

Список використаних джерел

- [1] Топ-7 библиотек для Python для компьютерного зрения [Он-лайн].
Доступно: <https://arboook.com/kompyuternoe-zrenie/moj-top-7-bibliotek-dlya-python-dlya-kompyuternogo-zreniya/>.
- [2] P. Viola and M. Jones. Robust real-time face detection. IJCV 57(2), 2004.
- [3] Lienhart R., Kuranov E., Pisarevsky V.: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In: PRS 2003, pp. 297-304 (2003).
- [4] Cascade Classifier Training [Он-лайн]. Доступно:
https://docs.opencv.org/3.4/dc/d88/tutorial_traincascade.html.

УДК 004.93.1

ВИЯВЛЕННЯ РУХОМИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

Васильков Д. Р., Пронін С.В.

Харківський національний автомобільно-дорожній університет, Харків

Задачі комп'ютерного зору у транспортних системах

Сучасні засоби відеоспостереження дозволяють вирішувати завдання моніторингу транспортних потоків шляхом аналізу відеоданих, що надходять з камер відеоспостереження. Результатом аналізу є ідентифікація транспортного засобу на дорозі, збережені зображення і параметри транспортних засобів у внутрішній базі даних. При необхідності система може сигналізувати про різні події [1].

Завдання, які вирішуються:

- розпізнавання державного реєстраційного знака;
- вимірювання швидкості руху транспортного засобу;
- детектування ДТП і «затор»;
- детектування порушень ПДР;
- визначення та класифікація транспортних засобів;

- збір статистики транспортного потоку;
- трансляція зображення в центр спостереження.

Відеоспостереження на транспорті дозволяє вирішувати широке коло завдань забезпечення безпеки пасажирів, збереження вантажів і самих транспортних засобів.

Огляд алгоритмів і методів пошуку об'єктів на зображенні

Завдання пошуку об'єкта за зразком є підзадачею розпізнавання образів.

При ненаправленим і несистематизированном пошуку схожих об'єктів з безлічі їх можна перераховувати досить довго і не досягти завершення із заданою точністю. У більшості випадків об'єкти характеризуються такими параметрами як форма, колір, положення і т.п. Об'єкти класифікуються залежно від цих параметрів. При цьому нерідко немає завдання класифікувати всі можливі об'єкти з вхідного відеопотоку, найчастіше потрібно виділити (розпізнати) тільки об'єкти певного типу.

На основі аналізу алгоритмів пропонується розглянути алгоритми які ґрунтуються на вирахуванні фону [2].

Задача вирахування фону на зрівнянні поточного кадру з попереднім. З відеопотоку береться кадр, зрівняється з попереднім та знаходиться різниця між ними (рис. 1).

В фреймворках таких як `opencv` для рішення задачі віднімання фону пропонується ряд методів серед яких можна виділити методи визначення порогу та алгоритм сегментації фону (переднього плану) на основі гаусової суміші.

При використанні методів віднімання порогу результатом буде маска, де в чорному є частини, які є зображеннями в обох зображеннях і білих частинах, які відрізняються.

В `opencv` методи віднімання порогу реалізуються за допомогою функції `cv2.threshold` [3].

Функція `cv2.threshold` використовується для визначення порогу. Перший аргумент - це вихідне зображення, яке повинно бути зображенням у відтінках

сірого. Другий аргумент - це граничне значення, яке використовується для класифікації значень пікселів. Третій аргумент - це максимальне значення, яке присвоюється значенням пікселів, що перевищує поріг. OpenCV надає різні типи порогових значень, які задаються четвертим параметром функції. Базова установка порогових значень виконується з використанням типу `cv2.THRESH_BINARY`:

```
gray_frame = cv2.GaussianBlur (gray_frame, (5, 5), 0)//використовуємо розмиття Gaussian.
```

```
difference = cv2.absdiff (first_gray, gray_frame) // Обчислити абсолютну різницю між масивами (картинками).
```

```
_, difference = cv2.threshold (difference, 25, 255, cv2.THRESH_BINARY)
```

Метод повертає два виходи. Перший - це поріг, який використовувався, а другий вихід - це зображення з граничним значенням (рис.1).

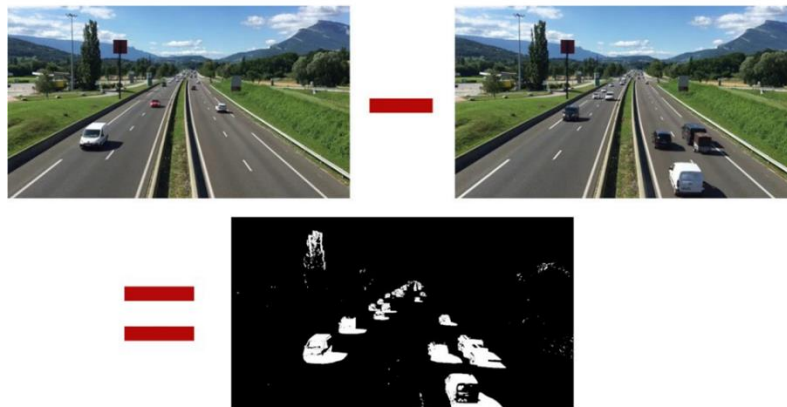


Рисунок 1 – Результат роботи віднімання фону

Метод визначення порогу за допомогою гаусової суміші у `opencv` реалізовано у вигляді функції `SubtractorMog2` [4] та має такий вигляд:

```
subtractor = cv2.createBackgroundSubtractorMOG2 (history=100, varThreshold=50, detectShadows=True)
```

`SubtractorMog2` має перевагу роботи з історією кадру, вона працює за замовчуванням з останніми 120 кадрами, але ви можете змінити її, і ми побачимо, як пізніше).

Гаусова суміш вивчає фон відповідно до історією кадрів. Для визначення

кількості кадрів які впнуть на побудову фону використовується параметр history. Виявлення тіней реалізується за допомогою порогу Тау який визначає, наскільки темніше може бути тінь. Якщо $Tau = 0,5$ це означає, що якщо піксель більш ніж в два рази темніше, то це не тінь

Алгоритм виявлення фону буде включати наступні дії:

- віднімається задане зображення з відомого фону і граничне значення для отримання маски переднього плану
- застосовується операція над маскою переднього плану і даними зображенням, щоб отримати об'єкт з можливими тінями.
- видаляються пікселі, які темніше, ніж їх відповідний піксель в фоновому режимі.

Результатом роботи методу буде наступне зображення (рис. 2).



Рисунок 2 – Робота програми за допомогою Manual way

Список використаних джерел

- [1] ВИДЕОКОНТРОЛЬ-Рубеж [Електронний ресурс]. Доступно: <http://www.ollie.com.ua/videocontrol/index.html>.
- [2] Огляд алгоритмів і методів пошуку об'єктів на зображенні [Електронний ресурс]. Доступно: https://ru.bmstu.wiki/Задачи_поиска_по_образцу

- [3] Image Thresholding [Електронний ресурс]. Доступно: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html
- [4] Background Subtraction [Електронний ресурс]. Доступно: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html

УДК 004.62.77

СТВОРЕННЯ РЕЛЕВАНТНОГО КОНТЕНТУ ДЛЯ АНАЛІЗУ ПОТРЕБ СПОЖИВАЧІВ

Циганок О. П., Пронін С.В.

Харківський національний автомобільно-дорожній університет, Харків

Особливості створення релевантного контенту. Сьогодні клієнт може контактувати з компанією в різних каналах - через мобільні пристрої, соціальні платформи, рекламні банери, магазини, телевізор і т.д., Це приводить до ситуації, що відслідковувати шлях клієнта стає все складніше. Для рішення цієї задачі на сьогодні існують такі інструменти як Web Mining, Big data, методи машинного навчання. Вони дозволяють зрозуміти, що призводить до продажу і які канали неефективні. Це допомагає компаніям оптимізувати комунікацію і конверсію в різних каналах. Споживачі використовують канали по-різному, і компанії повинні враховувати це. Наприклад, деякі сегменти клієнтів вважають за краще отримувати інформацію про продукт з блогів перед відвідуванням сайту. Тому головна сторінка сайту повинна містити релевантні пропозиції та посилання на додаткову інформацію. [1]

У якості підходу до вирішення задачі пропонується застосування сегментації клієнтів по самому різному набору параметрів.

Для того щоб була можливість застосувати метод сегментації необхідно провести збір та обробку первинної інформації. Для цього скористаємося