

# АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБНАРУЖЕНИЯ ДВИЖЕНИЯ НА ВИДЕО

А.С. Адушкина, М.А.Кудрина

Самарский государственный аэрокосмический университет им. академика С.П. Королёва  
(национально исследовательский университет)

Разработана автоматизированная система обнаружения движения на видео методом межкадровой разницы и методом базового кадра. Реализованы алгоритмы предобработки кадра для устранения шума – алгоритмы усреднения, медианной фильтрации, фильтр Гаусса, алгоритмы обнаружения движения, основанные на межкадровой разнице и построении базового кадра.

## Введение

За последние годы видеонаблюдение становится неотъемлемой частью комплексной системы безопасности объектов, поскольку современное оборудование позволяет не только наблюдать и записывать видео, но и программировать реакцию всей системы безопасности на возникновение тревожных событий.

Можно построить достаточно надежную охранную систему, которая будет, например, дистанционно следить за припаркованным под окнами автомобилем или выдавать предупредительный сигнал при обнаружении посторонних в охраняемой зоне (входная дверь, коридор, лестничный пролет и т.п.).

Такую задачу можно решить посредством алгоритмов детектирования наличия движения, для надежности сделав их нечувствительными к периодическому и хаотическому движению, к засветке и движению с малой или большой скоростью.

## Теоретическая часть

*Система обнаружения движения на видео (или детектор движения)* — это программный модуль, предназначенный для обнаружения перемещающихся в поле зрения web-камеры объектов [1].

Процесс обнаружения движения на видео включает в себя следующие этапы: предобработка кадров, выполнение алгоритма обнаружения движения, выполнение алгоритма выделения движущихся объектов в случае обнаружения движения.

Основной задачей этапа предобработки растровых данных является подготовка изображений к обнаружению на них движущихся объектов. Для корректного обнаружения движения необходимо минимизировать влияние шума [2].

*Алгоритм усреднения для шумоподавления изображения* основан на переходе к усреднённым значениям пикселей, поскольку значения компонентов цвета исходного пикселя заменяются усреднёнными значениями компонентов цвета данного пикселя и его соседей.

*Гауссово размытие или фильтр Гаусса* относится к классу линейной фильтрации изображения [3]. В основе линейной фильтрации лежит операция свёртки. В качестве растра  $a$  берётся ядро свёртки, которое представляет собой матрицу малой размерности. Обычно ядро свёртки является квадратной матрицей  $n \times n$ , где  $n$  — нечетное.

$$c[m, n] = \sum_{i=-u/2}^{u/2} \sum_{j=-u/2}^{u/2} a[i, j] \cdot b[m - i, n - j],$$

где  $c[m, n]$  – результирующий растр, полученный после свёртки,  $m, n$  – координаты  $x$  и  $y$ , в которых выполняется свёртка,  $a, b$  – исходные растры,  $j, k$  – счётчики по горизонтали и вертикали при вычислении свёртки,  $u$  – линейный размер ядра свёртки (ширина, высота).

Свёртка вычисляется для каждого пикселя исходного изображения.

*Медианный фильтр* представляет собой скользящее по изображению окно  $W$ , охватывающее нечетное число пикселей изображения [4]. Цвет текущего пикселя заменяется медианой цветов всех пикселей изображения, попавших в окно.

Алгоритм вычисления межкадровой разности двух кадров для случая обработки цветного видео в формате *RGB* выглядит следующим образом:

- 1) на вход алгоритма поступают два видео кадра;
- 2) производится вычисление попиксельных межкадровых разностей по формулам:

$$R_{res} = |R_{1i} - R_{2i}|, G_{res} = |G_{1i} - G_{2i}|, B_{res} = |B_{1i} - B_{2i}|,$$

где  $R_{res}, G_{res}, B_{res}$  - значения красной, зелёной и синей компонент цвета  $i$ -го пикселя результирующего раstra;

$R_{1i}, G_{1i}, B_{1i}, R_{2i}, G_{2i}, B_{2i}$  - значения красной, зелёной и синей компонент цвета  $i$ -го пикселя на первом и втором кадре;

- 3) для каждого пикселя вычисляется среднее значение между значениями трёх компонент цвета;

- 4) среднее значение сравнивается с заданным порогом. В результате сравнения формируется двоичная маска:

$$m_i = \begin{cases} 0, & p_i < T, \\ 1, & p_i > T, \end{cases}$$

где  $m_i$  - значение  $i$ -го элемента маски;

$p_i$  - среднее значение;

$T$  - порог сравнения, иногда называемый также порогом или уровнем чувствительности.

Работа метода основанного на построении базового кадра полностью аналогична работе алгоритма межкадровой разности с той лишь разницей, что разность вычисляется между текущим и базовым кадром.

Для построения базового кадра приближенным медианным методом [2] используется формула:

$$BF_{i,j} = \begin{cases} CF_{i,j} + \Delta, & \text{если } BF_{i,j} < CF_{i,j}, \\ CF_{i,j} - \Delta, & \text{если } BF_{i,j} > CF_{i,j}, \\ CF_{i,j}, & \text{если } BF_{i,j} = CF_{i,j}, \end{cases}$$

где  $BF_{ij}$  - яркость пикселя  $i$  строки  $j$  столбца сохранённого базового кадра,  $CF_{ij}$  - яркость пикселя  $i$  строки  $j$  столбца текущего кадра,  $\Delta$  - величина поправки.

### Программная реализация

Разработанная автоматизированная система обнаружения движения на видео может выполнять обнаружение движение несколькими алгоритмами, оповещать пользователя о тревожном событии, выделять движущийся объект и сохранять информацию в файл.

Пользователь может выбрать алгоритм обнаружения, алгоритм подавления шума, необходимость сохранения статистики.

На форму выводится текущее видео (из файла или с устройства), в случае обнаружения движения - движущийся объект выделять в рамку.

Диаграмма вариантов использования системы «Автоматизированная система обнаружения движения на видео» изображена на рисунке 1.

На диаграмме присутствует единственный актанта - пользователь. После запуска системы пользователь может запустить обнаружение движения на видео (вариант использования «Начать процесс обнаружения движения»), для этого ему необходимо настроить систему (вариант использования «Настроить систему»).

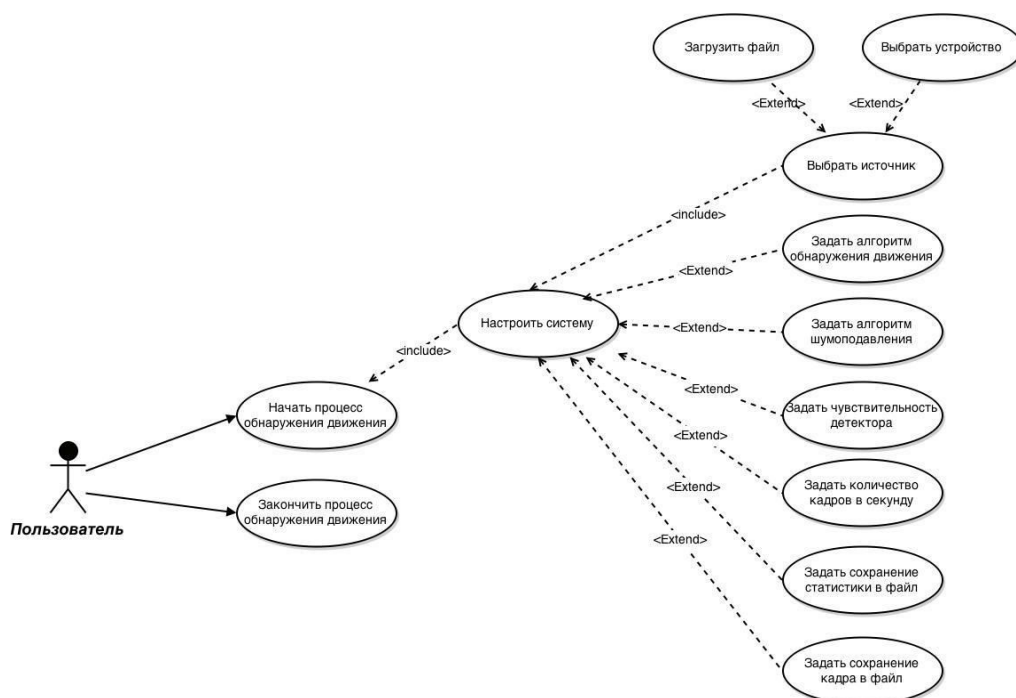


Рисунок 1 – Диаграмма вариантов использования системы

При входе пользователь видит главную экранную форму (рисунок 2). В верхней части экрана находятся следующие вкладки:

- «Видео» – выбор источников видео;
- «Алгоритмы» – выбор алгоритмов шумоподавления и детектирования движения;
- «Настройки» – открытие формы настройки параметров работы системы;
- «Справка» – открытие файла справки.

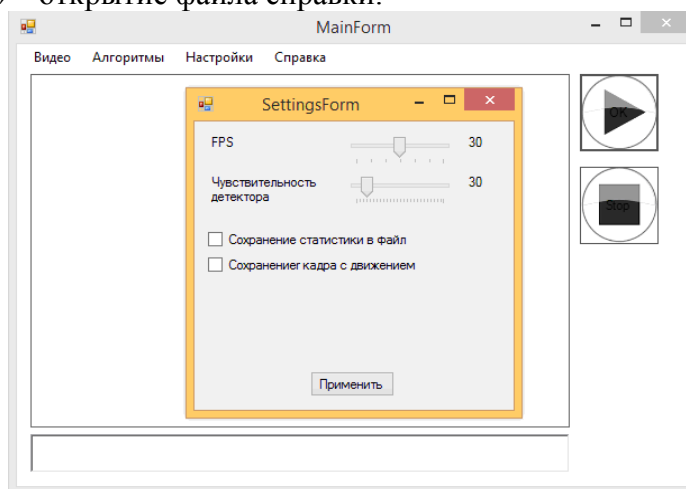


Рисунок 2 – Интерфейс разработанной системы

В правом верхнем углу находятся кнопки управления – «Старт» и «Стоп», при нажатии на которые начинается или заканчивается работа детектора.

В центре находится поле, в котором отображаются обработанные кадры. В случае обнаружения движения – кадры с выделенными красной рамкой движущиеся объекты.

Под полем отображения видео – окно статистики, в котором в случае обнаружения движения будут отображаться оповещения.

В форме «Настройки» пользователь может поменять настройки работы системы – задать *fps*, чувствительность детектора, необходимость сохранения статистики в файл и сохранения кадра с движением. При нажатии кнопки «Применить» настройки будут применены и форма «Настройки» закроется.

На рисунке 3 изображен результат работы программы.

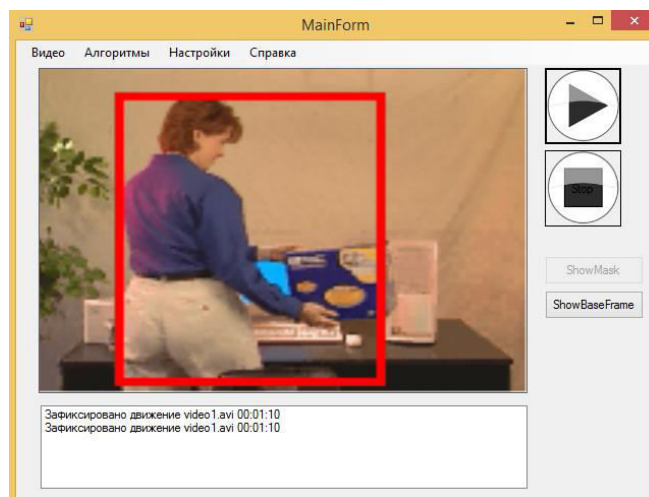


Рисунок 3 – Результат работы программы

### Экспериментальная часть

Время выполнения алгоритма обнаружения движения является важным критерием оценки эффективности работы алгоритма. Проведено исследование зависимости времени работы обработки одного кадра от типа детектора и размера кадра.

На рисунке 4 приведены графики зависимости времени работы алгоритмов, основанных на сравнении RGB компонент цвета, от размера обрабатываемого кадра.

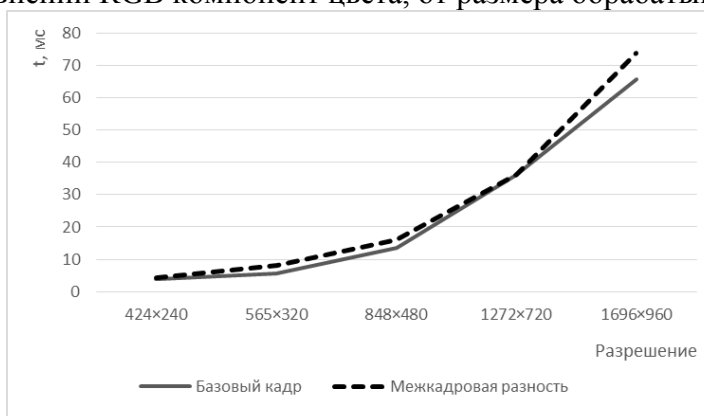


Рисунок 4 – Сравнение времени работы алгоритмов, основанных на сравнении RGB компонент цвета пикселя.

По графику на рисунке 4 можно сделать вывод, что время работы алгоритма межкадровой разности (обозначен пунктирной линией), основанном на сравнении RGB компонент цвета, немного больше, чем время работы алгоритма, основанном на построении базового кадра (обозначен сплошной линией).

На рисунке 5 приведены графики зависимости времени работы алгоритмов, основанных на сравнении яркостей пикселей, от размера обрабатываемого кадра.

По графику на рисунке 5 можно сделать вывод, что время работы алгоритма межкадровой разности (обозначен пунктирной линией), основанном на сравнении яркостей пикселей, меньше, чем время работы алгоритма, основанном на построении базового кадра (обозначен сплошной линией).

На рисунке 6 представлена диаграмма сравнения средней загрузки ЦП во время функционирования системы в зависимости от типов алгоритмов.

В результате анализа экспериментов можно сделать следующие выводы:

- 1) алгоритмы основанные на сравнении яркостных характеристик работают медленнее чем алгоритмы сравнения цвета;
- 2) алгоритм основанный на построении базового кадра не превосходит алгоритм межкадровой разности по времени выполнения, но более устойчив к возмущениям;

3) комбинируя алгоритмы можно получать различную загрузку ЦПУ. Меньшую нагрузку дает сочетание алгоритма межкадровой разности и размытия по Гауссу, но при этом увеличивается влияние шумов, частота ложных срабатываний выше, чем у более ресурсоемких алгоритмов.

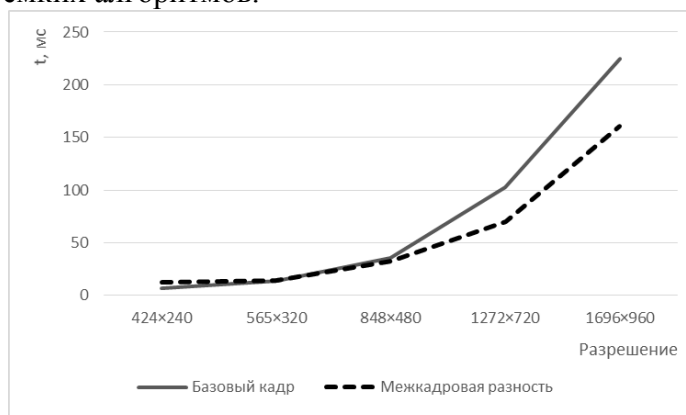


Рисунок 5 – Сравнение времени работы алгоритмов, основанных на сравнении яркости пикселя.

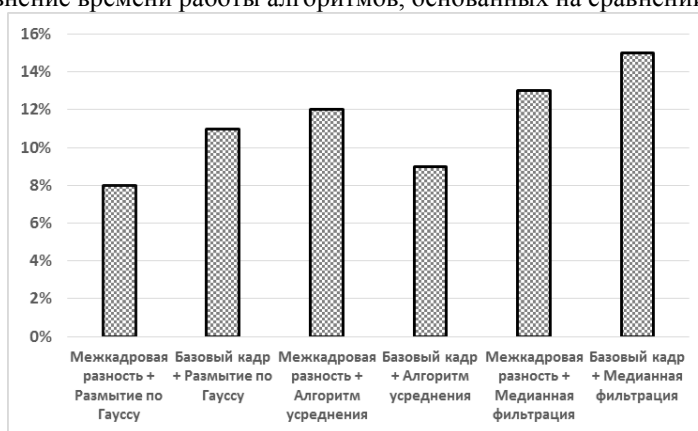


Рисунок 6 – Средняя загрузка ЦПУ

### Заключение

На основе анализа предметной области разработана информационно-логическая модель автоматизированной системы в нотации UML.

Реализованы два алгоритма обнаружения движения: алгоритм межкадровой разности и алгоритм с построением базового кадра; три алгоритма подавления шумов: алгоритм усреднения, медианная фильтрация и Гауссово размытие.

### Литература

1. Татарников, О.С. Системы видеонаблюдения и охраны на web-камерах // Компьютер Пресс – 2007
2. СюйЛэй. Исследование методов и алгоритмов обнаружения движущихся объектов в видеопотоке // Молодежный научно-технический вестник – 2013.
3. Конушин А. С. Введение в компьютерное зрение. Курс лекций // Учебное пособие – Москва: ВМК МГУ имени М.В. Ломоносова, 2013
4. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах // Учебное пособие – Новосибирск: НГТУ, 2002