

Дослідження методів та засобів виявлення об'єктів на відеозображеннях

© Стоян А.О. 2020

Розглянуто деякий ряд методів та засобів які дозволяють виявити об'єктів на відеозображенні. Проведено їхнє програмне моделювання на апаратній платформі обчислювальної плати Raspberry Pi 4 на базі процесора Quad core Cortex-A72 64-bit SoC. Здійснений порівняльний аналіз вибраних методів.

Ключові слова: виявлення об'єктів на відеозображенні, особливі точки, детектор ORB, комп'ютерний зір, виявлення рухомих об'єктів, HSV колірна модель

A number of methods and tools that allow detecting objects in a video image are considered. Their software simulations were performed on the hardware platform of the Raspberry Pi 4 computing board based on the Quad core Cortex-A72 64-bit SoC processor. A comparative analysis of the selected methods is carried out.

Keywords: object detection, feature points, keypoints, ORB detector, computer vision, motion detection, HSV model color

Вступ. Виявлення об'єктів є не від'ємною задачею в галузі комп'ютерного зору. Остатнім часом на неї покладають велику надію. Багато фахівців комп'ютерного зору намагаються розробити універсальний алгоритм, який би дав би можливість виявляти будь-які об'єкти на відеозображенні. Але як показала практика, це не цілком можливо. Причина в тому, що різні умови задачі вимагають різні підходи для її розв'язку. Наприклад, дві з головних причини які можуть виникати при реалізації виявлення об'єктів наступні:

- Продуктивність апаратури на якій реалізовується задача виявлення об'єктів відеозображення;
- Умови навколишнього середовища в якому здійснюється процес виявлення об'єктів з відео-камери;

В даній роботі, для демонстрації буде розглянуто наступні методи, а саме:

- Виявлення об'єктів за допомогою кольорових фільтрів;
- Виявлення рухомих об'єктів за допомогою між-кадрової різниці;
- Виявлення об'єктів за допомогою їхнього еталону через особливі точки детектором ORB

На основі цих методів, буде проведено порівняльний аналіз та моделювання на вбудованій системі для зняття показників. І відповідно за цими показниками буде прийняте рішення, в яких цілях використовувати дані методи та під які конкретні завдання.

Стан проблеми. При реалізації пошуку об'єктів на відеозображенні в комп'ютерному зорі, виникає проблема вибору необхідного метода (або методів) для реалізації її на конкретній апаратній платформі, а також виділення конкретних ознак шуканого об'єкта, яка має відрізнити їх серед інших об'єктів кандидатів. Дана проблема полягає в тому, що на даний момент не існує універсального методу який би дав змогу знаходити необхідні об'єкти на відеозображенні та мати хорошу продуктивність для будь-якого рівня систем обчислювальної складності [1].

Для прикладу, необхідно реалізувати метод виявлення об'єкта на апаратній платформі з середньою чи малою продуктивністю. При цьому, даний метод повинен знаходити об'єкт за мінімальний можливий час, та враховувати якомога більше ознак шуканого об'єкта для його пошуку на відеозображенні. Тому постає задача знаходження того методу, який би володів хорошим балансом між швидкістю (на апаратурі з малою продуктивністю) та враховування більшості ознак шуканого об'єкта.

Постановка задачі. Розглянути наступні методи виявлення об'єктів на відео-потоці, а саме:

- Виявлення об'єктів за допомогою кольорових фільтрів;
- Виявлення рухомих об'єктів за допомогою між-кадрової різниці;
- Виявлення об'єктів за допомогою їхнього еталону через особливі точки детектором ORB

Провести їхнє моделювання на апаратній платформі з середньою продуктивністю (в даному випадку проводиться на Raspberry PI 4) та на основі отриманих даних і порівняльного аналізу, зробити висновок щодо їхнього застосування.

Розв'язання задачі. Розглянемо основні принципи роботи 3-ох методів виявлення об'єктів на відеозображенні.

Почнемо з методу виявлення об'єктів за допомогою кольорових фільтрів. Даний метод базується на основі фільтрації зображення за кольором. Тобто пошук об'єкта здійснюється за ознакою його кольору. На практиці прийнято застосовувати цей метод, тоді коли об'єкт суттєво відрізняється за кольором від заднього фону і умови освітлення. Основний порядок алгоритму наступний:

1. Отримання кадру;
2. Конвертація отриманого RGB-кадру в HSV-кадр;

HSV (hue, saturation, value) – це колірна модель представлення зображення. В цій моделі є 3 головні параметри [2]:

a) Hue — колірний тон, (наприклад, червоний, зелений або синьо-блакитний). Варіюється в межах 0-360°, але іноді приводиться до діапазону 0-100.

b) Saturation — насиченість. Варіюється в межах 0-100.

c) Value — Brightness — яскравість. Також задається в межах 0-100.

Як видно з наведених вище параметрів, мета даної конвертації – це комфортне людське представлення параметрів кольорів, за допомогою яких можна в зрозумілому вигляді вибрати необхідний діапазон.

Процес RGB конвертації в HSV, проходить за наступною формулою [3]:

$$H = \begin{cases} 0, & \text{якщо } MAX = MIN \\ 60 \times \frac{G - B}{MAX - MIN} + 0, & \text{якщо } MAX = R \text{ і } G \geq B \\ 60 \times \frac{G - B}{MAX - MIN} + 360, & \text{якщо } MAX = R \text{ і } G < B \\ 60 \times \frac{B - R}{MAX - MIN} + 120, & \text{якщо } MAX = G \\ 60 \times \frac{R - G}{MAX - MIN} + 240, & \text{якщо } MAX = B \end{cases}$$
$$S = \begin{cases} 0, & \text{if } MAX = 0; \\ 1 - \frac{MIN}{MAX}, & \text{інакше} \end{cases}$$
$$V = MAX$$

(1.1.)

3. Встановлення необхідного порогу кольорів за допомогою HSV моделі;

4. Вибраний діапазон кольорів застосовується до кадру (відповідні пікселі які попадають в дану межу, зберігаються на матриці, інакше – позначаються чорними). Тут відповідно й застосовується “бінаризація” по кольору;

5. Отриманий кадр з виділеними об'єктами – бінаризується ще раз, з метою зменшення інформації. Але перед тим необхідно конвертувати RGB-кадр в півтонне зображення, а вже потім за порогом яскравістю здійснювати бінаризацію.

6. Видалення шумів за допомогою методу математичної морфології (наприклад операцією erode а потім dilate);

7. Здійснення пошуку об'єкта за контурами.

На рис. 1 відображено результат роботи даного методу. Алгоритм добре підходить для роботи на обчислювальних системах з малою продуктивністю.

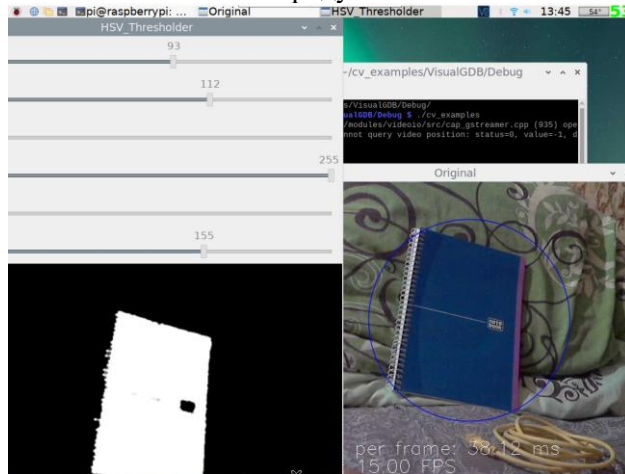


Рис. 1 Демонстрація виконання методу виявлення об'єктів за допомогою кольорових фільтрів

Метод обчислення між-кадрової різниці (frame difference) є простішим та більш розповсюдженим методом виявлення рухомих об'єктів у відео-поточці і призначений тільки для виявлення рухомих об'єктів на статичній камері. Тобто задній фон на якому виділяється об'єкт, повинен як мінімум не змінювати своє положення. Хорошим прикладом застосування є відеореєстратор машин чи людей на вулиці, де задній фон є статичним та відсутні зовнішні завади, такі як дерева і різкі перепади тіні [4].

Алгоритм роботи цього методу наступний:

1. Відеокамера у відповідний час, фіксує послідовно два кадра;
2. Послідовні два кадра, які в ЕОМ на програмному рівні представляються як двох-вимірні матриці, конвертується в монохромне (півтонне) зображення, з метою прискорення обробки алгоритму за рахунок зменшення обсягу інформації;
3. Відбувається по-піксельне обчислення різниці двох монохромних матриць [5]:

$$d_t(x, y) = I_t(x, y) - I_{t-1}(x, y) \quad (1.2.)$$

де:

- $I_t(x, y)$ – інтегральне значення пікселя з координатою x, y отриманого кадру в часі t
- $I_{t-1}(x, y)$ – інтегральне значення пікселя з координатою x, y отриманого кадру в часі $t-1$

4. Отримана матриця різниці $d_t(x, y)$ порівнюється з порогом T , де в результаті отримується бінарна маска:

$$m_t(x, y) = \begin{cases} 0, & \text{якщо } d_t(x, y) < T \\ 1, & \text{в іншому випадку} \end{cases} \quad (1.3.)$$

де:

- $m_t(x, y)$ - значення t -го елемента маски,
 - T – встановлений поріг (рівень чутливості)
5. Отримана маска обробляється методом математичної морфології;
 6. Здійснення пошуку об'єкта за контурами.

Також потрібно зазначити, що цей метод на практиці краще працює коли об'єкти далеко віддалені і на відеокамері пересуваються повільно. На рис. 2. продемонстровано роботу даного методу.

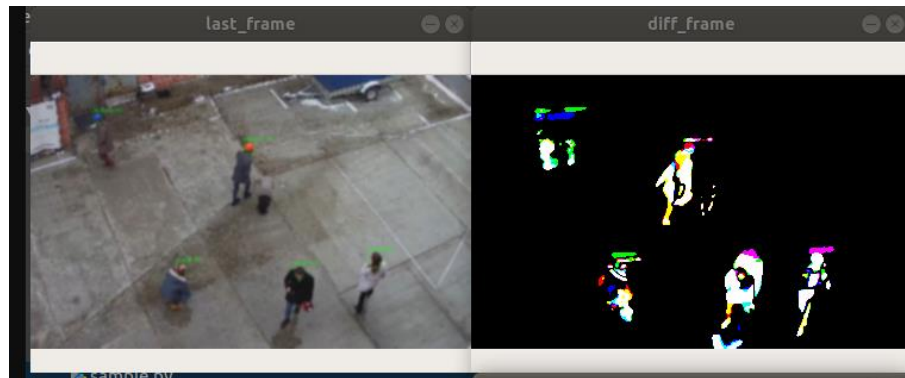


Рис. 2 Демонстрація виконання методу виявлення об'єктів за допомогою між-кадрової різниці

Хорошим методом виявлення конкретного об'єкта на кадрі, є пошук його за еталоном. Маючи оригінальне зображення об'єкта (тобто його еталон), можна порівняти його з кожним отриманим кадром із відео-потoku. Область кадра яка співпадає з еталоном зображення і має відповідні ознаки схожості - вважається шуканою.

Існує багато методів які дозволяють шукати об'єкти за їхніми еталонами, але з найбільш продуктивним та надійнішим вважають метод пошуку об'єкта за допомогою особливих точок. Основна причина в тому, що цей метод значно менше вимагає зберігання необхідної інформації про еталон об'єкта для його пошуку.

“Особливою точкою” – називають точкою зображення, яка має деякі особливості на відміну від інших точок даного зображення. Ознакою особливості точки може бути наприклад різкий перепад кольорів, яскравості околиці, кути, краї і т.д. І як правило ці ознаки повинні бути особливими, щоб їх можна було здійснити пошук на іншому зображенні. Процес виявлення та фіксування особливих точок досягається шляхом використання детектора та дескриптора.

“Детектор” - метод отримання особливих точок із зображення.

“Дескриптор” - називають кінцевий вектор, який узагальнює властивості особливої точки. Використовують дескриптор для ідентифікації точок, з метою виділення її як особливої серед інших точок.

Існує багато алгоритмів визначення особливих точок та їхніх дескрипторів які призначенні для різних областей застосування. В даній роботі розглянемо ORB алгоритм, який себе добре зарекомендував у використанні вбудованих системах.

Пошук особливих точок за алгоритмом ORB у деяких завданнях дозволяє виявляти особливості зображення в режимі реального часу. Такий підхід до пошуку особливих точок і їх опису, дозволяє використовувати його на менш потужних ЕОМ.

Для знаходження особливих точок використовується алгоритм FAST (Features from Accelerated Segment Test) [6], який вже давно зарекомендував себе як швидкий інструмент для детектування локальних особливостей. Метод FAST не є інваріантний до поворотів. Щоб це виправити, автори внесли в оригінальний алгоритм зміни.

Для опису (дескрипторів) особливих точок використовується дескриптор BRIEF (Binary Robust Independent Elementary Features) [7]. Основна його мета полягає в тому, щоб якусь область з зображення можна було розпізнати з різних ракурсів при невеликих обчислювальних витратах. В результаті роботи цього дескриптора кожна точка описується в вигляді бінарного вектора.

Алгоритм ORB використовує детектор кутів Харріса, щоб виявити особливі точки, залишивши при цьому з них тільки певну кількість. Отже, щоб можна було порівняти швидкість роботи, в якості вхідного параметра у алгоритму ORB необхідно задати середню кількість особливих точок, знайдених SURF [8-10].

Також алгоритм ORB використовує піраміду багатомасштабних зображень. Піраміда зображення (image pyramid) - це багатомасштабне зображення одного зображення, яке складається з послідовностей зображень, якими є версіями усіх зображення з різною роздільною здатністю. Кожен рівень піраміди містить зменшену версію зображення, ніж попередній рівень. Таким чином, ORB є частковим інваріантом щодо зміни масштабу зображення.

На рис. 3. продемонстровано роботу методу пошуку особливих точок детектором ORB. Як видно на рис. 3-б, алгоритм добре справляється з розпізнаванням обернутих об'єктів.

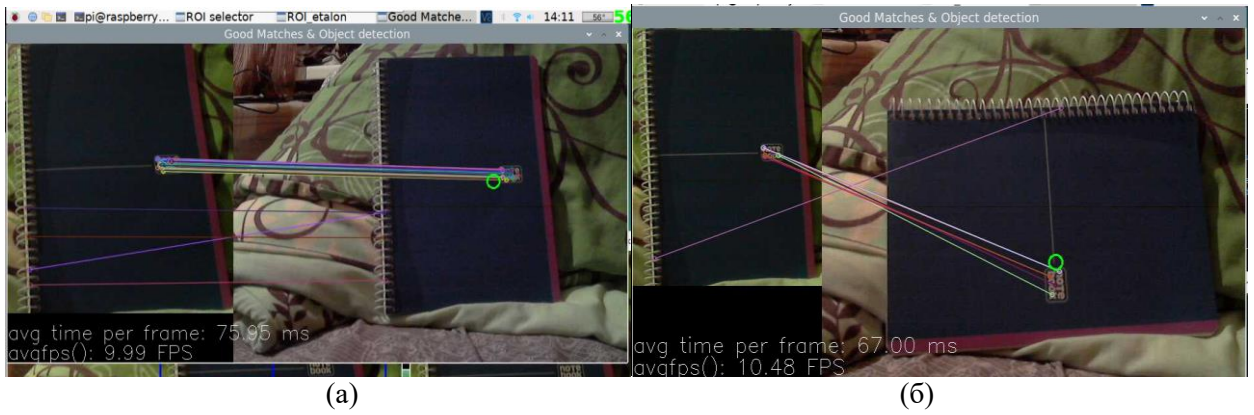


Рис. 3. а, б - Демонстрація виконання методу виявлення об'єктів через особливі точки детектором ORB

Моделювання даних методів проводилося на апаратній платформі Raspberry pi 4. Функціональна схема якої наведена на рис. 4.

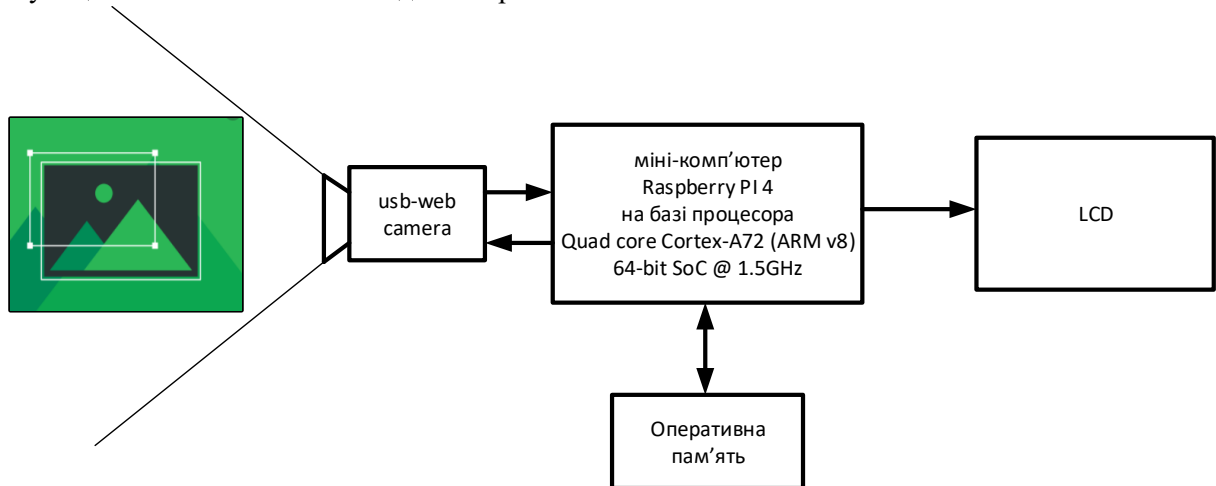


Рис. 4. Функціональна схема підключення апаратних засобів для пошуку об'єктів на відеозображенні

В таблиці 1.1. наведено середній час виконання алгоритмів, а також величину FPS, яка вказує на кількість кадрів в секунду яку камера встигає оновити при виконанні алгоритму. Також слід зазначити, що чистий показник FPS камери без виконання алгоритмів складає рівно 16 FPS.

Таблиця 1.1.

Порівняльна характеристика 3-ьох методів розпізнавання об'єктів на відеозображенні

	FPS (кількість кадрів в секунду)	Час виконання, мс
Метод виявлення об'єктів за допомогою кольорових фільтрів	14.1	34.5
Метод виявлення рухомих об'єктів за допомогою між-кадровою різницею	15.3	12.1
Метод виявлення об'єктів через особливі точки детектором ORB	9.6	71.39

Згідно з результатами таблиці 1.1. бачимо, що перші два методи працюють ефективно за показником FPS камери, при цьому другий метод за часом виконання є найшвидшим серед своїх кандидатів. Метод виявлення об'єктів через особливі точки детектором ORB хоч є повільнішим серед інших методів, але не так критично, як здавалося на перший погляд. Покази її FPS-у є стабільними. Тому з наведених вище 3-ьох методів, стає зрозуміло і очевидно, що чим більше ознак об'єкта за якими ми хочемо здійснювати пошук на відеозображенні, тим більше навантаження буде здійснюється на обчислювальну систему.

Також при моделюванні, всі три методи показали хороші результати в плані часу виконання. Але все ж таки методи, такі як виявлення об'єктів за допомогою кольорових фільтрів та метод виявлення рухомих об'єктів за допомогою між-кадрової різниці несуть малу інформацію

про об'єкт. І тому, якщо ставиться задача пошуку конкретного об'єкта на відеозображенні, в цьому випадку слід застосувати метод виявлення об'єктів за допомогою особливих точок детектором ORB.

Висновки. В результаті виконання роботи було розглянуто ряд методів, які відповідно до умов задач, дозволяють виявляти об'єкти на відеозображенні. Здійснено їхнє моделювання на апаратній платформі Raspberry pi 4 та проведено порівняльний аналіз даних методів. На основі отриманих показників можна зробити висновок, що для різних умов задач пошуку об'єктів на відеозображенні, існують різні підходи, щодо застосування методів для їхнього виявлення.

Література

1. Анализ и обработка изображений в задачах компьютерного зрения/ В.В. Селянкин, С.В. Скороход - 2015;
2. A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm / Monika Deswal1 , Neetu Sharma2 - International Journal of Science and Research (IJSR);
3. HSV (Wikipedia);
4. A Comparison between Background Modelling Methods for Vehicle Segmentation in Highway Traffic Videos / 2018;
5. Adaptive Motion Detection Algorithm using Frame Differences and Dynamic Template Matching Method // November 2012 Conference: The 9th International Conference on Ubiquitous Robots and Ambient Intelligence;
6. Rosten E., Drummond T. Machine learning for high speed corner detection // 9th European Conference on Computer Vision. Vol. 1. 2006. pp. 430–443.
7. An Improved ORB Algorithm of Extracting and Matching Features / Lei Yu1 , Zhixin Yu1 and Yan Gong2 - International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 8, No. 5 (2015), pp. 117-126;
8. Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // 11th European Conference on Computer Vision (ECCV), Heraklion, Crete. LNCS Springer, September 2010;
9. Introduction to ORB (Oriented FAST and Rotated BRIEF) / Deepanshu Tyagi;
10. ORB: an efficient alternative to SIFT or SURF / Ethan Rublee, Vincent Rabaud, Kurt Konolige Gary Bradski - Willow Garage, Menlo Park, California;
11. Пуйда В.Я. Реалізація алгоритму покадрового визначення координат об'єкта у монохромному відеопотоці. Вісник «Комп'ютерні науки та інформаційні технології».- Львів: НУ «Львівська політехніка», 2015.- №830.