

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДУ ПОШУКУ ОБ'ЄКТА У ВІДЕОПОТОЦІ НА
ОСНОВІ ДЕСКРИПТОРІВ КЛЮЧОВИХ ТОЧОК ЗОБРАЖЕННЯ

(тема)

Виконав:
студент 2 курсу, групи ІНФМ-19-2

Зарівчацький Р.М.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Кобилін О.А.
(прізвище, ініціали)

2020 р.

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Зарівчацькому Роману Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методу пошуку об'єкта у відеопотоці на основі дескрипторів ключових точок зображення

затверджена наказом по університету від «23» _____ жовтня _____ 2020 року № 1428Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 02 _____ грудня _____ 2020 р.

3. Вихідні дані до роботи детектори ключових точок SIFT, SURF, BRISK, ORB; алгоритми градієнтного спуску, Куна-Манкреса, RANSAC; набір відео файлів з різною роздільною здатністю; операційна система Windows 7, частота процесору 2,5 ГГц; середовище розробки – Visual Studio 2017 Enterprise, програмна бібліотека OpenCV 2.4

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд сучасних детекторів ключових точок

2. Огляд і дослідження методів пошуку перетинів дескрипторів ключових точок.

3. Побудова програмної моделі знаходження об'єкта у відеопотоці

4. Тестування на даній моделі різних детекторів ключових точок зображення

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Виявлення ключової точки, Пошук ключової точки в масштабному просторі, Процес знаходження локальних особливостей, Приклади особливостей, що не розглядаються, Побудова дескриптора розміром 2×2×8, Примітиви Хаара, Контекстна діаграма верхнього рівня, Контекстна діаграма першого рівня, Діаграма функції обчислення вектора дескрипторів зображення об'єкта, Діаграма виділення кадру з відео потоку, Діаграма функції обчислення дескрипторів кадру, Діаграма функції знаходження областей претендентів, Чотирьохпараметрична модель пошуку, Блок-схема роботи методу пошуку об'єкта у відеопотоці, Структурна діаграма, Структура модуля обчислення вектора дескрипторів об'єкта, Структура модуля виділення кадру, Структура модуля обчислення вектора ключових точок кадру, Структура модуля пошуку областей претендентів, ROC криві результатів тестів інваріантності відносно повороту, ROC криві результатів тестів інваріантності відносно зміни масштабу, ROC криві результатів тестів інваріантності відносно нахилу камери.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	23.10.2020	виконано
2	Аналіз завдання, підбір літератури	24.10.20-27.10.20	виконано
3	Аналіз літератури з досліджуваної проблеми	27.10.20-05.11.20	виконано
4	Аналіз технічних засобів	06.11.20-09.11.20	виконано
5	Дослідження методу	10.11.20-14.11.20	виконано
6	Програмна реалізація	15.11.20-20.11.20	виконано
7	Оформлення пояснювальної записки	21.11.20-01.12.20	виконано
8	Перевірка на плагіат	02.12.20	виконано
9	Рецензування	04.12.20	виконано
10	Підготовка презентації та доповіді	06.12.20	виконано
11	Занесення роботи в електронний архів	07.12.20	виконано
12	Попередній захист атестаційної роботи	09.12.20	виконано

Дата видачі завдання 23 жовтня 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Творошенко І.С.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 75 с., 5 табл., 28 рис., 40 джерел.

ДЕТЕКТОР КЛЮЧОВИХ ТОЧОК, ДЕСКРИПТОР, SIFT, SURF, BRISK, ORB, RANSAC, ГРАДІЄНТНИЙ СПУСК, ДЕТЕКЦІЯ ОБ'ЄКТА, ВІДЕОПОСЛІДОВНІСТЬ.

Метою дослідження є аналіз методів пошуку об'єкта у відеопотоці, що базуються на використанні детекторів ключових точок зображення, та можливість їх застосування для вирішення реальних задач.

Об'єктом дослідження є набір відео послідовностей з різною роздільною здатністю.

В ході дослідження були проаналізовані сучасні методи пошуку об'єкта у відеопотоці. Було проведено дослідження детекторів ключових точок зображення на предмет швидкодії та стійкості до завад. Також були проаналізовані методи пошуку перетину дескрипторів ключових точок зображення.

В результаті дослідження була здійснена програмна реалізація методу пошуку об'єкта у відеопотоці, на основі детекторів ключових точок зображення.

KEY POINT DETECTOR, DESCRIPTOR, SIFT, SURF, BRISK, ORB, RANSAC, GRADIENT DESCENT, OBJECT DETECTION, VIDEO SEQUENCE.

The aim of the study is to analyze the methods of searching for an object in the video stream, based on the use of detectors of key image points, and the possibility of their use to solve real problems.

The object of the study is a set of video sequences with different resolutions.

The study analyzed modern methods of searching for an object in a video stream. The study of key image point detectors was carried out for speed and noise immunity. Methods for finding the intersection of descriptors of key image points were also analyzed.

As a result of the research, a software implementation of the method of searching for an object in a video stream was carried out, based on key point detectors.

ЗМІСТ

Вступ.....	7
1 Аналіз існуючих методів пошуку об’єкта у відеопотоці	8
1.1 Класифікація основних методів пошуку об’єкта у відеопотоці	8
1.1.1 Детерміновані методи.....	8
1.1.2 Імовірнісні методи	26
1.1.3 Нейромережеві методи	27
1.1.4 Комбіновані методи	29
1.2 Аналіз проблем пошуку об’єкта у відеопотоці.....	32
1.3 Порівняльний аналіз існуючих методів пошуку об’єкта у відеопотоці.....	33
1.4 Постановка задачі дослідження.....	34
2 Розробка методу пошуку об’єкта у відеопотоці на основі дескрипторів ключових точок зображення.....	35
2.1 Визначення вимог та проектування функціональної моделі пошуку об’єкта у відеопотоці.....	35
2.2 Методи знаходження перетину дескрипторів ключових точок зображення.....	43
2.3 Метод ідентифікації досліджуваної області зображення.....	45
2.4 Метод пошуку об’єкта у відеопотоці	49
2.5 Розробка структурної та функціональної моделей пошуку об’єкта у відеопотоці.....	52
3 Дослідження методу пошуку об’єкта у відеопотоці на основі дескрипторів ключових точок зображення.....	56
3.1 Вибір інструментальних засобів та інформаційних технологій для створення системи пошуку об’єкта у відеопотоці.....	56
3.2 Тестування розробленого програмного засобу.....	58
3.3 Результати дослідження методу пошуку об’єкта у відеопотоці на основі дескрипторів ключових точок зображення	61

	6
3.3.1 Тестування інваріантності до проєктивних спотворень	61
3.3.2 Порівняння методів за ймовірністю виявлення об'єкта	65
3.3.3 Порівняння швидкодії методів	67
Висновки.....	69
Перелік джерел посилання	70

ВСТУП

Комп'ютерний зір є напрямком сучасної науки, що динамічно розвиваються. Методи комп'ютерного зору застосовуються в різних прикладних областях: від автоматизованої медичної діагностики до безпілотного керування автомобілем, від глибоководних апаратів, до штучних космічних супутників. Можливість застосування комп'ютерного зору багатогранна, а вирішення задач в одній сфері може бути перенесено на абсолютно не пов'язану з нею прикладну область.

Невід'ємною частиною комп'ютерного зору є розпізнавання образів, вирішальним завданням, якого є визначення приналежності вхідного зображення до одного зі збережених еталонних зображень об'єктів. Розпізнаванню образів присвячено велику кількість робіт, проте «універсального» вирішення будь-якого завдання на даний момент не існує. Кожна прикладна область накладає свої специфічні умови. Виходячи з даних умов необхідно побудувати алгоритм, який вирішує завдання з максимальною швидкістю і точністю, але як правило, залишаються умови, які подолати набагато складніше, ніж інші. Наприклад, більшість методів не є стійкими до зміни інтенсивності або напряму освітлення, в той час, як інші погано справляються зі зміною масштабу і т.д.

При створенні інтелектуальних систем також часто потрібно відстежувати стан рухомих об'єктів в реальному часі на основі зорової інформації, отриманої від відеокамери. Маючи в своєму розпорядженні множину послідовних цифрових зображень (кадрів), можна виділити спеціальну інформацію про об'єкт і потім використовувати її для виявлення поточного стану об'єкта, його ідентифікації та відстеження його переміщень.

В рамках даної роботи досліджується метод пошуку об'єкта у відеопотоці на основі ключових точок зображення, його ефективність та можливість застосування для вирішення прикладних задач.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ПОШУКУ ОБ'ЄКТА У ВІДЕОПОТОЦІ

1.1 Класифікація основних методів пошуку об'єкта у відеопотоці

1.1.1 Детерміновані методи

Детерміновані методи видають унікальний і передбачуваний результат для заданих вхідних даних. Детерміновані методи розглядають об'єкт спостереження, як об'єкт з ознаками, що не змінюються з часом. Ці методи можна розділити на такі групи:

- методи пошуку за шаблоном;
- методи пошуку оптичного потоку;
- методи пошуку особливих точок.

Методи пошуку за шаблоном. Пошук об'єктів на підставі – деякого шаблону передбачає, що є зображення об'єкта з виділеними ознаками – шаблон – і тестове зображення, яке співставляється з цим шаблоном.

У найпростішому випадку в якості шаблону може виступати матриця інтенсивностей кольору, які є найбільш характерними для об'єкта [1]. Більш складні методи даної групи в якості шаблону використовують набори векторів ознак (дескриптори), геометричне представлення об'єкта або імовірнісні моделі об'єктів, які містять інформацію про розподіли інтенсивностей пікселів.

Під співставленням (matching) з шаблоном слід розуміти порівняння описів тестового і шаблонного зображень за деякою обраною метрикою (як правило, вибирається Евклідова відстань, норма $L1$, зважена згортка квадратичних похибок або кореляція).

Методи пошуку за шаблоном застосовують в основному в якості допоміжних методів, так як ці методи дозволяють виявити деякі геометричні примітиви. Відзначимо, що дані методи ефективно працюють при пошуку

поодиноких об'єктів, тому що при виникненні перекриттів, в описі зникають деякі ознаки.

До основних методів пошуку об'єкта за шаблоном можна віднести:

- детектори ребер;
- детектори кутів;
- детектори кіл.

З точки зору детектора прямих ліній, перетворення можна представити як знаходження суми яскравостей точок на контурному зображенні уздовж усіх можливих напрямків. Напрямки однозначно задаються перпендикулярними їм векторами, проведеними з центру картинної площини. Вектори задаються в полярній системі координат довжиною і кутом відносно вертикалі. Результатом перетворення є функція, що залежить від двох аргументів – кута і відстані. За значеннями функції можна визначити кількість точок, що лежать уздовж певної прямої лінії.

Перший етап даного методу передбачає виділення контурів. На другому етапі відбувається знаходження суми яскравостей точок вздовж прямої, заданої кутом і довжиною. Результатом знаходження сум яскравостей вздовж усіх прямих ліній є двовимірною функцією, що залежить від кута і відстані. Отримана функція не несе інформації про розташування відрізків на лінії, вона лише говорить, що він є, тому, додатково до описаних операцій, потрібна реалізація алгоритму локалізації відрізка на прямій.

До переваг даного методу слід віднести високу надійність знаходження прямих ліній. Розриви контурної лінії уздовж прямої мають незначний вплив на роботу алгоритму.

До недоліків слід віднести необхідність проведення операції знаходження контурів, пошуку областей перетину траєкторій окремих точок в багатовимірному просторі параметрів і необхідність додаткових алгоритмів, для локалізації відрізків на знайдених прямих лініях.

Один з перших алгоритмів запропонував Бедет [2]. Він визначає положення кутів по максимумах визначника Гессіана від функції яскравості зображення:

$$H = I_{xx}I_{yy} + I_{xy}^2.$$

Цей метод добре працює з кутами рівними 90 градусів. І так як в методі використовуються другі похідні від функції яскравості, то результат сильно залежить від впливу шуму. У свою чергу Форстнер запропонував детектор кутів, що використовує тільки перші похідні від функції яскравості, і визначив кути, як локальні максимуми [3]:

$$F(x, y) = \frac{\bar{I}_x^2 \cdot \bar{I}_y^2 - (\bar{I}_x - \bar{I}_y)^2}{\bar{I}_x^2 + \bar{I}_y^2},$$

де рисочки над змінними позначають середнє значення в деякій області точки (x, y) .

Недоліком детекторів кутів, що використовують компоненти градієнта яскравості, є те, що визначення самих компонент градієнта базується на диференціальних масках, моделях горизонтального і вертикального контрастного перепадів. Вони погано працюють в місцях розташування кутів, так як маска передбачає, що контрастний перепад може бути продовжений по прямій до нескінченності.

Очевидним методом знаходження кіл на зображенні є відстеження кривизни контурних ліній. Алі Айдар Радий та інші запропонували алгоритм швидкого пошуку кіл на зображенні, використовуючи протилежну спрямованість пари векторів градієнта, що лежать на протилежних кінцях кола, а також той факт, що їх бази лежать на прямій паралельній їм [4].

Метод перевершує за швидкістю метод Хука для пошуку кіл і є більш стійким до наявності шуму типу «сіль і перець» (шум у вигляді випадкових білих і чорних крапок).

Методи пошуку за шаблоном дозволяють виявити деякі геометричні примітиви на зображенні. Процес виявлення швидкий є швидким, але алгоритм зіставлення примітивів може бути дуже обчислювально складним, тому що необхідно враховувати взаємне розташування примітивів з об'єктом на зображенні, на якому здійснюється пошук.

Методи пошуку оптичного потоку. Оптичний потік є базовим інструментом для обробки і аналізу відеопослідовностей. Він містить важливу інформацію про структуру сцени. Існує кілька визначень оптичного потоку:

- векторне поле явного руху об'єктів (пікселів), поверхонь і ребер в візуальній сцені між кадрами, викликане відносним рухом між спостерігачем (оком, камерою) і сценою;
- зображення видимого руху об'єктів, поверхонь або країв сцени, що отримується в результаті переміщення спостерігача щодо сцени.

Можна виділити кілька основних груп методів визначення оптичного потоку:

- диференціальні методи оцінки;
- метод фазової кореляції (phase correlation);
- метод зіставлення блоків (block matching algorithms).

Найбільш широко на практиці застосовуються методи зіставлення блоків. Це обумовлено універсальністю, невисокою обчислювальною складністю, високою ефективністю, а також простотою апаратної реалізації методів цього типу.

Загальна процедура роботи цих методів така:

- поточний кадр розбивається на безліч блоків, що не перетинаються;
- для кожного блоку поточного кадру проводиться пошук найбільш схожого блоку в попередньому кадрі.

Різниця між позиціями поточного і попереднього блоку називається вектором руху поточного блоку.

Одним з найпопулярніших методів пошуку оптичного потоку є метод Лукаса-Канаде [5].

Метод Лукаса-Канаде – локальний метод обчислення оптичного потоку, що має лінійну обчислювальну складність. Основне рівняння оптичного потоку містить дві невідомі і не може бути однозначно вирішеним.

Даний метод обходить неоднозначність за рахунок використання інформації про сусідні пікселі в кожній точці. Метод Лукаса-Канаде заснований на припущенні, що в локальній околиці кожного пікселя значення оптичного потоку однаково. Розглянемо піксель p , тоді, по методу Лукаса-Канаде, оптичний потік повинен бути однаковий для всіх пікселів, що знаходяться у вікні з центром в точці p . Вектор оптичного потоку (V_x, V_y) в точці p визначається за формулою:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i \omega_i I_x(q_i)^2 & \sum_i \omega_i I_x(q_i) I_y(q_i) \\ \sum_i \omega_i I_x(q_i) I_y(q_i) & \sum_i \omega_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i \omega_i I_x(q_i) I_t(q_i) \\ -\sum_i \omega_i I_y(q_i) I_t(q_i) \end{bmatrix},$$

де q_1, q_2, \dots, q_n – пікселі всередині вікна;

$I_x(q_i), I_y(q_i), I_t(q_i)$ – часткові похідні зображення I по координатам x, y і часу t , обчислені в точці q_i ;

ω_i – вага, присвоєна пікселю q_i . В якості ваг ω_i зазвичай використовується нормальний розподіл відстані між q_i і p .

Метод Лукаса-Канаде є суто локальним і не може визначити напрямок руху пікселів всередині однорідних областей. Деякі зображення можуть давати вироджену матрицю A , для якої не може бути знайдена зворотна матриця, відповідно для таких зображень неможливо визначити зсув.

На сьогоднішній день метод Лукаса-Канаде і має безліч модифікацій [6]. У методі Томасі-Канаде рухом вважається зсув і розраховується шляхом

ітеративного вирішення побудованої системи лінійних рівнянь. Метод Ши-Томасі-Канаде враховує афінні спотворення. Метод Джин-Фавара-Соатто враховує зміни освітленості.

Методи пошуку ключових точок. Дана група методів базується на понятті особливих точок – локальних особливостей зображення. Процес пошуку особливих точок здійснюється за допомогою детектора.

Особлива точка, або особливість – це точка зображення, що має ряд властивостей:

1. *Визначеність (distinctness)* – особливість повинна виділятися на тлі сусідніх точок.

2. *Стійкість (repeatability)* – зміна яскравості, контрастності і колірної гами не повинні впливати на місце особливої точки на об'єкті або сцені.

3. *Інваріантність (invariance)* – особливі точки повинні володіти стійкістю до повороту, зміни масштабу зображення і зміни ракурсу зйомки.

4. *Стабільність (stability)* – зашумленість зображення, що не перевищує певний поріг, не повинна впливати на роботу детектора.

5. *Інтерпретованість (interpretability)* – особливі точки повинні бути представлені в форматі, придатному для подальшої роботи.

6. *Кількість (quantity)* – кількість виявлених особливих точок має бути достатньою для виявлення об'єктів.

Дескриптор (від лат. *Descriptor* – той, що описує) – опис особливої точки, що визначає особливості її околу, та являє собою числовий або бінарний вектор певних параметрів. Довжина вектора і вид параметрів визначаються алгоритмом, що застосовується. Дескриптор дозволяє виділити особливу точку з їх множини на зображенні. Це необхідно для складання ключових пар особливостей, що належать одному об'єкту, при порівнянні різних зображень.

До популярних детекторів ключових точок можна віднести:

- ORB;
- BRISK;

- SIFT;
- SURF.

Алгоритм ORB використовує вдосконалений алгоритм виявлення ключових точок зображення FAST. Ідея даного алгоритму полягає в тому, що якщо піксель суттєво відрізняється від пікселів по-сусідству тоді він, швидше за все, є кутовим пікселем, а отже, може бути ключовою особливістю даного зображення.

Процес знаходження ключових в даному методі, має такі складові:

а) виявлення ключової точки зображення. Спочатку на зображенні вибирається піксель p з яскравістю I_p . Встановлюється поріг яскравості T . Потім вибирається 16 пікселів на колі з радіусом 3 і центром в точці p і порівнюються значення інтенсивності між пікселем p та іншими пікселями по колу (рис. 1.1). Якщо яскравість послідовних N точок на вибраному колі більше $I_p + T$ або менше $I_p - T$, тоді піксель p може розглядатися як особлива точка;

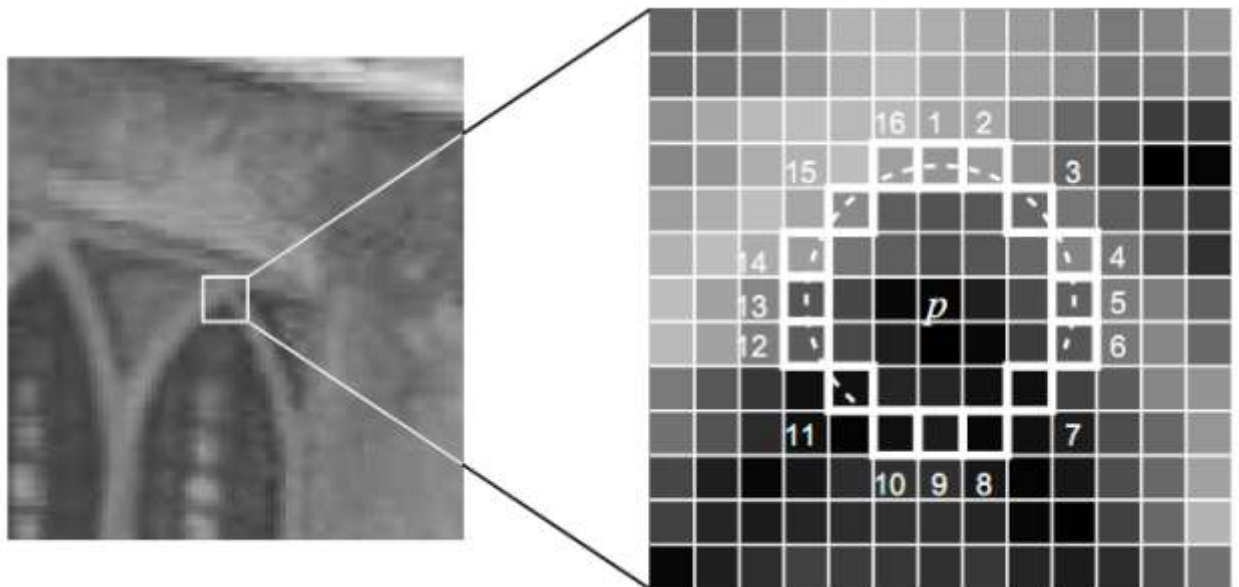


Рисунок 1.1 – Виявлення ключової точки

б) скрінінг особливих точок. Оскільки розрахунок кутової точки за методом FAST полягає лише в порівнянні різниці в яскравостей між

пікселями, то кількість потенційних ключових точок буде занадто великою, а для кожної такої точки буде відсутня інформація про її напрямок. Алгоритм ORB вдосконалює вихідний алгоритм FAST, який обчислює значення міри Харріса для вихідних кутових точок, сортуючи їх відповідно до значення інтенсивності кольору і беручи перші N точок [7]. Формули розрахунку значення міри Харріса:

$$R = \det(M) - k(\text{trace}(M))^2,$$

$$M = \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix},$$

де R – міра Харріса;

M – матриця 2×2 ;

$k \in [0,04; 0,06]$;

$w(x, y)$ – функція вікна;

I_x – варіація ключової точки по горизонталі;

I_y – варіація по вертикалі;

в) побудова пірамід масштабу. Піраміди масштабу зображення будуються для досягнення інваріантності до зміни масштабу зображення;

г) визначення напрямку характерної точки. Для того, щоб добути особливі точки мали інваріантність до обертання, отримують напрямок даних точок за допомогою знаходження центрального моменту зображення.

Завдяки вищевказаним крокам, отримані характерні точки мають інваріантність до зміни масштабу та обертання зображення, що значно покращує їх стійкість.

Після знаходження зорієнтованих характерних точок, алгоритм ORB використовує вдосконалений алгоритм BRIEF, для обчислення дескрипторів кожної точки. Даний дескриптор являє собою бінарний вектор, що складається лише з 0 та 1:

$$\tau(p; x, y) = \begin{cases} 1, & p(x) < p(y), \\ 0, & p(x) \geq p(y), \end{cases}$$

де $p(x)$ – значення інтенсивності в полі x навколо ключової точки зображення;

$p(y)$ – значення інтенсивності в полі x навколо ключової точки зображення.

Оскільки оригінальний дескриптор BRIEF не має інваріантності до обертання, легко втратити дані для поверненого зображення. Тому алгоритм ORB використовує алгоритм Steer BRIEF для обчислення основного напрямку кожної характерної ознаки, щоб дескриптор мав інформацію про напрямки ключової точки.

BRISK. Даний метод є поєднанням алгоритму AGAST (модифікація FAST) та дескриптора BRIEF [8]. З метою досягнення інваріантності до масштабу, яка має вирішальне значення для якісних ключових точок, у детекторі BRIEF, на відміну від AGAST виконується пошук максимумів не тільки в площині зображення, але і в масштабному просторі. Незважаючи на дискретизацію осі шкали за допомогою більш грубих інтервалів, ніж в альтернативних високопродуктивних детекторах (наприклад, Fast-Hessian), детектор BRISK оцінює справжній масштаб кожної ключової точки в безперервному масштабному просторі.

У методі BRISK багатомасштабний простір зображення складається з n октав c_i і n внутрішніх октав d_i , $i = \{0, 1, \dots, n - 1\}$. n зазвичай вибирається рівним 4. Октави знаходяться шляхом зменшення масштабу вихідного зображення в два рази (рис. 1.2).

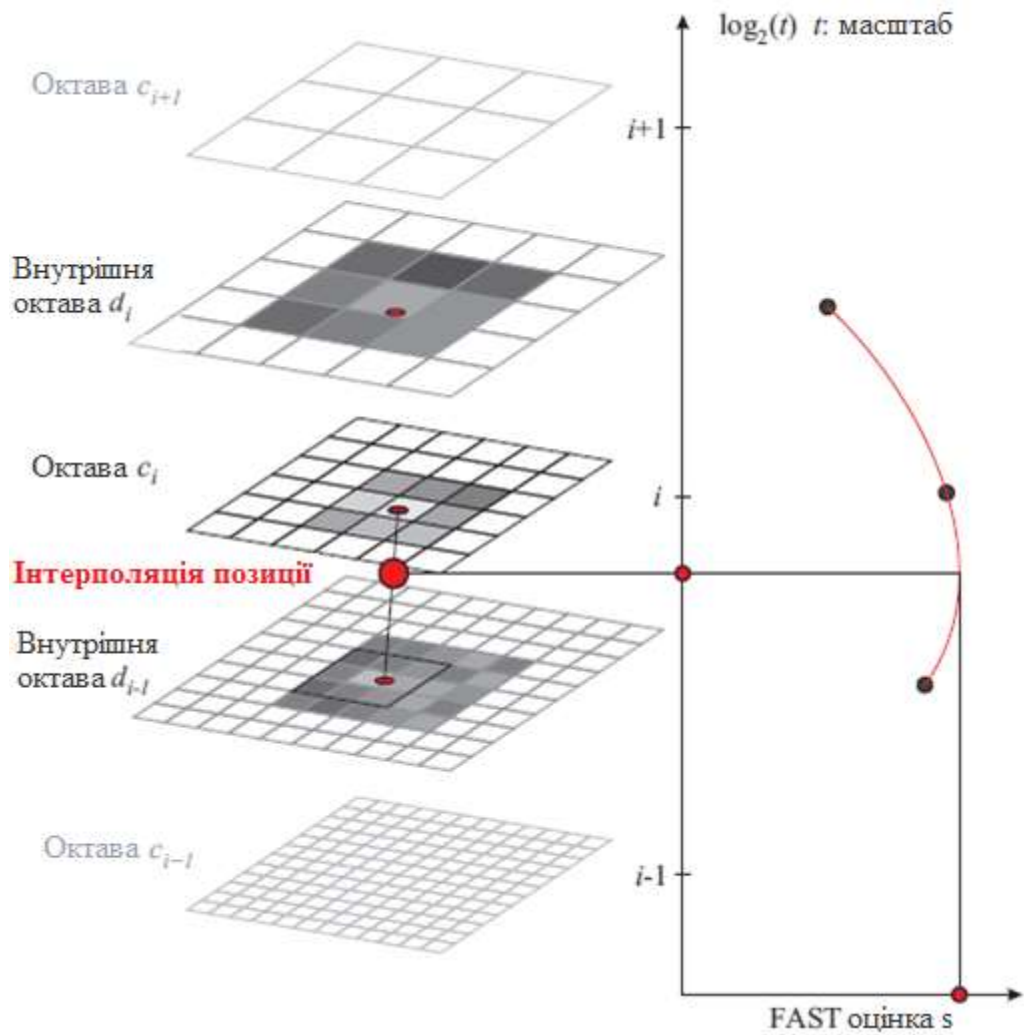


Рисунок 1.2 – Пошук ключової точки в масштабному просторі

Для формування дескриптора, нормованого для обертання та масштабування зображення, метод BRISK застосовує шаблон вибірки, повернутий на $\alpha = \arctan^2(g_y, g_x)$ навколо ключової точки k . Бінарний вектор (дескриптор) b_k обчислюється шляхом виконання порівнянь інтенсивності усіх пар точок $(p_i^\alpha, p_j^\alpha) \in S$, відстань між якими є такою, що кожен біт b відповідає умові:

$$b = \begin{cases} 1, \text{ якщо } I(p_i^\alpha, p_j^\alpha) > I(p_i^\alpha, p_k^\alpha), \\ 0, \text{ в іншому випадку} \end{cases},$$

$$\forall (p_i^\alpha, p_j^\alpha) \in S.$$

Хоча дескриптор BRIEF також обчислюється за допомогою порівняння яскравості, BRISK має деякі основні відмінності, окрім очевидного попереднього масштабування та обертання моделі вибірки.

По-перше, BRISK використовує детермінований шаблон вибірки, що приводить до єдиної щільності точок вибірки при заданому радіусі навколо ключової точки.

Крім того, BRISK використовує значно меншу кількість проб відбору точок ніж попарне порівняння, обмежуючи складність пошуку значень інтенсивності.

SIFT. Ідея методу SIFT зводиться до знаходження зображення об'єкта через унікальні локальні ознаки даного об'єкта [9]. Метод SIFT можна розділити на наступні етапи:

- визначення локальних особливостей (точок інтересу або ключових точок);
- локалізація особливостей;
- обчислення орієнтацій особливостей;
- опис локальних особливостей через дескриптор;
- зіставлення дескрипторів.

Для визначення особливостей використовується Гаусовий простір, що масштабується. Побудова простору, що масштабується являє собою операцію розкладання вихідного сигналу в сімейство версій сигналу, що поступово згладжуються та спрощуються, а також задовольняють вимогам лінійності, інваріантності до зсуву, масштабу і володіють властивістю напівгрупи та позитивності. Доведено, що Гаусовий простір, що масштабується, є лінійним, інваріантним до зсуву та масштабу. У методі SIFT область на зображенні є особливою, якщо вона є локальним екстремумом в масштабованому просторі (піраміді) різниць Гаусіанів. На етапі визначення характерних точок будуються піраміди Гаусіанів і різниці Гаусіанів (*DoG – Difference of Gaussian*) (рис. 1.3):

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

де L – значення Гаусіана в точці з координатами (x, y) ;

σ – радіус розмиття;

G – гаусове ядро;

I – значення вихідного зображення;

* – операція згортки.

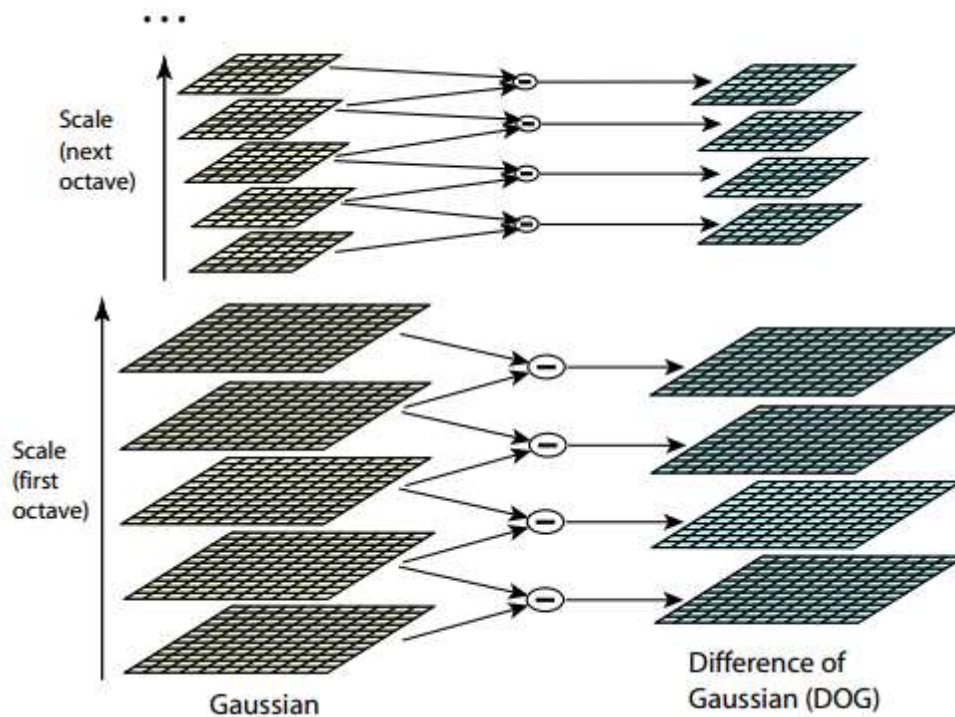


Рисунок 1.3 – Процес знаходження локальних особливостей

Гаусове ядро обчислюється за такою формулою:

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

Різниця Гаусіан DoG обчислюється таким чином:

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

де k – коефіцієнт зміни радіусу розмиття σ .

Після обчислення різниці Гаусіан відбувається передискретизація вихідного зображення: зменшується частота дискретизації в 2 рази по вертикалі і горизонталі. Потім над зменшеним зображенням обчислюють різниці Гаусіан. Таким чином, будується піраміда різниці Гаусіан до тих пір, поки розміри зображення не будуть менше певного порогу по ширині або по висоті, або поки передискретизацію НЕ буде зроблено максимально допустиму кількість разів. Після побудови піраміди знаходяться екстремуми шляхом порівняння кожної точки з вісьмома сусідами на поточному зображенні DoG (якщо такі є) і з дев'ятьма сусідами зображень DoG вище і нижче в піраміді.

Після визначення особливостей проводиться локалізація і перевірка особливостей на контрастність і на приналежність до кордону між областями зображення, тобто виключаються з подальшого розгляду особливості, які мають контрастність нижче певного порогу і які знаходяться на кордоні між об'єктами (рис. 1.4).

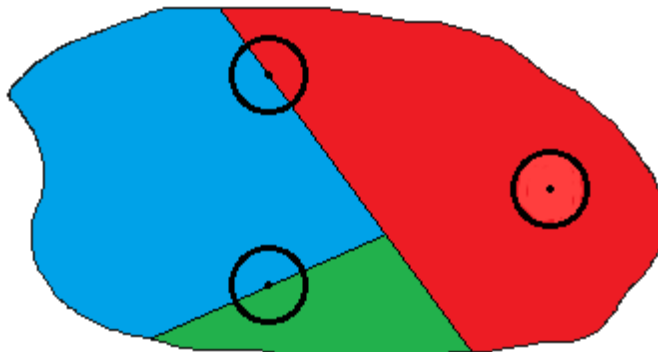


Рисунок 1.4 – Приклади особливостей, що не розглядаються

На початку етапу локалізації визначаються координати ключових точок з субпіксельною точністю, для цього апроксимується функція DoG многочленом Тейлора другого порядку в знайденому екстремумі:

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

де x – зміщення;

$$x = (x, y, \sigma).$$

Обчисливши похідну і прирівнявши даний вираз до нуля, отримаємо екстремум \hat{x} :

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}.$$

Якщо зсув \hat{x} більше, ніж 0,5, то це означає, що екстремум лежить ближче до іншої точки вибірки. У цьому випадку точка вибірки змінюється, і виконується інтерполяція заново для зміненої точки. Підсумкове зміщення \hat{x} додається до місця точки, щоб отримати інтерпольовану оцінку для розташування екстремуму.

Після обчислення екстремуму з субпіксельною точністю, обчислюється значення DoG в точці \hat{x} :

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

Значення $D(\hat{x})$ порівнюється з граничним значенням $D_{min} = 1$. Якщо $|D(\hat{x})| < D_{min}$, то особливість вважається такою, що має низьку контрастність і далі не розглядається.

Перевірка особливостей на приналежність до кордону між областями зображення здійснюється через обчислення матриці Гессе H [10]:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}.$$

Для знаходження орієнтації ключової точки обчислюються напрямки і величини градієнтів у вікні з радіусом 3σ в центрі особливості:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2},$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right),$$

де m – величина градієнта;

θ – напрямок градієнта.

Після обчислення напрямків градієнтів будується гістограма напрямків *Hist*, що складається з 36 компонент:

$$w_k = m(x, y) * G(x, y, 1.5\sigma),$$

де w_k – вага градієнта в точці (x, y) для k -ої компоненти *Hist*.

Напрямок градієнта вибирається з найближчого напрямку k гістограми *Hist*.

Напрямок особливості перебуває в проміжку, що покривається максимальною компонентою гістограми *Hist*. Максимальне значення і два сусідні значення інтерполюються параболою, точка максимуму параболі береться в якості напрямку особливості. Якщо в гістограмі є ще компоненти зі значеннями не менш 80% від максимального напрямку, то вони аналогічно інтерполюються і додаткові напрямки додаються до ключової точки.

Після визначення орієнтації вікно з радіусом 3σ повертають в центрі особливості в напрямку орієнтації – цим досягається інваріантність до повороту. Потім вся область у вікні ділиться на 16 однакових квадратних блоків (4×4), в яких обчислюються гістограми напрямків.

Дескриптор особливості складається з отриманих гістограм (рис. 1.5). Розмірність дескриптора оригінального методу SIFT становить $4 \times 4 \times 8 = 128$ елементів. Після опису дескриптора через вектор, що складається з 128 елементів, дескриптор нормалізується.

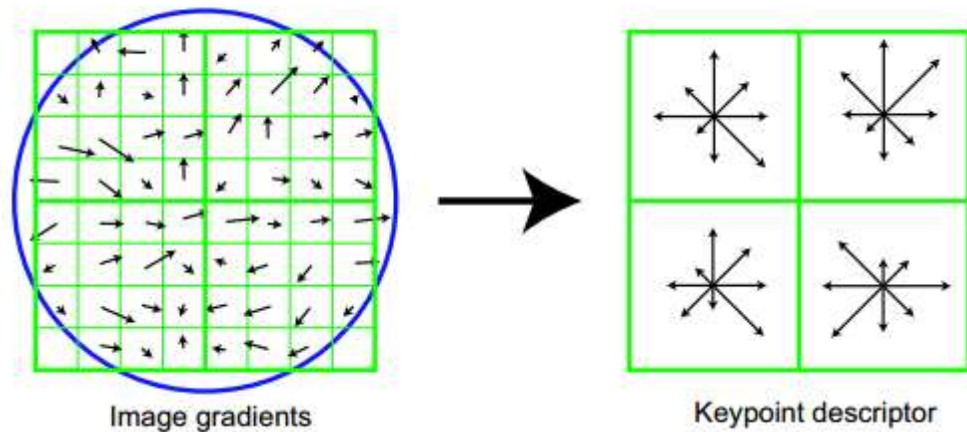


Рисунок 1.5 – Побудова дескриптора розміром $2 \times 2 \times 8$

SURF. Метод SURF є аналогом методу SIFT. Даний метод також шукає ключові точки і будує опис знайдених точок через дескриптори особливостей [11].

Ключова точка в SURF – це локальний екстремум детермінанта матриці Гессе. Для двовимірного випадку детермінант матриці Гессе визначається наступним чином

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

де H – матриця Гессе;

f – функція зміни градієнта.

На рисунку 1.6 показані фільтри компонент D_{xx} , D_{yy} , D_{xy} матриці Гессе H розміром 9×9 . Метод SURF використовує апроксимацію даних фільтрів.

Такі фільтри є стійкими до обертання і їх можна ефективно обчислити за допомогою інтегральної матриці.

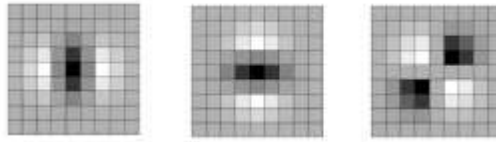


Рисунок 1.6 – Фільтри компонент D_{xx}, D_{yy}, D_{xy} матриці Гессе H

Детермінант матриці Гессе для фільтрів, зображених на рисунку 1.7 обчислюється за такою формулою:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0,9D_{xy})^2.$$

Гессіан є інваріантним до повороту, але не до масштабу, тому для визначення Гессіанів використовуються фільтри з різним масштабом.

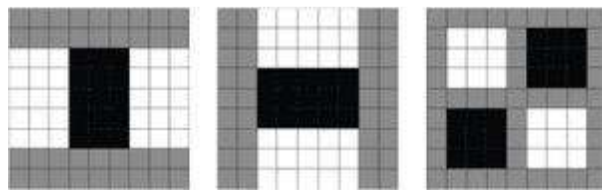


Рисунок 1.7 – SURF фільтри компонент D_{xx}, D_{yy}, D_{xy} матриці Гессе

Для визначення орієнтації особливості використовуються фільтри Хаара (рис. 1.8), розміри яких дорівнюють $4s$, де s – масштаб особливої точки. Значення вейвлета Хаара dX і dY для кожної точки множаться на значення Гаусіана з центром в особливій точці і сигмою рівною $2,5s$. Завдяки такому зважуванню, відсіюються випадкові перешкоди на далеких відстанях від ключової точки. Потім за допомогою кутового вікна вибирається напрямок особливості, при якому довжина сумарного вектора для тих точок, які потрапили у вікно, буде максимальна.



Рисунок 1.8 – SURF фільтри компонент D_{xx}, D_{yy}, D_{xy} матриці Гессе

Для обчислення дескриптора особливості формується прямокутна область (квадрат) уздовж пріоритетного напрямку розміру $20s$, де s – масштаб в якому була знайдена особливість. Квадрат розбивається на 16 квадратних областей, в кожній з яких береться регулярна сітка 5×5 . В кожній такій області обчислюється градієнт, за допомогою фільтра Хаара.

Після знаходження 25 точкових градієнтів областей, обчислюються чотири величини: $\sum dX, \sum |dX|, \sum dY, \sum |dY|$. На рисунку 1.9 наведені приклади поведінки цих величин для різних ділянок зображень.

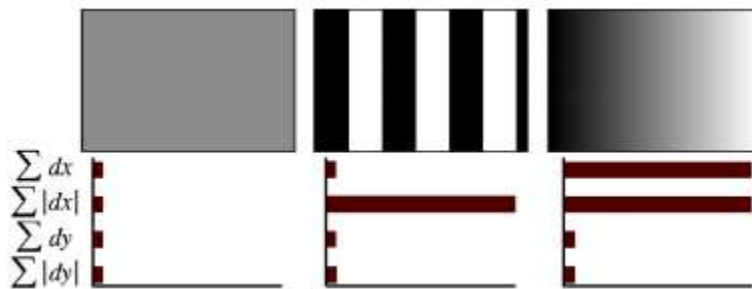


Рисунок 1.9 – Поведінка величин $\sum dX, \sum |dX|, \sum dY, \sum |dY|$ в різних ситуаціях

У разі, коли область однорідна, то всі коефіцієнти відносно малі. Для вертикальних повторюваних смужок усі величини крім $\sum |dX|$ малі. У разі збільшення яскравості уздовж осі x , величини $\sum dX, \sum |dX|$ мають великі значення відносно $\sum dY, \sum |dY|$.

Таким чином, для 16 областей отримується 64 значення величин, які і є компонентами дескриптора. Всі значення величин зважуються на Гаусіану, з центром в особливій точці і з сигмою $3,3 s$.

1.1.2 Імовірнісні методи

Імовірнісні методи використовують підхід, що базується на понятті простору станів. Вважається, що об'єкт, що рухається, має певний внутрішній стан, який вимірюється на кожному кадрі. У найпростішому випадку під станом розуміється стан об'єкта на зображенні. Щоб оцінити наступний стан об'єкта, потрібно максимально узагальнити отримані вимірювання, тобто визначити новий стан за умови, що отриманий набір вимірів для станів на попередніх кадрах. Типовими прикладами таких методів є методи на базі фільтра Кальмана і фільтра частинок.

Фільтр Калмана застосовують якщо відомий початковий стан об'єкта на зображенні, іншими словами застосовують для вирішення завдання супроводу об'єкта. Завдання супроводу можна розглядати як добре вивчену проблему теорії управління, яка полягає в тому, щоб оцінити стан системи на підставі послідовності зашумлених вимірювань.

Алгоритм складається з двох повторюваних фаз: передбачення і коригування. Протягом першої фази розраховується прогноз стану в наступний момент часу (з урахуванням неточності їх вимірювання). Протягом другої, нова інформація з датчика коригує передбачене значення (також з урахуванням неточності і зашумленості цієї інформації).

Для деяких практичних завдань, щоб отримати більш точну оцінку стану системи, необхідно відійти від припущення, що шум має Гауссовий розподіл [12]. У цьому випадку вводиться поняття мультимодального розподілу шуму, а для моделювання подібних систем використовуються фільтри частинок. Фільтри частинок є більш загальним підходом до вирішення завдання супроводу з застосуванням імовірнісних методів. Мультимодальним розподілом називається розподіл, що має кілька мод або локальних максимумів. Мультимодальний розподіл часто є сумішшю декількох розподілів.

Алгоритм відтворення умовної щільності (CONditional DENsity propAGATION, CONDENSATION) – базовий алгоритм фільтрації частинок, на підставі якого будується більшість алгоритмів даної групи, що застосовуються в комп'ютерному зорі.

Таким чином, імовірнісні методи спостереження дозволяють прогнозувати стан об'єкта на зображенні без збереження усіх даних про попередні стани та сприймати об'єкт зі змінними ознаками в відеопослідовності.

Імовірнісні методи застосовують на практиці в якості додаткових методів для підвищення стійкості до зміни зображення об'єкта [13].

1.1.3 Нейромережеві методи

У нейромережевих методах об'єкт в відеопотоці може розглядатись одночасно, як об'єкт з ознаками, що змінюються та об'єкт з незмінними ознаками. У цих методах невід'ємною частиною є етап навчання нейронної мережі [14]. Навчати мережу необхідно під кожен тип завдань.

Класична нейронна мережа. Основна ідея, що лежить в основі нейронних мереж – це послідовне перетворення сигналу, за допомогою паралельно працюючих елементарних функціональних елементів, нейронів [15]. Основний принцип налагодження нейронної мережі полягає в застосуванні оптимізаційних методів до мінімізації середньоквадратичної помилки, як наслідок – схильність до перенавчання. Головна перевага нейронних мереж – гнучкість.

Геометрично, поверхня, що розділяє класи, є множиною гіперплощин. Кожна з областей, на які гіперплощини розбивають простір ознак X відноситься до одного з класів. Існує безліч методів навчання нейромереж, проте всі вони зводяться до мінімізації середньоквадратичної помилки. Важливо відзначити, що знайдений мінімум, буде локальним. Також слід

зазначити, що вірно класифіковані прецеденти не вносять ніяких змін в функціонал, що оптимізується. Таким чином, знайдена розділяюча поверхня не буде ні єдиним, ні оптимальним рішенням.

Системи розпізнавання об'єктів на зображеннях засновані на нейронних мережах використовують ієрархічну архітектуру. Спочатку вектор ознак обробляється грубою мережею з високим рівнем помилок другого роду, далі, якщо вектор не класифікований як не об'єкт, розв'язок коригується більш точною і повільнішою мережею. В цілому нейронні мережі схильні до перенавчання, хоча і існують деякі методи, які в окремому випадку можуть вирішити цю проблему. Стійкість до шуму сильно залежить від конкретної архітектури мережі. У загальному випадку, нейромережа чутлива до шуму. Обчислювальна складність квадратично залежить від числа нейронів в прихованому шарі. Кожен нейрон вимагає обчислення функції активації. Для задач розпізнавання об'єктів на зображеннях швидкість обробки є недостатньою для вирішення задач в реальному часі.

SNoW – розріджена просіююча мережа. SNoW (Sparse network of Windows) – особливий вид нейронної мережі. Вектор ознак покладається бінарним. Дана мережа складається з двох (по числу можливих класів) лінійних нейронів, пов'язаних з компонентами вектора ознак [16]. Класифікація проходить за принципом переможець забирає все.

Геометрично, SNoW являє собою дві гіперплощини в просторі векторів ознак. Вектор відноситься до того класу, гіперплощина найближча. Таким чином, результуюча розділяюча поверхня є гіперплощиною в вихідному просторі X . Одна з переваг даної архітектури – можливість «проріджувати» вектор ознак на основі навчальної вибірки – компоненти вектора, що не несуть інформації, відкидаються.

SNoW вважається досить ефективним методом для вирішення завдання виявлення об'єктів на зображеннях. За рахунок просіювання компонент вектора ознак досягається висока швидкість – складність лінійна щодо кількості ефективних компонент вектора ознак.

Таким чином, нейронні мережі мають високий відсоток розпізнавання об'єкта на зображенні, низький відсоток помилкового розпізнавання, але необхідно враховувати, що нейронну мережу необхідно навчати під кожен тип завдань. Методи пошуку об'єкта з застосуванням класичних нейронних мереж мають низьку швидкість, на відміну від розріджених мереж.

1.1.4 Комбіновані методи

Особливість комбінованих методів полягає в тому, що вони складаються з декількох методів, комбінуючи їх за найвищими показниками різних критеріїв. Такі методи більш стійкі до шуму, до різних видів спотворень об'єкта. Комбіновані методи можуть поєднувати в собі детерміновані, імовірнісні, нейромережеві методи. Комбіновані методи можна розбити на дві групи: методи з учителем і методи без вчителя.

Метод Віоли-Джонса. Метод був розроблений і представлений в 2001 році Полом Віолою і Майклом Джонсом. Цей метод відноситься до методів з учителем. Етап навчання відбувається дуже повільно, але зате пошук відбувається дуже швидко.

Метод Віоли-Джонса використовує ознаки Хаара. Ознака – це відображення $f: X \Rightarrow D_f$, де D_f – множина допустимих значень ознаки. Якщо задані ознаки f_1, \dots, f_n , то вектор ознак $x = (f_1(x), \dots, f_n(x))$ називається описом об'єкта $x \in X$. Дані описи допустимо ототожнювати з самими об'єктами. При цьому множина $X = D_{f_1} * \dots * D_{f_n}$ називають простором ознак.

Ознаки поділяються на такі типи в залежності від множини D_f :

- бінарна ознака, $D_f = \{0,1\}$;
- номінальна ознака: D_f – скінченна множина;
- порядкова ознака: D_f – скінченна впорядкована множина;

– кількісна ознака: D_f – множина дійсних чисел.

Для кожного типу завдань необхідно підбирати ознаки індивідуально – від цього залежить якість пошуку. У стандартному методі Віоли-Джонса використовуються прямокутні ознаки (рис. 1.10). Ці ознаки називаються примітивами Хаара.

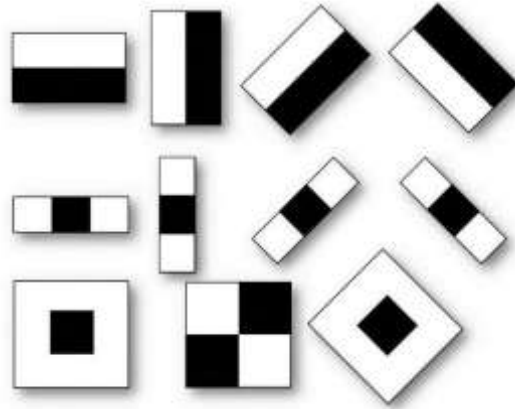


Рисунок 1.10 – Примітиви Хаара

Швидке обчислення ознак досягається за допомогою інтегрального представлення зображення. Інтегральне представлення дозволяє швидко розраховувати сумарну яскравість довільного прямокутника на даному зображенні, причому час розрахунку не залежить від розмірів прямокутника. Інтегральне представлення зображення – це матриця, що збігається за розмірами з вихідним зображенням. У кожному її елементі зберігається сума інтенсивностей всіх пікселів, що знаходяться лівіше і вище даного елемента. Кожен елемент даної матриці являє собою суму пікселів в прямокутнику від $(0,0)$ до (x, y) , тобто значення кожного пікселя (x, y) дорівнює сумі значень всіх пікселів лівіше і вище даного пікселя. Розрахунок матриці займає лінійний час, пропорційний числу пікселів в зображенні, тому інтегральне зображення прораховується за один прохід. За допомогою інтегральної матриці можна дуже швидко вирахувати суму пікселів довільного прямокутника, довільної площі.

Для вибору ознак використовується алгоритм бустінга. Даний алгоритм був опублікований в 1996 р. і послужив основою для всіх подальших досліджень в даній області [17].

На даний момент найбільш поширеними варіантами базового алгоритму є Gentle AdaBoost і Real AdaBoost, що перевершують базовий алгоритм за своїми характеристиками, але зберігають всі основні принципи.

Метод Віоли-Джонса використовує підхід на основі скануючого вікна (scanning window): сканується зображення вікном пошуку, а потім застосовується класифікатор до кожного положення [18]. Система навчання і вибору найбільш значущих ознак повністю автоматизована і не вимагає втручання людини, а тому цей підхід працює досить швидко.

Метод TLD. Даний метод дозволяє надійно супроводжувати заздалегідь невідомі об'єкти в природному середовищі протягом тривалого часу [19]. Метод витримує розриви між кадрами, швидкий рух камери, повне зникнення, а потім поява об'єкта. Підхід, який використаний в даному методі називається Супроводження-Моделювання-Виявлення (Tracking-Modeling-Detection (TMD)) [20], він поєднує адаптивний супровід об'єкта з навчанням детектора об'єкта в процесі розпізнавання. Після того як об'єкт був захоплений за допомогою будь-якого методу захоплення, траєкторія об'єкта починає спостерігатися двома процесами. Вони будують детектор об'єкта на льоту. Метод TLD є методом без учителя, але процес навчання все ж є. Навчання відбувається так само «на льоту», тобто в процесі виявлення об'єкта. Об'єкт супроводжується за допомогою короткострокового трекера. Траєкторія в просторі ознак аналізується двома подіями, які безперервно намагаються розширити або зменшити простір, що описується моделлю. Короткостроковий трекер ґрунтується на методі Лукаса-Канаде [21].

Спочатку множина ключових точок витягується з прямокутної решітки всередині прямокутника, описаного навколо об'єкта, що супроводжуваного. Потім трекер супроводжує точки від одного кадру до іншого, будуючи розріджене поле руху.

Ґрунтуючись на полі руху, зміщення і зміни масштабу прямокутника можуть бути надійно оцінені як середні значення за розподілом. У кожному новому кадрі супроводжується новий набір точок, це робить трекер адаптивним.

Таким чином, комбіновані методи мають високий рівень виявлення, дозволяють об'єднувати переваги різних методів пошуку. В комбінованих методах складною частиною є поєднання різних методів.

1.2 Аналіз проблем пошуку об'єкта у відеопотоці

Таким чином можна виділити основні проблеми процесу виявлення і стеження за об'єктом в відеопотоці:

1. Зміна яскравості. Яскравість зображення може змінюватися в просторі і протягом часу: деякі частини об'єкта можуть бути яскравішими за інші; яскраві об'єкти на поточному зображенні можуть стати темними на наступному.

2. Зміна масштабу об'єкта. Об'єкт може наближатися і віддалятися від камери.

3. Повороти об'єкта. Об'єкт може обертатися по всіх трьох осях тривимірного простору відносно камери.

4. Перекривання об'єкта. Об'єкт пошуку може перекриватися іншими об'єктами.

5. Шум на зображенні. Шум на зображенні може виникнути з різних причин: природних і штучно створених. До природним можна віднести некоректну роботу фото (відео) камери і всі природні умови, за яких відбувається отримання зображення (відео): снігопад, дощ і т.д. До штучно створених можна віднести всі ті дії людини, які спрямовані на зміну якості зображення: перекодування зображення в іншу колірну схему; стиснення зображення з втратами якості, розмиття й т.д.

6. Зміна геометричної форми об'єкта. Об'єкт спостереження може деформуватися, повернутися до камери тією частиною, яка за своєю формою є відмінною від форми, що спостерігається раніше.

7. Зміна колірної гистограми об'єкта. Об'єкт може мати на різних сторонах свого тіла різні гистограми кольору.

1.3 Порівняльний аналіз існуючих методів пошуку об'єкта у відеопотоці

Згідно з постановленою метою і описаними проблемами виявлення і стеження за об'єктом в відеопотоці, можна виділити основні критерії для порівняння:

- стійкість до зміни яскравості;
- стійкість до афінних перетворень;
- стійкість до закриття об'єкта іншими об'єктами;
- стійкість до шуму на зображенні;
- здатність до пристосування до зміни геометричної форми об'єкта;
- здатність до пристосування до зміни гистограми кольору об'єкта;
- швидкодія.

Таким чином, всі розглянуті методи в деяких межах стійкі до шуму і до перекриття об'єкта іншими об'єктами.

З детермінованих методів найстійкішим до розглянутих критеріїв є метод SIFT, однак він має велику обчислювальну складність. Також із даної групи варто відзначити методи SURF та BRISK, що враховують зміну масштабу зображення об'єкта та мають меншу обчислювальну складність, ніж метод SIFT, проте вони не є повністю стійкими до афінних перетворень. Найбільшою швидкістю володіє метод ORB, проте він також є чутливим до афінних перетворень.

Імовірнісні методи необхідно застосовувати у випадках, коли необхідно домогтися стійкості по якомусь одному критерію. Можна застосувати кілька фільтрів, стійких до зміни різних властивостей об'єкта, але тоді і складність етапу виявлення геометрично зростає. Методи, які стійкі до найбільшої кількості розглянутих критеріїв, є нейромережеві і комбіновані методи, але методи, засновані на класичних і розріджених нейронних мережах, і метод Віоли-Джонса мають тривалий процес навчання, який є набагато складнішим ніж сам етап виявлення.

1.4 Постановка задачі дослідження

На сьогоднішній день велику популярність здобули системи комп'ютерного зору. Автопілоти на автомобілях, системи відеонагляду, «розумні» будинки – це те, що оточує нас кожного дня. Ключовим аспектом функціонування усіх цих речей є детекція об'єктів у відеопотоці.

Об'єктом дослідження є набір відеопослідовностей з різною роздільною здатністю.

Метою дослідження є аналіз методів пошуку об'єкта у відеопотоці, що базуються на використанні детекторів ключових точок зображення, та можливість їх застосування для вирішення реальних задач.

Для досягнення поставленої мети необхідно вирішити наступні питання:

- розглянути сучасні детектори ключових точок;
- розглянути і дослідити методи пошуку перетинів дескрипторів ключових точок;
- побудувати програмну модель знаходження об'єкта у відеопотоці;
- протестувати на даній моделі різні детектори ключових точок зображення.

2 РОЗРОБКА МЕТОДУ ПОШУКУ ОБ'ЄКТА У ВІДЕОПОТОЦІ НА ОСНОВІ ДЕСКРИПТОРІВ КЛЮЧОВИХ ТОЧОК ЗОБРАЖЕННЯ

2.1 Визначення вимог та проектування функціональної моделі пошуку об'єкта у відеопотоці

Згідно зі сформульованою метою і проведеним аналітичним оглядом можна визначити вимоги до системи пошуку об'єкта в потоці. Ці вимоги можна розділити на дві групи: основні (функціональні) і додаткові (структурні).

Основні вимоги до системи пошуку об'єкта в відеопотоці формуються на підставі результатів аналітичного огляду і потреб практичного використання:

- алгоритм системи повинен бути заснований на пошуку ключових точок об'єкта, адже швидкість їх роботи є вищою, аніж швидкість роботи інших алгоритмів;
- обчислювальна складність методу пошуку повинна бути мінімально досяжною для застосування в рішенні задач в реальному часі;
- алгоритм роботи системи не повинен містити крок попереднього навчання;
- система повинна дозволяти керувати способом обробки результату: зберігати результат у вигляді текстової інформації, у вигляді набору зображень з виділеним знайденим об'єктом, або виводити на екран набір кадрів з виділеними об'єктами.

Виходячи з вимог швидкості обробки відеопотоку і простоти реалізації, структурні вимоги можуть бути зведені до наступного: модель структури системи повинна бути з мінімальною кількістю статичних зв'язків. З цього випливає, що бажана структура системи повинна бути модульною.

Згідно зі сформульованими вимогами, алгоритм роботи системи повинен бути заснований на пошуку ключових точок. З проведеного огляду

технічної літератури впливає, що методи пошуку ключових точок мають особливості, які необхідно враховувати при їх використанні:

- ключова точка може не бути частиною зображення тільки одного об'єкта, вона може зустрічатися і на зображеннях інших об'єктів, так як ключова точка є локальною ділянкою на зображенні порівняно невеликого розміру і на зображенні можуть бути «схожі» дублікати за рахунок малої інформативності;

- ключові точки можуть знаходитися не на всій області зображення об'єкта, а тільки в деяких підобластях, тобто за знайденими ключовими точками не можна судити про місцезнаходження всього об'єкта, тому визначення положення об'єкта на зображенні, ґрунтуючись тільки на положеннях ключових точок, є некоректним.

Наявність локальних ознак не гарантує надійну ідентифікацію через можливість повторюваності ознак на інших об'єктах і наявності спотворень. У зв'язку з цим для підвищення надійності ідентифікації необхідно розширити вектор локальних ознак однією з глобальних ознак, що дозволяє ідентифікувати об'єкт. З глобальних ознак слід виділити колірну гістограму, оскільки колірна гістограма швидко обчислюється і не залежить від повороту і масштабу зображення [22].

Для підвищення швидкодії пропонується перед початком аналізу відеопотоку здійснити наступні дії:

- створити набір зображень з проєктивними спотвореннями зразка;
- знайти на кожному створеному зображенні ключові точки;
- обчислити дескриптори кожній ключовій точці.

Описані кроки по обчисленню дескрипторів ключових точок на попередньо проєктивно спотворених зображеннях зразка дозволяють використовувати алгоритми для пошуку і опису ключових точок, які є нестійкими до проєктивних перетворень.

Таким чином, загальна схема алгоритму пошуку може бути представлена в вигляді двох етапів:

- швидкий пошук областей претендентів із зображенням об'єкта з допомогою ключових точок.

- ідентифікація об'єкта на зображенні на основі глобальної ознаки - колірної гистограми.

Базуючись на даній схемі, сформульованих вимогах до системи і запропонованому методі обчислення дескрипторів на проєктивно спотворених зображеннях зразка, можна визначити набір функцій, якими повинна володіти система та провести функціональне проектування. Функціональна модель описується на основі нотації IDEF0.

При розгляді системи з точки зору функції верхнього рівня, виділено такі положення (рис. 2.1):

- базовою функцією системи є знаходження об'єкта в відеопотоці;
- вхідними потоками системи є зображення об'єкта і відеопотік;
- механізмом, за допомогою якого функціонує система, є програмна модель;
- вихідними даними є набір зображень об'єкта в потоці.

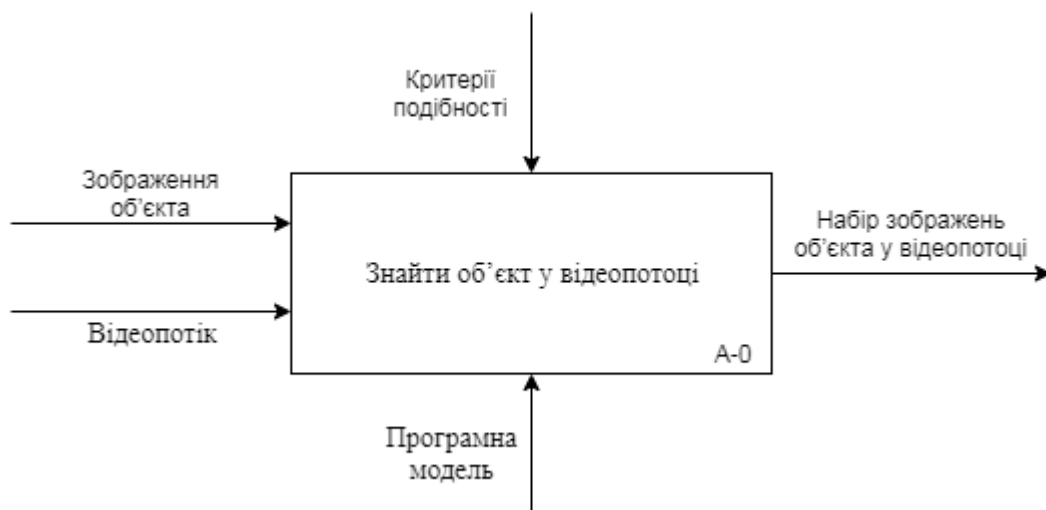


Рисунок 2.1 – Контекстна діаграма верхнього рівня

В результаті первинної декомпозиції діаграми верхнього рівня A-0 отримана діаграма A0 (рис. 2.2).

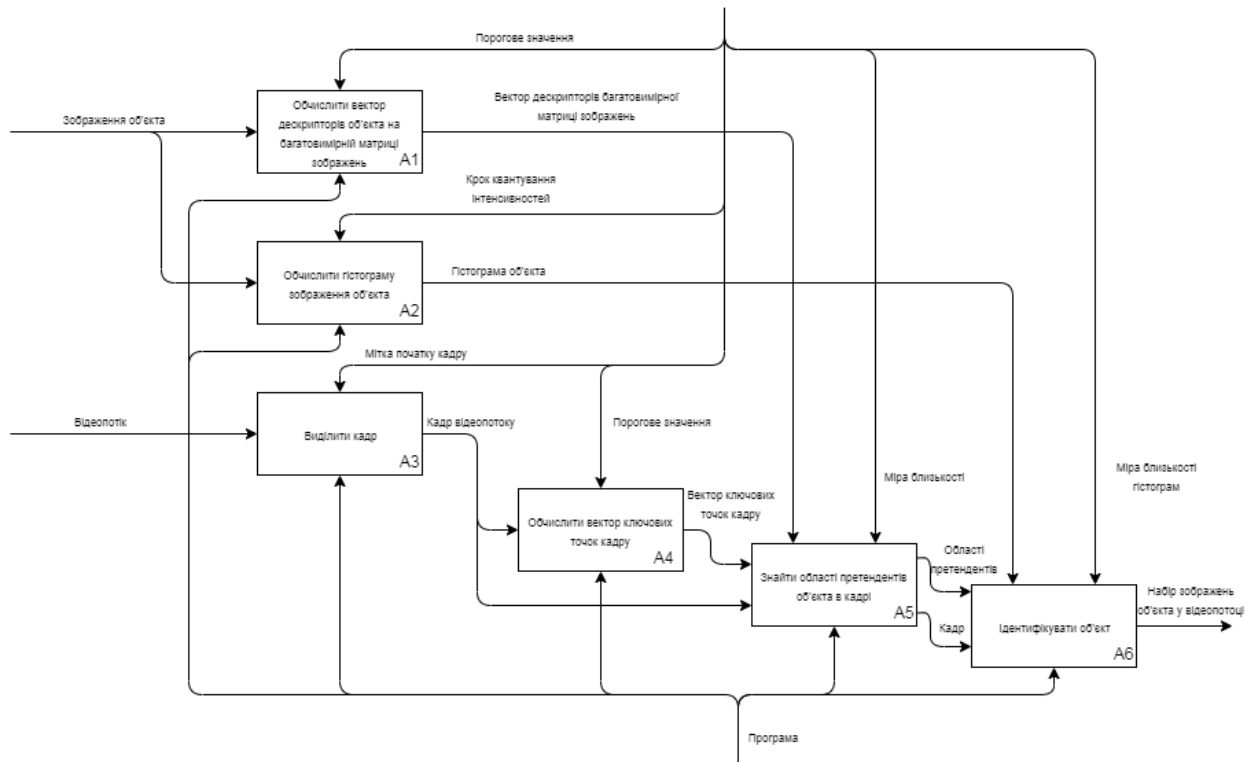


Рисунок 2.2 – Контекстна діаграма першого рівня

Після декомпозиції функції верхнього рівня системи було виділено такі основні функції:

- функція обчислення вектора дескрипторів по багатовимірній матриці проективно спотворених зображень об'єкта (A1);
- функція обчислення гістограми зображення об'єкта (A2);
- функція виділення кадру з відеопотоку (A3);
- функція обчислення вектора ключових точок кадру відеопотоку (A4);
- функція знаходження областей претендентів об'єкта на кадрі (A5);
- функція ідентифікації об'єкта в знайдених областях претендентів на підставі обчислення колірних гістограм (A6).

На діаграмі A0 виділені основні інформаційні потоки:

1. Потік, що містить зображення шуканого об'єкта.

2. Відеопотік. Даний потік включає в себе набір кадрів, на яких здійснюється пошук об'єкта.

У функції ідентифікації по виділених областях кадру знаходиться область з найбільшою відповідністю гистограми зображення шуканого об'єкта.

Функція обчислення вектора дескрипторів багатовимірної матриці зображень об'єкта $A1$ декомпозована на етапи (рис. 2.3):

1. Перетворення зображення у напівтонове ($A11$).
2. Генерація змін масштабу ($A12$). На цьому етапі створюється вектор масштабування зображень об'єкта.
3. Генерація змін нахилу камери ($A13$) відповідно до параметрів нахилу. Цей етап створює матрицю зображень з вектора масштабування зображень зміною нахилу камери.

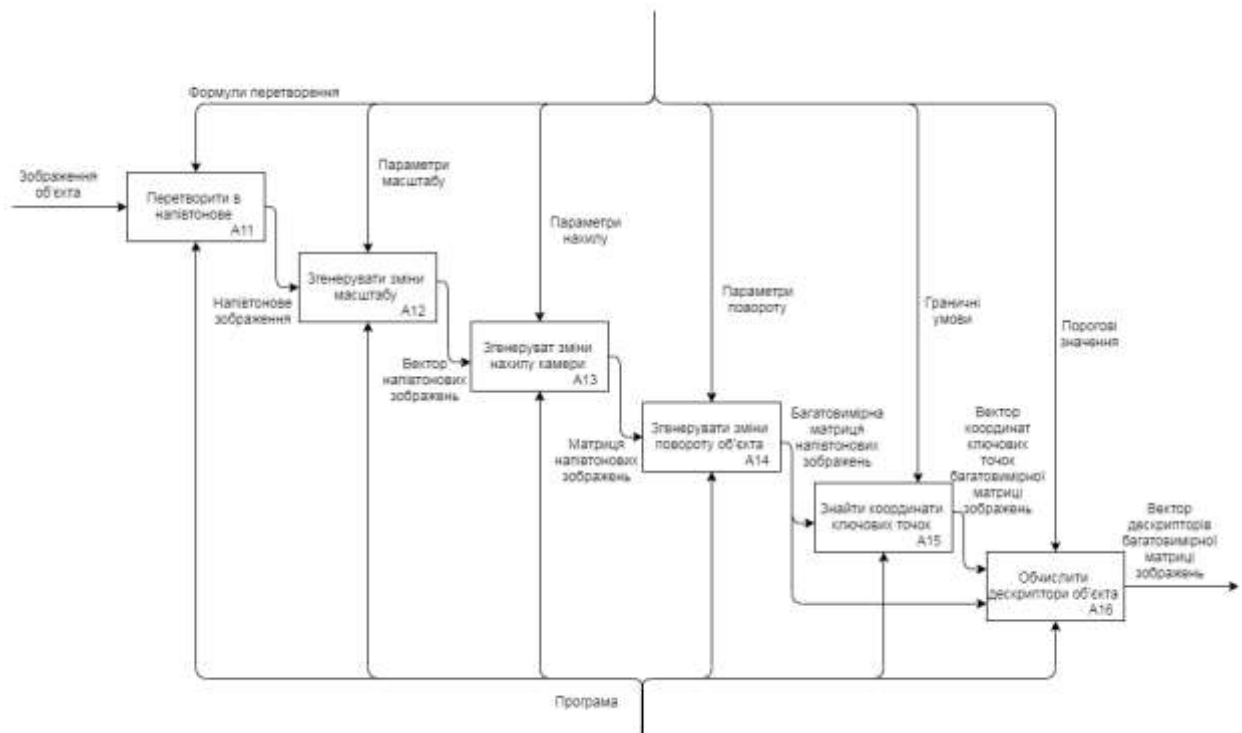


Рисунок 2.3 – Діаграма функції обчислення вектора дескрипторів зображення об'єкта

4. Генерація зміни повороту об'єкта відповідно до параметрів повороту (A14). Цей етап створює тривимірний масив зображень з матриці масштабування зображень, зміною кута повороту зображення об'єкта. Виходом функції є тривимірна матриця зображень об'єкта.

5. Знаходження координат ключових точок зображення об'єкта (A15). Ключова точка описується координатами областей зображення і дескриптором. На цьому кроці знаходяться координати унікальних ключових точок. Виходом функції є вектор координат з радіусом кіл ключових точок.

6. Обчислення дескрипторів ключових точок (A16) на тривимірній матриці зображень об'єкта. Дескриптор ключової точки – це вектор параметрів, обчислених на зображенні в області ключової точки. Ключова точка описується координатами центру та радіусом околу. Управління здійснюється вектором координат центру, радіусом кіл ключових точок, і граничним значенням. Виходом є вектор дескрипторів зображень об'єкта.

Описані етапи A11, A12, A13, A14, A15 і A16 – це кроки методу обчислення дескрипторів на проєктивно спотворених зображеннях зразка.

Результат декомпозиції функції вилучення кадру з відео потоку A3 представлений на рисунку 2.4. В процесі виконання функції відеопотік проходить через наступні етапи:

1. Демультіплексація. Цей етап необхідний для отримання закодованого відеосигналу з контейнера (A31).

2. Декодування. Етап необхідний для представлення закодованого відеосигналу в набір зображень, зручних для обробки в функції пошуку об'єкта в відеопотоці (A32).

3. Виділення кадру. На цьому етапі відповідно до параметрів кадру з декодованого відеопотоку, витягується кадр.

Функція обчислення вектора ключових точок кадру A4 декомпована на наступні етапи (рис. 2.5):

1. Перетворення зображення у напівтонове (A41).

2. Знаходження координат ключових точок кадру (A42).

3. Обчислення дескрипторів в знайдених координатах кадру (A43).

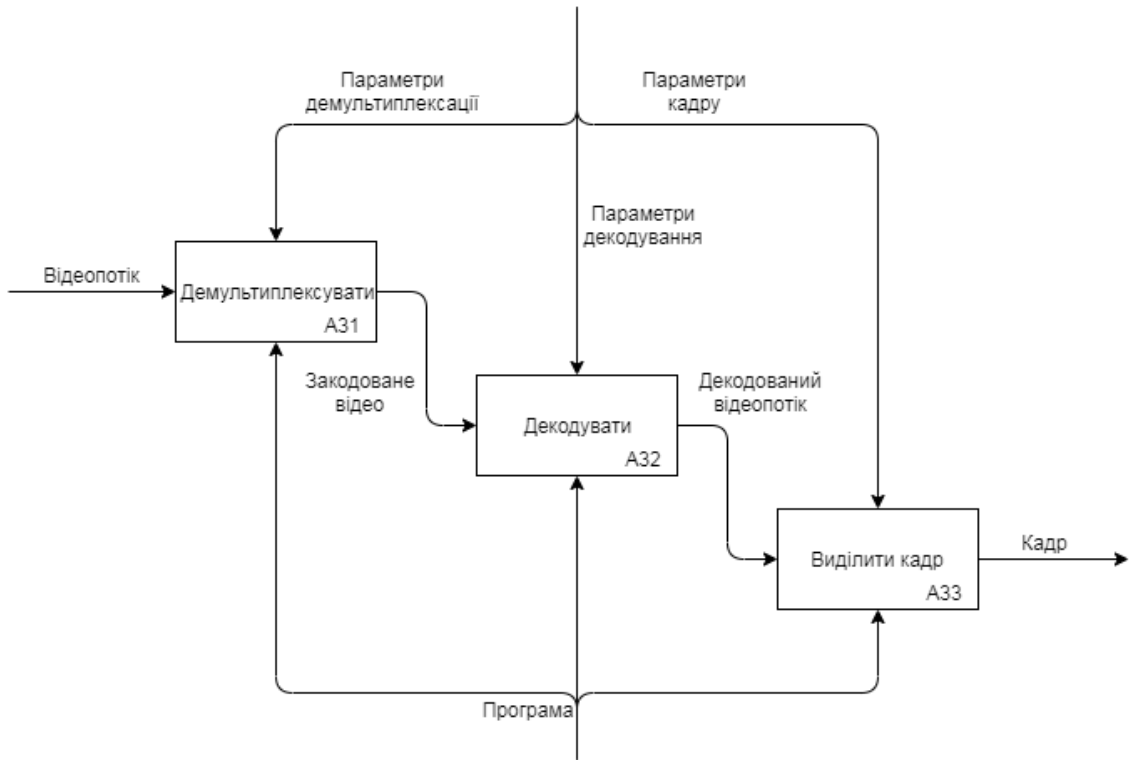


Рисунок 2.4 – Діаграма виділення кадру з відеопотоку

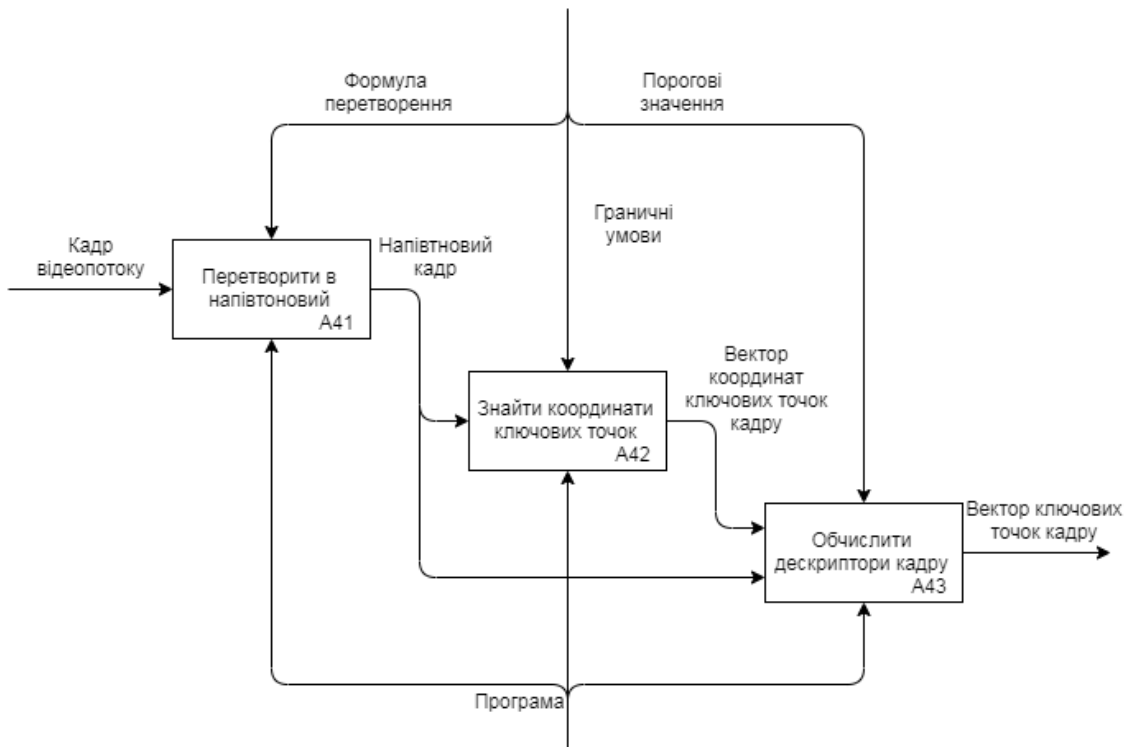


Рисунок 2.5 – Діаграма функції обчислення дескрипторів кадру

Вектор координат ключових точок необхідний для обчислення дескрипторів (A43). Виходом функції є вектор ключових точок з інформацією про номер кадру, координатах і дескрипторі по кожній ключовій точці.

Функція знаходження областей претендентів об'єкта на кадрі (A5) декомпозована на складові (рис. 2.6):

1. Знаходження загальних ключових точок (A51). На вхід надходить вектор ключових точок кадру. Управління здійснюється вектором дескрипторів об'єкта. Використовуючи міру близькості, визначаються ключові точки, дескриптори яких вважаються однаковими з дескрипторами зображення об'єкта, тобто зображення областей, на які вказують ключові точки, вважаються «схожими». У разі отримання не порожньої множини після перетину, виводяться загальні ключові точки і інформація про номер кадру.

2. Пропуск кадру (A52). Згідно з інформацією про номер кадру приймається рішення про пропуск вхідного кадру відеопотоку для подальшої обробки.

3. Отримання областей претендентів (A53). Вхідний вектор координат ключових точок кадру об'єднується відповідно до розміру вікна в області, які подаються на вихід функції.

Центральними функціями системи є функції знаходження ключових точок зображення об'єкта (A15) і кадру (A42), обчислення дескрипторів ключових точок зображення об'єкта (A16) і кадру (A43), знаходження спільних ключових точок по дескрипторів (A51) і ідентифікації кадру на предмет наявності зображення об'єкта (A6). Від результату виконання цих функцій залежить якість і швидкість виявлення об'єкта в потоці. Далі розглядаються докладніше алгоритми знаходження ключових точок спільно з обчисленням дескрипторів, алгоритми знаходження перетину дескрипторів і алгоритми, що дозволяють ідентифікувати зображення об'єкта в області кадру по обчисленню гістограми.

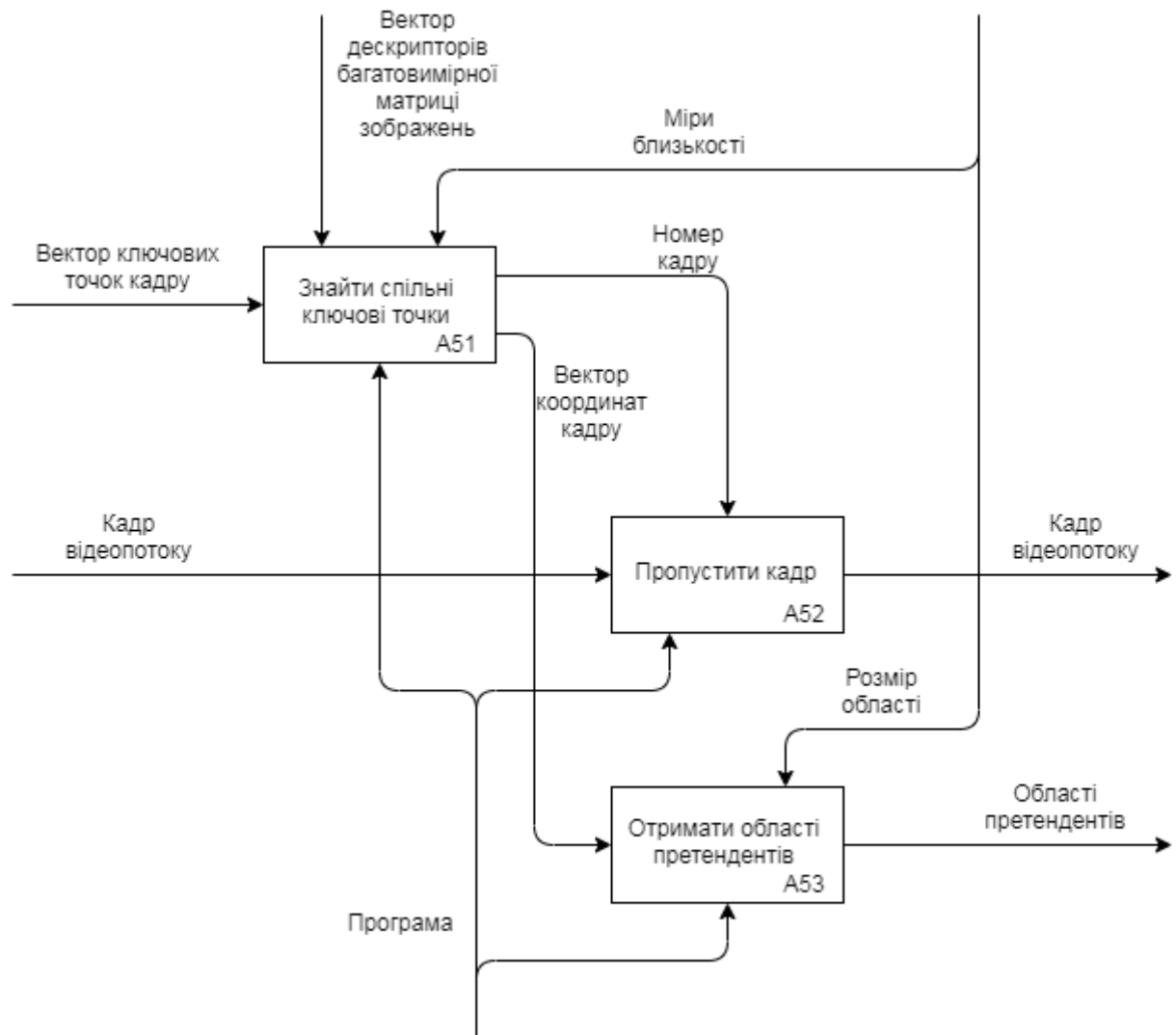


Рисунок 2.6 – Діаграма функції знаходження областей претендентів

2.2 Методи знаходження перетину дескрипторів ключових точок зображення

Для знаходження перетину двох множин дескрипторів, на сьогоднішній день, активно використовуються наступний підходи [23]:

- метод RANSAC;
- алгоритм Куна-Манкреса.

Метод RANSAC – це загальний метод, який використовується для оцінки параметрів моделі на підставі випадкових вибірок. При зіставленні модель являє собою матрицю перетворення (гомографію) [24]. На вході

алгоритму є дві множини дескрипторів. Схема роботи RANSAC складається з багаторазового повторення трьох етапів:

1. Вибір точок і побудова параметрів моделі. Зі вхідних множин дескрипторів вибираються випадковим чином без повторень набори фіксованого розміру. На підставі отриманих наборів будується матриця перетворення.

2. Перевірка побудованої моделі. Для кожного дескриптора зображення об'єкта знаходиться проекція на поточному кадрі і виконується пошук найбільш близького дескриптора із множини дескрипторів поточного кадру. Дескриптор позначається як викид, якщо відстань між проекцією і відповідним дескриптором поточного зображення більша деякого порогу.

3. Заміщення моделі. Після перевірки всіх точок перевіряється, чи є побудована модель кращою серед набору попередніх моделей. В результаті застосування RANSAC будується найкраща матриця гомографії. Обчисливши перспективну проекцію набору дескрипторів зображення об'єкта, досить виконати прохід по усім відповідностям, отриманих в процесі перебору, і перевірити, чи є відповідний дескриптор поточного кадру досить близьким до проекції дескриптора зображення об'єкта. Якщо ні, то пара відкидається.

Обчислювальна складність для однієї моделі складе $O(n)$, однак на практиці результати не є прийнятними для використання за рахунок великої кількості можливих помилок. Існують модифікації методу RANSAC. Наприклад, алгоритм Джі-Лінкейджа і алгоритм адаптації ядра, що дозволяють знаходити пари з меншою кількістю помилок, але володіють обчислювальною складністю $O(n^2)$.

Алгоритм Куна-Манкреса. Задачу співставлення дескрипторів можна представити у вигляді завдання про призначення [25]. Інтерпретуємо її у вигляді графа. Нехай параметри масок(дескриптори) – вершини графа, а значення міри схожості вершин – ребра цього графа. Складність оригінального алгоритму становить $O(n^4)$. Для вирішення задачі методом

Куна-Манкреса необхідно додати нові віртуальні вершини графа, які будуть нескінченно віддалені від інших вершин. Тоді $K_{n,n} [W]$ – зважений граф з долями X і Y . Результатом роботи даного методу є множина ребер оптимального паросполучення P в даному графі.

Метод Куна-Манкреса можна представити у вигляді наступних послідовних операцій:

а) задати в $K_{n,n} [W]$ довільну допустиму розмітку f і знайти підграф рівності $G_{W,f}$;

б) угорським алгоритмом знайти максимальне паросполучення P в графі $G_{W,f}$ і множину F вершин долі X , вільних відносно P ;

в) якщо $F = \emptyset$ – закінчити роботу;

г) знайти всі ланцюги, що чергуються в графі $G_{W,f}$, що починаються в F , покласти S і T рівними множині всіх вершин долі X (відповідно, долі Y), які зустрілися в цих ланцюгах;

д) якщо в T немає вільних вершин, покласти:

$$\Delta = \min_{x_i \in S, y_i \in Y \setminus T} \{f(x_i) + f(y_i) - w_{ij}\},$$

де $f(x) = f(x) - \Delta$ для усіх $x \in S$;

$f(y) = f(y) + \Delta$ для усіх $y \in T$.

Знайти новий граф $G_{W,f}$ і повернутися до кроку г).

2.3 Метод ідентифікації досліджуваної області зображення

Алгоритм ідентифікації повинен визначати, чи є область на кадрі зображенням або частиною зображення об'єкта. Для цього алгоритм повинен знайти параметри вікна на кадрі згідно зі знайденими областями, отриманих на підставі співставлення локальних ознак зображення – ключових точок.

Нехай алгоритм ідентифікації знаходить об'єкт еліптичних вікном. В алгоритмі пропонується використовувати метод, заснований на глобальній властивості зображення. Однією з найпоширеніших глобальних характеристик є гістограма кольору [26]. Гістограма кольору обчислюється швидко, однак при обчисленні, до уваги береться просторове розташування пікселів. Пропонується значення кольору точок вносити з певною вагою: чим ближче точка до центру вікна, тим більше у неї вагах [27]. Це необхідно і для того, щоб невеликі зсуви вікна приводили до невеликих змін помилки зіставлення. Такій умові відповідає ядро Епонечнікова [28]:

$$K(x) = \begin{cases} 1 - x^2, & |x| \leq 1, \\ 0, & |x| > 1. \end{cases}$$

Таким чином, колір пікселя x буде внесений до гістограми з певною вагою $K(x)$.

В основі ідентифікації об'єкту пропонується використовувати техніку *Mean Shift* [29]. Метод *Mean Shift* заснований на пошуку максимуму щільності ймовірності деякої функції, яка описує дискретні дані. Для локалізації об'єкту необхідно використовувати градієнтний спуск. В якості критерію схожості пропонується використовувати коефіцієнт Бхаттачарія [30].

Градієнтний спуск застосовується для вирішення задачі знаходження локального мінімуму. Пропонується використовувати чотирьохпараметричний пошук вікна зображення об'єкта (рис. 2.7) [30]. Для процесу ідентифікації сталого до незначних змін кольору і для зменшення розміру гістограми проводиться квантування значень гістограми кольору [31].

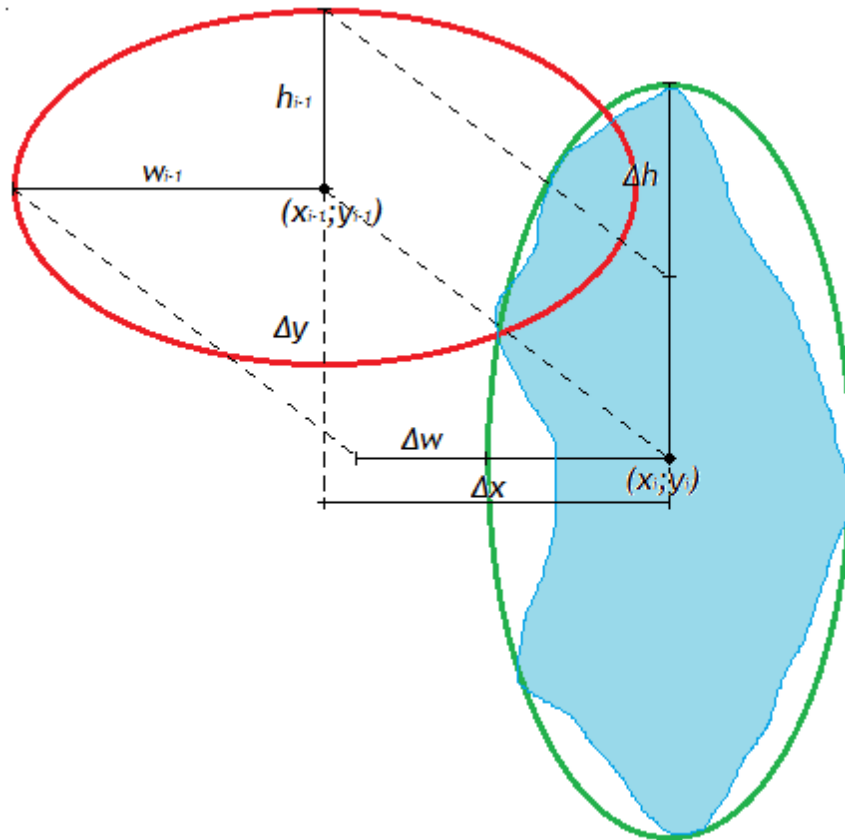


Рисунок 2.7 – Чотирьохпараметрична модель пошуку

Чотирьохпараметричний пошук вікна зображення об'єкта методом градієнтного спуску складається з наступних кроків:

- а) поставити ρ_{th} (мінімальне значення порога схожості), i_{max} (максимальну кількість ітерацій), w_0 і h_0 (довжини півосей еліпса);
- б) обчислити нормований вектор частот значень інтенсивностей (Гістограму) $hist_0$ в еліптичній області (x_0, y_0, w_0, h_0) ;
- в) обчислити міру схожості за допомогою коефіцієнта Бхаттачарія:

$$\rho_0 = \sum_{b \in B} \sqrt{hist_{refb} \cdot hist_{0b}},$$

де $hist_{refb}$ – гістограма зображення об'єкта для пошуку,

b – крок гістограми, $b \in B$;

г) $i = 1$;

д) поки $\rho_{i-1} < \rho_{th}$ та $i < i_{max}$ виконати:

1) обчислити градієнт $grad\rho_{i-1}$:

$$grad\rho_{i-1} = \left(\frac{\partial\rho_{i-1}}{\partial x}, \frac{\partial\rho_{i-1}}{\partial y}, \frac{\partial\rho_{i-1}}{\partial w}, \frac{\partial\rho_{i-1}}{\partial h} \right);$$

2) обчислити k_{i-1} :

$$k_{i-1} \approx \frac{1-\rho_{i-1}(x_{i-1}, y_{i-1}, w_{i-1}, h_{i-1})}{|grad\rho_{i-1}|^2};$$

3) обчислити величину кроку:

$$(\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i) = k_{i-1} \cdot grad\rho_{i-1};$$

4) змінити параметри еліпса:

$$(x_i, y_i, w_i, h_i) = (\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i) + (x_{i-1}, y_{i-1}, w_{i-1}, h_{i-1});$$

5) розрахувати відносну гістограму $hist_i$ еліптичної області (x_i, y_i, w_i, h_i) ;

б) обчислити міру схожості ρ_i :

$$\rho_i = \sum_{b \in B} \sqrt{hist_{refb} \cdot hist_{ob}};$$

7) $i = i + 1$;

е) стоп.

Початкові значення w_0 і h_0 беруться з параметрів області P , яка була отримана після виконання алгоритму обмеження області пошуку об'єкта в кадрі. Колірна гістограма обчислюється по компонентам U і V колірного простору YUV [32]:

$$U = -0.14713 \cdot R - 0.28886 \cdot G + 0.436 \cdot B + 128,$$

$$V = 0.615 \cdot R - 0.51499 \cdot G - 1.0001 \cdot B + 128,$$

де R, G, B – 8-ми бітові значення кольору.

Така гистограма має менший розмір на відміну від гистограми, що складається з трьох компонентів простору RGB і така гистограма є більш стійкою до зміни складової яскравості на зображенні.

Гradientний спуск, застосований для визначення параметрів чотирьохпараметричної моделі, має обчислювальну складність $O(n^2)$. Як тільки в методі ідентифікації значення критерію відповідності гистограми кольору стає вищим за певний поріг ρ_{th} , або кількість ітерацій перевищило максимальне значення i_{max} , процес ідентифікації припиняється. Зображення об'єкта вважається знайденим на кадрі, якщо перевищено мінімальне значення порога схожості ρ_{th} .

2.4 Метод пошуку об'єкта у відеопотоці

Декомпозиція функції пошуку об'єкта в відеопотоці дозволяє описати алгоритм пошуку (рис. 2.8):

- а) обчислити параметри зображення об'єкта для пошуку;
 - 1) створити набір Q^{ref} проєктивно спотворених зображень об'єкта Y^{ref} ;
 - 2) знайти ключові точки зображень об'єкта для пошуку:

$$K^{ref} = \{K^{ref}_0, \dots, K^{ref}_{l-1}\};$$

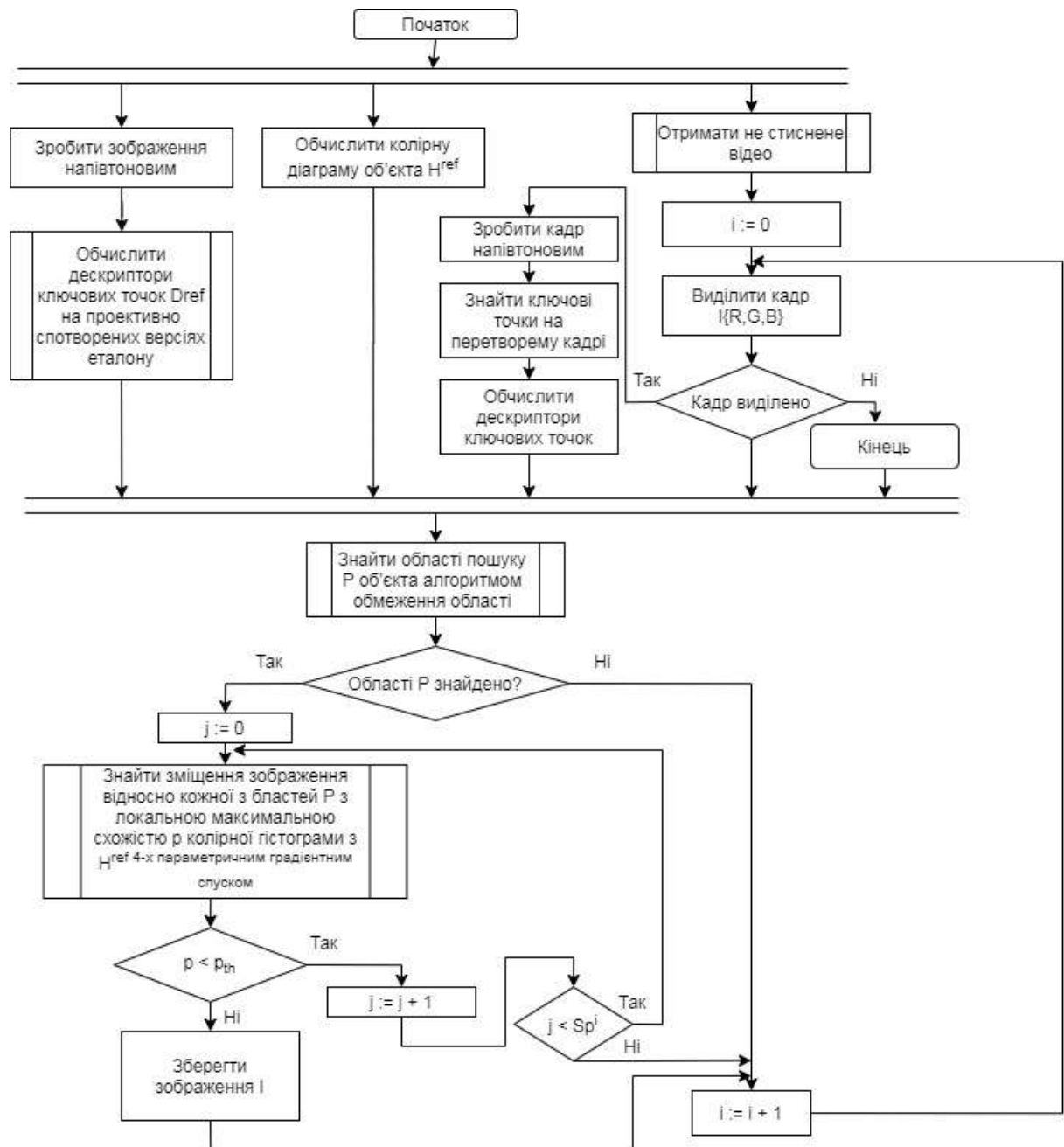


Рисунок 2.8 – Блок-схема роботи методу пошуку об'єкта у відеопотоці

3) Обчислити дескриптори ключових точок $Kref$ зображень об'єкта:

$$Dref = \begin{bmatrix} Dref_0^0 & \dots & Dref_{m-1}^0 \\ \vdots & \ddots & \vdots \\ Dref_0^{k-1} & \dots & Dref_{m-1}^{k-1} \end{bmatrix},$$

де $Dref$ – матриця значень дескрипторів об'єкта;

$Dref_j^i$ – j -е значення i -ого дескриптора;

m – розмір дескриптора;

k – кількість дескрипторів;

4) обчислити колірну гістограму зображення об'єкта:

$$H^{ref} = \{H_0^{ref}, \dots, H_{n-1}^{ref}\};$$

б) поки $i < N_{max}$ виконати:

1) знайти ключові точки i -ого кадру:

$$K^i = \{K_0^i, \dots, K_{m-1}^i\};$$

2) обчислити дескриптори ключових точок i -ого кадру:

$$D^i = \{D_0^i, \dots, D_{m-1}^i\};$$

3) порівняти дескриптори D^i кадру з рядками матриці $Dref$ і вибрати ті, що вище певного порогу Thr ;

4) знайти набір прямокутних областей P^i на кадрі, алгоритмом обмеження області пошуку об'єкта;

5) поки $j < Sp^i$ виконати:

– перевірити область P_j^i на наявність зображення об'єкта чотирьох параметричних градієнтним спуском;

– якщо міра схожості гістограм нижче порога ρ_{th} , потрібно прийняти $j := j + 1$ та перейти до кроку 5);

– інакше потрібно виділити прямокутним вікном зображення об'єкта, зберегти кадр, прийняти $i := i + 1$ та перейти до кроку б);

в) стоп.

2.5 Розробка структурної та функціональної моделей пошуку об'єкта у відео потоці

На основі побудованої функціональної моделі можна визначити основні модулі системи:

- модуль обчислення вектора дескрипторів об'єкта;
- модуль обчислення гистограми;
- модуль виділення кадрів;
- модуль обчислення вектора дескрипторів кадру;
- модуль пошуку областей претендентів;
- модуль ідентифікації.

На рисунку 2.9 представлена діаграма основних компонентів системи. Модуль обчислення вектора дескрипторів об'єкта виконує функції A11, A12, A13, A14, A15, A16, представлені в IDEF0-діаграмах. Модуль вилучення кадрів виконує функції A31, A32, A33. Кадри з нього надходять в модуль обчислення вектора дескрипторів кадру і в модуль пошуку областей претендентів. Функція A2 виконується модулем обчислення гистограми. Модуль обчислення вектора дескрипторів кадру виконує функції A41, A42, A43. Модуль пошуку областей претендентів реалізує функції A51, A52, A53. Ідентифікатор виконує функцію A6.

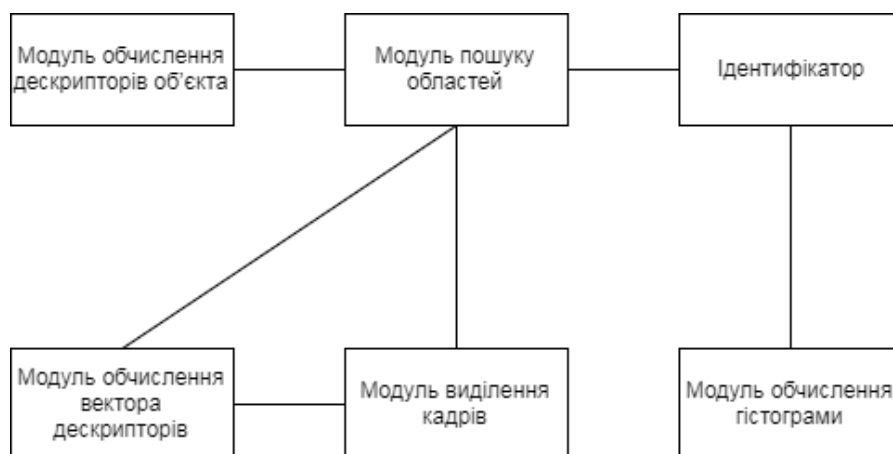


Рисунок 2.9 – Структурна діаграма

На рисунку 2.10 представлена структура модуля обчислення вектора дескрипторів об'єкта. Модуль складається з наступних елементів:

- напівтоновий перетворювач – виконує функцію A11;
- перетворювач масштабу – виконує функцію A12;
- перетворювач нахилу – моделює зміни нахилу камери на задані кути, реалізує функцію A13;
- генератор повороту – виконує функцію A14;
- блок пошуку ключових точок – виконує функцію A15;
- процесор дескрипторів – виконує функцію A16.



Рисунок 2.10 – Структура модуля обчислення вектора дескрипторів об'єкта

Структура модуля виділення кадру представлена на рисунку 2.11. Модуль складається з наступних елементів:

- демультимплексер – виконує функцію демультимплексації відеопотоку з контейнера в стиснений формат відео (A31);
- декодер – виконує функцію декодування стисненого формату відео (A32);
- компонент виділення кадру – виконує функцію виділення кадру з декодованого відеопотоку (A33).

Структура модуля обчислення вектора ключових точок кадру представлена на рисунку 2.12. Модуль складається з наступних елементів:

- блок перетворення зображення у напівтонове – виконує функцію A41;
- блок пошуку ключових точок – знаходить ключові точки на кадрі, виконує функцію A42;
- блок обчислення дескрипторів – обчислює дескриптори ключових точок, виконує функцію A43.

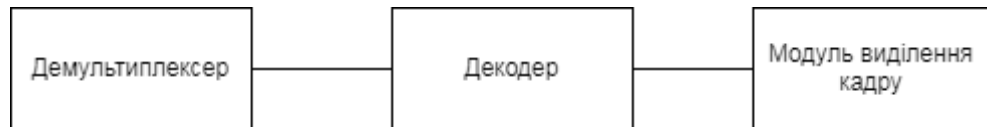


Рисунок 2.11 – Структура модуля виділення кадру



Рисунок 2.12 – Структура модуля обчислення вектора ключових точок кадру

Структура модуля пошуку областей претендентів представлена на рисунку 2.13. Модуль складається з наступних елементів:

- визначник перетину дескрипторів – виконує функцію A51;
- кадровий фільтр – пропускає кадр з певним номером, отриманим з визначника перетину дескрипторів, виконує функцію A52;
- визначник областей – виконує функцію A53.

Функціональний аналіз системи пошуку об'єкта показав, що в структурі програмної системи повинні бути включені в якості базових наступні функції:

- знаходження ключових точок;
- обчислення дескрипторів ключових точок;

- знаходження спільних ключових точок по дескрипторах;
- ідентифікація кадру на предмет наявності зображення об'єкта.

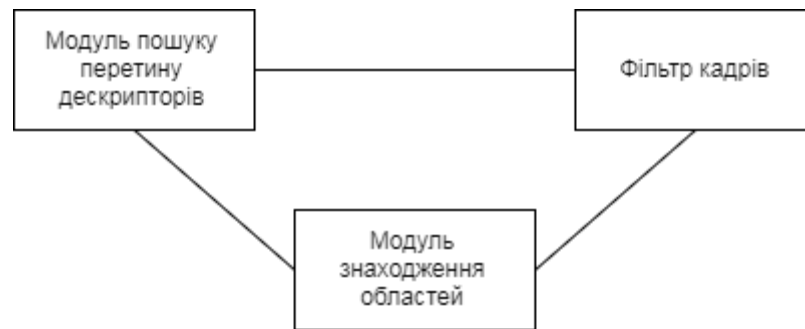


Рисунок 2.13 – Структура модуля пошуку областей претендентів

Алгоритм пошуку об'єкта повинен будуватися на основі обчислення дескрипторів, оскільки такий підхід дозволяє уникнути впливу проєктивних спотворень.

Для збільшення швидкості роботи алгоритму необхідно перед початком пошуку піддати зображення об'єкта проєктивним спотворенням для знаходження ключових точок і обчислення дескрипторів об'єкта [33].

Для підвищення ймовірності виявлення дескриптора пошук ключових точок необхідно доповнити кроком порівняння колірних гістограм [34].

3 ДОСЛІДЖЕННЯ МЕТОДУ ПОШУКУ ОБ'ЄКТА У ВІДЕОПОТОЦІ НА ОСНОВІ ДЕСКРИПТОРІВ КЛЮЧОВИХ ТОЧОК ЗОБРАЖЕННЯ

3.1 Вибір інструментальних засобів та інформаційних технологій для створення системи пошуку об'єкта у відеопотоці

Спроектовані класи і схеми взаємодії системи дозволяють здійснити реалізацію програми пошуку об'єкта в потоці. Реалізація програми повинна задовольняти вимогам до програми і відповідати побудованим функціональній і структурній моделі. Далі описуються технології розробки програми для ефективного виконання пошуку об'єкта.

Програма працює в двох режимах згідно описаних сценаріїв роботи системи:

- режим ініціалізації зображення об'єкта;
- режим пошуку кадрів об'єкта в потоці.

Пошук кадрів із зображенням об'єкта в відеопотоці вимагає виконання набагато більшої кількості обчислювальних операцій, ніж ініціалізація зображення об'єкта [35]. Реалізація програмної системи повинна бути орієнтована на ефективне використання обчислювальних ресурсів для того, щоб задовольнити вимоги до швидкості обробки. Пропонується використовувати наступні технології:

- конвеєрна обробка даних;
- пакетна обробка даних;
- CUDA для паралельних обчислень на відеокарті.

Конвеєрна архітектура дозволить виконувати обробку відео кадрів одночасно на всіх компонентах системи, тим самим зменшуючи середній час обробки кадру до максимального часу обробки одного з компонентів системи [36]. В ідеальному розбитті на n компонент, час обробки кадрів практично зменшиться в n разів.

Рішення про відсутність об'єкта на кадрі може бути прийнято на етапі визначення перетину дескрипторів кадру і зображення об'єкта, тому пропонується виділяти більше обчислювальних ресурсів операціям пошуку ключових точок, обчислення дескрипторів та визначення перетину дескрипторів.

Для мінімізації часу пошуку об'єкта в відеопотоці пропонується реалізувати основні функції з використанням пакетної обробки даних: технологія SSE2 (Streaming SIMD Extensions 2) [37]. Її призначення – підвищити ефективність операцій зі 128-розрядними даними в форматі з плаваючою комою з подвійною точністю і з цілочисельними даними. Всі сучасні процесори підтримують технологію SSE2, на відміну від технологій SSE4, AVX і AVX2 [38].

Для оптимального використання ресурсів пропонується при наявності відеокарти з підтримкою технології CUDA використовувати обчислювальні ресурси відеокарти. CUDA – це архітектура паралельних обчислень від NVIDIA, що дозволяє істотно збільшити обчислювальну продуктивність завдяки використанню GPU (графічних процесорів). На підставі результатів обчислення алгоритмічної складності детекторів ключових точок, процес пошуку ключових точок вимагає велику кількість обчислювальних операцій і його можна легко розпаралелити, на відміну від процесу обчислення дескриптора. Нехай пошук ключових точок здійснюється на відеокарті при її наявності. Тоді для операції обчислення дескриптора буде віддаватися більше процесорних обчислювальних ресурсів.

Для реалізації програм використані наступні програмні засоби:

- мова програмування C ++;
- бібліотека OpenCV 2.4, яка містить функції декодування відеопотоку, обчислення дескрипторів ключових точок та алгоритм RANSAC;
- набір інструментів CUDA SDK;
- операційна система Microsoft® Windows 7 x64;

- середовище розробки: Microsoft® Visual Studio 2017 року;
- система контролю версій: Git.

3.2 Тестування розробленого програмного засобу

Реалізована програма дозволяє здійснити автоматичне тестування розробленого алгоритму пошуку об'єкта у відеопотоці з використанням текстового формату виведення результатів виявлення. Необхідно розробити схему тестування і написати скрипти для проведення тестів. Відповідно до поставлених вимог система повинна бути інваріантною до проєктивних перетворень, шуму і знаходити об'єкт у відеопотоці в режимі реального часу, тому пропонується досліджувати кожну програму на відповідність цим вимогам. Для цього готуються тести з відеопотоками, на яких зображення певного об'єкта піддається проєктивним перетворенням.

У тестуванні пропонується використовувати метрику ROC для чисельного вимірювання якості знаходження об'єкта в відеопотоці різними методами. ROC-крива має особливу популярність в області аналізу класифікацій і відображає залежність частки вірно класифікованих позитивних подій в загальній кількості позитивних подій від частки невірно класифікованих негативних подій в загальній кількості негативних подій, тому на графіку точка $(0, 1)$ відповідає ідеальному класифікатору [38]. Для побудови ROC-кривих необхідно обчислити:

- суму істинно-позитивних значень (true-positive, TP);
- суму помилково-позитивних значень (false-positive, FP);
- суму істинно-негативних значень (true-negative, TN);
- суму помилково-негативних значень (false-negative, FN).

TP , FP , FN , TN визначаються за такими формулами:

$$TP_i = \sum_{\rho_j \geq Th_i} \rho_j,$$

$$FP_i = \sum_{\rho_j \geq Th_i} (1 - \rho_j),$$

$$FP_i = \sum_{\rho_j < Th_i} \rho_j,$$

$$FP_i = \sum_{\rho_j < Th_i} (1 - \rho_j).$$

Графік ROC-кривої є графіком залежності частки істинно-позитивних значень (TPR) від частки помилково-позитивних значень (FPR). Дані величини обчислюються за такими формулами:

$$TPR_i = \frac{TP_i}{TP_i + FN_i},$$

$$FPR_i = \frac{FP_i}{FP_i + TN_i}.$$

Чисельний показник площі під ROC кривою (AUC) визначається за наступною формулою:

$$AUC = \frac{1}{2} \sum_i (TPR_{i+1} + TPR_i) \cdot (FPR_{i+1} - FPR_1).$$

Програмна система працює ефективно, якщо використовує усі надані апаратно-програмні засоби протягом всього часу роботи. В якості критеріїв ефективності пропонується використовувати загальне середнє завантаження процесорів і середнє завантаження процесорів на обробку відеопослідовності в секунду.

Середнє завантаження процесорів обчислюється за такою формулою:

$$L = \frac{1}{n} \sum_{i=0}^{n-1} CPU_i,$$

де n – кількість процесорів;

CPU_i – середній відсоток завантаження i -ого процесора в секунду.

Середнє завантаження процесорів в секунду при обробці відеопослідовності обчислюється за формулою:

$$L_s = \frac{L}{T},$$

де L_s – середнє завантаження процесорів;

T – час виконання програми на обробку відеопослідовності.

Чим ближче значення L_s до 0, тим довше необхідно використовувати обчислювальні ресурси для обробки відеопослідовності.

Тест містить зображення об'єкта для пошуку, спеціально підготовлений відеопотік і інформацію про номерах кадрів із зображенням об'єкта. Виконання тесту здійснюється спеціальною програмою. Результат виконання тесту – значення обчисленої метрики. Завдання скрипта – отримати результат метрики по вихідних даних відпрацьованої програми на певному відеофайлі із зображенням об'єкта з використанням інформації про очікувані результати (список номерів кадрів відеофайлу з зображенням об'єкта).

Через знання системи, тестування проводиться в режимі «чорного ящика» через спеціальні інтерфейси взаємодії з програмою. Для проведення тестування написаний скрипт на мові python. Скрипт виконує послідовний запуск реалізованих програм пошуку об'єкта в відеопотоці з граничним значенням міри близькості і обчислює значення ROC кривої (FPR , TPR), використовуючи інформацію про номери кадрів із зображенням об'єкта для кожного відеопотоку. Всі розглянуті методи мають критерії подібності, значення яких змінюються від нуля до одиниці. Граничне значення міри близькості в скрипті змінюється від нуля до одиниці з кроком 0,05.

3.3 Результати дослідження методу пошуку об'єкта у відеопотоці на основі дескрипторів ключових точок зображення

Тестування проводилося на машині Lenovo B590 з двох ядерним процесором Intel Pentium 2030M з частотою 2,2 ГГц, об'ємом пам'яті рівним 4 Гб, операційною системою MS Windows 7 x64 і відеокартою GeForce GTX 720M з частотою 1033 МГц і обсягом пам'яті 1024 Мб з відключеним Turbo Boost.

Усі програмні моделі написані на мові C++ і використовують паралельні обчислення з технологіями CUDA і SSE2. Реалізацію детекторів ключових точок та процедури зіставлення дескрипторів методом RANSAC взято з бібліотеки OpenCV 2.4.

3.3.1 Тестування інваріантності до проєктивних спотворень

Для проведення тестування інваріантності до проєктивних перетворень, було підготовлено відеофайли зі зміненим поворотом, масштабом об'єкта і нахилом камери відносно об'єкта. У послідовності брали участь кадри, що не містять зображення об'єкта.

Для перевірки інваріантності до поворотів взято десять зображень різних об'єктів для пошуку, з яких змодельовані відеопослідовності зі зміненим кутом повороту. Для першого зображення кожного відеофайлу кут змінений на 5° від оригінальної картинки. Це зроблено для того, щоб зміна кута повороту відбувалася з іншим кроком, відмінним від кроків в порівнюваних методах. Решта змін кутів повороту відбуваються з кроком 10° .

Значення результатів тестів на різних вхідних відеопотоках для FPR і TPR усереднені і представлені в таблиці 3.1. На рисунку 3.1 наведені графіки ROC кривих (довірчий інтервал рівний 95%).

Таблиця 3.1 – Середні значення FPR і TPR результатів тестів інваріантності відносно повороту

Поріг міри схожості	SIFT		SURF		BRISK		ORB	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR
0,95	0	0,2341	0	0,1123	0	0,0184	0	0,2197
0,9	0	0,4173	0	0,2341	0	0,1396	0	0,3799
0,85	0	0,5693	0,0175	0,3559	0,1082	0,3008	0	0,5636
0,8	0	0,7812	0,094	0,4777	0,1712	0,4828	0,0291	0,7254
0,75	0,1102	0,9434	0,1275	0,5995	0,2342	0,5532	0,1823	0,8843
0,7	0,2267	1	0,2356	0,7213	0,4472	0,7944	0,3714	0,9832
0,65	0,3036	1	0,3637	0,8431	0,5702	0,8756	0,5386	1
0,6	0,4404	1	0,6218	0,9649	0,7932	0,9268	0,6271	1
0,55	0,5598	1	0,7499	1	0,9162	0,9881	0,6934	1
0,5	0,7023	1	0,8681	1	0,9932	1	0,7357	1
0,45	0,8819	1	0,9861	1	1	1	0,8832	1
0,4	0,9621	1	1	1	1	1	0,9794	1
0,35	0,9863	1	1	1	1	1	1	1

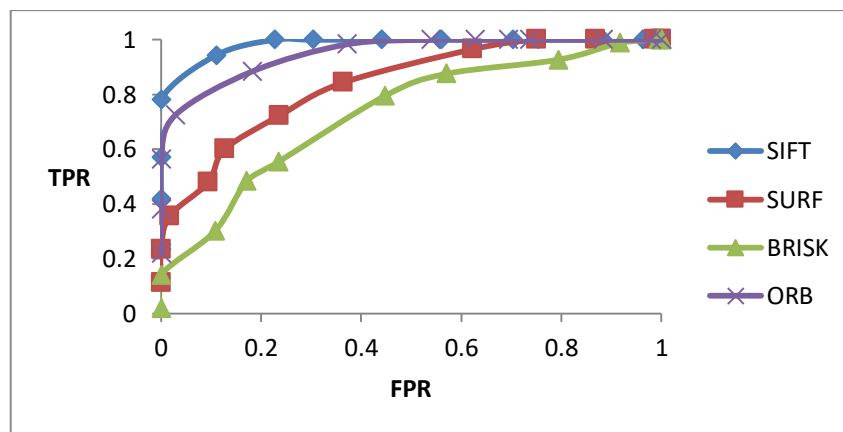


Рисунок 3.1 – ROC криві результатів тестів інваріантності відносно повороту

Графіки методів SIFT та ORB проходять ближче всіх до точки (0; 1) – це означає, що дані методи успішно класифікували практично всі кадри відеопослідовностей з наявністю зображення об'єкта і з його відсутністю.

Аналогічним чином проводилися тести зі зміною масштабу на десяти відео потоках (таблиця 3.2). Масштаб картинки зменшувався від 2 до 20 разів

по ширині і висоті, і потім збільшувався від 2 до 20 разів з кроком рівним 2. Результати для тесту зі зміною масштабу виглядають менш переконливими, ніж результати тесту на перевірку інваріантності до повороту. Це пов'язано з тим, що більшість методів засновані на дескрипторі, що є стійким до повороту на 180° , а для інваріантності до масштабу методи змінюють розміри картинки до певних встановлених максимальних і мінімальних значень.

Таблиця 3.2 – Середні значення FPR і TPR результатів тестів інваріантності відносно зміни масштабу

Поріг міри схожості	SIFT		SURF		BRISK		ORB	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR
0,95	0	0,1941	0	0,1543	0	0,0387	0	0,3212
0,9	0	0,3225	0	0,2754	0	0,1665	0	0,3799
0,85	0	0,5568	0,0175	0,4671	0,0753	0,3496	0,0184	0,5636
0,8	0,0521	0,8212	0,0642	0,5718	0,1591	0,4754	0,0476	0,7254
0,75	0,1465	0,9523	0,1347	0,6873	0,2264	0,5325	0,2081	0,8632
0,7	0,2687	0,9865	0,2356	0,7942	0,4345	0,6754	0,3892	0,9641
0,65	0,3114	1	0,3576	0,8987	0,5865	0,7912	0,5521	0,9987
0,6	0,4245	1	0,6018	0,9949	0,7471	0,9192	0,6483	1
0,55	0,5789	1	0,7621	1	0,8821	0,9961	0,7893	1
0,5	0,7532	1	0,8714	1	0,9578	1	0,8971	1
0,45	0,9787	1	1	1	1	1	1	1
0,4	0,9893	1	1	1	1	1	1	1
0,35	0,9863	1	1	1	1	1	1	1

Графіки ROC кривих (рис. 3.2) методів SIFT та ORB є ближчими за всіх до точки (0; 1) і при ймовірності помилки $\approx 0,1$ класифікації кадрів без зображення об'єкта, ними досягається 0,92 (і більше) ймовірності розпізнавання кадрів із зображенням об'єкта.

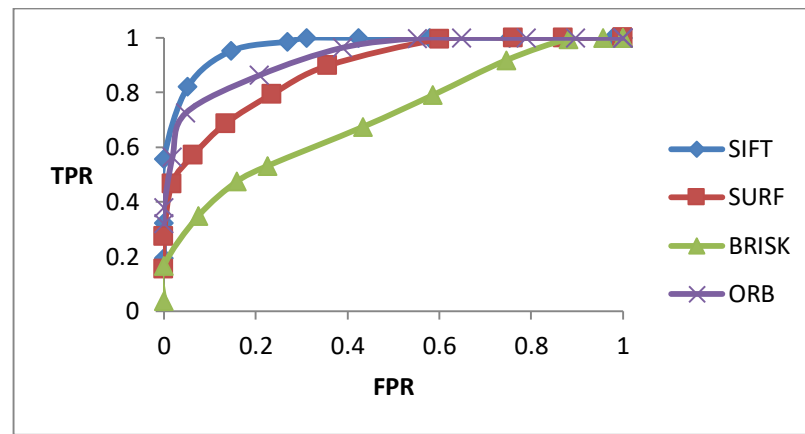


Рисунок 3.2 – ROC криві результатів тестів інваріантності відносно зміни масштабу

Для проведення тестів на перевірку інваріантності до змін нахилу камери підготовлені аналогічним чином 10 відеофайлів. Для зміни нахилу було використано 3 значення широти (0° , 45° , 170°) і 6 значень довготи (0° , 36° , 72° , 108° , 144° , 180°). Результати тесту наведені в таблиці 3.3, графіки ROC кривих відображені на рисунку 3.3.

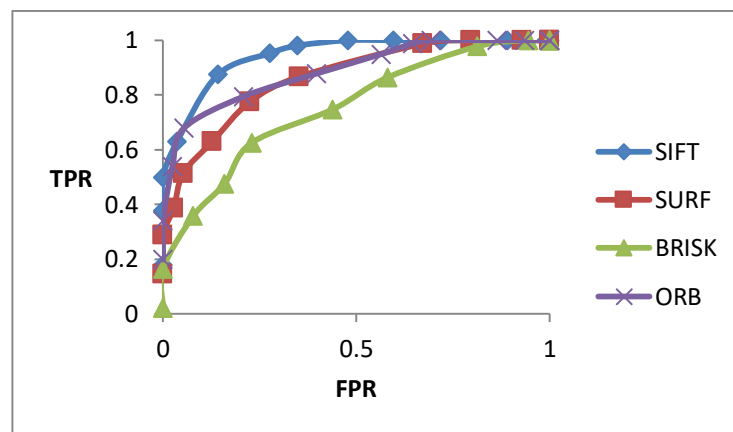


Рисунок 3.3 – ROC криві результатів тестів інваріантності відносно нахилу камери

Значення показників AUC представлені в таблиці 3.4 для чисельного порівняння результатів тесту.

Таблиця 3.3 – Середні значення FPR і TPR результатів тестів інваріантності відносно нахилу камери

Поріг міри схожості	SIFT		SURF		BRISK		ORB	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR
0,95	0	0,1786	0	0,1451	0	0,0213	0	0,1982
0,9	0	0,3765	0	0,2871	0	0,1642	0	0,3401
0,85	0	0,4976	0,0281	0,3873	0,0782	0,3574	0,0243	0,5391
0,8	0,0364	0,6293	0,0521	0,5132	0,1592	0,4754	0,0545	0,6792
0,75	0,1421	0,8744	0,1275	0,6295	0,2298	0,6256	0,2081	0,7942
0,7	0,2764	0,9523	0,2251	0,7754	0,4391	0,7473	0,3971	0,8781
0,65	0,3474	0,9801	0,3531	0,8672	0,5812	0,8641	0,5643	0,9491
0,6	0,4782	1	0,6721	0,9872	0,8132	0,9775	0,6451	0,9881
0,55	0,5962	1	0,7965	1	0,9454	1	0,6934	1
0,5	0,7184	1	0,9287	1	1	1	0,8654	1
0,45	0,8917	1	1	1	1	1	0,9365	1
0,4	0,9914	1	1	1	1	1	1	1
0,35	1	1	1	1	1	1	1	1

Таблиця 3.4 – AUC значення тестів

Тест	Метод			
	SIFT	SURF	BRISK	ORB
Поворот	0,9524	0,9081	0,8184	0,9375
Масштаб	0,9417	0,8913	0,8042	0,9254
Нахил	0,9236	0,8711	0,7958	0,9192

3.3.2 Порівняння методів за ймовірністю виявлення об'єкта

Під ймовірністю виявлення об'єкта p слід розуміти відношення суми кількості вірно виявлених кадрів об'єкта (TP) і кількості вірно

класифікованих кадрів з відсутністю об'єкта (TN) до загальної кількості кадрів (N):

$$p = \frac{TP+TN}{N}.$$

При використанні цієї формули необхідно, щоб розглянута відеопослідовність мала рівну кількість кадрів з наявністю і з відсутністю об'єкта.

Для обчислення ймовірності виявлення кожного методу підсумовані значення результатів TP і TN тестів. Граничне значення міри близькості для кожного методу обране на підставі результатів тестування інваріантності відносно проєктивних спотворень (таблиці 3.1 – 3.3), при яких досягається найменше число помилок. У таблиці 3.5 наведені значення ймовірностей виявлення кожного з методів на тестових відеофайлах. Загальна кількість кадрів $N = 944$.

Таблиця 3.5 – Вірогідність виявлення об'єкта у відеопотоці

Метод	TP (к-ть кадрів)	TN (к-ть кадрів)	Ймовірність виявлення
SIFT	485	362	0,89
SURF	431	317	0,79
BRISK	407	318	0,76
ORB	476	354	0,87

Значення ймовірностей виявлення методів показують, що найкраще з даною задачею справляються методи на основі детекторів SIFT та ORB.

3.3.3 Порівняння швидкодії методів

Для порівняння швидкодії методів пошуку об'єкта у відеопотоці було реалізовано програмну модель, що виконує послідовний пошук заданого об'єкта в відеопотоці за допомогою розглянутих методів спільно з методом RANSAC. Програмну реалізацію детекторів ключових точок та методу RANSAC взято з бібліотеки OpenCV 2.4. Дана програмна модель функціонує без використання паралельної обробки даних.

На рисунках 3.4 і 3.5 представлені результати виконання методів в відеопотоках, що містять і не містять зображення об'єкта відповідно.

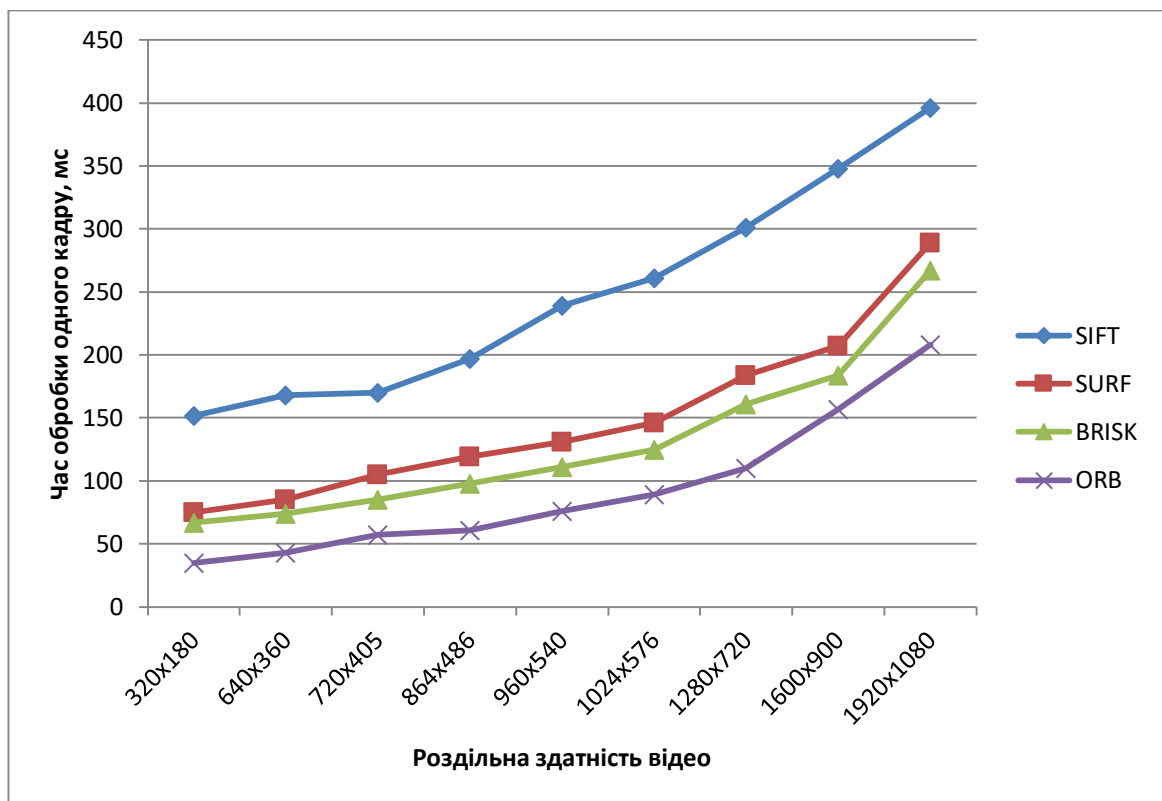


Рисунок 3.4 – Результати роботи методів, з відеопотоками, в яких об'єкт пошуку присутній

При обробці кадру, що містить зображення об'єкта, необхідний додатковий час на виконання функції ідентифікації об'єкта, тоді як при відсутності зображення об'єкта ця функція може бути не виконана у зв'язку з прийняттям рішення про зупинення процесу пошуку в функції знаходження

загальних ключових точок кадру і об'єкта. З цим пов'язано відмінність швидкостей обробки відеопослідовностей із зображенням об'єкта (рис. 3.4) і без нього (рис. 3.5).

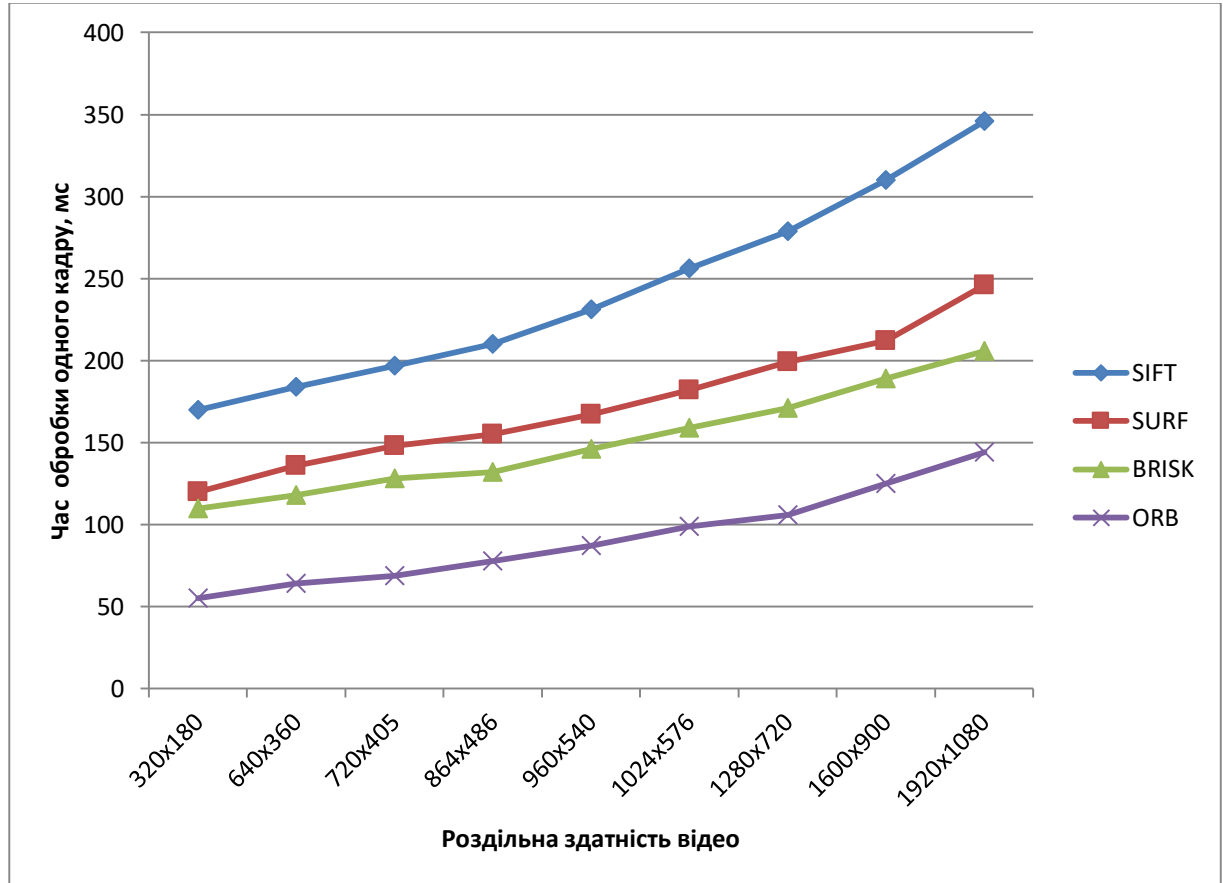


Рисунок 3.5 – Результати роботи методів, з відеопотоками, в яких об'єкт пошуку відсутній

Таким чином, в ході тестування було виявлено, що серед розглянутих детекторів ключових точок, для задачі виявлення об'єкта у відеопотоці найбільше підходить детектор ORB. Незважаючи на те, що даний детектор дещо програє детектору SIFT в плані якості виявлення об'єкта, швидкодія ORB дозволяє використовувати його для аналізу відеопослідовностей в режимі реального часу, що є досить важливим моментом для застосування методу у вирішенні реальних задач.

ВИСНОВКИ

У рамках даного дослідження було реалізовано і протестовано метод знаходження об'єкта у відеопотоці за допомогою детекторів ключових точок. Практична реалізація підтвердила теоретичне обґрунтування використання детекторів ключових точок для застосування в задачах пошуку об'єктів у відеопослідовностях з реальною швидкістю потоку кадрів (25 і більше кадрів в секунду). Серед детекторів рекомендується використовувати детектор ORB, через його швидкодію в поєднанні з високою якістю виявлення об'єкта.

Результати даного дослідження можуть бути використані для побудови програмних систем для вирішення цілого ряду задач, пов'язаних з аналізом відеопослідовностей та пошуком об'єкта у відеопотоці в режимі реального часу.

Результати даного дослідження апробовано на 24-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» [41] та на VI Міжнародній науково-практичній конференції «About the problems of science and practice, tasks and ways to solve them», (Мілан, Італія) [42].

Подальші дослідження можуть бути пов'язані з розробкою ефективного методу виявлення відразу декількох об'єктів. Це дозволить застосувати даний метод для вирішення значно більшої кількості реальних задач.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.
2. Chen B.Y., Shi C., Zhang J., and et al. (2017) Most reliable path-finding algorithm for maximizing on-time arrival probability, *Transportmetrica B: Transport Dynamics*, 5(3), pp. 253-269.
3. Szeliski R. (2010) *Computer Vision: Algorithms and Applications*, London, Great Britain: Springer-Verlag, 957 p.
4. Gorokhovatskyi V., and Tvoroshenko I. (2020) Image Classification Based on the Kohonen Network and the Data Space Modification, *In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020)*, 2608, pp. 1013-1026.
5. Gorokhovatskyi V., Gadetska S., and Stiahlyk N. (2020) Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set, *In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020)*, 2608, pp. 1027-1039.
6. Andrew V. Goldberg (2007) Point-to-Point Shortest Path Algorithms with Preprocessing, *Proceedings of 33rd Conference on Current Trends in Theory and Practice of Computer Science*, pp. 88-102.
7. Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*.
8. Weberruss, J., Kleeman, L., Boland, D., & Drummond, T. (2017, September). FPGA acceleration of multilevel ORB feature extraction for computer vision. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)* (pp. 1-8). IEEE.

9. Peters J.F. (2017) *Foundations of computer vision: Computational Geometry, Visual Image Structures and Object Shape Detection*, Cham, Switzerland: Springer International Publisher, 417 p.
10. Liu, Y., Zhang, H., Guo, H., & Xiong, N. N. (2018). A fast-brisk feature detector with depth information. *Sensors*, 18(11), 3908.
11. Pranata, Y. D., Wang, K. C., Wang, J. C., Idram, I., Lai, J. Y., Liu, J. W., & Hsieh, I. H. (2019). Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images. *Computer methods and programs in biomedicine*, 171, 27-37.
12. Gorokhovatskyi V.O., Tvoroshenko I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.
13. Gorokhovatskyi V.O., Gadetska S.V., and Stiahlyk N.I. (2019) Study of statistical properties of the block representation model for a set of key image descriptors, *Radio Electronics Computer Science Control*, 2, pp. 100-107.
14. Mistry, D., & Banerjee, A. (2017). Comparison of feature detection and matching approaches: SIFT and SURF. *GRD Journals-Global Research and Development Journal for Engineering*, 2(4), 7-13.
15. Gorokhovatsky V.A. (2016) Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions, *Telecommunications and Radio Engineering*, 75(14), pp. 1271-1283.
16. Prakash, C. S., Panzade, P. P., Om, H., & Maheshkar, S. (2019). Detection of copy-move forgery using AKAZE and SIFT keypoint extraction. *Multimedia Tools and Applications*, 78(16), 23535-23558.
17. Sun Y., Yu X., Bie R., and et al. (2017) Discovering time-dependent shortest path on traffic graph for drivers towards green driving, *Journal of Network and Computer Applications*, 83, pp. 204-212.

18. Liang Shen, Hu Shao, Long Zhang, and et al. (2017) The Global Optimal Algorithm of Reliable Path Finding Problem Based on Backtracking Method, *Mathematical Problems in Engineering*, 2017, pp. 1-10.

19. Gorokhovatskyi V.A. (2018) Image classification methods in the space of descriptions in the form of a set of the key point descriptors, *Telecommunications and Radio Engineering*, 77(9), pp. 787-797.

20. Gorokhovatskyi V.O., Tvoroshenko I.S., and Vlasenko N.V. (2020) Using fuzzy clustering in structural methods of image classification, *Telecommunications and Radio Engineering*, 79(9), pp. 781-791.

21. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

22. Leutenegger S., Chli M., and Siegwart R. (2011) BRISK: Binary Robust Invariant Scalable Keypoints, *Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2548-2555.

23. Deshpande, N. T., & Ravishankar, S. (2017). Face Detection and Recognition using Viola-Jones algorithm and Fusion of PCA and ANN. *Advances in Computational Sciences and Technology*, 10(5), 1173-1189.

24. Rublee E., Rabaud V., Konolige K., and Bradski G. (2011) ORB: an efficient alternative to SIFT or SURF, *Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2564-2571.

25. Nong Ye. (2013) *Data Mining: Theories, Algorithms, and Examples*, Florida, USA: CRC Press, 349 p.

26. Xu, Y., Yu, G., Wu, X., Wang, Y., & Ma, Y. (2016). An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery. *IEEE Transactions on Intelligent Transportation Systems*, 18(7), 1845-1856.

27. Sonka M., Hlavac V., and Boyle R. (2014) *Image Processing, Analysis, and Machine Vision*, Atlanta, USA: Thomson-Engineering, 920 p.

28. Duda R.O., Hart P.E., and Stork D.G. (2000) *Pattern classification*, Hoboken, USA: John Wiley & Sons, 738 p.
29. Flah P. (2015) *Machine learning. The science and art of building algorithms that extract knowledge from data*, Moscow, Russia: DMK Press, 400 p., (in Russian).
30. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.
31. Kido, S., Hirano, Y., & Hashimoto, N. (2018, January). Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN). In 2018 International Workshop on Advanced Image Technology (IWAIT) (pp. 1-4). IEEE.
32. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19). p. 6777.
33. Abdulkareem I., Ammar A. Shubber, and Sabah A. (2018) Multi-criteria decision making to select the best monorail route, *Global Journal of Engineering Science and Research Management*, pp. 16-32.
34. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, *Telecommunications and Radio Engineering*, 78(14), pp. 1303-1315.
35. Forstner, W. A feature based correspondence algorithm for image matching [Текст] / W. Forstner // International Archives of Photogrammetry and Remote Sensing. – 1986. – № 5. – 150–166.
36. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, and Vyacheslav Lyashenko (2019) Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach, *International Journal of Emerging Trends in Engineering Research*, 7(11), pp. 460-465.

37. Tvoroshenko Irina, Ahmad M. Ayaz, Mustafa Syed Khalid, Lyashenko Vyacheslav, and Alharbi Adel R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.

38. S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," (2018) *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, 2018, pp. 1-10.

39. Ahmad M. Ayaz, Tvoroshenko Irina, Baker Jalal Hasan, and Lyashenko Vyacheslav (2019) Computational Complexity of the Accessory Function Setting Mechanism in Fuzzy Intellectual Systems, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), pp. 2370-2377.

40. Baggio, D. L., Emami, S., Escriva, D. M., Ievgen, K., Saragih, J., & Shilkrot, R. (2017). *Mastering OpenCV 3*. Packt Publishing Ltd.

41. Зарівчацький Р.М. Порівняльний аналіз сучасних детекторів ключових точок зображень. Радіоелектроніка та молодь у XXI столітті: тези доповідей 24-го Міжнародного молодіжного форуму (Харків, 7–9 квітня 2020 р.). Харків: ХНУРЕ, 2020. Т. 7. С. 30-31.

42. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy, pp. 500-505. DOI: 10.46299/ISG.2020.II.VI