

ОТСЛЕЖИВАНИЕ ДВИЖЕНИЯ И ОЦЕНКА СКОРОСТИ ОБЪЕКТА В ВИДЕОПОТОКЕ

А.Т. Левинский

Н.М. Селюто

И.Д. Родионов

adam.levinskiy@yandex.ru

selyuto@icloud.com

ir@drp.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Статья посвящена разработке метода оценки скорости объекта в видеопотоке с движущейся камеры. Рассмотрены методы и алгоритмы решения указанной задачи, а также методы стабилизации видеопотока и методы отслеживания движущихся объектов в видеопотоке. Проведен обзор аналогичных систем слежения и разработан метод оценки скорости перемещения объекта в видеопотоке. Произведена программная реализация всех выбранных и разработанных методов. В ходе исследования разработано программное обеспечение, которое позволяет отслеживать скорость в двумерной плоскости.

Ключевые слова

Распознавание, погрешность, интерфейс, тестирование, метод Левенштейна, стабилизация, отслеживание движения объекта, перемещение объекта

Поступила в редакцию 31.10.2017

© МГТУ им. Н.Э. Баумана, 2017

Введение. Специалисты в области компьютерного зрения часто сталкиваются с ситуацией, когда необходимо проследить за каким-либо объектом и измерить его скорость передвижения. При этом в век информационных технологий, можно минимизировать затраты на дополнительное оборудование и заменить его программными методами решения задач.

Рассмотрим гипотетический пример из военной сферы. Предположим, что необходимо отследить скорость перемещения вражеского объекта. Традиционно будем использовать метод радиолокации, тогда вражеский объект можно обнаружить с помощью радиоволн и сменить тактику ведения боя. При этом измерить скорость передвижения вражеского объекта необходимо без использования средств, которые могут быть обнаружены. Подобные ситуации в действительности случаются нередко, поэтому разработка алгоритмов и методов оценки скорости движения объекта в видеопотоке актуальна.

Решение этой задачи позволит в дальнейшем отказаться от дополнительного оборудования, которое используется во многих системах слежения и взаимодействует с видеокamerой. Применение только камеры видеонаблюдения поможет оценить перемещение объектов в реальной жизни. В настоящей работе рассмотрен вариант реализации такого метода с использованием среды разработки программного обеспечения.

Выбор алгоритма стабилизации видеопотока. Алгоритм стабилизации включает в себя метод обнаружения смещения для текущего кадра и функцию сдвига/поворота изображения. Основной подход к определению смещения бу-

дет основываться на корреляционном методе, который можно описать следующим образом [1, 2]:

- выделяют центральную часть опорного изображения. Величина отступа определяется максимальным смещением, которое желаем определить. Центральная часть опорного изображения не должна быть слишком маленькой, иначе для функции расчета корреляции не хватит данных для стабильной работы;
- на текущем кадре выбирают часть изображения такого же размера, смещенную относительно центра изображения;
- для каждого смещения рассчитывают расстояние, описывающее корреляцию центральной части опорного и текущего изображений.

В качестве примера рассмотрим смещения, представленные на рис. 1.

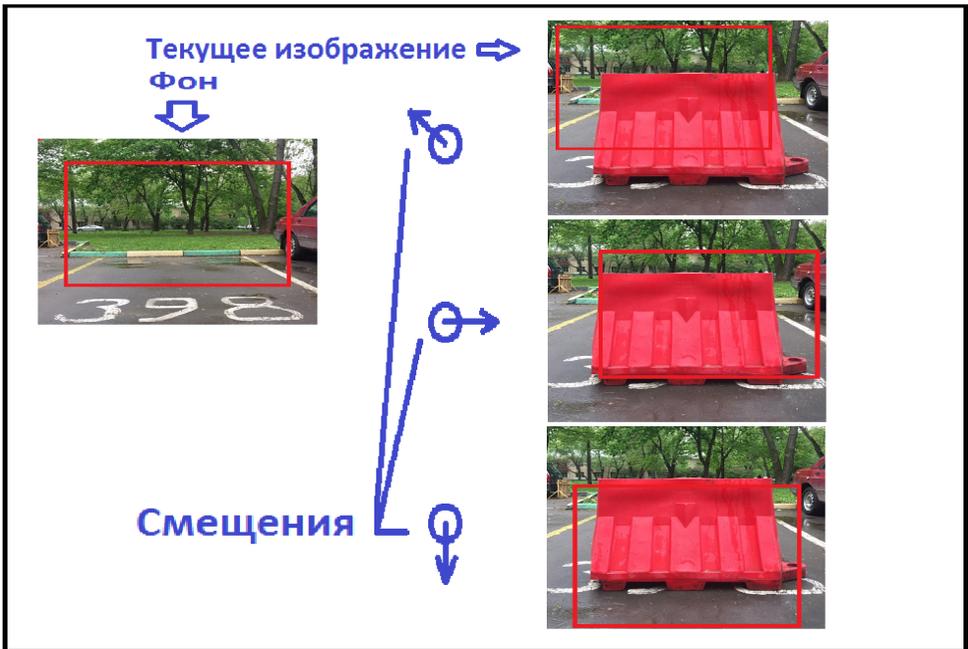


Рис. 1. Смещения текущего кадра относительно фона

При применении такого подхода без дополнительных методов скорость работы алгоритма будет крайне низкой даже при высокой скорости корреляционной функции, поскольку необходимо перебирать все варианты возможного смещения изображений относительно друг друга. Решить эту проблему можно путем использования пирамидальной структуры данных [3], что позволит:

- сократить время обработки изображения;
- определить более точные начальные приближения для обработки нижних уровней на основе результатов обработки верхних;
- находить локальный максимум корреляции для максимального масштаба, а затем для меньших масштабов.

Пирамидальная структура изображений представлена на рис. 2.



Рис. 2. Пирамидальная структура изображений

Выбор метода отслеживания объекта в видеопотоке. В качестве метода детектирования и отслеживания объекта в видеопотоке был выбран алгоритм Лукаса—Канаде [4].

Суть данного метода состоит в том, что отслеживается не фрагмент изображения, а конкретный пиксель (набор пикселей). При этом делается допущение, что значения пикселей переходят из одного кадра в другой без изменений. Может показаться, что такое допущение маловероятно, так как от кадра к кадру глобальное освещение сопровождаемого объекта может сильно меняться, но на практике получается, что этот алгоритм работает достаточно хорошо.

В данной работе алгоритм уместен, так как предполагается, что сопровождаемый объект находится в помещении и на его освещение ничего не влияет. Поскольку данный проект является прототипом, начальные эксперименты также будут проводиться в помещении с удовлетворительным освещением.

Тестирование программного модуля отслеживания объекта. Отслеживание объекта реализуем таким образом [5]:

- 1) выбираем объект на изображении, который будем отслеживать;
- 2) наводим курсор на объект и кликаем, появляется зеленая точка;
- 3) наблюдаем, что при перемещении объекта в видеопотоке точка двигается вместе с объектом.

Тестировать программу предлагается следующим образом:

- 1) вывести на экран отсчет времени, чтобы фиксировать положение объекта в определенный момент времени t ;
- 2) выделить объект;
- 3) перемещать объект.

Для фиксации отдельных кадров был записан видеофайл [6]. На рисунке 3 представлен захват фрагмента шариковой ручки камерой в момент времени $t = 3,3$ с.

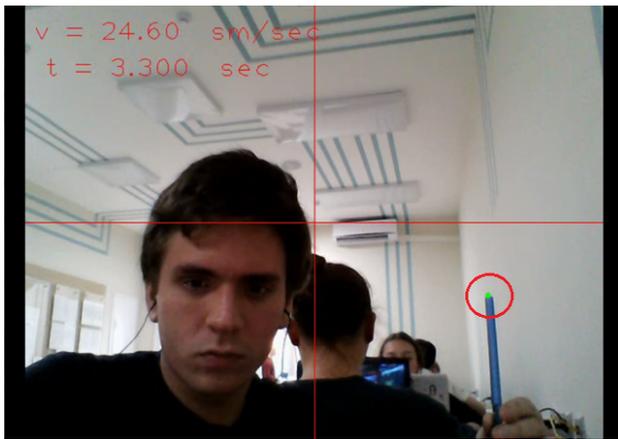
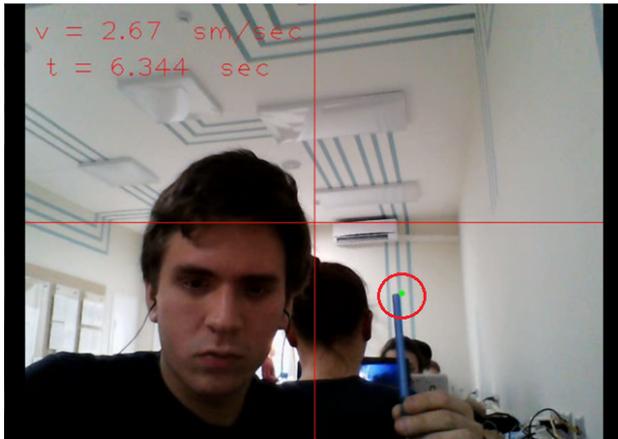
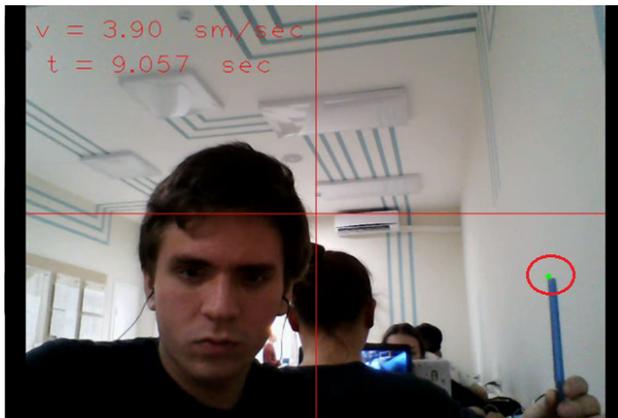


Рис. 3. Захват объекта видеокамерой

Перемещаем объект по оси абсцисс влево (рис. 4, а) и вправо (рис. 4, б) [7]. Зеленая точка на конце ручки означает, что объект захвачен видеокамерой.



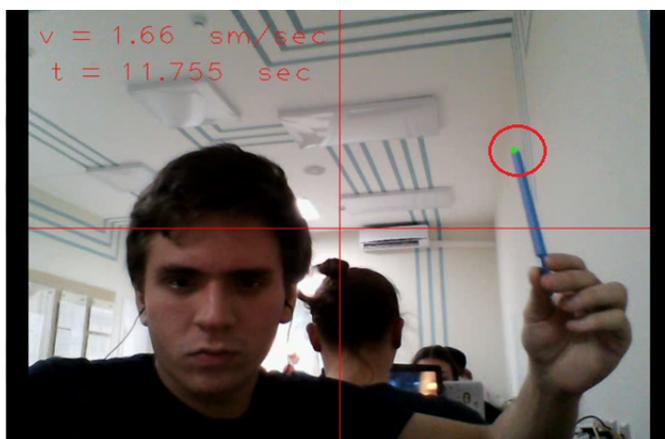
а



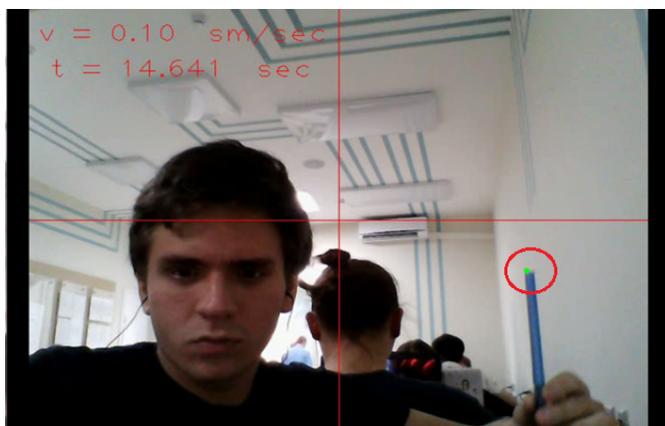
б

Рис. 4. Перемещение объекта влево (а) и вправо (б)

Далее перемещаем объект по оси ординат вверх (рис. 5, а) и вниз (рис. 5, б)



а



б

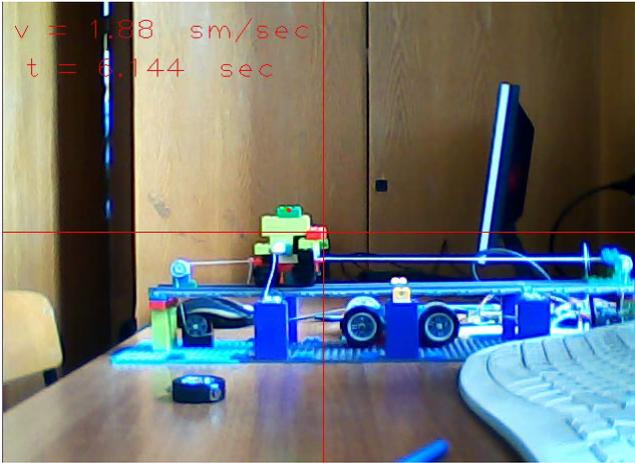
Рис. 5. Перемещение объекта вверх (а) и вниз (б)

В ходе тестирования программного модуля замечено, что точка следует за объектом по траектории перемещения объекта слежения. Программный модуль отслеживания объекта работает корректно [8].

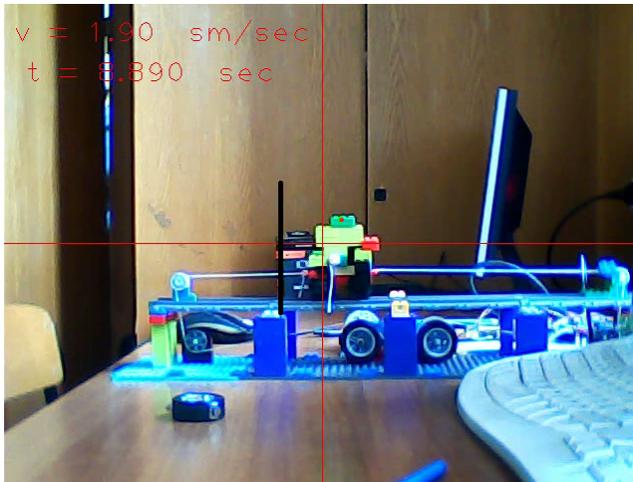
Тестирование программного модуля оценки скорости. Для подтверждения корректности работы программы будем измерять скорость перемещения объекта на экспериментальной установке. На экран будет выводиться изображение со значением скорости движения вагонетки V и отсчет времени t_c момента запуска программы. Проверка работы программы строится следующим образом:

- 1) выбор кадра в момент времени t_1 и t_2 ;
- 2) расчет координат точки отслеживания в момент времени t_1 и t_2 ;
- 3) расчет смещения точки отслеживания;
- 4) определение средней скорости V_{cp} и сравнение со значением V .

Первый кадр в момент времени t_1 представлен на рис. 6, а.



а



б

Рис. 6. Первый кадр в момент времени t_1 (а) и кадр в момент времени t_2 (б)

Фиксируем момент времени $t_1 = 6,144$ с. Точка отслеживания P отмечена красным цветом на зеленой крышке вагонетки (то же и для t_2). Значение координаты P по оси абсцисс $P_{x1} = 560$ пкс. Затем выводим второй кадр в момент времени t_2 (рис. 6, б). Отмечаем момент времени $t_2 = 8,89$ с. Значение координаты P по оси абсцисс $P_{x2} = 620$ пкс. Далее рассчитываем смещение и время:

$$\Delta P = P_{x2} - P_{x1} = 60 \text{ пкс}; \quad (1)$$

$$\Delta t = t_2 - t_1 = 2,746 \text{ с}, \quad (2)$$

где ΔP — смещение за время прошедшее с момента t_1 до t_2 ; Δt — время с момента t_1 до момента t_2 . Рассчитав смещение и время, за которое это смещение произошло, определим среднюю скорость:

$$V_{\text{cp}} = \frac{\Delta P}{\Delta t} = 21,8 \text{ пкс/с.} \quad (3)$$

Учитывая параметры калибровки перевода пикселей в сантиметры, рассчитаем V_{cp} в см/с. По данным калибровки $C = 10,5$ пкс/см, где C — количество пикселей на один сантиметр, следовательно

$$V_{\text{cp}} = \frac{21,8 \text{ пкс/с}}{C} \approx 2 \text{ см/с.} \quad (4)$$

Скорость передвижения V вагонетки, которая выводится на экран, составляет $1,88 \leq V \leq 2$, что совпадает со значением V_{cp} , рассчитанным опытным путем. Таким образом, можно утверждать, что программа оценки скорости движения объекта работает корректно. На точность измерения скорости влияет метод, которым будет производиться калибровка и масштабирование в программном обеспечении перед началом работы программы [9].

Масштабирование без учета параметров камеры. Когда параметры камеры не известны, производить масштабирование можно опытным путем. Для этого сначала надо рассчитать расстояние от камеры до объекта, размер объекта слежения, количество пикселей, которое занимает объект слежения на изображении, и перевести все эти значения в удобные единицы измерения. Разным разрешениям камеры соответствуют разные значения пикселей, которые занимает объект на изображении. Камера, для которой будет производиться калибровка, имеет разрешение 640×480 пикселей [10].

Первым шагом является размещение камеры напротив экспериментальной установки, которая состоит из платформы и вагонетки, перемещающейся по платформе (рис. 7). Для удобства измерений на изображении, которое выводится с камеры на экран наблюдения, красными линиями отмечен центр изображения.

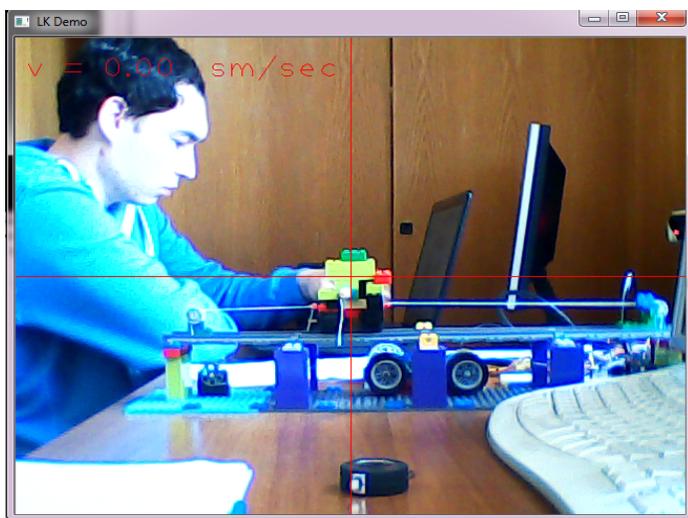


Рис. 7. Разработанная установка до начала эксперимента

Вагонетка находится в центре изображения, камера выравнена по горизонту относительно платформы. Расстояние S от камеры до вагонетки — 92 см. Используя редактор изображений PowerPoint, произведем расчет размера вагонетки в пикселях. Измерив разность значений пикселей по оси абсцисс левой L и правой R сторон вагонетки (рис. 8), получим размер вагонетки в пикселях.

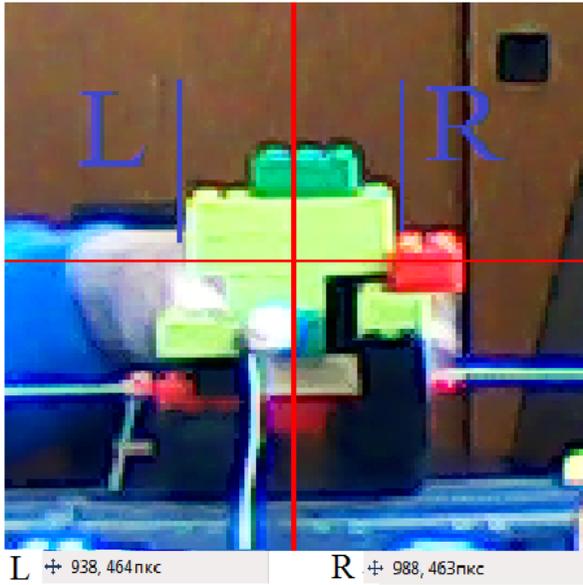


Рис. 8. Значения пикселей левой L и правой R частей вагонетки

Далее рассчитаем размер вагонетки:

$$N_{px} = R - L; \quad (5)$$

где N_{px} — размер вагонетки в пикселях; R — значение координаты пикселя по оси абсцисс правой части вагонетки; L — значение координаты пикселя по оси абсцисс левой части вагонетки.

Учитывая значения $R = 988$ и $L = 938$, по формуле (5) находим размер вагонетки N_{px} равный 50 пикселей. Затем рассчитаем реальный размер N_{real} вагонетки. Размер вагонетки N_{real} составляет 5 см. Теперь определим количество пикселей, приходящихся на один сантиметр:

$$C = \frac{N_{px}}{N_{real}}. \quad (6)$$

Согласно формуле (6), $C = 10$ пикселей на сантиметр. На основании результатов расчетов масштабирования можно оценить, сколько сантиметров в секунду проезжает вагонетка. Эти значения можно использовать в качестве опорных. Зная их, можно определять масштабы объектов на разных расстояниях. Используя экспериментальные данные, выведем формулу зависимости C от S .

Эксперимент определения замера и расчета масштаба объекта строится так же, как и предыдущий. Эталонный объект в ширину имеет размер 45 см.

Используем следующие экспериментальные данные: a — эталонный размер объекта, который будет опытным образцом в экспериментах № 1, 2 и 3, $a = 45$ см; S_1, S_2, S_3 — расстояние от камеры до объекта наблюдения; b_1, b_2, b_3 — размер эталонного объекта в пикселях; C_1, C_2, C_3 — число пикселей на сантиметр, приходящихся на определенное расстояние S .

Данные для эксперимента № 1: $S_1 = 55$ см; $b_1 = 470$ пкс; $C_1 = 10,4$ пкс/см.

Данные для эксперимента № 2: $S_2 = 93$ см; $b_2 = 281$ пкс; $C_2 = 6,244$ пкс/см.

Данные для эксперимента № 3: $S_3 = 118$ см; $b_3 = 223$ пкс; $C_3 = 4,95$ пкс/см.

Используя экспериментальные данные, рассчитаем зависимость C от S по формулам (6)–(12):

$$K_{C1} = \frac{C_1}{C_2} = 1,6655; \quad (7)$$

$$K_{C2} = \frac{C_2}{C_3} = 1,2614; \quad (8)$$

$$K_{C3} = \frac{C_1}{C_3} = 2,101; \quad (9)$$

$$K_{S1} = \frac{S_2}{S_1} = 1,69; \quad (10)$$

$$K_{S2} = \frac{S_3}{S_2} = 1,261; \quad (11)$$

$$K_{S3} = \frac{S_3}{S_1} = 2,14. \quad (12)$$

Сравним значения K_{C1} и K_{S1} , K_{C2} и K_{S2} , K_{C3} и K_{S3} и обнаружим совпадение до первого знака после запятой. Следовательно, эти значения почти равны, также прослеживается закономерность уменьшения C при увеличении S .

На основании примерного равенства значений K_C и K_S запишем

$$\frac{C_1}{C_2} = \frac{S_2}{S_1}. \quad (13)$$

Из выражения (11) получим формулу расчета масштабирования объекта:

$$C_2 = \frac{S_1 C_1}{S_2}, \quad (14)$$

где C_2 — значение масштаба объекта на изображении, пкс/см; C_1 — известная величина масштаба, полученная экспериментально на этапе калибровки; S_1 — известное расстояние, полученное на этапе калибровки; S_2 — новое расстояние

до объекта слежения, которое пользователь вводит вручную или которое автоматически передается с дальногомера.

Выводы. В заключение отметим, что после проведения калибровки, можно рассчитывать скорость объекта на разных расстояниях. Однако данный метод масштабирования недостаточно точен, в отличие от метода калибровки с использованием технических характеристик камеры. На основании результатов тестирования и оценки точности измерений, можно сделать вывод, что оценка скорости движения объекта в видеопотоке является корректной. Разработанный алгоритм позволяет определить скорость перемещения с некоторой погрешностью.

Литература

- [1] Красильщиков М.Н., Себрякова Г.Г., ред. Современные информационные технологии в задачах навигации и наведения беспилотных маневренных летательных аппаратов. Москва, Физматлит, 2009, 556 с.
- [2] Сакович И.О., Белов Ю.С. Применение метода Лукаса—Канаде для вычисления оптического потока. *Инженерный журнал: наука и инновации*. 2014, № 7. URL: <http://engjournal.ru/catalog/pribor/optica/1275.html> DOI: 10.18698/2308-6033-2014-7-1275.
- [3] Шилдт Г. С++ для начинающих. Шаг за шагом. Москва, Вильямс, 2011, 639 с.
- [4] Обухова Н.А. Обнаружение и сопровождение движущихся объектов методом сопоставления блоков. *Информационно управляющие системы*. 2004, № 1, с. 30–37.
- [5] Цифровая стабилизация изображения со стационарных камер — корреляционный подход. URL: <https://habrahabr.ru/post/219815/> (дата обращения 23.10.2017).
- [6] Bradski G., Kaehler A. Learning OpenCV. O'Reilly Media, 2008, pp. 513–516.
- [7] Кустикова В.Д. Отслеживание движения и алгоритмы сопровождения ключевых точек. Нижний Новгород, Нижегородский государственный университет им. Н.И. Лобачевского, 2013, 35 с.
- [8] Мураховский В., Симонович С. Большая книга цифровой фотографии. Санкт-Петербург, Питер, 2006, с. 11–15.
- [9] Визильтер Ю.В., Желтов С.Ю., Князь В.А., Ходарев А.Н., Моржин А.В. Обработка и анализ цифровых изображений с примерами на LabVIEW IMAQ Vision. Москва, ДМК Пресс, 2008, 464 с.
- [10] Коплович Е.А. Разработка алгоритмов стабилизации и компрессии изображений для систем видеонаблюдения мобильных робототехнических комплексов. Дис. ... канд. техн. наук. Москва, МИЭТ, 2008, 146 с.

Левинский Адам Тагирович — магистрант кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Селюто Наталия Максимовна — магистрант кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Родионов Илья Дмитриевич — магистрант кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

MOTION TRACKING AND ESTIMATION OF OBJECT SPEED IN VIDEO STREAM

A.T. Levinskiy

adam.levinskiy@yandex.ru

N.M. Selyuto

selyuto@icloud.com

I.D. Rodionov

ir@drp.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Keywords

The article is devoted to the development of object speed evaluation method from a moving camera video stream. The paper considers methods and algorithms for specified task solution, as well as methods for video stream stabilizing and tracking of moving objects in video stream. The article provides an overview of similar tracking systems and develops an object speed evaluation method from video stream. The software implementation for all selected and developed methods is produced. The software allowing to track a two-dimensional plane speed is developed.

Recognition, error, interface, testing, Levenshtein method, stabilization, object motion tracking, object movement

© Bauman Moscow State Technical University, 2017

References

- [1] Krasil'shchikov M.N., Sebyrakova G.G., ed. *Sovremennyye informatsionnyye tekhnologii v zadachakh navigatsii i navedeniya bespilotnykh manevrennykh letatel'nykh apparatov* [Modern information technologies in navigation and guidance problems of unmanned manoeuvring aircraft]. Moscow, Fizmatlit publ., 2009, 556 p.
- [2] Sakovich I.O., Belov Yu.S. Application of Lucas—Kanade method for computing the optical flow. *Inzhenernyy zhurnal: nauka i innovatsii* [Engineering Journal: Science and Innovation], 2014, no. 7. Available at: <http://engjournal.ru/catalog/pribor/optica/1275.html> DOI: 10.18698/2308-6033-2014-7-1275.
- [3] Shildt G. *S++ dlya nachinayushchikh. Shag za shagom* [C++ for beginners. Step by step]. Moscow, Vil'yams publ., 2011, 639 p.
- [4] Obukhova N.A. Detection and tracking moving object based on block matching algorithm. *Informatsionno upravlyayushchie sistemy* [Information and Control Systems], 2004, no. 1, pp. 30–37.
- [5] Tsifrovaya stabilizatsiya izobrazheniya so statsionarnykh kamer — korrelyatsionnyy podkhod [Image digital stabilization from stationary cameras — correlation approach]. Available at: <https://habrahabr.ru/post/219815/> (accessed 23 October 2017).
- [6] Bradski G., Kaehler A. *Learning OpenCV*. O'Reilly Media, 2008, pp. 513–516.
- [7] Kustikova V.D. Otslezhivanie dvizheniya i algoritmy soprovozhdeniya klyuchevykh toчек [Motion tracking and key points tracking algorithm]. Nijniy Novgorod, UNN publ., 2013, 35 p.
- [8] Murakhovskiy V., Simonovich S. *Bol'shaya kniga tsifrovoy fotografii* [Big book of digital photography]. Sankt-Petersburg, Piter publ., 2006, pp. 11–15.
- [9] Vizil'ter Yu.V., Zheltov S.Yu., Knyaz' V.A., Khodarev A.N., Morzhin A.V. Obrabotka i analiz tsifrovyykh izobrazheniy s primerami na LabVIEW i IMAQ Vision. [Digital image pro-

cessing and analysis with LabVIEW IMAQ and Vision examples]. Moscow, DMK Press publ., 2008, 464 p.

- [10] Koplovich E.A. Razrabotka algoritmov stabilizatsii i kompressii izobrazheniy dlya sistem videonablyudeniya mobil'nykh robototekhnicheskikh kompleksov. Dis. kand. tekhn. nauk [Developing image stabilization and compression algorithms for video observation of mobile robotic systems. Kand. tech. sci. diss.]. Moscow, MIET publ., 2008. 146 p.

Levinskiy A.T. — Master's Degree student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Celyuto N.M. — Master's Degree student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Rodionov I.D. — Master's Degree student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.