

# Разработка и исследование программного обеспечения для повышения эффективности обнаружения движений в системе интеллектуального видеонаблюдения

Е.Е. Истратова, А.О. Смирнова

**Аннотация**—В статье представлены результаты разработки и тестирования программного обеспечения для обнаружения движений, обладающего низкими требованиями к вычислительным ресурсам и при этом функциональными характеристиками, достаточными для работы системы интеллектуального видеонаблюдения в режиме реального времени. В ходе работы были рассмотрены особенности применения программно-аппаратных решений и алгоритмов обнаружения движений на устройствах с ограниченной вычислительной мощностью, определены требования к программному обеспечению. В результате выполнения программной реализации было разработано решение для обнаружения движений, отличительной особенностью которого является возможность работы с несколькими потоками камер одновременно в режиме реального времени. Программное обеспечение может быть установлено на встраиваемых системах и использовано для решения задач мониторинга и безопасности помещений и территорий. Готовый программный продукт позволяет задавать параметры потоков при добавлении камеры, а также в последующем изменять их, также предусмотрено редактирование значений по умолчанию и глобальных параметров потоков камер, что позволяет повысить эффективность процесса обнаружения движений в системах интеллектуального видеонаблюдения.

**Ключевые слова**—обнаружение движений, интеллектуальное видеонаблюдение, алгоритм, вычитание фона, программное обеспечение, разработка.

## I. ВВЕДЕНИЕ

Активное распространение на современном рынке систем интеллектуального видеонаблюдения обусловлено их значительными преимуществами перед процессом ручного анализа видеопотоков, что проявляется как в повышении точности обнаружения объектов, так и в скорости данного процесса [1].

Однако основным недостатком систем данного вида является необходимость их применения на встраиваемых платформах, обладающих небольшой вычислительной мощностью [2]. Так, в последние годы обработка видео выполняется с помощью методов на основе глубокого обучения, которые обеспечивают более высокую точность, но при этом более требовательны к вычислительным ресурсам [3].

В системах видеонаблюдения со стационарными

камерами первый этап обнаружения движущихся объектов имеет решающее значение для правильного моделирования поведения этих объектов, а также является наиболее сложным с точки зрения времени выполнения. Существует множество алгоритмов, которые обеспечивают надежную и адекватную маску сегментации, получая в реальном времени коэффициенты для уменьшенных размеров изображения. Однако из-за повышенной производительности оборудования камеры применение классических методов к последовательностям с более высоким разрешением не выполняется в режиме реального времени, что ставит под угрозу их использование в реальных системах видеонаблюдения [4]. Помимо этого, большое разнообразие видеоконтента делает обработку различных движений, краев, текстур и их комбинаций с помощью одного алгоритма очень сложной. Что в результате приводит к необходимости разработки комбинированного алгоритма для работы с видеопотоками в режиме реального времени [5].

Так как видеосервер размещается на встраиваемой платформе, то к алгоритму обнаружения движений предъявляется ряд требований, направленных на сохранение функционала при ограничении вычислительной мощности. В связи с этим, актуальным является разрешение противоречий между вычислительной сложностью алгоритмов обработки, точностью и аппаратными возможностями современной техники [6].

Таким образом, цель исследования заключалась в разработке и тестировании программного обеспечения для обнаружения движений, обладающего низкими требованиями к вычислительным ресурсам и при этом функциональными характеристиками, достаточными для работы системы видеонаблюдения в режиме реального времени.

Для реализации указанной цели были выполнены задачи, связанные с анализом аналогичных решений, разработкой на их основе собственного алгоритма обнаружения движений, а также его программной реализацией и исследованием результатов.

## II. ПРИМЕНЕНИЕ ПРОГРАММНО-АППАРАТНЫХ РЕШЕНИЙ ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ

В связи с активным распространением мультимедийных данных и онлайн-наблюдений в последнее время значительно возрастает важность

<sup>1</sup>Истратова Евгения Евгеньевна, Новосибирский государственный технический университет, istratova@mail.ru

Смирнова Арина Олеговна, Новосибирский государственный технический университет, smar979@yandex.ru

исследований в области компьютерного зрения, анализа и автоматического распознавания содержимого видеопотоков. При этом обнаружение движений определенного объекта в массиве видеоданных является ключевым предметом исследования в науке о компьютерном зрении, которая работает с рядом приложений, среди которых используются интеллектуальные системы видеонаблюдения, системы для индексирования видео и поиска мультимедиа [7].

Однако, несмотря на большое количество исследований, посвященных этой тематике, до сих пор существует необходимость в более детальном изучении программно-аппаратных инструментов реализации данного подхода. Так, для поддержки высокопроизводительных приложений все чаще применяются современные встраиваемые системы, что обусловлено не только их распространением в сфере мобильных устройств, но и потребностью в большом количестве услуг, реализуемых с их помощью. Обработка изображений — одна из основных функций в приложениях, устанавливаемых на встраиваемых системах, которая требует значительных вычислительных ресурсов без потерь производительности [8].

В литературных источниках приведены примеры применения программно-аппаратных решений для встраиваемых систем. Так, в работе [9] была предложена методология совместного проектирования процессор-ориентированных встраиваемых систем с аппаратным ускорением. Готовый программный продукт обрабатывает изображения размером 1600\*1200 пикселей со скоростью 32 кадра в секунду с расчетным потреблением энергии 17 мДж на кадр, что обеспечивает 16-кратное ускорение и 92-процентное энергосбережение по сравнению с существующими встроенными программными решениями. Данный результат доказывает полезность предлагаемой методологии для встроенных приложений обработки изображений в режиме реального времени.

Таким образом, обзор программно-аппаратных решений для встраиваемых систем позволил выделить основные характеристики данных платформ, оказывающие влияние на работу визуальных датчиков и видеокamer. Такими характеристиками являются следующие: уровень интеграции, необходимое дополнительное оборудование для обработки данных, уровень рассеивания энергии, применяемые радио и операционные системы.

### III. ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ДВИЖЕНИЙ НА УСТРОЙСТВАХ С ОГРАНИЧЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ МОЩНОСТЬЮ

Одной из основных целей мультимедиа, связанных с обработкой изображений и видео, является обеспечение сверхвысокого разрешения изображения в реальном времени или визуально приятного изображения с высоким разрешением на основе последовательностей изображений с низким разрешением. Изображения с высоким разрешением состоят из более высокой плотности пикселей с более мелкими и точными

деталью по сравнению с изображениями или видео с низким разрешением. Многие связанные приложения, такие как видеонаблюдение, телевидение сверхвысокой четкости, распознавание лиц с низким разрешением и дистанционное считывание изображений, основаны на методах сверхвысокого разрешения. Эти методы вызвали большой интерес как в научных кругах, так и в промышленности, и в настоящее время являются областью активных исследований в сфере обработки изображений и видео.

Вычитание фона — это очень важная задача обработки видео, цель которой — отделить передний план от видео, чтобы сделать задачи постобработки более эффективными. До сих пор для решения данной задачи было предложено несколько различных методов, но большинство из них не могут хорошо работать с видео одновременно на переднем и на заднем плане. В статье [10] предлагается метод вычитания фона, который направлен на постепенную подгонку определенного подпространства для фона, полученного в результате регуляризации матрицы низкого ранга  $L1$  с использованием циклического взвешенного медианного алгоритма и определенного распределения смеси Гауссовых шумов для переднего плана. Алгоритм максимизации ожидания применяется для оптимизации смешанной модели Гаусса. Кроме того, для устранения эффекта дрожания камеры используется оператор аффинного преобразования для выравнивания последовательных кадров. На основании сопоставления производительности предложенного метода с другими был сделан вывод, что разработанный алгоритм показывает лучший результат с точки зрения F-меры и вычислительной сложности.

Поскольку количество мест, наблюдаемых камерами, постоянно увеличивается, естественной тенденцией является отказ от участия человека в системах видеонаблюдения и разработка полностью автоматических устройств видеонаблюдения. Хотя относительная важность низкоуровневой обработки изображений может варьироваться от одной системы к другой, вычислительный вес низкоуровневых операторов, как правило, высок, поскольку они включают большой объем данных. Таким образом, способность систем видеонаблюдения обнаруживать соответствующее событие тесно связана с выполнением определенных функций обработки изображений. Таким этапом обработки является обнаружение движения, цель которого заключается в разделении пикселей каждого кадра последовательности изображений на два класса: фон, соответствующий пикселям, принадлежащим статической сцене (метка 0), и передний план, соответствующий пикселям, принадлежащим к движущемуся объекту (метка 1). Алгоритм обнаружения движения должен как можно точнее отличать движущиеся объекты от фона, не будучи слишком чувствительным к размерам и скоростям объектов или к изменяющимся условиям статической сцены. При этом для обеспечения длительной автономности и конфиденциальности система не должна потреблять слишком много вычислительных ресурсов. Именно

поэтому обнаружение движения обычно является наиболее требовательной к вычислительным ресурсам функцией системы видеонаблюдения [10].

В особенности это касается процесса определения оптического потока видео, который является ресурсоемкой задачей, необходимой для компьютерного зрения. Для выполнения обработки в режиме реального времени необходимо сначала проанализировать всю цепочку развертывания алгоритма с точки зрения эффективности. Поскольку реализация метода делится на две части: разработка алгоритма, отвечающего ограничениям точности; реализация и оптимизация его выполнения на целевой платформе. Было установлено, что объединение этих операций повышает производительность встроенного процессора. В итоге при бюджете мощности 30 Вт приведенный метод способен работать со скоростью 60 FPS, на изображениях размером 640\*512. Это позволяет судить о возможности переноса данного метода на параллельную платформу для выполнения алгоритмов компьютерного зрения на специализированном оборудовании [11].

Сети визуальных датчиков требуют методов, рассчитанных на небольшие вычислительные ресурсы, для сжатия большого количества видеоданных в каждом узле камеры из-за ограниченной энергии и ограниченной полосы пропускания. Поэтому необходима разработка энергоэффективной архитектуры для оценки движения [12].

В статье [13] приведен пример подобной высокопроизводительной, масштабируемой и унифицированной архитектуры для вычисления операций преобразования в видеокodeках для применения во встраиваемых системах. Также данную структуру можно использовать в качестве аппаратного ускорителя в современных встроенных системах для эффективного вычисления всех двумерных преобразований 4\*4 и 2\*2 стандарта H.264/AVC. Результаты показали, что при интеграции в многоядерную встраиваемую систему эта архитектура обеспечивает коэффициенты ускорения выше примерно в 120 раз по сравнению с чисто программными реализациями алгоритмов преобразования, что позволяет выполнять вычисления в реальном времени.

В исследовании [14] предложен еще один эффективный с точки зрения энергопотребления метод повышения точности видеонаблюдения в режиме реального времени. Данный метод основан на архитектуре граничных вычислений и развертывается в пограничном облаке, что повышает производительность при работе в режиме реального времени. Чтобы идентифицировать выполняемое действие, предложенная сквозная модель распознавания действий использует временную сверточную нейронную сеть (TCN) для извлечения локальных временных признаков и слой Gate Recurrent Unit (GRU) для извлечения глобальных временных признаков. Это повышает точность распознавания фрагментов действия. Эксперименты показали, что предложенный метод имеет хорошую производительность при работе в

режиме реального времени, а также высокую точность распознавания сегментов действий.

Еще одним важным аспектом при разработке алгоритмов обнаружения движений для встраиваемых систем, характеризующихся небольшими вычислительными ресурсами, является применение механизмов сегментации. Это связано с тем, что при обработке видео первым шагом является его сегментация на физические единицы, обычно называемые кадрами. Таким образом, кадр — это видеофрагмент, состоящий из одного непрерывного действия. Как правило, эти физические единицы необходимо сгруппировать, чтобы сформировать более семантически значимые единицы, такие как: сцены, последовательности, программы и т.д., что представляет собой структурирование видео на основе сюжета. Автоматическое структурирование видео имеет большое значение для его просмотра и обнаружения движений в нем [15].

На основе данного подхода также целесообразна разработка алгоритмов обнаружения движений для встраиваемых систем. Так, в статье [16] представлен метод сегментации видеообъектов, который использует границы движения и карты уровня класса на основе сверточной нейронной сети (CNN) для выполнения совместной сегментации кадров. Ключевой характеристикой предлагаемого подхода является сочетание этих двух источников информации для создания начальных объектов и фоновых областей. Данные области используются в рамках функции энергии косегментации. Карта границ движения определяет области, содержащие движение объекта, а карта заметности классов на основе CNN определяет области, оказывающие большее влияние на получение правильной классификации сети. Предлагаемый подход может быть реализован на неограниченных естественных видео, которые включают изменения внешнего вида объекта, быстро движущийся фон, деформацию объекта при нежестком движении, быстрое движение камеры и даже наличие статического объекта.

Таким образом, в ходе работы были исследованы и классифицированы различные методы и алгоритмы обнаружения движений, применяемые для различных сценариев видеонаблюдения. На основе анализа литературных источников были сделаны выводы о том, что наиболее актуальным и целесообразным для встраиваемых систем является применение алгоритмов, основанных на вычитании фона с граничными вычислениями, поскольку такой вариант позволяет работать в режиме реального времени с минимальными требованиями к вычислительным ресурсам оборудования. Также было установлено, что обнаружение движений с использованием периферийных устройств менее изучено и требует значительных изысканий для достижения современного уровня качества обнаружения объектов в системах интеллектуального видеонаблюдения.

#### IV. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ ДВИЖЕНИЙ

Для определения требований и ограничений при разработке программного обеспечения для обнаружения движений были рассмотрены условия эксплуатации системы видеонаблюдения. Так, в соответствии с данными эксплуатационными характеристиками, планируется, что проектируемая система интеллектуального видеонаблюдения будет установлена на встраиваемых системах и должна работать в режиме реального времени. Исходя из данных критериев, основными требованиями к разрабатываемому алгоритму будут следующие: низкая потребность в вычислительных ресурсах; высокая точность обнаружения движений; высокий уровень надежности системы; высокая производительность системы.

Наиболее оптимальным вариантом для реализации указанных требований является применение алгоритмов, основанных на вычитании фона, которые позволяют достаточно точно обнаруживать движения при работе в режиме реального времени, предъявляя при этом минимальные требования к вычислительным ресурсам используемого оборудования встраиваемых систем.

Для разработки собственного алгоритма были рассмотрены и проанализированы существующие аналоги.

Поскольку в качестве объекта исследования в работе применяется видеопоток, а не статичное изображение, то целесообразно предварительно выявить характерные для него особенности.

Видеопоследовательность содержит сильную пространственно-временную корреляцию между областями последовательных кадров. Наиболее важной характеристикой видео является воспринимаемое движение объектов переднего плана по кадрам. Движение объектов переднего плана резко меняет важность объектов в сцене и приводит к другой карте значимости кадра, представляющего сцену. Это делает анализ видео намного сложнее, чем анализ неподвижных изображений.

В статье [17] была предложена новая пространственно-временная модель оценки значимости видеопоследовательности, предназначенная для применения в системах видеонаблюдения. По сравнению с классическими моделями оценки видеопоследовательности, основанными на неподвижных изображениях, такими как: модель Итти и модели пространственно-временной значимости, предлагаемая модель больше коррелирует с визуальным восприятием значимости видеонаблюдения. В этой модели задействованы как восходящие, так и нисходящие механизмы, соответственно, анализируются стационарная заметность и значимость движения. В рамках работы был предложен новый алгоритм вычитания фона и выделения переднего плана, основанный на контент-анализе сцены в области видеонаблюдения.

Действия в реальном мире часто происходят в

многолюдных, динамичных средах. Это создает сложную проблему для современных подходов к обнаружению видеособытий, поскольку трудно отделить объект от фона из-за того, что движение отвлекает от других объектов в сцене. Авторами исследования [18] был предложен метод распознавания событий в многолюдных видео, который надежно идентифицирует действия при наличии частичной окклюзии и фоновых помех. Указанный подход был основан на трех ключевых идеях: сопоставление объемного представления события с пересегментированными пространственно-временными объемами видео; добавление к функции, основанной на форме, данных о видеопотоке; раздельное сравнение частей видео как в пространстве, так и во времени для обеспечения устойчивости к окклюзиям и изменчивости объектов. Проведенные эксперименты подтвердили точность обнаружения движений с небольшим количеством ложных срабатываний.

Большое разнообразие видеоконтента делает обработку различных движений, краев, текстур и их комбинаций с помощью одного алгоритма очень сложной. Поэтому в статье [19] авторами был представлен новый адаптивный к движению алгоритм с распознаванием контуров и гибридным обнаружением движения. Алгоритм распознавания краевых структур продемонстрировал гибкость в обработке как текстур, так и краев, которые выполняются отдельно с помощью усреднения линий. Кроме того, прогнозирование соседних пикселей для анализа образов и интерполяции дополнительно повысило адаптивность блока распознавания образов краев при включенном режиме обнаружения движения.

Обнаружение объектов на динамическом фоне — сложная задача для многих приложений компьютерного зрения. В некоторых ситуациях движение объектов можно предсказать, благодаря его регулярности, например, движение транспортных средств, движение пешеходов. В исследовании [20] был предложен метод выделения движущихся объектов переднего плана в последовательностях, снятых мобильной камерой. При этом была использована разность кадров с компенсацией движения камеры, дополненная оценкой функции плотности вероятности фоновых пикселей. Так как оценка была основана на ограниченном числе измерений; то были применены специальный пространственно-временной выбор точки выборки и метод адаптивной пороговой обработки. Объекты переднего плана были построены при помощи алгоритма DBSCAN из обнаруженных пикселей переднего плана.

При автоматической обработке видеопотоков с целью обнаружения движений возникают такие ограничивающие факторы, как: сильное изменение яркости, ограниченная видимость, сложный фон, свободное движение объекта. Для снижения негативного влияния данных факторов в статье [21] был предложен подход, объединяющий в себе два метода, которые используют разные потоки сверточной нейронной сети для объединения информации о

внешнем виде и движении для автоматического обнаружения объекта. Эти методы состоят из двух моделей Faster R-CNN, которые используют либо один и тот же классификатор. Наилучший показатель точности обнаружения составил 83,16%.

#### V. РАЗРАБОТКА АЛГОРИТМА ДЛЯ ОБНАРУЖЕНИЯ ДВИЖЕНИЙ

Разработка алгоритма осуществлялась с учетом таких требований, предъявляемых к программным решениям, реализуемым на встраиваемых системах, как: низкая потребность в вычислительных ресурсах; высокая точность обнаружения движений; высокая производительность системы и возможность работы с несколькими потоками камер одновременно.

Логика работы алгоритма заключается в отправке сообщения и записи об этом в лог-файлы при обнаружении движения. Данные действия осуществляются в заданной конфигурации, после чего с определенной периодичностью лог-файлы архивируются на диск.

Операции чтения и декодирования занимают значительную часть времени, поскольку перед обработкой кадра его необходимо полностью считать и декодировать. Для уменьшения задержки было принято решение вынести эти операции в отдельные потоки. В результате кадры считываются и декодируются в одном потоке, а затем обрабатываются в другом видеопотоке. Схема алгоритма для обработки видеопотока, в котором считываются и декодируются кадры, приведена на рис. 1.

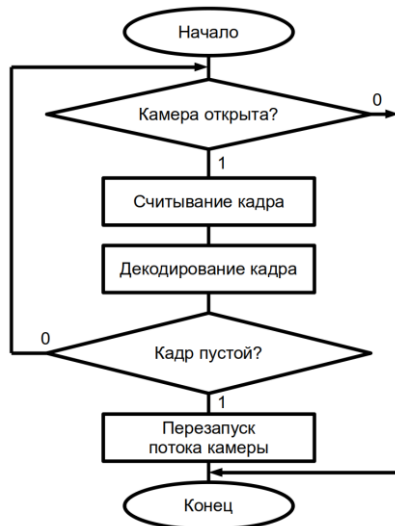


Рисунок 1 — Блок-схема процессов считывания и декодирования кадров видеопотока

Согласно блок-схеме, первым этапом обработки кадров является проверка видеокamer на предмет их работоспособности. Если видеокamera доступна, то осуществляется сначала считывание кадра, а потом его декодирование. В противном случае процесс завершается и выдает сообщение о том, что видеокamera закрыта. Информация записывается в виде лог-файлов и становится доступной для анализа подключения в системе видеонаблюдения. После считывания и декодирования кадров производится их

проверка. В случае, если кадр оказывается пустым, алгоритм выполняет перезапуск видеопотока, идущего от конкретной видеокamerы. Если кадр оказывается корректным, то его содержимое направляется для дальнейшей обработки на следующий этап для обнаружения движения.

Перед обнаружением движения изображение переводится в оттенки серого для сокращения времени выполнения дальнейших операций.

Для снижения уровня шумов была использована морфологическая операция по уменьшению размера изображения, включающая такие стадии, как эрозия и размытие.

Поскольку обнаруживать движение необходимо только в выбранных пользователем ячейках, изображение необходимо разбить и все дальнейшие действия выполнять для каждой ячейки отдельно. Наиболее оптимальным вариантом является разбивка изображения на фрагменты размером 32\*32.

Для обнаружения движения был применен метод вычитания фона. На первом шаге была определена начальная фоновая модель, а на втором этапе эта модель была обновлена для адаптации к возможным изменениям в сцене. В алгоритме фоновая модель строится на основании истории предыдущих кадров. Увеличение количества кадров позволяет снизить уровень шумов, но увеличивает уровень потребляемых вычислительных ресурсов. Эмпирическим путем было выявлено, что оптимальное количество кадров для формирования фоновой модели должно быть равно 50.

Следующим этапом алгоритма является операция вычитания фона, которая позволяет создать маску переднего плана, представляющую собой двоичное изображение с движущимися объектами в сцене. Для вычисления маски переднего плана выполняется вычитание между текущим кадром и фоновой моделью. В данном алгоритме использовался алгоритм вычитания на основе квадрата расстояния Махаланобиса. Высокое пороговое значение дает отсеивать шумы, не учитывая объекты, цвет которых слабо отличается от цвета фона, что делает алгоритм менее чувствительным при обнаружении движений. В качестве порогового значения использовалось значение по умолчанию, равное 16.

Далее с помощью метода простой цепной аппроксимации был определен внешний контур, ограничивающий движущиеся объекты. Движение считается существенным, если площадь внешнего контура больше, чем половина наименьшей допустимой ячейки (32\*32).

В алгоритме предусмотрено три события движения: начало, окончание и процесс. При обнаружении начала или окончания движения на видеосервер отправляются соответствующие сообщения. Также информация о обнаруженном движении фиксируется в журнале в виде лог-файлов. С заданной в конфигурации периодичностью информация из журнала движения архивируется и отправляется на резервное хранение. После архивирования лог-файлов выполняется проверка архивов на максимально допустимый размер и время

хранения.

Кроме того было решено сократить количество обрабатываемых кадров в секунду, что позволило значительно уменьшить скорость работы алгоритма, и практически не повлияло на его точность. Эмпирическим путем было выявлено, что оптимальным значением скорости обработки кадров в секунду является значение, равное 10 кадрам в секунду.

Таким образом, отличительными особенностями разработанного алгоритма является не только возможность работы с несколькими потоками камер одновременно в режиме реального времени, но и минимизация потребности в применяемых вычислительных ресурсах.

#### VI. РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ ДВИЖЕНИЙ

Разработка программного обеспечения осуществлялась при использовании языка программирования C++ и двух библиотек: OpenCV (Open Source Computer Vision Library) для работы с алгоритмами компьютерного зрения и ZeroMQ для обмена сообщениями в распределенных или параллельных приложениях.

Язык C++ наиболее часто применяется при разработке встраиваемых систем, выступая между программным и аппаратным обеспечением более высокого уровня, тем самым, обеспечивая доступ к оборудованию и управлению им напрямую. Данное решение обладает высокой производительностью, простотой в использовании и экономичностью с точки зрения распределения ресурсов. Все это в результате позволяет выполнять любые программы на встраиваемых системах с высокой скоростью в режиме реального времени.

Для работы с видео в режиме реального времени была выбрана библиотека алгоритмов компьютерного зрения с открытым исходным кодом OpenCV (Open Source Computer Vision Library), реализованная на языке программирования C++. Данный программный продукт имеет высокую производительность, низкое потребление ресурсов оперативной памяти, поддерживается практически на любом устройстве.

Для обмена сообщениями между модулями видеосервера применялась ZeroMQ — высокопроизводительная асинхронная библиотека обмена сообщениями, предназначенная для использования в распределенных или параллельных приложениях. Это обусловлено тем, что ZeroMQ поддерживает общие шаблоны обмена сообщениями, применяя различные протоколы, что в результате делает межпроцессный обмен сообщениями таким же простым, как и межпоточный. Данная библиотека также написана на языке программирования C++.

Разработанное программное обеспечение было исследовано на предмет обнаружения движений на нескольких потоках камер системы видеонаблюдения. При этом рассматривались разбиения по умолчанию и максимально возможное значение. Наиболее оптимальным оказалось подключение, состоящее не

более чем их четырех камер. В этом случае система видеонаблюдения может высвободить достаточно ресурсов, в том числе на решение других задач. Результаты исследования приведены в табл. 1.

В результате изучения программных аналогов для оценки качества предложенных алгоритмов было проведено их сравнение по показателям точности обнаружения движений и времени обработки кадров видеопоследовательности с разработанным программным обеспечением. Результаты представлены в табл. 2. Исходя из результатов сравнительного анализа, можно сделать вывод о том, что все указанные модели могут быть использованы для обнаружения движений на встраиваемых системах в режиме реального времени, показывая при этом хорошую точность и отличную скорость обработки кадров.

Таблица 1. Сравнение производительности модуля обнаружения движения

Количество потоков камер	Процент неиспользованных ресурсов процессора	
	10 x 10	32 x 32
0	95-98	95-98
1	76-79	76-79
2	69-73	65-68
3	58-61	50-53
4	42-48	31-38
5	32-36	18-23
6	19-23	9-11

Таблица 2. Результаты сравнительного анализа разработанного программного обеспечения с аналогами

Ссылка на источник	Точность, %	Время обработки, мс
[17]	88,43	24
[18]	88,52	25
[19]	77,46	13
[20]	78,85	14
[21]	83,16	17
Разработанная программа	88,31	20

#### VII. ЗАКЛЮЧЕНИЕ

Таким образом, в результате выполнения программной реализации было разработано программное обеспечение для обнаружения движений, позволяющее для всех потоков камер, на которых было зафиксировано движение, выполнять следующие действия над параметрами:

- 1) задавать время хранения и максимальный размер данных лог-файлов в резервном хранилище;
- 2) устанавливать промежуток времени, через который необходимо архивировать данные;
- 3) определять максимальное время непрерывной записи события;
- 4) вычислять число потоков, к которым необходимо подключиться для обнаружения движения при подключении новой камеры;
- 5) ограничивать максимальное количество одновременно открытых потоков.

В результате, готовый программный продукт

позволяет задавать параметры потоков при добавлении камеры, а также в последующем изменять их. Редактирование значений по умолчанию и глобальных параметров также было предусмотрено в предложенном решении. Помимо этого, разработанное программное обеспечение поддерживает ряд запросов, позволяющих получать список временных непрерывных диапазонов для лог-файлов, а также удалять или запрашивать файлы логирования за определенный период. Это в результате приводит к повышению эффективности процесса обнаружения движений в системе интеллектуального видеонаблюдения.

#### БИБЛИОГРАФИЯ

- [1] Проскурин А.В., Ковтуненко И.И. Отбор информативных снимков из последовательности на основе детекции движения объектов // Решетневские чтения. — 2018. — № 2. — С. 287-288.
- [2] Истратова Е.Е., Бухамер Е.А., Томилов И.Н. Разработка комбинированного метода детекции движений для системы интеллектуального видеонаблюдения // International Journal Of Open Information Technologies. — 2022. — №1. — С. 54-60.
- [3] Abbas Q., Ibrahim M.E., Jaffar M.A. A comprehensive review of recent advances on deep vision systems // Artif Intell Rev 52, 39–76 (2019). <https://doi.org/10.1007/s10462-018-9633-3>.
- [4] Molina-Cabello M.A., García-González J., Luque-Baena R.M. The effect of downsampling–upsampling strategy on foreground detection algorithms // Artif Intell Rev 53, 4935–4965 (2020). <https://doi.org/10.1007/s10462-020-09811-y>.
- [5] Lee G., Wang M.J., Li H.T. A motion-adaptive deinterlacer via hybrid motion detection and edge-pattern recognition // Image Video Proc. — 741290 (2008). <https://doi.org/10.1155/2008/741290>.
- [6] Нокеева Р.М. Разработка программного обеспечения для оптимальной работы видеорегистратора // Научные исследования. — 2019. — №3 (29). — С. 15-19.
- [7] Mohtavipour, S.M., Saeidi, M. & Arabsorkhi, A. A multi-stream CNN for deep violence detection in video sequences using handcrafted features. Vis Comput 38, 2057–2072 (2022). <https://doi.org/10.1007/s00371-021-02266-4>.
- [8] Chillet, D., Hübner, M. Special issue on design and architectures of real-time image processing in embedded systems. J Real-Time Image Proc 9, 1–3 (2014). <https://doi.org/10.1007/s11554-014-0401-6>.
- [9] Pedre, S., Krajník, T., Todorovich, E. et al. Accelerating embedded image processing for real time: a case study. J Real-Time Image Proc 11, 349–374 (2016). <https://doi.org/10.1007/s11554-013-0353-2>.
- [10] Jeon, G., Chetri, A. Special issue on deep learning for emerging embedded real-time image and video processing systems. J Real-Time Image Proc 18, 1167–1171 (2021). <https://doi.org/10.1007/s11554-021-01156-1>.
- [11] Lacassagne, L., Manzanera, A., Denoulet, J. et al. High performance motion detection: some trends toward new embedded architectures for vision systems. J Real-Time Image Proc 4, 127–146 (2009). <https://doi.org/10.1007/s11554-008-0096-7>.
- [12] Sez nec, M., Gac, N., Orioux, F. et al. Real-time optical flow processing on embedded GPU: an hardware-aware algorithm to implementation strategy. J Real-Time Image Proc 19, 317–329 (2022). <https://doi.org/10.1007/s11554-021-01187-8>.
- [13] Thevenin, M., Paindavoine, M., Schmit, R. et al. A templated programmable architecture for highly constrained embedded HD video processing. J Real-Time Image Proc 16, 143–160 (2019). <https://doi.org/10.1007/s11554-018-0808-6>.
- [14] Dias, T., López, S., Roma, N. et al. Scalable Unified Transform Architecture for Advanced Video Coding Embedded Systems. Int J Parallel Prog 41, 236–260 (2013). <https://doi.org/10.1007/s10766-012-0221-x>.
- [15] Pal, S.K., Bhoumik, D. & Bhunia Chakraborty, D. Granulated deep learning and Z-numbers in motion detection and object recognition. Neural Comput & Applic 32, 16533–16548 (2020). <https://doi.org/10.1007/s00521-019-04200-1>.
- [16] Xiong, W., Lee, C.M. & Ma, R.H. Automatic video data structuring through shot partitioning and key-frame computing. Machine Vision and Applications 10, 51–65 (1997). <https://doi.org/10.1007/s001380050059>.
- [17] Kamranian, Z., Naghsh Nilchi, A.R., Sadeghian, H. et al. Joint motion boundary detection and CNN-based feature visualization for video object segmentation. Neural Comput & Applic 32, 4073–4091 (2020). <https://doi.org/10.1007/s00521-019-04448-7>.
- [18] Yubing, T., Cheikh, F.A., Guraya, F.F.E. et al. A Spatiotemporal Saliency Model for Video Surveillance. Cogn Comput 3, 241–263 (2011). <https://doi.org/10.1007/s12559-010-9094-8>.
- [19] Lee, G., Wang, M.J., Li, H.T. et al. A Motion-Adaptive Deinterlacer via Hybrid Motion Detection and Edge-Pattern Recognition. J Image Video Proc 2008, 741290 (2008). <https://doi.org/10.1155/2008/741290>.
- [20] Ali, I., Mille, J. & Tougne, L. Adding a rigid motion model to foreground detection: application to moving object detection in rivers. Pattern Anal Applic 17, 567–585 (2014). <https://doi.org/10.1007/s10044-013-0346-6>.
- [21] Szolgay, D., Benois-Pineau, J., Megret, R. et al. Detection of moving foreground objects in videos with strong camera motion. Pattern Anal Applic 14, 311–328 (2011). <https://doi.org/10.1007/s10044-011-0221-2>.

**Истратова Евгения Евгеньевна.** Новосибирский государственный технический университет, г. Новосибирск, Россия. Кандидат технических наук, доцент кафедры автоматизированных систем управления. Количество печатных работ: 107. Область научных интересов: информационные технологии, информационные системы, системы компьютерного зрения. e-mail: [istratova@mail.ru](mailto:istratova@mail.ru) (Ответственная за переписку).

**Смирнова Арина Олеговна.** Новосибирский государственный технический университет, г. Новосибирск, Россия. Магистрант факультета автоматизации и вычислительной техники. Количество печатных работ: 2. Область научных интересов: системы компьютерного зрения, искусственный интеллект, информационные технологии.

# Development and research of software to improve the efficiency of motion detection in the intelligent video surveillance system

E.E. Istratova, A.O. Smirnova

**Abstract**—The article presents the results of the development and testing of software for motion detection, which has low requirements for computing resources and, at the same time, functional characteristics sufficient for the operation of an intelligent video surveillance system in real time. In the course of the work, the features of the application of software and hardware solutions and motion detection algorithms on devices with limited computing power were considered, and software requirements were determined. As a result of the software implementation, a motion detection solution was developed, a distinctive feature of which is the ability to work with multiple camera streams simultaneously in real time. The software can be installed on embedded systems and used to solve problems of monitoring and security of premises and territories. The finished software product allows you to set stream parameters when adding a camera, as well as subsequently change them, it also provides for editing the default values and global parameters of camera streams, which improves the efficiency of the motion detection process in intelligent video surveillance systems.

**Keywords**—motion detection, intelligent video surveillance, algorithm, background subtraction, software, processing.

## REFERENCES

- [1] Proskurin A.V., Kovtunenkov I.I. Selection of informative images from a sequence based on object motion detection // Reshetnevskiy readings. - 2018. - no. 2. - S. 287-288.
- [2] Istratova E.E., Bukhamer E.A., Tomilov I.N. Development of a combined motion detection method for an intelligent video surveillance system // International Journal Of Open Information Technologies. - 2022. - No. 1. - S. 54-60.
- [3] Abbas Q., Ibrahim M.E., Jaffar M.A. A comprehensive review of recent advances on deep vision systems // Artif Intell Rev 52, 39–76 (2019). <https://doi.org/10.1007/s10462-018-9633-3>.
- [4] Molina-Cabello M.A., García-González J., Luque-Baena R.M. The effect of downsampling–upsampling strategy on foreground detection algorithms // Artif Intell Rev 53, 4935–4965 (2020). <https://doi.org/10.1007/s10462-020-09811-y>.
- [5] Lee G., Wang M.J., Li H.T. A motion-adaptive deinterlacer via hybrid motion detection and edge-pattern recognition // Image Video Proc. — 741290 (2008). <https://doi.org/10.1155/2008/741290>.
- [6] Nokeeva R.M. Development of software for the optimal operation of the video recorder // Scientific research. - 2019. - No. 3 (29). - S. 15-19.
- [7] Mohtavipour, S.M., Saeidi, M. & Arabsorkhi, A. A multi-stream CNN for deep violence detection in video sequences using handcrafted features. Vis Comput 38, 2057–2072 (2022). <https://doi.org/10.1007/s00371-021-02266-4>.
- [8] Chillet, D., Hübner, M. Special issue on design and architectures of real-time image processing in embedded systems. J Real-Time Image Proc 9, 1–3 (2014). <https://doi.org/10.1007/s11554-014-0401-6>.
- [9] Pedre, S., Krajník, T., Todorovich, E. et al. Accelerating embedded image processing for real time: a case study. J Real-Time Image Proc 11, 349–374 (2016). <https://doi.org/10.1007/s11554-013-0353-2>.
- [10] Jeon, G., Chehri, A. Special issue on deep learning for emerging embedded real-time image and video processing systems. J Real-Time Image Proc 18, 1167–1171 (2021). <https://doi.org/10.1007/s11554-021-01156-1>.
- [11] Lacassagne, L., Manzanera, A., Denoulet, J. et al. High performance motion detection: some trends toward new embedded architectures for vision systems. J Real-Time Image Proc 4, 127–146 (2009). <https://doi.org/10.1007/s11554-008-0096-7>.
- [12] Seznec, M., Gac, N., Orioux, F. et al. Real-time optical flow processing on embedded GPU: an hardware-aware algorithm to implementation strategy. J Real-Time Image Proc 19, 317–329 (2022). <https://doi.org/10.1007/s11554-021-01187-8>.
- [13] Thevenin, M., Paindavoine, M., Schmit, R. et al. A templated programmable architecture for highly constrained embedded HD video processing. J Real-Time Image Proc 16, 143–160 (2019). <https://doi.org/10.1007/s11554-018-0808-6>.
- [14] Dias, T., López, S., Roma, N. et al. Scalable Unified Transform Architecture for Advanced Video Coding Embedded Systems. Int J Parallel Prog 41, 236–260 (2013). <https://doi.org/10.1007/s10766-012-0221-x>.
- [15] Pal, S.K., Bhounik, D. & Bhunia Chakraborty, D. Granulated deep learning and Z-numbers in motion detection and object recognition. Neural Comput & Applic 32, 16533–16548 (2020). <https://doi.org/10.1007/s00521-019-04200-1>.
- [16] Xiong, W., Lee, C.M. & Ma, R.H. Automatic video data structuring through shot partitioning and key-frame computing. Machine Vision and Applications 10, 51–65 (1997). <https://doi.org/10.1007/s001380050059>.
- [17] Kamranian, Z., Naghsh Nilchi, A.R., Sadeghian, H. et al. Joint motion boundary detection and CNN-based feature visualization for video object segmentation. Neural Comput & Applic 32, 4073–4091 (2020). <https://doi.org/10.1007/s00521-019-04448-7>.
- [18] Yubing, T., Cheikh, F.A., Guraya, F.F.E. et al. A Spatiotemporal Saliency Model for Video Surveillance. Cogn Comput 3, 241–263 (2011). <https://doi.org/10.1007/s12559-010-9094-8>.
- [19] Lee, G., Wang, M.J., Li, H.T. et al. A Motion-Adaptive Deinterlacer via Hybrid Motion Detection and Edge-Pattern Recognition. J Image Video Proc 2008, 741290 (2008). <https://doi.org/10.1155/2008/741290>.
- [20] Ali, I., Mille, J. & Tougne, L. Adding a rigid motion model to foreground detection: application to moving object detection in rivers. Pattern Anal Applic 17, 567–585 (2014). <https://doi.org/10.1007/s10044-013-0346-6>.
- [21] Szolgay, D., Benois-Pineau, J., Megret, R. et al. Detection of moving foreground objects in videos with strong camera motion. Pattern Anal Applic 14, 311–328 (2011). <https://doi.org/10.1007/s10044-011-0221-2>.