

УДК 004.93'1.021

К. Ю. ДЕРГАЧЁВ, Л. А. КРАСНОВ, Е. В. ПЯВКА

*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина***АЛГОРИТМЫ ОБНАРУЖЕНИЯ ОБЪЕКТОВ И ОЦЕНКИ ПАРАМЕТРОВ
ИХ ДВИЖЕНИЯ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ**

Предложены алгоритмы обнаружения и оценки параметров движения объектов по результатам видеонаблюдения, работа которых основана на анализе компонент цветового пространства HSV с последующей бинаризацией изображения с отсечением по порогу яркости и вычислением моментов бинарных изображений для точной локализации объекта наблюдения в кадре. Реализована простая и точная процедура оценки траекторных параметров (дальности, угловых и скоростных параметров) объекта. Предусмотрены меры по повышению порога устойчивой работы алгоритмов за счет использования интерактивного интерфейса. Результаты проведенных исследований могут быть полезны при проектировании систем технического зрения различного назначения.

Ключевые слова: *цветовое пространство HSV, выделение компонент цветового пространства, бинаризация изображения с отсечением по порогу яркости, вычисление моментов бинарных изображений, измерение траекторных параметров, пороги устойчивой работы алгоритмов.*

Введение

При проектировании современных систем технического зрения, предназначенных для транспорта, робототехники, беспилотных летательных аппаратов и ряда других актуальных приложений, разработчик всегда находится в условиях достаточно жестких ограничений. Во-первых, необходимо использовать простую, надежную и достаточно дешевую аппаратную платформу и широко распространенные стандартные средства программирования. При этом нужно обеспечить высокую точность измерения информационных параметров и большое быстродействие (обычно системы технического зрения должны функционировать в реальном масштабе времени).

Большинство современных систем связаны с обработкой и использованием видеoinформации с регистраторов, размещенных на мобильных платформах. В этих случаях приходится решать задачи управления движением в рамках указанных ограничений. Такие проекты ориентированы на использование одноплатных компьютеров Raspberry Pi и языка программирования Python в комбинации с библиотекой обработки изображений OpenCV (Open Source Computer Vision Library). Связка Python с OpenCV позволяет проводить обработку видеоданных в реальном времени с применением современных алгоритмов.

Неизменно актуальной проблемой технического зрения является эффективное решение задачи

обнаружения объектов и оценки параметров их движения с высокой степенью точности в реальном масштабе времени на базе относительно простых аппаратных и программных средств.

В задаче обнаружения и оценки параметров движения объекта, необходимо успешно решить три основных вопроса:

- надежно обнаружить наблюдаемый объект при различных условиях освещения сцены на фоне возможных помех;
- оценить его пространственную локализацию в кадре с точностью до пикселя;
- по видеопоследовательности рассчитать параметры траектории (дальность до начала координат, угловое положение и параметры скорости движения).

Традиционно решение этих задач осуществляется на базе алгоритмов анализа оптического потока (по Лукасу-Канаде), оценки движения объектов по методу сопоставления блоков и использования корреляционно-экстремальных алгоритмов [1 – 3]. Эти методы отличает недостаточная точность, большой объем вычислений, а, следовательно, и невысокое быстродействие. Такие ограничения не позволяют эффективно использовать эти алгоритмы для работы в мобильных системах технического зрения.

1. Постановка задачи

Целью данной работы является создание эффективных алгоритмов обнаружения объектов и оценки параметров их движения по результатам ви-

деонаблюдения, предназначенных для работы в реальном масштабе времени на базе компьютера Raspberry Pi с помощью языка программирования Python и библиотеки OpenCV. Для обнаружения и позиционирования положения наблюдаемых объектов в кадрах видеопоследовательности необходимо использовать их цветовые характеристики.

В соответствии с поставленной задачей для реализации алгоритмов был выбран наиболее существенный информационный признак – нужная компонента в соответствующей цветовой модели кадров регистрируемой видеопоследовательности, которая наилучшим образом соответствует цвету наблюдаемого объекта и надежно отличает его от фона сцены при различных условиях освещения.

Для точной пространственной локализации объекта наблюдения далее была использована процедура преобразования выделенной цветовой компоненты кадра в полутоновое изображение и последующая его бинаризация с отсечением по порогу яркости. Затем для бинарного изображения вычислялся набор инвариантных моментов нулевого ($m00$) и первого ($m01$ и $m10$) порядка, определяющих координаты центра наблюдаемого объекта.

На заключительном этапе обработки видеоданных осуществлялось преобразование системы координат изображения в правую центральную прямоугольную систему координат, и проводилась процедура вычисления траекторных параметров наблюдаемого объекта.

Существенное внимание было уделено построению интерактивного интерфейса программы и графическому отображению результатов обработки данных.

2. Методы синтеза программного кода

Одной из особенностей написания кодов на языке Python является необходимость формирования нужных возможностей программирования за счет подключения внешних библиотек.

Перечень основных ресурсов с учётом библиотеки OpenCV для решения нашей задачи невелик и приведен далее:

```
import sys, import cv2, import numpy as np, import math.
```

Видно, что такая конфигурация средств программирования не оказывает существенной нагрузки на процессор Raspberry Pi и создает хорошие предпосылки для обработки данных в реальном времени. Для увеличения быстродействия при обработке видео все основные процедуры обычно реализуются с помощью стандартных функций OpenCV, оптимизированных разработчиками библиотек ещё на этапе их создания.

Рассмотрим более внимательно некоторые подробности заложенных в разработке технических решений.

Для обнаружения и надежного детектирования объекта наблюдения в кадре была использована методика его выделения по цветовому признаку. Цветовое пространство является моделью представления данных, основанной на использовании определенных цветовых координат. Обычно сигналы с web-камеры являются последовательностью изображений, представленных в модели RGB, использование которой малопродуктивно, так как поиск областей нужного цвета предполагает анализ всех трех составляющих R, G и B. Для поиска на изображениях объектов по цвету и яркости, как правило, применяется цветовая модель HSV, где Hue, Saturation и Value – параметры, определяющие цветовой тон, насыщенность и яркость. Они показаны на диаграмме (рис. 1) и нормированы следующим образом:

- Hue – 0 – 360°;
- Saturation – (0 – 100)%;
- Value – (0 – 100)%.

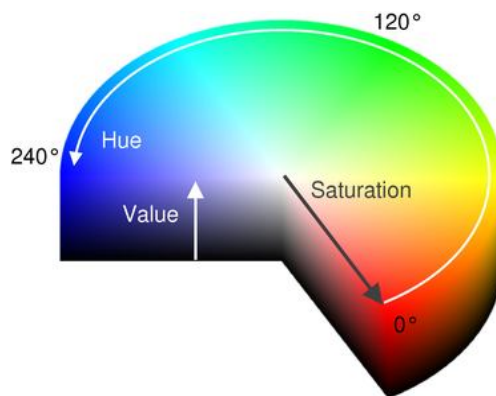


Рис. 1. Диаграмма цветовой модели HSV

Поэтому наиболее важной процедурой после ввода видеоданных было преобразование цветового пространства RGB в пространство HSV с помощью следующей функции OpenCV:

```
hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
```

Существенное влияние на качество обнаружения оказывает выбор компонент, соответствующих цвету наблюдаемого объекта при различных условиях освещения сцены. Любая точка на шкале H определяет двухмерную область с разными значениями S и V. Но для выбранного значения H целевая область в этом двухмерном пространстве надежнее определяется условием: $V > V_{min}$ и $S > S_{min}$, где V_{min} , S_{min} – некоторые константы. Поэтому принято выбирать целевой диапазон на шкале H, указывая два значения – H_{min} и H_{max} [4 – 8]. Так, например, диапазон значений параметров HSV для выделения синего цвета обычно лежит в пределах

```
low_range = np.array([100,100,100])
up_range = np.array([140,255,255]),
```

а диапазон для красного цвета

```
lower_range = np.array([0,50,100])
up_range = np.array([10,255,255]).
```

Следует отметить, что у такого подхода есть и существенный недостаток – надежно выделяются только яркие цветовые компоненты из-за ограничений на основе значений V_{min} и S_{min} . Поэтому практически невозможно выделить области для темных цветов (например, коричневого), а также для градаций в оттенках серого. Поэтому для успешного поиска и слежения за объектами на основе цветовых характеристик желательно, чтобы наблюдаемый объект имел маркеры ярких насыщенных цветов (красного, синего или зеленого).

Отметим, что процедуру выбора рабочего диапазона параметров пространства HSV для наблюдаемого объекта лучше осуществлять интерактивно в зависимости от текущих условий освещения и цвета самого объекта. Такой подход был реализован в рассматриваемом далее алгоритме с помощью элементов настройки (движков) интерфейса (рис. 2).

После выбора диапазона нужных цветовых компонент (например, синего цвета) производилась фильтрация (настройка цветовой маски) с помощью пороговой функции

```
mask_blue = cv2.inRange(hsv,
low_range, up_range).
```

После цветовой фильтрации изображение кадра представляется в оттенках серого цвета (диапазон значений яркости пикселей от 0 до 255). Такая процедура позволяет надежно детектировать объект по его цвету, однако не позволяет полностью подавить помехи фона, содержащие похожие цветовые компоненты. Эту трудность можно легко преодолеть путем бинаризации с отсечением по порогу яркости отфильтрованного по цвету изображения. Для этого использовалась функция

```
mask_blue = cv2.threshold(mask_blue, p,
255, cv2.THRESH_BINARY)[1].
```

Следует обратить внимание на выбор порога отсечения по яркости p в этой функции. Он зависит от уровня помех, и его желательно устанавливать интерактивно.

Следующим важным шагом при построении алгоритма траекторных измерений является определение координат объекта наблюдения в текущем кадре. При работе с бинарными изображениями наиболее продуктивным способом этого решения является вычисление моментных инвариантов такого изображения [9, 10], которые позволяют определить координаты с высокой степенью точности.

Моменты изображения вычисляются с помощью функции

```
moments = cv2.moments(thresh, 1).
```

Функция `moments` возвращает массив моментов вплоть до третьего порядка. Однако для вычисления координат центра объекта требуются только моменты первого порядка $m01$ и $m10$, а также момент нулевого порядка $m00$. Они определяются следующим образом:

```
dM01 = moments['m01']
dM10 = moments['m10']
dArea = moments['m00'].
```

Напомним, что момент $m00$ – это количество всех единичных пикселей принадлежащих выделенному объекту, а моменты $m01$ и $m10$ – суммы X и Y координат этих пикселей. Понятно, что для определения координат центра наблюдаемого объекта необходимо выполнить нормировку этих моментов на момент нулевого порядка.

При проведении этой процедуры целесообразно выполнить дополнительную пороговую операцию, позволяющую отфильтровать ложные объекты, вероятность появления которых при работе цветowego фильтра не исключается. В этом случае при наличии априорной информации о размерах наблюдаемого объекта можно по условию $\text{if } dArea > N$, где N – количество единичных пикселей момента $m00$, устранить ложные объекты. В приведенном далее примере программа будет реагировать только на моменты, содержащие больше 100 пикселей

```
if dArea > 100:
    x = int(dM10 / dArea)
    y = int(dM01 / dArea).
```

В этом примере будут устранены случайные вспышки синего цвета, имеющие малую площадь засветки в кадре (меньшую площади наблюдаемого объекта).

Отметим, что функция `moment` не способна определять количество объектов в кадре. Поэтому, если в кадре окажется, например, два независимых объекта, то они интерпретируются как один общий. Следовательно, и координаты центра будут определены между ними. А это приведет к грубым аномальным ошибкам при вычислении траекторных параметров. В таких случаях необходимо прибегать к более точным методам обработки информации, например, к оценке формы объекта или другим процедурам морфологического анализа. Однако, это не принципиальный аспект решаемой задачи. Он выходит за рамки нашего рассмотрения и в данной работе обсуждаться не будет.

Далее необходимо определить координаты центра наблюдаемого объекта относительно центра

кадра видео. Поскольку начало координат кадра соответствует левому верхнему углу снимка, нужно методом параллельного переноса перейти к центральной прямоугольной системе координат с началом в точке, координаты которой равны половине размера кадра. Такой переход, обеспечивающий удобство расчета и отображения параметров траектории и вычисления нужных показателей (дальности, угловых координат и скорости), затруднений не вызывает.

3. Варианты построения алгоритмов

В рамках принятых методов был проведен синтез различных алгоритмов обнаружения и оценки параметров движения наблюдаемых объектов, работа которых основана на селекции цвета.

В качестве примера рассмотрим два разных варианта построения такой обработки видеоданных.

В первом случае алгоритм предназначен для использования в системах стационарного видеонаблюдения для обнаружения и оценки траекторных параметров наблюдаемых объектов. Он реализуется на языке Python с использованием библиотеки OpenCV на базе ОС Windows. Это позволяет осуществлять его эксплуатацию на обычном компьютере. В связи с этим много внимания было уделено разработке и созданию интерактивного интерфейса для предварительной настройки всех режимов обработки информации и удобства визуального контроля.

Рассмотрим подробнее работу с интерфейсом. На рис. 2 показан специализированный рабочий стол программы, содержащий главную инструментальную панель, с помощью которой производится настройка параметров системы и оперативное управление режимами работы.

Для начала работы необходимо активировать кнопку «Open» и в выпадающем окне «OpenVideoCapture» (рис. 3) выбрать источник исходной информации – можно использовать видеоданные Web или IP-камер, а также данные предварительно сохраненные в видеофайл. Во всех этих случаях ресурсы программы обеспечивают обработку информации в реальном масштабе времени.

При нажатии кнопки настроек «Settings» выпадает одноименная панель (рис. 4, а), содержащая четыре окна для просмотра видеоданных и инструментарий настройки параметров системы. В верхнем левом окне демонстрируется исходная видеопоследовательность или отдельный кадр, отображающие наблюдаемую сцену. Различные варианты выбора содержимого этого окна можно получить по нажатию кнопки «Play/Pause» на инструментальной панели. Для выделения интересующего Вас объекта

необходимо воспользоваться органами настройки цветового фильтра, результаты работы которого отображаются в правом верхнем окне панели настроек. Движки, управляемые с помощью мышки, позволяют выделить нужный диапазон параметров H, S и V цветового пространства HSV в интервале `low_range - up_range`. В нашем примере выделяется объект для наблюдения синего цвета. Заметим, что помимо интересующего нас объекта в кадре могут присутствовать и другие предметы синего цвета. Также цветовой фильтр может обнаружить фрагменты сцены, содержащие похожие цветовые компоненты, что обусловлено условиями освещения сцены. Полученные в результате цветовой фильтрации данные подлежат дальнейшей обработке.

Левое нижнее окно панели настроек «Settings» отображает процедуру бинаризации изображения с отсечением по порогу яркости. Это производится с помощью движка «Threshold» и помогает окончательно подавить фрагменты малой интенсивности отфильтрованного изображения (см. рис. 4, б).

В правом нижнем окне показана исходная видеопоследовательность или текущий кадр с цветной отметкой на обнаруженном объекте наблюдения (в нашем случае синего цвета). Для привлечения внимания пользователя протектированный объект наблюдения в этой программе маркируется красным цветом. Однако легко заметить, что в приведенном примере на сцене присутствуют два предмета синего цвета. Это создает неопределенность при оценке координат наблюдаемого объекта, а, следовательно, нужна дополнительная процедура разделения. Как уже отмечалось ранее, в качестве информационного признака для этого целесообразно принять априорные сведения о размерах наблюдаемого объекта. На рис. 4, б видно, что на инструментальной панели предусмотрено два окошка для ввода данных о площади наблюдаемого объекта в пикселях (`minArea` и `maxArea`). После их заполнения эффективно подавляется объект меньших размеров.

По окончании процедуры настройки параметров фильтрации цвета и пороговых операций по обнаружению объекта наблюдения необходимо по нажатию кнопки «Start» перейти в рабочий режим наблюдения за объектом, построения траектории его движения и расчета основных параметров движения (дальности до объекта, его угловых координат и скорости).

Инструментальная панель программы содержит два рабочих окна для отображения текущей видеопоследовательности и результатов траекторных измерений для соответствующего интервала времени. Внешний вид и органы управления этой панели показаны на рис. 5, б.

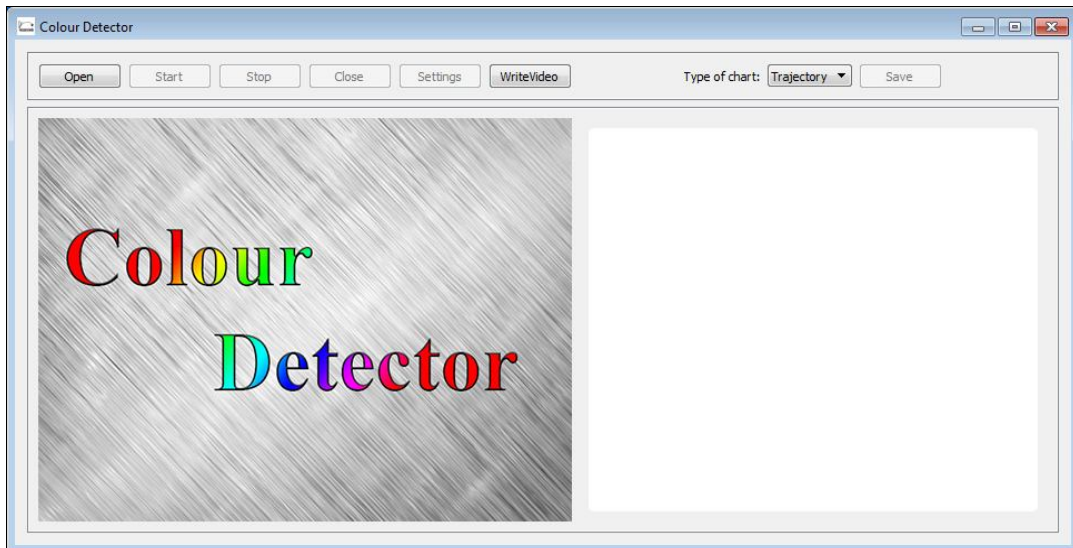


Рис. 2. Інтерфейс програми

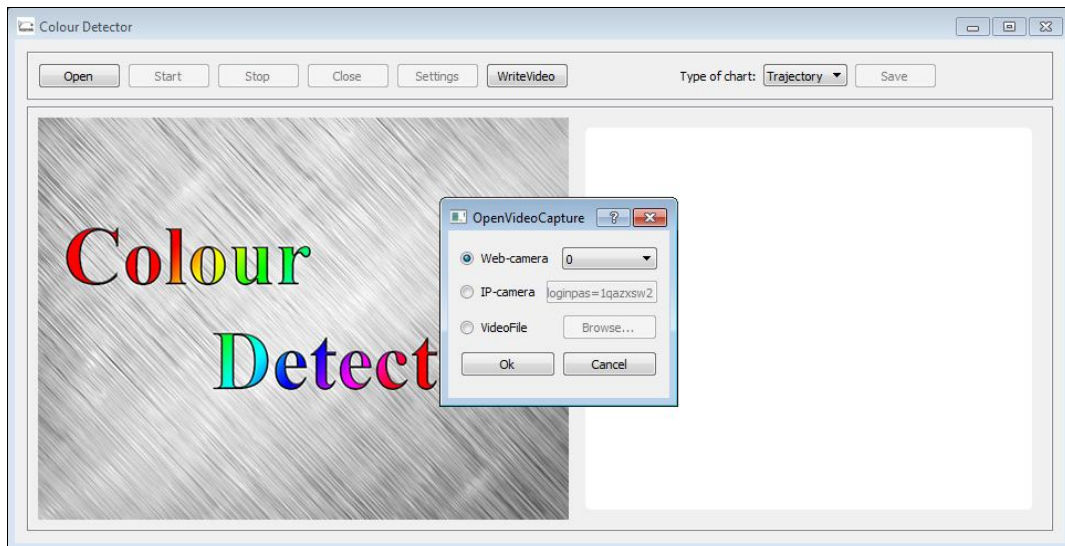
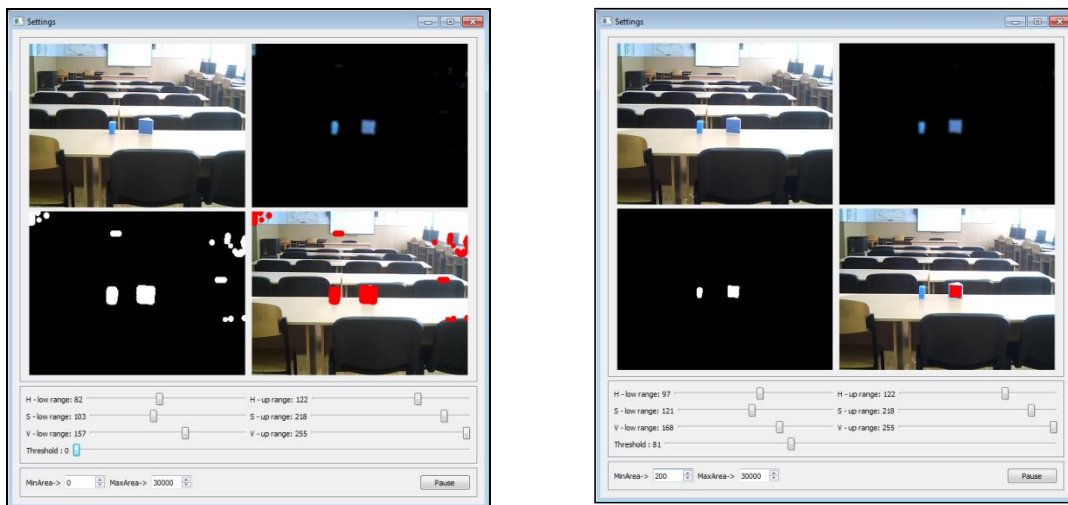


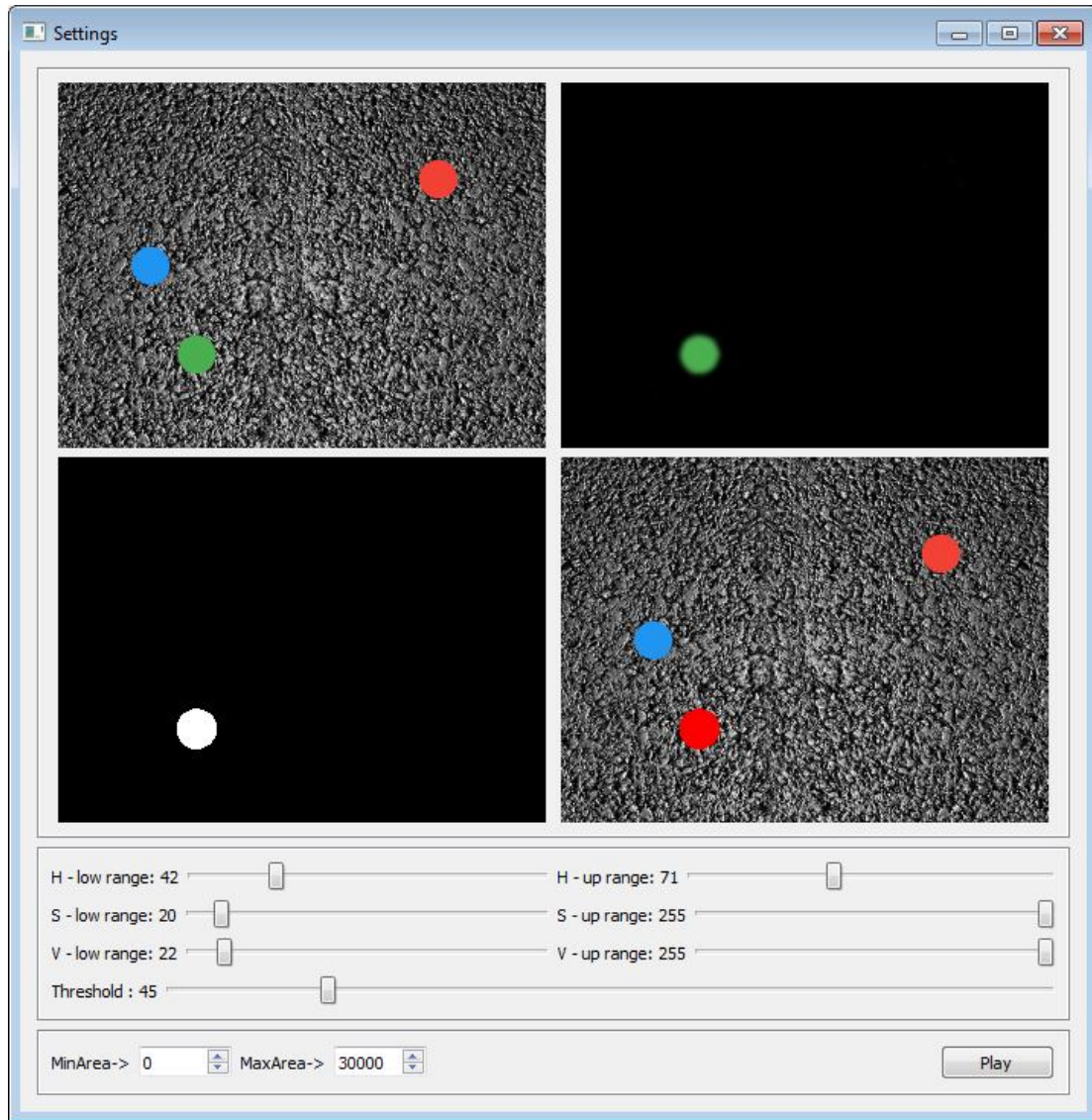
Рис. 3. Выбор источника видеоинформации



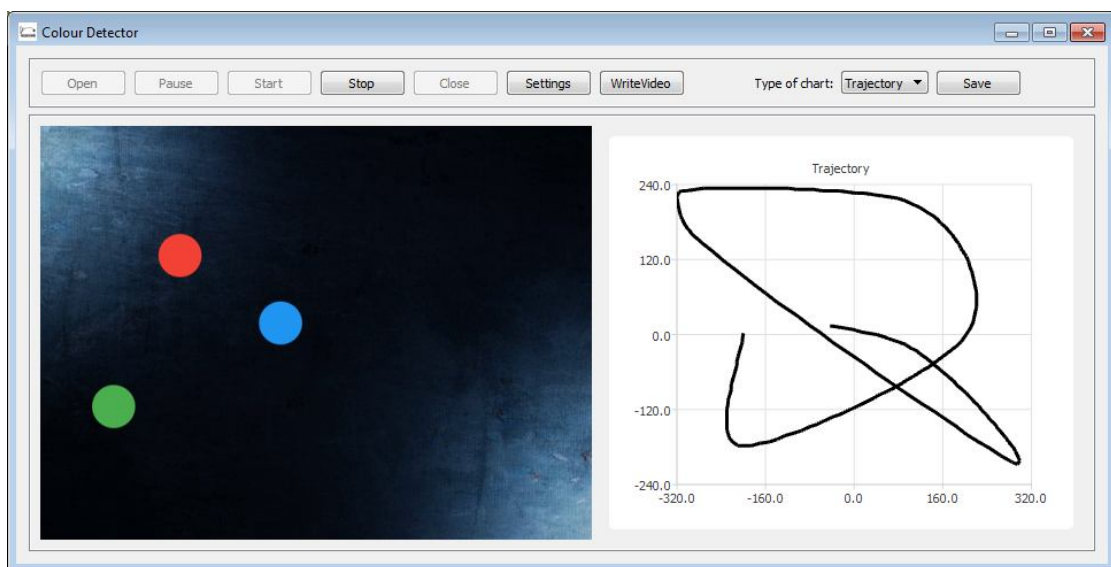
а

б

Рис. 4. Результаты обработки исходных видеоданных



а



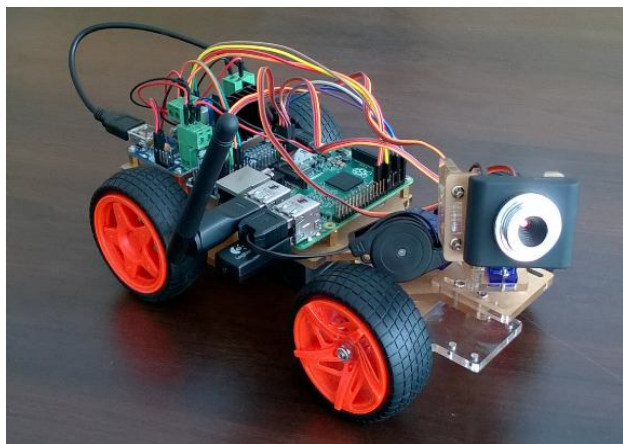
б

Рис. 5. Обработка тестовой видеопоследовательности

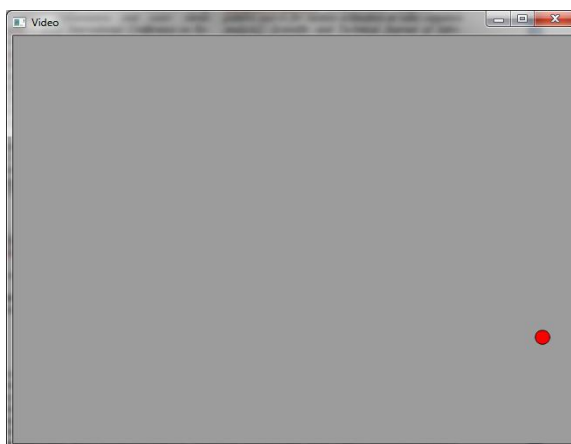
Приведем пример работы программы при анализе тестовой последовательности, когда на сцене хаотически движутся разноцветные шарики. Задача состоит в отслеживании траектории движения зеленого шарика. Настройки программы для решения этой задачи показаны на рис. 5, а. В левом окне (рис. 5, б) отображается исходная видеопоследовательность, а в правом окне в реальном масштабе времени производится отрисовка траектории движения нашего шара. Отметим, что по выбору пользователя в окошке «Type of chart» в правом окне можно отрисовывать тренды изменения угловых координат, дальности и скорости (в соответствии с пунктами выпадающего меню).

Рассмотрим еще один вариант построения алгоритма. Он был ориентирован на использование компьютера Raspberry Pi с операционной системой Raspbian, размещенного на мобильной платформе колесного робота (рис. 6, а). В этом случае основной является информация о движении робота относительно опорных точек сцены, обладающих опреде-

ленными цветовыми характеристиками. Покажем работу алгоритма на примере движения робота и съёмке с помощью его камеры опорной точки красного цвета. На рис. 6, б показан кадр текущей видеопоследовательности, а на рис. 6, в траектория взаимного перемещения. В нашем случае расчет параметров траектории проводился в центральной прямоугольной системе координат с началом в точке, координаты которой равны половине размера кадра. При расчете угловых координат была принята правая круговая система. Скачки на графике угловой координаты объекта связаны с переходом параметра из зоны нулевых углов в зону углов 360° . Эксперимент показал высокое качество выделения объекта на фоне помех и приемлемую точность оценки параметров траектории. Отметим большую информационную избыточность алгоритма. При скорости съемки 30 кадров в секунду для медленно движущихся объектов можно проводить обработку видео с прореживанием.



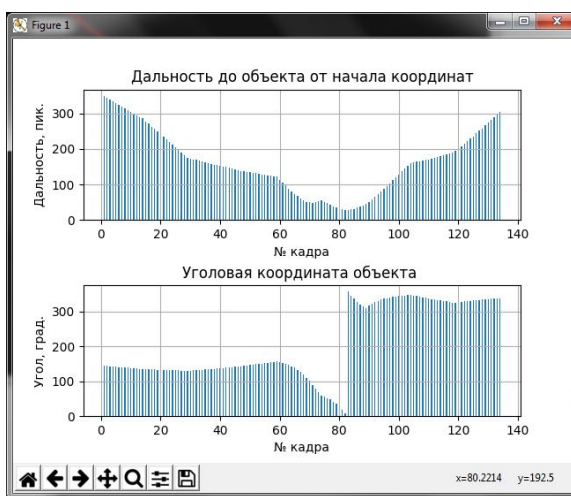
а



б



в



г

Рис. 6. Оценка параметров движения при размещении камеры на подвижном носителе

4. Результаты экспериментальных исследований

На рис. 7 показаны схема и результаты определения эффективности работы алгоритма оценки параметров движения колесного робота, на котором закреплен специальный круглый маркер синего цвета. Съемка проводилась на специальном полигоне. Камера закреплялась на фиксированной высоте над рабочей поверхностью и в течение записи её положение оставалось неизменным.

Из рис. 7, а (левое верхнее окно панели «Settings») видно, что съёмка проводилась фронтально по направлению сверху-вниз, поэтому введения поправок на угловое положение камеры не требовалось. Отметим, что никаких специальных условий освещения сцены не соблюдалось, а для создания достаточно сложной помеховой обстановки по поверхности полигона были хаотически распределены разноцветные маркеры различных размеров и формы.

Тем не менее, уже на этапе работы цветового фильтра было подавлено основное количество мешающих объектов, окрашенных в цвета отличные от синего. В кадре остались только метки, соответствующие предметам синего цвета. Затем с помощью пороговой процедуры сравнения площадей оставшихся объектов были подавлены более мелкие фрагменты помех и надёжно протектирован цветной маркер колёсного робота.

Рис. 7, б наглядно демонстрирует характер движения робота по поверхности полигона. Видно, что наличие большого количества помех (объектов различного цвета и формы в кадре) не влияет на точность построения траектории.

Кроме этого определялась степень влияния различных помех на качество оценки траекторных параметров, и выявлялись пороги устойчивой работы алгоритмов в зависимости от плотности действующих шумов. Характерные искажения формы траектории под действием гауссовых шумов приведены на рис. 8, б. В качестве критерия оценки влияния помех на качество определения информационных параметров использовалась величина среднеквадратической ошибки MSE (mean square error). Однако для наглядности представления результатов применён показатель среднеквадратического отклонения рассчитываемых параметров $SKO = \sqrt{MSE}$, отражающий степень разброса оценок в пикселях. При привязке кадра к реальным координатам достаточно умножить значение SKO на масштабирующий коэффициент, зависящий от условий проведения съёмки.

На рис. 9 приведены результаты оценки влияния гауссовых шумов различной интенсивности на качество оценки траекторных параметров (дальности наблюдаемого объекта от начала координат кадра и его углового положения). Также оценивалась степень разброса оценки координат объекта в кадре в зависимости от дисперсии гауссова шума σ .

Анализ показал, что наиболее чувствительной к действию шума является оценка параметра дальности объекта, а устойчивая работа алгоритма лежит в диапазоне дисперсий внешнего шума $\sigma = 0 - 0,45$. Дальнейшее повышение уровня шумов приводит к возникновению аномальных ошибок, и дисперсия оценок траекторных параметров существенно возрастает.

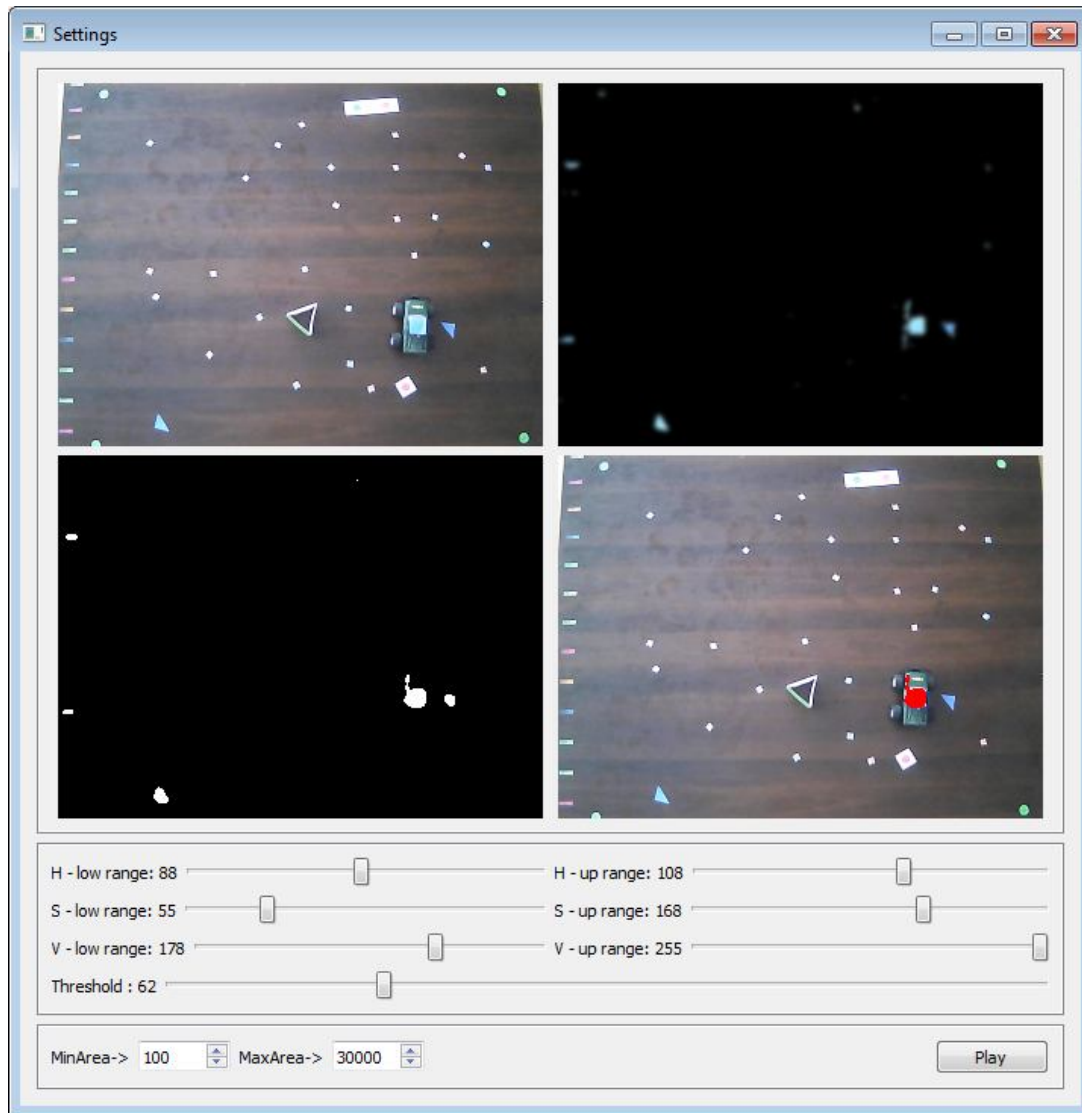
Заключение

Предложены алгоритмы оценки параметров движения по видеоданным на основе анализа компонент цветового пространства и даны варианты практической реализации этих методов. Проведена оптимизация структуры алгоритмов с использованием функций библиотеки OpenCV на языке программирования Python. Экспериментальные исследования показали достаточную устойчивость к действию различного рода помех и быстроедействие, позволяющее проводить обработку данных в реальном масштабе времени. Это в полной мере соответствует требованиям, сформулированным при постановке задачи на проектирование алгоритмов.

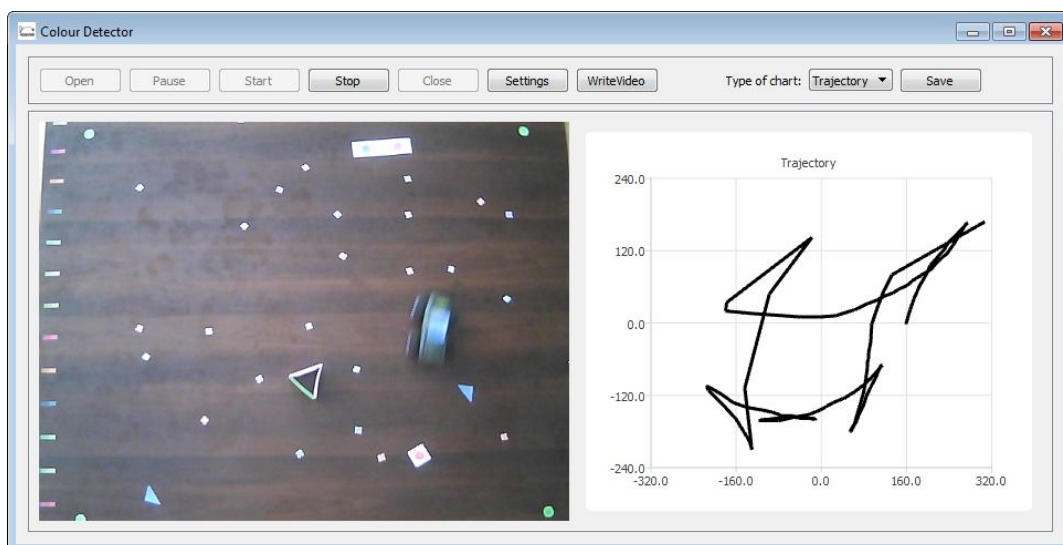
Приведённые данные могут быть полезны для адаптации алгоритмов и программного обеспечения оптических систем траекторных измерений к работе в реальном времени на подвижных носителях с использованием микрокомпьютеров Raspberry Pi.

Литература

1. *Определение координат и параметров движения объекта на основе обработки изображений [Текст] / Л. А. Мартынова, А. В. Корякин, К. В. Ланцов, В. В. Ланцов // Компьютерная оптика. – Самара, 2012. – Т. 36, № 2. – С. 266 – 273.*
2. *Lucas, B. An Iterative Image Registration Technique with an Application to Stereo Vision [Text] / B. Lucas, T. Kanade // 7th International Joint Conference on Artificial Intelligence (IJCAI): materials of the international seminar, 24-28 august, 1981. – Vancouver, 1981. – pp. 123-129.*
3. *Нгуен, В. Ч. Иерархический адаптивный алгоритм шаблонного поиска для оценки движения при анализе видеопоследовательности [Текст] / В. Ч. Нгуен, А. А. Тропченко // Научно-технический вестник информационных технологий, механики и оптики. – СПб, 2016. – Т. 16, № 3. – С. 474–481.*

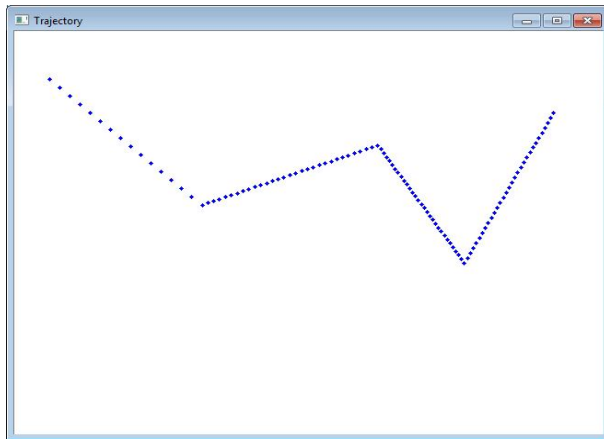


а

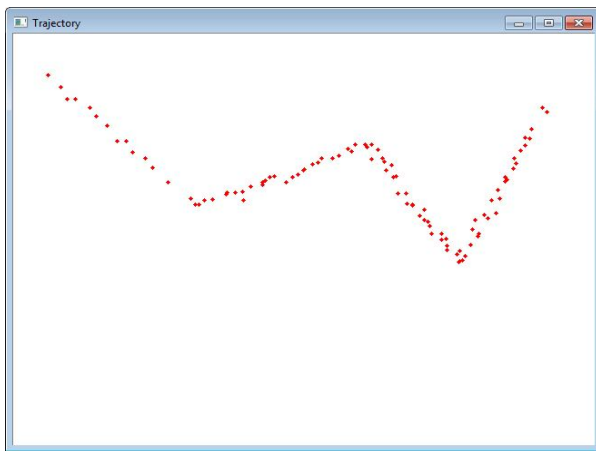


б

Рис. 7. Схема и результаты натурального эксперимента



а



б

Рис. 8. Влияние помех на качество траекторных измерений

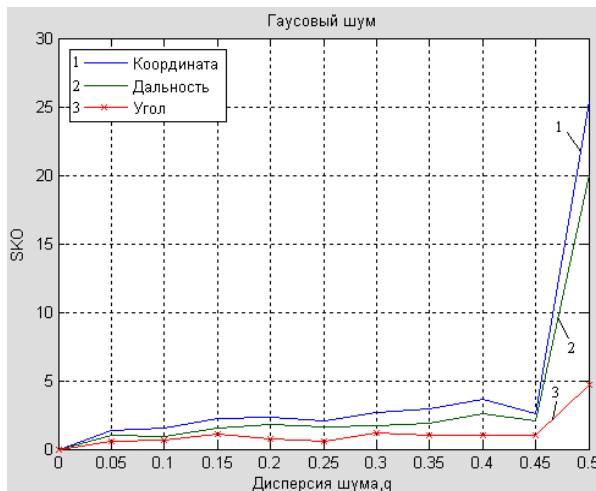


Рис. 9. SKO оценок координат (1), дальности (2) и углового положения (3) объекта при действии гауссовых шумов

4. Batavia, P. *Obstacle detection using adaptive color segmentation and color stereo homography* [Text] / P. Batavia // *IEEE International Conference on Robotics and Automation*. – 2001. – Vol. 1. – pp. 705-710.

5. Mojsilovic, A. *A computational model for color naming and describing color composition of images* [Text] / A. Mojsilovic // *IEEE Transactions on Image Processing*. – 2005. – Vol. 14. – pp. 690-699.

6. Barba-Guaman, L. *Detection of moving objects through color thresholding* [Detección de objetos en movimiento a través de la umbralización del color] [Text] / L. Barba-Guaman, C. Calderon-Cordova, P. A. Quezada-Sarmiento // *Iberian Conference on Information Systems and Technologies, CISTI, 2017*, article № 7975755.

7. *Object Tracking by Color and Active Contour Models Segmentation* [Text] / F. M. Q. Severgnini, M. L. Oliveira, V. M. S. Mendes, Z. M. A. Peixoto // *IEEE Latin America Transactions*, 2016. – № 14 (3). – pp. 1488-1493, article № 7459639.

8. Gao, X. *A high-speed color-based object detection algorithm for quayside crane operator assistance system* [Text] / X. Gao, H.-G. Yeh, P. A. Marayong // *11th Annual IEEE International Systems Conference, SysCon 2017 – Proceedings*, article № 7934703.

9. Глумов, Н. И. *Построение и применение моментных инвариантов для обработки изображений в скользящем окне* [Текст] / Н. И. Глумов // *Компьютерная оптика*. – 1995. – № 14. – С. 46-54.

10. Сабельников, П. Ю. *Сравнение контуров объектов с частично искажённой формой* [Текст] / П. Ю. Сабельников // *Journal of Qafqaz University. Mathematics and Computer Science*. – Baku, 2012. – № 34 – С. 47-58.

References

1. Martynova, L. A., Koryakin, A. V., Lantsov, K. V., Lantsov, V. V. *Opredelenie koordinat i parametrov dvizheniya objekta na osnove obrabotki izobrazheniy* [Determination of coordinates and parameters of movement of an object on the basis of image processing]. *Computer optics*, 2012, vol. 36, no. 2, pp. 266 – 273.

2. Lucas, B., Kanade, T. *An Iterative Image Registration Technique with an Application to Stereo Vision Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI). Materials of the international seminar, 24-28 august, 1981, Vancouver*, pp. 123-129.

3. Nguyen, V. T., Tropchenko, A. A. *Ier-arhicheskiiy adaptivniy algoritm shablonnogo poiska dlya ocienki dvizheniya pri analize videoposledovatelnosti* [Hierarchical adaptive rood pattern search for motion estimation at video sequence analysis]. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2016, vol. 16, no. 3, pp. 474–481.

4. Batavia, P., Singh, S. *Obstacle detection using adaptive color segmentation and color stereo homography. IEEE International Conference on Robotics and Automation*, 2001, vol. 1, pp. 705-710.

5. Mojsilovic, A. *A computational model for color naming and describing color composition of images. IEEE Transactions on Image Processing*, 2005, vol. 14, pp. 690-699.

6. Barba-Guaman, L., Calderon-Cordova, C., Quezada-Sarmiento, P. A. Detection of moving objects through color thresholding [Detección de objetos en movimiento a través de la umbralización del color]. *Iberian Conference on Information Systems and Technologies, CISTI*, 2017, article no. 7975755.

7. Silva, A. S., Severgnini, F. M. Q., Oliveira, M. L., Mendes, V. M. S., Peixoto, Z. M. A. Object Tracking by Color and Active Contour Models Segmentation. *IEEE Latin America Transactions*, 2016, no. 14 (3), pp. 1488-1493, article no. 7459639.

8. Gao, X., Yeh, H.-G., Marayong, P. A high-speed color-based object detection algorithm for quay-side crane operator assistance system. *11th Annual IEEE International Systems Conference, SysCon 2017*, Proceedings, article no. 7934703.

9. Glumov, N. I. Postroenie i primeneniye momentnykh invariantov dlja obrabotki izobrazhenij v skol'zjashhem okne [Construction and using of moment invariants for image processing in sliding window]. *Computer optics*, 1995, no. 14, pp. 46-54.

10. Sabelnikov, P. Y. Sravneniye konturov ob'ektov s chastichno iskazhjonnoj formoj [A comparison of objects contour with partially misrepresented shape]. *Journal of Qafqaz University. Mathematics and Computer Science*, Baku, 2012, no. 34, pp. 47-58.

Поступила в редакцію 27.10.2017, рассмотрена на редколлегии 22.11.2017

АЛГОРИТМИ ВИЯВЛЕННЯ ОБ'ЄКТІВ І ОЦІНКИ ПАРАМЕТРІВ ЇХ РУХУ В СИСТЕМАХ ТЕХНІЧНОГО ЗОРУ

К. Ю. Дергачов, Л. О. Краснов, Е. В. Пявка

Запропоновано алгоритми виявлення і оцінки параметрів руху об'єктів за результатами відеоспостереження, робота яких заснована на аналізі компонент колірному простору HSV з подальшою бінарizaцією зображення з відсіканням по порозу яскравості і обчисленням моментів бінарних зображень для точної локалізації об'єкту спостереження в кадрі. Реалізовано просту і точну процедуру оцінки траєкторних параметрів (дальності, кутових і швидкісних параметрів) об'єкту. Передбачені заходи по підвищенню порогу стійкої роботи алгоритмів за рахунок використання інтерактивного інтерфейсу. Результати проведених досліджень можуть бути корисні при проектуванні систем технічного зору різного призначення.

Ключеві слова: колірний простір HSV, виділення компонент колірному простору, бінарizaція зображення з відсіканням по порозу яскравості, обчислення моментів бінарних зображень, вимір траєкторних параметрів, пороги стійкої роботи алгоритмів.

ALGORITHMS OF FINDING OUT OBJECTS AND ESTIMATION OF PARAMETERS OF THEIR MOTION ARE IN SYSTEMS OF TECHNICAL SIGHT

K. Dergachov, L. Krasnov, E. Piavka

Most contemporary machine vision systems process video data contained in the recorder that is installed on a mobile platform. These designs are oriented to use the single-board computers Raspberry Pi and the Python programming language combined with OpenCV image processing library. The Python and OpenCV couple allows real-time video data processing while implementing modern algorithms. When doing this, there is a need to provide for high accuracy and fast response of information parameters measurements (machine vision systems should run in real time). The problem of moving objects reliable detection and appropriate parameters evaluation is especially topical in presence of interferences those resulted from scene lighting variations. Such a complex of troublesome factors were taken as a base of the proposed development, the results of which can be used for solving various control problems in robotics engineering as well as accurate positioning of unmanned aerial vehicles.

Algorithms are offered for moving objects detection and following motion parameters determination that are based on video monitoring (video camera surveillance) outcomes is proposed. A conception on synthesizing algorithms for moving objects detection and following appropriate parameters determination, which utilize video observations providing for image binarization on the base of HSV brightness-color components analysis is proposed; the binarization implements the image clipping on a brightness threshold and the following binary image moments detection that allows precise localizing the object within the frame A simple and accurate technique of the object trajectory parameters evaluation (distance to object, and angular and velocity parameters) has been proposed. Measures to raise thresholds of steady running of the algorithm realized by using an interactive interface are provided. Measures to raise thresholds of steady running of the algorithm involving an interactive interface are provided. Investigation outcomes could make use for designing various machine vision systems.

Keywords: HSV color space, allocation of color space components, binarization of the image with clipping on the brightness threshold, calculation of the moments of binary images, measurement of trajectory parameters, thresholds for stable operation of algorithms

Дергачев Константин Юрьевич – канд. техн. наук, ст. науч. сотр., зав. каф. Системы управления летательных аппаратов, Национальный аэрокосмический университет «Харьковский авиационный институт», Харьков, Украина, e-mail: kdergachev@ukr.net.

Краснов Леонид Александрович – канд. техн. наук, ст. науч. сотр., доц. каф. Системы управления летательных аппаратов, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: leonid.krasnov.1947@gmail.com.

Пявка Евгений Валентинович – аспирант кафедры Системы управления летательных аппаратов, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: zeek1910@gmail.com.

Kostiantyn Dergachov – PhD, Senior Researcher, Associate Professor of the Department of Aircraft Control Systems of the National Aerospace University «Kharkov Aviation Institute», Kharkiv, Ukraine.

Leonid Krasnov – PhD, Senior Researcher, Associate Professor of the Department of Aircraft Control Systems of the National Aerospace University «Kharkov Aviation Institute», Kharkiv, Ukraine, e-mail: leonid.krasnov.1947@gmail.com.

Evgeny Piavka – PhD student of the Department of Aircraft Control Systems of the National Aerospace University «Kharkov Aviation Institute», Kharkiv, Ukraine, e-mail: zeek1910@gmail.com.