

<https://doi.org/10.31891/2307-5732-2026-363-36>

УДК 621.396: 551.508

ЧИГІНЬ ВАСИЛЬ

Національний університет "Львівська політехніка"

<https://orcid.org/0000-0003-1593-6832>

e-mail: yhygin@gmail.com

МИХАЙЛИШИН ПАВЛО

Національна академія Сухопутних військ України

e-mail: nasv@post.mil.gov.ua

ШВИДКІСНА РЕЄСТРАЦІЯ І РЕАГУВАННЯ ПРИ ПРОНИКАННІ ОБ'ЄКТА У ПОЛЕ ЗОРУ RASPBERRY PI-КАМЕРИ

Створено програмне забезпечення та експериментальну установку для дослідження високошвидкісного проникання малого об'єкта у задану зону контролю та реагування системи, зокрема шляхом видачі відповідного сигналу. Особливістю системи є власний алгоритм аналізу мінімально необхідної кількості пікселів на віртуальному периметрі матриці камери одноплатного комп'ютера, що базується на відстеженні змін їхньої інтенсивності без опрацювання всього чи часткового об'єму матриці. Це дозволяє досягти миттєвої реакції на малопотужних комп'ютерних платформах. Програма реалізована мовою Python із використанням бібліотек OpenCV та NumPy. Для проведення досліджень використали власноруч сконструйований експериментальний стенд на базі одноплатного комп'ютера Raspberry Pi 4 з камерою Pi V2. Стенд включає повзунок для кріплення і контролюваного переміщення різнокольорових макетів об'єктів, зокрема у формі трикутника, диска, квадрата, через поле зору камери. У ролі поля переміщення об'єкта використали різнокольорові аркуші паперу (білі, зелені, сірі тощо). Швидкість переміщення макета об'єкта обчислювалася зі співвідношень відстаней від об'єктива камери до площини фону макета і до площини переміщення реальних об'єктів (спортивних м'ячів, птахів, тварин тощо), а також швидкостей цих об'єктів, і становила порядку 1–10 см/с. Програма записує у файли детальні параметри пікселів (координати, параметри кольорів R, G, B, яскравість) у моменти виявлення об'єкта при його проникненні через периметр спостереження. Отримані масиви чисел (0–255) використали у графічному редакторі типу Origin для візуалізації та аналізу процесів проникання. Створені та випробувані програмне забезпечення й експериментальний стенд дозволили досягнути високої швидкодії виявлення об'єкта при прониканні у зону спостереження (близько 1 мілісекунди) при опрацюванні мінімальної кількості відеопікселів матриці камери на віртуальному периметрі контролю (близько 1000 точок), а також забезпечити вмикання сигналу реагування на проникання за час порядку 1,5 мілісекунди (що є значно меншим від тривалості одного кадру, яка у нашому випадку дорівнює 33 мілісекунди)

Ключові слова: високошвидкісна реєстрація, програмне забезпечення, проникнення об'єкта, віртуальний периметр, Raspberry Pi, аналіз пікселів, Python.

CHYHIN VASYL

National University "Lviv Polytechnic"

MYKHAYLYSHYN PAVLO

Hetman Petro Sahaidachny National Army Academy

HIGH-SPEED REGISTRATION AND REACTION TO OBJECT PENETRATION INTO THE RASPBERRY PI CAMERA FIELD OF VIEW

Software and an experimental setup have been created to investigate the high-speed penetration of a small object into a specified control zone and the system's reaction, particularly by issuing a corresponding signal. A feature of the system is the developed algorithm for analyzing the minimal necessary number of pixels on the virtual perimeter of the single-board computer camera matrix. This is based on tracking changes in their intensity without processing the entire or partial volume of the matrix. This approach allows for instantaneous reaction on low-power computer platforms. The program is implemented in Python using the OpenCV and NumPy libraries. For the research, a custom-designed experimental stand based on the Raspberry Pi 4 single-board computer with a Pi V2 camera was used. The stand includes a slider for attaching and controlled movement of multi-colored object mock-ups, specifically in the form of a triangle, disk, and square, through the camera's field of view. Multi-colored sheets of paper (white, green, gray, etc.) were used as the object movement background. The movement speed of the object mock-up was calculated based on the ratios of distances from the camera lens to the background plane of the mock-up and to the movement plane of real objects (sports balls, birds, animals, etc.), as well as the speeds of these objects, and was of the order of 1–10 cm/s. The program records detailed pixel parameters (coordinates, color parameters R, G, B, brightness) to files at the moments of object detection when it penetrates the observation contour. The obtained arrays of numbers (0–255) are used in a graphical editor like Origin for visualization and analysis of penetration processes. The developed software and experimental stand achieved high-speed object detection (approximately 1 millisecond) by processing a minimal number of pixels on the virtual control perimeter (about 700 points), with a reaction signal trigger time of approximately 1.5 milliseconds (which is significantly less than the single frame duration of 33 milliseconds).

Keywords: high-speed registration, software, object detection, virtual perimeter, Raspberry Pi, pixel analysis, Python.

Стаття надійшла до редакції / Received 02.01.2026

Прийнята до друку / Accepted 11.02.2026

Опубліковано / Published 26.03.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Чигінь Василь, Михайлишин Павло

Постановка проблеми

В останні десятиліття проводилися дослідження, присвячені створенню ефективних систем для візуального виявлення та динамічного супроводу рухомих об'єктів, переважно великих розмірів (люди, автомобілі), що рухаються з невисокими швидкостями. Актуальність таких систем зумовлена широким спектром їх застосування, зокрема аналізом автомобільного трафіку, відеоспостереженням в охоронних зонах,

вивченням поведінки тварин тощо. При цьому класичні автоматизовані підходи здебільшого працюють із зображеннями на повних матрицях відеокамер, тому потребують значних обчислювальних ресурсів та високоякісного устаткування.

У більшості наявних рішень для моніторингу швидкісних спортивних об'єктів (наприклад, м'ячів у грі) або вивчення поведінки птахів керування системами покладається на оператора. Це призводить до помилок, швидкої втоми під час тривалого спостереження та знижує загальну надійність комплексу. Водночас досягти необхідної швидкості реакції при використанні простих та дешевих комп'ютерних платформ, необхідних для масового впровадження, за допомогою класичних методів неможливо.

У зв'язку з цим для створення швидкісної автоматичної системи реєстрації проникання малорозмірних об'єктів та усунення недоліків ручного керування, поставлено за мету дослідити систему на базі доступних пристроїв (одноплатний комп'ютер Raspberry Pi 4, Pi-камера). Дослідження передбачає створення власного програмного забезпечення з алгоритмом аналізу мінімально необхідної кількості пікселів на віртуальному периметрі матриці. Такий підхід базується на відстеженні змін інтенсивності пікселів, що дозволяє досягти миттєвої реакції за рахунок низької обчислювальної складності.

Аналіз останніх досліджень і публікацій

Попередні роботи авторів [1–4] присвячені створенню ефективних систем для візуального виявлення та динамічного супроводу рухомих об'єктів великих розмірів при невисоких швидкостях переміщень всередині поля зору відеокамер. Однак такі системи працюють із зображеннями на повних матрицях відеокамер, тому потребують значних обчислювальних ресурсів і дозволяють реєструвати переміщення об'єктів з невисокими швидкостями. Такі ж задачі візуального виявлення та динамічного супроводу рухомих об'єктів вирішувалися і в працях інших авторів. Актуальність таких систем зумовлена широким спектром їх застосування, зокрема аналізом автомобільного трафіку [5], вивченням поведінки тварин [6]. Зокрема, важливим є дистанційний екологічний моніторинг та дослідження біорізноманіття. Традиційні засоби спостереження за тваринами, так звані "фотопастки" (Camera Traps), здебільшого використовують пасивні інфрачервоні сенсори (PIR) для активації камери. Однак такі сенсори мають суттєвий недолік — час затримки спрацювання (trigger latency), який зазвичай становить від 0,5 до 1 секунди. Цього достатньо для фіксації великих тварин, але абсолютно недостатньо для реєстрації високошвидкісних об'єктів, таких як птахи у польоті або комахи, які встигають перетнути поле зору камери до моменту її активації.

У попередній нашій публікації [4] зроблено детальний аналіз сучасних одноплатних комп'ютерів типу Raspberry Pi, Odroid і LattePanda, підходів до реалізації систем візуального захоплення, виявлення та стеження за кольоровими об'єктами із використанням фотокамер. Проте варто зазначити, що всі відомі нам фотосистеми потребують потужного обладнання, а аналіз існуючих методів свідчить про недостатню кількість повністю відкритих та інтегрованих систем, які поєднували б функції виявлення і фотозахоплення рухомих об'єктів, особливо малих розмірів з використанням сучасних бортових комп'ютерів та алгоритмів опрацювання відеоданих у реальному часі.

На нашу думку, сучасні високоточні алгоритми є надто "важкими" для бюджетних платформ, а існуючі прості системи не забезпечують мілісекундної швидкості реакції. Це обумовлює необхідність створення спеціалізованого програмного забезпечення, яке б не виконувало повнокадрового аналізу на користь оптимізованої обробки ключових зон (віртуального периметра) для досягнення максимальної швидкодії.

Метою роботи є створення апаратного та програмного забезпечення для досягнення максимальної швидкодії реєстрації проникання об'єктом заданого периметру спостереження шляхом опрацювання мінімально необхідної кількості відеопікселів матриці камери (аналіз пікселів лише на віртуальному периметрі контролю) та посилання вчасного сигналу на реагування.

Виклад основного матеріалу

1. Експериментальна частина

Методика досліджень. Для проведення досліджень фотозахоплення макетів об'єктів із високою швидкістю реакції на їхнє проникання крізь поле зору камери Raspberry Pi за умови низького обчислювального навантаження створено експериментальну вимірювальну установку (Рис. 1). Вона передбачає фіксацію часових залежностей колірних параметрів об'єкта і фону, які характеризують процеси виявлення та реєстрації проникання об'єкта в зону контролю.

Експериментальна установка. На рисунку 1 зображено основні компоненти установки: 1 – одноплатний комп'ютер Raspberry Pi 4 з камерою (2); 3 – дві опори для фіксації верхньої пластини (4) з комп'ютером і камерою, а також основної пластини (5) з повзунком (7). Повзунок забезпечує переміщення макета об'єкта (6) у площині основи (5) перпендикулярно до оптичної осі камери (2). Макети об'єктів (6) виготовили у вигляді диска, трикутника та квадрата різних розмірів і кольорів (чорний, червоний, синій). На площину основи (5) розміщували плоский аркуш із різними кольорами фону – білим, зеленим, сірим.

Для забезпечення роботи комп'ютера до його USB-портів приєднували монітор, мишу та клавіатуру. Живлення подавали від перетворювача з допустимим струмом 5 А.

При вимірюваннях камера 2 (Raspberry Pi v2) здійснювала захоплення зображень макета об'єкта (6) із частотою 30 кадрів на секунду. Зображення макета об'єкта і його переміщення в межах заданих параметрів периферійної зони контролю опрацьовувалися власним програмним забезпеченням. Дослідження процесів перетинання заданої зони проводили з використанням різних макетів (чорний трикутник на білому тлі, кольорові

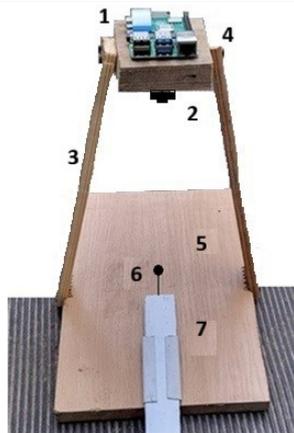


Рис.1. Фотографія експериментальної установки

- 1 – одноплатний мінікомп'ютер Raspberry Pi 4;
 2 - камера Raspberry Pi v2 №,
 3 – дві вертикальні опори висотою по 33 см, забезпечують віддалю об'єкта камери до площини основи 27 см,
 4 – верхня пластина для кріплення комп'ютера 1 і камери 2;
 5 – основна нижня пластина (21x30 см2) для кріплення листка-макета тла і повзунка (7) з макетом об'єкта (6)



Рис.2. Схема опрацьованого алгоритму

об'єкти на зеленому тлі і т.п.). Швидкість переміщення макета об'єкта обчислювалася із співвідношень відстаней від об'єкта камери до площини фону макета і до площини переміщення реальних об'єктів (спортивних м'ячів, птахів, тварин тощо), а також швидкостей цих об'єктів, і становила порядку 1–10 см/с.

2. Програмне забезпечення

Алгоритм роботи програми. Опрацьовано алгоритм високошвидкісного виявлення проникання об'єкта у поле зору камери, який ґрунтується на опрацьованні мінімально необхідної кількості пікселів матриці. Програмне забезпечення реалізовано мовою Python із використанням бібліотек OpenCV та NumPy, що забезпечує високу ефективність обчислень. В основу роботи покладено створення віртуального периметра. Замість опрацьовання сотень тисяч пікселів усієї матриці, система зчитує координати, значення R, G, B та яскравість лише з попередньо заданого набору (кілька сотень пікселів), чим підвищує на два порядки швидкість системи.

Алгоритм включає виконання таких кроків:

1) Ініціалізація: обчислення координат пікселів для заданого периметра.

2) Зчитування даних: з відеопотоку отримуються значення кольорів лише за координатами заданих пікселів.

3) Опрацьовання та аналіз: обчислюється усереднена яскравість пікселів. Система порівнює параметри у різних областях периметра. Якщо різниця перевищує заданий поріг (у відсотках), система переходить до блоку перевірки «Знайдено групу точок», які вказують на належність до об'єкта. При цьому відбувається фільтрація шуму і перевіряється достовірність сигналу (чи зафіксована зміна у групі сусідніх пікселів).

4) Реагування: при підтвердженні події миттєво генерується сигнал, який передається на вихід через GPIO (наприклад, для світіння LED-індикатора) та фіксується час реакції.

5) Логування: створений механізм "розумного" логування фіксує дані лише у моменти події. У файли записується повний набір параметрів: точна часова мітка, час реакції, просторові координати (пікселі та кути), фотометричні дані (RGB, яскравість).

На рисунку 2 зображено принципи роботи опрацьованого алгоритму. Після завершення операцій цикл повторюється, забезпечуючи безперервний моніторинг. Аналіз записаних даних дозволяє виявити обчислювальну складність та оптимізувати параметри.

3. Результати експериментальних досліджень

При вимірюваннях фотометричних характеристик пікселів на віртуальному периметрі матриці камери у моменти його перетинання макетами об'єктів отримали просторово-часові залежності параметрів R, G, B і яскравості кожного пікселя в залежності від його номера. Отримані масиви чисел (0–255) використали у графічному редакторі типу Origin для візуалізації та аналізу процесів проникання.

Як приклад, на рисунку 3 нанесені графіки таких залежностей для R, G, B компонент від номера пікселя при перетинанні периметра чорним об'єктом на білому тлі. Як видно з графіків, при використанні контрастних кольорів тла і об'єкта (у даному випадку - біле тло і чорний об'єкт) спостерігається чіткий "провал" R, G, B компонент пікселів в області перетинання периметра. Значення цих величин падають від рівня тла (порядку 200 одиниць) до значень, які задаються сірим кольором об'єкта (порядку 75 одиниць), незалежно від кольору (R, G чи B).

Ширина цього "провалу" змінюється з часом від моменту торкання макетом межової лінії віртуального периметра до повного перетинання макетом об'єкта цієї межі і корелює з розміром частини макета - трикутника, який перетинає периметр. Із графіків видно, що ця ширина змінюється в межах від 0 до 130 пікселів.

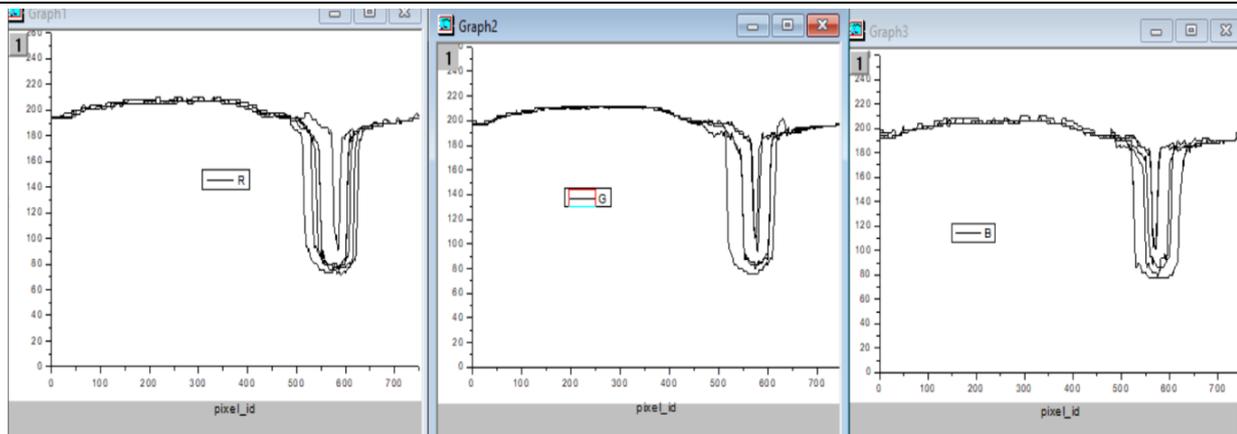


Рис.3 Графіки залежностей R, G, B компонент пікселів від їх номерів при перетинанні віртуального периметра білого тла чорним об'єктом – трикутником

Ширина цього "провалу" змінюється з часом від моменту торкання макетом межевої лінії віртуального периметра до повного перетинання макетом об'єкта цієї межі і корелює з розміром частини макета - трикутника, який перетинає периметр. Із графіків видно, що ця ширина змінюється в межах від 0 до 130 пікселів.

Завдяки оптимізації коду та використанню лише необхідних точок для аналізу, час опрацювання одного циклу проникання об'єкта у формі чорного трикутника крізь периметр зони спостереження та прийняття рішення про реагування становить близько 1 мілісекунди і є суттєво меншим від тривалості одного кадру, який у даному випадку становить 1/30 сек, або 33 мілісекунди. Оскільки існує можливість програмного підвищення частоти кадрів фотографування за допомогою Pi-камери, ефективність її використання у комплексі з мінікомп'ютером Raspberry 4 можна значно підвищити, при цьому значно підвищити також ефективність реєстрації швидких невеликих об'єктів.

Висновки

Створено програмне забезпечення та експериментальну установку для дослідження високошвидкісного проникання малого об'єкта у задану зону контролю, створеної камерою Raspberry Pi V2. Особливістю системи є власний алгоритм аналізу мінімально необхідної кількості пікселів на віртуальному периметрі матриці без опрацювання всього об'єму кадру. Це дозволило досягти миттєвої реакції на малопотужній платформі Raspberry Pi 4. Результати експериментів підтвердили, що система забезпечує швидкодію виявлення близько 1 мілісекунди та час реакції порядку 1,5 мілісекунди при опрацюванні близько 1000 точок периметра. Робота є актуальною для задач моніторингу швидкісних об'єктів та охорони периметрів, а отримані результати мають практичну цінність для створення бюджетних автоматизованих систем.

Література

1. Василь Чигінь, Павло Михайлишин. Експериментальний безпілотний авіаційний комплекс для фотозахоплення. *Вісник Хмельницького Національного університету*. 2019, № 2 (271), с. 202-205.
2. Василь Чигінь, Павло Михайлишин. Експериментальні дослідження безпілотного авіаційного комплексу при фотозахопленні. *Вісник Хмельницького Національного університету*. 2020, № 3 (285), с. 170-174.
3. В. Чигінь, М. Черненко. Експериментальна система і програмне забезпечення для дослідження фотопереслідування рухомих об'єктів безпілотним літальним апаратом. *Вісник Хмельницького Національного університету*. 2020, № 4 (287), с. 84-88.
4. Василь Чигінь, Василь Науличний, Павло Михайлишин. Система фотозахоплення та переслідування об'єктів у двох координатах. *Вісник Хмельницького Національного університету*. 2025, № 4 (355), с. 498-505.
5. Computer Vision Methods for Vehicle Detection and Tracking: A Systematic Review and Meta-Analysis. [Електронний ресурс]. – Режим доступу: <https://www.mdpi.com/2076-3417/15/22/12288>.
6. Real-Time Computer Vision Bird Feeder. [Електронний ресурс]. – Режим доступу: <https://www.tylerbisk.com/meng/>.

References

1. Chyhin V., Mykhailyshyn P. (2019). Experimental unmanned aerial vehicle for photo capture. *Bulletin of Khmelnytsky National University*, 2 (271), 202-205
2. Chyhin V., Mykhailyshyn P. (2020). Experimental studies of unmanned aerial vehicles during photo capture. *Bulletin of Khmelnytsky National University*, 3 (285), 170-174.
3. Chyhin V., Cherenko M. (2020). Experimental system and software for investigation of photo-pursuit of moving objects by unmanned aerial vehicle. *Bulletin of Khmelnytsky National University*, 4 (287), 84-88.
4. Chyhin V., Naulychnyi V., Mykhailyshyn P. (2025). System of image-based object capturing and following in two dimensions. *Bulletin of Khmelnytsky National University*, 4 (355), 498-505 [in Ukrainian].
5. Computer Vision Methods for Vehicle Detection and Tracking: A Systematic Review and Meta-Analysis Retrieved from: <https://www.mdpi.com/2076-3417/15/22/12288>.
6. Real-Time Computer Vision Bird Feeder. Retrieved from: <https://www.tylerbisk.com/meng/>.