



УДК 004.932.2

DOI: <https://doi.org/10.17721/AIT.2021.1.08>

М. В. Мороз, orcid.org/0000-0001-6953-683X,
Д. С. Берестов, orcid.org/0000-0002-3918-2978,
О. А. Курченко, orcid.org/0000-0002-3507-2392,

Київський національний університет імені Тараса Шевченка, Київ, Україна

Аналіз алгоритмів візуального відстежування об'єктів для систем реального часу

Проведено аналіз останніх досягнень і рішень у процесі візуального супроводу цільового об'єкта в галузі комп'ютерного зору, розглянуто підходи до вибору алгоритму візуального супроводження об'єктів на відеопослідовностях, виокремлено основні візуальні особливості, на яких може ґрунтуватись відстежування цільового об'єкта. Визначено критерії, які впливають на вибір алгоритму відстежування цільового об'єкта у режимі реального часу. Проте для відстежування у режимі реального часу за обмежених обчислювальних ресурсів питання вибору відповідного алгоритму є вирішальним. На вибір алгоритму візуального відстежування також впливають вимоги й обмеження щодо супроводжуваних об'єктів і попередні знання чи припущення щодо них. У результаті аналізу надано перевагу алгоритму відстежування Staple, за критерієм швидкодії, який є вирішальним показником у проектуванні та розробленні комплексу програмного та технічного забезпечення для автоматизованого візуального супроводу об'єкта у відеопотоці в режимі реального часу для різноманітних систем спостереження та безпеки, моніторингу трафіка дорожнього руху, розпізнавання діяльності й інших вбудованих систем.

Ключові слова: відеопоток, відстежувач, алгоритм, об'єкт, вбудована система, система реального часу.

Для цитування (for citation): М. В. Мороз, Д. С. Берестов, О. А. Курченко. "Аналіз алгоритмів візуального відстежування об'єктів для систем реального часу," *Сучасні інформаційні технології*, vol. 1, p. 59–65, 2021.

ВСТУП

Супроводження (відстеження, трекінг) об'єктів нині є досить важливим завданням у галузі комп'ютерного зору. Відстеження об'єктів має багато практичних застосувань [1–4], серед яких відеоспостереження, взаємодія між людиною та комп'ютером, навігація роботів, різноманітні системи спостереження та безпеки, моніторинг трафіка дорожнього руху, розпізнавання діяльності тощо.

Відстеження об'єктів націлено на те, щоб надати комп'ютеру можливість отримувати кращу модель реального світу. Оскільки комп'ютери за допомогою додаткових пристроїв не можуть бачити навколишнє середовище так само, як це робить людина, то відстеження об'єктів сприяє наближенню комп'ютерного сприйняття дійсності до людського, тим самим покращуючи взаємодію між людиною і комп'ютером.

ПОСТАНОВКА ЗАДАЧІ

Метою статті є огляд та аналіз наявних алгоритмів візуального супроводження об'єктів

на відеопослідовностях у режимі реального часу для вбудованих систем [5].

Проведений аналіз останніх досягнень і рішень [6–7] показав що у процесі візуального супроводу цільового об'єкта виокремлюють чотири важливі компоненти:

- 1) ініціалізацію цілі;
- 2) модель подання цілі на основі її особливостей;
- 3) прогноз руху;
- 4) пошук та отримання положення цілі.

Ініціалізація цілі полягає у виборі позиції об'єкта або певної зони навколо нього за допомогою таких допоміжних візуальних елементів: обмежувальна прямокутна зона, еліпс, центроїд, скелет фігури, контур, силует та ін.

Модель подання цілі зазвичай складається з певних візуальних особливостей, які притаманні об'єкту. Чим унікальнішими будуть особливості цілі, тим ефективнішим і точнішим буде виявлення. Вибір особливостей візуального представлення цільового об'єкта є важливим завданням, оскільки від них залежатиме ефективність і якість його супроводження.

© Мороз М. В., Берестов Д. С., Курченко О. А., 2021

Під час прогнозування руху в наступних кадрах оцінюється найбільш імовірне нове розташування цілі. На цьому етапі, зазвичай, використовують модель руху об'єкта. Це корисно у випадках, коли завчасно відомо про можливий рух цілі. Модель руху об'єкта може бути строго детермінованою, тобто, знаючи початкову точку руху, можна точно спрогнозувати розташування шуканого об'єкта в наступних кадрах. Також модель руху може мати локальну пам'ять, наприклад, якщо в декількох попередніх кадрах об'єкт рухався в одному напрямку, то зі значною вірогідністю, можна буде припустити, що в поточному кадрі тенденція руху в такому ж напрямку збережеться. Проте у випадках, коли рух цілі не детермінований або може стрімко змінюватись, використання моделі руху може бути надлишковим або повністю зайвим.

Під час пошуку цілі відбувається локалізація об'єкта на кадрі. За наявності прогнозування руху етап пошуку може залежати від його результату. У деяких випадках така взаємодія сприяє значному підвищенню продуктивності. Під час пошуку цілі, зазвичай, використовують різноманітні спеціалізовані детектори цілей.

Також наголошено на важливості гарної моделі подання цільового об'єкта і виокремлено такі основні візуальні особливості, на яких може ґрунтуватись відстежування об'єкта:

- а) колір;
- б) контури об'єктів;
- в) оптичний потік;
- г) текстура.

Зокрема представлено варіанти деяких підходів до розпізнавання, серед яких:

- а) точкові розпізнавачі;
- б) видалення фону;
- в) сегментація;

г) контрольоване навчання або навчання під наглядом;

- д) метод тимчасової різниці.

Також слід зазначити, що особливості супроводжувачів підтверджують комплексність та складність цієї царини. Можуть бути різні підходи та поєднання, які будуть надавати різні переваги і розв'язувати різні проблеми. Оскільки відстежування об'єктів є дослідницькою діяльністю, то кращим рішенням буде обрати певний алгоритм із кола доступних, адже розроблені алгоритми є надійними та чітко обґрунтованими.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Алгоритми відстеження можна розділили на дві загальні категорії [8–9]: відстежувачі з використанням кореляційних фільтрів (ВКФ) та

відстежувачі без використання кореляційних фільтрів (ВБКФ).

Дискримінаційні фільтри кореляції активно використовують у різних галузях комп'ютерного зору, серед яких розпізнавання об'єктів, реєстрація зображень, перевірка обличчя та розпізнавання дій. У відстеженні об'єктів кореляційні фільтри (КФ) використовують для поліпшення надійності й ефективності.

Схеми відстеження на основі КФ виконують обчислення, послуговуючись частотними характеристиками для мінімізації обчислювальної вартості. Загальна архітектура цих алгоритмів застосовує підхід «відстеження за виявленням» і зображена на рис. 1. Фільтри кореляції ініціалізують спеціально вказаною й обмеженою зоною, у якій розташовується ціль, на початковому кадрі. Під час подальшого відстеження місце розташування цілі в новому кадрі оцінюється завдяки використанню попереднього положення цілі. Щоб ефективно представити зовнішній вигляд цілі, застосовують відповідні методи вилучення характеристик, які використовують для побудови мапи характеристик (мапи особливостей). Крайні межі зони згладжуються завдяки застосуванню косинусного фільтра. Згладжування відбувається через те, що основні характеристики цілі зосереджені ближче до центра, а межі можуть вносити додатковий шум або ж можуть містити об'єкти, які не належать цілі.

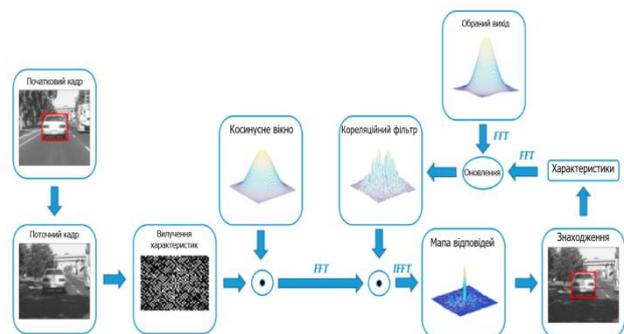


Рис. 1. Загальна схема алгоритмів відстеження на основі КФ

Операція кореляції виконується замість операції згортки, оскільки операція кореляції є швидшою. Мапа відповідей (результат пошуку в частотній області) обчислюється через поелементне множення між адаптивним фільтром навчання та характеристиками, вилученими за допомогою дискретної трансформції Фур'є (DFT). DFT працює із частотними даними за допомогою швидкого перетворення Фур'є (FFT). Мапу довіри (результат пошуку у просторовій області) отримують у просторовій області, застосовуючи



обернену FFT (IFFT) до мапи відповідей. Максимальне значення на мапі довіри вказує на нову позицію цілі. У результаті модель цілі оновлюється завдяки вилученню характеристик із нової позиції, після чого ці характеристики використовуються для оновлення кореляційного фільтра.

Схеми відстеження на основі КФ стикаються з різними труднощами, наприклад, із такими: можлива зміна із часом орієнтації та форми зовнішнього вигляду цілі. Іншим завданням є вибір ефективного представлення характеристик, оскільки деякі характеристики можуть покращувати ефективність ВКФ. Іншим важливим завданням для ВКФ є адаптація масштабу, оскільки розмір кореляційних фільтрів фіксований під час відстеження. Проте ціль із часом може змінювати свій розмір залежно від дальності до неї. Крім того, якщо ціль втрачено, її неможливо відновити знову.

Згідно із цією класифікацією всі відстежувачі, які не використовують кореляційні фільтри, називаються відстежувачами ВБКФ, які можна класифікувати на такі підкатегорії:

- відстежувачі з навчанням патчу;
- відстежувачі на основі розрідженості;
- відстежувачі на основі суперпикселів;
- відстежувачі на основі графів;
- відстежувачі на основі навчання з декількома екземплярами;
- відстежувачі на основі частин;
- відстежувачі на основі сіамських мереж.

Ці відстежувачі є дискримінаційними, за винятком відстежувачів на основі розрідженості, які є генеративними відстежувачами.

Відстежувачі з навчанням патчів використовують патчі цілі й патчі фону. Відстежувач навчається на позитивних і негативних зразках. Підготовлений відстежувач тестується на низці зразків, а максимальна відповідь відстежувача на одному з вхідних зразків вказує на положення цілі.

Під час *відстеження на основі навчання декількох екземплярів* замість розгляду певних патчів, як у попередньому методі, навчальні зразки розміщують у комірки й кожній комірці надається мітка. Позитивна мітка присвоюється комірці, якщо комірка містить принаймні один позитивний зразок, а негативна комірка містить лише негативні зразки. Позитивна комірка може містити позитивні та негативні екземпляри. Під час навчання мітки для екземплярів невідомі, але мітки на комірках відомі. За такого відстеження екземпляри використовують для побудови слабких класифікаторів, а кілька екземплярів вибирають та об'єднують для формування сильного класифікатора.

Відстежувач, що базується на сіамській мережі, здійснює відстеження на основі механізму відповідності. Процес навчання використовує загальні варіації зовнішнього вигляду цілі. Сіамські мережні відстежувачі узгоджують шаблони цілі зі зразками-кандидатами, щоб здобути схожість між патчами.

Відстежувачі на основі суперпикселів. Суперпикселі являють собою групу пикселів, що мають однакові значення. Цільова зона сегментується на декілька суперпикселів, що потім класифікуються для подальшої дискримінації об'єкта.

Відстежувачі на основі графів. Граф має вершини (це можуть бути пикселі, суперпикселі або частини об'єкта) та ребра (відповідність серед вершин). Графи використовують для прогнозування міток незазначених вершин. Як правило, відстежувачі на основі графів використовують суперпикселі як вузли для представлення зовнішнього вигляду об'єкта, а ребра представляють внутрішню геометричну структуру. Інша стратегія – побудувати графи між частинами об'єктів у різних кадрах.

Відстежувачі на основі частин. Частинне моделювання активно використовують для оброблення частин, які можуть змінюватись. Локальні частини об'єкта застосовують для моделювання трекера.

Відстежувачі на основі розрідженості. Усі вивчені до цього часу алгоритми є дискримінаційними методами відстеження. З іншого боку, генеративні методи вивчають представлення цілі та пошук цілі в кожному кадрі з мінімальною помилкою відновлення. Розріджене представлення – хороший приклад для генеративних моделей. Мета – виявити оптимальне подання цілі, яке є досить розрідженим і мінімізує помилку відновлення.

Для вибору алгоритму відстеження необхідно сфокусуватися на особливостях цільових об'єктів, особливостях середовища, обчислювальних можливостях пристрою та на перевагах наявних алгоритмів відстеження. Для більш зваженого рішення необхідно виконати порівняння наявних алгоритмів.

Щорічно спільнота «VOT Challenge» [10], діяльність якої спрямована на створення вільнодоступного сховища наявних алгоритмів візуального відстеження, проводить конкурс, у результаті якого порівнює і оцінює подані роботи. Це оцінювання служить для визначення якості відстежувача на основі підготовлених та анотованих відеопослідовностей. Тестові випадки включають основні проблеми відстеження, такі як: зміни освітлення, зміни розмірів



об'єкта, перекриття об'єкта тощо. Для автоматичного оцінювання якості відстежувачів використовують спеціально розроблений фреймворк. Також важливим є те, що у відкритому доступі перебувають як тестові дані з очікуваними результатами, так і результати порівняння трекерів за різними показниками.

Серед змагань «VOT Challenge» є також і змагання, які орієнтуються на відстежувачі, що придатні працювати в режимі реального часу. Здатність працювати в режимі реального часу означає, що відстежувач має повідомляти про розташування цілі на кожному кадрі із частотою, що перевищує або дорівнює частоті кадрів у відеопослідовності.

Спосіб тестування відстежувачів реального часу є таким. Спочатку відбувається ініціалізація відстежувача вхідними даними, а саме – цільовою зоною, у першому кадрі послідовності за допомогою відповідного інструментарію. Після цього інструмент оцінювання очікує вихідні дані відстежувача про положення цілі. Якщо відстежувач не встигне відповісти про нове положення цілі за наданий йому інтервал часу, то як відповідь відстежувача на такому кадрі беруть останню відповідь відстежувача.

Змагання «VOT Challenge» використовують методологію, що базується на перезапусках. У випадках, коли відстежувач надає результат у вигляді зони, в якій визначено нове розташування об'єкта, і ця зона має нульове перекриття з еталонними даними, то відбувається збій, а відстежувач повторно ініціалізується через п'ять кадрів після збою. Також у «VOT» є власні основні міри для оцінювання відстежувачів.

Точність і робастність – це основні міри, що застосовують для дослідження ефективності відстежувача в експериментах на основі перезапусків. Точність – це середнє перекриття між результатами відстежувача й еталонними даними впродовж успішних періодів супроводження. Робастність вимірює, як часто під час супроводу відстежувач втрачає ціль, унаслідок чого виконує повторну ініціалізацію. Потенційні відхилення в оцінюванні точності, які пов'язані з перезапусками, зменшуються завдяки ігноруванню десяти кадрів після повторної ініціалізації, що є досить консервативним запасом.

Третьою основною мірою оцінювання є очікуване середнє перекриття (ЕАО), яке поєднує значення точності в кожному кадрі та кількість перезавантажень відстежувача. Ця міра відповідає за очікуване середнє перекриття відстежувача на подібних за візуальними властивостями до такого набору даних короткотермінових послідовностях.

ЕАО вимірює очікуване перекриття зон відстежувача без перезавантажень на короткостроковій послідовності.

Для вибору й порівняння алгоритмів можна скористатися даними [11–14] щодо переможців у категорії реального часу за 2017–2020 рр. Порівняння алгоритмів наведено у таблиці, де є дані щодо оцінювання кожного алгоритму за значеннями середнього очікуваного перекриття, точності, робастності та використання алгоритмом обчислювальних ресурсів відеокарти (GPU) у процесі роботи. Також варто зазначити про трактування значень кожної з мір оцінювання. Кращими є більші значення очікуваного середнього перекриття, більші значення точності та менші значення робастності.

ТАБЛИЦЯ
Порівняння найкращих алгоритмів

Алгоритм	ЕАО	Точність	Робастність	GPU
AlphaRef	0,486	0,754	0,788	Так
OceanPlus	0,471	0,679	0,824	Так
AFOD	0,458	0,708	0,780	Так
SiamMargin	0,366	0,577	0,321	Так
SiamFCOT	0,350	0,601	0,386	Так
DiMP	0,321	0,582	0,371	Так
SiamRPN	0,383	0,586	0,276	Так
SA_Siam_R	0,337	0,566	0,258	Так
SA_Siam_P	0,286	0,533	0,337	Так
CSRDCF++	0,212	0,459	0,398	Так
SiamFC	0,182	0,502	0,604	Так
ECOhc	0,177	0,494	0,571	Ні
Staple	0,170	0,530	0,688	Ні

З наведеної таблиці видно, що збільшується тенденція до використання GPU та нейронних мереж у алгоритмах відстеження. Проте не всі вбудовані системи мають у своїй апаратній архітектурі GPU, а якщо й мають, то наявний GPU не завжди доцільний для виконання важких завдань. Наприклад, Raspberry Pi [15] хоч і має GPU, проте він не є достатньо потужним для виконання завдання відстежування і використовується, зазвичай, для завдань мультимедіа. Якщо ж реалізувати ці алгоритми без використання GPU, то їхня швидкість буде значно менша. Тому одним із критеріїв, який вплине на вибір алгоритму, є направленість алгоритму на виконання на CPU.

Для подальшого розгляду можна одразу відкинути алгоритми, які використовують GPU. Відповідно, залишилися лише алгоритми Staple та



ЕСОнс. Алгоритм ЕСОнс є кращим за показником робастності, а також має трохи кращий показник ЕАО. З іншого боку, Staple має кращий показник точності. Кожен із них запускався на процесорі Intel i7. ЕСОнс зумів досягти 60 кадрів у секунду, у той час як Staple – 80 кадрів у секунду, причому автори відстежувача Staple додатково вказали на модифікації, які збільшують швидкість відстежування до 100 кадрів у секунду. Саме тому перевага у виборі алгоритму відстежування буде надана алгоритму Staple, оскільки швидкодія є вирішальним показником у цьому випадку.

Алгоритм відстеження Staple у процесі роботи спирається на комбінацію фільтра кореляції (з використанням гистограми спрямованих градієнтів як характеристики) та глобальної кольорової гистограми. Він працює згідно з парадигмою відстежування на основі розпізнавання. Проблему пошуку об'єкта на зображенні x_t у порядку кадру t можна звести до задачі вибору такої прямокутної зони p_t з набору S_t , яка збільшуватиме оцінку у формулі

$$p_t = \arg \max_{p \in S_t} f(T(x_t, p), \theta), \quad (1)$$

де функція T – таке перетворення зображення, що $f(T(x, p), \theta)$ присвоює оцінку прямокутній зоні p на зображенні x відповідно до параметрів моделі θ ; θ – параметри моделі, які мають обиратися так, щоби мінімізувати значення функції витрат, яка залежить від попередніх зображень і положень об'єкта на цих зображеннях.

Для пошуку об'єкта на зображенні використовується оцінювальна функція. Оцінювальна функція є лінійною комбінацією шаблонної оцінки й оцінки гистограми і визначена як

$$f(x) = \gamma_{tmp} f_{tmp}(x) + \gamma_{hist} f_{hist}(x), \quad (2)$$

де γ_{tmp} , γ_{hist} – коефіцієнти, які визначають вплив шаблонної оцінки й оцінки гистограми на результуюче значення оцінки, причому

$$\gamma_{tmp} + \gamma_{hist} = 1.$$

Шаблонна оцінка є лінійною функцією характеристичного K -канального зображення φ_x , отриманого із зображення x та визначеного на скінченній сітці $\tau \subset Z^2$, обчислюється як

$$f_{tmp}(x; h) = \sum_{u \in \tau} h[u]^T \varphi_x[u], \quad (3)$$

де h – шаблон, інше характеристичне K -канальне зображення.

Оцінка гистограми є лінійною функцією середнього пікселя й обчислюється з характеристичного M -канального зображення ψ_x , отриманого із зображення x і визначеного на скінченній сітці $H \subset Z^2$:

$$f_{hist}(x; \beta) = \beta^T \left(\frac{1}{|H|} \sum_{u \in H} \psi_x[u] \right), \quad (4)$$

де β – ваговий вектор гистограми, шаблон.

Оскільки зовнішній вигляд об'єкта може суттєво змінюватися впродовж усієї відеопослідовності, то не ефективно будувати його модель лише на основі першого кадру та використовувати цю єдину модель, щоби знайти об'єкт у всіх інших кадрах. Тому цей алгоритм застосовує адаптацію моделі для того, аби скористатися інформацією, наявною в пізніших кадрах. Для цього він використовує прогнози (знайдене положення цілі) у нових кадрах як навчальні дані, завдяки яким можна оновити модель. Проте в цьому підході існує небезпека того, що невеликі помилки із часом накопичуються й у результаті можуть викликати дрейф моделі.

Отримати значення кореляційного фільтра (шаблону) для шаблонної оцінки можна за формулою

$$\hat{h}[u] = 1/(\hat{d}[u] + \lambda) \cdot \hat{r}[u], \quad (5)$$

де \wedge позначає операцію дискретного перетворення Фур'є; \hat{d} – знаменник шаблону, який визначено у формулі (6); \hat{r} – чисельник шаблону, який визначено у формулі (7); λ – коефіцієнт, що використовується для обмеження складності моделі.

$\hat{d}[u]$ є сумою елементів головної діагоналі матриці $\hat{s}[u]$ з розмірністю $K \times K$ з елементами $\hat{s}^{ij}[u] = (\hat{\phi}^j)^* \otimes \hat{\phi}^i$, де $\hat{\phi}$ є характеристичним K -канальним зображенням $\varphi_{T(x,p)}$, яке переведено до простору Фур'є. $\hat{d}[u]$ обчислюють за формулою

$$\hat{d}[u] = \sum_{i=1}^K (\hat{\phi}^i)^* \otimes \hat{\phi}^i, \quad (6)$$

де $*$ – комплексне спряжене число;

\otimes – операція поелементного множення.

$\hat{r}[u]$ є K -вимірним вектором з елементами $\hat{r}^i[u]$ й обчислюється за формулою

$$\hat{r}^i[u] = (\hat{y}^i)^* \otimes \hat{\phi}^i, \quad (7)$$

де y – бажаний вихід, зазвичай у формі функції Гаусса з піком в одиниці.

Формули (6) та (7) завдяки простору Фур'є представляють собою операції кроскореляції. Операція кореляції застосовується замість виснажливої щодо обчислювальних ресурсів операції згортки. Мапа відповідей обчислюється за допомогою поелементного множення між адаптивним кореляційним фільтром навчання й отриманими із зображення характеристиками за допомогою дискретної трансформації Фур'є



(DFT). DFT працює в частотній області за допомогою швидкого перетворення Фур'є (FFT). Мапа довіри у просторовій області отримується за допомогою оберненого FFT (IFFT) над мапою відповідей. Максимальні значення на мапі довіри вказують на нову цільову позицію об'єкта.

Навчання для шаблонної оцінки відбувається через адаптацію чисельника \hat{r}_t та знаменника \hat{d}_t до шаблону \hat{h} . Адаптація чисельника \hat{r}_t представлена формулою

$$\hat{r}_t = (1 - \eta_{mpl})\hat{r}_{t-1} + \eta_{mpl}\hat{r}_t', \quad (8)$$

де η_{mpl} – коефіцієнт адаптивності шаблонної моделі; \hat{r}_{t-1} – чисельник шаблону, отриманий із попередніх кадрів; \hat{r}_t' – чисельник шаблону, який обчислюється за формулою (7) для поточного кадру.

Адаптація знаменника \hat{d}_t представлена формулою

$$\hat{d}_t = (1 - \eta_{mpl})\hat{d}_{t-1} + \eta_{mpl}\hat{d}_t', \quad (9)$$

де \hat{d}_{t-1} – знаменник шаблону, отриманий із попередніх кадрів; \hat{d}_t' – знаменник шаблону, який обчислюється за формулою (6) для поточного кадру.

Отримати шаблон кольорової гістограми для характеристичного каналу зображення $j = 1 \dots M$ можна за формулою

$$\beta_j^j = \frac{p^j(O)}{p^j(O) + p^j(B) + \lambda}, \quad (10)$$

де $p^j(O)$ – частка пікселів у регіоні об'єкта, для якої характеристика j не дорівнює нулю; $p^j(B)$ – частка пікселів у регіоні фону об'єкта, для якої характеристика j не дорівнює нулю.

Навчання моделі на основі гістограми відбувається за допомогою адаптації моделей зони фону та зони об'єкта. Адаптація зони об'єкта представлена формулою

$$p_t(O) = (1 - \eta_{hist})p_{t-1}(O) + \eta_{hist}p_t'(O), \quad (11)$$

де η_{mpl} – коефіцієнт адаптивності гістограмної моделі; $p_t(O)$ – вектор з елементів $p_t^j(O)$ для $j = 1 \dots M$; $p_t'(O)$ – кольорова гістограма зони об'єкта, обчислена у кадрі t .

Адаптація зони фону представлена формулою

$$p_t(B) = (1 - \eta_{hist})p_{t-1}(B) + \eta_{hist}p_t'(B), \quad (12)$$

де $p_t(B)$ – вектор з елементів $p_t^j(B)$ для $j = 1 \dots M$; $p_t'(B)$ – кольорова гістограма зони фону біля об'єкта, обчислена у кадрі t .

ВИСНОВКИ

Візуальне відстежування об'єктів є актуальною проблемою в межах комп'ютерного зору. Завдяки її важливості нині існує багато підходів для розв'язання цієї проблеми. Проте для відстеження у режимі реального часу за обмежених апаратних ресурсів питання вибору відповідного алгоритму є вирішальним. На вибір відстежувача також впливають вимоги й обмеження щодо супроводжуваних об'єктів і попередні знання чи припущення щодо них. Для вибору відстежувача проведено огляд наявних алгоритмів, у результаті якого обрано алгоритм Staple та обґрунтовано доцільність його використання. Цей алгоритм має добру швидкодію, а також помірні вимоги щодо обчислювальних ресурсів, що є вирішальним для роботи на вбудованих системах. Подальші дослідження будуть направлені на проектування та розроблення комплексу програмного та технічного забезпечення для автоматизованого візуального супроводу об'єкта у відеопотоці в режимі реального часу для вбудованих систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Y. Zhang, Z. Chen and B. Wei, "A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement", in *IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1312–1316.
- [2] (2021) Оптико-електронна система керування стрільбою артустановок "Sens-2". [Online]. Available: <https://www.spetstechnoexport.com/product/sens-2>.
- [3] Byung-Kuk Seo, Jungsik Park and Jong-Il Park, "3-D visual tracking for mobile augmented reality applications", in *IEEE International Conference on Multimedia and Expo*, 2011, , pp. 1–4.
- [4] Mahendra Mallick; Vikram Krishnamurthy; Ba-Ngu Vo, "Tracking and Data Fusion for Ground Surveillance", in *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, IEEE, 2012, pp.203–254.
- [5] Peter Marwedel. (2021) Embedded System Design. [Online]. Available: <https://link.springer.com/content/pdf/10.1007%2F978-3-030-60910-8.pdf>.
- [6] Z. Soleimanitaleb, M. A. Keyvanrad and A. Jafari, "Object Tracking Methods: A Review", in 9th International Conference on Computer and Knowledge Engineering (ICCKE), 2019, pp. 282–288.
- [7] Sanna Ågren. (2021) Object tracking methods and their areas of application: A meta-analysis. A thorough review and summary of commonly used object tracking methods. [Online]. Available: <http://www8.cs.umu.se/education/examina/Rapporter/SannaAgrenFinal.pdf>.
- [8] Mustansar Fiaz, Arif Mahmood, Sajid Javed, and Soon Ki Jung. (2021) Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends. [Online]. Available: <https://arxiv.org/pdf/1812.07368.pdf>.
- [9] Yilmaz A., Javed O., and Shah M. (2021) Object tracking: A survey. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/1177352.1177355>.
- [10] (2021) The VOT website. [Online]. Available: <https://www.votchallenge.net>



- [11] (2017) The VOT website. [Online]. Available: <https://www.votchallenge.net/vot2017/results.html>
- [12] (2018) The VOT website. [Online]. Available: <https://www.votchallenge.net/vot2018/results.html>
- [13] (2019) The VOT website. [Online]. Available: <https://www.votchallenge.net/vot2019/results.html>
- [14] (2020) The VOT website. [Online]. Available: <https://www.votchallenge.net/vot2020/results.html>

- [15] L. Rosyidi, A. Prasetyo and M. S. Romadhon, "Object Tracking with Raspberry Pi using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM)", in *8th International Conference on Information and Communication Technology (ICoICT)*, 2020, pp. 1–6.

Стаття надійшла до редколегії

17.10.2021

Analysis of visual object tracking algorithms for real-time systems

The article analyzes the latest achievements and decisions in the process of visual support of the target object in the field of computer vision, considers approaches to the choice of algorithm for visual support of objects on video sequences, highlights the main visual features that can be based on tracking object. The criteria that influence the choice of the target object-tracking algorithm in real time are defined. However, for real-time tracking with limited computing resources, the choice of the appropriate algorithm is crucial. The choice of visual tracking algorithm is also influenced by the requirements and limitations for the monitored objects and prior knowledge or assumptions about them. As a result of the analysis, the Staple tracking algorithm was preferred, according to the criterion of speed, which is a crucial indicator in the design and development of software and hardware for automated visual support of the object in real-time video stream for various surveillance and security systems, monitoring traffic, activity recognition and other embedded systems.

Keywords: video stream, tracker, algorithm, object, embedded system, real-time system.



Микола Мороз. Студент четвертого курсу кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка, Київ, Україна. Наукові інтереси: Internet of Things, Embedded Systems.

Mykola Moroz. 4th-year student of the Department of Software Systems and Technologies, Faculty of Information Technology, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine. Research interests: Internet of Things, Embedded Systems.



Олег Курченко. Доцент кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка, Київ, Україна. Наукові інтереси: інженерія програмного забезпечення, технології віртуалізації, штучний інтелект.

Oleg Kurchenko. Associate Professor of the Department of Software Systems and Technologies, Faculty of Information Technology, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine. Research interests: Software Engineering, Virtualization Technologies, Artificial Intelligence.



Денис Берестов. Асистент кафедри програмних систем і технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка, Київ, Україна. Наукові інтереси: методи та інструменти Data Science, Industrial Internet of Things, Embedded Systems.

Denys Berestov. Assistant of the Department of Software Systems and Technologies, Faculty of Information Technology, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine. Research interests: methods and tools of Data Science, Industrial Internet of Things, Embedded Systems.