

Слежение за выделенными объектами на протяжении видеоряда

07, июль 2015

Волосатова Т. М.¹, Яблоков В. Е.^{1,*}

УДК: 004.93`12

¹Россия, МГТУ им. Н.Э. Баумана

*v.yablokov@gmail.com

Введение

Технология автоматизированного распознавания и дальнейшего отслеживания образов на протяжении серии изображений как подраздел области компьютерного зрения представляет в настоящий момент большой практический интерес. Это связано со следующими факторами. Существует множество научных работ, посвященных данной тематике, которые создают хорошую базу, востребованную для освоения данной области инженерами и разработчиками программного обеспечения. За последние 10 лет были достигнуты значительные результаты в практическом использовании методов распознавания образов. На данный момент созданы и активно поддерживаются разработчиками различные библиотеки и наборы инструментальных средств, облегчающие применение этих методов при решении различных актуальных задач. Развитие аппаратных средств, а именно возросшая вычислительная мощность носимой электроники и распространенность систем облачных вычислений, делает возможным повсеместное использование программных продуктов, основанных на алгоритмах компьютерного зрения, причем не только в сфере научной и инженерной деятельности, но и в бытовой потребительской сфере.

Возникающее вследствие этих факторов разнообразие практических задач требует не только широкого алгоритмического инструментария, но и комплексных программных продуктов, облегчающих решение этих задач на пользовательском уровне [1].

Одной из перспективных областей применения компьютерного зрения является обработка изображений и видеозаписей, полученных с помощью летательных и космических аппаратов. Растущей популярностью таких устройств, как мультикоптеры, обусловлена потребность в удобных средствах анализа подобных данных.

Кроме того, одним из направлений научно-исследовательской работы в МГТУ им. Н.Э. Баумана является разработка специализированного автоматизированного рабочего места (далее АРМ) для работы с результатами воздушной фото- и видео- разведки, в част-

ности отслеживания интересующих оператора объектов в видеопотоке [1]. В рамках этой работы был выполнен опытно-конструкторский проект, цель которого – разработка и изготовление программного комплекса для выделения на снимках интересующих пользователя объектов и последующего слежения за выбранными объектами на протяжении видеоряда.

1. Обзор методов распознавания объектов на изображении

Основной задачей, возникающей при отслеживании объектов в видеопотоке, является их автоматическое нахождение на каждом отдельно взятом кадре. В рамках проекта был проведен сравнительный обзор различных классов методов распознавания объектов на изображениях.

1.1. Сегментация изображений (Image Segmentation)

Сегментация – это процесс разделения цифрового изображения на несколько множеств пикселей, или присвоения каждому пикселю таких меток, чтобы пиксели с одинаковыми метками имели общие визуальные характеристики [2]. Цель сегментации состоит в упрощении представления изображения, чтобы его было легче анализировать. В результате сегментации обычно выделяют границы и объекты на изображениях.

Пиксели, принадлежащие к одним и тем же сегментам, похожи по некоторой вычисленной характеристике (например, по цвету, яркости), а соседние элементы значительно различаются по ней (в соответствии с рис. 1).

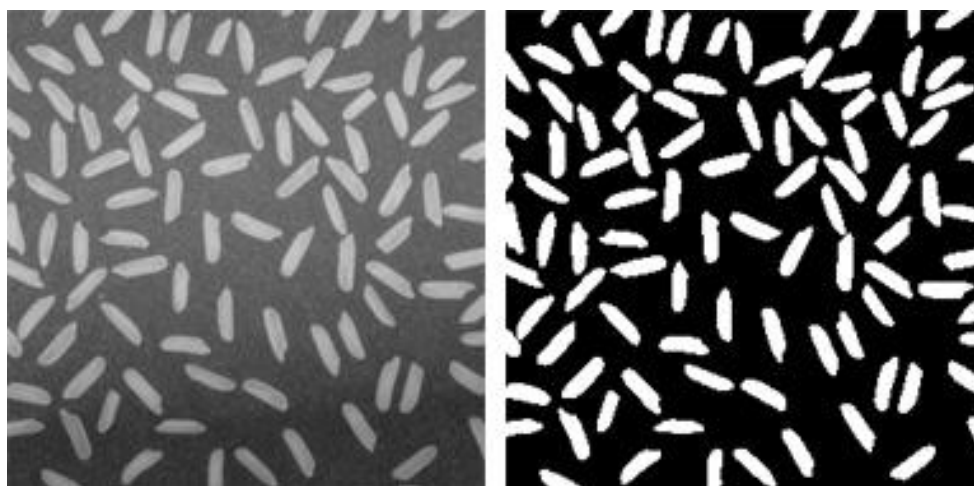


Рис. 1. Пример сегментации изображения

1.2. Поиск по шаблону (Template Matching)

Идея поиска объектов на изображении по шаблону заключается в сравнении шаблона, на котором изображен искомый объект, с подобластями обрабатываемого изображения (в соответствии с рис. 2) [3].



Рис. 2. Пример работы программного продукта, предназначенного для поиска объектов по шаблону Euresys EasyFind

1.3. Детектирование характерных признаков (Feature Detection)

Метод детектирования признаков призван преодолеть вышеприведенные недостатки. Признаками, или особенностями, в данном контексте называются характерные локальные особенности изображения. К простейшим относятся углы, грани. Поиск объекта в данном случае основан на сравнении характерных признаков обрабатываемого кадра и шаблона, на котором изображен искомый объект [4].

Локальные признаки должны быть повторяемыми (устойчивыми к изменениям ракурса или освещения на протяжении видеоряда), компактными (их количество должно быть значительно меньше общего числа пикселей изображения), уникальными (каждая особенность должна иметь свое собственное описание).

Для обнаружения характерных признаков используются детекторы. Одним из самых распространенных является детектор Харриса, распознающий признаки типа «угол» на изображении.

Угловые детекторы обладают недостатком: чувствительностью к масштабированию изображения. Чтобы избежать его, была предложена концепция так называемых капель (англ. *blob*) – каплевидных окрестностей с расположенной в центре особой точкой. Одним из наиболее общих blob-методов является LoG (The Laplacian of Gaussian). LoG – это фильтр, применяющий к изображению последовательно оператор Гаусса и оператор Лапласа (в соответствии с рис. 3). Распространен также детектор DoG на основе разности гауссиан.

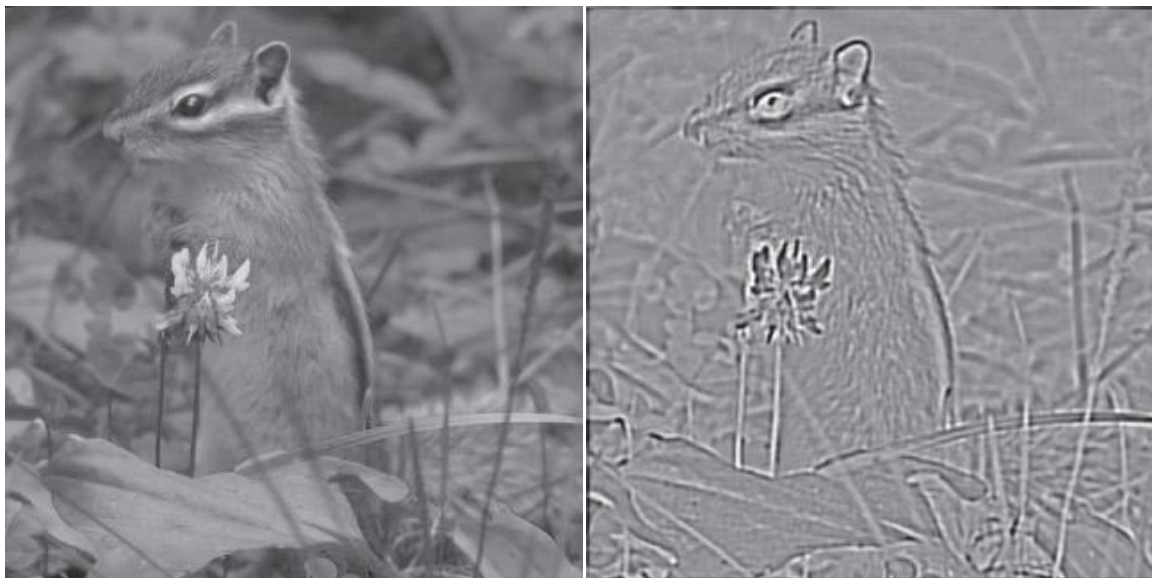


Рис. 3. Результат применения фильтра LoG к изображению

После нахождения особых точек необходимо их сравнивать. Для этого требуется способ компактного представления характерных особенностей. В практических задачах эталонным считается дескриптор SIFT (Scale-Invariant feature transform) и его производные, такие как SURF.

1.4. Категориальное распознавание

Методы детектирования характерных признаков хорошо подходят для решения задачи поиска по базе изображений [5]. Однако в соответствии с постановкой задачи, рассматриваемой в данной работе, требуется не просто обнаруживать на кадрах видеоряда области, соответствующие некоторому эталону, но и распознавать все объекты определенного класса. Рассматриваемую задачу можно было бы решить методами детектирования признаков, но при этом потребовалось бы создавать большое количество эталонов и тратить во время работы разработанной программы большое время на сравнение кадра с каждым из них. Избежать этого позволяет другая категория решений, основанных на классификации объектов, или категориальное распознавание [6]. Оно состоит из двух основных элементов: определения набора признаков или дескрипторов и машинного обучения классификатора.

В качестве набора признаков может использоваться, например, гистограмма направленных градиентов (HOG) или признаки Хаара.

Признаки HOG основаны на подсчете количества направлений градиента в локальных областях изображения. HOG вычисляется на плотной сетке равномерно распределенных ячеек (см. рис. 4).

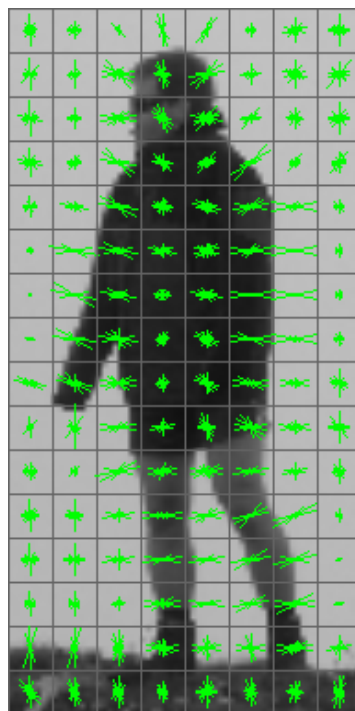


Рис. 4. Генерация HOG-признаков изображения

Признаки, или примитивы Хаара представляют собой прямоугольники, состоящие из смежных областей (см. рис. 5). Эти области позиционируются на изображении, далее происходит суммирование интенсивности пикселей в областях, а затем вычисляется разность между полученными суммами, которая и является значением определенного признака определенного размера, расположенного на изображении определенным образом. Пример использования признаков Хаара приведен на рис. 6. Достоинством признаков Хаара является сравнительно высокая скорость вычисления.

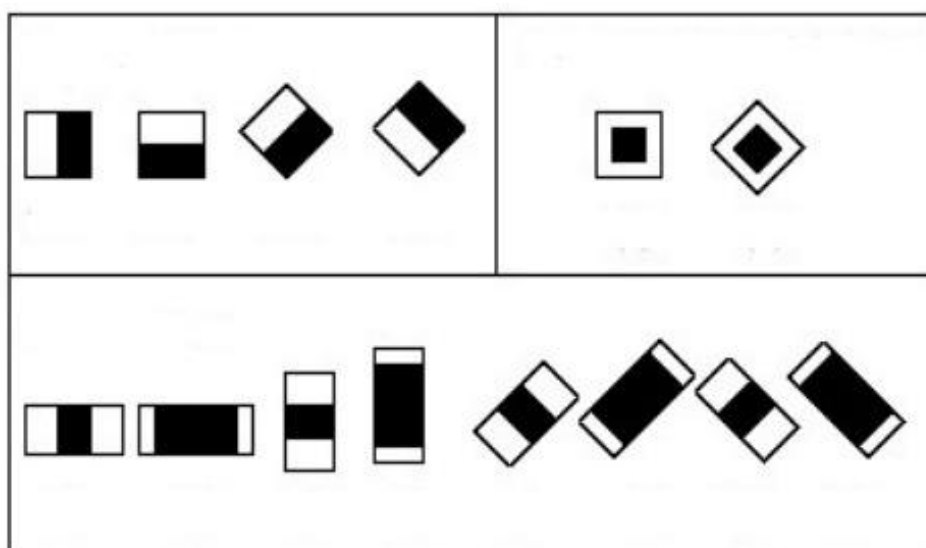


Рис. 6. Различные признаки Хаара

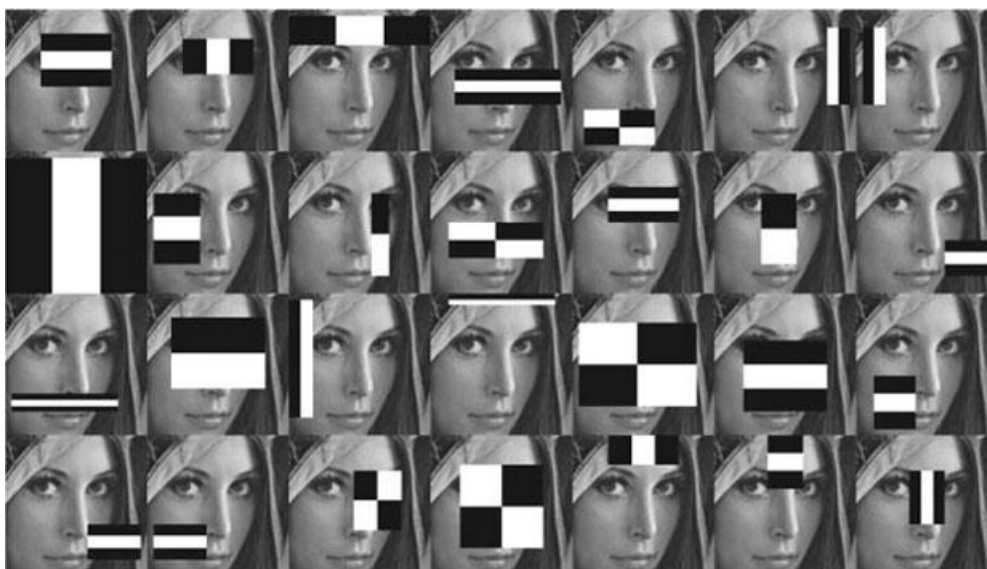


Рис. 6. Пример нахождения признаков Хаара на изображении

Машинное обучение используется для создания классификатора. Классификатор используется для того, чтобы определить, какие признаки принадлежат объекту. При этом для обучения используется некоторая база этих признаков.

Достоинства и недостатки вышеописанных подходов приведены на табл. 1.

Табл. 1. Сравнительные характеристики методов распознавания образов

Название класса методов	Достоинства	Недостатки
Сегментация [2]	Простота использования в тех ситуациях, когда исследуемые объекты значительно отличаются от остального фона по какому-либо параметру (такие предметные области как биоинформатика, поиск дефектов материалов).	Низкая универсальность методов, необходимость тонкой настройки параметров работы алгоритма распознавания в каждом отдельном случае, высокая чувствительность к изменениям освещения сцены
Поиск по шаблону [3]	Хорошая применимость при анализе сцен, в которых камера статична, а все экземпляры искомым объектов выглядят одинаково (например, при распознавании деталей на сборочной линии).	Неустойчивая работа в случае масштабирования, сдвига, поворота изображений, а также в случаях, когда распознаваемый объект виден не полностью.
Детектирование признаков [4]	Инвариантность к малым поворотам, масштабированию объектов, к изменению освещения сцены.	Невозможность определения объекта как экземпляра класса, сложность распознавания динамически видоизменяющихся объектов.
Категориальное распознавание [6]	Высокая скорость поиска объектов, наилучшее среди рассмотренных подходов качество распознавания (зависит от качества обучения классификатора).	Длительное время обучения классификатора объектов, потребность в большом количестве изображений искомого объекта для формирования обучающей выборки.

2. Метод Виолы – Джонса

По итогам проведенного обзора было принято решение использовать метод категориального распознавания объектов на изображении. В настоящий момент самым популярным в силу высокой скорости и эффективности является представленный в 2001 году метод Виолы – Джонса [7].

Основу метода Виолы – Джонса составляют примитивы Хаара. В оригинальной версии алгоритма использовались только примитивы без поворотов, а значение признака вычислялось как простая разность сумм яркостей пикселей в подобластях. В дальнейшем были предложены примитивы с наклоном на 45 градусов, а также несимметричных конфигураций. Вместо вычисления простой разности было предложено приписывать каждой подобласти определенный вес и вычислять значение признака как взвешенную сумму пикселей разнотипных областей. Признаки вычисляются в пределах перемещающегося по изображению сканирующего окна переменного размера.

Изображение при обработке методом Виолы – Джонса хранится в так называемом интегральном формате (англ. *Summed Area Table* – SAT): в каждом пикселе изображения записана сумма всех пикселей левее и выше данного. SAT-представление можно построить заранее один раз для всего изображения, в соответствии с формулами (1) и (2):

$$SAT(x, y) = SAT(x, y - 1) + SAT(x - 1, y) + I(x, y) - SAT(x - 1, y - 1) \quad (1)$$

$$SAT(-1, y) = SAT(x, -1) = SAT(-1, -1) = 0 \quad (2),$$

где x, y – координаты пикселя на изображении, $I(x, y)$ – интенсивность пикселя с координатами (x, y) исходного изображения. Благодаря этому проверка признака на конкретной позиции проводится за константное время.

В алгоритме Виолы – Джонса используется так называемый бустинг [8] (англ. *boosting*) – комплекс методов, способствующих повышению точности аналитических моделей, или усиление «слабых» моделей. Это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих. В данном случае применяется алгоритм бустинга AdaBoost. AdaBoost создает взвешенную комбинацию «слабых» классификаторов с целью создания одного «сильного». Под «слабыми» подразумеваются те классификаторы, которые получают правильный ответ ненамного чаще случайного угадывания, то есть более чем в 50% случаев.

Метод Виолы – Джонса использует каскадную модель сильных классификаторов. Она представляет собой дерево принятия решений, где каждый узел построен так, чтобы детектировать все интересующие образы и отклонять области, не являющиеся образами. Каскад применяется по следующим правилам:

- работа ведется с «простыми» классификаторами;
- только положительное значение одного классификатора запускает следующий, более приспособленный;
- отрицательное значение классификатора приводит к немедленному переходу к следующему сканирующему окну и отбрасыванию текущего;

- цепочка классификаторов становится все более сложной, и количество ошибок сокращается.

При обучении классификатора используется тестовая выборка из n изображений, гарантированно содержащих искомый объект, и m изображений, его не содержащих.

3. Обзор инструментальных средств для разработки программного продукта

В настоящий момент существует большое количество специализированных программных библиотек компьютерного зрения как общего, так и специального назначения. Чтобы отобрать наилучшие из них, сформулирован набор предъявляемых требований:

- разнообразие решаемых задач;
- широкое распространение;
- активное развитие и поддержка продукта компанией-разработчиком либо открытым сообществом;
- высокое качество технической документации;
- высокоуровневая иерархическая структура компонентов;
- относительная простота использования;
- стабильность и удовлетворительная скорость работы;
- кроссплатформенность и переносимость.

Несмотря на упомянутое выше многообразие средств, для создания программного продукта были выбраны два наиболее полно соответствующих изложенным требованиям продукта.

3.1. Библиотека OpenCV

Наиболее популярным среди разработчиков средством для решения задач компьютерного зрения является библиотека OpenCV (Open Source Computer Vision Library) [9].

OpenCV – библиотека с открытым исходным кодом, представленная в 2006 году и реализованная на языках программирования C/C++. Она также может быть использована совместно с языками программирования Python, Java, Ruby и другими. Библиотека выпускается по лицензии BSD и может свободно распространяться как в академических, так и в коммерческих целях.

Существуют реализации OpenCV для множества платформ: Microsoft Windows (компиляторы Microsoft Visual C++, Intel Compiler, MinGW), Linux (компиляторы GCC, Intel Compiler), Apple OS X (компилятор GCC). Поддерживаются распространенные мобильные операционные системы Android и Apple iOS.

Привлекательной для разработчиков является также возможность дополнительного ускорения OpenCV на платформах Intel с помощью инструментов Intel Performance Libraries.

OpenCV обладает обширным набором инструментов, подходящим для решения многих задач компьютерного зрения. Модули библиотеки предоставляют следующие возможности:

- реализация базовых структур, математических вычислений, XML- и YAML-сериализации и т.д.;
- обработка изображений (фильтрация, геометрические и цветовые преобразования);
- графический интерфейс для ввода / вывода изображений и видео;
- модели машинного обучения;
- распознавание плоских примитивов (Feature Detection);
- анализ движения, отслеживание объектов;
- обнаружение объектов (метод Виолы – Джонса, признаки HOG и т.д.);
- обработка стереоизображений.

Продолжается активное развитие библиотеки: новейшая на момент написания данного текста версия выпущена 4 июня 2015 года (версия 3.0).

Версии OpenCV оптимизированы для выполнения на разных платформах и аппаратных архитектурах. Присутствует поддержка технологии вычислений на графических процессорах NVidia CUDA [10].

К слабым сторонам продукта можно отнести недостаточно полную техническую документацию, что, безусловно, затрудняет его изучение и обеспечивает довольно высокий порог вхождения.

Тем не менее, библиотека OpenCV в достаточной степени отвечает предъявленным требованиям к инструментальным средствам.

3.2. Среда разработки MathWorks MATLAB

Продукт MATLAB, разработанный компанией MathWorks, включает высокоуровневый интерпретируемый язык программирования, интерактивную среду разработки и большое количество инструментов, предназначенных для решения задач самых разных отраслей науки [11].

Среда разработки MATLAB обладает всеми необходимыми инструментами для разработки объектно-ориентированных программ, включая отладчик, профилировщик, набор встроенных средств визуализации данных, а также удобный конструктор графических интерфейсов. Код, созданный на языке MATLAB, может исполняться отдельно от среды разработки, будучи собранным в независимое приложение с помощью компоненты MATLAB Compiler.

Созданный на языке MATLAB код предназначен для исполнения в большинстве современных операционных систем общего назначения: Microsoft Windows, GNU Linux, Apple OS X на базе аппаратной архитектуры x86. Кроме того, с помощью компонентов MATLAB Coder и Embedded Coder программы на языке MATLAB могут бы транслированы в код на языках C и C++ с соблюдением стандартов ANSI/ISO и в дальнейшем скомпилированы для других архитектур (например, для встраиваемых устройств на базе процес-

соров ARM). Эта функция обеспечивает высокую переносимость разработанных в среде MATLAB приложений.

Кроме того, имеется возможность взаимодействия MATLAB-кода с кодом, написанным на других языках, таких как C, C++, программной платформой Microsoft .NET, а также веб-сервисами по протоколам SOAP и WSDL.

Продукт MATLAB активно разрабатывается в настоящее время. Новейшая на момент написания данного текста версия была выпущена 5 марта 2015 года (версия 2015a).

Официальная документация MATLAB отличается полнотой и наглядностью, содержит большое количество практически полезных примеров использования функций и дает возможность осваивать язык максимально быстро.

Наибольший интерес в данном случае представляет набор инструментов MATLAB, предназначенный для решения задач компьютерного зрения, который выделен в библиотеку Computer Vision System Toolbox [12]. В ее состав включены следующие возможности:

- обработка изображений: методы улучшения, фильтрации, геометрических и цветовых преобразований;
- распознавание объектов: метод Виолы – Джонса, обучаемые классификаторы, средства OCR и др.;
- отслеживание объектов: фильтр Кальмана, алгоритмы CAMShift, Kanade-Lucas-Tomasi (KLT) и др.;
- обнаружение и сопоставление признаков: детекторы углов (Harris, Shi & Tomasi), дескрипторы SURF, MSER, обнаружение HOG-признаков, сопоставление признаков и др.;
- автоматическая калибровка камер, в том числе стереокамер;
- работа со стереоизображениями, в том числе восстановление трехмерных сцен;
- работа с видео: чтение, запись, воспроизведение, деинтерлейсинг, наложение и комбинирование кадров и т.д.;
- средства визуализации результатов обработки;
- поддержка генерации кода на языке ANSI/ISO C.

Продукт MATLAB является платным, однако программа распространения льготных академических лицензий позволяет избежать финансовых затрат в рамках академической деятельности.

Таким образом, данная библиотека в частности и пакет MATLAB в целом функционально полностью соответствуют приведенным требованиям.

3.3. Выбор наиболее подходящего средства разработки

В результате исследования инструментов, подходящих для решения поставленной задачи, было принято решение использовать в качестве среды разработки среду MATLAB. Оно основывается на том, что по сравнению с библиотекой OpenCV MATLAB обладает схожей необходимой и достаточной функциональностью, но позволяет разрабатывать

программы быстрее за счет более удобной работы с данными и богатых возможностей среды разработки, обладает лучшей документацией и аналогичными возможностями переносимости. При этом на основании консультаций с разработчиками, использующими оба рассматриваемых продукта, был сделан вывод о сопоставимой скорости работы функций MATLAB Computer Vision System Toolbox по сравнению с их аналогами из библиотеки OpenCV, причем в случае нехватки производительности имеется пространство для оптимизации MATLAB-кода путем его трансляции в язык C. Тем не менее библиотека OpenCV также является удобным средством решения большинства задач компьютерного зрения, причем в отличие от MATLAB не требует специальной среды исполнения.

4. Описание созданного программного продукта

После исследования методов отслеживания объектов на протяжении видеоряда с помощью языка MATLAB был разработан программный продукт, решающий данную задачу. Он состоит из нескольких модулей.

4.1. Модуль обучения классификатора объектов

При обучении классификатора в качестве входных данных используется обучающая выборка кадров (на одной их части, или позитивной выборке заведомо присутствует искомый объект, на другой, негативной выборке он заведомо отсутствует), и после указания типа распознаваемых признаков (признаки Хаара, признаки HOG) формируется обученный классификатор, представленный в виде XML-файла.

4.2. Модуль отслеживания объектов

Модуль последовательно считывает кадры видеопотока. На каждой итерации выполняются следующие действия:

- 1) методом Виолы – Джонса обнаруживаются все объекты в кадре с помощью обученных классификаторов (возможно одновременное распознавание нескольких экземпляров объектов разных классов);
- 2) анализируются уже известные и отслеживаемые траектории: на основе фильтра Кальмана предсказываются новые положения уже известных объектов;
- 3) далее с помощью алгоритма Куна – Манкреса [13] решается задача о назначениях, в результате чего часть найденных в кадре объектов сопоставляется известным траекториям;
- 4) положения тех найденных в кадре объектов, которым «не досталось» траектории, становятся началом новых траекторий;
- 5) траектории, новое предсказанное положение которых не соответствует ни одному из найденных объектов, маркируются как скрытые и удаляются спустя несколько кадров, если объект так и не будет найден в предсказанном положении на изображении.

В процессе своей работы созданная программа генерирует видеофайл, в котором отмечены и подписаны распознанные и отслеженные с ее помощью объекты (см. рис. 7).



Рис. 7. Примеры результатов работы созданного программного обеспечения

Источником исходных данных программы может выступать как записанный видеофайл, так и видеопоток, получаемый с подключенной к ЭВМ камеры. Это дает возможность использовать созданное приложение не только для последующей обработки заранее полученных данных, но и непосредственно во время видеосъемки.

4.3. Модуль графического интерфейса

Графический интерфейс призван упростить работу с программой для конечного пользователя и предоставляет следующие возможности:

- выбор источника видеоданных (камера или видеофайл);
- включение / выключение отображения видеоплеера во время работы;
- включение / выключение записи результатов работы в видеофайл;
- выбор одного или нескольких XML-файлов классификаторов, на основе которых будет производиться поиск объектов;
- ограничение количества обрабатываемых кадров видео;
- задание ограничений для отбрасывания распознанных объектов, не подходящих по размеру.

Графический интерфейс позволяет оператору АРМ использовать ее отдельно в интерактивном режиме и опционально просматривать результаты распознавания и отслеживания во время работы программы. В этом варианте конфигурации программа также является удобным инструментом для испытания качества обучаемых классификаторов образов.

Помимо работы в графическом режиме, функциональная часть программного продукта может быть использована в виде отдельного компонента, интегрируемого в состав комплексного многофункционального АРМ.

4.4. Переносимость созданного программного продукта

Компоненты программного продукта являются переносимыми и могут выполняться в среде MATLAB Runtime на базе широкого спектра клиентских и серверных операцион-

ных систем общего назначения (Microsoft Windows, GNU Linux, Apple OS X) и аппаратной архитектуры x86. Кроме того, исследована возможность транслирования кода, созданного на языке MATLAB, в язык C, что упрощает распространение и развертывание программного продукта, в том числе на ЭВМ с аппаратной архитектурой ARM.

Заключение

Полученные результаты могут быть интересны инженерам для использования программы в качестве составной части многофункциональной программно-аппаратной системы, а также студентам, изучающим компьютерное зрение.

Разработанный программный продукт призван решить задачу отслеживания объектов на протяжении видеопотока наиболее универсальным образом и может быть использован в большом количестве разнообразных предметных областей науки в качестве средства автоматизации анализа видеоданных.

В процессе тестирования программы была отмечена хорошая применимость метода Виолы – Джонса, изначально разработанного для распознавания человеческих лиц, к широкому кругу иных задач, в том числе задаче классификации автомобильного транспорта, фиксируемого движущейся камерой.

Экспериментально проверены удобство и высокая производительность библиотеки MATLAB Computer Vision System Toolbox как средства разработки программного обеспечения, решающего задачи компьютерного зрения.

Список литературы

1. Волосатова Т.М., Чичварин Н.В., Афиногенов Е.И. Специализированное АРМ для проектирования программного обеспечения комплексов навигации и воздушной разведки // Наука и образование: электронное издание МГТУ им. Н.Э. Баумана. 2013. № 8. С. 353 – 370. Режим доступа: <http://technomag.bmstu.ru/doc/585918.html> (дата обращения: 07.05.2015). DOI: [10.7463/0813.0585918](https://doi.org/10.7463/0813.0585918)
2. Arbelaez P., Maire M., Fowlkes C., Malik J. Contour Detection and Hierarchical Image Segmentation // IEEE TPAMI. 2011. Vol. 33(5). Pp. 898–916. DOI: [10.1109/TPAMI.2010.161](https://doi.org/10.1109/TPAMI.2010.161)
3. Brunelli R. Template Matching Techniques in Computer Vision. Theory and Practice. New York: Wiley. 2009. 348 p.
4. Lowe D.G. Object recognition from local scale-invariant features // International Conference on Computer Vision - ICCV. 1999. Vol. 2. Pp. 1150–1157. DOI: [10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410)
5. Dalal N., Triggs B. Histograms of oriented gradients for human detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005. Vol. 1. Pp. 886–893. DOI: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177)

6. Компьютерное зрение в MATLAB для задач детектирования и слежения / Videos and Webinars. // MathWorks. Режим доступа: <http://www.mathworks.com/videos/computer-vision-with-matlab-87719.html> (дата обращения: 21.05.2015).
7. Viola P., Jones M.J. Rapid Object Detection using a Boosted Cascade of Simple Features // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Kaia, Hawaii, USA. 2001. Vol. 1. Pp. 511–518.
8. Freund Y., Shapire R.E. A Short Introduction to Boosting // Journal of Japanese Society for Artificial Intelligence. 1999. Vol. 14(5). Pp. 771–780.
9. Bradski G.R., Kaehler A. Learning OpenCV. Sebastopol, California, USA: O`Reilly Media, 2008. 580 p.
10. CUDA. // OpenCV. Режим доступа: <http://opencv.org/platforms/cuda.html> (дата обращения: 27.04.2015).
11. Features / MATLAB. // MathWorks. Режим доступа: <http://www.mathworks.com/products/matlab/features.html> (дата обращения: 23.04.2015).
12. Features / Computer Vision System Toolbox. // MathWorks. Режим доступа: <http://www.mathworks.com/products/computer-vision/features.html> (дата обращения: 14.04.2015).
13. Munkres J. Algorithms for the Assignment and Transportation Problems // Journal of the Society for Industrial and Applied Mathematics. 1957. Vol. 5(1). Pp. 32–38.