

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.
ЧЕРНЫШЕВСКОГО»**

Кафедра математического анализа

Поиск движущихся объектов на видеоизображении

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 218 группы

направления *01.04.02 – Прикладная математика и информатика*

механико-математического факультета

Ковалева Никиты Александровича

Научный руководитель

профессор, д.ф.-м.н., профессор _____ С.Ф. Лукомский
подпись, дата

Зав. кафедрой

д.ф.-м.н., профессор _____ Д.В. Прохоров
подпись, дата

Саратов 2020

Введение Необходимость поиска движущихся объектов на видеоизображениях возникает в процессе решения большого количества задач, связанных с анализом объектов в видео. Такая потребность обусловлена тем, что для разных отраслей, будь то охрана предприятий или же распознавание номеров машин, и необходимо найти объект, который изменяет свое положение в пространстве для дальнейшей обработки данного объекта. Для решения таких целей используются детекторы объектов на видеоизображении.

Существует две больших группы детектора: детекторы активности и детекторы движения. Детекторы активности основаны на обнаружении активности в видео. Подобная активность может быть вызвана каким-либо движущимся объектом, однако не исключены и ложные срабатывания на изменения условий освещённости или шум при регистрации и передаче данных. Детекторы движения работают по иному принципу. Их задача – реагировать именно на движение в кадре. Источником движения может служить человек, животное, либо механическое устройство, изменяющее своё положение относительно устройства регистрации данных в течение нескольких кадров.

В данной работе будет реализован детектор движения, который будет состоять из двух этапов поиска движущихся объектов. Первым этапом является предобработка, которая включает в себя изменение размера кадра и детектор границ Канни. Вторым этапом является поиск движущихся объектов, который включает в себя межкадровую разность, операции математической морфологии и поиск связных областей.

Основное содержание работы В разделе 1 «Постановка задачи» описывается цель, которую в рамках данной работы необходимо достичь, а именно разработка программного обеспечения, которое выделяет движущиеся объекты на видео со статичным фоном. Описаны ключевые алгоритмы, которые будут реализованы в рамках данной работы, а именно: Уменьшение размера кадра, Поиск границ объектов в кадре, Межкадровая разность, Математическая морфология, а так же Поиск и выделение связных областей. В разделе 2 «Предобработка видеокадра» описываются этапы предобработки кадра, которые включают в себя: Уменьшение размера кадра и поиск границ объектов в кадре.

В подразделе 2.1 «Уменьшение размера кадра» рассматриваются следующие способы уменьшения размера кадра.

- Поиск ближайшего соседа. Данный способ был пропущен, из-за того, что в формировании конечного изображения используется малое количество пикселей в изображении, появляются неровности;
- Билинейное масштабирование изображения. Относительно алгоритма «Ближайший сосед», алгоритм «Билинейное масштабирование изображения» позволяет получить изображение с более высоким качеством, но наблюдается размытие. По этой причине данный алгоритм был так же пропущен;
- Бикубическая интерполяция. В отличие от билинейной интерполяции, которая учитывает только 4 пикселя (2×2), бикубическая интерполяция учитывает 16 пикселей (4×4). Изображения с повторной выборкой с бикубической интерполяцией являются более плавными и имеют меньше артефактов интерполяции.

На основе приведённых выше алгоритмов изменения размера изображения, был выбран «Бикубическая интерполяция». Так как необходимо выделять границы на изображении с измененным разрешением, то данный алгоритм подходит больше всего, ведь он позволяет добиться приемлемого качества изображения с измененным разрешением.

В подразделе 2.2 «Поиск границ объектов в кадре» рассматривается детектор границ Канни. Для определения данного детектора границ вводятся несколько понятий: конечной разности, дифференциальных методов и локальной обработки.

После чего рассматриваются этапы работы детектора границ Канни.

Для подавления шумов производится *сглаживание* изображения фильтром Гаусса. Уравнение для ядра фильтра Гаусса размером $(2k + 1) \times (2k + 1)$ определяется следующим образом:

$$H_{ij} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}}; 1 \leq i, j \leq (2k + 1) \quad (1)$$

Для поиска контура применяется алгоритм поиска градиентов. Контур отмечается там, где значение градиента изображения имеет наибольшее значение. На этом этапе используется оператор Собеля, который использует две маски размерности 3×3 .

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2)$$

Величина градиента и его направление определяются соответственно по формулам:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.4), \quad \alpha(x, y) = \arctan \frac{G_y}{G_x} \quad (3)$$

После применения вычисления градиента край, извлеченный из значения градиента, все еще довольно размыт, и для того, чтобы найти «сильный край» применяется следующий алгоритм. Подавление не максимумов – это метод истончения краев.

Цель алгоритма - проверить, являются ли пиксели в одном и том же направлении более или менее интенсивными, чем обрабатываемые.

1. Пиксель (i, j) обрабатывается, и пиксели в том же направлении выделяются синим цветом $(i, j - 1)$ и $(i, j + 1)$;
2. Если один из этих двух пикселей является более интенсивным, чем обрабатываемый, то сохраняется только более интенсивный. Пиксель $(i, j - 1)$ кажется более интенсивным, потому что он белый (значение 255). Следовательно, значение интенсивности текущего пикселя (i, j) устанавливается в 0. Если в направлении края отсутствуют пиксели, имеющие более интенсивные значения, то значение текущего пикселя

сохраняется.

Для фильтрации краевых пикселе, которые могут быть вызваны шумом и изменением цвета, применяется двойная пороговая фильтрация.

$$|\nabla f(x, y)| \leq L \Rightarrow f(x, y) = 0; \quad (4)$$

$$|\nabla f(x, y)| \geq T \Rightarrow f(x, y) = 255; \quad (5)$$

$$L < |\nabla f(x, y)| < T \Rightarrow f(x, y) = 55; \quad (6)$$

где T верхний порог, L нижний порог.

После двойной пороговой фильтрации остаются пиксели со слабым краем, но данные пиксели могут быть либо извлечены из истинного края, либо из-за изменений шума / цвета. Для этого необходимо отследить связан ли слабый пиксель с сильным краевым пикселем, так как шумовые отклики не будут связаны.

Для отслеживания краевого соединения применяется анализ изображения, которое является результатом работы предыдущих шагов. Рассмотрим более формальное определение данного алгоритма.

$$\left\{ \begin{array}{l} I_{f(x,y)} = 55 \\ I_{f(x+k,y+k)} = 255 \\ \alpha(x, y) = \alpha(x + k, y + k) \end{array} \right\} \Rightarrow I_{f(x,y)} = 255 \quad (7)$$

где $\alpha(x, y)$ это угол между направлением ∇f в точке (x, y) и осью x .

Для улучшения точности определения границ объектов были использованы следующие подходы.

Замена фильтра Гаусса на адаптивный фильтр, который основывается на 5 шагах:

- $K = 1$ установить итерацию по n и коэффициент амплитуды ребра h ;
- Рассчитать значение градиента $G_x(x, y)$ а также $G_y(x, y)$;
- Рассчитать по формуле ниже вес w :

$$d(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

$$w(x, y) = \exp\left(-\frac{\sqrt{d(x, y)}}{2h^2}\right)$$

- Определение адаптивного фильтра, которое сгладит изображение:

$$f(x, y) = \frac{1}{N} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j)w(x+i, y+j)$$

$$N = \sum_{i=-1}^1 \sum_{j=-1}^1 w(x+i, y+j)$$

- Когда $K = n$, необходимо остановить итерацию, в противном случае, $k = k + 1$, продолжение шага.

Замена Оператора Собеля 3×3 обусловлена тем, что данный оператор рассчитан на узкие и четкие края, в то время как целью работы является обработка видео с камер наблюдения и из-за этого, данный фильтр может дать двойной отклик на левой и правой части границы. Данной проблемы позволяет избежать оператор Собеля 5×5 данный фильтр позволяет захватить большую область, что позволяет избежать проблем, описанных ранее.

$$G_x = \begin{bmatrix} +2 & +2 & +4 & +2 & +2 \\ +1 & +1 & +2 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -2 & -1 & -1 \\ -2 & -2 & -4 & -2 & -2 \end{bmatrix}, \quad G_y = \begin{bmatrix} +2 & +1 & 0 & -1 & -2 \\ +2 & +1 & 0 & -1 & -2 \\ +4 & +2 & 0 & -2 & -4 \\ +2 & +1 & 0 & -1 & -2 \\ +2 & +1 & 0 & -1 & -2 \end{bmatrix} \quad (8)$$

В разделе 3 «Поиск движущихся объектов в видео» описываются этапы поиска движущихся объектов в видео, а именно: Межкадровая разность, Дилатация, Размыкание и Поиск и выделение объектов.

В подразделе 3.1 «Межкадровая разность» описываются шаги поиска разности кадров. Данные шаги представляют собой поиск попиксельной разности, при помощи вычисления RGB между пикселем первого и второго кадра. Следующим этапом является вычисления среднего значения для, рассчитанной выше, попиксельной разности. Финальным шагом данного алгоритма производится сравнение результата.

$$m_i = \begin{cases} 0, & p^i < 255 \\ 1, & p^i \geq 255 \end{cases} \quad (9)$$

где m_i – значение i -го элемента маски.

В подразделе 3.2 «Математическая морфология» описывается операция Дилатации и Разности, которая используется для устранения шумов и улучшения точности определения объектов на кадре.

Действие операции *дилатации* заключается в увеличении площади объекта.

$$A \oplus B = \bigcup_{b \in B} A_b \quad (10)$$

В приведенном выражении выше оператор объединения можно считать оператором, применяемым в окрестности пикселей. Структурный элемент B применяется ко всем пикселям бинарного изображения. Каждый раз, когда начало координат структурного элемента совмещается с единичным бинарным пикселем, ко всему структурному элементу применяется перенос и последующее логическое сложение (логическое ИЛИ) с соответствующими пикселями бинарного изображения. Результаты логического сложения записываются в выходное бинарное изображение, которое изначально инициализируется нулевыми значениями.

Для введения понятия Размыкания было рассмотрено понятие Эрозии.

$$A \ominus B = \{z \in A | B_z \subseteq A\} \quad (11)$$

При выполнении операции *эрозии* структурный элемент тоже проходит по всем пикселям изображения. Если в некоторой позиции каждый единич-

ный пиксель структурного элемента совпадет с единичным пикселом бинарного изображения, то выполняется логическое сложение центрального пиксела структурного элемента с соответствующим пикселом выходного изображения. В результате применения операции эрозии все объекты, меньшие чем структурный элемент, стираются, объекты, соединённые тонкими линиями, становятся разъединёнными и размеры всех объектов уменьшаются.

После чего было введено формальное определение операции Размыкания:

$$A \circ B = (A \ominus B) \oplus B \quad (12)$$

Таким образом, размыкание множества A по структурирующему элементу B строится как эрозия A по B , результат которой затем подвергается дилатации по тому же структурирующему элементу B .

В подразделе 3.3 «Поиск связных областей обходом контуров» рассматривается несколько алгоритмов связанных областей: рекурсивный поиск, поиск волновым методом и двухпроходный алгоритм поиска связных областей.

Рекурсивный поиск и поиск волновым методом были пропущены по причине низкой устойчивости к незакрытым контурам, а также низкой масштабируемостью в плане программной производительности.

В качестве осинового алгоритма был выбран двухпроходный алгоритм поиска связных областей, так как благодаря построчному обходу изображения есть возможно производить обход строк в параллельных потоках, а также данный алгоритм устойчив к незакрытым контурам.

Данный алгоритм базируется на двух шагах (проходах).

Первый проход

- Если все соседи - пиксели фона (со значением равным нулю), то $f(i, j)$ присваивается новая, ранее не использовавшаяся метка.
- Если имеется в точности один соседний пиксель с ненулевой меткой, присваиваем эту метку пикселу $f(i, j)$.
- Если имеется больше чем один ненулевой пиксель среди соседей, присваиваем метку любого из уже пронумерованных пикселов.

Второй проход. Все пиксели области пронумерованы метками областей в результате первого прохода, но некоторые области содержат пиксели с раз-

личными метками (в результате коллизий номеров). Чтобы избавиться от коллизий, просматриваем все изображение снова и перенумеровываем пиксели, используя информацию из таблицы эквивалентности.

В разделе 4 «Реализация алгоритмов», после рассмотрения теоретического материала, описываются способы реализации приведенных выше алгоритмов, а так же примеры их выполнения.

Первыми двумя алгоритмами, описанными в разделе «Реализация алгоритмов» были описаны алгоритмы пред обработки изображения.

Уменьшение размера кадра к заданному размеру происходит при помощи алгоритма «Бикубическая интерполяция». Для изменения размера кадра алгоритм производит кубическую интерполяцию в одном направлении с последующей кубической интерполяцией интерполированных значений в другом направлении.

Для реализации данного алгоритма была использована библиотека *opencv*, которая позволяет за приемлемое время (меньше 14 миллисекунд) обработать изображение.

После подраздела уменьшения размера кадра, была описана *собственная реализация «Детектор границ Канны»*. Реализация каждого шага данного алгоритма была так же описана, с приведением примеров кода. Следующие шаги были описаны:

1. Сглаживание;
2. Поиск градиентов;
3. Подавление не максимумов;
4. Двойная пороговая фильтрация;
5. Трассировка области неоднозначности.

А также были реализованы улучшения данного алгоритма:

1. Замена фильтра Гаусса на адаптивный фильтр, позволяющий не добавлять плавный эффект на границу объекта.
2. Замена Оператора Собеля 3×3 на оператор собеля 5×5 .

После реализации алгоритмов предобработки описываются алгоритмы поиска движущихся объектов.

Для реализации алгоритма поиска *межкадровой разности* был написан алгоритм, который получает два кадра с заданным интервалом и производит попиксельное вычитание. Так же для данного алгоритма было реализовано улучшение в виде избавления от непосредственного самого вычитание. Так как работа происходит с бинарным изображением необходимо лишь проверить, равны ли между собой значения интенсивности цвета. Если да, то устанавливается 0, так как изменений между кадрами не было, в противном случае 255 так как обнаружено изменение между выбранными кадрами.

Для улучшения качества изображения были использованы две *морфологических операции: Дилатации и Размыкания*. Реализация данных операций была использована из библиотеки *opencv*, так же было представлена работа данных операций из упомянутой ранее библиотеки.

Финальным реализованным алгоритмом для определения движения в видео является «Двухпроходный алгоритм поиска связных областей». Была описана детальная работа данного алгоритма, и его улучшение при помощи многопоточной обработки. Так же была сравнена скорость работы данного алгоритма со скоростью работы схожего алгоритма из библиотеки *opencv* и было выявлено, что при увеличении CPU собственная реализация работает быстрее.

Так же в работе предоставляются иллюстрирующие примеры работы каждого из алгоритмов.

Заключение Целью магистерской работы была разработка детектора движущихся объектов. В результате выполнения данной работы были рассмотрены и проанализированы алгоритмы, как для пост обработки кадров, так и для поиска движения объектов в видео. Были выявлены недостатки работы детектора границ Канни и последовательной работы двухпроходного поиска связных областей. Данные недостатки были устранены при помощи введения нового алгоритма сглаживания и реализацией параллельной обработки строк изображения.

В данной работе предложен вариант определения движущихся объектов в видео при помощи ряда алгоритмов.

- Бикубическая интерполяция;
- Детектор границ Канни и его улучшения;

- Межкадровая разность;
- Операции математической морфологии, а именно Дилатация и Размыкание;
- Двухпроходного алгоритма поиска связных областей.

Которые отлично справляются со статичным задним планом в видео, но текущая реализация не предназначена для работы с не статичным видео. Так как поиск движущихся связных объектов основывается на межкадровой разности и соответственно, если задний план будет изменяться, то алгоритм расценит смену заднего плана как движения. Для применения данного алгоритма необходимо ввести нейронную сеть, которая будет отсеивать изменившийся задний план от действительно движущихся объектов.