

УДК 004.021  
DOI 10.17513/snt.40210

## АЛГОРИТМ УЛУЧШЕНИЯ ПОВЕРХНОСТНО-ВРЕМЕННЫХ МЕТОДОВ РАСПОЗНАВАНИЯ ДВИЖЕНИЯ

Макаров О.С., Щенникова Е.В.

ФГБОУ ВО «Мордовский государственный университет им. Н.П. Огарёва», Саранск,  
e-mail: makaroff297991@gmail.com, schennikova8000@yandex.ru

Целью исследования является разработка универсального метода распознавания движения, комбинирующего пиксельный и блочный подходы и способного выдавать приемлемые результаты распознавания при малом потреблении вычислительных ресурсов. Описываемый в данной статье алгоритм основан на применении локальных бинарных шаблонов, которые позволяют анализировать связи между соседними пикселями. На предварительном этапе алгоритм использует фильтр Кувахара для сглаживания входящих видеок кадров. Для получения наиболее достоверного результата алгоритм подкрепляется дополнительными точечными проверками на основе сравнения интенсивностей пикселей, используемыми в стандартных методах вычитания фона. Также в сравнениях дескрипторов локальных бинарных шаблонов предлагается опираться на относительный порог дескриптора, в отличие от абсолютного, что приводит к большей гибкости и эффективности, и вводятся прочие улучшения для повышения быстродействия и качества распознавания. Разработанный метод является универсальным, что позволяет использовать его в большинстве сценариев, в том числе и в режиме реального времени, адаптируя процесс обработки под конкретные ситуации путем регулирования входных параметров алгоритма. Представленный метод был протестирован на наборе видеоданных, сегментированных вручную. Результаты тестирования показали, что для большинства сцен с наличием различных негативных факторов, препятствующих корректной сегментации объектов переднего плана, метод показал удовлетворительные результаты, в среднем превосходя другие аналогичные техники по основным показателям качества распознавания.

**Ключевые слова:** поверхностно-временной алгоритм, извлечение фона, распознавание движения, фильтр Кувахара, локальные бинарные шаблоны, дескрипторы подобия, относительное пороговое значение

## ALGORITHM FOR IMPROVING SPATIO-TEMPORAL MOTION DETECTION METHODS

Makarov O.S., Shchennikova E.V.

Ogarev Mordovian State University, Saransk,  
e-mail: makaroff297991@gmail.com, schennikova8000@yandex.ru

The aim of the research is to develop a universal method for motion recognition, combining pixel-based and block-based approaches, capable of providing acceptable recognition results with low computational resource consumption. The algorithm proposed in this article is based on the application of local binary patterns, which allow analyzing relationships between neighboring pixels. At the initial stage, the algorithm uses the Kuwahara filter to smooth incoming video frames. To achieve the most reliable result, the algorithm is supported by additional pixel checks based on comparing pixel intensities using standard background subtraction techniques. In addition, in comparisons of local binary pattern descriptors, it is suggested to rely on the relative descriptor threshold as opposed to the absolute, leading to greater flexibility and efficiency, and introducing other improvements to enhance speed and recognition quality. The developed method is universal, allowing its use in most scenarios, including real-time mode, adapting the processing process to specific situations by adjusting the algorithm's input parameters. The presented method has been tested on a set of manually segmented video data. The test results showed that for most scenes with various negative factors hindering proper segmentation of foreground objects, the method demonstrated satisfactory results, on average outperforming other similar techniques in main recognition quality indicators.

**Keywords:** spatio-temporal algorithm, background subtraction, motion detection, Kuwahara filter, local binary patterns, similarity descriptors, relative threshold

### Введение

Задача распознавания движения на видео заключается в создании алгоритмов, которые могут обнаруживать движение на видео и предоставлять информацию об этом движении в какой-либо форме, удобной для дальнейших шагов обработки видео. Алгоритмы, решающие такую задачу, классифицируются на две группы: пиксельные [1] и блочные [2]. Пиксельные алгоритмы основаны на анализе каждого пикселя в кадре видео. Они потребляют

меньше ресурсов в процессе работы и более просты в реализации, однако менее точны и подвержены влиянию различных негативных факторов. Блочные алгоритмы работают уже с группами пикселей вместо отдельных точек и извлекают из них определенные текстурные характеристики для дальнейшей обработки. Такие методы выдают более качественные результаты, но они более ресурсоемки и не всегда подходят для обработки в режиме реального времени.

**Целью исследования** является разработка универсального метода распознавания движения, комбинирующего пиксельный и блочный подходы и способного выдавать приемлемые результаты распознавания при малом потреблении вычислительных ресурсов.

В настоящий момент разработано множество пиксельных и блочных методов, большинство из которых опираются на уже проработанные фундаментальные алгоритмы, модифицируя их под определенные ситуации или объединяя их с прочими улучшениями для достижения наилучшего результата [3].

Наиболее распространенными пиксельными алгоритмами являются методики извлечения фона, такие как смесь гауссианов (GMM) с различными улучшениями [4], ядерная оценка плотности (KDE) [5], ядерный фильтр [6], алгоритмы Vibe и Vibe+ [7] и др. Эти методы хороши для использования в режиме реального времени, но иногда приводят к большому количеству ложных срабатываний.

В области применения блочных алгоритмов в настоящий момент достаточно распространена группа методов с применением локальных бинарных шаблонов (LBP). В работе [8] авторы успешно построили статическую модель фона на основе алгоритмов самоподобия (LSS-дескрипторы) и применяли специальную метрику для классификации пиксельных блоков. На этапе постобработки использовались информация о цвете и морфологические операции для уточнения границ модели.

В работе [9] был предложен новый алгоритм, который классифицирует все пиксели на несколько кластеров на основе сходства их изменений яркости и затем для каждого кластера определяет, чем были вызваны изменения. Изменения освещенности показывают схожие изменения яркости, и алгоритм использует эту информацию для фильтрации срабатываний.

В исследовании [10] проведен сравнительный обзор алгоритмов с применением локальных бинарных шаблонов для обработки обычных изображений. Авторы представили описания различных вариантов реализации LBP-дескрипторов и проанализировали некоторые современные технологии по применяемым алгоритмам и решаемым проблемам.

В целом, вышеуказанные алгоритмы на основе локальных бинарных шаблонов решают достаточно узкие проблемы либо имеют неудовлетворительные показатели по быстродействию по сравнению с пиксельными алгоритмами, что открывает область для исследования данной проблемы.

## Материалы и методы исследования

Стандартный подход расчета LBP-дескриптора представлен в формуле (1):

$$LBP(x, y) = \sum_{p=0}^{p-1} f(i_{x,y,p}, i_{x,y}) \cdot 2^p, \quad (1)$$

где  $i_{x,y}$  – рассматриваемый пиксель (обычно центральный),  $i_{x,y,p}$  –  $p$ -й сосед центрального пикселя,  $f(i_p, i)$  – функция сравнения пикселей, возвращающая 0 или 1 при различных условиях. В данном исследовании в качестве функции сравнения предлагается использовать признак «похожести» пикселей в совокупности с заменой абсолютного порога в традиционном подходе на относительный, который зависит от интенсивности центрального пикселя  $i$ . Это позволяет улучшить работу дескриптора в большинстве случаев изменения освещенности. Для этого достаточно определить функцию, как указано в формуле (2):

$$f(i_p, i) = \begin{cases} 1, & |i_p - i| < T_{rel} \cdot i \\ 0, & |i_p - i| \geq T_{rel} \cdot i \end{cases}, \quad (2)$$

где  $T_{rel}$  – коэффициент относительного порогового значения.

Еще проблемой LBP-дескрипторов является их неустойчивость к шуму. Для подавления такой проблемы обычно используют методы сглаживания. В данном исследовании предлагается использовать фильтр Кувахара – это нелинейный оператор для сглаживания изображений, особенностью которого является сохранение границ объектов [11]. Отметим, что такой фильтр добавляет на изображения эффект мультипликационного фильма, создается ощущение, что кадр грубо нарисован красками. Такой эффект усиливается с увеличением количества повторений или увеличением размеров окна фильтра. При сильном сглаживании теряются детали на изображении, что в некоторых сценариях довольно критично. Вследствие этого фильтр Кувахара имеет ограниченную область применения, однако он как нельзя лучше подходит к задаче распознавания движения, поскольку маска объектов переднего плана не требовательна к наличию деталей изображения, но требует сохранения границ для более точной сегментации.

Однако даже с приведенными изменениями полагаться исключительно на локальные бинарные шаблоны не всегда оправдано, поскольку такие вычисления нередко приводят к ложным срабатываниям. На рисунке 1 представлен пример работы алгоритма при расчете LBP-дескриптора в отдельном блоке.

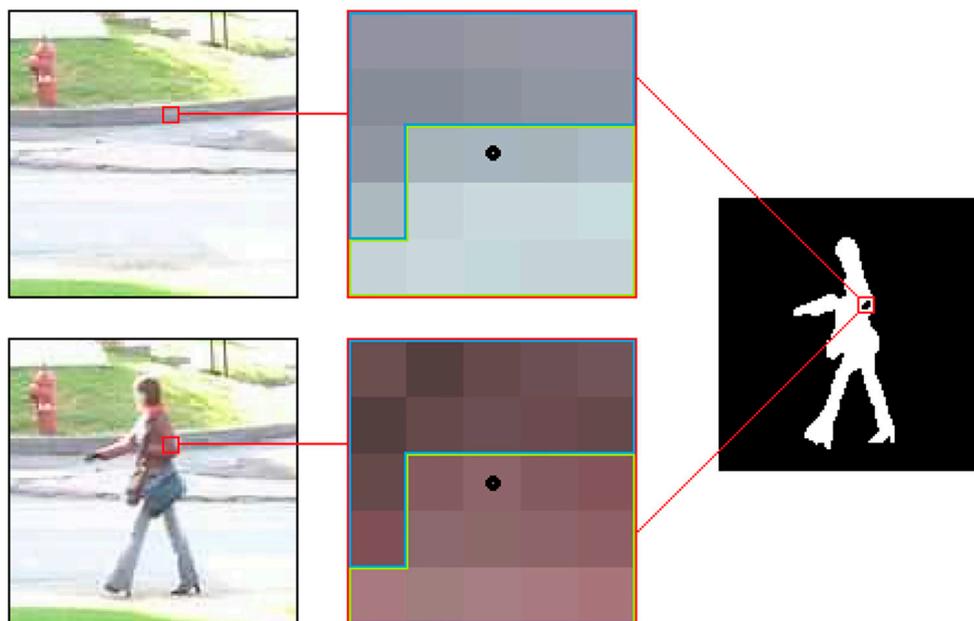


Рис. 1. Пример формирования одинаковых значений LBP-дескрипторов для фона и объекта переднего плана: сверху слева – фоновая модель, слева снизу – текущий кадр, в центре сверху – окно LBP-дескриптора для красного блока в фоновой модели, в центре снизу – окно LBP-дескриптора для соответствующего блока текущего кадра, справа – результат распознавания

Синие зоны на рисунке 1 объединяют пиксели, которые существенно отличаются от центральных пикселей (помечены черной точкой), зеленые зоны соответствуют «похожим» пикселям. При сравнении заметно, что значения LBP-дескрипторов в этой зоне будут одинаковыми, что приводит к неправильной классификации зоны на маске движения в правой части рисунка – алгоритм будет ложно классифицировать части перемещающегося объекта как фоновые, что неверно, поскольку в текущем кадре представлен двигающийся объект. Это объясняется тем, что текстурные характеристики блоков (связи между пикселями) совпадают.

Этот пример доказывает, что полагаться только на LBP-дескрипторы нежелательно, поэтому в данном исследовании предлагается объединить блочный подход со стандартными пиксельными методиками извлечения фона. Это позволит использовать преимущества обработки локального контекста пикселей при расчетах LBP-дескрипторов с дополнительными точечными проверками для фильтрации ложных срабатываний.

На рисунке 2 представлена блок-схема предлагаемого алгоритма.

Далее представлен псевдокод части алгоритма, отмеченный на рисунке 2 пунктирной линией:

```

1: function isPixelForeground(px,y)
2:   TotalDescriptorDistance = 0
3:   TotalIntensityDistance = 0
4:   for i = 1 to TotalChannelsNumber do
5:     DescriptorDistance = HammingDistance(Descriptor(px,y,c, i), Descriptor(px,y, i))
6:     If DescriptorDistance ≥ Tdesc · Intensity(px,y,c, i) · alpha
7:       return true
8:     IntensityDistance = Difference(Intensity(px,y,c, i), Intensity(px,y, i))
9:     If IntensityDistance > Tint · alpha
10:      return true
11:     TotalDescriptorDistance += DescriptorDistance
12:     TotalIntensityDistance += IntensityDistance
13:   end for
14:   return TotalDescriptorDistance > Tdesc · Intensity(px,y,c, i) or TotalIntensityDistance > Tint

```

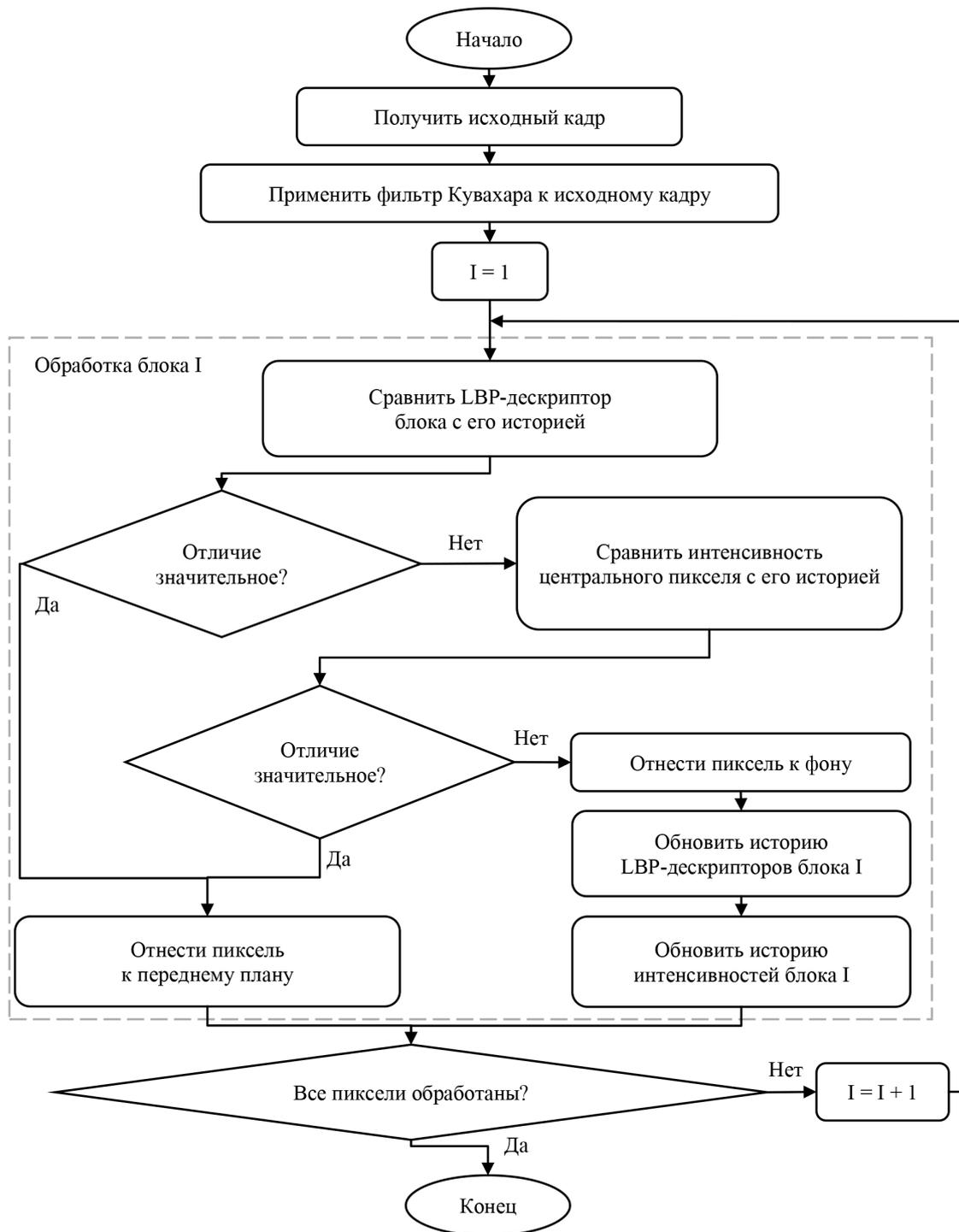


Рис. 2. Блок-схема предлагаемого алгоритма

Данный блок описывает упрощенное обнаружение изменений для сегментации изображения с дополнительной проверкой интенсивности.

Переменная `TotalChannelsNumber` – это количество цветных каналов в кадре (3 для RGB формата); `Difference` – абстракт-

ная функция определения разности между пикселями (может быть взят любой алгоритм вычитания фона в качестве реализации); `Descriptor` – функция вычисления значения дескриптора (формулы 1, 5); `HammingDistance` – функция вычисления расстояния Хэмминга.

Псевдокод описывает принятие решений при классификации рассматриваемого пикселя на основе блочной и пиксельной информации. Экономия вычислений происходит при использовании поканального сравнения – если изменения по одному из каналов в какой-то момент уже становятся существенными, дальнейшие сравнения отменяются, а пиксель относится к объекту переднего плана. Разбиение вычислений по каналам в первую очередь направлено на ускорение обработки, однако на практике выяснилось, что с помощью такого подхода алгоритм может обнаруживать незначительные изменения в рамках одного канала (например, когда темный объект перекрывается темно-синим).

В качестве алгоритма вычитания фона в этом исследовании был взят медианный фильтр, однако в зависимости от особенностей задачи и требований к производительности может быть использован любой алгоритм вычитания фона, который возвращает разность между пикселями.

#### Результаты исследования и их обсуждение

При тестировании разработанного алгоритма использовались наборы данных, которые были получены из видеокolleкций открытой онлайн базы данных Change

Detection 2014 [12]. Каждая видеозапись из этого набора уже разбита на видеокдры и вручную сегментирована для удобства сравнения ожидаемого и актуального результатов работы алгоритма. Всего было использовано 10 видеопоследовательностей из 5 различных категорий сцен, каждая из которых имеет свои проблемы и сложности. Метод тестировался на следующем универсальном наборе параметров:

- $T_{desc} = 12$ : суммарное пороговое значение для побитового сравнения значений LBP-дескрипторов на предмет обнаружения «похожести» (не имеет значения, в каких конкретно пикселях имеются различия, важно лишь общее количество несовпадений в области);

- $T_{int} = 75$ : суммарное пороговое значение для сравнения интенсивностей пикселей по всем цветовым каналам;

- $N = 15$ : размер буфера исторической информации – количество дескрипторов и интенсивностей цвета, хранящихся в виде выборок для каждого пикселя в опорной модели;

- $\alpha = 0,5$ : пороговый поправочный коэффициент, применяемый к  $T_{desc}$  и  $T_{int}$  для поканального сравнения;

- $T_{rel} = 0,365$ : коэффициент относительного порога LBP-дескриптора.

Примеры работы алгоритма для некоторых видеозаписей показаны на рисунке 3.

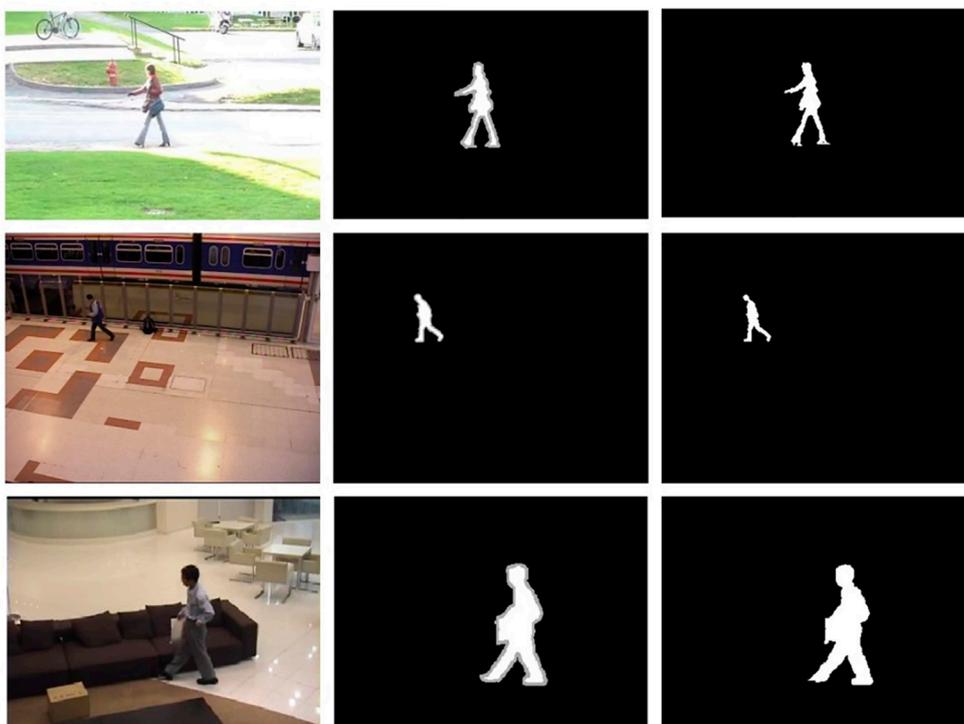


Рис. 3. Результаты работы предложенного метода: слева – оригинальные кадры; в центре – идеальные результаты распознавания, приближенные к реальности; справа – результаты работы алгоритма

Таблица 1

Показатели точности распознавания предлагаемого алгоритма

Категория видеозаписей	Prc	Rcl	PCC	F-балл
PETS 2006	0,84	0,89	99,2	0,86
Обычные видеозаписи	0,83	0,86	99,1	0,84
Динамический фон	0,66	0,70	96,9	0,68
Прерывистое движение объектов	0,79	0,79	98,3	0,79
Тень	0,76	0,76	96,6	0,76
Итого	0,78	0,80	98,0	0,79

Таблица 2

Показатели точности распознавания современных методов извлечения фона

Метод	Prc	Rcl	PCC	F-балл
Предлагаемый	0,78	0,80	98,0	0,79
ViVe+	0,67	0,81	98,9	0,73
ViVe	0,66	0,71	98,2	0,68
KDE	0,68	0,65	96,5	0,67
GMM	0,65	0,63	95,6	0,64
Медианный фильтр	0,59	0,47	91,2	0,52

Для оценки точности алгоритма учитывались распространенные показатели компьютерного зрения: точность (Prc), чувствительность (Rcl), процент корректных классификаций (PCC) и F-балл [3, 13]. В таблице 1 представлена полная информация о характеристиках предлагаемого алгоритма применительно ко всем пяти категориям видеопоследовательностей наряду с усредненными итоговыми значениями.

Наилучшие значения наблюдаются в категориях «PETS 2006» и «Обычные видеозаписи», что объясняется стабильностью фона и простотой моделирования сцен в этих видеозаписях. Наихудшие показатели алгоритм продемонстрировал на видеозаписях категории «Динамический фон». Видеозаписи из этой группы содержат колебания фона, что приводит к большому числу ложных срабатываний. Однако именно для этой категории применение фильтра Кувахаха показало наилучший прирост качества распознавания.

На практике выяснилось, что применение только одного из предложенных улучшений (фильтр Кувахаха, относительное пороговое значение, дополнительные пиксельные проверки) сказывается на общей эффективности алгоритма незначительно и иногда даже ухудшает показатели для некоторых видеозаписей, поэтому рекомендуется использовать все улучшения одновременно.

Например, при связке фильтра Кувахаха с LBP-дескрипторами успешно обрабатываются проблемы теней и изменения освещения, когда соответствующие зоны в кадре изменяются равномерно. Относительность значений интенсивностей между соседними пикселями в таких блоках сохраняется, поэтому медленные и нерезкие изменения фона между кадрами успешно отфильтровываются, не приводя к ложным срабатываниям.

В таблице 2 представлено сравнение показателей точности распознавания предлагаемого метода с основными современными методиками для решения подобных задач применительно к тем же наборам видеозаписей. Данные в таблице 2 отсортированы по убыванию F-балла.

В таблице 3 продемонстрировано сравнение тех же алгоритмов по потреблению вычислительных ресурсов на основе двух показателей: по среднему количеству обработанных в секунду кадров (FPS) и по среднему потреблению оперативной памяти в МБ. Каждая строка таблицы 3 соответствует одной из рассмотренных видеопоследовательностей.

Согласно результатам, представленным в таблицах 2 и 3, в среднем предлагаемый метод обрабатывает лучше, чем другие современные алгоритмы по большинству показателей точности, и демонстрирует хорошее качество распознавания.

Таблица 3

Показатели потребления вычислительных ресурсов современных методов извлечения фона

№	Предлагаемый		Vibe+		Vibe		KDE		GMM		Медианный фильтр	
	FPS	Память	FPS	Память	FPS	Память	FPS	Память	FPS	Память	FPS	Память
1	29,3	416	34,2	309	38,4	215	47,0	350	60,1	52	67,2	47
2	44,0	89	50,7	42	56,3	25	60,4	67	80,2	13	90,2	9
3	51,3	92	55,7	48	61,4	51	65,1	81	82,4	19	93,3	13
4	44,7	40	59,4	47	62,2	35	71,8	78	81,7	17	92,3	14
5	29,0	364	35,0	105	40,4	115	51,2	292	67,1	35	70,3	27
6	45,7	38	61,1	30	63,7	24	71,2	85	78,4	16	85,4	10
7	44,4	149	51,2	86	48,8	88	55,4	113	71,2	24	75,3	19
8	48,2	39	62,7	36	60,7	25	74,3	92	82,6	11	92,2	9
9	42,5	41	52,4	39	61,8	26	66,6	74	81,1	13	93,1	8
10	42,5	43	50,3	36	57,3	25	68,4	82	81,6	14	92,1	9

Однако качество работы компенсируется скоростью – в среднем, алгоритм работает немного медленнее аналогов и потребляет больше памяти из-за введения дополнительных шагов по фильтрации и операциям сравнения, а также ведения дополнительной истории LBP-дескрипторов. Более подробно результаты тестирования представлены в протоколах испытаний в репозитории GitHub [14].

### Заключение

В этой статье был представлен новый эффективный метод улучшения пространственно-временных алгоритмов вычитания фона, который позволяет существенно повысить качество распознавания при сравнительно небольших вычислительных затратах. Улучшение основано на одновременном применении нескольких модификаций.

Во-первых, это введение фильтра Куваха в качестве первого этапа предварительной обработки видеок кадров, что позволяет сглаживать изображения с одновременным сохранением границ. Потеря мелких деталей вследствие подобной фильтрации практически не оказывает влияния на итоговую сегментацию объектов переднего плана. Напротив, особенности такого фильтра добавляют на изображение «эффект рисунка», тем самым равномерно закрашивая однородные области, без размывания контуров объектов, что позволяет сохранить локальную контекстную информацию, используемую на последующих шагах.

Во-вторых, это введение локальных бинарных шаблонов в алгоритм распознавания,

что позволяет учитывать пространственные связи между соседними пикселями. В данном исследовании вместо стандартного подхода с разностью интенсивностей пикселей использовались двоичные дескрипторы подобия, определяющие степень «похожести» блоков друг на друга. Дополнительно было предложено использовать не абсолютный порог сравнения интенсивностей пикселей, а относительный, значение которого в каждом кадре зависит от интенсивности центрального пикселя рассматриваемого блока. Также предлагаемая методика предполагает отказ от применения сложных вычислений при сравнении интенсивностей пикселей. Каскад вычислений может быть значительно упрощен последовательным поканальным вычислением с ранним завершением цикла проверок, если локальной информации становится достаточно для классификации рассматриваемого пикселя на основе порога для сравнения по отдельному каналу.

В процессе тестирования было отмечено, что использование только одной из этих модификаций улучшает качество распознавания несущественно, в то время как одновременное применение нововведений в рамках максимальной конфигурации алгоритма позволяет достичь значительного прироста показателей эффективности алгоритма. Это объясняется тем, что комбинация фильтра Куваха и LBP-дескриптора взаимно компенсирует недостатки этих методик. Дополнительно было замечено, что в связи с особенностями реализации алгоритм успешно справляется с некоторыми негативными факторами распознавания, такими как из-

менение освещенности сцены, наличие теней и медленно перемещающихся объектов, а также точечные всплески интенсивности пикселей вследствие высокочастотных колебаний фона.

По результатам тестирования был сделан вывод о том, что качество распознавания предложенного алгоритма выше, чем у других современных методов по извлечению фона. В дальнейшем можно интегрировать предложенный метод с другими современными алгоритмами для получения более оптимальных результатов, например для сцен с динамическим фоном. Также в качестве улучшений в будущих исследованиях планируется оптимизировать потребление вычислительных ресурсов.

### Список литературы

1. Kulkarni V., Pol T., Mapari S. Comparative Study of Background Subtraction Algorithm for Moving Object Detection using OpenCV // Journal of Harbin Engineering University. 2024. Vol. 45, Is. 4. P. 318-327.
2. Fauzi N., Musa Z., Hujainah F. Feature-Based Object Detection and Tracking: A Systematic Literature Review // International Journal of Image and Graphics. 2023. Vol. 24, Is. 3. DOI: 10.1142/S0219467824500372.
3. Макаров О.С., Щенникова Е.В. Анализ алгоритмов вычитания фона // International Conference on Business Economics, Management, Engineering Technology, Medical and Health Sciences. 2021. С. 65–77.
4. Rakesh S., Nagaratna P.H., Gopalachari M.V., Jayaram D., Madhu B., Hameed M.A., Vankdothu R., Suresh Kumar L.K. Moving object detection using modified GMM based background subtraction // Measurement: Sensors. 2023. Vol. 30. URL: <https://www.sciencedirect.com/science/article/pii/S2665917423002349> DOI: 10.1016/j.measen.2023.100898. <https://doi.org/10.5565/REV%2FELCVIA.855>
5. Węglarczyk S. Kernel density estimation and its application // ITM Web of Conferences. 2018. Vol. 23. URL: [https://www.itm-conferences.org/articles/itmconf/pdf/2018/08/itmconf\\_sam2018\\_00037.pdf](https://www.itm-conferences.org/articles/itmconf/pdf/2018/08/itmconf_sam2018_00037.pdf) DOI: 10.1051/itmconf/20182300037.
6. Shah A., Bangash J., Khan A., Ahmed I., Khan A., Khan A., Khan A. Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images // Journal of King Saud University – Computer and Information Sciences. 2022. Vol. 34, Is. 3. P. 505-519. DOI: 10.1016/j.jksuci.2020.03.007.
7. Bou X., Ehret T., Facciolo G., Morel J.-M., Grompone R. Reviewing ViBe, a Popular Background Subtraction Algorithm for Real-Time Applications // Image Processing On Line. 2022. Vol. 12. P. 527-549. DOI: 10.5201/ipol.2022.434.
8. Jodoin J.P., Bilodeau G.A., Saunier N. Background subtraction based on Local Shape // arXiv. 2012. URL: <https://arxiv.org/pdf/1204.6326>. DOI: 10.48550/arXiv.1204.6326.
9. Yoshinaga, S., Shimada A., Nagahara H., Taniguchi R. Background model based on intensity change similarity among pixels // FCV 2013 – Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision. 2013. P. 276-280. DOI: 10.1109/FCV.2013.6485504.
10. Khaleefah S., Mostafa S., Mustapha A., Nasrudin M. Review of local binary pattern operators in image feature extraction // Indonesian Journal of Electrical Engineering and Computer Science. 2020. Vol. 19, Is. 1. P. 23-31. DOI: 10.11591/ijeecs.v19.i1.
11. Le H., Tran G. Free-size accelerated Kuwahara filter // Journal of Real-Time Image Processing. 2021. Vol. 18, Is. 4. P. 2049-2062. DOI: 10.1007/s11554-021-01081-3.
12. Change Detection 2014 – база данных для тестирования алгоритмов распознавания движения [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/maamri95/cdnet2014> (дата обращения: 12.10.2024).
13. Щенникова Е.В., Макаров О.С. Метод извлечения фона с использованием фильтра второй производной // XXI век: итоги прошлого и проблемы настоящего плюс. 2022. № 4(60). С. 42-51. DOI: 10.46548/21vek-2022-1160-0006.
14. Протоколы испытаний рассмотренных алгоритмов [Электронный ресурс]. URL: <https://github.com/bajitumor/LBP-Algorithm-Tests> (дата обращения: 12.10.2024).