

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**ЗБІРНИК НАУКОВИХ ПРАЦЬ**  
за матеріалами ІХ Всеукраїнської науково-практичної конференції  
**«ЕЛЕКТРОННІ ТА МЕХАТРОННІ СИСТЕМИ:**  
**ТЕОРІЯ, ІННОВАЦІЇ, ПРАКТИКА»**

10 листопада 2023 року



**Полтава 2023**

УДК 004.93

*М.Ю. Пророк, студент,*

*Б.Р. Боряк, к.т.н.*

*Національний університет «Полтавська політехніка імені Юрія Кондратюка»*

## **ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ВІДСТЕЖЕННЯ НА ВІДЕО З ОБ'ЄКТАМИ З ВИСОКОЮ ДИНАМІКОЮ РУХУ, НАЯВНИХ В OPENCV 4.8**

Відстежування об'єктів на відео є важливою задачею в області комп'ютерного зору. Інтеграція систем технічного зору в мобільні платформи та їх часткова або повна автоматизація за допомогою комп'ютерного зору надає багато переваг перед ручним керуванням. Однією із з ключових задач є задача відслідковування (трекінгу), тобто визначення руху та положення об'єкта на послідовних кадрах відео. У бібліотеці OpenCV [1] існує 7 алгоритмів (трекерів) для вирішення цієї задачі. У цьому дослідженні ми порівняли та дослідили поведінку різних алгоритмів відслідковування в умовах швидких та непередбачуваних рухів об'єктів на відео та визначили найкращий із запропонованих в бібліотеці OpenCV для подальшого інтегрування в мобільну платформу.

Для проведення дослідження були вибрані наступні алгоритми відслідковування, доступні в OpenCV [2]: MIL (Multiple Instance Learning), CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability), KCF (Kernelized Correlation Filters), BOOSTING, MedianFlow, MOSSE (Minimum Output Sum of Squared Error), TLD (Tracking, Learning, and Detection). Для кожного алгоритму було проведено аналіз, обрано початкові позиції об'єкта на першому кадрі, кожен алгоритм було випробувано на відео, з метою відстеження положення об'єкта на кожному кадрі. Результати були оцінені за допомогою таких метрик, як середня відстань між центром реального об'єкта та центром об'єкта, визначеного алгоритмом, та час, необхідний для обробки даних кожного кадра.

Для тестування було відібрано 14 відео з FPV-дронів (720x720px) з визначеними позиціями об'єктів, які відслідковувались. На мові програмування C++ було написано програму, за допомогою якої користувач обирає ROI (Rect Of Interest) для всіх трекерів одночасно. На кожному кадрі відеофайлу програма використовує вищезазначені алгоритми відслідковування, визначає різницю між положенням об'єкту, що визначається за допомогою трекера, та реальним положенням, а також час, витрачений на обробку одного кадру і записує ці дані в лог-файл. В свою чергу програма на Python зчитує дані з лог-файлу, систематизує їх, будує графіки та вираховує середні значення часу обробки одного кадру, помилки трекера, кількість втрат та відновлення відслідковування. Отримані дані представлені в таблицях 1 і 2. Результати показують, що ефективність алгоритмів може значно варіюватися в залежності від типу відеофрагменту. Наприклад, деякі алгоритми можуть бути ефективними на відео

зі спокійними об'єктами, але не ефективні при швидких та непередбачуваних рухах.

Таблиця 1. Середня похибка відстеження (рх)

Назва відеофайлу	Назва алгоритму						
	MIL	CSRT	KCF	BOOSTING	MedianFlow	MOSSE	TLD
drone2	35.9	∞	∞	113.2	31.3	∞	∞
drone3	45.4	∞	147.4	37.5	125.3	97.1	∞
drone4	18.6	18.3	∞	16.9	∞	∞	65.3
drone5	21.3	28.4	∞	39.9	107.6	∞	∞
drone6	29.3	∞	∞	57.6	58.9	35.4	53.5
drone7	38.0	117.7	∞	50.8	124.6	∞	52.9
drone8	60.2	14.0	∞	21.2	43.1	∞	∞
drone9	12.7	15.0	∞	15.2	118.4	∞	26.3
drone10	12.5	14.2	∞	13.3	∞	∞	∞
drone11	∞	∞	∞	∞	72.5	∞	150.7
drone12	37.7	7.5	∞	11.0	∞	∞	∞
drone13	9.4	9.9	∞	42.2	110.9	∞	150.4
drone14	8.1	∞	∞	∞	29.3	∞	∞
drone15	∞	∞	∞	6.1	30.0	∞	64.6
Середня похибка	27.4	28.1	147.4	35	77.5	66	80.5
Сер.% втрачених кадрів	0%	20.2%	12.7%	0%	7.8%	4.1%	0,3%

(середні значення похибки та відсотка втрачених кадрів розраховувалися лише для тих відео, на яких трекер зміг відслідкувати об'єкт)

Таблиця 2. Середній час обробки одного кадру (мс)

Назва відеофайлу	Назва алгоритму						
	MIL	CSRT	KCF	BOOSTING	MedianFlow	MOSSE	TLD
drone2	243.9	∞	∞	415.0	48.2	∞	∞
drone3	262.4	∞	158.6	698.0	153.4	73.9	∞
drone4	237.0	166.6	∞	432.6	∞	∞	949.6
drone5	215.0	122.7	∞	372.5	130.9	∞	∞
drone6	236.4	∞	∞	533.9	72.7	13.8	598.1
drone7	255.9	117.7	∞	475.5	144.1	∞	853.0
drone8	260.8	149.4	∞	395.1	53.7	∞	∞
drone9	206.5	103.4	∞	338.1	89.8	∞	966.0
drone10	193.3	119.2	∞	341.7	∞	∞	∞
drone11	∞	∞	∞		44.0	∞	∞
drone12	220.7	123.6	∞	393.8	∞	∞	∞
drone13	218.9	110.7	∞	372.0	81.8	∞	1276.3
drone14	195.3	∞	∞	∞	36.7	∞	∞
drone15	∞	∞	∞	308.1	41.8	∞	961.2
Сер. час обробки	228.8	126.6	158.6	426	81.5	43.9	1076.2

Аналізуючи результати дослідження варто зазначити, що вибір алгоритму відслідковування повинен враховувати конкретні вимоги завдання та характер відеоматеріалу. KCF та MOSSE швидко обробляють кадр (158.6 мс та 43.9 мс відповідно) та добре справляються з відстеженням об'єктів з рівномірним та плавним рухом, без різких переміщень об'єкта, що відслідковується, але для відслідковування об'єктів з непередбачуваною траєкторією ці алгоритми можуть втрачати об'єкт. TLD, у порівнянні з іншими алгоритмами відслідковування, більш ефективний для обробки відео, де об'єкт може зникати за перешкодою або виходити за рамки кадру, найкраще відновлює відслідковування, але він має великий час обробки кадру (1076.2 мс) та помилка відстежування більше ніж у інших трекерів. MedianFlow досить добре відстежує об'єкти з нерівномірним рухом, але точність дуже залежить від початкової вибраної ROI, потрібно точно виділити рамки об'єкту без захвату заднього фону. CSRT та BOOSTING, за результатами дослідження, забезпечують відслідковування об'єктів, але алгоритм BOOSTING дуже повільно обробляє кадр (426 мс), а CSRT частіше втрачає об'єкт (20.2%). Алгоритм MIL найбільше підходить під нашу ціль інтеграції в мобільну платформу завдяки балансу швидкодії (середній час обробки = 228.8 мс) та точності (середня похибка = 27.4 пкс, середній % втрачених кадрів = 0%) в задачах відслідковування. Він дозволяє відслідковувати об'єкти, навіть в умовах, де рух об'єкта може бути швидким і непередбачуваним. Один з його сильних боків полягає в тому, що він використовує інформацію з різних інстанцій (прикладів) об'єкта, що допомагає уникнути втрати відслідковування при різких переміщеннях.

#### ЛІТЕРАТУРА:

1. *OpenCV Tracking API Documentation [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: [https://docs.opencv.org/4.x/d9/df8/group\\_\\_tracking.html](https://docs.opencv.org/4.x/d9/df8/group__tracking.html)*
2. *Object Tracking using OpenCV (C++/Python) Tutorial [Електронний ресурс] – 2017. – Режим доступу до ресурсу: <https://learnopencv.com/object-tracking-using-opencv-cpp-python/>*

#### COMPARATIVE ANALYSIS OF TRACKING HIGH-DYNAMIC OBJECTS VIDEO USING ALGORITHMS AVAILABLE IN OPENCV 4.8

*M. Prorok, student,*

*B. Boriak, Ph.D.*

*National University «Yuri Kondratyuk Poltava Polytechnic»*