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

До захисту допускається
в.о. завідувача кафедри
_____ Рязанцев О.І.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Засоби нормалізації зображень для систем комп'ютерного зору

Освітній рівень"Магістр"
Спеціальність 123 "Комп'ютерна інженерія"

Науковий керівник роботи:	_____ (підпис)	М.Є.Щербакова (ініціали, прізвище)
Консультант з охорони праці:	_____ (підпис)	Я.О.Критська (ініціали, прізвище)
Студент:	_____ (підпис)	О.В.Лєвшин (ініціали, прізвище)
Група:		КІ-19зм

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній рівень магістр
Напрям підготовки _____
(шифр і назва)
Спеціальність 123 "Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри _____
_____ В.С.Кардашук
«_____» _____20____р.

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

Лєвшину Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Засоби нормалізації зображень для систем комп'ютерного зору

керівник проекту (роботи) Щербакова Марина Євгенівна, к.т.н., доц.
(прізвище, м.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «5» 10 2020 р. № 141/15.15
2. Строк подання студентом роботи 10.01.2021
3. Вихідні дані до роботи Матеріали науково-дослідної практики, методи нормалізації зображень з наявністю геометричних перетворень; математичні моделі дескрипторів SURF, SIFT, BRISK, ORB, KAZE, AKAZE; пошук відповідностей на зображенні та відкидання хибних зв'язків; найближчий сусід; математична модель алгоритму RANSAC; об'єктно орієнтована мова програмування Java; відкрита бібліотека комп'ютерного зору OpenCV; колекції тестових зображень.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Сучасний стан питання аналізу та нормалізації зображень, математичні моделі нормалізації зображень на основі дескрипторів їх характерних точок, реалізація та дослідження розробленого методу нормалізації на основі аналізу характерних точок дескрипторів surf, sifr, orb, kaze, akaze, brisk, охорона праці та безпека в надзвичайних ситуаціях, висновки

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

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання 14.10.2020

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	02.09.2020-15.09.2020	
2	Критичний аналіз літератури з досліджуваної проблеми	16.09.2020-22.09.2020	
3	Розробка методу	23.09.2020-25.09.2020	
4	Дослідження засобів	26.09.2020-06.10.2020	
5	Аналіз результатів дослідження	07.10.2020-25.11.2020	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2020-1.12.2020	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2020-09.01.2021	

Студент _____ Лєвшин О.В.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Щербакова М.Є.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Лєвшин О.В. Засоби нормалізації зображень для систем комп'ютерного зору.

Метою роботи є розробка методу нормалізації геометричних перетворень, що базуються на аналізі характерних точок, які дозволяють нормалізувати вхідне зображення до еталонного для подальшого його аналізу.

Об’єктом дослідження є послідовність геометрично-синтезованих та реальних зображень.

Використано дескриптори SURF, SIFT, ORB, KAZE, AKAZE, BRISK для детектування та опису характерних точок зображення, зіставлення відповідностей та методи k-NNN та RANSAC для детектування викидів та отримання найближчої матриці перетворення. Проведено дослідження на зображеннях за наявності масштабу, повороту та центроафінного перетворення. Отримані оцінки точності параметрів перетворення, зіставлення пікселів та швидкість аналізованих дескрипторів.

У результаті роботи обрано кращі детектори для задачі нормалізації зображень.

Ключові слова: дескриптор, surf, sift, orb, kaze, akaze, brisk, ransac, k-nn, виявлення характерних точок, відповідність особливостей, масштаб, поворот, центроафінне перетворення, нормалізація.

ABSTRACT

Levshin O.V. Image normalization tools for computer vision systems.

The goal is to develop a method of normalization of geometric transformations, based on the analysis of feature points, which allow to normalize the input image to the reference for its further analysis.

The object of the study is a sequence of geometrically synthesized and real images.

SURF, SIFT, ORB, KAZE, AKAZE, BRISK descriptors were used to detect and describe feature image points, match mappings, and k-NNN, RANSAC methods to detect emissions and obtain the closest conversion matrix. The study on the images in the presence of scale, rotation and central affine transformation. The obtained estimates of the accuracy of the transformation parameters, the comparison of points and the speed of the considered descriptors.

As a result of work, the best detector for the task of normalizing images was selected.

Keywords: descriptor, surf, sift, orb, kaze, akaze, brisk, ransac, k-nn, feature detection, feature matching, scale, rotation, central affine transformation, normalization.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ 6
ВСТУП 7
1 СУЧАСНИЙ СТАН ПИТАННЯ АНАЛІЗУ ТА НОРМАЛІЗАЦІЇ ЗОБРАЖЕНЬ 8
1.1 Задачі аналізу зображень та успіхи в їх вирішені 8
1.2 Питання нормалізації геометричних перетворень в задачах аналізу та обробки зображень 9
1.2.1 Проблема нормалізації геометричних перетворень 9
1.2.2 Огляд існуючих методів нормалізації геометричних перетворень 10
1.2.3 Отримання параметрів геометричних перетворень на основі аналізу характерних точок 12
1.3 Опис існуючих бібліотек для аналізу зображень 15
1.4 Постановка задачі дослідження 15
2 МАТЕМАТИЧНІ МОДЕЛІ НОРМАЛІЗАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ДЕСКРИПТОРІВ ЇХ ХАРАКТЕРНИХ ТОЧОК 17
2.1 Дескриптор SIFT 17
2.2 Дескриптор SURF 19
2.3 Дескриптор ORB 22
2.4 Дескриптор KAZE 23
2.5 Дескриптор AKAZE 24
2.6 Дескриптор BRISK 25
2.7 Отримання параметрів нормалізації геометричних перетворень 27
2.7.1 Пошук відповідностей методом найближчого сусіда 27
2.7.2 Отримання параметрів нормалізації на основі алгоритму RANSAC 28
3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО МЕТОДУ НОРМАЛІЗАЦІЇ НА ОСНОВІ АНАЛІЗУ ХАРАКТЕРНИХ ТОЧОК ДЕСКРИПТОРІВ SURF, SIFR, ORB, KAZE, AKAZE, BRISK 32
3.1 Обґрунтування вибору середовища програмної реалізації 32
3.2 Опис тестових зображень, умов та задач експериментів щодо дослідження інваріантності 33
3.3 Дослідження питання щодо інваріантності до геометричніх перетворень дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK 36
3.3.1 Дослідження нормалізації за наявності перетворення масштабу 37
3.3.2 Дослідження нормалізації за наявності перетворення повороту 47

3.3.3 Дослідження нормалізації за наявності центроафінних перетворень 58
3.3.4 Дослідження нормалізації реальних зображень 61
3.4 Вивчення питання швидкодії нормалізації на основі дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK 66
3.4.1 Опис умов та задач експериментів щодо дослідження швидкодії..... 66
3.4.2 Результати експериментів та висновки щодо швидкодії пошуку параметрів нормалізації..... 67
3.5 Висновки щодо використання дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK для нормалізації зображень 69
4 ОХОРОНА ПРАЦІ 70
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об’єкту, що мають вплив на персонал 70
4.2 Заходи щодо техніки безпеки 71
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці..... 74
4.4 Рекомендації по пожежній безпеці 77
ВИСНОВКИ..... 81
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 83
ДОДАТОК А. Тестові зображення 87
ДОДАТОК Б. Фрагмент лістингу 89
ДОДАТОК В. Результати експериментів за наявності перетворення масштабу 94
ДОДАТОК Г. Результати експериментів за наявності перетворення повороту 98
ДОДАТОК Д. Результати експериментів за наявності центроафінного перетворення 101
ДОДАТОК Е. Результати експериментів для реальних зображень 108
ДОДАТОК Ж. Електронні плакати..... 113

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

SURF – Speeded-Up Robust Features

SIFT – Scale Invariant Feature Transform

ORB – Oriented FAST and Rotated BRIEF

FAST – Features from Accelerated Segment Test

BRIEF – Binary Robust Independent Elementary Features

AKAZE – Accelerated-KAZE

BRISK – Binary Robust Invariant Scalable Keypoints

RANSAC - RANdom SAmple Consensus

NN – Nearest Neighbor

k-NN – k-Nearest Neighbor

ВСТУП

Сьогодні, з розвитком інформаційних технологій існує дуже багато методів для аналізу та обробки зображень. Кожен з методів виконує свої завдання та витягує необхідну корисну інформацію з зображень. При цьому дуже часто буває, що зображення спотворено геометричними перетвореннями та викривлює результати експерименту. Саме тоді використовується нормалізація зображення для більш точних досліджень.

Усі існуючі на даний момент методи нормалізації графічної інформації можна поділити на два напрямки: на основі функціоналів та аналізу характерних точок. Останній - пошук ключових точок і побудова локальних дескрипторів є однією з базових задач аналізу зображень. Даний процес можна розбити на чотири етапи:

- виділення ключових точок за допомогою детектора;
- опис знайдених точок за допомогою дескриптора;
- отримання відповідностей між ключовими точками;
- аналіз отриманих відповідностей з метою отримання матриці гомографії.

В роботі досліджується геометричні параметри нормалізації зображень на основі наступних дескрипторів: SURF, SIFT, ORB, KAZE, AKAZE та BRISK.

Таким чином, пошук ефективного виділення ключових точок, їх дескрипторів, а також методів нормалізації зображень, є актуальним завданням в області обробки зображень.

Незважаючи на численні дослідження та успіхи у вирішенні проблеми нормалізації геометричних перетворень зображень на основі аналізу характерних точок тема вимагає подальшого дослідження.

1 СУЧАСНИЙ СТАН ПИТАННЯ АНАЛІЗУ ТА НОРМАЛІЗАЦІЇ ЗОБРАЖЕНЬ

1.1 Задачі аналізу зображень та успіхи в їх вирішені

Аналіз зображень - це вилучення значимих даних із зображень за допомогою методів цифрової обробки зображень. Завдання аналізу зображень можуть бути такими ж простими, як читання штрих-кодів, або такими ж складними, як ідентифікація людини по його обличчю.

Аналіз зображень і фотографій полягає не тільки у визначенні якості отриманих знімків, але і в оцінці їх інформативності. Обробка і аналіз зображень дуже часто використовуються спільно [2, 3]. Наприклад, якщо потрібно виділити об’єкти на зображенні, перед тим як робити кількісний аналіз та проводити дослідження, потрібно прибрати шуми та визначити границі необхідних об’єктів. Редагування зображень потрібно і для того, щоб проводити візуальний аналіз: наприклад, попередня обробка рентгенівського зображення значно спрощує його дослідження.

Типові завдання аналізу та обробки зображень: колірна корекція; порівняння зображень; згладжування; сегментація; компенсація втрати різкості; геометричні перетворення, такі як обертання і масштабування; нормалізація перетворень. Загальна обробка зображень включає в себе поліпшення, відновлення, кодування та стиснення зображення.

Ще в середині XX століття обробка зображень була здебільшого аналоговою і виконувалась оптичними пристроями. Першою вдалою заявкою стала американська лабораторія American Jet Propulsion Laboratory (JPL). Вони використовували методи обробки зображень, такі як геометрична корекція, перетворення градацій, видалення шуму тощо на тисячах місячних фотографій, що були надіслані назад космічним детектором у 1964 році. Пізніше при обробці 100 000 фотографій, отриманих з космічного корабля, була створена топографічна карта, кольорова карта та панорама місяця. У1970-х роках з’явилась можливість робити обробку зображення у реальному часі. У 2000-х роках, цифрова обробка зображення стала найпоширенішою формою обробки зображення.

Дуже суттєві успіхи в вирішення задач аналізу зображень показали методи, які працюють на основі ознак (опису) зображень. Які саме ознаки використовувати залежить від типу задачі та самого зображення. Наприклад, для отримання опису характерних точок зображення розроблено велику кількість алгоритмів, що аналізують околицю навколо точок та приймають рішення, чи є точка цікавою для подальшого дослідження, якщо так, то формують для неї вектор ознак (дескриптор), який вже використовується у подальшому

аналізі зображення [7-13, 17]. В інших задачах, наприклад, під час роботи з текстурними зображенням вектор текстурних ознак формується для кожної точки. Таким чином кожна точка має текстурний опис, що в подальшому можна використати для рішення задач сегментації, пошуку, розпізнавання текстурних фрагментів [38-40].

В наш час стали застосовувати такі системи як машинне навчання та нейронні мережі [5]. Нейронна мережа - один із способів реалізації штучного інтелекту. У розробці штучного інтелекту існує велика область - машинне навчання. Вона вивчає методи побудови алгоритмів, здатних самостійно навчатися. Це необхідно, якщо не існує чіткого рішення будь-якої задачі. У цьому випадку простіше не шукати правильне рішення, а створити механізм, який сам придумає метод для його пошуку.

Однак, універсальних методів обробки зображень, які можна порівняти за ефективністю з інтелектуальними можливостями людини, ще не знайдено, що стимулює активну діяльність вчених в цьому напрямку. Головним дослідженням будь-якої задачі розпізнавання об’єктів є зіставлення зображень та аналізування на відповідність заданому еталону.

Проблеми виникають ще на етапі витягання корисної інформації зображення, деякі з них: еталон та вхідне зображення може відрізнятися положенням у полі зору, зображення можуть мати геометричні перетворення, спотворюючи зображення. Тим самим на цьому етапі застосовується нормалізація. При цьому нормалізація може використовуватися як кінцева мета, як рішення проміжної задачі (підготовчої) у вирішенні завдань аналізу аналітичними методами, а також при використанні нейронних мереж.

1.2 Питання нормалізації геометричних перетворень в задачах аналізу та обробки зображень

1.2.1 Проблема нормалізації геометричних перетворень

Процес обробки зображень можна розділити на наступні етапи:

– сегментація - виділення окремих областей зображення з фону;

– реставрація - фільтрація з метою поліпшення яскравості, видалення засвічувань, тощо;

– нормалізація - приведення заданого зображення до деякого еталонного зображення;

– розпізнавання - зображення класифікується залежно від початково заданої задачі.

Необхідність етапів нормалізації виявлена тим, що між еталонними і вхідними зображеннями є відмінності, що викликані геометричними перетвореннями.

В даний час проблема нормалізації зображень в реальних умовах може розглядатись як задача приведення отриманого зображення рухомого або стаціонарного об'єкта до деякого еталону з метою його розпізнавання

При вирішені реальних задач обробки зображень необхідно щоб виконувались наступні умови:

– операція переходу зображення до еталонного повинно бути простою для реалізації на програмному рівні;

– об’єкти повинні бути зручними для візуального сприйняття для прийняття рішення експертом;

– операція переходу до еталону повинна бути завадостійкою.

Виконання цих умов при реальних зображеннях, що не були спотворені штучно або програмно, є ідеальним випадком. Зображення може бути спотворено шумами, відблисками, деякими засвічуваннями тощо. Тому задача нормалізації зображень є досі актуальною і не вирішеною проблемою.

1.2.2 Огляд існуючих методів нормалізації геометричних перетворень

Суть нормалізації полягає в автоматичному обчисленні невідомих параметрів перетворень, які відрізняють вхідні зображення від еталона, та подальшому приведенні їх до еталонного вигляду. Процедура перетворень проводиться за допомогою операторів нормалізації, а обчислення параметрів виконується функціоналами, діючими на множині зображень [1, 4, 6].

Оператор F(B), який переводить вхідне ображення у нормалізоване, називають нормалізатором. Для знаходження нормалізаторів розроблені спеціальні методи. Методи нормалізації можна розділити на паралельні і послідовні, параметричні і слідкуючі. Нормалізатори знайшли ефективне застосування для базових перетворень: зсувів, поворотів, розтягнень, косих зрушень і деяких їх комбінацій.

Різниця параметричних та слідкуючих нормалізаторів полягає у способі реалізації, так, на відміну від параметричного, слідкуючий тип проводить оцінку тільки в кінці процедури нормалізації, коли вже відбулося поєднання еталонного та вхідного зображень. Нехай:

$$B(x,y) = gB_0(x - l, y - m),$$

де l та m - дійсні параметри. Нормалізатор зсувів має вид:

$$F(B) = F(B)B = B(x + F_1(B), y + F_2(B)),$$

де Ф - відображення $M \to G_c$, визначене двома дійсними функціоналами $\Phi_1(B)$, $\Phi_2(B)$. Фізичний сенс цього функціонала є у компенсації зсувів l та m.

Нормалізатором параметричного або координатного типу називається такий оператор, де спочатку визначаються параметри невідомого нормалізуючого перетворення g. Система реалізації описується як:

– визначення параметрів l, m розрахунком функціоналів $\Phi_1(B)$, $\Phi_2(B)$;

– застосування перетворень нормалізатору зсуву відповідно розрахованим значенням;

– вхідне зображення центрується (здійснюється перетворення g^{-1}).

Нормалізатором слідкуючого типу називають оператор, де проводяться багатократні перетворення зсувів (з деяким кроком за параметром l та m) і при цьому після кожного кроку розраховуються функціонали $\Phi_1(B)$ та $\Phi_2(B)$. Перетворення продовжується доки функціонали $\Phi_1(B)$ та $\Phi_2(B)$ не досягнуть деяких характерних значень та координати l та m знаходяться тільки на останньому кроці як результат сумування поточних прирощень Dl, Dm.

Послідовна нормалізація – багатокрокова процедура, де на кожному кроці відбувається нормалізація тільки однієї з базових груп перетворень. Таким чином, послідовна нормалізація – це розклад складного нормалізатора багато-параметричної групи у комбінацію простих нормалізаторів базових груп з метою спрощення реалізації.

Паралельна нормалізація – метод одночасного знаходження параметрів перетворення, що зв’язує вхідне та еталоне зображення. Вона особливо ефективна при чотирьох і більше параметричних групах, відсутні прості стійкі часткові нормалізатори та з нормалізаторів підгруп неможливо синтезувати нормалізатор повної групи.

Також нормалізацію зображення B (x, y) можна проводити на основі одновимірних нормалізації. Така нормалізація полягає в пошуку параметрів не тільки зображення B (x, y), а обчисленні параметрів індукованих перетворень обмежень на деякі прямі b (ξ). Під індукованими перетвореннями розуміються перетворення обмежень на прямі, які викликають перетворення площині [18–27].

Прямі, на які проводити обмеження, повинні мати властивість інваріантності щодо геометричного перетворення. Тому вибір прямих залежить від групи геометричних перетворень. На основі параметрів індукованих перетворень обчислюються параметри нормалізації перетворення площині. Даний метод має ряд переваг, однак істотно залежить від різних перешкод і якості попередньої сегментації об'єкта.

1.2.3 Отримання параметрів геометричних перетворень на основі аналізу характерних точок

1.2.3.1 Вирішення задач аналізу зображень на основі характерних точок

Знаходження відповідностей на зображеннях - одна з основних проблем в машинному зорі, заснована на виявленні і зіставленні характерних точок зображень. За допомогою відповідностей можливо розпізнання об'єктів і відновлення тривимірної структури, створення панорам з зіставлених зображень, нормалізування для виявлення геометричних перетворень тощо.

Методи пошуку відповідників на зображенні базується на процесі розпізнання. Важливим завданням на даному етапі є описи властивостей об'єкта - дескрипторів, для подальшої ідентифікації об'єкта.

Характерні точки (в різних джерелах – features / characteristic points / local feature points / interest point) – це точки (пікселі) з характерною (особливою) місцевістю – тобто такі, що відрізняються своєю областю від усіх сусідніх точок [33]. Використовуючи особливі точки, можна аналізувати як цілі зображення так і об'єкти на них. Коректні характерні точки дозволяють справитися з зміною масштабу, ракурсу та перекриття сцені або об'єкта.

Для ідентифікації характерних точок застосовуються детектори. Як результати роботи детектору є множина особливих точок, для яких необхідно розрахувати математичний опис.

Для усіх характерних точок, знайдених за допомогою детектору, розраховуються дескриптори – вектора ознак, які описують структуру навколо конкретної характерної точки. При цьому деякі дескриптори вирішують одразу дві задачі – пошук та опис характерної точки.

Основними достоїнствами використання характерних точок для задач виявлення є простота і швидкість виділення (у порівнянні з іншими використовуваними характерними ознаками). Крім того, на зображеннях не завжди вдається виділити інші характерні риси

(хороші і чіткі контури або області), в той час як локальні особливості в переважній більшості випадків виділити можна.

1.2.3.2 Аналіз досліджуваних дескрипторів

Для пошуку особливих точок і формування їх дескрипторів існує досить багато методів [28-37]. В роботі розглядаються наступні: SURF, SIFT, ORB, BRISK, KAZE та AKAZE. У таблиці 1.1 наведено загальну інформацію про їх тип та розмір.

Розрізняють дескриптори границь (Corners) та областей (Blobs). При використанні дескрипторів границь область яка досліджується може бути описана формою її границі, шляхом задання її характеристик. Він включає в себе наступні методи: цепні коди (подання границь у виді послідовності відрізків прямих ліній заданої довжини та напряму), сигнатури (подання границі у виді одномірної функції) та апроксимацію багатокутниками.

Застосування дескрипторів областей обмежена ситуаціями, коли досліджувані об’єкти розрізняються настільки, що для їх пошуку досить кілька обновних дескрипторів. Площа області – це кількість пікселів які розташовані в межах її границі. При цьому дескриптори задаються як числовими або бінарними векторами. Числовий вектор може описувати структуру або текстуру області особливої точки, наприклад у методах SIFT та SURF для розрахунку дескрипторів будуються гістограми градієнтів областей навколо особливої точки. При будуванні бінарних дескрипторів описується область навколо особливої точки двійковим рядком, отриманою як попарне порівняння яскравості пікселів заданої області. Тобто бінарний вектор – це спосіб опису, у якому напрямку зменшується яскравість в області характерної точки.

Таблиця 1.1 – Загальна інформація о типах та розмірах дескрипторів

Назва	Тип	Розмір
SIFT	Blobs	128 Floats
SURF (128D)	Blobs	128 Floats
SURF (64D)	Blobs	64 Floats
ORB	Corners	32 Bytes
KAZE	Blobs	128 Floats
AKAZE	Blobs	61 Bytes
BRISK	Corners	64 Bytes

1.2.3.3 Використання характерних точок для отримання параметрів нормалізації

У комп'ютерному зорі будь-які два зображень в просторі одного і того ж плоского об'єкта пов'язані гомографією. Це означає, що якщо є набір точок на одному зображенні і відповідний набір точок на іншому, можливо знайти відповідність між ними та представити у вигляді матриці H-гомографії. Таким чином, для точки на зображенні у площині B з координатами x та y перетворення у площину A з координатами x_i' та y_i' можна виразити формулою:

$$\lambda_i \times \begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = H \times \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \times \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix},$$

де $\lambda_i \times \begin{pmatrix} X' & Y' & 1 \end{pmatrix}$ – однорідні координати точки на зображені у тримірному просторі, H – матриця гомографії.

Оскільки гомографія - це проективне перетворення на площині, її можна розглядати як комбінацію наступних операцій: паралельний перенос; поворот; масштабування; афінне перетворення.

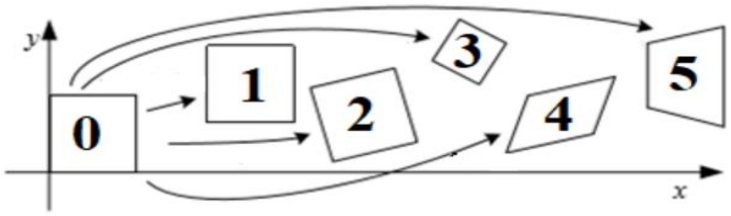


Рисунок 1.1 – Перетворення зображень: 0 - вхідне, 1 - паралельний перенос, 2 - поворот, 3 - масштабування, 4 – афінне, 5 - проективне

Отже, матриця гомографії H також є сукупністю вектору перенесення, матриці повороту тощо, для отримання параметрів геометричних перетворень і подальшої нормалізації, необхідно обчислити H для кожної пари зображень.

1.3 Опис існуючих бібліотек для аналізу зображень

OpenCV – це відкрита бібліотека для обробки зображень, що містить загальні числові алгоритми машинного зору з відкритим для користувачів кодом. Загалом бібліотека сприяє забезпеченню загального механізму для програм технічного зору.

Бібліотека має приблизно майже 3000 алгоритмів, спрямованих на вирішення таких завдань: ідентифікація об’єктів та тексту, усунення спотворень, розкриття подібності і форми сутностей, стеження за переміщенням об’єкта, розпізнавання рухів, жестів та багато іншого яке працює в реальному часі. OpenCV підтримує різноманітні мови, деякі з них: C++, Python та Java. Перша реліз версія бібліотеки вийшла 17 років тому – в червні 2000 року. Спочатку розроблявся корпорацією Intel, а пізніше підтримувався дослідною лабораторією робототехніки «Willow Garage».

В роботі для запуску методів SURF, SIFT, BRISK, ORB, KAZE та AKAZE використовувалась версія OpenCV 3.4.5, при цьому при інсталяції бібліотеки був вказаний флаг OPENCV_ENABLE_NONFRE=true, який дає доступ до платних ресурсів без комерційного використання. Нині, завдяки BSD ліцензії, є можливість застосовувати код OpenCV в комерційних і академічних цілях.

1.4 Постановка задачі дослідження

Незважаючи на досягнуті успіхи у вирішенні проблеми нормалізації зображень з геометричними перетвореннями на основі аналізу характерних точок тема вимагає подальшого дослідження. Пошук ефективного виділення ключових точок, їх дескрипторів, а також методів перевірки на збіги, як і раніше є актуальним завданням в області обробки зображень.

Метою даної роботи є дослідження нормалізації зображень до еталону та дослідження різних дескрипторів, порівняння різних кількісних характеристик тощо. Декомпозиція мети може дати список передбачуваних завдань дослідження:

- аналіз ідентифікації, опису та відповідності дескрипторів;
- розгляд дескрипторів SURF, SIFT, ORB, BRISK, KAZE та AKAZE;
- вивчення методів для відкидання поганих зв’язків;
- розробка методу нормалізації на основі аналізу характерних точок;
- створення тестової колекції зображень;

– дослідження у порівняльному аспекті точності та швидкодії методу нормалізації на основі аналізу дескрипторів SURF, SIFT, ORB, BRISK, KAZE та AKAZE, розробка програмного застосунку для проведення досліджень.

Для досягнення даної мети необхідно розглянути наступні теоретичні питання: виявлення ключових точок, побудова дескрипторів та зіставлення відповідностей, проблема відкидання хибних відповідностей на зображеннях, проаналізувати наступні дескриптори: SIFT, SURF, ORB, KAZE, AKAZE, BRISK.

На основі описаного матеріалу необхідно оцінити та отримати параметри матриці гомографії за допомогою алгоритму RANSAC розрахованої на основі аналізованих дескрипторів, виконати нормалізацію та оцінити її коректність.

Крім цього необхідно опрацювати ряд практичних питань: спроектувати технічне завдання та вивчити технологію роботи у програмному середовищі Java та відкритій бібліотеці OpenCV.

У якості тестової колекції зображень буде використано файли, які мають JPEG формат та розміри від 300 x 400 до 1000 x 1300 пікселів.

2 МАТЕМАТИЧНІ МОДЕЛІ НОРМАЛІЗАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ДЕСКРИПТОРІВ ЇХ ХАРАКТЕРНИХ ТОЧОК

2.1 Дескриптор SIFT

У 2004 році Д. Лоу, Університет Британської Колумбії, створив найвідоміший алгоритм «Scale Invariant Feature Transform» (SIFT) [7, 17].

Основним моментом у детектуванні характерних точок є побудова піраміди гауссіанів (Gaussian) та різниць гауссіанів (Difference of Gaussian, DoG). Гауссіаном є зображення:

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

де L - значення гауссіана в точці з вимірюванням (x, y), σ - радіус розмиття, G - гауссове ядро, I - значення вихідного зображення, * - операція згортки.

Різницею за Гауссом називають зображення, отримане шляхом віднімання одного гауссіана вихідного зображення з гауссіана іншого радіусу розмиття:

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

Після отримання різниці, проводиться порівняння пікселів по різниці гауссіанів зображень у тому ж самому масштабі для 8 сусідів та 9 відповідних йому сусідніх пікселів в кожному з сусідніх масштабів. Якщо значення пікселя є максимумом або мінімумом серед усіх порівнюваних точок, воно вибирається як кандидат ключовий точки. Далі здійснюється процес відкидання точок, які мають низький контраст.

Точне визначення положення характерної точки виконується за допомогою квадратичного розкладання Тейлора функції різниці гауссіанов масштабного простору з кандидатом в ключові точки, розташованому на початку координат. Це розкладання Тейлора задається рівнянням:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + 1/2 x^T \frac{\partial^2 D}{\partial x^2} x,$$

де D і її похідна обчислюються в точці-кандидата, а $x = (x, y, \sigma)^T$ є зміщенням від цієї точки.

Місцезнаходження екстремуму $\hat{x}$ визначається взяттям похідної цієї функції по x прирівняним до нуля. Наступними кроками алгоритму є відкидання ключових точок низького контрасту, вилучення вкладу ребер (для збільшення стабільності, слід виключати ключові точки, які мають погано певний розташування, але мають великий вклад від ребер) та призначення орієнтації.

Саме попередні кроки забезпечують інваріантність місцерозташування точки, повороту та масштабу, тому що знаходять характерні точки в заданих масштабах і призначають їх орієнтацію.

Для обчислення вектору дескриптора для кожної характерної точки в першу чергу створюється набір гістограм напрямків на 4 × 4 сусідніх пікселях з 8 областями в кожній. Ці гістограми обчислюються з значень величини і орієнтації елементів в області 16 × 16 навколо ключової точки. Величини зважуються функцією Гаусса з σ, яка дорівнює половині ширини вікна дескриптора. Дескриптор стає вектором всіх значень цих гістограм, який має 128 елементів, бо існує 16 гістограм з 8 областями у кожній. Цей вектор нормалізується до одиничної довжини, щоб забезпечити інваріантність аффінним змінам в освітленні. Щоб скоротити ефект нелінійного освітлення, застосовується поріг величиною 0,2 і вектор знову нормалізується. Процес застосування порога може поліпшити результати зіставлення, навіть якщо ефекти нелінійного освітлення відсутні. На рисунку 2.1 схематично показана частина зображення і отриманий на її основі дескриптор.

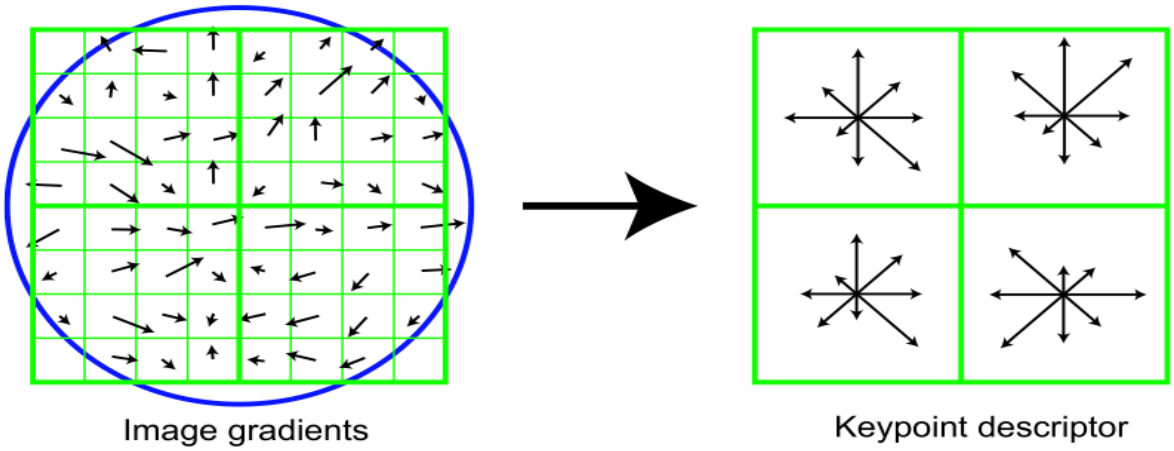


Рисунок 2.1 – Приклад масиву з дескрипторів 2x2
отриманого з вибірки 8x8

Хоча розмірність дескриптора (128) здається високою, дескриптори з меншою розмірністю настільки добре не працюють, а обчислювальні витрати залишаються

низькими. Довші дескриптори давали б кращі результати, але не на багато, і є небезпека збільшення чутливості до спотворень і перешкодам.

2.2 Дескриптор SURF

Іншим відомим алгоритмом є Speeded Up Robust Features (SURF) [8, 17]. Метод позиціонується як покращений варіант SIFT методу зі збільшеною швидкістю детектування і побудови дескриптора. Основні стадії алгоритму збігаються з SIFT: детектуються особливі точки, вважається кут домінуючого градієнта, і вважається дескриптор локальної області. Для детектування особливих точок вважається визначник матриці гессіан і визначаються локальні максимуми. Аналогічно з SIFT, відкидаються по пороговому значенням визначника нестабільні точки і для подальшої обробки беруться найбільш стабільні з максимальними значеннями визначника.

SURF вирішує два завдання - пошук особливих точок зображення та створення їх дескрипторів, інваріантних для масштабу та обертання. Це означає, що опис ключової точки буде однаковий, навіть якщо зразок змінить розмір і буде повернутий (тут і далі буде говоритись тільки про обертання в площині зображення). Крім того, сам пошук ключових точок теж повинен мати інваріантність. Так що, щоб повернутий об'єкт сцени мав той самий набір ключових точок, що і зразок.

Для виявлення характерних точок інваріантного способом в масштабі SURF використовує підхід каскадного фільтрації. Загалом, техніка для досягнення масштабної інваріантності полягає в тому, щоб дослідити зображення в різних масштабах, масштабному просторі, використовуючи гаусові ядра. SURF розподіляє масштабний простір на рівні та октави. Октава відповідає подвоєнню σ, а октава ділиться на рівномірно розподілені рівні.

SURF створює піраміди мап відповідей, з різними рівнями в октавах. Мапа відповідей є результатом операції на зображенні. Характерні точки є такими точками, що екстремуми серед 8 сусідів на поточному рівні та його 2×9 сусідів на рівень нижче і вище. Це так зване «придушення не-максимумів» (Non-Maximum Suppression) в районі $3 \times 3 \times 3$, співвідношення між рівнями, октавами та околицями яке ілюструється на рисунку 2.2.

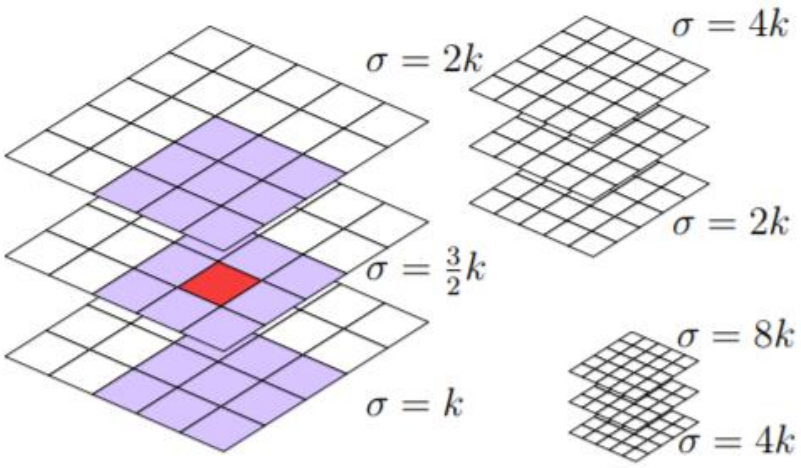


Рисунок 2.2 - 3 октави з трьома рівнями, виділено сусідство для «придушення не-максимумів» 3 × 3 × 3, що використовується для виявлення особливих точок

Тобто метод шукає особливі точки за допомогою матриць Гессе:

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix},$$

де H — матриця Гессе, f (x,y) — функція зміни градієнта яскравості.

Детермінант матриць Гессе (або гессіан) досягає екстремуму в точках максимального зміни градієнта яскравості:

$$\det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2.$$

Він добре виявляє плями, кути та границі ліній. Тому SURF пробігається фільтром ядра гауса по всьому зображенню і знаходить точки, в яких досягається максимальне значення детермінанта матриці Гессе. Відзначимо, що такий прохід виділяє як темні плями на білому тлі, так і світлі плями на темному тлі.

Гессіан інваріантний відносно обертання. Но не інваріантний масштаб. Тому SURF використовує різномасштабні фільтри для пошуку гессіанів.

Для кожної ключової точки вважається напрямок максимального зміни яскравості (градієнта) і масштабу, взятого з масштабного коефіцієнта матриці Гессе. Градієнт у точці обчислюється за допомогою фільтрів Хаара.

Після находження ключових точок, SURF формує їх дескриптори. Дескриптор представляє собою набір з 64 (або 128) цифр для кожної ключовій точки. Ці числа відображають коливання градієнта по ключовій точці.

Оскільки ключова точка представляє собою максимум гессіана, то це гарантує, що в окремій точці повинні бути ділянки з різними градієнтами. Таким чином, забезпечується різниця дескрипторів для різних ключових точок.

На підставі наявної інформації виконується побудова дескрипторів для кожної особливої точки (рис. 2.3):

– навколо точки будується квадратна околиця розміром 20s, де s - масштаб, на якому отримано максимальне значення детермінанта матриці Гессе;

– отримана квадратна область розбивається на блоки, в результаті вона буде розбита на 4 x 4 регіони;

– для кожного блоку обчислюються більш прості ознаки. Як наслідок, виходить вектор, що містить 4 компоненти: 2 - це сумарний градієнт по квадранту, 2 - сума модулів точкових градієнтів;

– дескриптор формується в результаті склеювання зважених описів градієнта для 16 квадрантів навколо особливої точки. Елементи дескриптора зважуються на коефіцієнти ядра Гауса. Ваги необхідні для більшої стійкості до шумів у віддалених точках;

– додатково до дескриптора заноситься відмітки матриці Гессе. Ці компоненти необхідні, щоб розрізняти темні і світлі плями. Для світлих точок на темному тлі відмітка негативна, для темних точок на світлому фоні - позитивна.

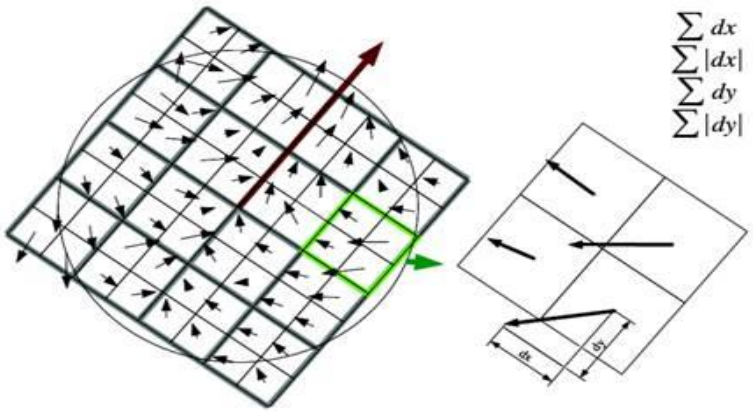


Рисунок 2.3 – Демонстрація побудови дескриптора

2.3 Дескриптор ORB

ORB представлений в 2011р [9,17]. В його основі лежить комбінація таких алгоритмів як детектор FAST (Features from Accelerated Segment Test) [10] і дескриптор BRIEF (Binary Robust Independent Elementary Features) [11] з деякими поліпшеннями.

Для пошуку кутових точок по черзі розглядаються області по 16 пікселів навколо кожного пікселя p.

Якщо існує N пікселів в її околиці довжиною 16 пікселів, та усі N яскравіше $I_p + t$ або темніше $I_p – t$, де I_p – яскравість точки p, t – порогова величина, то точка p помічається як можливо особлива. Якщо є помічена точка, тоді проводиться дослідження яскравості в області цієї точки – номера пікселів1, 5, 9, 13 (рис. 2.4). Точок тільки чотири, бо це дозволяє швидко відсіяти точки, що не підходять. Точка p вважається характерною, якщо не менше трьох пікселів виконують умову:

$$I_i < I_p - t \text{ OR } I_i > I_p + t;\ i = 1...4$$

За замовчуванням в алгоритмі ORB максимальна кількість характерних точок не більше 500, якщо їх більше, то до ним застосовується детектор кутів Харріса [12], для виключення найменш значущих.

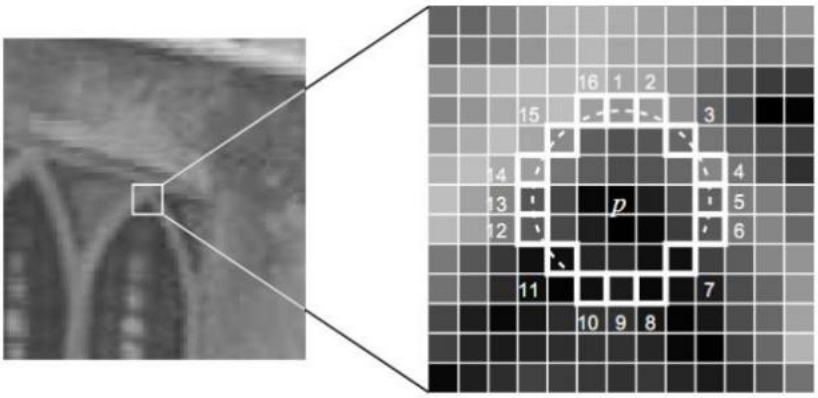


Рисунок 2.4 – Область точки p FAST детектора

Для інваріантності до масштабування застосовується алгоритм піраміди Гаусса. Початкове стиснуте зображення c_0 з заданим кроком і визначає октаву c_i. Введення параметра кутової орієнтації дозволяє домогтися стійкості детектування при обертанні об'єкта. Він заснований на напрямках градієнта яскравості щодо центру точки, напрямок з найбільшою інтенсивністю призначається орієнтацією особливої точки θ.

BRIEF дескриптор - це вектор довжиною 256, що складається з результатів бінарних тестів навколо особливої точки. В околиці 31 × 31 піксель порівнюються середні значення яскравості між x і y, де x, y - області 5 × 5 пікселів:

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

де I – середня яскравість обраної області.

Область обчислення дескриптора орієнтується по орієнтації особливої точки θ, для досягнення інваріантності. Матрицю S розмірністю 2 × n формують усі n = 256 наборів x_i та y_i. Далі за допомогою матриці повороту R_θ орієнтується відповідно кута θ, де $S_q = R_q S.$

А сам вектор дескриптора записується як:

$$g_n(\mathrm{I},\theta) := f_n(\mathrm{I}) \mid (x_i, y_i) \in S_\theta, f_n(\mathrm{I}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathrm{I}; x_i, y_i) .$$

2.4 Дескриптор KAZE

П. Ф. Алькантарілья висунув у 2012 році KAZE [17] особливості, які використовують нелінійну багато масштабну піраміду за допомогою нелінійної дифузійної фільтрації [13]. Це робить розмиття зображень локально пристосованим до характерних точок, тим самим зменшуючи шум і одночасно зберігаючи межі областей у предметних зображеннях.

Детектор KAZE заснований на нормованому детермінанті шкали матриці Гессіану, який обчислюється в декількох масштабних рівнях. Максимуми відгуку детектора вибираються у вигляді особливих точок з використанням рухомого вікна. Опис особливості вводить властивість інваріантності обертання шляхом знаходження домінуючої орієнтації в круговому сусідстві навколо кожного виявленого ознаки.

Характерні точки KAZE інваріантні до обертання, масштабу, обмеженої афінності та мають більшу виразність у різних масштабах при вартості помірного збільшення часу на обчислення. Стандартна формула нелінійної дифузії:

$$\frac{\partial L}{\partial t} = \mathrm{div}(c(x, y, t).\nabla L),$$

де c - функція провідності, div - дивергенція, ∇ - градієнтний оператор, а L - яскравість зображення.

2.5 Дескриптор AKAZE

У 2013 році П. Ф. Алькантарілья презентував алгоритм Accelerated-KAZE (AKAZE) [14,17], який також заснований на нелінійній дифузійній фільтрації, як KAZE, але його нелінійні багато масштабні піраміди будуються за допомогою Fast Explicit Diffusion (FED). Використання нелінійного коефіцієнту масштабування дозволяє збільшити швидкість пошуку необхідної характерної точки.

Детектор AKAZE заснований на детермінанті матриці Гессіана, тобто для кожної октави L^i в піраміді обчислюється визначник:

$$L^i_{\mathrm{Hessian}} = \sigma^2_{i,\mathrm{norm}}(L^i_{xx}L^i_{yy} - L^i_{xy}L^i_{xy}) ,$$

де $\sigma_{i,\mathrm{norm}} = \frac{\sigma_i}{2^{0^i}}$ - нормалізований відносно масштабу коефіцієнт, для розрахунку L^i з розрахунком розміру октави σ_i.

Якість інваріантності обертання покращується за допомогою фільтрів Шарра [15], за допомогою цього фільтра розраховується друга похідна з кроком $\sigma_{i,\mathrm{norm}}$. За допомогою цього підходу, відшуковуються такі точки в октаві, при яких значення фільтра вище заданого порогу та найбільший в області 3x3 пікселів.

Дескриптор AKAZE заснований на алгоритмі модифікованої локальної різниці (Modified Local Difference Binary - MLDB):

1) Вікно дескриптору орієнтується за орієнтацією характерної точки.

2) Інваріантність до масштабу отримана за допомогою вибору розміру вікна дескриптора в залежності від розміру октави в якій знайдена особлива точка.

2.6 Дескриптор BRISK

Даний метод представлений в 2011р [17, 41]. Детектування особливих точок здійснюється за допомогою FAST (Features from Accelerated Segment Test) та дескриптору BRIEF з деякими змінами.

Для досягнення інваріантності до масштабу обирається найкраща характерна точка з максимальним значенням інтенсивності в піраміді, яка складається з 4 октав c_i і 4 внутрішніх октав d_i, i = 0…3. Октави формуються як стиснення оригінального зображення c_0 в $2i$ рази. Внутрішні октави розташовані між c_i та $c_i + 1$ і представлені у вигляді стислої c_0 у $\frac{2}{3}2^i$ рази. Пошук особливих точок в октавах здійснюється детектором FAST (рис. 2.5).

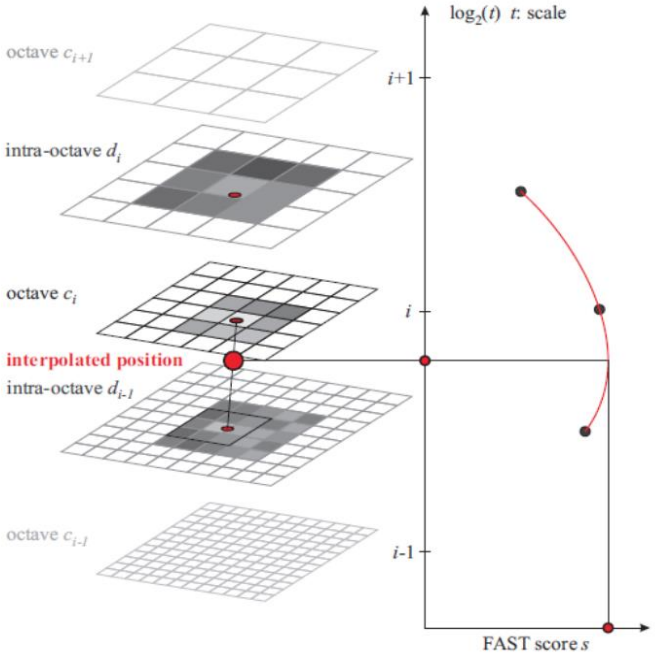


Рисунок 2.5 - Приклад пошуку особливої точки

Область навколо особливої точки розбивається на 60 ділянок p (рис. 2.6):

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

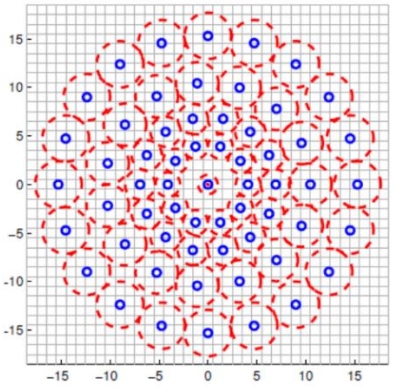


Рисунок 2.6 - Область обчислення дескриптора

Множина A розбивається на 2а підмножини:

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

де $d_{min} = 13.67t, d_{max} = 9.75t$, t - розмір особистої точки.

Обчислюється середнє значення градієнта множини L:

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

Дескриптор складається з бінарного рядка довжиною 512, заповненої результатами проведених тестів в множини S:

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

де $I(p_i^{\alpha}, \sigma_i)$ - інтенсивність околиці радіусу σ_i точки p_i, $\alpha = \arctan 2(g_y, g_x)$ - кут напрямку градієнта g.

2.7 Отримання параметрів нормалізації геометричних перетворень

2.7.1 Пошук відповідностей методом найближчого сусіда

Для порівняння пари зображень в основному використовують метод порівняння заснований на обчисленні відстаней всіх можливий пар дескрипторів $\rho(d_i, d_j')$, d – дескриптор першого зображення, вектор ознак α^k, d′– дескриптор другого зображення, вектор ознак α'^k, де $\forall d_i \in D, \forall d'_j \in D', i = 1...|D|, j = 1...|D'|$, розмірність вектору ознак |K| визначається в залежності від використовуваного методу опису точки.

Для визначення відстаней зазвичай використовується Евклідова метрика:

$$\rho(d_i, d'_j) = \sqrt{\sum_{k=0}^{|K|} |\alpha_k - \alpha'_k|^2} .$$

Дескриптори зіставляються з операцією, відомої як k пошук найближчих сусідів (пошук k-NN), який полягає в знаходженні k елементів, які найбільш схожі на даний дескриптор запиту. Пошук k-найближчих сусідів обчислює відстань між дескрипторами вхідного зображення і зображення з колекції і повертає k пар з найменшою відстанню до об’єкта, що класифікується.

Далі для кожного дескриптора d_i обираються два йому найближчих d'_j та навпаки. Якщо у вибраного d вже є відповідні йому два дескриптора, то він пропускається і пошук продовжується. У підсумку кожному дескриптору d_i будуть відповідати не більш двох взаємно найближчих з D′.

Задача пошуку найближчого сусіда є задачею оптимізації, яка полягає у відшуканні у множині елементів, розташованих у багатовимірному метричному просторі, елементів, близьких до заданого, відповідно до заданої функції близькості.

Формально ця задача ставиться наступним чином: надано множину точок S у просторі M та точку $q \in M$, необхідно знайти найближчу до q точку в S. Дональд Кнут у книзі «Мистецтво програмування» назвав це проблемою поштового відділення, посилаючись на застосування цієї задачі до пошуку найближчого поштового відділення.

Прямим узагальненням задачі пошуку найближчого сусіда є алгоритм пошуку k-NN, який призначений для пошуку k найближчих точок.

Вводиться параметр відносин довжин:

$$u = \frac{\rho_{i1}}{\rho_{i2}} \left(\rho_{i1} < \rho_{i2} \right),$$

де ρ_{i1} та ρ_{i2} відстані між можливими парами дескрипторів.

За даним параметром відсіваються дескриптори які не відповідають необхідному рівню визначеності. Якщо u більше заданого порогу u_{max}, то d_i далі не розглядається, інакше для d_i ставиться у відповідність дескриптор d_j з відстанню ρ_{i1}.

2.7.2 Отримання параметрів нормалізації на основі алгоритму RANSAC

Алгоритм RANdom SAMple Consensus (RANSAC) [3], запропонований Фішлером і Болсом, представляє собою загальний підхід оцінки параметрів, призначений для усунення значної долі викидів вхідних даних.

RANSAC є метод передискретизації, який генерує варіанти вирішення, використовуючи мінімальні спостереження числа (точки даних), необхідні для оцінки базових параметрів моделі. Як було відзначено Фішлера і Боллесом, на відміну від звичайних методів вибірки, які використовують якомога більше даних для отримання вихідного рішення, а потім приступають до скорочення викидів, RANSAC використовує найменший набір і продовжує збільшувати цей набір з послідовним точок даних.

Відфільтрувати дескриптори тільки по дистанції недостатньо для досягнення високої точності визначення схожих об'єктів на зображеннях. Якщо об'єкт рухається на сцені або знятий з іншого ракурсу, то при застосуванні трансформації «накладення» n точок одного зображення на відповідні по найближчому сусіду n точок іншого, можна виявити особливості, які не відносяться до загального об'єкту і тим самим зменшити кількість хибно певних зав’язків.

Схема роботи алгоритму RANSAC полягає в циклічному повторенні пошуку матриці трансформації H між випадково обираються чотирма особливими точками на одному зображенні і відповідно їм чотирьом точкам на другому (рис. 2.7). Зв’язок між двома відповідними точками x_i и x'_i в однорідних координатах може бути переписаний як:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \Leftrightarrow x_i = Hx'_i .$$

У неоднорідних координатах:

$$x'_i = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}},$$
$$y'_i = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}}.$$

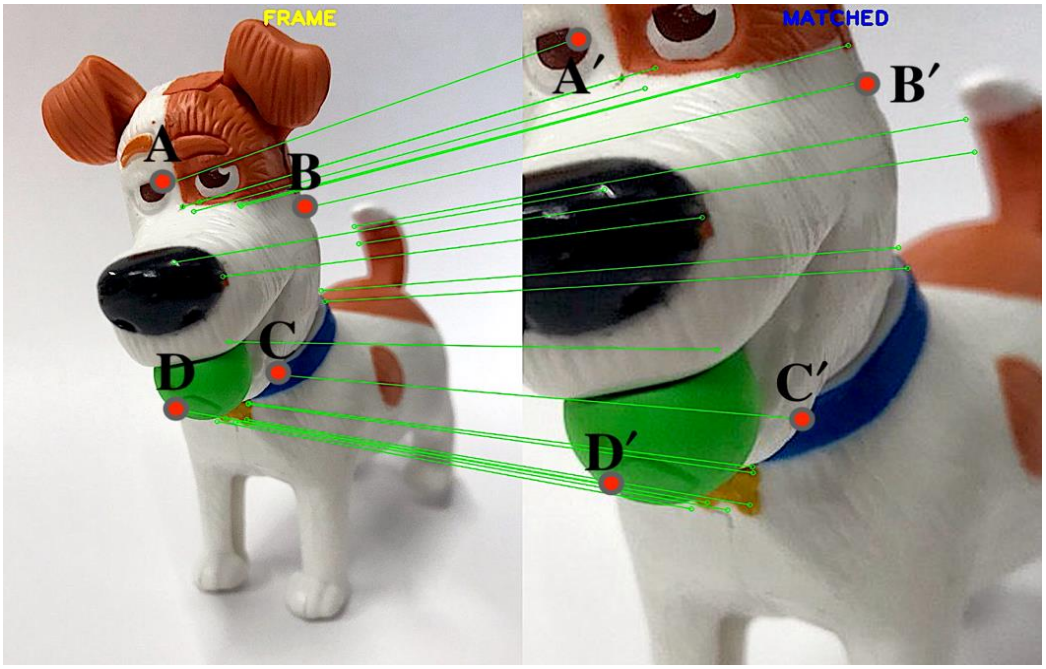


Рисунок 2.7 – Вибір чотирьох точок відповідностей

Іноді у технічній літературі вважають, що параметр $h_{33}\neq 0$, та розглядають проективні перетворення, що мають вигляд:

$$X_0(x,y) = X\left(\frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}, \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}\right).$$

Параметри проективного перетворення у цьому випадку для точок ABCD та A′B′C′D′ визначаються за формулами:

$$\begin{cases} h_{13} = x_{A'}; & h_{23} = y_{A'}; \\ h_{31} = \dfrac{d_1(y_{C'} - y_{B'}) - d_2(x_{C'} - x_{B'})}{AB(d_3 - d_4)}; & h_{32} = \dfrac{d_1(y_{C'} - y_{D'}) - d_2(x_{C'} - x_{D'})}{AD(d_4 - d_3)}; \\ h_{11} = \dfrac{x_{D'}(h_{31}AB + 1) - h_{13}}{AB}; & h_{12} = \dfrac{x_{B'}(h_{31}AD + 1) - h_{13}}{AD}; \\ h_{21} = \dfrac{y_{D'}(h_{31}AB + 1) - h_{23}}{AB}; & h_{22} = \dfrac{y_{B'}(h_{32}AD + 1) - h_{23}}{AD}; \end{cases}$$

де AB, AD – довжини відрізків та

$$\begin{aligned} d_1 &= x_{D'} - x_{C'} + x_{B'} - h_{13}, \\ d_2 &= y_{D'} - y_{C'} + y_{B'} - h_{23}, \\ d_3 &= (x_{C'} - x_{D'})(y_{C'} - y_{B'}), \\ d_4 &= (y_{C'} - y_{D'})(x_{C'} - x_{B'}). \end{aligned}$$

У разі, коли є більше 4 точки відповідності, проблема полягає в рішенні для гомографії, яка мінімізує вартість відповідної функції.

Геометрична відстань - визначає Евклідовому відстань зображення між тим, де гомографія відображає точку і де спочатку знаходилося відповідність точки. Іншим терміном для цього є помилка зворотної проекції. Припускаючи, що в другому зображенні є тільки помилки, загальна помилка проекції для набору відповідностей $x_i \rightarrow x'_i$ дорівнює:

$$\sum_i d(x'_i, Hx_i)^2,$$

де H – це оціночна гомографія, и d(~,~) – є евклідова відстань зображення між двома точками.

У більш реалістичному разі наявності помилок в обох зображеннях ми мінімізуємо симетричну помилку передачі, в якій враховуються як прямі (H), так і зворотні (H^{-1}) перетворення. При цьому симетрична помилка передачі:

$$\sum_i d(x'_i, Hx_i)^2 + d(x_i, H^{-1}x'_i)^2 =$$

$$\sum_i \left[\left(x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left(y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \right].$$

Оціночна гомографія H буде тією, для якої це рівняння мінімізовано.

Щоб мінімізувати цю або наступну функцію витрат, потрібно ітеративний підхід. Хоча результати часто більш точні, ітераційні методи мають недоліки в порівнянні з лінійними алгоритмами. Ітеративні алгоритми повільніше, ризик не сходиться, і виникають додаткові проблеми, такі як вибір початкових оцінок і критерії зупинки.

Всі вихідні дані можна розділити на два типи: хороші точки, що задовольняють моделі, «не-викиди» або «інлаери» (inlier) і помилкові точки, шуми - випадкові включення в вихідні дані, «викиди» або «аутлаери» (outlier).

Таким чином, схема алгоритму RANSAC:

– довільно вибрати мінімальну кількість точок (m≥4), необхідних для визначення параметрів моделі;

– вирішити параметри моделі;

– визначити, скільки точок з набору всіх точок відповідає визначеним допуску;

– якщо добова частина числа «не-викидів» на точки загального числа в наборі перевищує заданий пороговий τ, то переоцінюються параметри моделі, використовуючи всі ідентифіковані рівні і алгоритм завершується;

– в іншому випадку повторити кроки з 1 по 4 (максимум N раз).

Кількість ітерацій, N, вибирається досить високим, щоб ймовірність p (зазвичай рівна 0.99) полягала в тому, що хоча б один з наборів випадкових вибірок не містить викиду. Нехай u представляє ймовірність того, що будь-яка вибрана точка даних є «не-викид» та υ=1-u - це ймовірність спостереження «викиду». Необхідні N ітерації мінімальної кількості точок, позначених m, де $1-p=(1-u^m)^N$ і, таким чином, з деякими маніпуляціями:

$$N = \frac{\log(1-p)}{\log(1-(1-\upsilon)^m)}.$$

Перевагою алгоритму RANSAC є його здатність дати надійну оцінку параметрів моделі, тобто можливість оцінити параметри моделі з високою точністю, навіть якщо у вихідному наборі даних присутня значна кількість викидів.

Одним з недоліків методу RANSAC є відсутність верхньої межі часу, необхідного для обчислення параметрів моделі. Якщо використовувати в якості деякої межі часу максимальне число ітерацій, отримане рішення може бути не оптимальним, а також існує дуже мала ймовірність, що жодна модель не буде відповідати вихідним даним.

3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО МЕТОДУ НОРМАЛІЗАЦІЇ НА ОСНОВІ АНАЛІЗУ ХАРАКТЕРНИХ ТОЧОК ДЕСКРИПТОРІВ SURF, SIFR, ORB, KAZE, AKAZE, BRISK

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках магістерської роботи були проведені досліди нормалізації геометричних перетворень зображень за допомогою опису характерних точок дескрипторами SURF, SIFR, ORB, KAZE, AKAZE, BRISK. Для реалізації було обране високопродуктивне середовище IntelliJ IDEA 2019.1.1 для розроблення програм мовою Java.

Java - строго типізована об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems. Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якої комп'ютерної архітектурі за допомогою віртуальної Java-машини. У роботі застосовується Java Runtime Environment (JRE): 1.8.0_202-release-1483-b44 x86_64, Java Virtual Machine (JVM): OpenJDK 64-Bit Server VM by JetBrains s.r.o. Використання версії Java SE 8 обумовлено тим, що в роботі застосовуються потоки (stream) для роботи з колекціями та лямда-вирази.

Технічні характеристики використовуваної комп'ютерної системи: macOS 10.14.4, Intel Core i5 CPU при 2,9 ГГц, 2 ядра; кеш 2-го рівня (у кожному ядрі): 256 КБ; кеш 3-го рівня: 3 МБ та 16,00 ГБ оперативної пам’яті.

Для виконання експериментів використовувався OpenCV версії 3.4.5. Для дослідження було обрано дескриптори SIFT, SURF, BRISK, ORB, KAZE, AKAZE. При чому SURF розширений до SURF (64D), SURF (128D), які представляють 64 та 128 числовий вектор відповідно. ORB розширюються до ORB(1000), тим самим перший обмежується до сто тисяч, а другий до однієї тисячі характерних точок (Додаток Б). У таблиці 3.1 показані OpenCV об'єкти, що використовуються для детекторів функцій-дескрипторів, параметри, що не згадуються - використовуються за замовчанням OpenCV. Також необхідно зазначити, що використання дескрипторів SURF та SIFT є не комерційним, та несе тільки досліджувану мету.

Таблиця 3.1 - OpenCV об'єкти для визначення дескрипторів

Назва алгоритму	Об’єкт OpenCV
SIFT	SIFT sift = SIFT.create();
SURF (128D)	SURF surf128 = SURF.create(); surf128.setExtended(true);
SURF (64D)	SURF surf64 = SURF.create(); surf128.setExtended(false);
ORB	ORB orb = ORB.create(); orb.setMaxFeatures(100000);
ORB (1000)	ORB orb1000 = ORB.create(); orb1000.setMaxFeatures(1000);

KAZE	KAZE kaze = KAZE.create();
	kaze.setNOctaveLayers(3);
	kaze.setExtended(true);

AKAZE	AKAZE akaze = AKAZE.create(); akaze.setNOctaveLayers(3);
BRISK	BRISK brisk = BRISK.create();

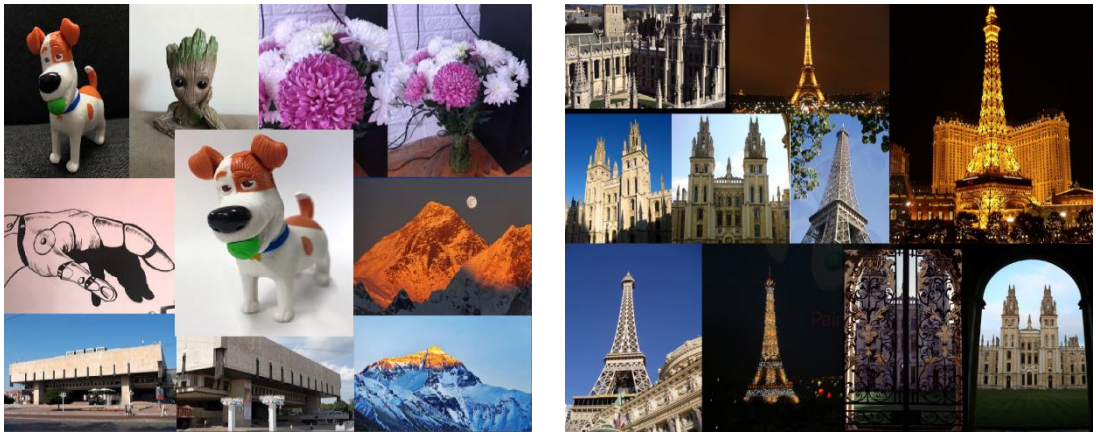
Методи що описані в таблиці описуються як: для дескриптору SURF метод setExended - false означає, що обчислюються основні дескриптори (по 64 елемента кожен), true - означає, що слід розраховувати розширені дескриптори (128 елементів у кожному); для методу ORB метод setMaxFeatures встановлює верхню границю для пошуку характерних точок, за замовчуванням OpenCV встановлює значення у 500 точок; у дескрипторах KAZE та AKAZE метод setNOctaveLayers – це кількість підрівнів на шкалі масштабу.

3.2 Опис тестових зображень, умов та задач експериментів щодо дослідження інваріантності

Для дослідження питання нормалізації за наявністю геометричних перетворень, було використано реальні та синтезовані зображення з заданими перетвореннями.

Синтезовані зображення – це зображення отримане за рахунок застосування геометричного перетворення зображення-еталону.

Для дослідження було використано три набори даних з зображеннями. Набір даних А (рис. 3.1 - а) є персональною збіркою, зображень зроблених власноруч або знайдені у загальному доступі. Набір даних Б (рис 3.1 - б) – зображення, які надаються університетом Oxford [16] зі збірок Oxford Buildings and Paris Datasets.



а) б)

Рисунок 3.1 – Набори зображень:
а - персональні зображення; б - зображення з сайту Oxford

Набір даних В (рис. 3.2) – це тестові дані, що пропонує OpenCV для тестування перетворень на реальних зображеннях та складається з 4 збірок, кожна з яких містить 6 зображень.

У якості тестової колекції зображень буде використано файли, які мають JPEG формат та розміри від 300 x 400 до 400 x 500 пікселів.

а) б) г) д)

Рисунок 3.2 – Набори зображень: а – збірка bark; б – збірка ubc;
г – збірка wall; д – збірка graf

Зображення з набору даних А, Б та В у кількості 50 штук використовувались для синтезування нових зображень, спотворених геометричними перетвореннями, порівняння різних аспектів детектування та опису характерних точок і отримання на їх основі матриці гомографії. Більше зображень представлено у додатку А. Крім того набір даних В використовувався для аналізу реальних зображень.

В роботі досліджуються наступні геометричні перетворення: масштаб, поворот, центроафінне та афінне перетворення. Так як всі алгоритми отримання дескрипторів, що розглядаються у роботі, інваріантні щодо перетворення зміщення, і в практичної реалізації існують лише невеликі відмінності дескрипторів для зміщених точок за рахунок дискретизації, то немає необхідності проводити дослідження нормалізації в умовах

перетворення зміщення. Як результат також будуть проведені дослідження на реальних зображеннях, коли параметри не будуть відомі.

Матрицю геометричного перетворення можна описати як:

$$H = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix},$$

де a_{11}, a_{22} — коефіцієнти масштабування відповідно по осях OX, OY, a_{11}, a_{12} , a_{21}, a_{22} - відповідають за поворот зображення; a_{13}, a_{23} - коефіцієнти які відповідають за зміщення по осях OX, OY та a_{12}, a_{21} — коефіцієнти відповідно косого зсуву по осях OX та OY.

В роботі будуть проводитись досліди для знаходження точності знайдених параметрів геометричних перетворень Dparam та точності зміщення пікселів після нормалізації Dpixel. Формула знаходження точності геометричних параметрів:

$$\Delta\mathrm{param} = \frac{\sum_{i,j=1}^{2} \rho(a_{ij}, a'_{ij})}{M}, \qquad (3.1)$$

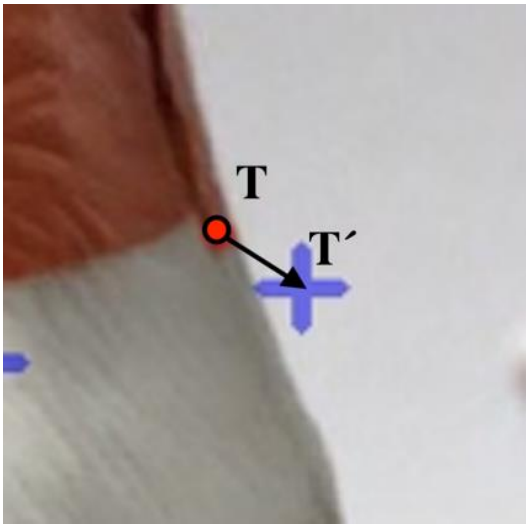
де a_{ij} – параметри заданої матриці центроафінного перетворення; a'_{ij} – параметри отриманої матриці центроафінного перетворення; M = 4 – кількість параметрів перетворення, $r(a_{ij}, a'_{ij})$ – міра подібності, яка у роботі буде розрахована як норма L1:

$$r(a_{ij}, a'_{ij}) = |a_{ij} - a'_{ij}|.$$

Алгоритм для розрахунку параметру точності піксельного зіставлення: на оригінальному зображенні B є характерно точка T та відповідна на синтезованому зображенні B_{new}, яке отримане за допомогою перетворення h, точка T_{new}. Після знаходження матриці H (яка найкращим чином описує перетворення h) відтворюється нормалізація зображення B_{new} за допомогою отриманої зворотної матриці H отримуючи нормалізоване зображення B′ на якому точка T_{new} переходить у точку T′.

Як результат отримується відстань між цими точками $r(T_i, T'_i)$. На рисунку 3.3 ілюстровано зображення після нормалізації та зіставлення зображень, де червоні крапки – це початкові характерні точки на еталоні T та сині хрести – нормалізовані точки T′.

а)



б)

Рисунок 3.3 – Результат зіставлення зображень: а – еталонне та нормалізоване зображення, б – приклад вимірювання відстаней між точками

Точність по пікселям буде дорівнювати сумі таких відстаней по всіх відповідних точках у зображенні і поділена на кількість відповідностей:

$$\Delta\text{pixel} = \frac{\sum_{i=1}^{N} \rho(T_i, T'_i)}{N}, \tag{3.2}$$

де N – кількість відповідностей характерних точок, $r(T_i, T'_i)$ – міра подібності, яка у роботі буде розрахована як норма L2:

$$r(T_i, T'_i) = \sqrt{(x_{T_i} - x_{T'_i})^2 + (y_{T_i} - y_{T'_i})^2}.$$

3.3 Дослідження питання щодо інваріантності до геометричних перетворень дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK

У рамках магістерської роботи проведене комплексне порівняння алгоритмів SIFT, SURF, KAZE, AKAZE, ORB та BRISK.

Метою є дослідження та вирішення питання щодо визначення найкращого за точністю та швидкодією алгоритму нормалізації за наявності таких геометричних

перетворень, як масштаб, поворот, перетворення групи подібності та центроафінного перетворення.

Тому в роботі досліджуються кількісні показники (3.1, 3.2), а також часові характеристики п'ятьох основних етапів нормалізації зображень на основі опису характерних точок:

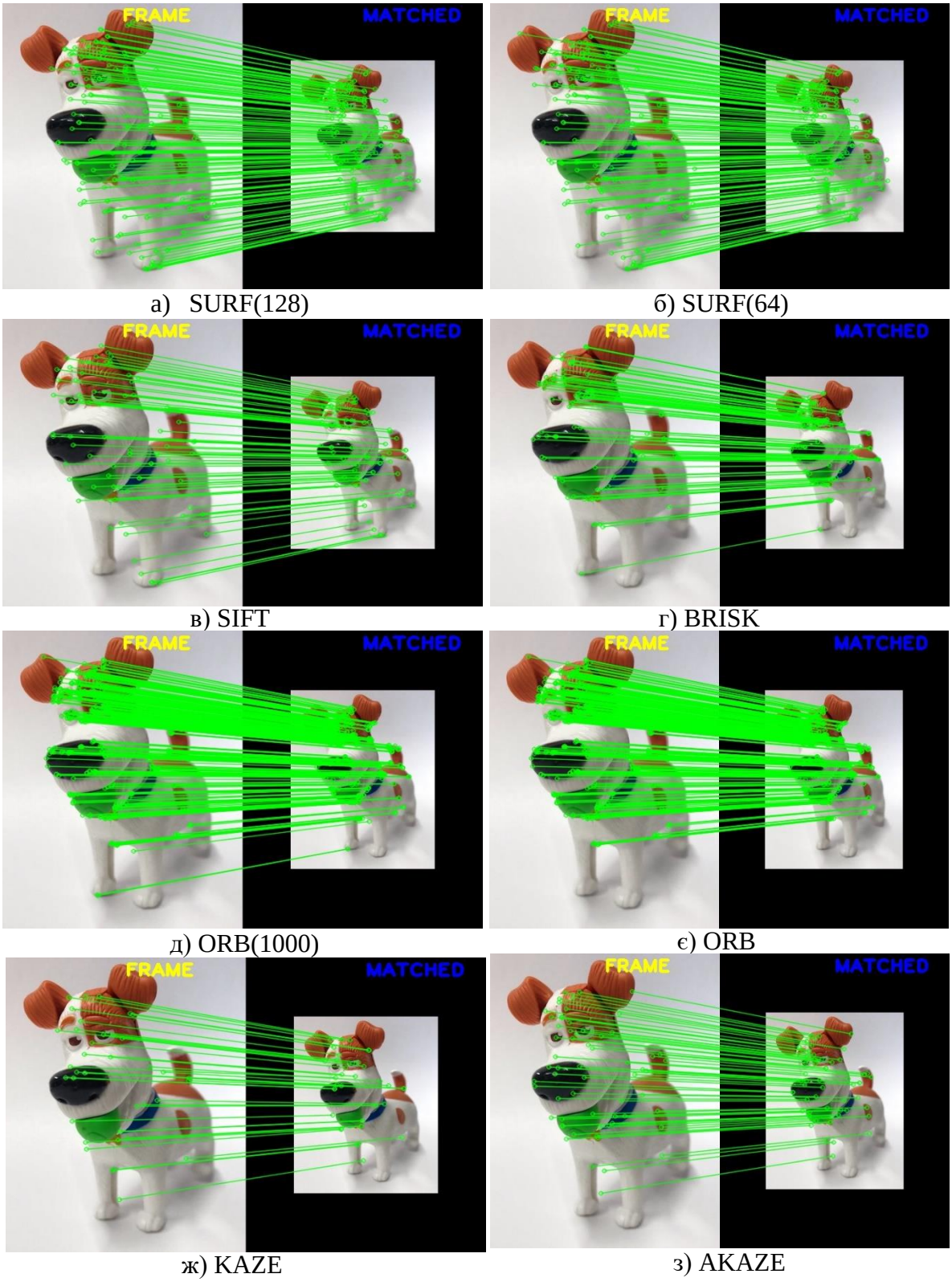
- виявлення характерних точок та їх опис;
- зіставлення на відповідність;
- видалення викидів у відповідностях;
- виведення матриці та параметрів перетворення;
- нормалізація зображень.

3.3.1 Дослідження нормалізації за наявності перетворення масштабу

Для дослідження нормалізації за наявності перетворення масштабу був обраний діапазон однорідного масштабування при коефіцієнтах $k \in [0.5, 2]$. На рисунку 3.4 зображені результати дослідження одного з зображень після зіставлення на відповідність за алгоритмами найближчого сусіду й RANSAC при вхідній матриці перетворення, де k=0.6:

$$h = \begin{bmatrix} 0.6 & 0 & 70.8 \\ 0 & 0.6 & 94 \\ 0 & 0 & 1 \end{bmatrix}.$$

В таблиці 3.2 зведено дані з точності отриманих параметрів та піксельного зіставлення і як результат на рисунку 3.5 їх діаграми.



а) SURF(128) б) SURF(64)

в) SIFT г) BRISK

д) ORB(1000) є) ORB

ж) KAZE з) AKAZE

Рисунок 3.4 – Відповідності для дескрипторів SURF, SIFT, ORB, KAZE, AKAZE, BRISK за наявністю перетворення масштабу з коефіцієнтом k=0.6

Таблиця 3.2 – Отримана матриця гомографії та параметри точності для одного зображення за наявністю перетворення масштабу k=0.6

Дескриптор	Отримана матриця гомографії	Δparam	Δpixel
SIFT	$H = \begin{bmatrix} 0.5683 & 0.0061 & 73.2103 \\ -0.002 & 0.5983 & 95.4062 \\ -0.0001 & 0 & 1 \end{bmatrix}$	0,0149	3,8746
SURF(128)	$H = \begin{bmatrix} 0.594 & 0.0013 & 71.3707 \\ 0.0001 & 0.5979 & 93.9643 \\ 0 & 0 & 1 \end{bmatrix}$	0,0024	1,9913
SURF(64)	$H = \begin{bmatrix} 0.5952 & 0.0001 & 71.3219 \\ 0.0008 & 0.5968 & 94.0552 \\ 0 & 0 & 1 \end{bmatrix}$	0,0023	2,1374
ORB	$H = \begin{bmatrix} 0.5944 & 0.0006 & 71.431 \\ -0.0035 & 0.5998 & 94.4018 \\ 0 & 0 & 1 \end{bmatrix}$	0,0025	1,5343
ORB(1000)	$H = \begin{bmatrix} 0.5942 & 0.0017 & 71.3868 \\ -0.0044 & 0.6016 & 94.3894 \\ 0 & 0 & 1 \end{bmatrix}$	0,0034	1,5077
KAZE	$H = \begin{bmatrix} 0.5462 & 0.0019 & 74.3039 \\ -0.0365 & 0.5893 & 96.2613 \\ -0.0002 & 0 & 1 \end{bmatrix}$	0,0257	8,0643
AKAZE	$H = \begin{bmatrix} 0.6008 & 0.007 & 70.317 \\ -0.0095 & 0.6129 & 94.2787 \\ 0 & 0 & 1 \end{bmatrix}$	0,0076	2,4403
BRISK	$H = \begin{bmatrix} 0.6005 & -0.0011 & 70.862 \\ 0.0001 & 0.6007 & 93.8803 \\ 0 & 0 & 1 \end{bmatrix}$	0,0006	1,7552

На основі розрахунку параметрів, які зображені у таблиці 3.2, були отримані діаграми залежності точності знайдених параметрів геометричного перетворення Δparam (3.1) від коефіцієнту масштабування k, де $k \in [0.5, 2]$, майже всі наведені дескриптори вказали схожий результат, але дескриптор KAZE видався найбільш нестійким починаючи від коефіцієнту 0.5 до 0.8 та показав стрибок для параметрів 1.6 до 2.

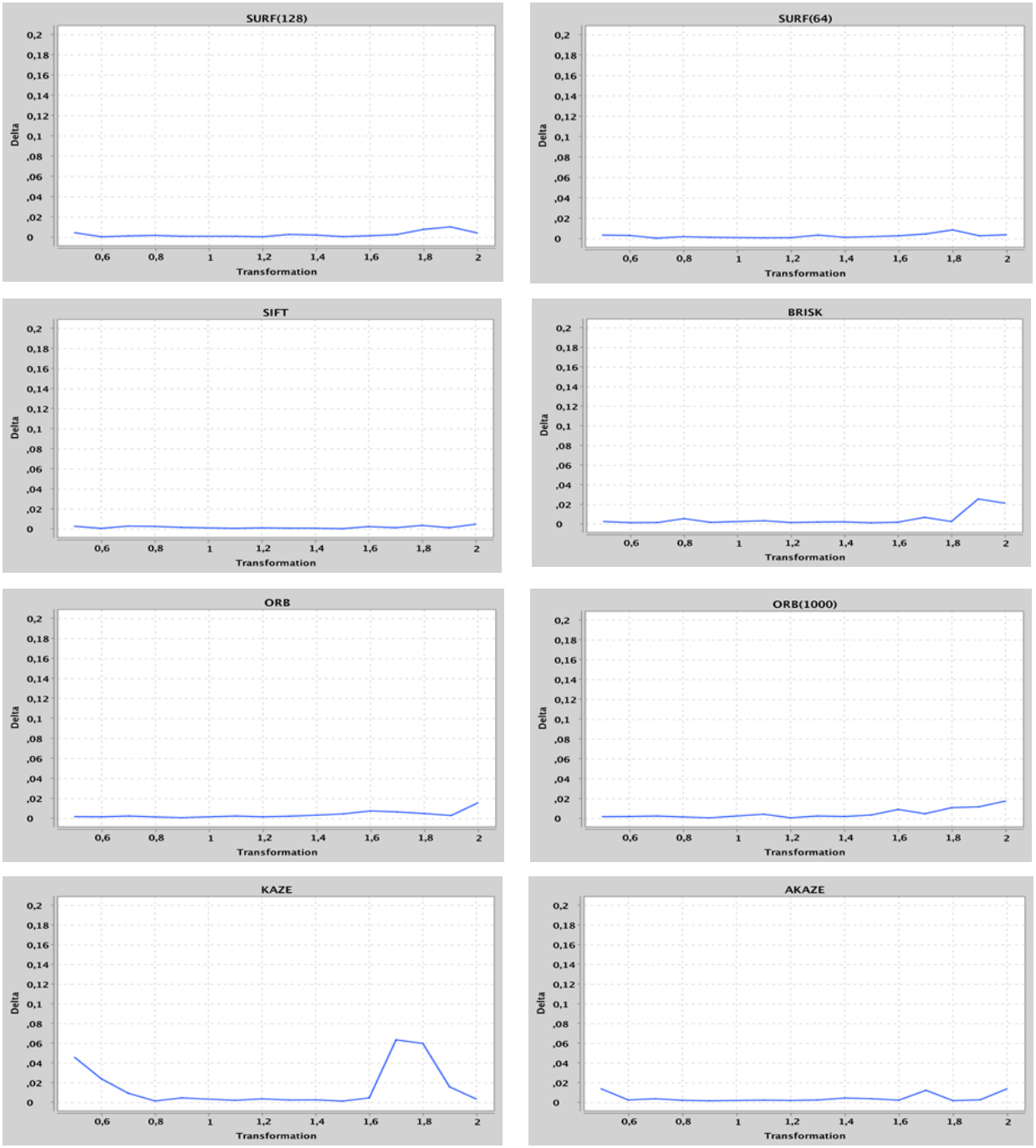


Рисунок 3.5 – Графіки залежності Δparam від параметру масштабу k,
де k∈[0.5, 2], для одного зображення

Також були отримані параметри точності зіставлення пікселів Δpixel (3.2) та побудовані діаграми залежності від коефіцієнту k, де k∈[0.5, 2] (рис. 3.6). На діаграмах видно чіткий скачок дескрипторів KAZE та AKAZE при збільшенні та зменшенні коефіцієнту масштабування та починаючи з 1.7 ORB росте у гору.

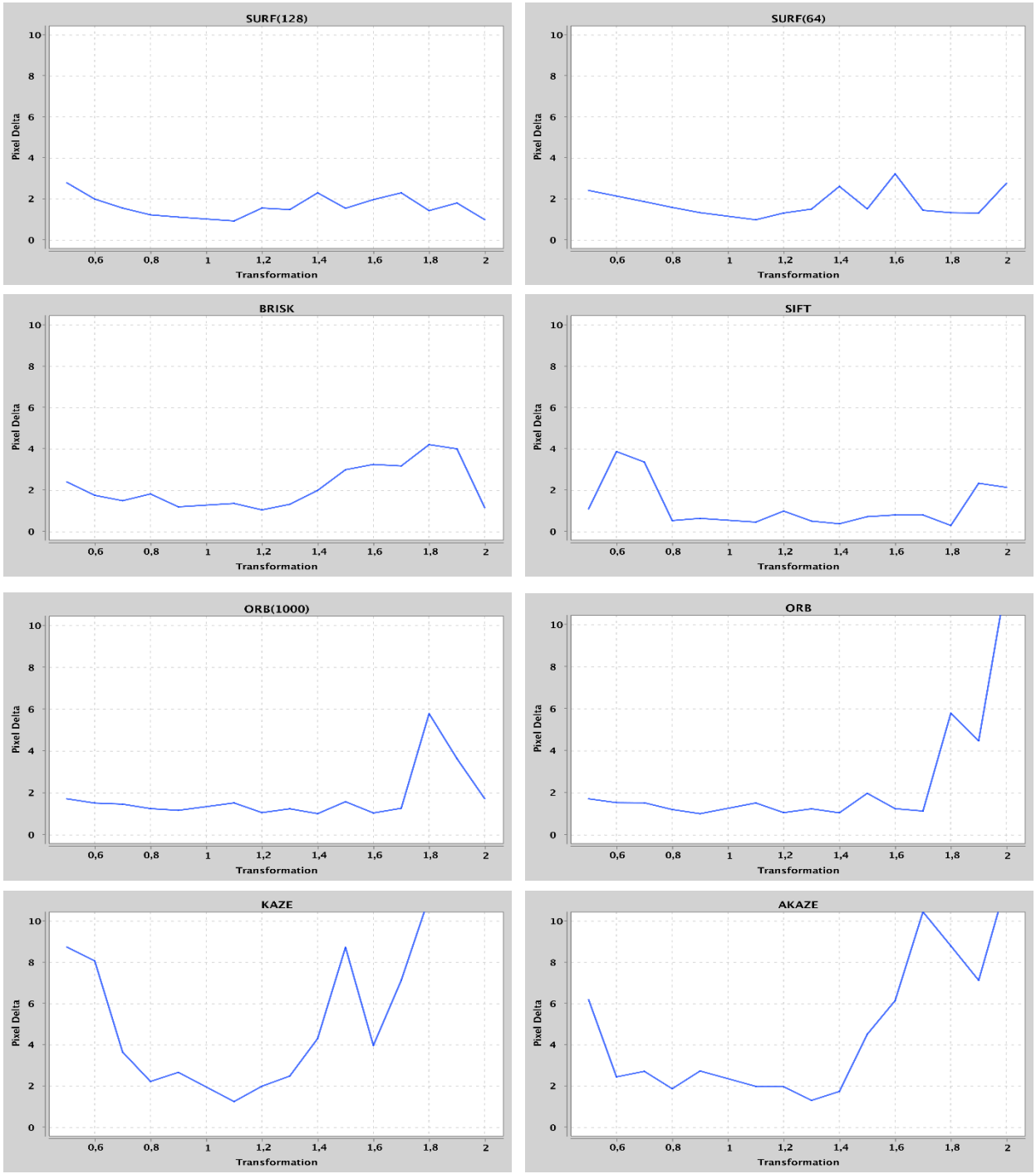


Рисунок 3.6 – Графіки залежності Δpixel від параметру масштабу k,
де k∈[0.5, 2], для одного зображення

На рисунку 3.7 наведено зіставлені нормалізовані зображення та 20 відповідних накладених характерних точок.

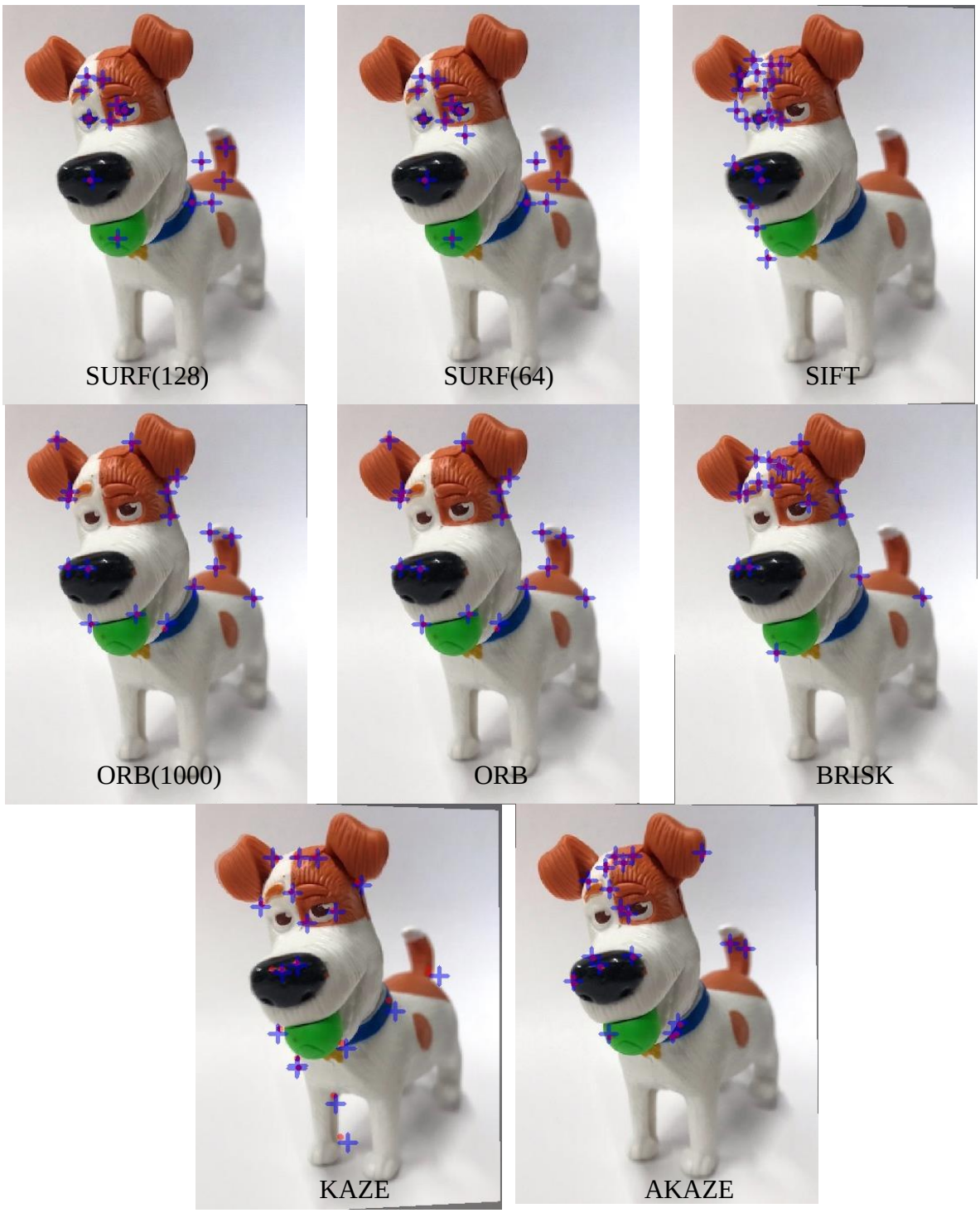


Рисунок 3.7 – Зіставленні зображення та їх характерні точки

Так як одне зображення не є показником у достовірності й дуже часто викиди можуть спотворювати данні, то було проведено досліди на 50 тестових зображеннях та побудовані таблиця для усередненої оцінки точності параметрів зображені у таблиці 3.3.

Таблиця 3.3 – Усереднені оцінки точності Δparam та Δpixel для аналізованих дескрипторів за наявності перетворення масштабу з коефіцієнтом k∈[0.5, 2]

k	Точність	SURF 128	SURF 64	SIFT	ORB 1000	ORB	KAZE	AKAZE	BRISK
0.5	Δparam	0.005	0.008	0.011	0.024	0.004	0.028	0.022	0.006
	Δpixel	2.991	4.292	4.312	7.219	2.539	10.94	6.878	4.108
0.6	Δparam	0.005	0.004	0.017	0.011	0.008	0.015	0.0108	0.003
	Δpixel	2.937	2.854	4.69	4.108	3.507	6.05	4.102	1.92
0.7	Δparam	0.002	0.002	0.002	0.005	0.001	0.018	0.01	0.003
	Δpixel	1.654	1.736	1.01	2.273	1.387	5.38	3.456	1.836
0.8	Δparam	0.003	0.0023	0.002	0.004	0.002	0.005	0.004	0.001
	Δpixel	1.691	1.699	0.771	1.883	1.399	2.201	2.02	1.246
0.9	Δparam	0.002	0.002	0.001	0.003	0.002	0.004	0.003	0.001
	Δpixel	1.239	1.381	0.549	1.617	1.561	1.645	1.24	0.962
1.1	Δparam	0.002	0.002	0.001	0.005	0.0014	0.003	0.002	0.0016
	Δpixel	1.123	1.27	0.326	1.985	1.069	1.369	0.962	0.921
1.2	Δparam	0.004	0.004	0.001	0.005	0.002	0.005	0.004	0.002
	Δpixel	1.605	1.648	0.331	1.855	1.052	1.784	1.477	1.032
1.3	Δparam	0.003	0.004	0.001	0.008	0.002	0.006	0.007	0.004
	Δpixel	1.34	1.488	0.46	2.321	1.195	1.815	2.026	1.435
1.4	Δparam	0.005	0.004	0.001	0.007	0.003	0.016	0.008	0.0027
	Δpixel	1.549	1.473	0.414	2.127	1.26	3.96	2.031	1.095
1.5	Δparam	0.006	0.005	0.001	0.009	0.004	0.009	0.013	0.005
	Δpixel	1.626	1.483	0.322	2.368	1.332	2.225	2.89	1.492
1.6	Δparam	0.009	0.009	0.001	0.0155	0.06	0.011	0.014	0.007
	Δpixel	2.64	2.428	0.345	3.547	1.612	2.48	2.965	1.781
1.7	Δparam	0.011	0.011	0.002	0.0142	0.007	0.0181	0.013	0.009
	Δpixel	2.319	2.681	0.415	2.962	1.641	3.843	2.689	2.328
1.8	Δparam	0.02	0.011	0.002	0.0864	0.0078	0.0195	0.025	0.01
	Δpixel	4.307	2.398	0.435	15.045	1.825	3.683	5.039	2.416
1.9	Δparam	0.0137	0.0105	0.003	0.11	0.011	0.048	0.045	0.009
	Δpixel	2.804	2.201	0.562	17.533	2.349	8.479	12.344	1.931
2.0	Δparam	0.008	0.013	0.003	0.113	0.017	0.036	0.241	0.012
	Δpixel	1.606	2.658	0.658	22.371	3.257	6.58	11.205	2.448

На основі таблиці 3.3 було побудовано графіки залежності Δparam та Δpixel від коефіцієнта масштабування зображених на рисунках 3.8 та 3.9.

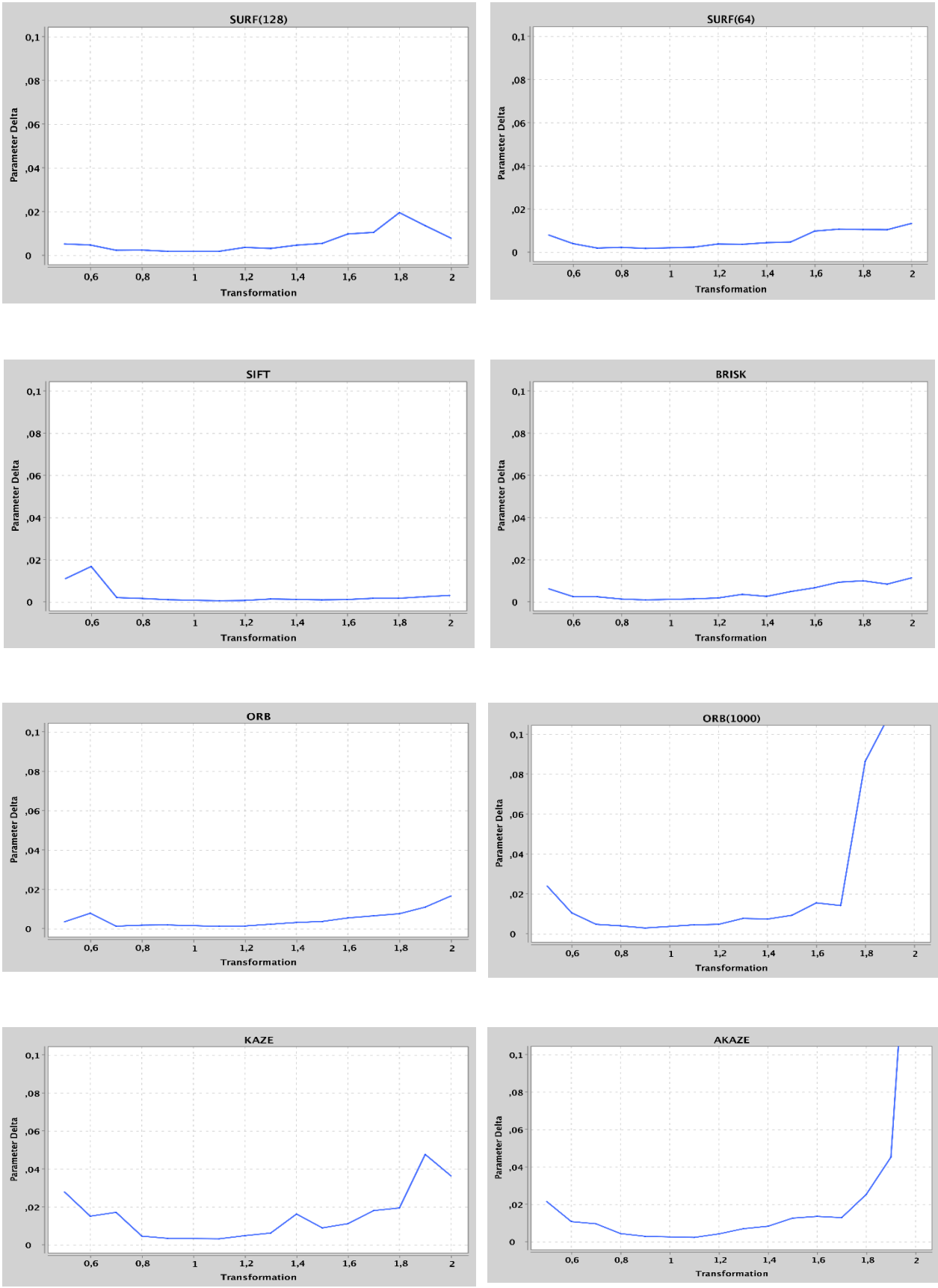


Рисунок 3.8 – Графіки оцінки точності Δparam для 50 зображень, де k∈[0.5, 2]

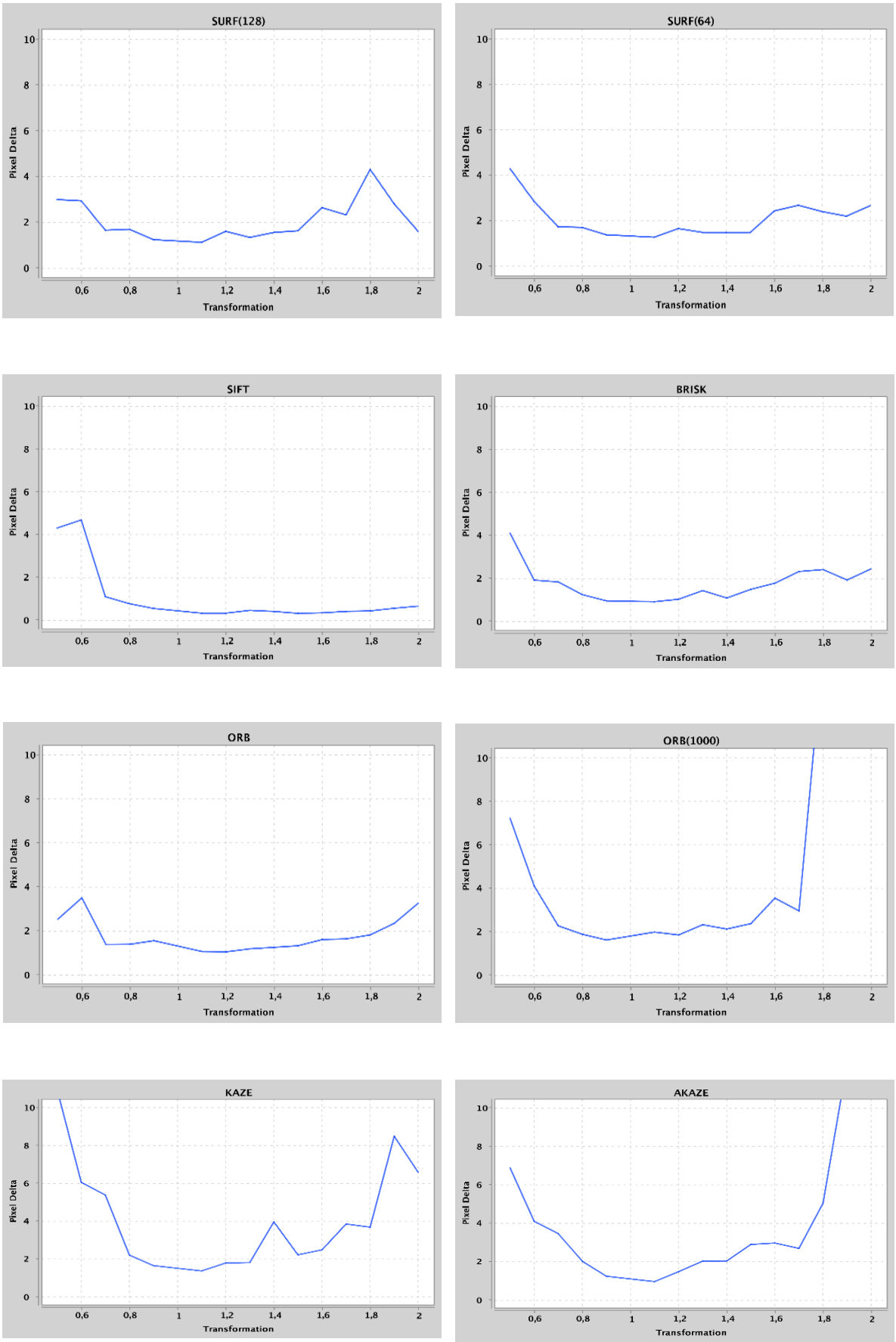


Рисунок 3.9 – Графіки оцінки точності Δpixel для 50 зображень, де k∈[0.5, 2]

Виводи щодо експериментів на 50 зображеннях в умовах однорідного масштабу, де $k \in [0.5,2]$ та змінюється з кроком 0.1.

В результаті досліджень зроблено такі висновки відносно точності відхилення параметрів вихідного перетворення від параметра нормалізації Δparam (3.1):

1) Найточніше коефіцієнт масштабу був визначений методами SIFT, BRISK, ORB, SURF(128), SURF(64):

– метод BRISK поводився найбільш стабільно і Δparam не перевищувало 0,012 на всьому діапазоні зміни коефіцієнта масштабу, а діапазоні [0.6; 1.5] не перевищує 0.005;

– метод SIFT у діапазоні [0.7; 2] показує відхилення найнижче (близьке до нуля, менш 0.002), однак при величині коефіцієнта менш 0.7 піднімалося до 0.017;

– методи ORB, SURF(128), SURF(64) у діапазоні [0.7; 1.5] мають відхилення менш ніж 0.006, але за межами цього діапазону показують незначні підвищення (для SURF(64) Δparam=0.013 при k=2; для ORB Δparam=0.017 при k=1.8; для SURF(128) Δparam=0.02 при k=2).

2) Методи KAZE, ORB (1000), AKAZE показали себе менш стабільно за наявності масштабу:

– на аналізованому діапазоні зміни коефіцієнта масштабу метод KAZE показує Δparam вище 0.02, але не більше 0.04, в діапазоні [0.6; 1.8] не перевищує 0.02;

– методи ORB (1000), AKAZE при коефіцієнті масштабу більш до 1.8 показують відхилення до 0.02, однак далі відхилення Δparam різко збільшується і дорівнює 0.11 для методу ORB (1000) і 0.24 для методу AKAZE, відповідно.

Висновки щодо середньої відстані Δpixel між відповідними характерними точками на оригінальному зображенні і після нормалізації:

1) Найбільш точні результати показали методи SURF (128), SURF (64), SIFT, ORB, BRISK, максимальне середня відстань Δpixel в розглянутому діапазоні не перевищило 4 пікселів, причому:

– метод SIFT в діапазоні [0.8; 2] показав саме низьке відхилення: менше 1 пікселя (<0.77);

– метод ORB в діапазоні [0.8; 1.8], метод BRISK в діапазоні [0.6; 1.6], методи SURF(128), SURF(64) в діапазоні [0.7; 1.5] показали відхилення менше 2 пікселів.

2) Методи ORB (1000), KAZE, AKAZE мають викиди величини Δpixel на краях розглянутого діапазону зміни коефіцієнта масштабу, так:

– у діапазоні [1.7;2] методи KAZE, AKAZE, ORB(1000) величина Δpixel, досягає значень: 8.479, 12.344, 22.371 відповідно;

– при найменшому коефіцієнті масштабу, що дорівнює 0.5, методи ORB (1000) і AKAZE показують відхилення Δpixel близько 7 піселей, метод KAZE близько 11 пікселів;

– відстань менш 4 пікселів метод показує ORB (1000) в діапазоні [0.7; 1.7], KAZE в діапазоні [0.8; 1.8], AKAZE в діапазоні [0.6; 1.7].

Отже, можна зробити висновок, що в розглянутому діапазоні зміни коефіцієнта масштабу [0.5; 2] методи SURF(128), SURF(64), SIFT, ORB, BRISK показують більш високу точність нормалізації, ніж методи ORB(1000), KAZE, AKAZE.

Найвищу точність дає метод SIFT: в діапазоні [0.7; 2] середня відстань між вихідними і нормалізованими точками менше 1 пікселя. Методи ORB(1000), KAZE, AKAZE рекомендується застосовувати тільки у випадках невеликих змін масштабу, а саме, якщо коефіцієнт масштабу від 0.8 до 1.7, проте на цьому діапазоні вони дають відхилення до 4-х пікселів, коли методи SURF(128), SURF(64), SIFT, ORB, BRISK тільки до 2-х.

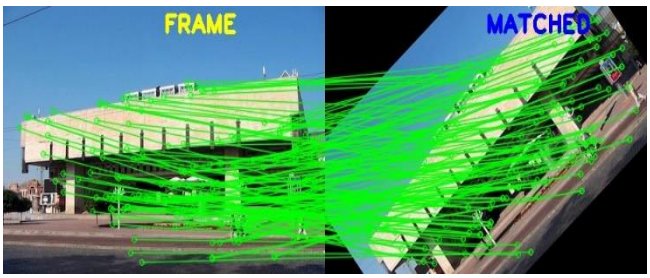
Таким чином, проведені дослідження точності нормалізації дозволяють розташувати методи у такій послідовності (починаючи з найкращого у сенсі найменших відхилень): SIFT < ORB, BRISK < SURF (64), SURF(128) < ORB(1000), AKAZE, KAZE.

Однак, усі методи показують достатню точність для застосування на практиці за наявностю маштабу з коефіцієнтом у діапазоні [0.8; 1.3], тому що Δparam має значення близько 0.008, а Δpixel – близько 2.3, що є припустимим для більшості практичних задач.

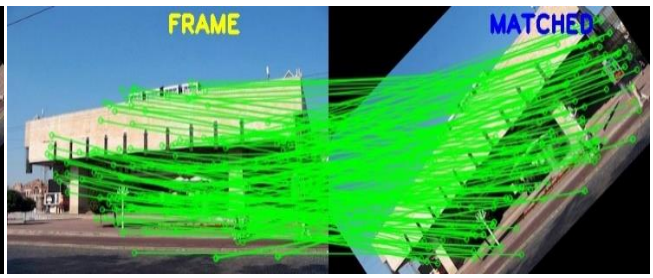
3.3.2 Дослідження нормалізації за наявності перетворення повороту

Для дослідження нормалізації за наявності перетворення повороту був обраний діапазон кута повороту α від 0 до 90 градусів. На рисунку 3.10 та 3.11 зображені результати дослідження одного з зображень після зіставлення на відповідність. Де матриця перетворення для повороту на 40 градусів навколо центру зображення:

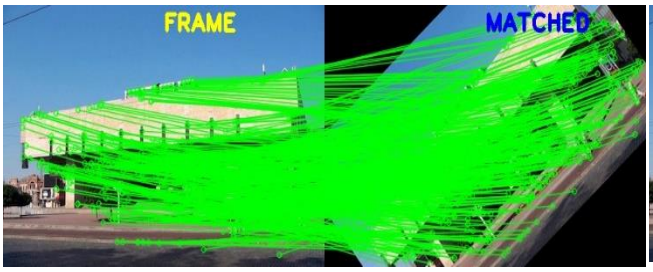
$$h = \begin{bmatrix} 0.766 & 0.6428 & -41.207 \\ -0.6428 & 0.766 & 178.7847 \\ 0 & 0 & 1 \end{bmatrix}.$$



SURF(128)



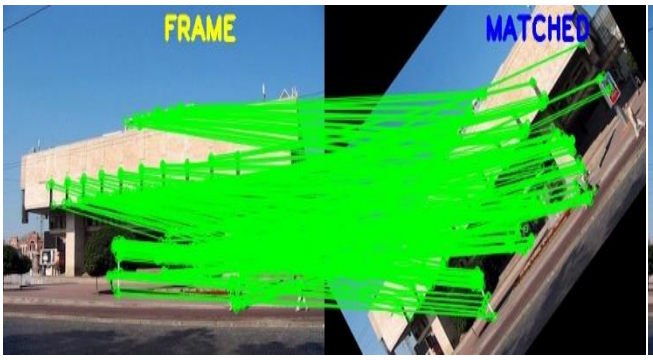
SURF(64)



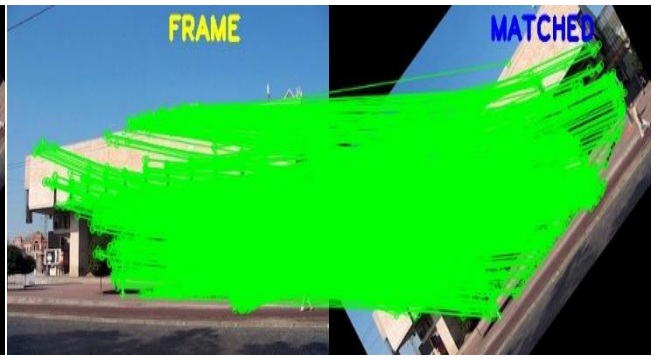
SIFT

BRISK

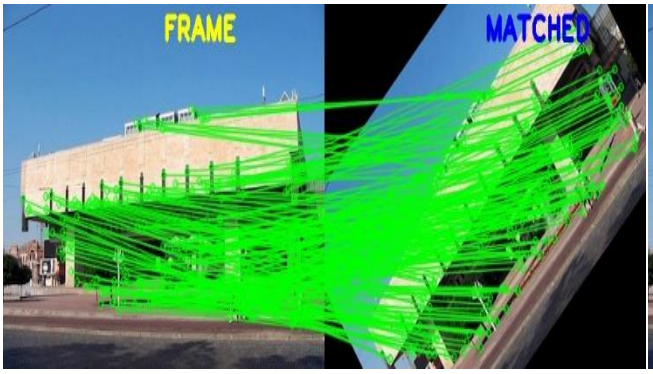
Рисунок 3.10 – Відповідності для дескрипторів SURF, SIFT, BRISK за наявності повороту з кутом α =40



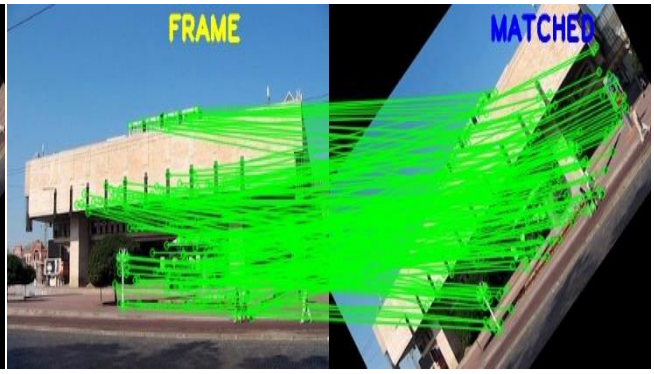
ORB(1000)



ORB



KAZE



AKAZE

Рисунок 3.11 – Відповідності для дескрипторів ORB, KAZE, AKAZE за наявності повороту з кутом α =40

Таблиця 3.4 – Отримана матриця гомографії та параметри Δparam, Δpixel для одного зображення за наявністю перетворення повороту α=40

Назва	Отримана матриця гомографії	Δparam	Δpixel
SIFT	$H = \begin{bmatrix} 0.766 & 0.6422 & -41.266 \\ -0.6424 & 0.765 & 178.9605 \\ 0 & 0 & 1 \end{bmatrix}$	0,0003	0,15
SURF(128)	$H = \begin{bmatrix} 0.7664 & 0.6326 & -40.7801 \\ -0.6373 & 0.7566 & 178.136 \\ 0 & 0 & 1 \end{bmatrix}$	0,0064	2,2979
SURF(64)	$H = \begin{bmatrix} 0.7682 & 0.645 & -41.8421 \\ -0.6451 & 0.7707 & 178.6536 \\ 0 & 0 & 1 \end{bmatrix}$	0,0028	1,8761
ORB	$H = \begin{bmatrix} 0.7688 & 0.6444 & -41.565 \\ -0.6425 & 0.7679 & 178.4756 \\ 0 & 0 & 1 \end{bmatrix}$	0,0016	1,1038
ORB(1000)	$H = \begin{bmatrix} 0.7728 & 0.6504 & -42.6497 \\ -0.6435 & 0.7719 & 178.4218 \\ 0 & 0 & 1 \end{bmatrix}$	0,0052	1,8951
KAZE	$H = \begin{bmatrix} 0.7712 & 0.6655 & -44.1287 \\ -0.6503 & 0.7748 & 180.731 \\ 0 & 0 & 1 \end{bmatrix}$	0,011	3,2895
AKAZE	$H = \begin{bmatrix} 0.7496 & 0.649 & -42.0043 \\ -0.6458 & 0.7714 & 179.3137 \\ 0 & 0 & 1 \end{bmatrix}$	0,0045	1,4603
BRISK	$H = \begin{bmatrix} 0.766 & 0.6397 & -40.8596 \\ -0.6411 & 0.7636 & 178.5932 \\ 0 & 0 & 1 \end{bmatrix}$	0,0018	0,8832

На основі розрахунку параметрів для α=40, які зображені у таблиці 3.4, були отримані графіки (рис 3.12 – 3.13) та точність зіставлення перших 20 характерних точок (рис. 3.14). Аналізуючи результати:

– найкращий результат показали дескриптори SIFT, ORB, ORB(1000) та BRISK, у яких на усьому діапазоні значень градусів повороту мають дуже маленький параметр оцінки – приближується до нуля;

– гарні результати показує дескриптори AKAZE (<0,01) та KAZE (<0,013), які мають невеликі викиди при 78 градусів та 42 градусів відповідно;

– дескриптор SURF (128) має трохи гірші результати ніж інші раніше перераховані дескриптори (<0.025);

– SURF (64) має викиди у діапазоні [50, 56], які досягають значень до 0.04.

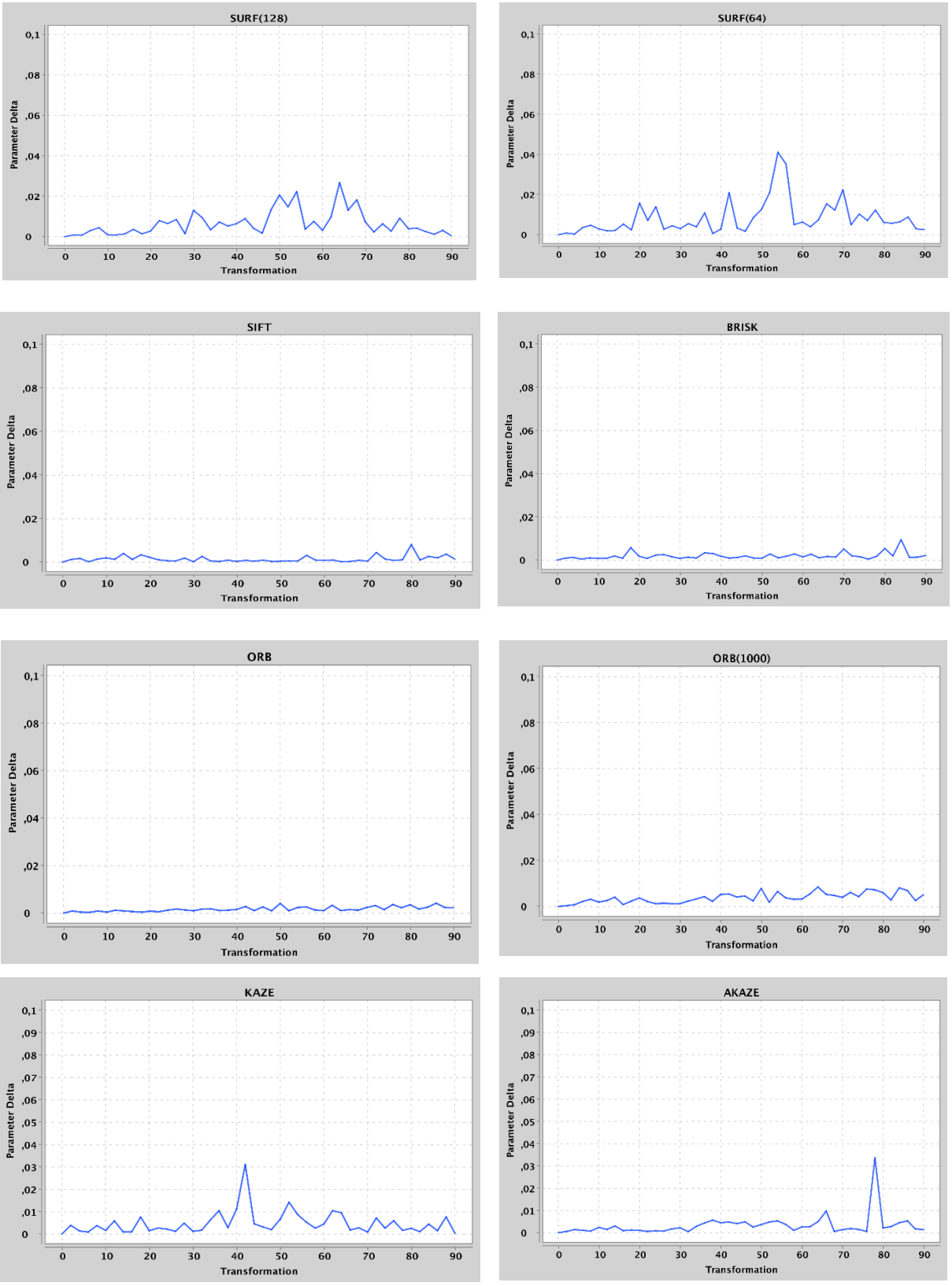


Рисунок 3.12 – Графіки оцінки точності Δparam при α∈[0, 90] для одного зображення

Оцінюючи точність зіставлення пікселів можна сказати що: кращі результати спостерігаються у методів SIFT, ORB, BRISK, ORB (1000); у дескриптора AKAZE є чіткі викиди (78 градусів), але на всьому діапазоні показую теж гарні результати (< 3 пікселів); відповідно дескриптори SURF (128), SURF (64), KAZE мають дуже багато стрибків Δpixel (до 8 пікселів) у продовж усього діапазону [20, 80] градусів.

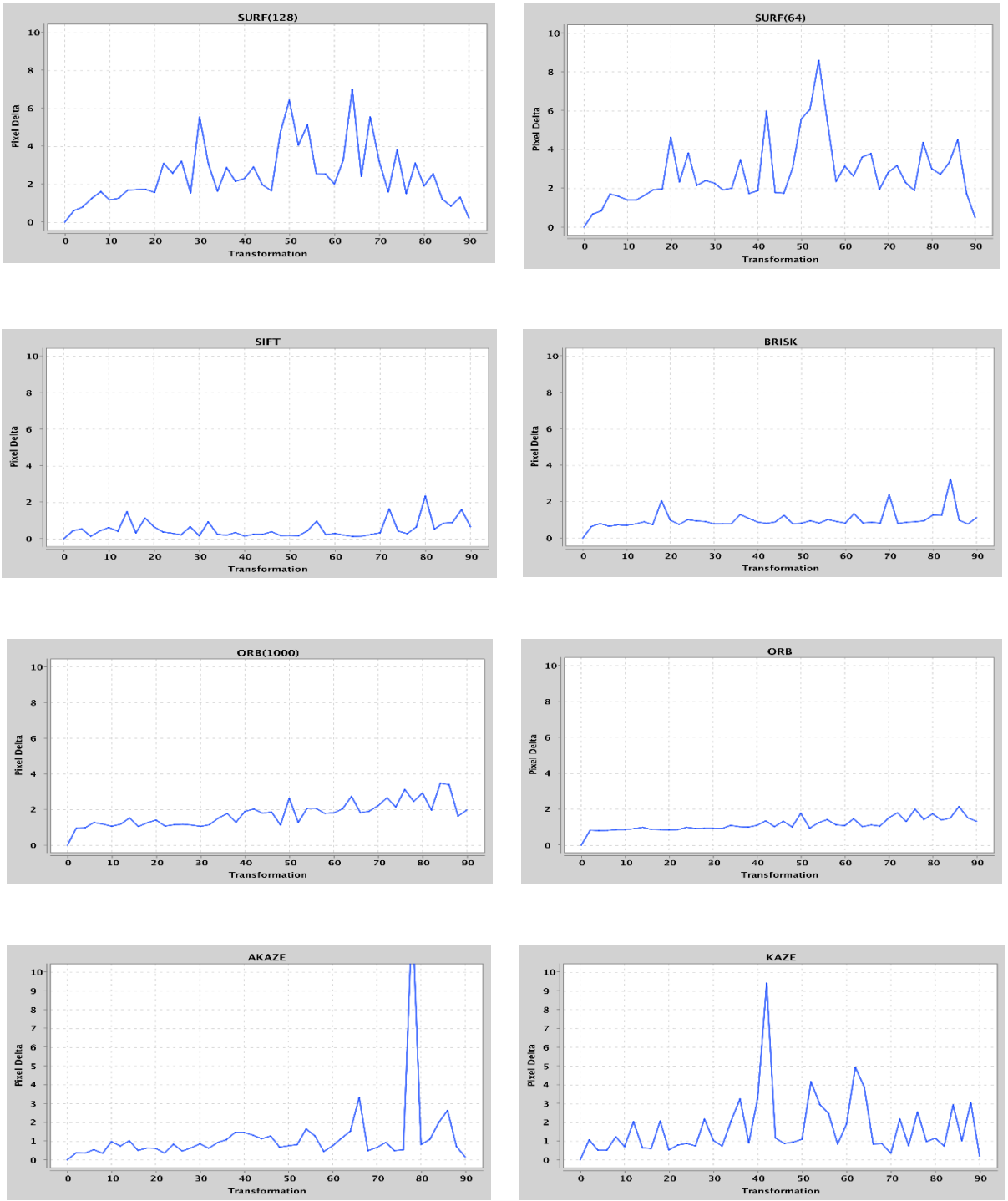
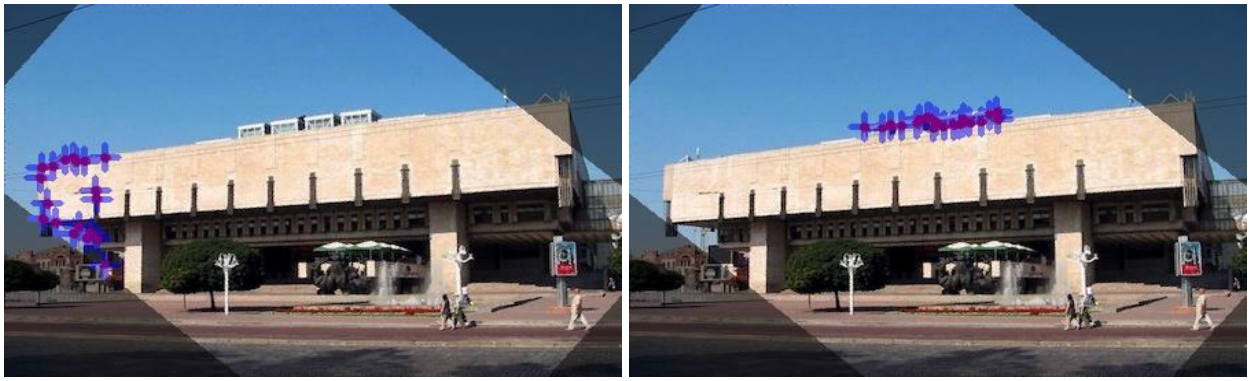


Рисунок 3.13 – Графіки оцінки точності Δpixel при α ∈ [0, 90] для одного зображення

SURF(128) SURF(64)

SIFT BRISK



ORB(1000) ORB

AKAZE KAZE

Рисунок 3.14 – Зіставлення характерних точок

Для мінімізації спотворення даних експеримент був проведений на 50 тестових зображеннях на діапазоні [0, 90] з кроком у 5 градусів та було побудована таблиця 3.5 з оцінками точності параметрів.

Таблиця 3.5 – Усереднені параметри точності Δparam та Δpixel для аналізованих дескрипторів за наявністю перетворення повороту, де $\alpha \in [0, 90]$

α	Точність	SURF 128	SURF 64	SIFT	ORB 1000	ORB	KAZE	AKAZE	BRISK
5.0	Δparam	0,0012	0,0016	0,0008	0,0022	0,0006	0,0022	0,0012	0,001
	Δpixel	0,9205	1,0528	0,3517	1,3584	0,8924	0,9333	0,4595	0,8363
10.0	Δparam	0,0023	0,0027	0,0007	0,0025	0,001	0,0021	0,0014	0,0011
	Δpixel	1,3625	1,5341	0,3306	1,3853	0,9855	1,005	0,5148	0,8971
15.0	Δparam	0,0043	0,0032	0,0009	0,0031	0,0009	0,0024	0,0016	0,0015
	Δpixel	2,0657	1,8238	0,3861	1,5607	0,9582	1,1019	0,6145	0,9543
20.0	Δparam	0,0042	0,0045	0,0009	0,0029	0,001	0,0042	0,0027	0,0013
	Δpixel	2,1018	2,1568	0,3559	1,459	0,9757	1,4388	0,919	0,9146
25.0	Δparam	0,0042	0,0038	0,001	0,004	0,001	0,0031	0,0016	0,0014
	Δpixel	2,1236	2,0445	0,4176	1,8022	0,968	1,297	0,6538	0,9488
30.0	Δparam	0,0057	0,0042	0,001	0,0039	0,0011	0,0039	0,0021	0,0015
	Δpixel	2,538	2,2879	0,4057	1,6467	1,0017	1,4348	0,7772	1,0104
35.0	Δparam	0,0061	0,005	0,0014	0,0036	0,0011	0,0035	0,0022	0,0011
	Δpixel	2,6686	2,323	0,4883	1,6463	1,01	1,3439	0,8343	0,9458
40.0	Δparam	0,0129	0,0044	0,0008	0,004	0,0017	0,0027	0,0022	0,0019
	Δpixel	4,4457	2,3247	0,3189	1,6943	1,1366	1,2886	0,837	1,0537
45.0	Δparam	0,0048	0,0043	0,0007	0,0039	0,0017	0,0029	0,0029	0,0019
	Δpixel	2,2875	2,0929	0,3203	1,7044	1,1281	1,1664	1,0091	1,1076
50.0	Δparam	0,0062	0,005	0,0009	0,0047	0,0013	0,0031	0,0023	0,0015
	Δpixel	2,6538	2,4691	0,3729	1,8772	1,0739	1,225	0,8432	1,008
55.0	Δparam	0,0111	0,0047	0,001	0,0049	0,0014	0,0048	0,0029	0,002
	Δpixel	4,1146	2,3455	0,397	1,8552	1,103	1,5845	1,0251	1,0966
60.0	Δparam	0,0042	0,0044	0,0008	0,0058	0,0015	0,0042	0,0028	0,0015
	Δpixel	2,1159	2,3414	0,3713	2,2465	1,1139	1,5633	0,9479	0,9785
65.0	Δparam	0,0049	0,0062	0,0009	0,0044	0,0015	0,0035	0,0023	0,0023
	Δpixel	2,4931	2,5894	0,3705	1,8472	1,1412	1,2503	0,9179	1,2246
70.0	Δparam	0,0056	0,0045	0,0009	0,0052	0,0014	0,0035	0,0024	0,0025
	Δpixel	2,6432	2,2442	0,4313	2,2335	1,1425	1,2831	0,9717	1,3796
75.0	Δparam	0,0046	0,0038	0,0011	0,0049	0,0017	0,0027	0,0028	0,0021
	Δpixel	2,249	2,0432	0,5186	2,1695	1,209	1,1986	0,9646	1,1068
80.0	Δparam	0,003	0,0033	0,001	0,0051	0,0018	0,0025	0,0027	0,0021
	Δpixel	1,5072	1,7294	0,4009	2,3524	1,2745	1,025	0,7708	1,1524
85.0	Δparam	0,0018	0,0025	0,0011	0,005	0,0021	0,0025	0,002	0,0014
	Δpixel	1,1099	1,3906	0,4021	2,1936	1,3581	0,9897	0,7461	0,9808
90.0	Δparam	0,0007	0,0008	0,0004	0,0054	0,0014	0,0018	0,0018	0,001
	Δpixel	0,371	0,3955	0,1694	2,4527	1,0661	0,6482	0,6205	0,6789

На основі таблиці 3.5 були побудовані графіки (рис. 3.15-3.18) для відображення поведінки дескрипторів відповідно до заданого кута повороту зображення.

В результаті експериментів на 50 зображеннях в умовах повороту навколо центру в діапазоні від 5 до 90 з кроком 5 градусів для усередненого відхилення параметра вихідного перетворення від параметра нормалізації алгоритми SIFT, ORB, BRISK, AKAZE, KAZE, ORB(1000), SURF(64) показали дуже точні результати, бо Δparam не перевищував 0.006, причому:

– метод SIFT показує найкращий результат (Δparam < 0.0015), тобто параметр наближається до нуля на всьому діапазоні значень коефіцієнтів поворотів;

– методи BRISK та ORB поводився теж стабільно і Δparam не перевищувало 0,0025 на всьому діапазоні зміни коефіцієнта масштабу;

– методи KAZE, AKAZE не перевищував 0.0029;

– методи ORB(1000), SURF(64) – 0.006;

Найбільш нестабільно себе показав алгоритм SURF (128), який має викиди до 0.0129 при значеннях 40 та 55 градусів.

Результати експериментів для середньої відстані між характерними точками на оригінальному зображенні і після нормалізації Δpixel:

1. Найбільш точні результати показали методи SIFT, BRISK, ORB, ORB(1000) максимальне середня відстань Δpixel в розглянутому діапазоні не перевищило 2.35 пікселів, причому:

– методи SIFT на всьому діапазоні показав саме низьке відхилення: менше 1 пікселя (<0,52); при цьому, методи SURF на діапазонах від [0,5] та [85,90] має низьке відхилення до 1 пікселя;

– метод AKAZE має відхилення до 1.03 пікселя; методи ORB та BRISK на всьому діапазоні показали відхилення менше ніж 1.37 пікселя;

– методи KAZE та ORB(1000) показали відхилення менше 2 пікселів;

2. Методи SURF(128) та SURF(64) мають істотні викиди величини Δpixel у середині розглянутого діапазону зміни коефіцієнта повороту, так:

– у діапазоні [40, 55] метод SURF(128) величина Δpixel досягає відхилення 4,4457;

– метод SURF(64) на діапазоні [20, 70] показує відхилення до 2.6 пікселя.

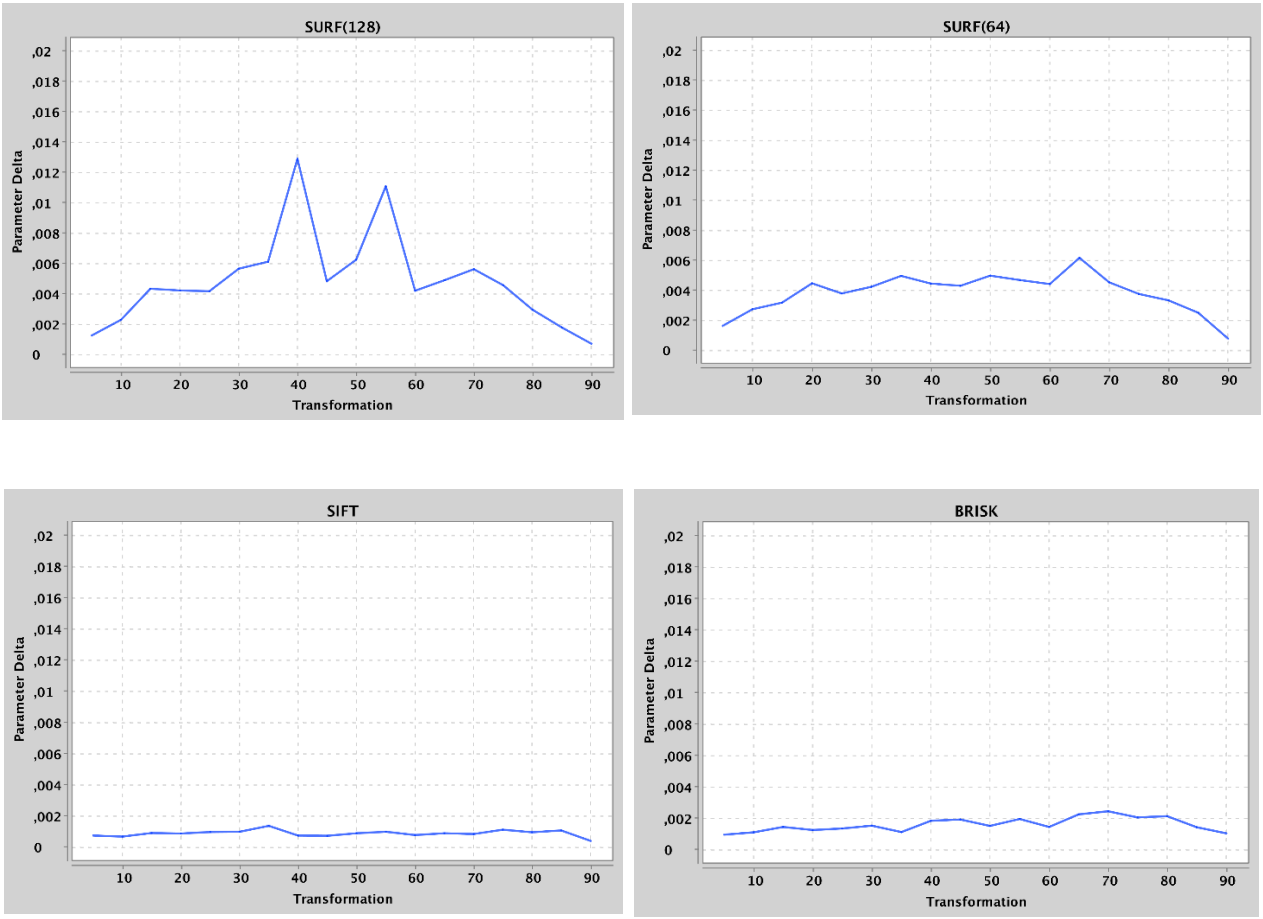


Рисунок 3.15 – Графіки оцінки точності Δparam для 50 зображень при $\alpha \in [0, 90]$ для дескрипторів SURF, SIFT, BRISK

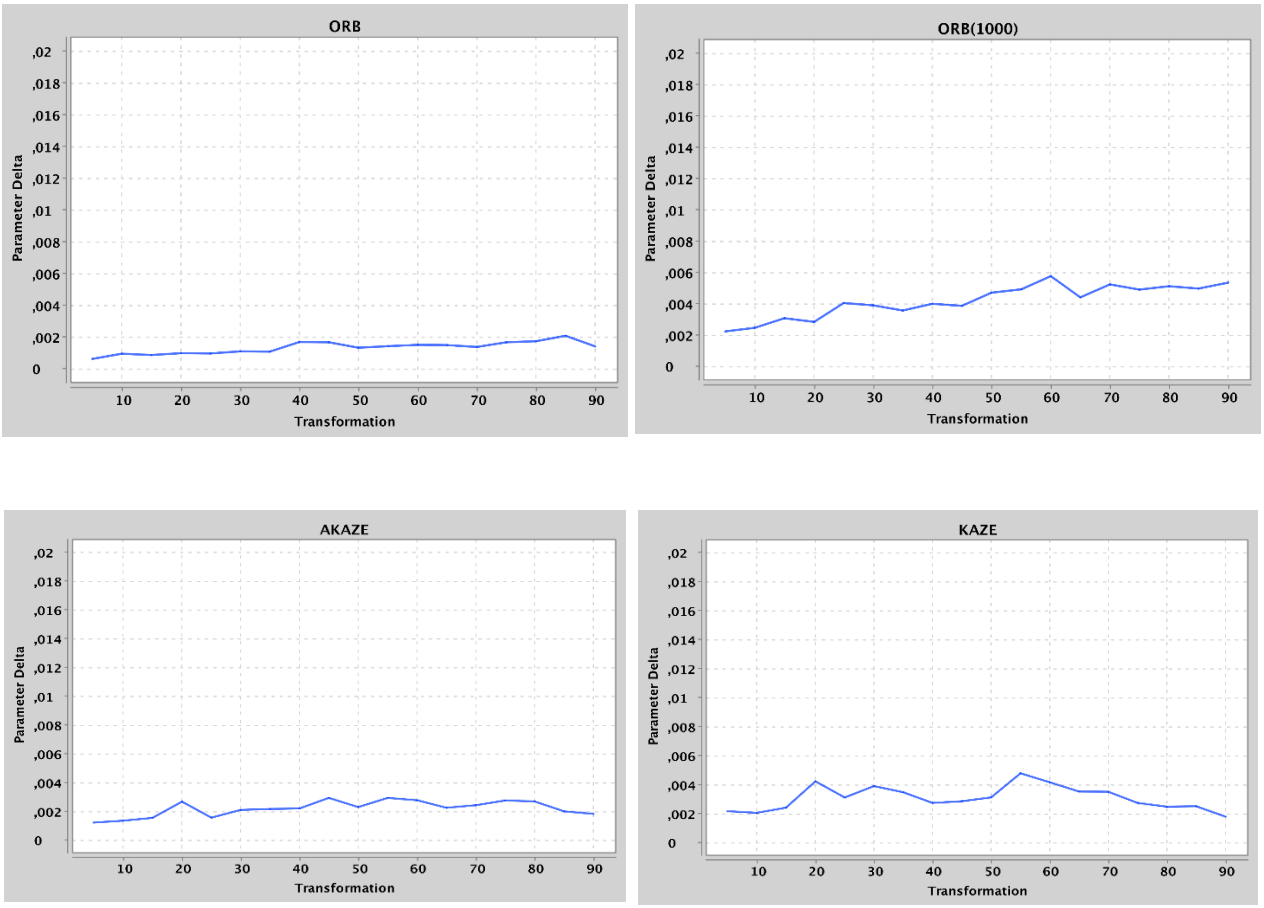


Рисунок 3.16 – Графіки оцінки точності Δparam для 50 зображень,
при α ∈ [0, 90] для дескрипторів ORB, KAZE, AKAZE

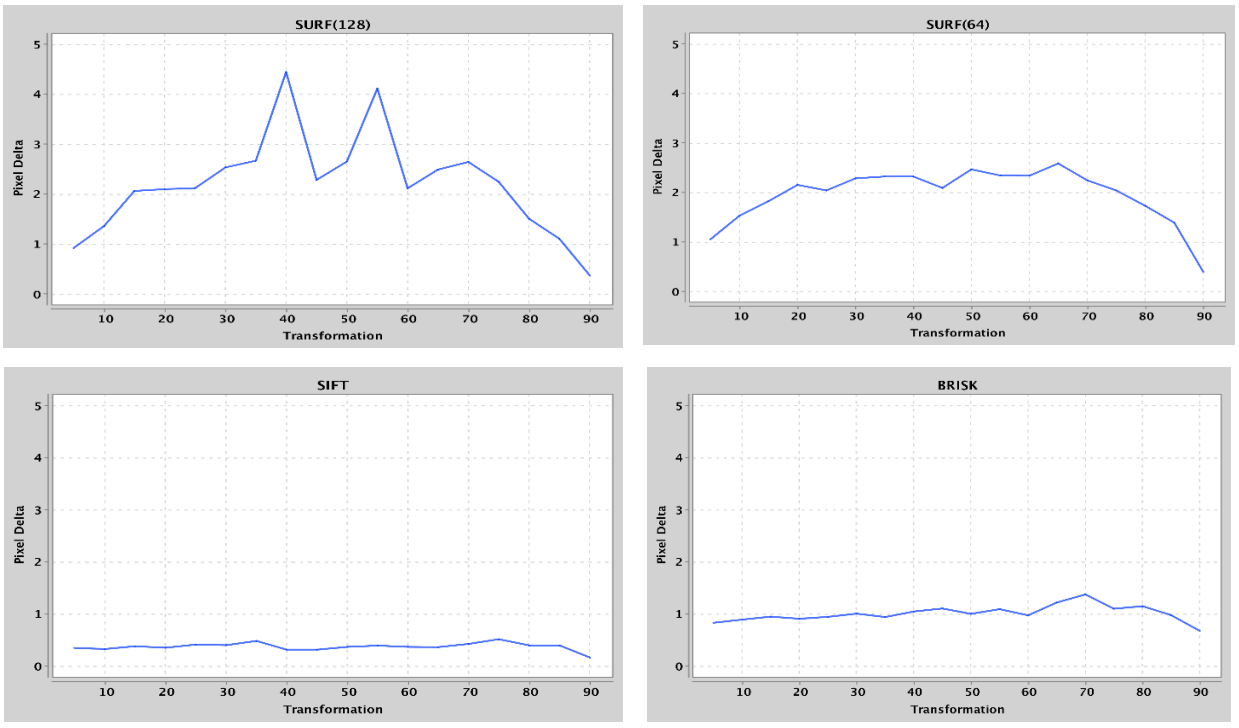


Рисунок 3.17 – Графіки оцінки точності Δpixel для 50 зображень,
при α ∈ [0, 90] для дескрипторів SURF, SIFT, BRISK

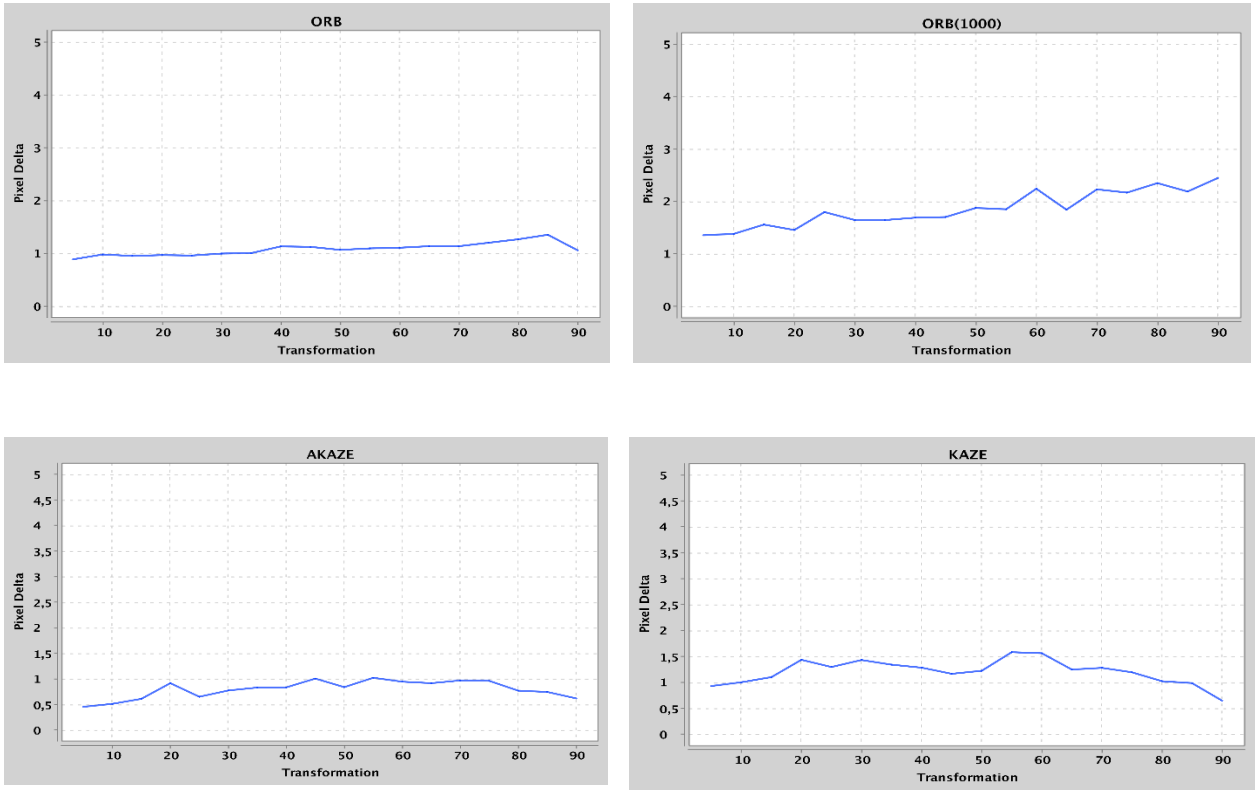


Рисунок 3.18 – Графіки оцінки точності Δpixel для 50 зображень при α ∈ [0, 90] для дескрипторів ORB, KAZE, AKAZE

Отже, можна зробити висновок, що в розглянутому діапазоні повороті зображення [0; 90] методи SIFT, ORB, BRISK, ORB (1000), KAZE, AKAZE показують більш високу точність нормалізації, ніж методи SURF (128), SURF(64). Найвищу точність дає метод SIFT: на всьому діапазоні середня відстань між вихідними і нормалізованими точками менше 0.5 пікселя.

Таким чином, проведені дослідження точності нормалізації за наявністю повороту дозволяють розташувати методи у такій послідовності (починаючи з найкращого у сенсі найменших відхилень): SIFT < ORB, BRISK, AKAZE < KAZE< ORB (1000) < SURF (64) < SURF(128).

Метод SURF рекомендується застосовувати у випадках, коли кут повороту належить до одного з діапазонів 0 до 5 та 85 до 90. В інших випадках рекомендується використовувати дескриптори SURF(128) та SURF(64) з обережністю, оскільки існують відхилення 2 і більше пікселів, але менш ніж 4.4 та 2.6 пікселя для SURF(128) та SURF(64) відповідно, що є припустимим для багатьох задач.

Тим самим, усі алгоритми показали достатню якість у визначенні параметрів перетворення та зіставлення зображень та можуть бути використані для нормалізації реальних зображень.

3.3.3 Дослідження нормалізації за наявності центроафінних перетворень

Для дослідження нормалізації за наявності центроафінного перетворення було використано розкладення матриці центрафінного перетворення на суперпозицію своїх підгруп (неоднорідний масштаб; косий зсув по осі X; поворот):

$$h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

де b – коефіцієнт масштабу по осі Y в діапазоні [0.6, 2] з кроком 0.2, h_x – коефіцієнт косого зсуву в діапазоні [0, 0.6] з кроком 0.1, α – кут повороту зображення в діапазоні [0, 90] з кроком 10 градусів.

Проаналізувавши 50 зображень, було сформовано графіки залежності параметрів оцінки точності нормалізації Δparam (3.1), Δpixel (3.2) від параметрів трансформації зображення b, h_x й α (додаток Г).

В ході експериментів було визначено, що при Δparam≤0.003 значення Δpixel становить близько 1 пікселя. Якщо значення Δparam досягає 0.005, це приводить до збільшення Δpixel до 2-х пікселів. Значення Δparam близько до 0,0075 приводить до збільшення Δpixel до 3-х пікселів. Яке значення Δpixel припустимо на практиці залежить від розміру зображення та кінцевої задачі. Для тестових зображень, наведених у додатку А, відхилення Δpixel у 3 пікселя було візуально не помітно.

Спочатку розглянемо окремий випадок, коли неоднорідний масштаб та поворот відсутні (b=1.0, α=0) (рис. Г.3-4), тобто присутній тільки косий зсув $h_x \in [0, 0.6]$. Експерименти показали, якщо $h_x \in [0, 0.4]$, метод SIFT має найменші відхилення Δparam≤0,002, Δpixel≤1, Методи BRISK, ORB мають такі невеличкі відхилення, коли $h_x \in [0, 0.2]$, а метод AKAZE при $h_x \in [0, 0.3]$. Далі при збільшені h_x до 0.6 метод SIFT показує відхилення Δparam до 0,005, Δpixel до 2 пікселів, за методом BRISK Δpixel збільшується до 4 пікселів, за методом ORB Δpixel збільшується до 7 пікселів, за методом AKAZE - до 9 пікселів, але ще при h_x=0.3 відхилення Δpixel=2,5. Інші методи KAZE, ORB(1000), SURF(128), SURF(64) показують відхилення Δpixel≤2, коли $h_x \leq 0.2$. Тобто можна зробити такий висновок, що $h_x \leq 0.2$ можно використовувати всі методи. Якщо $h_x \in [0.3, 0.6]$ найкращі результати показав SIFT (Δpixel<2), потім BRISK, потім ORB. Якщо $h_x \in [0.3, 0.6]$ найкращі результати показав SIFT (Δpixel<2), потім BRISK.

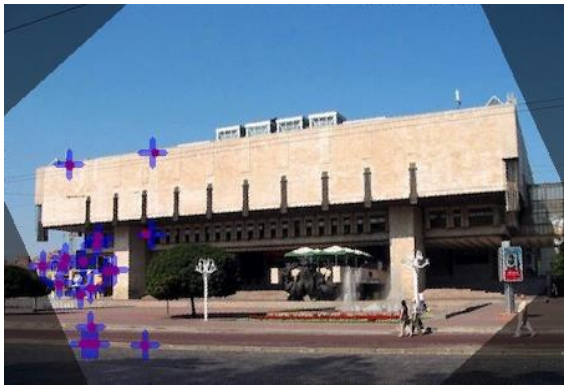
Якщо $h_x \geq 0.5$ інші методи застосовувати не рекомендується, бо відхилення становлять більше 4 пікселів. Тобто, за наявності косого зсуву методи можна розташувати так (починаючи від найточнішого): SIFT <BRISK <AKAZE <ORB, KAZE <ORB(1000), SURF(128), SURF(64).

Далі розглянемо випадок, коли присутні косий зсув та поворот (неоднорідний масштаб відсутній (b=1.0) , а h_x – коефіцієнт косого зсуву змінюється в діапазоні [0, 0.6] з кроком 0.1, α – кут повороту зображення в змінюється діапазоні [0, 90] з кроком 10 градусів) (рис. Г.3-4). У цьому випадку знову найкращі результати показав метод SIFT, далі BRISK, далі схожі результати показали методи ORB, AKAZE. Методи SURF(128), SURF(64), як вже згадувалось вище, чутливі до повороту та косого зсуву, отже їх рекомендується застосовувати, якщо $h_x \leq 0.2$, $\alpha \in [0, 10]$ и $\alpha \in [80, 90]$. Методи KAZE и ORB(1000), як зазначалось вище, чутливі до косого зсуву та показують схожі результати (KAZE має переваги, але несуттєві): Δparam до 0,0055, Δpixel до 2,5 при $h_x \leq 0.2$.

Далі було проведено дослідження в умовах центроафінних перетворень: коефіцієнт b неоднорідного масштабу по осі Y змінювався в діапазоні [0.6, 2] з кроком 0.2, а h_x – коефіцієнт косого зсуву змінювався в діапазоні [0, 0.6] з кроком 0.1, α – кут повороту зображення в діапазоні [0, 90] з кроком 10 градусів (додаток Г).

На рисунку 3.21 відображено приклади зіставлення зображень для методів SIFT та AKAZE для параметрів перетворення b=0.8, h_x=0.6, α=40.

а)



б)

в)

Рисунок 3.21 – Зіставлення ключових точок зображень:
а - б – методом SIFT, в – методом AKAZE

В результаті експериментів було отримано, що при всіх варіантах центроафінних перетворень найкращі результати показам метод SIFT, далі метод BRISK, потім метод ORB, але використання навіть цих найкращих методів обмежено. Для того, щоб відхилення не перевищувало 1.5 пікселів метод SIFT рекомендується використовувати за такому поєднанні параметрів: $b \in [0.8,\ 1.2]$, $h_x \in [0, 0.2]$, $\alpha \in [0, 90]$. Методи BRISK, ORB при такому же поєднання параметрів дають відхилення до 2,5 пікселів, при цьому відхилення методу BRISK мають трохи менші значення ніж відхилення методу ORB, але існують частіше незначні викиди. Метод AKAZE вже має відхилення до 3.3 пікселів, метод KAZE - до 4.7, ORB(1000), SURF(128), SURF(64) у середньому мають відхилення близько 4-5 пікселів, але існують великі викиди (до 7-10 пікселів). При іншому поєднанні параметрів b, h_x, α найкращий результат показує метод SIFT, але відхилення у крайніх значеннях параметрів становлять до 10 пікселів.

Методи SURF(128), SURF(64) чуттєві до повороту, тобто їх використання обмежено невеликим кутом нахилу до 5 градусів. При поєднання невеликого повороту, косого зсуву до 0.2, неоднорідного масштабу від 0.8 до 1.2 методи SURF(128), SURF(64) показують відхилення до 3-х пікселів.

Таким чином, експерименти за наявності центроафінних перетворень показали, що розглянуті методи за величиною параметрів Δparam, Δpixel методи можна розташувати у такої послідовності (починаючи від найкращого): SIFT < BRISK < ORB < AKAZE,KAZE,ORB(1000),SURF(128), SURF(64).

Отже, результати дослідження точності методів на основі аналізу параметрів Δparam, Δpixel в умовах окремих груп геометричних перетворень зведені у таблицю 3.6.

Отже, після проведення експериментів за наявності масштабу, повороту, косого зсуву, центроафінного перетворення можна зробити висновок, що найкращі за точністю результати за наявності перетворень із різних геометричних груп дає метод SIFT, на другому місці стоїть метод BRISK, далі ORB, тобто ці методи рекомендовано застосовувати на практиці до реальних зображень, якщо не відома група геометричних перетворень.

Таблиця 3.6 – Результати дослідження точності методів на основі аналізу параметрів Δparam, Δpixel

Геометричне перетворення	Розташування методів за точністю значень Δparam, Δpixel (починаючи від найкращого)
Масштаб	SIFT < ORB,BRISK< SURF (64), SURF(128)< ORB(1000),AKAZE,KAZE
Поворот	SIFT< ORB,BRISK,AKAZE<KAZE<ORB (1000)< SURF(64)< SURF(128)
Косий зсув	SIFT< BRISK< AKAZE< ORB,KAZE< ORB(1000),SURF(128),SURF(64)
Центроафінне перетворення	SIFT< BRISK< ORB< AKAZE, KAZE, ORB(1000), SURF(128),SURF(64)

3.3.4 Дослідження нормалізації реальних зображень

Для дослідження на реальних зображеннях була зроблена серія знімків й також використана колекція зображень які представляє науковий сайт університету Oxford (3.22), на основі яких проведені досліди на зіставлення.

а) б)

в)

Рисунок 3.22 – Пари реальних зображень:
а – з власної колекції; б, в – з колекції Oxford

На рисунках 3.23-3.25 проілюстровані результати експериментів над зображеннями у порядку зображення їх у рисунку 3.22.

Для другої пари зображень (рис. 3.22 - б) можна припустити, що вхідне зображення в порівнянні з еталонним має невелике перетворення повороту та масштабу з коефіцієнтами $k < 1.2$ та $\alpha < 20$.

Тим чином, як й зображено у висновках розділу 3.2.2, за наявністю невеликих значеннях параметрів масштабу та повороту усі методи відпрацювали з точним результатом.

Рисунок 3.23 – Зіставлення першої пари зображень з власної колекції

При експерименті з першою парою зображень (рис. 3.22 - а) можна припустити, що на зображенні присутнє перетворення однорідного масштабу з коефіцієнтом k, який приблизно дорівнює 1.8.

В ході експерименту усі алгоритми показали велику точність нормалізації, що підтверджує висновки розділу 3.2.1, зробленого за результатами експериментів над синтезованими зображеннями.

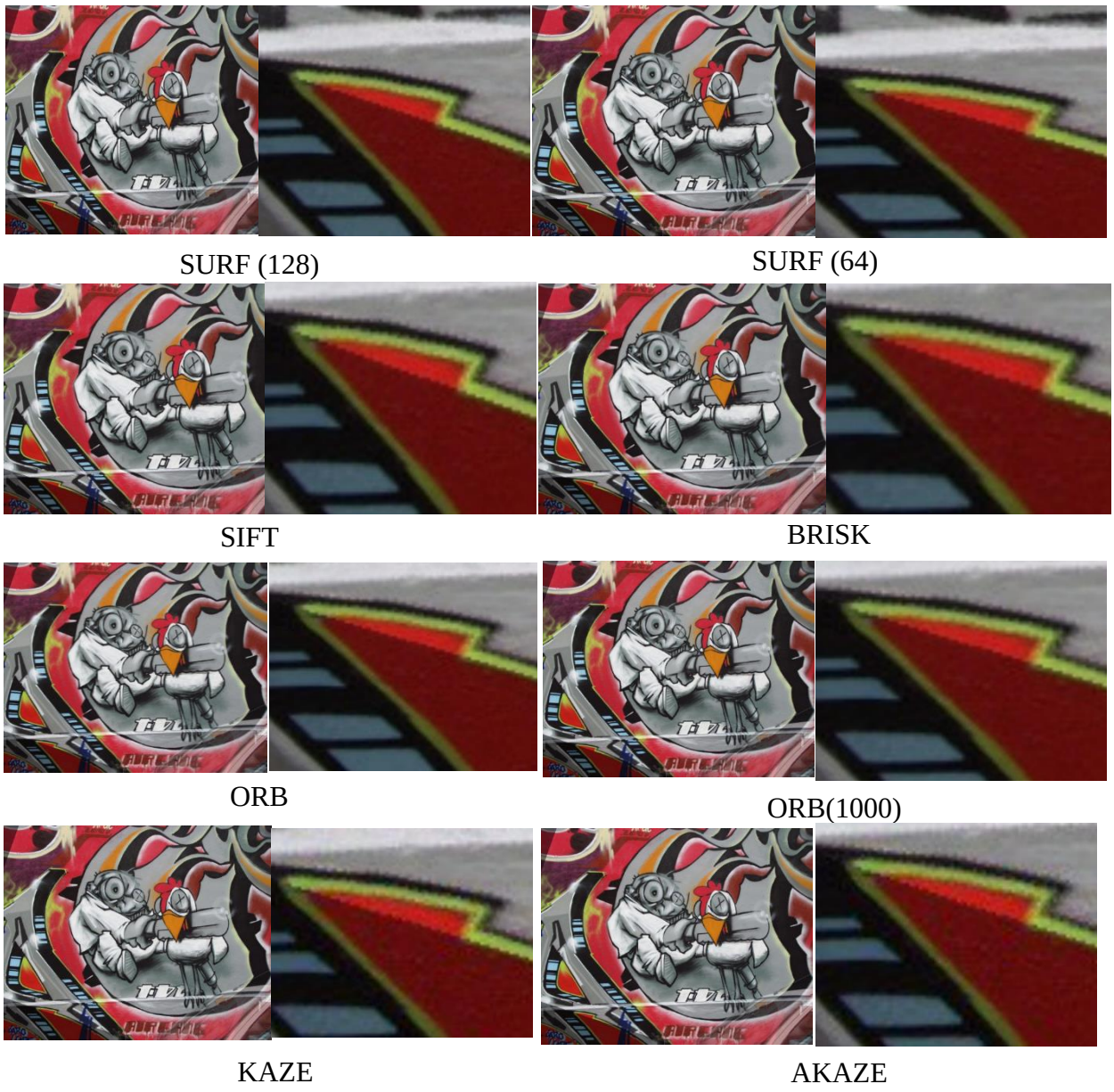


Рисунок 3.24 – Зіставлення другої пари зображень

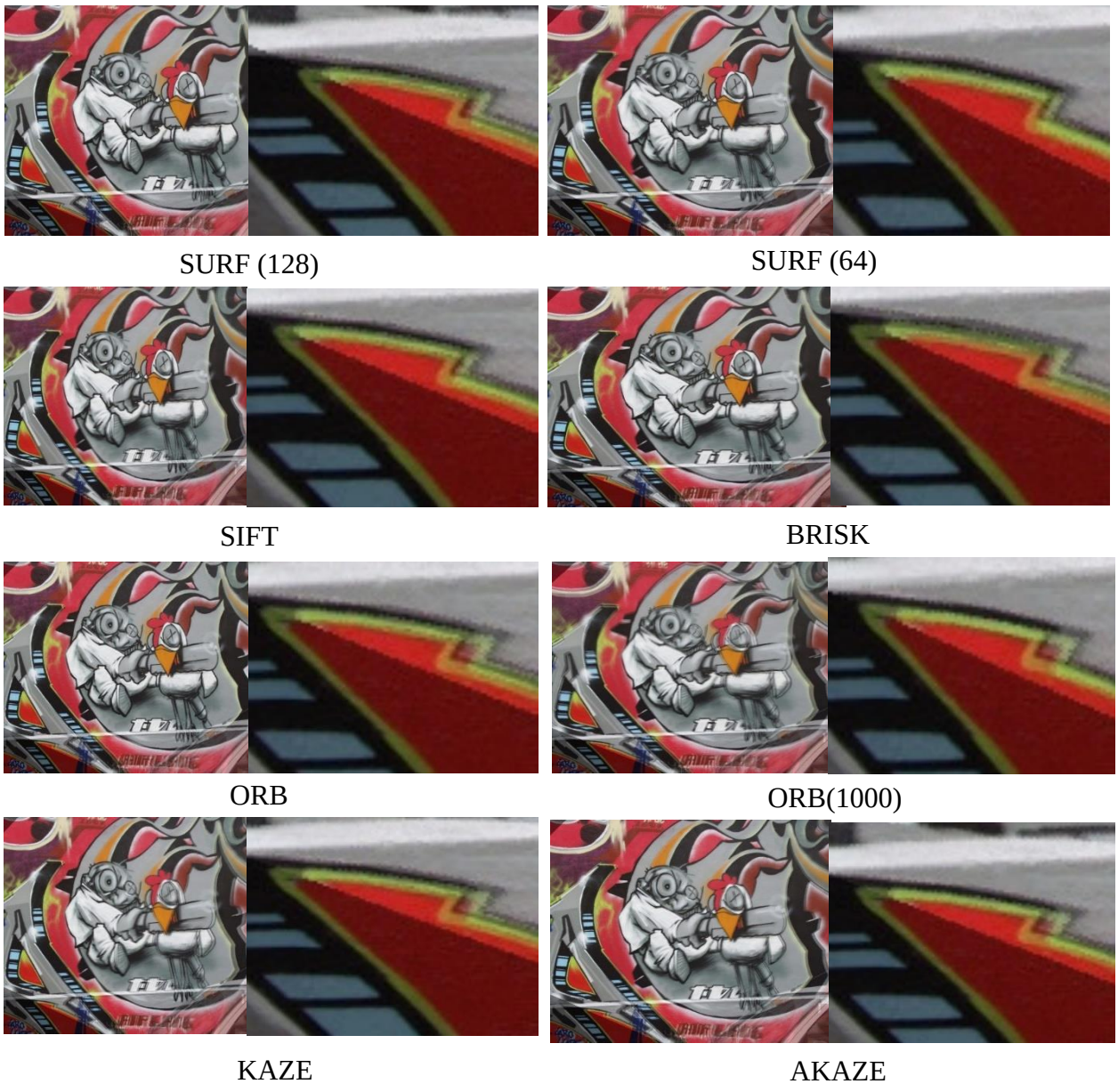


Рисунок 3.25 – Зіставлення третьої пари зображень

При порівнянні третьої пари зображень (рис. 3.22 - в) можна припустити наступні геометричні перетворення: неоднорідний масштаб по осі Y приблизно 0.8, поворот менше ніж 15 градусів, й навіть наявність проективних перетворень. За результатами експерименту можна зробити висновки, що при наявності серйозних геометричних перетворень усі методи показують не ідеальний результат, тим самим об’єднуючи результати зіставлення зображень на третій парі та висновки, які були зроблені у розділі 3.2.3, найкращий методами у даному прикладі є SIFT та BRISK.

Як і показано в розділі 3.3.3, за наявності істотних афінних і проективних перетворень нормалізація на основі аналізу дескрипторів не достатня точна (рис. Д.5).

Але якщо використовувати нормалізацію, як попередній етап для вирішення інших задач аналізу зображень, наприклад, розпізнавання або пошуку, то навіть при суттєвих

геометричних перетвореннях нормалізоване зображення стає істотно ближче до зображення-зразка, що підвищить якість вирішення завдань розпізнавання або пошуку.

3.4 Вивчення питання швидкодії нормалізації на основі дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK

3.4.1 Опис умов та задач експериментів щодо дослідження швидкодії

Для дослідження швидкодії дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK на вхід було прийнято еталонне зображення, коефіцієнт для однорідного масштабування k в діапазоні [0.5, 2] та згенеровано синтезоване зображення для кожного з коефіцієнтів масштабування. При цьому, перед тим як отримати параметри часу, кожен алгоритм з вхідними параметрами додатково був запущений 50 циклів для мінімізації комп'ютерної помилки та нормовано по кількості циклів.

Щоб аналізувати швидкодію було розраховано чотири параметри та для кожного з них підрахований час виконання. Для відображення кількості точок, введено такі параметри: $Img1_f$ – кількість знайдених характерних точок на першому зображенні, $Img2_f$ – кількість точок на другому зображенні, M_f – кількість відповідностей, отриманих після методу зіставлення, kNN_f – кількість відповідностей після виконання методу k-NN, R_f – кількість відповідностей після виконання методу RANSAC; відповідні їм параметри часу: $Img1_t$, $Img2_t$, M_t, kNN_t, R_t.

Після отримання усіх даних з кожного алгоритму по заданим 15-ти параметрам масштабу k розраховується час обробки одної ключової точки $Img1'_t$, $Img2'_t$, M'_t, kNN'_t, R'_t кожного з параметрів $Img1_t$, $Img2_t$, M_t, kNN_t, R_t:

$$Img1' = \frac{Img1_t}{Img1_f}; Img2' = \frac{Img2_t}{Img2_f}; M' = \frac{M_t}{M_f}; kNN' = \frac{kNN_t}{kNN_f}; R' = \frac{R_t}{R_f}.$$

Для отримання результуючої таблиці кожен з параметрів $Img1'_t$, $Img2'_t$, M'_t, kNN'_t, R'_t нормалізується відносно кількості проведених перетворень N=15.

3.4.2 Результати експериментів та висновки щодо швидкодії пошуку параметрів нормалізації

В ході експерименту щодо швидкодії пошуку параметрів нормалізації були побудовані ряд таблиць (табл. 3.7 – 3.8) для їх усереднення та отримання результуючої таблиці часу розрахунку за одну ключові точку (табл. 3.9).

Таблиця 3.7 – Отримані кількісні параметри для k=0.5

Назва	$Img1_f$	$Img2_f$	M_f	kNT_f	R_f	$Img1_t$, сек	$Img2_t$, сек	M_f, сек	kNN_t, сек	R_t, сек	Загальний час, сек
SIFT	206	110	206	76	67	0,0528	0,0471	0,0081	0,0002	0,0024	0,1105
SURF(128)	394	250	394	129	122	0,0535	0,0437	0,0135	0,0007	0,0007	0,1121
SURF(64)	394	250	394	139	127	0,0495	0,0417	0,0026	0,0001	0,0006	0,0945
ORB(1000)	870	874	870	250	243	0,0068	0,0059	0,0067	0,0003	0,0009	0,0205
ORB	871	874	871	250	243	0,0076	0,006	0,0093	0,0003	0,0012	0,0244
KAZE	168	131	168	34	33	0,2089	0,2091	0,0012	0,0001	0,0005	0,4198
AKAZE	161	139	161	52	51	0,0259	0,024	0,0006	0	0,0004	0,0511
BRISK	261	292	261	93	92	0,0092	0,0081	0,0011	0,0001	0,0005	0,0191

Таблиця 3.8 – Отримані кількісні параметри для k=2.0

Назва	$Img1_f$	$Img2_f$	M_f	kNN_f	R_f	$Img1_t$, сек	$Img2_t$, сек	M_f, сек	kNN_t, сек	R_t, сек	Загальний час, сек
SIFT	206	160	206	57	52	0,0514	0,0359	0,0016	0,00001	0,0008	0,0897
SURF(128)	394	357	394	75	70	0,0475	0,04	0,007	0,0001	0,0004	0,0949
SURF(64)	394	357	394	82	75	0,0443	0,0366	0,0031	0,0001	0,0004	0,0844
ORB(1000)	870	142	870	72	62	0,005	0,003	0,0021	0,0001	0,0004	0,0107
ORB	871	142	871	72	64	0,0056	0,0031	0,0021	0,0002	0,0004	0,0114
KAZE	168	101	168	21	13	0,1698	0,1539	0,0011	0,00001	0,0007	0,3255
AKAZE	161	54	161	23	20	0,0245	0,0223	0,0004	0,00001	0,0004	0,0477
BRISK	261	46	261	30	26	0,0071	0,0037	0,0006	0,00001	0,0003	0,0117

Таблиця 3.9 – Результуючий час виконання за одну ключову точку

Назва	Img1′$_t$, мкс	Img2′$_t$, мкс	M′$_t$, мкс	kNN′$_t$, мкс	R′$_t$, мкс
SIFT	244,030141	208,24554	13,737906	0,588449	7,480607
KAZE	1104,743008	1212,983609	9,600341	0,911238	15,288165
SURF(128)	136,940441	123,779191	20,688066	1,013253	3,723593
ORB(1000)	6,357703	9,965047	7,627997	0,972632	3,729259
AKAZE	170,749228	224,151566	4,108976	1,070238	11,055972
SURF(64)	127,418744	121,958063	10,341488	0,554862	3,587998
BRISK	30,676016	44,82015	3,895771	0,886459	8,38787
ORB	6,933812	9,753324	6,985975	1,140168	3,641543

Результати експерименту щодо швидкодії пошуку характерних точок:

– бінарні дескриптори ORB, ORB(1000), BRISK мають найкращу загальну швидкодію процесу нормалізації;

– ORB та ORB(1000) знаходять найбільшу кількість характерних точок (більше ніж в 2 рази за інші дескриптори), але мають суттєво менший час ніж інші дескриптори для виявлення точок та обчислення їх дескрипторів. В результаті, не зважаючи на найбільшу кількість точок, показують найкращу загальну швидкодію (порівнянну з BRISK). Дескриптор ORB(1000) обчислюється швидше ніж ORB, але ця різниця не суттєва;

– BRISK показав загальну швидкодію порівнянну з ORB, ORB(1000), має низьке значення часу на обчислення дескриптору, але у цьому поступається ORB, ORB(1000), також метод BRISK знаходить точок менш ніж ORB, ORB(1000);

– бінарний дескриптор AKAZE має наступну загальну швидкодію після ORB, ORB(1000), BRISK, але середній час на обчислення одного дескриптору значно поступається ORB, ORB(1000), BRISK, трохи SURF(64), SURF(128), SIFT, проте кращий ніж у KAZE;

– наступну загальну швидкодію показали числові дескриптори SURF(64), SIFT, SURF(128) (загальна швидкодія схожа). Дескриптори SURF(64), SURF(128) стоять на другому місці по кількості знайдених точок, поступаючись у цьому питанні тільки детекторам ORB та ORB(1000) (приблизно у 2 рази). Дескриптор SURF(64) виявився швидшим ніж SURF(128), але ця різниця не суттєва. Дескриптор SIFT знаходить менш характерних точок ніж SURF(64), SURF(128) (приблизно 1,5 рази), але середній час на обчислення одного дескриптору більше ніж у дескрипторів SURF(64), SURF(128);

– найдовше з всіх методів відпрацьовував KAZE, від суттєво відстає по всіх показниках (за загальною швидкодією та середнім часом на обчислення одного дескриптору KAZE відстає від одного з найкращих детекторів ORB приблизно у 25 разів).

Тим самим розташування методів за швидкодією (починаючи від найкращого): ORB(1000), ORB< BRISK< SURF(64), SURF(128) < AKAZE, SIFT < KAZE.

3.5 Висновки щодо використання дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK для нормалізації зображень

У результаті аналізу дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK було виведено синтезовані зображення з перетворенням однорідного масштабу в діапазоні [0.5, 2], з поворотом при діапазоні [0, 90] та з центроафінними перетвореннями при наявності двох останніх та косого зсуву при діапазоні [0, 1].

Перерахована комбінація перетворень була проаналізована та отримані наступні результати:

1) при наявності однорідного масштабу, де $k \in [0,5; 2]$ розташування дескрипторів у такому порядку: SIFT < ORB,BRISK< SURF (64), SURF(128)< ORB(1000),AKAZE,KAZE;

2) при наявності повороту зображення, де $\alpha \in [0; 90]$: SIFT< ORB,BRISK,AKAZE<KAZE<ORB (1000)< SURF(64)< SURF(128);

3) при наявності косого зсуву, де $h_x \in [0, 0.6]$: SIFT< BRISK< AKAZE< ORB,KAZE< ORB(1000),SURF(128),SURF(64);

4) при наявності центроафінних перетворень, де $b \in [0.6, 2]$, $h_x \in [0, 0.6]$ та $\alpha \in [0,90]$: SIFT< BRISK< ORB< AKAZE, KAZE, ORB(1000), SURF(128),SURF(64);

5) за швидкодією дескриптори розташовуються наступним чином: ORB(1000), ORB< BRISK< SURF(64), SURF(128)< AKAZE, SIFT < KAZE.

Спираючись на таблицю 3.6 і таблицю 3.9 можна зробити наступні висновки:

– за співвідношенням точність і швидкість мають перевагу дескриптори ORB і BRISK, які показують найкращу швидкодію та друге місце за точністю;

– якщо необхідно застосувати метод найкращий за точністю, рекомендовано дескриптор SIFT.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об’єкту, що мають вплив на персонал

У даному дипломному проекті розробляється програмне забезпечення.

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

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

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

Відповідно до ДСН 3.3.6.042-99 [41] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

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

Джерелами підвищеної небезпеки можуть служити наступні елементи:

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

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до НПАОП 0.00-7.15-18 [42], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

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

4.2 Заходи щодо техніки безпеки

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

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

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

– значення сили струму;
– електричного опору тіла людини і тривалості протікання через нього струму;
– роду і частоти струму;
– індивідуальних властивостей людини і навколишнього середовища.

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

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

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

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_K = \frac{\text{Uф}}{\text{Zп} + \frac{\text{Zт}}{\text{З}}}\ , \qquad (4.1)$$

де Uф - номінальна фазна напруга мережі, В;

Zп - повний опір петлі, створене фазними і нульовими дротами, Ом;

ZT - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [43]: Zт /3 = 0,1 Ом.

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

$$\text{Zп} = \sqrt{\text{Rп}^2 + \text{Xп}^2}\ , \qquad (4.2)$$

де Rп = Rф + Ro - сумарний активний опір фазного Rф і нульового Ro дротів, Ом;

Xп - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту S = 2.5 мм, тоді згідно таблицям 5 і 6 [43], має такий опір:

Xп = 0,11 Ом;

Rф = 7,55 Ом;

Ro = 7,55 Ом.

Отже, Rп = 7,55 + 7,55 = 15,1 Ом.

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Zп = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом).}$$

Струм однофазного короткого замикання рівний:

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

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$Iк \geq k * Iн , \quad (4.3)$$

де Iн - номінальний струм спрацьовування плавкої вставки, А;
k - коефіцієнт кратності нелінійного струму Iн, А.

Коефіцієнт кратності нелінійного струму Iн розраховується по формулі (4.4.) :

$$Iн = P / U , \quad (4.4)$$

де P = 220 Вт - споживана потужність;
U = 220 В - робоча напруга;
k = 3 А - для плавких вставок.

Отже, Iн = 220 / 220 = 1 А.

Підставивши значення у вираз (4.3), одержимо:

14,47> 3*1.

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

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються ДСН 3.3.6.042-99[41], ДБН, відповідними ГОСТами і ОСТамі з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

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

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

Відповідно до ДСН 3.3.6.042-99 [41] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні . За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

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

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

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

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

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні).

Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплену).

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

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

TCO'03 пред’являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [44].

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

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

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

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до ДБН В.2.5-28-2006 [47] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

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

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

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м2;

Z - поправочний коефіцієнт світильника = 1,2;

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

n - кількість світильників;

U - коефіцієнт використовування освітлювальної установки = 0,6;

M - кількість ламп у світильнику =2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k)/(F \cdot U \cdot M), \quad (4.6)$$

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

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

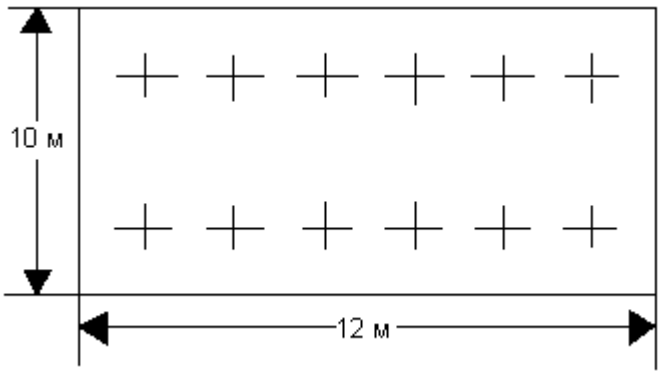


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

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

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

– поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;

– полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно ДСТУ Б В.1.1-36-2016 [48] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогненебезпечної зони класу П - IIа (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

– іскри при замиканні і розмиканні ланцюгів;

– іскри і дуги коротких замикань;

– перегріви від тривалого перевантаження і наявності перехідного опору.

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

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

Пожежна безпека об’єктів народного господарства регламентується і забезпечується системами запобігання пожежам і протипожежному захисту[48]. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

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

– застосування устаткування, що задовольняє вимогам електростатичної безпеки;

– застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;

– виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;

– підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з пальним середовищем, нижче гранично допустимої, становить 80% як найменшої температури самозаймання пального.

– заміна небезпечних технологічних операцій більш безпечними;

– ізольоване розташування небезпечних технологічних установок і устаткування;

– зменшення кількості пальних і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;

– запобігання можливості утворення пальних сумішей на лінії, вентиляційних системах і ін.;

– механізація, автоматизація та справність(потокова) виробництва;

– суворе дотримання стандартів і точне виконання встановленого технологічного режиму;

– запобігання можливості появи в небезпечних місцях джерел запалення;

– запобігання розповсюдженню пожеж і вибухів;

– використовування устаткування і пристроїв, при роботі яких не виникає джерел запалення;

– виконання вимог сумісного зберігання речовин і матеріалів;

– наявність громовідводу;

– ліквідація можливості самозаймання речовин і матеріалів.

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

– електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолоджування і кондиціонування;

– система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

– робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

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

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

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

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

4.5 Вплив на навколишнє природнє середовище

Діяльність за темою магістерської роботи в процесі її виконання впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища»[51], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення»[53], Законом України «Про відходи»[52].

В процесі діяльності виконанням дипломного проектування виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки.
- змінні носії інформації - IV клас небезпеки.
- відпрацьовані вогнегасники - IV клас небезпеки;
- макулатура - IV клас небезпеки

ВИСНОВКИ

У рамках магістерської роботи були проведені дослідження для нормалізації геометричних перетворень зображень на основі аналізу характерних точок. В ході дослідження проблематики був розглянутий такий список питань:

- огляд підходів для нормалізації геометричних перетворень;
- нормалізація зображень на основу аналізу характерних точок;
- формування дескрипторів SURF, SIFT, BRISK, ORB, KAZE та AKAZE;
- вирішення задачі усунення хибних відповідностей;
- отримання найкращої матриці геометричних перетворень та їх параметрів;
- дослідження у порівняльному аспекті точності та швидкодії методу нормалізації на основі аналізу дескрипторів SURF, SIFT, ORB, BRISK, KAZE та AKAZE, розробка програмного застосунку для проведення досліджень.

Для реалізації програми на мові Java була використана бібліотека комп'ютерного зору OpenCV. OpenCV - популярна бібліотека комп'ютерного зору. Вона написана на мові програмування C / C ++, її вихідний код відкритий, бібліотека включає більше 1000 функцій та алгоритмів. Бібліотека розповсюджується за ліцензією BSD, це означає, що її можна вільно і безкоштовно використовувати як у проектах з відкритим кодом, так і в закритих, комерційних проектах.

Після даних досліджень можна сказати, що усі дескриптори SURF, SIFT, BRISK, ORB, KAZE та AKAZE є доцільними в використанні для нормалізації зображень за наявністю геометричних перетворень на основі характерних точок, а саме: однорідного масштабу, повороту та невеликих центроафінних перетворень, але найкращу точність показали детектори SIFT, ORB, BRISK.

Також була проведена порівняльна характеристика швидкодії процесу нормалізації на основі розглянутих дескрипторів. Виявлено, що самими швидкими методами є ORB(1000), ORB, BRISK, а найдовше з всіх відпрацьовував KAZE.

Було зроблено підсумкові висновки, що за співвідношенням точність і швидкодія мають перевагу дескриптори ORB і BRISK, які показують найкращу швидкодію та друге місце за точністю, однак, якщо є потреба у методі, найкращому за точністю, рекомендовано застосовувати дескриптор SIFT. Проведені дослідження на реальних зображеннях підтвердили усі перераховані вище висновки.

Результати були апробовані на Міжнародній конференції «Майбутній науковець 2020».

У якості подальшого напрямку дослідження має сенс розглянути питання нормалізації зображень за наявністю більш значних геометричних перетворень, а також збільшення кількості зображень з 50 до 1000.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці. А також визначені основні екологічні аспекти впливу на навколишнє природнє середовище та зазначені заходи щодо поводження з ними.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) Шапиро, Л. (2006). Стокман Дж. Компьютерное зрение. М.: Бином, 120-124.

2) Форсайт, Д., & Понс, Ж. (2004). Компьютерное зрение. Современный подход (p. 928). М.: ИД Вильямс.

3) Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.

4) Шолле, Ф. (2018). Глубокое обучение на Python. СПб.: Питер.

5) Павлидис, Т. (1986). Алгоритмы машинной графики и обработки изображений. Радио и связь..

6) Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

7) Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359.

8) Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. R. (2011, November). ORB: An efficient alternative to SIFT or SURF. In ICCV(Vol. 11, No. 1, p. 2).

9) Rosten, E., & Drummond, T. (2006, May). Machine learning for high-speed corner detection. In European conference on computer vision (pp. 430-443). Springer, Berlin, Heidelberg.

10) Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In European conference on computer vision (pp. 778-792). Springer, Berlin, Heidelberg.

11) Harris, C. G., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, No. 50, pp. 10-5244).

12) Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012, October). KAZE features. In European Conference on Computer Vision (pp. 214-227). Springer, Berlin, Heidelberg.

13) Alcantarilla, P. F., & Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. IEEE Trans. Patt. Anal. Mach. Intell, 34(7), 1281-1298.

14) Weickert, J., & Scharr, H. (2002). A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. Journal of Visual Communication and Image Representation, 13(1-2), 103-118.

15) Visual Geometry Group. (2004). Image Retrieval. Retrieved from: http://www.robots.ox.ac.uk/~vgg/data.

16) Tareen, S. A. K., & Saleem, Z. (2018, March). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-10). IEEE.

17) Ересько, Ю. Н. (2002). Локализация изображений в автоматических визирах. М.: Компания Спутник+.

18) Браверман, Э. М., & Мучник, И. Б. (1983). Структурные методы обработки эмпирических данных. Наука. Гл. ред. физ.-мат. лит..

19) Воронцов, К. В. Лекции по методам отбора и синтеза признаков и моделей.

20) Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In iccv (Vol. 99, No. 2, pp. 1150-1157).

21) Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors.

22) Kim, S., & Kweon, I. S. (2006, January). Biologically motivated perceptual feature: Generalized robust invariant feature. In Asian Conference on Computer Vision (pp. 305-314). Springer, Berlin, Heidelberg.

23) Harris, C. G., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, No. 50, pp. 10-5244).

24) Schmid, C., & Mohr, R. (1997). Local grayvalue invariants for image retrieval. IEEE transactions on pattern analysis and machine intelligence, 19(5), 530-535.

25) Azimi, S., Lall, B., & Gandhi, T. K. (2019, May). Performance Evalution of 3D Keypoint Detectors and Descriptors for Plants Health Classification. In 2019 16th International Conference on Machine Vision Applications (MVA) (pp. 1-6). IEEE.

26) Koenderink, J. J., & van Doorn, A. J. (1987). Representation of local geometry in the visual system. Biological cybernetics, 55(6), 367-375.

27) Köthe, U. (2003, September). Edge and junction detection with an improved structure tensor. In Joint Pattern Recognition Symposium (pp. 25-32). Springer, Berlin, Heidelberg.

28) Wallraven, C., Caputo, B., & Graf, A. (2003, October). Recognition with local features: the kernel recipe. In null (p. 257). IEEE.

29) Gorokhovatskiy, A. V., Gorokhovatskiy, V. A., Vlasenko, A. N., & Vlasenko, N. V. (2014). Quality Criteria for Multidimensional Object Recognition Based Upon Distance Matrices. Telecommunications and Radio Engineering, 73(18).

30) Barghout, L., & Sheynin, J. (2013). Real-world scene perception and perceptual organization: Lessons from Computer Vision. Journal of Vision, 13(9), 709.

31) Федотов Н.Г. Методы стохастической геометрии в распознавании образов-М.:Радио и связь, 1990.- 144 с.

32) Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.

33) Даревич, Р. Р. (2005). Підвищення ефективності інтелектуального аналізу тексту шляхом зважування понять в моделі онтології. Искусственный интеллект, (3), 571-577.

34) Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis & Machine Intelligence, (4), 509-522.

35) Яковлева, Е. В., & Нестерова, Е. П. (2009). Сравнительный анализ мнтодов характеристик Лавса и матриц совпадений в задачах сегментации текстурных изображений.

36) Яковлева, Е. В., & Панченко, И. А. (2007). Применение энергетических характеристик Лавса для сегментации изображений.

37) Яковлева, Е. В., & Кускова, И. В. (2006). Исследование результатов сегментации изображений методом матриц совпадений.

38) LEUTENEGGER, Stefan; CHLI, Margarita; SIEGWART, Roland. BRISK: Binary robust invariant scalable keypoints. In: 2011 IEEE international conference on computer vision (ICCV). Ieee, 2011. p. 2548-2555.

39) Gorokhovatsky, V. (2014). Structural analysis and intellectual data processing in computer vision. SMIT: Kharkiv, Ukraine.

40) Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis & Machine Intelligence, (4), 509-522.

41) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99

42) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров`я працівників під час роботи з екранними пристроями. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18

43) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та

44) Номенклатура видів захисту. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/ru/v0037831-11

45) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

46) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0024858-13

47) ДБН В.2.5-28-2006. Природне і штучне освітлення. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0168667-06

48) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419

49) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0379-96

50) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

51) Закон України «Про охорону навколишнього природного середовища» . Вводиться в дію Постановою ВР № 1268-XII від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/1264-12

52) Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр

53) Закон України «Про забезпечення санітарного та епідемічного благополуччя населення». Відомості Верховної Ради України (ВВР), 1994, № 27, ст.218. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/4004-12

ДОДАТОК А.

Тестові зображення

Рисунок А.1 – Приклади тестових зображень збірки Oxford

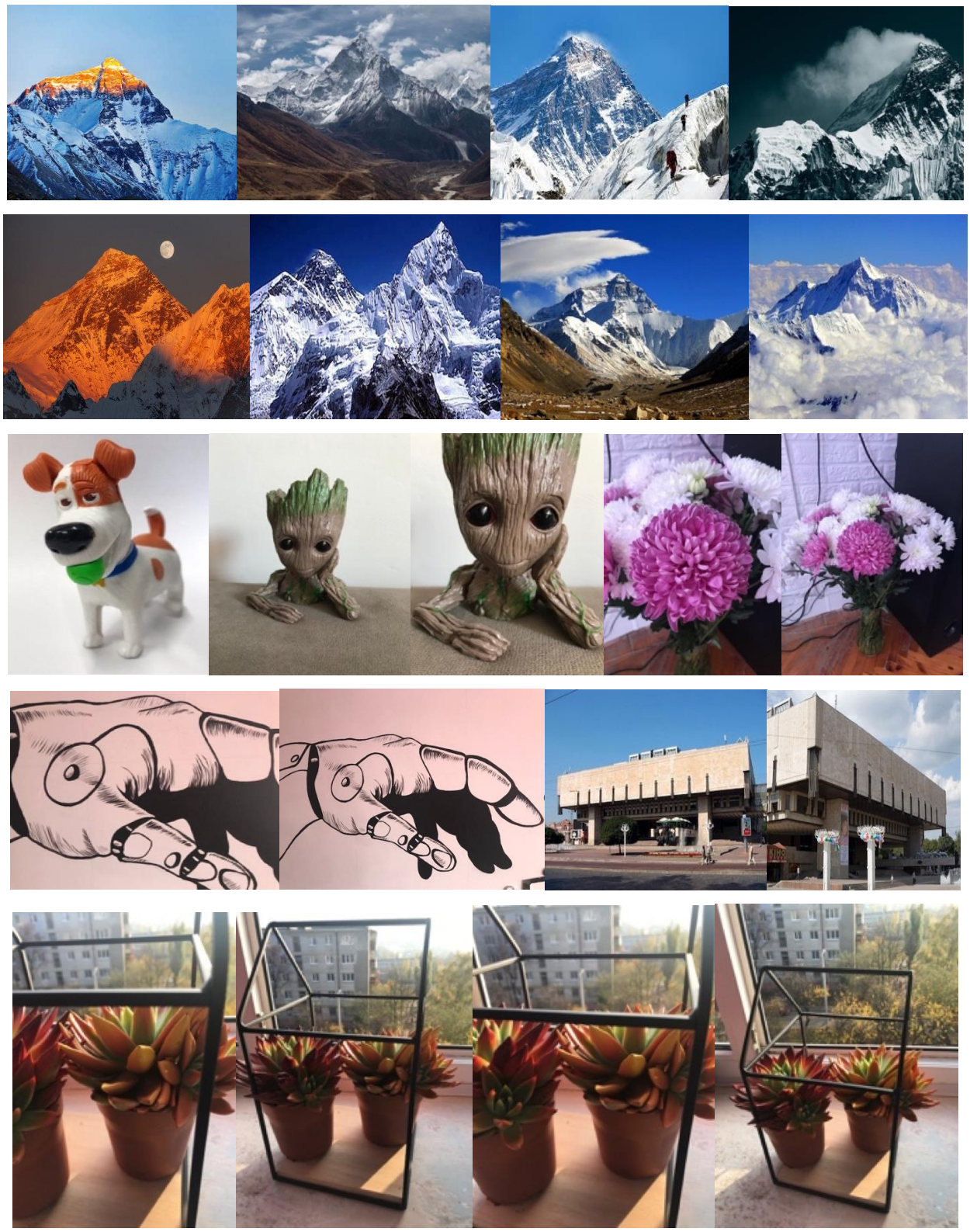


Рисунок А.2 – Приклади тестових зображень власної збірки

ДОДАТОК Б.

Фрагмент лістингу

```
package ijopencv.descript;

import net.imagej.ImageJ;
import ij.ImagePlus;
import ij.gui.GenericDialog;
import ij.gui.PointRoi;
import ij.plugin.filter.PlugInFilter;
import ij.process.ImageProcessor;
import ijopencv.ij.ImagePlusMatConverter;
import ijopencv.opencv.KeyPointVectorPointRoiConverter;
import ijopencv.ij.PointRoiKeyPointVectorConverter;
import java.util.Vector;
import org.bytedeco.javacpp.opencv_core;
import org.bytedeco.javacpp.opencv_core.KeyPointVector;
import org.bytedeco.javacpp.opencv_core.Mat;
import org.bytedeco.javacpp.opencv_features2d;
import org.bytedeco.javacpp.opencv_features2d.KAZE;
import org.bytedeco.javacpp.opencv_xfeatures2d;
import org.scijava.command.Command;
import org.scijava.plugin.Parameter;
import org.scijava.plugin.Plugin;

@Plugin(type = Command.class, headless = true, menuPath = "Plugins>IJ-OpenCV-
plugins>Keypoint detector")
public class KeyPointDetectorJ_ implements Command {

    @Parameter
    private ImagePlus imp;

    public static final String[] methods = {
        "AGAST", "AKAZE", "BRISK", "FAST", "GFFT", "KAZE", "MSER", "ORB",
"SIFT", "SimpleBlob", "SURF"
    };
    private String method = methods[0];
    private Vector choices;

    @Override
    public void run() {

        GenericDialog gd = new GenericDialog("Select Keypoint detector");
        gd.addChoice("Method", methods, method);
        gd.showDialog();
        if (gd.wasCanceled()) {
            return;
        }
        method = gd.getNextChoice();
        PointRoi res = keyPointDetection(method);
        imp.setRoi(res);

    }

    public PointRoi keyPointDetection(String method) {
        opencv_features2d.Feature2D f2d = new opencv_features2d.Feature2D();

        if (method == "SIFT") {
            f2d = opencv_xfeatures2d.SIFT.create();
        }
        if (method == "SURF") {
```

```
            f2d = opencv_xfeatures2d.SURF.create();
        }
        if (method == "ORB") {
            f2d = opencv_features2d.ORB.create();

        }
        if (method == "BRISK") {
            f2d = opencv_features2d.BRISK.create();
        }
        if (method == "MSER") {
            f2d = opencv_features2d.MSER.create();
        }
        if (method == "AKAZE") {
            f2d = opencv_features2d.AKAZE.create();
        }
        if (method == "SimpleBlob") {
            f2d = opencv_features2d.SimpleBlobDetector.create();
        }

        if (method == "KAZE") {
            f2d = KAZE.create();
        }
        if (method == "AGAST") {
            f2d = opencv_features2d.AgastFeatureDetector.create();
        }
        if (method == "FAST") {
            f2d = opencv_features2d.FastFeatureDetector.create();
        }
        if (method == "GFFT") {
            f2d = opencv_features2d.GFTTDetector.create();
        }

        // Converters
        ImagePlusMatConverter ic = new ImagePlusMatConverter();
        KeyPointVectorPointRoiConverter kpc = new
KeyPointVectorPointRoiConverter();
        PointRoiKeyPointVectorConverter roi2kp = new
PointRoiKeyPointVectorConverter();


        opencv_core.Mat imageOpenCV = ic.convert(imp, Mat.class);

        KeyPointVector kpv = new opencv_core.KeyPointVector();

        f2d.detect(imageOpenCV, kpv);

        PointRoi pr = kpc.convert(kpv, PointRoi.class);
        roi2kp.convert(pr, KeyPointVector.class);


        return pr;

    }

    public static void main(final String... args) throws Exception {
        // Launch ImageJ as usual.
        final ImageJ ij = new ImageJ();
        ij.launch(args);
    }

}

package com.dummys.learning;

import org.opencv.calib3d.Calib3d;
import org.opencv.core.*;
```

```
import org.opencv.features2d.*;
import org.opencv.highgui.Highgui;

import java.io.File;
import java.util.LinkedList;
import java.util.List;

public class SURFDetector {

    public static void main(String[] args) {

        File lib = null;
        String os = System.getProperty("os.name");
        String bitness = System.getProperty("sun.arch.data.model");

        if (os.toUpperCase().contains("WINDOWS")) {
            if (bitness.endsWith("64")) {
                lib = new File("libs//x64//" +
System.mapLibraryName("opencv_java2411"));
            } else {
                lib = new File("libs//x86//" +
System.mapLibraryName("opencv_java2411"));
            }
        }

        System.out.println(lib.getAbsolutePath());
        System.load(lib.getAbsolutePath());

        String bookObject = "images//bookobject.jpg";
        String bookScene = "images//bookscene.jpg";

        System.out.println("Started....");
        System.out.println("Loading images...");
        Mat objectImage = Highgui.imread(bookObject,
Highgui.CV_LOAD_IMAGE_COLOR);
        Mat sceneImage = Highgui.imread(bookScene,
Highgui.CV_LOAD_IMAGE_COLOR);

        MatOfKeyPoint objectKeyPoints = new MatOfKeyPoint();
        FeatureDetector featureDetector =
FeatureDetector.create(FeatureDetector.SURF);
        System.out.println("Detecting key points...");
        featureDetector.detect(objectImage, objectKeyPoints);
        KeyPoint[] keypoints = objectKeyPoints.toArray();
        System.out.println(keypoints);

        MatOfKeyPoint objectDescriptors = new MatOfKeyPoint();
        DescriptorExtractor descriptorExtractor =
DescriptorExtractor.create(DescriptorExtractor.SURF);
        System.out.println("Computing descriptors...");
        descriptorExtractor.compute(objectImage, objectKeyPoints,
objectDescriptors);

        // Create the matrix for output image.
        Mat outputImage = new Mat(objectImage.rows(), objectImage.cols(),
Highgui.CV_LOAD_IMAGE_COLOR);
        Scalar newKeypointColor = new Scalar(255, 0, 0);

        System.out.println("Drawing key points on object image...");
        Features2d.drawKeypoints(objectImage, objectKeyPoints, outputImage,
newKeypointColor, 0);

        // Match object image with the scene image
        MatOfKeyPoint sceneKeyPoints = new MatOfKeyPoint();
        MatOfKeyPoint sceneDescriptors = new MatOfKeyPoint();
        System.out.println("Detecting key points in background image...");
```

```
        featureDetector.detect(sceneImage, sceneKeyPoints);
        System.out.println("Computing descriptors in background image...");
        descriptorExtractor.compute(sceneImage, sceneKeyPoints,
sceneDescriptors);

        Mat matchoutput = new Mat(sceneImage.rows() * 2, sceneImage.cols() *
2, Highgui.CV_LOAD_IMAGE_COLOR);
        Scalar matchestColor = new Scalar(0, 255, 0);

        List<MatOfDMatch> matches = new LinkedList<MatOfDMatch>();
        DescriptorMatcher descriptorMatcher =
DescriptorMatcher.create(DescriptorMatcher.FLANNBASED);
        System.out.println("Matching object and scene images...");
        descriptorMatcher.knnMatch(objectDescriptors, sceneDescriptors,
matches, 2);

        System.out.println("Calculating good match list...");
        LinkedList<DMatch> goodMatchesList = new LinkedList<DMatch>();

        float nndrRatio = 0.7f;

        for (int i = 0; i < matches.size(); i++) {
            MatOfDMatch matofDMatch = matches.get(i);
            DMatch[] dmatcharray = matofDMatch.toArray();
            DMatch m1 = dmatcharray[0];
            DMatch m2 = dmatcharray[1];

            if (m1.distance <= m2.distance * nndrRatio) {
                goodMatchesList.addLast(m1);

            }
        }

        if (goodMatchesList.size() >= 7) {
            System.out.println("Object Found!!!");

            List<KeyPoint> objKeypointlist = objectKeyPoints.toList();
            List<KeyPoint> scnKeypointlist = sceneKeyPoints.toList();

            LinkedList<Point> objectPoints = new LinkedList<>();
            LinkedList<Point> scenePoints = new LinkedList<>();

            for (int i = 0; i < goodMatchesList.size(); i++) {

objectPoints.addLast(objKeypointlist.get(goodMatchesList.get(i).queryIdx).pt)
;

scenePoints.addLast(scnKeypointlist.get(goodMatchesList.get(i).trainIdx).pt);
            }

            MatOfPoint2f objMatOfPoint2f = new MatOfPoint2f();
            objMatOfPoint2f.fromList(objectPoints);
            MatOfPoint2f scnMatOfPoint2f = new MatOfPoint2f();
            scnMatOfPoint2f.fromList(scenePoints);

            Mat homography = Calib3d.findHomography(objMatOfPoint2f,
scnMatOfPoint2f, Calib3d.RANSAC, 3);

            Mat obj_corners = new Mat(4, 1, CvType.CV_32FC2);
            Mat scene_corners = new Mat(4, 1, CvType.CV_32FC2);

            obj_corners.put(0, 0, new double[]{0, 0});
            obj_corners.put(1, 0, new double[]{objectImage.cols(), 0});
            obj_corners.put(2, 0, new double[]{objectImage.cols(),
objectImage.rows()});
            obj_corners.put(3, 0, new double[]{0, objectImage.rows()});
```

```
            System.out.println("Transforming object corners to scene
corners...");
            Core.perspectiveTransform(obj_corners, scene_corners,
homography);

            Mat img = Highgui.imread(bookScene, Highgui.CV_LOAD_IMAGE_COLOR);

            Core.line(img, new Point(scene_corners.get(0, 0)), new
Point(scene_corners.get(1, 0)), new Scalar(0, 255, 0), 4);
            Core.line(img, new Point(scene_corners.get(1, 0)), new
Point(scene_corners.get(2, 0)), new Scalar(0, 255, 0), 4);
            Core.line(img, new Point(scene_corners.get(2, 0)), new
Point(scene_corners.get(3, 0)), new Scalar(0, 255, 0), 4);
            Core.line(img, new Point(scene_corners.get(3, 0)), new
Point(scene_corners.get(0, 0)), new Scalar(0, 255, 0), 4);

            System.out.println("Drawing matches image...");
            MatOfDMatch goodMatches = new MatOfDMatch();
            goodMatches.fromList(goodMatchesList);

            Features2d.drawMatches(objectImage, objectKeyPoints, sceneImage,
sceneKeyPoints, goodMatches, matchoutput, matchestColor, newKeypointColor,
new MatOfByte(), 2);

            Highgui.imwrite("output//outputImage.jpg", outputImage);
            Highgui.imwrite("output//matchoutput.jpg", matchoutput);
            Highgui.imwrite("output//img.jpg", img);
        } else {
            System.out.println("Object Not Found");
        }

        System.out.println("Ended....");
    }
}
```

ДОДАТОК В.

Результати експериментів за наявності перетворення масштабу

Таблиця Б.1 - Отримана матриця гомографії та параметри точності для одного зображення за наявності перетворення масштабу k=0.8

Назва	Матриці перетворення	Δparam	Δpixel
SIFT	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.8024 & 0.0005 & 35.1977 \\ 0.0027 & 0.8011 & 46.7214 \\ 0 & 0 & 1 \end{bmatrix}$	0,0017	0,5316
SURF(128)	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.7959 & 0.0012 & 35.7017 \\ -0.0017 & 0.8005 & 47.0312 \\ 0 & 0 & 1 \end{bmatrix}$	0,0019	1,2183
SURF(64)	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.8032 & 0.0015 & 34.883 \\ 0.0016 & 0.8031 & 46.6723 \\ 0 & 0 & 1 \end{bmatrix}$	0,0023	1,5859
ORB	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.7988 & -0.0014 & 35.7043 \\ 0.0004 & 0.7987 & 47.0297 \\ 0 & 0 & 1 \end{bmatrix}$	0,0011	1,2089
ORB(1000)	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.7991 & -0.002 & 35.7146 \\ 0.0009 & 0.7975 & 47.0387 \\ 0 & 0 & 1 \end{bmatrix}$	0,0016	1,2453
KAZE	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.816 & -0.0072 & 35.2393 \\ 0.01 & 0.7958 & 46.9013 \\ 0 & 0 & 1 \end{bmatrix}$	0,0094	2,2243
AKAZE	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.8046 & 0.0009 & 35.2353 \\ 0.0014 & 0.8067 & 46.3661 \\ 0 & 0 & 1 \end{bmatrix}$	0,0034	1,8717
BRISK	$h = \begin{bmatrix} 0.8 & 0 & 35.4 \\ 0 & 0.8 & 47 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.7916 & -0.0026 & 36.1499 \\ -0.0049 & 0.7948 & 47.5088 \\ 0 & 0 & 1 \end{bmatrix}$	0,0053	1,8208

SURF(128) SURF(64)

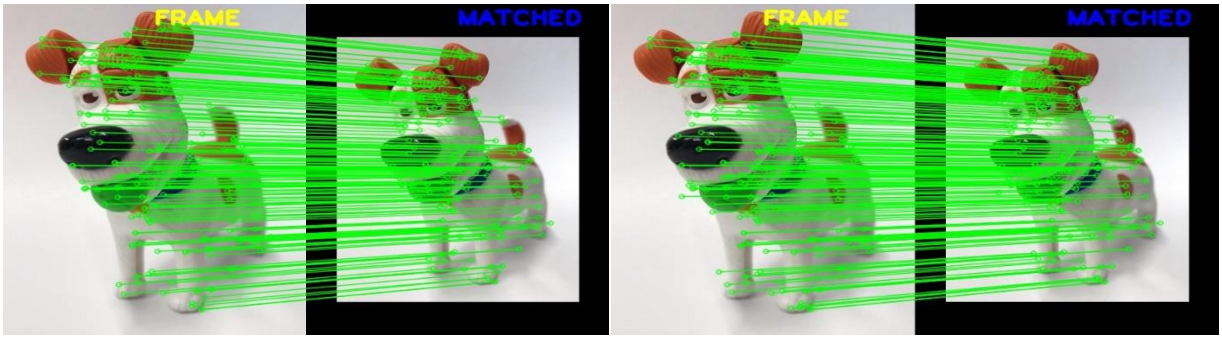
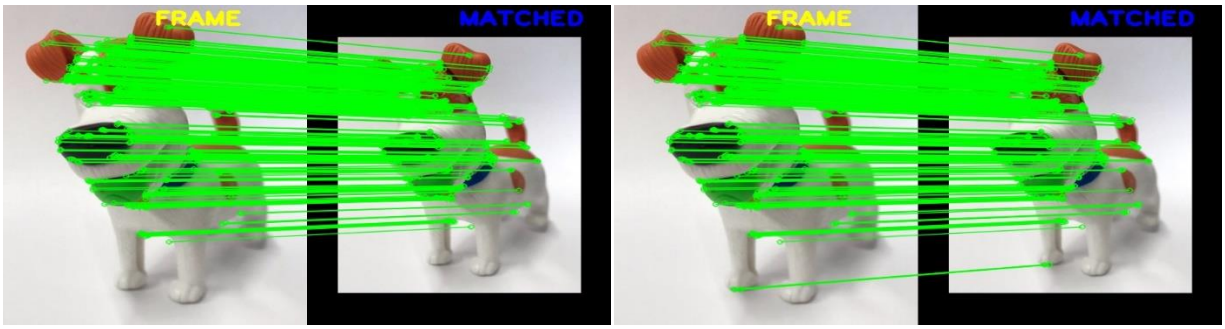
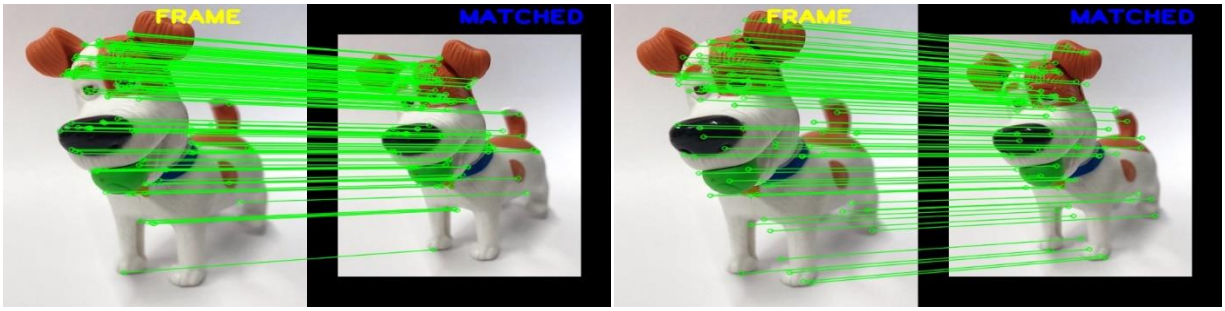


Рисунок Б.1 –Відповідності дескрипторів SURF при масштабі, де k=0.8

ORB(1000) ORB



BRISK SIFT



AKAZE KAZE

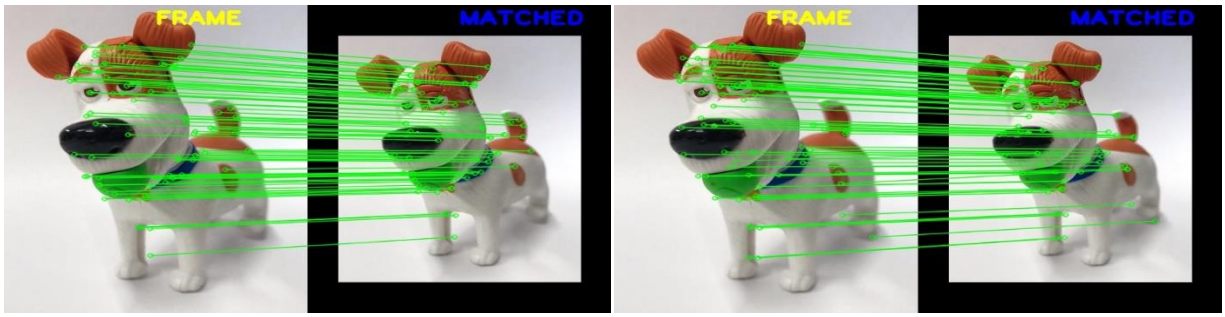


Рисунок Б.2 –Відповідності дескрипторів ORB, SIFT, BRISK, AKAZE, KAZE при масштабі, де k=0.8

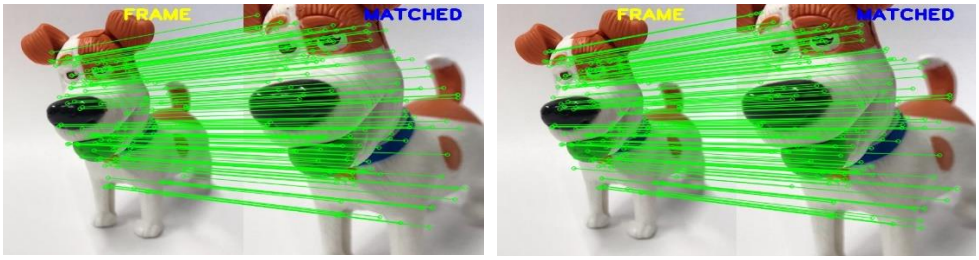
Таблиця Б.2 - Отримана матриця гомографії та параметри точності для одного зображення за наявністю перетворення масштабу k=1.4

Назва	Матриці перетворення	Δparam	Δpixel
1	2	3	4
SIFT	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1.3982 & 0.0012 & -70.6759 \\ 0.0009 & 1.4012 & -94.2659 \\ 0 & 0 & 1 \end{bmatrix}$	0,0013	0,3818
SURF(128)	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1.3818 & 0.0001 & -68.9283 \\ -0.0063 & 1.3911 & -92.401 \\ 0 & 0 & 1 \end{bmatrix}$	0,0084	2,2975
SURF(64)	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1.3852 & -0.004 & -68.7812 \\ 0.0009 & 1.3818 & -91.335 \\ 0 & 0 & 1 \end{bmatrix}$	0,0095	2,6082

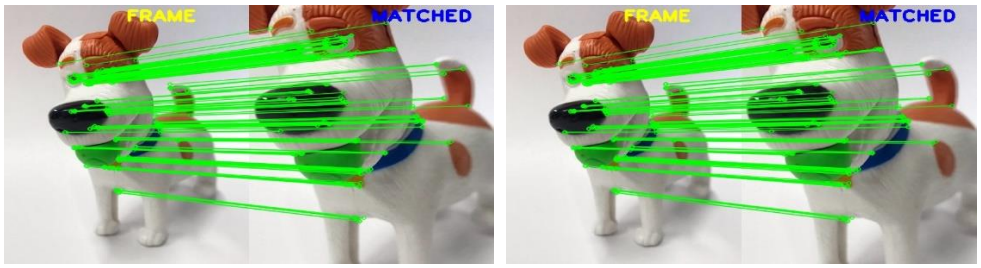
Продовження таблиці Б.2

1	2	3	4
ORB	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1.402 & -0.0005 & -71.0268 \\ 0.0013 & 1.3996 & -94.2061 \\ 0 & 0 & 1 \end{bmatrix}$	0,0011	1,049
ORB(1000)	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1.3948 & -0.0001 & -70.3363 \\ 0.0016 & 1.401 & -97.1447 \\ 0 & 0 & 1 \end{bmatrix}$	0,002	1,0112
KAZE	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1.4248 & 0.004 & -73.5916 \\ 0.0084 & 1.4271 & -97.683 \\ 0 & 0 & 1 \end{bmatrix}$	0,0161	4,3007
AKAZE	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1.4149 & 0.0027 & -72.896 \\ 0.0007 & 1.4112 & -95.0672 \\ 0 & 0 & 1 \end{bmatrix}$	0,0074	1,7338
BRISK	$h = \begin{bmatrix} 1.4 & 0 & -70.8 \\ 0 & 1.4 & -94 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1.4013 & -0.0021 & -71.104 \\ 0.0067 & 1.3812 & -92.9558 \\ 0 & 0 & 1 \end{bmatrix}$	0,0072	1,9976

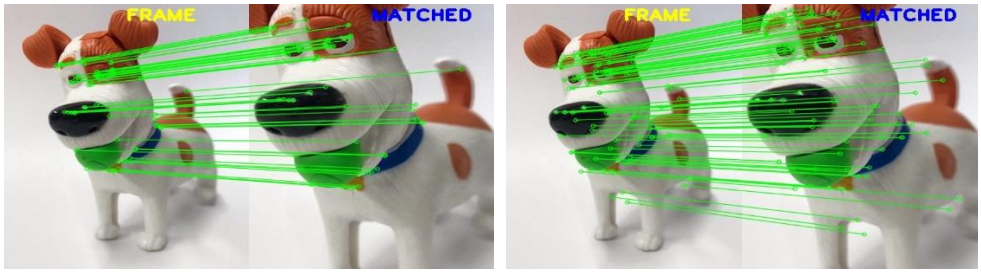
SURF(128) SURF(64)



ORB(1000) ORB



BRISK SIFT



AKAZE KAZE

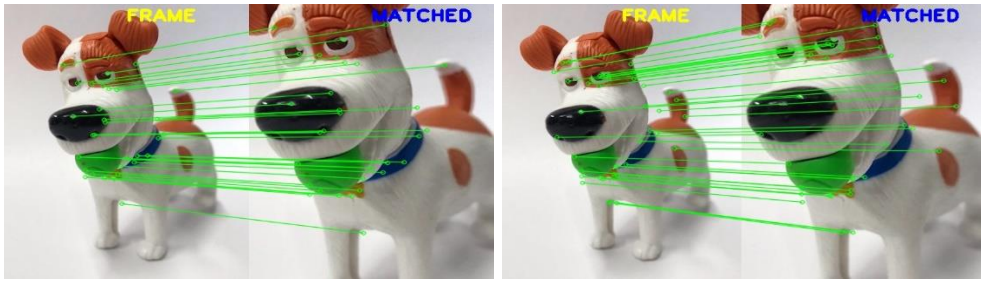


Рисунок Б.3 –Відповідності дескрипторів при масштабі, де k=1.4

ДОДАТОК Г.

Результати експериментів за наявності перетворення повороту

Таблиця В.1 - Отримана матриця гомографії та параметри точності для одного зображення за наявністю перетворення повороту α=20

Назва	Матриці перетворення	Δparam	Δpixel
SIFT	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9416 & 0.3452 & -36.9912 \\ -0.3426 & 0.9428 & 85.7659 \\ 0 & 0 & 1 \end{bmatrix}$	0,0022	0,668
SURF(128)	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9397 & 0.3406 & -36.1638 \\ -0.3399 & 0.9318 & 86.3253 \\ 0 & 0 & 1 \end{bmatrix}$	0,0028	1,5619
SURF(64)	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9536 & 0.3644 & -40.7602 \\ -0.346 & 0.9625 & 84.9654 \\ 0 & 0 & 1 \end{bmatrix}$	0,0157	4,6214
ORB	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9394 & 0.3429 & -36.3892 \\ -0.3426 & 0.941 & 85.585 \\ 0 & 0 & 1 \end{bmatrix}$	0,0008	0,8622
ORB(1000)	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9437 & 0.3463 & -37.1336 \\ -0.3412 & 0.9453 & 85.0074 \\ 0 & 0 & 1 \end{bmatrix}$	0,0037	1,4069
KAZE	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9497 & 0.3398 & -36.0317 \\ -0.3515 & 0.9372 & 86.1083 \\ 0 & 0 & 1 \end{bmatrix}$	0,0015	0,5305
AKAZE	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9376 & 0.3431 & -36.2866 \\ -0.342 & 0.9385 & 85.874 \\ 0 & 0 & 1 \end{bmatrix}$	0,0011	0,6241
BRISK	$h = \begin{bmatrix} 0.9397 & 0.342 & -36.3658 \\ -0.342 & 0.9397 & 85.7594 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.9376 & 0.3392 & -35.7794 \\ -0.3414 & 0.9381 & 85.5236 \\ 0 & 0 & 1 \end{bmatrix}$	0,0018	0,9961

SURF(128) SURF(64)

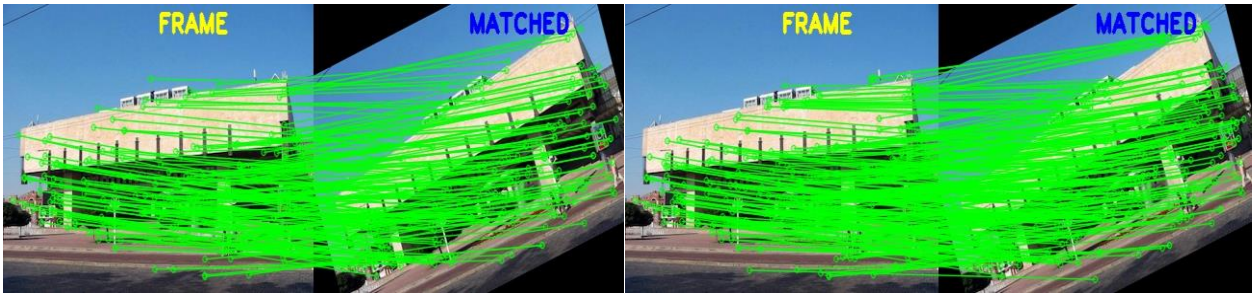


Рисунок В.1 –Відповідності дескрипторів SURF при повороті, де α=20

ORB(1000) ORB

SIFT BRISK

AKAZE KAZE

Рисунок В.2 –Відповідності дескрипторів ORB, SIFT, BRISK, AKAZE, KAZE при повороті, де α=20

Таблиця В.2 - Отримана матриця гомографії та параметри точності для одного зображення за наявності перетворення повороту α=80

Назва	Матриці перетворення	Δparam	Δpixel
SIFT	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.1413 & 0.9911 & 48.7439 \\ -0.9909 & 0.1396 & 349.1799 \\ 0 & 0 & 1 \end{bmatrix}$	0,001	0,5291
SURF(128)	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.132 & 0.9842 & 50.1548 \\ -0.9869 & 0.1392 & 347.0924 \\ 0 & 0 & 1 \end{bmatrix}$	0,0042	2,5509
SURF(64)	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.1522 & 0.9919 & 47.4113 \\ -0.9876 & 0.1338 & 349.5517 \\ 0 & 0 & 1 \end{bmatrix}$	0,0057	2,7224
ORB	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 0.1429 & 0.9924 & 48.6168 \\ -0.9905 & 0.1398 & 348.3965 \\ 0 & 0 & 1 \end{bmatrix}$	0,0017	1,3994

Продовження таблиці В.2

Назва	Матриці перетворення	Δparam	Δpixel
ORB(1000)	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.1457 & 0.9932 & 48.2881 \\ -0.9917 & 0.1389 & 348.8884 \\ 0 & 0 & 1 \end{bmatrix}$	0,0028	1,9703
KAZE	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.1409 & 0.9928 & 48.6855 \\ -0.99 & 0.1391 & 348.5997 \\ 0 & 0 & 1 \end{bmatrix}$	0,0011	0,7368
AKAZE	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.1381 & 0.9852 & 49.5361 \\ -0.9868 & 0.1375 & 347.6071 \\ 0 & 0 & 1 \end{bmatrix}$	0,0028	1,0989
BRISK	$h = \begin{bmatrix} 0.1392 & 0.9903 & 49.1069 \\ -0.9903 & 0.1392 & 348.491 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 0.1399 & 0.9947 & 48.997 \\ -0.9906 & 0.1417 & 348.3304 \\ 0 & 0 & 1 \end{bmatrix}$	0,002	1,2559

SURF(128) SURF(64)

ORB(1000) ORB

AKAZE KAZE

SIFT BRISK

Рисунок В.3 –Відповідності дескрипторів при повороті, де α=80

ДОДАТОК Д.

Результати експериментів за наявності центроафінного перетворення

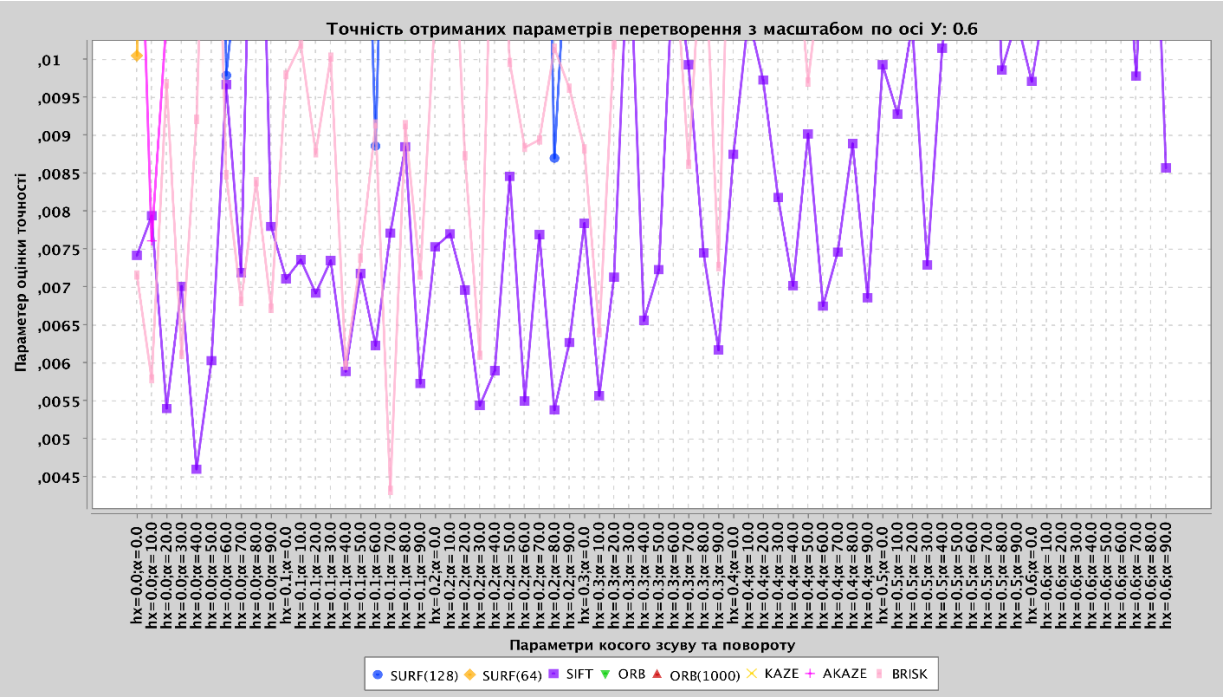


Рисунок Г.1 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=0.6

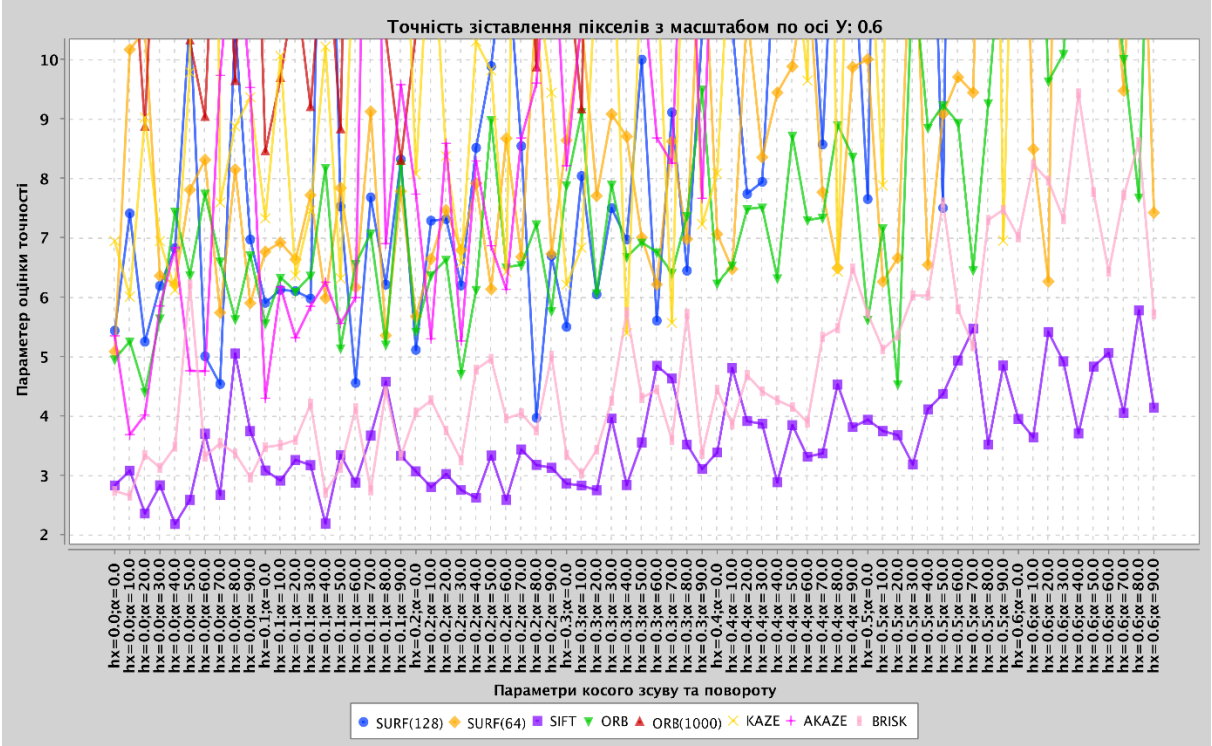


Рисунок Г.2 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=0.6

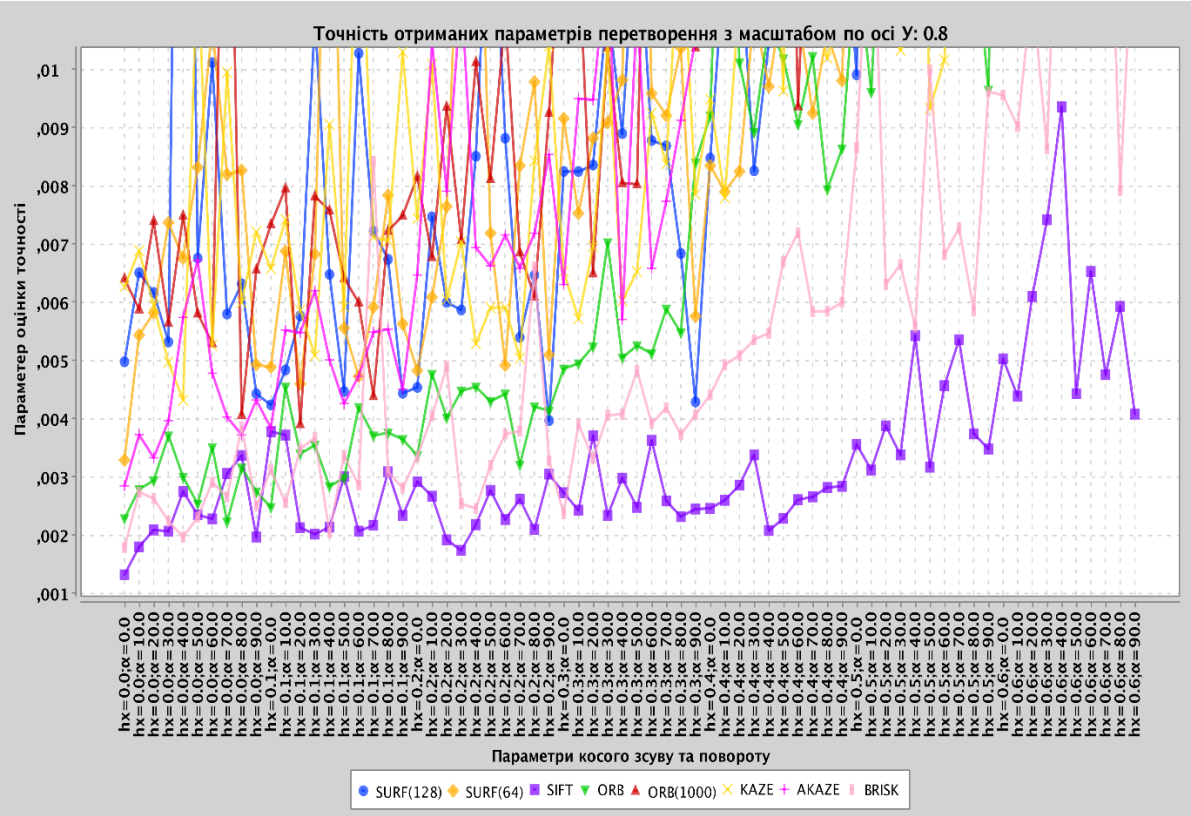


Рисунок Г.3 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=0.8

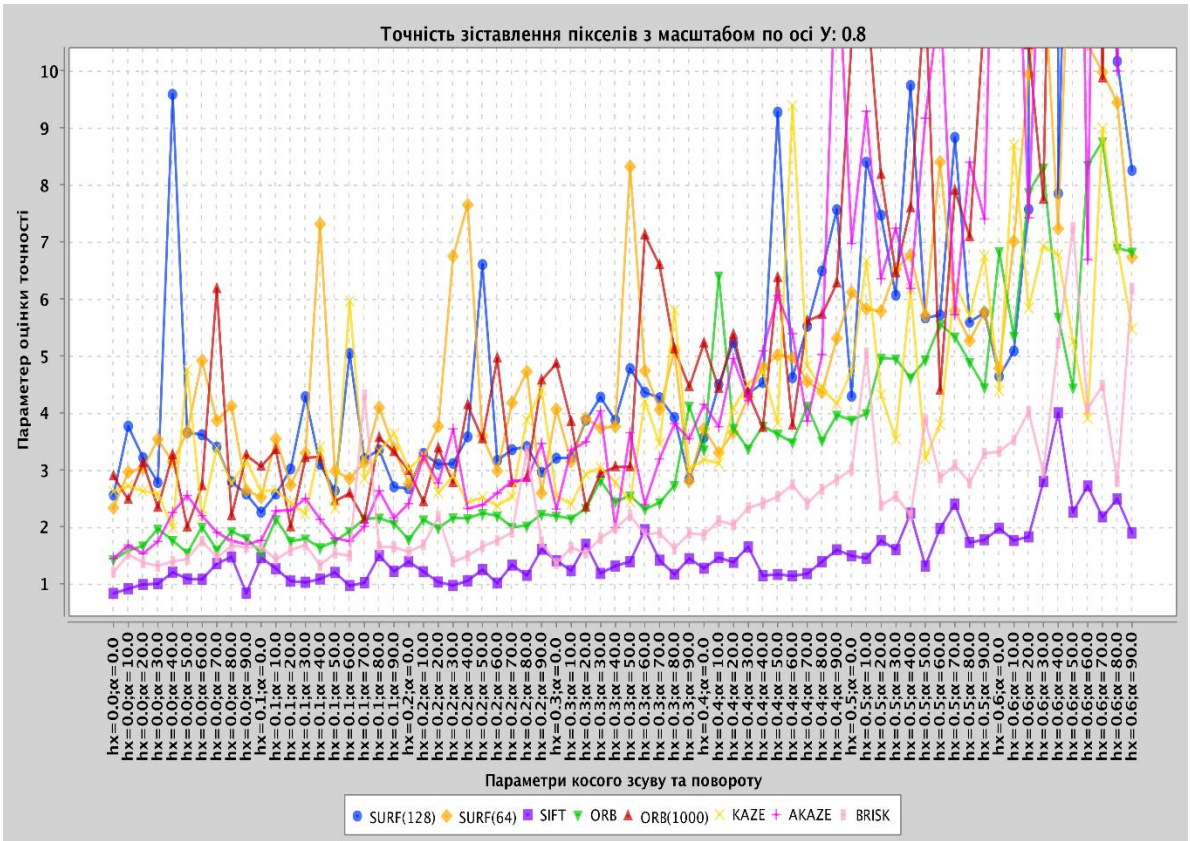


Рисунок Г.4 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=0.8

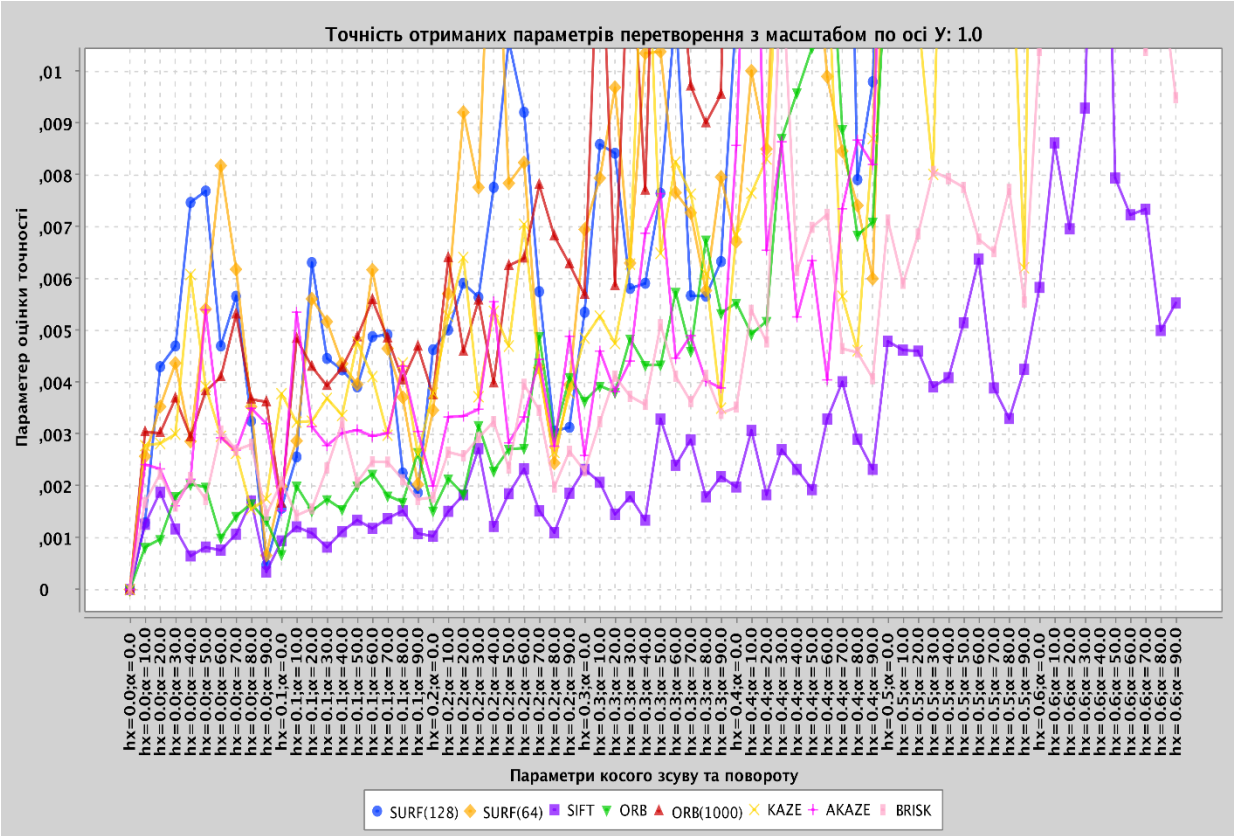


Рисунок Г.5 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.0

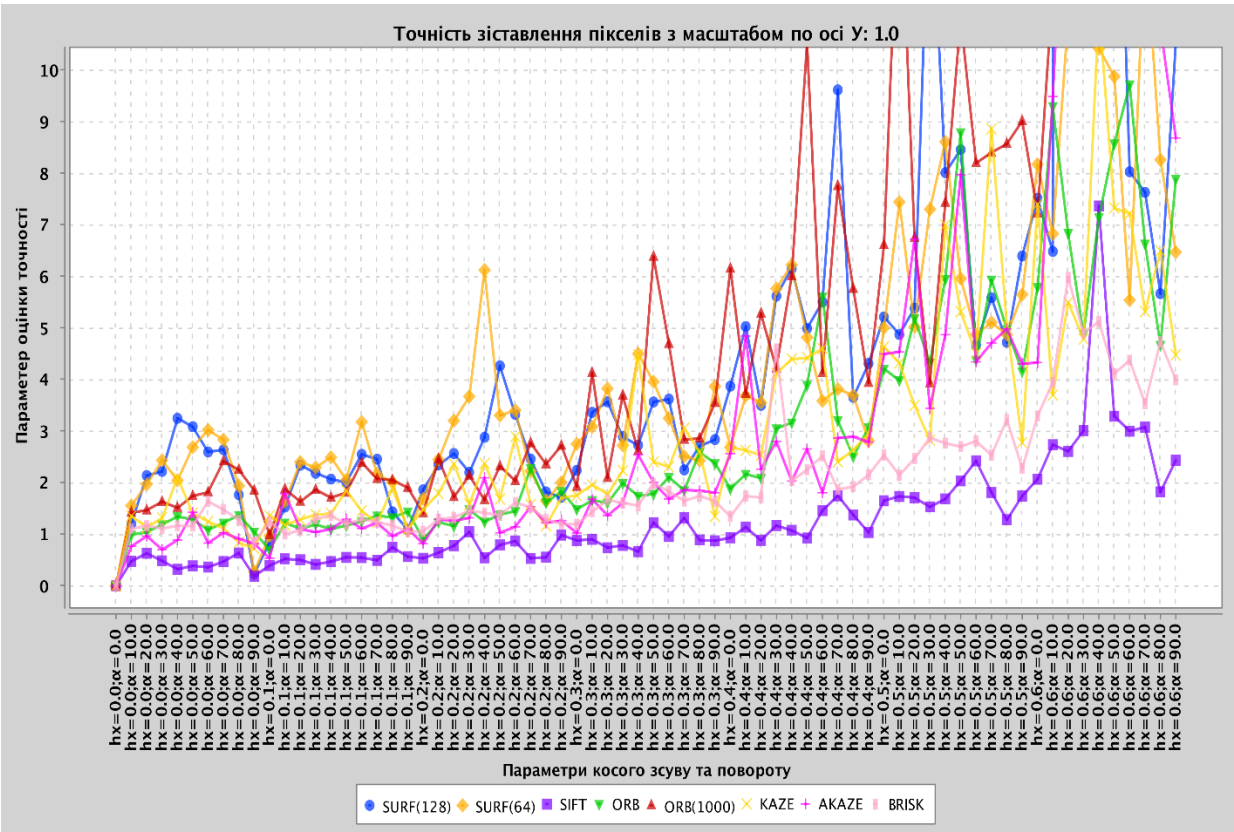


Рисунок Г.6 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=1.0

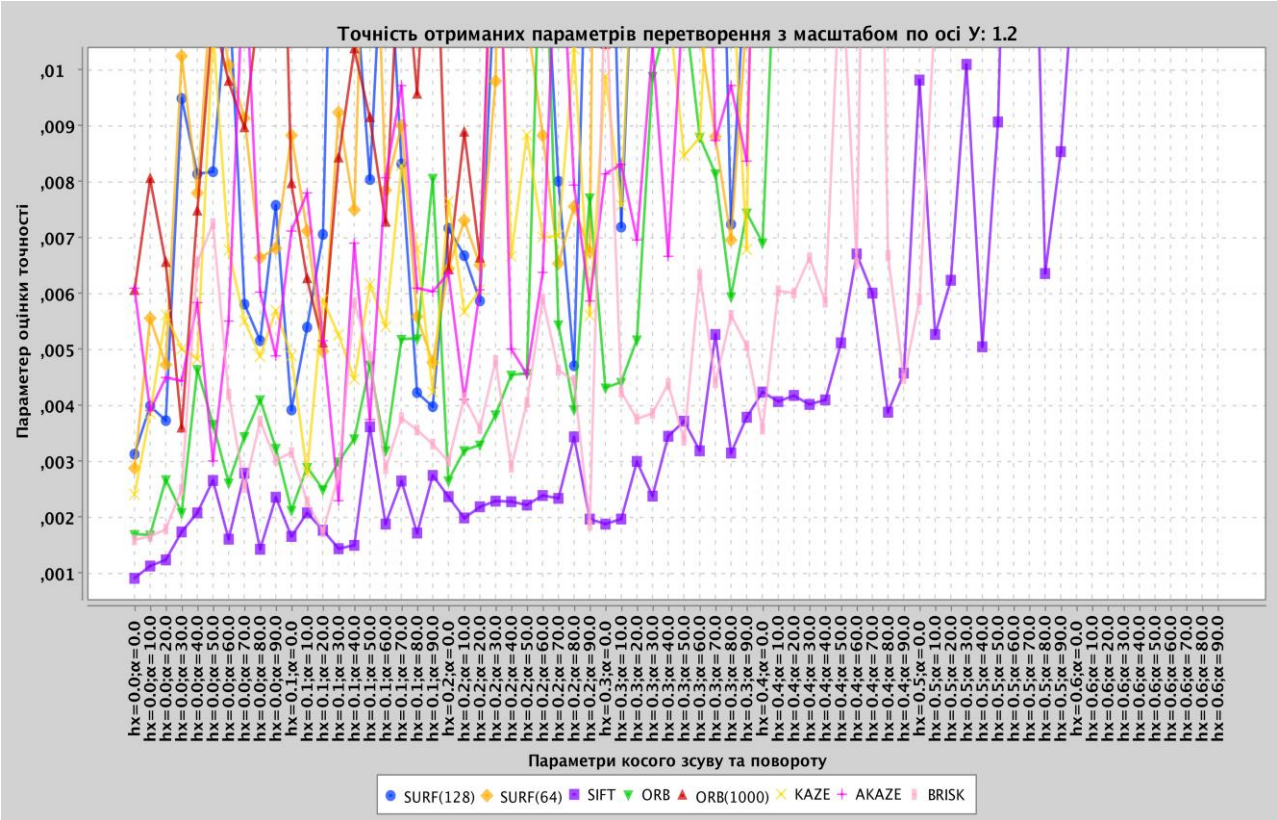


Рисунок Г.7 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.2

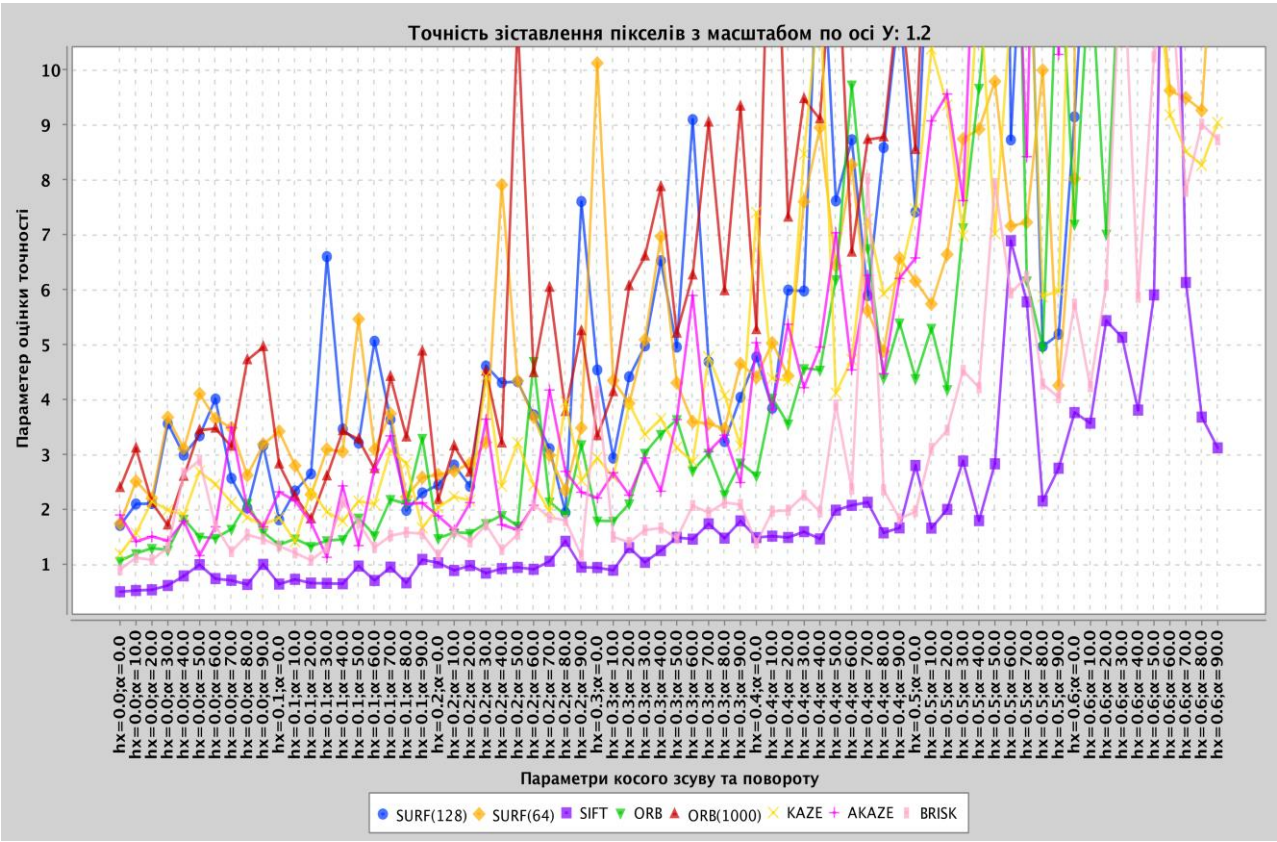


Рисунок Г.8 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=1.2

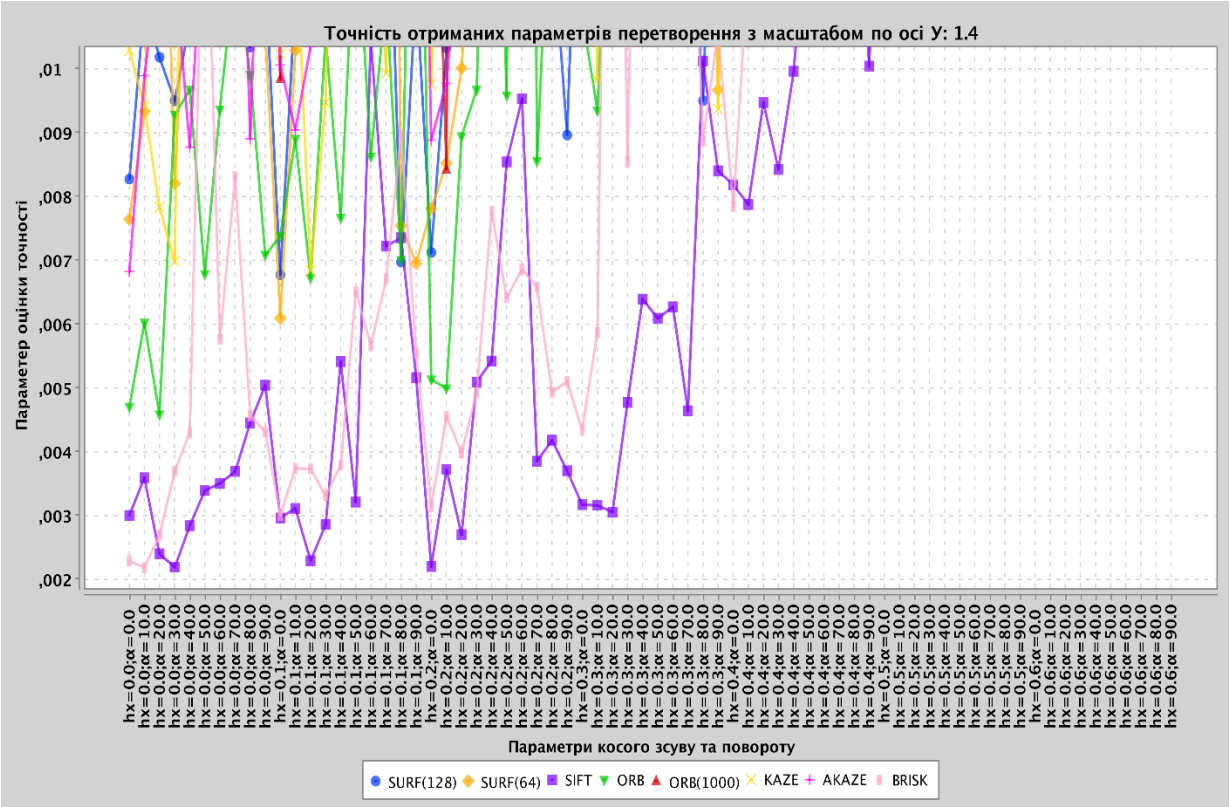


Рисунок Г.9 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.4

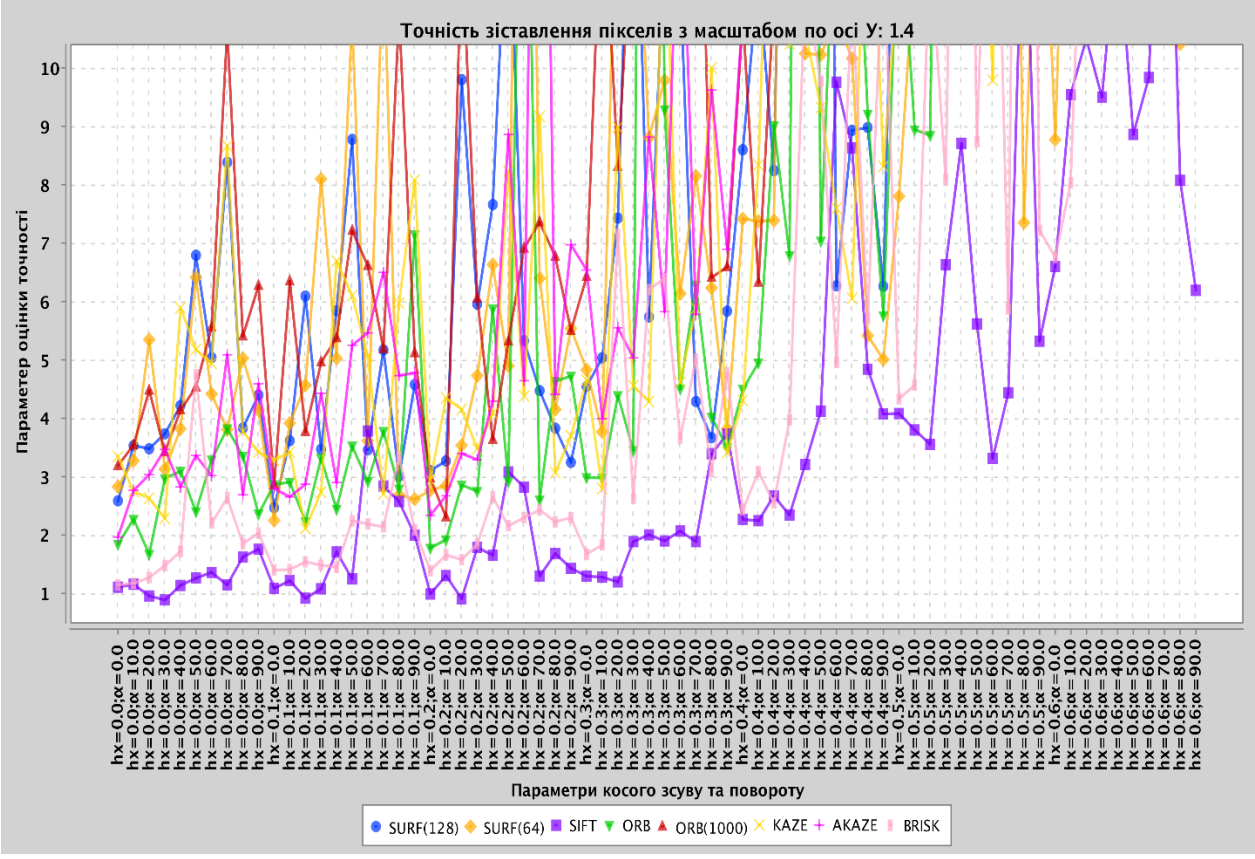


Рисунок Г.10 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=1.4

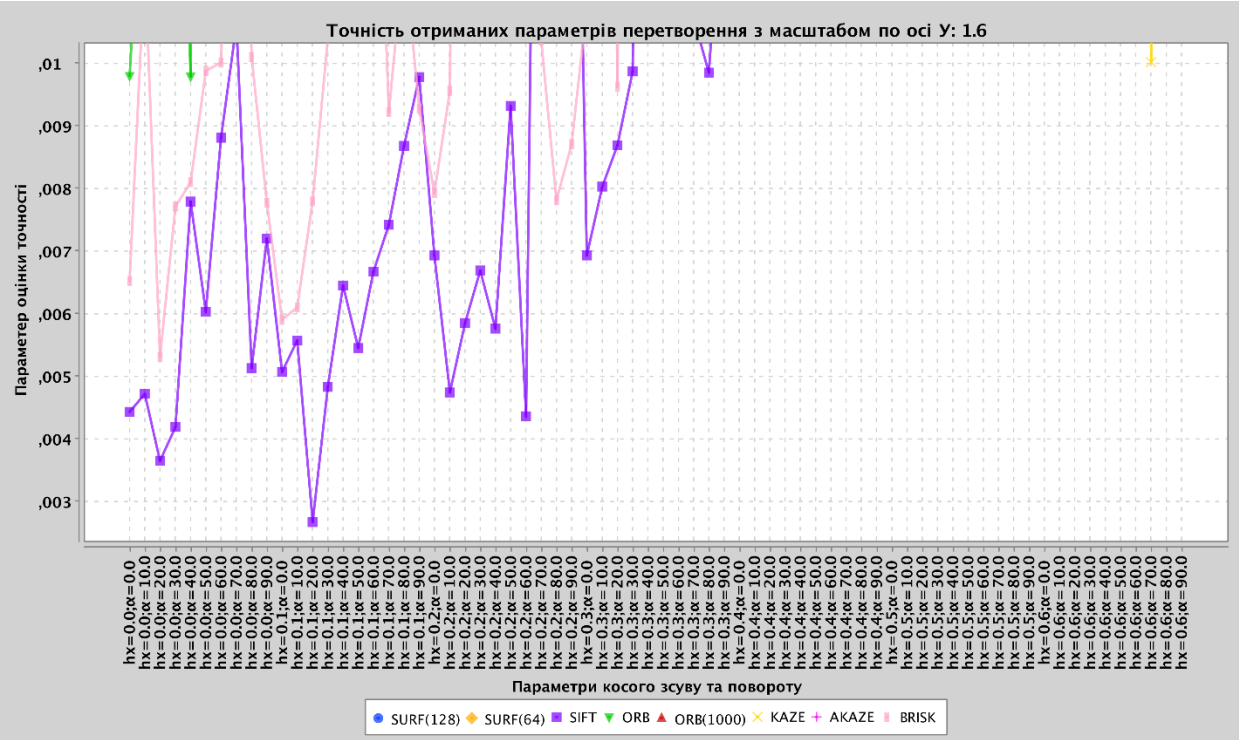


Рисунок Г.11 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.6

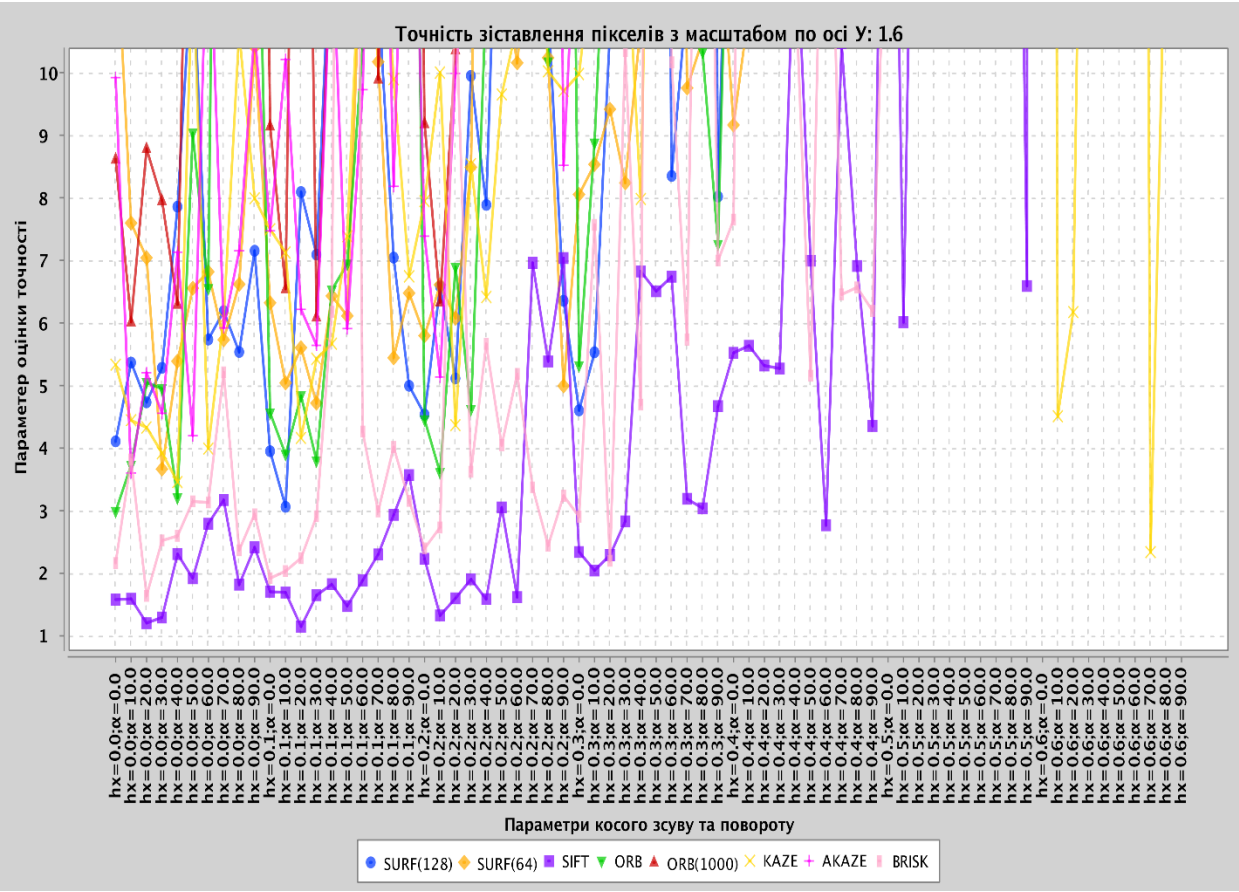


Рисунок Г.12 - Графік точності піксельного заставлення зображень для фіксованого параметру неоднорідного масштабу, де b=1.6

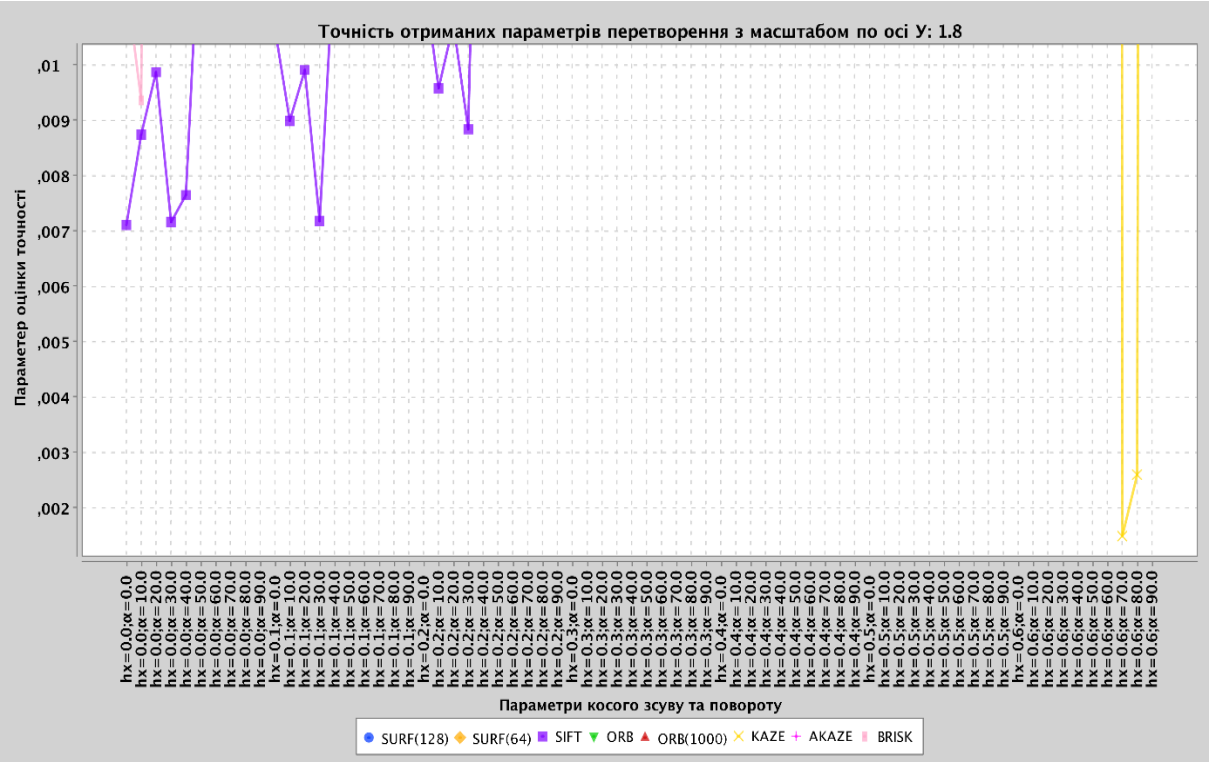


Рисунок Г.13 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.8

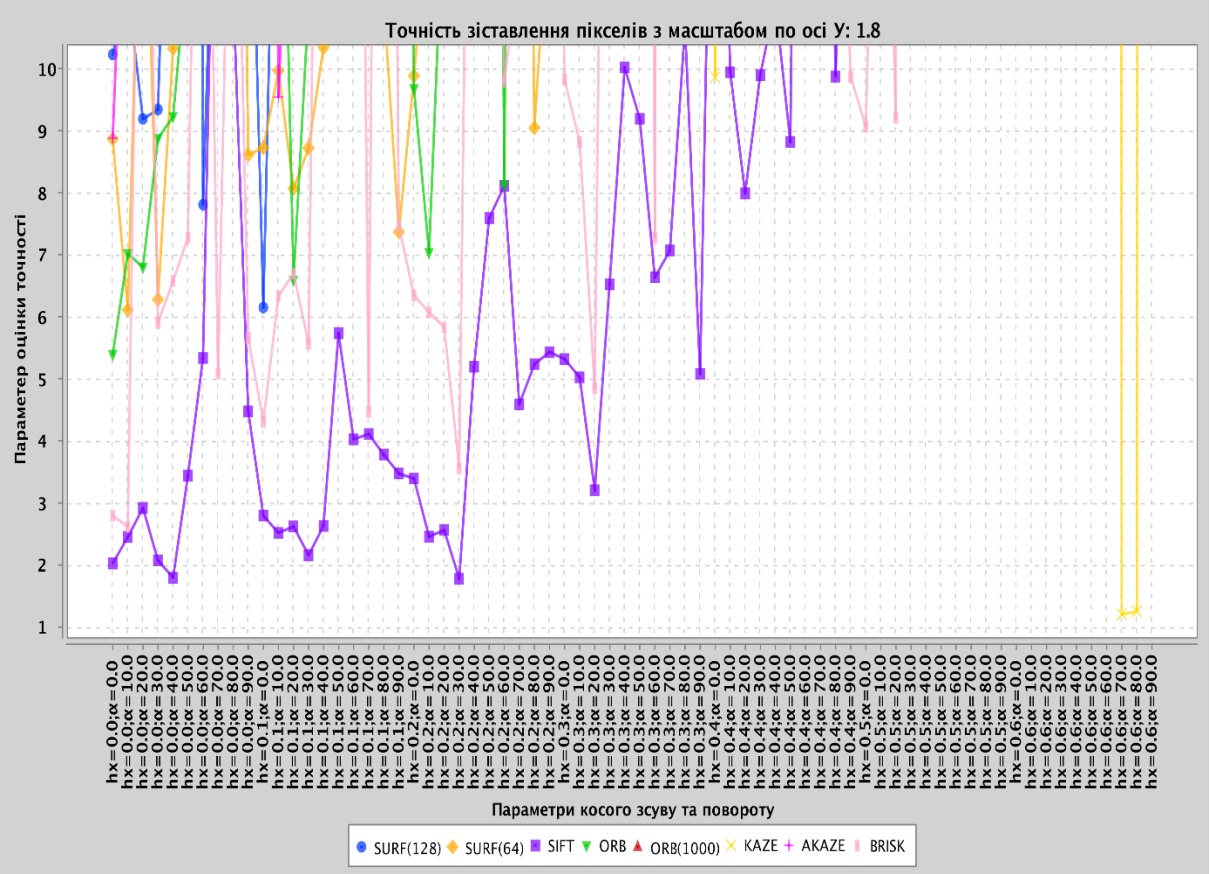


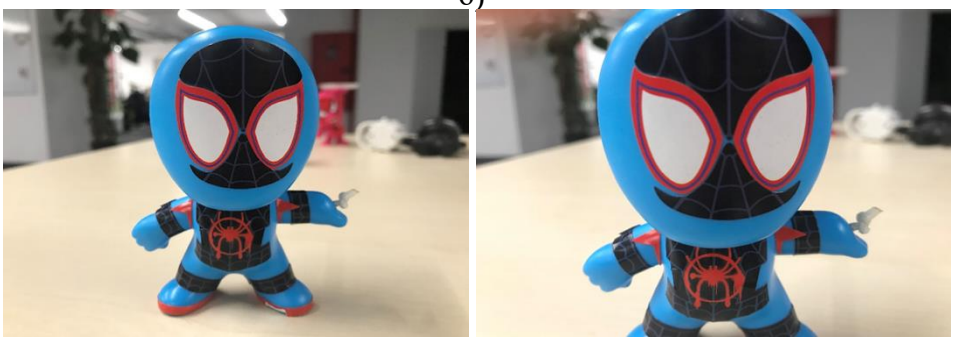
Рисунок Г.14 - Графік точності знайдених параметрів перетворення для фіксованого параметру неоднорідного масштабу, де b=1.8

ДОДАТОК Е.

Результати експериментів для реальних зображень

а)

б)



в)

г)

Рисунок Д.1 – Пари порівняльних зображень:

а – graph, б – boat , в – spider, г – dog

SURF(128)

SURF(64)

ORB(1000)

ORB

AKAZE

KAZE

BRISK

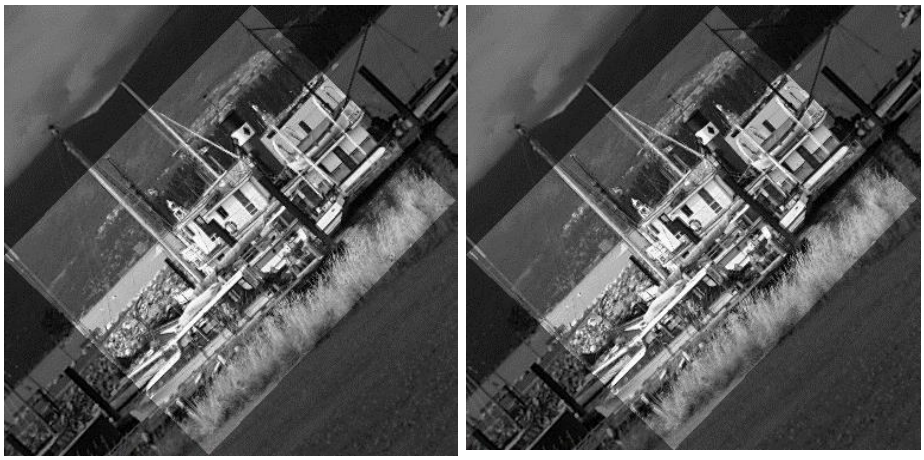
SIFT

Рисунок Д.2 – Результати нормалізації зображень пари graph

SURF(128) SURF(64)

ORB(1000) ORB

SIFT BRISK



AKAZE KAZE

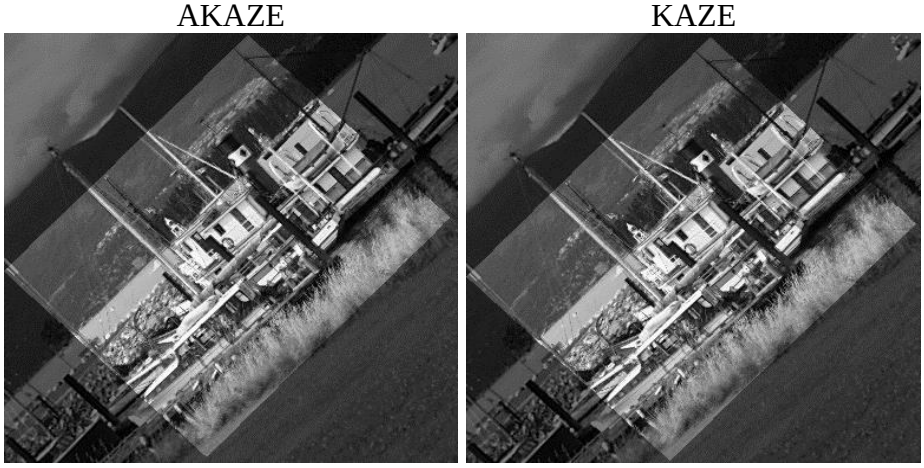


Рисунок Д.3 – Результати нормалізації зображень пари boat

SURF(128) SURF(64)

ORB(1000) ORB

SIFT BRISK

AKAZE KAZE

Рисунок Д.4 – Результати нормалізації зображень пари spider

SURF(128)

SURF(64)

ORB(1000)

ORB

SIFT

BRISK

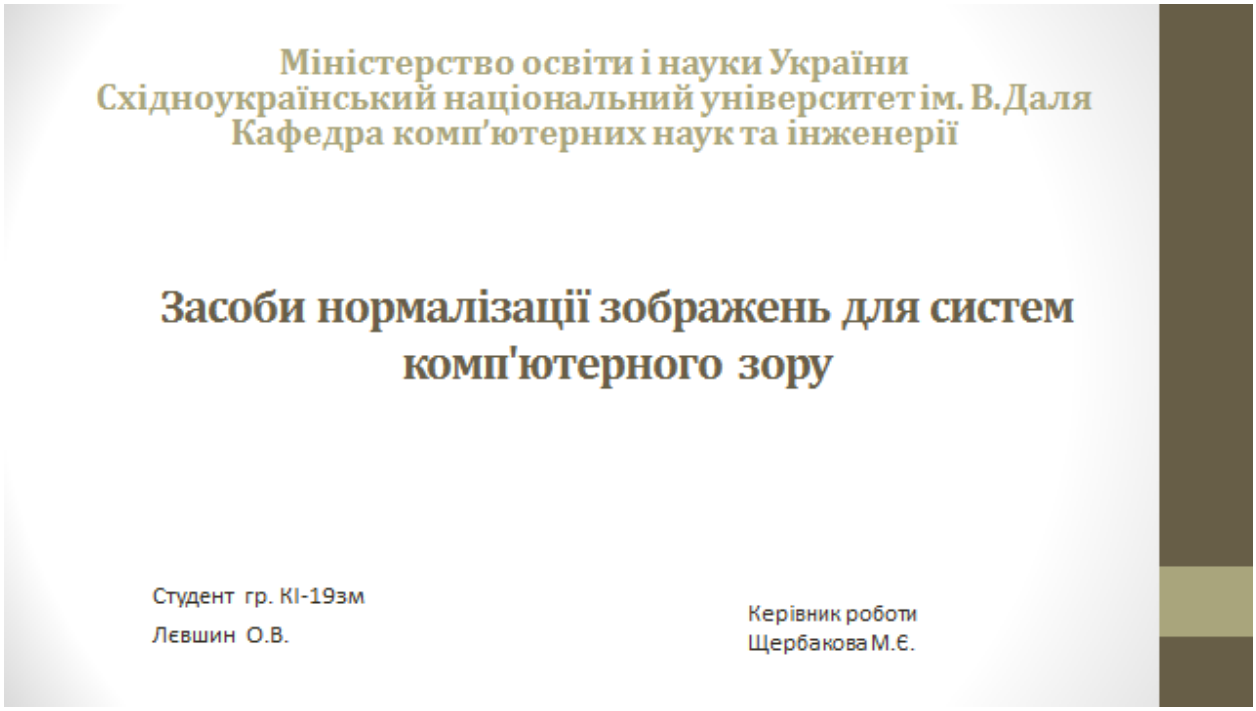
AKAZE

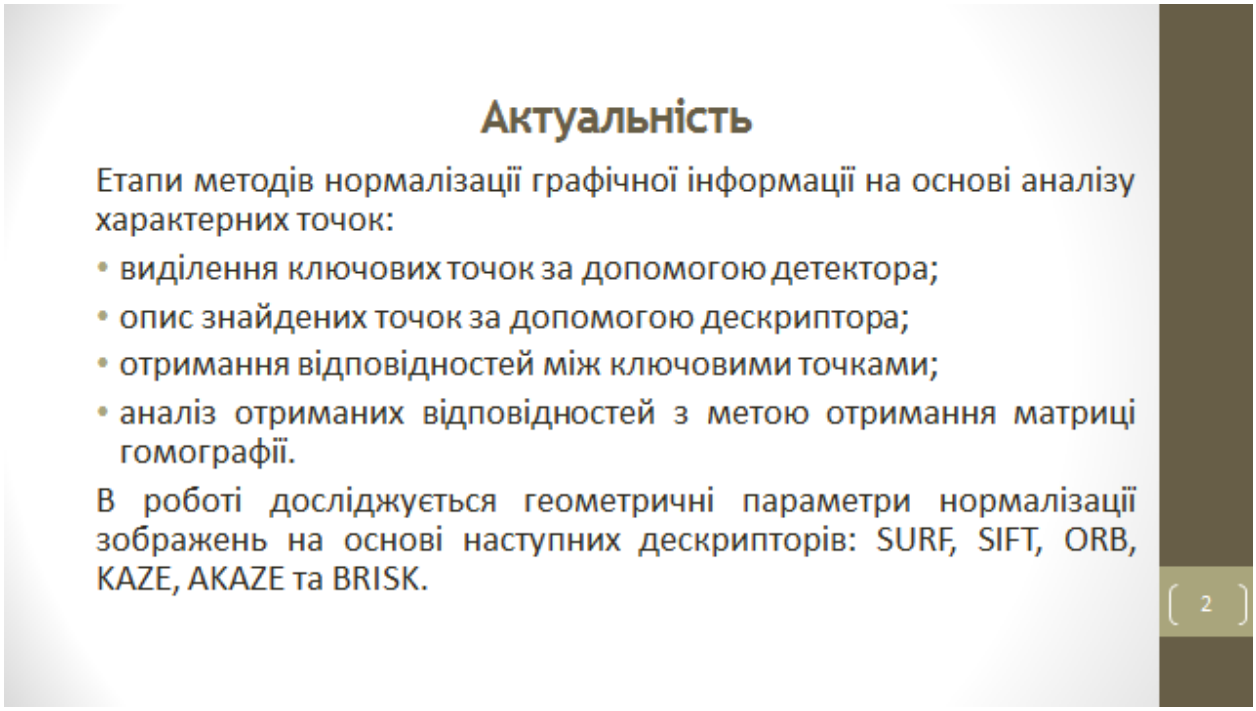
KAZE

Рисунок Д.5 – Результати нормалізації зображень пари dog

ДОДАТОК Ж.

Електронні плакати





Постановка задачі

Метою роботи є розробка методу нормалізації геометричних перетворень, що базуються на аналізі характерних точок, які дозволяють нормалізувати вхідне зображення до еталонного для подальшого його аналізу.

Об'єктом дослідження є послідовність геометрично-синтезованих та реальних зображень.

Задачі дослідження:

- аналіз ідентифікації, опису та відповідності дескрипторів;
- розгляд дескрипторів SURF, SIFT, ORB, BRISK, KAZE та AKAZE;
- вивчення методів для відкидання поганих зв'язків;
- розробка методу нормалізації на основі аналізу характерних точок;
- створення тестової колекції зображень;
- дослідження у порівняльному аспекті точності та швидкодії методу нормалізації на основі аналізу дескрипторів SURF, SIFT, ORB, BRISK, KAZE та AKAZE, розробка програмного застосунку для проведення досліджень

3

Питання нормалізації геометричних перетворень в задачах аналізу та обробки зображень

- сегментація - виділення окремих областей зображення з фону;
- реставрація - фільтрація з метою поліпшення яскравості, видалення засвічувань, тощо;
- нормалізація - приведення заданого зображення до деякого еталонного зображення;
- розпізнавання - зображення класифікується залежно від початково заданої задачі.

4

Аналіз досліджуваних дескрипторів

Назва	Тип дескриптору	Розмір дескриптору
SIFT	Blobs	128 Floats
SURF (128D)	Blobs	128 Floats
SURF (64D)	Blobs	64 Floats
ORB	Corners	32 Bytes
KAZE	Blobs	128 Floats
AKAZE	Blobs	61 Bytes
BRISK	Corners	64 Bytes

5

Математичні моделі нормалізації зображень на основі дескрипторів їх характерних точок

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

де L - значення гауссіана в точці з вимірюванням (x, y), σ - радіус розмиття, G - гауссове ядро, I - значення вихідного зображення, * - операція згортки.

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

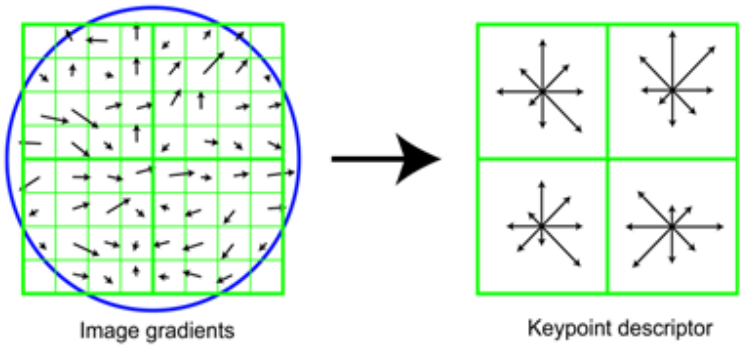
$$D(x) = D + \frac{\partial D^T}{\partial x} x + 1/2\, x^T \frac{\partial^2 D}{\partial x^2} x$$

де D і її похідна обчислюються в точці-кандидата,

$x = (x, y, \sigma)^T$ є зміщенням від цієї точки

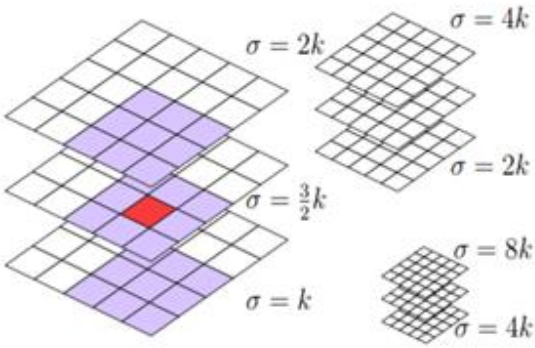
6

Приклад масиву з дескрипторів 2x2 отриманого з вибірки 8x8



7

3 октави з трьома рівнями, виділено сусідство для «придушення не-максимумів» 3 × 3 × 3, що використовується для виявлення особливих точок



8

OpenCV об'єкти для визначення дескрипторів

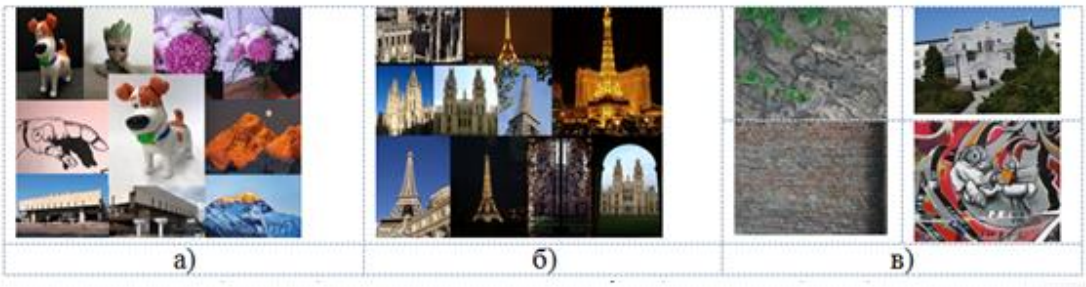
Назва алгоритму	Об'єкт OpenCV
SIFT	SIFT sift = SIFT.create();
SURF (128D)	SURF surf128 = SURF.create(); surf128.setExtended(true);
SURF (64D)	SURF surf64 = SURF.create(); surf128.setExtended(false);
ORB	ORB orb = ORB.create(); orb.setMaxFeatures(100000);
ORB (1000)	ORB orb1000 = ORB.create(); orb1000.setMaxFeatures(1000);

KAZE	KAZE kaze = KAZE.create();
	kaze.setNOctaveLayers(3);
	kaze.setExtended(true);

AKAZE	AKAZE akaze = AKAZE.create(); akaze.setNOctaveLayers(3);
BRISK	BRISK brisk = BRISK.create();

9

Набори зображень



а - персональні зображення; б - зображення з сайту Oxford; в – синтезовані зображення

10

Умови та задачі експериментів щодо дослідження інваріантності

$$\Delta param = \frac{\sum_{i,j=1}^{2} \rho(a_{ij}, a'_{ij})}{M}$$

де a_{ij} – параметри заданої матриці центроафінного перетворення; a'_{ij} – параметри отриманої матриці центроафінного перетворення; M = 4 – кількість параметрів перетворення, $\rho(a_{ij}, a'_{ij})$ – міра подібності, яка у роботі буде розрахована як норма L1

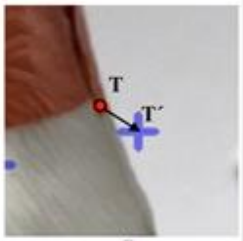
$$\rho(a_{ij}, a'_{ij}) = |a_{ij} - a'_{ij}|$$

$$\Delta pixel = \frac{\sum_{i=1}^{N} \rho(T_i, T'_i)}{N}$$

де N – кількість відповідностей характерних точок,
$\rho(T_i, T'_i)$ – міра подібності, яка у роботі буде розрахована як норма L2:

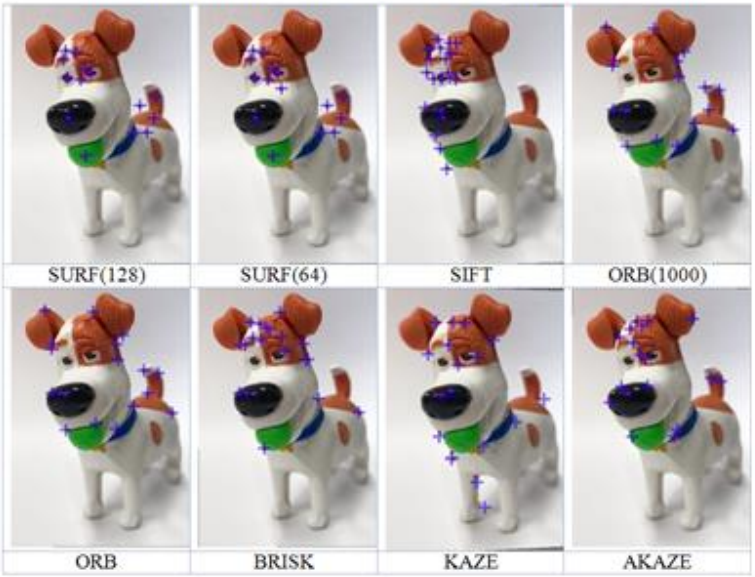
$$\rho(T_i, T'_i) = \sqrt{(x_{T_i} - x_{T'_i})^2 + (y_{T_i} - y_{T'_i})^2}$$

а)



б)

Зіставленні зображення та їх характерні точки



Усереднені оцінки точності Δparam та Δpixel для аналізованих дескрипторів за наявності перетворення масштабу з коефіцієнтом k=[0.5, 2]

k	Точність	SURF 128	SURF 64	SIFT	ORB 1000	ORB	KAZE	AKAZE	BRISK
0.5	Δparam	0.005	0.008	0.011	0.024	0.004	0.018	0.022	0.006
	Δpixel	2.991	4.292	4.212	7.219	2.529	10.94	6.878	4.108
0.6	Δparam	0.005	0.004	0.017	0.011	0.008	0.015	0.0109	0.003
	Δpixel	2.927	2.854	4.69	4.108	2.507	6.05	4.102	1.92
0.7	Δparam	0.002	0.002	0.002	0.005	0.001	0.018	0.01	0.003
	Δpixel	1.654	1.726	1.01	2.273	1.267	5.38	3.456	1.836
0.8	Δparam	0.003	0.0013	0.002	0.004	0.002	0.005	0.004	0.001
	Δpixel	1.691	1.699	0.771	1.883	1.299	2.201	2.02	1.146
0.9	Δparam	0.002	0.002	0.001	0.003	0.002	0.004	0.003	0.001
	Δpixel	1.239	1.281	0.549	1.617	1.561	1.645	1.24	0.962
1.1	Δparam	0.002	0.002	0.001	0.005	0.0014	0.003	0.002	0.0016
	Δpixel	1.123	1.27	0.226	1.985	1.069	1.369	0.962	0.921
1.2	Δparam	0.004	0.004	0.001	0.005	0.002	0.005	0.004	0.002
	Δpixel	1.605	1.648	0.221	1.855	1.052	1.784	1.477	1.022
1.3	Δparam	0.003	0.004	0.001	0.006	0.002	0.006	0.007	0.004
	Δpixel	1.24	1.488	0.46	2.221	1.195	1.815	2.036	1.425
1.4	Δparam	0.005	0.004	0.001	0.007	0.003	0.016	0.008	0.0027
	Δpixel	1.549	1.472	0.414	2.127	1.26	3.96	2.031	1.095
1.5	Δparam	0.006	0.005	0.001	0.009	0.004	0.009	0.013	0.005
	Δpixel	1.626	1.483	0.222	2.369	1.222	2.225	2.89	1.492
1.6	Δparam	0.009	0.009	0.001	0.0155	0.06	0.011	0.014	0.007
	Δpixel	2.64	2.428	0.245	3.547	1.612	2.48	2.985	1.791
1.7	Δparam	0.011	0.011	0.002	0.0142	0.007	0.0181	0.013	0.009
	Δpixel	2.219	2.681	0.415	2.962	1.641	2.843	2.689	2.228
1.8	Δparam	0.02	0.011	0.002	0.0864	0.0078	0.0195	0.025	0.01
	Δpixel	4.207	2.998	0.425	15.045	1.825	3.683	5.039	2.416
1.9	Δparam	0.0127	0.0105	0.003	0.11	0.011	0.048	0.045	0.009
	Δpixel	2.804	2.201	0.562	17.533	2.349	8.479	12.244	1.931
2.0	Δparam	0.008	0.013	0.003	0.113	0.017	0.036	0.241	0.012
	Δpixel	1.606	2.658	0.658	22.271	2.257	6.58	11.205	2.448

Дослідження методів

Таблиця 1 – Отримані кількісні параметри для k=0.5

Назва	img1,	img2,	M_T	kNT,	R_T	img1$_c$, сек	img2$_c$, сек	M_T, сек	kNN$_c$, сек	R_c, сек	Загальний час, сек
SIFT	206	110	206	76	67	0,0528	0,0471	0,0081	0,0002	0,0024	0,1105
SURF(128)	394	250	394	129	122	0,0535	0,0437	0,0135	0,0007	0,0007	0,1121
SURF(64)	394	250	394	139	127	0,0485	0,0417	0,0026	0,0001	0,0006	0,0945
ORB(1000)	870	874	870	250	243	0,0068	0,0059	0,0067	0,0003	0,0009	0,0205
ORB	871	874	871	250	243	0,0076	0,006	0,0093	0,0003	0,0012	0,0244
KAZE	168	131	168	34	33	0,2089	0,2091	0,0012	0,0001	0,0005	0,4198
AKAZE	161	139	161	52	51	0,0259	0,024	0,0006	0	0,0004	0,0511
BRISK	261	292	261	93	92	0,0092	0,0081	0,0011	0,0001	0,0005	0,0191

Таблиця 2 – Отримані кількісні параметри для k=2.0

Назва	img1,	img2,	M_T	kNN,	R_T	img1$_c$, сек	img2$_c$, сек	M_T, сек	kNN$_c$, сек	R_c, сек	Загальний час, сек
SIFT	206	160	206	57	52	0,0514	0,0359	0,0016	0,00001	0,0008	0,0897
SURF(128)	394	357	394	75	70	0,0475	0,04	0,007	0,0001	0,0004	0,0949
SURF(64)	394	357	394	82	75	0,0443	0,0366	0,0031	0,0001	0,0004	0,0844
ORB(1000)	870	142	870	72	62	0,005	0,003	0,0021	0,0001	0,0004	0,0107
ORB	871	142	871	72	64	0,0056	0,0031	0,0021	0,0002	0,0004	0,0114
KAZE	168	101	168	21	13	0,1698	0,1539	0,0011	0,00001	0,0007	0,3255
AKAZE	161	54	161	23	20	0,0245	0,0223	0,0004	0,00001	0,0004	0,0477
BRISK	261	46	261	30	26	0,0071	0,0037	0,0006	0,00001	0,0003	0,0117

Таблиця 3 – Результуючий час виконання за одну ключову точку

Назва	img1′$_c$, мкс	img2′$_c$, мкс	M′$_c$, мкс	kNN′$_c$, мкс	R′$_c$, мкс
SIFT	244,030141	208,24554	13,737906	0,588449	7,480607
KAZE	1104,743008	1212,983609	9,600341	0,911238	15,288165
SURF(128)	136,940441	123,779191	20,688066	1,013253	3,723593
ORB(1000)	6,357703	9,965047	7,627997	0,972632	3,729259
AKAZE	170,749228	224,151566	4,108976	1,070238	11,055972
SURF(64)	127,418744	121,958063	10,341488	0,554862	3,587998
BRISK	30,676016	44,82015	3,895771	0,886459	8,38787
ORB	6,933812	9,753324	6,985975	1,140168	3,641543

Аналізу дескрипторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK

Комбінація перетворень зображення:

- перетворенням однорідного масштабу в діапазоні [0.5, 2],
- поворот при діапазоні [0, 90],
- центроафінне перетворення при наявності двох останніх та косого зсуву при діапазоні [0, 1]

Результати аналізу:

1) при наявності однорідного масштабу, де $k \in [0,5; 2]$ розташування дескрипторів у такому порядку: SIFT < ORB,BRISK< SURF (64), SURF(128)< ORB(1000),AKAZE,KAZE;
2) при наявності повороту зображення, де $\alpha \in [0; 90]$: SIFT< ORB,BRISK,AKAZE<KAZE<ORB (1000)< SURF(64)< SURF(128);
3) при наявності косого зсуву, де $hx \in [0, 0.6]$: SIFT< BRISK< AKAZE< ORB,KAZE< ORB(1000),SURF(128),SURF(64);
4) при наявності центроафінних перетворень, де $b \in [0.6, 2]$, $hx \in [0, 0.6]$ та $\alpha \in [0,90]$: SIFT< BRISK< ORB< AKAZE, KAZE, ORB(1000), SURF(128),SURF(64);
5) за швидкодією дескриптори розташовуються наступним чином: ORB(1000), ORB< BRISK< SURF(64), SURF(128)< AKAZE, SIFT < KAZE.

15

ВИСНОВКИ

- Усі дескриптори SURF, SIFT, BRISK, ORB, KAZE та AKAZE є доцільними в використанні для нормалізації зображень за наявністю геометричних перетворень на основі характерних точок, а саме: однорідного масштабу, повороту та невеликих центроафінних перетворень, але найкращу точність показали детектори SIFT, ORB, BRISK.
- Також була проведена порівняльна характеристика швидкодії процесу нормалізації на основі розглянутих дескрипторів. Виявлено, що самими швидкими методами є ORB(1000), ORB, BRISK, а найдовше з всіх відпрацьовував KAZE.
- Було зроблено підсумкові висновки, що за співвідношенням точність і швидкодія мають перевагу дескриптори ORB і BRISK, які показують найкращу швидкодію та друге місце за точністю, однак, якщо є потреба у методі, найкращому за точністю, рекомендовано застосовувати дескриптор SIFT. Проведені дослідження на реальних зображеннях підтвердили усі перераховані вище висновки.
- У якості подальшого напрямку дослідження має сенс розглянути питання нормалізації зображень за наявністю більш значних геометричних перетворень, а також збільшення кількості зображень з 50 до 1000.

16