

## Помехоустойчивый морфологический алгоритм обнаружения вилочного погрузчика на видео\*

*В. О. Черноусов, А. В. Савченко*

*v.chernousov@mail.ru*

НИУ Высшая школа экономики, Москва, Россия

Исследуется задача обнаружения движущегося вилочного погрузчика на видео при наличии помех, в которой точность традиционного сопоставления локальных дескрипторов (SURF, SIFT, FAST, ORB) не достаточна. Предложен новый алгоритм, на первом этапе которого на кадре выделяются движущиеся объекты, после чего на передней части объекта находится потенциальная область вил и груза. На втором этапе выделяются контуры, затем с помощью морфологических преобразований вычисляются элементарные геометрические признаки объекта. Показано, что такой подход позволяет на 7% и 50% понизить вероятности ложной тревоги и пропуска события, соответственно, при детектировании пустого погрузчика по сравнению с методом FAST, является устойчивым к аддитивному шуму, а обработка одного кадра происходит в среднем на 30 мс быстрее.

**Ключевые слова:** *обнаружение объектов на видео; зашумленная среда; motion history image; морфология*

## A noise-resistant morphological algorithm of video-based moving forklift truck detection\*

*V. O. Chernousov, A. V. Savchenko*

National Research University Higher School of Economics, Moscow, Russian Federation

**Background:** The problem of video-based detection of the moving forklift truck is explored. It is shown that the detection quality of the state-of-the-art local descriptors (SURF, SIFT, FAST, ORB) is not satisfactory if the resolution is low and the lighting is changed dramatically.

**Methods:** In this paper, it is proposed to use a simple mathematical morphological algorithm to detect the presence of a cargo on the forklift truck. At first, the movement direction is estimated by the updating motion history image method and the front part of the moving object is obtained. Next, contours are detected and binary morphological operations in front of the moving object are used to estimate simple geometric features of empty forklift.

**Results:** The authors' experimental study shows that the best results are achieved if the bounding rectangles of empty forklift contours are used as an object validation rule. Namely, FAR and FRR of empty cargo detection is 7% and 50% lower than FAR and FRR of the FAST descriptor. The proposed method is much more resistant to the effect of additive noise. The average frame processing time for the morphological algorithm is 5 ms (compare with 35 ms of FAST method).

**Concluding Remarks:** The proposed morphological method is task specific and can be used only for forklift truck detection. Additional detection principles need to be added to adopt algorithm for other moving object detection in noisy environment.

**Keywords:** *video-based object detection; noisy environment; motion history image; binary morphology*

---

\*Работа выполнена за счет средств гранта Российского научного фонда (проект №14-41-00039).

## Введение

В настоящее время все актуальнее становится исследование способов применения алгоритмов машинного зрения в прикладных задачах автоматического контроля состояния производства. Например, в данной работе рассматривается прикладная задача обнаружения движущегося вилочного погрузчика и выделение его основных атрибутов (направление движения, наличие/отсутствие груза на вилах) для автоматизации производственных линий в существующей автоматизированной системе управления.

Классический подход для решения задачи детектирования текстурированного объекта основывается на использовании популярных алгоритмов компьютерного зрения SIFT [1], SURF [2], ORB [3], FAST [4] и пр. [5]. Эти алгоритмы ищут ключевые точки на эталонном изображении объекта, вычисляют их дескрипторы и сравнивают их с дескрипторами кадра, полученными аналогичным образом. К сожалению, такой подход существенно зависит от качества изображения и часто показывают высокие вероятности ошибок I (False Reject Rate, FRR) и/или II рода (False Accept Rate, FAR), особенно при наличии на изображении помех (вариативное освещение, частичное выпадение объекта из кадра и т. п.).

Целью проведенного исследования является снижение влияния фактора зашумленности изображения на точность обнаружения движущегося объекта на видео [6] при помощи использования морфологических операций [7]. В этом случае силуэт движущегося объекта может быть получен методом МНИ (Motion History Image) [8], после чего на основе полученной информации об объекте можно вычислить его ключевые характеристики, применяя дополнительные морфологические преобразования изображения [9, 10]

Работа организована следующим образом: во втором разделе формулируется задача обнаружения пустого движущегося погрузчика. В третьем разделе проводится экспериментальное сопоставление традиционных методов обнаружения и разработанного морфологического алгоритма и анализируется устойчивость точности к аддитивному шуму. В заключительном разделе представлены основные выводы, сделанные по результатам проведенного исследования.

## Материалы и методология

Пусть имеется база данных (БД) модельных изображений детектируемых объектов и заданы их атрибуты (размер, предполагаемая скорость движения и пр.). Задача распознавания состоит в том, чтобы на входящем потоке видеоданных определить положение движущегося объекта, очертив его ограничивающим прямоугольником и путем определения его ключевых признаков, отнести его к одному из искомым классов [6]. Нами рассматривается прикладной вариант такой задачи, в котором необходимо обнаружить движущийся вилочный погрузчик и определить его тип: пустой (рис. 1, а) или с грузом (рис. 1, б) [11]. Для указанной задачи известны следующие условия:

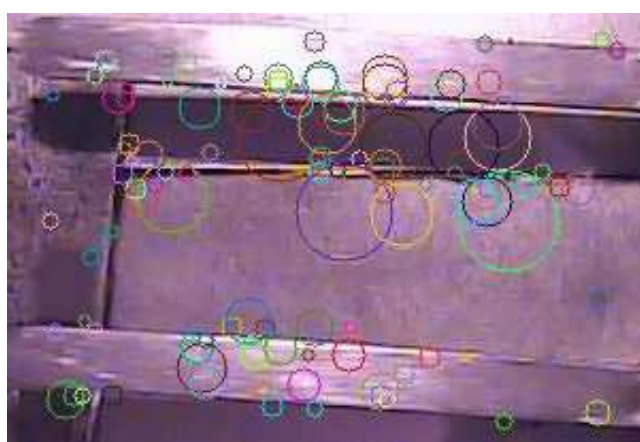
- 1) погрузчик движется в кадре горизонтально или вертикально с небольшими возможными отклонениями;
- 2) погрузчик является самым большим (занимает более половины кадра), но не единственным движущимся объектом на видео;
- 3) существуют два типа погрузчиков: вилочный и с одним длинным шестом для груза.

Поставленная задача традиционно решается методом поиска ключевых точек (алгоритм 1) на каждом изображении видеопоследовательности и сравнением их локальных дескрипторов с искомым изображением (SIFT [1], SURF [2]). Данный метод подразумевает устойчивость искомого объекта к аффинным преобразованиям, однако, исходя из особенностей задачи, где погрузчик движется только под углами  $0^\circ$  и  $180^\circ$ , сравнение



(а) Пустой вилочный погрузчик

(б) Погрузчик с грузом

**Рис. 1.** Примеры изображений детектируемого объекта**Рис. 2.** Искомое изображение пустых вилок с выделенными ключевыми точками

дескрипторов производится только с учетом горизонтальной или вертикальной ориентации объекта (в функции Compare). У данного алгоритма имеются следующие параметры:  $M_0$  — количество совпадений дескрипторов эталонного и найденного на видео объектов, при котором эти объекты признаются идентичными;  $M_1$  — порог количества кадров видео (в % от общего числа), на которых должен быть обнаружен объект, при достижении которого он считается найденным на видео;  $N_f$  — количество кадров с обнаруженным движущимся объектом;  $N_m$  — количество кадров, на которых был обнаружен искомый объект в процессе работы алгоритма. Функция Area отвечает за расчет площади переданной ей области. К сожалению, точность этого подхода (алгоритм 1) существенно зависит от качества входного видеосигнала. Поэтому уже небольшие помехи могут значительно ухудшить точность распознавания.

Для повышения качества обнаружения на зашумленных данных в настоящей работе предложено учитывать специфику задачи для разработки менее универсального, но более точного алгоритма. Используется информация о минимальном размере погрузчика относительно размера кадра (больше половины) — площадь также рассчитывается функцией Area, об отсутствии других объектов подобного размера и данные о том, что погрузчик движется на видео горизонтально или вертикально лишь с небольшими отклонениями. Предложенный алгоритм (алгоритм 2) состоит из двух частей. В первой части производится определение наличия движущегося объекта и направления его движения

```

Data: Последовательность кадров  $\{X(t)\}$ ,  $t = \overline{1, T}$ , изображение пустых вил  $X_{\text{trn}}$ ,
Порог совпадений дескрипторов  $M_0$ , Порог количества кадров с обнаруженным
объектом на видео  $M_1$ 
Result: True если пустые вилы найдены, False иначе
1 kpts_trn := DetectAndCalc( $X_{\text{trn}}$ ); // поиск ключевых точек и расчет их
  дескрипторов для  $X_{\text{trn}}$ 
2  $N_m := 0$ ,  $N_f := 0$ ;
3 for  $t = 1, \dots, T$  do
4   // проверка области движения на соответствие минимальному размеру в 0.5
   кадра
5   if Area(frame_moving_region) > 0.5 * Area(frame) then
6      $N_f := N_f + 1$ ; kpts_t := DetectAndCalc( $t$ ); // поиск ключевых точек и расчет
     их дескрипторов для  $t$ 
7     match := Compare(kpts_t, kpts_trn); // рассчитаем количество совпадающих
     дескрипторов
8     if match >  $M_0$  then
9       |  $N_{\text{match}} := N_{\text{match}} + 1$ ;
10    end
11  end
12 end
13 if ( $N_{\text{match}}/N_f$ ) >  $M_1$  then
14  | return True;
15 else
16  | return False;
17 end

```

**Algorithm 1:** Принцип работы алгоритма локальных дескрипторов

на видео: последовательно идущий кадр  $X(t)$  сравнивается с предыдущим  $X_{bg}$  методом МНП (применяется адаптивное пороговое преобразование методом Гаусса, где пороговое значение  $T(x, y)$  есть взвешенная сумма  $\text{blocksize} \times \text{blocksize}$  окрестности пикселя  $(x, y)$  —  $C$ ) (рис. 4, а) [8]. Затем для обнаруженной движущейся области строится ограничивающий прямоугольник. Заметим, что для традиционного алгоритма, основанного на поиске ключевых точек и сравнении их дескрипторов, задача обнаружения движения не является необходимой, поскольку оно может быть извлечено из матрицы аффинного преобразования, получаемой вследствие работы алгоритма RANSAC [9].

На втором этапе выполняются операции обнаружения контуров в передней части обнаруженного движущегося объекта (который был отмечен ограничивающим прямоугольником) с помощью оператора Кэнни [12] (рис. 4, б). После этого строятся ограничивающие прямоугольники для больших контуров (размер определяется исходя из размера кадра и найденной на нем движущейся области, чтобы отсеять контуры помех), среди которых затем выбирается пара прямоугольников при помощи определенных критериев сравнения. Они классифицируются как ограничивающие прямоугольники пустых вил погрузчика (рис. 4, в).

Алгоритм требует установки следующих параметров: тип фильтрации, используемый при предварительной обработке изображения и размерность ядра этого фильтра; минимальный размер движущегося сегмента кадра на видео (по условиям технического зада-

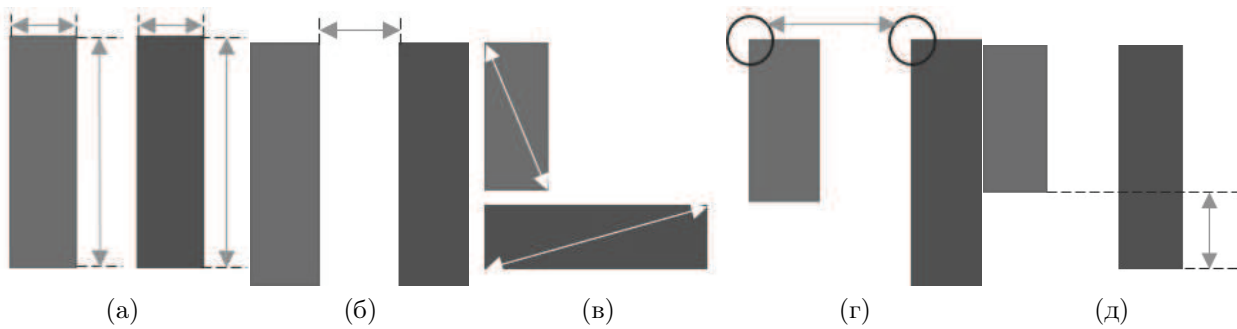
```

Data: Последовательность кадров  $\{X(t)\}$ ,  $t = \overline{1, T}$ 
Result: True если пустые вилы найдены, False иначе
1  $X_{bg} := X(1)$ ; // Считаем самый первый кадр фоновым
2 rectangles := 0, object_found := False, contours := 0;
3 for  $t = 2, \dots, T$  do
4    $X_{cur} := X(t) - X_{bg}$ ;
5   Filter( $X_{cur}$ );
6   //применяем адаптивное пороговое преобразование к  $X_{cur}$ ;
7    $X_g := ToGrayscale(X_{cur})$ ;
8   Filter( $X_g$ );
9   UpdateMotionHistory( $X_g$ ); //обновляем историю движений изображением  $X_g$ ;
10  bound_rect = BoundingBox( $X_g$ ); // ищем контур движущегося силуэта и строим
    его ограничивающий прямоугольник
11  // проверка области движения на соответствие минимальному размеру
12  if Area(bound_rect) > 0.5 * Area(frame) then
13    DetectMoveDirection( $X_g, X_{bg}$ ); contours := Canny( $X_g$ ); //ищем всевозможные
    контуры в движущейся области
14    MorphClosure( $X_g, rect, contours$ ); //выполняем морфологическое замыкание
    контуров с прямоугольным элементом
15    for each cont in contours do
16      // размер контура должен быть больше определенного порога
17      if Size(cont) > threshold then
18        rect := BoundingBox(cont);
19        if motion.isHorizontal AND rect.width >
        rect.height OR motion.isVertical AND rect.width < rect.height then
20          rectangles+ = BoundingBox(cont);
21        end
22      end
23    end
24    //производим выборку ограничивающих прямоугольников по заданным
    критериям
25    if  $\forall i \in \{1, \dots, 5\} C_i = true$  then
26      object_found := True;
27    end
28  end
29   $X_{bg} := X(t)$ ;
30 end
31 if object_found then
32   return True;
33 else
34   return False;
35 end

```

**Algorithm 2:** Принцип работы морфологического алгоритма

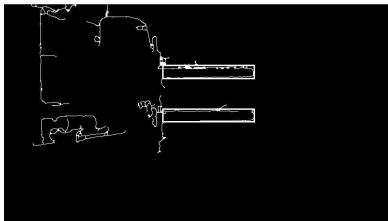
ния — не менее половины кадра). Также требуется выбрать критерии отбора ограничивающих прямоугольников, которые проверяются на завершающем этапе работы алгоритма.



**Рис. 3.** Критерии отбора ограничивающих прямоугольников: (а)  $Width > W_1$  AND  $Height > H_1$ ; (б)  $Distance(Rect1, Rect2) < D$ ; (в)  $|Area(Rect1) - Area(Rect2)| < A$ ; (г)  $R_2 < |Rect1.x - Rect2.x| < R_1$ ; (д)  $|Width1 - Width2| < W$



(а) Результат вычитания двух последовательных кадров (б) Погрузчик с грузом



(в) Кадр с погрузчиком после морфологической обработки с обнаруженными вилами (помечены белыми прямоугольниками)

**Рис. 4.** Обработка кадра в алгоритме

Были реализованы следующие критерии отбора:  $(C_1)$  — минимальная длина и ширина прямоугольника должна быть больше  $W_1 = const$  и  $H_1 = const$ , соответственно;  $(C_2)$  — максимальное расстояние между прямоугольниками (вилы не могут быть расставлены слишком широко) не больше фиксированной величины  $D = const$ ;  $(C_3)$  — максимальная разность площадей прямоугольников (поправка на небольшое отклонение вилок от горизонтального/вертикального расположения) должна быть меньше  $A = const$ ;  $(C_4)$  — максимальное и минимальное отклонение расположения по осям (разница в расположении верхних левых углов по оси  $X$  при горизонтальном движении и оси  $Y$  при вертикальном движении) находится в диапазоне между числами  $R_1 = const$  и  $R_2 = const$ ;  $(C_5)$  — максимальная разность длин меньше  $W = const$  (поправка на небольшой поворот либо частичное выпадение вилок из кадра).

## Результаты экспериментальных исследований

Популярные текстурные алгоритмы (SURF, SIFT, ORB и FAST), а также предложенный морфологический алгоритм были реализованы в прототипе программного продукта обнаружения вилочного погрузчика на видео для проведения сравнительного эксперимента. Для реализации использовалась библиотека OpenCV 2.4 [13]. На вход программе подается набор видеоданных, который затем обрабатывается любыми из указанных алгоритмов по выбору пользователя, после чего результаты работы программы выводятся в файл или непосредственно на экран. Агрегация результатов в процессе работы алгоритмов производится отдельно для каждого видео. Итоговая оценка качества работы алгоритмов определяются из показателей FRR и FAR обнаружения пустого погрузчика. Детектирование проводится независимо для каждого кадра. При оценке верного обнаружения (true detection) объекта для текстурных алгоритмов берется порог количества кадров  $M_1$  по отношению к общему количеству кадров видео (алгоритм 1). Для морфологического алгоритма такой порог не используется — если объект был обнаружен хотя бы на одном кадре, он считается присутствующим на видео.

Пользовательский интерфейс разработанной программы позволяет выбирать различные типы фильтрации изображения и размер ядра накладываемого фильтра, задействовать или отключать любой из пяти заложенных алгоритмов при обработке видео, также есть возможность выбора любого количества видеоданных, как с указанием конкретных файлов, так и загрузкой папок.

Для проведения эксперимента было выбрано 24 видеофайла, с разрешением  $1280 \times 720$ , на каждом из которых имеется движущийся в определенном направлении вилочный погрузчик (см. рис. 1) (загружены из БД компании Intelligent Security Systems) [14]. На 10 видео погрузчик движется без груза, на 14 оставшихся — с грузом. Камера, с которой записывалось видео, расположена строго перпендикулярно к плоскости пола, где движется погрузчик. Съемка велась со стационарных камер видеонаблюдения в больших складских помещениях во время процессов загрузки/отгрузки товара в разное время суток. Этим обусловлено наличие таких помех, как резкое изменение освещения и затенения, частичное выпадение объекта из кадра, наличие множества иных движущихся в разные стороны объектов. Продолжительность видео — от 15 до 25 с, средняя частота кадров в секунду — 25. Погрузчик (с грузом или без него) является самым крупным движущимся объектом на видео.

Эксперимент был разбит на три части. В первой было оценено качество работы алгоритма, основанного на сравнении локальных дескрипторов ключевых точек. Было использовано четыре метода вычисления и сравнения дескрипторов (SURF, SIFT, ORB, FAST). FAR определяется как  $N_{\text{fork\_miss}}/N_{\text{fork}}$ , а FRR как  $N_{\text{false\_det}}/N_{\text{no\_fork}}$ , где  $N_{\text{fork\_miss}}$  — количество видео, содержащих вилы с грузом, ошибочно распознанных как пустые,  $N_{\text{fork}}$  — количество видео с грузами,  $N_{\text{false\_det}}$  — число видео, на которых пустые вилы ошибочно не были обнаружены,  $N_{\text{no\_fork}}$  — количество видео с пустыми вилами. В эксперименте использовались различные значения параметров для каждого текстурного метода. Разрешение исходных видео данных:  $1280 \times 720$  и  $800 \times 465$  (искусственно уменьшено с исходного), разрешение изображения искомого объекта —  $303 \times 205$  (вырезано из исходных данных).

В результате проведения первой части эксперимента были определены оптимальные значения параметров первого алгоритма, при которых алгоритмы показывают наименьшее количество FAR и FRR обнаружения пустых вилок. Порог совпадений при сравнении дескрипторов  $M_0$ : SURF — 25, SIFT — 19, ORB — 40, FAST — 30. Рассмотренные значения порогов для SURF {16, 20, 25}, для SIFT {16, 20, 25}, ORB {16, 20, 25}, FAST {16, 20,

Таблица 1. Результаты работы алгоритма локальных дескрипторов

Значения	SURF	SIFT	ORB	FAST
Порог совпадений дескрипторов $M_0$	25	19	40	30
Порог кадров с обнаруженным объектом $M_1$ , %	1	2	20	5
FRR, %	0	65	87	75
FAR, %	64	53	0	13
Время обработки кадра, мс	75,1	188,8	35,5	44,5

25}. Оптимальное значение порога  $M_1$ : SURF — 1%, SIFT — 2%, ORB — 2%, FAST — 5%. Рассмотренные значения порога: SURF {0.5%, 1%, 2%}, SIFT {1%, 2%, 4%}, ORB {10%, 20%, 25%}, FAST {1%, 5%, 7%}. При оптимальных значениях параметров были получены результаты, представленные в табл. 1.

Как видно, значения FRR и FAR велики даже при оптимальном подборе значения параметров. Это в первую очередь связано с тем, что пустые вилы на выбранных видео практически не выделяются из фона (из-за схожей цветовой насыщенности). Очевидно, низкое качество обнаружения и вычислительная эффективность традиционного подхода не являются удовлетворительными для его практического применения.

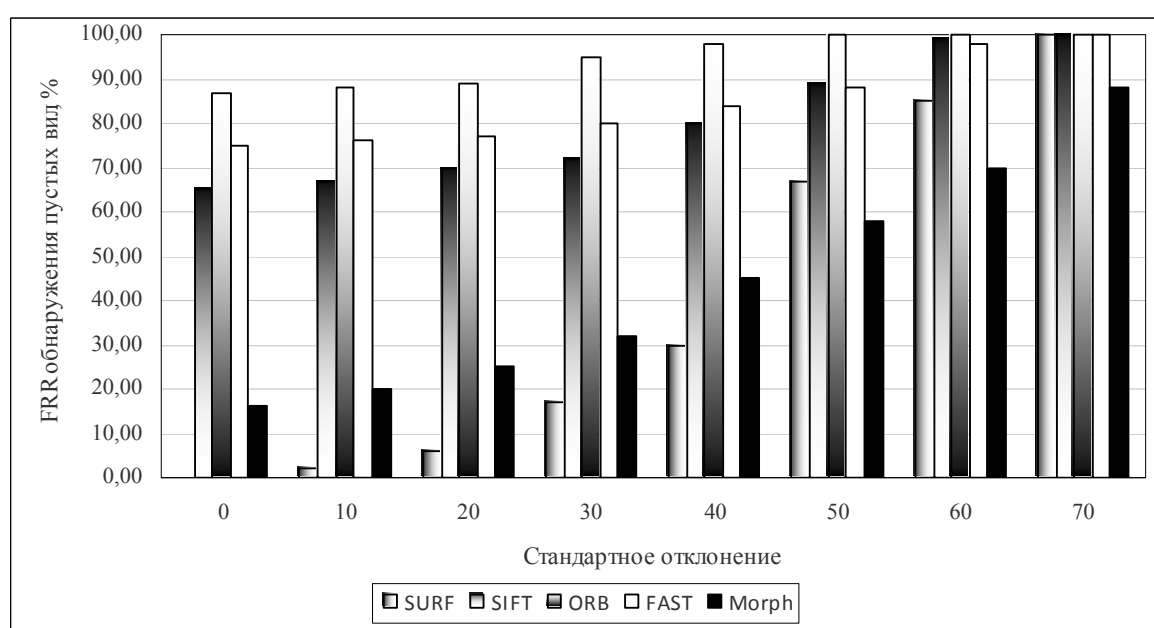
Во второй части эксперимента протестирована реализация морфологического алгоритма на качество обнаружения. Была выявлена зависимость точности алгоритма от условий отбора ограничивающих прямоугольников для контуров вилок. Были рассмотрены следующие значения параметров: тип фильтрации предварительной обработки — медианная, гауссова, оконный фильтр и нормализованный оконный фильтр [9, 15]; размеры ядра фильтра — от 1 до 9; минимальный размер движущейся области кадра — 0,3 и 0,5. Экспериментально были определены значения параметров, при которых алгоритм показывает наибольшую точность обнаружения при неизменных критериях отбора прямоугольников: медианная фильтрация в окрестности  $5 \times 5$ , при размере движущейся области в 0,5 кадра. Наилучшее качество детектирования было получено для следующих значений параметров критериев отбора прямоугольников:  $H_1 = 150$  (длина),  $W_1 = 35$  (ширина),  $D = 150$ ,  $A = 1500$ ,  $R_1 = 150$ ,  $R_2 = 80$ ,  $W = 20$ . В результате проведения эксперимента было выявлено, что все названные критерии, за исключением максимальной разности площадей прямоугольников, при добавлении их как условий в процесс отбора существенно понижают FRR поиска пустых вилок, лишь незначительно повышая значение FAR. Наименьшее FRR — 16% и 5% — FAR достигается при совместном использовании пяти критериев. Кроме того, разработанный алгоритм очень быстро обрабатывает каждый кадр — порядка 5 мс, что позволяет получать результаты в реальном времени. Зависимость морфологического алгоритма от критериев отбора ограничивающих прямоугольников контуров вилок представлена в табл. 2.

В третьей части эксперимента проводится тестирование реализованных алгоритмов на устойчивость к шумам различного типа. Всего рассматривалось три типа помех, являющихся типичными для условий поставленной задачи: аддитивный белый гауссов шум, случайный импульсный шум и частичное осветление/затемнение кадра. Так, импульсные помехи могут возникать от тиристорных регуляторов и ламп дневного света или из-за токов питания устройств, участвующих в обработке сигнала (синхронные с сетью помехи). Белый гауссов шум может возникать при плохих условиях приема сигнала с камеры. Пе-



**Таблица 2.** Зависимость точности морфологического алгоритма от критериев проверки

Критерии проверки	Минимальная длина и ширина	Разность площадей	Максимальное удаление друг друга	Максимальная ширина	Максимальная разность длин
FRR, %	0	0	7,5	11,8	16,3
FAR, %	78,9	78,9	34,1	15,5	5
Время обработки кадра, мс	4,8	4,9	4,9	5,1	5,2

**Рис. 5.** Зависимость значения FRR от дисперсии аддитивного белого шума

репады освещения типичны при движении по неравномерно освещенным помещениям или когда источники света расположены не на потолке, а на стенах и освещают помещение под разными углами. Для получения гауссова шума генерируется нормально-распределенная случайная величина с нулевым средним и переменной дисперсией (параметр, характеризующий уровень шума), прибавляющаяся к каждому пикселю каждого кадра, затем значение итогового пикселя устанавливается в пределах  $[0, \dots, 255]$ . Зависимость качества рассматриваемых алгоритмов (значения FRR и FAR) от дисперсии шума показана на рис. 5 и 6.

При использовании случайного импульсного шума (рис. 11), фиксируется один числовой параметр — количество модифицируемых пикселей изображения по отношению к общему числу пикселей одного кадра. Далее случайным образом выбирается координата модифицируемого пикселя и новое значение его яркости. Соответственно, чем больше количество изменяемых точек, тем сильнее помехи. Зависимость качества детектирования алгоритмов от этого параметра показана на рис. 7 (для значения FRR) и на рис. 8 (FAR).

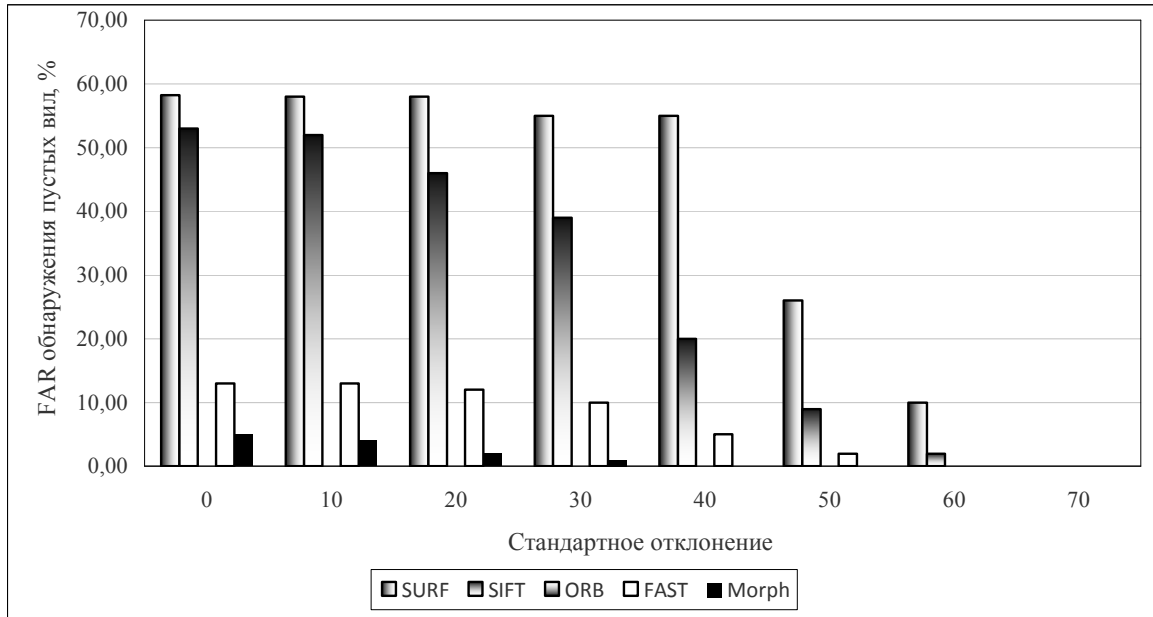


Рис. 6. Зависимость значения FAR от дисперсии аддитивного белого шума

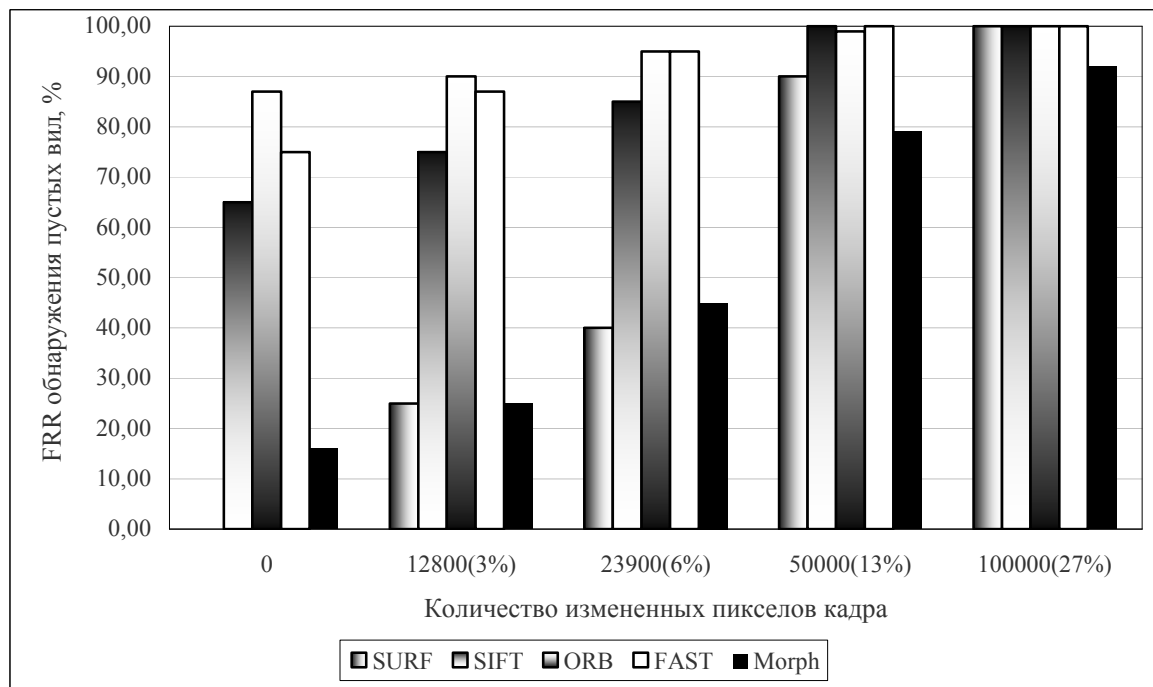


Рис. 7. Зависимость значения FRR количества пикселей с импульсным шумом

Для добавления перепадов освещения на видео использовалось динамическое увеличение яркости пикселей одной половины и ее понижение на другой половине кадра на фиксированную величину (интенсивность), которая и служит параметром этого шума. Динамика изменения значений FAR и FRR показана на рис. 9 (FRR) и на рис. 10 (FAR).

Результаты этого эксперимента подтверждают предположение о большей устойчивости к помехам предложенного алгоритма. Заметим, что при очень высоком уровне шума точность разработанного алгоритма (алгоритм 2) становится близкой к точности традицион-

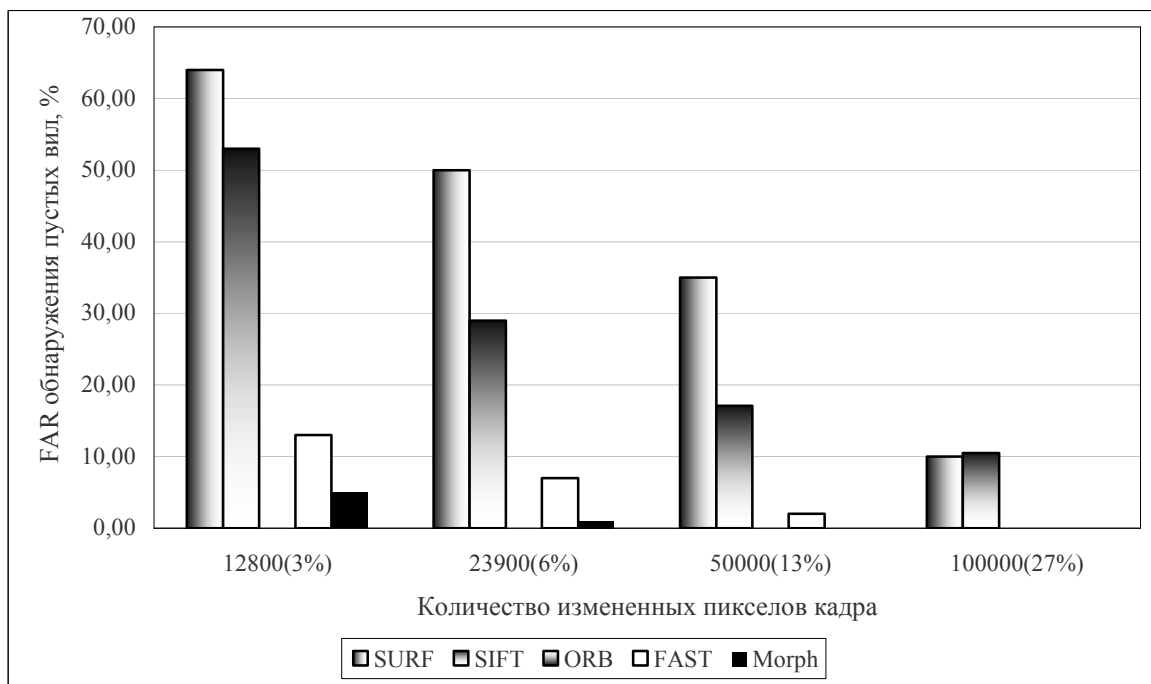


Рис. 8. Зависимость значения FAR от количества пикселей с импульсным шумом

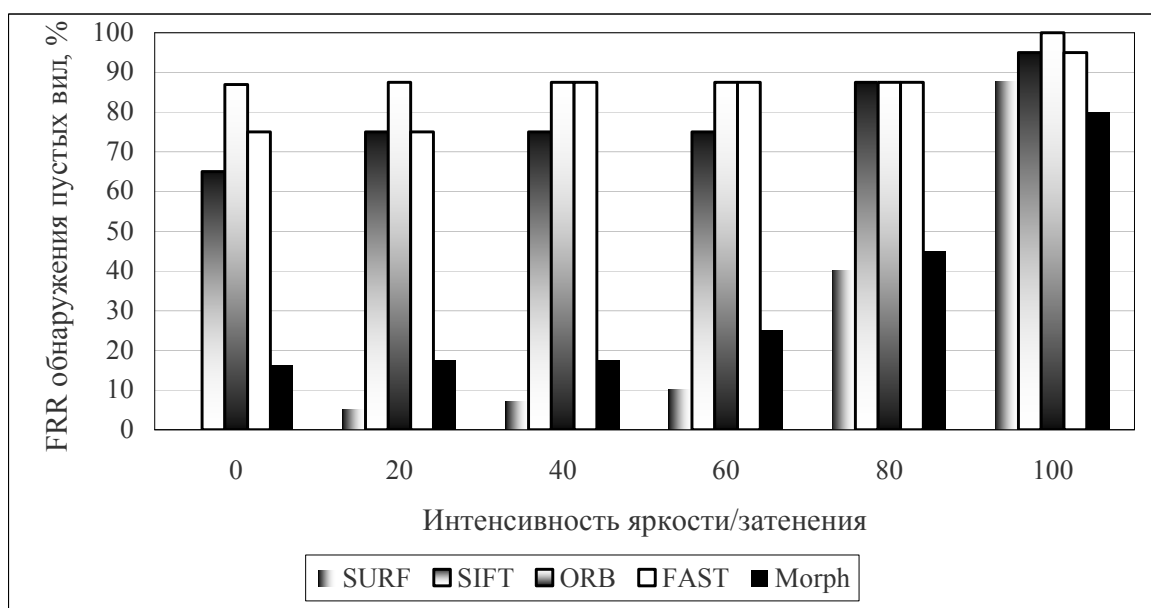


Рис. 9. Зависимость значения FRR от интенсивности яркости/затенения

ного подхода (алгоритм 1), что означает неудовлетворительное качество решения задачи при значительных помехах.

### Заключение

В результате проведения исследования был разработан морфологический алгоритм обнаружения движущегося объекта на видео, который обеспечивает на 40% и 27% (разница между усредненными значениями FRR и FAR  $(FAR + FRR/2)$  текстурных алгоритмов и морфологического алгоритма при обнаружении пустых вил) более точный резуль-

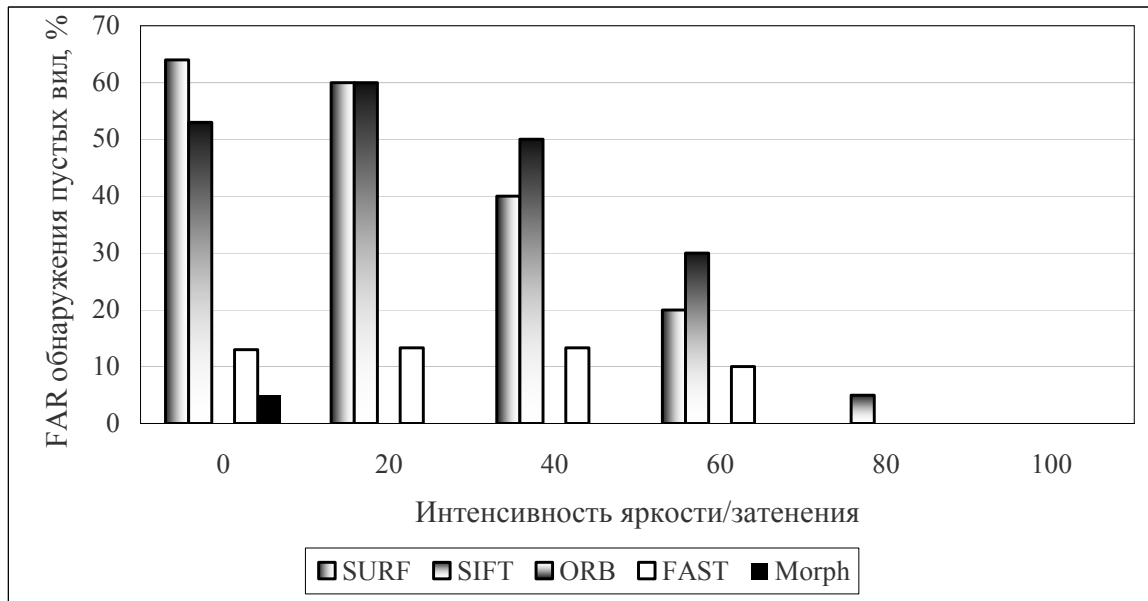


Рис. 10. Зависимость значения FAR от интенсивности яркости/затенения

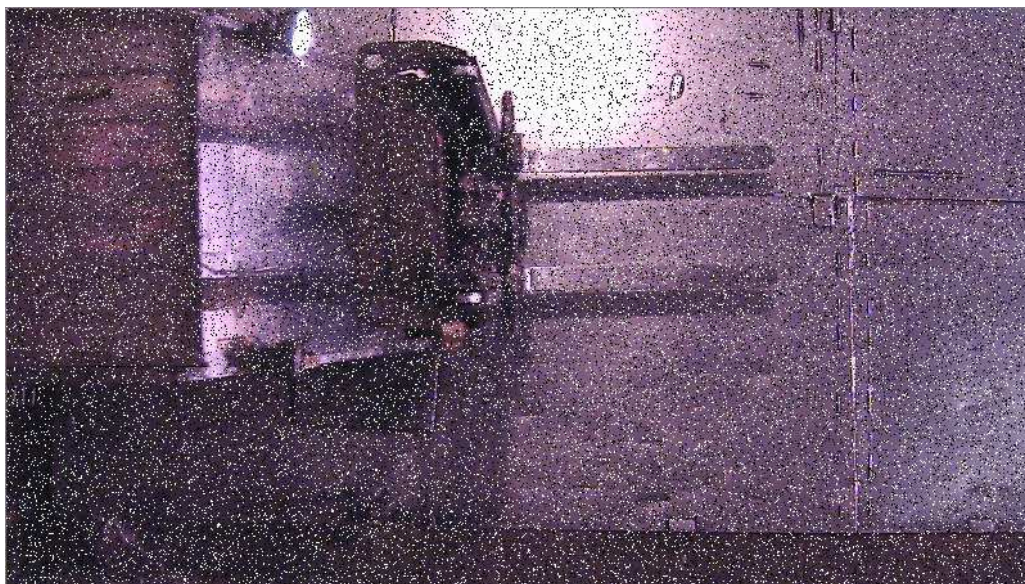


Рис. 11. Пример изображения погрузчика со случайным импульсным шумом

тат и в 7 раз (5 мс против 35 мс у ORB) более быструю обработку кадра [16], нежели традиционный метод, основанный на сравнении локальных дескрипторов ключевых точек. Экспериментально подтвержден факт сильного влияния помех вида игры света и тени, а также разного рода случайных помех на точность работы текстурных алгоритмов [1, 2, 3, 4] (рис. 5 и 7). Также было отмечено, что методы локальных дескрипторов неэффективны в плане скорости работы, когда изображение имеет много потенциальных ключевых точек и их приходится вычислять для каждого кадра. Оба недостатка были устранены в предложенном алгоритме, где игра света и тени обрабатывается оператором Кэнни [12], случайный шум удаляется фильтрацией и морфологическими преобразованиями. Производительность алгоритма повышена за счет использования только простых преобразований изображения.

Разработанный метод, однако, имеет существенный недостаток, так как во многом полагается на априорную информацию о специфике поставленной задачи (размеры движущегося объекта, наличие других объектов на видео и т. д.). Следовательно, его нельзя будет применить для обнаружения других типов объектов без внесения дополнительной логики работы в механизм распознавания. С точки зрения развития данной темы следующим шагом, на наш взгляд, может стать дальнейшее повышение точности алгоритма (с использованием комбинирования рассмотренных методов), а также его адаптация для других практически важных задач распознавания движущихся объектов в зашумленной среде.

## Литература

- [1] Lowe D. Distinctive image features from scale-invariant keypoints // *Int. J. Comput. Vision*, 2014. Vol. 60, no. 2. P. 91–110.
- [2] Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded up robust features // *Comput. Vision Image Understanding*, 2008. Vol. 110, no. 3. P. 346–359.
- [3] Rublee E., Rabaud V., Konolige K., Bradski G. ORB: An efficient alternative to SIFT or SURF // *IEEE Conference (International) on Computer Vision (ICCV)*, 2011.
- [4] Rosten E., Drummond T. Machine learning for high-speed corner detection // *European Conference on Computer Vision*, 2006. Vol. 1. P. 430–443.
- [5] Savchenko A. V. Probabilistic neural network with homogeneity testing in recognition of discrete patterns set // *Neural Networks*, 2013. Vol. 46. P. 227–241.
- [6] Savchenko A. V. Adaptive video image recognition system using a committee machine, optical memory and neural networks // *Optical Memory Neural Networks (Information Optics)*, 2012. Vol. 21, no. 4. P. 219–226.
- [7] Chien S.-Y., Ma S.-Y., Chen L.-G. Efficient moving object segmentation algorithm using background registration technique // *IEEE Trans. Circuits Syst. Video Technol.*, 2002. Vol. 12, no. 7. P. 577–586.
- [8] Ahad Md., Atiqur Rahman, Tan J. K., Kim H., Ishikawa S. Motion history image: its variants and applications // *Machine Vision Appl.*, 2012. Vol. 23, no. 2. P. 255–281.
- [9] Shapiro L. G., Stockman G. C. Computer vision. Prentice Hall, 2001.
- [10] Mathematical morphology: From theory to applications. / Eds. L. Najman, H. Talbot. Wiley-ISTE, 2010.
- [11] ISS (Intelligence Secure Systems). URL: <http://www.iss.ru/>
- [12] Canny J. A. Computational approach to edge detection // *IEEE Computer Society*, 1986. P. 679–698.
- [13] OpenCV library. URL: <http://opencv.willowgarage.com/wiki/>
- [14] ISS video dataset. URL: <ftp://isstemp:isstemp@ftpsupport.iss.ru/Loaders/Video/>
- [15] Sonka M., Hlavac V., Boyle R. Image processing, analysis, and machine vision. 4th ed. Cengage Learning, 2014.
- [16] Savchenko A. V. Directed enumeration method in image recognition // *Pattern Recognition*, 2012. Vol. 45, no. 8. P. 2952–2961.