

DOI: 10.15593/2499-9873/2018.3.07

УДК 004.021

**А.Д. Терехин<sup>1</sup>, С.В. Киндеркнехт<sup>2</sup>**

<sup>1</sup>Пермский государственный гуманитарно-педагогический университет, Пермь, Россия

<sup>2</sup>ЗАО «Технологии Инфокоммуникаций», Пермь, Россия

## **ОТСЛЕЖИВАНИЕ ОБРАЗОВ В ВИДЕОПОТОКЕ РЕАЛЬНОГО ВРЕМЕНИ**

Поставлена и решена задача подсчета машин на автомагистрали с применением стандартного оборудования видеонаблюдения. Для решения задачи использовалась библиотека компьютерного зрения OpenCV, позволяющая работать с каждым кадром видеопотока в реальном времени и обрабатывать поступающую информацию. Обработка данных производилась с помощью сегментирующей нейронной сети. Достигнута точность в 70 % правильного определения числа автомобилей.

**Ключевые слова:** видеопоток, трекинг, хэш-функция, автомобиль, бинаризация.

**A.D. Teryohin<sup>1</sup>, S.V. Kinderknecht<sup>2</sup>**

<sup>1</sup>Perm State Humanitarian-Pedagogical University, Perm, Russian Federation

<sup>2</sup>Company "Technologies of Infocommunications", Perm, Russian Federation

## **REAL-TIME VIDEO STREAM IMAGE TRACKING**

The problem of counting cars on a highway using standard video surveillance equipment was set and solved with the OpenCV computer vision library. The library made possible to grab each video stream frame in real time and process incoming information. Data processing was performed using a segmented neural network. Accuracy in 70% of correct definition of cars was reached.

**Keywords:** video stream, tracking, hash, automobile, binarization.

### **Введение**

Задача отслеживания (трекинга) автомобилей легко может быть решена в случае, если имеется возможность выделить и распознать государственный регистрационный номер транспортного средства [1]. Однако в некоторых случаях задача выделения и распознавания государственного регистрационного номера не может быть решена по имеющемуся изображению. Это может быть вызвано как особенностями применяемой камеры (разрешением камеры и оптики), так и ракурсом или расстоянием от камеры до потока транспортных средств.

В данной статье описывается метод отслеживания (трекинга) автотранспортных средств по изображению, формируемому обзорной камерой с разрешением 4К. Изображение, получаемое с данной камеры, исключает возможность распознавания государственного регистрационного знака, однако остается достаточно качественным для выделения автомобиля на фоне дорожного полотна.

Для осуществления трекинга был разработан собственный алгоритм, использующий API-библиотеку компьютерного зрения OpenCV [2, 3].

### **Постановка задачи**

Рассматривается задача подсчета количества автомобилей, движущихся по автодороге, с использованием имеющейся обзорной камеры. Для подсчета был выделен участок проезжей дороги, представляющий собой четыре полосы движения. Направление движения автомобилей – «от камеры». Требуется разработать программное обеспечение процесса отслеживания, которое по полученному видеопотоку позволяет оператору получать информацию о количестве автомобилей, проехавших по выделенному участку проезжей части за заданный интервал времени наблюдения.

Несмотря на высокое разрешение матрицы камеры и использование качественной оптики, фокусное расстояние объектива, а также расположение камеры над проезжей частью было такое, что средний размер автомобиля на изображении составлял 200×200 пикселей.

Для проведения исследования компанией «Технологии инфокоммуникаций» была предоставлена сегментирующая нейронная сеть, обученная находить автомобили на изображении и разделять их на классы: легковой и грузовой автомобиль. На вход сегментирующей нейронной сети отправлялось изображение интересующего нас участка дорожного полотна. На выходе определялись координаты автомобилей, которые смогла найти нейронная сеть. Требуется на основе нейросетевых технологий исследования видеопотока изображений разработать алгоритм отслеживания автомобилей в реальном времени, позволяющий определять количество проезжающих автомобилей с заданной точностью.

### **Обзор существующих методов отслеживания образов**

Использование библиотеки OpenCV открывает целый спектр готовых решений для отслеживания образов [2, 3]. Это алгоритм Лукаса – Канады, признаки Хаара, а также поиск фрагмента изображения на соседнем кадре.

Алгоритм Лукаса – Канады – точечный метод поиска объектов на соседних кадрах. Суть алгоритма заключается в поиске одной особой точки на двух соседних кадрах [2]. Особая точка – это такая точка изображения, окрестность которой можно отличить от окрестности любой другой точки изображения [4]. Алгоритм Лукаса – Канады очень удобен для непрерывного отслеживания отдельных перемещающихся объектов и позволяет уверенно сопровождать объект на протяжении десятков кадров. К сожалению, алгоритм неприменим в случае, если в процессе отслеживания один объект на время закрывается другим, что приводит к потере отслеживаемого объекта.

Признаки Хаара – это признаки цифрового изображения, которые используются в распознавании образов с помощью методов компьютерного зрения [5–7]. При использовании данного метода трекинга автомобилей необходимо иметь большое количество изображений автомобилей, а также изображений, где автомобиля нет. При этом все изображения используются для обучения каскада Хаара [7], который и будет определять местонахождение автомобиля.

Поиск фрагмента изображения на полном изображении реализован, как и вышеописанные методы, в библиотеке OpenCV [2, 3]. Данный метод ищет только полное совпадение с фрагментом и не всегда позволяет найти искомый объект на полном изображении [2]. При трекинге потока движения автотранспортных средств, в ситуации, когда автомобиль переезжает на соседнюю полосу, метод поиска фрагмента не срабатывает, поскольку нет полного совпадения с объектом, который ищем.

В результате анализа имеющихся алгоритмов можно сделать вывод, что наиболее перспективные результаты в рассматриваемом случае могут быть получены при использовании алгоритма частичного сравнения фрагментов изображения, описанного ниже.

### **Описание алгоритма трекинга**

Ввиду того что на изображении, полученном с видеокамеры, автомобили часто движутся достаточно кучно, частично перекрывая друг друга, требовалось реализовать алгоритм трекинга, который позволил бы осуществить визуальный трекинг в таких условиях. Был реализован следующий алгоритм:

1. На исходное изображение накладывается маска, ограничивающая область детекции автотранспортных средств.

2. Изображение с наложенной маской отправляется сегментирующей нейронной сети.

3. С исходного изображения вырезаются автомобили по координатам, которые определила нейронная сеть.

4. Изменяются размеры автомобиля и приводятся к некому стандарту, который используется в алгоритме. За стандарт взято изображение автомобиля размером  $128 \times 128$  пикселей. Стандартное значение размера было определено на основе вычислительных экспериментов.

5. Изображение автомобилей переводятся в бинарное представление. Для этого сначала рассчитывается средний цвет изображения, что позволяет сделать OpenCV, который используется как пороговый для бинаризации, и также при помощи OpenCV изображение переводится в бинарный вид, где белый цвет кодируется 0, а черный – 1 [7]. На рисунке представлен пример бинарного изображения автомобиля. На нем хорошо виден контур автомобиля.



Рис. Пример бинарного представления автомобиля

6. Автомобиль попадет в зону детекции, когда проехал первую невидимую линию, которая задана в алгоритме. Пока автомобиль не прошел следующую невидимую линию, которая также прописана в алгоритме, он находится в зоне детекции.

7. По бинарному представлению изображения автомобиля рассчитывается хэш-функция [6] и заносится в базу данных, в дальнейшем по данному значению хэш-функции (хэшу) сравниваются различные изображения и определяется степень их схожести.

8. При ситуации, когда нейронная сеть обнаружила автомобиль повторно, происходит поиск автомобиля в базе с похожим хэшем, процентный показатель их различия не должен превышать 10 %.

9. После обнаружения автотранспортного средства в базе происходит его выделение на исходном изображении при помощи API OpenCV.

При использовании предоставленной нейронной сети может возникнуть ситуация, когда автомобиль детектируется нейронной сетью не на каждом кадре. Данная ситуация решается путем использования хэша бинарного изображения автомобиля и его процентного различия с автомобилями, хранящимися в базе. Автомобиль, двигаясь по дорожному полотну, почти не изменяет своего угла направления, что позволяет детектировать его на протяжении всей зоны детекции без подтверждения его на каждом кадре.

Считается, что если автомобиль не находится нейронной сетью 300 и более кадров подряд, то он выехал за зону трекинга, и его удаляют из базы.

Для определения точности метода трекинга осуществлялось сравнение результатов компьютерного счета с данными ручного счета. С этой целью сначала просматривалось видео длительностью 5 мин, в котором подсчитывались автомобили вручную, после этого то же самое видео обрабатывалось на основе вышеизложенного алгоритма. Отношение данных ручного метода подсчета и подсчета алгоритмом позволяет оценить получаемую точность.

### **Заключение**

Как показали проведенные исследования, предложенный метод трекинга показывает неплохую точность для низкокачественных камер наблюдения. В пределах рассмотренной задачи описанный выше метод трекинга автомобилей достигает точности в 70 %. Дальнейшим развитием данного механизма трекинга может быть комбинированный механизм трекинга, использующий алгоритм Лукаса – Канады, что позволит, с одной стороны, получить большую точность отслеживания при отсутствии перекрытия объектов в поле изображения, с другой – обрабатывать перекрываемые объекты с использованием вышеописанного алгоритма.

### **Список литературы**

1. Мерков А.Б. Распознавание образов. Введение в методы статистического обучения. – М.: Едиториал УРСС, 2011. – 256 с.
2. Документация библиотеки компьютерного зрения OpenCV [Электронный ресурс]. – URL: [https://docs.opencv.org/2.4/modules/imgproc/doc/motion\\_analysis\\_and\\_object\\_tracking.html](https://docs.opencv.org/2.4/modules/imgproc/doc/motion_analysis_and_object_tracking.html).

3. Использование библиотеки компьютерного зрения OpenCV на языке программирования Java [Электронный ресурс]. – URL: <https://habr.com/company/intel/blog/333612/>
4. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б.А. Алпатов, П.В. Баба-ян, О.Е. Балашов, А.И. Степашкин. – М.: Радиотехника, 2008. – 176 с.
5. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. – М.: Вильямс, 2004. – 928 с.
6. Потапов А.С. Распознавание образов и машинное восприятие. – СПб.: Политехника, 2007. – 552 с.
7. Шапиро Л., Стокман Дж. Компьютерное зрение. – М.: Бином. Лаборатория знаний, 2006. – 752 с.

### References

1. Merkov A.V. Recognition of images. Introduction to statistical learning methods. М.: Editorial URSS, 2011 – 256 p.
2. Documentation of the library of computer vision OpenCV [Electronic resource]. –URL: [https://docs.opencv.org/2.4/modules/imgproc/doc/motion\\_analysis\\_and\\_object\\_tracking.html](https://docs.opencv.org/2.4/modules/imgproc/doc/motion_analysis_and_object_tracking.html).
3. Use of OpenCV computer vision library in Java programming language [Electronic resource]. URL: <https://habr.com/company/intel/blog/333612/>
4. Alpatov B.A., Babayan P.V., Balashov O.E., Stepashkin A.I. Methods of automatic detection and tracking of objects. Image processing and management. – Radio Engineering, 2008 – 176 p.
5. Forsythe D., Pons G. Computer vision. Modern approach. – Wil- liams, 2004 – 928 p.
6. Potapov A.S. Pattern Recognition and Machine Perception. – S-P.: Polytechnic, 2007 – 552 p.
7. Shapiro L., Stockman J. Computer vision. – Bean. Laboratory of Knowledge, 2006. 752 p.

Получено 28.05.2018

### **Об авторах**

**Терехин Александр Дмитриевич** (Пермь, Россия) – студент Пермского государственного гуманитарно-педагогического университета (614045, г. Пермь, ул. Пушкина, 42, e-mail: ateryohin@techinfocom.com).

**Киндеркнехт Сергей Владимирович** (Пермь, Россия) – директор компании ЗАО «Технологии инфокоммуникаций» (614000, Россия, г. Пермь, ул. Аркадия Гайдара, 8 Б, офис 201, e-mail: skinderknecht@techinfocom.com).

### **About authors**

**Alexander D. Teryokhin** (Perm, Russian Federation) – Student, Perm State Humanitarian-Pedagogical University (42, Pushkin st., Perm, 614045, Russian Federation, e-mail: ateryohin@techinfocom.com).

**Sergey V. Kinderknecht** (Perm, Russian Federation) – Director, Company “Technologies of Infocommunications” (Office 201, 8-B, Arkadiy Gaidar st., Perm, 614000, Russian Federation, e-mail: skinderknecht@techinfocom.com).