

МЕТОДИ ВІДСТЕЖЕННЯ ДОВІЛЬНОЇ КІЛЬКОСТІ ОБ'ЄКТІВ У РЕАЛЬНОМУ ЧАСІ НА МОБІЛЬНІЙ ПЛАТФОРМІ

Д. О. Кушнір

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин
E-mail: dmytro.o.kushnir@lpnu.ua

© Кушнір Д. О., 2023

Розглянуто проблему вибору методів відстеження розпізнаних об'єктів (ВО) у реальному часі для систем з обмеженими апаратними можливостями. Визначено, що для таких систем доречно інтеграція методів відстеження у пристрій, оминаючи використання передавання даних через мережу інтернет.

Розглянуто відомі методи відстеження довільної кількості об'єктів у реальному часі. Серед метрик оцінювання ефективності використано: MOTA, MOTP, F1, MT, ML, ID, FM.

На основі аналізу ефективності таких методів за згаданими вище метриками запропоновано використати метод відстеження V-IOU для відстеження розпізнаних об'єктів на мобільній платформі у мобільній кіберфізичній системі.

Ключові слова: V-IOU; відстеження довільного класу об'єктів; мобільна платформа; реальний час; MOTA; MOTP; кіберфізична система.

Вступ

Досвід використання UAV (Unmanned Aerial Vehicle) дронів на війні в Україні показав високу ефективність автономних безпілотників Saker Scout, які можуть автоматично розпізнавати, відстежувати та наводитися на заданий клас об'єктів без втручання людини [1]. Тому визначення ефективних автономних методів відстеження довільної кількості об'єктів для кіберфізичних систем з обмеженими апаратними можливостями надзвичайно актуальне.

Проаналізовані методи використано у дисертації “Методи та засоби пошуку та розпізнавання об'єктів у реальному часі на мобільній платформі” [2].

Огляд мобільних платформ для завдань відстеження об'єктів

Мобільна платформа (МП) – пристрій або операційна система з апаратними можливостями, обмеженими порівняно зі стаціонарними приладами.

Операції відстеження залежно від пропонованого алгоритму можуть по-різному навантажувати процесор вихідного пристрою, проте переважно такі алгоритми цілком підходять для роботи на МП. На рис. 1 зображено набір платформ, на яких можна виконувати операції відстеження об'єктів, зокрема на МП [2].

Дамо визначення кожному із понять:

- Мобільна операційна система (МОС) – операційна система для мобільних телефонів, розумних годинників, планшетів тощо.
- Вбудована система (embedded system, ВС) – спеціалізований обчислювальний пристрій для виконання обмеженої кількості завдань з обмеженнями системи реального часу (СРЧ).

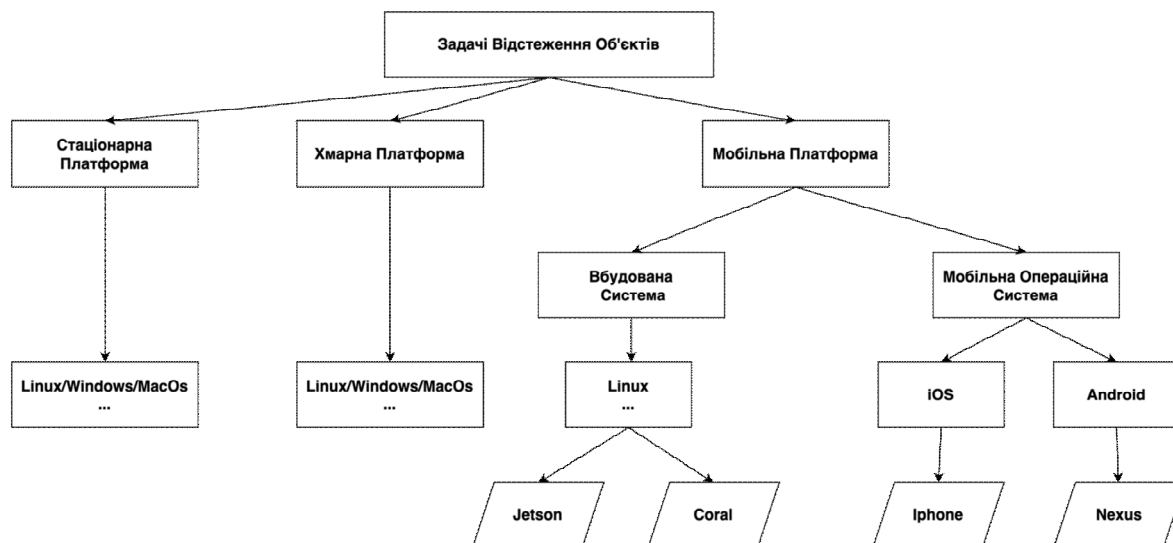


Рис. 1. Типи платформ, які підтримують задачі ВО.

Також зазначено деякі приклади операційних систем та прикладів пристроїв

- СРЧ – система, яка впливає на досліджене середовище в межах заданих часових проміжків, для якої важливий час отримання результату. Метрика таких систем тісно пов'язана з опрацюванням відеозображень для задач ВО [3].

- Відеозображення – технологія передавання візуальної інформації у вигляді потоку даних у СРЧ. Можна виділити такі характеристики: FPS, роздільна здатність, співвідношення сторін екрана, ширина відеопотоку.

- Хмарна платформа (ХП) – модель забезпечення ресурсів комп'ютерної системи без прямого доступу користувача. Великі хмарні сервіси забезпечують різні рівні захищеності, масштабування, реплікації та резервування даних.

- Стационарна платформа (СП) – не мобільна платформа, зазвичай її стационарно встановлюють у приміщенні. Переважно має більші апаратні можливості, порівняно з МП.

Також варто виділити кросплатформні рішення для МОС. Такі системи дають змогу запускати той самий додаток на таких платформах, як Android та iOS, використовуючи різні типи компіляторів залежно від типу МП, наприклад Just In Time Compiler (JIT) та Ahead Of Time Compiler (AOT). Такі підходи дають можливість істотно полегшити розроблення програмного забезпечення одночасно на декілька МП, проте специфічність завдань РО на таких пристроях не завжди вдається забезпечити кросплатформними рішеннями [2].

Аналіз літературних джерел показав, що для виконання завдань ВО для довільної кількості об'єктів доречно використати МОС iOS, які показали високі показники швидкодії та продуктивності для завдань ВО в МП. Причому систему доречно реалізовувати як повністю автономну, оскільки підключення до інтернету для завантаження даних не є гарантовано постійним [2, 4–6].

Методи відстеження об'єктів у реальному часі

Відстеження об'єктів (ВО) – процес пошуку одного чи більше об'єкта на відеозображенні. Головне завдання алгоритмів відстеження – це послідовний аналіз кадрів відео для оцінювання параметрів руху та положення вихідного об'єкта [8]. Серед основних проблем, які виникають під час відстеження, можна виділити: визначення траєкторії руху за високого рівня FPS у реальному часі; визначення положення об'єкта на цільовому відеопотоці, у разі великої кількості вхідних

об'єктів, які можуть перекривати один одного. Отже, створюючи моделі ВО, важливо визначити, як може змінюватись об'єкт під час руху в реальному часі [7, 8].

Системи ВО можна розділити на два основні етапи:

Target Representation and Localization (представлення та локалізація об'єкта).

Цей етап є послідовним та висхідним, тобто наступні кроки ВО не зачіпають попередніх. Складність алгоритму для обчислень на цьому етапі невисока [7]. Серед прикладів таких алгоритмів можна виділити:

- Відстеження точкових особливостей сцени (Point feature tracking): полягає у отриманні максимально точних послідовностей координат проекції точок вхідного відеопотоку [9].
- Відстеження контурів (Contour Tracking): полягає у пошуку граничної межі об'єкта. Ітераційно модифікує початковий контур, ініціалізований з попереднього кадру, на його нове положення в поточному кадрі. Цей підхід розвиває контур за рахунок мінімізації енергії контуру, використовуючи алгоритм градієнтного спуску (gradient descent) [9].

Водночас розглядають різні підходи до ВО у реальному часі за допомогою моделей згорткових нейронних мереж (ЗНМ) (рис. 2) [10].

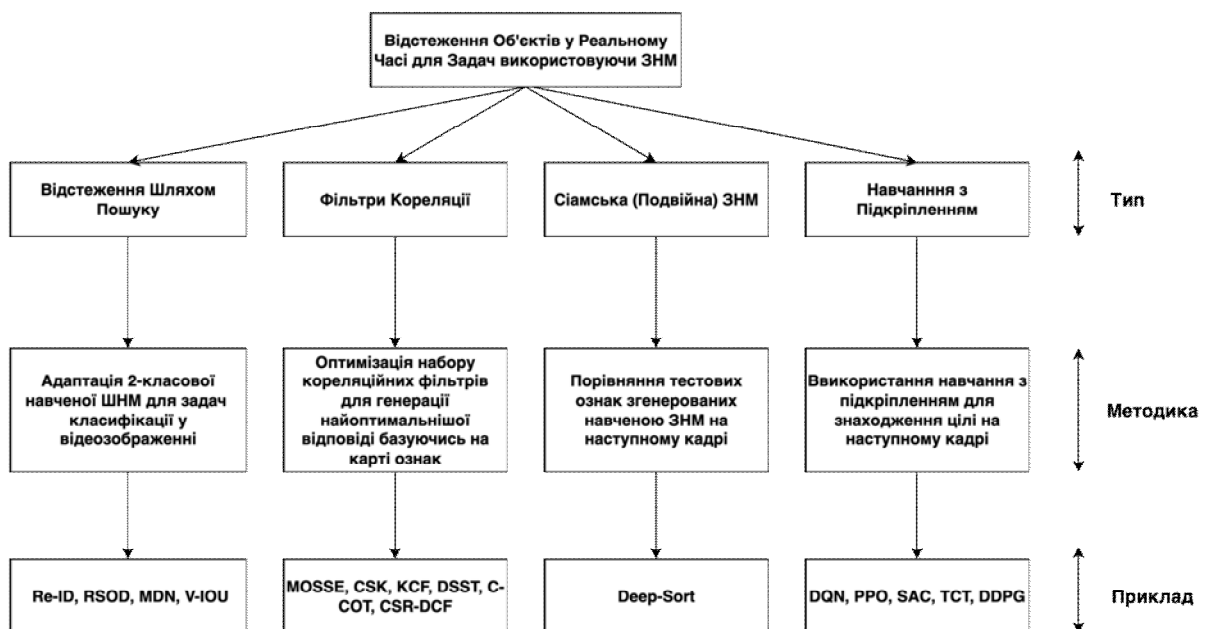


Рис. 2. Типові приклади методів ВО

Із розвитком підходів глибокого навчання (ГН) зростає попит на методи пошуку та відстежування об'єктів за допомогою ЗНМ. Наприклад, у змаганні на найефективніший алгоритм 2018 Visual Object Tracking (VOT 2018) усі призіві місця отримали алгоритми, які використовують ГН [10].

Такі алгоритми умовно можна розділити на чотири групи: відстеження із пошуком (Tracking by Detection); фільтри кореляції; сіамські (подвійні) ЗНМ; навчання із підкріпленням.

Відстеження із пошуком – підхід для багатокласового відстеження, який полягає у відбиранні ознак з вхідного зображення під час аналізу регіонів такого зображення. Позитивні та негативні випадки в навчальних вибірках відбирають, оцінюючи Intersection Over Union (IoU) з основною істиною. Типовою проблемою для таких систем є обмежена продуктивність базового детектора, який може призвести до хибно позитивних (FP) і пропущених виявлень. Розглянемо приклади реалізації такого підходу.

Зокрема, модель повторної ідентифікації (Re-ID) [11] основана на оптимізованому хребті (backbone) DenseNet121 із функцією втрат. Ця модель застосовує блок Squeeze-and-Excitation (SE), щоб автоматично визначити важливість кожної функції каналу та присвоїти їй відповідну вагу. Ознаки (features) переносять у глибокий шар, регулюючи відповідні ваги, що знижує передавання зайвої інформації під час повторного використання функції в DenseNet121. Запропонована модель використовує додаткові переваги ознак середніх особливостей ЗНМ для підвищення здатності ознак функцій [11].

Водночас алгоритм виявлення малих об'єктів у реальному часі (RSOD) [12] підвищує точність виявлення невеликих об'єктів за допомогою

- використання карт ознак меншого шару, що містить дрібнозернистішу інформацію для прогнозування місця розташування;
- злиття локальних і глобальних особливостей неглибоких і глибоких ознак функцій в Feature Pyramid Network (FPN) для підвищення здатності до вилучення репрезентативніших функцій;
- присвоєння ваг вихідним особливостям FPN та їх адаптивне об'єднання;
- покращення шару збудження в механізмі уваги Squeeze-and-Excitation (SE), щоб точніше налаштувати реакції функцій кожного каналу [12].

Такі підходи можна вважати хорошим початком для реалізації власних методів і засобів алгоритму відстеження. Однак такі системи застосовували тільки для завдань автомобільного руху, де об'єкти зазвичай рухаються за строгими схемами. Тому додатково можна розглянути системи відстеження руху мурах [13]. У публікації [13] автори наводять свій фреймворк виявлення та відстеження руху мурах. Вони пропонують:

- прийняття двоступеневої структури виявлення об'єктів із використанням ResNet-50 (Residual Network with 50 Layers) як хребта (backbone) та кодування позиції областей, що становлять інтерес для точного знаходження мурах;
- використання моделі ResNet-50 для розроблення дескрипторів траєкторії мурах;
- побудова довгострокових послідовностей розпізнавання та об'єднання їх із інформацією про рух для досягнення онлайн-відстеження.

Розглянемо також таку нейронну мережу, як MDNet (Multi-Domain Convolutional Network Tracker). У задачах ВО є деякі бажані властивості для навчання цільового представлення, такі як: інваріантність щодо освітлення, масштабу, перспективи та розмиття руху. Мета використання мультидоменного навчання – вивчити дискримінаційну модель, яка вивчає спільне представлення цілі в різних областях. Щоб досягти цього, MDNet навчається в автономному режимі з великим набором відеопослідовностей, кожна послідовність розглядається як домен. Для точної локалізації об'єкта під час відстеження використовується техніка регресії обмежувальної рамки на обмежувальних прямокутниках із високими показниками.

Для алгоритмічного та ефективного пошуку на базі мінімізаційного фільтра IOU запропонований алгоритм V-IOU [14].

Він полягає у динамічному перекриванні локалізованих регіонів двох об'єктів за допомогою IOU.

Водночас у статті [15] запропонована онлайн-структура тестів Multiple Object Tracking (MOT) для відстеження руху мурах. Цей фреймворк поєднує в собі як розпізнавання, так і відстеження руху таких малих об'єктів. Методи та засоби цього фреймворку дають змогу ефективно запобігати перериванню фрагментів траєкторії, та ID (True Positive Id's), що є частою проблемою, оскільки мурахи можуть змінювати траєкторію руху [13].

ЗНМ на базі фільтрів кореляції – полягає у навчанні моделі кореляційного фільтра, для відокремлення об'єкта від заднього фону для кожного кадру відеопотоку в реальному часі. Параметри такого фільтра підбирають так, щоб результуюча згортка виглядала як розподіл Гауса з

піком у центрі об'єкта відстеження. Для навчання використовують вибірку з мінімізованою помилкою між розподілом Гаусса та вихідним результатом за допомогою методу найменших квадратів [16].

Кореляційні фільтри – це клас класифікаторів, які оптимізовані для створення різких граничних значень кореляційного виходу, здебільшого для досягнення точної локалізації цілей у сценах.

Простежимо історичний розвиток застосування кореляційних фільтрів.

Використовуючи сірі зображення, фільтр мінімальної вихідної суми квадратичних помилок (MOSSE) вперше застосовує кореляційний фільтр у галузі відстеження. Цей фільтр легко обчислити, він може швидко відстежувати об'єкти, але не гарантує точного відстеження змін зовнішнього вигляду об'єкта. Після цього, у 2012 р., Henriques et al. запропонував *circulant structure tracking with kernels (CSK)*. Згодом, у 2014 р., Danwelljan et al. за допомогою *Kernels correlation filter (KCF)* додатково налаштувати характеристики каналу для багатоканальних функцій і ввів функції CN для відстеження. Функція CN покращує дискримінаційну здатність фільтра. Однак адаптивність фільтра до обертання, поза межами поля зору та швидкого руху, все ще потребує вдосконалення. Наступним етапом була робота Danelljan et al., яка пропонує *discriminative scale space tracker (DSST)*, використовуючи піраміду ознак для вирішення проблеми багатомасштабної варіації, а також запропонував вдосконалений алгоритм DSST. Зі швидким розвитком методів ГН алгоритм C-COT дає змогу ефективно подати інформацію про просторову позицію із поверхневими функціями ЗНМ, що є комбінацією кореляційної фільтрації та ЗНМ. Алгоритм переміг у конкурсі VOT2016. Подібно до C-COT, алгоритм CSR-DCF також застосовує функції ЗНМ до алгоритму кореляційної фільтрації, що підвищує стійкість алгоритмів [2].

Сіамські (подвійні) ЗНМ – це різновид ШНМ, що складається із двох ідентичних нейронних підмереж з однаковими наборами ваг. Цей вид мереж дає змогу порівняти вектор ознак двох об'єктів, щоб виділити їх семантичну подібність або відмінність. Сіамська нейронна мережа є нелінійним відображенням даних з метою наблизити один до одного схожі об'єкти і рознести різні об'єкти на максимально можливу відстань. Сіамські мережі отримали свою назву від сіамських близнюків, що фізично зростаються один з одним, через використання відразу двох підмереж, які розділяють один набір ваг.

Прикладом таких ЗНМ є Deep Sort [17]. Цей підхід використовують для завдань відстеження людей (*Human Tracking*). Принцип роботи ґрунтується на відомому фільтрі Калмана та відстані Махаланобіса.

Відстань Махаланобіса – така міра відстані між векторами випадкових величин, що узагальнює значення евклідової відстані. Евклідова відстань – відстань між двома точками в евклідовому просторі. Застосовується у випадках, коли евклідова відстань для середнього значення не може забезпечити правильного розподілу для нових вимірів, за сильної кореляції. Визначається формулою:

$$d(X, Y, S) = \sqrt{(X - Y)^T \times S^{-1} \times (X - Y)}, \quad (1)$$

де X, Y – вектори, де різниця визначається між новою точкою (X) та середнім значенням для кожної змінної (Y); S – коваріаційна матриця; T – операція транспонування.

Коваріаційна матриця – це квадратна матриця, яка визначає коваріацію між кожною парою елементів заданого випадкового вектора.

Задача відстані Махаланобіса полягає у: забиранні коваріацій змінних, прийнятті дисперсії змінних такою, що дорівнює 1, та використанні евклідової відстані для трансформації даних. Отже,

що вища кореляція між змінними, то швидше можна зменшити відстань, оскільки відбуватиметься множення, обернене до найбільшого значення [2].

Отже, алгоритм DeepSort використовує поняття відстані Махаланобіса та фільтра Калмана для того, щоб переносити інформацію від одного кадру до наступного. На першій фазі, за допомогою методів РО, визначаються регіони та класи об'єкта розпізнавання. На наступній фазі застосовують такі алгоритми відстеження, як угорський алгоритм, для того щоб зв'язати певні об'єкти з об'єктами, які раніше відстежено за допомогою фільтрів Калмана.

Також у Deep Sort, порівняно з попередніми рішеннями, додається нова метрика – appearance, побудована на основі сіамської ЗНМ. Вона полягає у тренуванні окремої ШНМ на основі картинок скупчень людей. Така ШНМ вирішує проблему, коли один об'єкт (людина) може закривати на декілька ітерацій кадрів інший об'єкт (іншу людину), при цьому відбувається заміщення ідентифікатора. Така ШНМ дає змогу “поглибити” пам'ять об'єктів алгоритму [2].

Недоліком алгоритму Deep Sort є використання лише для задач Human Tracking. Для відстеження іншого класу об'єктів необхідно перенавчати RE-ID ШНМ, що не завжди можливо реалізувати на практиці.

Навчання з підкріпленням для завдань ВО – підхід навчання з підкріпленням полягає у навчанні знаходити оптимальний шлях у середовищі, ґрунтуючись на отриманій нагороді. Агент, як правило, отримує спостереження за дискретними часовими кроками з винагородами і вибирає дію з набору доступних опцій [18].

Є різні алгоритми для навчання системи з неперервним простором дій у завданнях пошуку та ВО [2]:

- Deep Q-Network (DQN) – алгоритм, який комбінує Q-навчання із глибокими нейронними мережами, що дає змогу вирішувати високомасштабовані завдання в ігровій та роботехнічній індустрії.

- Deep Deterministic Policy Gradient (DDPG) – модель, розроблена OpenAi, із використанням моделі “Actor–Critic”. Цей підхід тісно пов'язаний із DQN: виконуючи команди, Actor (актор) намагається навчитися рухатись у певному напрямку, залежно від заданих умов Critic(критика) [18].

- Proximal Policy Optimization (PPO) – суть алгоритму полягає у наближеній оптимізації стратегії, тобто коли відбувається пошук поля стратегії (field policy), без призначень значень до пар “стан – дія”.

- Target Candidate Track (TCT) – полягає у безперервному слідкуванні за об'єктами, із прив'язуванням попереднього та поточного положення об'єкта у суміжних кадрах. Оснований на фільтрі Калмана. Як алгоритм збіжності між двома розпізнаними об'єктами використовується угорський алгоритм [18].

Припускаючи результати авторів, тестову структуру MOT [19] можна використати для перевірки продуктивності алгоритмів відстеження.

Ефективність алгоритмів відстеження довільного класу рухомих об'єктів у реальному часі

Основні метрики для виконання дослідження: тип та методика алгоритму, **F1** (F-міра). Використано попередньо визначені метрики із еталонної вибірки **MOT17**: **MOTA** (багатоцільова точність), **MOTP** (багатоцільова влучність), **MT** (кількість відстежуваних траєкторій), **ML** (кількість втрачених траєкторій), **ID** (кількість перемикачів ідентифікатора), **FM** (кількість фрагментацій шляху відстеження).

Для оцінювання ефективності розроблених методів та засобів відстеження попередньо імплементовано три алгоритми пошуку та відстеження об'єктів [2] та один сторонній алгоритм DeepSort для порівняння характеристик метрик.

У таблиці визначено основні типи пропонувані для аналізу алгоритмів та метрики, на основі еталонної вибірки **MOT17** [19].

Визначення ефективності алгоритмів відстеження за метриками еталонної вибірки MOT17

	Метрика / Алгоритм	Тип	MOTP ↑	MOTA ↑	F1, % ↑	MT, % ↑	ML, % ↓	ID ↓	FM ↓
A1	Алгоритмічний пошук та наведення [20]	Відстеження шляхом пошуку	44,2	34,2	68,23	10,34	54,2	2852	3256
A2	Автоматизований пошук та наведення [18]	Навчання із підкріпленням	74,4	32,8	72,1	14,1	32,1	610	1318
A3	Алгоритмічний пошук за перетину регіонів [8]	Відстеження із пошуком	82,9	44,3	82,4	36,2	27,2	2374	3836
A4	DeepSort [17]	Сіамські ЗНМ	61,4	79,1	73,82	32,8	18,2	781	2008

Примітка. Стрілочка вгору вказує, що найліпші результати є найбільшими, стрілочка вниз – що найліпші результати є найменшими.

У дослідженні порівнюють три типи алгоритмів: відстеження за допомогою пошуку, навчання з підкріпленням та сіамські ЗНМ [2].

Згідно із результатами, найвищий показник **F1** оцінки влучності позиціонування розпізнаних об'єктів (**MOTP**) та **MT** має алгоритм **A3**. Причому як алгоритм збіжності для цього методу вибрано угорський алгоритм, що істотно збільшив показники ефективності порівняно з відомим рішенням з алгоритмом *k-d tree* (наприклад, значно зменшилась кількість перемикачів **ID** та збільшився показник **MT**, що продемонстровано у попередніх дослідженнях). Загалом метрики **MOTP** та **MT** для алгоритму **A3** показали на 12 % ліпші результати порівняно з другим за ефективністю алгоритмом **A4**. Ліпші результати на 10 % у метрики **F1**. Недоліком алгоритму **A3** є все ще високий рівень перемикачів **ID** та втрат траєкторії **FM** порівняно з прямими аналогами [2].

На рис. 3 зображено візуальне порівняння основних характеристик, отриманих у дослідженні.

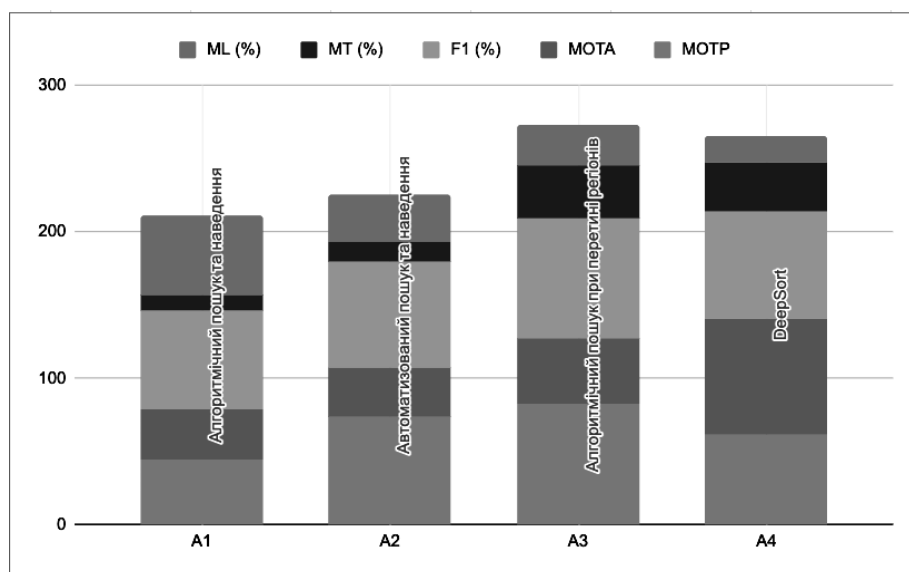


Рис. 3. Метрики ML, MT, F1, MOTA, MOTP ефективності відстеження об'єктів для алгоритмів A1-A4

Алгоритм DeepSort (A4) виявився ефективним для генерації метрики MOTA та ML, причому значення перемикань ID було невисоким.

Алгоритм A2 із використанням навчання з підкріпленням показав середні результати для більшості метрик, проте мінімальні значення перемикань ID та кількості втрачених траєкторій за принаймні 20 % тривалості відстеження (ML). З цього можна зробити висновок, що алгоритм A2 цілком підходить для виконання завдань у реальному часі, якщо кількість вхідних об'єктів велика (рис. 4).

Значення ID та FM істотно залежать від вхідного значення мінімізаційного фільтра IOU, тобто наскільки максимально часто можуть перетинатися ймовірні об'єкти. Що більше це значення, то меншим буде значення ID та FM, проте й менша кількість об'єктів пройде фільтрацію [2].

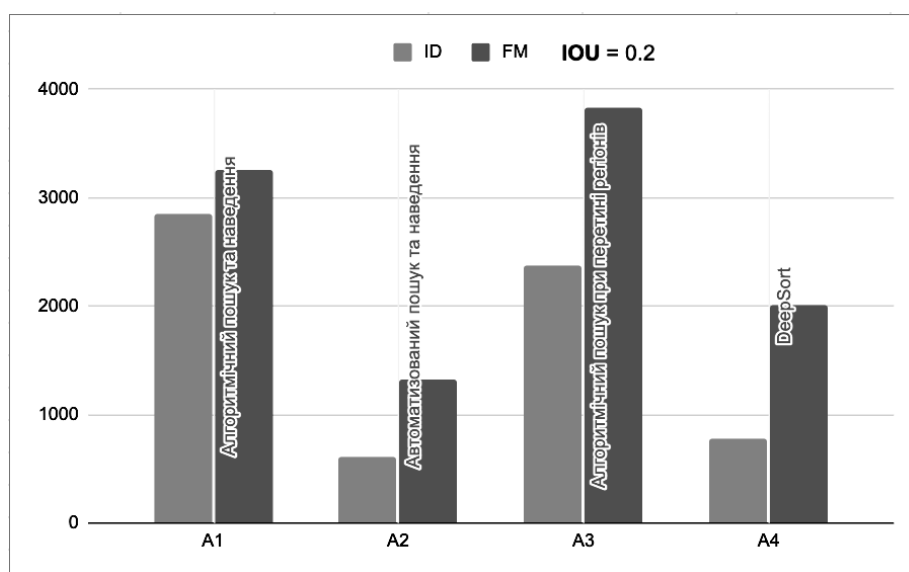


Рис. 4. Метрики ID, FM ефективності відстеження об'єктів для алгоритмів A1-A4. Що менше перемикань ідентифікаторів об'єктів ID та відхилень траєкторії FM, то ефективніший алгоритм відстеження під час опрацювання багатьох рухомих об'єктів

Найгірші результати показав алгоритм A1 через жадібний (greedy) підхід до перебору об'єктів для виявлення правильної траєкторії. А значення F1 для цього підходу ще у прийнятних межах – становить 68,23 %.

Висновки

Розглянуто основні підходи до інтеграції систем відстеження об'єктів на мобільні платформи. Проаналізовано основні види мобільних платформ: мобільні операційні пристрої та вбудовані системи, які можуть бути базою для створення кіберфізичної системи відстеження об'єктів у реальному часі.

Описано основні підходи до відстеження об'єктів у реальному часі на відеозображеннях. Визначено чотири типи алгоритмів відстежування із використання штучних нейронних мереж: відстеження за допомогою пошуку, кореляційні фільтри, сіамські нейронні мережі та навчання з підкріпленням. Описано імплементації таких алгоритмів, визначення їх переваги та недоліки. Визначено метрику оцінки результатів відстеження (MOT).

За результатами тестів аналізу чотирьох вибраних методів відстеження об'єктів визначено, що алгоритм пошуку в разі перетину регіонів V-IOU (A3) дав найвищі результати за метриками F1, MOTP та MT, з використанням угорського алгоритму як алгоритму збіжності. Загалом метрики MOTP та MT для алгоритму A3 показали на 12 % ліпші результати порівняно з другим за

ефективністю алгоритмом DeepSort. Ліпші результати на 10 % показала метрика F1. Недоліком алгоритму A3 є все ще високий рівень перемикань ID та втрат траєкторії FM порівняно з прямими аналогами. Водночас алгоритм навчання з підкріпленням A2 продемонстрував найменшу кількість перемикань ID, що також дає змогу використовувати його у реальному часі для відстеження багатьох об'єктів.

Список літератури

1. Hambling, D. (2023). Drones killing without oversight? *New Scientist*, 260(3461), 1–8. [https://doi.org/10.1016/S0262-4079\(23\)01937-1](https://doi.org/10.1016/S0262-4079(23)01937-1).
2. Kushnir, D. (2023). *Methods and means of searching and recognizing objects in video images on the mobile platform in real-time [Doctoral dissertation, Lviv Polytechnic National University]. Institutional repository of Lviv Polytechnic National University.* <https://lpnu.ua/sites/default/files/2023/radaphd/23565/diskushnir.pdf>
3. Niu, W., Ma, X., Lin, S., Wang, S., Qian, X., Lin, X., ... & Ren, B. (2020). Patdnn: Achieving real-time dnn execution on mobile devices with pattern-based weight pruning. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 907–922. <https://doi.org/10.1145/3373376.3378534>.
4. Thakkar, M. (2019). *Beginning machine learning in iOS: CoreML Framework.* Apress. <https://doi.org/10.1007/978-1-4842-4297-1>.
5. Dewantoro, G., Mansuri, J., & Setiaji, F. D. (2021). Comparative Study of Computer Vision Based Line Followers Using Raspberry Pi and Jetson Nano. *Jurnal Rekayasa ElektriKa*, 17(4). <https://doi.org/10.17529/jre.v17i4.21324>.
6. Puchtler, P., & Peinl, R. (2020, September). Evaluation of deep learning accelerators for object detection at the edge. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, Cham. 320–326. https://doi.org/10.1007/978-3-030-58285-2_29.
7. Parmar, M. (2016). A survey of video object tracking methods. *IJEDR*. 4(1). 519–524. <https://www.ijedr.org/papers/IJEDR1601086.pdf>.
8. Kushnir, D. (2022) *Methods and means for small dynamic objects recognition and tracking.* *Computers, Materials & Continua*, 73(2), 3649–3655. <https://doi.org/10.32604/cmc.2022.030016>.
9. Patel, S. K., & Mishra, A. (2013). Moving object tracking techniques: A critical review. *Indian Journal of Computer Science and Engineering*, 4(2), 95–102. <https://dx.doi.org/10.2139/ssrn.3548453>.
10. Taufique, A. M. N., Minnehan, B., & Savakis, A. (2020). Benchmarking deep trackers on aerial videos. *Sensors*, 20(2), 547. <https://doi.org/10.3390/s20020547>.
11. Zhang, X., Chen, X., Sun, W., & He, X. (2021). Vehicle Re-Identification Model Based on Optimized DenseNet121 with Joint Loss. *Cmc-computers materials & continua*, 67(3), 3933–3948. <https://doi.org/10.32604/cmc.2021.016560>.
12. Sun, W., Dai, L., Zhang, X., Chang, P., & He, X. (2022). RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Applied Intelligence*, 52(8), 8448–8463. <https://doi.org/10.1007/s10489-021-02893-3>.
13. Wu, M., Cao, X., & Guo, S. (2020). Accurate detection and tracking of ants in indoor and outdoor environments. *bioRxiv*. <https://doi.org/10.1101/2020.11.30.403816>.
14. Bochinski, E., Senst, T., & Sikora, T. (2018, November). Extending IOU based multi-object tracking by visual information. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) IEEE*, 1–6. <https://doi.org/10.1109/AVSS.2018.8639144>.
15. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*. <https://doi.org/10.48550/arXiv.1603.00831>.
16. Panteleev, O., & Oliynik, V. (2018). Method of visual multittracking in real time based on correlation filters. *Adaptive systems of automatic control*, 1(32), 97–106. <https://doi.org/10.20535/1560-8956.32.2018.145620>.
17. Wojke, N., & Bewley, A. (2018, March). Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 748–756. <https://doi.org/10.1109/WACV.2018.00087>.
18. Kushnir, D., & Paramud, Y. (2020). The intelligence algorithm of Cyber-Physical system targeting on a movable object using the smart sensor unit. *Computer Systems And Networks*, 2(1), 44–52. <https://doi.org/10.23939/csn2020.01.044>.

19. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*. <https://doi.org/10.48550/arXiv.1603.00831>.

20. Kushnir, D., & Paramud, Y. (2020). The algorithm of Cyber-Physical system targeting on a movable object using the smart sensor unit. *Advances In Cyber-Physical Systems*, 5(1), 16–22. <https://doi.org/10.23939/acps2020.01.016>.

METHODS OF TRACKING AN ARBITRARY NUMBER OF OBJECTS IN REAL-TIME ON A MOBILE PLATFORM

D. Kushnir

Lviv Polytechnic National University,
Computer Engineering Department

© Kushnir D., 2023

The problem of choosing methods for tracking recognized objects in real-time for systems with limited hardware capabilities is considered. It was determined that for such scenarios, it is appropriate to integrate tracking methods into the device, bypassing data transmission via the Internet.

Existing methods of tracking an arbitrary number of objects in real-time are considered. Among the performance evaluation metrics, the following were used: MOTA, MOTP, F1, MT, ML, ID and FM.

Based on the primary analysis of the effectiveness of such methods according to the metrics mentioned above, it was proposed to use the V-IOU tracking method to track recognized objects on a mobile platform in a mobile cyber-physical system.

Key words: V-IOU; Arbitrary object class tracking; Mobile platform; Real-time; MOTA; MOTP; Cyber-physical system.