення	16
2.2 Огляд методів визначення особливостей зображення	19
2.2.1 Ключові точки зображення	19
2.2.2 Вибір сучасних детекторів ключових точок	21
2.2.3 Перцептивний хеш	31
2.3 Аналіз проблем і вибір критеріїв для узгодження зображень	33
2.3.1 Надмірність у детекторах ключових точок зображення	33
2.3.2 Критерій накладання детекторів	36
3 Програмна реалізація	39
3.1 Мова програмування Python та бібліотека OpenCV	39
3.2 Алгоритм розв’язання задачі узгодження зображень на основі ключових точок та перцептивного хешу	41
3.3 Опис програми	43
4 Результати обчислювального експерименту та їх аналіз	44
Висновки	48
Перелік джерел посилання	49
Додаток А Програмний код	51

ВСТУП

Актуальність теми. Комп'ютерний зір є одною з ключових дисциплін, яка використовується у роботі з зображенням. Ці знання нерідко використовуються для розробки різноманітних додатків в яких можливо реалізувати аналітичні, дослідницькі та наукові задачі. Одним з найважливіших видів використання таких технологій є обробка медичних зображень. Викладений у роботі алгоритм для узгодження цифрових зображень з використанням ключових точок та перцептивного хешу зображення допомагає більш легше слідкувати за прогресом лікування та вивчати томографічні знімки.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є вивчити алгоритми узгодження зображень, ключових точок та перцептивного хешу. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «узгодження зображень»;
- застосування перцептивного хешу для первісного відсіювання;
- знаходження ключових точок зображення;
- описання ключових точок зображення за допомогою дескриптора;
- узгодження зображень за допомогою ключових точок пари цифрових зображень через зіставлення їхніх дескрипторів.

Об'єктом дослідження є обробка зображень

Предметом дослідження є методи узгодження зображень

Методи дослідження. У кваліфікаційній роботі використовуються методи узгодження зображень, алгоритми знаходження ключових точок, перцептивний хеш, порівняння ключових точок за критеріями надмірності, накладення, швидкодії.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблем узгодження зображень на основі ключових точок та перцептивного хешу

1.1.1 Вербальна модель системи

Об'єкт аналізу – «Цифрове зображення».

Предмет аналізу – «Алгоритми визначення ключових точок зображення та p-hash».

Точка зору: дослідник.

На виході з системи ми маємо отримати оціночне порівняння методів узгодження зображень. На вхід до нашої системи поступають дані у вигляді цифрового зображення. Управління системи – алгоритми узгодження зображень з ключових точок та перцептивного хешу. Механізми системи – дослідник та його програмне забезпечення.

Щоб описати, як модель функціонує у взаємодії із зовнішнім середовищем на рисунку 1.1 представлена модель «чорний ящик».

Вона включає в себе модель границі, модель зовнішнього середовища, виходи та входи.

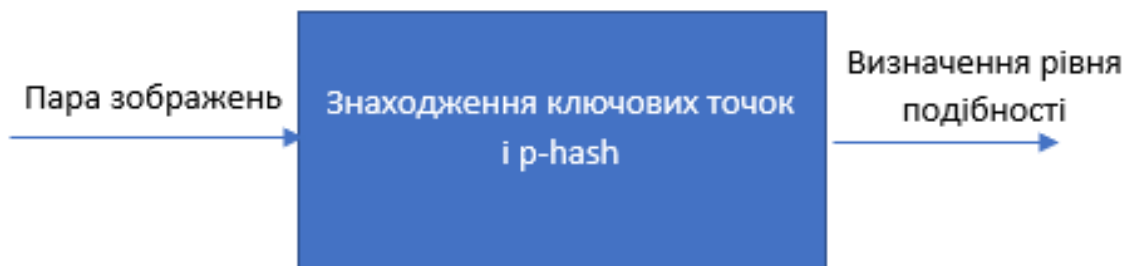


Рисунок 1.1 – Модель «чорний ящик»

1.1.2 Морфологічний опис системи

Морфологічний описання нашої задачі включає в себе, як одну з невід'ємних частин огляд поняття навколишнє середовище. Почнемо розгляд з рис. 1.2.

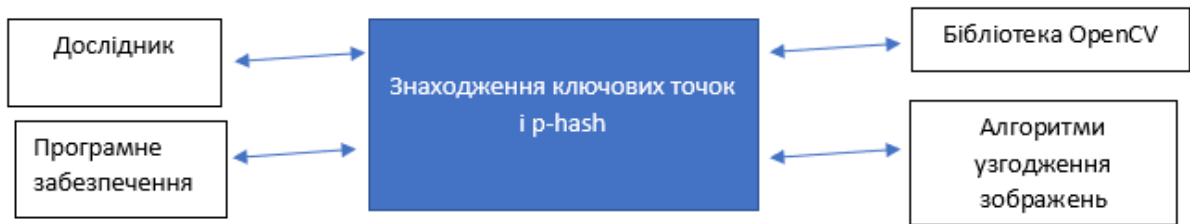


Рисунок 1.2 – Модель зовнішнього середовища системи

Морфологічним описом системи є побудова та структуризація системи. Для цієї задачі виконується опис всієї сукупності множин елементів поданої системи і те, що необхідно досягти з метою набору відносин між ними. Уся сукупність об'єктів, що знаходиться поза системою називається зовнішнім середовищем.

1.1.3 Функціональна модель системи

Зробимо функціональний опис моделі за допомогою IDEF0 (рис. 1.3).

Одним з сталих стандартів документування технологічних процесів можна вважати IDEF3. Таке документування відбуваються задля наочного моделювання сценаріїв дослідження. Сценарій – це послідовний опис змін властивостей об'єкту у рамках даного процесу, що зображено на рис. 1.6.Р



Рисунок 1.3 – Контекстна діаграма (рівню A-0)

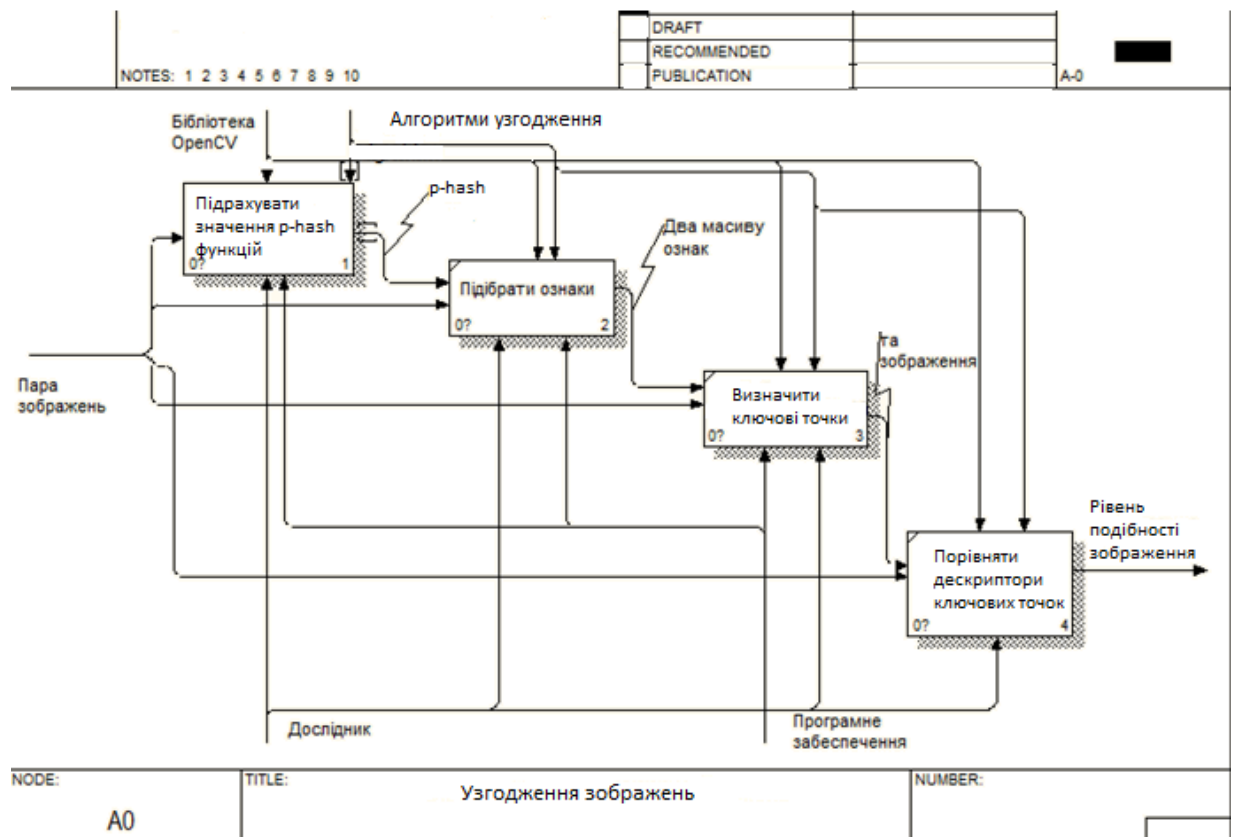


Рисунок 1.4 – Декомпозиція процесу роботи «Узгодження зображень» на рівні A0

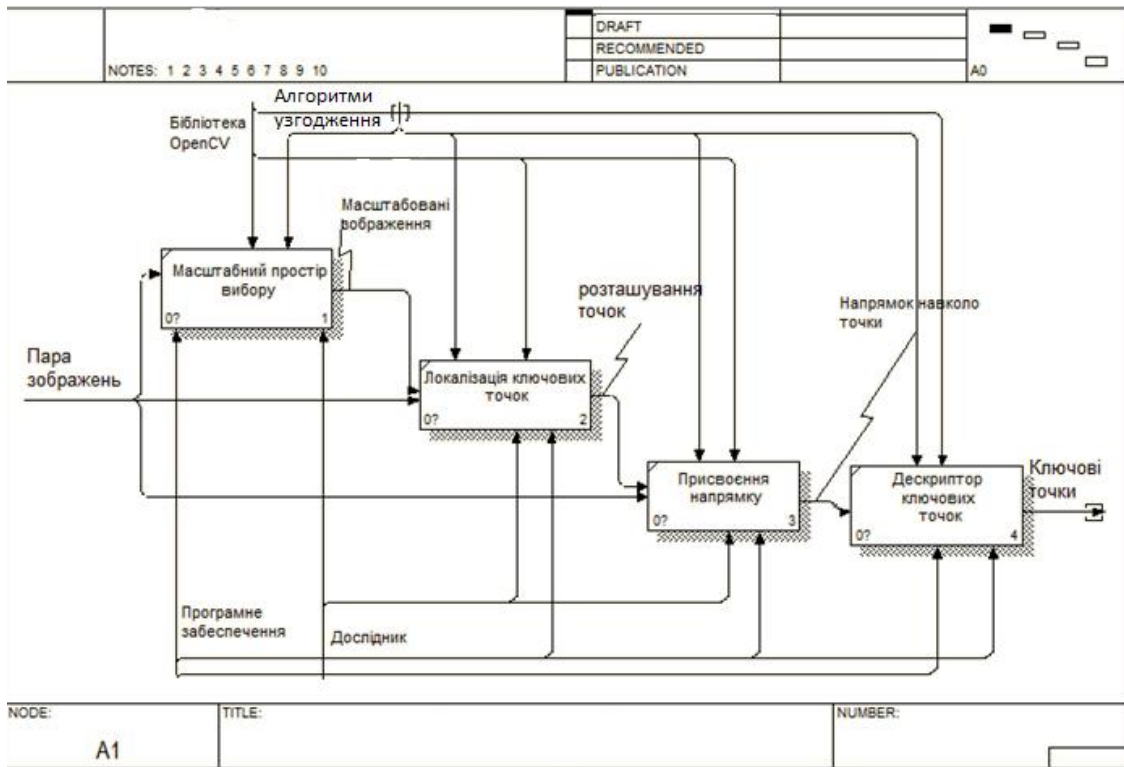


Рисунок 1.5 – Декомпозиція визначень ключових точок на зображенні



Рисунок 1.6 – Опис визначення ключових точок зображення

1.1.4 Інформаційна модель системи

Діаграма потоків даних, яка називається DFD є одним з основним шляхів по моделюванню функціональних вимог проектованої системи. Ця діаграма все більше використовується, щоб мати змогу описати шляхи обробки інформації і відображає змодельовану систему мережами пов'язаних між собою робіт, відображаючи операцію обміну інформацією в корпоративних системах обробки інформації. Показано на рисунку 1.7.

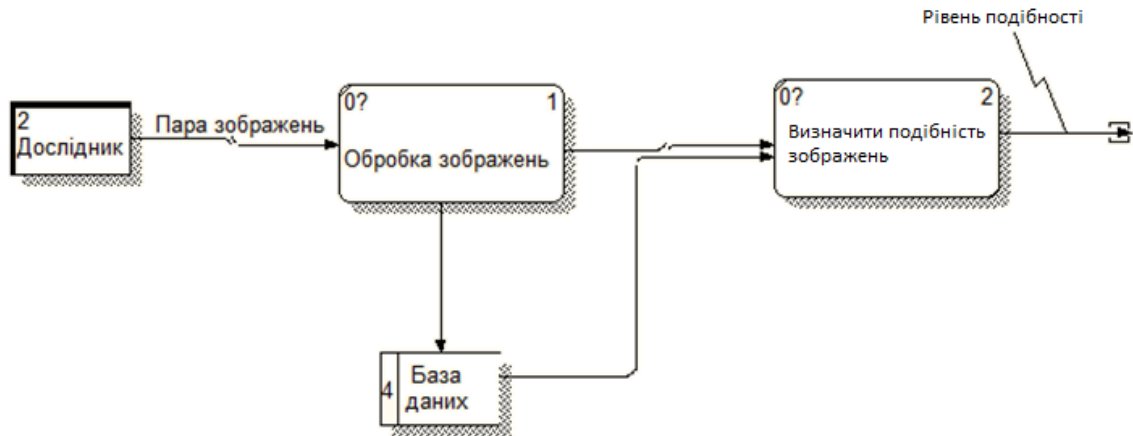


Рисунок 1.7 – Схема DFD

1.2 Аналіз сценаріїв вирішення задачі узгодження ключових точок

Методи узгодження ключових точок зображення опираються на алгоритми по знаходженню ключових точок зображення. У цій роботі вибір пав на 3 таких алгоритмі серед них: ORB, BRISK, SIFT.

SIFT або Scale Invariant Feature Transform – це ключові точки, які отримано за алгоритмом, що надає сталість до інваріантності до зміни розміру зображення та повороту. Також виявляє стійкість до зміни освітлення, шуму та зміни позиції спостерігача.

ORB або Oriented FAST and Rotated Brief – це алгоритм, який обчислює орієнтацію ключової точки зображення та бінарне узгодження з вже отриманих орієнтацій.

BRISK – це новий метод виявлення, опису та відповідності ключових точок. Комплексна оцінка контрольних наборів даних показує адаптивну, високоякісну продуктивність BRISK, як і в найсучасніших алгоритмах, хоча і при значно менших обчислювальних витратах (у випадках на порядок швидше, ніж SURF). Ключ до швидкості полягає у застосуванні нового детектора на базі FAST на основі масштабу і простору в поєднанні зі збіркою дескриптора бітового рядка з порівнянь інтенсивності, отриманих шляхом виділення вибірки

кожної околиці ключової точки.

Тепер ми можемо обрати метод, яким буде розв'язуватися задача.

Вибір методу дуже важливий для цієї задачі, бо є визначним в точності подальшого узгодження. Схема на рис. 1.8 відображає наступні критерії:

- критерій 1: розмір дескриптора;
- критерій 2: розмір патчу;
- критерій 3: кількість ключових точок, за уваги надмірності;
- критерій 4: складність програмної реалізації.

Тепер наведемо м:

- альтернатива 1: SIFT;
- альтернатива 2: ORB;
- альтернатива 3: BRISK.

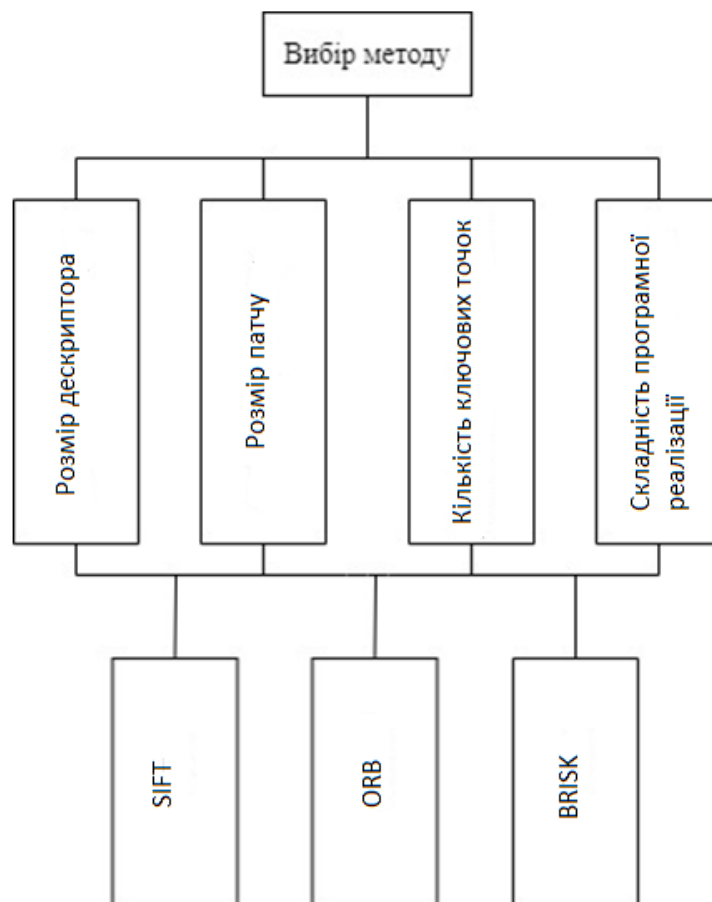


Рисунок 1.8 – Ієрархічна модель вибору

1.3 Формальна та змістовна постановка задачі

Узгодження зображень – це процес співставлення детекторів ключових точок зображення з метою дізнатися про ступінь їх схожості.

Сучасні алгоритми знаходження ключових точок зображення включають в себе детектори та дескриптори, що вимагають великої обчислювальної потужності. Для цього в роботі попереднім кроком до визначення ключових точок обчислюється перцептивний хеш. Це дозволяє не витратити час на повністю різні зображення.

Перцептивний хеш допомагає системі по визначенню обробляти більш великі масиви даних, що дозволяє використовувати такі алгоритми на реальних задачах.

Розглянемо формальну постановку задачі. Нехай дано деяка послідовність з двовірних RGB зображень.

$$F_1(x, y, u), \dots, F_N(x, y, u),$$

де $(x, y) \in \mathbb{R}^2$ – це декартові координати пікселя;

$u \in \mathbb{R}^3$ – це значення кольору.

Таким чином, для будь-якої пари сусідніх зображень виконується умова:

$$F_k(x, y, u) \cap F_{k+1}(x, y, u) \neq 0.$$

Необхідно знайти таке відображення Z , що

$$Z : F_1(x, y, u), \dots, F_N(x, y, u) \rightarrow S(x, y, u),$$

де $S(x, y, u)$ – панорамне зображення.

1.4 Постановка задач дослідження

Метою написання даної кваліфікаційної роботи є застосування алгоритмів пошуку ключових точок зображення та перцептивного хешу у задачі узгодження зображень. Розроблені етапи за якими може бути досягнена мету нашого дослідження:

- необхідність провести системний аналіз предметної області;
- вивчення методів знаходження ключових точок зображення та перцептивного хешу;
- розробити програмну реалізацію алгоритму узгодження;
- виконати експерименти на тестових зображеннях.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Комп'ютерне бачення

Комп'ютерне бачення – це сфера в науці, яка займається вивченням того, як комп'ютери можуть сприймати цифрові зображення та відео. З погляду інженерії, яка спрямована на розуміння та автоматизацію завдань, що можуть виконуватися у сфері діяльності людини.

Комп'ютерний бачення сприймається, як складова частина комп'ютерної науки, мета вивчення якої сфокусована на відтворення деяких можливостей зорового апарату людини, що дозволяє комп'ютерним програмам визначати і аналізувати об'єкти зображення та відео так само, як би це зробила людина, а в деяких випадках переважати здібності наших органів чуття.

Сучасне цифрове обладнання може обробляти зображення, які є недосяжними для недосконалої системи зору людини. Комп'ютерні програми також здатні порівнювати і вимірювати відмінності між кольором об'єкта з точністю яка неспівставна з людським оком. Задача аналізу змістовної частини зображення, на жаль, є складною задачею, яку ставлять комп'ютерам. Вчені та розробники вштовхувались з нею протягом всієї історії комп'ютерного зору. З погляду комп'ютера, цифрове зображення предмета є ніщо інше, як числове значення, або масив пікселів, представлений колірними моделями.

Для того щоб ми змогли отримати інформацію, яку нам надає наше навколишнє середовище, в якості інструмента вводу, використовується різне обладнання технічного зору. Ці датчики надають необхідну кількість вхідних даних.

Розглянемо перетворення простого растрового зображення у інформацію, яка може бути аналізована математичними алгоритмами комп'ютерного бачення. У цьому прикладі ми можемо побачити такі прості перетворення, що набуває інформація, рис. 2.1.

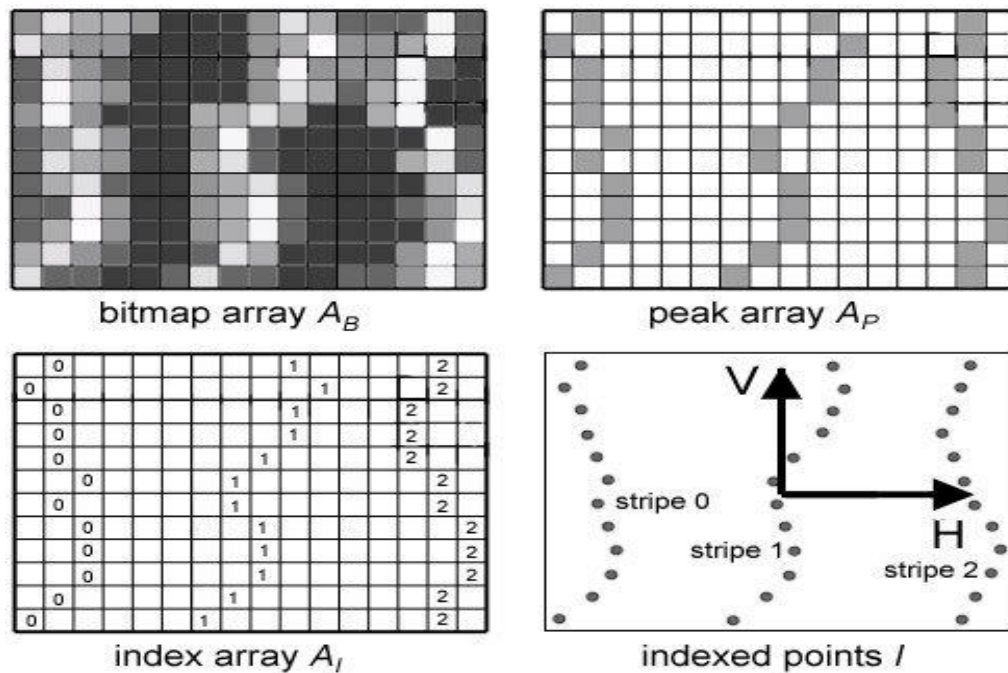


Рисунок 2.1 – Інформація з цифрового зображення як масив пікселів

Це дозволяє передати інформацію з навколишнього світу у сприятливий для аналізу об'єм даних. Що грає важливу роль у нашому сучасному житті.

Одна з важливих гілок застосування комп'ютерного бачення полягає в алгоритмах для розпізнавання осіб. Це набір технологій, що дозволяють нам узгоджувати зображення особливих точок людського профілю – ключових точок, з їх особистістю. Алгоритми комп'ютерного бачення обробляють риси людини в зображенні і порівнюють їх з вже обробленими даними в базах даних. Потенціальні користувачі можуть використовувати ці можливості у задачах розпізнавання людини задля знаходження та маркування зображення користувачів продукту (рис. 2.2).

За минулий час досягли значного розвитку технології змішаної реальності, що дозволяє сучасним електронним пристроям розміщати віртуальні об'єкти в зображення взяті з навколишнього середовища. Інструменти та апаратура комп'ютерного зору визначають об'єкти на зображенні, визначають їх місце у просторі та форми, щоб обробити їх. Це дозволить розмістити віртуальні об'єкти враховуючи глибину та інші геометричні параметри фізичного світу. (рис 2.3).

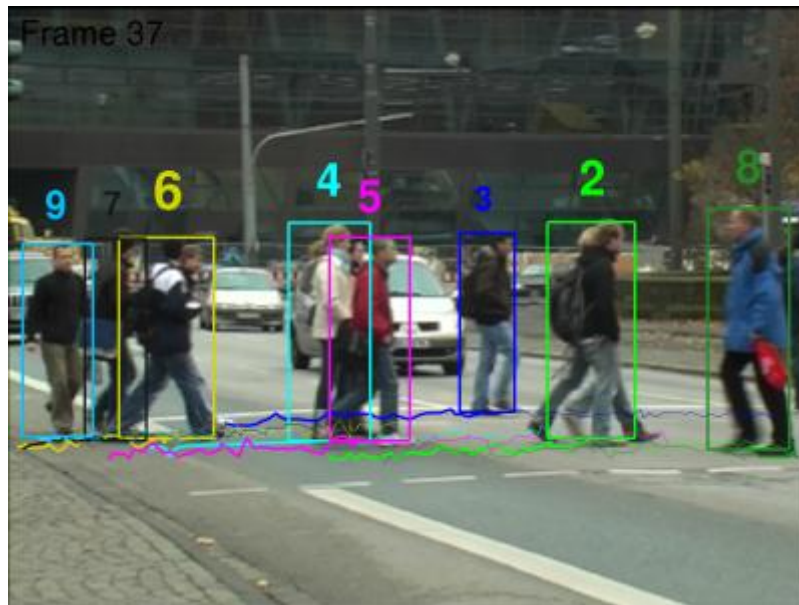


Рисунок 2.2 – Люди окреслені bound boxes

Однією з повсякденних задач комп'ютерного бачення є задача узгодження зображень. Ця технологія використовується компаніями по всьому світу за для зменшення обсягів даних, через зменшення обсягу дублікатів. Алгоритми комп'ютерного бачення визначають особливі місця на зображеннях. Описують ці місця інтересу та узгоджують їх за вибраними параметрами. Це допомагає бізнесу, який займаний у нерухомості скоротити частину повторюваних оголошень, що значно спрощує користувачам її вибір.



Рисунок 2.3 – AR-технології

2.2 Огляд методів визначення особливостей зображення

У нашому дослідженні для вирішення поставлених задач у якості базових методів були обрані ORB, BRISK та SIFT. В цьому розділі буде детально розглянуто наведені вище методи визначення ключових точок та їх математичні моделі.

2.2.1 Ключові точки зображення

Для узгодження зображень нам необхідно визначити їх особливості. Далі розглянемо особливості цифрового зображення – ключові точки та особливості загалом.

Особливість – це інформація, характерна риса, що має визначну роль для задачі узгодження зображень. На сам перед, в зображеннях, особливостями зображень виступають краї, кути та грані поверхонь об'єктів.

Ключова точки зображення – це точка, що виділяється серед усіх точок зі своїх околиць. Вона відповідає наступним якостям.

Визначеність – особлива точка має бути унікальною на тлі інших точок зображення.

Стійкість – вплив кольорової гами, контрастності та яскравості зображення не мають впливати на місцезнаходження точки інтересу.

Інваріантність – зміни повороту, масштабу та зміна ракурсу камери не повинні впливати на ключову точку.

Стабільність – детектор ключової точки повинен бути стійким до шуму в зображення певного рівня.

Зрозумілість – формат даних має бути придатним для подальшого аналізу.

Кількість – детектор має забезпечити потрібну кількість ключових точок для виявлення об'єкту на сцені.

Дескриптор ключової точки – це описова характеристика точки інтересу, що закодована в числовому або бінарному вигляді. Він дозволяє описати ключову точку, виділити її особливості на тлі множин пікселів на сцені. За допомогою цього складаються ключові пари особливостей певного об'єкту, при узгодженні двох різних цифрових зображень (рис. 2.4).

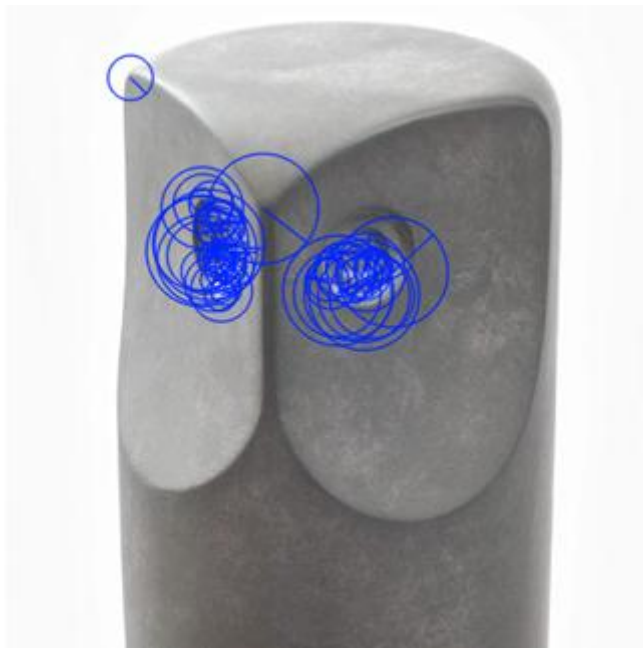


Рисунок 2.4 – КТ BRSK з визначеними регіонами

Така інформація не має змінюватися при геометричних змінах в зображенні – інваріантною. Це дає можливість визначити певну точку інтересу після геометричних перетворень.

Бінарні дескриптори – це стисле уявлення патчу або області навколо точки інтересу у вигляді двійкового рядка. Такий вид обумовлений результатом порівняння яскравості пікселів зображення у визначених алгоритмах місцях [1]. Такі бінарні дескриптори дуже цікаві для нашого дослідження, бо вони мають на меті зменшити час вилучення функції та необхідного розміру пам'яті для цифрових зображень у порівнянні з іншими детекторами.

Меша кількість вимог до пам'яті, що зазвичай вимагають 128 або 64 біт для зберігання. На відміну двійкові дескриптори займають лишень 256 або 512 біт. Це дає вигреш в пам'яті у чотири-вісім разів.

То того вони мають меншу складність обчислень, це можна побачити порівнюючи інтенсивність згладжених пікселів з часом на розрахунок градієнту. Так двійкові дескриптори витрачають менший часу на обчислення.

Отже, двійкові дескриптори мають переваги у прикладах з обмеженим часом. До таких можна віднести застосунки на мобільних пристроях, бо вони мають обмежену обчислювану потужність та пам'ять.

2.2.2 Вибір сучасних детекторів ключових точок

У роботі [2] викладено, що такий метод як ORB – Oriented FAST і Rotated BRIEF був розроблений як альтернатива SIFT та SURF, він забезпечує порівняну ефективність та швидкість. Його відмінність від цих методів полягає не лише у ліцензійних обмеженнях. Він забезпечує порівняно аналогічні показники відповідності коли йдеться про одночасну обчислювальну швидкість. ORB розроблений на базі детектору ключових точок і дескриптора. Вони розроблені на основі перевірених часом детектору ключових точок FAST [3] та дескриптору BRIEF [4]. На відміну від FAST та BRIEF, які не мають інваріантності до обертання ORB її має. До того ж залишаючись з гарними показниками обчислювальної вартості.

ORB починається з виявлення набору ключових точок із зображення за допомогою детектора FAST та використовує кутовий захід Гарріса, щоб визначити ключові точки для виключення некоректних точок. Для створення багато масштабних функцій використовується масштабна піраміда зображення і на кожному рівні піраміди виявляються набір функцій FAST, відфільтрованих Гаррісом. Оскільки FAST не забезпечує вимірювання кутової орієнтації, то для обчислення орієнтації ключових точок застосовується міра центроїда інтенсивності.

Будується дескриптор ключових точок зображення ORB на базі обертового бінарного дескриптора BRIEF з додаванням обертання. BRIEF дає нам шви-

дкість та ефективність. Але ж його результати стрімко падають, коли йде мова про обертання об'єкту в площині. Одне з можливих вирішень цього питання це обмежити – керувати BRIEF (sBRIEF) відповідно до змін, що відбуваються з орієнтацією точок інтересу. Цього можна досягти, якщо помножити матрицю двійкових тестів на кут їхньої орієнтації в просторі сцени. Тоді, BRIEF стане орієнтованим по напрямку особливих точок зображення, його відношення інтенсивності між оберненими парами зображень зміниться ближче до якоїсь сталої схеми і це підніме відношення між парами, що використано задля визначення двійкового дескриптора. Такі зміни у кореляції мають вплив у дискримінаційних можливостях дескриптора ключових точок. Існують дискримінаційні ознаки з двома властивості у парах вибірки. Перше – це коли не кореляція присутня у двох парах вибірки, що дозволить кожній окремій парі вносити зміну і донести якісно нові дані до дескриптора ключової точки зображення, таким чином, збільшується корисна інформація, яка надається дескриптором ключової точки зображення. Друга ознака – це визначена більша дисперсія пар, що має на меті особливість пор-різному реагувати на вхідну інформацію. Задля дотримання наведених вище особливостей використовується метод навчання, що допомагає у пошуку серед множини можливих пар вибірки та вибору оптимального набору пар з 256 проб з такими властивостями як викладеними вище. Такий бінарний дескриптор і є повернутим BRIEF або rotated-BRIEF він на порядок покращує дисперсію та кореляцію, алгоритму BRIEF.

Задля знаходження кутів у детекторі FAST розглядаються за чергою околиці по шістнадцять пікселів на околицях всіх пікселів множини p .

Така точка p може вважатися кандидатом на особливу точку зображення, коли в ній знаходиться N пікселів (у даній роботі ця кількість дорівнює 9) в патчі круглої форми з довжиною 16 пікселів, якщо всі значення N більше за $IP + t$ або мають значення менше $IP - t$, де IP – це яскравість пікселя p , а t – величина порогового значення. То при виконанні всіх цих умови, має місце далі досліджувати яскравість пікселів, що знаходяться на колі пронумеровані як: 1, 5, 9, 13. Коли трапляється, що у трьох пікселів патчу з чотирьох ми маємо вико-

нану умова $I_i < IP - t$ чи $I_i > IP + t$, $i = 1, 2, 3, 4$, можна вважати, що p є особливою точкою цифрового зображення.

Вибір лише чотирьох пікселів на патчі дозволяє алгоритму швидше відкинути непідходящих кандидатів на особливі точки, але в можуть траплятися випадки в яких мають місце набір різних особливостей зображені в межах одного патчу. Розглядаючи алгоритм ORB рекомендовано обирати максимальну кількість ключових точок зображення в кількості 500. В випадках коли їх кількість стає більшою за 500 то до них рекомендовано застосовувати детектор кутів Харріса [5]. Це дозволить зменшити множину ключових точок шляхом виключення найменш значущих з них.

Такий дескриптор ключових точок буде представлений у вигляді вектору. Його довжина буде складати 256, що зібрані на тлі результатів бінарних тестів околиць навколо точки інтересу. Розмір такого патчу складає 31 на 31 піксель. В ньому відбувається порівняння середніх значень яскравості між x і y , де x , y – патчі з розміром 5 на 5 пікселів:

$$\tau(I; x, y) := \begin{cases} 1: I_x < I_y, \\ 0: I_x \geq I_y, \end{cases}$$

де I – середня яскравість обраного патчу.

Щоб досягти сталої інваріантності до деяких фізичних перетворень область розрахунків дескриптора точки інтересу має бути орієнтована до цієї точки θ .

На усьому наборі $n = 256$ точок x_i і y_i створюється матриця S з розмірністю 2 на n . Наступним кроком S із залученням матриці повороту R_θ орієнтується на кут θ :

$$S_\theta = R_\theta S.$$

Таким чином ми записуємо вектор дескриптора ключових точок:

$$g_n(I, \theta) := f_n(I) | (x_i, y_i) \in S_\theta,$$

BRISK – один з найшвидших детекторів особливих точок зображення, який розроблено за методологією AGAST. Переваги цього детектора пов'язані з модифікаціями, які мають на меті виявляти якісно вищі особливі точки на зображенні та досягти стійкості до геометричних перетворень. Задля цього BRISK працює за принципом, який шукає локальні максимуми у площині зображення, окрім того, у просторі с масштабованим зображенням, додається FAST як оцінка надійності роботи. Але, дескриптор BRISK має приклад вибірки, у якому особливі точки розподілені на масштабованих концентричних колах у патчі навколо точки інтересу. Кожний ступінь сортування спроби розглядається, як малий патч у його теренах і розгладжується із застосуванням фільтра згладжування Гауса. Набори для великих відстаней засовуються як оцінка міри направлення патчу. Маленькі відстані застосовуються задля обрахунку кінцевого варіанту кандидата на дескриптор ключової точки. Використовуючи цей шаблон вибірки та граничних порогів відстані дескриптор ключової точки, що обмежений бітовим рядком, де $l = 512$ біт. Вважається, що дескриптор BRISK є класичним двійковим дескриптором ключових точок, складеним за допомогою згладжених порівнянь яскравості пікселів зображення, які подібні до BRIEF та ORB, BRISK має принципові переваги та недоліки.

Насамперед, BRISK користується з фіксованими схемами підбору спроб, яка відбирається вручну, що має на меті отримання більш рівномірної щільності точок множини на деякому заданому радіусі на околицях ключової точки зображення.

Ще дескриптор використовує значно меншу кількість особливих точок для відбору спроб, які будуть порівнюватися паралельно, це має на меті, обмеження складності знаходження значень інтенсивності.

Значною перевагою, є порівняння алгоритму BRISK обмежені в просторі

таким чином, що виходить зміна яскравості має бути в близьких до точки околицях. Він був представлений в 2011 році. Знаходження точок інтересу забезпечується алгоритмом FAST. Опис цих точок дескриптором BRIEF. В їх роботу внесено деякі вище описані зміни.

Для стійкості до зміни масштабу, повороту та зміни ракурси камери пропонується обирати точку інтересу з огляду на максимальне значення інтенсивності у піраміді, що складається з чотирьох октав c_i і 4 внутрішніх октав d_i , $i = 0, 1, 2, 3$. Вони сформовані, як зменшення зображення оригіналу у 2^i рази. Октави які представлені на (рис. 2.5) розміщені між c_i і c_{i+1} та представлені у вигляді зменшеної c_0 у 4^i разів. Знаходження точок інтересу в цих октавах здійснюється детектором FAST.

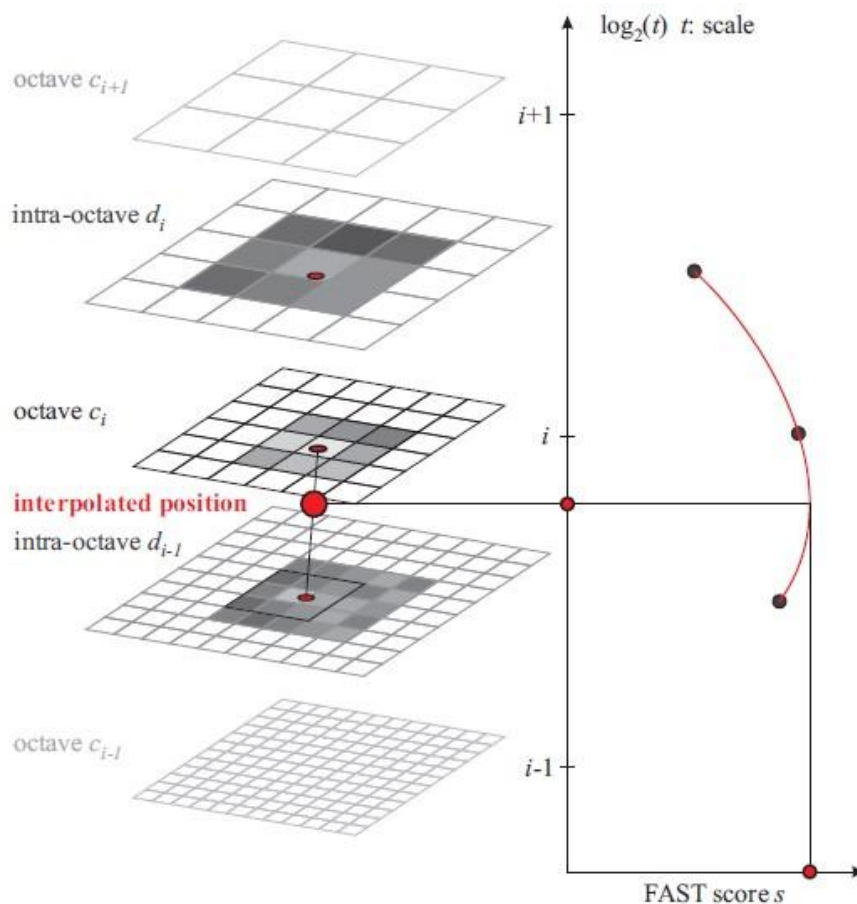


Рисунок 2.5 – Пошук точки інтересу за допомогою FAST

Ділянка навколо точки інтересу розділяється на 60 патчів p , що показано

на рисунку 2.6.

$$A = \{(p_i, p_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\}.$$

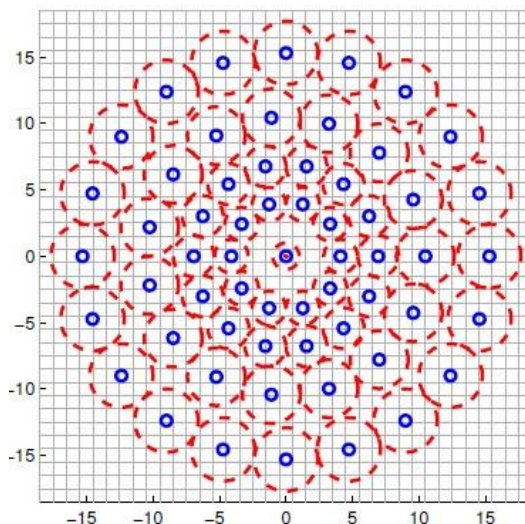


Рисунок 2.6 – Области для пошуку дескриптором BRISK

Множина A розділяється на 2а підмножин:

$$S = \{(p_i, p_j) \in A \mid \|p_j - p_i\| < \delta_{\max}\} \subseteq A,$$

$$L = \{(p_i, p_j) \in A \mid \|p_j - p_i\| < \delta_{\min}\} \subseteq A,$$

де $\delta_{\min} = 13,67t$, $\delta_{\max} = 9,75t$, t – розмір особливої точки зображення.

Розрахунок середнього значення градієнту множини має вигляд:

$$g \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{|L|} \sum_{p_j, p_i \in L} \left[(p_j - p_i) \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \right].$$

Дескриптор ключової точки зображення складений з бінарних рядків довжиною 512, вони складаються з результатів проведених тестів на множині S :

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_j^\alpha, \sigma_i); \\ 0, & \text{інакше,} \end{cases} \quad \forall (p_i^\alpha, \sigma_j^\alpha) \in S,$$

де $I(p_j^\alpha, \sigma_j)$ – інтенсивність на околі радіусу σ_j точки p , $\alpha = \arctg 2$, (g_y, g_x) – кут напрямку градієнта g .

Алгоритм знаходження ключових точок SIFT – масштабно-просторовий детектор. Його першим кроком є обчислення точки інтересу на зображеннях з різним масштабом. Далі задля підвищення стабільності точок інтересу відбувається фільтрація знайдених на сцені точок. Це дозволяє залишити найстабільніші точки. Для кожної найстабільнішої точки обирається головна орієнтація, вираховується дескриптор [2].

Загалом SIFT складається з таких етапів [6]:

- а) вираховується різниця піраміди гаусіанів, що дозволяє забезпечити інваріантність до масштабування;
- б) розрахунок екстремумів;
- в) уточнення точок інтересу;
- г) побудова дескрипторів ключових точок (які мають стійкість до геометричних перетворень та висвітлення, шуму, зміни положення камери).

SIFT здатний знаходити точки інтересу у формі крапель. Це є перевагою даного методу, бо такі точки інтересу інваріантні до всіх афінних перетворень. Краплі представляють собою такі структурні одиниці, які можуть бути описані масштабом і напрямком, координатами центру. Така структура виявляється найбільш складною, що трапляється серед всіх різноманітних форм ключових точок цифрового зображення і, безперечно, забезпечує стійке виявлення.

У SIFT [7, 8] будується масштабований простір зображення – набір зображень, згладжених фільтром Гауса

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

де (x, y) – координати точки;

σ – радіус розмиття.

На основі отриманого масштабу зображень здійснюється гаусова різниця $D(x, y, \sigma)$ – по-піксельне віднімання зображень в одній октаві з різним радіусом розмиття. Октаву формують зображення одного масштабу, розмиті фільтром Гауса з різним радіусом розмиття (4 зображення в одній октаві).

Гаусіан – зображення, розмите гаусовим фільтром:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

де $L(x, y, \sigma)$ – значення гаусіану у точці з координатами (x, y) ;

σ – радіус розмитнення;

$G(x, y, \sigma)$ – гаусове ядро;

$I(x, y)$ – значення початкового зображення.

Різниця гаусіанів – зображення, отримане шляхом по піксельного віднімання одного гаусіана вихідного зображення з гаусіаном з іншим радіусом зображення:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma).$$

Масштабований простір зображення – множина різних, згладжених деяким фільтром, варіантів вихідного зображення. Гаусовий масштабований простір є лінійним і інваріантним щодо зрушень, обертань, масштабу, що не зміщує локальні екстремуми, і має властивість напівгруп. Різна ступінь розмиття зображення гаусовим фільтром може бути прийнята за вихідне зображення, взяте в деякому масштабі.

Інваріантність до зміни масштабу зображення досягається у SIFT за допомогою визначення особливих точок для вихідного зображення, яке було масштабовано. Увесь масштабований простір пірамід гаусіан розбивається на де-

які ділянки, до того кожна наступна частина цього простору, займаного наступною октавою, менша за попередню у два рази. На кожному октаву, яку ми маємо на зображенні буде безліч гаусіанів зображення. У зв'язку з такою особливістю необхідно побудувати тільки деяке число гаусіанів N з визначеним кроком радіусу розмиття. З кроком такого ж радіусу будуються 2 додаткових гаусіанів, виходить, що ми маємо $N + 2$ гаусіанів. Розмір першого необробленого зображення наступної октави дорівнює розміру зображення попередньої октави номеру N . Згладжуючи зображення за допомогою гаусового фільтра за алгоритмом який наведено вище, ми будемо зображення як на рисунку 1.3.

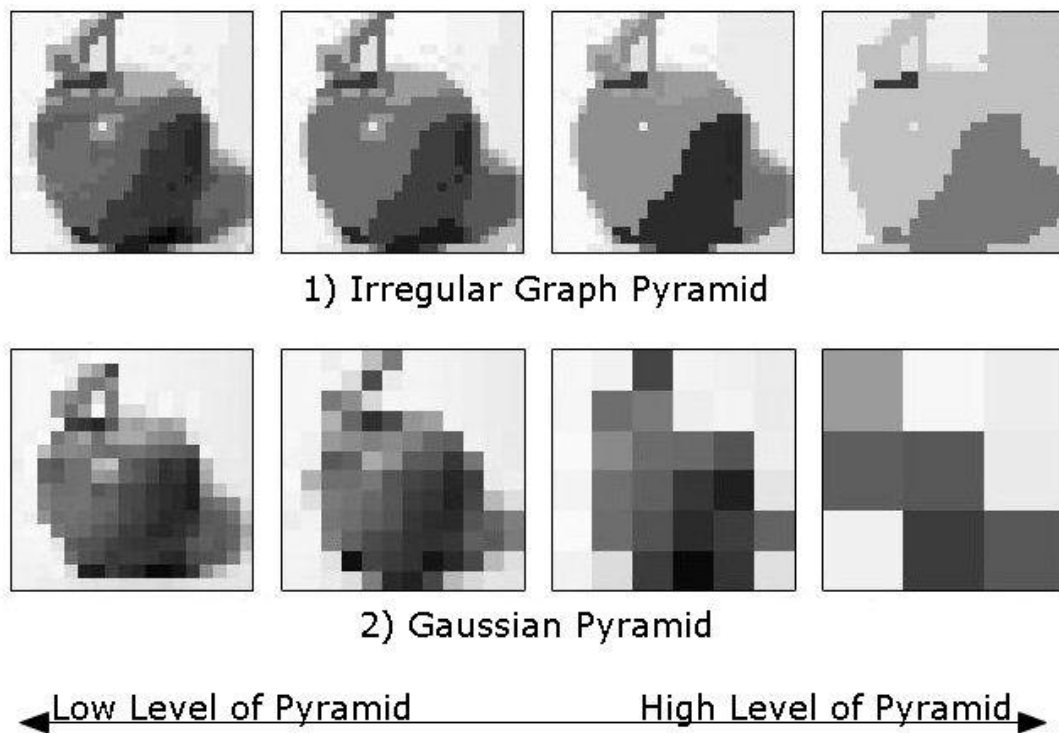


Рисунок 2.7 – Зміна зображення після перетворення

За допомогою такого виду маніпуляцій ми забезпечили інваріантність до зміни масштабу. Наступним кроком ми повинні визначити екстремуми, що будуть розглядатись як потенційний список точок інтересу цифрового зображення. На цьому ж кроці будується піраміда різниць гаусіанів, вона вираховується як різниця N та $N+1$ зображень піраміди гаусіанів. Кількість зображень в такій піраміді буде дорівнювати $N + 1$. Наведені вище піраміди наведені на рисунку

2.8. На схемі представлено, після яких кроків отримано піраміду різниць гаусіанів.

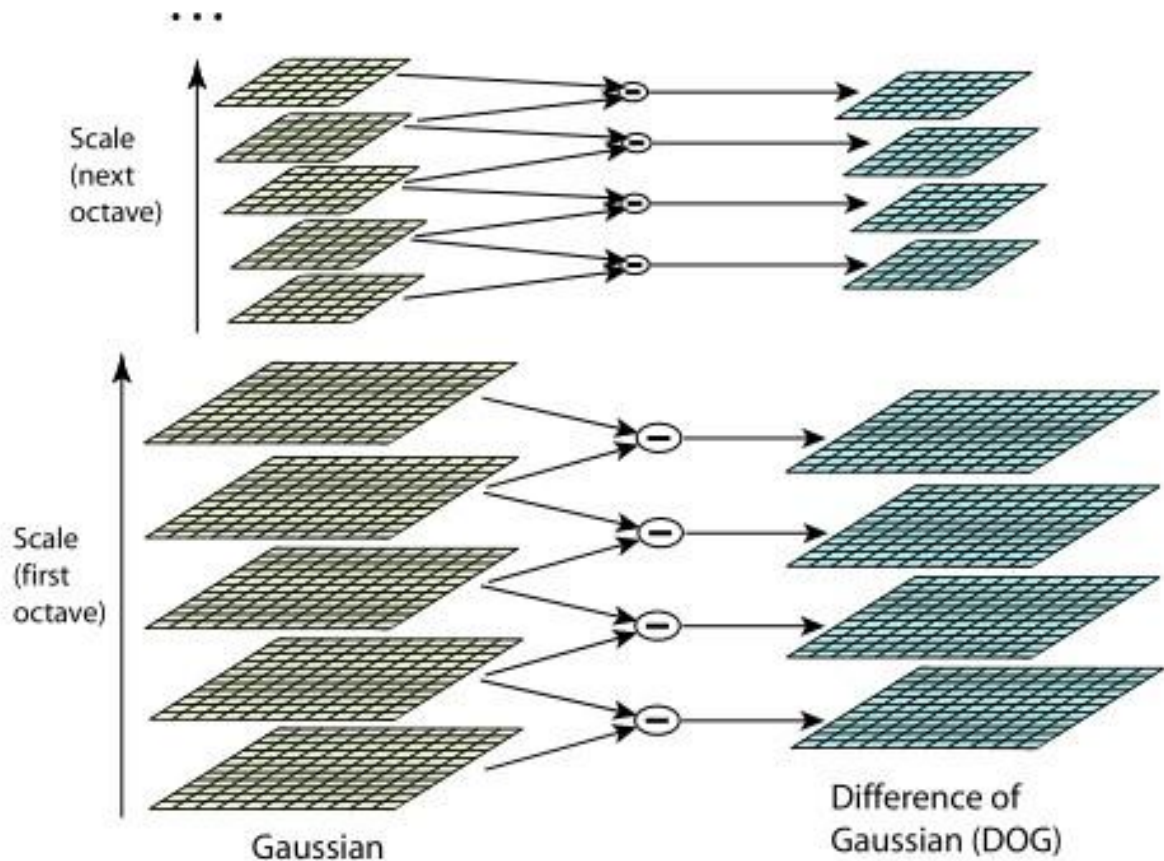


Рисунок 2.8 – Схема відмінностей між Gaussian та DOG

На рисунку наведеному нижче приведена піраміда гаусіанів, а поряд з нею піраміда їх різниць. На рисунку схематично зображено, як кожна різниця гаусіан складається з двох сусідніх гаусіан, кількість різниць гаусіан на один менше за кількість гаусіан. Коли відбувається перехід до наступної октави масштаб цифрового зображення зменшується у два рази.

Коли крок з побудови пірамід гаусіан виконано вважається, що точка є особливою, якщо вона є локальним екстремумів різниці гаусіан. Щоб знайти екстремуми вибрано метод, який схематично зображено на рисунку 2.9.

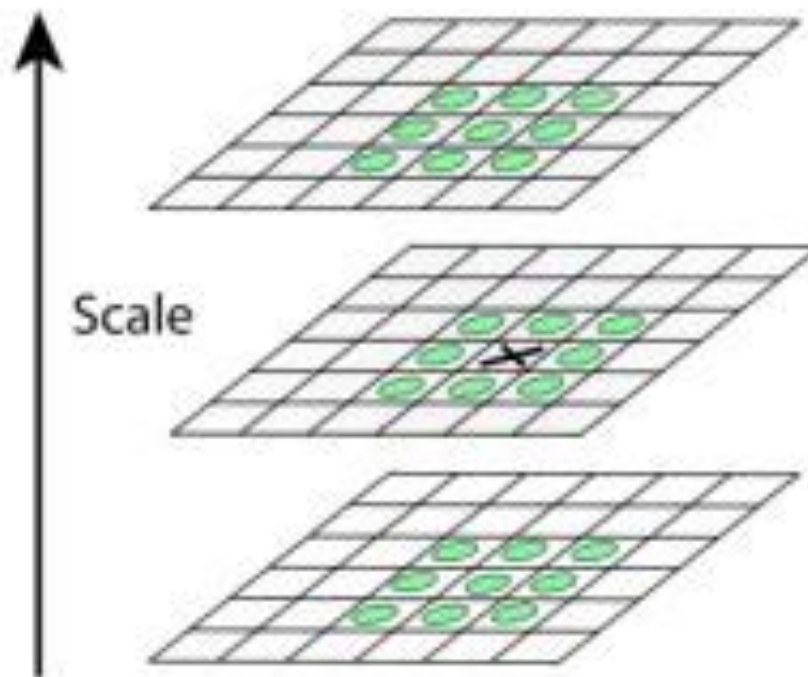


Рисунок 2.9 – Екстремуми різниці гаусіан

Коли значення різниці гаусіан позначено хрестиком, і таких позначень більше або менше усіх значень в точках, які позначають кружечками, говорять, що така точка може вважатися точкою екстремуму. У вибраному алгоритмі SIFT ми мали одні кращі результати визначення особливих точок зображення і підрахунку дескрипторів таких точок [9, 10]. Тому цей метод був використаний для досліджень в нашій роботі.

2.2.3 Перцептивний хеш

Через постійно зростаючу цифровізацію аутентифікація мультимедійного вмісту стає все більш важливою. Аутентифікація в цілому означає відповідь на питання про автентичність об'єкта чи ні. Тобто, якщо деякий об'єкт збігається з заданим оригінальним об'єктом. Аутентифікація сильно залежить від типу об'єкта. Під час автентифікації виконуваного файлу важливо, щоб кожен окре-

мий біт точно відповідав вихідному виконуваному файлу. Для ров'язання таких задач використовують функції криптографічного хешу. Для перевірки автентичності мультимедійного вмісту краще підходять інші методи. Мультимедійний об'єкт, наприклад зображення, може мати різні цифрові аналогічні або схожі зображення, які всі виглядають однаково зі сторони сприйняття людиною. Різні цифрові подання можуть бути результатом таких етапів обробки зображень, як обрізання, стиснення або вирівнювання гістограми. Кожен із цих кроків обробки зображень змінює двійкове представлення зображення. Тому використання криптографічної хеш-функції для аутентифікації модифікованих зображень не працює для мультимедійного контенту. Для визначення "рівності" мультимедійного вмісту пропонуються так звані перцептивні хеш-функції. В останні роки спостерігається зростання наукового та промислового інтересу до технологій перцептивного хешування. Такі функції були розроблені для різних типів цифрових носіїв (наприклад, аудіо, зображення або відео). Перцептивні хеш-функції визначають певні особливості з мультимедійного вмісту та обчислюють хеш-значення на основі цих особливостей. Під час автентифікації мультимедійного об'єкта порівнюються хеш-значення початкового об'єкта та об'єкта, що підлягає автентифікації за допомогою специфічних функцій. Такі функції обчислюють оцінку відстані або подібності між двома перцептивними хеш-значеннями. Остаточний вердикт ґрунтується на обраному пороговому значенні. У даній роботі загальний термін "медіа-об'єкт" використовується для мультимедійного вмісту, такого як аудіо, зображення або відеофайлу. Медіа об'єкт можна змінити, використовуючи різні операції. Прикладом операції із зображенням є обрізання зображення на 10%. Операція - загальний термін для модифікації або маніпуляції. Модифікація – операція, яка не змінює суттєвого змісту медіа-об'єкта. Після модифікації медіа-об'єкт все ще очікується виявити автентичним за допомогою функції перцептивного хешу. Маніпуляція - операція, яка суттєво змінює вміст медіа-об'єкта. Після маніпуляції медіа-об'єкт, як очікується, буде виявлений як неавтентичний за допомогою перцептивної хеш-функції.

2.3 Аналіз проблем і вибір критеріїв для узгодження зображень.

2.3.1 Надмірність у детекторах ключових точок зображення

Узгодження цифрових зображень є невід'ємною складовою багатьох додатків на основі технологій комп'ютерного бачення. В них входять реєстрація цифрових зображень та розпізнавання об'єктів на сцені, оскільки такі задачі визначаються у знаходженні відповідності ключових точок між об'єктами зображення декількох сцен. Узгодження цифрових зображень вважається одним з основних прийомів для отримання інформації про подібність таких об'єктів.

Прийнятий підхід до узгодження цифрових зображень полягає у виявленні множини особливих точок на сцені цифрових зображень та визначення локального сусідства в певному радіусі кандидатів на ключової точки цих зображень. Далі аналізуються особливості, після знаходження найближчих сусідів у патчі навколо дескриптора ключових точок. Так відбувається узгодження ключових особливостей та прийняття рішення щодо їх подібності. Розглянемо, функцію f_1 на цифровому зображенні I_1 . Яким чином можна визначити кращу узгодженість кандидата на цифровому зображенні I_2 . По-перше треба визначити функцію відстані ключових точок, яка аналізує два дескриптора цих функцій. Далі, обчислюється відстань у всіх функціях на цифровому зображенні I_2 . Залишилось визначити найближчих сусідів, що мають мінімальну відстань.

Одним з простіших способом пошуку подібності у зображенні є підрахунок суми квадратних різниць або евклідової норми. Складність відкинути неоднозначні результати збігів є найважливішим недоліком розгляданого методу. Бо значна частина функцій цифрового зображення може не мати подібної відповідності на іншому зображенні. На Рис. 2.10 приводиться некоректне визначення кращого кандидата f_2 для визначених характеристик f_1 для підрахунку взято мінімальну відстань. Для того щоб уникнути таких результатів нам потрібно використати точніший спосіб узгодження зображень. Функції мають замале чи-

сло коректних збігів. Рішенням такого питання є підрахунок співвідношення відстаней між найкращим матчем зі збігом який на другому місці. Це ілюстровано на рис. 2.11. Виходить, що для правильного узгодження відстаней у співвідношенні ці відстані повинні бути менше ніж певна порогова величина, а неперевірена чи недостатня відповідність буде знаходитись на малій відстані.

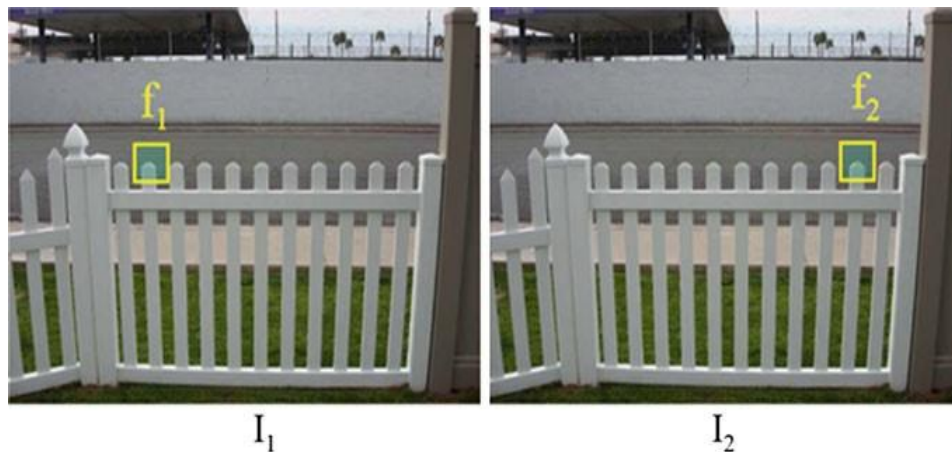


Рисунок 2.10 – Узгодження пари функцій на двох цифрових зображень за допомогою SSD, де f_2 – ближчий сусід до f_1

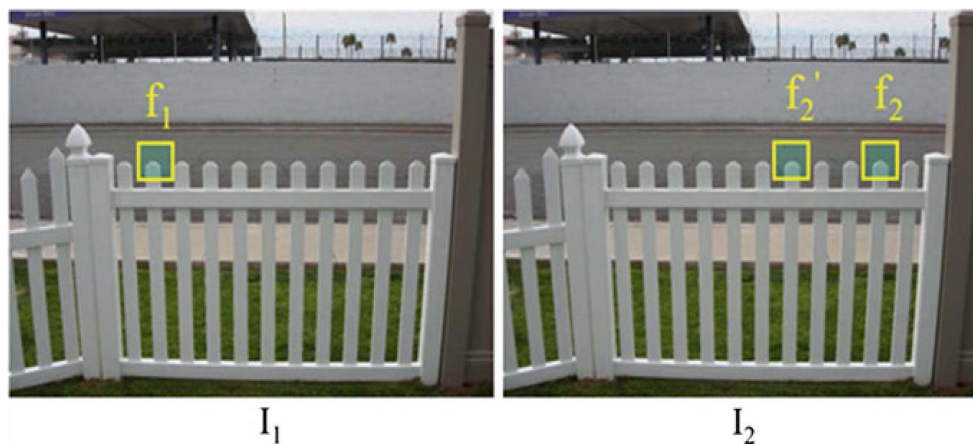


Рисунок 2.11 – Узгодження пари характеристик, з використанням відношення відстаней, де f_1 та f_2 – кращий та другий краща відповідність

Вибраний спосіб оцінки відповідності працює добре, оскільки для правильної відповідності потрібна найближча відповідність, ніж найближча неправильна відповідність для того, щоб було можливе досягнення надійності. Слід за-

значити, що відповідність ближчого сусіда в множині характеристик дескрипторів зображень за алгоритмами SSD або евклідової норми може аналізуватись як відповідності основних характеристик в основу якого полягає вектор. У той же час при практичному використанні алгоритм найближчого сусіда, а на сам перед його параметри залежать від якості даних. Такі алгоритми погано показують себе на бінарних функціях, якими є ORB, BRIEF, FREAK, BRISK та LATCH. Вони порівнюються із залученням відстані Хемінга, яка обчислюється за допомогою розрахунків побітової операції XOR з її подальшим розрахунком кожного біту на результат обчислення. Таке формулювання дійсне лише для операцій з маніпулюванням біту, що виконуються завдяки швидким розрахункам. Вважається, що ефективність алгоритмів, які використовують за основу точки інтересу, сподіваються на властивості основних детекторів ключових точок зображення та вибору взаємопов'язаних з ним дескрипторів цифрових зображень. Можна сказати, що доцільним є вибір цих детекторів та дескрипторів для вмісту цифрових зображень у програмах даного типу. Випадкові зразки (RANSAC) [15], почали використовуватися з метою фільтрації у множинах функцій, що підлягають оцінці геометричної трансформації, що виявляються корисними для узгодження функцій і подальшої реєстрації цифрових зображень і різних застосунків для задачі узгодження об'єктів на сцені.

Для того щоб використовувати оцінку якості дескрипторів ключових точок зображення визначені наступні основні характеристики з роботи [16]:

- «коректні збіги» CM – рівність пари дескрипторів точок інтересу, вказують на одну й ту саму точку на реальній сцені;

- «некоректні збіги» FM – рівність пари дескрипторів точок інтересу, вказує на точки на реальній сцені, які є відмінними одна від одної;

- «загальна кількість збігів» TNM – сума усіх випадків, коли дескриптори рівні між собою, еквівалентна сумі позитивних та негативних збігів;

- «відповідність» C – множина усіх дійсних відповідностей точок інтересу на парі цифрових зображень, дескриптори яких підійшли до порівняння.

Далі буде введено деякі характеристики для подальшої оцінки:

– точність відгуку

$$R = \frac{CM}{C}, P = \frac{CM}{TNM}, 1 - P = \frac{FM}{TNM}.$$

Ця метрика підраховується, як сума коректних збігів, до суми коректних збігів з додаванням суми некоректних збігів, повнота відгуку (recall),

$$R = \frac{CM}{C}.$$

Метрика, яка має на меті максимізування, в статистиці вона має назву «recall», по-іншому можливість даної моделі до знаходження усіх можливих випадків узгодження в множині даних. Формальне визначення recall – це сума коректних узгоджень до суми коректних узгоджень плюс сума помилкових узгоджень. Коректні узгодження – це така одиниця даних, що класифікується, як коректна і справедлива за поглядом моделі, яка визначена коректною. В той же час інші помилкові узгодження є одиницями даних, що визначена модель визначає як помилкові, які насправді є неправильними. У нашій праці коректні узгодження – це коректно отримані узгодження особливих точок зображення на методах BRISK та ORB. Некоректні узгодження – наведені як узгодження пари дескрипторів ключових точок цифрового зображення, які показують на різні елементи реальної сцени.

2.3.2 Критерій накладання детекторів

Загалом критерій накладання детекторів базується на ідеї, аналізу перекривання дескрипторів ключових точок зображення.

Ціль дослідження знайти особливі точки цифрового зображення, які бу-

дуть мати стійкість до зашумлення зображення, різних точок освітлення сцени та різноманітних геометричних спотворень. До них входять зміни масштабу, зміни повороту та інші. Необхідно обробити інформацію з усіх точок визначених точок цифрового зображення, дізнатися про межові особливості. До основних входять: фактури, грані, кути та краплі. Також треба забезпечити розподіл особливих точок по всій сцені. Тоді, існує якась множина детекторів або дескрипторів, що мають в основі різні ідеї з різною логікою. Для чесного аналізу детекторів особливих точок зображення треба обрати критерії їх оцінювання.

Так введемо коефіцієнт повторюваності детекторів для двох цифрових зображень $u_a(x)$, $u_b(x)$. Нехай, воно визначається як відношення між сумою одночасних знаходжень в парі різних цифрових зображеннях. Узагальнюючи, ми аналізуємо повторні знаходження і загальна сума у патчі навколо двох цифрових зображень. У загальному випадку для знаходження повторень робиться підрахунок еліптичної області, яка позначена $R(x, \Sigma)$, що параметризована її центром x та матрицею Σ .

$$R(x, \Sigma) = \{x' \in \Omega \mid (x' - x)^T \Sigma^{-1} (x' - x) \leq 1\}.$$

Два детектори (еліптичних регіонів $R(x_a, \Sigma_a)$ і $R(x_b, \Sigma_b)$) з зображень $u_a(x)$, $u_b(x)$ будуть вважатися ідентичними, коли:

$$1 - \frac{|R(x_a, \Sigma_a) \cap R(x_b, \Sigma_b)|}{|R(x_a, \Sigma_a) \cup R(x_b, \Sigma_b)|} \leq \varepsilon_{overlap}.$$

Досліджено, що об'єднання та перетин таких областей на вихідне зображення $u_a(x)$, спроектувавши його детектор u_b на цифрове зображення u_a . Таке об'єднання аналізує всю область, що позначається як символ $|R(x_a, \Sigma_a) \cup R(x_b, \Sigma_b)|$ поки $|R(x_a, \Sigma_a) \cap R(x_b, \Sigma_b)|$ це ми розуміємо як область їх перетину. Складова накладання – це найбільш допустима похибка в на-

кладанні. У значній частині випадків знайдених орієнтирів їх значення дорівнює значенню встановленому до 0.40. Знаходження $R(x_b, \Sigma_b)$ на сцені u_b проєктується на опорній сцені u_a . Коли помилка накладання є меншою за $\epsilon_{overlap}$, детектори ключової точки вважатимуться повтореними. Показано два патча радіусом r з різними центрами від d . Максимальна відстань для повторного виявлення, складається з функції радіуса r для 4 його значень.

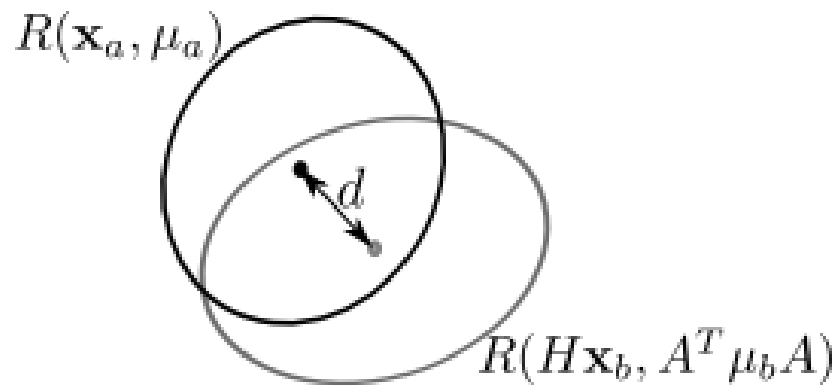


Рисунок 2.12 – Ілюструє критерій повторюваності.

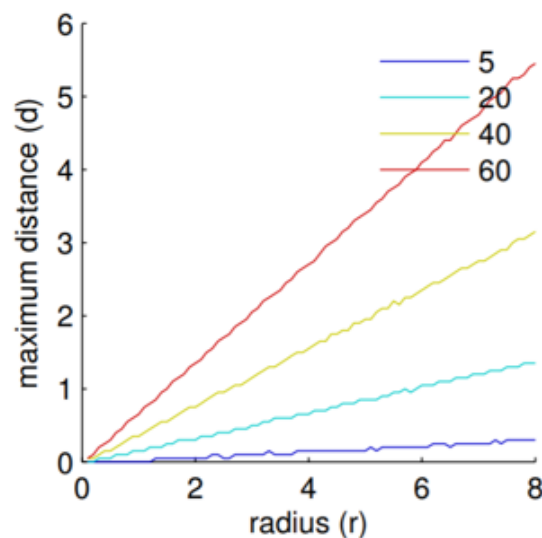


Рисунок 2.13 – Графік залежності радіуса від максимальної дистанції виявлення

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Мова програмування Python та бібліотека OpenCV

Python – об’єктно-орієнтована мова програмування, яка винайдена з метою значно підвищити продуктивність в написанні коду. На ряду з іншими мовами програмування, Python має хороші та погані сторони. Однак позитивні зміни у мові переважають недоліки. Про це говорить кількість розробників, що зростає з кожним роком. Тому й кількість проектів розроблених на цій мові програмування стрімко іде в гору.

Розглянемо переваги мови, які можна виділити в першу чергу:

- лаконічний та простий синтаксис;
- добра читабельність коду та логічний синтаксис;
- всі основні структури даних підтримуються обраною мовою;
- можливість використання на багатьох платформах;
- зручна оптимізація;
- необхідність у спеціалістах;
- необмежена кількість користувацьких бібліотек з обробки даних
- одні з кращих графічних бібліотек для відображення даних.

Python є процедурно-орієнтованою мовою програмування, до того ж поєднує в собі парадигму об’єктно-орієнтованого програмування. Процедурні мови програмування орієнтовані на побудову програм на тлі процедур або функцій, які по своїй суті є багаторазовими програми. На відміну від об’єктно-орієнтованих мов програмування, які мають на меті використання об’єктів, які з’єднують функціональні можливості з даними. Python має одну з найпотужніших, але значно спрощених шляхів відтворити парадигму ООР. Це значно відрізняється в порівняльному аналізі з старими, популярними мовами програмування, Java або C ++.

В сучасному програмуванні при виконанні задач часто виникає потреба у використанні фрагменту коду з іншої мови програмування. Така можливість є у

мови Python – реалізувати цю частину коду програми на мовах C або C ++ інтегровано в програму Python.

Так і навпаки присутня можливість вбудовувати Python у своєму дослідженні на мові C / C ++, щоб надати можливість сценарію для користувачів програми.

До того одною з переваг мови Python є підтримка управління пакетами, вона називається `pip`. І має на меті інсталяцію та управління пакетами, що були написані на мові програмування Python.

`Pip` дає можливість управління усіма доступними пакетами та всім різноманіттям версіями. Це зазвичай робиться за допомогою файлу `requirements.txt`. Що дозволяє ефективно відтворювати весь необхідний список пакетів в окремому оточенні (наприклад, на іншому комп'ютері) або в віртуальному оточенні. Це досягається за допомогою правильно відформатованого файлу `requirements.txt` і команди.

`OpenCV` – найрозвиненіша бібліотека алгоритмів комп'ютерного бачення, задач обробки цифрових зображень та безлічі алгоритмів загального призначення, що мають відкритий код. Реалізована ця бібліотека на мовах C / C ++, але вона розповсюджується на мови Python, Java, Ruby, Matlab, Lua та декілька інших мов програмування. Її дозволено вільно використовуватися як в академічних так і в комерційних цілях – вона поширена на умовах ліцензії BSD.

`OpenCV` має більше ніж 300 гарно оптимізованих алгоритмів, в яких трапляється повний набір, як класичних математичних так і деяких практичних алгоритмів комп'ютерного бачення і машинного навчання. Алгоритми на базі `OpenCV` використовуються у таких сферах:

- обробка та аналіз цифрових зображень;
- додатки з розпізнавання обличчя людини;
- знаходження об'єктів на сцені;
- розпізнавання рухів на відео;
- трекінг об'єктів розміщених на сцені;
- VR 3D моделі;

- відтворення 3D хмар точок зі стерео камер;
- розпізнавання руху на відео.

– Бібліотека широко використовується для затвердження загального базового інтерфейсу комп'ютерного бачення у додатках в цій області. Для того, щоб кількість додатків значно зросла та відбулось розширення нових моделей на РС.

Через наведені вище причини OpenCV була обрана для використання у розробці нашого експерименту. Вона задовольняє всім вимогам та виконує всі необхідні задачі. Це значно спрощує їх розробку.

3.2 Алгоритм розв'язання задачі узгодження зображень на основі ключових точок та перцептивного хешу

Першочергова фільтрація зображень відбувається за допомогою перцептивного хешу. Алгоритм показаний на рис 3.1.

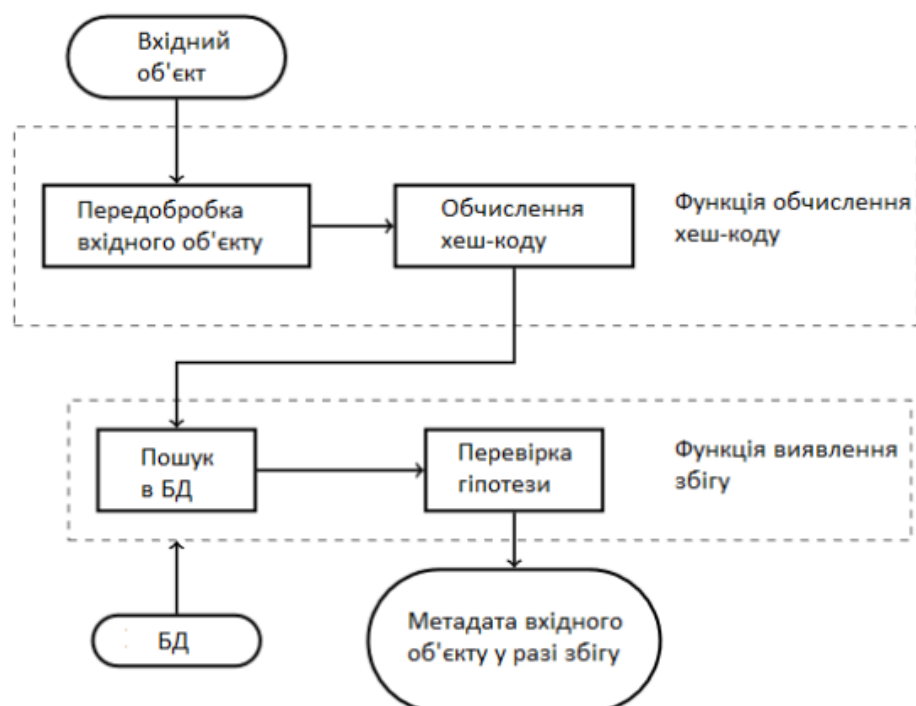


Рисунок 3.1 – Загальна схема ідентифікації контенту за допомогою p-hash

Після того визначаються ключові точки зображення та їх дескриптори. Між дескрипторами відбувається порівняння відстаней, після якого і визначається рівень подібності зображень. Загальна схема наведена на рис. 3.2.



Рисунок 3.2 – Загальна схема додатку

3.3 Опис програми

Структурно програма складається з бібліотек комп'ютерного бачення, математичних та для робот з різними типами даних. До них входять такі:

- а) `opencv2` – бібліотека комп'ютерного бачення;
- б) `pillow` – бібліотека для роботи з цифровими зображеннями;
- в) `numpy` – математична бібліотека, що дає змогу користуватися масивами та матрицями, разом із різними математичними функціями;
- г) `Matplotlib` – бібліотека, що дозволяє графічно відображати дані.

Програма розроблена у середовищах розробки `Pycharm` та `JupyterLab`.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

Перше, що необхідно зробити для початку роботи з зображеннями – це їх підготовка. Для цього нам треба з кольорової моделі RGB перейти на відтінки сірого або «grayscale». Це зроблено тому, що наведені вище алгоритми працюють саме в такій кольоровій схемі. Перші перетворення зображені на рис. 4.1.

Також, для формування вибірки на якій будуть проходити тести нам необхідно скласти набори з зображень різної якості.

```
# Load the image
test_image = cv.imread('turtle1.jpg')

training_image = cv.imread('turtle1.jpg')
#training_image = training_image.astype('uint8')

# Convert the image to RGB
training_image = cv.cvtColor(training_image, cv.COLOR_BGR2RGB)
test_image = cv.cvtColor(test_image, cv.COLOR_BGR2RGB)
# Convert the image to grayscale
training_gray = cv.cvtColor(training_image, cv.COLOR_RGB2GRAY)
test_gray = cv.cvtColor(test_image, cv.COLOR_RGB2GRAY)
```

Рисунок 4.1 – Предобробка даних

Після того, як дані підготовлені, ми можемо розпочинати застосування перцептивного хешу. Приклад захешованого зображення рис. 4.2.



Рисунок 4.2 – зображення захешоване за допомогою p-hash

Після цього кроку до відібраних зображень ми можемо використати алгоритми для знаходження ключових точок зображення. Фрагменти реалізації таких алгоритмів показано на рис. 4.3

```

orb = cv.ORB_create()

train_keypoints_orb, train_descriptor_orb = orb.detectAndCompute(training_gray, None)
test_keypoints_orb, test_descriptor_orb = orb.detectAndCompute(test_gray, None)

keypoints_without_size = np.copy(training_image)
keypoints_with_size = np.copy(training_image)

cv.drawKeypoints(training_image, train_keypoints_orb, keypoints_without_size, color = (255, 36 , 0))

cv.drawKeypoints(training_image, train_keypoints_orb, keypoints_with_size, flags = cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

# Display image with and without keypoints size
fx, plots = plt.subplots(1, 2, figsize=(20,10))

plots[0].set_title("Train keypoints With Size")
plots[0].imshow(keypoints_with_size, cmap='gray')

plots[1].set_title("Train keypoints Without Size")
plots[1].imshow(keypoints_without_size, cmap='gray')

# Print the number of keypoints detected in the training image
print("Number of Keypoints Detected In The Training Image: ", len(train_keypoints_orb))

# Print the number of keypoints detected in the query image
print("Number of Keypoints Detected In The Query Image: ", len(test_keypoints_orb))

```

Рисунок 4.3 – Фрагмент коду застосування алгоритму ORB

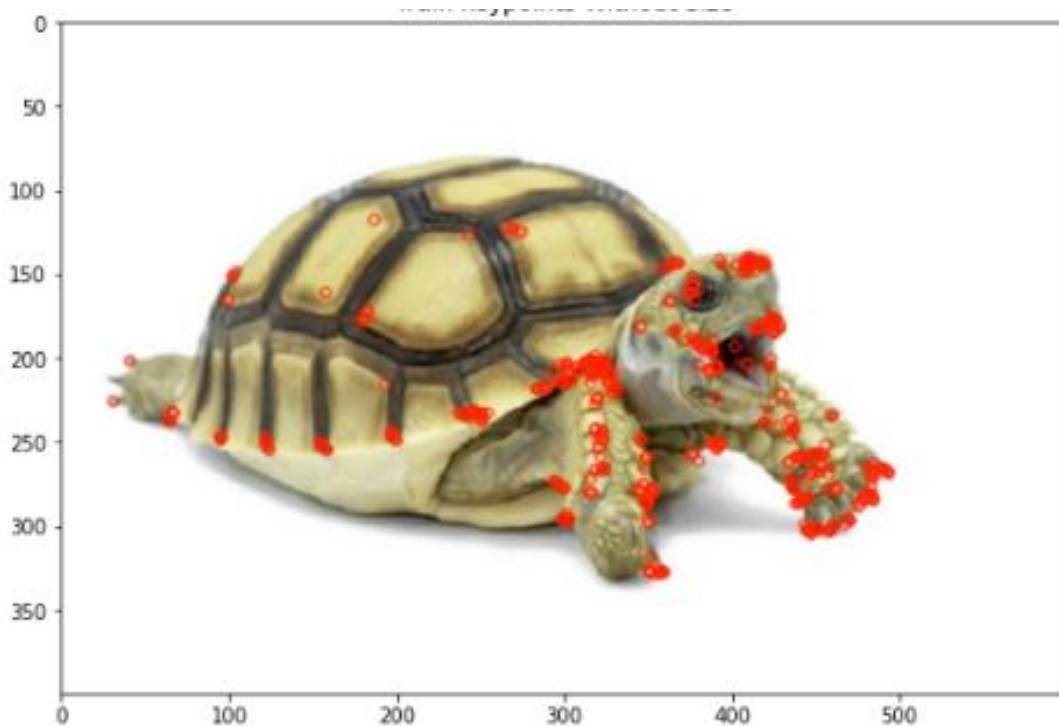


Рисунок 4.4 – Відображення ключових точок ORB

Щоб визначити області зображення в які є спільні попадання алгоритмів, розіб'ємо зображення на патчі різного розміру. Обчислимо матриці попадання алгоритмів знаходження ключових точок.

Далі зробимо підрахунки інших критеріїв порівняння дескрипторів, які розглядалися в другому розділі.

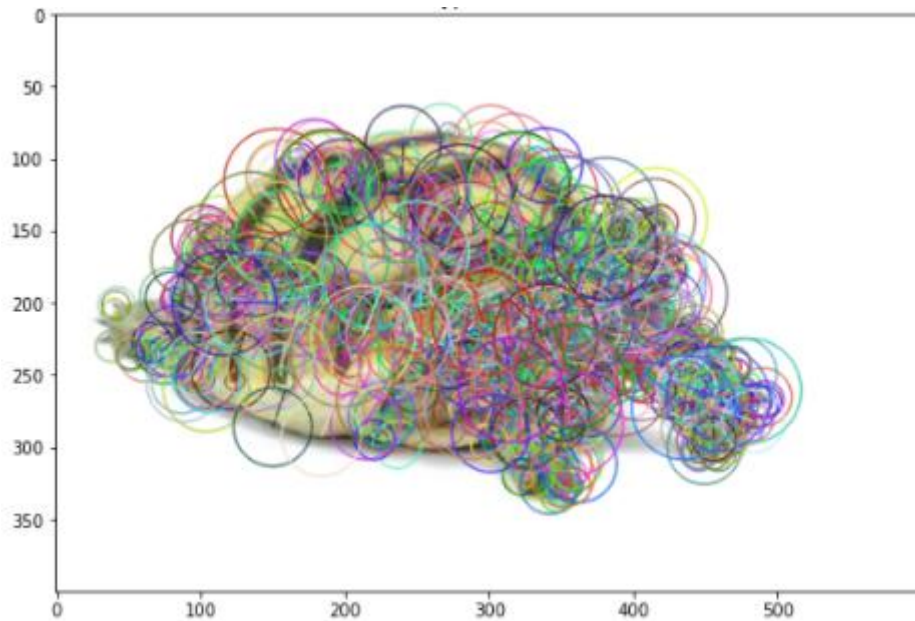


Рисунок 4.5 – Відображення ключових точок з розміром патчу

Порівняємо ключові точки нашого зображення за допомогою відстані Хемінга та експериментально визначеним порогом для максимального значення (рис. 4.6).

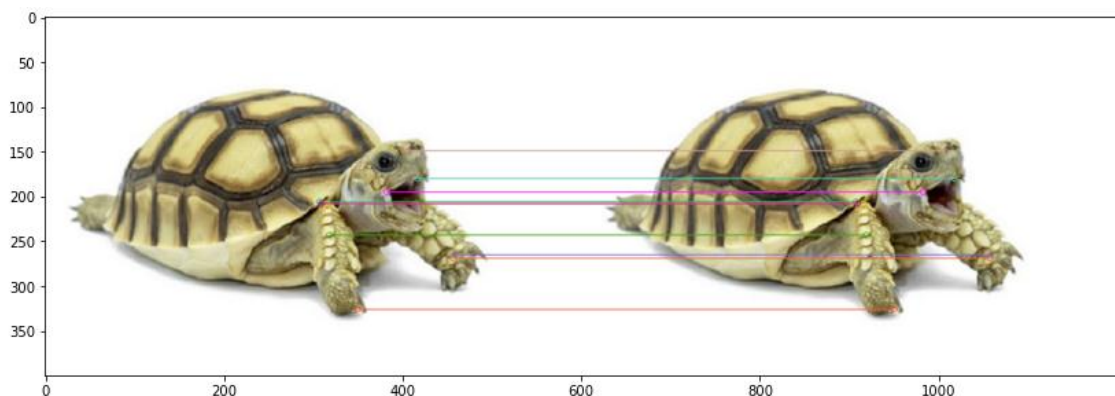


Рисунок 4.6 – Зіставлення ключових точок зображення за допомогою відстані Хемінга

Після проведення всіх тестів та перетворень, виконавши їх аналіз, маємо на меті скласти висновки про модель, яку ми реалізували в цій роботі.

Кращою з трьох методів продуктивністю володіє метод ORB. Хоча кількість знайдених ним ключових точок зображення значно менша за кількість у

конкурентів.

З огляду на геометричні перетворення ORB теж показав кращі результати, але з змінами масштабу краще впорався алгоритм BRISK.

Застосування перцептивного хешу було дуже доречним в нашому дослідженні, бо дозволило відкинути різні зображення за менші витрати часу аніж алгоритми знаходження ключових точок зображення.

ВИСНОВКИ

У рамках нашої роботи було проведено системний аналіз проблеми узгодження зображень за допомогою ключових точок зображення та перцептивного хешу. Завдяки цьому було отримано дані про доцільність використання моделі узгодження зображень за допомогою КТ та прцептивного хешу. На практиці було продемонстровано, що перцептивний хеш доцільно використовувати задля фільтрації зображень, що мають велику відмінність одне від одного. Але його точності не вистачає для повноцінного порівняння пари зображень. Натомість, застосування алгоритмів SIFT, ORB та BRISK загально впоралися зі своєю задачею.

Були описані усі основні теоретичні моменти, що потрібні для створення моделі узгодження.

В кваліфікаційній роботі було розроблено модель узгодження ключових точок цифрового зображення на основах методів SIFT, ORB, BRISK, Продецьована їх робота за допомогою мови програмування Python та бібліотеки комп'ютерного бачення OpenCV. Було розраховано прцептивний хеш зображень за допомогою бібліотеки numpy.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Alshazly H. A., Hassaballah M., Ali A. A. An experimental evaluation of binary feature descriptors // International Conference on Advanced Intelligent Systems and Informatics. 2015. P. 181-191.
2. Wang R., Xia Y., Wang. License plate localization in complex scenes based on oriented FAST and rotated BRIEF feature // Journal of Electronic Imaging. 2015. 24 (5). P. 8.
3. Rosten E., Porter R., Drummond T. Faster and better: A machine learning approach to corner detection // IEEE transactions on pattern analysis and machine intelligence. 2008. 32 (1). P. 105-119.
4. Calonder M., Lepetit V., Ozuysal M. BRIEF: Computing a local binary descriptor very fast // IEEE transactions on pattern analysis and machine intelligence, 2011. 34 (7). P. 1281-1298.
5. Rublee E., Rabaud V., Konolige K. ORB: An efficient alternative to SIFT or SURF // 2011 International conference on computer vision. 2011. P. 2564-2571.
6. Tuytelaars, T., Mikolajczyk, K. Local invariant feature detectors: a survey. Foundations and trends in computer graphics and vision. 2008. 3 (3). P. 177-280.
7. Lindeberg, T. Feature detection with automatic scale selection // International journal of computer vision. 1998. 30 (2). P. 79-116.
8. Шапиро Л., Стокман Д. Компьютерное зрение. 2-е изд. Москва : БИНОМ. Лаборатория знаний, 2013. 752 с.
9. Meng Y., Tiddeman P. Implementing the scale invariant feature transform (sift) method. Department of Computer Science University of St. Andrews. В. 2006. 9 p.
10. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Техносфера. Москва : Техносфера, 2019. 1104 с.
11. Bay H., Ess A., Tuytelaars T. Features, S. U. R. M.: ETH Zurich, BIWI. 2008. 14 p.

12. Методы компьютерной обработки изображений / Гашников, М. В., Глумов, Н. И., Ильясова [и др.]. Москва : ФИЗМАТЛИТ, 2007. 544 с.
13. Skowron, A., Rauszer, C. The discernibility matrices and functions in information systems // *Intelligent decision support*. 1992. P. 331-362.
14. Bonikowski Z., Bryniarski E., Wybraniec-Skardowska U. Extensions and intentions in the rough set theory // *Information sciences*. 1998. 107 (1-4). P. 149-167.
15. Веричев А. В. Обучаемый детектор особых точек изображения // Сборник трудов III международной конференции и молодежной школы «Информационные технологии и нанотехнологии». 2012. С. 670-675.
16. Mikolajczyk K., Schmid C. A performance evaluation of local descriptors // *IEEE transactions on pattern analysis and machine intelligence*. 2012. 27 (10). P. 1615-1630.
17. Myasnikov V. V. Model-based gradient field descriptor as a convenient tool for image recognition and analysis // *Computer optics*. 2012. P. 596-604.
18. Гороховатский В. А. Структурный анализ и интеллектуальная обработка данных в компьютерном зрении. Харьков : Компания СМИТ, 2014. 316 с.
19. Duda R. O., Hart P. E., Stork D. G. *Pattern classification*. 2000. 668 p.
20. Гороховатский В. А., Гороховатський В. О. Методы определения релевантности изображений на основе медианной обработки структурных описаний. Харків : 2017. С. 100-106.
21. Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення / V. O. Gorokhovatsky, D. V. Pupchenko, K. G. Solodchenk // Системи управління, навігації та зв'язку. Збірник наукових праць. Полтава : ПНТУ, 2018. С. 93-98.