

<https://doi.org/10.31891/2219-9365-2024-78-22>

УДК 004

МРАК Василь

Національного університету “Львівська політехніка”

<https://orcid.org/0009-0002-6066-5592>

[vasyl.b.mrak@lpnu.ua](mailto:vasyl.b.mrak@lpnu.ua)

КЛИМАШ Михайло

Національного університету “Львівська політехніка”

<https://orcid.org/0000-0003-2867-1482>

[mykhailo.m.klymash@lpnu.ua](mailto:mykhailo.m.klymash@lpnu.ua)

БАБИНЕЦЬ Василь

Національного університету “Львівська політехніка”

<https://orcid.org/0009-0004-2059-0260>

[vasyl.v.babynets@lpnu.ua](mailto:vasyl.v.babynets@lpnu.ua)

## УДОСКОНАЛЕННЯ МЕТОДУ ВИЯВЛЕННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ У ВІДЕО ПОСЛІДОВНОСТЯХ

Віднімання фону відіграє важливу роль в системах відеоспостереження, оскільки це один із найбільш використовуваних інструментів виявлення руху. Якщо науковий прогрес дозволив розробити складне обладнання для цього завдання, використовувані алгоритми також повинні бути вдосконалені. За останнє десятиліття техніка віднімання фону під назвою ViBE набирає поширення. Однак оригінальний алгоритм має два головних недоліки. Перший — це явище привидів, яке з'являється, якщо початковий кадр містить рухомий об'єкт або у разі раптової зміни фонові ситуації. По-друге, він погано працює на складному тлі. У цій статті представлено ефективний підхід віднімання фону на основі ViBE для вирішення цих двох проблем. Він базується на адаптивному радіусі для роботи зі складним фоном, на сукупному середньому та механізмі підрахунку пікселів для швидкого усунення явища фантома та адаптації до раптової зміни моделі фону.

Ключові слова: віднімання фону, ViBE, адаптивний радіус, кумулятивне середнє, підрахунок пікселів.

MRAK Vasyl, KLYMASH Mykhailo, BABYNETS Vasyl

Lviv Polytechnic National University

## IMPROVING THE METHOD OF DETECTION OF DYNAMIC OBJECTS IN VIDEO SEQUENCES

Many computer vision programs must first find moving objects in a video frame. As an example of video surveillance, we need to detect an intruder in a monitored scene. To achieve this, the image can be averaged over time during the initialization period. This is done in order to separate moving objects from a static background. The main process used for this task is background subtraction. A wide variety of background subtraction techniques have been proposed over the years, which can be grouped into codebook-based, probabilistic-based, sample-based, subspace-based, compressed sensing-based, and more recently, deep learning-based approaches.

Background subtraction plays an important role in video surveillance systems, as it is one of the most widely used motion detection tools. If scientific progress has made it possible to develop sophisticated equipment for this task, the algorithms used must also be improved. Over the past decade, a background subtraction technique called ViBE has been gaining popularity. However, the original algorithm has two main drawbacks. The first is the ghosting phenomenon, which appears if the initial frame contains a moving object or if the background situation suddenly changes. Second, it doesn't work well on complex backgrounds. This paper presents an efficient ViBE-based background subtraction approach to address these two issues. It is based on an adaptive radius for working with complex backgrounds, a cumulative average and a pixel counting mechanism to quickly eliminate the phenomenon of phantom and adapt to the sudden change of the background model.

The paper presents an efficient background subtraction algorithm based on ViBE for a complex background. The proposed method was explained in detail with all the necessary parameters. It combines advanced ViBE, which uses cumulative averaging and a pixel-counting engine for fast de-ghosting, and adaptive ViBE, which calculates an adaptive radius based on background changes. The obtained results are given, which demonstrate the effectiveness of the method in comparison with the existing ones. With an average frame rate of 30 frames per second, it can also be used in real-time applications.

Keywords: background subtraction, ViBE, adaptive radius, cumulative average, pixel count.

## ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Багато програм комп'ютерного зору спочатку повинні знайти рухомі об'єкти у відеокадрі. Як приклад відеоспостереження, нам потрібно виявити зловмисника на спостережуваній сцені. Щоб досягти цього, можна усереднювати зображення протягом певного часу протягом періоду ініціалізації[1]. Це робиться з метою відокремлення рухомих об'єктів від статичного фону. Основним процесом, який використовується для цього завдання, є віднімання фону. Протягом багатьох років було запропоновано широкий спектр методів віднімання фону, які можна згрупувати в підхід на основі кодової книги, на основі

ймовірностей, на основі вибірки, на основі підпростору, на основі стисненого зондування та нещодавно підхід на основі глибокого навчання. У [2] автори пропонують хороший стан техніки різних методів віднімання фону від попередніх до нових.

### АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Основна ідея віднімання фону складатиметься лише з різниці між поточним значенням пікселя та значенням фонові моделі з наступним порівнянням цієї різниці з пороговим значенням для прийняття рішення. Однак у реальних життєвих ситуаціях у фоновому режимі виникає велика складність через велику кількість проблем, а саме зміни освітлення, динамічний фон, тремтіння камери та багато інших фонових проблем. Це призводить до розробки більш складних методів віднімання фону.

Методи імовірнісного віднімання фону, які також називають методами параметричного віднімання фону, зазвичай стосуються підходів, які моделюють фон за допомогою нормального (гаусового) розподілу значень інтенсивності пікселів зображення. Одне з перших рішень було запропоновано Stauffer C [6]. Після цього з'явилося багато вдосконалених методів STGMM[9], SKMGM[10], SKMGM[10], TAPPMOG[11] і STAPPMOG[12] (де MOG означає суміш Гауса), кожен із яких намагався вирішити один або більше недоліків оригінального алгоритму. Однак через його чутливість MOG не може бути точно налаштований, і його здатність успішно справлятися зі змінами у фоновому режимі є дискусійною.

Щоб вирішити недоліки ручного вибору параметрів у кожному середовищі, були запропоновані непараметричні підходи (Ahmed Elgamma [13]). Зазвичай вони стосуються методів на основі вибірки, але існує багато інших, таких як оцінка щільності ядра (KDE)[14], підхід Байєса[15] або рекурсивна оцінка щільності (RDE)[16].

Інша група методів — метод кодового словника, де для кожної позиції пікселя фон моделюється за допомогою кодового словника [17]. Тут кожному пікселю призначається кодове слово, яке складається з інтенсивності, кольору та часових характеристик. Щоб виконати сегментацію, інтенсивність і колір вхідних пікселів порівнюються з інтенсивністю та кольором кодових слів у кодовому словнику. Пізніше алгоритм обчислює відстані між пікселями та кодовими словами, порівнює їх із пороговим значенням і призначає мітку переднього плану, якщо відповідності не знайдено, або мітку фону в іншому випадку. Пізніше відповідне кодове слово оновлюється відповідним фоновим пікселем.

Підпросторовий підхід [18], в якому одна модель призначена для точного виявлення руху, а інша спрямована на досягнення представлення порожньої сцени. Відмінності у виявленні переднього плану додаткових моделей використовуються для визначення нових статичних областей. У цьому підході обчислюється набір зображень і відповідне середнє значення та коваріаційна матриця. Після цього виконується обчислення PCA коваріаційної матриці, проєкції з певною кількістю векторів і порівняння вхідних зображень з їх проєкціями на власні вектори, обчислюється відстань між зображенням і проєкцією та порівнюється з відповідним пороговим значенням, щоб позначити піксель як піксель переднього плану чи фону.

Алгоритми стиснення [19, 20, 21, 22], дозволяють зменшити кількість вимірювань, необхідних для представлення відео, використовуючи попередні знання про розрідженість вихідного сигналу. Сигнал може бути реконструйований з невеликого набору випадкових проєкцій, за умови, що сигнал є розрідженим у певному базисі. Однак ці методи накладають певні умови на матрицю проектування.

Також важливо згадати методи, які використовують виявлення країв [23] або виявлення патчів [24]. У той час як перший метод найчастіше використовується для сегментації зображення на основі різких змін інтенсивності зображення, другий використовує патчі для таких операцій, як зменшення шуму зображення.

Метод розроблений Брахамом і Дрогенбруком, полягає в тому, що використання згорткової нейронної мережі зазвичай відноситься до підходу глибокого навчання. Він використовує модель фіксованого фону, яка була згенерована на основі операції тимчасової медіани над  $N$  відеокадрами [25]. Його архітектура дуже схожа на мережу LeNet-5 для рукописної класифікації цифр [26], за винятком того, що субвибірка виконується з максимальним об'єднанням замість усереднення, а приховані сигмоїдні одиниці замінюються випрямленими лінійними одиницями для швидшого навчання. Однак цей метод, як правило, залежить від конкретної сцени.

### ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

**Метою даної роботи** – Розроблення ефективного методу віднімання фону на основі ViBE, який базується на адаптивному радіусі для роботи зі складним фоном, на сукупному середньому та механізмі підрахунку пікселів для швидкого усунення явища фантома та адаптації до раптової зміни моделі фону.

### ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Запропонований метод має три основні модифікації. Першою зміною є введення нового механізму оновлення, який використовує формулу Кеніга для скорочення загальних обчислень. Це поєднується з використанням  $n$  останніх кадрів для моделювання фону. Тут значення  $n$  вибрано емпірично. Друга

модифікація полягає у впровадженні механізму підрахунку пікселів для врахування пікселів переднього плану, які довго залишаються пікселями фону. Зазвичай це відбувається, при раптовій зміні фонові моделі. Ці дві модифікації призначені для швидкого усунення феномена привидів. Третя модифікація стосується впровадження модифікованої адаптивної політики радіуса з коригуванням параметрів, яка дозволяє алгоритму працювати зі складним фоном, коли це необхідно, і поводитися по-іншому (майже як оригінальний ViBE), коли фон є відносно простим.

Загальне функціонування оригінального ViBE можна розділити на три основні частини, а саме ініціалізацію фонові моделі, фонову сегментацію та оновлення фонові моделі. Ідея полягає в тому, що замість використання першого кадру для ініціалізації  $N$  значень вибірки кожного пікселя використати  $n$  перших кадрів. Це дає змогу враховувати зв'язок між  $n$  останніми кадрами  $i$ , таким чином, діє як фільтр. Крім того, з'являється механізм підрахунку пікселів і посилює процес, якщо піксель переднього плану залишається на передньому плані протягом  $K$  послідовних кадрів; він, безумовно, став фоновим пікселем, оскільки тепер він є статичним пікселем у сцені

Щоб побачити переваги використання  $n$  останніх кадрів для моделювання фону, потрібно проаналізувати рис. 1.1. На ньому показано різницю в результатах між використанням одного кадру та використанням  $n$  останніх кадрів у порівнянні з ідеальним результатом очікування. Можна побачити, що краще вибрати  $n$  останніх кадрів, щоб результати були більш точними. Недоліком цього підходу є те, що він потребує трохи більше обчислювальних ресурсів (особливо пам'яті та часу), ніж той, що використовує один кадр, але це прийнятно порівняно з отриманими результатами.

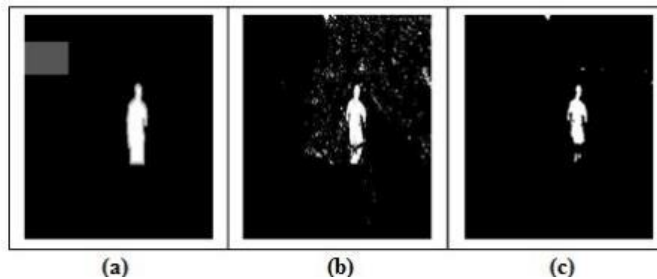


Рис. 1 Порівняння результатів набору даних (а) еталон (b) з використанням одного кадру (c) з використанням  $n$  останніх кадрів

З іншого боку, коли деякі зміни відбуваються на задньому плані, як у випадку руху гілок, повітря, дрібних часток або навіть невеликих хвиль на поверхні моря, алгоритм повинен впоратися з цією складністю. Це пов'язано з тим, що ці елементи в русі не належать до переднього плану, тому їх потрібно розглядати як елементи фону. Однак використання радіуса фіксації  $R$  може помилково класифікувати ці пікселі як пікселі переднього плану. Для цього використано адаптивний радіус ( $R_{ad}$ ), який має тенденцію бути сталим, якщо середня відстань між поточним пікселем і значеннями фону його  $N$  вибірки поступово збільшується для складного динамічного фону і має тенденцію до незначного зменшення, якщо середнє значення відстані має тенденцію бути стабільним.

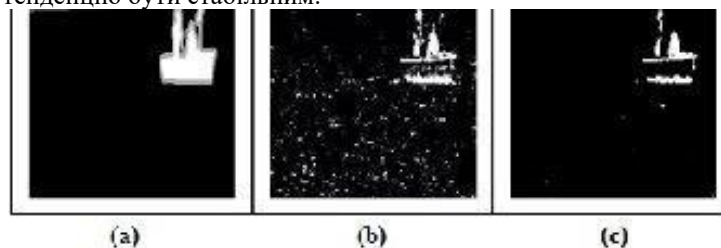


Рис. 2 Порівняння результатів на наборі даних човнів (а) еталон (b) використанням фіксованого радіуса (c) з використанням адаптивного радіуса

На малюнку 1.2 показано різницю в результатах між використанням фіксованого радіуса та адаптивного радіуса в порівнянні з ідеальним результатом очікування.

Все це дозволило формалізувати весь процес схематично, і він представлений на рис. 1.3

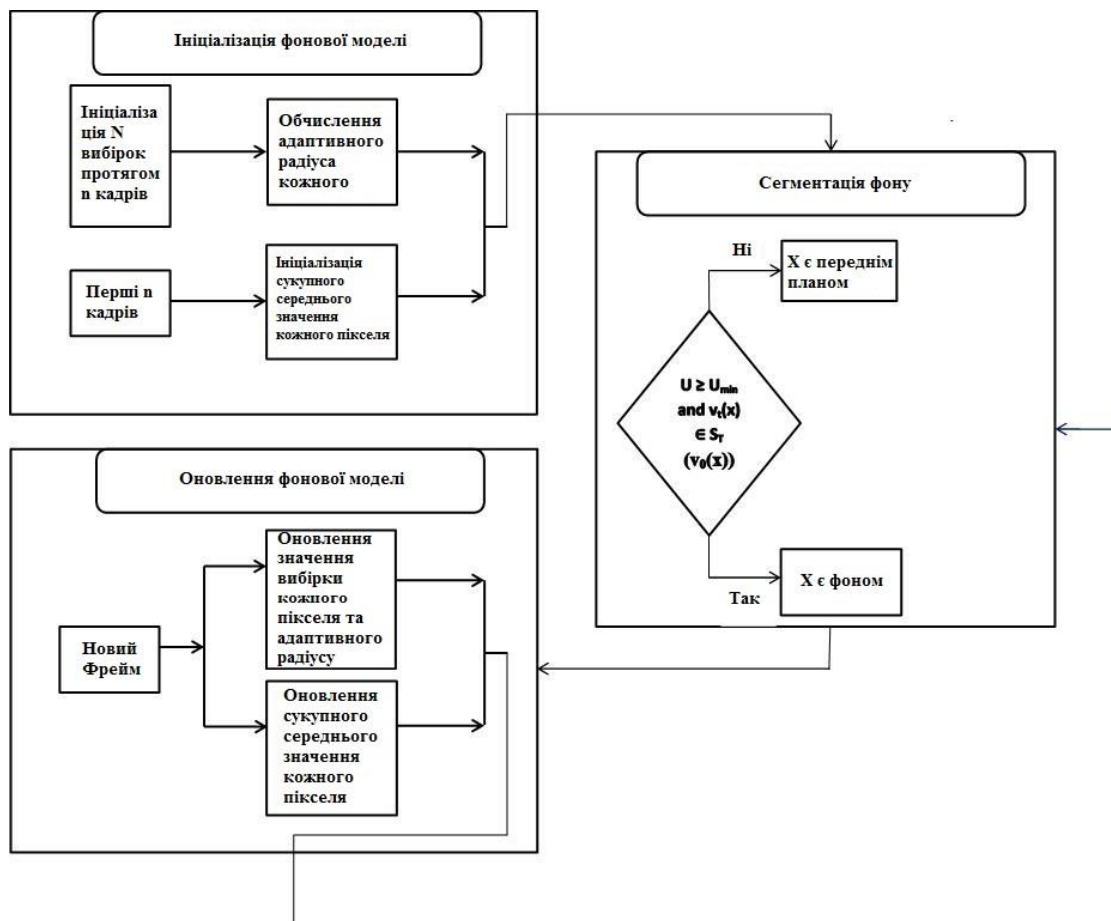


Рис. 3 Архітектура функціональної системи

Запропонований метод складається з трьох основних частин, а саме фоновій ініціалізації, фоновій сегментації та фоновому оновленню, як показано на рис. 1.3. Перша частина виконується під час процесу ініціалізації. У той час як перші  $n$  кадрів збираються для обчислення сукупного середнього для кожного пікселя, вибіркові значення також ініціалізуються. У той же час для кожного пікселя також обчислюється адаптивний радіус  $R_{ad}$ . Друга частина — це частина сегментації, яка класифікує піксель як піксель фону або переднього плану, якщо умови в блоці сегментації на рис. 3 перевірені чи ні. Третя частина виконується після процесу сегментації. Він використовує кожен новий вхідний кадр для оновлення своїх параметрів. У цій частині ми намагаємося оновити параметри, які використовуються за допомогою певного підходу. Оскільки обчислення середнього значення та стандартного відхилення (на кожній ітерації) для кожного пікселя є складним, можна використати той факт, що для обчислення нового середнього потрібно лише отримати значення першого пікселя та додати нове значення до попереднього середнього та зробити те саме для стандартного відхилення. Нове середнє буде функцією попереднього середнього, першого значення та нового спостережуваного значення. Нове стандартне відхилення використовуватиме той самий принцип, воно буде функцією попереднього стандартного відхилення, першого значення та нового спостережуваного значення. Крім того, щоб обчислити новий адаптивний радіус, уникається його обчислення (на кожній ітерації), замість цього обчислюється новий радіус тоді і тільки тоді, коли значення відповідного пікселя змінилися. У загальному вигляді, якщо позначити змінну, яка вказує, чи змінилися значення пікселя чи ні, механізм оновлення використовуватиме значення пікселя та значення зміни для обчислення нового радіуса.

Для ініціалізації фонової моделі використовується три підходи. По-перше, використовується процес ініціалізації звичайного методу ViBE для моделювання кожного фонового пікселя за допомогою  $N$  вибірових значень, взятих із їхнього безпосереднього сусідства. По-друге, використовується  $n$  перших кадрів, як у [30], але зі зміною його значення, оскільки збільшення кількості кадрів також підвищує точність результатів. З іншого боку, якщо  $n$  занадто велике, алгоритм стає повільнішим. Тому  $n$  вибране рівним 30 замість 20, як в оригінальному методі. Вибрані кадри використовуються для обчислення середнього значення кожного пікселя, потрібно додати це значення до значень вибірки, щоб отримати  $N + 1$  значення вибірки. Це призводить до моделювання кожного пікселя

$$M = \{v_0(x), v_1, v_2, v_3 \dots v_N\}, \quad (1.1)$$

де

$$v_0(x) = \frac{1}{n} \sum_{t=1}^n v^t(x), \quad (1.2)$$

По-третє, використовується  $N$  вибірових значень, як у [31], щоб обчислити адаптивний радіус для кожного пікселя. Значення отримані емпіричним шляхом у порівнянні зі значеннями, вибраними в оригінальному методі. Після кількох тестів зауважено, що вихідні значення можна змінити, тобто  $\varepsilon_c$ , трохи збільшити, а  $\varepsilon_d$  зменшити. Вибрано  $\varepsilon_c$  рівним 0,08, а не 0,06, а  $\varepsilon_d$  рівним 0,35, а не 0,4. Вирішили позначити адаптивний радіус  $R_{ad}$  як

$$R_{ad} = \begin{cases} R(1 + \varepsilon_c), & \text{якщо } R \leq d_{mean} \delta \\ R(1 + \varepsilon_c) \end{cases}, \quad (1.3)$$

$$d_{mean}(x) = \sum_{i=1}^N D^i(x), \quad (1.4)$$

Де  $D^i(x)$  це відстань між пікселем  $x$  і значенням вибірки  $v_i$ .

Якщо ми позначити  $V$  сегментоване зображення,  $V(x)$  буде значенням пікселя  $x$  у матриці  $V$ , яке дорівнює 0, якщо  $x$  є пікселем фону, і 255, якщо  $x$  є пікселем переднього плану. Тому, щоб класифікувати піксель як фоновий піксель або піксель переднього плану, використовуються такі рівняння для:

$$R_{ad} = \begin{cases} 0, & \text{якщо } U_{\min} i v^t(x) \in S_T(v_0(x)) \\ 255, \end{cases}, \quad (1.5)$$

$$U = |S_{R_{ad}}(v^t(x)) \cap \{v_1, v_2, v_3 \dots v_N\}|, \quad (1.6)$$

$$T = \begin{cases} R_{ad}, & \text{якщо } \sigma \leq \beta R_{ad} \\ \sigma / \beta, \end{cases}, \quad (1.7)$$

$$\sigma = \sqrt{\sum_{t=1}^n (v^t(x) - v_0(x))^2}, \quad (1.8)$$

Для параметра  $U_{\min}$  буде встановлено значення 2, як у [5,30,31], а для  $\beta$  буде встановлено значення 3, як у [30]. Крім того, зауважено, що в конкретній ситуації тремтіння камери та температурних умов вибір коефіцієнта 1,5 замість 4,5, як в оригінальній статті, для оновлення останньої історії фонові моделі призводить до кращих результатів

Щоб оновити модель, використано наступні припущення. По-перше, як зазначено в [5], вважається, що сусідні фонові пікселі мають подібний часовий розподіл і що новий фоновий зразок пікселя також повинен оновити моделі сусідніх пікселів. З цієї причини використано ту саму політику оновлення дати без пам'яті, а також політику вибірки часу. Це дає змогу вибіровим значенням кожного пікселя адаптуватися до зміни фону.

По-друге, використано механізм підрахунку пікселів, який класифікує піксель, який залишається на передньому плані протягом  $K$  послідовних кадрів, як піксель фону. Для досягнення цієї мети підтримується матриця з розміром кадру. У цій матриці кожен елемент представляє кількість разів, коли піксель з'являвся на передньому плані протягом  $K$  послідовних кадрів. Коли піксель класифікується як піксель переднього плану, алгоритм збільшує його відповідний лічильник у матриці. Якщо лічильник перевищує максимальне значення, позначене  $counter_{\max}$ , піксель класифікується як піксель фону. Експерименти показали, що значення 10 для  $counter_{\max}$  призводить до хороших результатів.

Щоб уникнути обчислення параметрів алгоритму після кожного нового кадру, використовується формулу Кеніга, яка справляється краще. Щоб оновити параметри, просто потрібно додати нове значення та відняти попереднє, тому замість того, щоб зберігати середнє значення та стандартні відхилення, зберігається сума та сума квадратів. Це призводить до таких рівнянь:

$$\sum_{t=2}^{n+1} v^t(x) = \sum_{t=1}^n v^t(x) + v^{n+1}(x) - v^1(x), \quad (1.9)$$

$$\sum_{t=2}^{n+1} (v^t(x))^2 = \sum_{t=1}^n (v^t(x))^2 + (v^{n+1}(x))^2 - (v^1(x))^2, \quad (1.10)$$

Якщо позначити через  $v_0^1(x), \sigma^1(x), v_0^{t+1}(x), \sigma^{t+1}(x)$  значення кумулятивного середнього та стандартного відхилення в момент часу  $t$  і  $t + 1$  відповідно, отримуємо їх оновлення за такою формулою:

$$v_0^{t+1}(x) = \frac{1}{n} \sum_{t=2}^{n+1} v^t(x), \quad (1.11)$$

$$\sigma^{t+1} = \sqrt{\left( \sum_{t=2}^{n+1} (v^t(x))^2 \right) - (v^{t+1}(x))^2}, \quad (1.12)$$

Тепер, щоб оновити  $d_{\text{mean}}$ , використано той факт, що нове значення для даного пікселя має бути обчислене, якщо його вибіркові значення були оновлені. Таким чином, використана логічна матриця, де всі елементи ініціалізуються значенням false, і щоразу, коли значення вибірки даного пікселя змінюються, відбувається зміна на значення true для елемента у відповідній позиції в логічній матриці. Кожного разу, коли потрібно обчислити евклідову відстань, щоб переконатися, що вибіркові значення пікселя належать колу радіуса  $R_{\text{ad}}$ , перевіряється, чи його відповідне булеве значення встановлено як істинне. Якщо це так, нове  $d_{\text{mean}}$  обчислюється за допомогою рівняння 9, щоб отримати новий  $R_{\text{ad}}$ . Інакше нічого не робиться, ця методика дозволяє значно скоротити непотрібні обчислення. У процесі прийняття рішення використовуватимуться рівняння 6 – рівняння 13 із заміною  $v_0(x)$  на  $v_0^{t+1}(x)$  та  $\sigma$  на  $\sigma^{t+1}$ .

Реалізація виконана з використанням мови C/C++, платформи openCV [32]. OpenCV використовувався лише для того, щоб скористатися його функціями для захоплення зображень. Набір даних, який використовується для вимірювання продуктивності алгоритмів Changetection.net[33].

Програмна реалізація базується на [34] випуску в липні 2014 року. Також використано BSLibrary[35] для моделювання. Це структура, яка містить до 53 алгоритмів фоновго віднімання, які вже реалізовані від найстарішого до найновішого та складного. Тому модифікувавши структуру, додано три нові алгоритми, а саме вдосконалений ViBE, який використовує  $n$  останніх кадрів для моделювання фону, адаптивний ViBE, який відповідно змінює радіус, і запропонований ViBE.

Коли серія пікселів виявляється як рухомі цілі, але насправді ці пікселі не є рухомими об'єктами [27], утворюється явище привида. Таким чином, алгоритм повинен швидко усунути його видимість. Однак оригінальний ViBE не видаляє його так швидко, як хотілося б, це один із його головних недоліків. Щоб подолати цю складність, використано кумулятивне середнє, обчислене з використанням  $n$  останніх кадрів, щоб посилити класифікацію пікселів.

Механізм підрахунку пікселів, який класифікує як фоновий піксель, який залишається на передньому плані більше ніж  $\text{counter}_{\text{max}}$  послідовних кадрів. Це дозволяє покращити усунення привидів. Щоб чітко проілюструвати, як запропонований метод усуває привид швидше, ніж оригінальний ViBE, використано набір даних зимового під'їзду, де два автомобілі стоять на місці протягом тривалого періоду часу (понад 1800 кадрів), перш ніж один із них вирішить рухатися. Було зроблено наступні зауваження:

- Оригінальний ViBE, запропонований у 2011 році (див. лінію c на малюнку 1.4), потребує більш тривалого періоду часу (понад 600 кадрів), щоб усунути появу привидів
- Для версії, запропонованої в 2014 році тим же автором (див. лінію d на малюнку 1.4), усунення феномену привидів набагато краще. Однак точність і якість результатів знизилася.
- Запропонований метод (див. лінію e на малюнку 1.4) потребує менше 200 кадрів для виконання того самого завдання. Крім того, виявлений привид швидко видаляється (уже видаляється в кадрі 2100), поки він все ще присутній у кадрі 2100 і кадрі 2200 як в оригінальному, так і в покращеному ViBE.

Як було сказано раніше, для складного динамічного фону радіус слід збільшити, щоб фон не можна було легко визначити як передній план. Це пояснюється тим, що в базовому алгоритмі ViBE зіставлення моделі завжди використовує фіксований глобальний радіус. Тим не менш, у [31] показано, що проста політика радіуса ігнорує складність місцевого середовища та невизначеність змін. Тому важко ефективно виявити ціль у складному середовищі. Тому третій внесок полягає в обчисленні для кожного пікселя модифікованого адаптивного радіуса ( $R_{\text{ad}}$ ), який залежить від змінного параметра  $d_{\text{mean}}$ , обчисленого за допомогою рівняння 9. Він також використовує фіксовані параметри  $\epsilon_d = 0,08$ ,  $\epsilon_d = 0,35$ ,  $\delta = 5$ .

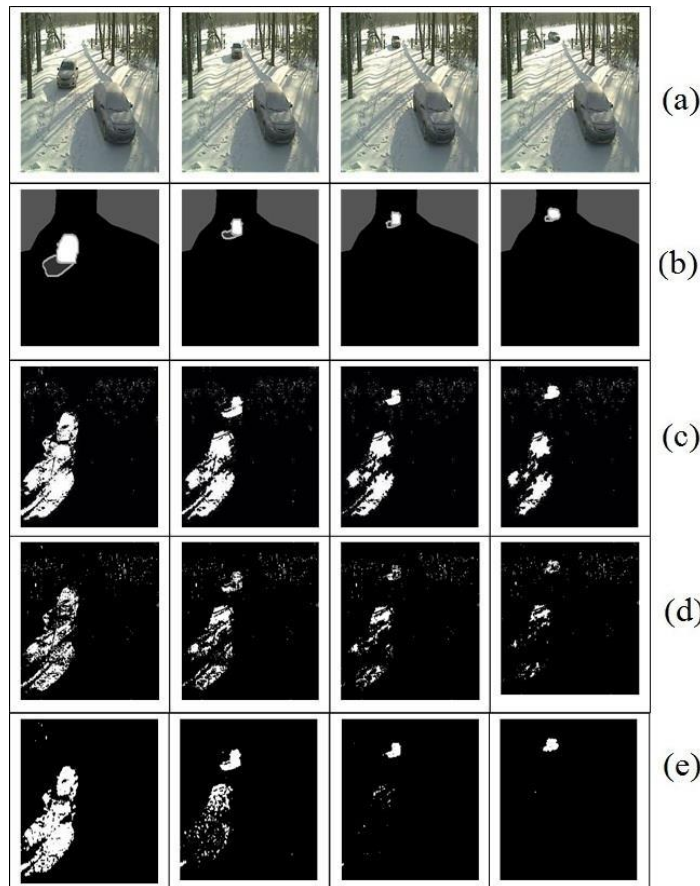


Рис. 4 Ціль рухається зі стаціонарного положення (зимовий шлях руху)

(a) 1900-й, 2000-й, 2100-й, 2200-й кадри; (b) еталон, наданий набором даних; (c) результати виявлення оригінального ViBE; (d) результати виявлення покращеної версії ViBE; (e) результати виявлення запропонованого методу.

Щоб проілюструвати ефект адаптивного радіуса, використано набір даних каное, що рухається по морю, з невеликими хвилями на поверхні. Було зроблено наступні зауваження:

- Зазвичай хвилі на поверхні моря відносяться до фону, але ViBE (див. лінію c на малюнку 1.5) має тенденцію виявляти його як передній план
- Покращення (2014) намагається вирішити цю проблему (див. рядок d на малюнку 1.5), але якість виявленого рухомого човна стає гіршою.
- Запропонований метод вирішує ці проблеми, обчислюючи радіус адаптивним способом, вплив хвилястого фону значно зменшується (див. лінію e на малюнку 1.5).

Виміряна точність, відкликання та F-вимірювання дев'яти алгоритмів: оригінальний алгоритм ViBE, виконання ViBE з використанням кумулятивного середнього та механізму підрахунку пікселів, виконання ViBE з використанням адаптивного радіуса, запропонований метод та деякі складні алгоритми наприклад Codebook[17], KDE[14], MOG[6], PBAS[36] і SUBSENSE[37].

Використані набори даних CDNet 2014, і щоб отримати результати, використано програмне забезпечення BMC-wizard для обчислення цих значень. Процес полягає в наступному: беремо сегментовані зображення, отримані за допомогою вибраного алгоритму, як перший набір параметрів, а другий набір параметрів для майстра BMC є оригінальною основною істинністю вхідних даних, наданих набором даних.

Математично точність  $P_r$ , відкликання  $R$  і F-measure обчислюються за формулою

$$Precision(P_r) = \frac{TP}{TP + FP}, \quad (1.13)$$

$$Recall(R) = \frac{TP}{TP + FN}, \quad (1.14)$$

$$F - Measure(F) = \frac{2PrR}{Pr+R}, \quad (1.15)$$



де **TP** позначає кількість пікселів переднього плану, дійсно класифікованих як передній план, **FP** кількість пікселів фону, помилково класифікованих як передній план, і **FN** кількість пікселів переднього плану, помилково класифікованих як фон.

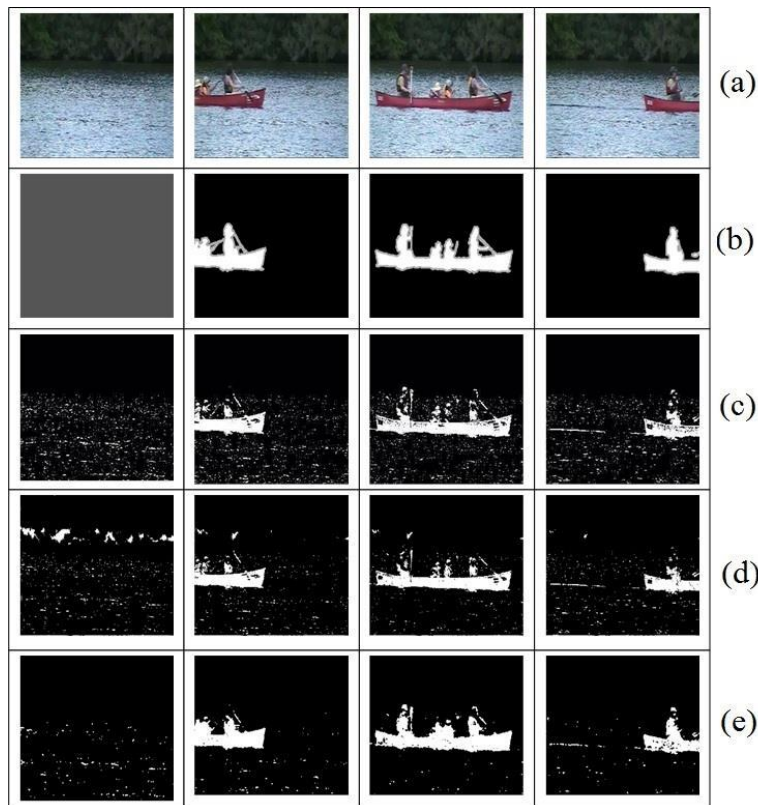


Рис. 5. Оцінка адаптивного радіусу (каное). (а) 600-й, 900-й, 960-й, 1040-й кадри; (б) реальний фон (а); (с) результати виявлення оригінального ViBE; (д) результати виявлення вдосконаленої версії ViBE; (е) результати виявлення запропонованого методу

Ефективність модифікованого алгоритму з точки зору запам'ятовування порівняно зі згаданими алгоритмами перевершує всі вибрані алгоритми із середнім відсотком у 65% випадків (тобто серед 53 різних складних фонів). ситуації, запропоновані набором даних), за винятком алгоритму PBAS, де він перевершує лише 30% випадків. Зрештою, об'єднання цих результатів для вимірювання F-міри (також називається F-показником) показує середній відсоток перевищення продуктивності становить 85% для інших алгоритмів і 40% для алгоритму PBAS.

На малюнку 1.6 представлено результати, отримані від різних виконання алгоритмів для кадрів, взятих випадковим чином у чотирьох наборах даних, а саме: хуртовина (1), парк (2), диван (3) і шлагбаум (4). Рядок (A) — вхідний кадр, рядок (B) — основне зображення істинності, рядок (C) — виконання алгоритму PBAS, рядок (D) — виконання алгоритму Subsense, рядок (E) — виконання оригінального алгоритму ViBE, рядок (F) ) виконання покращеного алгоритму ViBE з ініціалізацією n кадрів, виконання Line (G) адаптивного алгоритму ViBE та Line (H) виконання модифікованого алгоритму.

Таким чином, можна побачити, наскільки точними є запропоновані результати підходу порівняно з іншими алгоритмами щодо основного зображення.

Також можна помітити, що, незважаючи на кількісний аналіз, результати PBAS перевершували модифікований метод. Це сталося тому, що PBAS виявляє багато пікселів фону як пікселя переднього плану, що призвело до модифікації параметрів виявлення (TP, FP і FN), і, як наслідок, були отримані хороші кількісні результати.



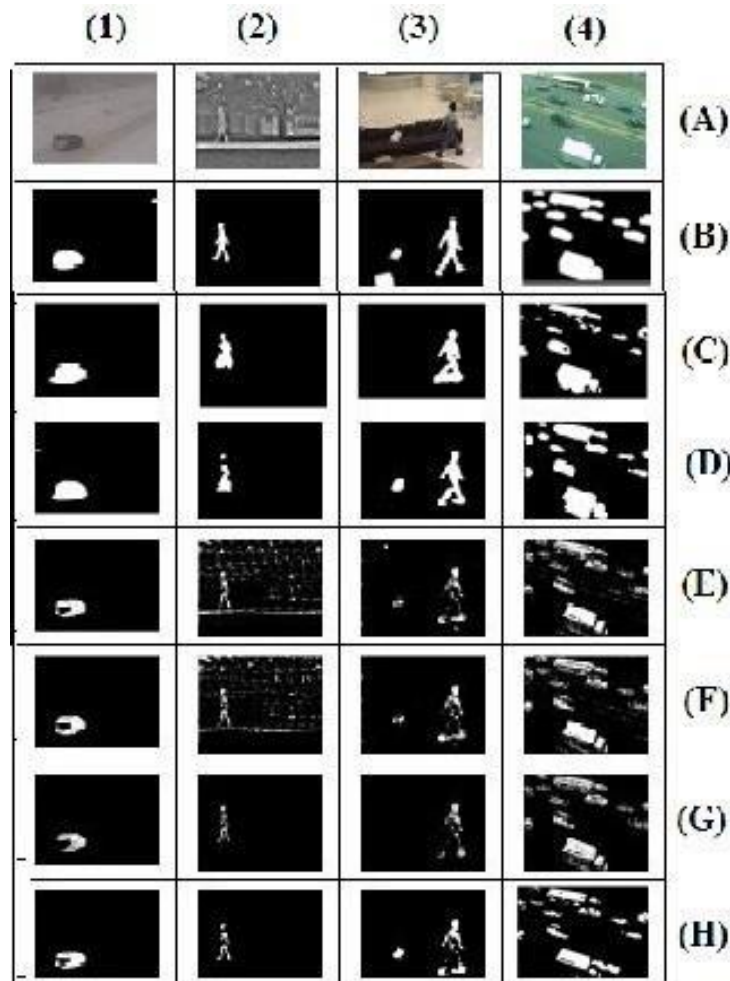


Рис. 6 Аналіз результатів

### ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У роботі представлено ефективний алгоритм віднімання фону на основі ViBE для складного фону. Запропонований метод був детально пояснений із усіма необхідними параметрами. Він поєднує в собі вдосконалений ViBE, який використовує кумулятивне середнє значення та механізм підрахунку пікселів для швидкого усунення привидів, і адаптивний ViBE, який обчислює адаптивний радіус залежно від зміни фону. Наведено отримані результати, що демонструють ефективність методу в порівнянні з існуючими. З середньою частотою 30 кадрів в секунду його також можна використовувати в програмах реального часу.

### References

- 1 S. S. Hossain, S. Khalid, C. Nabendu, Moving Object Detection Using Background Subtraction, Springer Briefs in Computer Sciences, Springer, Cham, 2014.
- 2 G. Yao, T. Lei, J. Zhong, P. Jiang, W. Jia, Comparative evaluation of background subtraction algorithms in remote scene videos captured by mwir sensors, Sensors (2017).
- 3 F. Zeng, G. Zhang, J. Jiang, Text image with complex background filtering method based on harris corner-point detection, journal of software 8 (8) (2013).
- 4 D. Li, Y. Li, F. He, S. Wang, Object detection in image with complex background, Third International Conference on Multimedia Technology (2013).
- 5 O. Barnich, M. V. Droogenbroeck, Vibe: A powerful random technique to estimate the background in video sequences, International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2009) 19–24.
- 6 C. Stauffer, E. Grimson, Adaptive background mixture models for real-time tracking, Computer Vision and Pattern Recognition (1999) 246–252.
- 7 B. Han, X. Lin, Update the GMMs via adaptive Kalman filtering, International Society for Optical Engineering (2005) 1506–1515.
- 8 Y. Hong, Y. Tan, J. Tian, J. Liu, Accurate dynamic scene model for moving object detection, International Conference on Image Processing (ICIP) (2007) 157–160.
- 9 W. Zhang, X. Fang, X. Yang, J. Wu, Spatio-temporal Gaussian mixture model to detect moving objects in dynamic scenes., Journal of Electronic Imaging (2007).
- 10 P. Tang, L. Gao, Z. Liu, Salient moving object detection using stochastic approach filtering, Fourth International Conference on Image and Graphics (ICIG) (2007) 530–535.

- 11 M. Harville, A framework for high-level feedback to adaptive, perpixel, mixture-of-Gaussian background models, 7th European Conference on Computer Vision (ECCV) (2002) 543–560
- 12 M. Cristani, V. Murino, A spatial sampling mechanism for effective background subtraction, 2nd International Conference on Computer Vision Theory and Applications (VISAPP) (2007) 403–410
- 13 A. Elgammal, D. Harwood, L. Davis, Non-parametric model for background subtraction, European Conference on Computer Vision(2000).
- 14 A. Elgammal, R. Duraiswami, D. Harwood, L. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE 90 (2002) 1151–1163.
- 15 Park, C. Lee, Bayesian rule-based complex background modeling and foreground detection optical engineering, Optical Engineering (2010)
- 16 P. Angelov, P. S. Tehran, R. Ramezani, An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving Takagi-Sugeno Fuzzy systems, International Journal of Intelligent Systems (2011) 189–205.
- 17 K. Kim, T. H. Chalidabhongse, D. Harwood, L. Davis, Real-timeforeground background segmentation using codebook model, Real time imaging 11 (2005) 172–185.
- 18 N. M. Oliver, B. Rosario, A. P. Pentland, A bayesian computer vision system for modeling human interactions, IEEE transactions on pattern analysis and machine intelligence 22 (2000) 831–843.
- 19 D. Kuzin, O. Isupova, L. Mihaylova, Compressive sensing approaches for autonomous object detection in video sequences, Sensor Data Fusion: Trends, Solutions, Applications (SDF) (2015).
- 20 V. Cevhe Aswin, S. M. F., D. Dikpal, R. R. G., B. R. Chellappa, Compressive sensing for background subtraction, European Conference on Computer Vision (2008) 155–168.
- 21 V. Cevher, A. Sankaranarayanan, M. F., D. Dikpal, R. R. G., B. R. Chellappa, Background subtraction using spatiotemporal group sparsity recovery, European Conference on Computer Vision(2008) 155–168.
- 22 A. Azeroual, K. Afdel, Ieee transactions on circuits and systems for video technology: Background subtraction based on low-rank and structured sparse decomposition, IEEE Trans. Image Process 24 (08) (2017) 2502–2514.
- 23 A. Azeroual, K. Afdel, Fast image edge detection based on faber schauder wavelet and otsu threshold, Heliyon 3 (2017), doi: 10.1016/j.heliyon.2017.e00485.
- 24 M. Nishio, C. Nagashima, S. Hirabayashi, A. Ohnishi, K. Sasaki, T. Sagawa, M. Hamada, T. Yamashita., Convolutional auto-encoder for image denoising of ultralow-dose ct, Heliyon 3 (2017), doi:10.1016/j.heliyon.2017.e00393.
- 25 M. Braham, M. V. Droogenbroeck, Deep background subtraction with scene-specific convolutional neural networks, International Conference on Systems, Signals and Image Processing (2016).
- 26 Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradientbased learning applied to document recognition, Proceeding of IEEE 86 (1998) 2278–2324.
- 27 O. BARNICH, M. V. DROOGENBROECK, Vibe : A universal background subtraction algorithm for video sequences., IEEE Transactions on Image Processing (2011) 1709–1724.
- 28 A. Elgammal, D. Harwood, L. Davis, Non-parametric model for background subtraction, 6th European Conference on Computer Vision Part II, Springer 1843 (2000) 751–767.
- 29 H. Wang, D. Suter, A consensus-based method for tracking: Modeling background scenario and foreground appearance, Pattern Recognition 40 (2007) 1091–1105.
- 30 H. Batao, Y. Shaohua, An improved background subtraction method based on vibe, Chinese Conference on Pattern Recognition, Springer 662 (2016) 356–368
- 31 L. Chang, Zhenghua, Y. Ren, Improved adaptive vibe and the application for segmentation of complex background, Hindawi Publishing Corporation, Mathematical Problems in Engineering (2016).
- 32 Itseez, Open source computer vision library, <https://github.com/itseez/opencv> (2015).
- 33 Y. Wang, G. P. M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Y. Benezeth, P. Ishwar, Cdnet 2014: An expanded change detection benchmark dataset, 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (2014) 16–21.
- 34 B. Laugraud, vibe-sources, source code in c/c++ original implementation. example for opencv, University of Liege, Belgium, <http://orbi.ulg.ac.be/bitstream/2268/145853/5/vibe-sources.zip> (2014).
- 35 A. Sobral, Bgslibrary: An opencv c++ background subtraction library, In Proceedings of the 2013 IX Workshop de Viso Computacional, Rio de Janeiro, Brazil (2013) 3–5.
- 36 M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: The pixel-based adaptive segmenter, In Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA (2012).
- 37 St-Charles, G. A. Bilodeau, R. Bergevin, Flexible background subtraction with self-balanced local sensitivity, In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA (2014) 414–419.